

## SISTEM INFORMASI GEOGRAFIS PENENTUAN JARAK BTS TERDEKAT PADA KEGIATAN SURVEY PELANGGAN DENGAN METODE HAVERSINE BERBASIS WEB (STUDI KASUS: PADINET SURABAYA)

Muhammad Pearson Ibnu S.A, Joseph Dedy Irawan, Ahmad Faisol

Teknik Informatika, Institut Teknologi Nasional Malang

Jalan Raya Karanglo km 2 Malang, Indonesia

1918128@scholar.itn.ac.id

### ABSTRAK

Perkembangan teknologi informasi pada era digital semakin berkembang pesat, Internet digunakan diberbagai bidang, termasuk dalam industri. Untuk mengakses Internet diperlukan *Internet Service Provider* atau jaringan seluler. *Internet Service Provider* merupakan lembaga yang memberikan pelayanan kepada konsumen agar bisa mengakses Internet secara berlangganan. *Internet Service Provider* (ISP) menyediakan berbagai layanan, layanan tersebut bisa disesuaikan oleh kebutuhan calon pelanggan, kebutuhan ini bisa dalam bentuk layanan *Internet*, *Voice Over Internet Protocol* (VoIP), *Internet Protocol Camera* (IP Camera), dan *Virtual Private Network* (VPN). Dalam melaksanakan survey calon pelanggan, *Technical Support* memerlukan posisi BTS terdekat yang terlihat secara jelas dan tanpa halangan. atau bisa disebut *Line of Sight* (LoS). *Technical Support* terkadang mengalami kesulitan untuk mengetahui BTS mana yang berada pada jangkauan mereka. Sistem Informasi Geografis (SIG) adalah media yang bisa digunakan sebagai penentuan BTS terdekat dan mencantumkan informasi yang berkaitan dengan BTS tersebut. Sistem Informasi ini mengimplementasikan metode haversine untuk mendapatkan jarak antara pengguna dan BTS terdekat. Oleh karena itu, dipilihnya Sistem Informasi Geografis dan *Haversine Formula*. berdasarkan data yang diperoleh merupakan data koordinat dari setiap bts dan Sistem Informasi yang cocok adalah sistem informasi geografis, untuk formula haversine dipilih dikarenakan jarak untuk bts ke pelanggan merupakan jarak udara yang mana gelombang *wireless* hanya memanfaatkan garis lurus. Dari hasil pengujian sistem yang dilakukan pada *desktop* maupun *mobile* memperoleh *error* dari keakuratan jarak dibandingkan google *maps* rata-rata sebesar 0.29% dari data 9 BTS dengan 2 lokasi *user* yang berbeda.

**Kata kunci :** Sistem, Informasi Geografi, Haversine Formula, BTS, NuxtJS

### 1. PENDAHULUAN

Perkembangan teknologi informasi pada era digital semakin berkembang pesat, *Internet* digunakan diberbagai bidang, termasuk dalam industri. Untuk mengakses *Internet* diperlukan *Internet Service Provider* atau jaringan seluler. *Internet Service Provider* merupakan lembaga yang memberikan pelayanan kepada konsumen agar bisa mengakses *Internet* secara berlangganan. *Internet Service Provider* (ISP) [14] menyediakan berbagai layanan, layanan tersebut bisa disesuaikan oleh kebutuhan calon pelanggan, kebutuhan ini bisa dalam bentuk layanan *Internet*, *Voice Over Internet Protocol* (VoIP), *Internet Protocol Camera* (IP Camera), dan *Virtual Private Network* (VPN). Hal ini, memerlukan pemasangan perangkat radio untuk mendapatkan jaringan yang disediakan oleh *Internet Service Provider* (ISP). Perangkat radio ini digunakan untuk, mendapatkan jaringan dari *Base Transceiver Station* (BTS). BTS ini telah dihubungkan dengan jaringan yang mulai dari *Network Operation Center* (NOC) tetapi, juga dihubungkan ke BTS lain sebagai Backup jika, salah satu dari BTS mengalami gangguan bisa dialihkan ke BTS lain baik secara *wireless* maupun dengan kabel fiber optik.

Dalam melaksanakan survey calon pelanggan, *Technical Support* memerlukan posisi BTS terdekat yang terlihat secara jelas dan tanpa halangan. atau bisa

disebut *Line of Sight* (LoS). *Technical Support* terkadang mengalami kesulitan untuk mengetahui BTS mana yang berada pada jangkauan mereka. Oleh karena itu, dengan adanya Sistem Informasi Geografis (SIG) dalam Penentuan jarak terdekat BTS ke Calon pelanggan tadi, dapat membantu *Technical Support* untuk memilih BTS mana saja yang nantinya akan digunakan dalam Instalasi jaringan Internet pada pelanggan agar pelanggan mendapatkan layanan paling optimal setelah pemasangan tadi. .

Sistem Informasi Geografis (SIG) adalah media yang bisa digunakan sebagai penentuan BTS terdekat dan mencantumkan informasi yang berkaitan dengan BTS tersebut. Dalam pengembangannya dapat diintegrasikan data lokasi BTS dan calon pelanggan (Lokasi pengguna). Sehingga, mendapatkan jarak antara kedua lokasi dan menampilkan data BTS terdekat. Dengan menggunakan teknologi *Location-Based Service* (LBS) untuk layanan sistem informasi dan metode *Haversine Formula* untuk menentukan jarak antara 2 titik, diharapkan calon pelanggan menerima layanan yang terbaik. Apabila, jarak BTS dengan lokasi pelanggan mempunyai jarak yang optimal dan memiliki pandangan terlihat dari lokasi pelanggan.

Pada penelitian terdahulu yang dilaksanakan pada tahun 2018 oleh Yulianto dengan judul "Penerapan Formula Haversine Pada Sistem Informasi

Geografis Pencarian Jarak Terdekat Lokasi Lapangan Futsal” pada penelitian tersebut memiliki perbedaan yaitu, pada sistem tersebut menampilkan rute terdekat dari lokasi user menuju futsal, tetapi untuk sistem yang akan dibuat hanya menampilkan garis lurus dari posisi user menuju bts.

## 2. TINJAUAN PUSTAKA

### 2.1. Website

*Website* adalah himpunan *web page* yang mana terkoneksi dengan halaman *web* yang lainnya, tetapi halaman-halaman yang tidak terhubung dengan halaman yang lain bisa disebut halaman mandiri adalah *web page*, dengan kata lain *website* adalah situs yang bisa diakses oleh seorang yang memiliki akses *Internet* [1]. *Website* sendiri tidak hanya digunakan untuk bertukar informasi tetapi, juga dapat untuk membantu kegiatan sehari-hari. *Website* adalah himpunan dari situs yang memiliki halaman yang telah dimasukkan dalam *domain* maupun *subdomain*. Sering kali *website* terdiri atas HTML (*Hyper Text Markup Language*) [2].

### 2.2. OpenStreetMap

OpenStreetMap adalah merupakan salah satu layanan pemetaan dunia yang open source, yang dibangun oleh komunitas *mapper* yang berkontribusi dalam pengelolaan data jalan, rel kereta kafe, stasiun dan lainnya. OpenStreetMap mengedepankan pengetahuan lokal tentang daerah sekitarnya. [3]. OpenStreetMap adalah proyek yang digunakan untuk membuat *database* peta dunia secara gratis. Tujuannya untuk memiliki data setiap fitur geografis yang ada di dunia. Selain, memetakan jalan, OpenStreetMap juga memetakan jalur pejalan kaki, bangunan dan bahkan pohon secara satu persatu.

*Database* dari OSM sendiri diolah oleh *mappers*, yang mengumpulkan informasi dengan mengemudi, bersepeda dan berjalan sepanjang jalan, jalur dan area sekitar berdasar pergerakan mereka yang menggunakan GPS (*Global Positioning System*). Informasi ini kemudian, digunakan membuat kumpulan garis dan titik yang diubah menjadi peta atau digunakan sebagai navigasi [4].

### 2.3. GPS

Sistem lokasi global (GPS) menggunakan teknologi satelit untuk memberi tahu pengguna dimana mereka berada di permukaan bumi secara *global*. Data *digital* diperoleh dari satelit melalui sinyal radio. Di mana Tempat pengguna berada, GPS dapat membantu menunjukkan arah dan layanan GPS ini gratis [5]

Peningkatan penggunaan perangkat yang menggunakan GPS membuka peluang pengguna, menggunakan informasi geografi mereka untuk diolah menjadi yang bias digunakan oleh *Location Based Service* (LBS). Data dari pengguna ini biasanya digunakan oleh OpenStreetMap yang merupakan

pemetaan aplikasi dunia yang memanfaatkan GPS dari *user* untuk memetakan tempat yang ada di dunia [6].

### 2.4. Sistem Informasi Geografis

Sistem Informasi Geografis adalah sistem yang mempunyai tujuan untuk menyimpan data pada *database*, menganalisa, merekayasa, mengatur, dan mempresentasikan jenis yang lengkap data lokasi [7]. Sistem Informasi Geografis sendiri memiliki kapabilitas menghubungkan titik tertentu dibelahan bumi menganalisa, mengabungkannya dan mempresentasikan hasilnya. Data yang dikelola oleh Sistem Informasi Geografis ialah data yang berorientasi geografi dan ialah koordinat lokasi tertentu sebagai acuannya (data spasial). Oleh karena itu, aplikasi Sistem Informasi Geografis dapat mempermudah pengguna dalam melakukan kegiatan yang berorientasi geografis [8].

### 2.5. Firebase

Firebase merupakan Layanan yang dikenalkan oleh Google, Firebase, memudahkan pengembang perangkat lunak untuk membuat aplikasi, terutama aplikasi *mobile*. Pada tahun 2011, Andrew Lee dan James Tamplin mendirikan Firebase dengan nama perusahaan Encolce. Salah satu produk pertama mereka adalah Firebase, yang memiliki *database real-time* [9]. Firebase API dari Google memungkinkan Anda membuat basis data aplikasi secara *realtime*. Data yang tersimpan di firebase akan disimpan dalam *format JSON* dan dapat diakses di semua *platform*, dan basis data ini dapat berjalan secara *realtime* berkat data *synchronization*. Oleh karena itu, jika terjadi perubahan data, semua perangkat yang terhubung akan segera menerima perubahan tersebut [10].

### 2.6. Metode Haversine

Metode haversine, persamaan navigasi penting, menunjukkan jarak lingkaran yang signifikan antara dua titik pada permukaan bola (bumi) berdasarkan bujur dan lintang. Rumus ini cukup akurat untuk sebagian besar perhitungan, dan juga mengabaikan ketinggian bukit dan lembah di permukaan bumi [11]. Rumus ini ditemukan oleh James Andrew pada tahun 1805, dan Josefe Mendoza y Ríos menggunakannya pada tahun 1801. Pada tahun 1835, Prof. James Inman memperkenalkan istilah haversine untuk pertama kali. Dalam penelitian Masalah Utama Astronomi Laut, Josef de Mendoza y Ríos juga menggunakan rumus *Haversine* untuk mengukur jarak antar bintang [12].

Dalam Implementasinya ini memungkinkan penggunaan garis lintang dan bujur pada Google Maps. Perhitungan yang dihasilkan dari penggunaan ini adalah jarak dari titik yang menghasilkan gambar dalam peta *Application Programming Interface* (API) Google Maps [13].

Rumus dapat dituliskan dengan persamaan sebagai berikut :

$$x = (\text{lon}2 - \text{lon}1) \frac{\cos((\text{lat}1 + \text{lat}2))}{2} \quad (1)$$

$$y = \text{lat}2 - \text{lat}1 \quad (2)$$

$$d = R \cdot \sqrt{x^2 + y^2} \quad (3)$$

Keterangan :

R = jari-jari bumi sebesar 6371 (km)

lon1= Longitude atau garis lintang tempat pertama

lat1= Latitude atau garis bujur tempat pertama

lon2= Longitude atau garis lintang tempat kedua

lat2= Latitude atau garis bujur tempat kedua

x = Longitude (Lintang)

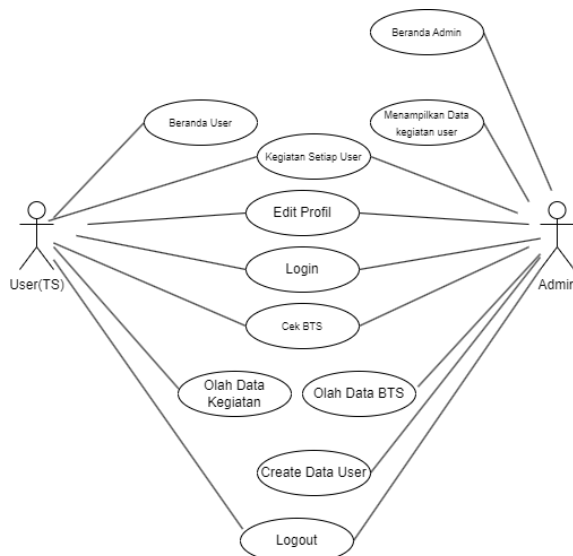
y = Latitude (Bujur)

d = jarak (km)  $1^\circ = 0.0174532925$  radian [13]

### 3. METODE PENELITIAN

#### 3.1. Use Case Diagram

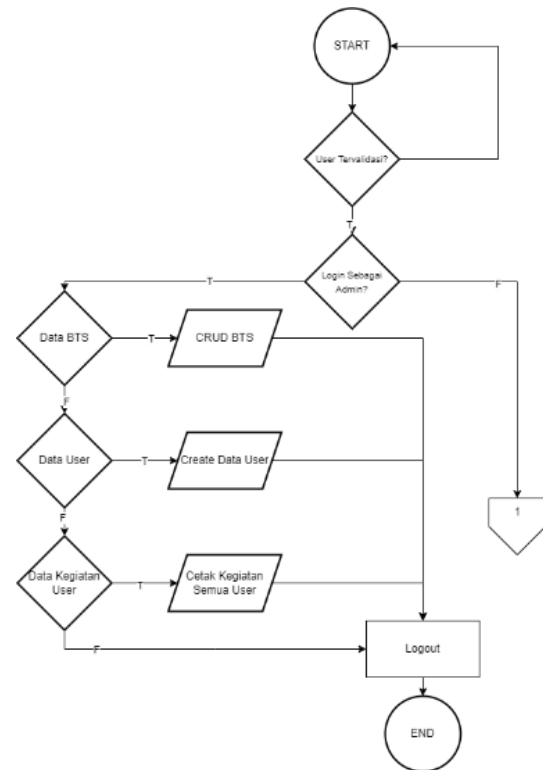
Pada gambar 1 menggambarkan bagaimana *actor* atau pengguna menggunakan sistem yang dibuat. Terdapat 2 hak akses yaitu Admin dan User. Setiap Hak Akses memiliki fitur yang berbeda, pada hak akses admin bisa mengolah data *user*, data *bts* dan melihat kegiatan setiap *user*. Pada hak akses *user* terdapat fitur cek BTS, olah data kegiatan dan cetak kegiatan *user* berdasarkan *Login*.



Gambar 1. Use case diagram

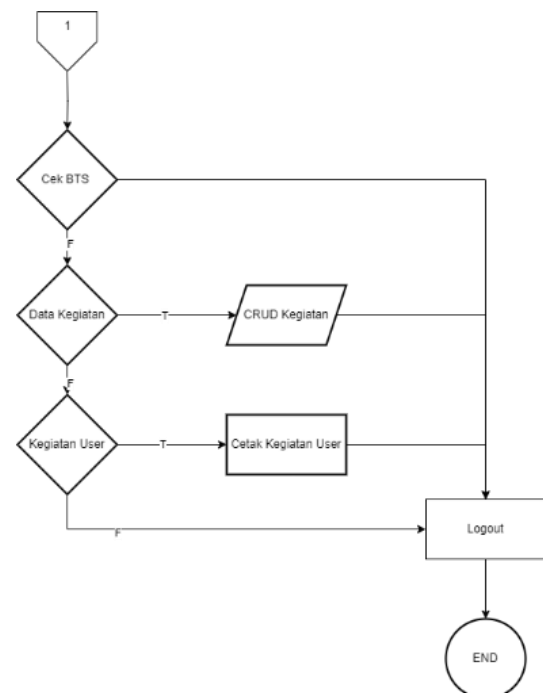
#### 3.2. Flowchart Sistem

Pada gambar 2 diperlihatkan bahwa diperlukan validasi untuk masuk kedalam sistem apabila *User* tervalidasi maka, akan dicek kembali untuk rolenya apabila *user* merupakan admin maka, akan diarahkan ke Halaman *admin* yang berisi Data BTS. Data BTS ini apabila dipilih akan menampilkan CRUD BTS. Apabila Data BTS tidak dipilih, maka lanjut pada menu *create User*. Apabila Data *User* tidak dipilih maka, akan diarahkan pada data kegiatan *user*. Apabila semua hal tadi telah dipilih *User* bisa memilih *logout* dan program berakhir.



Gambar 2. Flowchart admin

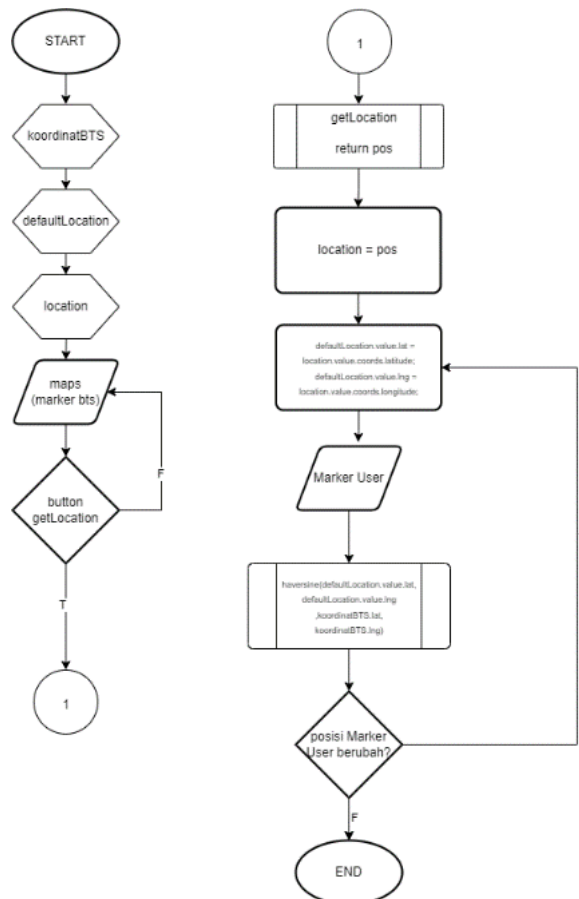
Pada Gambar 3 apabila *user* yang tervalidasi merupakan *user* biasa. Maka akan, diarahkan pada halaman *user* yang berisi Cek BTS, Data Kegiatan, Kegiatan *User*. Apabila *user* tidak memilih Cek BTS maka, akan diarahkan pada Data kegiatan. Apabila, *user* tidak memilih Data kegiatan maka, akan diarahkan pada Cetak Kegiatan *User*. Semua Pilihan tadi terhubung dengan *Logout* apabila *user* ingin mengakhiri program.



Gambar 3. Flowchart User

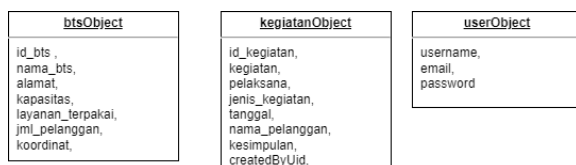
### 3.3. Flowchart Haversine

Pada Gambar 4 apabila *user* yang berada pada halaman *maps* pertama kali akan muncul peta yang berisi dengan BTS yang telah dipetakan sesuai koordinat yang ada. Setelah *button locateMe* di klik maka, *marker* pengguna akan muncul dengan koordinat yang didapatkan dari *Web API* dan memunculkan *list* jarak BTS dengan metode haversine. Apabila *marker* pengguna digeser maka, jalankan lagi metode haversine dengan lokasi pengguna yang terbaru.



Gambar 4. Flowchart Haversine

### 3.4. Desain Database



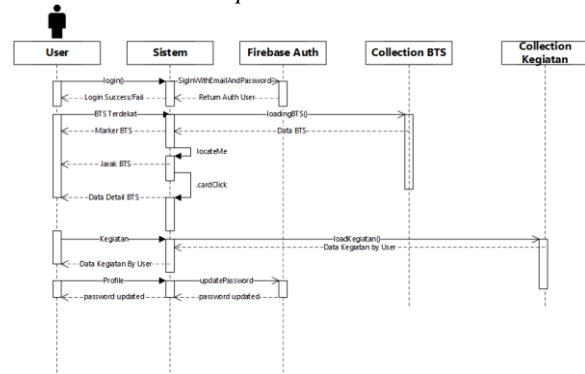
Gambar 5. Desain Database

Pada gambar 5 terdapat 3 object yaitu *btsObject*, *KegiatanObject*, *user Object*. Setiap object tadi, digunakan untuk operasi CRUD pada aplikasi. Untuk *btsObject* terdiri atas *id\_bts*, *nama\_bts*, *alamat*, *kapasitas*, *layanan\_terpakai*, *jml\_pelanggan*, dan *koordinat*. Untuk *kegiatanObject* terdiri atas *id\_kegiatan*, *kegiatan*, *pelaksana*, *jenis\_kegiatan*, *tanggal*, *nama\_pelanggan*, *kesimpulan*,

*createByUid*. Untuk *userObject* terdiri atas *username*, *email* and *password*. Data objek tadi akan diolah oleh sistem dan diinputkan pada *firebase* pada setiap *collection* yang telah dibuat.

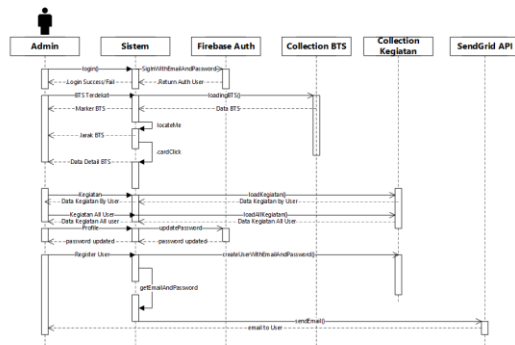
### 3.5. Sequence Diagram

Pada gambar 6 merupakan *sequence diagram* untuk *user* yang kebanyakan interaksinya dilalui Sistem terlebih sebagai jalan untuk akses informasi dari *Database* yang kebanyakan interaksi yang dilakukan dengan database dilakukan *asynchronous*. Sistem akan mengirimkan request pada interaksi tertentu akan mengembalikan data berubah sukses atau gagal setiap permintaan *user*. Sebagai contoh untuk login. *User* pertama kali melakukan login dengan memasukkan kredensial yang dia punya kedalam sistem. Ketika mengklik tombol login maka sistem akan mengirimkan data yang telah diinputkan oleh *user* untuk dibandingkan dengan *firebase auth* apabila, *user* terdaftar maka dia akan dialihkan ke halaman dashboard. Apabila tidak, terdaftar akan muncul notifikasi *email* atau *password* salah.



Gambar 6. Sequence Diagram User

Pada gambar 7 merupakan *sequence diagram* untuk admin yang kebanyakan interaksinya dilalui Sistem terlebih sebagai jalan untuk akses informasi dari *Database* yang kebanyakan interaksi yang dilakukan dengan database dilakukan *asynchronous*. Secara keseluruhan *sequence diagram* *user* dan admin tidak berbeda jauh hanya saja ada beberapa fitur yang membedakan seperti *register user* dan kegiatan *all user*. Untuk *Register user* terdapat perbedaan pada diagramnya. Dapat dilihat bahwa ketika, sistem melakukan *request* ke *firebase* untuk *register user*. Sistem masih melakukan *request* ke eksternal aplikasi yaitu API untuk melakukan pengiriman email dan password untuk masuk kedalam sistem yang mendapatkan data inputan dari *form register* sebagai *output* pada email yang akan dikirim.



Gambar 7. *Sequence Diagram Admin*

#### 4. HASIL DAN PEMBAHASAN

#### 4.1. Halaman Login

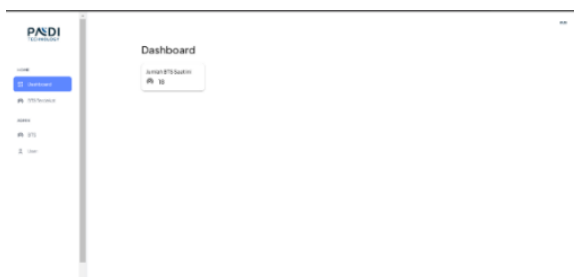
Pada Gambar 8 dibawah ini merupakan tampilan awal dari aplikasi Ketika pertama dibuka pada halaman login ini akan muncul. Halaman login ini digunakan untuk mengakses aplikasi tersebut dengan memasukka email dan password yang telah ditambahkan.



Gambar 8. Halaman *Login*

## 4.2. Halaman Dashboard

Pada Gambar 9 dibawah ini merupakan tampilan awal dari aplikasi Setelah login, pengguna akan diarahkan pada halaman dashboard ini berisi tentang informasi Jumlah BTS yang ada di Surabaya.



Gambar 9. Halaman *Login*

### 4.3. Halaman Dashboard

Pada Gambar 10 dibawah ini merupakan tampilan dari peta yang berisi BTS yang ada di Surabaya yang dapat diakses oleh *user* dan admin dan dapat digunakan untuk mencari jarak terdekat dari BTS.



Gambar 10. Halaman *Maps*

#### 4.4. Halaman BTS Terdekat (LBS Aktif)

Pada Gambar 11 dibawah ini merupakan tampilan dari peta yang berisi BTS yang ada dan marker dari lokasi pengguna apabila. Fitur LBS diaktifkan maka, akan muncul *marker* dari pengguna yang berwarna merah.



Gambar 11. Halaman *Maps*

#### 4.5. Halaman BTS Terdekat (Rute Aktif)

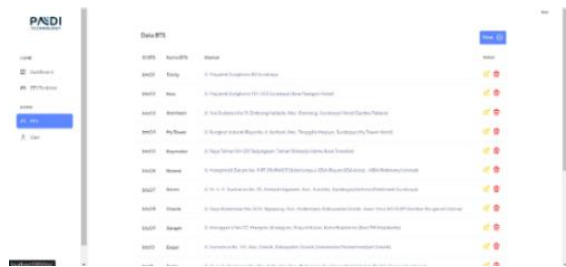
Pada Gambar 12 dibawah ini merupakan tampilan dari peta yang berisi BTS dan marker pengguna. Apabila list dari BTS tadi diklik maka, akan muncul garis lurus yang menghubungkan antara marker pengguna dan BTS yang diklik.



Gambar 12. Halaman BTS terdekat (Rute Aktif)

#### 4.6. Halaman BTS

Pada Gambar 15 dibawah ini merupakan tampilan dari halaman data BTS yang ada di Surabaya. Halaman ini bisa digunakan untuk CRUD untuk data BTS yang hanya dapat diakses oleh *admin*.



Gambar 13. Halaman BTS

#### 4.7. Halaman Tambah Data BTS

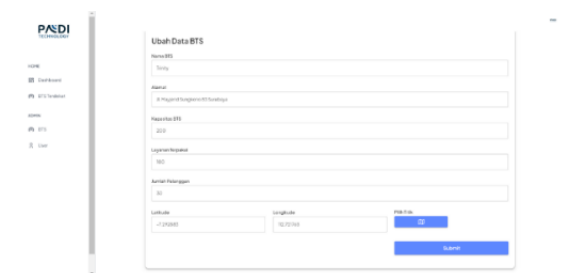
Pada Gambar 14 dibawah ini merupakan tampilan dari halaman yang digunakan untuk menambahkan data BTS. Untuk menambahkan koordinat bisa menggunakan *marker* atau diisi langsung pada kolom *latitude* dan *longitude*.



Gambar 14. Halaman Tambah Data BTS

#### 4.8. Halaman Ubah Data BTS

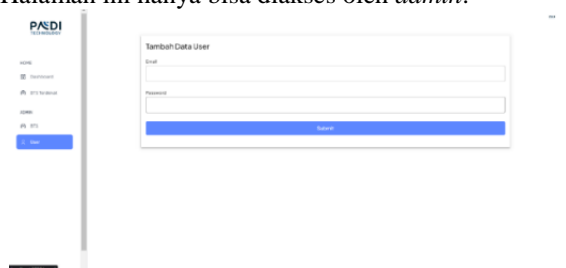
Pada Gambar 15 dibawah ini merupakan tampilan dari halaman yang digunakan untuk mengubah data BTS. Untuk mengubah koordinat bisa menggunakan *marker* atau diisi langsung pada kolom *latitude* dan *longitude*.



Gambar 15. Halaman Ubah Data BTS

#### 4.9. Halaman Tambah Data User

Pada Gambar 16 dibawah ini merupakan tampilan dari halaman yang digunakan untuk menambahkan data *user* untuk mengakses aplikasi ini. Halaman ini hanya bisa diakses oleh *admin*.



Gambar 16. Halaman Tambah Data User

#### 4.10. Halaman Forgot Password

Pada Gambar 17 dibawah ini merupakan tampilan dari halaman yang digunakan untuk melakukan aksi *forgot password*. Apabila *email* dan diklik buttonnya maka, akan ada *email* diterima untuk mengganti *password*.

Gambar 17. Halaman *Forgot Password*

#### 4.11. Halaman Kegiatan

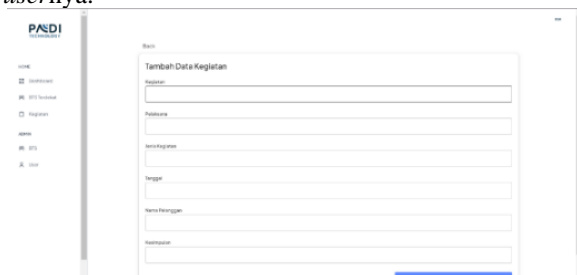
Pada Gambar 18 dibawah ini merupakan tampilan dari halaman kegiatan yang menampilkan data kegiatan untuk setiap *user*-nya. *User* Bisa menambah kegiatan mengubah dan menghapus data kegiatan.



Gambar 18 Halaman Kegiatan

#### 4.12. Halaman Tambah Data Kegiatan

Pada Gambar 19 dibawah ini merupakan tampilan dari halaman tambah data kegiatan yang digunakan untuk menambahkan data bagi setiap *user*-nya.



Gambar 19. Halaman Tambah Data Kegiatan

#### 4.13. Halaman Tambah Data Kegiatan



Gambar 20. Halaman Ubah Data Kegiatan

Pada Gambar 20 dibawah ini merupakan tampilan dari halaman mengubah data kegiatan yang ada pada setiap *user*nya.

#### 4.14. Pengujian keakuratan

Pada Tabel 1 dan Tabel 2 merupakan tahapan pengujian yang dilakukan untuk memvalidasi apakah keakuratan aplikasi yang telah dibuat sesuai. Sebagai contoh menggunakan aplikasi google maps dengan fitur *Measure Distancenya*.

Koordinat Pengguna saat menggunakan aplikasi pada *desktop*

*Latitude* : -7.298262

*Longitude*: 112.7219518

Tabel 1 Pengujian Keakuratan(*Dekstop*)

No	Nama BTS	Jarak(km)		Selisih	Persentase (%)
		Apps	GMaps		
1	Neo	0.727	0.718	0.009	1.25
2	Mercury	1.257	1.26	0.003	0.24
3	Architect	4.217	4.22	0.003	0.07
4	MyTower	5.421	5.42	0.01	0.18
5	Seraph	37.960	37.97	0.10	0.26

Koordinat Pengguna saat menggunakan aplikasi pada *mobile*

*Latitude* : -7.292793

*Longitude*: 112.721940

Tabel 2 Pengujian Keakuratan(*Mobile*)

No	Nama BTS	Jarak(km)		Selisih	Persentase (%)
		Apps	GMaps		
1	Dozer	18.759	19	0.241	1.27
2	Tank	7.221	7.3	0.079	1.08
3	Lexy	23.059	23	-0.059	-0.26
4	Master Steel	22.269	22	-0.269	-1.22
5	Option	8,190	8,2	0.01	0.12

Berdasarkan pengujian keakuratan pada tabel 1 dan tabel 2 dapat disimpulkan bahwa, pengujian yang dilakukan pada perangkat *mobile* atau *desktop* memiliki persentase error dari yang cukup banyak jika dibandingkan aplikasi Google Maps dengan fiturnya *Measure Distance*.

#### 4.15. Pengujian Metode

Pada Tabel 3 digunakan untuk memvalidasi metode yang telah diimplementasikan pada sistem dengan perhitungan yang menggunakan *excel*

Koordinat Pengguna saat menggunakan aplikasi pada *desktop*

*Latitude* : -7.2928209

*Longitude* : 112.7219239

Tabel 3 Pengujian Metode

No	Nama BTS	Jarak(km)		Selisih	Persentase (%)
		Apps	Excel		
1	My Tower	5.4269	5.4269	0	0
2	Twins	7.7137	7.7137	0	0
3	Keymaker	12.2786	12.2785	0.0001	0.0008
4	Oracle	17.7058	17.7058	0	0
5	Soren	6.6658	6.6657	0.0001	0.0015

Berdasarkan pengujian metode dapat disimpulkan bahwa, perhitungan metode dengan *Excel* memiliki selisih perhitungan yang minim. Hasil ini disebabkan oleh pembulatan angka yang selisih antara implementasi pada sistem dengan rumus dari *excel*.

#### 4.16. Pengujian BlackBox

Pada Tabel 4 merupakan tahapan pengujian yang dilakukan untuk memvalidasi apa hal apa saja yang sudah sesuai dengan fungsionalitas khususnya pada *input output* aplikasi.

Tabel 4 Pengujian Blackbox

Jenis Pengujian	Kondisi Pengujian	Hasil yang Diharapkan	Hasil yang Didapatkan	Keterangan	
				Sesuai	Tidak
Halaman Login	Jika diklik Sign in dengan <i>username</i> dan password kosong	Aplikasi menampilkan notif email atau password tidak boleh kosong	Aplikasi berhasil menampilkan notif email atau password tidak boleh kosong	✓	
	Jika diklik Sign in dengan <i>username</i> dan password yang salah	Aplikasi menampilkan Halaman Dashboard	Aplikasi menampilkan berhasil Halaman Dashboard	✓	
	Jika diklik Forgot Password muncul field input email	Aplikasi menampilkan field input email dan button submitnya	Aplikasi berhasil menampilkan field input email dan button submitnya	✓	
Halaman BTS terdekat	Tampilan BTS terdekat	Aplikasi menampilkan Halaman BTS Terdekat	Aplikasi berhasil menampilkan Halaman BTS Terdekat	✓	
	Jika diklik button getLocation	Aplikasi menampilkan marker dengan posisi saat ini	Aplikasi berhasil menampilkan marker dengan posisi saat ini	✓	
Halaman BTS	Jika di klik Tombol Edit(icon pensil)	Aplikasi menampilkan Halaman Ubah Data BTS	Aplikasi berhasil menampilkan Halaman Ubah Data BTS	✓	

Jenis Pengujian	Kondisi Pengujian	Hasil yang Diharapkan	Hasil yang Didapatkan	Keterangan	
				Sesuai	Tidak
	Jika di klik Tombol Hapus (Icon sampah)	Aplikasi menampilkan Popup untuk hapus data BTS	Aplikasi berhasil menampilkan Popup untuk hapus data BTS	✓	

Setelah dilakukan pengujian blackbox didapatkan hasil pengujian blackbox. Pada setiap *case* yang diberikan sistem dapat memenuhi setiap *test* diberikan baik pada perangkat *desktop* atau *mobile*. Untuk perangkat *desktop* pada semua tampilan bisa ditampilkan dengan baik. Untuk perangkat *mobile* ada beberapa tampilan yang terpotong.

## 5. KESIMPULAN DAN SARAN

Berdasarkan implementasi dan pengujian disimpulkan bahwa aplikasi dapat mengatasi *error* input yang dilakukan pengguna, kemudian untuk menampilkan jarak bts terdekat sistem berjalan dengan baik jika, lokasi *user* didapatkan dengan benar. Apabila, lokasi *user* belum tepat bisa digeser untuk mendapatkan lokasi saat ini.

Sedangkan, untuk fitur *location based services* berjalan lebih baik pada perangkat *mobile*. Untuk pengujian keakuratan mendapatkan rata-rata *error* total (*desktop* dan *mobile*) 0.29% dari data 9 BTS yang berbeda. Untuk pengujian metode dapat disimpulkan bahwa, perhitungan metode dengan Excel memiliki selisih perhitungan yang minim. Hasil ini disebabkan oleh pembulatan angka yang selisih antara implementasi pada sistem dengan rumus dari excel. Untuk pengujian blackbox, pada setiap *case* yang diberikan sistem dapat memenuhi setiap *test* diberikan baik pada perangkat *desktop* atau *mobile*. Untuk perangkat *desktop* pada semua tampilan bisa ditampilkan dengan baik. Untuk perangkat *mobile* ada beberapa tampilan yang terpotong. Adapun saran pengembangan sistem mencakup aplikasi untuk mengimplementasi *Login multirole* terpisah dari firebase yang sudah ada, hasil jarak pada metode bisa ditampilkan dalam bentuk kilometer maupun meter, pengguna dapat menambahkan hasil dari jarak bts yang ada pada kegiatan surveynya

## DAFTAR PUSTAKA

- [1] Wahidin A (2013). Analisa Kepuasan Mahasiswa Terhadap Website Universitas Negeri Yogyakarta (Uny)
- [2] Nofyat, Ibrahim, A. , & Ambarita, A. (2018). Sistem Informasi Pengaduan Pelanggan Air Berbasis Website Pada Pdam Kota Ternate. IJIS - Indonesian Journal On Information System
- [3] Haklay, Muki dan Weber. (2008). OpenStreetMap (OSM): *User generated street maps*. *Journal of Pervasive Computing IEEE University College of London*.
- [4] Bennet, Jonathan. (2010). OpenStreetMap. Birmingham: Packt
- [5] Mahdia, F. , & Noviyanto, F. (2013). Pemanfaatan Google Maps API untuk pembangunan sistem informasi manajemen bantuan logistik pasca bencana alam berbasis mobile web (studi kasus: badan penanggulangan bencana daerah Kota Yogyakarta) (Doctoral dissertation, Universitas Ahmad Dahlan)
- [6] Neis P & Zielstra D (2014). *Recent Development and Future Trends in Volunteered Geographic Information Research: The Case of OpenStreetMap*. *Future Internet*
- [7] Yulianto. , Rahmadani, & Awang H K. (2018) Penerapan Formula Haversine Pada Sistem Informasi Geografis Pencarian Jarak Terdekat Lokasi Lapangan Futsal Jurnal Ilmiah Ilmu Komputer Vol. 13 No. 1
- [8] A. Annugrah, I. F. Astuti & A. H Kridalaksana. "Sistem Informasi Geografis Berbasis Web Pemetaan Lokasi Toko Oleh-Oleh Khas Samarinda". *Jurnal Informatika Mulawarman* Vol. 11, No. 2
- [9] A.P. Putra, "Aplikasi Manajemen Data dan Aplikasi Katalog Pemasaran Bisnis Properti Berbasis Android Menggunakan Firebase
- [10] E. Tanuwijaya, "Rancang Bangun Aplikasi Penitipan Hewan Peliharaan Berbasis Android," vol. 4, 2018.
- [11] Bahranyan P. , Hengky A. , & M Azhar I. (2016). Sistem Informasi Geografis Perumahan dan Fasilitas Sosial Terdekat dengan Metode Haversine Formula. *Jurnal Sistem dan Teknologi Informasi (JUSTIN)* Vol. 1, No. 1.
- [12] Abdul H. , & M Saefudin. (2021). Aplikasi Sistem Informasi Geografis Menggunakan Metode Haversine Formula Pencarian Rumah Kost Daerah Jakarta Selatan *Journal of Information System, Informatics and Computing* Vol. 5 No. 2
- [13] Suriyanto, M. , Febrinda, B. , & Dirgantara, W. (2019). Implementasi Haversine Formula pada Smart Garbage Can. *JEECAE (Journal of Electrical, Electronics, Control, and Automotive Engineering)*, 4(2), 257-264.
- [14] Salim, A., Surono, G., Pabelan, E. B., & Raizaldi, A. (2023). Penerapan Load Balancing Metode Per Connection Classifier Berbasis Router Mikrotik di PT. Asuransi Jiwa Nasional. *Jurnal Bidang Penelitian Informatika*, 1(1), 1–10.