

BAB 2

LANDASAN TEORI

2.1 Sistem Presensi Karyawan

Sistem presensi karyawan merupakan mekanisme yang digunakan untuk mencatat kehadiran, keterlambatan, dan ketidakhadiran karyawan dalam suatu organisasi. Sistem ini memiliki peran penting dalam manajemen sumber daya manusia, khususnya dalam evaluasi kinerja, penggajian, serta peningkatan kedisiplinan karyawan [3, 5].

Perkembangan teknologi telah mendorong perubahan sistem presensi dari metode manual menjadi sistem digital yang lebih akurat dan efisien [10, 3]. Salah satu pendekatan yang digunakan adalah sistem presensi berbasis lokasi yang memanfaatkan teknologi GPS untuk memperoleh koordinat pengguna sebagai dasar validasi kehadiran [3, 12].

2.2 Global Positioning System (GPS)

Global Positioning System (GPS) merupakan sistem navigasi berbasis satelit yang digunakan untuk menentukan posisi suatu perangkat di permukaan bumi [13]. GPS bekerja dengan menerima sinyal dari beberapa satelit dan menghitung posisi menggunakan metode trilaterasi [13].

Dalam sistem presensi berbasis lokasi, GPS digunakan untuk memperoleh koordinat posisi pengguna secara *real-time* sebagai dasar dalam proses validasi kehadiran [3, 12].

2.3 Geofencing

Geofencing merupakan teknik yang digunakan untuk menentukan batas geografis virtual pada suatu area tertentu [14]. Dengan teknologi ini, sistem dapat mendeteksi apakah suatu perangkat berada di dalam atau di luar area yang telah ditentukan [14].

Pada penelitian ini, geofencing digunakan untuk membatasi area presensi sehingga pengguna hanya dapat melakukan presensi jika berada dalam radius tertentu dari lokasi yang telah ditetapkan [11, 12].

2.4 Perhitungan Jarak dengan Rumus Haversine

Rumus Haversine merupakan metode yang digunakan untuk menghitung jarak antara dua titik pada permukaan bumi berdasarkan koordinat lintang (*latitude*) dan bujur (*longitude*). Dalam perhitungan ini, bumi diasumsikan berbentuk bola dengan jari-jari tertentu, sehingga jarak antara dua titik dihitung berdasarkan panjang busur pada permukaan bumi. Rumus ini banyak digunakan pada sistem berbasis lokasi karena mampu memperhitungkan kelengkungan bumi dan tetap memiliki proses komputasi yang relatif sederhana [9].

Secara umum, jarak antara dua titik pada permukaan bumi dapat dihitung menggunakan konsep *great-circle distance*. Apabila terdapat dua titik koordinat, yaitu titik pertama (φ_1, λ_1) dan titik kedua (φ_2, λ_2) , maka jarak permukaan antara kedua titik tersebut dapat diperoleh dari sudut pusat yang terbentuk di antara keduanya. Persamaan dasarnya dapat dituliskan sebagai berikut:

$$d = R \cdot c \quad (2.1)$$

dengan d merupakan jarak antara dua titik, R merupakan jari-jari bumi, dan c merupakan sudut pusat antara dua titik dalam satuan radian. Dengan demikian, inti dari perhitungan jarak adalah mencari nilai sudut pusat c terlebih dahulu.

Pada rumus Haversine, sudut pusat tersebut dihitung dengan menggunakan fungsi *haversine*. Fungsi *haversine* didefinisikan sebagai berikut:

$$\text{hav}(\theta) = \sin^2\left(\frac{\theta}{2}\right) \quad (2.2)$$

Berdasarkan fungsi tersebut, hubungan antara dua titik koordinat pada permukaan bumi dapat dituliskan sebagai berikut:

$$\text{hav}(c) = \text{hav}(\Delta\varphi) + \cos(\varphi_1) \cos(\varphi_2) \text{hav}(\Delta\lambda) \quad (2.3)$$

dengan $\Delta\varphi$ merupakan selisih lintang dan $\Delta\lambda$ merupakan selisih bujur. Karena fungsi *haversine* bernilai $\sin^2\left(\frac{\theta}{2}\right)$, maka persamaan tersebut dapat diubah menjadi bentuk berikut:

$$\sin^2\left(\frac{c}{2}\right) = \sin^2\left(\frac{\Delta\varphi}{2}\right) + \cos(\varphi_1)\cos(\varphi_2)\sin^2\left(\frac{\Delta\lambda}{2}\right) \quad (2.4)$$

Agar lebih mudah diimplementasikan ke dalam program, bagian kanan dari persamaan tersebut disederhanakan ke dalam variabel sementara a . Oleh karena itu, bentuk implementasinya menjadi:

$$a = \sin^2\left(\frac{\Delta\varphi}{2}\right) + \cos(\varphi_1)\cos(\varphi_2)\sin^2\left(\frac{\Delta\lambda}{2}\right) \quad (2.5)$$

Setelah nilai a diperoleh, sudut pusat antara dua titik dapat dihitung dengan persamaan berikut:

$$c = 2 \cdot \arctan 2\left(\sqrt{a}, \sqrt{1-a}\right) \quad (2.6)$$

Pada tahap akhir, jarak antara dua titik dihitung dengan mengalikan sudut pusat c dengan jari-jari bumi R , sehingga diperoleh persamaan:

$$d = R \cdot c \quad (2.7)$$

Keterangan:

- d = jarak antara dua titik (meter)
- R = jari-jari bumi (6371000 meter)
- φ_1 = lintang titik pertama dalam radian
- φ_2 = lintang titik kedua dalam radian
- λ_1 = bujur titik pertama dalam radian
- λ_2 = bujur titik kedua dalam radian
- $\Delta\varphi = \varphi_2 - \varphi_1$ = selisih lintang dalam radian
- $\Delta\lambda = \lambda_2 - \lambda_1$ = selisih bujur dalam radian
- a = nilai antara pada perhitungan Haversine

- c = sudut pusat antara dua titik dalam radian

Sebelum dilakukan perhitungan, nilai lintang dan bujur yang diperoleh dari GPS umumnya masih berada dalam satuan derajat. Oleh karena itu, nilai tersebut perlu dikonversi terlebih dahulu ke dalam satuan radian menggunakan persamaan berikut:

$$\text{radian} = \text{derajat} \times \frac{\pi}{180} \quad (2.8)$$

Dalam sistem ini, rumus Haversine digunakan untuk menghitung jarak antara lokasi pengguna dan titik presensi yang telah ditentukan. Hasil jarak tersebut kemudian dibandingkan dengan radius area presensi. Apabila jarak pengguna lebih kecil atau sama dengan radius yang ditentukan, maka pengguna dianggap berada di dalam area presensi. Sebaliknya, apabila jarak pengguna melebihi radius tersebut, maka proses presensi tidak dapat dilakukan.

2.5 REST (Representational State Transfer) API

REST (Representational State Transfer) API merupakan arsitektur komunikasi yang digunakan dalam pengembangan sistem berbasis web untuk pertukaran data antara *client* dan *server*. REST API memanfaatkan protokol HTTP dengan metode seperti GET, POST, PUT, dan DELETE untuk melakukan operasi terhadap *resource* yang tersedia pada *server* [15].

Dalam sistem presensi berbasis lokasi, REST API digunakan sebagai penghubung antara aplikasi *mobile* dengan *server* untuk mengirim dan menerima data seperti data kehadiran, lokasi pengguna, serta informasi autentikasi. Penggunaan REST API memungkinkan sistem menjadi lebih modular, *scalable*, dan mudah diintegrasikan dengan berbagai platform [15, 4].

Karakteristik utama REST API meliputi *stateless*, *client-server separation*, *cacheable*, serta penggunaan *resource* berbasis URL yang unik [15].

2.6 User Acceptance Testing (UAT)

User Acceptance Testing (UAT) merupakan tahap pengujian perangkat lunak yang dilakukan oleh pengguna akhir untuk memastikan bahwa sistem telah sesuai dengan kebutuhan dan harapan pengguna. UAT dilakukan setelah proses

pengujian fungsional dan sebelum sistem benar-benar diimplementasikan secara penuh [16].

Dalam penelitian ini, UAT dilakukan melalui pemberian skenario penggunaan dan kuesioner kepada responden. Penilaian responden diukur menggunakan skala Likert untuk mengetahui tingkat penerimaan pengguna terhadap sistem yang dikembangkan [17].

2.7 Skala Likert

Skala Likert merupakan metode pengukuran yang digunakan untuk mengukur sikap, pendapat, atau persepsi seseorang terhadap suatu pernyataan. Skala ini umumnya digunakan dalam kuesioner penelitian untuk mendapatkan data kuantitatif dari aspek kualitatif [17].

Dalam penelitian ini, skala Likert digunakan pada proses UAT untuk mengukur tingkat kepuasan dan penerimaan pengguna terhadap sistem yang dikembangkan. Responden diberikan beberapa pernyataan dan diminta untuk memilih tingkat persetujuan berdasarkan skala berikut:

- 1 = Sangat Tidak Setuju
- 2 = Tidak Setuju
- 3 = Netral
- 4 = Setuju
- 5 = Sangat Setuju

Data yang diperoleh kemudian dapat diolah menjadi nilai persentase untuk menentukan tingkat kepuasan pengguna terhadap sistem.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A