

Simulasi Penentuan Posisi Target Menggunakan Teknik *Time Difference of Arrival* (TDoA) dengan MATLAB

Ribka Mariska ^{#1}, Sofia Saidah ^{*2}, Aisyah Novfitri ^{#3}

¹ *Departemen Teknik Telekomunikasi, Universitas Telkom
Daan Mogot KM. 11, Jakarta Barat, 11710, Indonesia*

¹ Mariskaribka@student.telkomuniversity.ac.id

² sofiasaidahsfi@telkomuniversity.ac.id

³ aisyahnovf@telkomuniversity.ac.id

Received on dd-mm-yyyy, revised on dd-mm-yyyy, accepted on dd-mm-yyyy

Abstract

Saat ini penentuan posisi sangatlah penting karena selain untuk mencari barang di luar ruangan, penentuan posisi ini juga dapat menentukan posisi suatu objek bergerak dengan memanfaatkan perbedaan waktu kedatangan sinyal pada beberapa titik sensor. Dalam kehidupan sehari-hari, TDOA dapat digunakan dalam berbagai aplikasi seperti navigasi, komunikasi seluler, dan pemantauan lingkungan, mencari korban saat terjadi bencana alam.

Ada beberapa solusi untuk masalah tersebut salah satunya adalah menggunakan *Teknik Time Difference of Arrival* (TDoA) merupakan metode pemosisian yang menggunakan perbedaan waktu kedatangan sinyal dari beberapa stasiun basis untuk menentukan posisi suatu benda di luar ruangan. Metode ini umumnya digunakan dalam sistem pemosisian global seperti GPS untuk mendapatkan koordinat yang akurat.

Dalam teknik TDoA sinkronisasi waktu hanya diperlukan di antara BS tetap. Biasanya, mereka terhubung ke kabel *backbone*, yang menyederhanakan sinkronisasi di antara mereka. Sehingga mempermudah dalam pencarian lokasi dibandingkan metode yang lain. Berdasarkan penelitian yang dilakukan, diperoleh bahwa TDOA pada BTS (Sensor) dapat ditentukan posisi target MS dengan tingkat kesalahan terbesar 230.03 meter, dan kesalahan terkecil adalah 0.185 meter, running time yang diperlukan untuk menjalankan program simulasi TDOA ini antara 0.01 sampai 0.18 detik, dan dengan algoritma modified TDOA mampu memperbaiki kesalahan terbesar 62.14 meter atau 72.986% dan terkecil adalah 0.14 meter atau 24.324%.

Keywords: Matlab, *Time of Arrival*, *Time Difference of Arrival*, Simulasi

I. INTRODUCTION

Pada dasarnya memori manusia dalam mengingat terbatas sehingga manusia gampang melupakan segala sesuatu termasuk barang barangnya. Hal tersebut sangatlah merugikan dalam hal waktu, uang, dan tenaga jika kehilangan barang tersebut. Untuk itu perlu adanya pencegahan agar tidak menimbulkan banyak kerugian.[1]

Saat ini penentuan posisi sangatlah penting karena selain untuk mencari barang di luar ruangan, penentuan posisi ini juga dapat menentukan posisi suatu objek bergerak dengan memanfaatkan perbedaan waktu kedatangan sinyal pada beberapa titik sensor. Dalam kehidupan sehari-hari, TDOA dapat digunakan dalam berbagai aplikasi seperti navigasi, komunikasi seluler, dan pemantauan lingkungan, mencari korban saat terjadi bencana alam. Misalnya, dalam navigasi GPS, TDOA digunakan untuk menentukan posisi pengguna berdasarkan perbedaan waktu kedatangan sinyal dari satelit. Ini membantu kita mengetahui lokasi kita dengan akurat dan efisien [2]

TDOA atau *Time Difference of Arrival* adalah metode untuk menentukan posisi seseorang atau objek dengan membandingkan waktu kedatangan sinyal radio dari beberapa titik referensi yang diketahui. Dalam konteks evakuasi

korban, TDOA dapat digunakan dengan memasang beberapa sensor di sekitar area evakuasi dan memantau waktu kedatangan sinyal dari korban yang memerlukan bantuan.

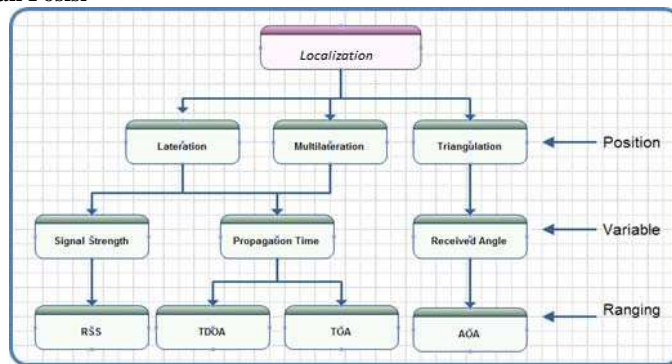
Ketika korban memancarkan sinyal darurat, sensor-sensor tersebut akan merekam waktu kedatangan sinyal tersebut. Dengan menggunakan perbedaan waktu kedatangan sinyal di antara sensor-sensor tersebut, sistem dapat memperkirakan posisi korban secara akurat. Hal ini memungkinkan tim penyelamat untuk merespons dengan cepat dan efektif dalam evakuasi korban.

Penting untuk melatih tim penyelamat dalam penggunaan sistem TDOA dan memastikan peralatan dan sensor-sensornya berfungsi dengan baik. Selain itu, koordinasi yang baik antara tim penyelamat dan korban juga sangat diperlukan untuk memastikan evakuasi berjalan lancar dan aman.

Dalam menentukan posisi benda menggunakan teknik TDoA nirkabel dapat didefinisikan sebagai estimasi kordinat *mobile unit* (MU) di area yang diperlukan. Ini dapat diperoleh dengan mentransmisikan secara berkala yang dirancang dengan benar sinyal dari MU dan menerimanya di sejumlah *Base Station* (BS) dengan diketahui koordinat posisi ke sistem. Koordinat MU kemudian diperkirakan dengan memproses sinyal yang diterima. Karakteristik sinyal yang diterima digunakan untuk melokalisasi berbasis MU pada interaksi antara sinyal yang ditransmisikan dan saluran nirkabel. BS bisa jadi disebut *Access point* (AP) atau *Reference Point* (RP). Ada beberapa metode yang digunakan untuk pemosisian luar ruangan, seperti *Received Signal Strength Indication* (RSSI), *Angle of Arrival* (AoA), *Time of Arrival* (ToA), *Time Difference of Arrival* (TDOA) [3].

II. TINJAUAN PUSTAKA

2.1 Konsep Penentuan Posisi

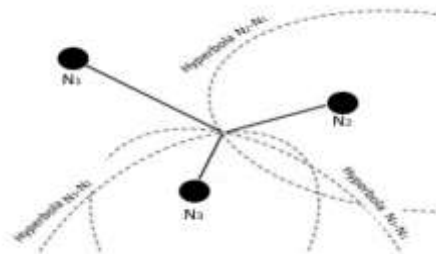


Gambar 2. 1 Taksonomi Penentuan Lokasi

Konsep yang digunakan untuk menentukan posisi obyek atau target di dalam maupun diluar ruangan, dapat merujuk pada taksonomi pada gambar 2.1 dimana terdapat *triangulation*, dimana konsep ini menggunakan dua konsep *lateration* (laterasi) dan *angulation*. Sedangkan dalam penelitian ini menggunakan Metode *multilateration*.

2.1.1 Metode Time Difference of Arrival (TDoA)

Time Difference of Arrival (TDoA) ini mirip dengan ToA, tetapi terkadang waktunya tidak diketahui, jadi perbedaannya antara waktu perjalanan digunakan. Lokasi dapat diperkirakan oleh persimpangan hiperbola ToA diferensial sinyal[7]. TDoA ditunjukkan pada Gambar 2.4.



Gambar 2. 2 Model Pemosisian dengan Metode TDoA

Metode yang lebih sederhana yang menghindari sinkronisasi antara MU dan *node* adalah TDoA. Namun, sinkronisasi antara *node* itu sendiri masih merupakan prasyarat. TDoA praktis adalah perbedaan waktu kedatangan sinyal yang dipancarkan, diterima pada sepasang sensor seperti yang diilustrasikan pada gambar 2.4. Selain itu, mirip dengan ToA, lokasi yang tepat dari semua stasiun pangkalan dianggap diketahui oleh infrastruktur lokalisasi. Setiap TDoA yang dihitung membentuk hiperbola di ruang lokalisasi dan persimpangan semua hiperbola mewakili kemungkinan lokasi terminal seluler. Model pengukuran TDoA dapat diasumsikan bahwa sumber memancarkan sinyal pada waktu yang tidak diketahui t_0 dan sensor ke- l menerimanya pada waktu t_l , $l = 1, 2, \dots, L$ dengan $L > 3$. Ada $L(L-1)/2$ TDoA yang berbeda dari semua pasangan sensor yang mungkin, dilambangkan oleh $tk,l = (t_k - t_0) - (t_l - t_0) = t_k - t_l$, dimana $l = 1, 2, \dots, L$ dengan $k > l$. Namun hanya ada $(L-1)$ TDoA yang tidak sama. Misalnya jika $L = 3$, yang berbeda. TDoA adalah t_{21} , t_{31} dan t_{32} . Namun, $t_{32} = t_{31} - t_{21}$ [8].

2.2 Konsep dasar Operasi Matrik

Operasi matriks yang digunakan untuk menyelesaikan persamaan linier dalam bentuk umum sistem persamaan linear pada artikel mengenai Sistem Persamaan Linier (SPL), mempunyai matriks yang bersesuaian yang disebut matriks yang diperluas atau *augmented matrix* sebagai berikut sebagai berikut:

$$[A|b] = \left[\begin{array}{cccc|c} a_{11} & a_{12} & a_{13} & \dots & a_{1N} & b_1 \\ a_{21} & a_{22} & a_{23} & \dots & a_{2N} & b_2 \\ a_{31} & a_{32} & a_{33} & \dots & a_{3N} & b_3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{M1} & a_{M2} & a_{M3} & \dots & a_{MN} & b_M \end{array} \right] \quad (2.9)$$

Dalam mencari penyelesaian suatu sistem persamaan linear dengan metode eliminasi dan substitusi. Berikut ini adalah pembahasan mengenai cara mencari penyelesaian sistem persamaan linear dengan metode eliminasi dan substitusi), langkah-langkah yang dilakukan dapat dibedakan menjadi 3 macam, yaitu:

1. menukar letak dua persamaan.
2. mengalikan suatu persamaan dengan skalar tak nol.
3. menambah suatu persamaan dengan kelipatan persamaan yang lain.

Langkah-langkah tersebut berpengaruh pada matriks yang diperluas $[A|b]$ yang selanjutnya dikenal dengan sebutan **operasi baris elementer** yang dibagi menjadi 3, yaitu:

1. menukar letak dua baris;
2. mengalikan suatu baris dengan skalar tak nol;
3. menambah suatu baris dengan kelipatan baris yang lain.

Operasi-operasi baris elementer tersebut mempunyai tujuan membawa matriks yang diperluas menjadi matriks dengan bentuk lebih sederhana, atau lebih tepatnya dibawa ke **bentuk eselon**.

Proses menghasilkan bentuk eselon baris ini disebut **eliminasi Gauss**. Jika matriks yang dihasilkan merupakan matriks bentuk eselon baris tereduksi, prosesnya disebut **eliminasi Gauss-Jordan** dengan Matlab dapat diselesaikan dengan fungsi rref (*reduced row echelon form*) seperti contoh berikut[11]:

Diberikan persamaan linier sebagai berikut:

$$\begin{aligned} 2x + y + 2z &= 10 \\ x + 2y + z &= 8 \\ 3x + y - z &= 2 \end{aligned} \quad (2.10)$$

Untuk menyelesaikan persamaan diatas, dapat dilakukan dengan menggunakan metoda Eliminasi Gauss-Jordan, yaitu membuat koefisien dari persamaan diatas menjadi matrik diagonal yang bernilai 1 dan bernilai 0 untuk yang lainnya.

Langkah 1: Membuat matrik dari koefisien persamaan diatas seperti berikut:

$$\begin{bmatrix} 2 & 1 & 2 \\ 1 & 2 & 1 \\ 3 & 1 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 10 \\ 8 \\ 2 \end{bmatrix} \begin{matrix} \rightarrow b_1 \\ \rightarrow b_2 \\ \rightarrow b_3 \end{matrix} \quad (2.11)$$

Langkah 2: Menukar baris 1 atau b_1 dengan baris 2 atau b_2 dan menyembunyikan variable, x,y,z seperti berikut:

$$\begin{bmatrix} 1 & 2 & 1 & | & 8 \\ 2 & 1 & 2 & | & 10 \\ 3 & 1 & -1 & | & 2 \end{bmatrix} \begin{matrix} \rightarrow b_1 \\ \rightarrow b_2 \\ \rightarrow b_3 \end{matrix} \quad (2.12)$$

Langkah 3: Pada baris ke-2 kolom ke-1 bernilai 0 dengan rumus $-2b_1 + b_2$ lalu selesaikan baris ke-2 seperti berikut:

$$\begin{bmatrix} 1 & 2 & 1 & | & 8 \\ 0 & -3 & 0 & | & -6 \\ 3 & 1 & -1 & | & 2 \end{bmatrix} \rightarrow -2b_1 + b_2 \quad (2.13)$$

Langkah 4: Pada baris ke-3 kolom ke-1 dibuat bernilai 0, dengan rumus $-3b_1 + b_3$ lalu selesaikan baris ke-3 seperti berikut:

$$\begin{bmatrix} 1 & 2 & 1 & | & 8 \\ 0 & -3 & 0 & | & -6 \\ 0 & -5 & -4 & | & -22 \end{bmatrix} \rightarrow -3b_1 + b_3 \quad (2.14)$$

Langkah 5: Pada baris ke-2 dan kolom ke-2 bernilai 1 dengan rumus $b_2 : (-3)$ lalu selesaikan baris ke-2 seperti berikut:

$$\begin{bmatrix} 1 & 2 & 1 & | & 8 \\ 0 & 1 & 0 & | & 2 \\ 0 & -5 & -4 & | & -22 \end{bmatrix} \rightarrow b_2 : (-3) \quad (2.15)$$

Langkah 6: Pada kolom ke-2 selain diagonal dibuat bernilai 0, seperti berikut:

$$\begin{bmatrix} 1 & 0 & 1 & | & 4 \\ 0 & 1 & 0 & | & 2 \\ 0 & 0 & -4 & | & -12 \end{bmatrix} \rightarrow -2b_2 + b_1 \text{ dan } 5b_2 + b_3 \quad (2.16)$$

Langkah 7: Pada baris ke-3 diagonal bernilai 1 dan selain diagonal dibuat bernilai 0, seperti berikut:

$$\begin{bmatrix} 1 & 0 & 1 & | & 4 \\ 0 & 1 & 0 & | & 2 \\ 0 & 0 & 1 & | & 3 \end{bmatrix} \rightarrow b_3 : (-4) \quad (2.17)$$

Langkah 8: Pada kolom ke-3 selain diagonal dibuat bernilai 0, seperti berikut:

$$\begin{bmatrix} 1 & 0 & 0 & | & 1 \\ 0 & 1 & 0 & | & 2 \\ 0 & 0 & 1 & | & 3 \end{bmatrix} \rightarrow -b_3 + b_1 \quad (2.18)$$

Dengan mengembalikan variable $x, y,$ dan z diperoleh:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \quad (2.19)$$

Sehingga diperoleh $x=1, y=2$ dan $z=3$. Metode eliminasi Gauss-Jordan tersebut diatas dapat diselesaikan dengan menggunakan Matlab dengan cara sebagai berikut:

```
>> A=[2 1 2;1 2 1;3 1 -1]
```

```
A =
```

```
2 1 2
1 2 1
3 1 -1
```

```
>> b=[10 8 2]';
```

```
>> M=[A b]
```

```
M =
```

```
2 1 2 10
1 2 1 8
3 1 -1 2
```

```
>> R = rref(M)
```

```
R =
```

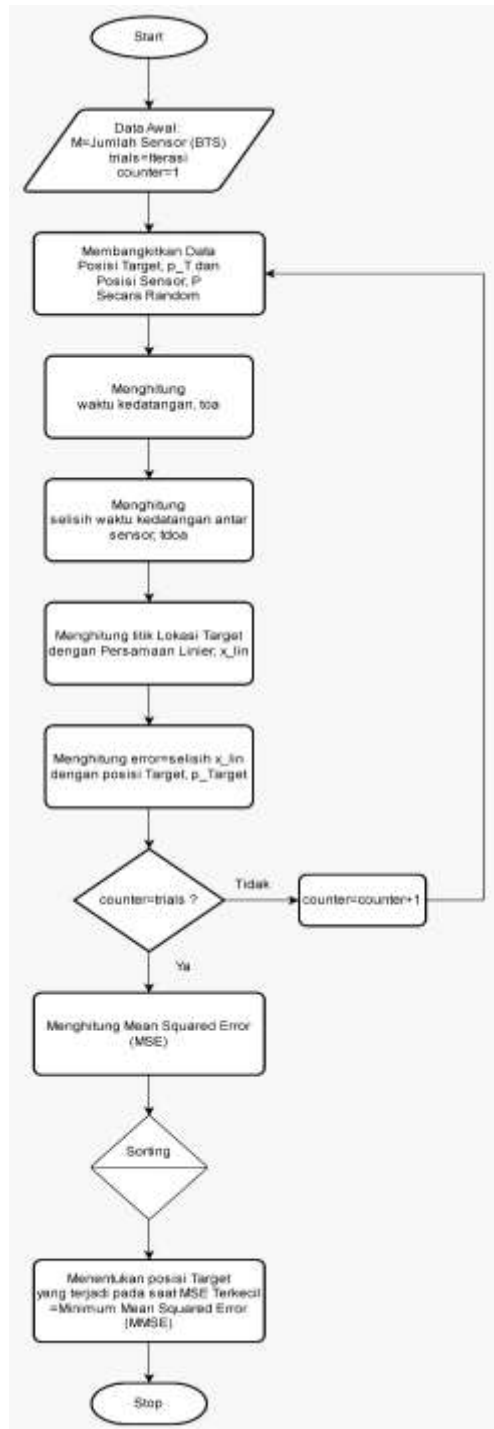
```
1 0 0 1
0 1 0 2
0 0 1 3
```

Sehingga diperoleh $x=1, y=2$ dan $z=3$.

III. RANCANGAN SISTEM

3.1 Flowchart

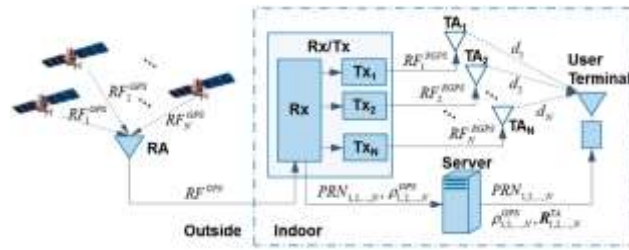
Pada proses pembuatan simulasi penentuan posisi menggunakan Teknik *Time Difference of Arrival (TDoA)* dengan MATLAB diperlukan *flowchart* yang berguna untuk memberikan gambaran untuk pengerjakan atau proses pembuatannya. Gambar dibawah ini merupakan *flowchart* alur pengerjaannya.



Gambar 3. 1 Flowchart

3.2 Desain Sistem

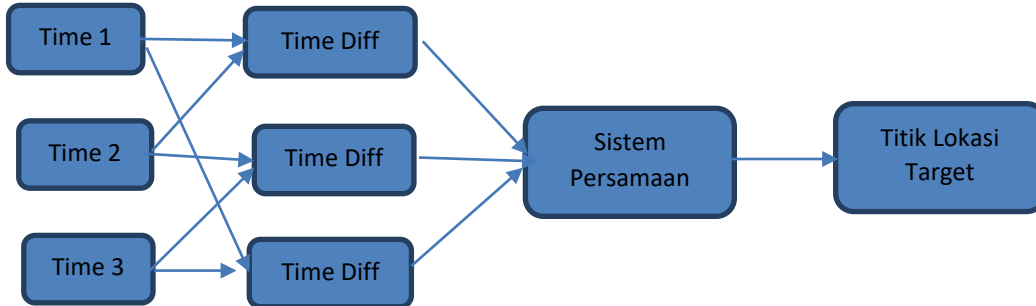
Secara umum untuk memberikan gambaran simulasi sistem penentuan posisi menggunakan teknik TDoA menggunakan MATLAB yang dibangun dapat merujuk pada gambar arsitektur pada gambar 3.2.



Gambar 3. 2 Arsitektur Sistem Pemosisian di dalam dan di luar ruangan

Gambar 3.2. menjadi rujukan untuk membuat simulasi dengan menggunakan bahasa pemrograman MATLAB, sehingga memudahkan dalam melakukan pemodelan, perancangan dan pembuatan. Dari gambar 3.2, terdapat bagian-bagian yang mempunyai fungsi dan fitur yang berbeda untuk membentuk satu sistem penentuan lokasi menggunakan TDoA. Dimana *User Terminal* berfungsi sebagai *Mobile Unit* (MU) yang bergerak untuk dilacak posisinya. Sedangkan *Transmitter* Tx1, Tx2, Tx3 dan seterusnya berperan sebagai *Base Station* (BS), yang riilnya digunakan Access Point (AP). Sedangkan yang berperan sebagai komputer server dalam hal ini adalah laptop.

3.3 Diagram Blok

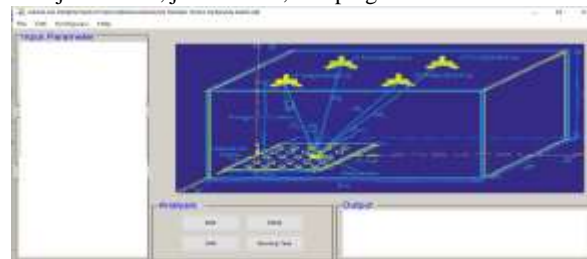


Gambar 3. 3 Diagram Blok TDoA

- *Time 1* waktu yang diperlukan untuk sinyal dari target sampai ke receiver 1. *Time 2* waktu yang diperlukan untuk signal dari target sampai ke receiver 2. *Time 3* waktu yang diperlukan untuk sinyal dari target sampai ke receiver 3.
- Menghitung Time difference dari masing waktu
- Menghitung Jarak didapat dari selisih waktu dikalikan kecepatan cahaya sehingga jaraknya didapat dari *time 1,2,* dan 3.
- Membuat system persamaan linier dari matrik jarak yang diperoleh
- Menghitung solusi persamaan linier sebagai titik lokasi dari target dengan menggunakan algoritma TDoA

3.4 Desain software

Perancangan model simulasi sistem penentuan lokasi menggunakan Teknik TDoA dengan MATLAB *Graphical User Interface* (GUI) agar lebih interaktif dan *User friendly*. Dalam hal ini, *User* dapat memasukkan variasi nilai inputan untuk mendapatkan hasil yang berbeda. Sehingga UI yang dikembangkan terdapat Parameter Masukan, dalam hal ini sebagai inputan antara lain adalah jumlah BS, jumlah MU, dan pergerakan MU.



Gambar 3. 4 Contoh Gambar GUI

3.4.1 Spesifikasi Sub Sistem

Spesifikasi sub sistem yang digunakan untuk memodelkan sistem, antara lain:

1. *Transmitter*, didalamnya *Data Source*, Jenis Modulasi
2. Model Kanal, AWGN
3. *Receiver*, didalamnya terdapat Jenis Demodulasi

3.4.2 Parameter

Dalam penelitian simulasi sistem membutuhkan parameter masukan sebagai berikut:

1. *Data Source* = *Biner*, Simbol dengan parameter $M = \text{modulo Simbol}$.
2. Jenis Modulasi = BPS, QPSK, QAM
3. Model Kanal AWGN dengan parameter SNR
4. Posisi *receiver*
5. Sinyal dari target

Dalam simulasi yang dirancang juga dimungkinkan untuk beberapa skenario model simulasi.

3.5 Parameter Performasi Sistem

Sebagai bahan analisa dalam penelitian ini, maka parameter kinerja atau performansi dari simulasi sistem antara lain adalah:

1. Selisih atau *error* posisi estimasi dengan posisi yang sebenarnya dalam persentase maupun dalam jarak (meter).
2. *Running time* dari *program* simulasi.

IV. HASIL DAN ANALISIS

4.1 Hasil Implementasi

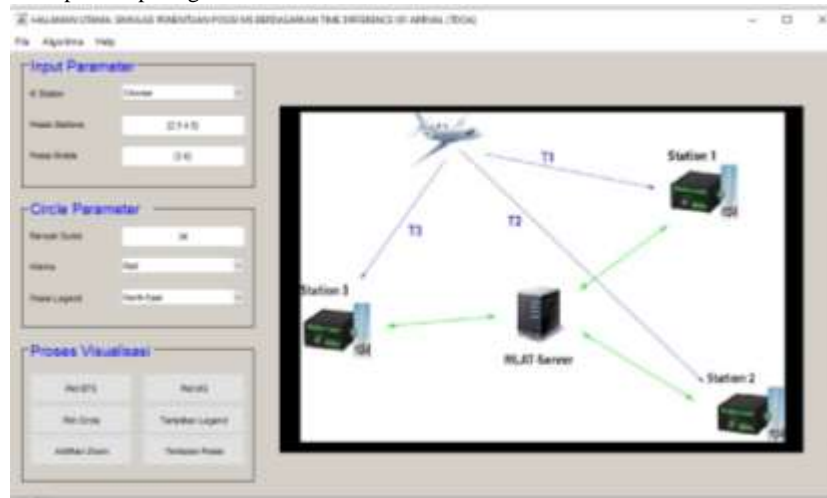
Untuk menjalankan program dari hasil Simulasi Penentuan Posisi Target Menggunakan *Teknik Time Difference of Arrival* (TDoA) dengan MATLAB dapat dilakukan dengan cara mengetikkan perintah pada command Matlab sebagai berikut:

`cd`

`D:\tdoa_program (1)`

`tdoa_gue.m`

Sehingga diperoleh tampilan seperti gambar berikut:

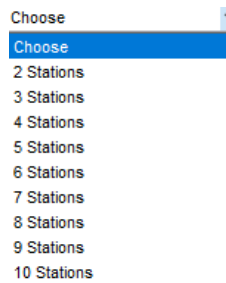


Gambar 4. 1 Halaman Utama

4.2 Hasil Pengujian

Untuk pengujian dari simulasi Penentuan Posisi Target Menggunakan *Teknik Time Difference of Arrival* (TDoA) dengan MATLAB ini dilakukan dengan tahapan sebagai berikut:

1. Menentukan jumlah Base Transceiver Station (BTS) dalam hal ini untuk simulasi penentuan posisi target di luar ruangan dimodelkan dengan Access Point (AP). Dalam hal ini dapat ditentukan dengan pilihan sebagai berikut:



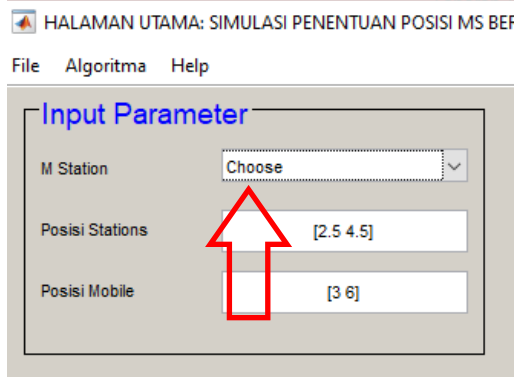
Gambar 4. 2 Jumlah BTS

Pilihan jumlah BTS yang bisa diinputkan adalah 1,2,3, dan 4 BTS.

Untuk menentukan lokasi station penulis mengatur masing masing *stations* sebagai berikut

- Station 1 = [2.5 4.5]
- Stations 2 = [2.5 4.5;6.5 8.5;]
- Stations 3 = [2.5 4.5;6.5 8.5; 2.5 8.5]
- Stations 4 = [2.5 4.5; 6.5 8.5; 2.5 8.5; 6.5 4.5]
- Stations 5= [2.5 4.5; 6.5 8.5; 2.5 8.5; 6.5 4.5; 8.5 4.5]
- Stations 6= [2.5 4.5; 6.5 8.5; 2.5 8.5; 6.5 4.5; 8.5 4.5; 4.8 9.5]
- Stations 7 =[2.5 4.5; 6.5 8.5; 2.5 8.5; 6.5 4.5; 8.5 4.5; 4.8 9.5; 9.4 4.5]
- Stations 8 =[2.5 4.5; 6.5 8.5; 2.5 8.5; 6.5 4.5; 8.5 4.5; 4.8 9.5; 9.4 4.5; 8.5 7.5]');
- Stations 9 : [2.5 4.5; 6.5 8.5; 2.5 8.5; 6.5 4.5; 8.5 4.5; 4.8 9.5; 9.4 4.5; 8.5 7.5; 10.5 9.5]');
- Stations 10: [2.5 4.5; 6.5 8.5; 2.5 8.5; 6.5 4.5; 8.5 4.5; 4.8 9.5; 9.4 4.5; 8.5 7.5; 10.5 9.5; 11.5 11.5]

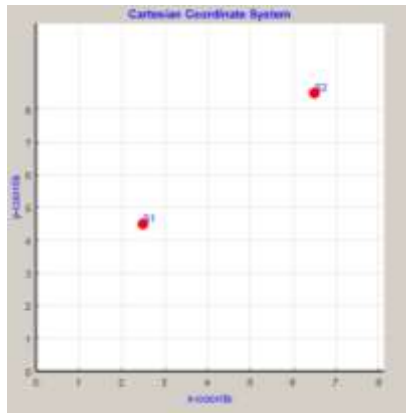
2. Menentukan titik BTS



Gambar 4. 3 Titik BTS

Untuk mengubah titik BTS dapat dilakukan dengan mengganti angka pada gambar 4.3 dengan angka yang lainnya.

Untuk menampilkan pada bidang kartesian, dapat diklik tombol Plot BTS, sehingga diperoleh tampilan sebagai berikut:



Gambar 4. 4 Tampilan Ketika tombol plot BTS di klik

3. Menentukan titik Mobile Station (MS) yang menjadi target penentuan Posisi, dalam hal ini dapat dilakukan dengan dua cara:

a. Mengetikkan titik dalam koordinat (x,y)

Gambar 4. 5 Menentukan Titik Ms

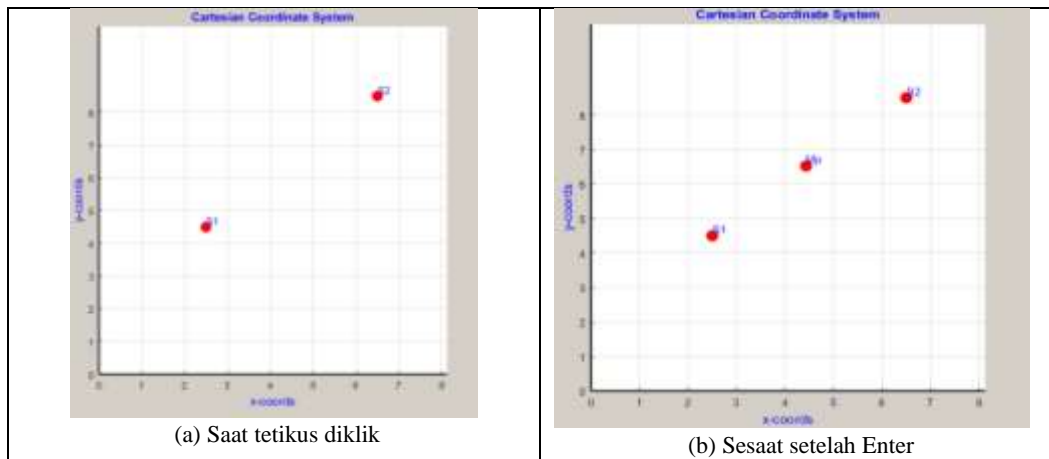
Untuk mengganti atau mengubah titik posisi MS dapat dilakukan dengan mengganti nilai koordinat pada gambar 4.5.

b. Menentukan titik dengan tetikus (mouse) dengan mengklik pada bidang kartesian 2 dimensi, yang dilanjutkan dengan enter.

Gambar 4.6 Titik MS

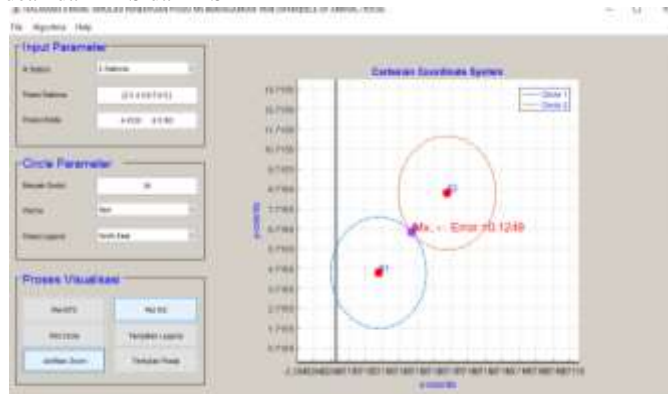
Titik posisi yang diperoleh setelah menentukan titik MS dengan tetikus dan menekan tombol enter.

4. Menampilkan titik BTS dan MS pada koordinat kartesian



Gambar 4. 6 proses menentukan titik MS dengan tetikus

5. Menghitung Jarak Euclidean dari BTS dan MS



Gambar 4. 7 Jarak antara BTS dan MS

Untuk menghitung jarak dua titik dapat dilakukan dengan menggunakan rumus jarak Euclidean. Sehingga pencarian posisi target dilakukan di sepanjang lingkaran dengan radius sama dengan jarak BTS dan MS tersebut dengan tahapan sebagai berikut:

1. menghitung jarak, d dengan rumus Euclidean berikut:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$
2. menentukan titik terdekat dengan MS dengan membuat titik-titik sepanjang lingkaran dengan radius, d . Dari gambar 4.7. dimisalkan jumlah titik sepanjang lingkaran adalah 36.
3. menghitung error atau selisih atau jarak antara titik-titik tersebut dengan titik MS.
4. mencari titik terdekat dengan MS, dengan mencari titik yang errornya paling kecil, itulah titik yang dianggap sebagai titik MS.

6. Menghitung Perbedaan Waktu Kedatangan Sinyal (TDOA)

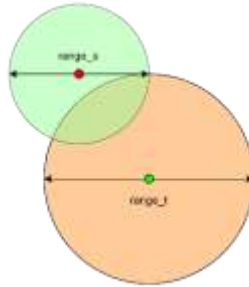
Sedangkan proses yang dilakukan untuk menentukan posisi target dengan TDOA dilakukan dengan cara sebagai berikut:

1. Menentukan cakupan atau range, baik range dari BTS (dalam hal ini dianggap sebagai sensor), dan range dari target atau MS. Dalam hal ini menggunakan parameter atau variabel:

$$\text{range}_s = 30$$

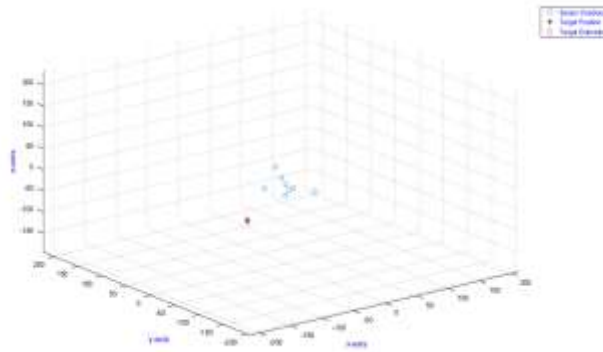
$$\text{range}_t = 100$$

range_s merupakan radius maksimal yang dapat digunakan sensor untuk mencari obyek, sedangkan range_t merupakan radius maksimal yang dapat dijangkau oleh sinyal Target.



Gambar 4. 8 Cangkupan sensor dan cangkupan target

2. Menentukan Jumlah Sensor atau jumlah BTS
M bervariasi dari 2 sampai dengan 10
3. Menentukan secara random waktu kedatangan (TOA)
Untuk menghitung waktu kedatangan (TOA) dilakukan dengan cara sebagai berikut:
 - a. Menentukan Vektor posisi dari sensor dengan rumus:
 $P = \text{range_s} * 2 * (\text{rand}(3,1) - 0.5);$
Berdasarkan gambar 4.8, maka sebaran posisi sensor P berada pada 2 kali radius atau range_s, dengan metode penyebarannya dilakukan secara acak atau random.
 - b. Menentukan Vektor posisi dari Target dengan rumus:
 $p_T = \text{range_t} * 2 * (\text{rand}(3,1) - 0.5)$
Berdasarkan gambar 4.8 pula, maka sebaran posisi target p_T berada pada 2 kali radius atau range_t, dengan metode penyebarannya dilakukan secara acak atau random.
 - c. Membuat Matrik dummy untuk semua kemungkinan dengan variasi P dan p_T, yang dilakukan dengan mengulang posisi p_T sebanyak jumlah sensor M untuk dikurangi dengan posisi sensor P.
 $\text{dummy} = \text{repmat}(p_T, 1, M) - P;$
dummy merupakan selisih atau jarak antara p_T dengan P dari masing-masing sensor.
 - d. Menghitung waktu kedatangan (TOA):
Waktu kedatangan dihitung dengan cara membagi variasi jarak, dummy dengan kecepatan rambat gelombang, dalam hal ini menggunakan variabel, c, sehingga TOA dihitung dengan operasi berikut:
 $\text{toa}(\text{ii}) = \text{norm}(\text{dummy}(:, \text{ii})) / c;$
toa yang didapat sebanyak jumlah sensor. Jika M = 10, maka terdapat 10 nilai toa.
4. Menghitung selisih waktu kedatangan (TDOA) dari masing-masing sensor dengan menggunakan rumus sebagai berikut:
 $\text{tdoa} = \text{toa} - \text{toa}(1); \text{tdoa}(1) = [];$
 $\text{tdoa} = \text{tdoa} + \text{err_std} * \text{randn}(M-1, 1);$
tdoa dihitung dengan mengurangkan toa sensor 1, toa(1) dengan toa sensor 2, toa(2) dan seterusnya. Sebagai koreksinya ditambahkan standar deviasi error secara random.
5. Menentukan posisi target dengan solusi persamaan linier dengan menggunakan *Root Mean Squared Error (RMSE)* untuk mendapatkan nilai terdekat dengan target, sehingga diperoleh posisi target seperti ditunjukkan pada gambar 4.8.
 $A = 2 * [(p_1(1) - \text{dummy}(:, 1)), (p_1(2) - \text{dummy}(:, 2)), (p_1(3) - \text{dummy}(:, 3)), -c * \text{tdoa}];$
 $b = (c * \text{tdoa})^2 + \text{norm}(p_1)^2 - \text{sum}((\text{dummy}^2), 2);$
 $x_lin = \text{pinv}(A) * b;$
 $\text{rmse}(k) = \text{norm}(p_T - x_lin(1:3))^2;$



Gambar 4. 9 Menentukan Posisi dengan TDoA

4.4.1 Pengaruh Jumlah Sensor/BTS

Untuk menguji performansi dari simulasi yang dihasilkan, maka dapat dihitung error atau selisih antara posisi target yang dihasilkan dari TDOA dengan posisi target yang sebenarnya, seperti yang ditunjukkan pada tabel 4.1. Tabel 4. 1 Pengaruh Jumlah Sensor (BTS) 1

Tabel 4. 1 Pengaruh Jumlah Sensor (BTS) Terhadap Error

M	Trials	e_x	e_y	e_z	Rata-rata Error (meter)
2	1	13.849	-15.006	-188.166	141.207 m
	2	-118.053	-234.964	-39.642	
	3	268.735	288.270	50.429	
	4	89.439	254.289	-20.357	
	5	-226.151	151.660	107.709	
	6	271.080	198.433	9.554	
	7	0.717	-222.827	157.473	
	8	193.528	97.679	112.577	
	9	269.312	282.986	35.454	
	10	-108.385	157.833	51.638	
3	1	56.716	-36.209	37.017	147.841 m
	2	-134.513	201.408	107.162	
	3	229.107	149.842	301.604	
	4	57.649	-83.831	293.690	
	5	-25.823	55.405	-291.376	
	6	24.174	-287.448	-17.861	
	7	-34.645	254.058	-283.857	
	8	244.467	285.693	65.259	
	9	-43.422	-114.071	-298.994	
	10	-107.944	125.110	-186.866	
4	1	227.272	-23.147	-26.190	130.425 m
	2	-227.306	-31.715	205.784	
	3	219.303	33.606	184.233	

4	4	117.687	179.666	-0.347	
4	5	172.598	117.678	174.095	
4	6	-97.450	-192.359	-202.598	
4	7	48.261	-153.772	-33.339	
4	8	148.762	255.542	-81.610	
4	9	200.278	160.746	-73.195	
4	10	-192.924	57.336	-73.957	
5	1	0.071	-0.030	0.033	1.262 m
5	2	-0.422	0.306	-0.444	
5	3	0.384	0.017	0.368	
5	4	-0.095	0.040	0.106	
5	5	-1.870	-1.565	-0.527	
5	6	-0.114	-0.102	0.180	
5	7	-0.052	-3.144	-4.900	
5	8	0.501	0.564	-0.054	
5	9	0.125	-0.263	0.159	
5	10	7.271	6.110	8.047	
6	1	-1.001	-0.164	0.301	0.580 m
6	2	0.212	0.173	0.095	
6	3	-3.558	-0.398	4.622	
6	4	-0.116	0.061	0.018	
6	5	-0.330	0.359	0.232	
6	6	0.641	-0.113	-0.042	
6	7	0.967	-0.900	-0.237	
6	8	0.781	-0.094	0.864	
6	9	0.188	0.442	0.151	
6	10	-0.170	0.125	-0.037	
7	1	0.000	0.301	0.451	0.632 m
7	2	-3.305	3.082	-1.204	
7	3	-0.207	-0.406	-1.047	
7	4	0.160	0.570	-0.295	
7	5	-1.485	-0.095	-1.385	
7	6	0.046	-0.048	-0.001	
7	7	0.036	-0.046	0.252	
7	8	0.124	0.027	0.026	
7	9	-0.360	-0.669	-3.078	
7	10	0.111	-0.020	0.133	
8	1	0.052	-0.005	-0.098	0.485 m
8	2	0.297	0.862	-0.377	
8	3	0.555	-0.434	-0.407	

8	4	0.391	0.964	-0.002	
8	5	-1.102	0.952	0.378	
8	6	-0.242	-0.174	0.145	
8	7	-0.146	0.766	-0.935	
8	8	-1.467	0.753	1.116	
8	9	-0.359	0.367	0.801	
8	10	-0.322	-0.073	0.007	
9	1	-1.379	-1.066	-0.805	0.288 m
9	2	0.007	0.083	0.043	
9	3	1.183	-0.448	-0.263	
9	4	-0.049	-0.028	0.070	
9	5	-0.017	0.057	-0.064	
9	6	0.122	-0.227	0.046	
9	7	-0.283	-0.082	-0.248	
9	8	0.081	-0.142	0.046	
9	9	-0.411	0.472	0.330	
9	10	0.207	0.174	0.197	
10	1	-0.378	0.491	-0.508	0.243 m
10	2	-0.753	-0.608	0.376	
10	3	0.242	0.830	0.615	
10	4	0.025	0.024	-0.067	
10	5	-0.335	0.331	0.195	
10	6	-0.180	0.052	-0.131	
10	7	-0.081	0.035	-0.050	
10	8	0.201	-0.322	0.047	
10	9	0.015	0.108	0.131	
10	10	0.006	0.025	-0.119	

Pengujian dilakukan untuk setiap nilai M = jumlah sensor, diuji sebanyak 10 kali pengujian, dan hasilnya dirata-ratakan.

4.4.2 Pengaruh Jumlah Sensor (BTS) Terhadap Running Time

Pengaruh jumlah Sensor atau BTS terhadap lama waktu proses dari program atau running time, dapat dilihat pada tabel 4.2. Pengujian dilakukan untuk setiap nilai M = jumlah sensor, dan masing-masing diuji sebanyak 10 kali pengujian, dan hasilnya dirata-ratakan.

Tabel 4. 2 Pengaruh Jumlah Sensor (BTS) Terhadap Running Time

M (N Sensor)	Trials	RunTime	Rata-rata (detik)
2	1	0.00623	0.016
	2	0.01259	
	3	0.01608	
	4	0.01657	
	5	0.01722	

	6	0.01741	
	7	0.01753	
	8	0.01766	
	9	0.01779	
	10	0.01791	
3	1	0.02506	0.028
	2	0.02805	
	3	0.02826	
	4	0.02845	
	5	0.02864	
	6	0.02881	
	7	0.02898	
	8	0.02916	
	9	0.02932	
	10	0.02949	
4	1	0.03069	0.032
	2	0.03097	
	3	0.03119	
	4	0.03140	
	5	0.03159	
	6	0.03176	
	7	0.03194	
	8	0.03210	
	9	0.03226	
	10	0.03243	
5	1	0.03370	0.035
	2	0.03386	
	3	0.03406	
	4	0.03437	
	5	0.03453	
	6	0.03467	
	7	0.03480	
	8	0.03496	
	9	0.03508	
	10	0.03538	
6	1	0.03596	0.037
	2	0.03617	
	3	0.03637	
	4	0.03651	
	5	0.03663	
	6	0.03676	
	7	0.03687	

	8	0.03696	
	9	0.03705	
	10	0.03714	
7	1	0.03811	0.039
	2	0.03823	
	3	0.03836	
	4	0.03855	
	5	0.03868	
	6	0.03888	
	7	0.03907	
	8	0.03921	
	9	0.03937	
	10	0.03954	
8	1	0.04006	0.041
	2	0.04019	
	3	0.04032	
	4	0.04044	
	5	0.04126	
	6	0.04165	
	7	0.04198	
	8	0.04226	
	9	0.04248	
	10	0.04270	
9	1	0.04348	0.044
	2	0.04368	
	3	0.04390	
	4	0.04410	
	5	0.04428	
	6	0.04445	
	7	0.04462	
	8	0.04486	
	9	0.04506	
	10	0.04526	
10	1	0.04595	0.047
	2	0.04621	
	3	0.04645	
	4	0.04669	
	5	0.04690	
	6	0.04712	
	7	0.04733	
	8	0.04760	
	9	0.04798	

	10	0.4828	
--	----	--------	--

4.4.3 Jarak Target dengan Sensor (BTS)

Proses penentuan posisi target obyek dari posisi sensor dilakukan dengan cara menghitung TOA, TDOA dan selanjutnya menghitung jarak, antara target obyek dengan posisi sensor-sensor yang ada, dalam hal ini menggunakan jumlah sensor atau M=8. Sehingga diperoleh tabel 4.3.

Tabel 4.3 Jarak Target Obyek dengan Sensor (BTS)

Posisi Target			Posisi Sensor			Jarak (m)	Path Loss (dB)
x	Y	Z	x	y	z		
-137.21	-124.07	-203.26	-19.49	-23.45	15.78	268.25	88.62
-137.21	-124.07	-203.26	-23.67	-9.07	29.81	283.62	89.11
-137.21	-124.07	-203.26	-11.66	-18.51	6.55	266.32	88.56
-137.21	-124.07	-203.26	18.27	15.36	-15.82	280.62	89.02
-137.21	-124.07	-203.26	19.46	9.56	-0.84	288.75	89.26
-137.21	-124.07	-203.26	-9.86	-27.14	15.88	271.36	88.72
-137.21	-124.07	-203.26	22.01	-4.46	9.77	291.62	89.35
-137.21	-124.07	-203.26	-12.47	-24.70	-2.87	256.11	88.22

4.4.4 Pengaruh Perulangan terhadap Jarak

Pengujian selanjutnya adalah Pengaruh Perulangan terhadap Jarak Target Obyek dengan Sensor (BTS) dalam satuan meter, dengan nilai perulangan=5, 10, dan 15. Hasil pengujian ini seperti ditunjukkan pada tabel 4.4.

Tabel 4.4 Pengaruh Perulangan terhadap Jarak Target Obyek dengan Sensorn(BTS)

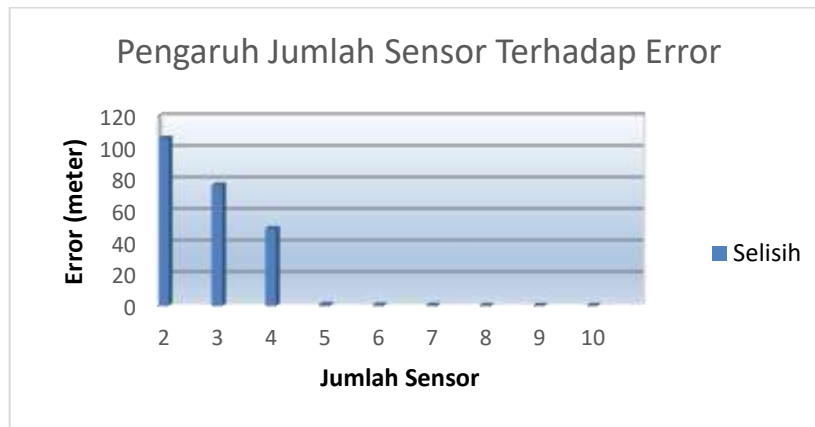
M	Jarak (meter)					
	Ulang=5		Ulang=10		Ulang=15	
	TDOA	MTDOA	TDOA	MTDOA	TDOA	MTDOA
2	129.643	120.563	63.224	53.221	60.429	55.139
3	130.667	117.813	59.072	43.328	63.136	55.126
4	131.325	101.176	57.855	34.231	61.822	42.057
5	0.412	0.073	0.136	0.025	1.039	0.025
6	0.172	0.034	0.048	0.015	0.044	0.012
7	0.138	0.034	0.056	0.013	0.037	0.011
8	0.063	0.022	0.034	0.007	0.025	0.010
9	0.056	0.020	0.017	0.009	0.034	0.008
10	0.106	0.019	0.012	0.007	0.020	0.008

4.3 Analisa Data

Berdasarkan data-data pengujian yang dilakukan untuk menentukan posisi target dengan menggunakan Time Difference of Arrival (TDOA) ini dapat dianalisa sebagai berikut:

4.3.1 Pengaruh Jumlah Sensor Terhadap Selisih (Error)

Dengan TDOA pada BTS (sensor) dapat ditentukan posisi target MS dengan tingkat kesalahan rata-rata sebesar 48.92654 untuk M=4, dan kesalahan rata-rata sebesar 0.432441 untuk M=8, jumlah sensor atau jumlah BTS.

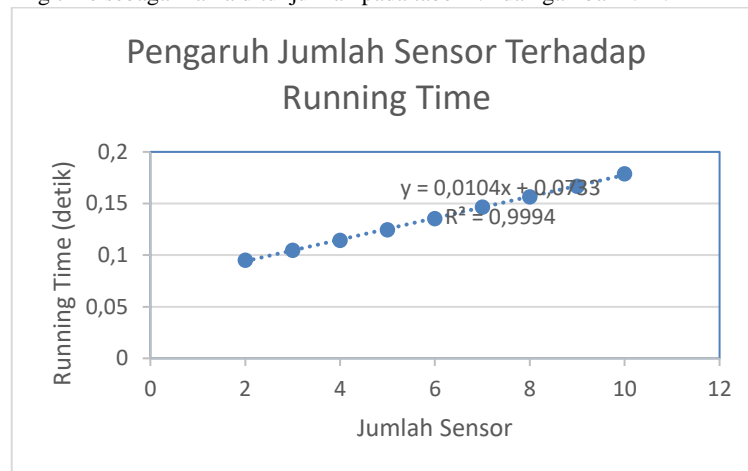


Gambar 4.10 Pengaruh Jumlah Sensor pada Error

Semakin banyak jumlah BTS semakin kecil kesalahan atau selisih posisi target Dari TDOA dengan posisi yang sesungguhnya Dan sebaliknya semakin sedikit jumlah sensor semakin besar kesalahan yang diperoleh Sebagaimama ditunjukkan oleh tabel 4.1 dan gambar 4.10.

4.3.2 Pengaruh Jumlah Sensor Terhadap Running Time

Dari data pengujian sub-Bab sebelumnya, diperoleh informasi bahwa semakin banyak jumlah sensor posisi yang didspat lebih mendekati posisi target yang ditunjukkan dengan error yang semakin kecil. Namun jumlah sensor mempengaruhi running time sebagaimana ditunjukkan pada tabel 4.2 dan gambar 4.11.

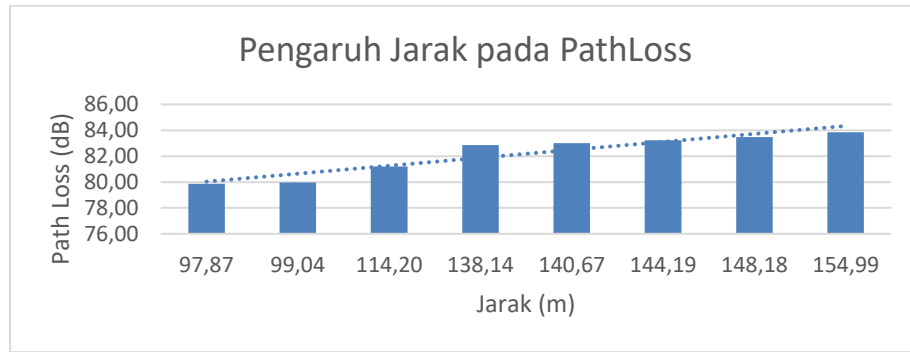


Gambar 4.11 Pengaruh jumlah sensor terhadap running time

Semakin banyak jumlah BTS (sensor) semakin lama waktu yang diperlukan untuk proses nopencarian posisi target, sebaliknya semakin sedikit jumlah sensor semakin kecil waktu proses yang diperlukan.

4.3.3 Pengaruh Jarak Terhadap PathLoss (PL)

Data pengujian yang dilakukan untuk mengetahui pengaruh Jarak pada PathLoss adalah seperti ditunjukkan pada gambar 4.12.



Gambar 4.12 Pengaruh Jarak pada PathLoss

Data pengujian tersebut diatas sesuai dengan rumus PathLoss yaitu:

$$PL = 20\log_{10}(4 \cdot \pi \cdot \text{jarak} / \lambda)$$

Dengan merujuk data tersebut maka dapat diketahui bahwa semakin besar jarak dari posisi target dengan sensor semakin besar PathLoss yang diperoleh sebagaimana ditunjukkan pada gambar 4.12. Dan sebaliknya semakin kecil jarak antara sensor dengan target, Maka semakin kecil pula rugi-rugi yang dialami oleh sinyal transmisi.

4.3.4 Pengaruh Perulangan (Trials) terhadap Jarak

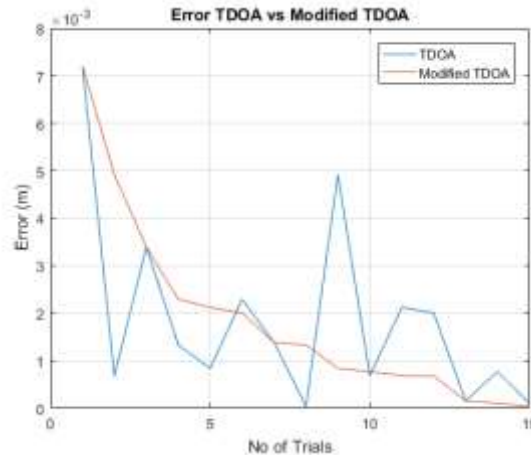
Berdasarkan data pengujian pada tabel 4.4. menunjukkan bahwa semakin banyak jumlah sensor menyebabkan Jarak Obyek Target dengan hasil posisi dari TDOA semakin kecil, yang menunjukkan bahwa semakin banyak sensor menyebabkan hasil penentuan lokasi obyek semakin presisi. Sedangkan hasil pengujian berdasarkan jumlah perulangan menunjukkan bahwa untuk perulangan (Trials)=10 menghasilkan data posisi yang lebih presisi dibandingkan Trials=5 dan Trials=15. Hal ini berarti bahwa nilai yang terbaik tidak selalu dengan perulangan yang paling besar. Sedangkan perbandingan antara data pengujian TDOA dengan Modified TDOA (MTDOA) menunjukkan bahwa MTDOA memberikan hasil yang lebih baik atau lebih presisi dibandingkan dengan TDOA saja.

4.4 Algoritma Modified TDOA

Data hasil pengujian menggunakan TDOA dengan nilai Trials=1, jika menggunakan Trials dengan nilai yang lebih dari satu, maka hasil penentuan lokasi target adalah dari Trials yang terakhir. Dari data juga menunjukkan bahwa hasil Trials yang terakhir belum tentu hasil yang terbaik. Oleh sebab itu, untuk nilai Trials lebih besar satu, menggunakan Algoritma Modified TDOA, dimana algoritma ini:

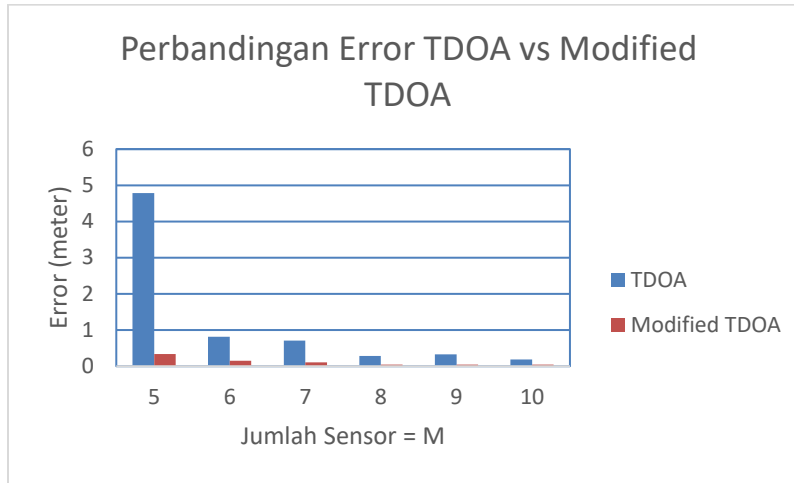
1. memerlukan buffer untuk menyimpan sementara error dari setiap iterasi (trials).
2. melakukan sorting error dari yang terbesar ke yang terkecil.
3. menentukn posisi target dengan error terkecil.

Hasil dari algoritma Modified TDOA adalah sebagaimana ditunjukkan pada gambar 4.13.



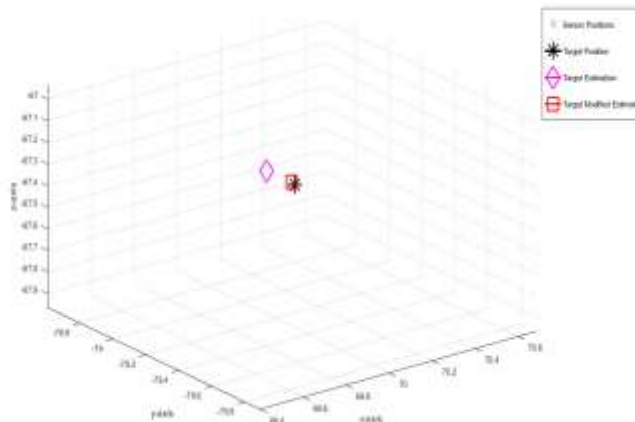
Gambar 4.13 Grafik Error TDoA vs Modified TDoA

Grafik 4.13 menunjukkan bahwa Modified TDOA mencari dan menentukan titik terdekat posisi target dengan mencari kuadrat error terkecil. Hasil ini sejalan dengan perbandingan rata-rata kesalahan antara TDOA dan Modified TDOA sebagaimana ditunjukkan pada gambar 4.14.



Gambar 4. 14 Rata-rata error TDOA vs Modified TDoA

Perbedaan atau selisih terbesar antara algoritma TDOA dan modified TDOA adalah 62.14 meter, dan selisih terkecil adalah 0.14 meter. Gambar 4.15 merupakan tampilan 3 dimensi antara posisi target, estimasi target dengan TDOA dan estimasi target dengan Modified TDOA yang nampak jelas bahwa modified TDOA menghasilkan estimasi yang lebih dengan dengan posisi target dibandingkan dengan algoritma TDOA.



Gambar 4. 15 Posisi Target, Estimasi TDoA, dan Estimasi Modified TDoA

V. KESIMPULAN

5.1 Kesimpulan

Berdasarkan data-data pengujian dan analisa yang telah dilakukan untuk menentukan posisi target dengan menggunakan Time Difference of Arrival (TDOA) dapat ditarik kesimpulan sebagai berikut:

1. Dengan TDOA pada BTS (Sensor) dapat ditentukan posisi target MS dengan tingkat kesalahan terbesar 230.03 meter, dan kesalahan terkecil adalah 0.185 meter.
2. Semakin banyak jumlah BTS semakin kecil kesalahan atau selisih posisi target Dari TDOA dengan posisi yang sesungguhnya.
3. Semakin banyak jumlah BTS (sensor) semakin lama proses pencarian posisi target.

4. Running time yang diperlukan untuk menjalankan program simulasi TDOA ini antara 0.01 sampai 0.18 detik.
5. Dengan algoritma modified TDOA mampu memperbaiki kesalahan terbesar 62.14 meter atau 72.986% dan terkecil adalah 0.14 meter atau 24.324%.

REFERENCES

- [1] A. Sandi and N. Neviyarni, "Ingatan II : Pengorganisasian, Lupa dan Model-Model Ingatan," *Edukatif J. Ilmu Pendidik.*, vol. 3, no. 1, pp. 115–123, 2021, doi: 10.31004/edukatif.v3i1.191.
- [2] B. Prasetya, U. A. Tatang, and D. Arseno, "Penentuan Posisi User Pada Sistem Komunikasi Seluler Dengan Metoda Time of Arrival (Toa) Dan Time Difference of Arrival (Tdoa)," *Semin. Nas. Inform. 2008 (semnasIF 2008)*, vol. 2008, no. semnasIF, pp. 335–343, 2008.
- [3] S. M. Sheikh, H. M. Asif, K. Raahemifar, and F. Al-Turjman, "Time Difference of Arrival Based Indoor Positioning System Using Visible Light Communication," *IEEE Access*, vol. 9, pp. 52113–52124, 2021, doi: 10.1109/ACCESS.2021.3069793.
- [4] P. Wang, H. B. Ji, L. Liu, and W. B. Zhang, "Data Association Based on Multidirection-Ordered Association in AOA," *Tien Tzu Hsueh Pao/Acta Electron. Sin.*, vol. 49, no. 3, 2021, doi: 10.12263/DZXB.20181053.
- [5] J. J. Pérez-Solano, S. Ezpeleta, and J. M. Claver, "Indoor localization using time difference of arrival with UWB signals and unsynchronized devices," *Ad Hoc Networks*, vol. 99, Mar. 2020, doi: 10.1016/j.adhoc.2019.102067.
- [6] A. Garg and A. Gupta, "Indoor tracking using BLE-brief survey of techniques," *Conf. Int. Conf. Data Sci. Mach. Learn. (ICDSML-2020) At Amritsar Coll. Eng. Technol. Amritsar, India*, no. March, 2020, [Online]. Available: <https://www.researchgate.net/publication/346962774>
- [7] J. Zhang, F. Gao, Y. Li, and X. Wu, "Simulation of multilateration system based on Chan algorithm and conjugate gradient optimisation algorithm," *Int. J. Simul. Process Model.*, vol. 14, no. 5, 2019, doi: 10.1504/IJSPM.2019.104117.
- [8] A. Naeem, N. U. Hassan, M. A. Pasha, C. Yuen, and A. Sikora, "Performance analysis of TDOA-based indoor positioning systems using visible LED Lights," 2018. doi: 10.1109/IDAACS-SWS.2018.8525567.
- [9] Y. Miftahuddin, S. Umaroh, and F. R. Karim, "Perbandingan Metode Perhitungan Jarak Euclidean, Haversine, Dan Manhattan Dalam Penentuan Posisi Karyawan," *J. Tekno Insentif*, vol. 14, no. 2, pp. 69–77, 2020, doi: 10.36787/jti.v14i2.270.
- [10] T. O. Hodson, "Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not," *Geosci. Model Dev.*, vol. 15, no. 14, pp. 5481–5487, 2022, doi: 10.5194/gmd-15-5481-2022.
- [11] R. Snm, "Kuliah 2 Analisis Numerik : Metoda Numerik untuk Solusi Persamaan Linear".
- [12] S. Kurt and B. Tavli, "Path-Loss Modeling for Wireless Sensor Networks: A review of models and comparative evaluations," *IEEE Antennas Propag. Mag.*, vol. 59, no. 1, pp. 18–37, 2017, doi: 10.1109/MAP.2016.2630035.
- [13] K. Putri, "Pengertian Teknologi Informasi, Serta Tujuan dan Fungsinya," <https://Teknologi.Id/Insight/Pengertian-Teknologi-Informasi-Serta-Tujuan-Dan-Fungsinya/>. 2018.
- [14] A. Oulasvirta, N. R. Dayama, M. Shiripour, M. John, and A. Karrenbauer, "Combinatorial Optimization of Graphical User Interface Designs," *Proc. IEEE*, vol. 108, no. 3, pp. 434–464, 2020, doi: 10.1109/JPROC.2020.2969687.
- [15] L. Keviczky, R. Bars, J. Hetthéssy, and C. Bányász, "Introduction to MATLAB," *Advanced Textbooks in Control and Signal Processing*. 2019. doi: 10.1007/978-981-10-8321-1_1.
- [16] M. R. Noviansyah, W. Suharso, D. R. Chandranegara, M. S. Azmi, and M. Hermawan, "Sistem Pendukung Keputusan Pemilihan Laptop Pada E-Commerce Menggunakan Metode Weighted Product," *Pros. SENTRA (Seminar Teknol. dan Rekayasa)*, vol. 0, no. 5, 2019.
- [17] S. D'Alu, H. Rivano, and O. Simonin, "TDoA for In-Flight Relative Localization in UAV Swarm using Ultra-Wide Band," *CEUR Workshop Proc.*, vol. 3581, 2023.
- [18] J. Pospisil, R. Fujdiak, and K. Mikhaylov, "Investigation of the performance of tdoa-based localization over lorawan in theory and practice," *Sensors (Switzerland)*, vol. 20, no. 19, pp. 1–22, 2020, doi: 10.3390/s20195464.
- [19] M. Aernouts, N. BniLam, R. Berkvens, and M. Weyn, "TDAoA: A combination of TDOA and AoA localization with LoRaWAN," *Internet of Things (Netherlands)*, vol. 11, 2020, doi: 10.1016/j.iot.2020.100236.
- [20] J. Qu, H. Shi, N. Qiao, C. Wu, C. Su, and A. Razi, "New three-dimensional positioning algorithm through integrating TDOA and Newton's method," *Eurasip J. Wirel. Commun. Netw.*, vol. 2020, no. 1, 2020, doi: 10.1186/s13638-020-01684-7.
- [21] R. Muppala *et al.*, "Feasibility of Standalone TDoA-based Localization Using LoRaWAN To cite this version : HAL Id : hal-03288706 Feasibility of Standalone TDoA-based Localization Using LoRaWAN," 2021.