

PENGEMBANGAN SISTEM POSISI INDOOR BERGERAK MENGUNAKAN TRILATERASI DAN FILTER KALMAN



Disusun Oleh:

N a m a : Fathan Firmansyah
NIM : 20523006

**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
2024**

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**PENGEMBANGAN SISTEM POSISI INDOOR BERGERAK
MENGUNAKAN TRILATERASI DAN FILTER KALMAN**

TUGAS AKHIR



المعهد الإسلامي
Yogyakarta, 8 Juli 2024

Pembimbing 1,

Pembimbing 2,

(Fayruz Rahma, S.T., M.Eng.)

(Kurniawan Dwi Irianto, S.T., M.Sc.)

HALAMAN PENGESAHAN DOSEN PENGUJI

**PENGEMBANGAN SISTEM POSISI INDOOR BERGERAK
MENGUNAKAN TRILATERASI DAN FILTER KALMAN****TUGAS AKHIR**

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 8 Juli 2024

Tim Penguji

Fayruz Rahma, S.T., M.Eng.

Anggota 1

Erika Ramadhani, S.T., M.Eng.

Anggota 2

Rahadian Kurniawan, S.Kom., M.Kom.

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana
Fakultas Teknologi Industri
Universitas Islam Indonesia



(Ir. Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Fathan Firmansyah

NIM : 20523006

Tugas akhir dengan judul:

PENGEMBANGAN SISTEM POSISI INDOOR BERGERAK MENGUNAKAN TRILATERASI DAN FILTER KALMAN

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 8 Juli 2024



(Fathan Firmansyah)

HALAMAN PERSEMBAHAN

Segala puji bagi Allah Swt. atas segala nikmat yang telah diberikan kepada kita. Shalawat dan salam kita sampaikan kepada junjungan kita Nabi Muhammad saw. yang kita harapkan syafaatnya di hari akhir nanti.

Saya mengucapkan terima kasih yang sebesar-besarnya kepada kedua orang tua saya yang telah mengasuh dan mendidik saya sejak dalam kandungan hingga saat ini, serta kepada kedua adik saya yang telah memberikan dukungan baik secara moral maupun material. Semoga kita semua senantiasa diberikan kesehatan, kebahagiaan, dan umur yang panjang.

Terima kasih saya sampaikan kepada Dosen Pembimbing saya, Ibu Fayruz Rahma dan Pak Kurniawan Dwi Irianto, yang selalu dengan sabar melatih dan membimbing saya, serta selalu memperhatikan saya di setiap pertemuan bimbingan.

Terima kasih juga kepada semua pihak yang tidak dapat disebutkan satu per satu atas segala dukungan dan bantuannya, baik secara langsung maupun tidak langsung.

HALAMAN MOTO

“Terus Berjuang, Tak Kenal Menyerah”

KATA PENGANTAR

Assalamu'alaikum warahmatullahi wabarakatuh.

Dengan menyebut nama Allah Yang Maha Pengasih lagi Maha Penyayang, segala puji dan syukur penulis panjatkan ke hadirat Allah Swt. Berkat rahmat dan izin-Nya, penulisan skripsi berjudul "Pengembangan Sistem Penentuan Posisi Objek Bergerak dalam Ruang Menggunakan Trilaterasi dan Filter Kalman" ini dapat terselesaikan. Selawat dan salam tak lupa penulis sampaikan kepada Nabi Muhammad saw. Teladan bagi kita hingga akhir zaman.

Selama proses penulisan tugas akhir ini, penulis menyadari bahwa penyelesaiannya tidak mungkin dilakukan secara mandiri. Banyak pihak yang telah memberikan dukungan dari berbagai aspek. Oleh karena itu, penulis ingin menyampaikan rasa terima kasih kepada:

1. Orang tua dan kakak penulis yang telah memberikan dukungan serta doa kepada penulis.
2. Prof. Fathul Wahid, S.T., M.Sc., Ph.D. selaku Rektor Universitas Islam Indonesia.
3. Dr. Raden Teduh Dirgahayu, S.T., M.Sc. selaku Ketua Jurusan Informatika UII.
4. Ir. Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D. selaku Ketua Prodi Informatika – Program Sarjana UII.
5. Fayruz Rahma, S.T., M.Eng. selaku Dosen Pembimbing 1 yang selalu memberikan arahan, petunjuk, dan motivasi dalam menyelesaikan tugas akhir.
6. Kurniawan Dwi Irianto, S.T., M.Sc. selaku Dosen Pembimbing 2 yang memberikan masukan-masukan penting dalam penulisan tugas akhir.
7. Teman-teman seperjuangan, Arfi Wanda Safitri, Bintang Ananda, Muhammad Alwan Abdurra'uf, Muhammad Yoga Kurniawan, dan Bima Al Fatih Haryadina Putra yang selalu menemani dan membantu penulis dalam menyelesaikan tugas akhir.

Penulis menyadari bahwa terdapat banyak kekurangan dalam penyusunan tugas akhir ini. Namun, penulis tetap berharap bahwa tugas akhir ini dapat memberikan manfaat dan mungkin dapat dikembangkan menjadi sesuatu yang lebih besar di masa depan.

Yogyakarta, 8 Juli 2024



(Fathan Firmansyah)

SARI

Global Positioning System (GPS) adalah layanan yang sering digunakan untuk melakukan deteksi lokasi di lingkungan *outdoor*, namun GPS memiliki kelemahan dengan turunnya akurasi jika melakukan deteksi lokasi pada lingkungan *indoor* karena struktur bangunan. Dalam dua dekade terakhir, telah dilakukan berbagai upaya besar untuk meningkatkan teknologi deteksi lokasi di dalam ruangan atau *Indoor Positioning System* (IPS). Layanan IPS sangat dibutuhkan untuk menentukan lokasi barang, orang, atau bahkan perangkat *Internet of Things* (IoT) secara akurat di dalam ruangan. Penelitian ini mengembangkan sistem IPS menggunakan metode Trilaterasi dan Kalman Filter. Trilaterasi digunakan untuk menentukan lokasi, sementara Kalman Filter digunakan untuk menyaring sinyal yang diterima dari *Access Point* (AP). Penelitian ini melibatkan dua pengujian untuk mengevaluasi hasil sistem yang dikembangkan serta dampak penggunaan Kalman Filter pada IPS. Pengujian pertama dilakukan dalam kondisi statis, sedangkan pengujian kedua dilakukan dalam kondisi bergerak. Selain itu, pengujian juga dilakukan di dua ruangan berbeda dan dalam dua kondisi berbeda, yaitu ruangan kosong dan ruangan terisi. Pengujian ini juga mencakup pengubahan posisi AP untuk menentukan apakah letak AP memengaruhi hasil IPS. Hasil pengujian dalam kondisi statis menunjukkan bahwa Kalman Filter memberikan dampak positif dengan menurunkan tingkat kesalahan hasil IPS. Penurunan kesalahan berkisar antara 2,93% hingga 59,62%, tergantung pada titik pengujian. Pada kondisi bergerak, Kalman Filter juga membantu menurunkan tingkat kesalahan IPS. Kondisi ruangan, baik kosong maupun terisi, memengaruhi tingkat akurasi. Namun penelitian ini menunjukkan bahwa bentuk ruangan tidak terlalu berpengaruh pada akurasi hasil. Sebaliknya, posisi peletakan AP memiliki pengaruh yang lebih signifikan terhadap hasil IPS. Hasil penelitian ini diharapkan dapat memberikan pengetahuan yang dapat membantu peneliti lain dalam mengembangkan sistem IPS yang bisa digunakan untuk bergerak.

Kata kunci: Trilaterasi, Kalman Filter, *Indoor Positioning System*, *Access Point*.

GLOSARIUM

Access Point (AP)	Perangkat jaringan yang memungkinkan perangkat Wi-Fi untuk terhubung ke jaringan kabel.
Indoor Positioning System	Sistem yang digunakan untuk melacak dan mengidentifikasi lokasi objek atau orang di dalam ruangan.
Kalman Filter	Algoritma matematis yang menggunakan serangkaian pengukuran yang diamati dari waktu ke waktu, yang mengandung gangguan dan ketidakakuratan, untuk menghasilkan estimasi dari variabel-variabel yang tidak diketahui.
N Path Loss	Reduksi kekuatan sinyal saat bergerak melalui suatu medium dari pengirim ke penerima dalam komunikasi nirkabel.
Receiver	Perangkat atau komponen yang menerima sinyal atau data dari sumber pengirim.
RSSI	<i>Received Signal Strength Indicator</i> , pengukuran kekuatan sinyal yang diterima oleh perangkat penerima dalam jaringan nirkabel.
Transmitter	Perangkat atau komponen yang mengirimkan sinyal atau data ke penerima (receiver).
Trilaterasi	Metode untuk menentukan posisi suatu titik dengan mengukur jarak ke tiga titik referensi yang diketahui lokasinya.

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING.....	ii
HALAMAN PENGESAHAN DOSEN PENGUJI.....	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR.....	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTO	vi
KATA PENGANTAR	vii
SARI.....	viii
GLOSARIUM.....	ix
DAFTAR ISI.....	x
DAFTAR TABEL.....	xii
DAFTAR GAMBAR	xiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	3
1.5 Sistematika Penulisan.....	3
BAB II KAJIAN PUSTAKA.....	5
2.1 Kajian Teori.....	5
2.1.1 Sistem Posisi <i>Indoor (Indoor Positioning System - IPS)</i>	5
2.1.2 Estimasi Jarak (<i>Signal Propagation Model</i>).....	7
2.1.3 Menghitung Jarak Antar Titik (<i>Euclidean Distance</i>).....	8
2.1.4 Filter Kalman	8
2.1.5 Trilaterasi pada IPS.....	9
2.2 Kajian Literatur	12
BAB III METODE PENELITIAN	16
3.1 Alur Pengerjaan Tugas Akhir.....	16
3.1.1 Kajian Literatur	16
3.1.2 Perencanaan Eksperimen	17
3.1.3 Eksperimen.....	18
3.1.4 Dokumentasi Eksperimen dan Membuat Laporan.....	19

3.2	Alur Eksperimen.....	19
3.2.1	Mempersiapkan Lingkungan Eksperimen	20
3.2.2	Mengambil Nilai Rata-Rata RSSI AP.....	27
3.2.3	Menentukan Nilai N Path Loss AP	28
3.2.4	Melakukan Estimasi Posisi	28
3.3	Alur Kerja Sistem	29
3.3.1	Mengambil Nilai RSSI.....	30
3.3.2	Memproses RSSI dengan Kalman Filter.....	30
3.3.3	Mengubah Nilai RSSI dengan Model Propagasi Sinyal	30
3.3.4	Mengestimasi Posisi dengan Trilaterasi.....	30
BAB IV HASIL DAN PEMBAHASAN		31
4.1	Mengembangkan IPS Menggunakan Trilaterasi dan Kalman Filter	31
4.1.1	Mendapatkan Nilai RSSI	31
4.1.2	Mengolah Nilai RSSI Menggunakan Kalman Filter.....	32
4.1.3	Mencari Nilai Eksponen (N).....	33
4.1.4	Melakukan Estimasi Jarak (<i>Signal Propagation Model</i>)	33
4.1.5	Melakukan Estimasi Posisi Menggunakan Trilaterasi	34
4.1.6	Melakukan Plotting Data Trilaterasi (<i>Animate</i>)	35
4.2	Skenario Uji Coba	38
4.2.1	Skenario Uji Coba 1	38
4.2.2	Skenario Uji Coba 2.....	42
4.3	Diskusi.....	47
BAB V KESIMPULAN DAN SARAN		49
5.1	Kesimpulan.....	49
5.2	Saran.....	50
DAFTAR PUSTAKA		51
LAMPIRAN.....		54

DAFTAR TABEL

Tabel 2.1 Penelitian Terkait	13
Tabel 3.1 Perencanaan Skenario Uji Coba	18
Tabel 3.2 Spesifikasi laptop	27
Tabel 3.3 Spesifikasi Access Point	27
Tabel 4.1 Nilai N pada AP1	39
Tabel 4.2 Nilai N pada AP2	39
Tabel 4.3 Nilai N pada AP3	39
Tabel 4.4 Nilai perubahan trilaterasi statis	42
Tabel 4.5 Referensi RSSI Ruang 1 pada kondisi kosong	43
Tabel 4.6 Referensi RSSI Ruang 2 pada kondisi kosong	43
Tabel 4.7 Nilai N Ruang 1 pada kondisi kosong	43
Tabel 4.8 Nilai N Ruang 2 pada kondisi kosong	43
Tabel 4.9 Data penurunan eror pada Ruang 1	45
Tabel 4.10 Data penurunan eror pada Ruang 2	46

DAFTAR GAMBAR

Gambar 2.1 Berbagai teknologi yang digunakan pada IPS	6
Gambar 2.2 Metode trilaterasi untuk menemukan posisi objek	10
Gambar 3.1 Alur Pengerjaan Tugas Akhir	16
Gambar 3.2 Alur Eksperimen	20
Gambar 3.3 Ruang 1 posisi 1 (2D)	21
Gambar 3.4 Ruang 1 posisi 2 (2D)	21
Gambar 3.5 Ruang 2 posisi 1 (2D)	22
Gambar 3.6 Ruang 2 posisi 2 (2D)	22
Gambar 3.7 Ruang 1 posisi 1 dalam kondisi terisi	23
Gambar 3.8 Ruang 1 posisi 1 dalam kondisi kosong.....	23
Gambar 3.9 Ruang 1 posisi 2 dalam kondisi terisi	24
Gambar 3.10 Ruang 1 posisi 2 dalam kondisi kosong.....	24
Gambar 3.11 Ruang 2 posisi 1 dalam kondisi terisi	25
Gambar 3.12 Ruang 2 posisi 1 dalam kondisi kosong.....	25
Gambar 3.13 Ruang 2 posisi 2 dalam kondisi terisi	26
Gambar 3.14 Ruang 2 posisi 2 dalam kondisi kosong.....	26
Gambar 3.15 Alur Kerja Sistem Trilaterasi	29
Gambar 4.1 Kode untuk mendapatkan nilai RSSI.....	32
Gambar 4.2 Kode untuk menerapkan Kalman Filter.....	33
Gambar 4.3 Kode untuk mencari nilai eksponen.....	33
Gambar 4.4 Kode untuk mengubah RSSI menjadi Jarak.....	34
Gambar 4.5 Kode Trilaterasi untuk prediksi posisi.....	34
Gambar 4.6 Kode untuk melakukan plotting hasil trilaterasi.....	37
Gambar 4.7 Hasil perhitungan trilaterasi statis pada titik 1.....	40
Gambar 4.8 Hasil perhitungan trilaterasi statis pada titik 2.....	40
Gambar 4.9 Hasil perhitungan trilaterasi statis pada titik 3.....	41
Gambar 4.10 Hasil perhitungan trilaterasi statis pada titik 4.....	41
Gambar 4.11 Hasil pengujian IPS pada Ruang 1.....	44
Gambar 4.12 Hasil pengujian IPS pada Ruang 2.....	44
Gambar 4.13 Proses pengujian IPS dengan berjalan mengikuti pola yang ditentukan	45
Gambar 4.14 Grafik nilai eror IPS pada Ruang 1	45
Gambar 4.15 Grafik nilai eror IPS pada Ruang 2.....	46

BAB I

PENDAHULUAN

1.1 Latar Belakang

Sistem Pemosisian Global (*Global Positioning System - GPS*) menyediakan informasi posisi dan waktu yang tepat di luar ruangan (Sohan et al., 2019). Namun, sinyal GPS melemah secara signifikan di dalam ruangan karena terhalang oleh struktur bangunan, yang menyebabkan konektivitas tidak dapat diandalkan dan akurasi berkurang (Sohan et al., 2019). Masalah ini penting untuk diatasi karena individu menghabiskan banyak waktu di dalam ruangan menggunakan jaringan nirkabel. Sistem Penentuan Posisi dalam Ruangan (*Indoor Positioning System - IPS*) digunakan untuk menavigasi layanan lokasi di dalam ruangan untuk menemukan barang, orang, atau perangkat *Internet of Things* (IoT) di dalam gedung atau ruang tertutup lainnya (Sohan et al., 2019).

IPS memiliki berbagai penerapan praktis yang memperluas fungsinya dalam kehidupan sehari-hari. Sebagai contoh, dalam perawatan kesehatan, IPS dapat digunakan untuk melacak pergerakan pasien lansia di panti jompo, memastikan bahwa mereka tidak memasuki area berbahaya dan membantu staf dalam mengawasi mereka dengan lebih efektif. Di industri pertambangan, IPS membantu dalam pelacakan real-time lokasi para pekerja di dalam tambang, yang sangat penting untuk respons cepat dalam kasus keadaan darurat seperti kebocoran gas atau runtuhnya terowongan.

Di lingkungan akademis, universitas dapat memanfaatkan IPS untuk sistem absensi otomatis yang memastikan mahasiswa hadir di kelas tanpa perlu absensi manual, yang seringkali memakan waktu. Selain itu, di pusat perbelanjaan besar, IPS dapat digunakan untuk membantu pengunjung menemukan toko atau produk tertentu dengan mudah, serta untuk manajemen keamanan dalam melacak keberadaan staf dan pengunjung secara real-time. Kemudian di tempat rekreasi seperti museum, IPS bisa digunakan sebagai sistem yang interaktif dengan pengunjung, sehingga pengunjung bisa mendapatkan informasi yang sesuai dengan tempat dimana dia berada. Dengan kemampuan untuk menyediakan data lokasi yang akurat dan tepat waktu, IPS menjadi alat yang sangat berharga dalam memperkuat keamanan, efisiensi, dan kenyamanan dalam berbagai setting kehidupan sehari-hari.

IPS adalah sebuah sistem yang digunakan untuk memprediksi posisi terkini seseorang atau sebuah objek di dalam ruangan (Wibisono & Wicaksono, 2022a). Dalam dua dekade

terakhir, telah dilakukan usaha besar dalam meningkatkan teknologi pendeteksian lokasi di dalam ruangan. Hasil dari penelitian terdahulu adalah ketersediaan komersial beberapa IPS yang sudah diimplementasikan serta sejumlah sistem yang masih sedang diuji dan diperbaiki (Batistic & Tomic, 2018). Terdapat berbagai teknik yang digunakan untuk memprediksi titik lokasi, salah satunya adalah teknik trilaterasi (Batistic & Tomic, 2018). Dalam penelitian ini, tiga titik yang memiliki koordinat yang diketahui serta jaraknya terhadap objek tertentu digunakan untuk mengestimasi posisi objek tersebut di dalam ruangan. Dalam konteks IPS yang diteliti, tiga titik ini merujuk kepada lokasi *access points* (APs), sementara objek yang diprediksi lokasinya adalah laptop.

Untuk memungkinkan prediksi lokasi yang akurat dalam IPS, komunikasi antara laptop dan APs menggunakan gelombang elektromagnetik yang merambat melalui udara. Pengukuran jarak antara laptop dan APs memerlukan kekuatan sinyal dan frekuensi tertentu. Terkadang, sinyal ini tidak stabil karena berbagai kondisi (Mari, Kiong, & Loong, 2018a). Sebagai contoh, ada situasi di mana halangan seperti dinding atau objek lainnya di sekitar perangkat dapat memantulkan sinyal, mengakibatkan adanya sinyal ganda atau variasi dalam kekuatan sinyal yang diterima. Ketidakstabilan sinyal juga dapat muncul ketika energi perangkat tidak konsisten. Pada akhirnya, semua hal ini berdampak pada ketepatan prediksi lokasi.

Dalam upaya untuk memprediksi lokasi, teknik trilaterasi digunakan sebagai metode utama. Teknik ini melibatkan penggunaan tiga titik akses (APs) untuk memperkirakan posisi objek berdasarkan jarak yang terukur dari masing-masing AP. Namun, ketidakstabilan sinyal yang telah disebutkan sebelumnya dapat menyebabkan kesalahan dalam pengukuran jarak ini, yang pada gilirannya mengakibatkan estimasi posisi yang tidak akurat. Oleh karena itu, diperlukan metode tambahan untuk mengurangi kesalahan ini dan meningkatkan keakuratan prediksi lokasi.

Teknik trilaterasi menggunakan tiga APs untuk memperkirakan posisi. Kesalahan dalam mengukur salah satu jarak dapat mengakibatkan kesalahan dalam estimasi posisi. Filter Kalman merupakan salah satu algoritma yang mampu mengurangi ketidakakuratan sinyal, yang membantu menghasilkan prediksi lokasi objek yang lebih tepat (Fariz et al., 2018a). Beberapa penelitian telah menggabungkan algoritma filter Kalman dan trilaterasi untuk menentukan lokasi suatu objek (Fariz et al., 2018a), (Andika Satrugna Mahardhika, Gautama Putra, & Abdurrohman, 2020a), (Wibisono & Wicaksono, 2022b). Meskipun demikian, penelitian sebelumnya hanya menghasilkan prediksi lokasi dalam konteks statis. Penelitian ini mencoba untuk mengembangkan sistem IPS yang bersifat *real-time*.

1.2 Rumusan Masalah

- a. Bagaimana mengembangkan sebuah sistem penentuan posisi objek dalam ruangan (*Indoor Positioning System*) yang bergerak menggunakan metode trilaterasi dan filter Kalman?
- b. Bagaimana dampak penggunaan filter Kalman pada sistem IPS yang bergerak?

1.3 Batasan Masalah

Penelitian yang efektif adalah penelitian yang memiliki tujuan yang jelas dan terfokus, yang memungkinkan hasil penelitian yang paling baik. Oleh karena itu, penting untuk mempertimbangkan beberapa pembatasan berikut:

- a. Penelitian ini akan membatasi diri pada pengembangan IPS yang menggunakan metode trilaterasi dan filter kalman.
- b. Pengujian dilakukan dengan hanya satu perangkat yang terhubung dengan AP (*Access Point*).
- c. Sistem yang dikembangkan akan fokus pada penggunaan dalam perangkat laptop.
- d. Sistem yang dikembangkan akan bisa digunakan dengan kondisi bergerak, sehingga mampu mengakomodasi perubahan posisi objek secara *real-time*.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk mengembangkan sebuah IPS yang bisa digunakan dalam menentukan posisi objek yang bergerak. IPS ini akan menghitung titik lokasi menggunakan metode trilaterasi, menstabilkan sinyal yang diterima melalui penggunaan filter Kalman, dan menerapkannya dalam konteks laptop. Selain itu, penelitian ini bertujuan untuk mengkaji dampak penggunaan filter Kalman pada IPS yang bergerak.

1.5 Sistematika Penulisan

Dalam penulisan laporan Tugas Akhir, struktur penyajian membantu peneliti dalam mengorganisir dokumen sesuai dengan standar penelitian yang membantu menjadikan penelitian lebih terarah. Berikut adalah kerangka laporan penulisan Tugas Akhir:

- a. BAB I PENDAHULUAN

Bab Pendahuluan membahas mengenai awal penulisan laporan penelitian. Bagian pendahuluan memuat Latar Belakang, Rumusan Masalah, Batasan Masalah, Tujuan Penelitian, dan Sistematika Penulisan.

b. **BAB II KAJIAN PUSTAKA**

Bagian ini memuat berbagai dasar kerangka pemikiran dalam penelitian, yang didapatkan dari studi pustaka untuk memperoleh solusi dari permasalahan yang ada. Bagian ini terdiri Kajian Teori dan Kajian Literatur.

c. **BAB III METODOLOGI**

Bagian metodologi membahas analisis teori yang diterapkan dalam penelitian serta cara mengimplementasikannya ke dalam sistem yang dikembangkan. Materi yang digunakan dalam penelitian ini akan dijelaskan sesuai dengan alur pengembangan yang dilakukan, serta penguraian perancangan sistem.

d. **BAB IV HASIL DAN PEMBAHASAN**

Bagian ini membahas secara rinci hasil penelitian yang dilaksanakan. Hasil penelitian akan disajikan dalam bentuk gambaran alur sistem, gambar kode, tabel, foto ataupun bentuk lainnya yang diuraikan sesuai dengan tahapan penelitian.

e. **BAB V KESIMPULAN DAN SARAN**

Kesimpulan dan saran merupakan bagian terakhir dari penulisan laporan yang membahas tentang kesimpulan hasil penelitian yang telah dilakukan dan berisi saran serta wawasan untuk penelitian selanjutnya yang berkaitan dengan topik bahasan pada penelitian ini.

BAB II

KAJIAN PUSTAKA

Pada era digital yang semakin maju, kebutuhan akan sistem penentuan posisi dalam ruangan (Indoor Positioning System - IPS) telah menjadi semakin mendesak. Bab ini akan membahas kajian teori yang menjadi landasan penting dalam pengembangan IPS, termasuk teori-teori utama mengenai IPS, teknik trilaterasi, dan algoritma filter Kalman. Selain itu, kajian literatur mengenai penelitian-penelitian sebelumnya yang telah mengintegrasikan trilaterasi dan filter Kalman dalam konteks IPS akan dieksplorasi. Pemahaman mendalam tentang aspek-aspek teoritis ini akan memberikan wawasan yang komprehensif, mempersiapkan dasar yang kuat untuk mengembangkan sistem IPS yang lebih canggih dan efektif.

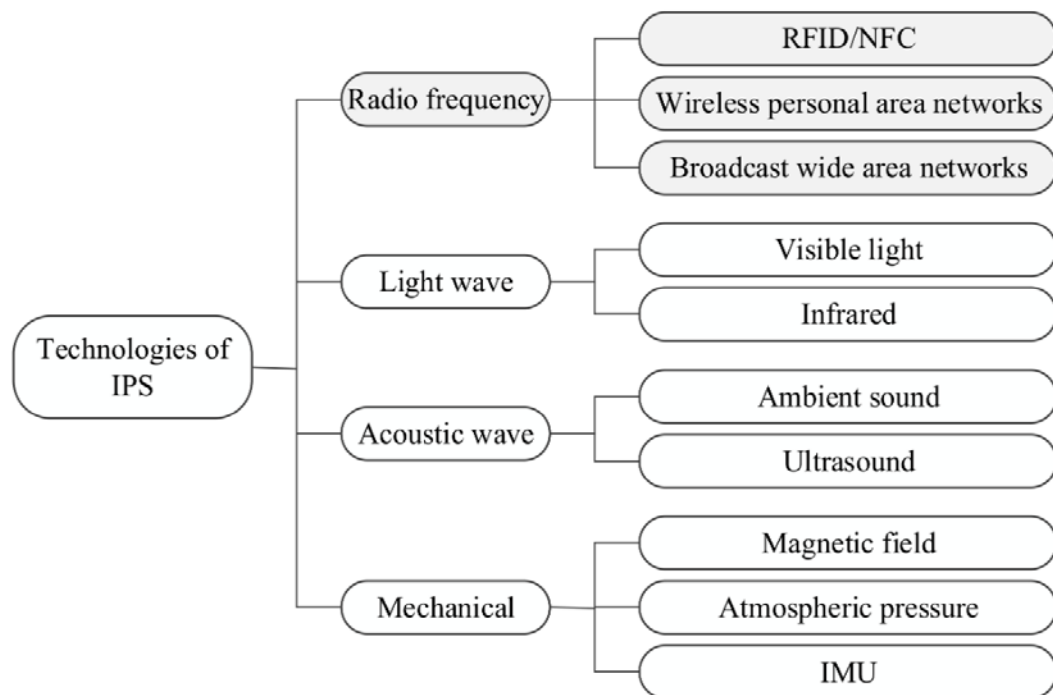
2.1 Kajian Teori

Seiring dengan kemajuan teknologi dan perkembangan kebutuhan akan penentuan posisi dalam ruangan, berbagai teori dan konsep telah berkembang dalam literatur terkait (Batistic & Tomic, 2018). Selain itu, penelitian ini juga akan mengeksplorasi berbagai penelitian terkait yang telah mengintegrasikan trilaterasi dan filter Kalman dalam konteks IPS. Dalam bab ini, landasan teoritis yang mendukung pendekatan ini serta beberapa studi kasus yang mengilustrasikan implementasinya akan diperlihatkan. Teori lain yang digunakan dan berkaitan juga dibahas salah satunya adalah *Signal Propagation Model*, digunakan untuk mengukur jarak AP dengan perangkat yang digunakan. Kemudian *Euclidian Distance* sebagai rumus untuk menghitung jarak antara 2 titik.

2.1.1 Sistem Penentuan Posisi *Indoor* (*Indoor Positioning System - IPS*)

Gelombang radio dipengaruhi utamanya oleh dua peristiwa, yaitu penurunan kekuatan sinyal seiring jarak yang ditempuh dalam atmosfer dan respon dari setiap objek yang berinteraksi dengan sinyal tersebut (Lassabe, Canalda, Chatonnay, Spies, & Baala, 2005). Peristiwa pertama berkaitan dengan penurunan intensitas sinyal sejalan dengan jarak dari sumber pemancar. Sebaliknya, peristiwa kedua melibatkan penurunan atau peningkatan intensitas sinyal sebagai akibat dari pemantulan, respon, dan atenuasi. Dalam penelitian ini, model penyebaran sinyal digunakan untuk lingkungan dalam ruangan (Lassabe et al., 2005).

Berbagai teknologi telah diterapkan dalam pengembangan IPS, seperti yang dapat dilihat pada Gambar 2.1.



Gambar 2.1 Berbagai teknologi yang digunakan pada IPS

Sumber: (Subedi & Pyun, 2020)

Sistem IPS yang dikembangkan pada penelitian ini merupakan bagian dari *Wireless Personal Area Networks*, merujuk pada penelitian Subedi & Pyun (2020) yang berfokus pada cabang *Radio Frequency-Wireless Personal Area Networks* (RF-WPAN) karena menggunakan APs sebagai pemancar sinyal. RF-WPAN digunakan untuk mengembangkan sistem penentuan posisi dalam ruangan (IPS) dengan memanfaatkan sifat-sifat unik gelombang radio. Dalam konteks ini, model penyebaran sinyal yang digunakan memperhitungkan penurunan kekuatan sinyal seiring dengan jarak dan interaksi sinyal dengan objek-objek di sekitar. Hal ini memungkinkan deteksi dan prediksi posisi yang lebih akurat, sesuai dengan kebutuhan aplikasi dalam ruangan. Teknologi RF-WPAN memungkinkan komunikasi yang efektif antara perangkat, seperti laptop dan APs, yang mendukung pengembangan IPS yang lebih canggih dan andal.

Kemudian AP yang digunakan pada penelitian ini termasuk sebagai teknologi yang digunakan sebagai pemancar sinyal. Selain itu, terdapat teknologi yang dipertimbangkan juga sebagai penerima sinyal. *Smartphones* dan *Personal Computer* (PC) diketahui sebagai 2 perangkat yang paling banyak digunakan juga oleh peneliti sebagai perangkat penerima sinyal

(Firmansyah, Rahma, Irianto, & Shiddiqi, 2024). Perangkat *smartphones* tersebut dipilih oleh para peneliti dengan mempertimbangkan bahwa penggunaan IPS nantinya banyak penggunaan *smartphones*, sedangkan PC dipilih karena peneliti sedang melakukan pengembangan sistem IPS. Selanjutnya metode yang sering digunakan untuk penentuan lokasi oleh para peneliti terdahulu adalah Trilaterasi dan *Fingerprinting* (Firmansyah et al., 2024). *Fingerprinting* sendiri menggunakan dua tahap yaitu *offline phase* dan *online phase*. Tahap *offline* adalah tahap untuk melakukan *mapping* lokasi kemudian menyimpan data tersebut ke *database*. Selanjutnya tahap *online* adalah tahap penggunaan sistem IPS yang kemudian membandingkan nilai tersebut ke *database* yang sudah dibuat sebelumnya. Hal lain yang dipertimbangkan dalam pengembangan IPS adalah teknik *signal filtering*, pada penelitian terdahulu diketahui jika kalman filter dan particle filter terpilih sebagai metode yang paling banyak digunakan oleh peneliti (Firmansyah et al., 2024). Particle filter sendiri memiliki sifat nonlinear, sifat tersebut mengartikan jika perubahan dalam nilai input tidak menghasilkan perubahan yang proporsional atau langsung dalam output.

2.1.2 Estimasi Jarak (*Signal Propagation Model*)

Jarak antara perangkat dan titik akses diperoleh dengan menggunakan model propagasi sinyal berdasarkan kekuatan sinyal yang telah distabilkan oleh filter. Propagasi sinyal (N) memerlukan eksperimen lapangan untuk menentukan nilainya yang diukur pada jarak satu meter dari pemancar (Wibisono & Wicaksono, 2022b).

$$RSSI = -(10n \log_{10} d + A) \quad (2.1)$$

$$d = 10^{\frac{A-RSSI}{10n}} \quad (2.2)$$

$$n = -\left(\frac{RSSI - A}{10 \log_{10} d}\right) \quad (2.3)$$

Variabel-variabel pada (2.1), (2.2), dan (2.3) meliputi RSSI sebagai kekuatan sinyal yang diterima oleh perangkat, n adalah konstanta propagasi sinyal, d adalah jarak antara titik akses dan perangkat, dan A adalah sinyal RSSI yang dikumpulkan pada jarak 1 meter antara perangkat dan titik akses.

2.1.3 Menghitung Jarak Antar Titik (*Euclidean Distance*)

Rumus jarak Euclidean digunakan untuk menghitung jarak antara dua titik dalam bidang dua dimensi. Dengan mempertimbangkan dua titik, A dan B, rumus ini (2.4) menghitung jarak langsung (jarak garis lurus) antara kedua titik tersebut.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (2.4)$$

Dapat dijelaskan jika d adalah jarak antara titik A dan B, kemudian (x_1, y_1) adalah koordinat titik A, dan (x_2, y_2) adalah koordinat titik B.

2.1.4 Filter Kalman

Algoritma filter Kalman, yang sering disebut juga dengan nama Linear Quadratic Estimation (LQE), adalah suatu algoritma yang memanfaatkan hasil pengamatan atau pengukuran selama periode waktu tertentu yang dapat mencakup gangguan, dan menghasilkan estimasi yang cenderung lebih akurat. Filter Kalman bertujuan untuk mengurangi varians kesalahan estimasi. Oleh karena itu, algoritma filter Kalman merupakan salah satu metode yang tepat untuk digunakan dalam upaya penentuan posisi (Fariz et al., 2018a).

Kalman Filter memiliki dua proses utama, yaitu prediksi dan koreksi (Andika Satrugna Mahardhika et al., 2020a). Fungsi prediksi adalah untuk menghitung nilai estimasi dari suatu keadaan (*state*) berdasarkan nilai-nilai sebelumnya. Koreksi berfungsi untuk memperbaiki nilai estimasi tersebut agar menghasilkan nilai prediksi yang lebih akurat.

Seperti yang terlihat pada (2.5), (2.6), (2.7), (2.8), (2.9) terdapat beberapa istilah dalam perhitungan Kalman Filter. X adalah matriks keadaan (*state matrix*). P adalah matriks kovarians proses (*process covariance matrix*). A dan B adalah matriks adaptasi yang mengonversi keadaan menjadi input keadaan proses. U adalah matriks variabel kontrol. Q adalah matriks kovarians *noise* proses. W adalah matriks *noise* keadaan yang diprediksi. K adalah *gain*. R adalah matriks kovarians pengukuran (*noise sensor*).

Persamaan untuk *prediction* filter Kalman.

$$X_{k-1} = AX_{k-1} + BU_k + W_k \quad (2.5)$$

$$P_k = P_{k-1} + Q_k \quad (2.6)$$

Persamaan untuk melakukan *correction*.

$$K = P_k(P_k + R)^{-1} \quad (2.7)$$

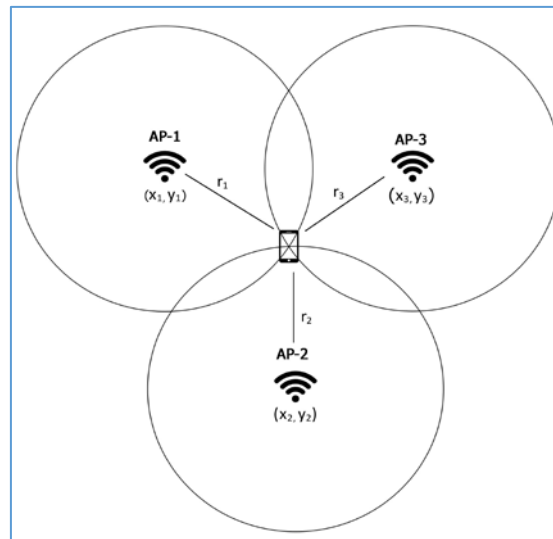
$$X_k = X_k + K(Y_k - X_k) \quad (2.8)$$

$$P_k = (1 - K)P_k \quad (2.9)$$

Kalman Filter adalah alat penting dalam pemrosesan data dan estimasi yang dapat digunakan dalam berbagai aplikasi, termasuk sistem Python. Dalam penelitian ini, implementasi Kalman Filter di Python dapat dilakukan menggunakan perpustakaan "filterpy". Dengan mengimpor Kalman Filter, pengguna dapat memanfaatkan kemampuan Kalman Filter untuk mengestimasi variabel-variabel keadaan suatu sistem berdasarkan serangkaian pengamatan yang terkontaminasi oleh kebisingan. Proses ini melibatkan prediksi keadaan berikutnya dan memperbarui estimasi tersebut berdasarkan data pengamatan baru, sehingga menghasilkan estimasi yang lebih akurat dan dapat diandalkan. Filter ini sangat efektif dalam berbagai aplikasi, seperti pelacakan objek, navigasi, dan pengendalian sistem bergerak.

2.1.5 Trilaterasi pada IPS

IPS yang bergantung pada Access Point (AP) menggunakan metode Received Signal Strength (RSS) untuk menghitung jarak antara AP dan laptop. Dalam perkiraan ini, RSS dimodelkan dengan mengacu pada penyebaran sinyal. Trilaterasi adalah suatu teknik yang digunakan untuk menentukan posisi suatu objek berdasarkan jaraknya dari beberapa titik dengan lokasi yang sudah diketahui (Pakanon, Chamchoy, & Supanakoon, 2020). Setiap AP memiliki cakupan komunikasi yang mampu mendeteksi perangkat berbasis WiFi (lihat Gambar 2.2). Berdasarkan kekuatan sinyal, dapat diperkirakan posisi perangkat berbasis WiFi yang berada dalam jangkauan ketiga AP yang memiliki koordinat yang telah diketahui (Pakanon et al., 2020).



Gambar 2.2 Metode trilaterasi untuk menemukan posisi objek

Sumber: (Wibisono & Wicaksono, 2022b)

Pada Gambar 2.2, ilustrasi dari proses trilaterasi dalam konteks penentuan posisi dalam ruangan menggunakan tiga AP dapat dilihat. Setiap AP ditandai dengan simbol Wifi dan memiliki koordinat tertentu dalam bidang dua dimensi. Koordinat (x_1, y_1) , (x_2, y_2) , (x_3, y_3) mewakili posisi dari masing-masing AP dalam sistem koordinat kartesian dua dimensi. (x_1, y_1) adalah koordinat dari AP-1, (x_2, y_2) adalah koordinat dari AP-2, dan (x_3, y_3) adalah koordinat dari AP-3. Koordinat ini menunjukkan lokasi tetap dari setiap AP yang akan digunakan sebagai referensi dalam proses trilaterasi. Selanjutnya r_1, r_2, r_3 adalah radius atau jarak yang mengindikasikan jarak antara perangkat yang akan ditentukan posisinya (ditandai dengan gambar *handphone* di pusat gambar) dengan masing-masing AP. Nilai-nilai jarak ini diperoleh melalui pengukuran sinyal yang diterima oleh perangkat dari setiap AP.

Dalam proses trilaterasi, posisi perangkat yang tidak diketahui dihitung dengan menggunakan jarak r_1, r_2, r_3 dari masing-masing AP dan posisi tetap dari AP ((x_1, y_1) , (x_2, y_2) , (x_3, y_3)). Dengan menggambarkan lingkaran-lingkaran yang berpusat di setiap AP dan memiliki radius sesuai dengan jarak yang diukur, titik perpotongan dari ketiga lingkaran ini akan menentukan posisi perangkat tersebut. Proses ini memanfaatkan prinsip dasar dari trilaterasi, yaitu menentukan posisi suatu titik dengan menggunakan jarak dari tiga titik referensi yang diketahui posisinya.

Pada tahap estimasi posisi, akan dilakukan trilaterasi untuk memperoleh estimasi posisi dari penerima. Dalam proses trilaterasi, koordinat x dan y dari penerima akan diperoleh dari persamaan jarak berikut (2.20), (2.21) (Sujana, 2018).

$$r_1^2 = (x - x_1)^2 + (y - y_1)^2 \quad (2.10)$$

$$r_2^2 = (x - x_2)^2 + (y - y_2)^2 \quad (2.11)$$

$$r_3^2 = (x - x_3)^2 + (y - y_3)^2 \quad (2.12)$$

Dari persamaan jarak tersebut (2.10), (2.11), (2.12), masing-masing dijelaskan secara terperinci.

$$r_1^2 = x^2 - 2x_1x + x_1^2 + y^2 - 2y_1y + y_1^2 \quad (2.13)$$

$$r_2^2 = x^2 - 2x_2x + x_2^2 + y^2 - 2y_2y + y_2^2 \quad (2.14)$$

$$r_3^2 = x^2 - 2x_3x + x_3^2 + y^2 - 2y_3y + y_3^2 \quad (2.15)$$

Kurangkan persamaan (2.13) dengan persamaan (2.14) dan kurangkan persamaan (2.14) dengan persamaan (2.15) untuk menyederhanakan sistem persamaan.

$$r_1^2 - r_2^2 - x_1^2 + x_2^2 - y_1^2 + y_2^2 = (-2x_1 + 2x_2)x + (-2y_1 + 2y_2)y \quad (2.16)$$

$$r_3^2 - r_4^2 - x_3^2 + x_4^2 - y_3^2 + y_4^2 = (-2x_3 + 2x_4)x + (-2y_3 + 2y_4)y \quad (2.17)$$

Substitusi menggunakan variabel A, B, C untuk persamaan (2.16) dan D, E, F untuk persamaan (2.17).

$$Ax + By = C \quad (2.18)$$

$$Dx + Ey = F \quad (2.19)$$

Berdasarkan persamaan (2.18), dan (2.19) didapatkan solusi dari trilaterasi.

$$x = \frac{CE - FB}{EA - BD} \quad (2.20)$$

$$y = \frac{CD - AF}{BD - AE} \quad (2.21)$$

2.2 Kajian Literatur

Subbab ini akan menggali lebih dalam kajian literatur yang relevan dengan penelitian ini. Kajian literatur merupakan tonggak penting dalam setiap penelitian karena membantu memahami pengetahuan yang telah ada tentang topik penelitian. Dalam konteks ini, berbagai sumber literatur yang berfokus pada konsep-konsep yang relevan dengan IPS, teknik trilaterasi, dan algoritma filter Kalman akan dijelajahi.

Pentingnya kajian literatur adalah untuk memahami kerangka kerja dan teori yang telah ada, mengidentifikasi kekosongan pengetahuan, serta melihat kontribusi penelitian ini terhadap pemahaman yang ada. Selain itu, penelitian-penelitian terdahulu yang relevan dengan topik ini juga akan diulas. Ini akan memberikan pandangan yang lebih luas tentang evolusi penelitian dalam domain IPS dan cara pendekatan yang digunakan dalam penelitian ini sejalan dengan penelitian sebelumnya. Melalui pemahaman yang mendalam tentang literatur yang telah ada, kerangka penelitian ini akan dapat dirancang dan dijelaskan dengan lebih baik.

Tabel 2.1 Penelitian Terkait

Penulis	Teknologi	Metode	Kelebihan	Kekurangan	Keterangan
(Wang et al., 2023a)	WiFi	Geomagnetic Signals, Pedestrian Dead Reckoning, Particle Filter	Menggabungkan teknologi WiFi, PDR, dan Geomagnetic Signals. Kemudian menggunakan Particle Filter untuk mengurangi eror/meningkatkan akurasi	Pengukuran hanya dilakukan pada satu jenis ukuran ruangan yaitu 24x12 m	Gabungan metode-metode tersebut mampu meningkatkan akurasi yang lebih tinggi dari pada tidak digabung
(Spachos & Plataniotis, 2020a)	BLE (Beacon)	Trilaterasi, Kalman Filter	Keunggulan dari sistem ini adalah simpel, biaya rendah, dan instalasi yang mudah	Akurasi dapat turun drastis ketika ada gangguan atau perubahan yang signifikan di lingkungan tersebut	Penelitian ini dilakukan pada museum agar dapat menjadi <i>smart</i> museum
(Mari et al., 2018a)	WiFi	Trilaterasi, Fingerprinting, Machine Learning (KOS-ELM)	Akurasi yang didapat dari penggabungan tiga metode tersebut menunjukkan hasil yang baik	Aplikasi pada tempat yang padat cenderung lebih susah karena harus melakukan <i>scanning</i> ulang untuk menghasilkan akurasi yang lebih baik	Butuh pembelajaran lebih lanjut mengenai Machine Learning tersebut
(Fariz et al., 2018a)	WiFi	Trilaterasi, Kalman Filter	Filter Kalman menunjukkan hasil yang baik dengan meningkatkan akurasi titik lokasi perangkat	Pengujian dilakukan dalam kondisi statis, dan hanya diuji pada 1 titik saja.	Pengujian dilakukan di ruangan yang terhalang dinding
(Sohan et al., 2019)	WiFi	Trilaterasi, ITU <i>path loss</i> (Model yang digunakan dalam <i>indoor positioning</i> yang memikirkan hambatan-hambatannya)	Penggunaan ITU <i>path loss</i> dapat meningkatkan akurasi jika digunakan di dalam ruangan ruangan yang bersekat atau banyak objek	Perbandingan hasil sebelum penggunaan ITU <i>path loss</i> dan sesudahnya tidak ditunjukkan dengan detail	Peneliti memberikan masukan untuk penggunaan kalman filter ke depannya

(Pakanon et al., 2020)	BLE Beacons	Trilaterasi	Menunjukkan secara detail nilai akurasi yang didapat dari beberapa pengukuran	Pengukuran hanya berfokus pada kondisi statis	Semakin banyak pengukuran menunjukkan semakin tinggi akurasi yang didapat
(Andika Satrugna Mahardhika et al., 2020a)	WiFi	Trilaterasi, Kalman Filter, Feedback Filter	Menunjukkan perbandingan hasil dari tiap-tiap metode yang digunakan	Pengukuran lebih membandingkan hasil dari filter, jadi hanya dalam kondisi statis	Kalman Filter menunjukkan akurasi yang lebih baik dari yang lainnya
(Bottigliero, Milanesio, Saccani, & Maggiora, 2021)	UWB	Trilaterasi, Mean Filter, Median Filter	Menggunakan Time Different of Arrival untuk atributnya dan menggunakan Trilateration untuk metodenya	Tidak mencoba perbandingan dengan metode lain	Hasil penelitian menunjukkan akurasi mencapai angka 10cm
(Wibisono & Wicaksono, 2022b)	WiFi	Trilaterasi, Kalman Filter	Menjelaskan detail hasil penggunaan Kalman Filter	Tidak dijelaskan kondisi pengukuran, Tidak dibandingkan antara sebelum menggunakan kalman dan sesudahnya	Dengan melakukan kalibrasi <i>signal propagation model</i> maka akurasi menjadi lebih baik
(Chouhan et al., 2022a)	WiFi	Trilaterasi	Menunjukkan penggunaannya dalam gambar 2D, hasil juga cukup baik	Penelitian berfokus pada pencarian titik AP apakah sudah akurat atau belum, tidak meneliti akurasi prediksi posisi dengan jelas dan hanya dilakukan dalam kondisi statis.	Di ruangan persegi (188cm x 188cm) yang kosong, 2 wifi di atas, 1 wifi di bawah

Dari penelitian Tabel 2.1, dapat ditarik kesimpulan di antaranya:

- a. Metode trilaterasi menonjol sebagai salah satu metode yang relatif mudah dalam pengaplikasiannya. Ini melibatkan perhitungan jarak dengan mengacu pada tiga titik referensi, yang umumnya didukung oleh perangkat dengan biaya relatif rendah. Beberapa poin kunci tentang trilaterasi:
 1. Pengaplikasiannya sederhana dan mudah diimplementasikan.

2. Biaya implementasi cenderung terjangkau.
 3. Memerlukan tiga titik referensi yang memiliki koordinat yang diketahui.
- b. Kalman filter adalah algoritma yang dianggap mampu meningkatkan akurasi dalam penentuan posisi. Ia berfungsi untuk meminimalkan kesalahan estimasi dan mengatasi ketidakstabilan sinyal. Beberapa poin kunci tentang Kalman filter:
1. Berperan penting dalam meningkatkan akurasi estimasi posisi.
 2. Mampu mengatasi ketidakpastian dan variabilitas dalam data sinyal.
 3. Menghasilkan prediksi yang lebih akurat, terutama dalam lingkungan yang penuh dengan gangguan atau perubahan.
- c. Penelitian yang sudah dilakukan sebelumnya berfokus pada pengembangan sistem IPS yang pengujiannya dilakukan dalam kondisi statis (*receiver* diam), sistem yang dikembangkan dalam kondisi statis tersebut belum bisa digunakan jika dijalankan dalam kondisi bergerak. Oleh karena itu, penelitian ini mencoba mengembangkan sistem IPS yang mampu digunakan dalam kondisi bergerak.

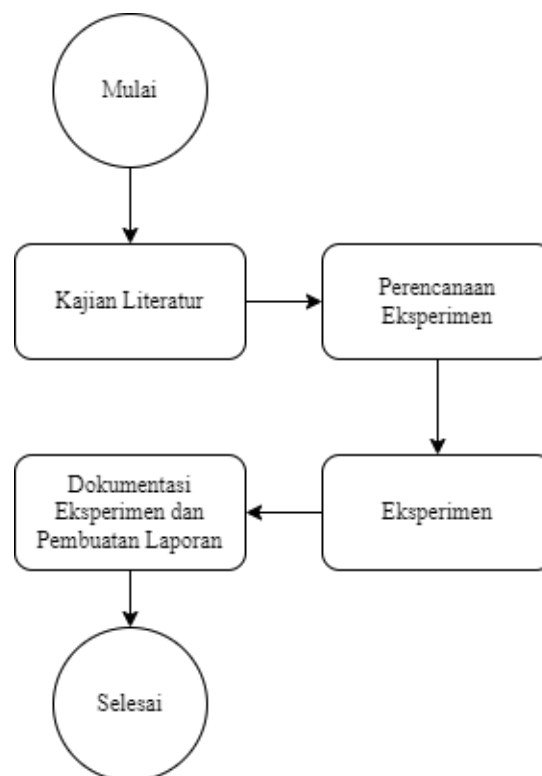
Penelitian ini bertujuan untuk mengembangkan teknik trilaterasi dan filter Kalman dalam konteks IPS yang bisa digunakan untuk bergerak. Dalam upaya ini, konsep trilaterasi akan diintegrasikan untuk estimasi posisi berdasarkan jarak dari beberapa titik referensi yang sudah diketahui. Algoritma filter Kalman yang terbukti mampu meningkatkan akurasi estimasi posisi akan diterapkan, terutama dalam situasi yang melibatkan ketidakpastian dan variasi dalam sinyal. Dengan menggabungkan kedua pendekatan ini, sistem IPS yang mampu beradaptasi dengan perubahan posisi objek secara real-time akan diciptakan. Penelitian ini fokus pada pengujian kinerja metode trilaterasi dan filter Kalman dalam berbagai kondisi yang berbeda, guna mengevaluasi efektivitas keduanya dalam meningkatkan akurasi estimasi posisi. Berbagai bentuk dan kondisi ruangan akan digunakan sebagai variabel eksperimen untuk mengkaji pengaruhnya terhadap hasil trilaterasi dan filter Kalman. Selain itu, penelitian ini juga akan meneliti pengaruh peletakan AP terhadap tingkat kesalahan pada IPS. Diharapkan bahwa pengembangan ini akan memberikan kontribusi signifikan dalam kemajuan teknologi IPS, terutama dalam aplikasi di lingkungan yang kompleks dan penuh tantangan.

BAB III METODE PENELITIAN

Bab ini menjelaskan secara rinci seluruh rangkaian langkah-langkah yang ditempuh dalam pengerjaan tugas akhir ini, dimulai dari tahap awal hingga tahap akhir. Penjelasan ini mencakup beberapa bagian utama yang terdiri dari alur pengerjaan tugas akhir, alur eksperimen, dan alur kerja sistem.

3.1 Alur Pengerjaan Tugas Akhir

Proses pelaksanaan tugas akhir ini dijelaskan melalui sebuah alur yang memberikan gambaran umum mengenai penelitian yang dilakukan, dimulai dari tahap awal hingga tahap penyelesaian. Alur pengerjaan tersebut dapat dilihat pada Gambar 3.1.



Gambar 3.1 Alur Pengerjaan Tugas Akhir

3.1.1 Kajian Literatur

Tahap awal dalam pelaksanaan tugas akhir ini adalah melakukan kajian pustaka, yaitu dengan mencari dan membaca jurnal serta artikel penelitian yang berkaitan dengan “*Indoor Positioning System*”, “*Indoor Positioning*”, dan “*Kalman Filter*”. Kajian pustaka dilakukan

dengan mencari artikel ilmiah dan jurnal dari IEEE Xplore dan ScienceDirect. Artikel dan jurnal yang diperoleh kemudian diseleksi berdasarkan kriteria berikut:

- a. Publikasi makalah penelitian diterbitkan pada atau setelah tahun 2018.
- b. Makalah yang diterima adalah artikel konferensi dan jurnal, sementara *literature review* dikecualikan.
- c. Makalah penelitian harus berkaitan dengan *Indoor Positioning System*.

Setelah diseleksi berdasarkan kriteria tersebut, tahap berikutnya adalah membaca dan mengekstrak informasi menggunakan alat *spreadsheet*. Hasil ekstraksi informasi ini telah dipresentasikan pada konferensi ilmiah internasional SIML 2024 (Firmansyah dkk., 2024) dan digunakan dalam perencanaan eksperimen.

3.1.2 Perencanaan Eksperimen

Langkah kedua dalam pelaksanaan tugas akhir ini adalah merencanakan secara rinci eksperimen yang akan dilakukan. Pada tahap ini, dirumuskan berbagai komponen utama yang akan digunakan dalam eksperimen. Pertama, dipilih *transmitter* berupa perangkat WiFi yang akan berfungsi sebagai sumber sinyal. Kedua, ditetapkan *receiver* berupa laptop yang akan menerima sinyal dari *transmitter* untuk kemudian dianalisis. Selanjutnya, untuk menentukan lokasi secara akurat, digunakan metode penentuan lokasi dengan pendekatan Trilaterasi, yang memanfaatkan jarak dari beberapa titik referensi untuk memperkirakan posisi. Terakhir, untuk meningkatkan akurasi hasil pengukuran dan mengurangi *noise*, digunakan Metode Penyaringan Kalman Filter yang dikenal efektif dalam memproses data yang berubah-ubah dan menghasilkan estimasi yang lebih halus.

Selanjutnya dalam tahap perencanaan eksperimen ini, ditentukan pula dua skenario uji coba.

- a. Skenario uji coba pertama.
 1. Menguji trilaterasi dalam kondisi statis, yang dilakukan di Ruang 1 seperti yang ditunjukkan pada Gambar 3.3. Skenario pengujian ini diambil untuk melihat apakah sistem yang dikembangkan(*code*) sudah berjalan sesuai dengan yang diharapkan.
- b. Skenario uji coba kedua.
 1. Menguji IPS dalam kondisi bergerak di dua ruangan, sesuai dengan yang ditunjukkan pada Gambar 3.3 dan Gambar 3.5. Pengujian dengan skenario ini digunakan untuk melihat apakah kalman filter mampu membantu menurunkan nilai error dari sistem IPS pada kondisi bentuk ruangan yang berbeda. Hal ini dilakukan karena pengujian

sebelumnya seperti (Wang et al., 2023b), (Mari, Kiong, & Loong, 2018b), dan (Chouhan et al., 2022b) hanya melakukan pengujian pada satu jenis bentuk ruangan.

2. Menguji trilaterasi dalam posisi AP yang berbeda seperti pada Gambar 3.4 dan Gambar 3.6. Skenario ini diambil karena penelitian sebelumnya tidak membahas apakah peletakan AP yang berbeda memiliki pengaruh, hal penting dari skenario ini adalah apakah Kalman Filter masih mampu menurunkan nilai eror tersebut.
3. Menguji trilaterasi dengan dua kondisi pada tiap ruangan, yaitu kondisi ruangan kosong dan kondisi ruangan terisi perabotan. Skenario ini diambil untuk melihat dampak dari kondisi ruangan terhadap hasil dari IPS, kemudian penelitian-penelitian sebelumnya juga tidak menunjukkan hasil pengujian kondisi ruangan terhadap hasil IPS. Hipotesis ini diambil karena pada penelitian lain juga disebutkan jika kehadiran objek mampu mengganggu sinyal antara penerima sinyal dan pengirim sinyal (Andika Satrugna Mahardhika, Gautama Putrada, & Abdurohman, 2020b). Kemudian Kadang-kadang, sinyal ini bisa menjadi tidak stabil akibat berbagai faktor (Mari et al., 2018b). Sebagai contoh, ada kondisi di mana hambatan seperti dinding atau objek lain di sekitar perangkat dapat memantulkan sinyal, menyebabkan munculnya sinyal ganda atau variasi dalam kekuatan sinyal yang diterima. Oleh karena itu, penelitian ini ingin membuktikan hipotesis tersebut dan menunjukkan hasil pengujian kondisi ruangan.

Dengan perencanaan yang terstruktur seperti pada Tabel 3.1, eksperimen diharapkan dapat memberikan hasil yang valid dan dapat diandalkan.

Tabel 3.1 Perencanaan Skenario Uji Coba

Skenario	Kondisi Laptop		Bentuk Ruang		Kondisi Ruang		Posisi AP	
	Statis	Bergerak	Ruang 1	Ruang 2	Kosong	Isi	Posisi 1	Posisi 2
Skenario 1	✓		✓		✓		✓	
Skenario 2.1		✓	✓		✓		✓	
Skenario 2.2		✓	✓			✓	✓	
Skenario 2.3		✓	✓		✓			✓
Skenario 2.4		✓	✓			✓		✓
Skenario 2.5		✓		✓	✓		✓	
Skenario 2.6		✓		✓		✓	✓	
Skenario 2.7		✓		✓	✓			✓
Skenario 2.8		✓		✓		✓		✓

3.1.3 Eksperimen

Langkah ketiga dalam pengerjaan tugas akhir ini adalah melaksanakan eksperimen sesuai dengan perencanaan yang telah dibuat pada langkah kedua. Pelaksanaan eksperimen dimulai

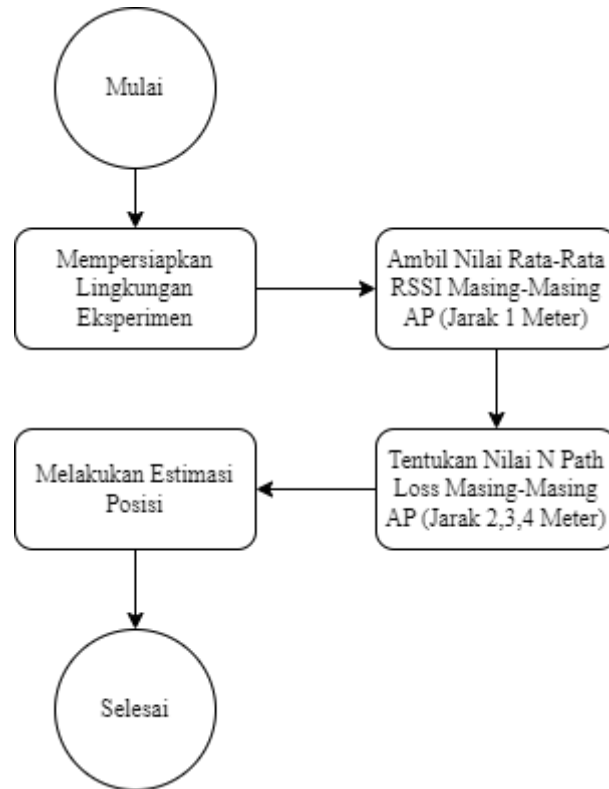
dengan mempersiapkan lingkungan eksperimen yang sesuai. Kemudian, dilakukan pengambilan nilai rata-rata RSSI dari AP, diikuti dengan penentuan nilai N Path Loss. Tahap akhir dari eksperimen ini adalah melakukan estimasi posisi berdasarkan data yang telah dikumpulkan.

3.1.4 Dokumentasi Eksperimen dan Membuat Laporan

Langkah keempat dalam pengerjaan tugas akhir ini adalah menyimpan hasil eksperimen, termasuk model dan hasil prediksi yang diperoleh. Selain itu, tahap ini juga melibatkan pelaporan penelitian dengan menyusun laporan tugas akhir yang mendokumentasikan seluruh proses dan temuan penelitian.

3.2 Alur Eksperimen

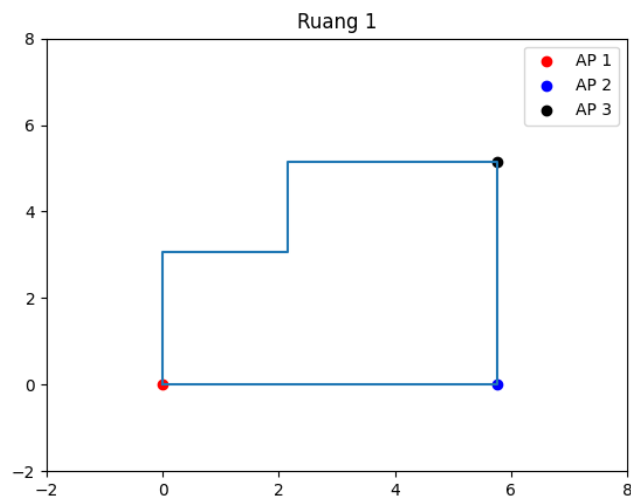
Pada bagian ini, akan diuraikan secara rinci mengenai alur eksperimen yang dilakukan dalam penelitian tugas akhir ini. Alur eksperimen ini dirancang untuk memastikan bahwa setiap langkah diambil secara sistematis dan sesuai dengan tujuan penelitian. Berikut adalah tahapan-tahapan yang tergambar dalam alur eksperimen pada Gambar 3.2. Dengan mengikuti alur eksperimen, penelitian diharapkan dapat memberikan hasil yang valid dan dapat dipertanggungjawabkan, serta memberikan kontribusi yang signifikan dalam bidang studi yang diteliti.



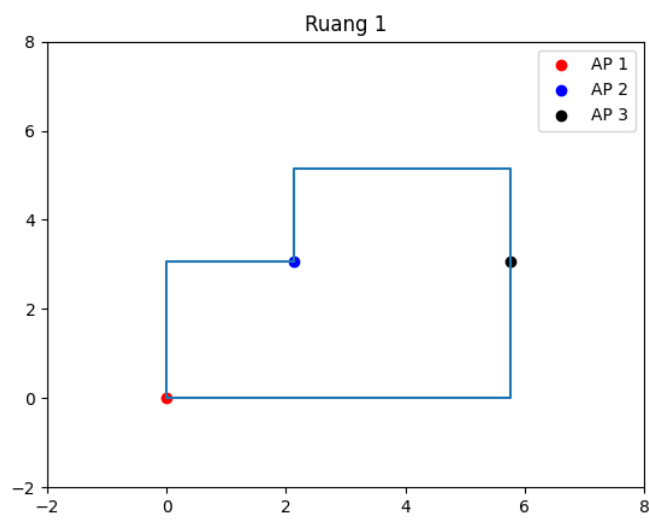
Gambar 3.2 Alur Eksperimen

3.2.1 Mempersiapkan Lingkungan Eksperimen

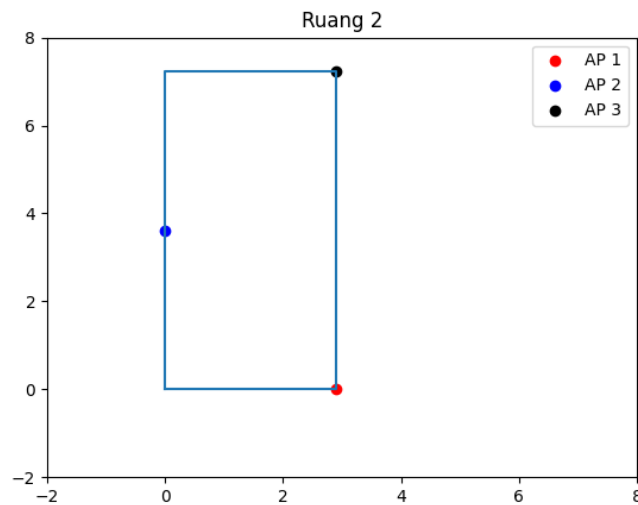
Pada tahap ini, lingkungan eksperimen dipersiapkan dengan teliti. Hal ini mencakup pengaturan ruang, penempatan perangkat, dan memastikan kondisi yang sesuai untuk mendapatkan hasil yang akurat. Dalam eksperimen ini, dua ruangan yang berbeda digunakan, yaitu Ruang 1 dan Ruang 2, seperti yang terlihat pada Gambar 3.3 dan Gambar 3.5. Kemudian pada tiap ruang juga dilakukan dua kali percobaan dengan posisi AP yang berbeda. Selanjutnya untuk pengujiannya nanti akan dilakukan dua kondisi pada tiap ruang, terdapat kondisi saat ruang kosong dan kondisi saat ruang terisi barang-barang perabotan.



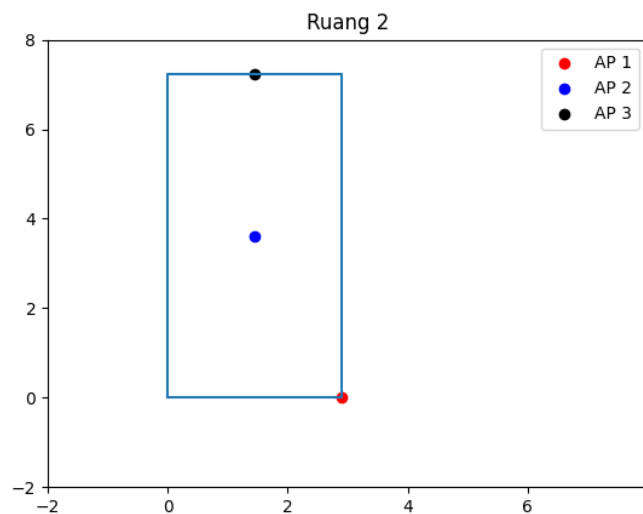
Gambar 3.3 Ruang 1 posisi 1 (2D)



Gambar 3.4 Ruang 1 posisi 2 (2D)



Gambar 3.5 Ruang 2 posisi 1 (2D)



Gambar 3.6 Ruang 2 posisi 2 (2D)

Ruang 1 memiliki konfigurasi dengan beberapa titik akses (AP) yang ditempatkan pada posisi tertentu untuk memastikan cakupan dan kekuatan sinyal yang optimal. AP 1 ditandai dengan warna merah, AP 2 dengan warna biru, dan AP 3 dengan warna hitam. Penempatan titik-titik ini dilakukan di posisi strategis untuk mengukur sinyal secara menyeluruh di seluruh ruang. Ruang 2 juga diatur dengan cara yang serupa, tetapi dengan konfigurasi yang sedikit berbeda. Di Ruang 2, penempatan AP mencakup AP 1, AP 2, dan AP 3 dengan warna yang sama seperti di Ruang 1. Setiap AP ditempatkan pada titik-titik tertentu untuk memastikan lingkungan yang stabil dan pengukuran sinyal yang akurat.



Gambar 3.7 Ruang 1 posisi 1 dalam kondisi terisi



Gambar 3.8 Ruang 1 posisi 1 dalam kondisi kosong



Gambar 3.9 Ruang 1 posisi 2 dalam kondisi terisi



Gambar 3.10 Ruang 1 posisi 2 dalam kondisi kosong



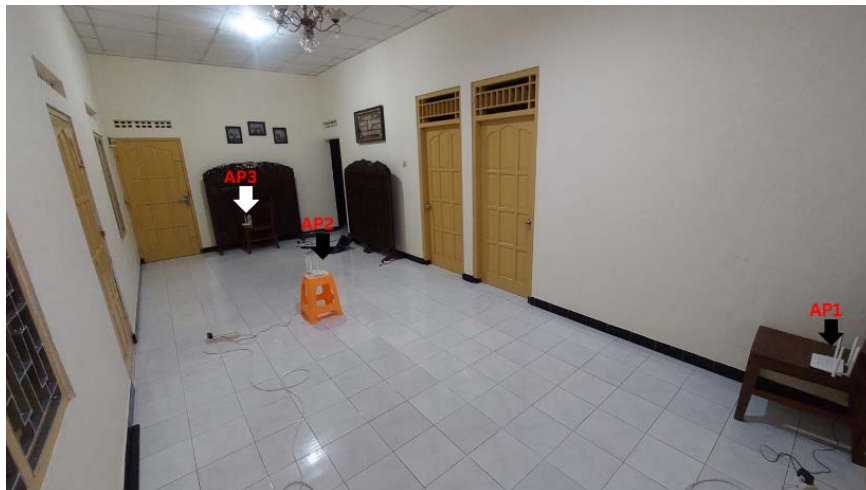
Gambar 3.11 Ruang 2 posisi 1 dalam kondisi terisi



Gambar 3.12 Ruang 2 posisi 1 dalam kondisi kosong



Gambar 3.13 Ruang 2 posisi 2 dalam kondisi terisi



Gambar 3.14 Ruang 2 posisi 2 dalam kondisi kosong

Kondisi ruangan yang digunakan pada Ruang 1 untuk penelitian kali ini seperti pada Gambar 3.7, Gambar 3.8, Gambar 3.9, dan Gambar 3.10. Selanjutnya Gambar 3.11, Gambar 3.12, Gambar 3.13, dan Gambar 3.14 adalah kondisi ruangan untuk Ruang 2. Kemudian untuk spesifikasi perangkat laptop dan AP yang digunakan pada penelitian ini dapat dilihat pada Tabel 3.2 dan Tabel 3.3.

Tabel 3.2 Spesifikasi laptop

Spesifikasi	
Laptop	MSI GL65 Leopard 10SDK
Processor	Intel Core i7-10750H
GPU	NVIDIA GeForce GTX 1660 Ti
RAM	16 GB DDR4
Sistem Operasi	Windows 11 Home
Tipe Sistem	64-bit
Wireless Network Interface Card	Intel(R) Wi-Fi 6 AX201 160MHz

Tabel 3.3 Spesifikasi Access Point

Spesifikasi	
Nama	Ruijie RG-EW1200
Standar Wi-Fi	Wi-Fi 5 (802.11ac)
Antena	4
Penguatan Antena	2.4 GHz 5 GHz

Persiapan yang teliti ini sangat penting untuk memastikan bahwa setiap variabel dalam eksperimen dapat dikontrol dan hasil yang diperoleh adalah representatif dan dapat diandalkan. Dengan dua ruangan yang berbeda ini, perbandingan dan analisis terhadap variasi dalam konfigurasi ruang serta pengaruhnya terhadap hasil sistem *positioning indoor* yang sedang diuji dapat dilakukan dengan lebih baik.

Pada 2 langkah selanjutnya dijelaskan cara melakukan kalibrasi AP agar hasil pengukuran menjadi lebih akurat dalam sistem IPS yang bergerak. Kalibrasi dibutuhkan untuk memastikan akurasi dan konsistensi hasil pengukuran, sehingga sistem dapat berfungsi dengan optimal dan memberikan estimasi posisi yang tepat. Langkah tersebut mengacu pada cara yang digunakan oleh (Wibisono & Wicaksono, 2022c).

3.2.2 Mengambil Nilai Rata-Rata RSSI AP

Langkah selanjutnya adalah mengambil nilai rata-rata RSSI dari setiap AP pada jarak 1 meter menggunakan rumus seperti (2.1). Pengukuran ini dilakukan dengan hati-hati untuk memastikan akurasi dan konsistensi data yang diperoleh. Setiap AP diuji secara terpisah untuk mendapatkan nilai RSSI yang spesifik pada jarak tersebut, yang kemudian dicatat dan dianalisis. Data ini sangat penting untuk mengetahui kekuatan sinyal yang diterima dari masing-masing AP, yang akan digunakan sebagai dasar untuk perhitungan dan pemetaan lebih

lanjut dalam IPS. Mengetahui kekuatan sinyal pada jarak 1 meter memungkinkan prediksi perilaku sinyal pada jarak yang lebih jauh. Hal ini juga membantu dalam mengkalibrasi sistem untuk meningkatkan keakuratan dan keandalan dalam menentukan posisi objek di dalam ruangan. Analisis ini juga berguna dalam mengidentifikasi potensi gangguan atau hambatan yang mungkin memengaruhi kekuatan dan kestabilan sinyal, sehingga langkah-langkah penyesuaian dapat diambil untuk mengoptimalkan kinerja sistem secara keseluruhan.

3.2.3 Menentukan Nilai N Path Loss AP

Setelah mendapatkan nilai RSSI pada jarak 1 meter, langkah berikutnya adalah melakukan pengukuran nilai N Path Loss pada jarak 2, 3, dan 4 meter dari masing-masing Access Point (AP) menggunakan rumus (2.2). Pengukuran ini dilakukan dengan metode yang konsisten untuk memastikan data yang akurat dan dapat diandalkan. Setiap jarak diuji sebanyak 200 kali untuk mendapatkan nilai rata-rata yang representatif, yang kemudian digunakan untuk analisis lebih lanjut. Langkah ini sangat penting untuk memahami cara sinyal melemah seiring bertambahnya jarak dari AP, yang dikenal sebagai *path loss*. Proses ini mengikuti penelitian (Wibisono & Wicaksono, 2022b).

Path loss adalah fenomena di mana kekuatan sinyal menurun saat sinyal bergerak menjauh dari sumbernya. Pemahaman mendalam tentang pola penurunan ini sangat krusial dalam merancang dan mengoptimalkan sistem *positioning indoor*. Dengan mengetahui nilai N Path Loss pada berbagai jarak, dapat dilakukan kalibrasi yang lebih tepat terhadap model prediksi sinyal, sehingga akurasi dalam menentukan lokasi objek dapat ditingkatkan. Pengukuran ini juga membantu dalam mengidentifikasi dan mengatasi potensi masalah yang disebabkan oleh penghalang fisik atau interferensi lainnya yang mungkin memengaruhi transmisi sinyal. Dengan demikian, langkah ini tidak hanya memberikan wawasan tentang karakteristik sinyal, tetapi juga berperan penting dalam memastikan keandalan dan keakuratan sistem *positioning* yang sedang dikembangkan.

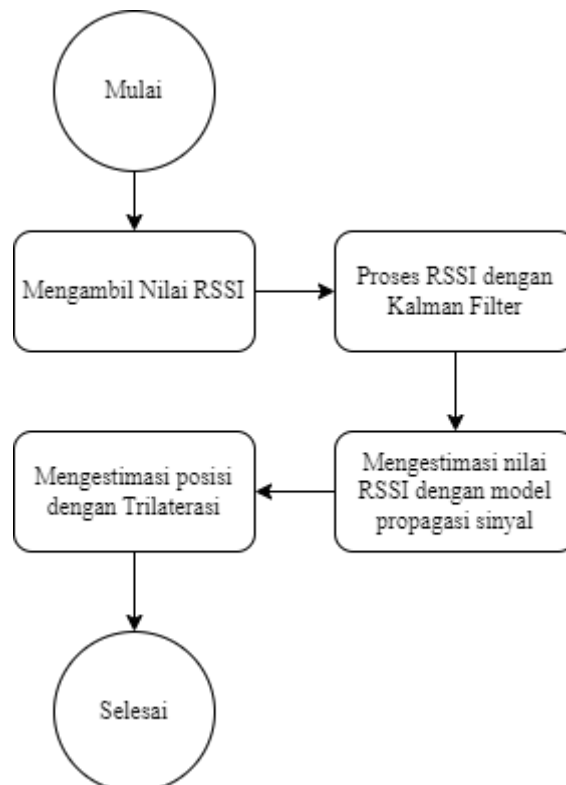
3.2.4 Melakukan Estimasi Posisi

Berdasarkan data yang telah dikumpulkan, dilakukan estimasi posisi menggunakan rumus (2.19) dan (2.20). Proses ini melibatkan penggunaan metode yang telah direncanakan sebelumnya untuk menentukan lokasi secara akurat. Setiap data RSSI yang telah diperoleh dari berbagai jarak dan dari setiap AP dianalisis secara mendalam untuk mengidentifikasi pola dan korelasi yang relevan. Metode yang digunakan melibatkan algoritma trilaterasi.

Dengan melakukan estimasi posisi berdasarkan data yang telah dikumpulkan, sistem dapat memberikan informasi lokasi yang lebih presisi. Hal ini sangat penting untuk aplikasi yang memerlukan penentuan lokasi yang akurat, seperti navigasi dalam ruangan, manajemen aset, atau aplikasi lain yang bergantung pada teknologi IPS. Proses ini memastikan bahwa setiap variabel telah dipertimbangkan dan bahwa hasil yang diperoleh dapat diimplementasikan dengan sukses dalam lingkungan nyata.

3.3 Alur Kerja Sistem

Untuk memahami cara IPS bekerja, penting untuk melihat alur kerja yang digunakan dalam proses ini. Alur kerja sistem ini dirancang untuk memastikan bahwa setiap tahap dari pengumpulan data hingga estimasi posisi dilakukan dengan teliti dan efisien. Diagram alur kerja yang disajikan memberikan gambaran menyeluruh tentang langkah-langkah yang diambil dalam sistem ini. Setiap tahap dalam alur kerja memainkan peran penting dalam memastikan keakuratan dan keandalan estimasi posisi yang dihasilkan. Berikut adalah penjelasan rinci mengenai setiap tahap dalam alur kerja sistem yang ditunjukkan pada Gambar 3.15.



Gambar 3.15 Alur Kerja Sistem Trilaterasi

3.3.1 Mengambil Nilai RSSI

Alur kerja sistem dimulai dengan tahap pengambilan nilai RSSI. Pada tahap ini, sistem mengukur kekuatan sinyal yang diterima dari berbagai AP pada posisi tertentu. Pengukuran ini dilakukan untuk mendapatkan data awal yang diperlukan untuk proses estimasi posisi.

3.3.2 Memproses RSSI dengan Kalman Filter

Setelah nilai RSSI berhasil diambil, langkah selanjutnya adalah memproses nilai-nilai RSSI tersebut menggunakan Kalman Filter. Kalman Filter digunakan untuk menghaluskan data RSSI yang mungkin mengandung *noise* atau fluktuasi, sehingga data yang dihasilkan menjadi lebih akurat dan dapat diandalkan untuk langkah-langkah berikutnya.

3.3.3 Mengubah Nilai RSSI dengan Model Propagasi Sinyal

Tahap berikutnya dalam alur kerja sistem adalah mengestimasi nilai RSSI dengan model propagasi sinyal menggunakan rumus (2.2). Pada tahap ini, model matematis digunakan untuk memprediksi perilaku sinyal dalam berbagai kondisi dan jarak. Hasil estimasi ini kemudian dibandingkan dengan data RSSI yang telah diproses untuk memperbaiki keakuratan estimasi posisi.

3.3.4 Mengestimasi Posisi dengan Trilaterasi

Setelah proses estimasi nilai RSSI selesai, langkah selanjutnya adalah mengestimasi posisi dengan metode trilaterasi seperti rumus (2.20) dan (2.21). Trilaterasi adalah teknik yang menggunakan jarak dari beberapa AP untuk menentukan posisi objek secara geometris. Dengan menggunakan data jarak yang telah diperoleh dari tahap sebelumnya, posisi objek dapat dihitung dan dipetakan dengan akurat.

Dengan mengikuti alur kerja sistem yang terstruktur ini, setiap tahap dari pengambilan data hingga estimasi posisi dapat dilakukan secara sistematis dan efisien, memastikan hasil yang akurat dan dapat diandalkan dalam sistem IPS.

BAB IV

HASIL DAN PEMBAHASAN

Bab ini akan membahas hasil penelitian yang telah dilakukan serta menjawab berbagai rumusan masalah yang telah dirumuskan sebelumnya. Pembahasan ini mencakup beberapa aspek penting dalam pengembangan dan evaluasi sistem IPS. Fokus utama penelitian ini adalah untuk mengembangkan sebuah IPS yang bisa digunakan untuk bergerak dengan menggunakan metode trilaterasi dan filter Kalman, serta mengevaluasi dampak penggunaan filter Kalman baik pada IPS yang statis maupun bergerak.

4.1 Mengembangkan IPS Menggunakan Trilaterasi dan Kalman Filter

Pada subbab ini, akan dibahas secara mendetail mengenai proses pengembangan sistem IPS dengan menggunakan metode trilaterasi dan filter Kalman. Pembahasan akan berisi struktur kode Python yang diterapkan pada sistem ini. Penjelasan akan dibuat secara terstruktur berdasarkan jalannya sistem agar mudah dimengerti. Untuk kode versi lengkap ditampilkan pada Lampiran A. Untuk memastikan agar kode dapat dijalankan, semua modul yang digunakan dalam kode penelitian ini perlu diinstal. Daftar modul yang perlu diinstal terdiri dari *matplotlib*, *pandas*, *pywifi*, *filterpy*, *numpy*.

4.1.1 Mendapatkan Nilai RSSI

Langkah awal yang krusial dalam sistem ini adalah mendapatkan nilai RSSI. Seperti yang ditunjukkan pada Gambar 4.1, sistem ini menggunakan modul *pywifi* yang diimpor dari *library* Python. Proses *scanning* RSSI dilakukan dengan menggunakan antarmuka WiFi dari laptop. *Scanning* dilakukan setiap 0,5 detik sekali untuk memastikan data yang diperoleh konsisten dan akurat.

```

1 def scan_rssi(ssid, outputRSSI, max_data, stop_event):
2     wifi = pywifi.PyWiFi()
3     iface = wifi.interfaces()[0] # Menggunakan antarmuka pertama (biasanya wlan0)
4
5     try:
6         # Inisialisasi hitungan data yang diambil
7         data_count = 0
8
9         while data_count < max_data and not stop_event.is_set():
10            iface.scan()
11            time.sleep(0.5)
12            scan_results = iface.scan_results()

```

```

13
14     for result in scan_results:
15         if ssid in result.ssid:
16             # Fungsi untuk mendapatkan nilai RSSI
17             rssi = result.signal
18             with open(outputRSSI, "a") as file:
19                 file.write(f"RSSI {ssid} = {rssi} \n")
20             print(f"{data_count+1}.RSSI {ssid} = {rssi} dBm")
21
22             # Menambah jumlah data yang diambil
23             data_count += 1
24
25             if data_count >= max_data:
26                 print(f"Pengambilan Data {ssid} Selesai")
27                 break
28         else:
29             print(f"Tidak dapat menemukan {ssid}")
30             # Set stop_event jika SSID tidak ditemukan
31             stop_event.set()
32             break
33
34     except KeyboardInterrupt:
35         print("Dihentikan oleh pengguna (Ctrl+C)")

```

Gambar 4.1 Kode untuk mendapatkan nilai RSSI.

Dalam kode tersebut, modul `pywifi` diinisialisasi dan digunakan untuk memindai sinyal WiFi setiap 0,5 detik. Hasil pemindaian kemudian dievaluasi untuk menemukan nilai RSSI dari SSID yang diinginkan. Jika SSID ditemukan, nilai RSSI dicatat dan proses pemindaian berlanjut hingga jumlah data yang diinginkan tercapai atau proses dihentikan oleh pengguna.

4.1.2 Mengolah Nilai RSSI Menggunakan Kalman Filter

Langkah berikutnya setelah memperoleh nilai RSSI adalah mengolah nilai tersebut menggunakan Kalman Filter. Filter yang digunakan pada Gambar 4.2 memanfaatkan library `filterpy` yang sudah tersedia di Python. Seperti yang telah dijelaskan pada bab sebelumnya, Kalman Filter bekerja melalui dua proses utama, yaitu *predict()* dan *update()*.

```

1     """
2     Calculate the distance based on RSSI, reference RSSI (A), and path loss
3 exponent(n).
4     Parameters:
5         RSSI (float): Received Signal Strength Indication.
6         A (float): Reference RSSI at 1 meter.
7         n (float): Path loss exponent.
8     Returns:
9         float: Distance between transmitter and receiver.
10    """
11
12    # Update Filter Kalman dengan pengukuran RSSI
13    reference_rssi = reference_rssi_dict[ssid]
14    n = n_dict[ssidN]
15    if data_count == 0:
16        kfAP1.x = np.array([reference_rssi])
17    kfAP1.predict()

```

```
18 kfAP1.update(np.array([rssi]))
```

Gambar 4.2 Kode untuk menerapkan Kalman Filter.

Pada kode tersebut, nilai referensi RSSI dan eksponen diperoleh dari *dictionary* yang sudah ditentukan. Jika data yang diproses adalah data pertama, Kalman Filter diinisialisasi dengan nilai referensi RSSI. Selanjutnya, proses *predict()* dilakukan untuk memperkirakan nilai selanjutnya, kemudian diikuti dengan *update()* untuk memperbarui perkiraan tersebut berdasarkan nilai RSSI yang baru diperoleh. Dengan menggunakan Kalman Filter, nilai RSSI yang berfluktuasi dapat dihaluskan sehingga menghasilkan estimasi jarak yang lebih akurat.

4.1.3 Mencari Nilai Eksponen (N)

Sebelum mencari nilai trilaterasi, terdapat langkah yang harus dilakukan yaitu mendapatkan nilai eksponen *path loss*. Nilai eksponen (n) terdapat pada Gambar 4.3 baris 14 yang merupakan penerapan dari rumus (2.3). Nilai ini diperoleh melalui pengukuran langsung dengan jarak 2, 3, dan 4 meter, dari masing-masing AP kemudian dilakukan sebanyak 100 kali pengambilan data.

```
1 def calculate_path_loss_exponent(RSSI, ssid, d):
2     """
3     Calculate the path loss exponent (n) based on RSSI, reference RSSI (A), and
4 distance (d).
5     Parameters:
6     RSSI (float): Received Signal Strength Indication.
7     A (float): Reference RSSI at 1 meter.
8     d (float): Distance between transmitter and receiver.
9     Returns:
10    float: Path loss exponent (n).
11    """
12
13    A = reference_rssi_dict[ssid]
14    n = -1*(RSSI - A) / (10 * math.log10(d))
15    return n
```

Gambar 4.3 Kode untuk mencari nilai eksponen.

Pada Gambar 4.3, eksponen *path loss* dihitung berdasarkan nilai RSSI yang diterima, referensi RSSI pada jarak 1 meter, dan jarak antara pemancar dan penerima. Hasil pengukuran ini kemudian digunakan untuk langkah selanjutnya dalam proses trilaterasi.

4.1.4 Melakukan Estimasi Jarak (*Signal Propagation Model*)

Setelah mendapatkan nilai RSSI yang sudah diolah oleh Kalman Filter, sistem akan mengubah nilai RSSI tersebut menjadi jarak dalam satuan meter.

```

1 estimated_rssi = kfAP1.x[0]
2 distanceKalman = 10 ** ((reference_rssi - estimated_rssi) / (10 * n))

```

Gambar 4.4 Kode untuk mengubah RSSI menjadi Jarak.

Pada kode ini, nilai RSSI yang telah diperkirakan (*estimated_rssi*) digunakan untuk menghitung jarak (*distanceKalman*) dengan rumus yang memperhitungkan referensi RSSI dan nilai eksponen (*n*). Proses ini memastikan bahwa hasil pengukuran lebih stabil dan akurat.

4.1.5 Melakukan Estimasi Posisi Menggunakan Trilaterasi

Setelah mendapatkan nilai RSSI yang sudah diolah oleh Kalman Filter, sistem akan mengubah nilai RSSI tersebut menjadi jarak dalam satuan meter.

```

1 def trilateration(ssid1,rAP1,ssid2,rAP2,ssid3,rAP3):
2     """
3     Trilateration untuk menentukan posisi pengguna berdasarkan sinyal yang diterima
4 dari beberapa beacon.
5     Args:
6         beacons (list): List koordinat beacon dalam format (x, y).
7         distances (list): List jarak antara pengguna dan masing-masing beacon.
8     Returns:
9         tuple: Koordinat perkiraan posisi pengguna dalam format (x, y).
10    """
11
12    xAP1 = beacon_positions[ssid1][0]
13    yAP1 = beacon_positions[ssid1][1]
14    xAP2 = beacon_positions[ssid2][0]
15    yAP2 = beacon_positions[ssid2][1]
16    xAP3 = beacon_positions[ssid3][0]
17    yAP3 = beacon_positions[ssid3][1]
18
19    A = 2*xAP2 - 2*xAP1
20    B = 2*yAP2 - 2*yAP1
21    C = rAP1**2 - rAP2**2 - xAP1**2 + xAP2**2 - yAP1**2 + yAP2**2
22    D = 2*xAP3 - 2*xAP2
23    E = 2*yAP3 - 2*yAP2
24    F = rAP2**2 - rAP3**2 - xAP2**2 + xAP3**2 - yAP2**2 + yAP3**2
25    x = (C*E - F*B) / (E*A - B*D)
26    y = (C*D - A*F) / (B*D - A*E)
27    return x,y

```

Gambar 4.5 Kode Trilaterasi untuk prediksi posisi.

Pada kode ini, trilaterasi digunakan untuk menentukan posisi pengguna berdasarkan sinyal yang diterima dari beberapa AP. Baris kode 19 sampai 24 pada Gambar 4.5 merupakan

penerapan dari rumus (2.16) dan (2.17). Kemudian baris kode 25 dan 26 pada Gambar 4.5 merupakan penerapan dari rumus (2.20) dan (2.21). Jarak antara pengguna dan masing-masing AP, yang telah dihitung dari nilai RSSI, kemudian koordinat masing-masing AP juga digunakan sebagai input untuk rumus trilaterasi. Rumus tersebut menghitung koordinat (x, y) dari pengguna, kemudian menghasilkan estimasi posisi yang lebih akurat berdasarkan data yang telah diolah oleh Kalman Filter.

4.1.6 Melakukan Plotting Data Trilaterasi (*Animate*)

Setelah mendapatkan nilai dari trilaterasi, langkah terakhir dalam sistem ini adalah melakukan *plotting* data secara *real-time*. Proses *plotting* ini bertujuan untuk memvisualisasikan posisi pengguna berdasarkan hasil perhitungan trilaterasi yang telah dilakukan. Data yang diperoleh dari pengukuran dan estimasi menggunakan Kalman Filter dan metode trilaterasi akan ditampilkan dalam bentuk grafis, sehingga memudahkan untuk memantau dan menganalisis pergerakan pengguna secara langsung. Proses *plotting* ini mencakup menggambarkan posisi AP serta posisi estimasi pengguna pada peta ruangan, memungkinkan identifikasi dan penyesuaian yang diperlukan untuk meningkatkan akurasi sistem posisi dalam ruangan.

```

1 def minDistance(A, B, E):
2     AB = [B[0] - A[0], B[1] - A[1]]
3     BE = [E[0] - B[0], E[1] - B[1]]
4     AE = [E[0] - A[0], E[1] - A[1]]
5
6     AB_BE = AB[0] * BE[0] + AB[1] * BE[1]
7     AB_AE = AB[0] * AE[0] + AB[1] * AE[1]
8
9     reqAns = 0
10
11     if AB_BE > 0:
12         reqAns = sqrt(BE[0] * BE[0] + BE[1] * BE[1])
13     elif AB_AE < 0:
14         reqAns = sqrt(AE[0] * AE[0] + AE[1] * AE[1])
15     else:
16         reqAns = abs(AB[0] * AE[1] - AB[1] * AE[0]) / sqrt(AB[0] * AB[0] + AB[1] *
17 AB[1])
18
19     return reqAns
20
21 def animate(i):
22     try:
23         dataKF = pd.read_csv('outputTrilaterasiKF.csv')
24         data = pd.read_csv('outputTrilaterasi.csv')
25         # Inisialisasi objek Lock
26         lock = Lock()
27         fieldnames = ["error"]
28         if not dataKF.empty and not data.empty:
29             xKF = dataKF['x']
30             yKF = dataKF['y']

```

```

31
32     x = data['x']
33     y = data['y']
34
35     plt.cla()
36
37     # Gambar persegi panjang
38     plt.plot([0, 5.76, 5.76, 2.14, 2.14, 0, 0], [0, 0, 5.16, 5.16, 3.06,
39 3.06, 0])
40
41     A = [0, 1.53]
42     B = [3.95, 1.53]
43     C = [3.95, 1.53]
44     D = [3.95, 4.10]
45     EKf = [dataKF['x'].iloc[-1], dataKF['y'].iloc[-1]]
46     E = [data['x'].iloc[-1], data['y'].iloc[-1]]
47
48     # Calculate minimum distances
49     min_dist_ABKf = minDistance(A, B, EKf)
50     min_dist_CDKf = minDistance(C, D, EKf)
51     if min_dist_ABKf < min_dist_CDKf:
52         with lock: # Menggunakan lock untuk mengamankan akses ke file CSV
53             with open("outputErrorTitikGaris(KF).csv", 'a') as csv_file:
54                 csv_writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
55                 info = {
56                     "error": min_dist_ABKf,
57                 }
58                 csv_writer.writerow(info)
59                 print("Jarak Titik ke Garis Jalan (KF):", min_dist_ABKf)
60     elif min_dist_ABKf > min_dist_CDKf:
61         with lock: # Menggunakan lock untuk mengamankan akses ke file CSV
62             with open("outputErrorTitikGaris(KF).csv", 'a') as csv_file:
63                 csv_writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
64                 info = {
65                     "error": min_dist_CDKf,
66                 }
67                 csv_writer.writerow(info)
68                 print("Jarak Titik ke Garis Jalan (KF):", min_dist_CDKf)
69     elif min_dist_ABKf == min_dist_CDKf:
70         with lock: # Menggunakan lock untuk mengamankan akses ke file CSV
71             with open("outputErrorTitikGaris(KF).csv", 'a') as csv_file:
72                 csv_writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
73                 info = {
74                     "error": min_dist_ABKf,
75                 }
76                 csv_writer.writerow(info)
77                 print("Jarak Titik ke Garis Jalan (KF):", min_dist_ABKf)
78
79     # Calculate minimum distances
80     min_dist_AB = minDistance(A, B, E)
81     min_dist_CD = minDistance(C, D, E)
82     if min_dist_AB < min_dist_CD:
83         with lock: # Menggunakan lock untuk mengamankan akses ke file CSV
84             with open("outputErrorTitikGaris.csv", 'a') as csv_file:
85                 csv_writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
86                 info = {
87                     "error": min_dist_AB,
88                 }
89                 csv_writer.writerow(info)
90                 print("Jarak Titik ke Garis Jalan :", min_dist_AB)
91     elif min_dist_AB > min_dist_CD:
92         with lock: # Menggunakan lock untuk mengamankan akses ke file CSV
93             with open("outputErrorTitikGaris.csv", 'a') as csv_file:
94                 csv_writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
95                 info = {
96                     "error": min_dist_CD,
97                 }

```

```

98         csv_writer.writerow(info)
99         print("Jarak Titik ke Garis Jalan :", min_dist_CD)
100     elif min_dist_AB == min_dist_CD:
101         with lock: # Menggunakan lock untuk mengamankan akses ke file CSV
102             with open("outputErrorTitikGaris.csv", 'a') as csv_file:
103                 csv_writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
104                 info = {
105                     "error": min_dist_AB,
106                 }
107                 csv_writer.writerow(info)
108             print("Jarak Titik ke Garis Jalan :", min_dist_AB)
109
110         plt.plot([A[0], B[0]], [A[1], B[1]], marker='o', label='Line Segment
111 AB',color='green')
112         plt.plot([C[0], D[0]], [C[1], D[1]], marker='o', label='Line Segment
113 CD',color='green')
114
115         plt.scatter(xKF, yKF, label='Prediksi Histori Kalman', color='blue')
116 # Menggunakan plt.scatter() untuk memplot titik-titik
117         plt.scatter(x, y, label='Prediksi Tanpa Kalman', color='red')
118 #Menggunakan plt.scatter() untuk memplot titik-titik
119         plt.scatter(dataKF['x'].iloc[-1], dataKF['y'].iloc[-1], label='Prediksi
120 Kalman Sekarang', color='black') # Menggunakan plt.scatter() untuk memplot titik-
121 titik
122
123         plt.scatter(0, 0, label = "AP1", color = 'orange')
124         plt.scatter(2.14, 3.06, label = "AP2", color = 'yellow')
125         plt.scatter(5.76, 3.06, label = "AP3", color = 'purple')
126
127         plt.legend(loc='upper left')
128         plt.tight_layout()
129     except Exception as e:
130         pass

```

Gambar 4.6 Kode untuk melakukan plotting hasil trilaterasi.

Kode tersebut dimulai dengan fungsi *minDistance()* yang bertujuan untuk menghitung jarak minimum dari suatu titik ke garis yang dihasilkan oleh dua titik lainnya. Fungsi ini menghitung jarak berdasarkan proyeksi ortogonal dari titik ke garis atau jarak langsung jika proyeksi jatuh di luar segmen garis.

Fungsi *animate(i)* kemudian membaca data posisi yang diprediksi dengan dan tanpa Kalman Filter dari file CSV. Data ini digunakan untuk melakukan *plotting real-time* posisi pengguna dan perhitungan eror jarak minimum dari titik ke garis jalan yang telah ditentukan.

Di dalam fungsi *animate(i)*, data posisi dari kedua sumber (dengan dan tanpa Kalman Filter) dibaca dan diproses. Jarak minimum dari titik ke garis jalan dihitung untuk setiap metode prediksi posisi (dengan Kalman Filter dan tanpa Kalman Filter). Hasil perhitungan jarak eror disimpan ke dalam file CSV untuk analisis lebih lanjut.

Plotting dilakukan dengan menggambarkan posisi AP, garis jalan, serta titik-titik posisi pengguna yang diprediksi. Posisi histori dari prediksi Kalman, prediksi tanpa Kalman, dan prediksi Kalman terbaru juga ditampilkan dalam plot dengan warna yang berbeda untuk memudahkan visualisasi dan perbandingan.

Fungsi *animate(i)* diatur untuk berjalan secara terus-menerus dalam interval waktu tertentu untuk memperbarui plot secara *real-time*, memberikan gambaran langsung tentang pergerakan pengguna dan akurasi prediksi posisi dari sistem IPS yang dikembangkan.

4.2 Skenario Uji Coba

Pada subbab ini, penelitian akan memaparkan hasil uji coba yang telah dilakukan secara rinci. Penjabaran hasil ini akan dibagi ke dalam beberapa anak subbab untuk memudahkan pemahaman dan analisis lebih mendalam. Pertama, akan disajikan hasil trilaterasi dalam kondisi statis, di mana objek yang dipantau tidak bergerak. Penelitian kemudian akan melanjutkan dengan memaparkan hasil trilaterasi dalam kondisi objek bergerak.

Selain itu, pengaruh variasi bentuk ruangan terhadap hasil trilaterasi juga akan dianalisis secara komprehensif, mengingat bahwa bentuk ruangan dapat memengaruhi distribusi sinyal dan akurasi penentuan posisi. Terakhir, akan dibahas pengaruh posisi Access Points (AP) yang berbeda terhadap hasil trilaterasi, mengingat penempatan AP yang strategis sangat penting untuk memastikan cakupan sinyal yang optimal dan akurasi posisi yang tinggi. Dengan penjabaran yang terstruktur ini, diharapkan dapat memberikan gambaran yang jelas dan lengkap mengenai efektivitas dan efisiensi sistem posisi dalam ruangan yang dikembangkan melalui penelitian ini.

4.2.1 Skenario Uji Coba 1

Pada skenario uji coba pertama, sistem diuji dalam kondisi trilaterasi yang tetap atau statis. Pengujian ini melibatkan empat titik yang telah ditentukan sebelumnya di dalam ruangan, seperti yang terlihat pada Gambar 3.3. Keempat titik ini dipilih untuk memastikan cakupan area yang merata dan untuk menguji keakuratan sistem dalam berbagai posisi yang berbeda.

Pada tahap awal, dilakukan kalibrasi setiap AP dengan mengumpulkan 200 data RSSI dari jarak 1 meter, lalu menghitung rata-ratanya. Hasilnya, nilai rata-rata RSSI untuk AP1, AP2, dan AP3 masing-masing adalah -23, -24, dan -25. Selanjutnya, nilai eksponen N untuk masing-masing AP ditentukan dengan mengukur dari jarak 2, 3, dan 4 meter. Untuk setiap jarak, dikumpulkan 100 data. Setelah memperoleh nilai N, dilakukan pengujian dengan memindai jarak untuk melihat nilai eror terkecil di antara 2, 3, dan 4 meter. Nilai N dengan eror terkecil kemudian dipilih untuk digunakan pada AP tersebut. Hasil pengujian tersebut

dapat dilihat pada Tabel 4.1, Tabel 4.2, dan Tabel 4.3. Untuk nilai-nilai lain yang digunakan pada penelitian ini bisa dilihat pada Lampiran D.

Tabel 4.1 Nilai N pada AP1

Meter	RSSI	N	MSE (Mean Square Error)	RMSE (Root Mean Square Error)	Rata-Rata Error (m)
2	-31,14	2,70	0,441	0,664	0,56
3	-37,55	3,05	1,798	1,341	0,796
4	-43,44	3,40	5,625	2,371	1,886

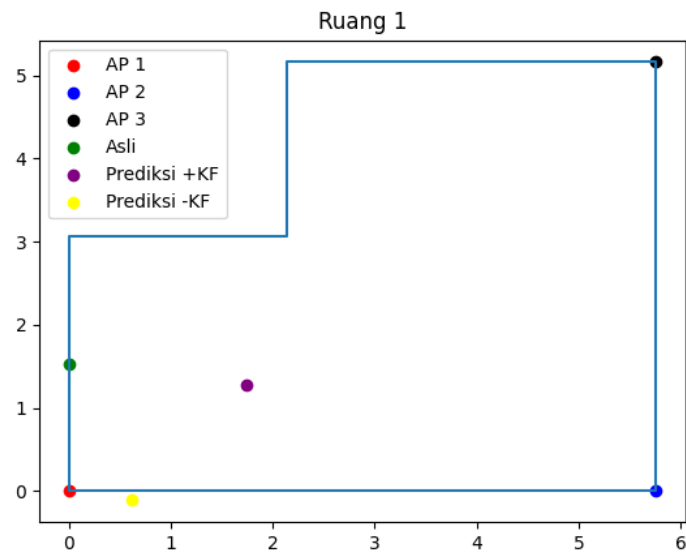
Tabel 4.2 Nilai N pada AP2

Meter	RSSI	N	MSE (Mean Square Error)	RMSE (Root Mean Square Error)	Rata-Rata Error (m)
2	-33,18	2,70	0,306	0,553	0,403
3	-33,04	3,05	3,241	1,8	0,99
4	-38,08	3,40	13,47	3,67	2,611

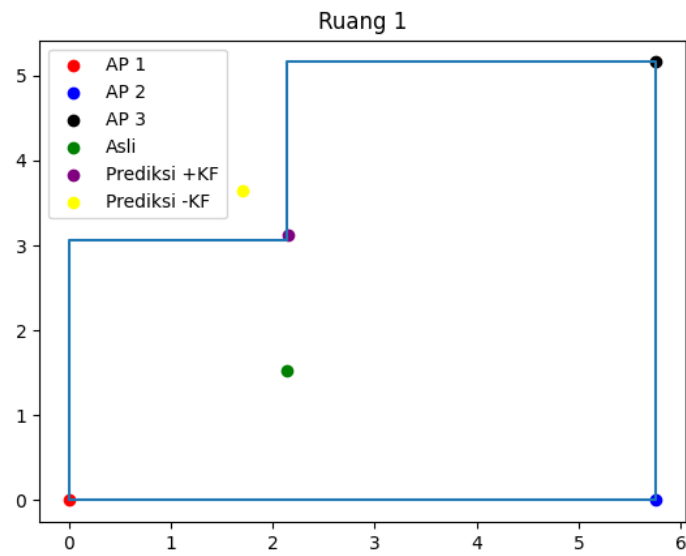
Tabel 4.3 Nilai N pada AP3

Meter	RSSI	N	MSE (Mean Square Error)	RMSE (Root Mean Square Error)	Rata-Rata Error (m)
2	-31,89	2,70	1,57	1,253	1,069
3	-33,32	3,05	4,061	2,015	1,182
4	-34,63	3,40	19,787	4,448	1,505

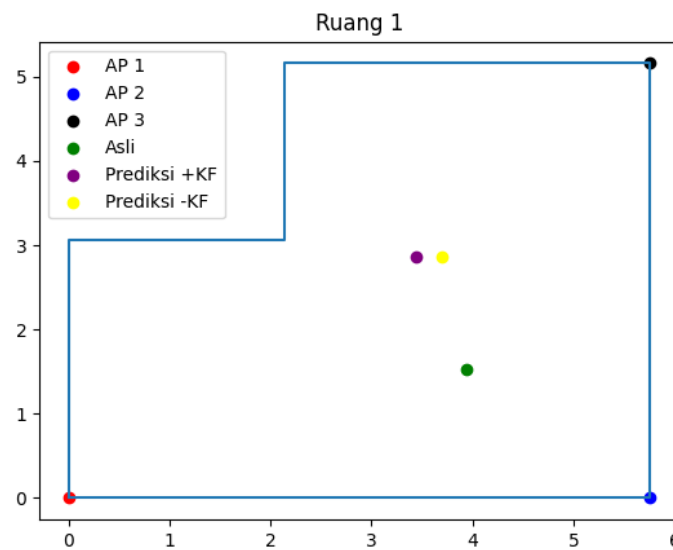
Berdasarkan hasil pengujian, nilai N untuk masing-masing AP ditentukan sebagai 2,70 untuk AP1, 3,05 untuk AP2, dan 2,29 untuk AP3. Penentuan ini didasarkan pada MSE, RMSE, dan rata-rata eror terkecil dibandingkan dengan jarak lainnya. Tahap selanjutnya adalah menguji trilaterasi dalam kondisi statis di Ruang 1 dengan melakukan percobaan pada empat titik di ruangan tersebut. Setiap titik diuji sebanyak lima kali, kemudian nilai rata-rata dari lima pengujian ini disajikan dalam hasil penelitian kali ini.



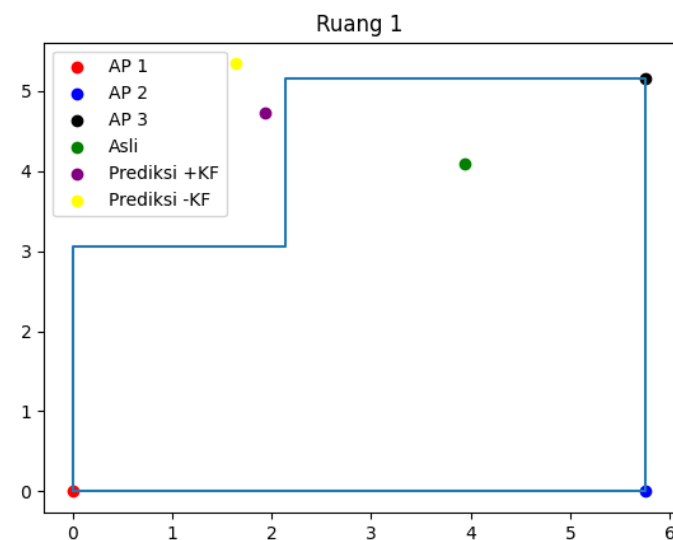
Gambar 4.7 Hasil perhitungan trilaterasi statis pada titik 1.



Gambar 4.8 Hasil perhitungan trilaterasi statis pada titik 2.



Gambar 4.9 Hasil perhitungan trilaterasi statis pada titik 3.



Gambar 4.10 Hasil perhitungan trilaterasi statis pada titik 4.

Dari hasil pengujian trilaterasi statis yang sudah divisualisasikan seperti Gambar 4.7, Gambar 4.8, Gambar 4.9, dan Gambar 4.10, dapat disimpulkan bahwa metode trilaterasi memberikan hasil yang cukup akurat. Gambar visualisasi hasil pengujian lengkap dapat dilihat pada Lampiran B, sedangkan untuk gambar sesi pengujian lainnya ditampilkan pada Lampiran C. Dilakukan juga pencarian nilai eror dari titik prediksi ke titik asli pengujian, dapat dilihat pada Tabel 4.4. Cara yang digunakan untuk menghitung jarak antara titik prediksi ke jarak titik asli adalah menggunakan rumus jarak Euclidean seperti pada (2.4).

Tabel 4.4 Nilai perubahan trilaterasi statis

Titik	Koordinat Asli (x,y)	Nilai Jarak (-KF) ke Titik Asli (m)	Nilai Jarak (+KF) ke Titik Asli (m)	Penurunan Nilai Error
1	(0, 1.53)	4,482	1,544	65,55%
2	(2.14, 1.53)	2,604	1,818	30,18%
3	(3.94, 1.53)	1,95	1,502	22,97%
4	(3.94, 4.09)	1,806	1,494	17,28%

Berdasarkan data yang ditampilkan pada Tabel 4.4, dapat disimpulkan bahwa penggunaan Kalman Filter (KF) secara signifikan meningkatkan akurasi prediksi lokasi dengan mengurangi nilai eror jarak prediksi ke titik lokasi asli. (-KF) sendiri memiliki makna jika hasil pengujian tidak menggunakan KF sedangkan untuk (+KF) memiliki makna sebaliknya yaitu menerapkan KF dalam hasil trilaterasi tersebut. Hal ini terlihat dari perbandingan antara jarak prediksi tanpa KF dan dengan KF ke titik asli, di mana penerapan Kalman Filter secara konsisten menunjukkan jarak yang lebih kecil.

Secara keseluruhan, dapat disimpulkan bahwa Kalman Filter efektif dalam meningkatkan akurasi sistem trilaterasi statis, dengan penurunan eror jarak prediksi yang bervariasi antara 17.28% hingga 65.55%, tergantung pada titik lokasi yang diukur. Error jarak prediksi ke titik lokasi asli juga relatif kecil, berada di kisaran 1,494 meter hingga 1,818 meter, menunjukkan bahwa sistem ini cukup andal untuk aplikasi yang memerlukan ketelitian lokasi yang tinggi.

4.2.2 Skenario Uji Coba 2

Pada skenario kedua ini, eksperimen trilaterasi akan dilakukan dalam kondisi bergerak, dengan pengujian yang berlangsung di ruangan yang digambarkan pada Gambar 3.3 dan Gambar 3.4. Kemudian untuk setiap gambar proses pengujian bisa dilihat pada Lampiran E. Selain itu, percobaan akan mencakup perubahan posisi AP untuk memahami pengaruh variasi dalam penempatan AP terhadap akurasi sistem. Selanjutnya, uji coba juga akan dilakukan dalam dua kondisi ruangan yang berbeda: satu dengan ruangan yang berisi berbagai jenis furnitur, dan satu lagi dengan ruangan yang kosong tanpa isi. Dengan melakukan pengujian dalam berbagai kondisi ini, diharapkan dapat diperoleh data yang lebih komprehensif mengenai kinerja sistem trilaterasi dalam lingkungan yang beragam.

Tabel 4.5 Referensi RSSI Ruang 1 pada kondisi kosong

Posisi	AP	RSSI
Posisi 1	AP1	-23
	AP2	-24
	AP3	-25
Posisi 2	AP1	-23
	AP2	-27
	AP3	-23

Tabel 4.6 Referensi RSSI Ruang 2 pada kondisi kosong

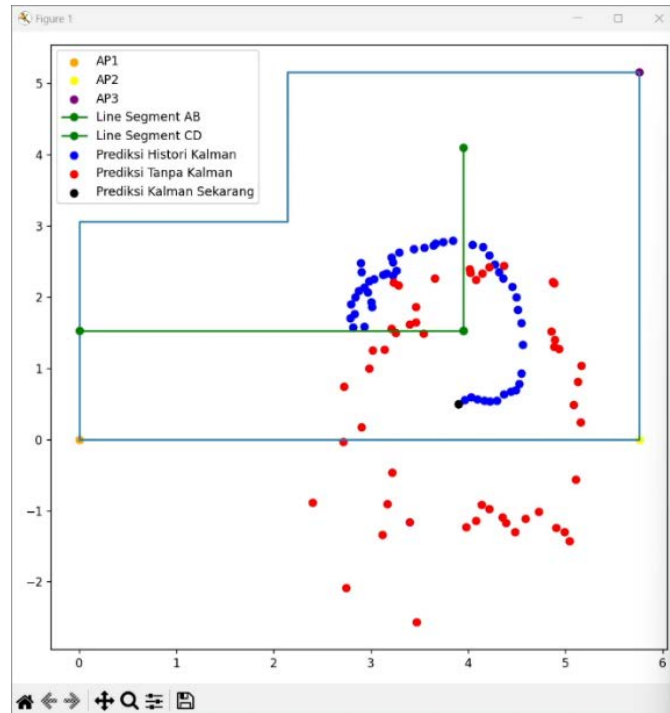
Posisi	AP	RSSI
Posisi 1	AP1	-23
	AP2	-23
	AP3	-21
Posisi 2	AP1	-23
	AP2	-19
	AP3	-20

Tabel 4.7 Nilai N Ruang 1 pada kondisi kosong

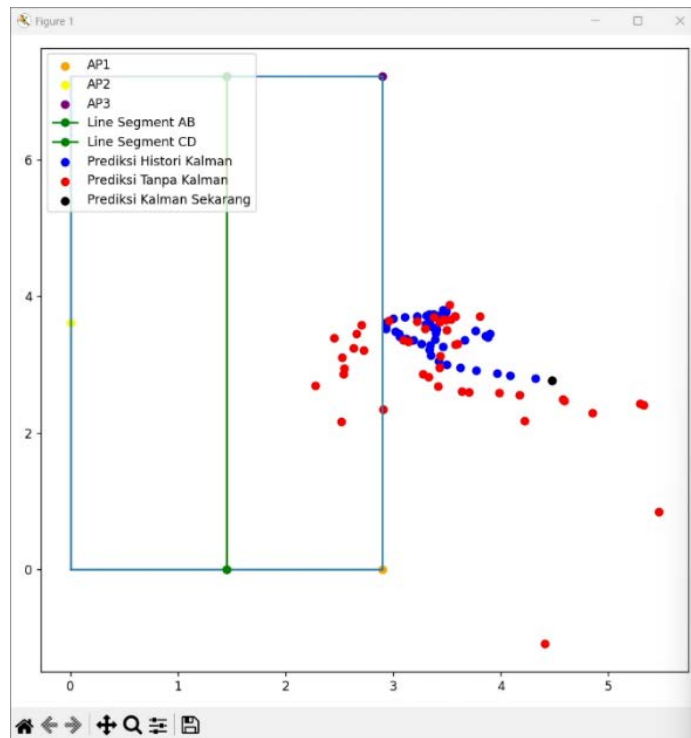
Posisi	AP	N
Posisi 1	AP1	2,7
	AP2	3,05
	AP3	2,29
Posisi 2	AP1	2,7
	AP2	2,46
	AP3	2,55

Tabel 4.8 Nilai N Ruang 2 pada kondisi kosong

Posisi	AP	N
Posisi 1	AP1	3,34
	AP2	2,3
	AP3	2,99
Posisi 2	AP1	3,34
	AP2	2,5
	AP3	4,97



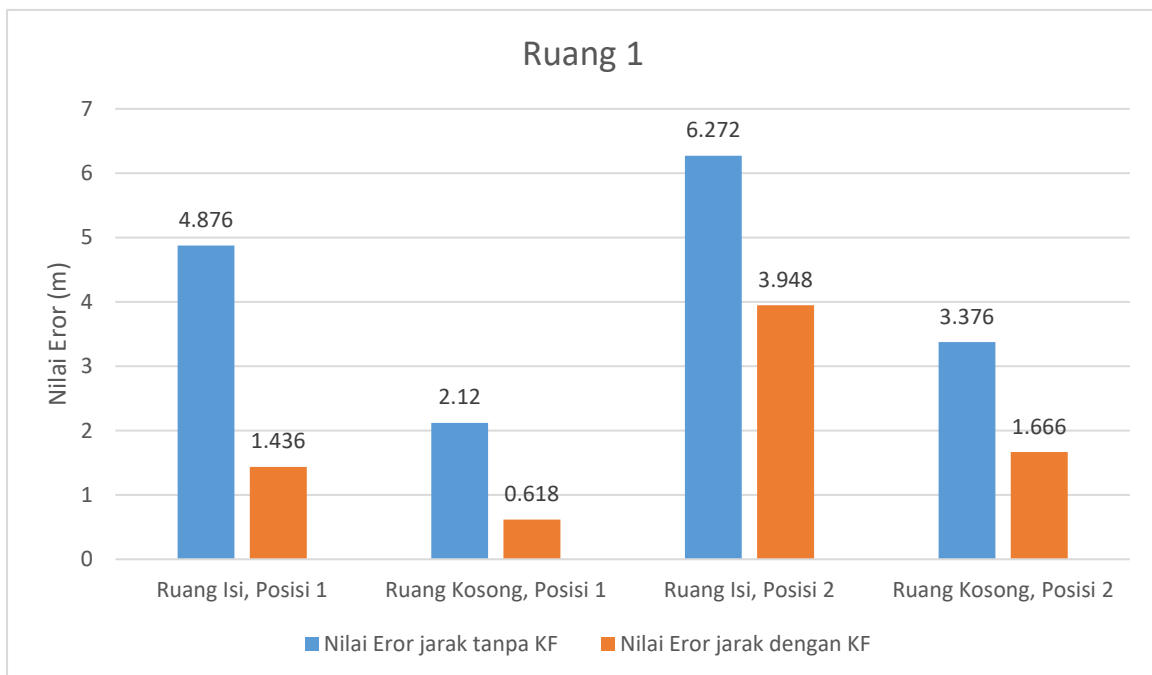
Gambar 4.11 Hasil pengujian IPS pada Ruang 1



Gambar 4.12 Hasil pengujian IPS pada Ruang 2



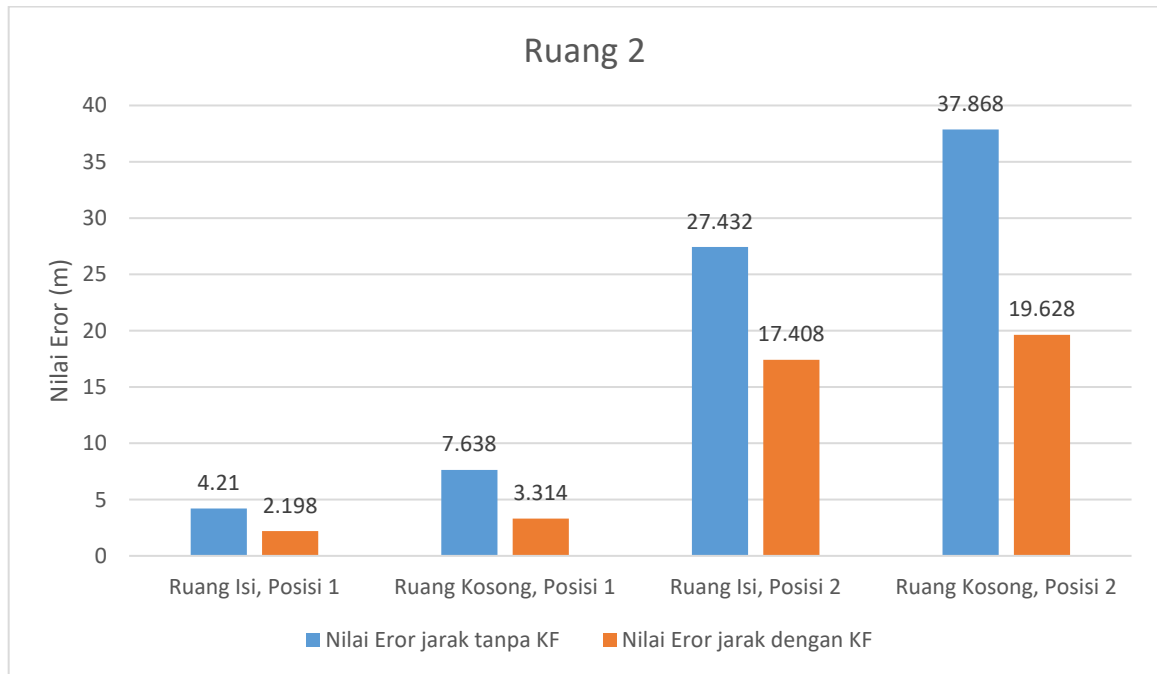
Gambar 4.13 Proses pengujian IPS dengan berjalan mengikuti pola yang ditentukan



Gambar 4.14 Grafik nilai eror IPS pada Ruang 1

Tabel 4.9 Data penurunan eror pada Ruang 1

Posisi	Kondisi Ruang	Rata-Rata Eror Jarak (-KF) (m)	Rata-Rata Eror Jarak (+KF) (m)	Penurunan Nilai Eror
1	Isi	4,876	1,436	70,55%
	Kosong	2,12	0,618	70,85%
2	Isi	6,272	3,948	37,05%
	Kosong	3,376	1,666	50,65%



Gambar 4.15 Grafik nilai error IPS pada Ruang 2

Tabel 4.10 Data penurunan error pada Ruang 2

Posisi	Kondisi Ruang	Rata-Rata Error Jarak (-KF) (m)	Rata-Rata Error Jarak (+KF) (m)	Penurunan Nilai Error
1	Isi	4,21	2,198	47,79%
	Kosong	7,638	3,314	56,61%
2	Isi	27,432	17,408	36,54%
	Kosong	37,868	19,628	48,17%

Dari data yang sudah didapat saat melakukan pengujian, ditemukan bahwa penggunaan filter Kalman dalam IPS yang bergerak menunjukkan dampak signifikan dalam mengurangi kesalahan pengukuran. Berdasarkan data yang disajikan pada Tabel 4.9 dan Tabel 4.10, dapat dilihat bahwa penerapan filter Kalman menghasilkan penurunan rata-rata kesalahan (*average error*) baik pada kondisi ruangan yang diisi maupun yang kosong. Pada Ruang 1, penggunaan filter Kalman mengurangi kesalahan rata-rata hingga 70,55% pada posisi pertama dan 37,05% pada posisi kedua dalam kondisi ruangan terisi, serta 70,85% dan 50,65% pada kondisi ruangan kosong. Demikian pula, pada Ruang 2, filter Kalman mengurangi kesalahan rata-rata hingga 47,79% pada posisi pertama dan 36,54% pada posisi kedua dalam kondisi ruangan terisi, serta

56,61% dan 48,17% pada kondisi ruangan kosong. Penurunan signifikan ini menunjukkan efektivitas filter Kalman dalam meningkatkan akurasi sistem IPS yang bergerak.

Selanjutnya, mengenai kondisi ruangan, baik terisi maupun kosong, memengaruhi hasil sistem IPS yang bergerak. Pada Tabel 4.9, dalam kondisi ruangan terisi, kesalahan rata-rata tanpa filter Kalman mencapai 4,876 meter pada posisi pertama dan 6,272 meter pada posisi kedua, yang kemudian berkurang menjadi 1,436 meter dan 3,948 meter setelah penerapan filter Kalman. Sementara itu, dalam kondisi ruangan kosong, kesalahan rata-rata tanpa filter Kalman adalah 2,12 meter pada posisi pertama dan 3,376 meter pada posisi kedua, yang berkurang menjadi 0,618 meter dan 1,666 meter setelah penerapan filter Kalman. Data dari Tabel 4.10 menunjukkan pola yang berbeda pada Ruang 2. Pada Ruang 2 menunjukkan jika kondisi terisi menghasilkan nilai eror yang lebih kecil dibandingkan kondisi kosong. Hal ini menunjukkan bahwa kondisi ruangan memiliki pengaruh terhadap hasil dari sistem IPS, tetapi penerapan filter Kalman tetap mampu mengurangi kesalahan tersebut secara signifikan.

Berdasarkan hasil uji coba penelitian ini, bentuk ruangan tidak memiliki pengaruh yang signifikan terhadap kinerja sistem IPS yang bergerak. Data yang disajikan menunjukkan bahwa Ruang 1 memiliki bentuk yang lebih kompleks dibandingkan dengan Ruang 2. Namun, nilai eror di Ruang 2 cenderung lebih tinggi dibandingkan dengan Ruang 1, meskipun bentuk Ruang 1 lebih kompleks. Seperti yang telah dijelaskan sebelumnya, penerapan filter Kalman tetap mampu mengurangi nilai kesalahan dalam deteksi lokasi di kedua ruangan tersebut.

Penempatan AP memiliki pengaruh besar terhadap hasil sistem IPS yang bergerak. Dari gambar yang disajikan, terlihat bahwa posisi AP 1, AP 2, dan AP 3 di kedua ruangan berbeda. Penempatan AP yang lebih strategis, memungkinkan pengukuran yang lebih akurat dengan kesalahan yang lebih rendah, baik sebelum maupun setelah penerapan filter Kalman. Sebaliknya, posisi AP yang kurang optimal menyebabkan kesalahan yang lebih besar. Data ini menegaskan pentingnya perencanaan penempatan AP yang tepat untuk memaksimalkan akurasi sistem IPS, terutama dalam lingkungan yang kompleks.

4.3 Diskusi

Hasil dari skenario uji coba pertama menunjukkan bahwa Kalman Filter berhasil mengurangi nilai error pada sistem IPS hingga 1,4 hingga 1,8 pada kondisi statis. Pada skenario uji coba kedua, pengurangan nilai error pada berbagai kondisi juga berhasil dicapai dengan menggunakan Kalman Filter. Selain itu, hipotesis bahwa bentuk ruangan mempengaruhi hasil IPS terbukti tidak benar berdasarkan hasil penelitian ini. Hingga penelitian ini dilakukan pada

tanggal 8 Juli 2024, belum ada penelitian lain yang menguji pengaruh beberapa bentuk ruangan terhadap hasil IPS.

Selanjutnya, hasil penelitian ini menemukan bahwa penempatan Access Point (AP) memiliki pengaruh signifikan terhadap kinerja sistem IPS. Hingga penelitian ini dilakukan pada tanggal 8 Juli 2024, belum ada penelitian lain yang secara spesifik menguji pengaruh penempatan AP dalam sistem IPS. Oleh karena itu, disarankan untuk melakukan riset yang lebih mendalam mengenai strategi penempatan AP guna menghasilkan nilai yang lebih optimal. Rekomendasi ini didasarkan pada temuan bahwa peletakan AP terbukti mempengaruhi hasil IPS secara signifikan dalam pengujian yang dilakukan.

Kemudian, hipotesis lainnya yang diuji menunjukkan bahwa nilai hasil IPS dipengaruhi oleh kondisi ruangan. Penelitian sebelumnya juga mendukung temuan ini. (Spachos & Plataniotis, 2020b) menemukan bahwa akurasi dapat turun drastis ketika ada perubahan signifikan di lingkungan, yang sejalan dengan temuan bahwa hasil IPS dipengaruhi oleh kondisi ruangan. Melakukan penambahan metode penentuan lokasi seperti fingerprinting juga menjadi rekomendasi agar IPS dapat menghasilkan nilai eror yang semakin kecil. Saran ini dipertimbangkan dengan melihat hasil dari (Mari et al., 2018b) yang menunjukkan nilai eror lebih kecil jika trilaterasi dan fingerprinting digabungkan.

(Fariz et al., 2018b) menunjukkan bahwa akurasi titik lokasi dapat ditingkatkan oleh filter kalman, memperkuat hasil uji coba yang menunjukkan penurunan nilai eror. Selain itu, (Wibisono & Wicaksono, 2022c) menyatakan bahwa peningkatan akurasi dapat dicapai melalui kalibrasi signal propagation model, dan peningkatan lebih lanjut dapat diberikan oleh penggunaan kalman filter, mendukung hasil bahwa filter ini efektif dalam berbagai kondisi. Dengan demikian, hasil penelitian ini tidak hanya menunjukkan keunggulan kalman filter dalam mengurangi nilai eror pada sistem IPS, tetapi juga konsisten dengan penelitian sebelumnya yang menyoroti pentingnya kalibrasi dan penyesuaian lingkungan untuk meningkatkan akurasi.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan data yang disajikan, penggunaan Kalman Filter secara signifikan meningkatkan akurasi prediksi lokasi pada IPS dengan metode trilaterasi dalam kondisi statis dan bergerak. Pada Tabel 4.4, terlihat bahwa penerapan Kalman Filter secara konsisten mengurangi nilai eror jarak prediksi ke titik lokasi asli, dengan penurunan eror berkisar antara 2.93% hingga 59.62%, tergantung pada titik lokasi yang diukur. Efektivitas Kalman Filter dalam meningkatkan keandalan sistem trilaterasi statis ditunjukkan dengan eror jarak prediksi ke titik lokasi asli yang relatif kecil, yaitu antara 0.21 meter hingga 2.65 meter.

Dalam konteks IPS yang bergerak, Tabel 4.9 dan Tabel 4.10 menunjukkan penurunan rata-rata kesalahan yang signifikan setelah penerapan Kalman Filter. Baik dalam kondisi ruangan terisi maupun kosong, penggunaan Kalman Filter terbukti efektif dalam mengurangi kesalahan pengukuran. Di Ruang 1, kesalahan rata-rata berkurang hingga 70,55% pada posisi pertama dan 37,05% pada posisi kedua dalam kondisi ruangan terisi, serta 70,85% dan 50,65% dalam kondisi ruangan kosong. Sementara itu, di Ruang 2, kesalahan rata-rata berkurang hingga 46,17% pada posisi pertama dan 36,54% pada posisi kedua dalam kondisi ruangan terisi, serta 56,61% dan 48,17% dalam kondisi ruangan kosong. Penurunan signifikan ini menunjukkan efektivitas Kalman Filter dalam meningkatkan akurasi sistem IPS yang bergerak.

Kondisi ruangan, baik terisi maupun kosong, memengaruhi hasil sistem IPS yang bergerak secara signifikan. Dalam kondisi ruangan terisi, pada Tabel 4.9, kesalahan rata-rata tanpa penggunaan Kalman Filter mencapai 4,876 meter pada posisi pertama dan 6,272 meter pada posisi kedua, yang berkurang menjadi 1,436 meter dan 3,948 meter setelah penerapan Kalman Filter. Dalam kondisi ruangan kosong, kesalahan rata-rata tanpa Kalman Filter adalah 2,12 meter pada posisi pertama dan 3,376 meter pada posisi kedua, yang berkurang menjadi 0,618 meter dan 1,666 meter setelah penerapan Kalman Filter. Data dari Tabel 4.10 menunjukkan pola yang berbeda di Ruang 2. Data menunjukkan bahwa ruangan terisi cenderung menghasilkan kesalahan pengukuran lebih kecil dibandingkan ruangan kosong. Hal ini menyimpulkan jika Kalman Filter tetap mampu mengurangi kesalahan tersebut secara signifikan di kedua kondisi tersebut, kemudian ruangan kondisi terisi dan kosong memiliki pengaruh juga terhadap hasil IPS.

Penelitian ini juga menemukan bahwa bentuk ruangan tidak memiliki pengaruh signifikan terhadap kinerja sistem IPS yang bergerak. Meskipun Ruang 1 memiliki bentuk lebih kompleks dibandingkan Ruang 2, nilai eror di Ruang 2 cenderung lebih tinggi dibandingkan di Ruang 1. Namun, Kalman Filter tetap mampu mengurangi nilai kesalahan dalam deteksi lokasi di kedua ruangan tersebut.

Penempatan AP memiliki pengaruh besar terhadap hasil sistem IPS yang bergerak. Posisi AP 1, AP 2, dan AP 3 yang strategis memungkinkan pengukuran lebih akurat dengan kesalahan lebih rendah. Sebaliknya, posisi AP yang kurang optimal menyebabkan kesalahan lebih besar. Data ini menegaskan pentingnya perencanaan penempatan AP yang tepat untuk memaksimalkan akurasi sistem IPS, terutama dalam lingkungan yang kompleks. Nilai eror dari dampak penempatan AP yang bervariasi tersebut tetap bisa diantisipasi menggunakan Kalman Filter.

Secara keseluruhan, Kalman Filter terbukti efektif dalam meningkatkan akurasi sistem IPS baik dalam kondisi statis maupun bergerak. Efektivitas Kalman Filter terlihat dari penurunan eror yang signifikan pada berbagai kondisi ruangan. Selain itu, perencanaan penempatan AP yang baik sangat diperlukan untuk hasil optimal. Dapat disimpulkan bahwa penggunaan Kalman Filter dan penempatan AP yang strategis adalah kunci untuk mengoptimalkan kinerja sistem IPS.

5.2 Saran

Penelitian ini masih memiliki berbagai kekurangan. Oleh karena itu, agar penelitian selanjutnya dapat lebih ditingkatkan, berikut beberapa saran yang dapat dipertimbangkan, yaitu:

- a. Melakukan pengujian IPS apabila posisi AP ditempatkan di lantai yang berbeda, guna memastikan kualitas dan kestabilan jaringan yang optimal.
- b. Melakukan pengujian dengan objek yang sedang berlari dan bergerak secara acak, agar dapat mengukur performa sistem di kondisi yang lebih variatif.

DAFTAR PUSTAKA

- Andika Satrugna Mahardhika, M., Gautama Putra, A., & Abdurrohman, M. (2020a). Improving Indoor Positioning Systems Accuracy in Closed Buildings with Kalman Filter and Feedback Filter. *2020 Fifth International Conference on Informatics and Computing (ICIC)*, 1–8. IEEE. <https://doi.org/10.1109/ICIC50835.2020.9288539>
- Andika Satrugna Mahardhika, M., Gautama Putra, A., & Abdurrohman, M. (2020b). Improving Indoor Positioning Systems Accuracy in Closed Buildings with Kalman Filter and Feedback Filter. *2020 Fifth International Conference on Informatics and Computing (ICIC)*, 1–8. IEEE. <https://doi.org/10.1109/ICIC50835.2020.9288539>
- Batistic, L., & Tomic, M. (2018). Overview of indoor positioning system technologies. *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 0473–0478. IEEE. <https://doi.org/10.23919/MIPRO.2018.8400090>
- Bottiglieri, S., Milanesio, D., Saccani, M., & Maggiora, R. (2021). A Low-Cost Indoor Real-Time Locating System Based on TDOA Estimation of UWB Pulse Sequences. *IEEE Transactions on Instrumentation and Measurement*, 70, 1–11. <https://doi.org/10.1109/TIM.2021.3069486>
- Chouhan, R. S., Rajawat, A. S., Goyal, S. B., Bedi, P., Constantin, N. B., Raboaca, M. S., & Verma, C. (2022a). Experimental Analysis for Position Estimation using Trilateration and RSSI in Industry 4.0. *2022 11th International Conference on System Modeling & Advancement in Research Trends (SMART)*, 904–908. IEEE. <https://doi.org/10.1109/SMART55829.2022.10047276>
- Chouhan, R. S., Rajawat, A. S., Goyal, S. B., Bedi, P., Constantin, N. B., Raboaca, M. S., & Verma, C. (2022b). Experimental Analysis for Position Estimation using Trilateration and RSSI in Industry 4.0. *2022 11th International Conference on System Modeling & Advancement in Research Trends (SMART)*, 904–908. IEEE. <https://doi.org/10.1109/SMART55829.2022.10047276>
- Fariz, N., Jamil, N., Md Din, M., Ezanee Rusli, M., Sharudin, Z., & Afendee Mohamed, M. (2018a). An Improved Indoor Location Technique using Kalman Filter. *International Journal of Engineering & Technology*, 7(2.14), 1. <https://doi.org/10.14419/ijet.v7i2.14.11141>
- Fariz, N., Jamil, N., Md Din, M., Ezanee Rusli, M., Sharudin, Z., & Afendee Mohamed, M. (2018b). An Improved Indoor Location Technique using Kalman Filter. *International*

Journal of Engineering & Technology, 7(2.14), 1.
<https://doi.org/10.14419/ijet.v7i2.14.11141>

- Firmansyah, F., Rahma, F., Irianto, K. D., & Shiddiqi, A. M. (2024). Indoor Positioning System: A Brief Review of Its Technologies and Signal-Filtering Techniques. *2024 International Conference on Smart Computing, IoT and Machine Learning (SIML)*, 1–7. IEEE. <https://doi.org/10.1109/SIML61815.2024.10578093>
- Lassabe, F., Canalda, P., Chatonnay, P., Spies, F., & Baala, O. (2005). A Friis-based calibrated model for WiFi terminals positioning. *Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks*, 382–387. IEEE. <https://doi.org/10.1109/WOWMOM.2005.2>
- Mari, S. K., Kiong, L. C., & Loong, H. K. (2018a). A hybrid trilateration and fingerprinting approach for indoor localization based on wifi. *2018 Fourth International Conference on Advances in Computing, Communication & Automation (ICACCA)*, 1–6. IEEE. <https://doi.org/10.1109/ICACCAF.2018.8776729>
- Mari, S. K., Kiong, L. C., & Loong, H. K. (2018b). A hybrid trilateration and fingerprinting approach for indoor localization based on wifi. *2018 Fourth International Conference on Advances in Computing, Communication & Automation (ICACCA)*, 1–6. IEEE. <https://doi.org/10.1109/ICACCAF.2018.8776729>
- Pakanon, N., Chamchoy, M., & Supanakoon, P. (2020). Study on Accuracy of Trilateration Method for Indoor Positioning with BLE Beacons. *2020 6th International Conference on Engineering, Applied Sciences and Technology (ICEAST)*, 1–4. IEEE. <https://doi.org/10.1109/ICEAST50382.2020.9165464>
- Sohan, A. A., Ali, M., Fairouz, F., Rahman, A. I., Chakrabarty, A., & Kabir, Md. R. (2019). Indoor Positioning Techniques using RSSI from Wireless Devices. *2019 22nd International Conference on Computer and Information Technology (ICCIT)*, 1–6. IEEE. <https://doi.org/10.1109/ICCIT48885.2019.9038591>
- Spachos, P., & Plataniotis, K. N. (2020a). BLE Beacons for Indoor Positioning at an Interactive IoT-Based Smart Museum. *IEEE Systems Journal*, 14(3), 3483–3493. <https://doi.org/10.1109/JSYST.2020.2969088>
- Spachos, P., & Plataniotis, K. N. (2020b). BLE Beacons for Indoor Positioning at an Interactive IoT-Based Smart Museum. *IEEE Systems Journal*, 14(3), 3483–3493. <https://doi.org/10.1109/JSYST.2020.2969088>

- Subedi, S., & Pyun, J.-Y. (2020). A Survey of Smartphone-Based Indoor Positioning System Using RF-Based Wireless Technologies. *Sensors*, 20(24), 7230. <https://doi.org/10.3390/s20247230>
- Sujana, D. (2018). *Android Smartphone Location Detection on Indoor Using Trilateration Method and Kalman Filter*. 5, 29–35jo.
- Wang, T., Han, L., Kong, Q., Li, Z., Li, C., Han, J., ... Chen, Y. (2023a). An improved particle filter indoor fusion positioning approach based on Wi-Fi/ PDR/ geomagnetic field. *Defence Technology*. <https://doi.org/10.1016/j.dt.2023.03.021>
- Wang, T., Han, L., Kong, Q., Li, Z., Li, C., Han, J., ... Chen, Y. (2023b). An improved particle filter indoor fusion positioning approach based on Wi-Fi/ PDR/ geomagnetic field. *Defence Technology*. <https://doi.org/10.1016/j.dt.2023.03.021>
- Wibisono, W., & Wicaksono, A. D. (2022a). An Improved Accuracy of Indoor Positioning System Based on Trilateration using Kalman Filter. *2022 10th International Conference on Information and Communication Technology (ICoICT)*, 174–178. IEEE. <https://doi.org/10.1109/ICoICT55009.2022.9914863>
- Wibisono, W., & Wicaksono, A. D. (2022b). An Improved Accuracy of Indoor Positioning System Based on Trilateration using Kalman Filter. *2022 10th International Conference on Information and Communication Technology (ICoICT)*, 174–178. IEEE. <https://doi.org/10.1109/ICoICT55009.2022.9914863>
- Wibisono, W., & Wicaksono, A. D. (2022c). An Improved Accuracy of Indoor Positioning System Based on Trilateration using Kalman Filter. *2022 10th International Conference on Information and Communication Technology (ICoICT)*, 174–178. IEEE. <https://doi.org/10.1109/ICoICT55009.2022.9914863>

LAMPIRAN A

A. Kode Utama.py

```
import os
import multiprocessing
import signal

def run_script(script_name):
    try:
        os.system(f"python {script_name}")
    except KeyboardInterrupt:
        pass # Tangani KeyboardInterrupt di sini

if __name__ == "__main__":
    signal.signal(signal.SIGINT, signal.SIG_IGN) # Menangani sinyal
    KeyboardInterrupt untuk proses anak

    # Membuat proses untuk tes2.py
    p1 = multiprocessing.Process(target=run_script, args=("Wifi_RssiV6.py",))
    p1.start()

    # Membuat proses untuk tes3.py
    p2 = multiprocessing.Process(target=run_script, args=("Animate.py",))
    p2.start()
    print("-----START-----")

    # Menunggu kedua proses selesai
    p1.join()
    p2.join()
```

B. Kode Animate.py

```
import signal
import sys
import time
import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation
import pandas as pd
from math import sqrt
from multiprocessing import Lock
import csv

def signal_handler(sig, frame):
    print("\n-----EXIT-----")
    sys.exit(0)

def minDistance(A, B, E):
    AB = [B[0] - A[0], B[1] - A[1]]
    BE = [E[0] - B[0], E[1] - B[1]]
    AE = [E[0] - A[0], E[1] - A[1]]

    AB_BE = AB[0] * BE[0] + AB[1] * BE[1]
    AB_AE = AB[0] * AE[0] + AB[1] * AE[1]

    reqAns = 0

    if AB_BE > 0:
        reqAns = sqrt(BE[0] * BE[0] + BE[1] * BE[1])
    elif AB_AE < 0:
        reqAns = sqrt(AE[0] * AE[0] + AE[1] * AE[1])
    else:
```

```

    reqAns = abs(AB[0] * AE[1] - AB[1] * AE[0]) / sqrt(AB[0] * AB[0] + AB[1] *
AB[1])

    return reqAns

def animate(i):
    try:
        dataKF = pd.read_csv('outputTrilaterasiKF.csv')
        data = pd.read_csv('outputTrilaterasi.csv')
        # Inisialisasi objek Lock
        lock = Lock()
        fieldnames = ["error"]
        if not dataKF.empty and not data.empty:
            xKF = dataKF['x']
            yKF = dataKF['y']

            x = data['x']
            y = data['y']

            plt.cla()

            # Gambar persegi panjang
            plt.plot([0, 5.76, 5.76, 2.14, 2.14, 0, 0], [0, 0, 5.16, 5.16, 3.06,
3.06, 0])

            A = [0, 1.53]
            B = [3.95, 1.53]
            C = [3.95, 1.53]
            D = [3.95, 4.10]
            EKf = [dataKF['x'].iloc[-1], dataKF['y'].iloc[-1]]
            E = [data['x'].iloc[-1], data['y'].iloc[-1]]

            # Calculate minimum distances
            min_dist_ABKf = minDistance(A, B, EKf)
            min_dist_CDKf = minDistance(C, D, EKf)
            if min_dist_ABKf < min_dist_CDKf:
                with lock: # Menggunakan lock untuk mengamankan akses ke file CSV
                    with open("outputErrorTitikGaris(KF).csv", 'a') as csv_file:
                        csv_writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
                        info = {
                            "error": min_dist_ABKf,
                        }
                        csv_writer.writerow(info)
                    print("Jarak Titik ke Garis Jalan (KF):", min_dist_ABKf)
            elif min_dist_ABKf > min_dist_CDKf:
                with lock: # Menggunakan lock untuk mengamankan akses ke file CSV
                    with open("outputErrorTitikGaris(KF).csv", 'a') as csv_file:
                        csv_writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
                        info = {
                            "error": min_dist_CDKf,
                        }
                        csv_writer.writerow(info)
                    print("Jarak Titik ke Garis Jalan (KF):", min_dist_CDKf)
            elif min_dist_ABKf == min_dist_CDKf:
                with lock: # Menggunakan lock untuk mengamankan akses ke file CSV
                    with open("outputErrorTitikGaris(KF).csv", 'a') as csv_file:
                        csv_writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
                        info = {
                            "error": min_dist_ABKf,
                        }
                        csv_writer.writerow(info)
                    print("Jarak Titik ke Garis Jalan (KF):", min_dist_ABKf)

            # Calculate minimum distances
            min_dist_AB = minDistance(A, B, E)
            min_dist_CD = minDistance(C, D, E)
            if min_dist_AB < min_dist_CD:

```

```

        with lock: # Menggunakan lock untuk mengamankan akses ke file CSV
            with open("outputErrorTitikGaris.csv", 'a') as csv_file:
                csv_writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
                info = {
                    "error": min_dist_AB,
                }
                csv_writer.writerow(info)
            print("Jarak Titik ke Garis Jalan :", min_dist_AB)
    elif min_dist_AB > min_dist_CD:
        with lock: # Menggunakan lock untuk mengamankan akses ke file CSV
            with open("outputErrorTitikGaris.csv", 'a') as csv_file:
                csv_writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
                info = {
                    "error": min_dist_CD,
                }
                csv_writer.writerow(info)
            print("Jarak Titik ke Garis Jalan :", min_dist_CD)
    elif min_dist_AB == min_dist_CD:
        with lock: # Menggunakan lock untuk mengamankan akses ke file CSV
            with open("outputErrorTitikGaris.csv", 'a') as csv_file:
                csv_writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
                info = {
                    "error": min_dist_AB,
                }
                csv_writer.writerow(info)
            print("Jarak Titik ke Garis Jalan :", min_dist_AB)

    plt.plot([A[0], B[0]], [A[1], B[1]], marker='o', label='Line Segment
AB',color='green')
    plt.plot([C[0], D[0]], [C[1], D[1]], marker='o', label='Line Segment
CD',color='green')

    plt.scatter(xKF, yKF, label='Prediksi Histori Kalman', color='blue') #
Menggunakan plt.scatter() untuk memplot titik-titik
    plt.scatter(x, y, label='Prediksi Tanpa Kalman', color='red') #
Menggunakan plt.scatter() untuk memplot titik-titik
    plt.scatter(dataKF['x'].iloc[-1], dataKF['y'].iloc[-1], label='Prediksi
Kalman Sekarang', color='black') # Menggunakan plt.scatter() untuk memplot titik-
titik

    plt.scatter(0, 0, label = "AP1", color = 'orange')
    plt.scatter(2.14, 3.06, label = "AP2", color = 'yellow')
    plt.scatter(5.76, 3.06, label = "AP3", color = 'purple')

    plt.legend(loc='upper left')
    plt.tight_layout()
except Exception as e:
    pass

if __name__ == "__main__":
    signal.signal(signal.SIGINT, signal_handler)

    fig, ax = plt.subplots()
    ani = FuncAnimation(fig, animate, interval=1000, cache_frame_data=False)

    fieldnames = ["error"]
    outputErrorTitikGarisKf = f"outputErrorTitikGaris(KF).csv"
    with open(outputErrorTitikGarisKf, 'w') as csv_file:
        csv_writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
        csv_writer.writeheader()

    outputErrorTitikGaris = f"outputErrorTitikGaris.csv"
    with open(outputErrorTitikGaris, 'w') as csv_file:
        csv_writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
        csv_writer.writeheader()

try:

```

```

plt.tight_layout()
plt.show()
except KeyboardInterrupt:
    print("Keyboard interrupt detected. Closing the plot...")
    plt.close() # Menutup plot saat menerima sinyal KeyboardInterrupt

```

C. Kode Wifi_RssiV6.py

```

import signal
import sys

def signal_handler(sig, frame):
    print("\n-----EXIT-----")
    sys.exit(0)

# Penanganan sinyal SIGINT (Ctrl+C)
signal.signal(signal.SIGINT, signal_handler)

try:
    from pywifi import PyWiFi, const
    import time
    import pywifi
    import math
    from math import sin, cos, sqrt, atan2, radians, pi
    from filterpy.kalman import KalmanFilter
    import numpy as np
    from multiprocessing import Process, Event, Manager, Lock
    import matplotlib.pyplot as plt
    from matplotlib.animation import FuncAnimation
    import csv
    import pandas as pd
except ImportError as e:
    print("Modul yang diperlukan belum diinstal:", e)
    print("Silakan instal modul yang diperlukan dengan menggunakan pip.")
    exit(1)

reference_rssi_dict = {
    "RuijieAP1": -24, # RSSI pada jarak referensi untuk SSID RuijieAP1 (dalam dBm)
    "RuijieAP2": -33, # RSSI pada jarak referensi untuk SSID RuijieAP2 (dalam dBm)
    "RuijieAP3": -21, # RSSI pada jarak referensi untuk SSID RuijieAP3 (dalam dBm)
}

n_dict = {
    "RuijieAP1": 1.68, # RSSI pada jarak referensi untuk SSID RuijieAP1 (dalam dBm)
    "RuijieAP2": 1.11, # RSSI pada jarak referensi untuk SSID RuijieAP2 (dalam dBm)
    "RuijieAP3": 3.42, # RSSI pada jarak referensi untuk SSID RuijieAP3 (dalam dBm)
}

beacon_positions = {
    "RuijieAP1": np.array([0, 0]),
    "RuijieAP2": np.array([2.14, 3.06]),
    "RuijieAP3": np.array([5.76, 3.06])
}

# Inisialisasi Filter Kalman
kfAP1 = KalmanFilter(dim_x=1, dim_z=1)
kfAP1.F = np.array([[1.0]]) # Matriks transisi
kfAP1.H = np.array([[1.0]]) # Matriks pengukuran
kfAP1.R = 5 # Kovariansi pengukuran (dalam contoh ini, disesuaikan dengan skala pengukuran)
kfAP1.P = np.eye(1) # Matriks kovariansi awal (dalam contoh ini, matriks identitas)

kfAP2 = KalmanFilter(dim_x=1, dim_z=1)
kfAP2.F = np.array([[1.0]]) # Matriks transisi

```

```

kfAP2.H = np.array([[1.0]]) # Matriks pengukuran
kfAP2.R = 5 # Kovariansi pengukuran (dalam contoh ini, disesuaikan dengan skala
pengukuran)
kfAP2.P = np.eye(1) # Matriks kovariansi awal (dalam contoh ini, matriks identitas)

kfAP3 = KalmanFilter(dim_x=1, dim_z=1)
kfAP3.F = np.array([[1.0]]) # Matriks transisi
kfAP3.H = np.array([[1.0]]) # Matriks pengukuran
kfAP3.R = 5 # Kovariansi pengukuran (dalam contoh ini, disesuaikan dengan skala
pengukuran)
kfAP3.P = np.eye(1) # Matriks kovariansi awal (dalam contoh ini, matriks identitas)
def calculate_distanceKalmanAP1(rssi, ssid, ssidN, data_count):
    """
    Calculate the distance based on RSSI, reference RSSI (A), and path loss exponent
    (n).

    Parameters:
        RSSI (float): Received Signal Strength Indication.
        A (float): Reference RSSI at 1 meter.
        n (float): Path loss exponent.

    Returns:
        float: Distance between transmitter and receiver.
    """
    # Update Filter Kalman dengan pengukuran RSSI
    reference_rssi = reference_rssi_dict[ssid]
    n = n_dict[ssidN]
    if data_count == 0:
        kfAP1.x = np.array([reference_rssi])
    kfAP1.predict()
    kfAP1.update(np.array([rssi]))

    estimated_rssi = kfAP1.x[0]
    distanceKalman = 10 ** ((reference_rssi - estimated_rssi) / (10 * n))
    return distanceKalman

def calculate_distanceKalmanAP2(rssi, ssid, ssidN, data_count):
    """
    Calculate the distance based on RSSI, reference RSSI (A), and path loss exponent
    (n).

    Parameters:
        RSSI (float): Received Signal Strength Indication.
        A (float): Reference RSSI at 1 meter.
        n (float): Path loss exponent.

    Returns:
        float: Distance between transmitter and receiver.
    """
    # Update Filter Kalman dengan pengukuran RSSI
    reference_rssi = reference_rssi_dict[ssid]
    n = n_dict[ssidN]
    if data_count == 0:
        kfAP2.x = np.array([reference_rssi])
    kfAP2.predict()
    kfAP2.update(np.array([rssi]))

    estimated_rssi = kfAP2.x[0]
    distanceKalman = 10 ** ((reference_rssi - estimated_rssi) / (10 * n))
    return distanceKalman

def calculate_distanceKalmanAP3(rssi, ssid, ssidN, data_count):
    """
    Calculate the distance based on RSSI, reference RSSI (A), and path loss exponent
    (n).

    Parameters:

```

```

    RSSI (float): Received Signal Strength Indication.
    A (float): Reference RSSI at 1 meter.
    n (float): Path loss exponent.

Returns:
    float: Distance between transmitter and receiver.
"""
# Update Filter Kalman dengan pengukuran RSSI
reference_rssi = reference_rssi_dict[ssid]
n = n_dict[ssidN]
if data_count == 0:
    kfAP3.x = np.array([reference_rssi])
kfAP3.predict()
kfAP3.update(np.array([rssi]))

estimated_rssi = kfAP3.x[0]
distanceKalman = 10 ** ((reference_rssi - estimated_rssi) / (10 * n))
return distanceKalman

def calculate_distance(RSSI, ssid, ssidN):
    """
    Calculate the distance based on RSSI, reference RSSI (A), and path loss exponent
    (n).

    Parameters:
        RSSI (float): Received Signal Strength Indication.
        A (float): Reference RSSI at 1 meter.
        n (float): Path loss exponent.

    Returns:
        float: Distance between transmitter and receiver.
    """
    A = reference_rssi_dict[ssid]
    n = n_dict[ssidN]
    d = 10 ** ((A - RSSI) / (10 * n))
    return d

def calculate_path_loss_exponent(RSSI, ssid, d):
    """
    Calculate the path loss exponent (n) based on RSSI, reference RSSI (A), and
    distance (d).

    Parameters:
        RSSI (float): Received Signal Strength Indication.
        A (float): Reference RSSI at 1 meter.
        d (float): Distance between transmitter and receiver.

    Returns:
        float: Path loss exponent (n).
    """
    A = reference_rssi_dict[ssid]
    n = -1*(RSSI - A) / (10 * math.log10(d))
    return n

def trilateration(ssid1,rAP1,ssid2,rAP2,ssid3,rAP3):
    """
    Trilateration untuk menentukan posisi pengguna berdasarkan sinyal yang diterima
    dari beberapa beacon.

    Args:
        beacons (list): List koordinat beacon dalam format (x, y).
        distances (list): List jarak antara pengguna dan masing-masing beacon.

    Returns:
        tuple: Koordinat perkiraan posisi pengguna dalam format (x, y).
    """
    xAP1 = beacon_positions[ssid1][0]

```

```

yAP1 = beacon_positions[ssid1][1]
xAP2 = beacon_positions[ssid2][0]
yAP2 = beacon_positions[ssid2][1]
xAP3 = beacon_positions[ssid3][0]
yAP3 = beacon_positions[ssid3][1]

A = 2*xAP2 - 2*xAP1
B = 2*yAP2 - 2*yAP1
C = rAP1**2 - rAP2**2 - xAP1**2 + xAP2**2 - yAP1**2 + yAP2**2
D = 2*xAP3 - 2*xAP2
E = 2*yAP3 - 2*yAP2
F = rAP2**2 - rAP3**2 - xAP2**2 + xAP3**2 - yAP2**2 + yAP3**2
x = (C*E - F*B) / (E*A - B*D)
y = (C*D - A*F) / (B*D - A*E)
return x,y

def ambilDataKF(ssid,lock,outputTrilaterasiKF,kolom):
    data = pd.read_csv('outputKF.csv')
    AP1 = data[ssid[0]]
    AP2 = data[ssid[1]]
    AP3 = data[ssid[2]]

    meanAP1 = np.nanmean(AP1)
    print(f"Rata-Rata Jarak {ssid[0]} ke titik:",meanAP1)
    meanAP2 = np.nanmean(AP2)
    print(f"Rata-Rata Jarak {ssid[1]} ke titik:", meanAP2)
    meanAP3 = np.nanmean(AP3)
    print(f"Rata-Rata Jarak {ssid[2]} ke titik:", meanAP3)

    x,y = trilateration(ssid[0],meanAP1,ssid[1],meanAP2,ssid[2],meanAP3)
    print("Trilaterasi Kalman(x,y)=",x,y)

    with lock: # Menggunakan lock untuk mengamankan akses ke file CSV
        with open(outputTrilaterasiKF, 'a') as csv_file:
            csv_writer = csv.DictWriter(csv_file, fieldnames=kolom)
            info = {
                "x": x,
                "y": y
            }
            csv_writer.writerow(info)

def ambilData(ssid,lock,outputTrilaterasi,kolom):
    data = pd.read_csv('outputJarak.csv')
    AP1 = data[ssid[0]]
    AP2 = data[ssid[1]]
    AP3 = data[ssid[2]]

    meanAP1 = np.nanmean(AP1)
    meanAP2 = np.nanmean(AP2)
    meanAP3 = np.nanmean(AP3)

    x,y = trilateration(ssid[0],meanAP1,ssid[1],meanAP2,ssid[2],meanAP3)
    print("Kalkulasi Tanpa Kalman (x,y)=",x,y)

    with lock: # Menggunakan lock untuk mengamankan akses ke file CSV
        with open(outputTrilaterasi, 'a') as csv_file:
            csv_writer = csv.DictWriter(csv_file, fieldnames=kolom)
            info = {
                "x": x,
                "y": y
            }
            csv_writer.writerow(info)

def scan_location(ssid, outputRSSI, outputJarak, outputKalmanFilter, stop_event,
lock, outputTrilaterasiKF,kolom,outputTrilaterasi):
    wifi = pywifi.PyWiFi()
    iface = wifi.interfaces()[0] # Menggunakan antarmuka pertama (biasanya wlan0)

```

```

try:
    # Inisialisasi hitungan data yang diambil
    data_count = 0
    AP1 = ssid[0]
    AP2 = ssid[1]
    AP3 = ssid[2]

    countAP1 = 0
    countAP2 = 0
    countAP3 = 0

    countKFAP1=0
    countKFAP2=0
    countKFAP3=0
    while not stop_event.is_set():
        # Menambah jumlah data yang diambil
        data_count += 1
        iface.scan()
        time.sleep(1)
        scan_results = iface.scan_results()
        countAP1 = 0
        countAP2 = 0
        countAP3 = 0

print("=====")
    for result in scan_results:
        if countAP1 == 0:
            if AP1 in result.ssid:
                countAP1 += 1
                # Fungsi untuk mendapatkan nilai RSSI
                rssi = result.signal
                with open(outputRSSI, "a") as file:
                    file.write(f"RSSI {AP1} = {rssi} dBm\n")
                print(f"{data_count}.RSSI {AP1} = {rssi} dBm")

                # Fungsi untuk mendapatkan nilai jarak TANPA Kalman Filter
                distanceAP1 = calculate_distance(rssi, AP1, AP1)
                print(f"Jarak {AP1} = {distanceAP1} meter")

                # Fungsi untuk menghitung jarak device ke AP yang sudah
                menggunakan kalman filter
                distanceKalmanAP1 = calculate_distanceKalmanAP1(rssi, AP1,
AP1, countKFAP1)
                print(f"Jarak KF {AP1} = {distanceKalmanAP1} meter")

                countKFAP1 +=1
            if countAP2 == 0:
                if AP2 in result.ssid:
                    countAP2 +=1
                    # Fungsi untuk mendapatkan nilai RSSI
                    rssi = result.signal
                    with open(outputRSSI, "a") as file:
                        file.write(f"RSSI {AP2} = {rssi} dBm\n")
                    print(f"{data_count}.RSSI {AP2} = {rssi} dBm")

                    # Fungsi untuk mendapatkan nilai jarak TANPA Kalman Filter
                    distanceAP2 = calculate_distance(rssi, AP2, AP2)
                    print(f"Jarak {AP2} = {distanceAP2} meter")

                    # Fungsi untuk menghitung jarak device ke AP yang sudah
                    menggunakan kalman filter
                    distanceKalmanAP2 = calculate_distanceKalmanAP2(rssi, AP2,
AP2, countKFAP2)
                    print(f"Jarak KF {AP2} = {distanceKalmanAP2} meter")

                    countKFAP2 +=1

```

```

        if countAP3 == 0:
            if AP3 in result.ssid:
                countAP3 +=1
                # Fungsi untuk mendapatkan nilai RSSI
                rssi = result.signal
                with open(outputRSSI, "a") as file:
                    file.write(f"RSSI {AP3} = {rssi} dBm\n")
                print(f"{data_count}.RSSI {AP3} = {rssi} dBm")

                # Fungsi untuk mendapatkan nilai jarak TANPA Kalman Filter
                distanceAP3 = calculate_distance(rssi, AP3, AP3)
                print(f"Jarak {AP3} = {distanceAP3} meter")

                # Fungsi untuk menghitung jarak device ke AP yang sudah
                menggunakan kalman filter
                distanceKalmanAP3 = calculate_distanceKalmanAP3(rssi, AP3,
                AP3, countKFAP3)
                print(f"Jarak KF {AP3} = {distanceKalmanAP3} meter")

                countKFAP3 +=1
    for AP in ssid:
        if countAP1 == 0:
            print(f"AP {AP1} Tidak Ditemukan")
            exit()
        if countAP2 == 0:
            print(f"AP {AP2} Tidak Ditemukan")
            exit()
        if countAP3 == 0:
            print(f"AP {AP3} Tidak Ditemukan")
            exit()
    if countAP1 == 1 and countAP2 == 1 and countAP3 ==1:
        with lock: # Menggunakan lock untuk mengamankan akses ke file CSV
            with open(outputKalmanFilter, 'a') as csv_file:
                csv_writer = csv.DictWriter(csv_file,
                fieldnames=target_ssids)
                info = {
                    "RuijieAP1": distanceKalmanAP1,
                    "RuijieAP2": distanceKalmanAP2,
                    "RuijieAP3": distanceKalmanAP3
                }
                csv_writer.writerow(info)
                print(f"AP1:{distanceKalmanAP1},    AP2:{distanceKalmanAP2},
                AP3:{distanceKalmanAP3}")
            with lock: # Menggunakan lock untuk mengamankan akses ke file CSV
                with open(outputJarak, 'a') as csv_file:
                    csv_writer = csv.DictWriter(csv_file,
                    fieldnames=target_ssids)
                    info = {
                        "RuijieAP1": distanceAP1,
                        "RuijieAP2": distanceAP2,
                        "RuijieAP3": distanceAP3
                    }
                    csv_writer.writerow(info)
                ambilDataKF(ssid,lock,outputTrilaterasiKF,kolom)
                ambilData(ssid,lock,outputTrilaterasi,kolom)
        # if countAP1 == 1 and countAP2 == 1 and countAP3 == 1 and data_count >=
max_data:
        #     print(f"Pengambilan Data Selesai")
        #     exit()
    except KeyboardInterrupt:
        print("Dihentikan oleh pengguna (Ctrl+C)")

def scan_N(ssid, outputRSSI, outputN, max_data, stop_event, jarak):
    wifi = pywifi.PyWiFi()
    iface = wifi.interfaces()[0] # Menggunakan antarmuka pertama (biasanya wlan0)

    try:

```

```

# Inisialisasi hitungan data yang diambil
data_count = 0

while data_count < max_data and not stop_event.is_set():
    iface.scan()
    time.sleep(0.5)
    scan_results = iface.scan_results()

    for result in scan_results:
        if ssid in result.ssid:
            # Fungsi untuk mendapatkan nilai RSSI
            rssi = result.signal
            with open(outputRSSI, "a") as file:
                file.write(f"RSSI {ssid} = {rssi} \n")
            print(f"{data_count+1}.RSSI {ssid} = {rssi} dBm")

            # Fungsi Calculate Path Loss Exponent
            nValue = calculate_path_loss_exponent(rssi,ssid,jarak)
            with open(outputN, "a") as file:
                file.write(f"N-{ssid}-{jarak}m = {nValue}\n")
            print(f"N-{ssid}-{jarak}m = {nValue}")

            data_count += 1 # Menambah jumlah data yang diambil

            if data_count >= max_data:
                print(f"Pengambilan Data {ssid} Selesai")
                break
        else:
            print(f"Tidak dapat menemukan {ssid}")
            stop_event.set() # Set stop_event jika SSID tidak ditemukan
            break

except KeyboardInterrupt:
    print("Dihentikan oleh pengguna (Ctrl+C)")

def scan_rssi(ssid, outputRSSI, max_data, stop_event):
    wifi = pywifi.PyWiFi()
    iface = wifi.interfaces()[0] # Menggunakan antarmuka pertama (biasanya wlan0)

    try:
        # Inisialisasi hitungan data yang diambil
        data_count = 0

        while data_count < max_data and not stop_event.is_set():
            iface.scan()
            time.sleep(0.5)
            scan_results = iface.scan_results()

            for result in scan_results:
                if ssid in result.ssid:
                    # Fungsi untuk mendapatkan nilai RSSI
                    rssi = result.signal
                    with open(outputRSSI, "a") as file:
                        file.write(f"RSSI {ssid} = {rssi} \n")
                    print(f"{data_count+1}.RSSI {ssid} = {rssi} dBm")

                    # Menambah jumlah data yang diambil
                    data_count += 1

                    if data_count >= max_data:
                        print(f"Pengambilan Data {ssid} Selesai")
                        break
                else:
                    print(f"Tidak dapat menemukan {ssid}")
                    # Set stop_event jika SSID tidak ditemukan
                    stop_event.set()
                    break
    
```

```

except KeyboardInterrupt:
    print("Dihentikan oleh pengguna (Ctrl+C)")

def scan_distance(ssid, outputRSSI, outputJarak, max_data, stop_event):
    wifi = pywifi.PyWiFi()
    iface = wifi.interfaces()[0] # Menggunakan antarmuka pertama (biasanya wlan0)

    try:
        # Inisialisasi hitungan data yang diambil
        data_count = 0

        while data_count < max_data and not stop_event.is_set():
            iface.scan()
            time.sleep(0.5)
            scan_results = iface.scan_results()

            for result in scan_results:
                if ssid in result.ssid:
                    # Fungsi untuk mendapatkan nilai RSSI
                    rssi = result.signal
                    with open(outputRSSI, "a") as file:
                        file.write(f"RSSI {ssid} = {rssi} \n")
                    print(f"{data_count+1}.RSSI {ssid} = {rssi} dBm")

                    # Fungsi untuk mendapatkan nilai jarak TANPA Kalman Filter
                    distance = calculate_distance(rssi, ssid, ssid)
                    with open(outputJarak, "a") as file:
                        file.write(f" Jarak {ssid} = {distance} \n")
                    print(f"Jarak {ssid} = {distance} meter")

                    # Menambah jumlah data yang diambil
                    data_count += 1

                    if data_count >= max_data:
                        print(f"Pengambilan Data {ssid} Selesai")
                        break
                else:
                    print(f"Tidak dapat menemukan {ssid}")
                    # Set stop_event jika SSID tidak ditemukan
                    stop_event.set()
                    break

    except KeyboardInterrupt:
        print("Dihentikan oleh pengguna (Ctrl+C)")

if __name__ == "__main__":
    # Ganti dengan daftar SSID yang ingin Anda lacak
    target_ssids = ["RuijieAP1", "RuijieAP2", "RuijieAP3"]
    # Event untuk menghentikan proses jika SSID tidak ditemukan
    stop_event = Event()
    manager = Manager()
    # Dictionary untuk menyimpan array berdasarkan SSID
    array_dict = {}
    processes = []
    # Inisialisasi objek Lock
    lock = Lock()

    print("Daftar Pilihan:")
    print("1. Mencari Titik Lokasi (+Kalman Filter)")
    print("2. Mencari Nilai N (Path Loss Exponent)")
    print("3. Mencari Nilai RSSI")
    print("4. Mencari Nilai Jarak ke 1 AP")

    pilihan = float(input("Masukkan Pilihan Anda: "))

    kolom = ["x", "y"]

```

```

if pilihan == 1:
    lock = Lock()

    outputKalmanFilter = f"outputKF.csv"
    with open(outputKalmanFilter, 'w') as csv_file:
        csv_writer = csv.DictWriter(csv_file, fieldnames=target_ssids)
        csv_writer.writeheader()

    outputTrilaterasiKF = f"outputTrilaterasiKF.csv"
    with open(outputTrilaterasiKF, 'w') as csv_file:
        csv_writer = csv.DictWriter(csv_file, fieldnames=kolom)
        csv_writer.writeheader()

    outputJarak = f"outputJarak.csv"
    with open(outputJarak, 'w') as csv_file:
        csv_writer = csv.DictWriter(csv_file, fieldnames=target_ssids)
        csv_writer.writeheader()

    outputTrilaterasi = f"outputTrilaterasi.csv"
    with open(outputTrilaterasi, 'w') as csv_file:
        csv_writer = csv.DictWriter(csv_file, fieldnames=kolom)
        csv_writer.writeheader()

    outputRSSI = f"outputRSSI.csv"

scan_location(target_ssids,outputRSSI,outputJarak,outputKalmanFilter,stop_event,lock,outputTrilaterasiKF,kolom,outputTrilaterasi)

elif pilihan == 2:
    max_data = 100
    jarak = float(input("Masukkan Jarak Lokasi ke AP: "))
    for ssid in target_ssids:
        outputRSSI = f"outputRSSI_{ssid}_{jarak}.m.txt"
        outputN = f"outputN_{ssid}_{jarak}.m.txt"
        process = Process(target=scan_N, args=(ssid, outputRSSI, outputN,
max_data, stop_event, jarak))
        processes.append(process)
        process.start()

    # Tunggu proses selesai
    for process in processes:
        process.join()

    # Cek apakah SSID tidak ditemukan
    if stop_event.is_set():
        print("Scanning dihentikan karena salah satu SSID tidak ditemukan.")

elif pilihan == 3:
    max_data = 200
    for ssid in target_ssids:
        outputRSSI = f"outputRSSI_{ssid}.txt"
        process = Process(target=scan_rssi, args=(ssid, outputRSSI, max_data,
stop_event))
        processes.append(process)
        process.start()

    # Tunggu proses selesai
    for process in processes:
        process.join()

    # Cek apakah SSID tidak ditemukan
    if stop_event.is_set():
        print("Scanning dihentikan karena salah satu SSID tidak ditemukan.")

elif pilihan == 4:

```

```
max_data = 100
for ssid in target_ssids:
    outputRSSI = f"outputRSSI_{ssid}.txt"
    outputJarak = f"outputJarak_{ssid}.txt"
    process = Process(target=scan_distance, args=(ssid, outputRSSI,
outputJarak, max_data, stop_event))
    processes.append(process)
    process.start()

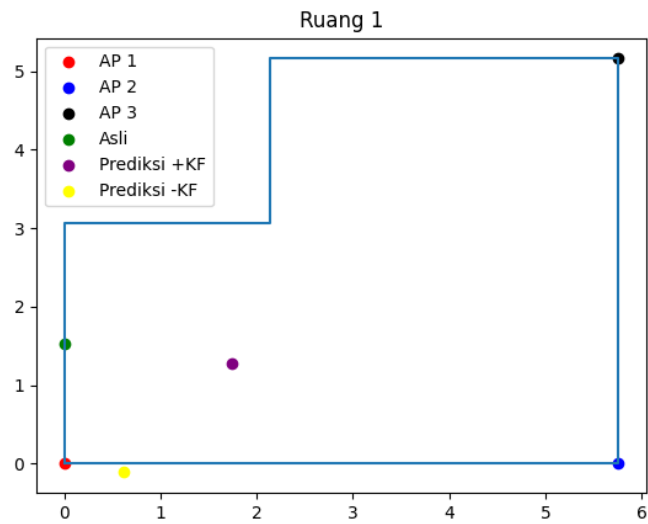
# Tunggu proses selesai
for process in processes:
    process.join()

# Cek apakah SSID tidak ditemukan
if stop_event.is_set():
    print("Scanning dihentikan karena salah satu SSID tidak ditemukan.")
else:
    exit()
```

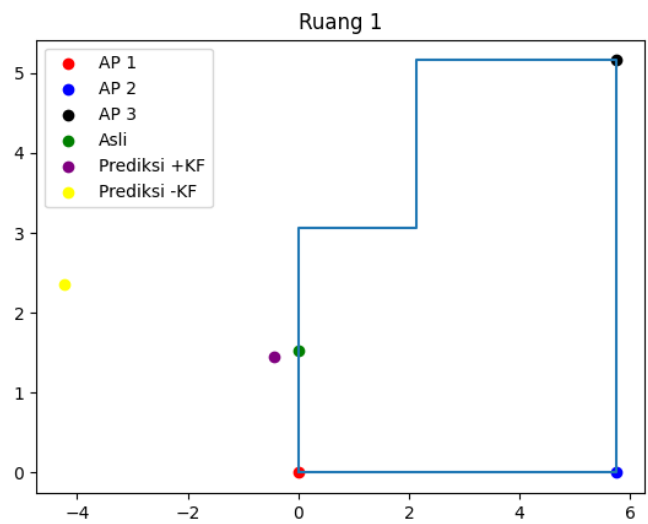
LAMPIRAN B

A. Visualisasi Data Statis

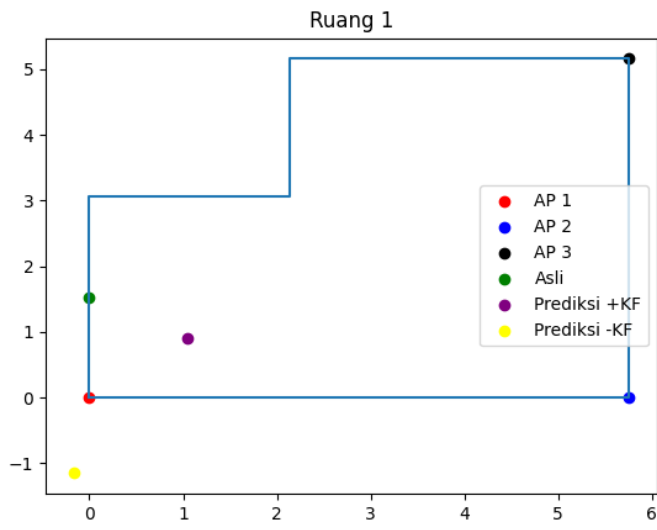
a. Titik 1, pengujian ke-1



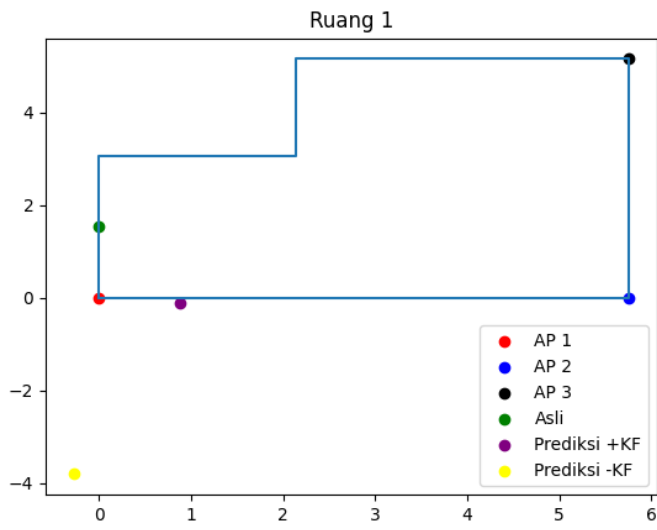
b. Titik 1, pengujian ke-2



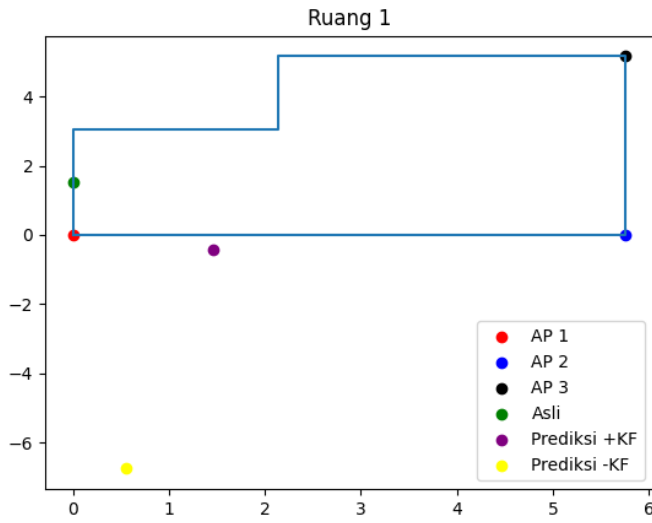
c. Titik 1, pengujian ke-3



d. Titik 1, pengujian ke-4



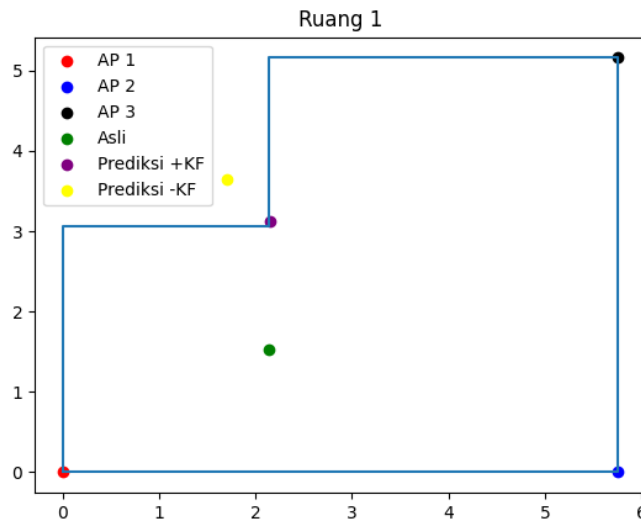
e. Titik 1, pengujian ke-5



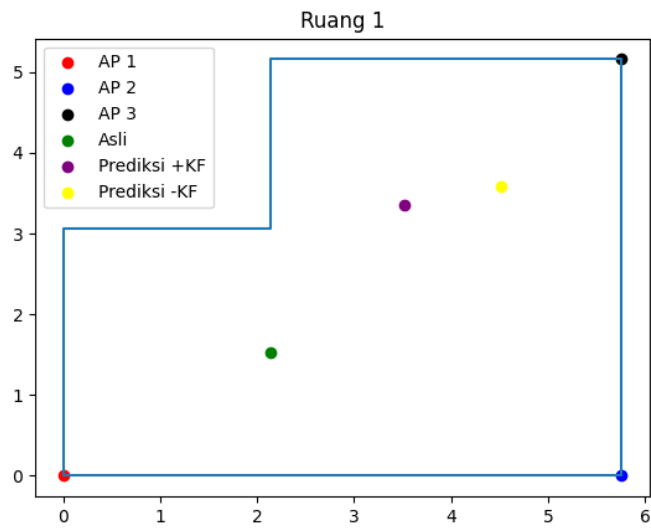
f. Tabel Data Pengujian Titik 1 Kondisi Statis

Pengujian ke-	Tanpa KF (x,y)	Error (m)	Dengan KF (x,y)	Error (m)
1	0.62 , -0.11	1.75	1.74, 1.27	1.76
2	-4.24 , 2.35	4.32	-0.43, 1.45	0.44
3	-0.17 , -1.15	2.69	1.05, 0.91	1.22
4	-0.28 , -3.81	5.35	0.88, -0.11	1.86
5	0.55 , -6.75	8.3	1.46, -0.42	2.44
Rata-Rata		4.482		1.544

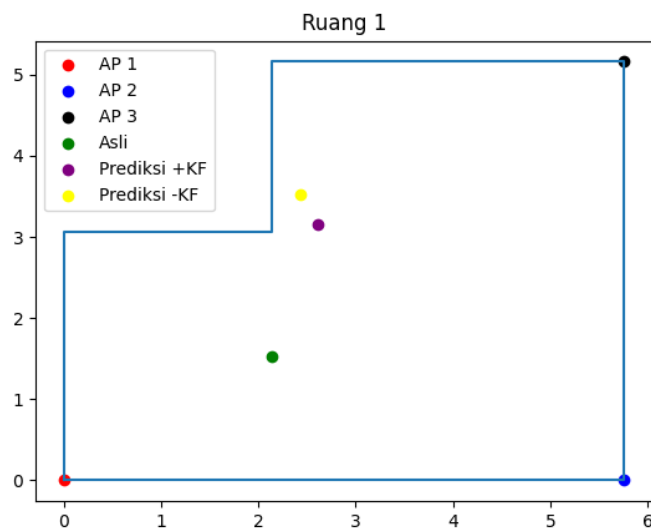
g. Titik 2, pengujian ke-1



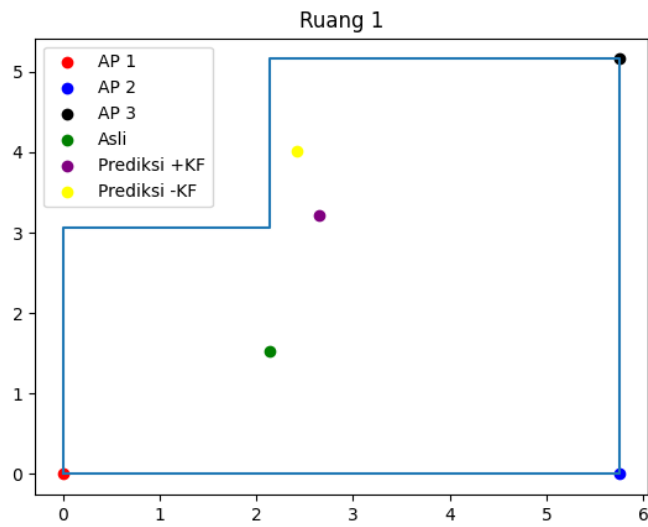
h. Titik 2, pengujian ke-2



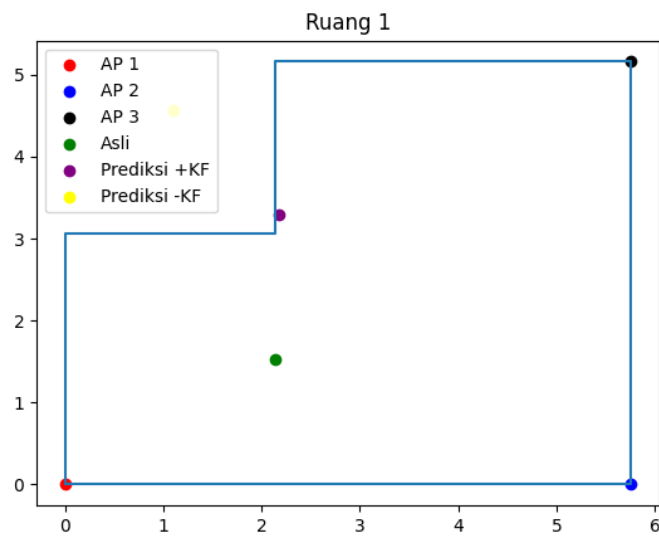
i. Titik 2, pengujian ke-3



j. Titik 2, pengujian ke-4



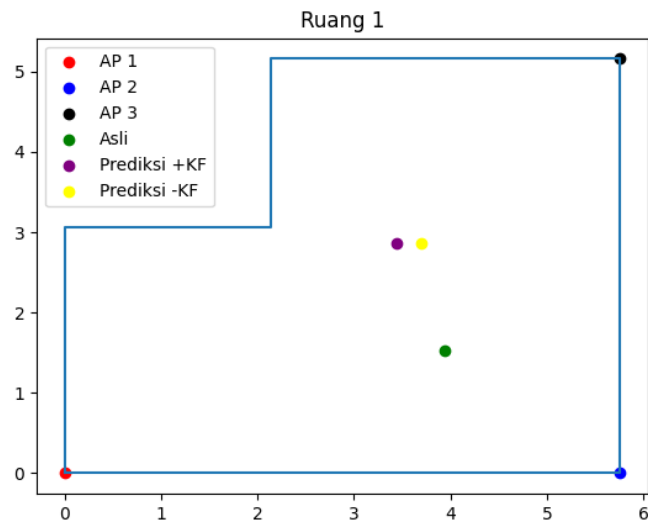
k. Titik 2, pengujian ke-5



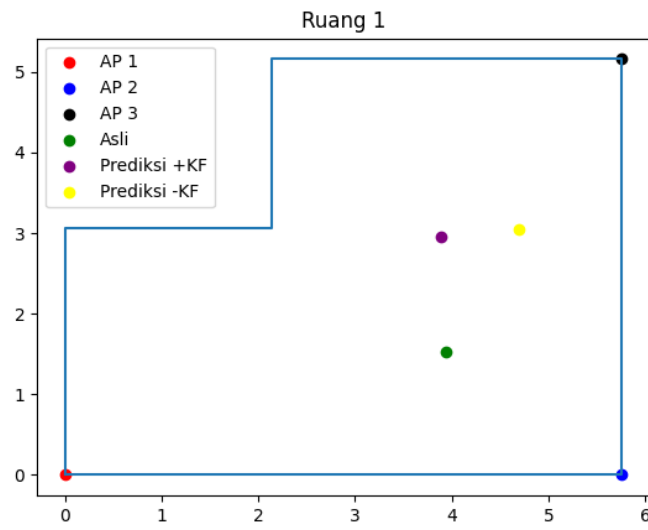
l. Tabel Data Pengujian Titik 2 Kondisi Statis

Pengujian ke-	Tanpa KF (x,y)	Error (m)	Dengan KF (x,y)	Error (m)
1	1.71, 3.65	2.16	2.15, 3.12	1.59
2	4.52, 3.58	3.14	3.52, 3.35	2.28
3	2.43, 3.52	2.01	2.61, 3.15	1.69
4	2.42, 4.01	2.5	2.65, 3.22	1.77
5	1.11, 4.57	3.21	2.18, 3.29	1.76
Rata-Rata		2.604		1.818

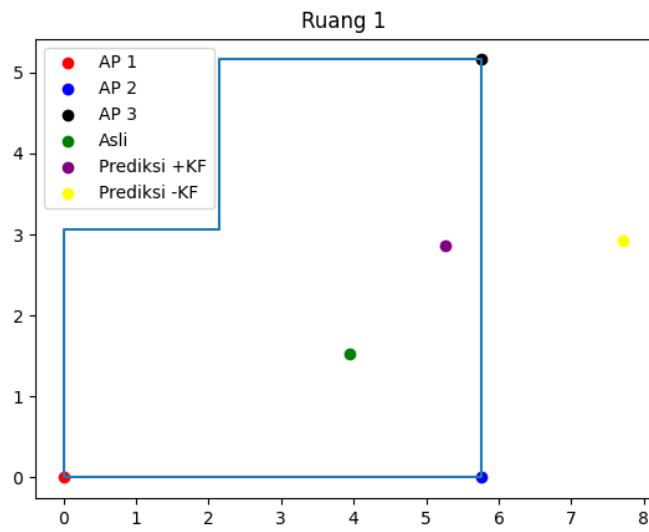
m. Titik 3, pengujian ke-1



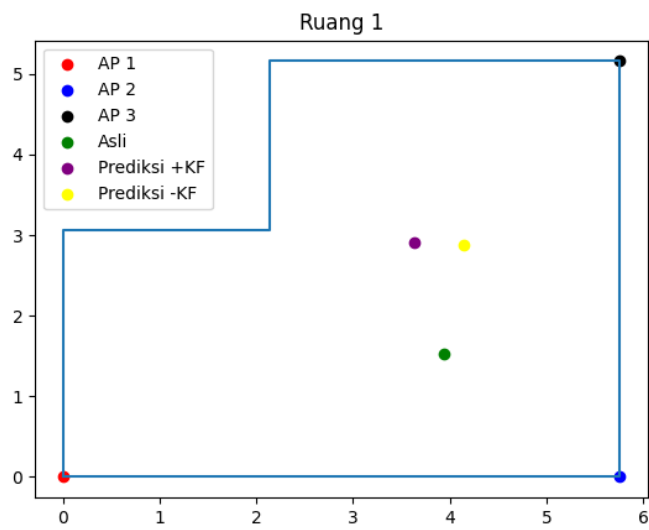
n. Titik 3, pengujian ke-2



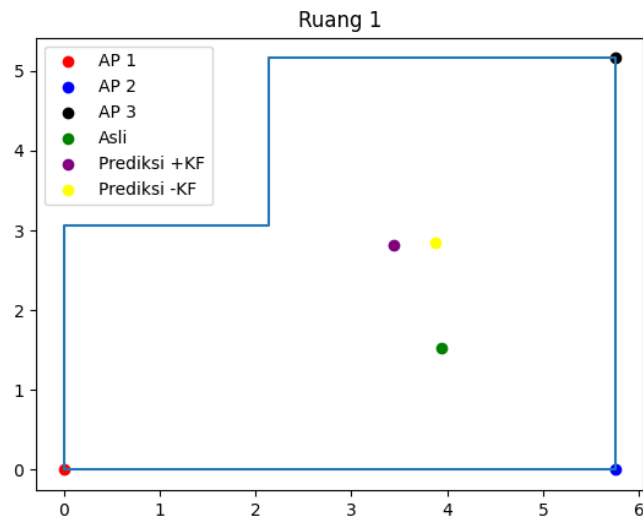
o. Titik 3, pengujian ke-3



p. Titik 3, pengujian ke-4



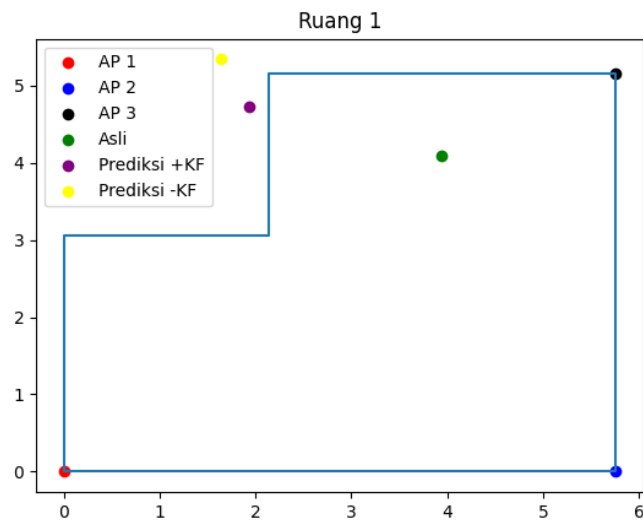
q. Titik 3, pengujian ke-5



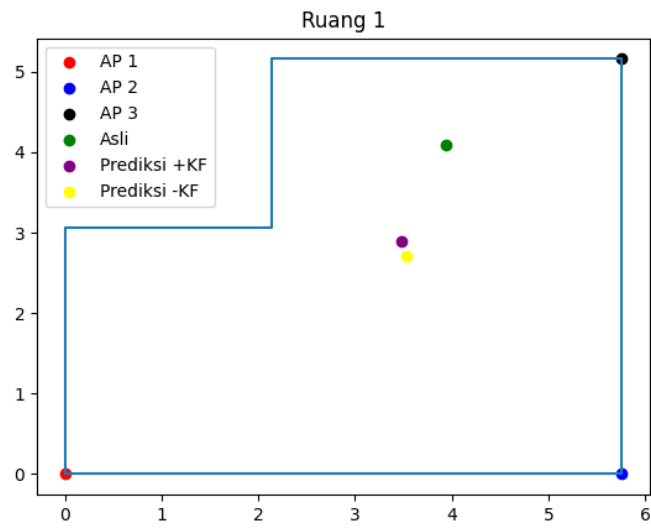
r. Tabel Data Pengujian Titik 3 Kondisi Statis

Pengujian ke-	Tanpa KF (x,y)	Eror (m)	Dengan KF (x,y)	Eror (m)
1	3.7, 2.86	1.35	3.44, 2.86	1.42
2	4.69, 3.05	1.69	3.89, 2.96	1.43
3	7.72, 2.93	4.03	5.26, 2.86	1.87
4	4.15, 2.88	1.37	3.64, 2.91	1.41
5	3.87, 2.84	1.31	3.44, 2.82	1.38
Rata-Rata		1.95		1.502

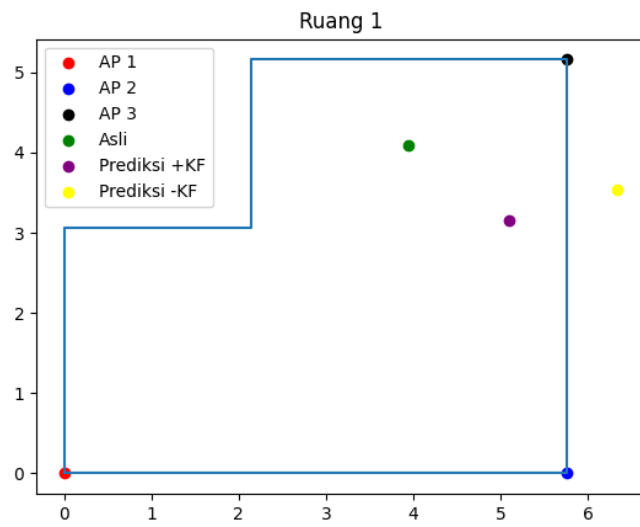
s. Titik 4, pengujian ke-1



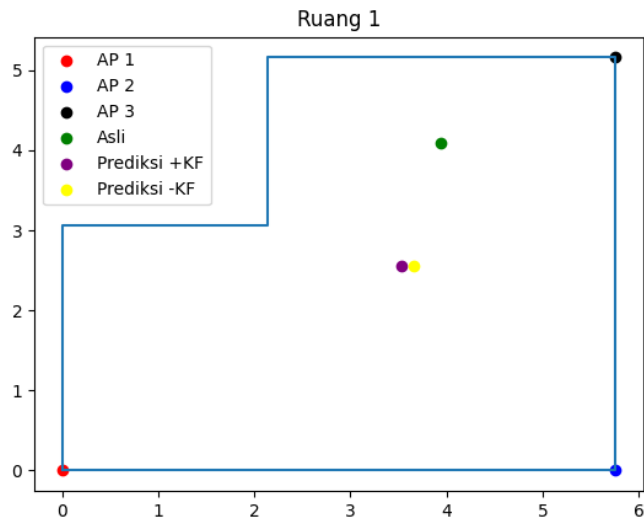
t. Titik 4, pengujian ke-2



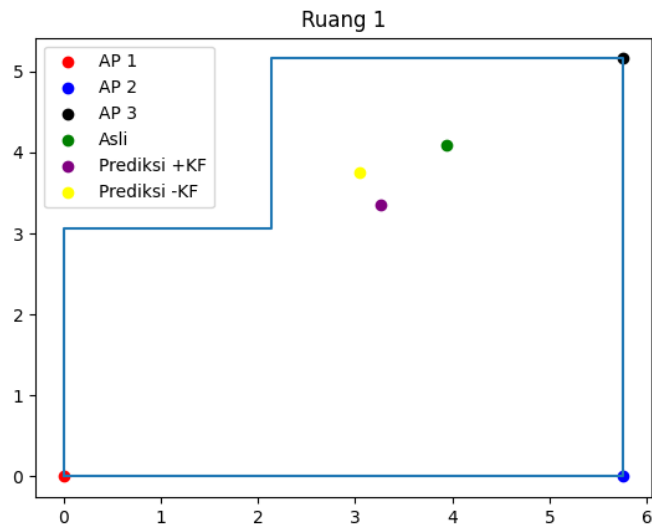
u. Titik 4, pengujian ke-3



v. Titik 4, pengujian ke-4



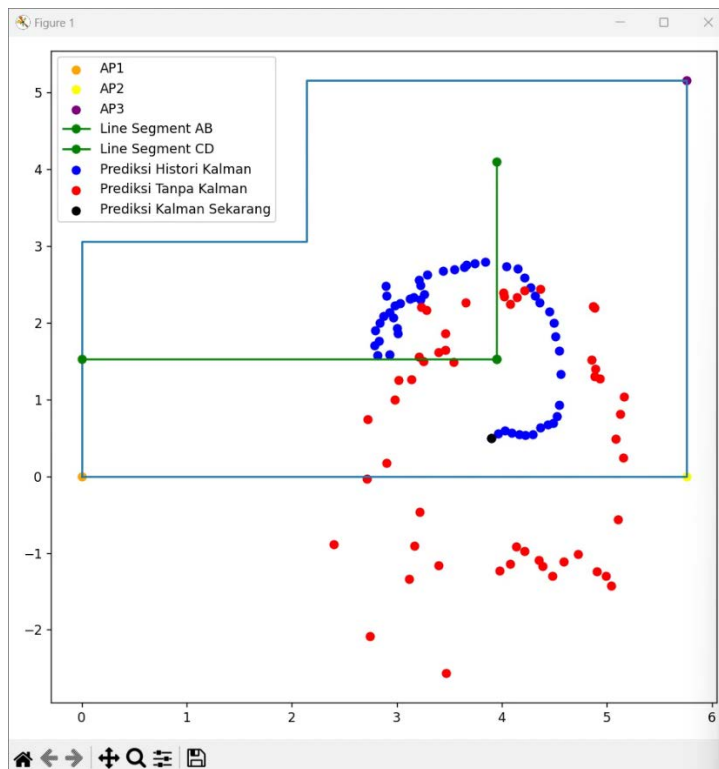
w. Titik 4, pengujian ke-5



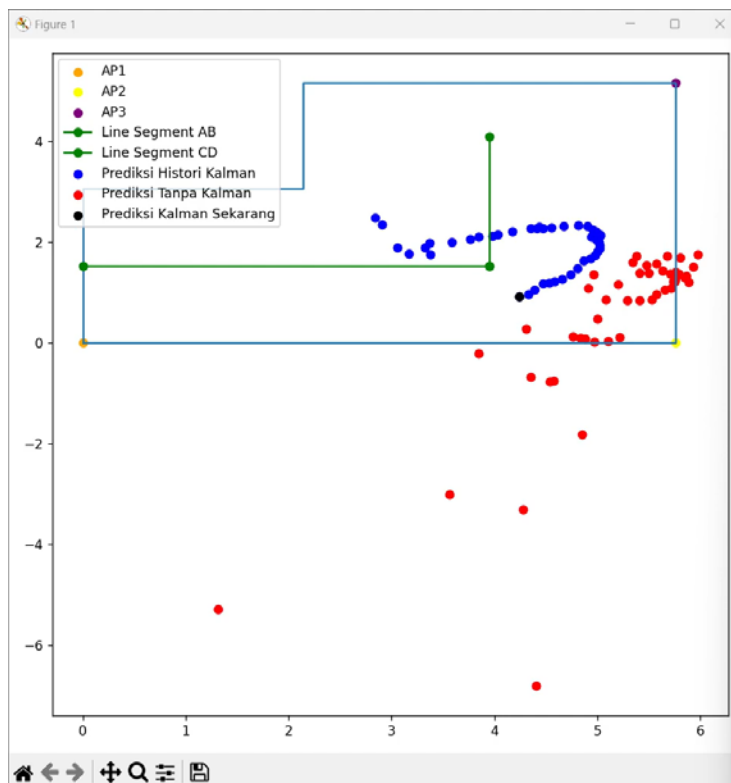
x. Tabel Data Pengujian Titik 4 Kondisi Statis

Pengujian ke-	Tanpa KF (x,y)	Error (m)	Dengan KF (x,y)	Error (m)
1	1.64, 5.34	2.62	1.94, 4.72	2.1
2	3.53, 2.71	1.44	3.48, 2.89	1.29
3	6.34, 3.54	2.46	5.09, 3.15	1.49
4	3.66, 2.56	1.56	3.53, 2.55	1.59
5	3.05, 3.75	0.95	3.26, 3.35	1
Rata-Rata		1.806		1.494

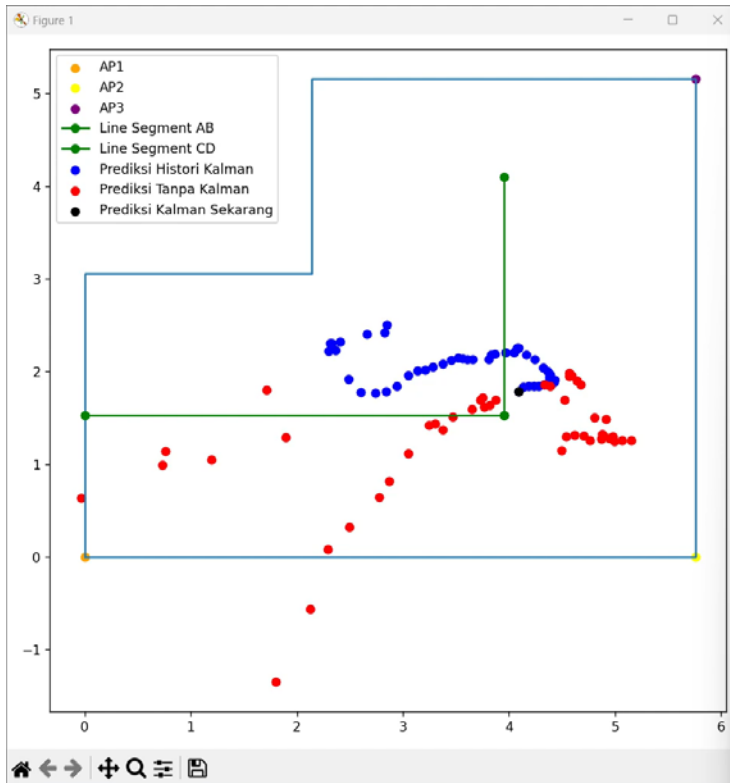
B. Visualisasi Data IPS Bergerak Ruang 1 Kondisi Kosong
a. Pengujian ke-1, kondisi AP 1



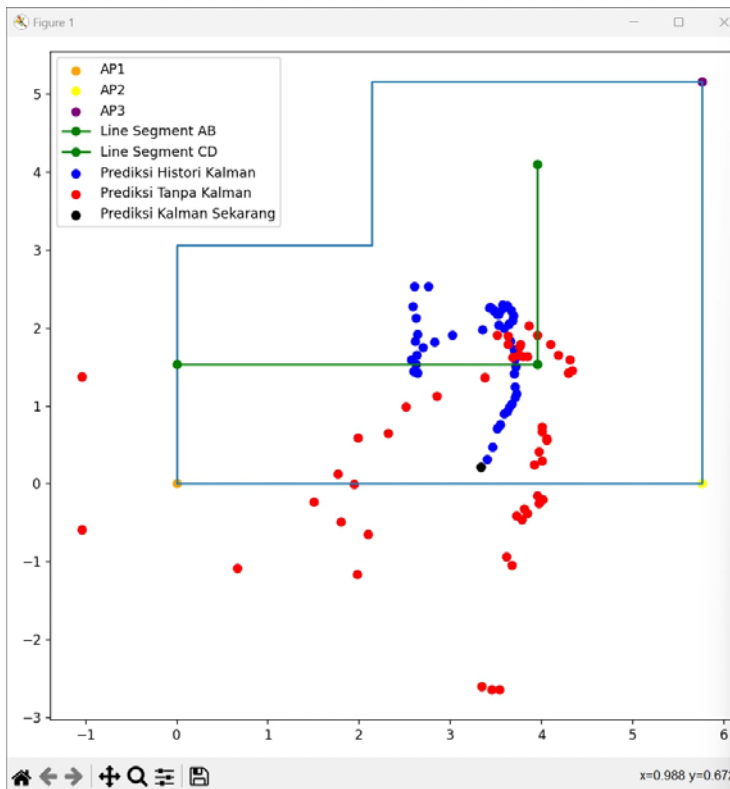
b. Pengujian ke-2, kondisi AP 1



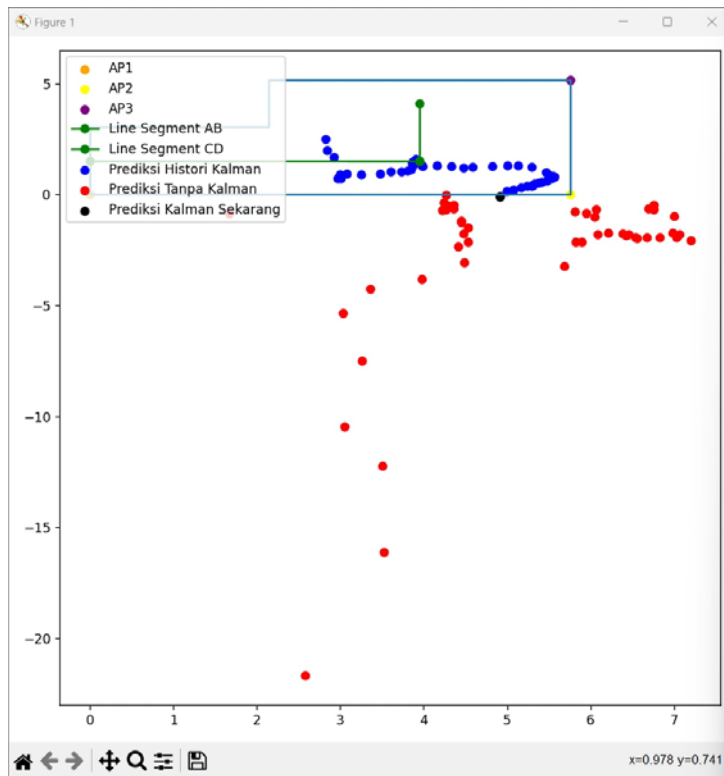
c. Pengujian ke-3, kondisi AP 1



d. Pengujian ke-4, kondisi AP 1



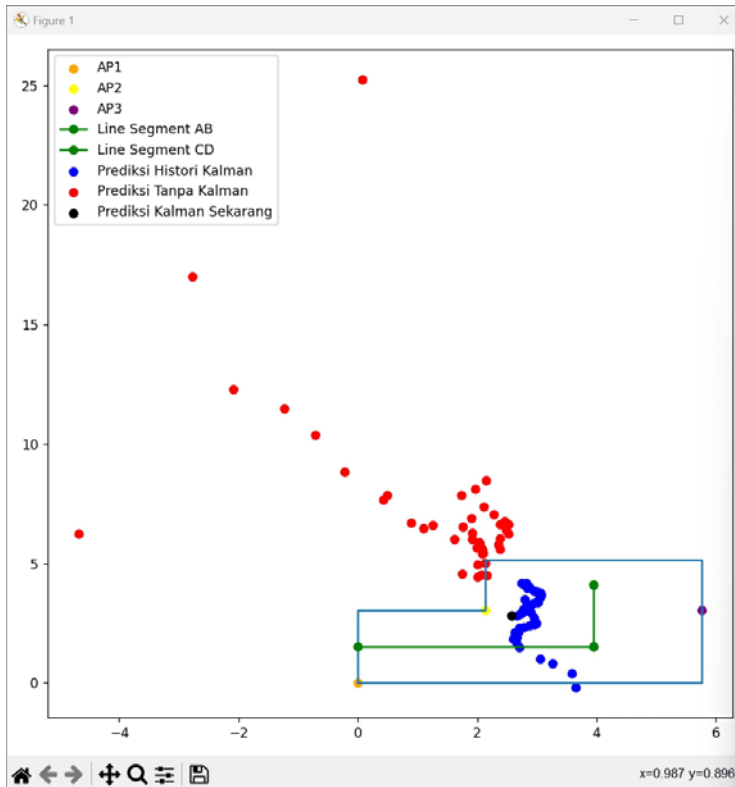
e. Pengujian ke-5, kondisi AP 1



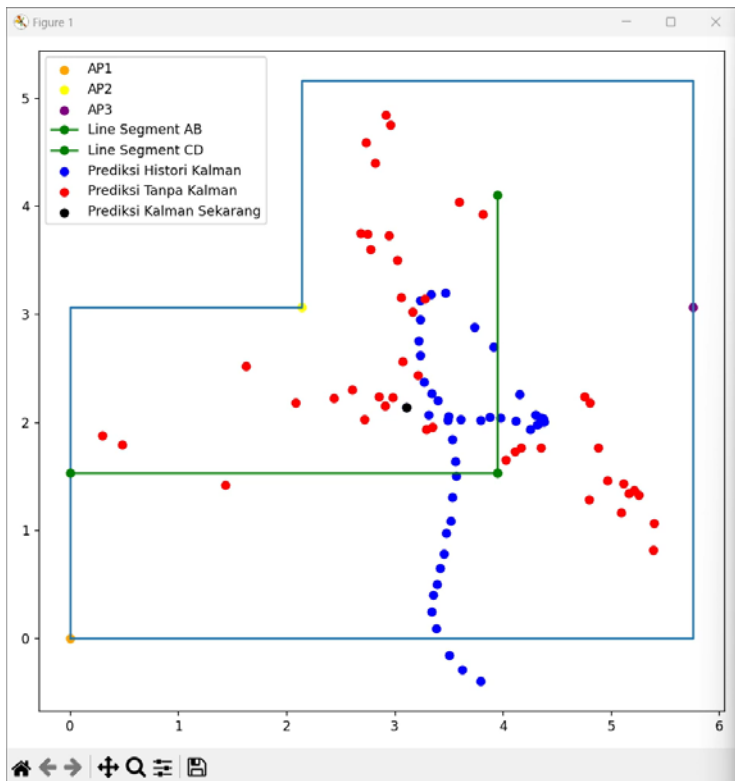
f. Tabel Data Pengujian Ruang 1, Kondisi Kosong, Posisi AP 1

Pengujian ke-	Error Tanpa KF (m)	Error Dengan KF (m)	Persentase Penurunan Error
1	1.48	0.59	60.14%
2	2.15	0.71	66.98%
3	0.68	0.39	42.65%
4	1.32	0.45	65.91%
5	4.97	0.95	80.89%
Rata-Rata	2.12	0.618	70.85%

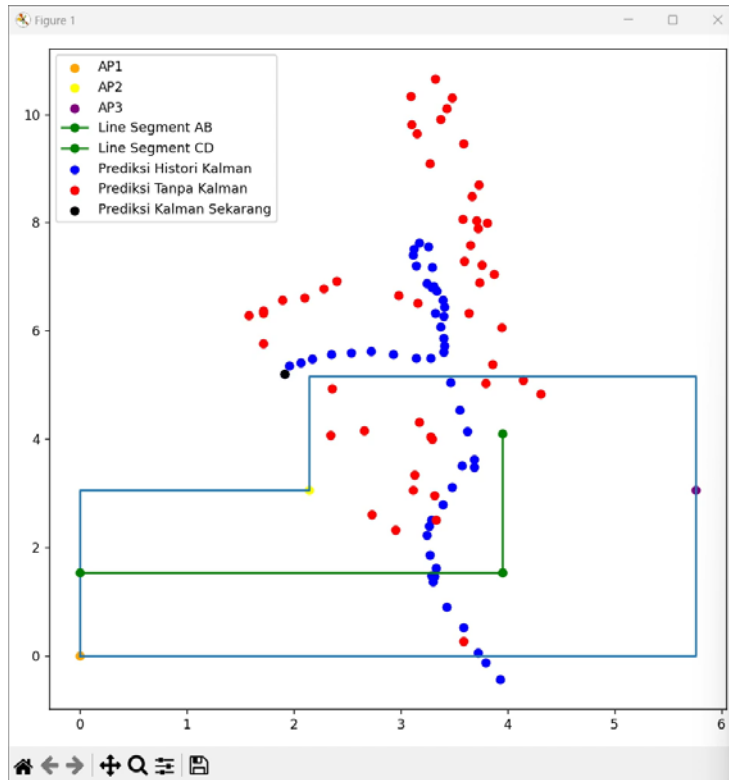
g. Pengujian ke-1, kondisi AP 2



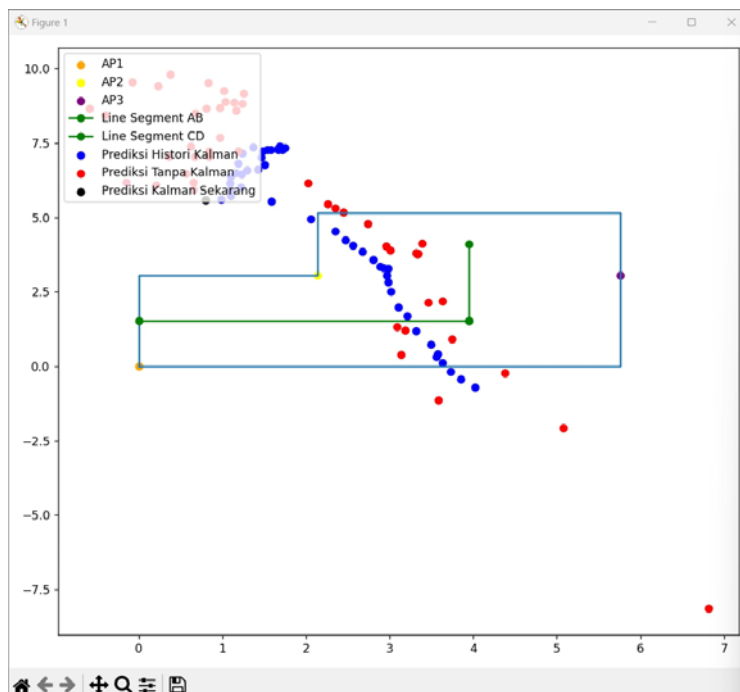
h. Pengujian ke-2, kondisi AP 2



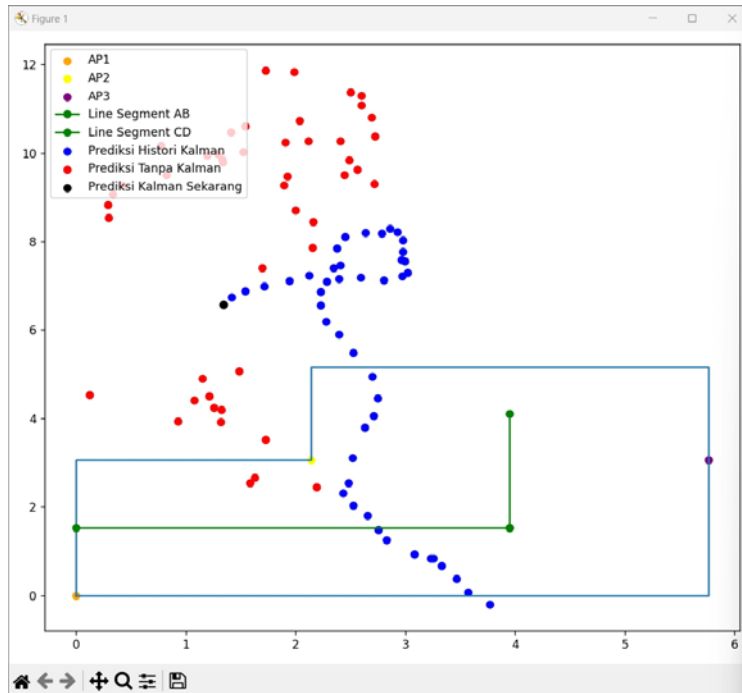
i. Pengujian ke-3, kondisi AP 2



j. Pengujian ke-4, kondisi AP 2



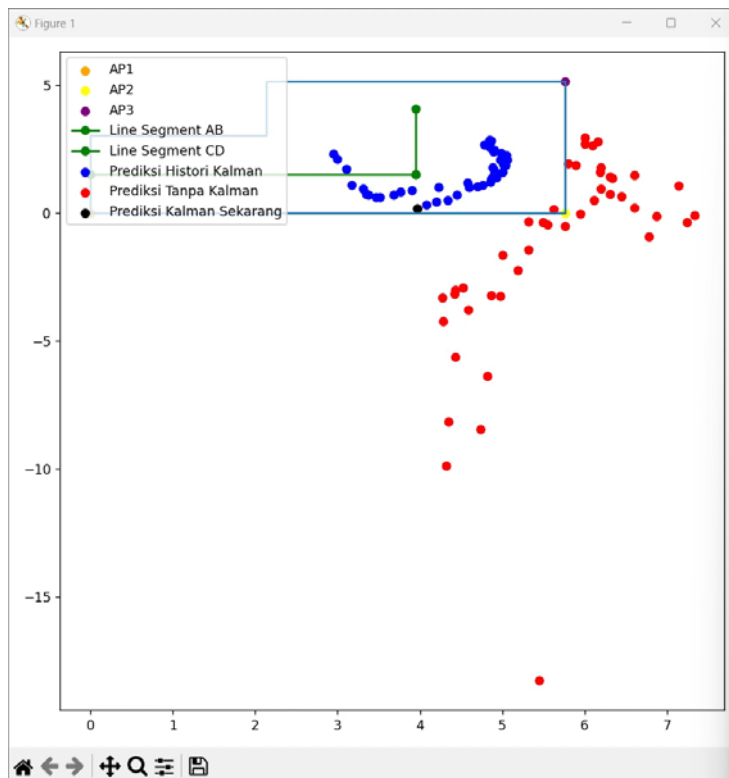
k. Pengujian ke-5, kondisi AP 2



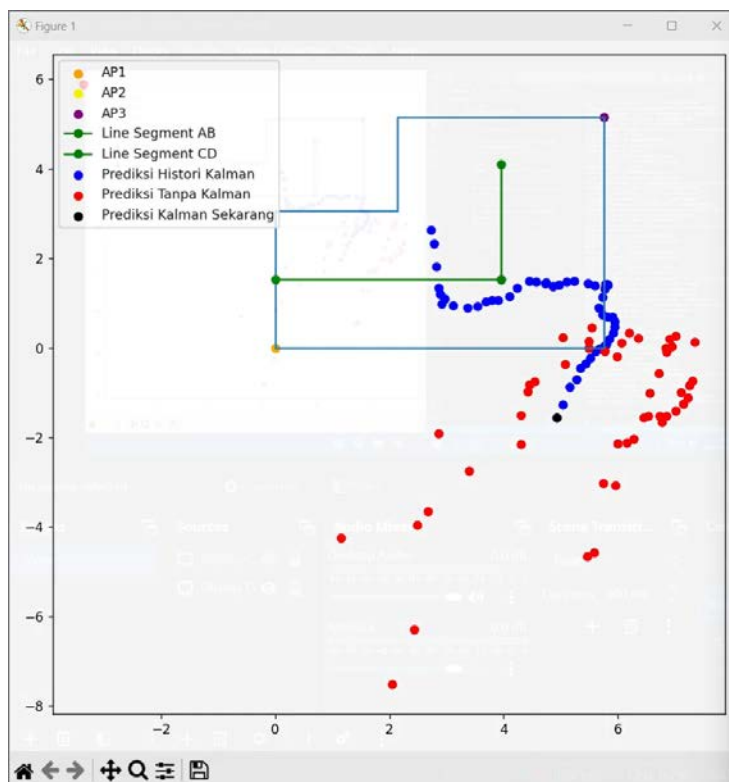
1. Tabel Data Pengujian Ruang 1, Kondisi Kosong, Posisi AP 2

Pengujian ke-	Error Tanpa KF (m)	Error Dengan KF (m)	Persentase Penurunan Error
1	4.18	0.93	77.75%
2	0.83	0.56	32.53%
3	3.01	1.68	44.19%
4	3.77	2.6	31.03%
5	5.09	2.56	49.71%
Rata-Rata	3.376	1.666	50.65%

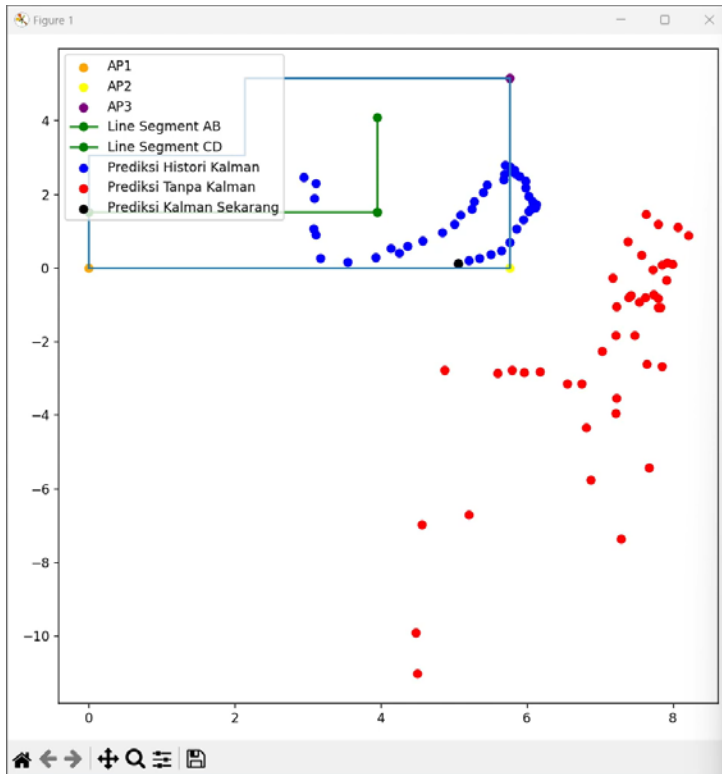
C. Visualisasi Data IPS Bergerak Ruang 1 Kondisi Isi
a. Pengujian ke-1, kondisi AP 1



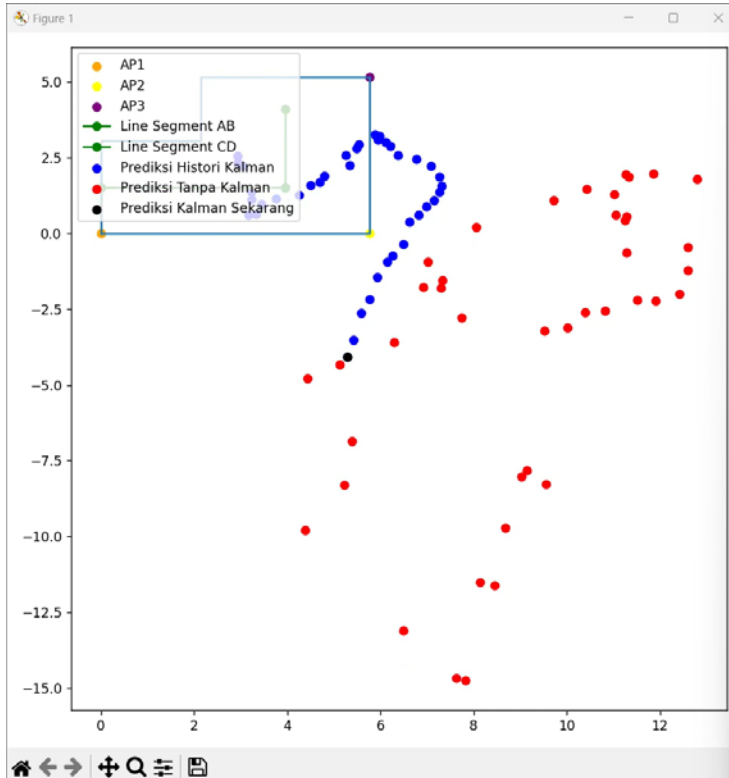
b. Pengujian ke-2, kondisi AP 1



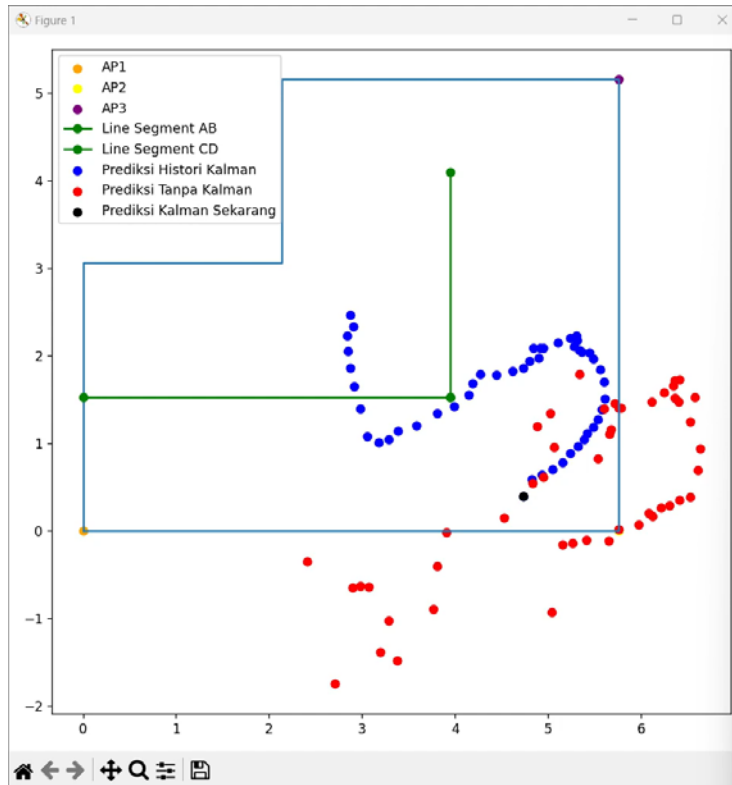
c. Pengujian ke-3, kondisi AP 1



d. Pengujian ke-4, kondisi AP 1



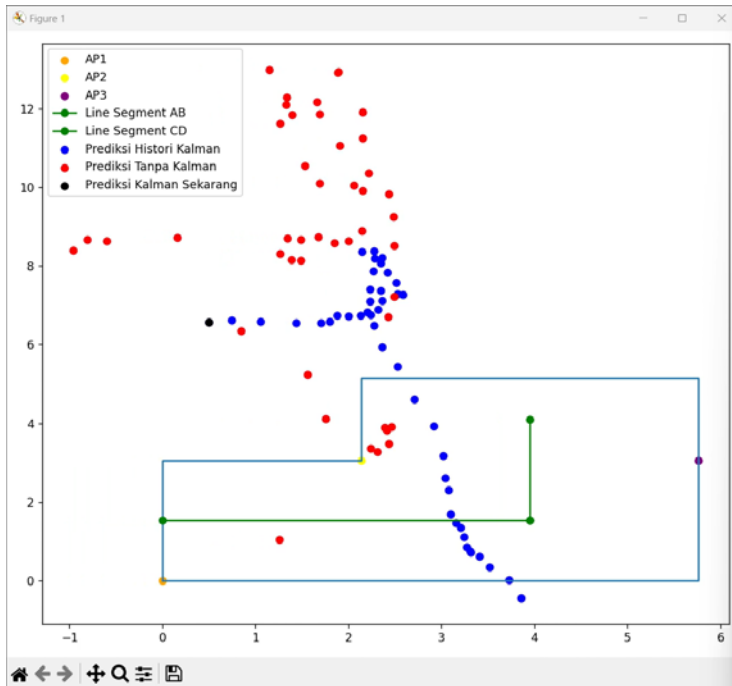
e. Pengujian ke-5, kondisi AP 1



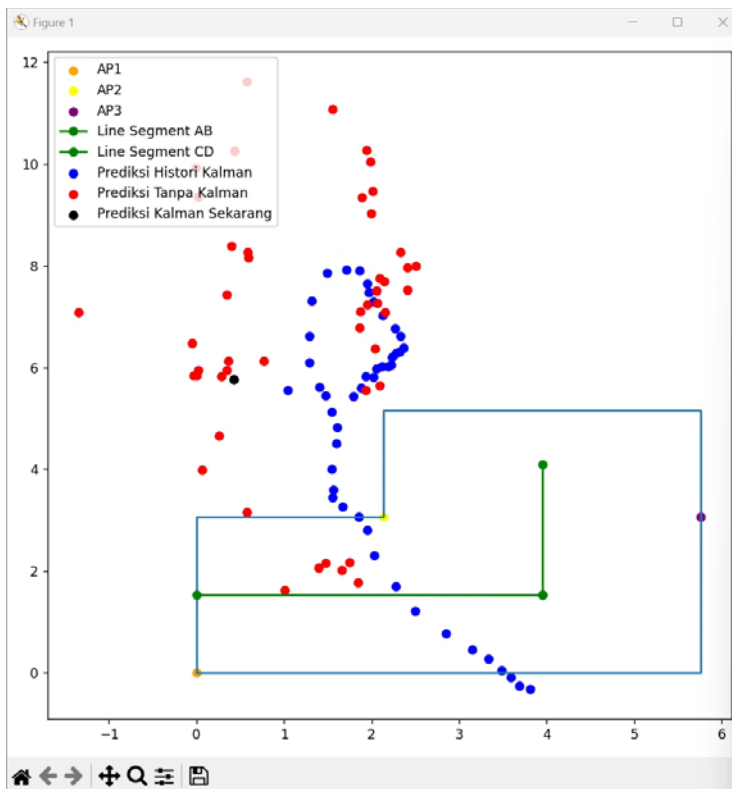
f. Tabel Data Pengujian Ruang 1, Kondisi Isi, Posisi AP 1

Pengujian ke-	Error Tanpa KF (m)	Error Dengan KF (m)	Persentase Penurunan Error
1	4.23	0.89	78.96%
2	3.85	1.49	61.30%
3	5.38	1.56	71.00%
4	8.71	2.24	74.28%
5	2.21	1	54.75%
Rata-Rata	4.876	1.436	70.55%

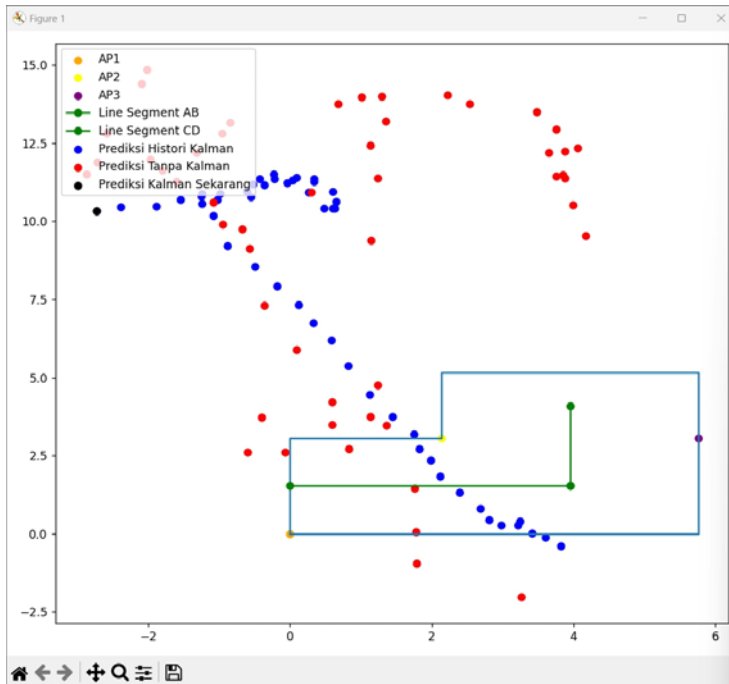
g. Pengujian ke-1, kondisi AP 2



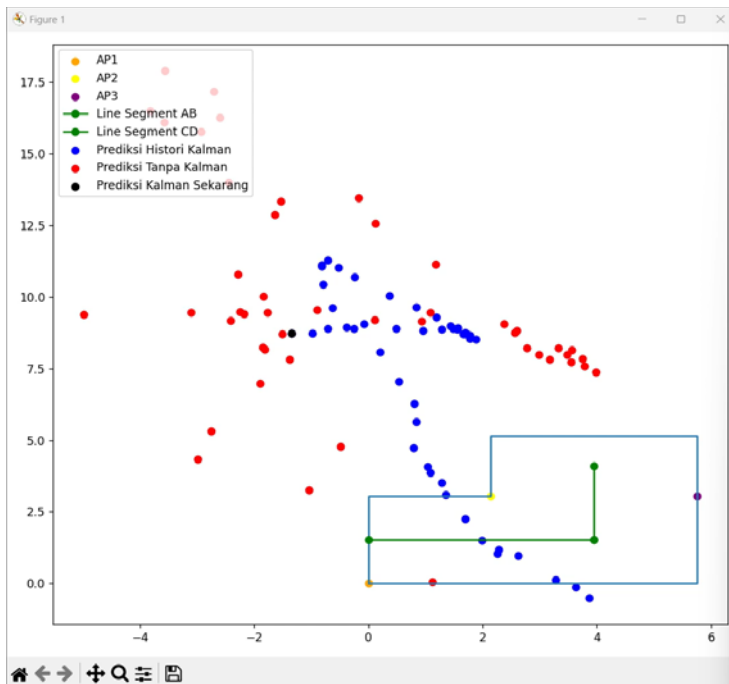
h. Pengujian ke-2, kondisi AP 2



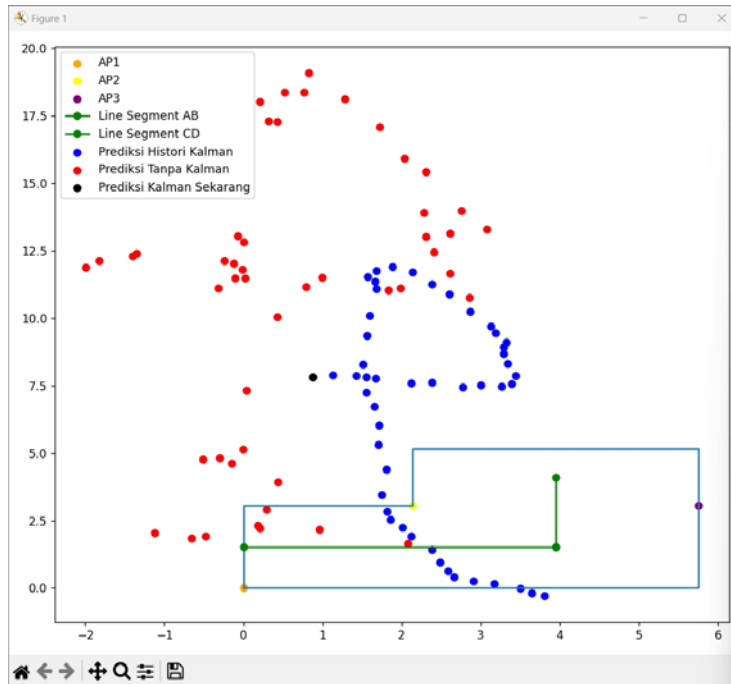
i. Pengujian ke-3, kondisi AP 2



j. Pengujian ke-4, kondisi AP 2



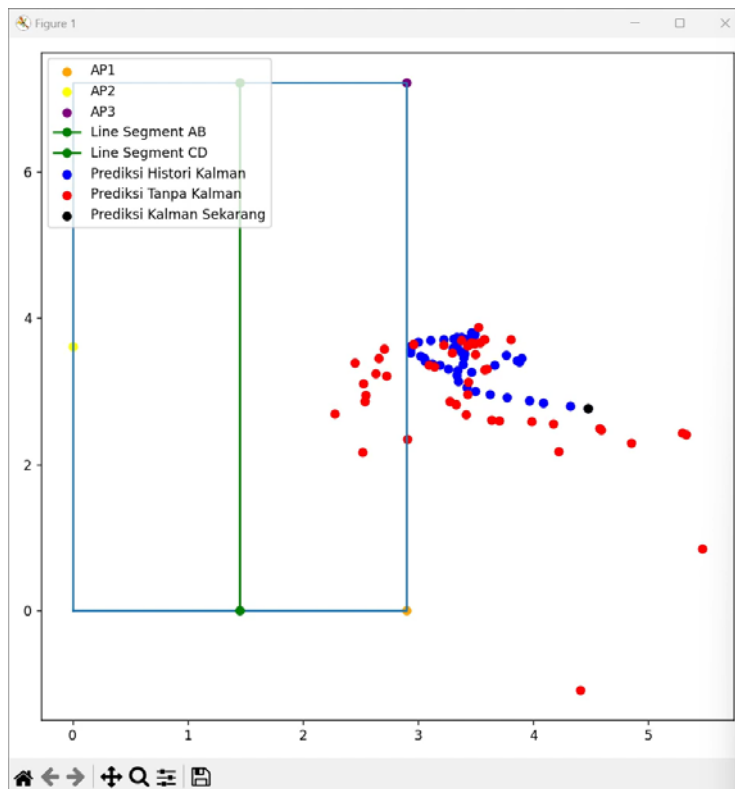
k. Pengujian ke-5, kondisi AP 2



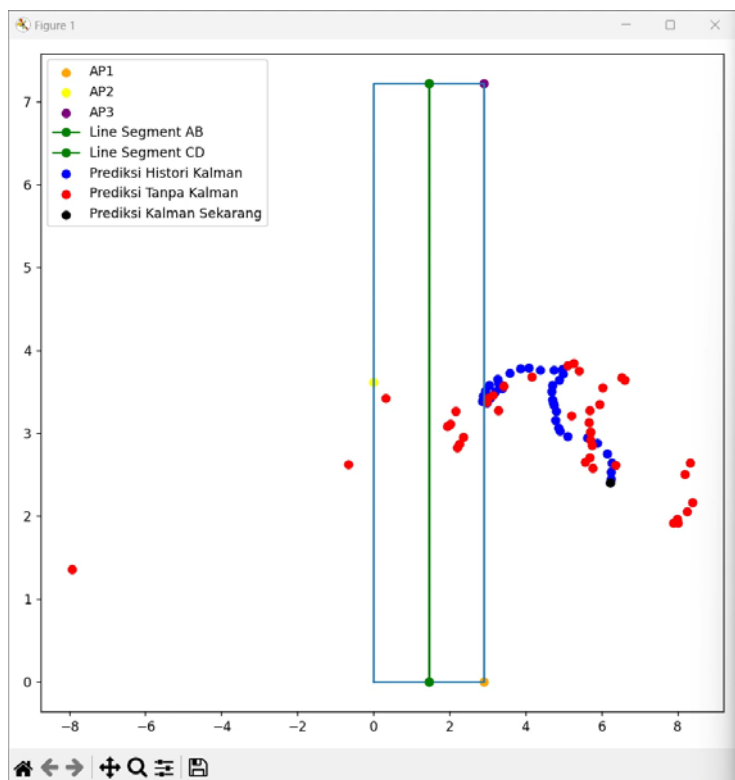
l. Tabel Data Pengujian Ruang 1, Kondisi Isi, Posisi AP 2

Pengujian ke-	Error Tanpa KF (m)	Error Dengan KF (m)	Persentase Penurunan Error
1	5.36	2.65	50.56%
2	4.07	2.53	37.84%
3	6.93	5.73	17.32%
4	7.26	4.86	33.06%
5	7.74	3.97	48.71%
Rata-Rata	6.272	3.948	37.05%

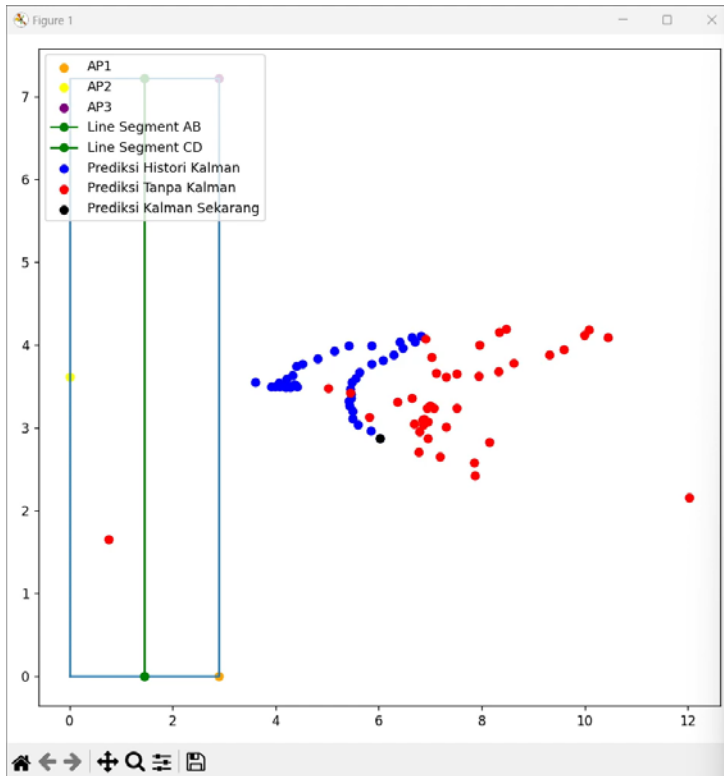
D. Visualisasi Data IPS Bergerak Ruang 2 Kondisi Kosong
a. Pengujian ke-1, kondisi AP 1



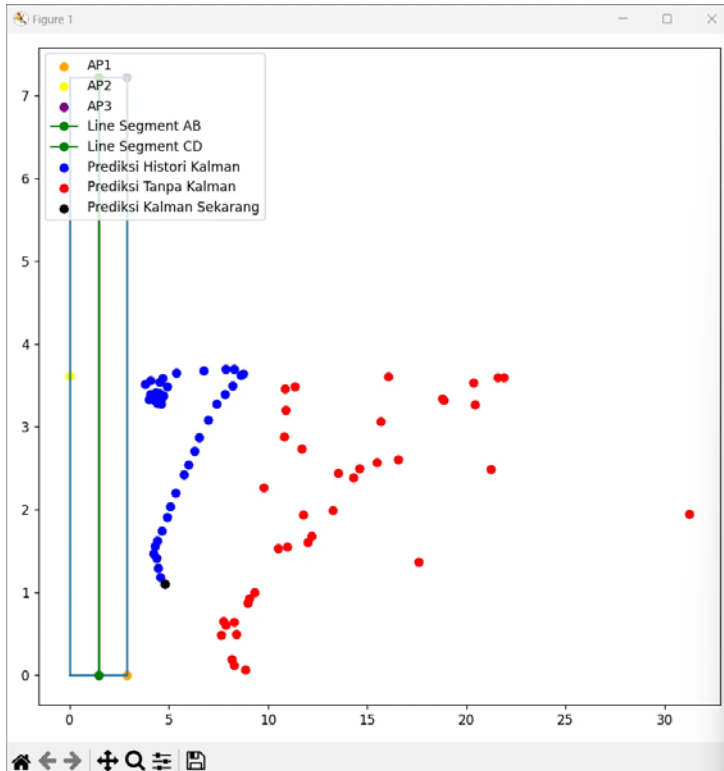
b. Pengujian ke-2, kondisi AP 1



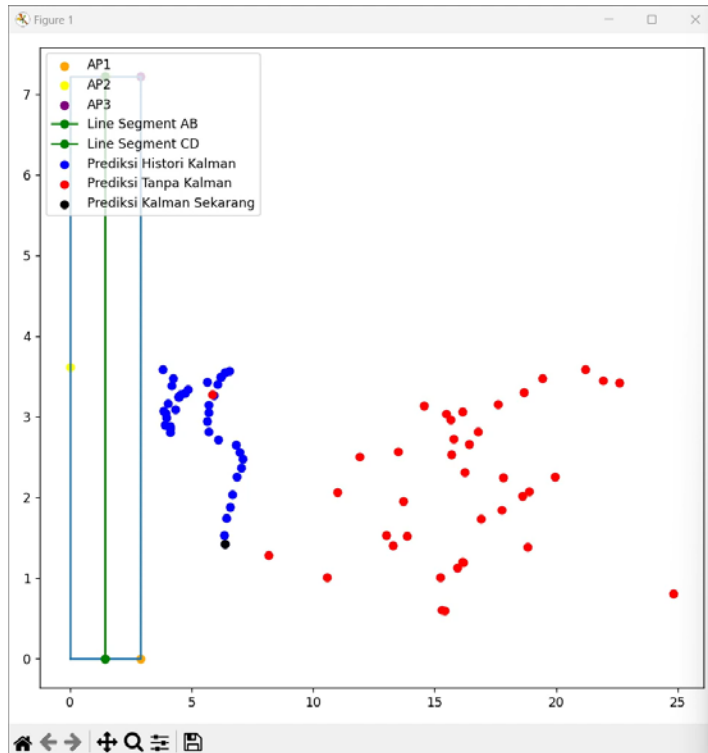
c. Pengujian ke-3, kondisi AP 1



d. Pengujian ke-4, kondisi AP 1



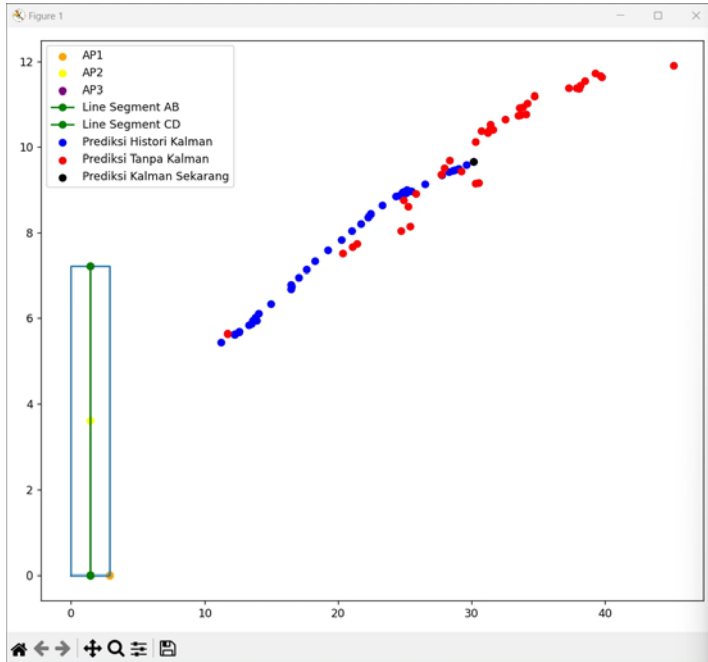
e. Pengujian ke-5, kondisi AP 1



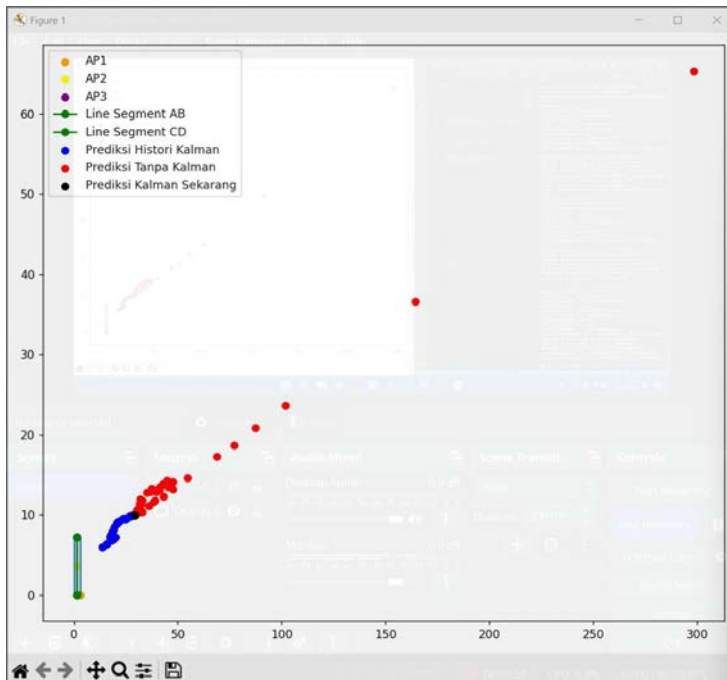
f. Tabel Data Pengujian Ruang 2, Kondisi Kosong, Posisi AP 1

Pengujian ke-	Error Tanpa KF (m)	Error Dengan KF (m)	Persentase Penurunan Error
1	2.08	2.01	3.37%
2	3.77	2.88	23.61%
3	6	3.75	37.50%
4	11.79	3.97	66.33%
5	14.55	3.96	72.78%
Rata-Rata	7.638	3.314	56.61%

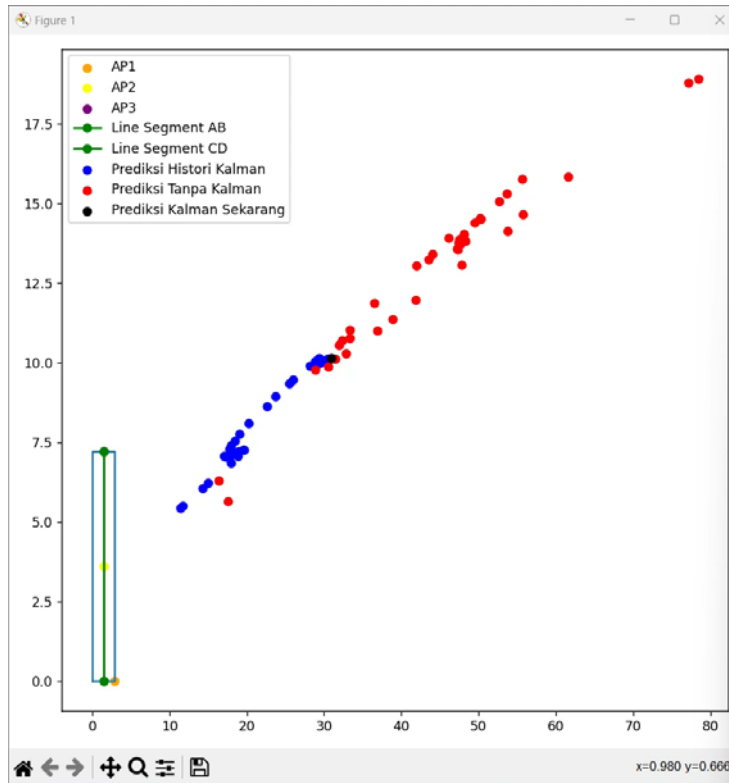
g. Pengujian ke-1, kondisi AP 2



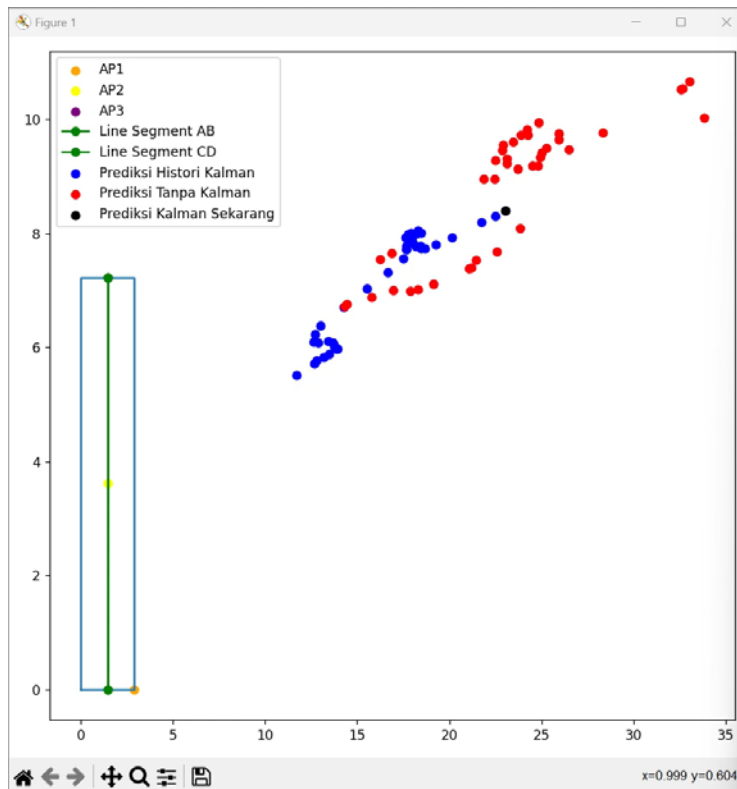
h. Pengujian ke-2, kondisi AP 2



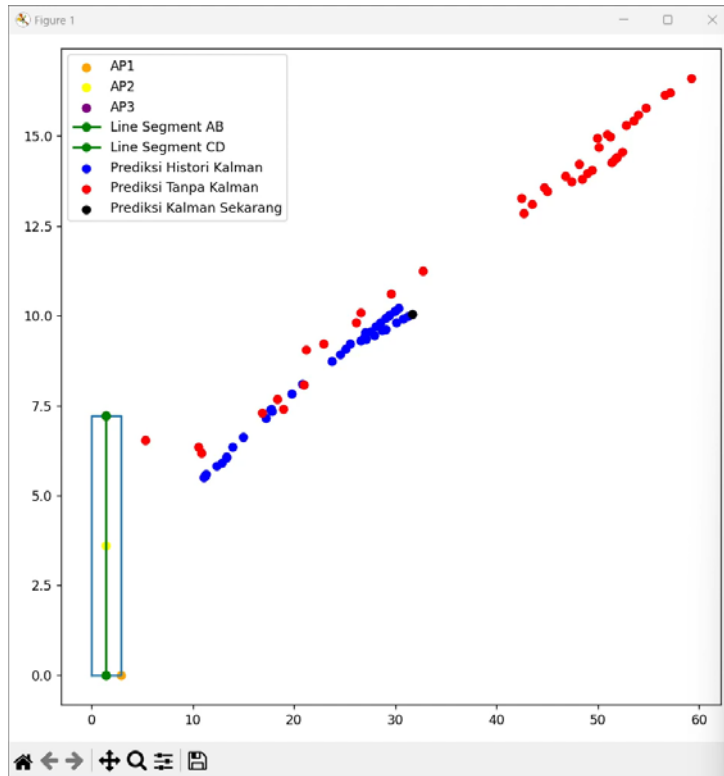
i. Pengujian ke-3, kondisi AP 2



j. Pengujian ke-4, kondisi AP 2



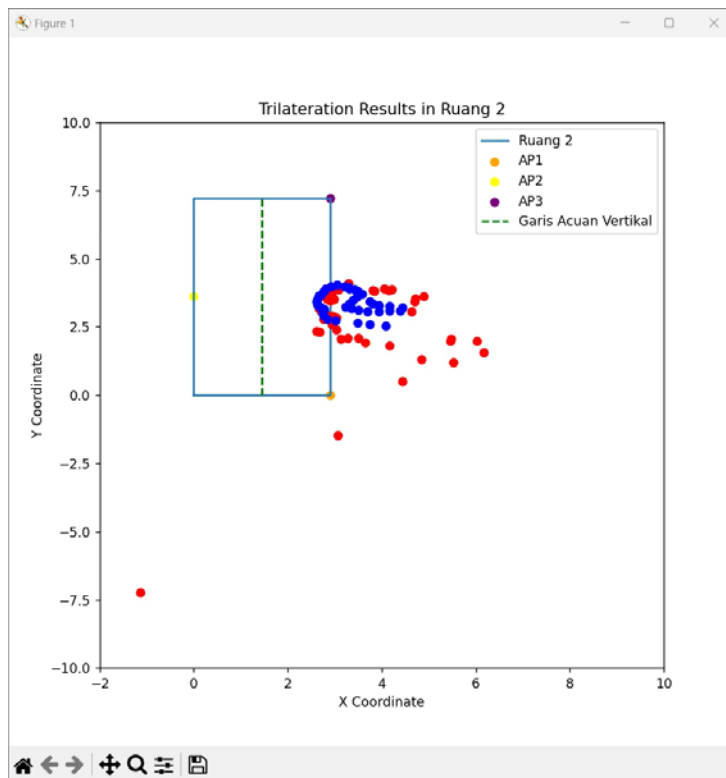
k. Pengujian ke-5, kondisi AP 2



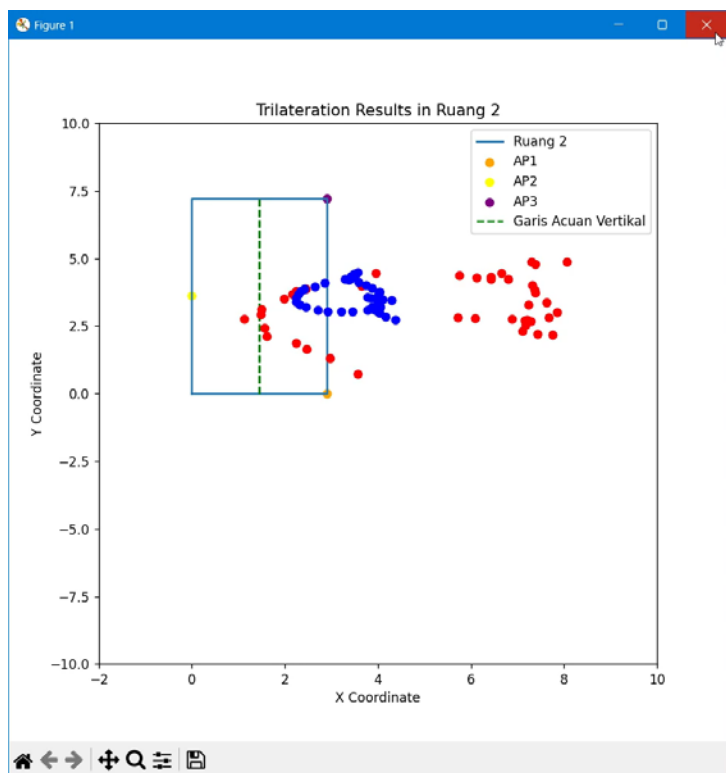
l. Tabel Data Pengujian Ruang 2, Kondisi Kosong, Posisi AP 2

Pengujian ke-	Error Tanpa KF (m)	Error Dengan KF (m)	Persentase Penurunan Error
1	30.34	19.79	34.77%
2	54.02	19.87	63.22%
3	43.63	21.55	50.61%
4	21.93	15.03	31.46%
5	39.42	21.9	44.44%
Rata-Rata	37.8868	19.628	48.17%

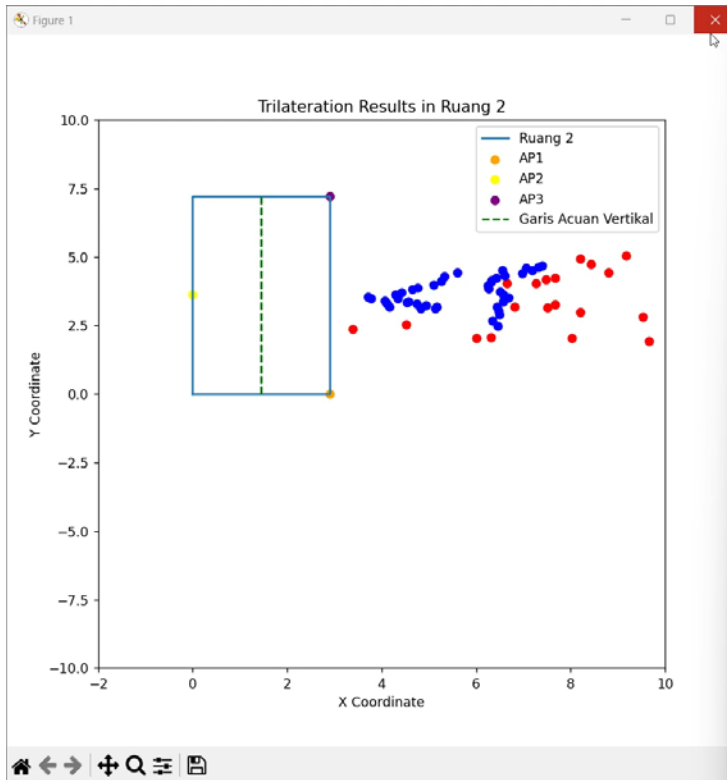
E. Visualisasi Data IPS Bergerak Ruang 2 Kondisi Isi
a. Pengujian ke-1, kondisi AP 1



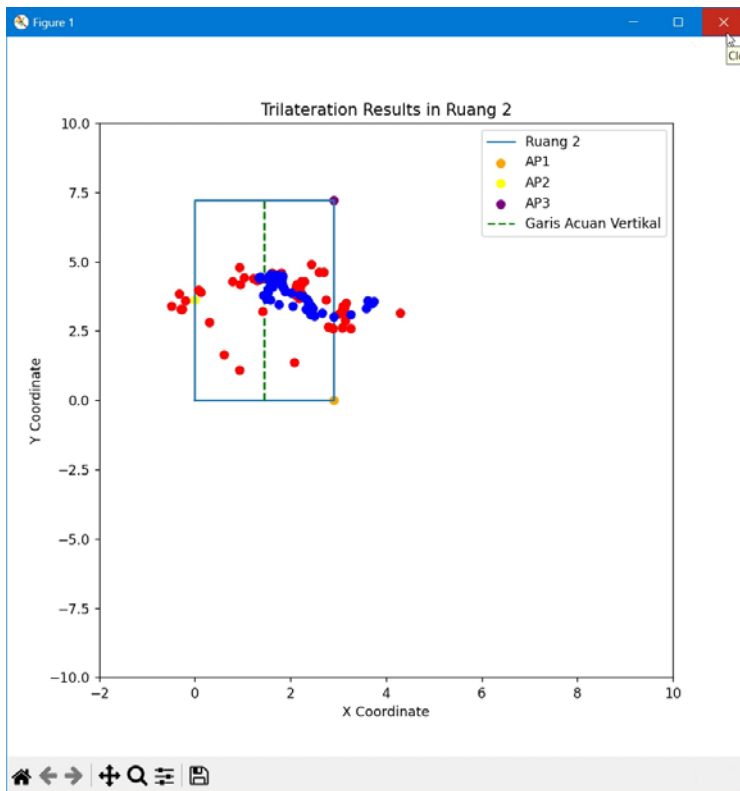
b. Pengujian ke-2, kondisi AP 1



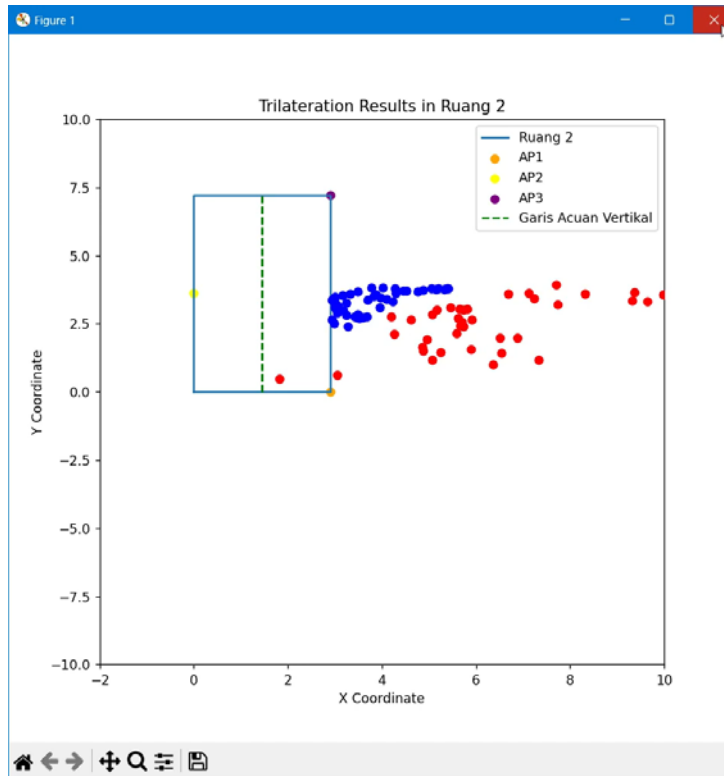
c. Pengujian ke-3, kondisi AP 1



d. Pengujian ke-4, kondisi AP 1



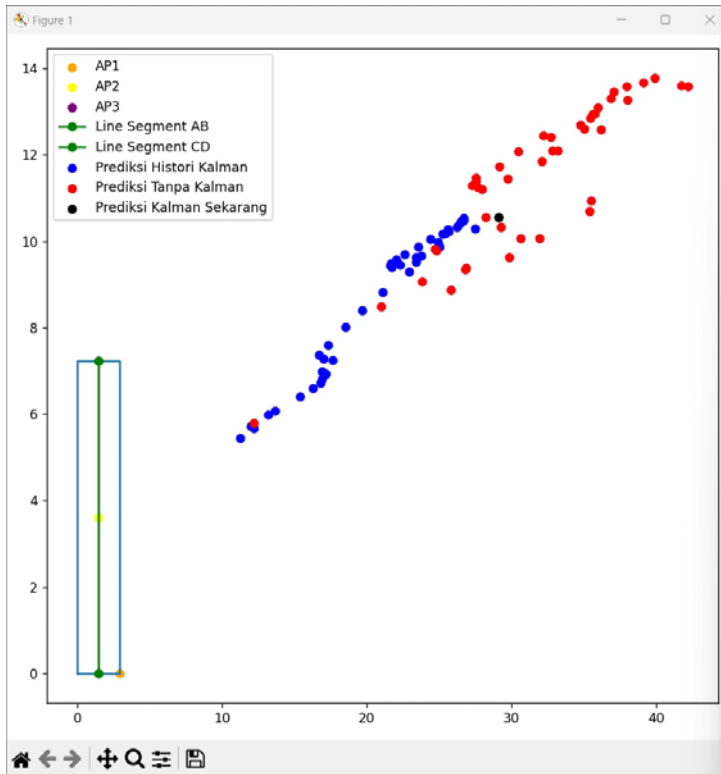
e. Pengujian ke-5, kondisi AP 1



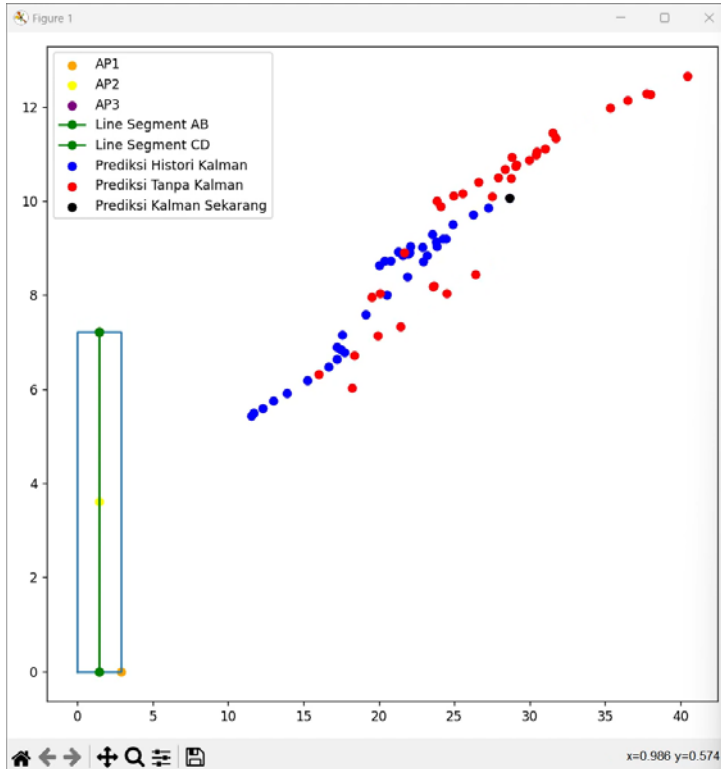
f. Tabel Data Pengujian Ruang 2, Kondisi Isi, Posisi AP 1

Pengujian ke-	Error Tanpa KF (m)	Error Dengan KF (m)	Persentase Penurunan Error
1	2.43	1.84	24.28%
2	3.88	1.97	49.23%
3	8.12	4.19	48.40%
4	1.06	0.64	39.62%
5	5.56	2.35	57.73%
Rata-Rata	4.21	2.198	47.79%

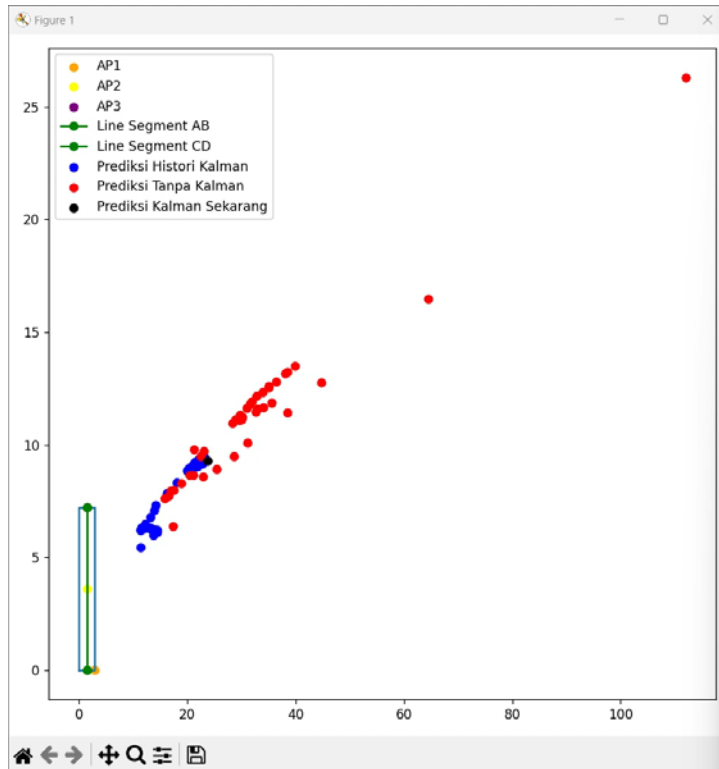
g. Pengujian ke-1, kondisi AP 2



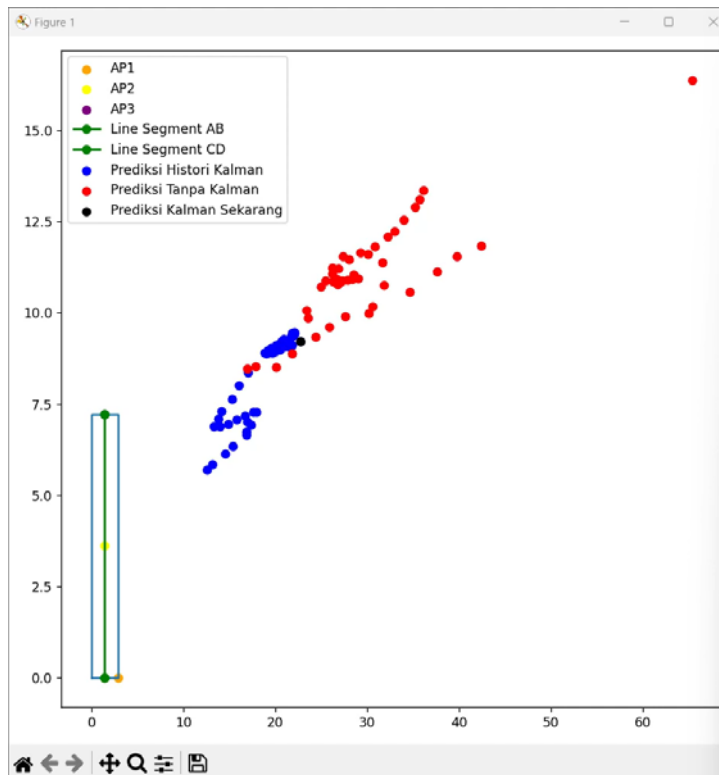
h. Pengujian ke-2, kondisi AP 2



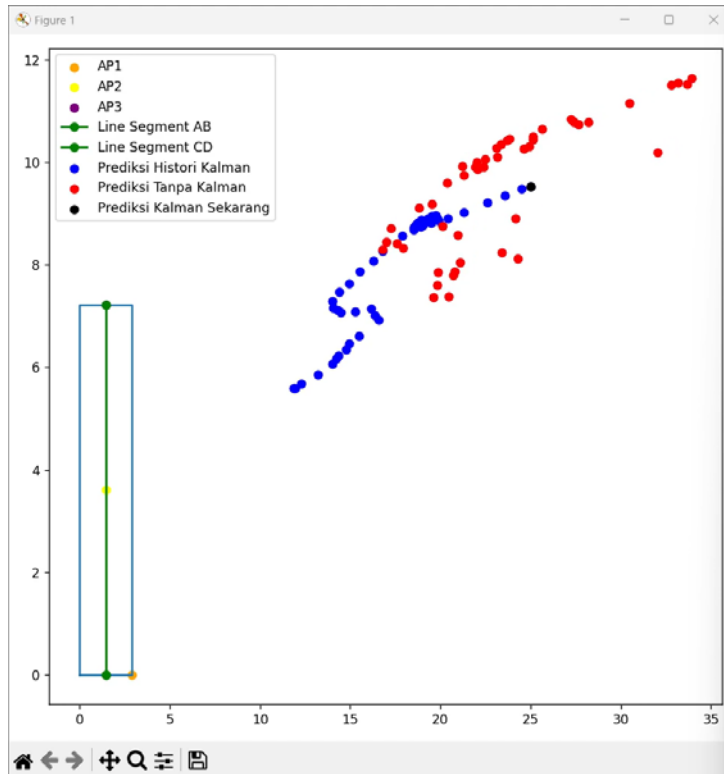
i. Pengujian ke-3, kondisi AP 2



j. Pengujian ke-4, kondisi AP 2



k. Pengujian ke-5, kondisi AP 2



a. Tabel Data Pengujian Ruang 2, Kondisi Isi, Posisi AP 2

Pengujian ke-	Error Tanpa KF (m)	Error Dengan KF (m)	Persentase Penurunan Error
1	30.23	19.48	35.56%
2	25.59	18.75	26.73%
3	30.83	16.13	47.68%
4	28.5	16.83	40.95%
5	22.01	15.85	27.99%
Rata-Rata	27.432	17.408	36.54%

LAMPIRAN C

a. Ruang 1, Isi, Posisi 1



b. Ruang 1, Isi, Posisi 2



c. Ruang 1, Kosong, Posisi 1



d. Ruang 1, Kosong, Posisi 2



e. Ruang 2, Isi, Posisi 1



f. Ruang 2, Isi, Posisi 2



g. Ruang 2, Kosong, Posisi 1



h. Ruang 2, Kosong, Posisi 2



LAMPIRAN D

A. Referensi RSSI Ruang 1 Posisi 1

AP1	AP2	AP3
-23	-24	-25

B. Referensi RSSI Ruang 1 Posisi 2

AP1	AP2	AP3
-23	-27	-23

C. Tabel Nilai N Ruang 1 Posisi 1 dan Posisi 2 AP1

Meter	RSSI	N	MSE (<i>Mean Square Error</i>)	RMSE (<i>Root Mean Square Error</i>)	Rata-Rata Error (m)
2	-31,14	2,70	0,441	0,664	0,56
3	-37,55	3,05	1,798	1,341	0,796
4	-43,44	3,40	5,625	2,371	1,886

D. Tabel Nilai N Ruang 1 Posisi 1 AP2

Meter	RSSI	N	MSE (<i>Mean Square Error</i>)	RMSE (<i>Root Mean Square Error</i>)	Rata-Rata Error (m)
2	-33,18	3,05	0,306	0,553	0,403
3	-33,04	1,89	3,243	1,8	0,992
4	-38,08	2,34	13,47	3,67	2,611

E. Tabel Nilai N Ruang 1 Posisi 1 AP3

Meter	RSSI	N	MSE (<i>Mean Square Error</i>)	RMSE (<i>Root Mean Square Error</i>)	Rata-Rata Error (m)
2	-31,89	2,29	1,57	1,253	1,069
3	-33,32	1,74	4,06	2,015	1,182
4	-34,63	1,60	19,787	4,44	1,505

F. Tabel Nilai N Ruang 1 Posisi 2 AP2

Meter	RSSI	N	MSE (<i>Mean Square Error</i>)	RMSE (<i>Root Mean Square Error</i>)	Rata-Rata Error (m)
2	-36,27	3,08	1,38	1,17	0,926
3	-38,73	2,46	1,269	1,12	0,828
4	-40,58	2,26	6,07	2,46	1,864

G. Tabel Nilai N Ruang 1 Posisi 2 AP3

Meter	RSSI	N	MSE (<i>Mean Square Error</i>)	RMSE (<i>Root Mean Square Error</i>)	Rata-Rata Error (m)
2	-30,68	2,55	1,591	1,261	0,651
3	-28,34	1,12	2,681	1,637	1,305
4	-32,01	1,50	42,653	6,530	3,039

H. Referensi RSSI Ruang 2 Posisi 1

AP1	AP2	AP3
-23	-23	-21

I. Referensi RSSI Ruang 2 Posisi 2

AP1	AP2	AP3
-23	-19	-20

J. Tabel Nilai N Ruang 2 Posisi 1 dan Posisi 2 AP1

Meter	RSSI	N	MSE (<i>Mean Square Error</i>)	RMSE (<i>Root Mean Square Error</i>)	Rata-Rata Error (m)
2	-33,06	3,34	0,061	0,248	0,197
3	-38,58	3,27	0,685	0,827	0,579
4	-37,85	2,47	8,179	2,859	2,148

K. Tabel Nilai N Ruang 2 Posisi 1 AP 2

Meter	RSSI	N	MSE (<i>Mean Square Error</i>)	RMSE (<i>Root Mean Square Error</i>)	Rata-Rata Error (m)
2	-29,93	2,30	0,260	0,510	0,370
3	-31,18	1,71	2,136	1,461	1,219
4	-36,63	2,26	0,776	0,881	0,652

L. Tabel Nilai N Ruang 2 Posisi 1 AP 3

Meter	RSSI	N	MSE (<i>Mean Square Error</i>)	RMSE (<i>Root Mean Square Error</i>)	Rata-Rata Error (m)
2	-30,01	2,99	0,180	0,425	0,354
3	-33,07	2,53	3,558	1,886	0,776
4	-32,92	1,98	4,078	2,019	1,059

M. Tabel Nilai N Ruang 2 Posisi 2 AP 2

Meter	RSSI	N	MSE (Mean Square Error)	RMSE (Root Mean Square Error)	Rata-Rata Error (m)
2	-26,33	2,43	10,51	3,241	1,119
3	-34,22	3,19	0,853	0,923	0,822
4	-34,05	2,5	0,313	0,56	0,34

N. Tabel Nilai N Ruang 2 Posisi 2 AP 2

Meter	RSSI	N	MSE (Mean Square Error)	RMSE (Root Mean Square Error)	Rata-Rata Error (m)
2	-34,95	4,97	0,121	0,348	0,227
3	-35,63	3,28	0,561	0,749	0,542
4	-40,95	3,48	2,832	1,683	1,152

LAMPIRAN E

a. Data Nilai Error Ruang 1, Kondisi Kosong, Posisi 1

Tanpa Kalman (m)	Dengan Kalman (m)
2.44	0.946
4.093	0.539
2.689	0.403
1.995	0.329
3.618	0.055
2.419	0.045
1.564	0.18
1.352	0.235
0.787	0.369
0.529	0.47
0.279	0.562
0.265	0.606
0.03	0.691
0.032	0.729
0.117	0.78
0.085	0.786
0.042	0.721
0.332	0.686
0.637	0.736
0.295	0.657
0.074	0.511
0.128	0.4
0.196	0.305
0.068	0.284

0.269	0.204
0.419	0.106
0.934	0.096
0.917	0.204
0.905	0.268
0.947	0.325
0.959	0.363
1.018	0.41
1.307	0.502
1.379	0.545
1.54	0.562
1.764	0.598
2.393	0.638
3.152	0.846
3.012	0.945
2.658	0.982
2.719	0.991
2.876	1.037
2.741	1.03
2.65	1.009
2.516	0.97
2.448	0.938
2.672	0.976
2.761	1.033
3.541	1.228

6.816	0.95
4.542	0.819
8.348	0.358
4.845	0.241
3.466	0.218
1.742	0.359
1.297	0.46
1.492	0.367
1.322	0.185
1.058	0.102
1.303	0.033
1.03	0.085
1.395	0.22
1.73	0.406
1.526	0.461
1.43	0.484
1.627	0.603
1.854	0.726
2.027	0.865
1.983	0.952
1.964	1.003
1.911	1.033
1.832	1.035
1.925	1.071
1.849	1.078
1.766	1.066
1.688	1.042

1.557	0.989
1.813	1.02
1.828	1.044
1.813	1.064
1.819	1.075
1.774	1.071
1.723	1.049
1.718	1.026
1.51	0.92
1.903	0.861
1.893	0.806
1.821	0.756
1.72	0.707
1.688	0.672
1.621	0.635
2.362	0.649
2.378	0.682
2.242	0.678
2.878	0.976
0.538	0.874
0.899	0.794
0.388	0.775
0.479	0.693
0.274	0.773
0.24	0.7
1.446	0.39
1.209	0.247

0.882	0.241
0.711	0.254
0.415	0.316
0.112	0.427
0.094	0.481
0.155	0.487
0.021	0.521
0.069	0.555
0.168	0.492
0.19	0.43
0.093	0.386
0.075	0.293
0.575	0.138
0.389	0.117
0.435	0.08
0.686	0.017
0.726	0.094
0.642	0.116
0.617	0.127
0.62	0.143
0.855	0.209
0.965	0.291
1.077	0.371
1.047	0.414
1.025	0.436
0.956	0.428
1.236	0.477

1.054	0.452
0.961	0.409
0.951	0.381
0.857	0.332
0.786	0.287
0.699	0.235
0.635	0.182
0.662	0.139
2.366	1.004
1.054	1.002
2.615	0.744
2.019	0.39
1.408	0.302
2.697	0.006
2.176	0.107
1.537	0.084
0.945	0.058
0.879	0.119
0.548	0.218
0.401	0.29
0.166	0.383
0.396	0.447
0.357	0.419
0.36	0.312
0.235	0.269
0.149	0.257
0.001	0.287

0.088	0.327
0.184	0.376
0.103	0.377
0.122	0.42
0.255	0.46
0.377	0.521
0.321	0.515
0.19	0.476
0.101	0.437
0.801	0.361
0.976	0.294
0.952	0.193
0.864	0.158
1.118	0.072
1.239	0.028
1.292	0.117
1.787	0.286
1.738	0.376
1.685	0.421
1.911	0.506
1.85	0.541
1.941	0.636
2.575	0.775
2.467	0.821
4.17	1.059
4.169	1.215
4.135	1.316

4.466	1.477
2.373	0.974
23.187	0.458
13.753	0.142
17.639	0.641
11.966	0.795
9.031	0.785
6.882	0.677
5.767	0.604
5.325	0.614
4.603	0.582
3.902	0.507
3.699	0.496
3.326	0.473
2.832	0.391
2.198	0.221
2.214	0.131
2.02	0.064
1.585	0.045
1.897	0.043
2.243	0.009
3.064	0.205
2.769	0.247
2.972	0.303
3.097	0.455
3.277	0.622
3.042	0.694

3.5	0.908
3.435	1.072
3.554	1.197
3.934	1.376
4.852	1.61
4.636	1.683
4.556	1.745
4.442	1.772
4.488	1.778
4.416	1.766
4.363	1.755
4.175	1.734
4.184	1.741
4.164	1.758
4.096	1.763
3.959	1.729
3.944	1.706
4.151	1.718
4.116	1.732
5.071	1.885

b. Data Nilai Error Ruang 1, Kondisi Kosong, Posisi 2

Tanpa Kalman (m)	Dengan Kalman (m)
6.646	1.706
21.499	1.135
9.028	0.528
10.166	0.047
7.817	0.209
6.319	0.335
5.111	0.4
4.026	0.572
3.719	0.626
3.684	0.778
3.019	0.806
2.447	0.889
2.566	0.968
2.507	0.983
2.624	1.002
2.983	1.049
3.276	1.127
3.476	1.152
4.742	1.116
4.356	1.21
3.755	1.154
3.387	1.074
2.993	1.004
3.065	0.945
2.751	0.889
2.578	0.878
2.517	0.9
2.32	0.925
2.167	0.94
2.405	1.012

2.281	1.095
2.025	1.121
2.126	1.184
1.838	1.193
1.926	1.228
2.252	1.304
0.349	1.926
0.261	1.819
0.989	1.438
0.653	1.281
0.69	1.132
0.495	1.034
0.704	0.883
0.878	0.556
0.738	0.444
0.786	0.224
0.674	0.026
0.893	0.108
0.929	0.311
1.001	0.455
1.175	0.607
1.265	0.677
1.309	0.713
1.17	0.728
1.186	0.713
1.27	0.715
0.132	0.481
0.801	0.21
0.85	0.037
1.604	0.202
1.521	0.349

1.276	0.425
1.226	0.432
1.172	0.432
1.016	0.397
0.928	0.355
0.881	0.301
0.402	0.166
0.216	0.03
0.162	0.07
0.074	0.155
0.4	0.462
0.621	0.538
0.767	0.61
1.262	1.97
1.07	1.669
1.292	1.008
1.609	0.638
1.791	0.164
0.838	0.053
0.622	0.063
0.632	0.079
0.804	0.695
0.669	0.683
0.657	0.664
0.945	0.555
1.279	0.468
1.96	0.38
1.0	0.26
2.948	0.331
3.12	0.594
3.975	1.056
5.378	1.71

5.835	2.313
6.224	3.148
6.587	3.521
6.286	3.613
5.771	3.505
5.605	3.408
5.035	3.205
4.597	2.8
4.393	2.71
3.901	2.529
3.804	2.409
3.501	2.243
3.206	2.062
2.8	1.852
2.244	1.605
2.734	1.616
3.218	1.789
3.164	1.959
3.119	2.057
3.161	2.253
3.182	2.297
3.229	2.358
2.789	2.314
10.073	2.248
1.141	1.715
0.214	1.415
0.324	1.203
3.781	1.129
1.808	0.797
0.607	0.353
0.49	0.463
0.614	0.936

0.632	0.966
0.612	0.988
0.563	0.964
0.992	1.065
1.392	1.148
1.85	1.279
2.003	1.387
2.157	1.482
4.283	2.071
6.141	2.771
6.433	3.337
6.769	3.683
6.476	3.868
6.724	4.089
5.539	3.972
5.743	3.948
5.92	3.98
5.446	3.911
5.61	3.952
5.276	3.878
5.535	3.956
5.493	3.977
4.654	3.826
4.164	3.62
4.401	3.563
4.68	3.633
4.635	3.643
4.134	3.517
3.89	3.387
3.77	3.279
4.591	3.482
5.006	3.785

1.008	1.732
2.7	1.159
2.629	0.862
1.993	0.694
0.912	0.693
1.136	0.607
2.406	0.284
2.769	0.277
2.878	0.496
2.913	0.779
2.647	1.01
4.995	1.322
4.171	1.238
4.701	1.256
5.562	1.517
6.329	1.984
6.457	2.376
6.859	3.004
6.935	3.263
6.476	3.424
6.896	3.665
6.434	3.695
7.98	4.283
7.418	4.304
7.116	4.249
7.324	4.327
6.389	4.052
5.925	3.792
5.691	3.58
6.359	3.618
5.343	3.332
5.604	3.269

5.742	3.238
6.398	3.374
6.261	3.426
6.243	3.615
6.267	3.651

6.153	3.674
5.977	3.658
5.741	3.596

c. Data Nilai Error Ruang 1, Kondisi Isi, Posisi 1

Tanpa Kalman (m)	Dengan Kalman (m)
19.845	0.795
9.688	0.601
10.002	0.2
11.413	0.431
7.944	0.572
7.169	0.778
5.338	0.893
4.564	0.893
3.955	0.798
3.328	0.677
3.245	0.616
3.73	0.577
3.793	0.71
3.738	0.931
3.345	0.986
2.965	0.971
2.647	0.956
2.477	0.93
3.218	1.063
2.654	1.025
2.144	0.946
2.053	0.881
2.213	0.915
2.056	0.896

1.945	0.867
2.246	0.898
2.238	0.952
2.363	1.032
2.39	1.103
2.317	1.107
2.381	1.093
2.517	1.051
2.154	0.944
2.714	0.946
2.528	0.912
2.424	0.882
2.291	0.823
4.863	0.95
4.82	1.073
4.476	1.101
4.691	1.21
4.837	1.346
5.505	1.1
5.786	0.797
9.044	0.284
5.483	0.334
5.178	0.547
3.442	0.428
4.282	0.582

3.696	0.641
3.062	0.594
2.548	0.497
2.412	0.458
2.363	0.459
2.207	0.405
1.7	0.346
1.93	0.502
2.178	0.621
2.433	0.785
2.07	0.793
2.674	0.925
2.549	1.034
2.552	1.152
3.362	1.529
3.278	1.651
3.681	1.881
3.318	1.865
3.239	1.836
3.323	1.831
3.476	1.823
3.645	1.833
4.045	1.949
4.075	2.054
4.052	2.135
4.215	2.21
4.235	2.266
4.237	2.3
4.213	2.316
4.259	2.353
4.124	2.355
3.996	2.327

3.981	2.315
4.261	2.394
4.21	2.417
5.017	2.606
4.888	2.686
6.314	3.005
6.372	3.238
4.398	0.928
5.729	0.773
8.339	0.352
12.563	0.452
8.516	0.623
11.451	1.259
9.5	1.37
7.89	1.24
6.376	1.008
7.854	1.16
6.032	1.006
5.54	1.056
4.868	1.109
4.17	1.141
4.343	1.294
3.688	1.327
3.799	1.451
3.524	1.499
4.306	1.735
3.857	1.745
4.132	1.831
3.686	1.753
4.289	1.817
4.201	1.88
4.09	1.883

4.157	1.947
4.376	2.028
4.401	2.032
4.664	2.123
4.509	2.181
4.351	2.162
4.152	2.102
4.162	2.071
4.686	2.012
4.882	1.967
5.459	1.992
5.354	2.003
4.884	1.947
4.807	1.886
4.686	1.827
4.69	1.792
4.735	1.798
4.316	1.008
6.331	0.796
14.843	0.164
9.911	0.406
11.339	0.932
8.501	0.889
5.971	0.567
5.624	0.402
5.736	0.399
4.577	0.548
4.714	0.751
3.93	0.84
7.377	1.384
5.787	1.306
7.069	1.532

6.473	1.586
8.854	2.001
7.905	1.987
7.315	1.927
7.389	2.036
7.157	2.255
7.645	2.419
8.874	2.82
9.074	3.129
9.177	3.322
8.801	3.358
8.428	3.312
7.989	3.228
7.655	3.106
7.635	3.018
7.307	2.906
11.293	3.157
10.81	3.233
10.679	3.301
12.197	3.569
13.892	4.131
13.694	4.481
16.73	5.257
16.614	5.76
3.273	0.937
2.915	0.801
1.878	0.704
2.176	0.525
2.164	0.33
2.169	0.123
2.559	0.134
3.016	0.453

2.427	0.521
1.931	0.484
1.549	0.389
1.5	0.323
1.323	0.187
1.357	0.116
1.254	0.201
1.09	0.318
1.736	0.498
1.764	0.667
1.766	0.786
1.658	0.855
1.811	0.953
1.389	0.894
1.839	0.971
1.768	1.003
2.401	1.159
2.466	1.289
2.413	1.357
2.301	1.367
2.166	1.332
2.459	1.385
2.41	1.412
2.628	1.495
2.597	1.54
2.791	1.656
2.817	1.66
2.73	1.639
2.666	1.608
2.591	1.575
2.512	1.529

2.561	1.515
2.497	1.479
2.361	1.441
2.367	1.418
2.189	1.374
2.13	1.328
2.074	1.29
2.691	1.38
3.079	1.55

d. Data Nilai Error Ruang 1, Kondisi Isi, Posisi 2

Tanpa Kalman (m)	Dengan Kalman (m)
0.491	1.97
3.828	1.182
2.644	0.922
2.195	0.811
1.71	0.773
1.636	0.672
1.538	0.415
1.514	0.063
1.561	0.151
3.449	0.771
3.019	0.912
4.643	0.929
6.878	1.34
8.158	1.956
7.984	2.433
8.6	2.917
8.088	3.459
8.39	3.733
9.063	4.122
9.059	4.418
9.316	4.633
7.367	4.404
7.254	4.283
6.495	4.034
5.931	3.765
5.351	3.446
6.421	3.648
5.112	3.403
4.934	3.23
4.958	3.161

5.158	3.202
5.196	3.267
5.287	3.357
4.997	3.318
5.976	3.509
6.419	3.816
6.593	4.082
6.518	4.24
0.533	1.847
0.091	1.795
0.486	1.617
0.239	1.48
0.637	1.254
1.618	0.755
2.462	0.313
3.12	0.165
4.309	0.776
4.117	1.281
4.048	1.528
4.316	1.726
4.338	1.918
4.05	2.061
5.36	2.391
5.267	2.447
5.574	2.614
6.566	2.962
7.033	3.334
7.085	3.662
8.244	4.156
7.377	4.49
6.257	4.335

5.703	4.075
5.642	3.918
5.302	3.74
4.467	3.438
4.159	3.159
4.155	2.995
4.021	2.744
4.099	2.753
3.9	2.717
3.48	2.603
3.686	2.613
3.718	2.653
3.652	2.67
2.969	2.579
3.402	2.647
2.414	2.557
2.482	2.538
4.659	3.246
5.715	3.9
1.243	1.927
1.081	1.503
2.689	1.116
3.544	1.252
2.482	1.251
1.459	1.087
0.084	0.724
1.18	0.21
1.946	0.828
2.222	1.197
2.795	1.657
4.263	2.216
5.384	2.85

7.309	3.961
7.596	4.482
8.231	5.002
9.068	5.627
10.897	7.043
12.301	7.892
11.95	8.301
7.753	8.557
5.995	8.267
7.356	8.274
8.101	8.509
7.293	8.206
6.432	7.761
5.444	7.206
8.257	7.154
8.846	7.313
9.415	7.625
10.088	8.023
9.756	8.113
9.477	8.182
10.299	8.374
10.206	8.474
8.804	8.383
10.252	8.385
10.012	8.492
9.663	8.583
9.486	8.65
10.26	8.985
10.079	9.141
1.481	2.03
4.094	1.413
9.304	0.56

4.676	0.35
2.016	0.502
5.771	0.023
8.101	1.558
7.134	1.983
6.872	2.337
6.443	2.542
9.141	3.213
10.732	3.468
11.787	4.519
13.539	5.467
14.156	6.377
14.632	7.167
15.706	7.921
13.806	8.493
10.221	8.549
9.285	8.239
7.566	7.816
4.154	6.94
4.064	6.347
3.746	5.874
3.272	5.136
3.918	5.045
3.643	4.883
4.281	4.949
4.86	5.138
4.905	5.378
3.81	5.172
3.995	5.088
6.071	5.389
5.88	5.461
6.38	5.601

8.276	6.383
7.83	6.499
8.201	6.682
8.017	6.772
8.51	7.027
0.125	1.815
0.685	1.725
0.634	1.547
1.231	1.382
0.723	1.258
0.619	1.113
1.368	0.579
2.383	0.104
3.273	0.395
3.097	0.711
3.604	1.298
5.06	1.913
6.904	2.157
8.208	2.545
8.908	3.49
9.056	3.958
9.562	4.412
9.806	4.833
13.635	6.433
15.306	7.35
14.67	7.61
14.403	7.802
14.609	7.984
13.152	7.817
11.968	7.326
11.443	6.915
9.943	6.242

9.223	5.647
9.951	5.397
9.071	4.87
8.471	4.618
7.675	4.26
6.745	3.79
7.282	3.516
7.959	3.543

7.734	3.546
8.658	3.839
8.415	3.942
9.787	4.302
9.887	4.709
9.788	4.819
10.163	5.054

e. Data Nilai Error Ruang 2, Kondisi Kosong, Posisi 1

Tanpa Kalman (m)	Dengan Kalman (m)
3.151	2.313
3.399	2.446
2.767	2.428
1.458	2.216
1.064	2.012
0.822	1.811
1.087	1.734
1.091	1.669
1.072	1.611
1.271	1.605
1.179	1.566
0.996	1.484
1.209	1.483
1.254	1.488
1.512	1.551
1.772	1.655
1.976	1.771
2.036	1.859
2.006	1.922
2.354	2.023
2.071	2.012

2.127	2.034
2.116	2.04
2.086	2.036
2.013	2.009
1.845	1.953
2.045	1.963
1.645	1.882
1.689	1.848
2.143	1.916
2.128	1.951
1.982	1.943
1.979	1.939
1.822	1.892
1.878	1.883
1.965	1.9
2.188	1.974
2.255	2.044
2.724	2.317
3.126	2.517
3.14	2.636
3.843	2.87
3.877	3.021

5.397	3.39
9.384	1.911
1.133	1.824
0.482	1.876
0.713	1.865
1.825	1.935
1.542	1.883
0.917	1.752
0.742	1.611
0.81	1.526
0.574	1.404
1.533	1.437
1.686	1.489
1.975	1.593
2.715	1.806
3.654	2.139
3.805	2.404
4.562	2.936
5.149	3.298
5.064	3.511
4.498	3.527
4.223	3.431
3.756	3.252
4.21	3.238
4.256	3.248
4.242	3.272
4.254	3.299
4.282	3.345
4.106	3.336
4.304	3.403
4.23	3.453
4.912	3.649

6.875	4.17
6.943	4.693
6.787	4.805
6.53	4.782
6.548	4.794
6.437	4.781
0.699	2.147
10.585	2.46
5.85	2.631
5.401	2.739
6.061	2.97
4.917	2.887
5.198	2.925
4.369	2.836
4.004	2.75
3.575	2.612
5.863	2.764
5.666	2.87
5.454	2.947
5.572	3.07
6.499	3.363
6.889	3.686
7.029	3.964
8.997	4.963
8.627	5.2
8.535	5.364
7.859	5.25
7.174	5.017
6.878	4.832
6.488	4.631
6.057	4.414
5.542	4.18

5.623	4.103
5.481	4.034
5.434	4.001
5.499	4.009
5.418	4.007
5.233	3.964
5.498	4.04
5.33	4.039
5.742	4.143
6.41	4.39
6.424	4.57
29.807	2.604
8.342	2.368
16.164	2.611
10.335	2.55
19.776	3.145
14.23	3.076
13.148	3.255
12.099	3.167
10.229	2.972
9.363	2.846
9.43	2.904
9.915	3.095
9.395	3.237
14.624	3.932
20.125	5.286
20.435	6.41
18.907	6.852
18.975	7.294
17.33	7.168
15.108	6.749
14.041	6.362

11.828	5.529
10.747	5.082
10.566	4.842
9.519	4.535
9.067	4.29
7.875	3.873
7.612	3.619
7.517	3.463
6.844	3.212
6.418	2.979
6.306	2.861
6.189	2.78
6.943	2.929
6.702	3.0
6.848	3.115
7.412	3.35
7.744	3.642
4.402	2.348
23.388	2.8
11.842	2.721
17.375	3.105
12.257	3.043
9.126	2.878
6.709	2.508
14.695	2.656
12.423	2.659
11.551	2.672
9.544	2.447
12.039	2.493
10.45	2.378
14.205	2.59
14.016	3.27

13.116	3.398
21.146	4.199
20.469	4.749
19.748	5.112
17.988	4.926
17.225	4.779
16.153	4.627
15.328	4.469
14.32	4.264
14.958	4.245
14.23	4.177
14.785	4.259

16.361	4.65
18.502	5.362
17.444	5.522
17.174	5.652
15.46	5.421
14.719	5.237
14.494	5.143
13.786	4.967
13.967	4.881
13.849	4.907
13.798	4.978

f. Data Nilai Error Ruang 2, Kondisi Kosong, Posisi 2

Tanpa Kalman (m)	Dengan Kalman (m)
10.267	9.767
43.912	10.812
28.918	11.116
29.162	11.875
23.942	12.063
23.273	12.447
19.956	12.326
19.61	12.593
23.853	13.534
27.847	15.039
24.438	15.052
23.48	15.026
26.329	15.604
26.62	16.202
26.994	16.81
28.961	17.793
29.903	18.805

30.03	19.609
30.153	20.276
30.121	20.811
29.419	21.023
32.323	21.916
33.505	22.908
32.945	23.426
31.263	23.521
30.299	23.189
32.227	23.425
32.401	23.712
32.817	24.111
36.057	25.126
38.067	26.412
37.263	26.92
36.929	27.213
36.618	27.388
36.828	27.648

38.45	28.273
38.532	28.77
39.183	29.351
302.431	12.142
165.462	14.278
86.86	14.11
101.785	16.657
76.514	17.507
68.061	18.569
53.589	18.012
46.574	17.448
41.87	17.119
37.782	16.778
37.029	16.886
34.71	16.895
31.409	16.423
28.364	15.863
29.722	16.009
28.88	16.11
31.057	16.644
30.309	16.965
31.813	17.653
30.78	17.997
36.332	19.346
34.15	19.405
37.106	20.334
35.656	20.575
43.706	22.169
41.994	22.736
40.706	22.921
40.131	23.104
38.845	23.018

38.525	23.038
45.218	24.905
45.002	25.843
44.539	26.444
46.775	27.464
45.789	27.877
44.716	27.941
44.404	28.041
16.104	9.998
14.961	10.269
77.854	12.825
52.77	13.518
76.578	16.473
60.764	17.388
54.809	18.182
46.727	18.056
40.639	17.533
37.684	17.274
35.656	17.168
30.169	16.095
29.247	15.892
27.518	15.619
32.101	16.28
30.639	16.528
31.048	16.967
32.095	17.61
35.402	18.758
42.516	21.174
40.932	22.313
45.158	24.119
43.056	24.603
54.908	26.817

52.785	27.581
51.777	28.044
49.25	27.736
48.605	27.834
47.199	27.699
46.56	27.633
47.03	27.857
46.414	28.002
47.107	28.408
47.291	28.862
46.748	29.152
46.237	29.334
46.33	29.611
45.454	29.644
19.617	10.241
32.505	11.219
22.405	11.368
21.143	11.706
19.737	12.035
19.994	12.484
17.679	12.39
16.852	12.324
16.436	12.334
15.49	12.251
14.311	11.963
12.842	11.42
12.984	11.184
15.424	11.559
22.358	12.812
23.686	14.077
24.622	15.229
24.633	16.06

22.571	16.207
22.964	16.567
23.576	17.017
22.15	16.799
22.938	16.861
21.588	16.496
21.557	16.328
21.142	16.201
21.759	16.354
21.786	16.522
20.501	16.216
21.066	16.222
23.943	16.737
23.13	17.033
23.445	17.247
25.128	17.811
27.024	18.696
31.765	20.304
31.371	21.072
31.29	21.625
3.889	9.658
9.141	9.814
9.447	9.871
19.532	10.886
17.512	11.411
15.385	11.879
19.808	12.512
21.524	13.52
31.502	15.775
28.33	16.395
25.263	16.199
24.814	16.294

43.742	18.319
41.476	19.42
55.884	22.365
51.962	23.209
50.077	23.719
49.131	24.171
53.988	25.696
58.51	27.948
56.37	29.044
52.712	28.63
53.212	28.653
50.359	28.09
47.208	27.189

45.806	26.732
44.01	26.177
42.498	25.636
41.574	25.252
46.434	25.73
47.979	26.541
48.454	27.351
47.463	27.7
51.46	28.754
50.942	29.412
50.692	29.918
50.435	30.331

g. Data Nilai Error Ruang 2, Kondisi Isi, Posisi 1

Tanpa Kalman (m)	Dengan Kalman (m)
57.818	10.772
44.188	11.776
40.686	14.807
41.899	16.881
36.045	17.69
29.479	17.038
25.962	16.309
22.776	15.422
21.034	14.747
20.159	14.296
19.117	13.905
29.234	14.791
30.132	15.717
30.079	16.497
30.691	17.363
27.734	17.256

26.696	17.28
28.266	17.673
27.909	17.944
26.283	17.829
25.512	17.727
25.118	17.68
25.699	17.935
25.257	18.05
24.689	18.058
24.608	18.127
24.968	18.357
25.566	18.735
25.808	19.112
25.648	19.357
26.611	19.818
27.451	20.803
27.727	21.251

31.467	22.364
33.006	23.593
43.199	26.354
43.143	27.96
49.234	10.617
35.198	11.354
36.416	12.666
33.983	15.367
28.278	15.164
23.231	14.311
21.143	13.703
18.084	12.927
16.835	12.335
14.159	11.437
15.608	11.176
17.392	11.161
17.796	11.513
17.61	11.783
18.377	12.235
25.024	13.644
27.552	15.237
46.514	20.208
47.294	21.202
45.263	21.555
44.582	21.898
44.504	22.384
41.109	22.341
47.465	23.322
50.874	24.974
52.413	26.624
52.89	28.007
56.638	29.931

59.307	31.959
58.022	32.627
55.938	32.619
57.262	33.167
57.48	34.349
60.386	35.598
19.071	10.201
49.167	13.242
38.868	14.419
37.508	15.718
72.749	20.127
64.048	22.598
48.104	22.85
43.769	22.102
39.75	21.292
44.715	21.781
38.173	21.105
40.3	21.294
43.619	22.401
45.3	24.972
52.761	27.264
51.068	28.539
50.268	29.39
83.835	37.101
82.188	40.371
82.579	42.975
78.234	43.18
77.312	43.451
74.228	43.058
70.08	41.546
71.138	41.755
71.721	42.294

71.368	42.816
72.254	43.653
77.036	45.498
77.48	47.766
74.976	47.791
75.033	48.029
76.434	48.804
75.935	49.354
74.87	49.59
14.389	9.913
24.761	10.857
21.094	11.41
22.795	12.271
18.583	12.32
14.47	11.66
8.276	9.967
8.409	9.321
6.952	8.522
7.359	8.048
8.293	8.074
8.861	8.258
12.914	8.85
11.652	8.853
14.416	9.459
12.031	9.256
22.548	10.411
20.967	11.504
19.397	11.309
19.454	11.24
19.114	11.287
17.233	10.898
17.67	10.907

22.012	11.999
21.56	12.517
22.054	13.061
21.852	13.431
20.935	13.378
21.532	13.714
21.418	13.895
21.833	14.18
24.844	14.997
27.418	16.197
27.13	17.322
26.85	17.601
27.177	17.929
27.775	18.366
28.408	18.906
28.366	19.544
62.373	10.918
52.46	12.152
46.404	13.602
43.0	15.228
34.289	15.724
27.369	15.138
40.797	17.203
43.733	18.162
38.802	18.226
39.468	18.736
35.229	18.366
33.806	18.139
39.271	18.74
41.55	19.914
52.875	22.678
49.263	23.46

63.549	26.277
83.968	31.628
79.836	33.362
78.28	34.546
74.66	34.731
67.725	34.382
65.905	33.36
63.46	32.845
59.352	31.935
55.859	30.756
55.151	30.009

50.42	28.025
49.54	27.481
49.212	27.165
47.65	26.737
51.755	27.164
51.829	28.233
52.069	28.654
52.434	29.179
52.493	29.726
52.089	30.129

h. Data Nilai Eror Ruang 2, Kondisi Isi, Posisi 2

Tanpa Kalman (m)	Dengan Kalman (m)
10.732	9.805
34.252	10.726
19.582	10.54
29.327	11.743
24.445	12.217
34.112	13.935
30.64	14.835
28.524	15.365
25.5	15.736
22.468	15.452
27.982	16.209
23.539	15.561
23.399	15.276
26.964	15.907
30.978	17.12
31.747	18.268
34.017	19.718

37.066	21.586
34.8	22.066
34.448	22.444
38.209	23.621
37.084	24.185
35.954	24.348
39.013	25.464
36.15	25.486
34.651	25.19
33.776	24.962
31.737	24.402
31.18	23.957
29.424	23.139
28.084	22.268
26.425	21.309
26.405	20.747
26.157	20.368
26.488	20.288

26.804	20.408
28.607	20.976
32.109	22.116
35.161	23.745
41.258	27.867
16.749	10.098
14.55	10.26
16.931	10.823
18.452	11.538
19.979	12.458
23.078	13.778
24.977	15.233
22.197	15.764
22.141	16.239
18.641	15.991
18.079	15.765
20.283	16.121
27.501	19.094
28.755	20.464
29.189	21.549
27.856	21.797
29.28	22.441
29.802	23.096
27.879	22.888
26.643	22.425
24.263	21.522
22.787	20.556
27.115	20.66
25.352	20.176
23.679	19.383
22.551	18.627
27.619	18.978

30.342	19.918
30.525	20.703
34.208	22.168
36.661	24.908
36.917	25.926
39.38	27.343
15.918	9.986
112.229	12.324
63.737	12.935
43.608	12.82
37.278	13.019
29.817	12.702
24.048	12.368
21.516	12.031
19.705	11.688
17.524	11.144
15.55	10.48
14.413	9.968
15.121	9.922
16.089	10.117
19.015	10.785
21.183	11.668
21.716	12.46
19.98	12.639
32.91	14.762
34.027	16.586
37.045	18.725
38.97	20.813
37.49	21.719
35.403	21.686
33.944	21.402
31.719	20.705

30.407	19.909
29.897	19.716
28.569	19.284
27.727	18.886
27.201	18.596
28.424	18.664
28.937	18.912
28.411	19.006
28.883	19.297
31.725	20.143
31.445	20.635
32.887	21.388
34.416	22.382
34.457	23.084
64.541	11.085
38.541	11.661
41.225	13.128
33.35	13.941
36.332	15.389
28.835	15.407
29.284	15.883
24.551	15.522
30.59	16.479
23.001	15.262
20.47	14.302
18.704	13.453
16.464	12.558
15.546	11.851
22.291	12.346
22.1	12.723
25.572	13.838
25.037	14.634

26.289	15.601
35.228	18.088
34.762	19.585
34.243	20.489
32.979	20.757
31.927	20.754
31.182	20.685
29.691	20.332
28.143	19.724
26.895	19.268
25.732	18.702
25.057	18.238
24.312	17.813
23.795	17.484
25.098	17.628
25.483	17.874
25.846	18.262
26.107	18.652
26.634	19.119
27.331	19.665
27.184	20.011
27.786	20.454
30.492	21.409
30.75	10.404
19.029	10.505
18.156	10.836
22.875	11.771
21.959	12.577
19.271	12.772
18.396	12.908
18.424	13.495
19.652	14.056

22.781	15.143
19.568	14.943
18.719	14.716
16.525	13.816
15.37	13.026
16.21	12.846
15.585	12.627
15.859	12.57
17.488	12.961
18.173	13.494
19.067	14.115
20.013	14.857
19.942	15.378
22.628	16.477
22.48	17.119
22.149	17.436
21.857	17.581
20.771	17.335

20.679	17.265
24.447	17.932
23.902	18.175
23.901	18.383
23.374	18.357
23.665	18.453
21.915	18.014
21.132	17.527
20.786	17.141
26.042	17.653
26.202	18.106
26.45	18.536
27.004	19.04
29.277	19.949
31.635	21.234
32.022	22.237
32.506	23.666