

**PROTOTYPE SISTEM PERINGATAN KEDALAMAN LAUT PADA PERAIRAN NATUNA  
BERBASIS ANDROID DENGAN FUNGSI SIGNUM**

**Dico Eggy Refaxa<sup>1</sup>, Hendra Kurniawan, S.Kom., M.Sc.Eng.<sup>2</sup>,  
Muhamad Radzi Rathomi, S.Kom., M.Cs.<sup>3</sup>**

Mahasiswa Program Studi Teknik Informatika Fakultas Teknik UMRAH<sup>1</sup>  
Dosen Program Studi Teknik Informatika Fakultas Teknik UMRAH<sup>2</sup>  
Dosen Program Studi Teknik Informatika Fakultas Teknik UMRAH<sup>3</sup>

**ABSTRAK**

Indonesia merupakan negara kepulauan dengan laut yang luas dan memiliki kedalaman yang berbeda-beda. Masyarakat mengalami kesulitan dalam menentukan kedalaman laut secara langsung (fisik), sehingga membutuhkan alat bantu yang dapat memberikan informasi kedalaman laut secara akurat. Untuk mengetahui lokasi kedalaman laut tertentu diperlukan informasi kedalaman laut berbentuk peta digital, serta penggunaan teknologi *Global Positioning System (GPS)* yang dapat memberikan informasi secara real-time. Dalam penelitian ini dibangun prototype sistem peringatan kedalaman laut pada perairan natuna berbasis android dengan fungsi signum. Dengan menggunakan aplikasi ini akan mendapatkan informasi berupa notifikasi peringatan keberadaan lokasi *user* secara real-time pada area dangkal, dalam, atau sangat dalam. Hasil akurasi pengecekan kedalaman laut dipengaruhi oleh lokasi *user* yang dibaca oleh *GPS* dengan range kesalahan 2 hingga 5 m.

**Kata Kunci:** *Android, Fungsi Signum, Global Positioning System (GPS).*

**ABSTRACT**

Indonesia is an archipelago nation with vast sea and have variant of sea depth. Civilian having trouble to determine sea depth directly (physically), they need a device could give the sea depth information accurately. To determine the certain sea depth location needs an information of sea depth in form of digital map, and also by using *Global Positioning System (GPS)* technology who could give the information in real-time. This research will be build the prototype of depth warning system on Natuna sea based android with signum function. With this application, it will get the information in form of warning notification of user location in real-time on shallow, deep or very deep sea depth. Sea depth checking accuracy results affected by user's location with *GPS* error range of 2 until 5 metres

**Key Words:** *Android, Signum, Global Positioning System (GPS).*

## I. PENDAHULUAN

Indonesia merupakan negara kepulauan dengan laut yang luas dan memiliki banyak pulau yang dikelilingi dengan laut. Tetapi masih ada masyarakat yang tidak mengetahui seberapa jauh jarak antara laut yang kedalaman dangkal, sedang dan dalam dari suatu pulau yang mereka tempati.

Dalam ilmu kelautan, laut dibagi menjadi 4 menurut kedalamannya, antara lain : zona litoral, zona neritik, zona batial dan zona abbasal. Setiap zona diatas memiliki kedalaman laut yang berbeda-beda. Zona litoral yaitu merupakan daerah landai yang berbentuk pantai, zona neritik yaitu merupakan kedalaman laut yang masih bisa ditembusi sinar matahari sampai ke bawah permukaan air sehingga banyak hewan laut dan tumbuhan laut hidup di zona ini, kemudian zona batial yaitu kedalaman laut yang sangat dalam bahkan sinar matahari tidak bisa menembus sampai ke dasar laut. Sedangkan zona abbasal yaitu kedalaman laut yang paling dalam biasanya berbentuk jurang (palung) dan zona ini suhunya sangat rendah.

Untuk itulah perlu dibangun sebuah prototype sistem peringatan kedalaman laut pada perairan natuna berbasis android yang dapat memberikan informasi yang berupa peringatan kedalaman laut kepada *user* berdasarkan posisi *GPS user* tersebut menggunakan peta digital (*Google Maps*).

## II. TINJAUAN PUSTAKA

### 2.1 Kajian Terdahulu

Sebagai bahan pertimbangan dalam penelitian ini akan dicantumkan beberapa hasil penelitian terdahulu antara lain:

Yuwono dkk. (2015) “Sistem Informasi Geografis Berbasis *Android* Untuk Pariwisata di Daerah Magelang” memaparkan penelitian ini menghasilkan sebuah aplikasi Sistem Informasi Geografi Pariwisata di daerah Magelang berbasis *Android*. Aplikasi ini terhubung langsung dengan *Google Maps*. Tujuan sistem ini dibangun untuk memberikan informasi lokasi pariwisata di Daerah Magelang dengan mudah dan sederhana karena dapat dioperasikan dimanapun *user* berada menggunakan piranti *mobile device* berbasis android.

Dalam penelitian Zainudin dkk. (2014), yang berjudul “Rancang Bangun Aplikasi Peringatan Dini Batas Luar Wilayah Indonesia Berbasis *Android*” memaparkan bahwa aplikasi peringatan dini batas luar

wilayah dapat dikembangkan berbasis *Android* dengan menggunakan *Google Maps* sebagai peta digital dan titik koordinat dari *GPS* yang digunakan sebagai penanda bila *user* melewati batas wilayah tertentu.

Selanjutnya Dalam penelitian yang dilakukan oleh Hati dkk (2013), yang berjudul “Aplikasi Penanda Lokasi Peta Digital Berbasis Mobile GIS Pada *Smartphone Android*” memaparkan Dengan bantuan *Global Positioning System (GPS)* yang berfungsi sebagai penunjuk lokasi, *Location Based Service (LBS)* yang menyediakan informasi berdasarkan letak geografis perangkat mobile, melalui visualisasi *Google Maps*, maka aplikasi ini akan mudah digunakan.

Selanjutnya penelitian yang pernah dilakukan oleh Pakpahan dkk (2013), yang berjudul “Aplikasi Mobile Peta Rawan Bencana Kota Manado Berbasis *Android*” menyimpulkan bahwa *Google Maps* memudahkan dalam pengembangan aplikasi layanan pemetaan karena telah terintegrasi dengan *Android* dan aplikasi yang mereka kembangkan memberikan kemudahan kepada *user* untuk mengetahui wilayah rawan terjadi bencana banjir dan tanah longsor melalui perangkat mobile yang mereka miliki.

Kurniawan (2013) “Membuat Peta Geografis Potensi Wisata Kabupaten Klaten Menggunakan Aplikasi Berbasis *Android*” memaparkan bahwa pembuatan aplikasi berbasis *Android* ini dibuat untuk mempermudah wisatawan dalam pencarian tempat wisata di Kabupaten Klaten. Metode pembuatan aplikasi yaitu menggunakan metode pengumpulan data dari Dinas Pariwisata Klaten lalu mengumpulkan gambar dan deskripsi wisata di daerah Klaten serta menyiapkan *software* yang diperlukan untuk membangun aplikasi.

### 2.2 *Global Positioning System (GPS)*

*GPS* merupakan sistem navigasi menggunakan lebih dari 24 satelit *MEO (Medium Earth Orbit* atau *Middle Earth Orbit)* yang mengelilingi bumi sehingga penerima-penerima sinyal di permukaan bumi dapat menangkap sinyalnya. Satelit mengorbit pada ketinggian 12.000 mil di atas bumi dan mampu mengelilingi bumi dua kali dalam 24 jam (Apriyani dkk, 2012). Penentuan koordinat lokasi dengan menggunakan *GPS* tidak selamanya akurat, hal ini dipengaruhi oleh beberapa faktor kesalahan dan bias

(systematic error). Beberapa kesalahan pada akurasi *GPS* adalah sebagai berikut (Pusdiklat Kemendag, 2016):

**a. Kesalahan Ephemeris**

Posisi satelit sebagai fungsi waktu diprediksikan dari Observatori *GPS* di stasiun control. Akan tetapi posisi satelit tidaklah selalu sempurna seperti yang dilaporkan sehingga menyebabkan kesalahan ephemeris. Biasanya besar kesalahan ini berorde 2 hingga 5 m.

**b. Kesalahan jam satelit dan receiver**

Jam satelit walaupun berakurasi tinggi tidaklah sempurna. Jam satelit memiliki kesalahan sekitar 8,64 hingga 17,28 ns per hari.] dengan *range error* 2,59 m hingga 5,18 m (mengalikan kesalahan jam dengan kecepatan cahaya). Sedangkan untuk *receiver GPS*, sebaliknya, menggunakan jam Kristal yang lebih murah sehingga keakuratannya jauh di bawah jam satelit. Kesalahan tersebut mengurangi keakuratan dalam pengukuran *GPS*.

**c. Kesalahan Multipath**

*Multipath* adalah sumber kesalahan utama bagi pengukuran *pseudorange* maupun fase pembawa. Kesalahan tersebut terjadi apabila sinyal *GPS* sampai ke *receiver* melewati lintasan yang berbeda. Lintasan itu dapat berupa lintasan sinyal yang diterima secara langsung dan sinyal yang terpantulkan dari objek di sekitar antenna *receiver*.

**d. Noise dalam pesawat penerima (receiver)**

*Noise* dalam *receiver* berasal dari keterbatasan alat elektronik. Sistem *GPS* yang baik harus memiliki level noise yang minimal. Pada umumnya, *receiver GPS* melakukan tes ketika dinyalakan akan tetapi untuk mengevaluai performa sistem *GPS* tersebut dapat dilakukan dengan tes *baseline nol* dan tes *baseline pendek*.

**e. Efek Atmosfer**

Efek atmosfer terdiri dari efek yang disebabkan oleh ionosfer dan troposfer. Ionosfer akan mempengaruhi kecepatan, arah, polarisasi, dan kekuatan dari sinyal

*GPS* yang melaluinya. Pengaruh terbesar dari efek ini adalah pada kecepatan sinyal dan akan langsung mempengaruhi ukuran jarak dari pengamat ke satelit. Akan tetapi efek ini dapat dilakukan dengan menggunakan *receiver dual frequency*.

**f. Geometri Satelit dan DOP (*Dilution of Precision*)**

Geometri satelit memiliki peran penting dalam penentuan posisi yang akurat. Hal tersebut dapat tercapai apabila satelit tersebar secara merata di angkasa. Baik tidaknya geometri satelit dapat diukur dengan *dilution of precision* (DOP), yaitu bilangan yang digunakan untuk merefleksikan kekuatan geometri dari satelit yang diamati.

**2.3 Java**

*Java* adalah bahasa pemrograman yang dapat membuat seluruh bentuk aplikasi, desktop, web, mobile dan lainnya, sebagaimana dibuat dengan menggunakan bahasa pemrograman konvensional yang lain. Bahasa Pemrograman *Java* ini berorientasi objek (*OOP-Object Oriented Programming*), dan dapat dijalankan pada berbagai platform sistem operasi (Agrianza, 2014).

**2.4 Google Maps**

*Google Maps* adalah sebuah jasa peta globe virtual gratis dan online disediakan oleh *Google*. Fasilitas *Google Maps* dihadirkan oleh google sejak tahun 2005 dan terus dikembangkan hingga sekarang ini. Di alam *Google Maps*, anda tidak hanya mendapatkan tampilan peta dunia, namun juga informasi pendukung berupa informasi jalan, lokasi layanan public, bisnis, dan sebagainya. Untuk menjalankan layanan *Google Maps* ini dengan mengetikkan alamatnya (Hartati, 2012)

**2.5 Android**

*Android* adalah system operasi berbasis *Linux*. *Android* memiliki keunggulan sebagai perangkat lunak yang dapat didistribusikan secara terbuka (*open source*) sehingga pengguna bisa membuat aplikasi baru di dalamnya. Kerangka aplikasi : itu memungkinkan penggunaan dan penghapusan komponen yang tersedia.

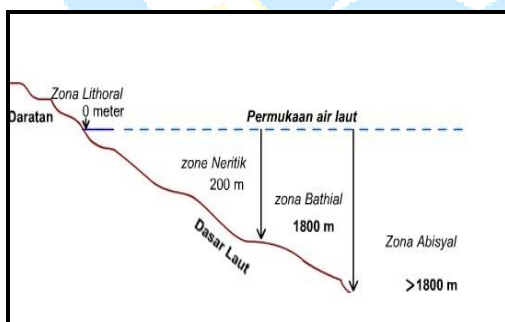
- Dalvik mesin virtual: mesin virtual dioptimalkan untuk perangkat *mobile*.
- Grafik: grafik di 2D dan grafis 3D berdasarkan pustaka OpenGL.
- *SQLite*: untuk penyimpanan data.
- Mendukung media: audio, video, dan berbagai format gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
- *GSM, Bluetooth, EDGE, 3G, dan Wi-Fi (hardware dependent)*
- Kamera, *Global Positioning System (GPS)*, kompas, dan *accelerometer* (tergantung *hardware*) (Apriyani dkk, 2012).

## 2.6 Zona Laut Berdasarkan Kedalaman

Berdasarkan kedalamannya, laut terbagi menjadi beberapa zona kelautan, seperti :

### a. Zona Litoral (Wilayah Pasang Surut)

Zona litoral adalah zona atau wilayah laut yang apabila pada saat terjadi air pasang, wilayah ini akan tergenang oleh air, dan pada saat terjadi air surut, wilayah ini akan mengering dan berubah menjadi pantai. Oleh karena itulah maka zona ini seringkali disebut dengan daerah pasang surut.



Gambar 2.1. Zonasi laut

### b. Zona Neritik (Laut dangkal)

Zona Neritik adalah wilayah perairan dangkal yang terletak dekat dengan pantai. Kedalaman dari zona ini adalah berkisar antara 50 hingga 200 meter. Kawasan ini dapat tertembus sinar matahari dengan sangat baik, sehingga menjadikannya

sebagai habitat yang sangat cocok bagi berbagai jenis spesies laut seperti ubur-ubung, fitoplankton. Zooplankton, rumput laut, serta jenis spesies lainnya.

### c. Zona Bathial (Laut Dalam)

Zona Bathial merupakan Wilayah perairan yang memiliki kedalaman yang berkisar antara 200 hingga 2000 meter. Wilayah ini tidak dapat ditembus oleh sinar matahari. Hal tersebutlah yang menjadikan kehidupan di wilayah zona bathial tidak seramai di zona neritik.

### d. Zona Abisal (Laut sangat Dalam)

Ini merupakan bagian laut yang memiliki kedalaman lebih dari 2000 meter. Wilayah ini memiliki suhu yang sangat dingin. Hal inilah yang menjadikan zona abisyal hanya memiliki beberapa spesies hewan laut. Dan di zona ini tidak dapat ditemui spesies tumbuh-tumbuhan laut. (Yulia, 2015).

## 2.7 Fungsi Signum

Fungsi signum merupakan fungsi matematika yang berfungsi membalikkan nilai *real x* yang diperoleh dari hasil perhitungan jarak titik *user*.

$$\text{Sgn}(x) = ((x_2 - x) * (y_t - y) - (x_t - x) * (y_2 - y)) \quad (2.1)$$

Keterangan :

$(x, y)$  = Titik sebelum

$(x_2, y_2)$  = Titik setelah

$(x_t, y_t)$  = Titik *user*

Contohnya hasil *x* adalah 0.167960, maka fungsi signum membalikkan nilai +1, karena  $0.167960 > 0$ . Fungsi signum bernilai +1 bila *x* positif, -1 bila *x* negatif, dan 0 bila *x* sama dengan 0 (Mabrur, 2009).

$$\text{Sgn}(x) = \begin{cases} +1 & \text{untuk } x > 0 \\ 0 & \text{untuk } x = 0 \\ -1 & \text{untuk } x < 0 \end{cases} \quad (2.2)$$

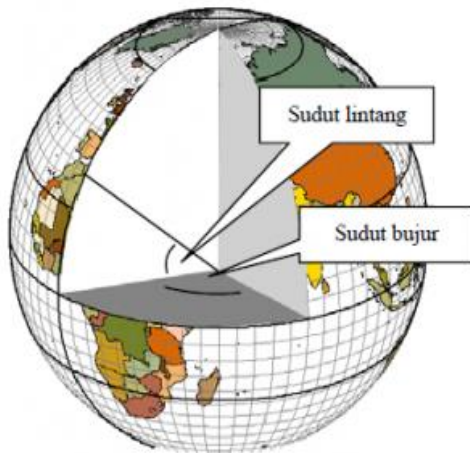
## 2.8 Sistem Koordinat Geografik

Jarak diukur dalam satuan derajat sudut yang dibentuk dari titik pangkal ke posisi tersebut melalui pusat bumi. Sedangkan titik



pangkal ditetapkan berada diperpotongan belahan utara-selatan bumi (garis khatulistiwa) dengan garis yang membelah bumi timur-barat melalui kota Greenwich di Inggris.

Posisi suatu tempat dialamatkan dengan nilai koordinat garis bujur (*longitude*) dan lintang (*latitude*) yang melalui tempat itu. Garis bujur (*longitude*), sering juga disebut garis meridian, yaitu merupakan garis lurus yang menghubungkan kutub utara dan selatan bumi.

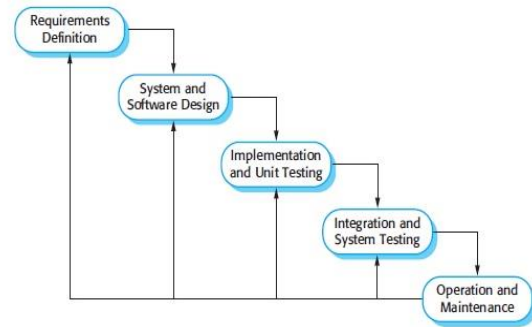


**Gambar 2.2.** Sistem Koordinat Geografik

Adapun nilai koordinat lintang dimulai dari garis lingkaran khatulistiwa yang diberi nilai  $0^\circ$ . Selanjutnya garis-garis lintang yang lain berupa lingkaran-lingkaran paralel (sejajar) khatulistiwa berada di sebelah utara dan selatan khatulistiwa. Lingkaran paralel di selatan disebut garis lintang selatan (LS) dan diberi nilai negatif, sedangkan lingkaran paralel di utara diberi nilai positif dan disebut garis lintang utara (LU). Nilai maksimum koordinat garis lintang adalah  $90^\circ$  yaitu terletak di kutub-kutub bumi (Rahayu, 2015).

### III. METODE PENELITIAN

Metode pengembangan sistem pada penelitian ini penulis menggunakan metode *waterfall*. Tahapan utama dari *waterfall model* langsung mencerminkan aktifitas pengembangan dasar. Terdapat 5 tahapan pada *waterfall model*, yaitu *requirement analysis and definition*, *system and software design*, *implementation and unit testing*, *integration and system testing*, dan *operation and maintenance* (Sommerville, 2011).



**Gambar 3.1.** *Waterfall Model* (Sommerville, 2011)

Berikut adalah penjelasan dari tahapan-tahapan tersebut :

#### 1. *Requirement Analysis and Definition*

Merupakan tahapan penetapan fitur, kendala dan tujuan sistem melalui konsultasi dengan pengguna sistem. Semua hal tersebut akan ditetapkan secara rinci dan berfungsi sebagai spesifikasi sistem.

#### 2. *System and Software Design*

Dalam tahapan ini akan dibentuk suatu arsitektur sistem berdasarkan persyaratan yang telah ditetapkan. Dan juga mengidentifikasi dan menggambarkan abstraksi dasar sistem perangkat lunak dan hubungan-hubungannya.

#### 3. *Implementation and Unit Testing*

Dalam tahapan ini, hasil dari desain perangkat lunak akan direalisasikan sebagai satu set program atau unit program. Setiap unit akan diuji apakah sudah memenuhi spesifikasinya.

#### 4. *Integration and System Testing*

Dalam tahapan ini, setiap unit program akan diintegrasikan satu sama lain dan diuji sebagai satu sistem yang utuh untuk memastikan sistem sudah memenuhi persyaratan yang ada. Setelah itu sistem akan dikirim ke pengguna sistem.

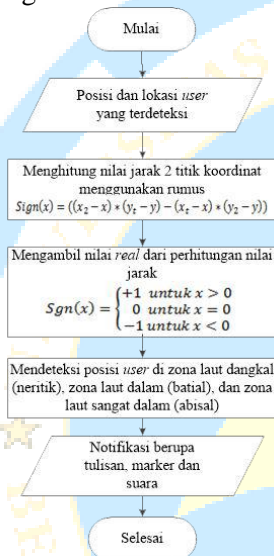
#### 5. *Operation and Maintenance*

Dalam tahapan ini, sistem diinstal dan mulai digunakan. Selain itu juga memperbaiki *error* yang tidak ditemukan pada tahap pembuatan. Dalam tahap ini juga dilakukan pengembangan sistem seperti penambahan fitur dan fungsi baru.

#### IV. PERANCANGAN

##### 4.1 Flowchart menampilkan Notifikasi

Flowchart diagram ini merupakan gambaran awal proses perancangan *Prototype* Sistem Peringatan Kedalaman Laut Perairan Natuna Berbasis *Android* Menggunakan Fungsi Signum untuk menentukan zona laut dangkal (neritik), zona laut dalam (batial), dan zona laut sangat dalam (abisal). Dari rancangan flowchart diagram inilah aplikasi *Prototype* Sistem Peringatan Kedalaman Laut Perairan Natuna Berbasis *Android* akan dibangun. Berikut merupakan gambar flowchart diagram.



**Gambar 4.1.** Flowchart menampilkan notifikasi

##### a. Deteksi Posisi dan Mengambil Nilai Titik User

Sistem akan mendeteksi lokasi serta mengambil secara otomatis nilai titik posisi *user* melalui *GPS* yang terdapat pada *smartphone*. Yang bertujuan untuk mendapatkan nilai *longitude* dan *latitude* posisi *user* yang akan digunakan untuk menghitung jarak *user* dengan titik zona kedalaman laut.

##### b. Menghitung Nilai Jarak Antar 2 Titik

Sistem akan mendapatkan nilai *longitude* dan *latitude user*, dan kemudian akan menghitung jaraknya antara titik *user* dengan titik zona kedalaman laut menggunakan rumus signum pada persamaan (2.2).

##### c. Mendapatkan Nilai Real dari Perhitungan Jarak

Kemudian, sistem mendapatkan nilai real dari hasil perhitungan jarak, untuk pengecekan zona posisi *user* berada pada area dangkal atau dalam.

##### d. Mendeteksi Keberadaan User di Area Dangkal, Dalam atau Sangat Dalam

Sistem akan mencoba memeriksa hasil dari nilai real yang diperoleh, dan sistem akan mengetahui posisi *user* saat itu berada di zona laut dangkal (neritik), zona laut dalam (batial), atau zona laut sangat dalam (abisal).

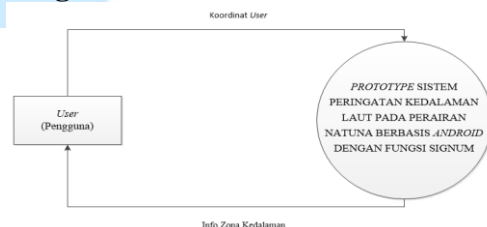
##### e. Menampilkan Notifikasi Berupa Peringatan

Pada proses ini sistem memberikan sebuah pesan berupa peringatan dalam bentuk tulisan yang berisikan informasi zona kedalaman (dangkal, dalam, atau sangat dalam), marker yang berubah warna sesuai kedalaman laut tertentu, jarak kedalaman sesuai lokasi *user* dan suara jika *user* memasuki area dangkal, dalam, atau sangat dalam.

##### 4.2 Perancangan alur sistem

Setelah mengetahui bagaimana alur program dan data berjalan secara umum pada sub bab sebelumnya. Pada sub bab ini akan dijelaskan bagaimana perancangan sistem yang dirancang untuk membangun *Prototype* Sistem Peringatan Kedalaman Laut Perairan Natuna Berbasis *Android* Menggunakan Fungsi Signum. Alat bantu yang digunakan adalah Diagram Konteks dan Data Flow Diagram (DFD).

##### a. Diagram Konteks



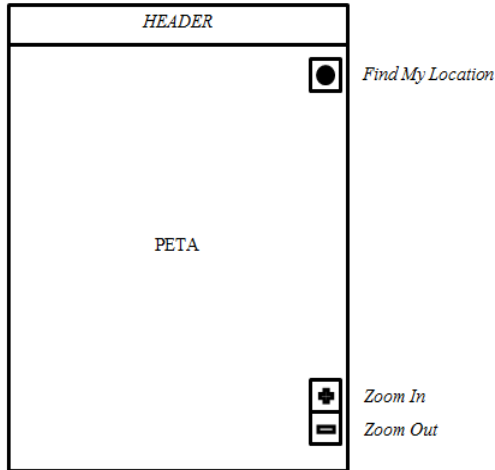
**Gambar 4.2.** Diagram Konteks

Pada Gambar 4.2. Diagram konteks menjelaskan bahwa *user* dapat memeriksa

lokasi pada sistem, kemudian sistem menampilkan peta dan info zona kedalaman kepada *user*.

### 4.3 Perancangan User Interface

Tampilan Form Peta dapat dilihat pada Gambar 4.3. dibawah ini.



Gambar 4.3. Perancangan Form Peta

Perancangan form peta pada Gambar 4.3. dirancang agar *user* bisa melihat lokasi keberadaannya disaat menggunakan aplikasi ini. Dalam form peta terdapat *header*, peta, tombol *find my location*, *zoom in*, dan *zoom out*. Dengan menggunakan tombol *find my location*, aplikasi ini dapat mengetahui posisi dan arah pandangan *user*. Tombol *zoom in* berfungsi sebagai tools untuk memperbesar jarak pandang peta, sedangkan tombol *zoom out* berfungsi sebagai tools untuk memperkecil jarak pandang peta.

## V. ANALISA DAN PEMBAHASAN

### 5.1 Deteksi Lokasi

Tujuan dari deteksi lokasi adalah untuk menguji keakuratan aplikasi yang dibangun dalam menentukan titik lokasi *user* secara sebenarnya (*real-time*). Hasil uji coba deteksi lokasi ini menggunakan *smartphone* yang bisa dilihat pada Tabel 5.1. sebagai berikut:

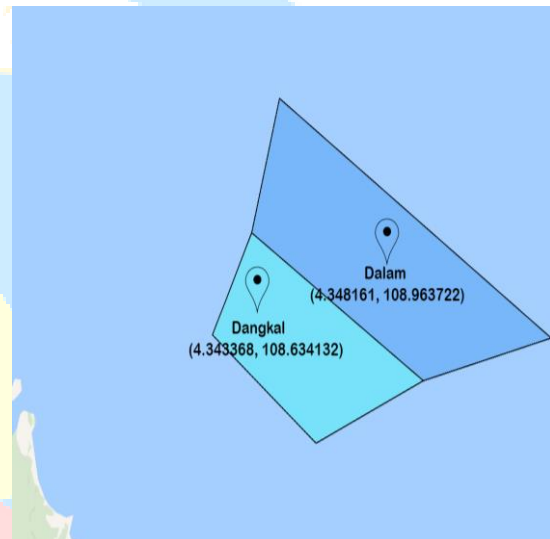
Tabel 5.1. Pengujian Deteksi Lokasi

No	Lokasi	Hasil Deteksi
1	Tanjungpinang City Center	± 3 meter dari posisi asli
2	Morning Bakery	± 3 meter dari posisi asli
3	STIE	± 3 meter dari posisi asli
4	Rumah Detensi Imigrasi	± 2 meter dari posisi asli
5	Pasar Raya Bintang 21	± 3 meter dari posisi asli

6	Kedai Kopi Uncu	± 2 meter dari posisi asli
7	Kantor Kejaksaan Tinggi	± 5 meter dari posisi asli
8	Toko Buku Lotus Km.9	± 4 meter dari posisi asli
9	Pamedan	± 2 meter dari posisi asli
10	Hotel Aston Tanjungpinang	± 3 meter dari posisi asli

### 5.2 Deteksi Area Kedalaman Laut Natuna

Titik koordinat yang sudah ditentukan akan dijadikan sebagai informasi area kedalaman laut berbentuk poligon. Berikut merupakan contoh kawasan kedalaman laut yang terletak di laut Natuna :



Gambar 5.1. Kawasan kedalaman laut Natuna

Keterangan titik koordinat yang terhubung dan menjadi kawasan kedalaman laut:

Area Dangkal : A(4.322827, 108.540190), B(4.457015, 108.608039), C(4.262572, 108.904670), D(4.180398, 108.719276).

Area Dalam : A(4.457015, 108.608039), B(4.633612, 108.656104), C(4.318719, 109.125770), D(4.262572, 108.904670).

Jika hasil perhitungan fungsi Signum adalah +1 maka *user* akan dinyatakan berada di dalam suatu area poligon, jika hasilnya  $\leq 0$  maka dinyatakan *user* berada di luar area poligon.

Pengecekan awal terhadap  $y_1$  apakah  $\leq y$ , jika benar kemudian dilakukan pengecekan pada kondisi pertama, jika tidak maka dilakukan pengecekan kondisi kedua.

Kondisi pertama :  $y_2 > y$  dan signum  $> 0$ , jika benar keduanya maka hasil (output)

yang diberikan adalah +1, jika salah satunya tidak benar maka output yang diberikan adalah 0.

Kondisi kedua :  $y_2 \leq y$  dan  $\text{signum} < 0$ , jika benar keduanya maka hasil (output) yang diberikan adalah -1, jika salah satunya tidak benar maka output yang diberikan adalah 0.

### 1. Titik User di Kawasan Dangkal (4.343368, 108.634132)

Garis pertama (Titik A ke Titik B)

$$y_1 \leq y$$

$$108.540190 \leq 108.634132 \text{ (benar)}$$

$$y_2 > y$$

$$108.608039 > 108.634132 \text{ (salah)}$$

$$y_2 \leq y$$

$$108.608039 \leq 108.634132 \text{ (benar)}$$

$$\begin{aligned} \text{Signum} &= ((4.457015 - 4.322827) * \\ & (108.634132 - 108.540190) - \\ & (4.343368 - 4.322827) * \\ & (108.608039 - 108.540190)) < \\ & 0 \end{aligned}$$

$$= (0.134188 * 0.093942) -$$

$$(0.020541 * 0.067849) < 0$$

$$= 0.012605889096 -$$

$$0.001393686309 < 0$$

$$= 0.011212202787 < 0 \text{ (salah)}$$

Maka *output* yang diperoleh adalah 0

Garis kedua (Titik B ke Titik C)

$$y_1 \leq y$$

$$108.608039 \leq 108.634132 \text{ (benar)}$$

$$y_2 > y$$

$$108.904670 > 108.634132 \text{ (benar)}$$

$$\begin{aligned} \text{Signum} &= ((4.262572 - 4.457015) * \\ & (108.634132 - 108.608039) - \\ & (4.343368 - 4.457015) * \\ & (108.904670 - 108.608039)) > \\ & 0 \end{aligned}$$

$$= (-0.194443 * 0.026093) -$$

$$(-0.113647 * 0.296631) > 0$$

$$= -0.005073601199 +$$

$$0.033711223257 > 0$$

$$= 0.011212202787 > 0 \text{ (benar)}$$

Maka *output* yang diperoleh adalah + 1

Garis ketiga (Titik C ke Titik D)

$$y_1 \leq y$$

$$108.904670 \leq 108.634132 \text{ (salah)}$$

$$y_2 \leq y$$

$$108.719276 \leq 108.634132 \text{ (salah)}$$

Maka *output* yang diperoleh adalah 0

Garis keempat (Titik D ke Titik A)

$$y_1 \leq y$$

$$108.719276 \leq 108.634132 \text{ (salah)}$$

$$y_2 \leq y$$

$$108.540190 \leq 108.634132 \text{ (benar)}$$

$$\begin{aligned} \text{Signum} &= ((4.322827 - 4.180398) * \\ & (108.634132 - 108.719276) - \\ & (4.343368 - 4.180398) * \\ & (108.540190 - 108.719276)) < \\ & 0 \end{aligned}$$

$$= (0.142429 * (-0.085144)) -$$

$$(0.162970 * (-0.179086)) < 0$$

$$= -0.012126974776 +$$

$$0.029185645420 < 0$$

$$= 0.017058670644 < 0 \text{ (salah)}$$

Maka *output* yang diperoleh adalah 0

Jadi, *output* yang didapatkan dari seluruh pengecekan adalah +1

### 2. Titik User di Kawasan Dalam (4.348161, 108.963722)

Garis pertama (Titik A ke Titik B)

$$y_1 \leq y$$

$$108.608039 \leq 108.963722 \text{ (benar)}$$

$$y_2 > y$$

$$108.656104 > 108.963722 \text{ (salah)}$$

$$y_2 \leq y$$

$$108.656104 \leq 108.963722 \text{ (benar)}$$

$$\begin{aligned} \text{Signum} &= ((4.633612 - 4.457015) * \\ & (108.963722 - 108.608039) - \\ & (4.348161 - 4.457015) * \\ & (108.656104 - 108.608039)) < 0 \end{aligned}$$

$$= (0.176597 * 0.355683) -$$

$$(-0.108854 * 0.048065) < 0$$

$$= 0.062812550751 +$$

$$0.005232067510 < 0$$

$$= 0.068044618261 < 0 \text{ (salah)}$$

Maka *output* yang diperoleh adalah 0

Garis kedua (Titik B ke Titik C)

$$y_1 \leq y$$

$$108.656104 \leq 108.963722 \text{ (benar)}$$

$$y_2 > y$$

$$109.125770 > 108.963722 \text{ (benar)}$$

$$\begin{aligned} \text{Signum} &= ((4.318719 - 4.633612) * \\ & (108.963722 - 108.656104) - \\ & (4.348161 - 4.633612) * \\ & (109.125770 - 108.656104)) > 0 \\ &= (-0.314893 * 0.307618) - (- \\ & 0.285451 * 0.469666) > 0 \\ &= -0.096866754874 + \\ & 0.134066629366 > 0 \\ &= 0.037199874492 > 0 \text{ (benar)} \end{aligned}$$



Maka *output* yang diperoleh adalah + 1

Garis ketiga (Titik C ke Titik D)

$$y_1 \leq y$$

$$109.125770 \leq 108.963722 \text{ (salah)}$$

$$y_2 \leq y$$

$$108.904670 \leq 108.963722 \text{ (benar)}$$

$$\begin{aligned} \text{Signum} &= ((4.262572 - 4.318719) * \\ & (108.963722 - 109.125770) - \\ & (4.348161 - 4.318719) * \\ & (108.904670 - 109.125770)) < 0 \\ &= (-0.056147 * (-0.162048)) - \\ & (0.029442 * (-0.221100)) < 0 \\ &= 0.009098509056 + \\ & 0.006509626200 < 0 \\ &= 0.015608135256 < 0 \text{ (salah)} \end{aligned}$$

Maka *output* yang diperoleh adalah 0

Garis keempat (Titik D ke Titik A)

$$y_1 \leq y$$

$$108.904670 \leq 108.963722 \text{ (benar)}$$

$$y_2 > y$$

$$108.608039 > 108.963722 \text{ (salah)}$$

$$y_2 \leq y$$

$$108.608039 \leq 108.963722 \text{ (benar)}$$

$$\begin{aligned} \text{Signum} &= ((4.457015 - 4.262572) * \\ & (108.963722 - 108.904670) - \\ & (4.348161 - 4.262572) * \\ & (108.608039 - 108.904670)) < 0 \\ &= (0.194443 * 0.059052) - \\ & (0.085589 * (-0.296631)) < 0 \\ &= 0.011482248036 + \\ & 0.025388350659 < 0 \\ &= 0.036870598695 < 0 \text{ (salah)} \end{aligned}$$

Maka *output* yang diperoleh adalah 0

Jadi, *output* yang didapatkan dari seluruh pengecekan adalah +1

### 5.3 Menampilkan Informasi Area Kedalaman Laut

Berdasarkan perhitungan di atas bahwa *user* diindikasikan berada di dalam suatu area atau diluar area yang dijadikan sebagai sample. Maka informasi area yang ditampilkan berupa pesan notifikasi, marker dan suara sesuai area kedalamannya. Pesan bertuliskan " Perairan dangkal " dan pesan suara berbunyi "Anda berada di kawasan dangkal dengan kedalaman 50 - 200 meter" untuk di area dangkal. Pesan bertuliskan " Perairan dalam " dan pesan suara berbunyi "Anda berada di kawasan dalam dengan

kedalaman 200 - 2000 meter", serta Warna marker berubah sesuai warna kedalaman laut yang ditentukan.

### 5.4 Menampilkan Informasi Area Kedalaman Laut

Dilakukan pengujian simulasi pergerakan titik *user* dari titik A menuju titik B dengan melewati kawasan dangkal, dalam atau sangat dalam pada aplikasi prototype sistem peringatan kedalaman laut pada perairan natuna berbasis android. Contoh rute simulasi pergerakan titik *user* seperti gambar berikut :



**Gambar 5.2.** Simulasi Pergerakan *User* dari Titik A ke Titik B

Untuk melakukan simulasi pergerakan titik *user* ini diperlukan aplikasi android yang dapat meletakkan titik *user* di tempat tertentu dan melakukan perjalanan sesuai yang ditentukan, penulis menggunakan *software android* bernama *Mock Location (fake path)*. Pengecekan titik *user* dilakukan setelah *user* menempuh jarak 300 M, dalam pengujian ini *user* berjalan dengan kecepatan ± 60 Km/jam dan berjarak 11.351 Km. Setelah dilakukan simulasi pergerakan titik *user*, berikut tabel hasil dari simulasi pergerakan titik *user* :

**Tabel 5.4.** Hasil Pengujian Simulasi Pergerakan *User* dari Titik A ke Titik B

No	Latitude	Longitude	Hasil Deteksi Posisi User
1	5.798848	110.099034	Kawasan Dangkal
2	5.801034	110.100663	Kawasan Dangkal
3	5.803269	110.102328	Kawasan Dangkal
4	5.805624	110.104082	Kawasan Dangkal
5	5.807912	110.105870	Kawasan Dangkal
6	5.810146	110.107451	Kawasan Dalam

No	Latitude	Longitude	Hasil Deteksi Posisi User
7	5.812515	110.109216	Kawasan Dalam
8	5.814808	110.110924	Kawasan Dalam
9	5.816984	110.112545	Kawasan Dalam
10	5.819248	110.114232	Kawasan Dalam
11	5.821486	110.115899	Kawasan Dalam
12	5.823865	110.117672	Kawasan Dalam
13	5.826156	110.119376	Kawasan Dalam
14	5.828400	110.121050	Kawasan Dalam
15	5.830657	110.122731	Kawasan Dalam
16	5.833024	110.124495	Kawasan Dalam
17	5.835316	110.126203	Kawasan Dalam
18	5.837544	110.127862	Kawasan Dalam
19	5.839778	110.129527	Kawasan Dalam
20	5.842011	110.131190	Kawasan Dalam
21	5.844241	110.132851	Kawasan Dalam
22	5.846648	110.134645	Kawasan Dalam
23	5.848935	110.136349	Kawasan Dalam
24	5.851318	110.138124	Kawasan Dalam
25	5.853570	110.139801	Kawasan Dalam
26	5.855851	110.141501	Kawasan Dalam
27	5.858150	110.143213	Kawasan Dalam
28	5.860495	110.144961	Kawasan Dalam
29	5.862770	110.146781	Kawasan Dalam
30	5.864820	110.148184	Kawasan Dalam
31	5.867318	110.150044	Kawasan Dalam
32	5.869941	110.151998	Kawasan Dalam
33	5.872225	110.153699	Kawasan Dalam
34	5.874601	110.155470	Kawasan Dalam
35	5.876962	110.157228	Kawasan Dalam
36	5.879201	110.158897	Kawasan Dalam

## VI. KESIMPULAN DAN SARAN

Berdasarkan analisis dan pembahasan yang telah penulis lakukan dalam penelitian ini, maka dapat diambil kesimpulan *Prototype* sistem peringatan kedalaman laut perairan Natuna berbasis android dengan fungsi signum ini mampu memberikan peringatan kepada *user* secara real-time keberadaan *user* pada area dangkal, dalam, atau sangat dalam serta kedalamannya dan hasil akurasi pengecekan kedalaman laut dipengaruhi oleh lokasi *user* yang dibaca oleh *GPS* dengan range kesalahan 2 hingga 5 m.

Beberapa saran yang dapat penulis sampaikan guna perkembangan dalam penelitian ini adalah sebagai berikut:

1. Penelitian selanjutnya perlu menambahkan suatu metode yang berfungsi untuk memprediksi kondisi pasang surutnya air laut.
2. Penelitian selanjutnya perlu dikembangkan dengan menggunakan

metode lain seperti algoritma *Ray Casting* untuk melihat keakuratan deteksi lokasi *user* di dalam poligon dibandingkan dengan fungsi signum.

3. Penelitian selanjutnya data koordinat kedalaman laut dimasukkan kedalam *database*.
4. Penelitian selanjutnya dapat dikembangkan dengan menggunakan batas poligon yang tumpang tindih (*overlapping polygon*).

## DAFTAR PUSTAKA

Agrianza H., R. (2015), Pengertian dan Sejarah Bahasa Pemrograman Java, <http://natery.blog.widyatama.ac.id/2015/09/12/pengertian-dan-sejarah-bahasa-pemrograman-java/>, Diakses 14 Oktober 2016.

Apriyani, M. E., Giovanny, R., & Haris, P. Y. (2012). Sistem Pelacakan Posisi Kapal Berbasis Mobile *Android* dan Web Server. *Jurnal Integrasi*, 4(2).

Hartati, E. (2012). Mengenal dan Menjalankan Google Maps. <http://news.palcomtech.com/2012/11/mengenal-dan-menjalankan-google-maps/>. Diakses pada 29 Oktober 2016.

Hati, G. M., & Akhir, J. T. (2013). Aplikasi Penanda Lokasi Peta Digital Berbasis Mobile GIS pada Smartphone *Android*. *Jurnal Geodesi UNDIP*, 4.

Iansyah, (2016). Apa itu pengujian kotak hitam. <http://www.iansyahbelajar.com/2016/06/apa-itu-pengujian-kotak-hitam.html>. Diakses pada 2 Juli 2017

Kurniawan, E. (2013). *Membuat Peta Geografis Potensi Wisata Kabupaten*

*Klaten Menggunakan Aplikasi Berbasis Android* (Doctoral dissertation, Universitas Muhammadiyah Surakarta).

Peringatan Dini Batas Luar Wilayah Indonesia Berbasis *Android*.

Mabrur, Andik., (2009), Fungsi Matematika. <http://its-matematika.blogspot.co.id/2009/12/fungsi-matematika.html?m=1>. Diakses pada 2 Juli 2017

Pakpahan, T., Rindengan, Y. D., & Najoan, X. (2013). Aplikasi Mobile Peta Rawan Bencana Kota Manado Berbasis *Android*. *Jurnal Teknik Elektro dan Komputer UNSRAT*, 2(4), 77-82.

Pusdiklat Kemendag, (2016). Mengenal Lebih Dalam *Global Positioning System (GPS)* <http://pusdiklat.kemendag.go.id/bdpm-b/kolom/mengenal-lebih-dalam-global-positioning-system-GPS/> Diakses pada 20 Juli 2017

Rahayu, E. (2015). Sistem Koordinat Geografik. <http://www.ekorahayu.com/sistem-koordinat-geografik.html> Diakses pada 29 Oktober 2016.

Yulia, M.SI., (2015). Zona Laut Berdasarkan Kedalaman, <http://ilmugeografi.com/ilmu-bumi/laut/zona-laut>, Diakses 17 Oktober 2016

Yuwono, B., & Aribowo, A. S. (2015). Sistem Informasi Geografis Berbasis *Android* untuk Pariwisata di Daerah Magelang. In Seminar Nasional Informatika (SEMNASIF) (Vol. 1, No. 1).

Zainudin, M. Y., Bettiza, M., & Kurniawan, H. (2014), Rancang Bangun Aplikasi