Faktor Exacta

Vol. 19, No. 3, Sept 2026, pp. 164~175
eISSN: 2502-339X, pISSN: 1979-276X, https://doi.org/10.30998/faktorexacta.v19i3.5987

Haversine-Based Multi-Site Visit Validation for Real-Time GPS Tracking

Ardhiat Gilang Pratama[1], Nur Rofiq[2]
[1,2]Department of Informatics Engineering, Faculty of Computer Science, Universitas Pamulang, Indonesia

Article Info

Article history:

Received 20/10/2026
Revised 06/11/2026
Accepted 06/11/2026

Keywords:

GPS tracking
Haversine formula
Geofencing
Visit validation
Location-based
workforce monitoring

ABSTRACT

Field administrators who visit multiple project sites are often supervised through manual reports that cannot verify the actual location of a visit. This study designed and implemented LACAKIN, a web-based, real-time GPS tracking system that validated each visit report against the coordinates and radius of the selected site using the Haversine formula. The system was developed using PHP, MySQL, and Leaflet.js to facilitate monitoring for both field administrators and supervisors. The evaluation combined black-box, white-box, and API testing with a pilot trial consisting of six visit reports from five accounts across four sites, utilizing a 700 m radius and 37 logged GPS points. All 44 black-box and 25 API test cases passed, and the core functions exhibited a mean cyclomatic complexity of 6.4. Independent recalculations reproduced the stored distances within 0.05 m, confirming computational consistency rather than positioning accuracy. Furthermore, two on-site reports (54.75 m and 93.35 m) were accepted, whereas four off-site reports (4.5 to 20.8 km) were rejected; these classification decisions remained unchanged for any configured radius between 93.35 m and 4,500 m. The mean reported GPS accuracy of 115.16 m indicated that setting a radius below approximately 100 m would inadvertently reject genuine visits. These findings suggest that larger field trials are required before the system can be deployed to assess workforce productivity.

This work is licensed under a CC-BY-NC

Corresponding Author:

Ardhiat Gilang Pratama,
Department of Informatics Engineering,
Universitas Pamulang,
Jl. Surya Kencana No. 1, Pamulang, Tangerang Selatan, Banten.
Email: gilangardhiat2@gmail.com

1. INTRODUCTION

Many companies and public agencies employ field administrators whose daily work consists of visiting several project sites to check progress, collect documents, and report conditions to a supervisor. Because this work happens away from the office, supervisors rarely observe it directly. In practice, visits are reported through chat messages, telephone calls, photographs, or written forms that are sent after the visit and then compiled manually. The reported location and time cannot be verified, recapitulation is slow and prone to recording errors, and the supervisor cannot see where staff are while the work is taking place. A report can therefore claim a visit that was never made, and the supervisor has no objective way to tell.

Smartphones make location evidence available at low cost. The Global Positioning System (GPS) and other satellite constellations used by phones provide latitude and longitude that a web or mobile application can send to a server and display on a map in near real time [1]. These positions are not exact, and their error depends strongly on the environment. Near buildings, average smartphone GPS errors of 7 to 13 m have been reported, with occasional errors above 100 m close to large structures [2]. Under open sky, smartphone

receivers can reach horizontal errors of about 1 to 3 m when static, but errors grow to 5 to 19 m when the user is moving [3]. In a harsh urban environment, the 95th-percentile horizontal error of a smartphone was 10 to 15 m depending on the constellation and integrity checks used [4]. The accuracy radius reported by phones also rises sharply indoors, with medians of 16 to 48 m indoors against about 4 m outdoors [5]. Any system that decides from GPS whether a person is at a site must therefore set its acceptance threshold with this uncertainty in mind.

The Haversine formula computes the great-circle distance between two points from their latitudes and longitudes and is the method most often used for this decision in Indonesian attendance systems. Table 1 summarises representative studies in terms of setting, threshold, evaluation metric, and the limitation most relevant to field visits. Three patterns stand out. First, the decision model is the same in almost every study: one fixed workplace, one geofence, and a check-in event that is either accepted or rejected [6]–[11]. Second, thresholds are chosen without a stated procedure. Sidiq *et al.* [7] used 20 m and Painem and Soetanto [8] used 100 m, but neither related the value to measured GPS error at the site, even though the studies cited above show that a 20 m radius is comparable to the typical indoor error of a phone. Third, evaluation focuses on computation or usability rather than on decision quality. Miftahuddin *et al.* [12] found that Haversine gave a distance accuracy of 98.66% and a location-decision accuracy of 90% when compared with Euclidean and Manhattan distances, and Antono and Dwiasnati [6] reported a difference of about 0.0018 m between system and manual calculations. Such comparisons show that the formula is implemented correctly, but they use the same coordinates for both values and say nothing about how often a genuine visit is rejected or a false one accepted. Usability scores, such as the System Usability Scale (SUS) score of 85.83 in [10] or 95.2 in [11], measure acceptance of the interface, not the reliability of the location decision.

Table 1. Critical comparison of previous GPS-based attendance and tracking studies

Study	Setting and method	Threshold	Evaluation metric	Limitation for field visits
Robert [1]	Mobile attendance with GPS tracking, face and fingerprint recognition	Not used	17 functional tests passed	Tracking only; no site-level validation of visits
Miftahuddin et al. [12]	Employee positioning; Euclidean, Manhattan, and Haversine compared	Campus area	Distance accuracy 98.66%; decision accuracy 90%	One campus; compares distance methods, not thresholds
Antono & Dwiasnati [6]	Android attendance with Haversine	Office radius	System vs manual difference about 0.0018 m	Computational check only; single office
Sidiq et al. [7]	Android staff attendance with Haversine	20 m	Accept or reject check-in by radius	Small fixed radius not related to GPS error
Painem & Soetanto [8]	Web service (REST, JWT) with Haversine	100 m	Attendance status and lateness	Single office; one check-in per day
Fauzan et al. [9]	Android attendance for regional government staff	Office radius	Attendance recording	Single office; no positioning error analysis
Anita & Rahmawati [10]	Web attendance with leave and permission requests	Office radius	SUS 85.83	Usability only; no decision-error measure
Che Fauzi et al. [11]	Geofence-based class attendance in higher education	Classroom geofence	SUS 95.2 (50 students); 100% match with verified presence	Fixed rooms; no multi-site movement
This study	Web tracking with per-site Haversine validation of each visit report	Per site (700 m in trial)	Testing; recalculation; radius sensitivity; accept and reject decisions	Small trial; no surveyed ground truth (Section 3.7)

Field administrators do not fit the single-workplace model. They move between several sites in one day, sites differ in size, and the supervisor needs to know whether each reported visit took place at the intended site, not only whether the employee was present somewhere. A second gap concerns the indicators derived from location data. Counts of validated visits, visits per working hour, and travel distance are attractive because they are computed automatically, but they describe activity rather than the quality or outcome of the work. Monitoring of this kind is also sensitive, because employees' acceptance of surveillance depends strongly on its purpose and context [13]. Travel distance is a further problem, because straight-line distance underestimates road distance. Road distance is usually approximated with a detour (circuity) factor, but this factor varies with network structure, travel mode, and trip length, and a constant factor is a simplification [14]–[16].

This study designs and implements LACAKIN, a web-based real-time tracking system for field administrators. Its building blocks, namely GPS positions, the Haversine formula, geofences, and photographs, are established. The contribution lies in how they are combined and evaluated. The contributions are: (1) a visit-level validation model in which every report is checked against the coordinate and individual radius of the site selected for that visit, so that one session can contain several sites with different tolerances; (2) an

evaluation that separates the computational consistency of the distance calculation from the quality of the accept and reject decisions, together with a radius sensitivity analysis that relates the threshold to observed GPS accuracy; and (3) a supervisor view that attaches distance, validity, reported accuracy, and photographic evidence to each report and derives activity indicators whose limitations are stated explicitly. The study does not claim to measure productivity. The indicators are treated as supervisory aids, and the trial reported here is a feasibility evaluation.

2. METHOD

This research followed a software engineering approach in which a working system was designed, built, and tested. Development used the waterfall model, whose phases run in sequence: requirements gathering, analysis, design, coding, testing, and maintenance [17]. The model was chosen because the requirements could be stated completely after analyzing the existing reporting process, and the scope was fixed at two user roles. The subsections below describe the analysis of the existing process, the system design, the distance and validation model, the activity indicators, the database, and the evaluation procedure, including the trial setup.

2.1. Analysis of the Existing Process

Observation of the existing process showed that field administrators reported their visits after the fact through a messaging application, typically as a short message accompanied by a photograph, and the supervisor compiled these messages manually. Six problems were identified: (1) the position of field administrators could not be monitored while visits were in progress; (2) there was no check that an administrator was at the intended project site; (3) visit data were recorded and recapitulated manually, which caused recording errors; (4) supervisors had difficulty monitoring several administrators at once; (5) there was no information on the distance between an administrator and the site that could serve as objective evidence; and (6) visit reports were not linked to location data. These problems defined the functional requirements of the system.

2.2. System Design

LACAKIN was designed as a web application accessed through the browser of the field administrator's smartphone and through the supervisor's desktop browser. The server side was implemented in PHP with a MySQL database on an Apache web server, and maps were rendered with the Leaflet library on OpenStreetMap tiles. Positions were obtained from the browser's geolocation service, which returns latitude, longitude, an accuracy radius, and speed. The accuracy radius is the device's own estimate of its error and is stored with every point so that it can be used when a report is reviewed.

The system has two actors and twelve use cases. The field administrator can log in, start and stop tracking, view their own position, submit visit reports, and browse their history; the supervisor can monitor positions, review, filter, and export reports, and view activity indicators and GPS accuracy statistics. Figure 1 shows the workflow across the field administrator, the system, and the supervisor. After the administrator accepts a consent dialog that explains why location access is needed and starts a tracking session, the page watches the device position and sends the latest coordinates to the server every 30 seconds without reloading. The server stores each point and updates the active session. The supervisor's monitoring page requests the latest position of every administrator every 10 seconds, and an administrator whose last point is no more than five minutes old is shown as online.

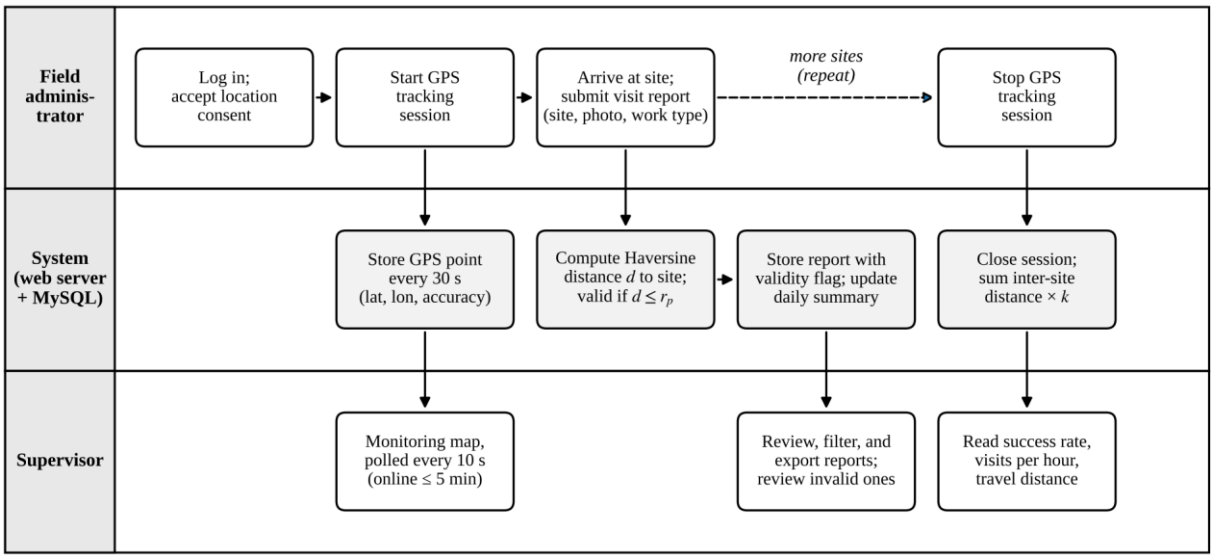


Figure 1. Workflow of LACAKIN across the field administrator, the system, and the supervisor

To submit a visit report, the administrator selects an active project site, attaches a photograph in JPG or PNG format no larger than 5 MB, selects the type of work, and adds optional notes. The server checks the form and the photograph, stores the photograph, and reads its EXIF metadata, including capture time and embedded coordinates, as supplementary evidence. It then computes the Haversine distance between the administrator's position and the site coordinate, compares it with the site radius, stores the report with its validity flag, and updates the administrator's daily summary. Reports outside the radius are kept and flagged rather than discarded, so that the supervisor can review exceptions.

2.3. Distance Calculation and Visit Validation

Let (φ_1, λ_1) be the latitude and longitude of the field administrator and (φ_2, λ_2) those of the project site, converted from degrees to radians. The Haversine distance d is computed with (1) to (3) [12], [18]:

$$a = \sin^2\left(\frac{\Delta\varphi}{2}\right) + \cos\varphi_1 \cdot \cos\varphi_2 \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right) \quad (1)$$

$$c = 2 \cdot \operatorname{atan2}\left(\sqrt{a}, \sqrt{1-a}\right) \quad (2)$$

$$d = R \cdot c \quad (3)$$

where $\Delta\varphi = \varphi_2 - \varphi_1$, $\Delta\lambda = \lambda_2 - \lambda_1$, a is the square of half the chord length between the two points, c is the angular distance in radians, and R is the mean radius of the Earth, taken as 6,371 km. Treating the Earth as a sphere introduces an error of well under 1% of the distance, which is negligible compared with smartphone positioning error at the distances considered here.

Each project site p has its own validation radius r_p, stored in the database. A visit report is accepted when the computed distance does not exceed this radius, as in (4). Because the decision uses the radius stored for the selected site, sites of different sizes can use different tolerances within the same tracking session.

$$v_p = \begin{cases} 1 \text{ (valid)}, & d \le r_p \\ 0 \text{ (invalid)}, & d > r_p \end{cases} \quad (4)$$

In the trial reported in Section 2.6, all four sites were configured with r_p = 700 m. This value was an operational setting for the trial. It was not derived from a calibration procedure, and Section 3.6 therefore examines how the decisions would change under other radii and relates the result to the GPS accuracy observed in the trial. For operational use, a site-specific radius is proposed in (5), where e_p is the half-extent of the site (the distance from the stored coordinate to the farthest point where work is normally done) and u_p is a high percentile, for example the 95th, of the accuracy radius reported by devices at that site.

$$r_p = e_p + u_p \quad (5)$$

2.4. Activity Indicators

The daily summary stores, for each administrator, the number of visits, the number of valid visits, the working duration, and the estimated travel distance between consecutive sites. Two indicators are derived from these values. The success rate *SR* in (6) is the share of visits that were validated, and the efficiency score *ES* in (7) is the number of visits per working hour, where V_{valid} is the number of valid visits, V the total number of visits, and T the working duration in hours.

$$SR = \frac{V_{\text{valid}}}{V} \times 100\% \tag{6}$$

$$ES = \frac{V}{T} \tag{7}$$

These indicators are intended as supervisory aids. *SR* reflects how consistently reported visits match the intended sites, which is the property the system is designed to check, and a low value signals reports that need review. *ES* and travel distance describe how much movement and how many visits occurred. None of the three indicators measures the quality or outcome of the work done at a site, and *ES* can be misleading when sites require visits of different lengths. They are therefore not used in this study to rank or evaluate individual performance, and their validity as productivity measures remains to be established.

The estimated travel distance for a working day is computed with (8), where d_i is the Haversine distance between the i-th and (i+1)-th sites visited, n the number of sites visited, and k a detour factor that converts straight-line distance into an approximate road distance.

$$D_{\text{road}} = k \cdot \sum_{i=1}^{n-1} d_i \tag{8}$$

Constant detour factors in the literature typically lie between 1.25 and 1.41, and the mean detour ratio falls as trip length increases [14]. Circuity also differs between driving, cycling, and walking and across parts of the same city [15], [16]. The value k = 1.5 used in the system is above that typical range and is therefore a conservative, uncalibrated assumption. It is stored as a parameter so that it can be replaced by a value calibrated against routed distances for the area served. In the trial, no administrator visited more than one site on the same day, so (8) did not contribute to any result reported in Section 3.

2.5. Database Design

The database consists of six operational tables and one table for Haversine test records, summarised in Table 2. A user owns many GPS points, tracking sessions, visit reports, and daily summaries, and a project site has many visit reports. Each GPS point stores the reported accuracy, and each visit report stores the computed distance and validity flag, so that every decision can be audited later.

Table 2. Main tables of the LACAKIN database

Table	Content	Key attributes and relationships
users	Field administrators and supervisors; the role attribute controls access	id, username, role, status; one-to-many with the four tables below
project_locations	Project sites with coordinates and an individual validation radius (m)	id, latitude, longitude, radius_valid; one-to-many with visit_reports
gps_tracking	Every GPS point captured during an active session	admin_id, latitude, longitude, accuracy, speed, timestamp
tracking_sessions	Start and end of each tracking session	admin_id, session_start, session_end, is_active
visit_reports	Visit reports with Haversine distance, validity flag, photo, and verification status	admin_id, project_id, jarak_dari_project, is_valid
daily_summary	Daily totals of distance, visits, duration, visits per hour, and success rate	admin_id, tanggal, total_jarak_tempuh, success_rate
haversine_validation	Test routes with expected and computed distance for error analysis	lat1, lon1, lat2, lon2, error_meter

2.6. Evaluation Procedure and Trial Setup

The evaluation had four parts. First, black-box testing checked each function against its expected output with valid and invalid inputs, without reference to the source code [6]. Second, white-box testing with the basis-path technique examined the control flow of five core functions [19]. For each function a flow graph was drawn and the cyclomatic complexity $V(G)$ was calculated with (9) [20], where E is the number of edges, N the number of nodes, and P the number of predicate nodes. $V(G)$ gives the number of independent paths to be executed, and a value of 10 or less is generally considered acceptable [20].

$$V(G) = E - N + 2 = P + 1 \tag{9}$$

Third, the API endpoints for authentication, GPS tracking, visit reports, and supervisor functions were tested with valid requests, invalid parameters, wrong HTTP methods, and requests from unauthorised roles. Fourth, the visit validation was evaluated on the data recorded in a trial. The stored distance of every visit report was recomputed independently from the stored coordinates outside the system, and the absolute difference was calculated. Because the system and the recalculation use the same coordinates, this step measures computational consistency only. The accept and reject decisions were then examined for a range of radii, and the resulting acceptance of off-site reports (false acceptance) and rejection of on-site reports (false rejection) were counted.

Table 3 summarises the trial. The system ran on Windows 11 Pro (64-bit) with XAMPP (Apache, PHP, MySQL) on a computer with an Intel Core i5-12450HX processor, 20 GB of RAM, and a 512 GB SSD, and was tested in Google Chrome. Six visit reports were recorded for four configured sites. The two reports for the Universitas Pamulang site were made on site by the first author on two different days. The other four reports were submitted from one location, with coordinates within 5 m of each other, within three minutes, while four different sites 4.5 to 20.8 km away were selected. They therefore act as controlled off-site cases rather than as independent field visits. Separately, the tracking log for 21 May to 20 June 2026 contained 37 GPS points from one account, which were used to describe the accuracy radius reported in the field.

Table 3. Trial setup

Item	Value
Field administrator accounts	5 (one used for on-site visits by the first author; four used for off-site cases)
Supervisor accounts	1
Project sites	4 in Jakarta and South Tangerang (Universitas Pamulang, Monas, TMII, GBK); radius 700 m each
Visit reports	6: 2 on site (4 and 25 April 2026); 4 off site (27 August 2026)
GPS tracking log	37 points from one account, 21 May to 20 June 2026
Ground truth	Operational labels (on site or off site) from the trial design; no surveyed reference points
Software tests	44 black-box cases (10 modules); 5 basis-path functions; 25 API cases

3. RESULT AND DISCUSSION

This section presents the implemented system, the results of the software tests, the trial results with the radius sensitivity analysis, and a discussion of what the evidence does and does not show.

3.1. System Implementation

The field administrator interface consists of a dashboard with daily statistics (visits, valid visits, distance, and working time), controls to start or stop tracking, a project search, and the five latest reports. The supervisor interface consists of a dashboard, a real-time monitoring page, a report page, and an activity analysis page. Figure 2 shows the monitoring page. The upper table lists each administrator's status, distance, reported GPS accuracy, and last update, and the map shows online administrators and each site as a red marker surrounded by its validation radius. In the view shown, the reported accuracy of the active administrator was 134 m, larger than the 20 to 100 m radii used in earlier studies.

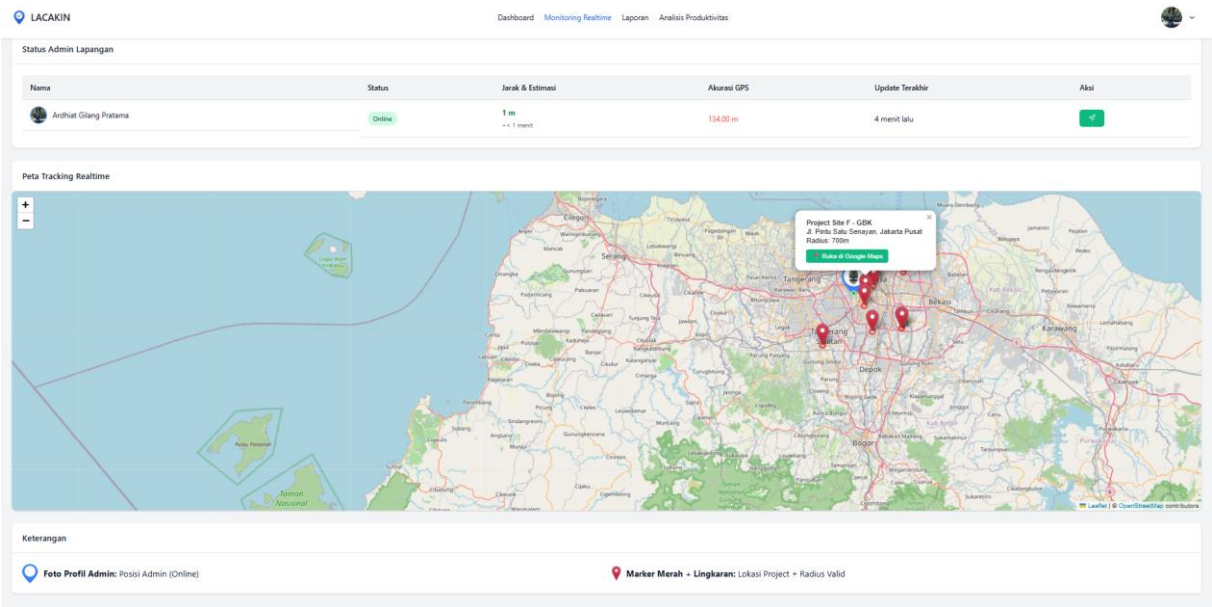


Figure 2. Real-time monitoring page for the supervisor

Figure 3 shows the visit report form. The GPS panel displays the current latitude, longitude, and accuracy, and a button retrieves the position that is used in the Haversine calculation. Below it are the site selection, photo upload, work type, and notes, followed by a map of the selected site so that the administrator can confirm the destination before submitting.

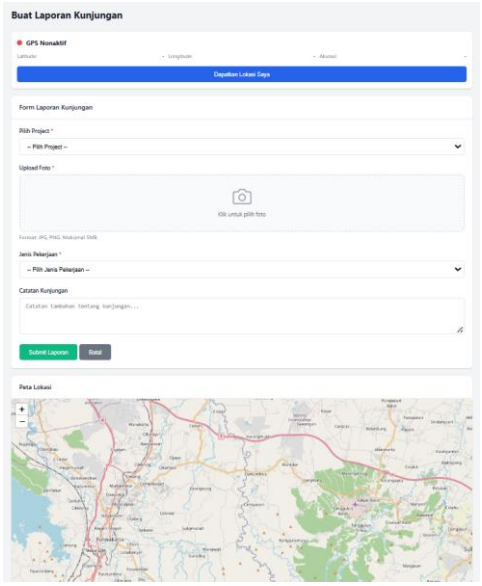


Figure 3. Visit report form for the field administrator

3.2. Black-Box Testing

Black-box testing covered 44 test cases across ten modules, including negative cases such as empty credentials, unregistered usernames, missing site selection, unsupported file formats, files larger than 5 MB, and attempts by one role to open pages of the other role. As Table 4 shows, all 44 cases produced the expected output.

Table 4. Summary of black-box testing

Module	Test cases	Passed	Failed
Login	7	7	0
GPS permission dialog	3	3	0
Submit visit report	7	7	0
Visit report history	3	3	0

Module	Test cases	Passed	Failed
Field administrator profile	4	4	0
Real-time position monitoring	5	5	0
Visit reports (supervisor)	5	5	0
Activity analysis	4	4	0
Logout	2	2	0
Unauthorised access	4	4	0
Total	**44**	**44**	**0**

Table 5 lists the seven cases of the submit-report module, which contains the validation rule. In these cases the test site radius was set to 50 m, so a position 20 m from the site had to be accepted and one 150 m away had to be stored as outside the radius. Both behaved as expected, which shows that (4) uses the radius stored for each site rather than a constant in the code. These results establish functional correctness under controlled inputs. They do not show how reliable the decisions are under real positioning error, which is examined in Sections 3.5 and 3.6.

Table 5. Black-box test cases of the submit-report module

No	Scenario	Input	Expected result	Result
1	Complete report inside radius	Site selected, photo attached, position 20 m from site (radius 50 m)	Report stored as valid	Passed
2	Complete report outside radius	Site selected, photo attached, position 150 m from site (radius 50 m)	Report stored as outside radius	Passed
3	No site selected	Project field empty	Validation error shown	Passed
4	No photo	Photo field empty	Message “Photo is required”	Passed
5	Unsupported format	PDF file uploaded	Message “File format not supported”	Passed
6	File too large	8 MB image uploaded	Message “Maximum file size 5 MB”	Passed
7	Haversine distance	Known administrator and site coordinates	Distance equal to formula result	Passed

3.3. White-Box Testing

Table 6 lists the cyclomatic complexity of the five core functions. Every independent path derived from $V(G)$ was executed with suitable input, and each produced its expected result. The values range from 4 to 9 with a mean of 6.4, within the commonly accepted limit of 10 [20]. The submit-report function has the highest value because it combines input and photo validation, the Haversine calculation, database insertion, and the daily summary update in one request. Because it is the function that accepts or rejects a visit, it is the first candidate for refactoring into smaller, separately testable units.

Table 6. Cyclomatic complexity of the core functions

Function	V(G)	Category	Main decision points
Login	6	Moderate	Empty input, user found, account status, password check, role
Submit visit report	9	Moderate	Session and role, request method, completeness, site exists, photo type and size, insert result
Update location	6	Moderate	Session and role, request method, required fields, insert result
Real-time monitoring	4	Low	Login check, query result, result loop
Activity analysis	7	Moderate	Role, result loops, distance-range interpretation
Mean	**6.4**		

3.4. API Testing

Table 7 summarises the 25 API test cases by endpoint group. Ten were positive cases with valid requests, and fifteen were negative cases: five requests without the required session or role, five invalid or missing parameters, two files of an unsupported type or size, one wrong HTTP method, and two login attempts with invalid credentials or an inactive account. All 25 returned the expected response. In particular, every request from a user without the required role was rejected by the server, so access control does not depend only on the menus shown by the interface. One weakness was found in the design rather than in the tests: error

responses are returned with HTTP status 200 and an error flag in the body. Returning 401 for unauthorised requests and 422 for invalid input would make the API easier to integrate with other clients.

Table 7. API test results by endpoint group

Group	Endpoints	Cases	Positive	Negative	Passed
Authentication	Login, logout	5	2	3	5
GPS tracking	Update location, get tracking data	6	2	4	6
Visit reports	Submit, edit, delete report	9	4	5	9
Supervisor	Dashboard statistics, export reports, export analysis	5	2	3	5
Total		**25**	**10**	**15**	**25**

3.5. Haversine Consistency and Trial Decisions

Table 8 lists the six visit reports recorded in the trial, with the stored distance, the independently recomputed distance, the difference, and the decision under the 700 m radius. Administrators are coded FA-1 to FA-5.

Table 8. Stored and recomputed Haversine distances of recorded visits

No	Admin	Project site	System (m)	Recomputed (m)	Diff. (m)	Case	Decision
1	FA-1	Site UNPAM	93.35	93.36	0.01	On site	Valid
2	FA-1	Site UNPAM	54.75	54.75	0.00	On site	Valid
3	FA-2	Site UNPAM	4,500.52	4,500.47	0.05	Off site	Invalid
4	FA-3	Site A (Monas)	19,851.75	19,851.75	0.00	Off site	Invalid
5	FA-4	Site G (TMII)	20,830.29	20,830.24	0.05	Off site	Invalid
6	FA-5	Site F (GBK)	14,437.36	14,437.36	0.00	Off site	Invalid

The largest difference between stored and recomputed distances was 0.05 m, and four of the six records were identical to two decimal places; the mean relative agreement was 99.99%. The differences come from rounding of coordinates and of the stored value. This agreement confirms that (1) to (3) are implemented correctly, as in [6], but it is not a measure of positioning accuracy, because both values were derived from the same GPS coordinates. If the phone reported a position 100 m from the true one, both values would carry the same 100 m error.

Figure 4 plots the distances against the radius on a logarithmic scale. The two on-site reports were 93.35 m and 54.75 m from the site coordinate and were accepted. The four off-site reports were 4.5 to 20.8 km away and were rejected. Under the trial labels there was therefore no false acceptance and no false rejection. With only two on-site and four off-site cases, however, these rates cannot be generalised, and the off-site cases were far from their sites, which makes them easy to reject.

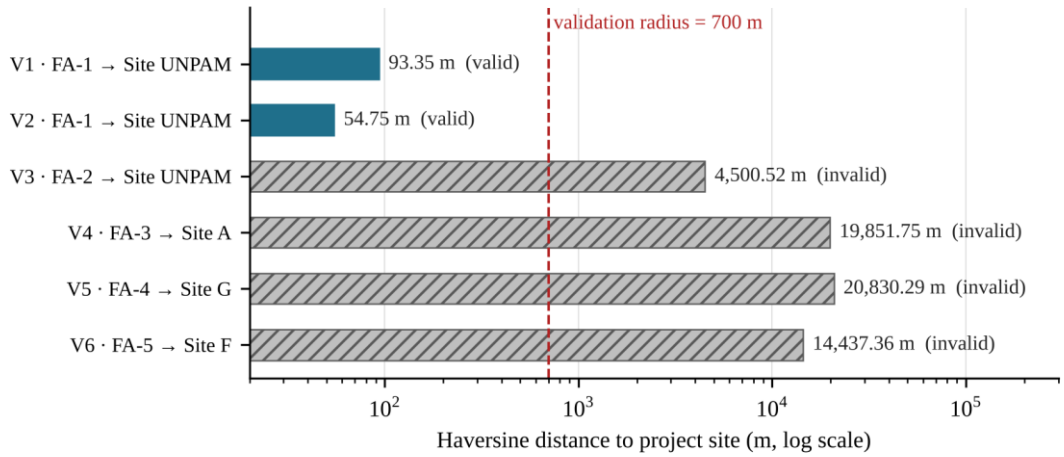


Figure 4. Haversine distance of each recorded visit compared with the 700 m validation radius

3.6. Radius Sensitivity and GPS Accuracy

To examine whether the decisions depend on the 700 m setting, the six recorded distances were re-evaluated with (4) for radii from 25 m to 25 km. Table 9 reports, for each radius, how many on-site reports

would be rejected (false rejections) and how many off-site reports would be accepted (false acceptances). The decisions are identical for every radius from 93.35 m, the larger on-site distance, up to just below 4,500.52 m, the nearest off-site distance. The choice of 700 m therefore did not determine the trial outcome, and any radius in that interval would have produced the same decisions.

Table 9. Decisions on the trial data under different validation radii

Radius (m)	Accepted	False rejections (of 2)	False acceptances (of 4)	Same as 700 m
25	0	2 (100%)	0 (0%)	No
50	0	2 (100%)	0 (0%)	No
100	2	0 (0%)	0 (0%)	Yes
200 to 4,000	2	0 (0%)	0 (0%)	Yes
5,000	3	0 (0%)	1 (25%)	No
15,000	4	0 (0%)	2 (50%)	No
25,000	6	0 (0%)	4 (100%)	No

The lower end of the interval is informative. A radius of 50 m, the value used in the black-box tests, or the 20 m used by Sidiq *et al.* [7], would have rejected both genuine visits. This is consistent with the accuracy observed in the field. The 37 logged GPS points had a mean reported accuracy of 115.16 m, and the monitoring page in Figure 2 shows 134 m for one administrator. These values are an order of magnitude larger than the 7 to 15 m errors reported for smartphone GNSS outdoors or near buildings [2], [4] and closer to the indoor accuracy radii of 16 to 48 m reported in [5]. This suggests that many browser positions in the trial came from indoor or network-assisted fixes rather than from a clear satellite view. Thresholds of 20 to 100 m used for single offices in earlier studies [7], [8] would therefore be too strict for browser-based field reporting under these conditions.

The upper end of the interval is not informative, because the off-site cases were kilometres away. A radius as large as 700 m would accept an administrator who is in the neighbourhood but not at the site, and the trial contains no case that tests this. For this reason the 700 m value should not be read as a recommended setting. The calibration in (5) is proposed instead. For a site such as the university campus, where work may take place up to a few hundred metres from the stored coordinate, a half-extent of 200 m and a 95th-percentile accuracy of about 150 m would give a radius of about 350 m. A compact site with good sky view would receive a much smaller radius. Reports whose reported accuracy exceeds the radius could be routed to manual review instead of being decided automatically. Calibrating e_p and u_p requires accuracy data collected at each site, which this trial did not provide.

3.7. Discussion

Compared with the studies in Table 1, LACAKIN keeps the same distance method but changes the unit of evaluation from the attendance day to the individual visit, and the reference point from one office to the site selected for each report. This matters for field work because a single daily check-in, as in [7]–[9], says nothing about the second or third site visited. The evaluation also differs. Earlier studies reported a computational difference [6], a distance-method comparison [12], or usability [10], [11]. This study separates three questions: whether the formula is computed correctly (Table 8), whether the decision depends on the chosen threshold (Table 9), and how that threshold relates to the accuracy observed in the field. The first question is answered conclusively. The second and third are answered only for the small trial, and they show that the strict radii used in earlier single-office systems would fail under the accuracy observed here.

For supervisors, the main practical change is that evidence is attached to each report when it is created. Instead of comparing a photograph with an administrator's message, the supervisor sees the distance, the validity flag, the reported accuracy, and the time next to the photograph, and can filter reports by administrator and date. Invalid reports are kept, so legitimate exceptions such as an inaccurate stored site coordinate can still be accepted manually. The activity indicators in Section 2.4 are updated automatically, but, as discussed there, they describe activity rather than performance, and the trial was too small to show whether supervisors make better or fairer judgements with them.

Scalability was not measured, but the expected load can be estimated from the design. One administrator sending a point every 30 seconds produces 120 points per hour, or 960 rows in an eight-hour day. One hundred administrators would generate about 3.3 location requests per second and 96,000 rows per day, which a single MySQL server can handle if the tracking table is indexed by administrator and time and old points are archived. The 10-second polling of the monitoring page adds one query per open supervisor page. Load testing under these conditions is needed before deployment at that scale.

Continuous tracking also raises privacy concerns, because location data reveal where a person is and how they move, and employees' acceptance of monitoring depends on its purpose and context [13]. LACAKIN starts tracking only after the administrator has accepted a consent dialog that states the purpose of location access, lets the administrator start and stop tracking so that positions are recorded only during working sessions, and restricts location data to the supervisor role, which was confirmed by the unauthorised-access and API tests. In Indonesia, location data linked to an identifiable employee are personal data under Law No. 27 of 2022 on Personal Data Protection [21]. Organisations that adopt such a system should define a lawful basis and purpose for processing, a retention period for raw GPS points, and how the data may and may not be used in evaluation.

The study has several limitations. First, the trial is small: six visit reports from five accounts at four sites, of which only two were genuine on-site visits made by one person and four were controlled off-site cases submitted from one location. The absence of false acceptances and false rejections therefore demonstrates that the validation logic works as designed, but it does not estimate error rates in daily operation. Second, no surveyed ground-truth coordinates were collected, so positioning error was described only through the accuracy radius reported by the device. Third, the detour factor of 1.5 was not validated against routed distances and did not affect any reported result. Fourth, browser location can be falsified with mock-location tools, and photo EXIF data can be edited. Fifth, tracking reliability under different environments and devices, usability, and the effect of the system on supervisors' decisions were not evaluated. Future work should collect visits at several sites over a longer period with surveyed reference points, include near-site negative cases at 100 to 1,000 m, estimate false acceptance and false rejection rates for calibrated radii, validate the detour factor against routed distances, add mock-location detection, and evaluate the system with supervisors and administrators in daily use.

4. CLOSING

This study designed and implemented LACAKIN, a web-based real-time GPS tracking system that validates each field visit report against the coordinate and individual radius of the selected project site using the Haversine formula. The technical contribution is a visit-level validation model for multi-site field work, together with an evaluation that separates the correctness of the distance calculation from the quality and sensitivity of the accept and reject decisions.

The implementation passed all 44 black-box and 25 API test cases, and its core functions had cyclomatic complexity values between 4 and 9 (mean 6.4). Independent recalculation reproduced the stored distances within 0.05 m, which confirms computational consistency but not positioning accuracy. In a small trial with a 700 m radius, two on-site reports were accepted and four off-site reports were rejected. The same decisions would have been reached with any radius from 93.35 m to 4,500 m. The mean reported GPS accuracy of 115.16 m showed that radii below about 100 m would reject genuine visits under the conditions observed.

These results show that the system works as designed and that visit-level validation is feasible. They do not yet show that the system improves the evaluation of field administrators' productivity. That claim requires larger field trials with surveyed reference points, site-specific radius calibration, validation of the detour factor against routed distances, and an evaluation of whether the activity indicators support fair and accurate supervisory decisions.

ACKNOWLEDGMENT

The authors thank the Informatics Engineering Study Program, Faculty of Computer Science, Universitas Pamulang, for its support during this research.

REFERENCES

[1] Robert, “Implementasi GPS tracking and monitoring, pengenal wajah, pengenal sidik jari pada sistem presensi karyawan berbasis mobile,” *Jurnal Ilmiah KOMPUTASI*, vol. 19, no. 2, pp. 203–216, 2020, doi: 10.32409/jikstik.19.2.2796.

[2] K. Merry and P. Bettinger, “Smartphone GPS accuracy study in an urban environment,” *PLoS ONE*, vol. 14, no. 7, Art. no. e0219890, 2019, doi: 10.1371/journal.pone.0219890.

[3] W. Magiera *et al.*, “Accuracy of code GNSS receivers under various conditions,” *Remote Sensing*, vol. 14, no. 11, Art. no. 2615, 2022, doi: 10.3390/rs14112615.

[4] G. Cappello, C. Gioia, A. Angrisano, S. Del Pizzo, G. Portelli, and S. Gaglione, “Reliability of smartphone positioning in harsh environment,” *Engineering Proceedings*, vol. 54, no. 1, Art. no. 44, 2023, doi: 10.3390/ENC2023-15429.

[5] J. R. Palathinkal, H. I. Nabil, M. I. Rochman, H. Nasiri, F. A. Gatsi, and M. Ghosh, “Evaluating smartphone GNSS accuracy for geofenced 6 GHz operations,” arXiv:2603.26706, 2026.

[6] F. Antono and S. Dwiasnati, "Implementasi absensi karyawan menggunakan algoritma Haversine dengan global positioning system berbasis Android," *Jurnal Esensi Infokom: Jurnal Esensi Sistem Informasi dan Sistem Komputer*, vol. 6, no. 1, pp. 1–10, 2022, doi: 10.55886/infokom.v6i1.459.

[7] H. Sidiq, D. Satria, and Humaira, "Implementasi algoritma Haversine pada absensi kepegawaian berbasis Android," *Journal of Applied Computer Science and Technology (JACOST)*, vol. 3, no. 1, pp. 120–126, 2022, doi: 10.52158/jacost.v3i1.263.

[8] Painem and H. Soetanto, "Sistem presensi pegawai berbasis web service menggunakan metode RESTful dengan keamanan JWT dan algoritma Haversine," *Fountain of Informatics Journal*, vol. 5, no. 3, pp. 6–11, 2020, doi: 10.21111/fij.v5i3.4906.

[9] M. Fauzan, E. Christian, and P. B. A. A. Putra, "Rancang bangun aplikasi presensi menggunakan algoritma Haversine (Studi kasus Badan Kepegawaian Daerah Provinsi Kalimantan Tengah) berbasis Android," *Journal of Information Technology and Computer Science (JOINTECOMS)*, vol. 4, no. 1, pp. 81–88, 2024, doi: 10.47111/jointecoms.v4i1.13205.

[10] D. Anita and Rahmawati, "Pengembangan sistem presensi online berbasis web dengan validasi lokasi menggunakan GPS dan Haversine formula," *OKTAL: Jurnal Ilmu Komputer dan Sains*, vol. 4, no. 12, pp. 828–839, 2025.

[11] A. A. Che Fauzi, N. Ruzana, A. Mat, N. Ashafiqa, and A. Fauzi, "TECH-CLASS: A geofencing-driven attendance management system for higher education," *Journal of Mathematics and Computing Science*, vol. 11, no. 2, pp. 61–70, 2025, doi: 10.24191/jmcs.v11i2.9707.

[12] Y. Miftahuddin, S. Umaroh, and F. R. Karim, "Perbandingan metode perhitungan jarak Euclidean, Haversine, dan Manhattan dalam penentuan posisi karyawan (Studi kasus: Institut Teknologi Nasional Bandung)," *Jurnal Tekno Insentif*, vol. 14, no. 2, pp. 69–77, 2020, doi: 10.36787/jti.v14i2.270.

[13] J. Vitak and M. Zimmer, "Surveillance and the future of work: Exploring employees' attitudes toward monitoring in a post-COVID workplace," *Journal of Computer-Mediated Communication*, vol. 28, no. 4, Art. no. zmad007, 2023, doi: 10.1093/jcmc/zmad007.

[14] H. Yang, J. Ke, and J. Ye, "A universal distribution law of network detour ratios," *Transportation Research Part C: Emerging Technologies*, vol. 96, pp. 22–37, 2018, doi: 10.1016/j.trc.2018.09.012.

[15] M. Costa, M. Marques, and F. Moura, "A circuity temporal analysis of urban street networks using open data: A Lisbon case study," *ISPRS International Journal of Geo-Information*, vol. 10, no. 7, Art. no. 453, 2021, doi: 10.3390/ijgi10070453.

[16] K. M. Cubukcu, "Using circuity as a network efficiency measure: The example of Paris," *Spatial Information Research*, vol. 29, pp. 163–172, 2021, doi: 10.1007/s41324-020-00342-w.

[17] R. S. Pressman and B. R. Maxim, *Software Engineering: A Practitioner's Approach*, 9th ed. New York, NY, USA: McGraw-Hill Education, 2020.

[18] C. Husada, K. D. Hartomo, and H. P. Chernovita, "Implementasi Haversine formula untuk pembuatan SIG jarak terdekat ke RS rujukan COVID-19," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 4, no. 5, pp. 874–883, 2020, doi: 10.29207/resti.v4i5.2255.

[19] J. B. L. Sie, I. A. Musdar, and S. Bahri, "Pengujian white box testing terhadap website Room Menggunakan teknik basis path," *KHARISMA Tech*, vol. 17, no. 2, pp. 45–57, 2022, doi: 10.55645/kharismatech.v17i2.235.

[20] T. J. McCabe, "A complexity measure," *IEEE Transactions on Software Engineering*, vol. SE-2, no. 4, pp. 308–320, 1976, doi: 10.1109/TSE.1976.233837.

[21] Republic of Indonesia, *Undang-Undang Nomor 27 Tahun 2022 tentang Pelindungan Data Pribadi* [Law No. 27 of 2022 on Personal Data Protection]. Jakarta, Indonesia, 2022.