Design and Implementation of an Android-Based Indoor Signal Strength Positioning System Using Multivariate Gaussian Mixture Model with Wi-Fi RSSI Fingerprinting

Kusuma Wardhana[1*], Lusia Rakhmawati[2]
[1,2]Department of Electrical Engineering, Universitas Negeri Surabaya
A5 Building Ketintang Campus, Surabaya 60231, Indonesia
[1]kusuma.21038@mhs.unesa.ac.id
[2]lusiarakhmawati@unesa.ac.id

Abstract – Indoor positioning systems (IPS) are crucial where GPS accuracy is limited, but Wi-Fi RSSI-based methods face challenges from signal fluctuations and computational complexity. This research designed and implemented an Android application for indoor signal-strength positioning using a Multivariate Gaussian Mixture Model (MGMM) algorithm based on Wi-Fi RSSI fingerprinting. The system utilized three 2.4 GHz access points to collect Received Signal Strength Indicator (RSSI) data, building a fingerprint database. MGMM was integrated with Maximum Likelihood Estimation (MLE) for parameter estimation and Bayes' Theorem for probabilistic position determination. Testing was conducted in furnished and unfurnished rooms (30 trials per condition). Results showed 90% accuracy (within a 1-meter tolerance radius), a Mean Absolute Error (MAE) of 0.433 meters, and a Root Mean Square Error (RMSE) of 0.796 meters in furnished environments. In unfurnished rooms, the system achieved 100% accuracy (MAE and RMSE = 0 meters). The average system latency was 62 ms, confirming real-time responsiveness. This study demonstrates MGMM's effectiveness in modeling RSSI distributions and enhancing IPS accuracy.

Keywords: Indoor Positioning System, Wi-Fi RSSI-Fingerprint, Multivariate Gaussian Mixture Model, Android Application, Accuracy.

I. INTRODUCTION

Indoor Positioning Systems (IPS) have become indispensable in modern applications where Global Positioning System (GPS) accuracy is severely limited, such as hospitals, shopping malls, warehouses, and emergency response scenarios [1,2]. While GPS excels in open environments, its signals attenuate significantly indoors due to structural obstructions (e.g., walls, ceilings), leading to positioning errors exceeding 5–10 meters [3]. This limitation has spurred interest in alternative technologies, with Wi-Fi Received Signal Strength Indicator (RSSI)-based methods emerging as a cost-effective solution due to the ubiquity of existing Wi-Fi infrastructure [4]. However, traditional RSSI-fingerprinting approaches face critical challenges: signal fluctuations caused by multipath fading, interference from electronic devices, and environmental dynamics (e.g., moving people or furniture) degrade accuracy [5,6]. Additionally, computational complexity in algorithms like K-Nearest Neighbors (KNN) and Support Vector Machines (SVM) hinders real-time performance on mobile devices [7,8]. These issues necessitate robust, efficient solutions for reliable indoor localization.

To address these challenges, this research proposes the integration of a Multivariate Gaussian Mixture Model (MGMM) algorithm into an Android-based IPS. Unlike deterministic methods, MGMM probabilistically models the joint distribution of RSSI signals across multiple access points (APs), capturing inherent uncertainties caused by environmental noise [9]. By combining Maximum Likelihood Estimation (MLE) for parameter optimization and Bayes' Theorem for posterior probability estimation, MGMM adapts dynamically to signal variations while maintaining computational efficiency [10,11]. Prior studies demonstrate MGMM's superiority over KNN and SVM in handling multimodal RSSI distributions—common in indoor environments due to multipath effects [12]—but few have implemented it in a practical, end-to-end mobile application. For instance, Alfakih et al. [13] validated MGMM's theoretical accuracy but did not deploy it on resource-constrained devices, while Zhu et al. [14] used neural networks for fingerprinting at the cost of high computational overhead. This research bridges this gap by developing a lightweight, real-time Android application leveraging MGMM's strengths for scalable IPS deployment.

The primary objectives of this study are threefold:

- Design and implement an Android application that estimates indoor positions using MGMM-based Wi-Fi RSSI fingerprinting, supported by a backend system (Express.js) and MySQL database for fingerprint storage.
- Evaluate the system's accuracy and error metrics—including success rate within a 1-meter tolerance radius, Mean Absolute Error (MAE), and Root Mean Square Error (RMSE)—in both furnished and empty environments.
- Measure system latency to ensure real-time responsiveness suitable for navigation applications.

Relevant literature underscores the viability of this approach. Rizk et al. [15] achieved high accuracy using hybrid RSSI/RTT methods but noted added complexity from specialized hardware. In contrast, our system relies solely on commercial Wi-Fi APs. Liu et al. [16] confirmed MGMM's effectiveness in modeling RSSI distributions but highlighted sensitivity to initial parameters—a risk mitigated here through MLE-based calibration. Meanwhile, Xia et al. [17] emphasized fingerprinting's superiority over trilateration in cluttered spaces but identified database-update bottlenecks, which our static fingerprint database streamlines. Crucially, this research extends Li et al.'s [18] probabilistic framework by integrating MGMM into a user-friendly mobile platform, demonstrating practical viability where prior works focused on simulations. By addressing signal instability through probabilistic modeling and optimizing for mobile efficiency, this work offers a significant step toward accessible, high-accuracy indoor navigation.

II. METHODS

Figure 1 illustrates the research methodology flowchart employed by the authors to acquire the necessary data. The process initiates with a Literature Review, during which the authors analyze prior studies and academic publications to identify areas requiring further investigation. The insights derived from this review provide the foundation for selecting a research topic and subsequently formulating the problem to be addressed.

System Architecture Design

Figure 2 illustrates the RSSI data processing workflow for the Wi-Fi fingerprint-based indoor localization system. The Android smartphone acquires RSSI measurements from three access points (APs) during the data acquisition phase. A dataset of 1000 RSSI samples is compiled through random selection of 900 values from collected measurements, supplemented by 100 unique readings.

This dataset undergoes Maximum Likelihood Estimation (MLE) processing to compute critical parameters: the mean values and covariance matrix characterizing RSSI distribution. These parameters form the basis for Multivariate Gaussian Mixture Models (MGMM) at designated locations. Using Retrofit, the results are transmitted to an Express.js backend server via RESTful API, where they are persistently stored in a MySQL database. The database's MLE table systematically organizes location coordinates (x, y), mean values, and covariance matrices for subsequent positioning operations.

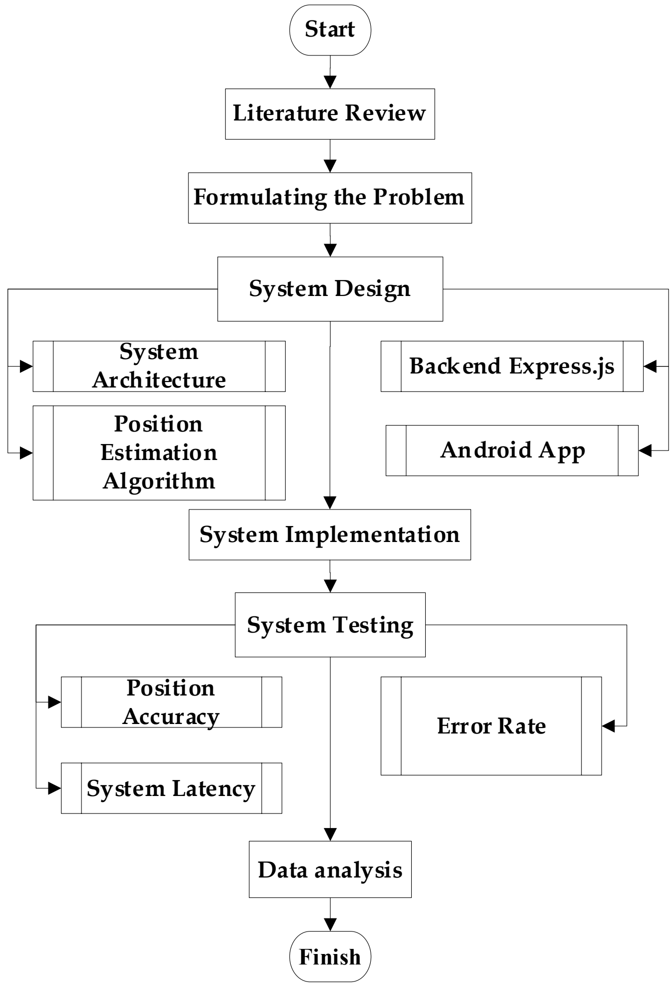

Figure 1. Research Flowchart

The Wi-Fi fingerprint-based indoor localization system depicted in Figure 3 operates through a two-phase workflow. In the initial setup phase (executed once), precomputed statistical parameters—including mean and covariance values for all monitored locations—are retrieved from a MySQL database via an Express.js backend using Retrofit. These parameters serve as spatial priors to inform subsequent Bayesian inference.

During the operational phase (continuously repeated), the system first acquires real-time RSSI measurements from three access points using an Android smartphone. A Multivariate Gaussian Mixture Model (MGMM) then calculates location-specific likelihoods by comparing these live RSSI values against the stored statistical profiles. Bayesian inference integrates the computed likelihoods with spatial priors to derive posterior probabilities for each potential location. The position associated with the highest posterior probability is selected as the final estimated location.

By leveraging historical signal distribution data, this probabilistic framework enhances reliability in GPS-denied environments, addressing Wi-Fi signal variability more effectively than deterministic approaches. The integration of spatial priors and real-time signal analysis ensures robust performance in dynamic indoor settings.

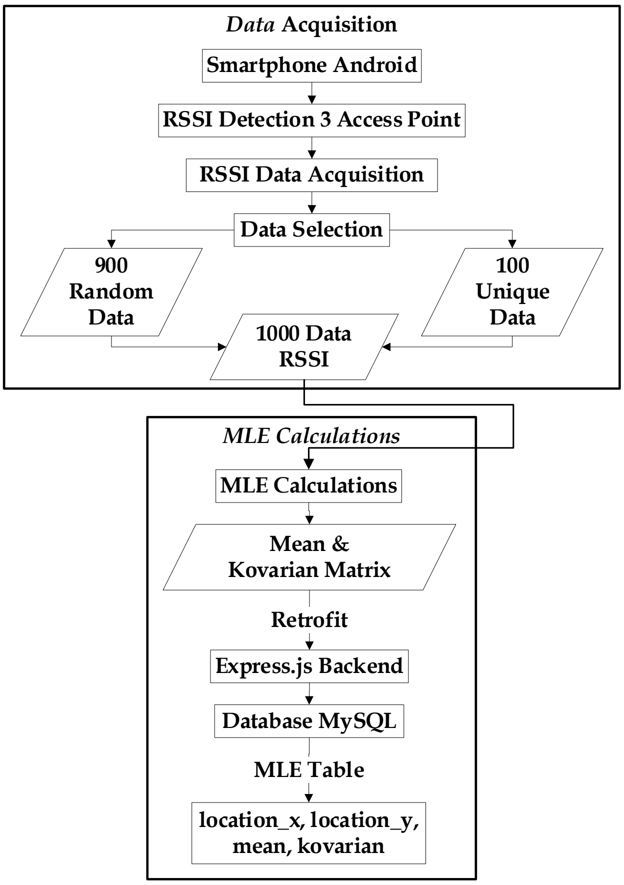

Figure 2. RSSI Data Processing Flowchart

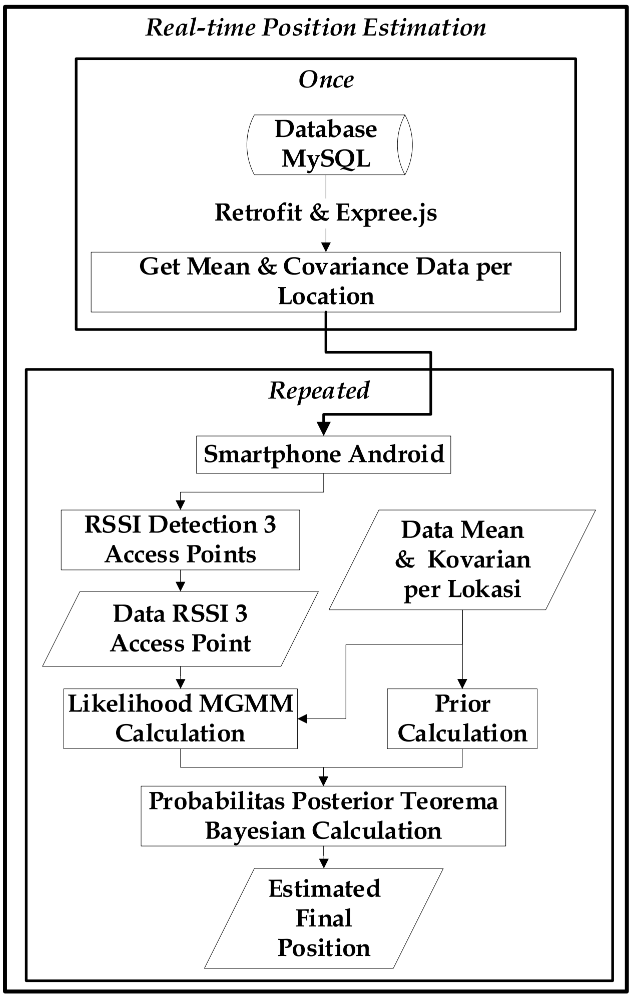

Figure 3. Position Estimation Flowchart

The data collection was conducted in a furnished indoor space measuring approximately 9.06 × 6 meters (Figure 4), containing typical household furniture including tables, wardrobes, beds, and sofas. The room was functionally divided into sleeping areas, living spaces, and a motorcycle parking area, with three access points strategically positioned for comprehensive coverage. While the furniture introduced multipath effects and potential signal attenuation, the environment remained stable with minimal human activity during measurements, ensuring consistent RSSI data collection and adequate signal variation for accurate fingerprint construction.

RSSI measurements were systematically collected using a 1-meter grid spacing across the research area (Figure 5). At each measurement point, an Android smartphone collected 100 unique RSSI readings from the three access points to ensure statistical robustness and reduce environmental noise effects. Data collection incorporated device orientation variations by positioning the smartphone on flat surfaces and holding it statically while standing, ensuring the fingerprint database reflected realistic signal conditions across different usage scenarios. The measurement system relied exclusively on RSSI data without additional sensors, emphasizing Wi-Fi fingerprinting reliability as the sole positioning basis.

The 1-meter grid interval was selected as an optimal compromise between spatial accuracy, data collection efficiency, and computational feasibility. According to Torres-Sospedra & Moreira (2017) [19] grid spacing below 1 meter does not significantly improve positioning accuracy, as the decorrelation distance of indoor RSSI signals is approximately 1 meter. Additionally, considering Indonesian housing standards requiring 7.2 m^2 per person [20], intervals exceeding half the room dimension (~1.8 m) would increase location misidentification risks. Therefore, the 1-meter interval provides optimal spatial resolution while maintaining practical feasibility for smartphone-based indoor positioning systems.

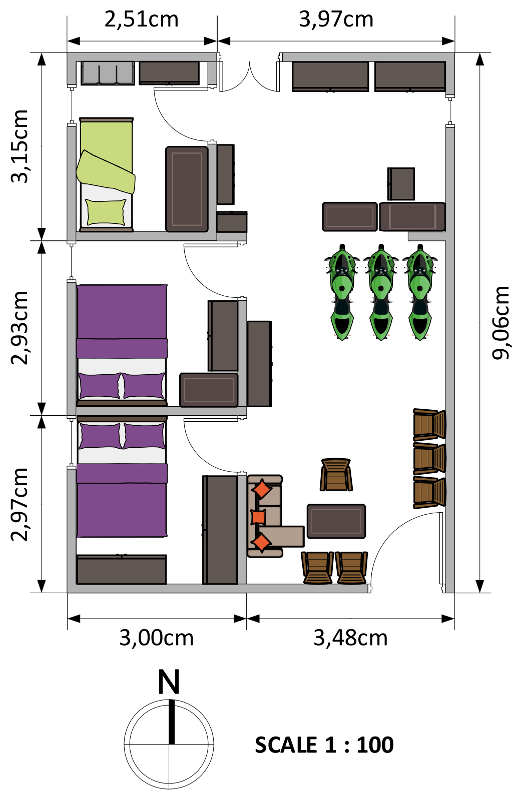

Figure 4. Furnished Space Research Area Plan

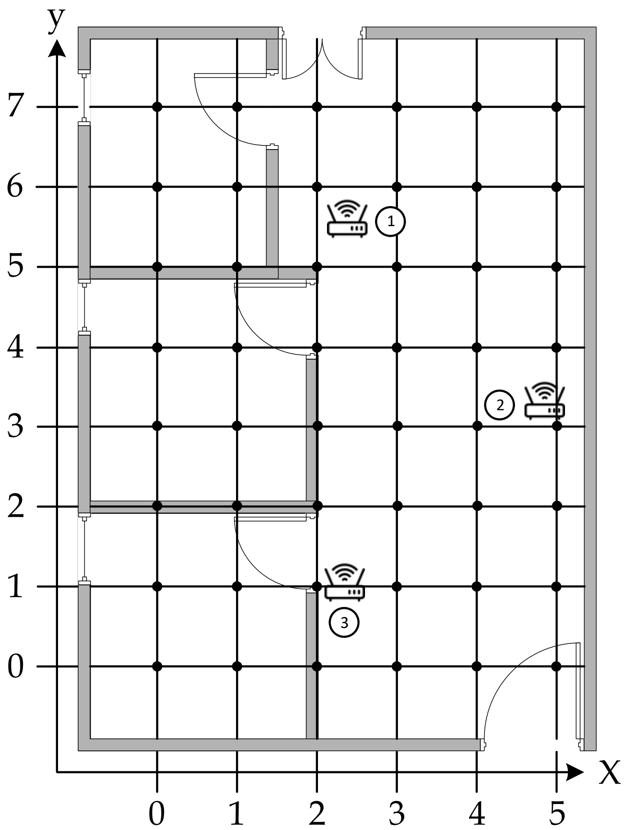

Figure 5. RSSI Measurement Coordinate Point Layout for Furnished Room

The data collection was conducted in an unfurnished indoor space measuring 4.65 × 3.65 meters, as illustrated in Figure 6. The room was divided into a 1-meter grid system along both X and Y axes, creating 12 measurement points marked by black dots. Three access points (AP) were positioned on flat surfaces at approximately 1.2 meters height to ensure optimal signal coverage across the measurement area. Due to the completely empty room without furniture or other objects, signal interference such as multipath effects and attenuation originated solely from wall surfaces, providing a controlled environment for baseline RSSI measurements and minimizing environmental variables that could affect signal propagation patterns.

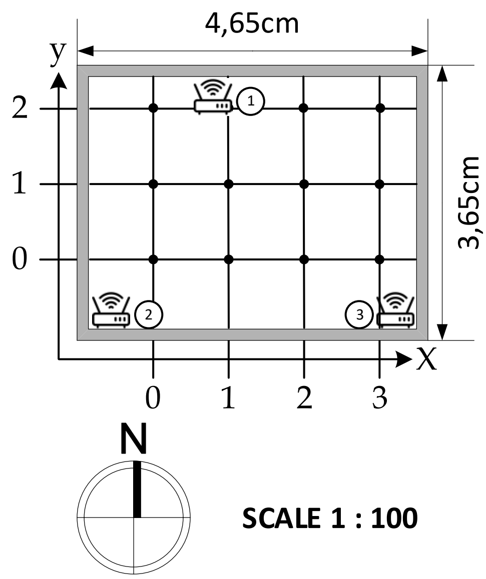

Figure 6. RSSI Measurement Coordinate Point Layout for Unfurnished Room

The experimental setup comprises a Vivo Y95 Android smartphone as the RSSI data collection device and a three-node wireless access point network for indoor positioning measurements. Table 1 details the smartphone specifications, including its Snapdragon 439 processor, 4 GB RAM, and Wi-Fi 802.11 b/g/n connectivity capabilities that enable reliable RSSI signal detection and processing. The wireless infrastructure consists of three access points with specifications outlined in Tables 2, 3, and 4: an EchoLife EG8245H5 GPON terminal (Table 2) and two ZTE ZXHN F609 GPON terminals (Tables 3 and 4), all operating on IEEE 802.11 b/g/n standards at 2.4 GHz with 2×2 MIMO configuration and supporting transmission speeds up to 300 Mbps.

Table 1. Android Smartphone specifications as a RSSI data collection and positioning tool

Category	Detail
Model & System	Vivo Y95, Android 8.1 (Oreo) with Funtouch OS 4.5 interface
Chipset & CPU	Qualcomm SDM439 Snapdragon 439 (12 nm); Octa-core (4×1.95 GHz Cortex-A53 & 4×1.45 GHz Cortex-A53)
GPU	Adreno 505
Wi-Fi Connectivity	Wi-Fi 802.11 b/g/n; Wi-Fi Direct

Table 2. Access Point Specification 1

Category	Detail
Model & Type	EchoLife EG8245H5 - GPON Optical Network Terminal (ONT) for Huawei's FTTH solution
Wi-Fi Standard & Speed	IEEE 802.11 b/g/n (2.4 GHz) with 2×2 MIMO, up to 300 Mbps
Power Supply	DC 12 V ⎓ 1 A (adapter input 100-240 VAC, 50/60 Hz)

Table 3. Access Point Specification 2

Category	Detail
Model & Tipe	ZTE ZXHN F609 – GPON Optical Network Terminal (ONT)
Standar & Kecepatan Wi-Fi	IEEE 802.11b/g/n (2×2 MIMO) @ 2.4 GHz, hingga 300 Mbps
Daya & Catu Daya	Power: 12 V DC 1.5 A (adapter input 100–240 VAC, 50–60 Hz); Rata-rata konsumsi: ~7–11 W

Table 4. Access Point Specification 3

Category	Detail
Model & Type	ZTE ZXHN F609 V2.0 - GPON Optical Network Terminal (ONT) for FTTH, desktop & wall mounting
Wi-Fi Standard & Speed	IEEE 802.11b/g/n (2×2 MIMO @ 2.4 GHz), two 5 dBi external antennas
Power Supply	12 V DC 1.0 A via adapter (100–240 VAC, 50/60 Hz)

Android Application Design

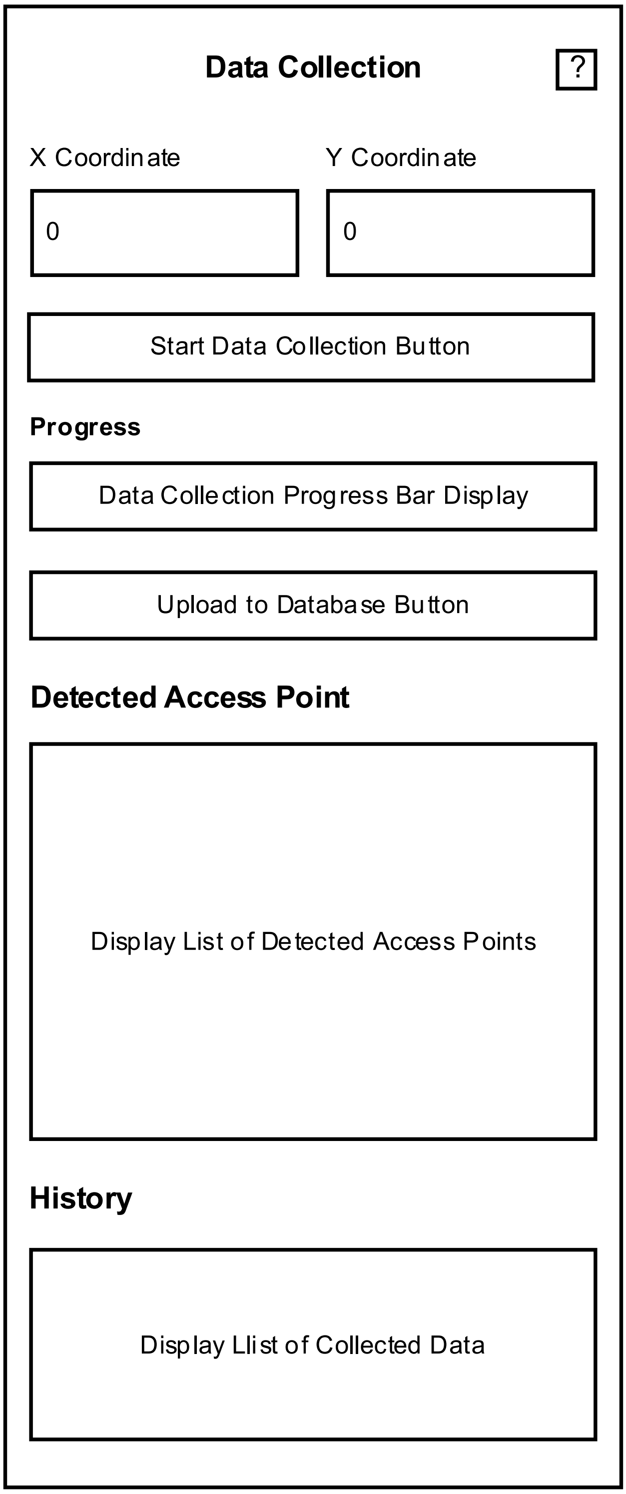

Figure 7. Data Collection Interface Design

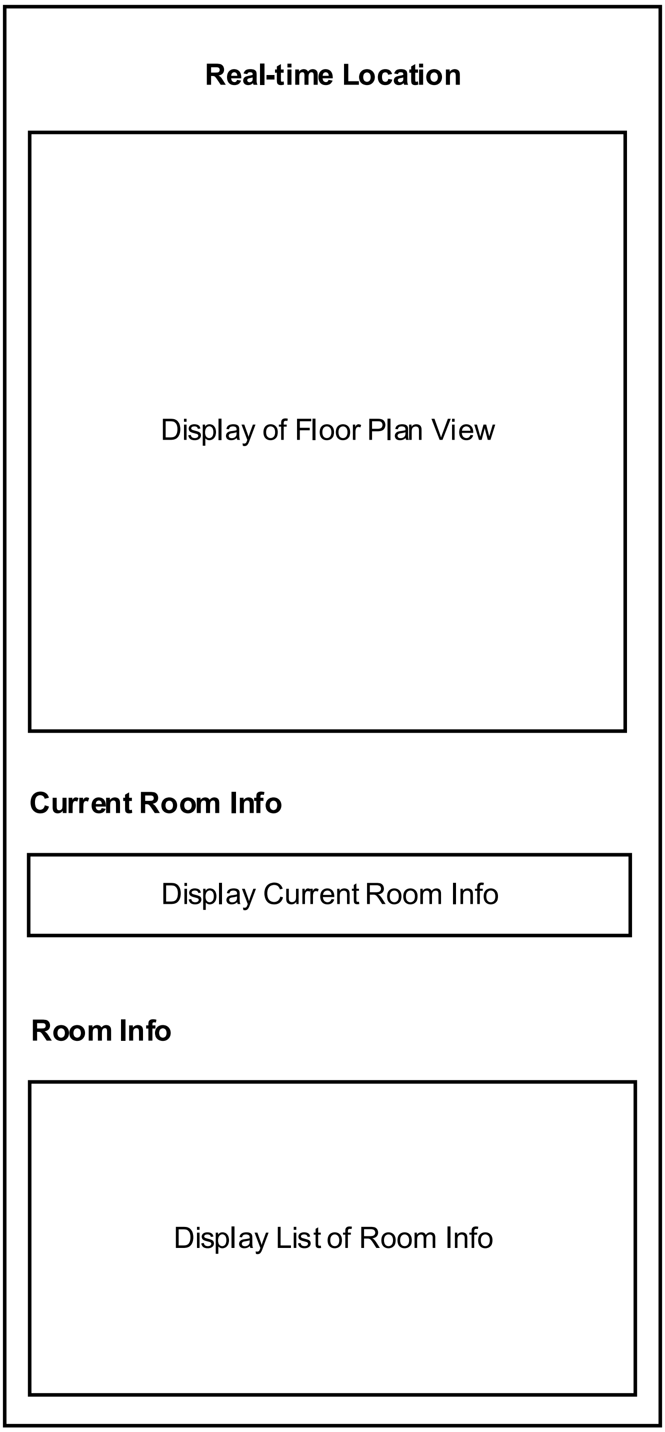

Figure 8. Real-time Location Interface Design

The indoor positioning system implements two distinct Android applications with specific purposes and target users. The Data Collection Application (Figure 8) is designed for developers to collect and map location data through a one-time setup process. This application features coordinate input fields for manual X and Y coordinate entry with validation to prevent duplicate entries, a "Start Data Collection" button to initiate the data gathering process, a progress bar for real-time monitoring, and an "Upload to Database" button for data storage. The interface also displays detected access points with SSID, MAC address, and RSSI values, along with a history section showing previously collected coordinate data.

The Real-time Location Application (Figure 9) serves end-users by utilizing the developer-collected data to display current indoor positioning on a floor plan map. The interface consists of three main components: a floor plan display providing spatial context of the building layout, current room information showing the user's present location with room name and description, and a comprehensive room list displaying all available rooms with their respective descriptions. Room data and descriptions are hardcoded into the application, requiring code modification and recompilation for any changes to floor plans or room information. This dual-application approach provides an efficient solution where the data collection app enables accurate location database construction while the real-time app delivers accessible positioning information to end-users.

Express.js Backend Design

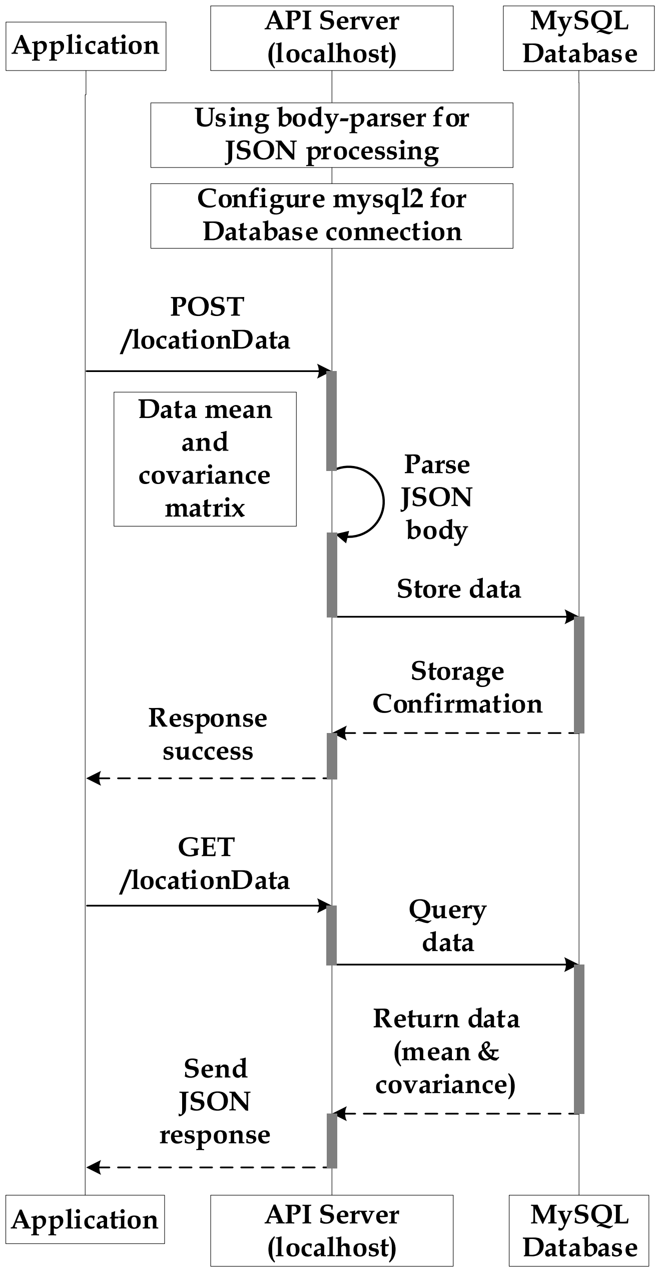

Figure 9. API Service for Location Data Management Flowchart

The system implements a RESTful API service for managing location data through two primary operations, as illustrated in Figure 7. The POST request workflow enables data storage by sending mean values and covariance matrices in JSON format to the /locationData endpoint on the localhost server. The API server utilizes body-parser middleware for JSON processing and the mysql2 library to store data in the MySQL database, returning a success confirmation to the client application upon completion.

The GET request workflow facilitates data retrieval when the application requests location data from the same endpoint. The API server queries the database using mysql2, retrieves the stored mean and covariance matrix data, converts it to JSON format, and sends the response back to the requesting application. This bidirectional communication system ensures efficient data management for the fingerprinting database, with all operations running on localhost to maintain system performance and data integrity during the positioning process.

Position Estimation Algorithm Design

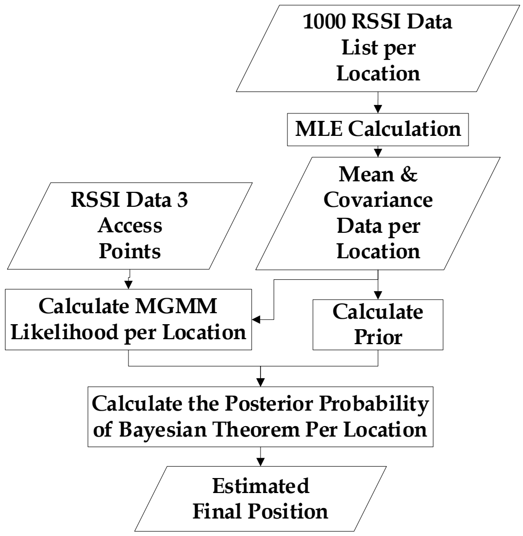

Figure 10. Algorithm for Position Estimation Flowchart

The position estimation algorithm integrates three sequential stages: Maximum Likelihood Estimation (MLE), Multivariate Gaussian Mixture Model (MGMM), and Bayes' theorem, as illustrated in Figure 10. The MLE stage processes 1000 RSSI samples from three access points at each location coordinate (x,y) to calculate statistical parameters. For each location, the mean values are computed as:

$$\mu_1 = \frac{1}{n}\sum_{i=1}^{n} r_i 1, \quad \mu_2 = \frac{1}{n}\sum_{i=1}^{n} r_i 2, \quad \mu_3 = \frac{1}{n}\sum_{i=1}^{n} r_i 3 \qquad (1)$$

where r_{i1}, r_{i2}, r_{i3} represent RSSI values from the three APs for sample I, while n is the number of samples (1000 in this case). The covariance matrix computed as:

$$\Sigma = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)(x_i - \mu)^T \qquad (2)$$

captures the statistical relationships between RSSI signals from different access points, forming a 3×3 matrix containing variances and covariances. Where x_i is the i-th data vector out of a total of n samples

The MGMM stage utilizes MLE parameters to model the multivariate Gaussian distribution for each location. The likelihood function

$$p(\boldsymbol{R}|L_k) = \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{R} - \mu_k)^T \Sigma_k^{-1}(\boldsymbol{R} - \mu_k)\right) \qquad (3)$$

calculates the probability of observing RSSI vector R at location L_k, where d represents data dimensionality (3 APs), and μ_k, Σ_k are location-specific parameters. This likelihood quantifies how well new RSSI measurements match the statistical characteristics of each reference location.

The final stage applies Bayes' theorem to determine the most probable location by combining MGMM likelihood with prior probabilities. The posterior probability

$$P(L_k|R) = \frac{P(R|L_k) \cdot P(L_k)}{P(R)} \qquad (4)$$

integrates the likelihood $P(R|L_k)$ from MGMM with prior probability $P(L_k)$, calculated as the proportion of historical data at location L_k or assumed uniform if no information exists. The denominator $P(R) = \sum_{i=1}^{N} P(R|L_i) \cdot P(L_i)$ normalizes the posterior probabilities across all locations. The location with the highest posterior probability is selected as the final position estimate, providing a probabilistically robust indoor positioning solution.

System Implementation

The indoor positioning system implementation was conducted following the completion of architecture, application, and algorithm design phases. The system was developed as a functional prototype encompassing three primary components: Android application development, backend infrastructure, and database configuration.

The Android application was developed using Java programming language to handle user interface interactions and RSSI data processing. An Express.js backend server was implemented to manage data communication between the mobile application and database. MySQL database was configured for storing RSSI fingerprint data and Maximum Likelihood Estimation (MLE) calculation results.

The implementation process involved strategic placement of three WiFi access points according to the predetermined floor plan design. The system was deployed in a real-world environment that accurately represents actual usage conditions, incorporating considerations for physical obstacles and signal strength variations that typically affect indoor wireless propagation.

System Testing

To ensure the Android indoor positioning application using Multivariate Gaussian Mixture Model (MGMM) algorithm functions effectively, comprehensive testing was conducted to evaluate accuracy, error rates, and system latency. Testing was performed in a pre-mapped indoor environment using a grid-based approach with 1-meter intervals, where each test point had predetermined (x, y) coordinates for position estimation based on RSSI data from three access points.

- Position Estimation Accuracy Testing

Accuracy assessment measured the system's ability to estimate user positions within a tolerance radius of ≤1 meter. Euclidean distance was calculated using:

$$d = \sqrt{(x_e - x_r)^2 + (y_e - y_r)^2} \quad (5)$$

where (x_e, y_e) represents estimated coordinates, (x_r, y_r) represents actual coordinates, and d represents distance between estimated position and actual position. Accuracy percentage was calculated as:

$$Accuracy(\%) = \left(\frac{N_{correct}}{N_{total}}\right) \times 100 \quad (6)$$

where $N_{correct}$ represents the number of estimates that are within the tolerance radius (≤1 meter) and N_{total} represents total number of tests performed.

- Error Rate Testing

Two primary error metrics were employed to evaluate error rate. The Mean Absolute Error (MAE) measures the average absolute deviation between estimated positions and ground-truth values, calculated as:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}(|x_e - x_r| + |y_e - y_r|) \quad (7)$$

where (x_e, y_e) represents coordinates of the estimated user position on the i-th test, (x_r, y_r) represents actual coordinates, and n represents total number of trials. Complementing this, the Root Mean Square Error (RMSE) accounts for error variability by assigning greater weight to extreme deviations, making it sensitive to significant outliers. This metric, computed using:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}((x_e - x_r)^2 + (y_e - y_r)^2)} \quad (8)$$

- Latency Testing

System latency was measured across three stages: RSSI data acquisition ($T_{collect}$), processing time using MGMM ($T_{process}$), and result display ($T_{display}$). Total latency was calculated as:

$$T_{total} = T_{collect} + T_{process} + T_{display} \quad (9)$$

The average latency was calculated with the formula:

$$T_{average} = \frac{T_{total1} + T_{total2} + \ldots + T_{totaln}}{Total\ testing\ (n_{total})} \quad (10)$$

Testing the position estimation accuracy and error rate is done 30 times at three test points in the room to ensure statistical significance, with all measurements performed under consistent environmental conditions. In addition, the delay time (Latency) test will also be conducted 30 times.

Data Analysis

Data analysis is carried out to evaluate the results of system testing in determining the position of users in the room using the Multivariate Gaussian Mixture Model (MGMM) algorithm based on Wi-Fi RSSI-Fingerprint. The data obtained from the tests will be analyzed to determine the level of accuracy, error rate, and latency. The results of this analysis will be compared with predetermined success standards to determine the extent to which the system can be effectively used in indoor navigation.

- Accuracy Classification:
  - ≥80%: Sufficient for indoor navigation
  - 60-79%: Usable but requires algorithm optimization
  - <60%: Inadequate, requiring system improvement
- Error Categories:
  - Very Accurate: MAE ≤0.3m, RMSE ≤0.5m
  - Accurate: MAE 0.3-0.7m, RMSE 0.5-1.0m
  - Moderately Accurate: MAE 0.7-1.5m, RMSE 1.0-2.0m
  - Poor Accuracy: MAE >3.0m, RMSE >4.0m
- Latency Performance:
  - Very Fast: ≤0.5 seconds
  - Fast: 0.5-1.0 seconds
  - Slow: ≥1.0 seconds

III. RESULT AND DISCUSSION

The signal strength heatmaps visualize RSSI distribution from each access point using color gradations in dBm units. Figure 11 shows the Ekahau Best Practices color scale ranging from -85 dBm (gray, very weak) to -30 dBm (bright green, very strong). This visualization identifies signal strength zones and potential blind spots for fingerprinting analysis.

Figures 12-14 display RSSI distributions for each AP in the furnished room, while Figures 15-17 show distributions in the unfurnished room. The heatmaps were generated using Ekahau AI Pro version 11.1.4 software.

RSSI values are primarily influenced by distance between the smartphone and access points, with signal attenuation occurring due to free space path loss. Physical obstacles (walls, furniture, construction materials) create shadowing effects and multipath fading. Additional factors include antenna radiation patterns, interference from neighboring devices, and smartphone orientation during measurements.

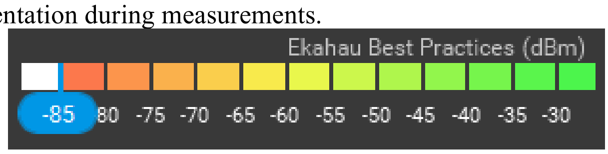

Figure 11. RSSI signal strength color scale based on Ekahau Best Practices (dBm)

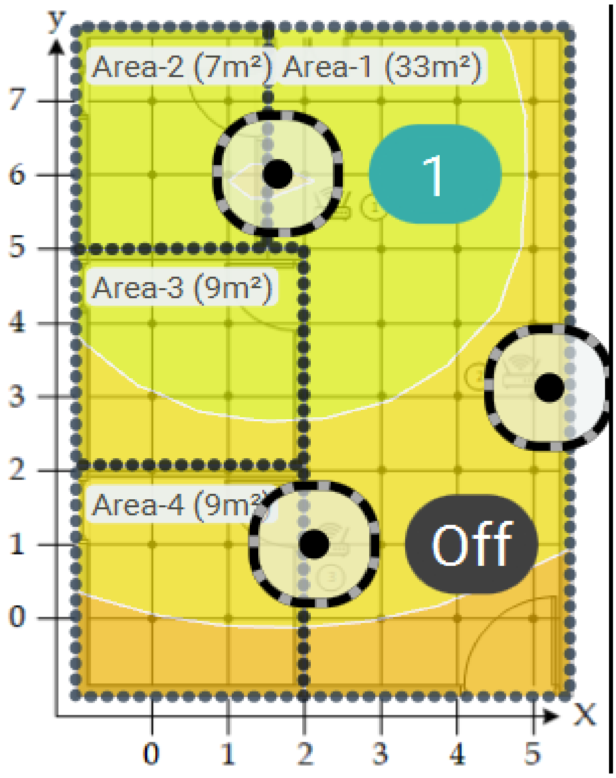

Figure 12. Access Point 1 furnished room RSSI distribution heatmap

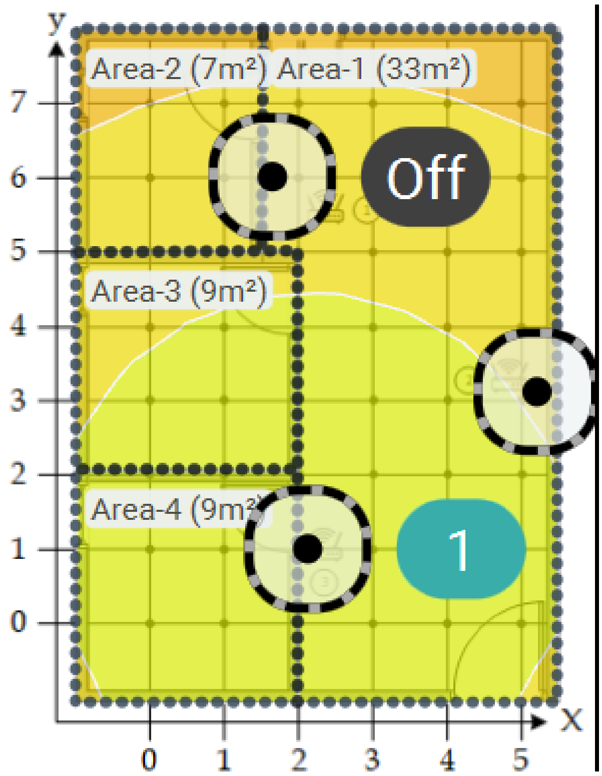

Figure 14. Access Point 3 furnished room RSSI distribution heatmap

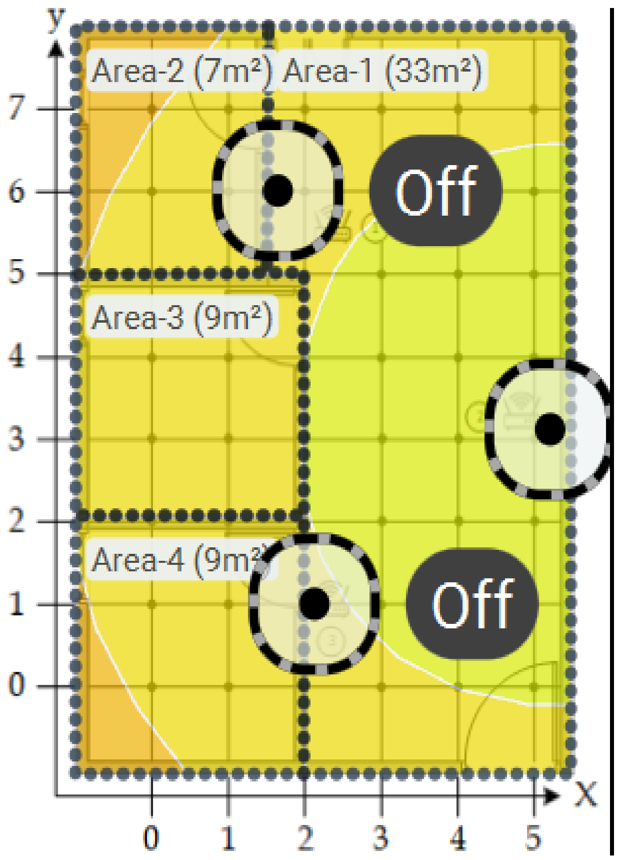

Figure 13. Access Point 2 furnished room RSSI distribution heatmap

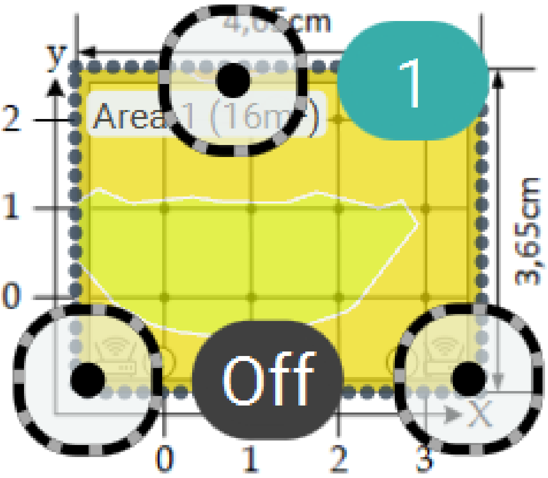

Figure 15. Access Point 1 unfurnished room RSSI distribution heatmap

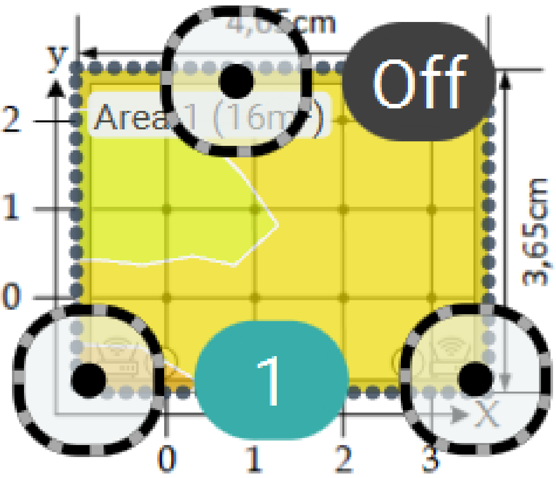

Figure 16. Access Point 2 unfurnished room RSSI distribution heatmap

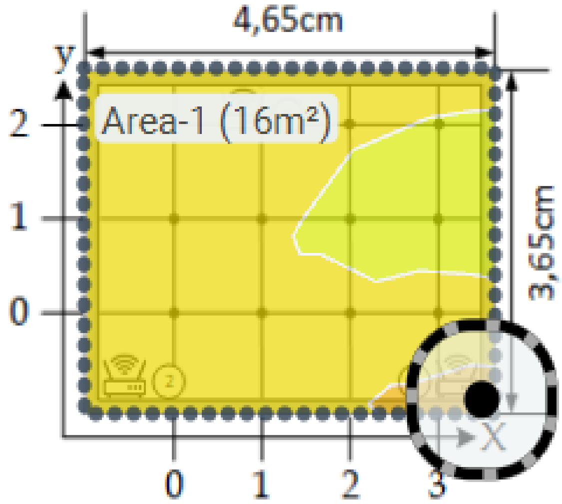

Figure 17. Access Point 3 unfurnished room RSSI distribution heatmap

The Android application prototype consists of two distinct components: the Data Collection App and Real-Time Location App, as illustrated in Figures 18 and 19. Both applications were developed according to the interface design specifications outlined in Figures 8 and 9.

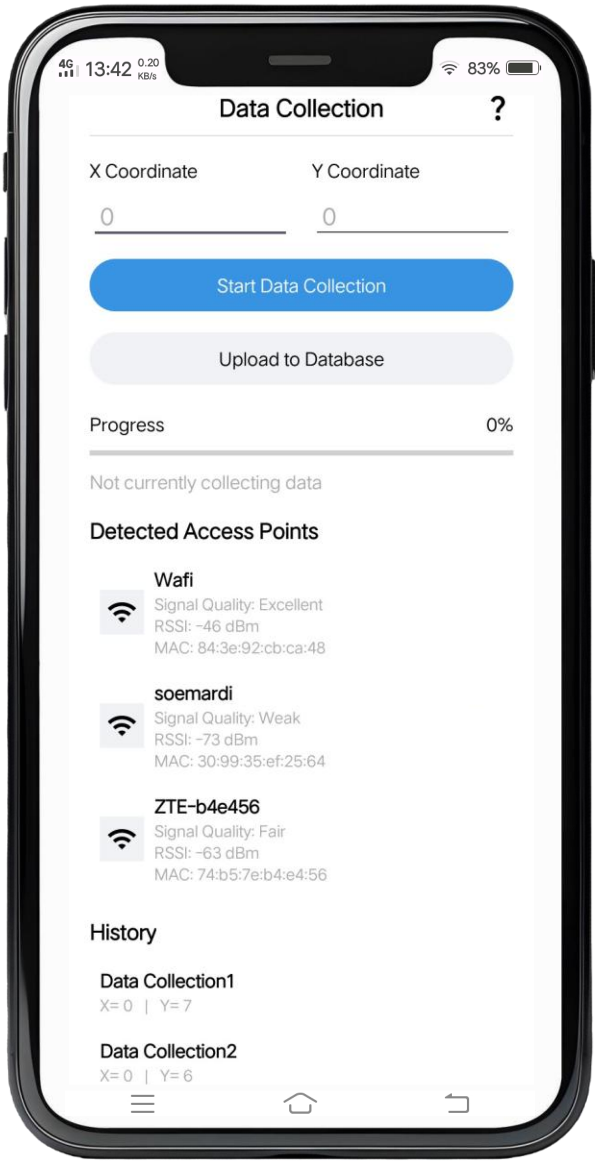

Figure 18. Data Collection App

The Data Collection App (Figure 18) serves as a specialized tool for gathering RSSI data from three access points at each grid point within the research location floor plan. The application interface displays detected access points with their corresponding SSID, BSSID (MAC Address), and signal strength (RSSI) values. The data collection process requires users to position themselves at specific grid coordinates and input the corresponding x and y coordinates into the application.

Upon initiating data collection via the "Start Data Collection" button, the application monitors progress through a visual progress bar. The system incorporates validation mechanisms to prevent duplicate data collection, displaying a "Coordinate data has been taken" warning for previously sampled locations. The History section maintains a record of completed coordinate measurements.

During active data collection, the application status changes to "Data Collection in Progress," temporarily disabling the "Upload to Database" function. Upon completion (100% progress), the status updates to "Finish Collect Data." At this stage, the application executes Maximum Likelihood Estimation (MLE) algorithms in the background to calculate mean values and covariance matrices from the collected data. Subsequently, the "Upload to Database" button becomes active, enabling users to upload the processed MLE-calculated parameters to the MySQL database. A brief instructional guide is accessible through the question mark icon.

During the operational phase, the Android application collects real-time RSSI measurements simultaneously from all three access points using the Android OS's built-in Wi-Fi scanning capabilities. The process leverages the standard WifiManager.getScanResults() API, which returns a list of all detectable Wi-Fi networks within range during a single scan cycle. The implementation follows these steps:

1. Scan Trigger: The application initiates a Wi-Fi scan via the Android WifiManager when position estimation is requested.
2. Simultaneous Data Capture: The scan detects signals from all nearby APs concurrently. This ensures RSSI values from different APs are captured at nearly the same instant, minimizing temporal discrepancies caused by signal fluctuations.
3. AP Filtering: The raw scan results are filtered using the pre-configured MAC addresses (BSSIDs) of the three target access points. Only RSSI values from these specific APs are extracted for processing.
4. Vector Formation: The filtered RSSI values (in dBm) from the three APs are combined into a single 3-dimensional vector: $[RSSI\_AP1, RSSI\_AP2, RSSI\_AP3]^T$
5. Latency Optimization: The entire scan-to-vector process occurs in the $T_{collect}$ phase (Table 7), with an average duration of 11–25 ms in typical conditions. This efficiency enables real-time operation without specialized hardware.

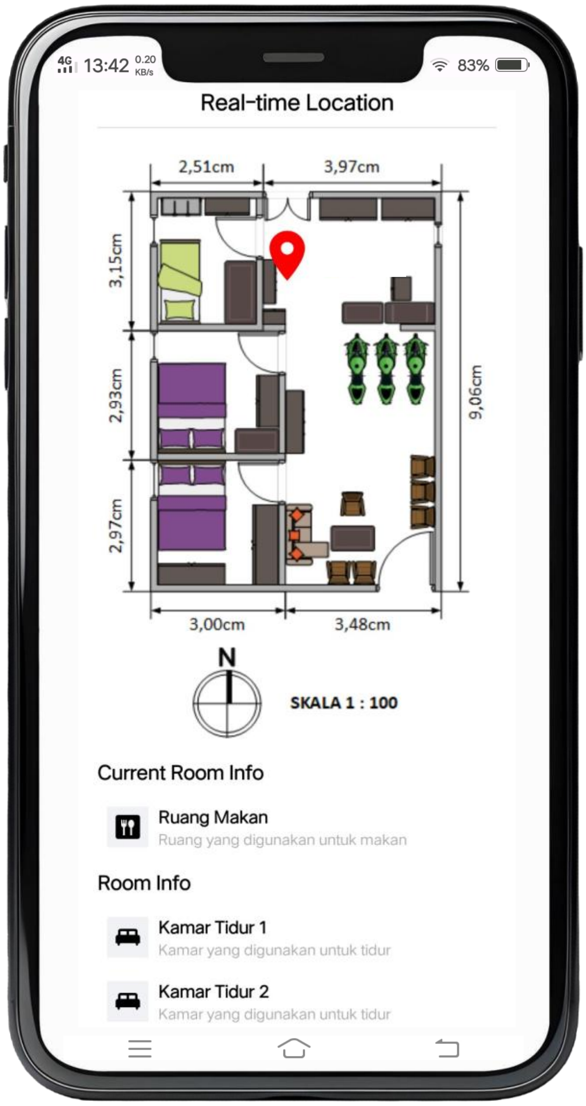

Figure 19. Real-time Location App Furnished Room

The Real-Time Location App (Figure 19) provides real-time user positioning based on previously collected coordinate data. The main interface displays a location map with a red marker indicating the user's current position. Position calculations are performed automatically in the background whenever the application detects RSSI signals from the three access points.

Below the map display, the "Current Room Info" section provides information about the user's present location, including room descriptions. The "Room Info" section lists additional rooms within the mapped area with corresponding descriptions, enabling users to monitor their position accurately and access comprehensive room information.

Figure 20 demonstrates the Real-Time Location App variant for unfurnished rooms, maintaining identical functionality to the furnished room version (Figure 19). The primary difference lies in the floor plan layout, which has been adapted to reflect the empty room conditions used in this research. This version omits the "Current Room Info" feature for displaying room names or descriptions and excludes the comprehensive room listing, resulting in a simplified interface focused solely on user position mapping within the empty floor plan.

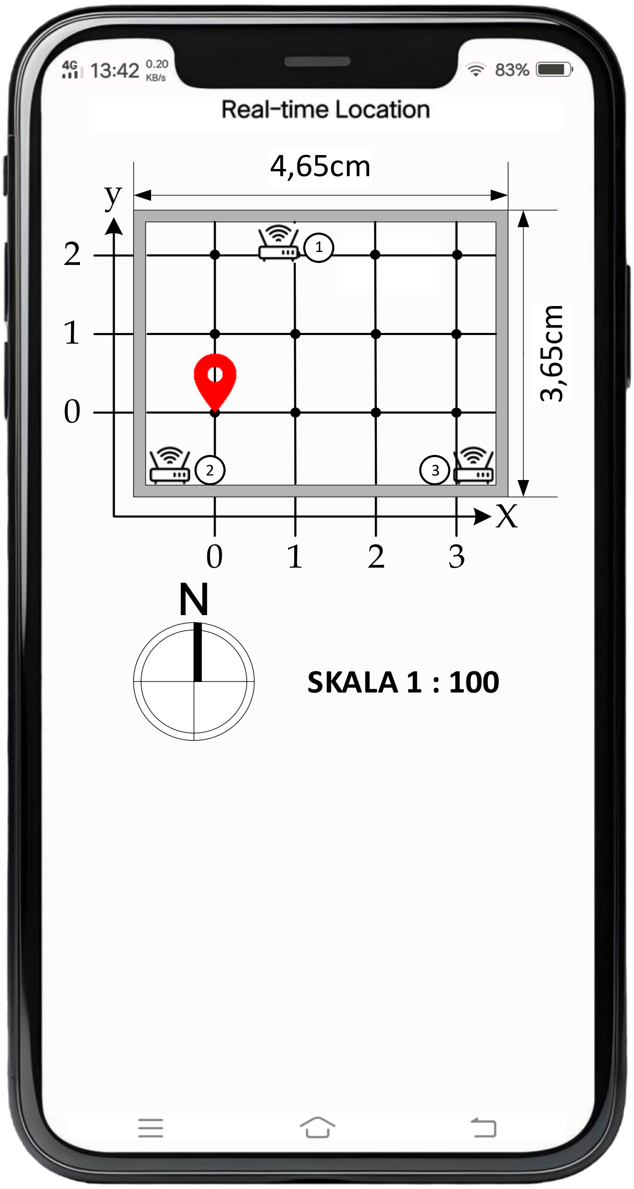

Figure 20. Real-time Location App Unfurnished Room

Figure 20 demonstrates the Real-Time Location App variant for unfurnished rooms, maintaining identical functionality to the furnished room version (Figure 19). The primary difference lies in the floor plan layout, which has been adapted to reflect the empty room conditions used in this research. This version omits the "Current Room Info" feature for displaying room names or descriptions and excludes the comprehensive room listing, resulting in a simplified interface focused solely on user position mapping within the empty floor plan.

This research developed two distinct Android applications: the Data Collection App (Figure 18) for development team use during initial data gathering phases, and the Real-Time Location App (Figure 19) for end-user implementation. The Data Collection App is specifically designed for developers to capture RSSI values at each coordinate point according to the grid sampling pattern. As its function is limited to reference data collection, this application is not intended for end-users but serves as a one-time development tool during system preparation.

Conversely, the Real-Time Location App represents the end-user interface for indoor positioning within mapped environments. Users can download and install this application on Android smartphones without requiring technical data input. The application automatically displays real-time positioning based on collected RSSI fingerprints.

Given the prototype stage of development, the floor plan maps and room listings in the Real-Time Location App remain static and cannot be modified or expanded by users. All mapping information is hardcoded within the source code rather than retrieved from servers or external files, limiting user customization capabilities for room addition or editing.

Accuracy and Error Rate Testing Results

The system was tested in a furnished room environment at three coordinate points: (4,1), (3,3), and (2,6), with 10 trials conducted at each position (30 total trials). The accuracy was measured using a tolerance radius of ≤1 meter from the actual position, calculated using Euclidean distance. The detailed furnished room accuracy and error rate testing results are presented in Table 5.

Table 5. Furnished Room Accuracy and Error Rate Testing Results

Trial No.	Actual Position (x,y)	Estimated Position (x, y)	Euclidean Distance (meter)
1	(4,1)	(4,1)	0
2	(4,1)	(4,1)	0
3	(4,1)	(4,1)	0
4	(4,1)	(2,3)	2,8
5	(4,1)	(4,1)	0
6	(4,1)	(4,1)	0
7	(4,1)	(4,1)	0
8	(4,1)	(4,0)	1
9	(4,1)	(4,0)	1
10	(4,1)	(4,0)	1
11	(3,3)	(3,3)	0
12	(3,3)	(3,3)	0
13	(3,3)	(3,3)	0
14	(3,3)	(3,3)	0
15	(3,3)	(3,3)	0
16	(3,3)	(3,3)	0
17	(3,3)	(3,3)	0
18	(3,3)	(3,3)	0
19	(3,3)	(3,3)	0
20	(3,3)	(3,3)	0
21	(2,6)	(2,6)	0
22	(2,6)	(2,6)	0
23	(2,6)	(2,6)	0
24	(2,6)	(2,6)	0
25	(2,6)	(2,6)	0
26	(2,6)	(2,6)	0
27	(2,6)	(2,6)	0
28	(2,6)	(2,6)	0
29	(2,6)	(3,5)	1,4
30	(2,6)	(3,5)	1,4

The results demonstrated that the system achieved 90% accuracy, with 27 out of 30 position estimates falling within the acceptable tolerance range. The Mean Absolute Error (MAE) was 0.433 meters, while the Root Mean Square Error (RMSE) was 0.796 meters. These values indicate low error rates suitable for indoor navigation applications.

The lower MAE compared to RMSE suggests that extreme errors (such as the 2.8-meter deviation in trial 4) occur infrequently, indicating stable system performance. The occasional larger errors are likely attributed to RSSI signal fluctuations caused by physical obstacles and limitations in the Gaussian distribution assumptions of the MGMM algorithm.

The 90% accuracy achieved in furnished environments meets the acceptable threshold for indoor navigation systems (≥80%). The error metrics (MAE: 0.433m, RMSE: 0.796m) fall within the "low error" category, confirming the system's suitability for indoor positioning applications.

To establish baseline performance, identical testing was conducted in an empty room without furniture at coordinates (0,0), (2,2), and (3,0). The detailed unfurnished room accuracy and error rate testing results are presented in Table 6.

Table 6. Unfurnished Room Accuracy and Error Rate Testing Results

Trial No.	Actual Position (x,y)	Estimated Position (x, y)	Euclidean Distance (meter)
1	(0,0)	(0,0)	0
2	(0,0)	(0,0)	0
3	(0,0)	(0,0)	0
4	(0,0)	(0,0)	0
5	(0,0)	(0,0)	0
6	(0,0)	(0,0)	0
7	(0,0)	(0,0)	0
8	(0,0)	(0,0)	0
9	(0,0)	(0,0)	0
10	(0,0)	(0,0)	0
11	(2,2)	(2,2)	0
12	(2,2)	(2,2)	0
13	(2,2)	(2,2)	0
14	(2,2)	(2,2)	0
15	(2,2)	(2,2)	0
16	(2,2)	(2,2)	0
17	(2,2)	(2,2)	0
18	(2,2)	(2,2)	0
19	(2,2)	(2,2)	0
20	(2,2)	(2,2)	0
21	(3,0)	(3,0)	0
22	(3,0)	(3,0)	0
23	(3,0)	(3,0)	0
24	(3,0)	(3,0)	0
25	(3,0)	(3,0)	0
26	(3,0)	(3,0)	0
27	(3,0)	(3,0)	0
28	(3,0)	(3,0)	0
29	(3,0)	(3,0)	0
30	(3,0)	(3,0)	0

The system achieved 100% accuracy with all Euclidean distances measuring 0 meters. Both MAE and RMSE values were 0 meters, indicating perfect position estimation in obstacle-free environments.

This ideal performance demonstrates the MGMM model's capability under optimal conditions without physical obstructions or multipath fading effects. However, this represents a baseline scenario, as real-world applications must account for furniture and interference factors.

The stark contrast between furnished room (90% accuracy) and empty room (100% accuracy) performance highlights the significant impact of environmental factors on positioning accuracy. Physical obstacles, furniture, and multipath propagation contribute to signal degradation and positioning errors.

Latency Testing Results

System responsiveness was evaluated by measuring three key processing stages: RSSI data acquisition time ($T_{collect}$), processing time ($T_{process}$), and result display time ($T_{display}$). The total latency (T_{total}) was calculated as the sum of these components. The detailed latency testing results are presented in Table 7.

Table 7. Latency Testing Results

Trial No.	$T_{collect}$ (ms)	$T_{process}$ (ms)	$T_{display}$ (ms)	T_{total} (ms)
1	11	25	4	40
2	20	13	2	35
3	25	4	2	31
4	14	3	2	19
5	15	3	2	20
6	19	2	2	23
7	19	2	2	23
8	16	3	2	21
9	1189	2	2	1193
10	14	1	3	18
11	21	2	2	25
12	19	2	2	23
13	22	2	2	26
14	17	1	2	20
15	15	2	1	18
16	18	1	3	22
17	19	2	2	23
18	17	1	2	20
19	17	2	2	21
20	22	1	2	25
21	15	1	2	18
22	16	1	2	19
23	15	2	2	19
24	21	2	3	26
25	15	2	2	19
26	19	1	3	23
27	24	2	2	28
28	23	2	4	29
29	16	1	2	19
30	15	1	2	18
$T_{average}$ (ms)				62

Across 30 trials, the system achieved an average latency of 62 ms, with the lowest recorded latency of 18 ms and highest of 1,193 ms (trial 9). The majority of trials demonstrated consistent performance below 30 ms, meeting real-time navigation requirements (≤500 ms threshold).

The anomalous high latency in trial 9 (1,193 ms) was primarily due to the data acquisition phase (1,189 ms), likely caused by Android background processes. This suggests the need for multithreading implementation to ensure consistent performance.

The average 62 ms latency demonstrates excellent real-time performance, well within acceptable limits for indoor navigation applications. This rapid response time enables smooth user experience and practical deployment feasibility.

IV. CONCLUSION

The conclusions of this study are as follows:

- The system demonstrated 90% accuracy in furnished environments and 100% accuracy in obstacle-free conditions within a ≤1-meter tolerance radius. The Mean Absolute Error (MAE) was 0.433 meters for furnished rooms and 0 meters for unfurnished rooms, while Root Mean Square Error (RMSE) measured 0.796 meters and 0 meters respectively. These results confirm the MGMM algorithm's effectiveness in modeling RSSI signal distributions for indoor positioning.
- The system exhibited excellent responsiveness with an average latency of 62 ms (range: 18-1,193 ms), well below the 1-second threshold required for real-time navigation applications.
- The integration of MySQL database, Retrofit, and Express.js as backend infrastructure proved efficient for storing and managing RSSI fingerprint data, enabling responsive and stable Android application operation.

For practical deployment, several enhancements are recommended:

- Increase the number of access points to improve signal variation and reduce blind spots, particularly in areas with physical obstructions. Position access points at optimal heights (≥190 cm) with appropriate antenna orientations.
- Implement periodic recalibration at positions with high estimation errors and increase RSSI sample size during training phases to strengthen model accuracy.
- Integrate inertial sensors (gyroscope, accelerometer) to complement RSSI data and reduce dependency on static environmental conditions.

Here are also some suggestions for future research:

- Implement automatic fingerprint database updates to create adaptive systems that respond to environmental changes.
- Explore Ultra-Wideband (UWB) technology integration for enhanced positioning precision and investigate hybrid approaches combining MGMM with deep learning algorithms (e.g., CNN) for dynamic environments.
- Implement multithreading in Android applications to reduce latency in data acquisition and processing phases.

ACKNOWLEDGMENT

The author would like to express his gratitude to Dr. Ir. Lusia Rakhmawati, S.T., M.T. as the research supervisor for all the guidance and motivation so that the author could complete this research. Thanks are also extended to Dr. Farid Baskoro, S.T., M.T. and Miftahur Rohman, S.T., M.T. for all the suggestions given to make this research better. Additionally, the author would like to thank Syifa Kamilah for their continuous support and encouragement throughout this journey.

REFERENCES

[1] Alarifi, A., Al-Salman, A., Alsaleh, M., Alnafessah, A., Al-Hadhrami, S., Al-Ammar, M. A., & Al-Khalifa, H. S. (2016). Ultra wideband indoor positioning technologies: Analysis and recent advances. In Sensors (Switzerland) (Vol. 16, Issue 5). MDPI AG. https://doi.org/10.3390/s16050707

[2] Xiao, J., Zhou, Z., Yi, Y., & Ni, L. M. (2016). A survey on wireless indoor localization from the device perspective. In ACM Computing Surveys (Vol. 49, Issue 2). Association for Computing Machinery. https://doi.org/10.1145/2933232

[3] Syazwani, C. J. N., Wahab, N. H. A., Sunar, N., Ariffin, S. H. S., Wong, K. Y., & Aun, Y. (2022). Indoor Positioning System: A Review. International Journal of Advanced Computer Science and Applications,

13(6), 477–490. https://doi.org/10.14569/IJACSA.2022.0130659

[4] Huang, H., & Gartner, G. (2018). Current trends and challenges in location-based services. In ISPRS International Journal of Geo-Information (Vol. 7, Issue 6). MDPI AG. https://doi.org/10.3390/ijgi7060199

[5] Li, Y., Barthelemy, J., Sun, S., Perez, P., & Moran, B. (2020). A Case Study of WiFi Sniffing Performance Evaluation. IEEE Access, 8, 129224–129235. https://doi.org/10.1109/ACCESS.2020.3008533

[6] Shafie, M. S., & Nasir Akhyari. (2023). Wi-Fi Received Signal Strength Indicator (RSSI) Factors in Influencing Indoor Signal: A Review. International Journal of Business and Technology Management, 5(2nd International Conference on Business Innovation, Entrepreneurship and Technology (ICOBET) 2023), 385–394. https://doi.org/10.55057/ijbtm.2023.5.s5.42

[7] Kadry, R., & Ismael, O. (2020). A New Hybrid KNN Classification Approach based on Particle Swarm Optimization. International Journal of Advanced Computer Science and Applications, 11(11), 291–296. https://doi.org/10.14569/IJACSA.2020.0111137

[8] Rathnayake, R. M. M. R., Maduranga, M. W. P., Tilwari, V., & Dissanayake, M. B. (2023). RSSI and Machine Learning-Based Indoor Localization Systems for Smart Cities. In Eng (Vol. 4, Issue 2, pp. 1468–1494). Multidisciplinary Digital Publishing Institute (MDPI). https://doi.org/10.3390/eng4020085

[9] Raitoharju, M., García-Fernández, Á. F., Hostettler, R., Piché, R., & Särkkä, S. (2020). Gaussian mixture models for signal mapping and positioning. Signal Processing, 168, 107330. https://doi.org/https://doi.org/10.1016/j.sigpro.2019.107330

[10] Bonakdarpour, M. (2016, January 22). Introduction to EM: Gaussian Mixture Models. Https://Stephens999.Github.Io/FiveMinuteStats/Intro_to_em.Html.

[11] Mulyawan, R. (2025, January 5). Bayes Theorem Adalah: Definisi, Rumus, dan Contohnya! Https://Rifqimulyawan.Com/Kamus/Bayes-Theorem/.

[12] Alfakih, M., Keche, M., & Benoudnine, H. (2015). Gaussian mixture modeling for indoor positioning WIFI systems. 2015 3rd International Conference on Control, Engineering & Information Technology (CEIT), 1–5. https://doi.org/10.1109/CEIT.2015.7233072

[13] Alfakih, M., Keche, M., Hadjira, B., & Abdelkrim, M. (2020). Improved Gaussian mixture modeling for accurate Wi-Fi based indoor localization systems. Physical Communication, 43. https://doi.org/10.1016/j.phycom.2020.101218

[14] Zhu, H., Cheng, L., Li, X., & Yuan, H. (2023). Neural-Network-Based Localization Method for Wi-Fi Fingerprint Indoor Localization. Sensors, 23(15). https://doi.org/10.3390/s23156992

[15] Rizk, H., Elmogy, A., & Yamaguchi, H. (2022). A Robust and Accurate Indoor Localization Using Learning-Based Fusion of Wi-Fi RTT and RSSI. Sensors, 22(7). https://doi.org/10.3390/s22072700

[16] Liu, F., Liu, J., Yin, Y., Wang, W., Hu, D., Chen, P., & Niu, Q. (2020). Survey on WiFi-based indoor positioning techniques. In IET Communications (Vol. 14, Issue 9, pp. 1372–1383). Institution of Engineering and Technology. https://doi.org/10.1049/iet-com.2019.1059

[17] Xia, S., Liu, Y., Yuan, G., Zhu, M., & Wang, Z. (2017). Indoor fingerprint positioning based on Wi-Fi: An overview. In ISPRS International Journal of Geo-Information (Vol. 6, Issue 5). MDPI AG. https://doi.org/10.3390/ijgi6050135

[18] Li, Y., Williams, S., Moran, B., Kealy, A., & Retscher, G. (2018). High-dimensional probabilistic fingerprinting in wireless sensor networks based on a multivariate gaussian mixture model. Sensors (Switzerland), 18(8). https://doi.org/10.3390/s18082602

[19] Torres-Sospedra, J., & Moreira, A. (2017). Analysis of sources of large positioning errors in deterministic fingerprinting. Sensors (Switzerland), 17(12). https://doi.org/10.3390/s17122736

[20] Fadli, A., & Alexander, H. B. (2021). Berapa Luas Ideal Rumah Orang Indonesia? Simak Penjelasannya. Kompas.com. https://www.kompas.com/properti/read/2021/10/22/193000021/berapa-luas-ideal-rumah-orang-indonesia-simak-penjelasannya?page=all