IMPLEMENTASI *POINT INCLUSION IN POLYGON TEST* UNTUK PENDATAAN KEHADIRAN PEGAWAI BERBASIS BIOMETRIK DI PERANGKAT *MOBILE*

SKRIPSI

UIN SUNAN AMPEL
S U R A B A Y A

Disusun Oleh:

BAYU UTOMO

H76216055

PROGRAM STUDI SISTEM INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUNAN AMPEL
SURABAYA

2019

LEMBAR PERNYATAAN KEASLIAN

Saya yang bertanda tangan dibawah ini,

Nama	:	Bayu Utomo
NIM	:	H76216055
Program Studi	:	Sistem Informasi
Angkatan	:	2016

Menyatakan bahwa saya tidak melakukan plagiat dalam penulisan skripsi saya yang berjudul: "IMPLEMENTASI *POINT INCLUSION IN POLYGON TEST* UNTUK PENDATAAN KEHADIRAN PEGAWAI BERBASIS BIOMETRIK DI PERANGKAT *MOBILE*". Apabila suatu saat nanti terbukti saya melakukan tindakan plagiat, maka saya bersedia menerima sanksi yang telah ditetapkan.

Demikian pernyataan keaslian ini saya buat dengan sebenar-benarnya.

Surabaya, 29 November 2019

Yang menyatakan,

METERAI TEMPEL

62C37AHF115007337

6000

ENAM RIBU RUPIAH

Bayu Utomo

NIM. H76216055

LEMBAR PERSETUJUAN PEMBIMBING

Skripsi oleh

NAMA : BAYU UTOMO

NIM : H76216055

JUDUL : IMPLEMENTASI *POINT INCLUSION IN POLYGON TEST* UNTUK PENDATAAN KEHADIRAN PEGAWAI BERBASIS BIOMETRIK DI PERANGKAT *MOBILE*

Ini telah diperiksa dan disetujui untuk diujikan.

Surabaya, 30 DESEMBER 2019

Dosen Pembimbing 1	Dosen Pembimbing 2
Achmad Teguh Wibowo, MT	Mujib Ridwan, S.Kom., M.T
NIP. 198810262014031003	NIP. 198604272014031004

PENGESAHAN TIM PENGUJI SKRIPSI

Skripsi Bayu Utomo ini telah dipertahankan
Di depan tim penguji skripsi
Di Surabaya, 23 Desember 2019

Mengesahkan,
Dewan Penguji

Penguji I

Achmad Teguh Wibowo, MT
NIP. 198810262014031003

Penguji II

Mujib Ridwan, S.Kom., M.T
NIP. 198604272014031004

Penguji III

Ahmad Yusuf, M. Kom
NIP. 199001202014031003

Penguji IV

Indri Sudanawati Rozas, M. Kom
NIP. 198207212014032001

Mengetahui,
Dekan Fakultas Sains dan Teknologi
UIN Sunan Ampel Surabaya

Dr. Eni Purwati, M.Ag
NIP. 196512211990022001

KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI SUNAN AMPEL SURABAYA
PERPUSTAKAAN
Jl. Jend. A. Yani 117 Surabaya 60237 Telp. 031-8431972 Fax.031-8413300
E-Mail: perpus@uinsby.ac.id

LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI
KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademika UIN Sunan Ampel Surabaya, yang bertanda tangan di bawah ini, saya:

Nama : BAYU UTOMO
NIM : H76216055
Fakultas/Jurusan : SAINS DAN TEKNOLOGI / SISTEM INFORMASI
E-mail address : by@uinsby.ac.id

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Perpustakaan UIN Sunan Ampel Surabaya, Hak Bebas Royalti Non-Eksklusif atas karya ilmiah :
☑ Sekripsi ☐ Tesis ☐ Desertasi ☐ Lain-lain (.....)
yang berjudul :
IMPLEMENTASI POINT INCLUSION IN POLYGON TEST UNTUK PENDATAAN KEHADIRAN PEGAWAI BERBASIS BIOMETRIK DI PERANGKAT MOBILE

beserta perangkat yang diperlukan (bila ada). Dengan Hak Bebas Royalti Non-Ekslusif ini Perpustakaan UIN Sunan Ampel Surabaya berhak menyimpan, mengalih-media/format-kan, mengelolanya dalam bentuk pangkalan data (database), mendistribusikannya, dan menampilkan/mempublikasikannya di Internet atau media lain secara ***fulltext*** untuk kepentingan akademis tanpa perlu meminta ijin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan atau penerbit yang bersangkutan.

Saya bersedia untuk menanggung secara pribadi, tanpa melibatkan pihak Perpustakaan UIN Sunan Ampel Surabaya, segala bentuk tuntutan hukum yang timbul atas pelanggaran Hak Cipta dalam karya ilmiah saya ini.

Demikian pernyataan ini yang saya buat dengan sebenarnya.

Surabaya, 03 JANUARI 2020

Penulis

(BAYU UTOMO)
nama terang dan tanda tangan

ABSTRAK

IMPLEMENTASI *POINT INCLUSION IN POLYGON TEST* UNTUK PENDATAAN KEHADIRAN PEGAWAI BERBASIS BIOMETRIK DI PERANGKAT *MOBILE*

Oleh:

Bayu Utomo

Sistem kehadiran menggunakan *fingerprint* sudah banyak digunakan pada perusahaan. Tetapi sistem ini memiliki beberapa kekurangan, salah satunya adalah mobilitas. Banyak penelitian yang dilakukan terkait dengan peningkatan sistem kehadiran, salah satunya memanfaatkan teknologi *mobile*. Penelitian ini mengembangkan aplikasi pendataan kehadiran pegawai berbasis biometrik di perangkat mobile. Sistem yang dikembangkan menggunakan metode *Point Inclusion in Polygon Test* untuk membatasi penggunaan aplikasi hanya di dalam wilayah poligon yang sudah ditentukan. Aplikasi yang dihasilkan dapat dijalankan pada sistem operasi Android dan IOS yang memiliki sensor *fingerprint*. Hasil dari percobaan yang dilakukan untuk menganalisa metode yang diimplementasikan ke dalam aplikasi menunjukan keberhasilan sebesar 87%, tergantung dengan sensor GPS yang digunakan pada perangkat *mobile*. Percobaan kedua dilakukan untuk melihat *network latency* menghasilkan kesimpulan bahwa jaringan 4G pada Android 8 adalah solusi terbaik ketika menggunakan aplikasi ini.

Kata kunci: Fingerprint, Geographic Information Retrieval, Mobile, PNPOLY, Sistem Kehadiran

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

ABSTRACT

IMPLEMENTATION OF POINT INCLUSION IN POLYGON TEST FOR RECORDS ATTENDANCE SYSTEM BASED ON MOBILE DEVICE BIOMETRIC SYSTEM

By:

Bayu Utomo

The fingerprint attendance system widely used in companies, but this system has several disadvantages. Much research has done to improve the attendance systems; one of them utilizes mobile technology. This study developed a biometric-based records attendance system on mobile devices. The application developed using the Point Inclusion in Polygon Test method to limit application usage only within predetermined polygon areas. The application result could run on Android and IOS operating systems that have a fingerprint sensor. The results of an experiment to analyze the method that implemented into the application showed 87% of successfully, depending on the GPS sensor used on the mobile device. The second experiment conducted to see the network latency resulting in the conclusion that the 4G network on Android 8 is the best solution when using this application.

Keywords: Attendance System, Fingerprint, Geographic Information Retrieval, Mobile, PNPOLY

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

DAFTAR ISI

PENGESAHAN TIM PENGUJI SKRIPSI ..... **i**
LEMBAR PERNYATAAN KEASLIAN ..... **ii**
LEMBAR PERNYATAAN PUBLIKASI KARYA ILMIAH ..... **iii**
MOTTO ..... **iv**
KATA PENGANTAR ..... **v**
ABSTRAK ..... **vii**
DAFTAR ISI ..... **ix**
DAFTAR TABEL ..... **xii**
DAFTAR GAMBAR ..... **xiii**
BAB I PENDAHULUAN **1**
1.1 Latar Belakang 1
1.2 Perumusan Masalah 4
1.3 Batasan Masalah 4
1.4 Tujuan Penelitian 4
1.5 Manfaat Penelitian 5
1.5.1 Akademik 5
1.5.2 Aplikatif 5
BAB II TINJAUAN PUSTAKA **6**
2.1 Tinjauan Penelitian Terdahulu 6
2.2 Teori-Teori Dasar 8
2.2.1 *Geographic Information System* (GIS) 8
2.2.2 *Geographic Information Retrieval* (GIR) 9
2.2.3 *Inclusion of a Point in a Polygon* 9
2.2.4 *Point Inclusion in Polygon Test* (PNPOLY) 10
2.2.5 *Biometric Information System* 12
2.2.6 *Fingerprint* 13
2.2.7 *Smartphone* 13
2.2.8 *Database* 13

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

2.2.9 *Javascript* 15
2.2.10 *Framework React Native* 16
2.2.11 *PHP* 16
2.2.12 *Unified Modeling Language* (UML) 17
2.2.13 *Mock Location* 19
2.2.14 *Network Latency* 19
2.2.15 *Test Case Scenario* 20
2.2.13 *Device ID* 20
2.3 Integrasi Keilmuan 21
BAB III METODOLOGI PENELITIAN 24
3.1 Desain Penelitian 24
3.1.1 Perumusan Masalah 24
3.1.2 Studi Pustaka 25
3.1.3 Pengembangan Aplikasi 25
3.1.4 Rancangan Pengujian 30
3.1.5 Evaluasi Sistem 32
3.2 Tempat dan Waktu Penelitian 32
BAB IV HASIL DAN PEMBAHASAN 33
4.1 Analisis Kebutuhan 33
4.2 Desain Arsitektur Sistem 34
4.3 Alur Penggunaan Aplikasi 35
4.4 *UML Class Diagram* 36
4.5 *UML Use Case Diagram* 39
4.6 *UML Activity Diagram* 40
4.7 *UML Sequence Diagram* 43
4.8 *Entity Relationship Diagram* (ERD) 46
4.9 *Conceptual Data Model* (CDM) 46
4.10 *Physical Data Model* (PDM) 48
4.11 Implementasi Metode 48
4.12 Desain *User Interface* (UI) Aplikasi 49
4.13 Pengujian Fungsionalitas Aplikasi 54

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

4.14 Evaluasi Aplikasi 56
BAB V PENUTUP 66
5.1 Kesimpulan 66
5.2 Saran 67
DAFTAR PUSTAKA 68
LAMPIRAN 71

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

DAFTAR TABEL

Tabel 3.1: *Test Case Scenario* Pengujian Fungsionalitas Aplikasi 27
Tabel 3.2: Daftar Lokasi Uji Coba Aplikasi 31
Tabel 4.1: Hasil Pengujian Berdasarkan *Test Case Scenario* 54
Tabel 4.2: *Raw Data* Hasil Pengujian Metode PNPOLY 56
Tabel 4.3: Hasil Pengujian Metode PNPOLY Berdasarkan Koordinat GPS 59
Tabel 4.4: Pengelompokan Kegagalan Pengujian Berdasarkan Sistem Operasi .. 60
Tabel 4.5: Rata-Rata *Network Latency (Seconds)* 64

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

DAFTAR GAMBAR

Gambar 2.1: Metode *crossing number* (cn) 10
Gambar 2.2: Metode *winding number* (wn) 10
Gambar 2.3: *Flowchart* Metode PNPOLY 11
Gambar 2.4: Komponen DBMS 15
Gambar 2.5: Sejarah UML 17
Gambar 2.6: Tampilan Diagram UML 18
Gambar 2.7: Daftar Pengaruh *Network Latency* 19
Gambar 2.8: Diagram Alur Pengambilan *Device ID* 20
Gambar 3.1: Diagram Alur Penelitian Skripsi 24
Gambar 3.2: Diagram Alur Pengembangan Aplikasi 25
Gambar 4.1: Desain Arsitektur Sistem 34
Gambar 4.2: Alur Penggunaan Aplikasi 35
Gambar 4.3: *UML Class Diagram* 36
Gambar 4.4: *UML Use Case Diagram* 40
Gambar 4.5: *UML Activity Diagram Register Device* 41
Gambar 4.6: *UML Activity Diagram Insert Arrival Time* 42
Gambar 4.7: *UML Activity Diagram Update Leaving Time* 42
Gambar 4.8: *UML Sequence Diagram Register Device* 43
Gambar 4.9: *UML Sequence Diagram Insert Arrival Time* 45
Gambar 4.10: *UML Sequence Diagram Update Leaving Time* 45
Gambar 4.11: ERD Aplikasi 46
Gambar 4.12: CDM Aplikasi 47
Gambar 4.13: PDM Aplikasi 48
Gambar 4.14: *Function Javascript* Metode PNPOLY 49
Gambar 4.15: UI *Register Screen* 50
Gambar 4.16: UI *Home Screen* Saat Posisi Pengguna di Dalam Poligon 51
Gambar 4.17: UI *Home Screen* Saat Posisi Pengguna di Luar Poligon 52
Gambar 4.18: UI *Home Screen* Saat Menggunakan *Fake GPS* 53

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Gambar 4.19: *Network Latency* di Android 8 61
Gambar 4.20: *Network Latency* di Android 9 62
Gambar 4.21: *Network Latency* di IOS 12 63
Gambar 4.22: *Network Latency* di IOS 13 64
Gambar 4.23: Grafik Rata-Rata *Network Latency* 65

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

BAB I
PENDAHULUAN

1.1 Latar Belakang

Penggunaan identitas di dunia modern sudah berkembang menjadi berbagai macam jenis dan kegunaanya. Salah satunya adalah penggunaan identitas biometrik pada perangkat tertentu yang berguna untuk mengamankan data dan informasi bersifat pribadi. Bentuk pemanfaatan identitas biometrik dalam dunia modern adalah penggunaan *fingerprint system*. Teknologi ini membutuhkan dua komponen utama untuk memindai sidik jari manusia. Pertama adalah *fingerprint scanner* sebagai alat pemindai sidik jari. Kedua adalah *fingerprint authentication* yang merupakan sebuah metode untuk membandingkan kecocokan sidik jari antar manusia (Narinder, 2009).

Teknologi *fingerprint* berkembang setiap tahunya, data yang diperoleh dari *Global Biometric Market Projection by Technology* pada tahun 2005 sampai 2012 menunjukan pertumbuhan pasar biometrik yang sangat pesat terjadi terutama pada teknologi *fingerprint*. Pada tahun 2005 penjualan teknologi ini menghasilkan lebih dari $0,5 *billion* dan di tahun 2012 naik 5 kali lipat, menjadi $2,5 *billion* (Mil'shtein, 2005). Saat ini banyak instansi yang menggunakan teknologi *fingerprint* untuk keperluan pendataan kehadiran. Selain itu *fingerprint* juga dapat diimplementasikan untuk keperluan keamanan perangkat *mobile*, dikarenakan beberapa tahun terakhir permintaan akan *smartphone* terus meningkat setiap tahunya, dan setiap vendor ingin berinovasi untuk memberikan keamanan, kenyamanan dan kemudahan dalam penggunaannya.

Fingerprint scanner konvensional masih memiliki kelemahan. Teknologi ini memiliki beberapa kompleksitas dalam memperoleh gambar berkualitas tinggi dari pola sidik jari. Karena masalah kotor, luka, sobek, dan aus yang dapat dengan mudah mempengaruhi bentuk sidik jari manusia (Alsaadi, 2015). Seperti halnya yang terjadi pada perangkat *fingerprint scanner* yang terdapat di salah satu gedung di Universitas Islam Negeri Sunan Ampel Surabaya. Alat tersebut memerlukan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

waktu yang lama untuk melakukan proses *scan* sidik jari, bahkan tidak dapat digunakan. Kasus lain yang terjadi adalah tidak dapat mendeteksi sidik jari yang sudah aus, meskipun sidik jari yang digunakan di perangkat *fingerprint scanner* pendata kehadiran dapat digunakan pada *fingerprint* yang terdapat di *smartphone*. Hal ini dikarenakan perbedaan sensor yang terdapat di setiap perangkat berbeda-beda, daya tangkap gambar di setiap sensor dipengaruhi oleh *Local Binary Patterns* (LBPs), *Histograms of Oriented Gradients* (HoGs), dan *Gabor Responses* yang berfungsi sebagai *texture descriptors* untuk membedakan gambar/ pola sidik jari yang ditangkap (AlShehri, Hussain, AboAlSamh, & AlZuair, 2018).

Masalah utama dari *fingerprint scanner* konvensional adalah keterbatasan perangkat untuk dijalankan di luar lokasi yang sudah ditentukan, sehingga teknologi ini tidak memiliki mobilitas dalam penggunaanya. Pegawai perusahaan atau instansi ketika ingin melakukan pencatatan kehadiran menggunakan alat ini harus datang ke tempat *fingerprint scanner* yang sudah disediakan, efek yang ditimbulkan adalah saat jam masuk kerja terjadi antrean pendataan kehadiran pada perangkat *fingerprint scanner*. Sehingga diperlukan solusi untuk mengatasi masalah tersebut.

Fingerprint yang terdapat pada *smartphone* memiliki mobilitas yang lebih besar dibandingkan dengan *fingerprint* konvensional, karena *smartphone* dapat digunakan kapanpun dan dimanapun. Penggunaan *fingerprint* pada *smartphone* berfungsi sebagai akses masuk pengguna ketika layar dimatikan, tetapi sekarang sudah banyak implementasi dari penggunaan *fingerprint smartphone* untuk keperluan keamanan data pengguna, contohnya adalah untuk mengizinkan transaksi pada aplikasi finansial (Jo, Jeon, Im, & Lee, 2016). *Fingerprint scanner* yang dimiliki *smartphone* juga dapat digunakan untuk melakukan pendataan kehadiran pegawai, dengan memadukan sensor *Global Positioning System* (GPS) dan *fingerprint* dapat menjadikan sistem pendataan lebih akurat dalam hal mendeteksi lokasi berbasis koordinat pengguna (Soewito, Gaol, Simanjuntak, & Gunawan, 2015).

Penggunaan *Geographic Information Retrieval* (GIR) memberikan solusi dalam pendataan kehadiran pegawai berbasis sensor GPS yang dimiliki oleh *smartphone*. GIR merupakan pengembangan dari *Geographical Information*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

System (GIS) yang beberapa tahun ini populer, penggunaan GIR didasarkan atas pengembangan sistem komputer yang membutuhkan informasi spesifik suatu geografis tertentu dengan sumber data berasal dari seluruh dunia (Jones & Purves, 2008). Sehingga GIR berguna untuk mendapatkan informasi yang dibutuhkan dari sebuah tata letak geografis. Salah satu cara mendapatkan informasi dari GIR adalah dengan menggunakan *georeferencing*. Proses ini memetakan peta digital dengan bersumber dari koordinat nyata wilayah geografis. Hasilnya berupa *cartographic projection* dengan skala koordinat yang sama dengan titik koordinat nyata (Herbei, Ciolac, & Smuleac, 2010).

Ada dua metode yang dapat digunakan untuk memvalidasi koordinat pengguna berada di dalam wilayah poligon, yaitu *crossing number* (cn) dan *winding number* (wn). *Crossing number* digunakan untuk wilayah poligon yang tidak memiliki kompleksitas tinggi, sebaliknya menggunakan *winding number* (Sunday, 2012). Metode *crossing number* memiliki beberapa bagian, salah satunya adalah *ray casting*. Algoritma ini menggunakan garis *ray* dari titik poin ke jarak tak terbatas. Sehingga dihasilkan potongan garis antara poin dengan poligon (Ye, Guangrui, & Shiqi, 2013). Pengembangan algoritma *ray casting* menghasilkan metode *Point Inclusion in Polygon Test* (PNPOLY) yang dibuat oleh W. Randolph Franklin (WRF). Metode ini bekerja dengan menjalankan *semi-infinite ray* secara horizontal (meningkatkan y dan memperbaiki x) dari titik uji dan menghitung berapa banyak sudut yang dilintasi (Franklin, 2006). Menjadikan metode ini hanya menghitung garis *ray* sebanyak sudut yang dimiliki oleh wilayah poligon.

Penggabungan GIR, PNPOLY dan *fingerprint mobile* dapat digunakan sebagai solusi teknologi untuk mempermudah pendataan kehadiran pengguna. Selain itu teknologi ini dapat menjadi *disruptive innovation* dalam hal pencatatan kehadiran pegawai. Berawal dari latar belakang diatas, maka perlu dilakukan penelitian dengan judul **"IMPLEMENTASI *POINT INCLUSION IN POLYGON TEST* UNTUK PENDATAAN KEHADIRAN PEGAWAI BERBASIS BIOMETRIK DI PERANGKAT *MOBILE*"**, sehingga hasil penelitian dapat membangun aplikasi yang mampu menggantikan perangkat *fingerprint* konvensional.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

1.2 Perumusan Masalah

Hasil dari latar belakang diatas, dirumuskan kedalam beberapa poin masalah, antara lain.

1. Bagaimana membuat aplikasi pendataan kehadiran pegawai berbasis perangkat *mobile*?
2. Bagaimana hasil evaluasi metode PNPOLY yang telah diimplementasikan terhadap pengaruh sistem operasi *mobile* (Android dan IOS) dan jaringan (4G, 3G dan 2G) serta *network latency* ketika menggunakan aplikasi?

1.3 Batasan Masalah

Untuk menjaga penelitian dari masalah-masalah yang ada di luar cakupan maka dibuatlah beberapa batasan masalah, sebagai berikut.

1. Perangkat *smartphone* yang digunakan harus mempunyai sensor *fingerprint*, serta pengguna harus sudah mendaftarkan sidik jarinya ke perangkat *smartphone*.
2. Satu perangkat *smartphone* hanya dapat digunakan untuk satu pengguna.
3. Lokasi wilayah yang didaftarkan menjadi poligon untuk perhitungan metode PNPOLY adalah wilayah Universitas Islam Negeri Sunan Ampel Surabaya (UINSA Surabaya) di Jl. A. Yani 117, serta data koordinat diambil dari penyedia layanan data peta dari OpenStreetMap.
4. Aplikasi yang dibuat berbasis aplikasi *mobile* yang hanya dapat dijalankan di perangkat Android (versi 8, 9) dan IOS (versi 12, 13).
5. Penelitian yang dilakukan menggunakan studi kasus UINSA Surabaya.

1.4 Tujuan Penelitian

Penelitian ini dilakukan dengan tujuan berikut.

1. Membuat aplikasi pendataan kehadiran pegawai berbasis perangkat mobile.
2. Mendapatkan hasil evaluasi metode PNPOLY yang telah diimplementasikan terhadap pengaruh sistem operasi *mobile* (Android dan IOS) dan jaringan (4G, 3G dan 2G) serta *network latency* ketika menggunakan aplikasi.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

1.5 Manfaat Penelitian

Manfaat dari penelitian yang dilakukan berdampak luas kepada pihak dan lingkungan dalam bidang terkait, termasuk di dalamnya ada akademik dan aplikatif.

1.5.1 Akademik

1. Mengimplementasikan ilmu pengetahuan yang telah didapat untuk menyelesaikan permasalahan yang dihadapi dalam masyarakat.
2. Mendapat pengetahuan baru dalam mengembangkan metode yang sudah ada ke dalam inovasi aplikasi.
3. Sebagai bahan referensi ilmu pengetahuan pada pengembangan sistem autentikasi biometrik, khususnya menggunakan *fingerprint scanner smartphone* dengan metode PNPOLY.

1.5.2 Aplikatif

1. Dengan adanya aplikasi ini dapat menjadi bahan pertimbangan untuk menggantikan sistem pendataan kehadiran pegawai berbasis pada *fingerprint* konvensional.
2. Memudahkan pegawai untuk melakukan *scan fingerprint* pendataan kehadiran dimanapun, asalkan masih berada di dalam wilayah perusahaan.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

BAB II
TINJAUAN PUSTAKA

2.1 Tinjauan Penelitian Terdahulu

Untuk memperoleh wawasan dan pemahaman, maka dilakukan tinjauan penelitian terdahulu yang relevan dengan penelitian yang akan dilaksanakan nantinya. Berdasarkan penelitian dengan judul "*Attendance System on Android Smartphone*" dengan metode *Fingerprint method with GPS positioning* menyatakan pemanfaatkan teknologi *fingerprint* dan GPS *smartphone* dapat memecahkan masalah yang ada pada sistem pendataan kehadiran. Sistem ini dapat menghindari antrian panjang yang terjadi di depan perangkat *fingerprint* pendataan kehadiran, karena sistem yang dikembangkan menggunakan perangkat *smartphone* yang mempunyai mobilitas tinggi (Soewito et al., 2015).

Penelitian yang dilakukan oleh yang berjudul "*Attendance System Using Android integrated Biometric Fingerprint Recognition*" dengan metode *Fingerprint method with GPS positioning*, menyatakan dengan memadukan GPS dan *fingerprint* kedalam sistem pendataan kehadiran memudahkan sistem untuk mengaudit kehadiran siswa secara *real time* serta membuat *attendance tracking system* lebih efektif (Wadhwa, 2017).

Penelitian ketiga, dengan judul "*A Smart, Location Based Time and Attendance Tracking System Using Android Application*" yang ditulis oleh dengan metode *GPS positioning*, menyatakan bahwa penggunaan lokasi pada pendataan kehadiran pegawai mempermudah *attendance tracking system*. lokasi yang diatur untuk *tracking* menggunakan GPS dan koordinat pegawai di dalam batas wilayah, menggambarkan bahwa karyawan hadir dalam organisasi (Sultana, Enayet, & Mouri, 2015).

Berdasarkan penelitian dengan judul "*Remote Biometric Authentication System using Android Phone*" oleh dengan menggunakan metode *Fingerprint method with local connection (Wi-Fi)*, berhasil menyatakan aplikasi yang dibuat berhasil memberikan keandalan dalam manajemen waktu serta memudahkan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

kontrol pengguna, baik dari segi pegawai maupun *administrator* (S., D., & Waghmare, 2018).

Penelitian selanjutnya, berasal dari dengan judul "*Geographic Information Retrieval Method for Geography Mark-Up Language Data*" menggunakan metode *Geographic Information Retrieval*, menghasilkan pernyataan bahwa tingkat penarikan dan presisi hasil pencarian lebih dari 90%, yang memenuhi persyaratan pengguna, dibandingkan dengan menggunakan metode GML tradisional (Fang & Zhang, 2018).

Penelitian oleh dengan judul "*An Algorithm for Judging Points Inside or Outside A Polygon*" menggunakan metode *Ray Casting*, berhasil menyatakan algoritma baru yang dibuat berhasil mendapatkan hasil yang identik dengan algoritma *ray casting* dari berapa banyak poin yang terdapat di poligon di setiap pengujian (Ye et al., 2013).

Penelitihan terahir dengan judul "*An Extension to Winding Number and Point in Polygon Algorithm*" menggunakan metode *Point Inclusion in Polygon Test* (PNPOLY), menyatakan peningkatan algoritma *axis-crossing* untuk perhitungan disajikan dalam bentuk *winding number*. Penelitian yang dilakukan menunjukan bahwa, masalah yang dihadapi dalam algoritma sebelumnya diselesaikan dengan perbaikan yang dilakukan untuk menangani berbagai jenis poligon tanpa *pre-processing* (Kumar & Bangi, 2018).

Dari beberapa penelitian yang pernah dilakukan, belum ada penelitian tentang implementasi penggunaan *fingerprint smartphone* untuk pendataan kehadiran pegawai menggunakan *Geographic Information Retrieval* (GIR) berbasis metode *Point Inclusion in Polygon Test* (PNPOLY). Penelitian terdahulu dengan judul "*Attendance System on Android Smartphone*", "*Attendance System Using Android integrated Biometric Fingerprint Recognition*" dan "*A Smart, Location Based Time and Attendance Tracking System Using Android Application*" menggunakan metode yang hampir mirip yaitu menggunakan GPS. Penggunaan GPS hanya terbatas pada *tracking* lokasi pegawai ketika melakukan pendataan kehadiran, tanpa menggunakan metode pengecekan lokasi.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Penelitian keempat juga menggunakan perangkat *mobile* dalam mengembangkan metode. Berbeda dengan penelitian yang lain, penelitian ini menggabungkan teknologi *fingerprint mobile* dengan gelombang *Wi-Fi*. Keempat penelitian inilah yang menjadi dasar teori penggunaan *fingerprint mobile* dalam penelitian yang akan dilakukan. Karena kemudahan yang dimiliki, serta mobilitas perangkat yang tinggi.

Karena *fingerprint smartphone* dapat digunakan saat dimana saja dan kapan saja, harus ada indikator untuk membatasi penggunaanya, sehingga metode PNPOLY sangat tepat diterapkan dalam penelitian ini. Metode ini didapatkan dari beberapa penelitian terdahulu yang memuat tentang penggunaan GIR sebagai cara pengambilan informasi geografis wilayah, penelitian tersebut berjudul "*Geographic Information Retrieval Method for Geography Mark-Up Language Data*". Kemudian penelitian yang memuat penggunaan algoritma *ray casting* dengan judul "*An Algorithm for Judging Points Inside or Outside A Polygon*". Serta, penelitian yang menggunakan metode PNPOLY berjudul "*An Extension to Winding Number and Point in Polygon Algorithm*".

Berdasarkan penelitian sebelumnya, metode PNPOLY memiliki keakuratan yang cukup tinggi. Karena metode ini mengoptimalkan penggunaan algoritma *ray casting*, dengan hanya menggunakan *semi-infinite ray* dalam algoritmanya (Franklin, 2006). Sehingga, metode PNPOLY di implementasikan kedalam aplikasi pendataan kehadiran pegawai berbasis *fingerprint mobile* dalam penelitian ini.

2.2 Teori-Teori Dasar

2.2.1 *Geographic Information System* (GIS)

Geographic Information System (GIS) merupakan salah satu informasi yang tersaji dalam bentuk peta geografis. Penggunaan GIS mulai populer ketika diperkenalkannya sistem navigasi menggunakan peta digital. Pada awal mula pengembangannya, GIS digambarkan sebagai perangkat lunak yang menggabungkan teknologi analisa dan kartografi menggunakan data digital (Wood, 2011). Perkembangan dunia digital menjadikan kebutuhan akan teknologi GIS

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

semakin banyak, oleh karena itu lahirlah cabang ilmu informasi geografis yaitu *Geographic Information Retrieval* (GIR).

2.2.2 *Geographic Information Retrieval* (GIR)

Kebutuhan akan informasi geografis yang dapat digunakan semua kalangan dalam pengembangan aplikasi berbasis geografis menjadikan motivasi utama pihak akademisi mengembangkan *Geographic Information Retrieval* (GIR) (Jones & Purves, 2008). GIR memproses data yang dimiliki oleh sistem informasi geografis atau yang lebih dikenal dengan nama GIS, menjadi sebuah informasi yang dapat digunakan untuk keperluan sistem dari sebuah aplikasi. Penggunaan GIR saat ini sudah digunakan di berbagai bidang, mulai dari kesehatan, sosial masyarakat, dan lain sebagainya. Banyak dari peneliti berfokus pada pengembangan aspek individu GIR, contohnya adalah *georeferencing, ranking, indexing or evaluation*. Namun beberapa dekade terakhir, banyak peneliti yang mulai mengembangkan rantai proses GIR dengan tujuan menambah performa dalam penggunaanya (Purves, Clough, Jones, Hall, & Murdock, 2018).

Penggunaan GIR berguna dalam pemetaan peta digital untuk keperluan pengembangan aplikasi. Untuk dapat melakukan pengambilan data geografis dapat menggunakan layanan pihak ketiga, seperti Google Map (Google Inc, 2019) maupun OpenStreetMap (OpenStreetMap Foundation, 2019). Keuntungan penggunaan layanan tersebut adalah memudahkan pengembang aplikasi dalam mengimplementasikan peta digital kedalam aplikasi. Sekarang ini terdapat banyak *library* yang dapat digunakan untuk melakukan fungsi GIR, salah satu diantaranya adalah *library react native map* (Airbnb, 2017). *Library* ini dibuat dari implementasi Google Map ke dalam *framework react native* yang berbasis pada bahasa pemrograman *javascript*.

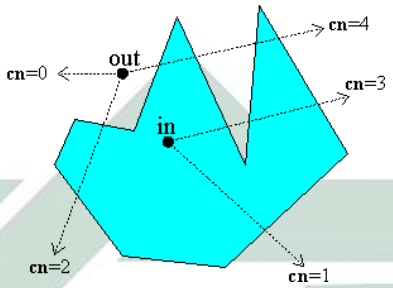
2.2.3 *Inclusion of a Point in a Polygon*

Untuk menentukan valid atau tidak sebuah titik P didalam area poligon untuk menyelesaikan permasalahan geometrik, dapat menggunakan dua metode, yaitu:

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

1. *Crossing number* (cn)

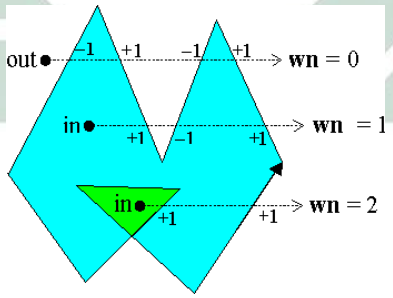
   Metode ini menghitung berapa kali ray melintas dari titik P ke sisi tepi poligon. Jika *crossing number* berada di luar sisi tepi maka bernilai genap, dan sebaliknya. Metode cn digunakan untuk poligon yang tidak memiliki kompleksitas tinggi (Sunday, 2012). Gambar 2.1 menunjukan metode cn.



Gambar 2.1: Metode *crossing number* (cn) (Sunday, 2012)

2. *Winding number* (wn)

   Metode wn Menghitung berapa kali poligon berputar di sekitar titik P. Titik dinyatakan berada di dalam poligon ketika *winding number* (wn) bernilai 0, jika tidak maka titik berada di dalam. Penggunaan wn ditujukan untuk poligon yang memiliki kompleksitas tinggi (Sunday, 2012). Gambar 2.2 menunjukan metode wn.



Gambar 2.2: Metode *winding number* (wn) (Sunday, 2012)

2.2.4 *Point Inclusion in Polygon Test* (PNPOLY)

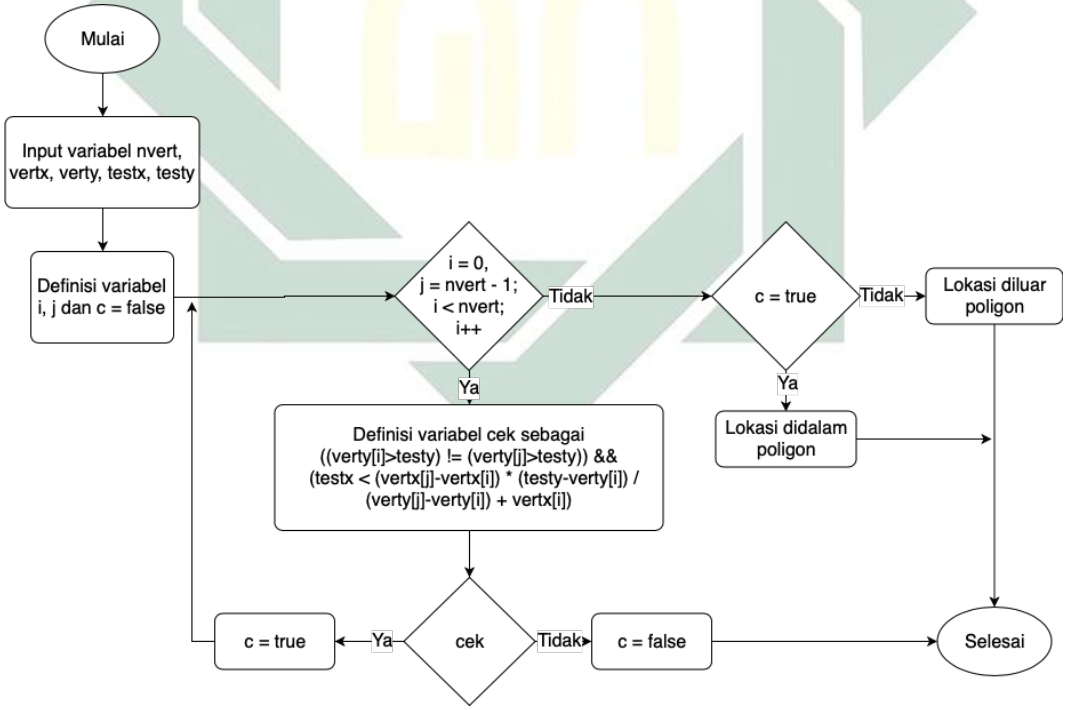
PNPOLY adalah sebuah metode untuk menentukan posisi titik berada dalam sebuah wilayah poligon atau tidak. Metode ini dikembangkan oleh W. Randolph Franklin (WRF) pada tahun 1970. Berikut adalah *pseudo code* metode PNPOLY.

```
1. Mulai dengan mendefinisikan nvert,vertx,verty,testx,testy
2. nvert=banyaknya sudut area poligon
```


digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

```
3. vertx dan verty=array yang berisi koordinat x dan y dari sudut-sudut
   area poligon
4. testx dan testy=koordinat x dan y titik uji
5. Atur i,j,c
6. i=0
7. j=nvert-1
8. c=false
9. Jika i<nvert adalah salah kemudian keluar dari perulangan
10. Jika (verty ke i>testy !=verty ke j>testy)  dan
11. Jika testx<(vertx ke j-vertx ke i)*((testy-verty ke i))/((verty ke j-
    verty ke i) )+vertx ke i adalah benar kemudian
12. c !=c
13. Atur j=i dan kemudian
14. Tambahkan 1 ke i dan kemudian
15. Kembali ke tahap 9
16. Selesai jika c=false atau tahap 9 selesai
```

Pseudo code diatas digambarkan kedalam bentuk alur perhitungan metode PNPOLY yang terlihat pada Gambar 2.3.



Gambar 2.3: *Flowchart* Metode PNPOLY

PNPOLY merupakan pengembangan dari algoritma *ray casting*. Berbeda dengan algoritma *ray casting*, metode ini menjalankan *semi-infinite ray* secara horizontal (meningkatkan y dan memperbaiki x) dari titik uji dan menghitung

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

berapa banyak sudut yang dilintasi. Ketika melintasi sudut tersebut, *ray* akan berpindah di antara posisi *inside* dan *outside*. Perpindahan itu disebut dengan *Jordan Curve Theorem*. Tujuan menguji *ray* pada setiap sudut adalah untuk mengecek beberapa kemungkinan berikut.

- Apakah titik uji berada dalam setengah bidang sebelah kiri dari sisi perpanjangan?
- Apakah titik uji y berada dalam koordinat dari barisan sudut y?

Pengujian tersebut dilakukan terus menerus sampai batas banyaknya sudut yang terdapat pada area poligon. Hasil dari pengecekan menggunakan metode ini berupa *false* dan *true* untuk menyatakan bahwa titik uji berada di dalam area poligon (dinyatakan dengan *true*) atau berada diluar area poligon (dinyatakan dengan *false*) (Franklin, 2006). Metode PNPOLY dinyatakan dalam persamaan:

$$f = \sum_{x=0,\ y=0}^{n,\ m} (y - y_m)(x_n - x_m) > (y_n - y_m)(x - x_m) \quad (1)$$

Dimana:

f : *flag*

x, y : koordinat titik uji

x_n, y_n : koordinat titik potong *ray* dengan sisi tepi poligon

x_m, y_m : koordinat sisi poligon

2.2.5 *Biometric Information System*

Biometrik adalah sebuah teknologi untuk menganalisis dan mengukur data biologis dari individu. Dengan adanya teknologi ini identitas seseorang dapat ditentukan tanpa adanya otoritas atau pengetahuan apapun. Sistem Biometrik menyediakan metode verifikasi *user* dengan lebih baik. *Biometric Verification* adalah sebuah teknik otomatis, dimana identitas seseorang dikonfirmasi melalui keunikan dari karakteristik perilaku atau fisiologis, seperti *palm print, iris* dan *fingerprint* (Malathi & Jeberson Retna Raj, 2016). Pemanfaatan teknologi biometrik sudah sangat banyak ditemukan di berbagai bidang kehidupan saat ini, dan yang paling banyak digunakan adalah teknologi *fingerprint*. Hampir semua perangkat *smartphone* yang beredar di *marketplace* sekarang sudah dilengkapi

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

dengan sensor *fingerprint* untuk memudahkan dan meningkatkan efektivitas penggunaan.

2.2.6 *Fingerprint*

Fingerprint merupakan salah satu bentuk *Biometric Authentication* menggunakan sidik jari yang terdapat pada setiap individu. *Fingerprint* bersifat unik dan tidak sama dengan yang lain, sebagai contoh dari sepuluh jari yang ada di tangan manusia terdapat sidik jari yang berbeda-beda di setiap jarinya, oleh karenanya *fingerprint* berperan besar dalam teknologi biometrik di dunia modern (Wadhwa, 2017). Teknologi ini banyak digunakan dalam alat pendataan kehadiran pegawai pada perusahaan maupun instansi menggunakan perangkat khusus *fingerprint scanner*, akan tetapi sekarang ini *fingerprint* sudah banyak ditemukan pada perangkat *smartphone* yang beredar di pasaran. Penggunaannya ditujukan sebagai salah satu media keamanan data dan informasi dari perangkat *smartphone*.

2.2.7 *Smartphone*

Perkembangan teknologi dan informasi menjadikan kebutuhan akan sarana pendukung sangat diperlukan oleh setiap orang. Pada beberapa dekade terakhir permintaan masyarakat akan perangkat *mobile* sangat besar, hal ini menjadikan *smartphone* sangat laku di pasaran. *Smartphone* digunakan untuk menggantikan kamera digital, jam, perekam video, dan masih banyak lagi. Mempunyai perangkat ini seperti memiliki sebuah komputer kecil yang dapat masuk kedalam kantong. Dengan didukung kecanggihan teknologi internet serta aplikasi pendukung di dalamnya, menjadikan *smartphone* tidak hanya digunakan untuk melakukan panggilan suara, akan tetapi dapat juga digunakan untuk menjelajah internet, seperti halnya mengirim email, *chatting*, *sharing* foto dan dokumen, membaca berita, membuka situs internet, dan berbelanja *online* (Al Fawareh & Jusoh, 2017).

2.2.8 *Database*

Di era teknologi informasi, kebutuhan akan data dan informasi sangat penting. Komponen penyimpanan data terpusat disebut dengan *database*. Dalam perkembangan *database* dibagi menjadi dua bagian, yaitu *traditional database* dan

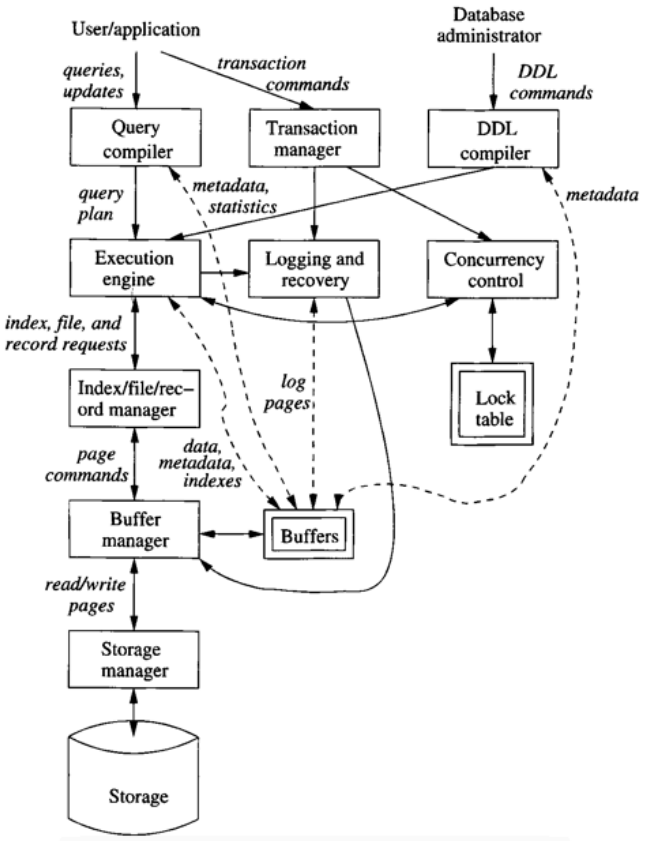
digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

modern database. *Database* lama menyimpan data yang dapat diakses dengan cara *textual* atau *numeric*. Teknologi *database* yang baru memungkinkan untuk menyimpan gambar, suara, video. Jenis file ini menjadi komponen penting dari multimedia *database* (Foster, Godbole, Foster, & Godbole, 2016). Dalam *database* terdapat teknologi yang dikenal dengan nama *Database Management System* (DBMS) seperti yang ditunjukan pada Gambar 2.4.

Teknologi DBMS berguna untuk mengelola data dan informasi dari sebuah *database*. Pengelolaan *database* sendiri dapat dilakukan oleh beberapa *user* terkait, antara lain *administrator, designers* dan *end users*. *Administrator* mempunyai hak tertinggi dari sebuah *database*, tugasnya adalah memelihara DBMS dan mengelola *database*. *Administrator* dapat membatasi pengguna yang lain untuk menjaga keamanan *database*. Tugas lainnya adalah mengurus sumber daya DBMS, seperti lisensi sistem, pemeliharaan alat yang diperlukan, dan pemeliharaan *software* maupun *hardware* yang terkait. Pengguna kedua adalah *designers*, merupakan bagian kelompok yang berkerja untuk merancang desain dari *database*, termasuk merancang rangkaian *entities, relations, contraints* dan *views*. Terakhir adalah *end users*, pengguna ini adalah pengguna biasa dari sebuah sistem yang mendapatkan manfaat sebenarnya dari DBMS yang ada. *End users* dapat diartikan sebagai pengguna sederhana, yang hanya menggunakan aplikasi atau sistem, maupun pengguna ahli seperti bisnis analis (Garcia-Molina, Ullman, & Widom, 2009).

DBMS menggungakan perintah SQL untuk mengeksekusi proses yang dilakukan oleh *user*. Perintah SQL dibagi kedalam 2 macam, yaitu *Data Manipulation Language* (DML), dengan *statement* didalmnya termasuk *SELECT, INSERT, UPDATE* dan *DELETE*, serta *Data Definition Language* (DDL). *Statement* DDL adalah *CREATE TABLE* dan *CREATE INDEX* (Garcia-Molina et al., 2009). Pembuatan sebuah database perlu melewati tahap *data warehouse life cycle*, di dalamnya terdapat *requirements analysis and specification*, *logical database design*, *physical database design*, *data distribution*, dan yang terakhir adalah database *implementation, monitoring, and modification* (Garcia-Molina et al., 2009).

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id



Gambar 2.4: Komponen DBMS (Garcia-Molina et al., 2009)

2.2.9 *Javascript*

Beberapa dekade terakhir, *javascript* menjadi bahasa pemrograman yang sangat populer, dengan didukung banyaknya *library* dan *framework* yang dapat digunakan, serta dukungan dari berbagai komunitas pengembang yang melakukan *update* dalam jangka waktu relatif cepat. Melihat sejarahnya, *javascript* pertama kali dibuat pada tahun 1995 dalam waktu 10 hari oleh Brendan Eich. Ketika dia bekerja di *Netscape*, dan mengimplementasikan *javascript* untuk *web browser* mereka, yaitu *Netscape Navigator* (Rauschmayer, 2019).

Bahasa *javascript* juga menganut penggunaan *Object Oriented Programming* (OOP) untuk memudahkan *function* dalam mengeksekusi sebuah proses. Bahasa ini juga dapat mengeksekusi *values* dengan tipe yang berbeda-beda, diantaranya *numbers*, *strings*, *booleans* maupun *undefined values* (Garcia-Molina et al., 2009). Penggunaan javascript sekarang ini fokus dalam pengembangan aplikasi berbasis *web browser*. Menggabungkan teknologi HTML dan CSS sebagai *styling* nya dapat

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

membuat aplikasi berbasis *web* dapat berfungsi maksimal. Pengembangan *jvascript* dalam teknologi *website* mengubah sistem *Model View Controller* (MVC) menjadi sebuah *Single Page Application* (SPA) dengan bantuan *framework* yang berbasis *javascript*. Dengan bantuan *framework* tersebut, bahasa *javascript* juga dapat digunakan dalam teknologi *desktop* maupun *mobile*. Aplikasi *mobile* berbasis *javascript* dikembangkan melalui *framework react native*. *Framework* ini dapat mengkonversikan baris kode *javascript* kedalam bentuk kode pengembangan aplikasi Android dan IOS (Facebook Inc, 2019).

2.2.10 *Framework React Native*

Teknologi pemrograman berbasis *javascript* mulai berkembang pesat beberapa tahun terakhir. Karena bahasa ini merupakan salah satu bahasa pemrograman yang cocok digunakan untuk *Machine Learning* di era data sekarang. Penggunaan *javascript* dahulu ketika awal era internet digunakan untuk membuat aplikasi berbasis *website*, akan tetapi sekarang sudah banyak bermunculan *framework* berbasis *javascript* yang mengubah fungsi lama, yaitu untuk pembuatan *web* menjadi membuat aplikasi *mobile*. Salah satu *framework* yang banyak digunakan oleh para *developer* aplikasi *mobile* saat ini adalah *react native*. *Framework* ini merupakan bagian dari *framework react js* yang digunakan untuk membuat aplikasi *web*. Keuntungan menggunakan *framework react native* adalah dapat membuat aplikasi Android dan IOS dengan menggunakan baris kode yang sama (Facebook Inc, 2019), hal ini lah yang jarang ditemukan di *framework* lain.

2.2.11 *PHP*

Dalam sejarahnya, PHP pertama kali dibuat pada tahun 1995 oleh seorang *developer* yang bernama Rasmus Lerdorf ketika ingin membuat *script* dari *Perl/ CGI* yang membuat dia dapat mengetahui berapa banyak pengunjung yang membaca resume *online* miliknya (Gilmore, 2010). Bahasa pemrograman ini berjalan di atas *web server* yang memungkinkan untuk mengeksekusi program PHP. Ada 2 *web server* yang sering digunakan, yaitu *Apache* dan *Microsoft Internet Information Services* (IIS). Dan kedua layanan tersebut mendukung penggunaan bahasa pemrograman PHP. Dahulu PHP digunakan untuk membuat *website* secara

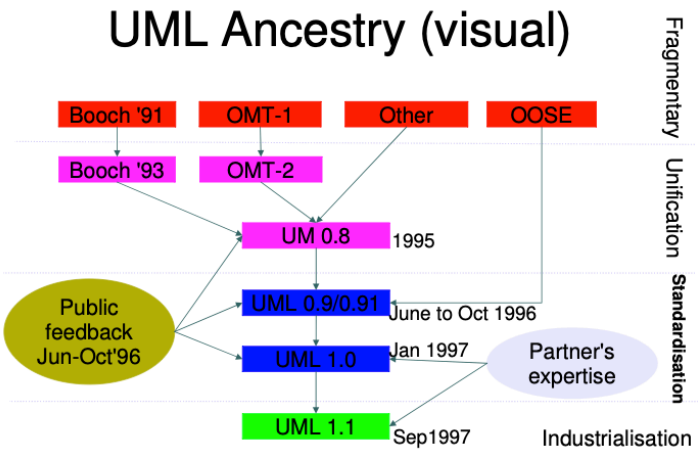
digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

keseluruhan, dengan memadukan HTML, JS dan CSS. Akan tetapi penggunaan PHP sekarang lebih difokuskan untuk menangani *back-end* dari sebuah aplikasi *web*, seperti halnya membuat *RESTful API* ataupun untuk berkomunikasi kedalam *database*.

2.2.12 *Unified Modeling Language* (UML)

Sebuah model untuk memvisualkan rancangan pengembangan sistem dibuat kedalam bentuk UML. UML merupakan bahasa standar untuk memvisualkan, menentukan dan membangun dari sebuah dokumentasi pengembangan sistem perangkat lunak (B'Far, Fielding, Brady, B'Far, & Fielding, 2009). UML memungkinkan untuk menggambarkan rancangan pengembangan sistem dengan teks dan gambar. Teknologi ini dapat memodelkan berbagai sistem seperti sistem perangkat lunak, sistem bisnis, maupun sistem yang lain.

Dalam sejarahnya, UML merupakan penggabungan Bahasa permodelan dari seluruh dunia yang distandarisasi oleh *Object Management Group* (OMG) (Grässle, Baumann, & Baumann, 2005). Sejarah pengembangan UML dapat dilihat pada Gambar 2.5.



Gambar 2.5: Sejarah UML (Cachia, 2004)

Konsep UML dikembangkan kedalam beberapa tampilan diagram sebagai permodelan sebuah konstruksi yang mewakili beberapa rancangan sistem yang dibuat (James Rumbaugh, Ivar Jacobson, 2013). Tampilan tersebut dibagi menjadi beberapa bagian, diantaranya adalah.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

a. *Structural Classification.*

Model ini mendeskripsikan hubungan antara bagian-bagian yang terdapat didalam sistem dengan hal-hal lain yang terkait, yang termasuk kedalam bagian ini adalah *classes, use case, compinents* dan *nodes*.

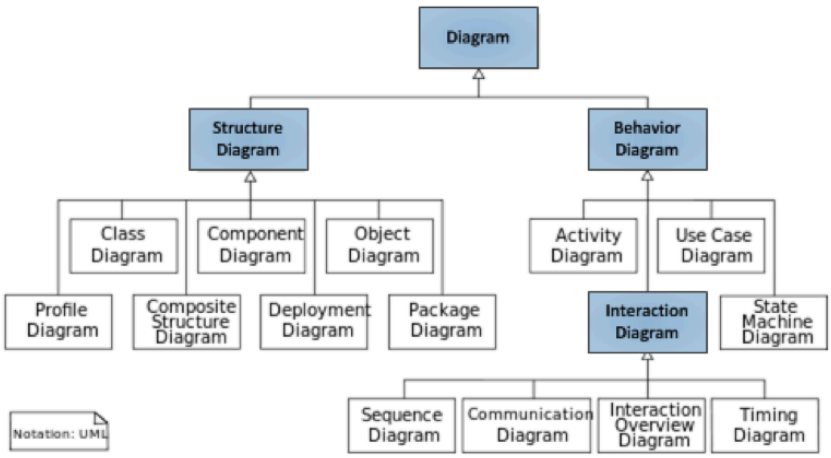
b. *Dynamic Behavior.*

Bagian ini menggambarkan perilaku sistem dari waktu ke waktu. Perilaku sistem dideskripsikan dengan serangkaian perubahan pada *snapshots* dari pengambilan tampilan *static* sebuah sistem. Tampilan *dynamic behavior* meliputi *state machine view, activity view* dan *interaction view*.

c. *Model Management.*

Pada bagian ini mendeskripsikan organisasi dari model terkait kedalam unit hierarki. Tampilan *model management* melintasi tampilan lain dan mengaturnya untuk keperluan pengembangan dan kontrol konfigurasi.

Sumber lain menyebutkan pembagian tampilan UML dibagi menjadi 2 bagian yaitu, *structure diagram* dan *dynamic diagram*. *Dynamic diagram* sendiri merupakan gabungan dari *behavior diagram* dan *interaction diagram* (Padmanabhan, 2012). Setiap pengembangan sistem perangkat lunak diperlukan rancangan UML untuk mempermudah dokumentasi pekerjaan. UML yang dibuat harus mencakup kedua jenis, yaitu bentuk *structure* dan *dynamic*. Pada Gambar 2.6 dijelaskan pembagian struktural diagram UML.



Gambar 2.6: Tampilan Diagram UML (Padmanabhan, 2012)

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

2.2.13 *Mock Location*

Salah satu fitur tersembunyi di *developer option* yang dimiliki oleh perangkat Android adalah *mock location*. Fungsi ini berguna untuk mengsimulasikan penggunaan GPS, menghasilkan nilai palsu dari sebuah titik koordinat. *Mock location* di desain untuk *Location-Based Service (LBS) application developer* yang dijalankan pada perangkat tanpa modul GPS (Chang, 2018). Tetapi dalam implementasinya fitur *mock location* ini banyak disalah gunakan untuk menentukan lokasi palsu dari sebuah aplikasi yang berbasis pada peta digital. Untuk mengurangi hal tersebut, banyak *developer* yang mengembangkan *library* untuk mendeteksi penggunaan *mock location*, salah satunya adalah *library* yang terdapat pada *framework react native*.

2.2.14 *Network Latency*

Latency dari hasil komunikasi antara jaringan satu dengan yang lain didefinisikan sebagai waktu yang dibutuhkan untuk mengirim informasi dari pengirim ke penerima. Salah satu cara untuk mengukur *network latency* dapat menggunakan Round-Trip-Time (RTT) (ESOA, 2019). *Latency* berpengaruh terhadap penggunaan aplikasi yang memanfaatkan jaringan internet, daftar pengaruh *latency* terhadap *user experience* penggunaan aplikasi dimuat kedalam Gambar 2.7.

Fibre Cellular Public Wi-Fi Satellite

Application	Sensitivity to Latency	Mitigation Techniques	Likely Transmission Medium
Television	Low	Some	
SCADA, and other telematics applications	Low	Some	
Streaming services	Low	Many	
Over-the-Air (OTA) updates	Low	Some	
Internet browsing	Medium	Many, very effective	
Encrypted Internet Browsing	Medium	Few	
Voice & Videoconferencing	Medium	Few	
Cloud-computing & ERP	Medium - High	Some	
High-frequency trading	Extreme	Few	
V2V and V2X	Depends on application	Few	
Mobility connectivity (In-flight/maritime/Cars /train)	Depends on application	Many	
IoT	Depends on application	Very Few	

Gambar 2.7: Daftar Pengaruh *Network Latency* (ESOA, 2019)

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

2.2.15 *Test Case Scenario*

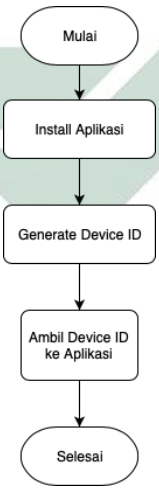
Skenario pengujian merupakan salah satu tahap pada pengujian *black box* dengan tujuan mengilustrasikan kinerja suatu aplikasi. Di dalamnya ada beberapa tujuan, antara lain.

- Mempelajari produk yang dibuat.
- Menghubungkan pengujian ke *document requirement*.
- Melihat kegagalan sistem.
- Mempelajari jalanya sistem, untuk menguji fungsionalitas aplikasi terhadap *system requirement* yang dibuat.

(Kaner, 2003)

2.2.13 *Device ID*

Device id merupakan *unique id* yang diambil dari perangkat Android maupun IOS, salah satu cara untuk mengambil *device id* adalah menggunakan *library framework react native*. Pengembangan react native terbaru mengganti nama *device id* menjadi *installation id*, meskipun begitu hasil yang didapat tetap sama. Penggunaan nama *device id* masih tetap digunakan sebagai alias dari fungsi yang baru (Expo, 2018). Alur pengambilan *device id* dimuat kedalam Gambar 2.8.



Gambar 2.8: Diagram Alur Pengambilan *Device ID*

Device id dibuat ketika aplikasi di *install*, dan akan berubah saat pengguna melakukan *reinstall* aplikasi. Tetapi id yang dihasilkan dari proses *generate device id* akan berbeda setiap aplikasi maupun setiap perangkat Android dan IOS.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

2.3 Integrasi Keilmuan

Pendataan pegawai merupakan bagian dari layanan *Human Resource*. Data yang didapat dari pendataan dapat digunakan untuk mencatat riwayat kehadiran pegawai, sehingga memudahkan pimpinan untuk membuat laporan kinerja serta mempermudah dalam menentukan gaji karyawan. Semakin disiplin dalam bekerja pegawai akan mendapat upah yang sebanding dengan pekerjaannya.

Berdasarkan hasil wawancara yang dilakukan dengan pakar bernama KH Mohammad Khusnu Milad, M.MT selaku pengasuh pondok pesantren Darul Mukhlasin, Probolinggo dan Drs. KH. Sumarkan, M.Ag selaku Imam Masjid Surabaya, mengatakan banyak sekali firman Allah SWT tentang kedisiplinan. Ada 3 surah terkait hal itu, yang pertama adalah surah Al-'Asr ayat 1-3, kemudian surah Al-Ma'idah ayat 1, dan surah An-Nisa' ayat 58. Dalam Al-Qur'an surah Al-'Asr ayat 1-3 dijelaskan sebagai berikut:

وَالْعَصْرِ ۙ - ﴿١﴾ اِنَّ الْاِنْسَانَ لَفِيْ خُسْرٍ ۙ - ﴿٢﴾ اِلَّا الَّذِيْنَ اٰمَنُوْا وَعَمِلُوا الصّٰلِحٰتِ وَتَوَاصَوْا
بِالْحَقِّ ۙ وَتَوَاصَوْا بِالصَّبْرِ - ﴿٣﴾

Artinya:

"Demi masa, sungguh, manusia berada dalam kerugian, kecuali orang-orang yang beriman dan mengerjakan kebajikan serta saling menasihati untuk kebenaran dan saling menasihati untuk kesabaran. (QS. Al-'Asr [103] : 1-3)"

Ayat diatas menjelaskan bahwa setiap orang beriman harus senantiasa menghargai waktu. Maksudnya adalah manusia dituntut untuk memanajemen waktu dengan baik, jangan menunda ataupun meninggalkan pekerjaan. Iman dari segi *dalalah* berarti aman dan dipercayai, jika seseorang sudah meninggalkan kewajiban pekerjaan yang sudah dibuat, itu merupakan salah satu bentuk kurangnya iman mereka. Manusia memiliki tanggung jawab individu untuk beriman dan beramal sholeh serta memiliki tanggung jawab kepada masyarakat untuk saling menasehati dalam hal mentaati kebenaran dan kesabaran (Ramadhani, 2018).

Tafsir diatas apabila diintegrasikan dengan penelitian ini adalah pentingnya kedisiplinan dalam suatu pekerjaan. Salah satu cara melihat kedisiplinan seseorang dapat dilakukan dengan melihat daftar hadir mereka, semakin banyak daftar

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

keterlambatan maupun ketidak hadiran, dapat berpotensi buruk dalam kedisiplinan orang tersebut. Karena atasan menilai mereka sebagai orang yang tidak menghargai waktu. Di dalam surah Al-Maidah ayat 1 terdapat penjelasan tentang perjanjian, seperti berikut.

يٰٓاَيُّهَا الَّذِيْنَ اٰمَنُوْٓا اَوْفُوْا بِالْعُقُوْدِۗ اُحِلَّتْ لَكُمْ بَهِيْمَةُ الْاَنْعَامِ اِلَّا مَا يُتْلٰى عَلَيْكُمْ غَيْرَ مُحِلِّى الصَّيْدِ وَاَنْتُمْ حُرُمٌۗ اِنَّ اللّٰهَ يَحْكُمُ مَا يُرِيْدُ - ١

Artinya:

"Wahai orang-orang yang beriman! Penuhilah janji-janji. Hewan ternak dihalalkan bagimu, kecuali yang akan disebutkan kepadamu, dengan tidak menghalalkan berburu ketika kamu sedang berihram (haji atau umrah). Sesungguhnya Allah menetapkan hukum sesuai dengan yang Dia kehendaki. (QS. Al-Ma'idah [5] : 1)"

Aqad yang dimaksud dalam surah tersebut adalah perjanjian yang dilakukan. Contohnya adalah perjanjian dari pegawai kepada perusahaan mengenai hak dan kewajiban. Salah satu perjanjian itu adalah datang tepat pada waktunya.

Orang disiplin menjalankan kewajiban yang diberikan kepadanya. Salah satu contohnya adalah dalam pegawai datang tepat waktu di perusahaan, merupakan salah satu pemenuhan kewajiban dari pegawai. Jika dia melanggarnya maka akan mendapat sanksi yang sesuai. Di Al-Qur'an surah An-Nisa' ayat 58 berisi tentang penyampaian amanat (kewajiban), sebagai berikut.

اِنَّ اللّٰهَ يَأْمُرُكُمْ اَنْ تُؤَدُّوا الْاَمٰنٰتِ اِلٰٓى اَهْلِهَاۙ وَاِذَا حَكَمْتُمْ بَيْنَ النَّاسِ اَنْ تَحْكُمُوْا بِالْعَدْلِۗ اِنَّ اللّٰهَ نِعِمَّا يَعِظُكُمْ بِهٖۗ اِنَّ اللّٰهَ كَانَ سَمِيْعًاۢ بَصِيْرًا - ٥٨

Artinya:

"Sungguh, Allah menyuruhmu menyampaikan amanat kepada yang berhak menerimanya, dan apabila kamu menetapkan hukum di antara manusia hendaknya kamu menetapkannya dengan adil. Sungguh, Allah sebaik-baik yang memberi pengajaran kepadamu. Sungguh, Allah Maha Mendengar, Maha Melihat. (QS. Al-Nisa' [4] : 58)"

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Pendataan kehadiran dalam dunia modern sudah menggunakan sistem yang terstruktur untuk memudahkan pekerjaan. Banyak perusahaan sudah menggunakan *fingerprint scanner* untuk melakukan pendataan kehadiran pegawai. Akan tetapi penggunaan sistem ini terdapat beberapa kendala, seperti yang sudah dijelaskan dalam latar belakang penelitian ini. Sehingga untuk dapat mendata kehadiran dengan lebih maksimal dibuatlah penelitian dengan judul implementasi *Point Inclusion in Polygon Test* dalam aplikasi pendataan kehadiran pegawai berbasis perangkat *mobile*.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

BAB III
METODOLOGI PENELITIAN

3.1 Desain Penelitian

Alur rancangan penelitian dideskripsikan ke dalam bentuk diagram alur. Diagram ini dipilih dengan tujuan agar mempermudah penyampaian informasi dari alur penelitian. Pada penelitian implementasi *Point Inclusion in Polygon Test* (PNPOLY) pada aplikasi pendataan kehadiran pegawai berbasis perangkat *mobile* menggunakan metode pengembangan dari *Research and Development* (R&D). Metode yang digunakan disesuaikan dengan penelitian yang dilakukan. Berikut adalah alur metode dari penelitian skripsi yang disajikan pada Gambar 3.1.

Gambar 3.1: Diagram Alur Penelitian Skripsi

Berikut adalah penjelasan dari diagram alur metodologi penelitian yang telah tergambarkan pada Gambar 3.1 diatas.

3.1.1 Perumusan Masalah

Pada tahap ini terdapat proses perumusan masalah yang menjadi latar belakang untuk membuat penelitian implementasi *Point Inclusion in Polygon Test* (PNPOLY) pada aplikasi pendataan kehadiran pegawai berbasis *mobile*, sesuai dengan rumusan yang tertera dalam bab pertama di poin latar belakang. Masalah yang diangkat pada penelitian ini adalah pemanfaatan teknologi *fingerprint* untuk pendataan kehadiran pegawai menggunakan perangkat *smartphone*. Dengan menggunakan metode *Point Inclusion in Polygon Test* (PNPOLY) untuk mengecek lokasi pengguna ketika akan melakukan *fingerprint* melalui *smartphone*, menjadikan sistem pendataan kehadiran lebih mudah dan akurat.

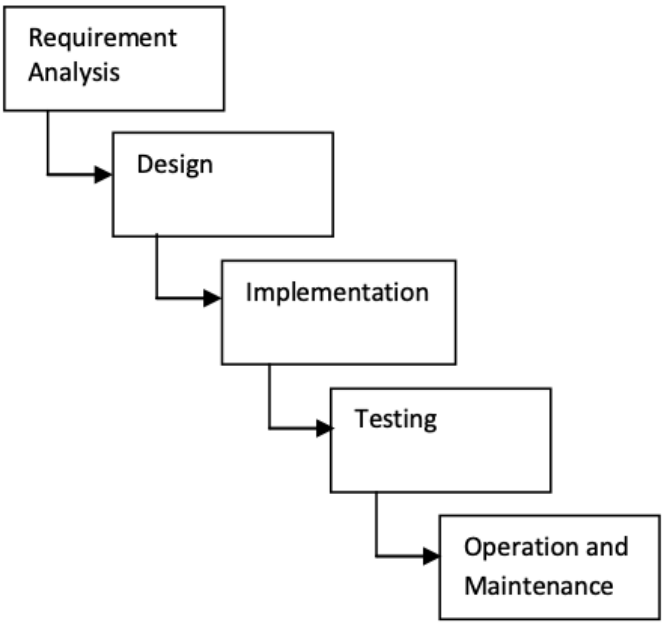
digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

3.1.2 Studi Pustaka

Tahapan kedua adalah untuk melakukan studi pustaka yakni melakukan analisis kebutuhan sistem lama dan sistem *fingerprint* yang baru. Didapatkan hasil untuk pengembangan *fingerprint* pada tahapan yang lebih maju, digunakanlah perangkat *mobile*. Akan tetapi sistem pendataan kehadiran yang menggunakan perangkat *mobile* juga masih belum maksimal, oleh karena itu digunakanlah metode PNPOLY untuk memaksimalkan penggunaan teknologi *fingerprint*. Metode ini dipilih berdasarkan penelitian terdahulu yang pernah dilakukan, menghasilkan keakuratan data yang baik.

3.1.3 Pengembangan Aplikasi

Setelah dilakukanya analisis yang mendalam, tahap selanjutnya adalah mengembangkan aplikasi yang tepat. Pengembangan aplikasi menggunakan model *waterfall*, berikut alur pengembangan yang dilakukan terlihat seperti Gambar 3.2 berikut.



Gambar 3.2: Diagram Alur Pengembangan Aplikasi (Adetokunbo & Basirat, 2014)

Setiap tahapan yang terdapat pada model *waterfall* memiliki tujuan berbeda antara satu dengan yang lainya. Tahapan berikutnya dapat dikerjakan setelah benar-

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

benar menyelesaikan tahap sebelumnya, karena model ini tidak ada kesempatan untuk kembali ke tahap sebelumnya, seperti yang digambarkan pada Gambar 3.2 di atas. Sehingga model ini mendukung pendekatan terstruktur dan berpusat pada proses (Adetokunbo & Basirat, 2014). Rincian alur yang terdapat pada Gambar 3.2 sebagai berikut.

1. Tahap pertama adalah *requirement of analysis*, dalam tahap ini dilakukan analisis kebutuhan pembuatan sistem untuk mengetahui batasan pembuatan aplikasi. Informasi didapatkan dari perumusan latar belakang dan analisa penelitian terdahulu untuk mendapatkan data kebutuhan sistem. Untuk menggunakan aplikasi diperlukan spesifikasi minimum perangkat menggunakan Android 8 atau IOS 12 serta mempunyai sensor *fingerprint*. Pengembangan aplikasi menggunakan bahasa pemrograman *javascript* dengan *framework react native*. Dalam penggunaan aplikasi, *user* dibatasi untuk hanya dapat menggunakan menu *register device* dan menu *home*, serta tidak dapat menggunakan aplikasi *mock location*.
2. Tahap *design* berisi pembuatan arsitektur dari hasil analisis yang dilakukan. Beberapa desain yang dibuat dalam pengembangan aplikasi ini adalah desain arsitektur, desain alur penggunaan aplikasi, desain UML, dan desain basis data. Dari beberapa diagram UML yang harus dimiliki oleh sistem seperti yang ditunjukan pada Gambar 2.6, pengembangan aplikasi ini menggunakan 4 desain UML, antara lain *UML Class Diagram, UML Use Case Diagram, UML Activity Diagram* dan *UML Sequence Diagram*. Sedangkan, desain basis data yang dibuat berupa *Entity Relationship Diagram* (ERD), *Conceptual Data Model* (CDM) dan *Physical Data Model* (PDM).
3. Tahap selanjutnya adalah *implementation*, pada tahap ini dilakukan pembuatan program berdasarkan dengan desain yang terlah dibuat. Aplikasi yang dikembangkan berbasis *mobile* dengan menggunakan *framework react native*. *Framework* ini dipilih karena memudahkan untuk membuat aplikasi Android dan IOS dalam waktu bersamaan. Dari segi *back-end*, digunakan PHP untuk memudahkan *react native* dalam melakukan eksekusi data.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Komunikasi antara *front-end* dan *back-end* menggunakan teknologi *RESTful API*, teknologi ini mengirim request data berupa *JSON*.

4. Setelah aplikasi sudah dikembangkan, langkah berikutnya adalah melakukan *testing* fungsionalitasnya untuk mengecek keberhasilan aplikasi dalam mengeksekusi fungsi. *Test case scenario* dibuat sebagai daftar pengujian yang dilakukan terhadap fungsi aplikasi, sebagaimana yang ditunjukan pada Tabel 3.1.

Tabel 3.1: *Test Case Scenario* Pengujian Fungsionalitas Aplikasi

No.	***Timestamp***	***Untitled Question***	***Pre-Condition***	***Test Step***	***Expected Result***
1.	*Check device*	*Valid*	• Terhubung dengan jaringan *internet* • DeviceId perangkat sudah tersimpan pada *database*	1. Buka aplikasi 2. Tunggu *splash screen* berhasil dimuat	Tampilan dialihkan ke *home screen*
		Invalid	• Tidak mempunyai sensor *fingerprint*	1. Buka aplikasi 2. Tunggu *splash screen* berhasil dimuat	Tampil *pop up* pemberitahuan bahwa perangkat *smartphone* tidak mempunyai sensor *fingerprint*
		Invalid	• Terhubung dengan jaringan *internet* • DeviceId perangkat belum tersimpan pada *database*	1. Buka aplikasi 2. Tunggu *splash screen* berhasil dimuat	Tampilan dialihkan ke *register screen*
		Invalid	• Tidak terhubung dengan *internet*	1. Buka aplikasi	Tampilan dialihkan ke *register screen*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Tabel 3.1: *Test Case Scenario* Pengujian Fungsionalitas Aplikasi lanjutan

No.	***Timestamp***	***Untitled Question***	***Pre-Condition***	***Test Step***	***Expected Result***
2.	*Register device*	*Valid*	• Terhubung dengan *internet* • Mengisi *form* NIP • NIP terdapat pada *database employee*	1. Masukan NIP secara benar 2. Tekan tombol "Daftar" 3. Lakukan *fingerprint* ketika muncul *pop up* pemberitahuan	Muncul *pop up* informasi sukses, kemudian tampilan dialihkan ke *home screen*
		Invalid	• Terhubung dengan jaringan *internet* • Mengisi *form* NIP	1. Masukan NIP secara *random* dengan panjang kurang dari 7 karakter 2. Tekan tombol "Daftar" 3. Lakukan *fingerprint* ketika muncul *pop up* pemberitahuan	Muncul *pop up* pemberitahuan *register* gagal dilakukan karena NIP kurang dari 7 karakter
		Invalid	• Terhubung dengan jaringan *internet* • Mengisi *form* NIP • NIP tidak terdapat pada *database employee*	1. Masukan NIP secara *random* dengan panjang lebih dari 7 karakter 2. Tekan tombol "Daftar" 3. Lakukan *fingerprint* ketika muncul *pop up* pemberitahuan	Muncul *pop up* pemberitahuan *register* gagal dilakukan karena NIP tidak ditemukan dalam *database employee*
		Invalid	• Tidak terhubung dengan jaringan *internet*	1. Masukan NIP dengan benar atau *random* 2. Tekan tombol "Daftar" 3. Lakukan *fingerprint* ketika muncul *pop up* pemberitahuan	Tombol "Daftar" akan tetap menjadi ikon *loading*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Tabel 3.1: *Test Case Scenario* Pengujian Fungsionalitas Aplikasi lanjutan

No.	***Timestamp***	***Untitled Question***	***Pre-Condition***	***Test Step***	***Expected Result***
3.	*Shows Map*	*Valid*	• Terhubung dengan *internet*	1. Buka *home screen* 1. Tunggu sampai muncul tampilan *map*	Tampil peta digital
		Invalid	• Tidak terhubung dengan *internet*	2. Buka *home screen*	Peta digital tidak dapat ditampilkan, dan berwarna abu-abu
4.	*GPS Position*	*Valid*	• GPS dinyalakan	3. Buka *home screen* 4. Tunggu sampai muncul tampilan *map* 5. Gunakan aplikasi dengan berjalan dalam beberapa langkah	Tampil posisi pengguna pada peta digital, posisi pengguna bergerak mengikuti langkah
		Invalid	• GPS dimatikan	1. Buka *home screen*	Muncul *pop up* pemberitahuan untuk menyalakan GPS
5.	*Mock Location*	*Valid*	• Menjalankan aplikasi *mock location*	1. Jalankan aplikasi *mock location* 2. Atur koordinat awal berada di UIN Sunan Ampel Surabaya 3. Buka aplikasi pendataan kehadiran berbasis *mobile* 4. Masuk ke *home screen* 5. Tunggu sampai tampil pemberitahuan	Tampil pemberitahunan bahwa lokasi terdeteksi menggunakan *mock location* dan tombol *fingerprint* dinonaktifkan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Tabel 3.1: *Test Case Scenario* Pengujian Fungsionalitas Aplikasi lanjutan

No.	***Timestamp***	***Untitled Question***	***Pre-Condition***	***Test Step***	***Expected Result***
6.	*Fingerprint Attendance*	*Valid*	• *Fingerprint ID* perangkat sudah diatur sebelumnya • Berada di dalam wilayah poligon • Terhubung dengan *internet*	1. Buka aplikasi 2. Tekan ikon *fingerprint* 3. Taruh jari pada tempat *fingerprint scanner* perangkat *mobile* 1. Tunggu sampai data tersimpan	Muncul *pop up* pemberitahuan sukses, lalu tampil *attendance time* pada *home screen*
		Invalid	• *Fingerprint ID* perangkat sudah diatur sebelumnya • Berada di dalam wilayah poligon • Tidak terhubung dengan *internet*	2. Buka aplikasi 3. Tekan ikon *fingerprint* 4. Taruh jari pada tempat *fingerprint scanner* perangkat *mobile*	Ikon *fingerprint* tetap berubah mendaji *loading* dan data tidak dapat tersimpan

5. Langkah terakhir adalah *operation and maintenance*, pada langkah ini dilakukan perbaikan *bug* yang ditemukan ketika dilakukan *testing* untuk mendapatkan hasil aplikasi yang dapat berjalan secara optimal.

3.1.4 Rancangan Pengujian

Tahap terakhir adalah melakukan evaluasi implementasi metode. Data didapatkan dari hasil implementasi di Universitas Islam Negeri Sunan Ampel Surabaya (UINSA Surabaya), dengan menggunakan lokasi yang berbeda-beda. Daftar lokasi tersebut terdapat pada Tabel 3.2.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Tabel 3.2: Daftar Lokasi Uji Coba Aplikasi

No.	Lokasi	Tipe Pengujian
1.	Fakultas Sains dan Teknologi (Rektorat Lama)	*Network Latency*
2.	Fakultas Psikologi dan Kesehatan (Rektorat Lama)	*Network Latency*
3.	Fakultas Tarbiyah dan Keguruan	*Network Latency*
4.	Fakultas Adab dan Humaniora	*Network Latency*
5.	Fakultas Ilmu Sosial dan Politik	*Network Latency*
6.	Fakultas Ekonomi dan Bisnis Islam	*Network Latency*
7.	Fakultas Dakwah dan Kominukasi	*Network Latency*
8.	Fakultas Ushuludin dan Filsafat	*Network Latency*
9.	Fakultas Syariah dan Hukum	*Network Latency*
10.	*Twin Tower* Gedung A	*Network Latency*
11.	*Twin Tower* Gedung B	*Network Latency*
12.	Masjid	*Network Latency*
13.	Lab Integrasi	*Network Latency*
14.	Auditorium	*Network Latency*
15.	Sport Center	*Network Latency*
16.	Gedung *Self Access Center* (SAC)	*Network Latency*
17.	Gedung Transit	*Network Latency*
18.	Pagar Utara (bagian dalam)	Metode PNPOLY
19.	Pagar Timur (bagian dalam)	Metode PNPOLY
20.	Pagar Barat (bagian dalam)	Metode PNPOLY
21.	Pagar Timur Laut (bagian dalam)	Metode PNPOLY
22.	Pagar Barat Laut (bagian dalam)	Metode PNPOLY
23	Pagar Samping Masjid (bagian luar)	Metode PNPOLY

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Pengujian pertama dijalankan untuk menguji metode PNPOLY dapat menentukan posisi pengguna berada dalam poligon. Pengujian ini dilakukan sebanyak 2 kali pada 6 tempat dan 4 sistem operasi yang berbeda. Sistem operasi yang digunakan meliputi Android 8, Android 9, IOS 12 dan IOS 13. Sehingga hasil pengujian mendapatkan 48 data mentah yang nantinya akan dievaluasi. Pengujian kedua untuk melihat *network latency* ketika menggunakan aplikasi ini, dilakukan sebanyak 100 kali dengan posisi pengguna berada di dalam poligon. Koordinat titik uji menggunakan 17 titik lokasi yang sudah ditentukan pada Tabel 3.2. Ketika melakukan pengujian ini menggunakan jaringan seluler 2G, 3G dan 4G serta sistem operasi Android 8, Android 9, IOS 12 dan IOS 13. Tetapi pengujian yang dialukan pada IOS hanya terbatas pada jaringan 3G dan 4G. Sehingga didapatkan 1000 data mentah, yang menjadi sumber evaluasi dari *network latency* penggunaan aplikasi.

3.1.5 Evaluasi Sistem

Evaluasi yang pertama dilakukan adalah untuk menentukan persentase metode untuk menentukan posisi pengguna aplikasi ini. Selanjutnya dilakukan evaluasi *network latency*. Data yang sudah dikumpulkan dari hasil pengujian akan ditampilkan kedalam bentuk grafik, sehingga memudahkan proses evaluasi. Lalu dihitung rata-rata *network latency* per jaringan dan sistem operasi yang nantinya disajikan kedalam bentuk tabel dan grafik. Tujuan dilakukan perhitungan rata-rata adalah untuk melihat jaringan dan sistem operasi yang paling baik ketika menggunakan aplikasi ini.

3.2 Tempat dan Waktu Penelitian

Penelitian dengan judul implementasi *Point in Polygon Test* pada aplikasi pendataan kehadiran pegawai berbasis perangkat *mobile*, dilaksanakan mulai pertengahan bulan Agustus 2019 sampai dengan akhir bulan November 2019. Penelitian ini dilakukan di UINSA Surabaya.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

BAB IV
HASIL DAN PEMBAHASAN

4.1 Analisis Kebutuhan

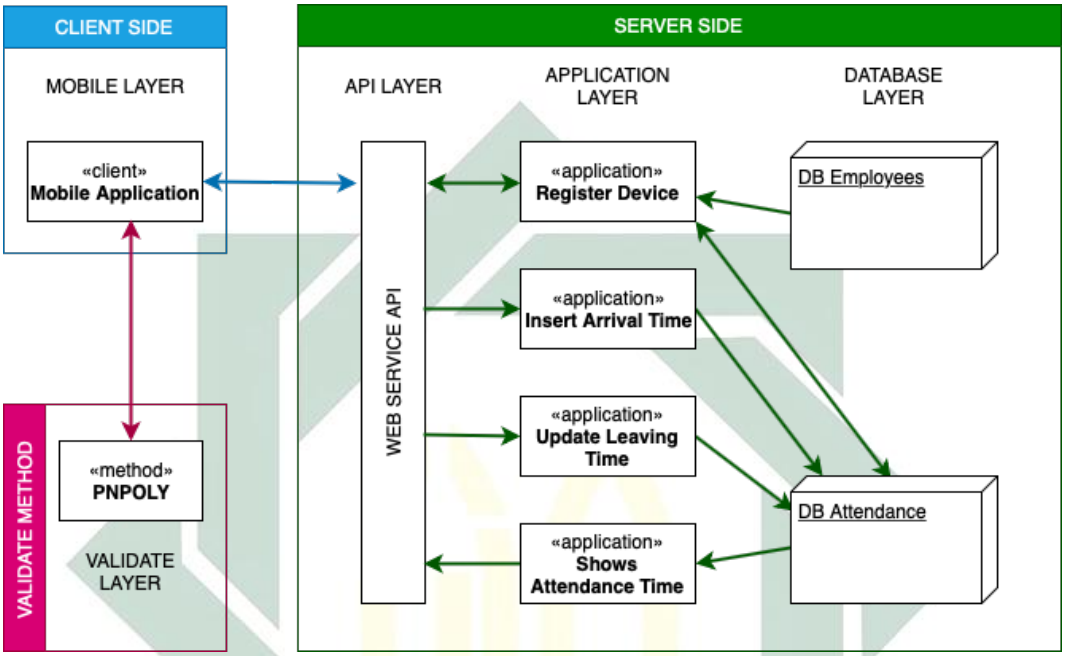
Berdasarkan latar belakang serta rumusan masalah terkait dengan implementasi *Point Inclusion in Polygon Test* (PNPOLY) untuk pendataan kehadiran pegawai berbasis biometrik pada perangkat *mobile*, dihasilkan analisis kebutuhan sistem dengan spesifikasi minimum perangkat menggunakan Android 8 atau IOS 12 serta mempunyai sensor *fingerprint*. Hasil aplikasi yang dikembangkan menggunakan *framework react native* diatur untuk hanya dapat digunakan di wilayah Universitas Islam Negeri Sunan Ampel Surabaya (UINSA Surabaya). Fungsi utama aplikasi ini adalah untuk melakukan pendataan kehadiran, jadi pengaturan menu terbatas pada pendaftaran perangkat yang digunakan dan menu pendataan kehadiran.

Pada menu pendaftaran perangkat, disajikan sebuah *form* yang berisi kotak *input* yang hanya bisa diisi dengan Nomor Induk Pegawai (NIP). Hasil masukan pengguna akan di bandingkan dengan data yang sudah ada di *database*, jika data NIP sama maka dilakukan proses penyimpanan data *user*. Ketika sudah masuk ke dalam aplikasi, akan ditampilkan peta digital yang bersumber dari *react native map*, *library* ini merupakan bagian dari pengembangan peta digital milik Google yang dikhususkan untuk *framework react native* (Airbnb, 2017). Pada aplikasi juga ditampilkan posisi pengguna secara *real time*, dengan bantuan sensor *Global Positioning System* (GPS) pada perangkat *mobile*. Koordinat pengguna yang diambil oleh GPS diolah oleh metode PNPOLY untuk menentukan posisi pengguna berada di dalam wilayah poligon UINSA Surabaya atau tidak. Jika pengguna berada di dalam poligon yang sudah ditentukan, aplikasi memberikan izin untuk melakukan pendataan kehadiran menggunakan autentikasi *fingerprint mobile*. Untuk membatasi penyalahgunaan aplikasi menggunakan lokasi palsu maka ditambahkan keamanan berlapis pada algoritma pengambilan lokasi. Fungsi *logout session* pada aplikasi ini juga ditiadakan untuk membatasi penyalahgunaan pemakaian aplikasi oleh user dalam sistem pendataan kehadiran berbasis *mobile*.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

4.2 Desain Arsitektur Sistem

Aplikasi pendataan kehadiran pegawai dikembangkan berdasarkan hasil analisis kebutuhan menjadi bentuk desain arsitektur sistem, seperti yang ditunjukan pada Gambar 4.1.



Gambar 4.1: Desain Arsitektur Sistem

Terdapat 3 bagian utama pada desain arsitektur yang terbentuk, yaitu *server side, client side* dan *validate method*. Bagian *server side* bertugas untuk melakukan pekerjaan yang berhubungan dengan penggunaan *server*. Ada 3 layer yang termasuk didalamnya, anatara lain *API layer* yang bertugas membuat *RESTful API* untuk melakukan pertukaran data *JSON* dari *back-end* ke *front-end* ataupun sebaliknya. *Layer* kedua adalah *application*, yang berisi berkas PHP sebagai *back-end* dari sistem ini. Kegunaan *back-end* adalah menjalankan fungsi *query sql* ke dalam *database*, hasilnya berupa data yang nantinya akan dikirimkan ke *API layer* untuk diproses menjadi *JSON*. Terakhir adalah *database layer*, berfungsi untuk menyimpan data dari aplikasi. Pada sistem ini menggunakan *database* dari MySQL untuk memudahkan manajemen penggunaan.

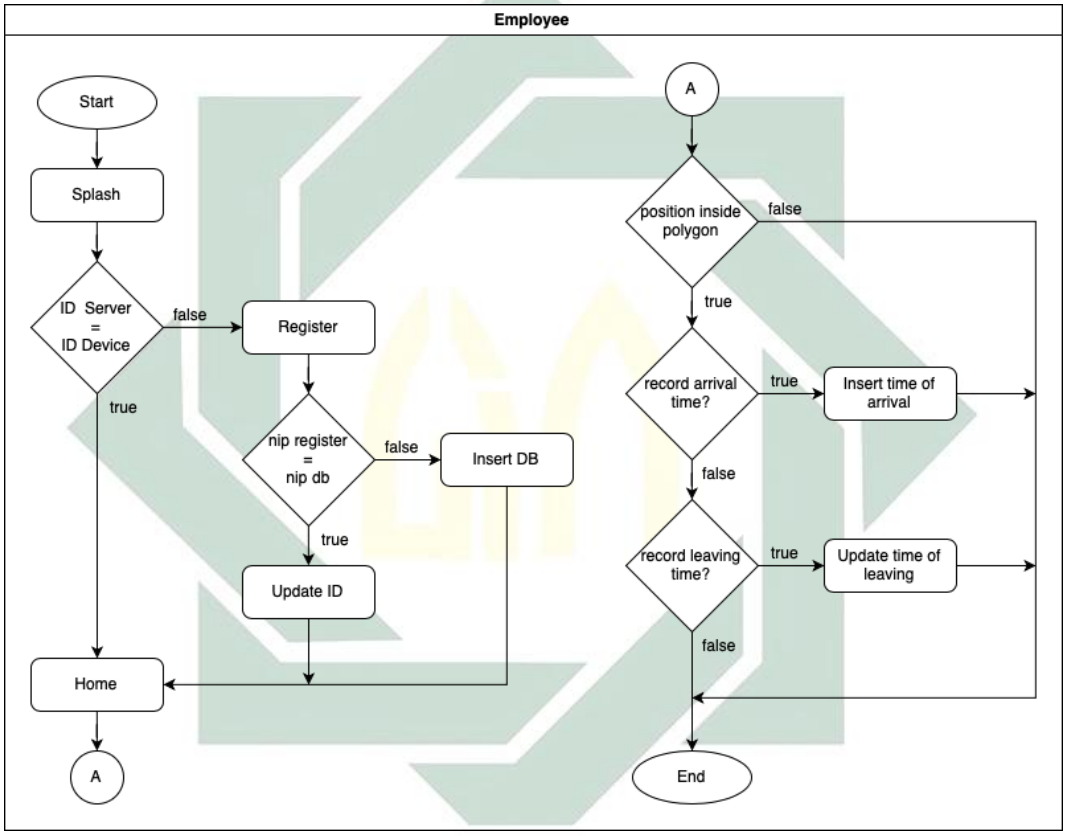
Di bagian *client side* hanya berisi *mobile layer*, didalamnya terdapat *client application* yang digunakan oleh pengguna. Bagian ketiga adalah *validated method*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

dalam pengembangan sistem. *Layer* yang terdapat di bagian ini berfungsi untuk melakukan verifikasi pada *mobile layer*. Proses ini berguna dalam menentukan lokasi pengguna ketika menggunakan aplikasi.

4.3 Alur Penggunaan Aplikasi

Alur penggunaan aplikasi dijelaskan pada *flowchart* yang digambarkan kedalam Gambar 4.2 dibawah ini.



Gambar 4.2: Alur Penggunaan Aplikasi

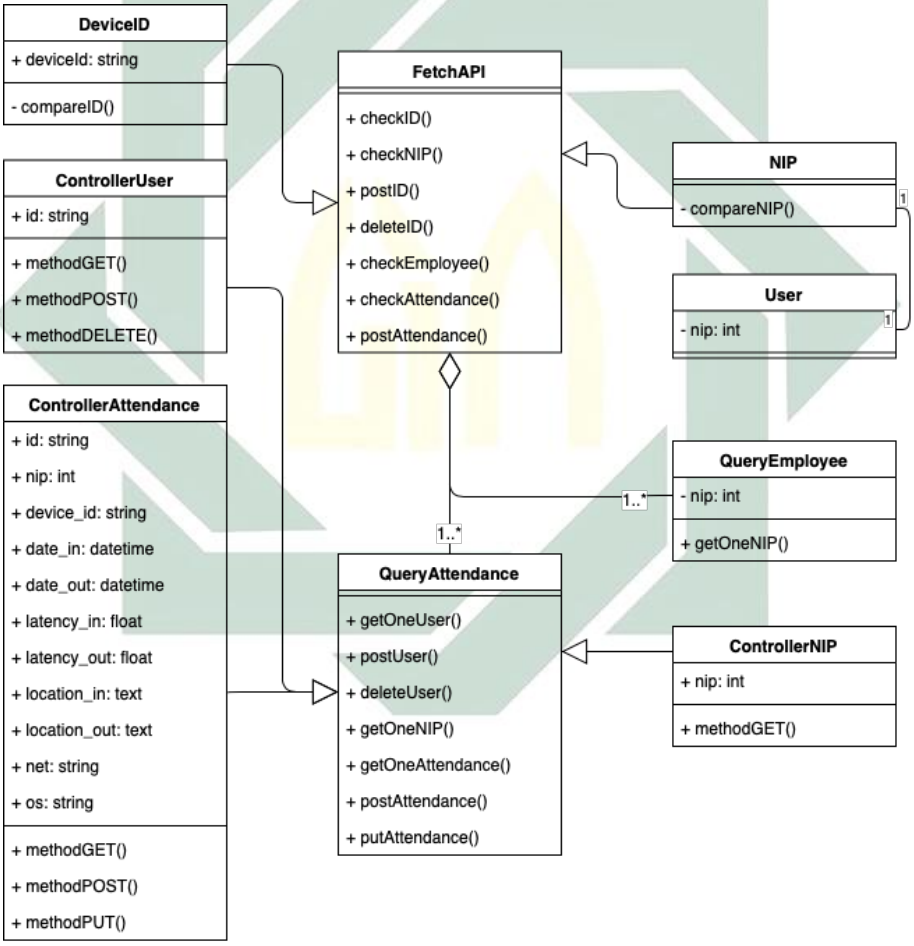
Alur penggunaan aplikasi ini dimulai dengan proses *splash* yang menampilkan *splash screen* aplikasi. Pada proses ini dilakukan pengecekan ID, apakah ID yang berada di dalam *database* sama dengan *id device*. Jika benar maka akan beralih tampilan ke menu *home* dan sebaliknya, jika bernilai salah maka akan dialihkan ke tampilan *register*. Pada tampilan *register* pengguna diminta untuk memasukan NIP yang dimiliki, NIP hasil masukan tersebut akan dibandingkan dengan NIP pada data pegawai di *database*. Jika bernilai benar, dilakukan *update id device* kedalam tabel *user*, dan jika salah maka akan melakukan fungsi *insert*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

user baru ke dalam *database*. Tampilan aplikasi akan diarahkan ke menu *home* setelah proses pengecekan selesai. Di menu *home* dilakukan validasi posisi oleh metode PNPOLY, jika hasil bernilai benar maka pengguna dapat melakukan fungsi pendataan kehadiran ketika datang maupun waktu pulang.

4.4 *UML Class Diagram*

Diagram yang bersifat stuktural ini berfungsi untuk menggambarkan rancangan *class* yang terdapat pada aplikasi. Struktur *class* divisualisasikan kedalam Gambar 4.3 dibawah ini.



Gambar 4.3: *UML Class Diagram*

Terdapat 9 *class* yang terdapat dalam rancangan *class diagram* aplikasi ini. Beberapa *class* tersebut dibagi kedalam bentuk *front-end* dan *back-end*. Di dalam *front-end* terdapat *class name DeviceID, FetchAPI, NIP* dan *User*, sedangkan pada bagian *back-end* terdapat *class name QueryAttendance, QueryEmployee,*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

ControllerNIP, ControllerUser dan *ControllerAttendance*. Setiap bagian *class* memiliki fungsi yang berbeda, berikut adalah deskripsi fungsi dari setiap *class*.

1. *User*

   Pada bagain *classname user*, terdapat *attribute* berupa nip dengan tipe *string* dan batasan akses secara *private*. *Attribute* nip diperoleh dari hasil masukan *user*. Bagian ini berelasi dengan *class* NIP, dengan tipe relasi berupa *association*.

2. NIP

   Dalam *class* NIP tidak terdapat *attribute*, tetapi ada *private operation* yang dijalankan yaitu *compareNIP()*. Ada dua relasi yang terdapat pada *class* ini, yang pertama adalah relasi bertipe *association* ke *class User*. Kedua adalah relasi dengan *class FetchAPI*, tipe relasi yang digunakan berupa *inheritance* atau dapat disebut dengan *generalization*. Tipe relasi ini menghasilkan relasi antara *parent* dan *child* yang menjalankan *function* secara virtual dari *function* yang dimiliki oleh *parent*. Operasi *compareNIP()* dijalankan pada bagian ini untuk melakukan perbandigan nip dari hasil masukan *user* dengan nip yang berasal dari *database employee*.

3. *DeviceID*

   Class ini memiliki *attribute* berupa *deviceId* yang bersifat *public* dengan tipe data *string*. *Private operation* yang dijalankan adalah *compareID()*, operasi ini digunakan untuk membandingkan *deviceId* yang didapat dari perangkat dengan *deviceId* yang tersimpan di *database*. Hanya terdapat satu relasi pada bagian ini yaitu *generalization* dengan *class FetchID*.

4. *FetchAPI*

   Di dalam *class FetchAPI* tidak terdapat *attribute*, akan tetapi *class* ini menjalankan beberapa *public operation* didalamnya, karena *class* ini merupakan *parent* dari 2 *class* lain yaitu *class NIP* dan *class DeviceID*. Sehingga bagian ini memiliki *generalization* terhadap kedua class tersebut. Operasi yang terdapat pada *class* antara lain, *checkID()* untuk melakukan pengecekan *deviceId* ke *database, checkNIP()* berguna untuk pengecekan *nip* yang terdapat pada *database, postID()* untuk menyimpan data *user,*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

deleteID() untuk menghapus data *user*, *checkEmployee()* digunakan untuk mengecek data *employee*, *checkAttendance()* untuk melakukan pengecekan data *attendance* dan *postAttendance()* digunakan untuk menyimpan data *attendance* dengan *method* POST dan PUT. Tidak terdapat *generalization* saja, pada *class* ini juga relasi *agregation* dengan *class QueryAttendance* dan *class QueryEmployee*.

5. *QueryAttendance*

   Class ini merupakan *parent* dari 3 *class child* dibawahnya, dan diwaktu yang sama *class* ini juga termasuk *child* dari *parent class FetchAPI*. Ada 3 relasi yang menghubungkan *class* ini dengan *class ControllerUser, ControllerNIP* dan *ControllerAttendance*. Semua relasi tersebut bertipe *generalization*. Dalam *class* ini tidak ada *attribute* yang digunakan, akan tetapi ada beberapa *public operation* yang dijalankan, antara lain operasi *getOneUser()* berguna untuk melakukan *query* pengambilan data *user* berdasarkan *deviceId* yang dikirim, *postUser()* untuk melakukan *query* penyimpanan data *user, deleteUser()* digunakan dalam *query* penghapusan data *user* berdasarkan nip yang dikirim, *getOneNIP()* berguna dalam *query* pengambilan data *user* berdasarkan nip yang dikirim, *getOneAttendance()* untuk melakukan *query* pengambilan data *attendance* pada hari itu juga, *postAttendance()* digunakan untuk *query* penyimpanan data *attendance* ketika *user* datang, dan yang terakhir adalah operasi *putAttendance()* yang berguna untuk melakukan *query update* data *attendance* ketika *user* pulang.

6. *QueryEmployee*

   Dalam *class* ini terdapat *attribute private* nip dengan tipe data *integer* dan sebuah operasi *public* yaitu *getOneNIP()*. Operasi ini digunakan untuk melakukan *query* pengambilan data pegawai dari *database employee*. Hanya terdapat sebuah *relasi* dalam *class* ini, relasi tersebut bertipe *aggregation* yang tersambung dengan *parent class FetchAPI*.

7. *ControllerUser*

   Pada bagian ini terdapat sebuah *attribute id* dengan tipe data *string* dan bersifat *public*, serta terdapat 3 *public operation* di dalamnya. Operasi

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

tersebut antara lain, *methodGET(), methodPOST()* dan *methodDELETE()*. Fungsi dari operasi tersebut sama, yaitu untuk melakukan pemanggilan *function* ke client-*side* dengan tipe *method* yang sesuai. Relasi yang terdapat pada *class* ini hanya satu, yaitu *generalization* menuju *parent class QueryAttendance*.

8. *ControllerNIP*

   Dalam *class ControllerNIP* terdapat sebuah *public attribute nip* dengan tipe data *integer* dan sebuah *public operation methodGET()*. Relasi yang ada juga hanya satu, yaitu *generalization* ke *parent class QueryAttendance*.

9. *ControllerAttendance*

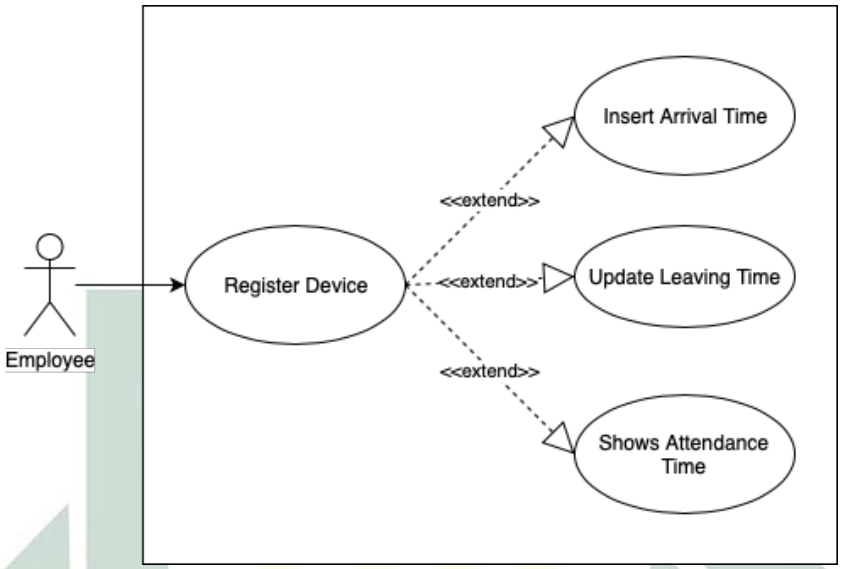
   Class yang terakhir ini memiliki banyak sekali *attribute* yang bersifat *public*, diantaranya ada *id* dengan tipe data *string, nip* bertipe data *integer, deviceId, net* dan *os* yang bertipe data *string*, lalu *attribute date_in* dan *date_out* dengan tipe data *datetime*, kemudian terdapat *attribute latency_in* dan *latency_out* yang bertipe data *float*, serta terakhir ada *attribute location_in* dan location_*out* dengan tipe data *text*. Dalam *class* ini juga terdapat 3 *public operation*, yaitu *methodGET(), methodPOST()* dan *methodPUT()*. Hanya ada sebuah relasi yang terdapat pada bagian ini, yaitu sebuah *generalization* ke *parent class QueryAttendance*.

4.5 *UML Use Case Diagram*

Diagram ini menampilkan skenario penggunaan aplikasi dari beberapa pengguna yang saling berhubungan. Skenario yang dimiliki user satu dengan yang lain berbeda. *Use case diagram* yang dibuat dalam aplikasi ini menggambarkan pengguna dapat melakukan beberapa fungsi pekerjaan seperti yang ditunjukan pada Gambar 4.4. Dalam aplikasi ini hanya terdapat satu orang pengguna, yaitu pegawai. Sesuai dengan hasil analisis didapatkan bahwa pengguna aplikasi dapat melakukan pendaftaran perangkat, kemudian dapat melihat waktu ketika melakukan pendataan kehadiran dan tentunya melakukan fungsi pendataan kehadiran, baik ketika datang maupun pulang. *User* hanya dibatasi menggunakan keempat fungsi tersebut untuk mempermudah mengoperasikan aplikasi. *Use case insert arrival time, update*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

leaving time dan *shows attendance* time merupakan *extend* dari *register device*. Artinya ketiga *case* mempunyai sifat ketergantungan dengan *case* yang pertama, sehingga tidak dapat dijalankan tanpa melewati *register device*.



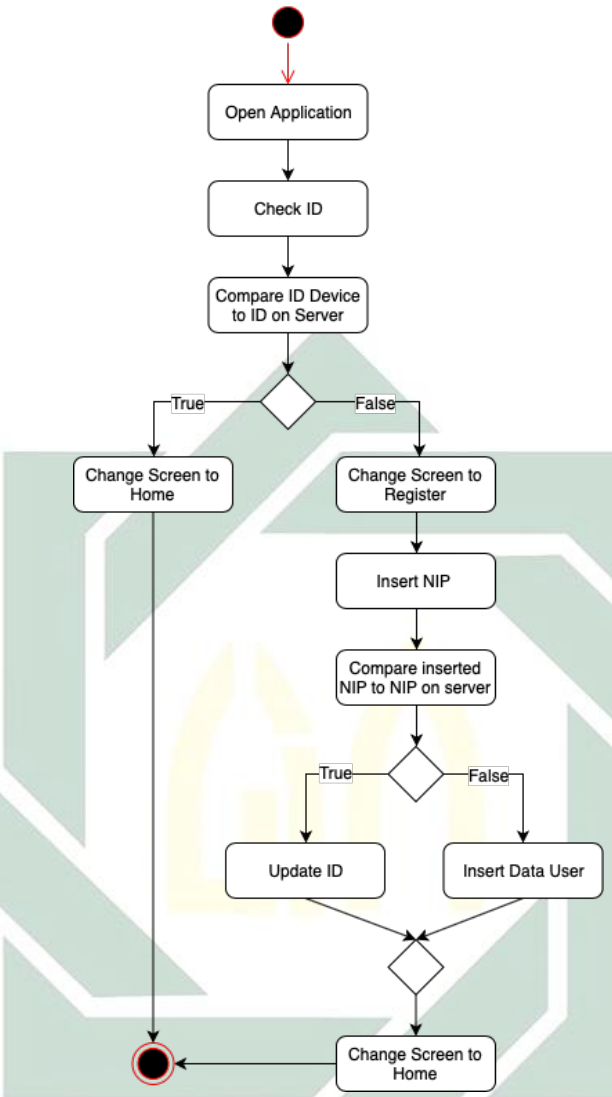
Gambar 4.4: *UML Use Case Diagram*

4.6 *UML Activity Diagram*

Pada diagram ini menggambarkan aktivitas sistem dari awal permulaan sanpai selesai, memperlihatkan beberapa jalanya proses terkait aktivitas yang dikerjakan sistem. Dari empat fungsi pekerjaan yang digambarkan pada Gambar 4.4, dibuat menjadi *activity diagram* sebanyak 3 diagram, masing-masing meliputi *register device, insert arrival time* dan *update leaving time.* Activity diagram untuk proses register device dimuat kedalam Gambar 4.5.

Diagram aktivitas untuk *register device* dimulai dari *open application*, kemudia dilakukan pengecekan ID. Dalam pengecekan ini, ID *device* dibandingkan dengan ID yang disimpan dalam *database*, jika hasilnya bernilai benar maka tampilan dirubah ke menu *Home*. Jika bernilai salah tampilan diubah ke menu *register* dan disajikan *form input* NIP. Setelah didapat masukan NIP, lalu dilakukan verifikasi NIP ke *database*. JIka hasil yang didapat benar maka dilanjutkan ke proses *update ID*. Home. Akan tetapi sebaliknya, jika salah maka dilakukan *insert* data pengguna yang baru. Setelah proses ini selesai tampilan aplikasi dirubah ke menu *Home*.

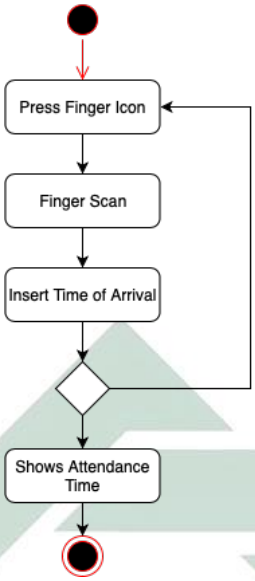
digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id



Gambar 4.5: *UML Activity Diagram Register Device*

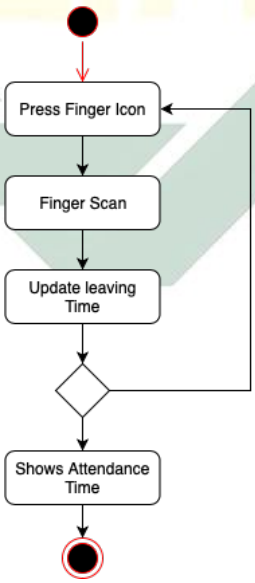
Activity diagram yang kedua menjelaskan alur aktivitas pada proses *insert arrival time*, yang digambarkan pada Gambar 4.6. Aktifitas dalam *insert arrival time* dimulai ketika posisi pengguna dinyatakan berada didalam poligon. *User* melakukan *fingerprint scanner*, setelah itu waktu kedatangan dicatat dan diproses ke *database*. Jika hasil *insert* bernilai benar maka dilanjutkan ke proses *shows attendance time*.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id



Gambar 4.6: *UML Activity Diagram Insert Arrival Time*

Diagram ketiga yang digambarkan pada Gambar 4.7 menunjukan *activity diagram* pada proses *update leaving time*. Alur proses ini hampir sama dengan proses pada *insert arrival time*. Perbedaanya terdapat pada aktivitas proses kedua yaitu *update leaving time*.

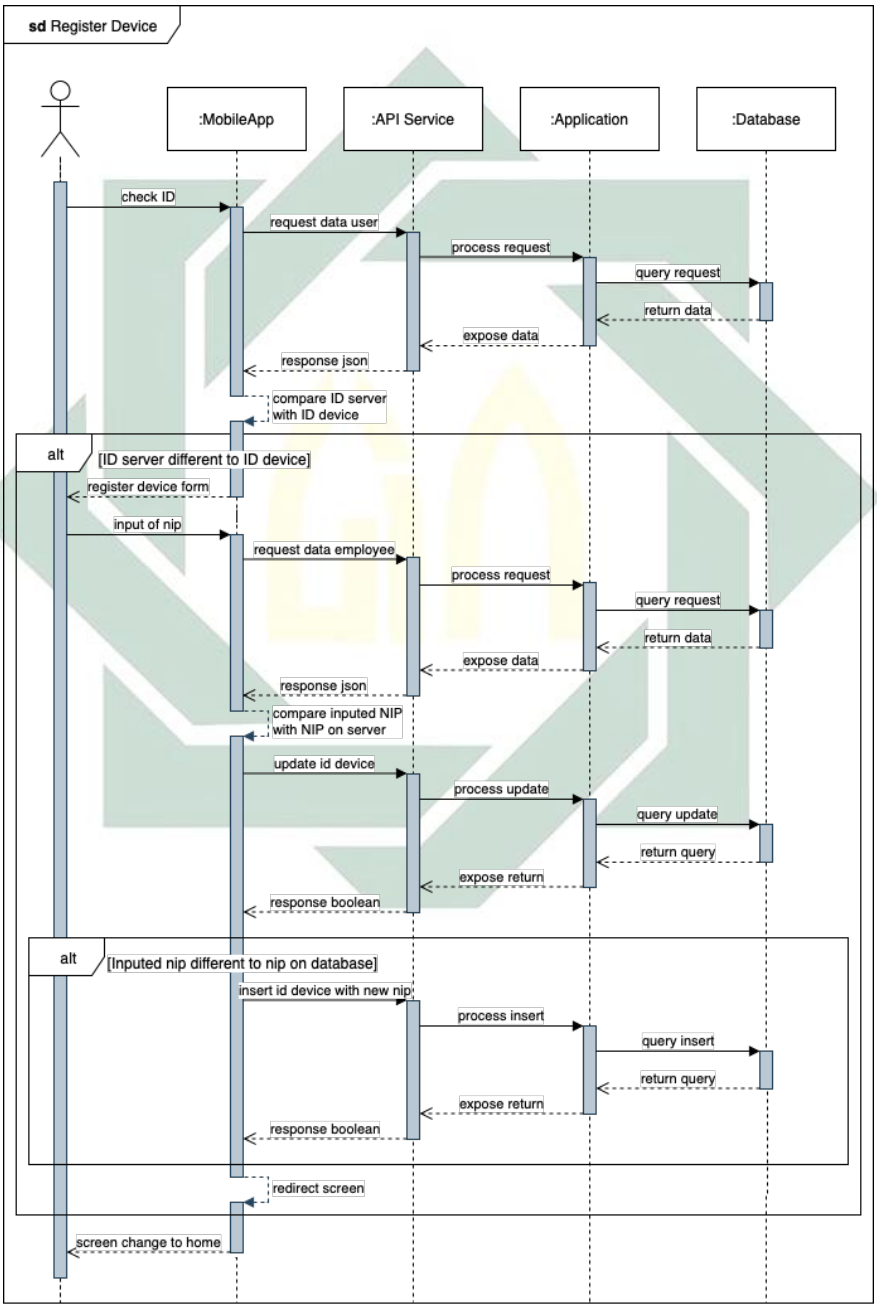


Gambar 4.7: *UML Activity Diagram Update Leaving Time*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

4.7 *UML Sequence Diagram*

Interaksi antar *object* yang terdapat pada aplikasi ini digambarkan kedalam bentuk *UML Sequence Diagram*. Terdapat 4 fungsi utama yang digambarkan kedalam 3 bentuk *sequence diagram*, meliputi *register device*, *insert arrival time* dan *update leaving time*. *Register device* dijelaskan pada Gambar 4.8.



Gambar 4.8: *UML Sequence Diagram Register Device*

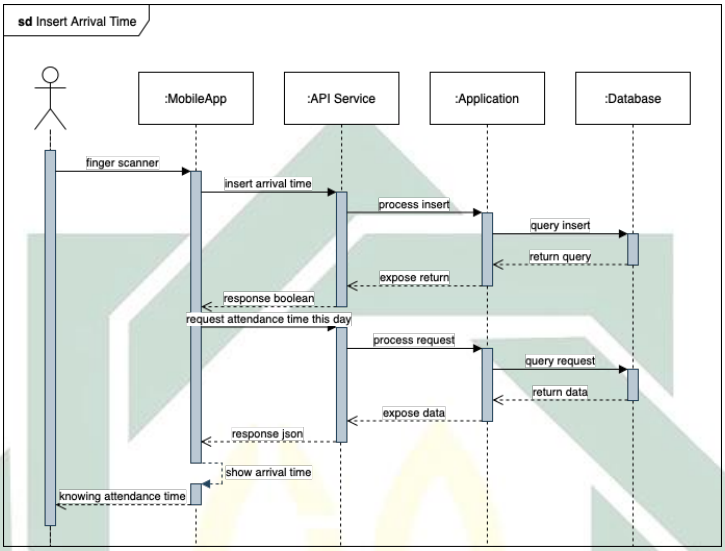
digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Pembuatan *sequence diagram* untuk fungsi *register device* menghasilkan sebuah *actor* dengan 4 buah *object* yang berbeda. *Actor* yang dimaksud adalah pengguna aplikasi, sedangkan *object* yang ada meliputi *mobile app*, *api service*, *application* dan *database*. Alur interaksi object dimulai dari *actor* mengirimkan pesan berupa *check ID* kepada *mobile app*, oleh *object* ini kemudian diteruskan ke *api service*, lalu menuju *application*. Pada *application* dilakukan *query* untuk mengambil data di *database*. Hasil *query* berupa *raw data* yang di *expose* ke *api service*. Data yang didapat oleh *api service* dikonversikan menjadi *JSON* agar dapat dikirim ke bagian *mobile app*. Di *mobile app*, ID yang terdapat pada data *user* dibandingkan dengan *id device*. Jika sama maka akan dialihkan menuju *home screen*. Akan tetapi jika hasilnya tidak sama maka dialihkan menuju *form register device*. Pada *form* tersebut *actor* harus memasukan NIP sebagai indikator yang digunakan oleh aplikasi untuk membandingkan data pegawai pada *database*. Jika perbandingan NIP menghasilkan nilai yang sama maka dilakukan proses *update* oleh *mobile app* dengan mengirimkan *request update id device* kepada *api service*, proses dilanjutkan ke *application object*, lalu dilakukan *query update* ke *database*. Hasilnya berupa *boolean* (*true or false*) yang diterskan ke *mobile app*, melalui *application* dan *api service*. Jika hasil perbandingan NIP tadi bernilai tidak sama maka dilakukan proses *insert user* baru, dengan proses yang hampir sama dengan *update*, akan tetapi berbeda *query* yang digunakan. Ketika semua proses sudah dilakukan *mobile app* akan mengalihkan tampilan menuju menu *home*.

Sequence diagram yang kedua menggambarkan interaksi yang ada pada proses *insert arrival time*. *Sequence* ini mempunyai *actor* dan *object* yang sama dengan *sequence diagram register device*. Ketika *actor* melakukan *fingerprint scanner*, *mobile app* akan merespon perintah tersebut dengan mengirimkan *request* ke *api service*, kemudian meneruskanya kepada *application*. *Appication* akan melakukan *query insert attendance* kedalam *database*. Hasil *query* berupa *boolean* yang dikirimkan ke *mobile app*, melalui application dan *api service*. Setelah *request* pertama selesai, lalu dilanjutkan untuk resuest selanjutnya, yaitu *request data attendance system at that day*. *Api service* meneruskan permintaan itu ke *application*, kemudian dilakukan *query*. Hasil *raw data* dibawa menuju *api service*

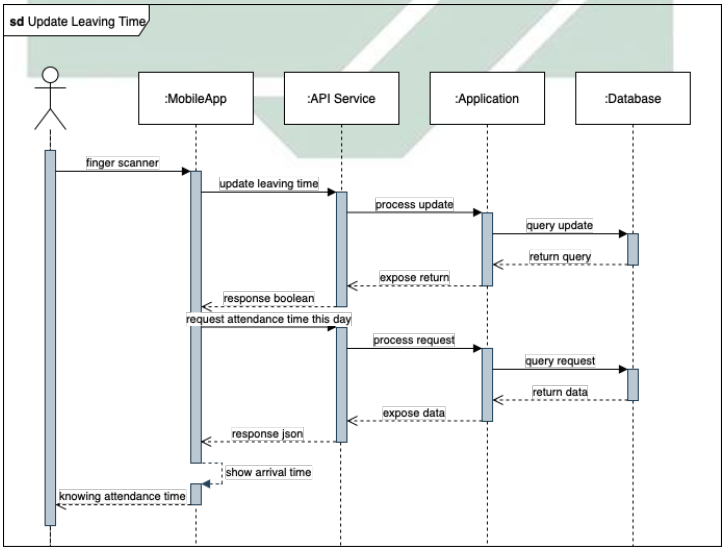
digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

yang nantinya akan dikonversikan ke bentuk *JSON*. *Mobile app* menerima *response JSON* yang dikirimkan oleh *api service*, lalu diproses untuk ditampilkan waktu yang tercatat ketika melakukan pendataan kehadiran kepada *actor*. Berikut adalah bentuk *sequence diagram insert arrival time*, ditunjukan pada Gambar 4.9.



Gambar 4.9: *UML Sequence Diagram Insert Arrival Time*

Sequence diagram ketiga adalah *update leaving time*. Alur interaksi antar *object* sama dengan *sequence* yang kedua, akan tetapi berbeda *query* yang digunakan. Bentuk ketiga ini digambarkan kedalam Gambar 4.10.

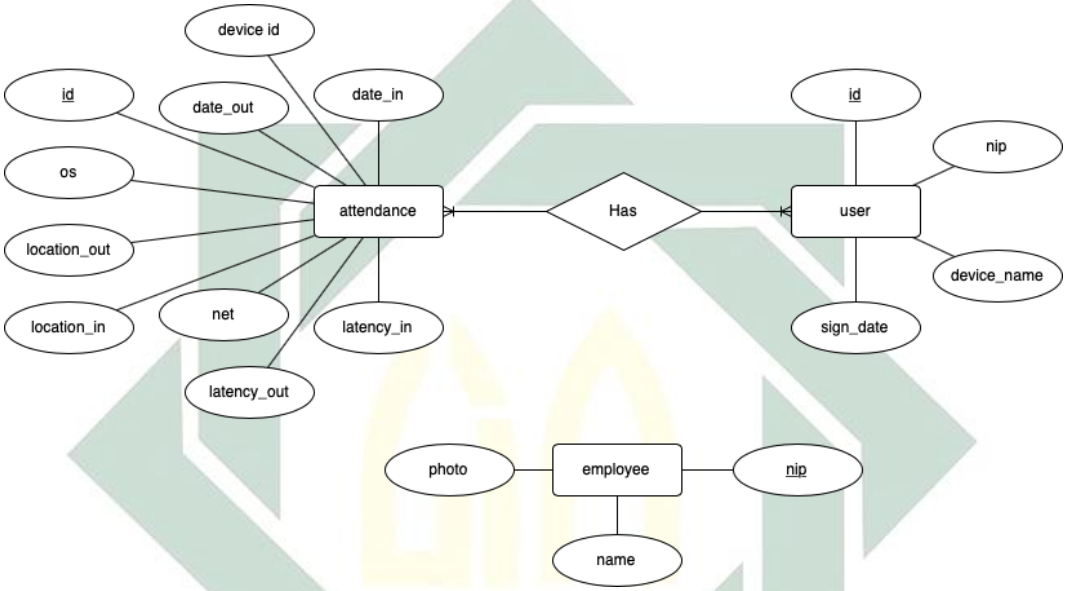


Gambar 4.10: *UML Sequence Diagram Update Leaving Time*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

4.8 *Entity Relationship Diagram* (ERD)

Pada aplikasi ini menggunakan dua *database*, yaitu *db employee* dan *db attendance*. Databse pertama berisi data pegawai yang digunakan untuk melakukan pengecekan NIP ketika *register device*. *Database* yang kedua merupakan *database* utama dari aplikasi ini, dimana seluruh data *user* dan *attendance* disimpan dalam *database* ini. Rancangan ERD digambarkan pada Gambar 4.11 di bawah ini.



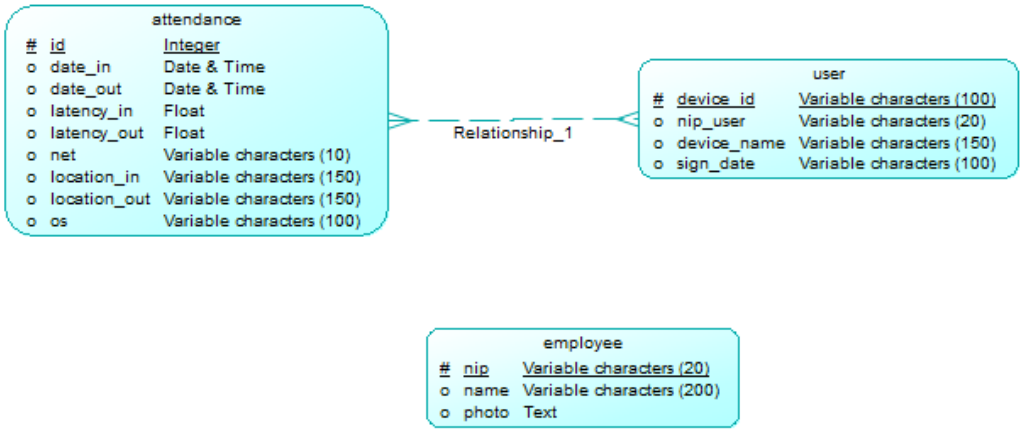
Gambar 4.11: ERD Aplikasi

Database db employee menghasilkan sebuah tabel yang bernama *employee*, di dalamnya ada 3 atribut yaitu, nip, *name* dan *photo*. *Database* yang kedua membentuk dua tabel, yaitu *user* dan *attendance*. Tabel *user* berisi atribut id, nip, *device_name* dan *sign_date*. Tabel yang kedua berisi id, *device_id, date_in, date_out, latency_in, latency_out, net, location_in, location_out* dan *os*. Pada tabel yang terbentuk tidak disertai dengan relasi dikarenakan kebutuhan data pada tabel satu dengan yang lainya tidak saling ketergantungan.

4.9 *Conceptual Data Model* (CDM)

CDM merepresentasikan rancangan diagram database berdasarkan ERD pada Gambar 4.11. Pada model yang dibuat berisi rancangan tabel *database* beserta dengan *field* dan tipe data yang digunakan. Rancangan model aplikasi ini ditampilkan pada Gambar 4.12.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id



Gambar 4.12: CDM Aplikasi

Dalam rancangan CDM terdapat 3 desain tabel utama dari *database db employee* dan *db attendance*. Struktur tabel *employee* pada *database db employee* terdiri dari *field* nip sebagai *primary key* dengan tipe data *varchar* 20 *character*, kemudian *field name* bertipe data *varchar* dengan panjang 200 *character* dan *field photo* yang bertipe data *text*. *Field* terahir ini berisi *link* foto dai pegawai.

Database db attendance terdiri dari dua tabel, tabel pertama adalah *user*. Isi dari tabel ini yang pertama adalah *device_id* sebagai *primary key*, dengan tipe data *vharchar* 100 *character*. Kemudian nip bertipe data *vharchar* 20 *character*, lalu ada *device_name* bertipe *varchar* dengan panjang *character* sebanyak 150. Terakhir adalah *field sign date* dengan tipe data *varcher* 100 *character*. *Field* ini tidak dibuat tipe *date* karena untuk memudahkan pengisian data waktu dan tanggal dari *client*.

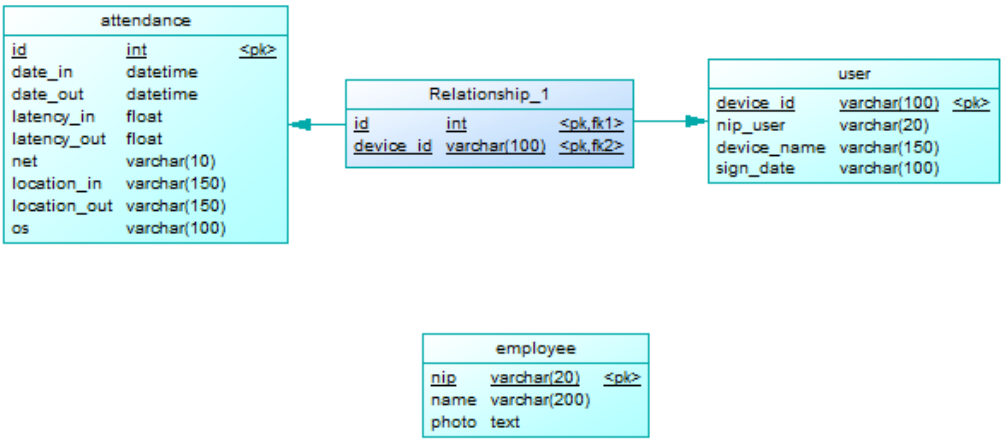
Tabel kedua yang terbentuk adalah *attendance*, struktur dari tabel ini terdiri dari *field id* dengan tipe data *int auto increment*. *Field* ini menjadi *primary key* dari tabel *attendance*. Lalu ada *field* nip bertipe data *varchar* 100 *character*, *field device_id* bertipe *varchar* 150 character. Selanjutnya ada *field date_in* dan *date_out* dengan tipe data yang sama yaitu *datetime*. *Field net* dengan tipe data *vharchar* 10 *character*. *Field* ini digunakan untuk menyimpan jaringan seluler ketika menggunakan aplikasi ini. Kemudian terdapat *field latency_in* dan *latency_out* bertipe data *float*, bagian ini digunakan untuk menyimpan data *latency* ketika melakukan *fingerprint attendance*, baik ketika waktu datang dan pulang. Hal ini berguna untuk memudahkan pengujian *latency* jaringan. Tiga *field* terakhir adalah

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

location_in, location_out dan os dengan tipe data yang sama yaitu *varchar*, untuk *field location* panjangnya 150 *character* sedangkan os panjangnya 100 *character*. *Field* os berisi nama sistem operasi yang digunakan untuk melakukan *attendance*.

4.10 *Physical Data Model* (PDM)

Diagram dari Gambar 4.12 diubah kedalam bentuk PDM. Model ini merepresentasikan bentuk data sebenarnya dari *database*. Dalam PDM yang dibuat berisi rancangan tabel *database* beserta dengan *field* dan tipe data yang digunakan. Bentuk PDM hampir sama dengan CDM, perbedaannya terdapat pada deskripsi relasi yang digunakan lebih diperjelas. *Field device_id* pada tabel *attendance* mengambil data dari hasil relasi antara *field device_id* yang menjadi *primary key* pada tabel *user*. Rancangan model aplikasi ini ditampilkan pada Gambar 4.13 berikut.



Gambar 4.13: PDM Aplikasi

4.11 Implementasi Metode

Penelitian ini menggunakan metode PNPOLY untuk membatasi penggunaan aplikasi hanya digunakan dalam wilayah poligon yang sudah ditentukan sebelumnya. Dalam rancangan aplikasi yang dibuat menggunakan bahasa pemrograman *javascript*, sehingga metode PNPOLY dibuat kedalam bentuk *javascript*. Sebagai *function* dalam *javascript* metode ini dipanggil berulang kali sebanyak lokasi pengguna yang diambil oleh sensor GPS. Jadi aplikasi ini dapat

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

melakukan *live tracking* dengan lebih baik. Gambar 4.14 memperlihatkan baris kode PNPOLY yang sudah diubah kedalam bahasa *javascript*.

```
function inside(point, vs) {
  var x = point.latitude, y = point.longitude;

  var inside = false;
  for (var i = 0, j = vs.length - 1; i < vs.length; j = i++) {
      var xi = vs[i].latitude, yi = vs[i].longitude;
      var xj = vs[j].latitude, yj = vs[j].longitude;

      var intersect = ((yi > y) != (yj > y)) && (x < (xj - xi) * (y - yi) / (yj - yi) +
xi);
      if (intersect) inside = !inside;
  }

  return inside;
};
```

Gambar 4.14: *Function Javascript* Metode PNPOLY

Function ini memiliki *parameter* berupa *point* dan *vs*. Parameter *point* adalah *object* koordinat pengguna. Sedangkan *vs* merepresentasikan *object array* dari koordinat batas poligon UINSA Surabaya. Hasil dari *function* ini berupa nilai *true* atau *false*. Baris kode pada Gambar 4.14 ini adalah hasil implementasi metode PNPOLY, sesuai dengan *flowchart* metode ini yang ditampilkan pada Gambar 2.3.

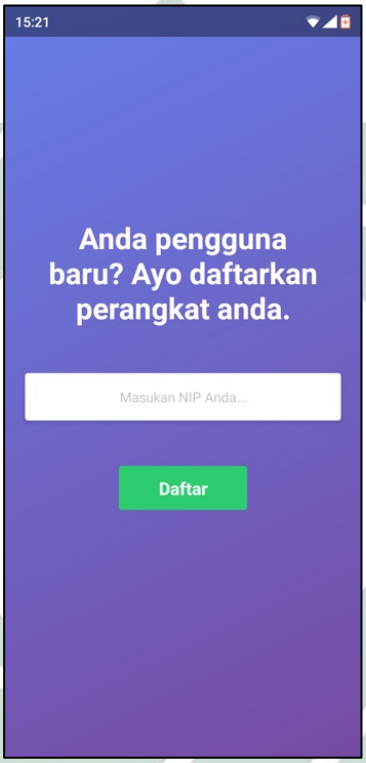
4.12 Desain *User Interface* (UI) Aplikasi

Ada beberapa desan tampilan yang berhasil dibuat dari pengembangan aplikasi ini. Tampilan tersebut meliputi *splash screen, register screen* dan *home screen*. Pada tampilan *splash screen* terdapat logo aplikasi dengan *background* warna bertema gradasi. Proses pengecekan awal dilakukan pada saat tampilan ini dipanggil. Tampilan kedua adalah tampilan *register screen*, di sini terdapat *form* pendaftaran perangkat, dengan NIP pengguna digunakan sebagai id pendaftaran. Berikut adalah tampilan *register screen*, yang digambarkan pada Gambar 4.15.

Pada bagian ini hanya terdapat sebuah *input text* dalam *form* yang disediakan. Masukan dibatasi hanya dapat diisi dengan angka, karena fungsinya memang digunakan untuk mengisi NIP pengguna. Ketika tombol daftar ditekan akan memicu fungsi *fingerprint*. Pengguna harus melakukan *fingerprint* pada perangkat sebagai bukti izin untuk melakukan pendaftaran NIP menggunakan id pada

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

perangkat yang bersangkutan. Pesan kesalahan akan muncul ketika *user* salah melakukan *scan fingerprint* jari mereka, atau NIP yang dimasukan tidak terdaftar pada *database*. Kebalikan dari itu, proses akan dilanjutkan menuju tampilan home, setelah proes pendaftaran berhasil. *Register screen* tidak ditampilkan ketika pengecekan awal id pada *splash screen* menghasilkan nilai benar. Aplikasi akan langsung menampilkan *home screen*.

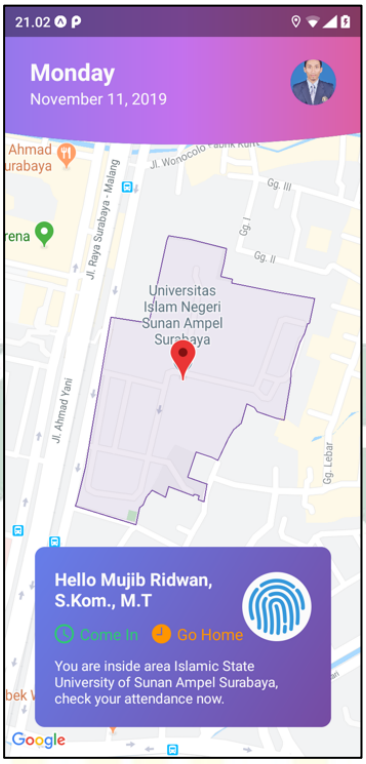


Gambar 4.15: UI *Register Screen*

Terdapat beberapa fitur yang disediakan pada tampilan *home screen*. Fitur yang termasuk didalamnya adalah *tracking live location* dan menampilkanya pada peta aplikasi, memperlihatkan batas wilayah poligon yang sudah diatur sebelumnya, fungsi *fingerprint scanner* dan menampilkan waktu hasil melakukan *fingerprint attendance*. *Tracking* yang dilakukan sistem bekerja dengan mengambil data lokasi menggunakan sensor GPS, lalu di proses oleh PNPOLY untuk menentukan posisi pengguna berada di dalam poligon atau sebaliknya. Hasil proses tersebut ditampilkan kedalam peta aplikasi beserta dengan wilayah poligon yang digunakan untuk perhitungan. Jika hasil proses perhitungan bernilai benar dan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

berada di dalam poligon maka fungsi *fingerprint* dapat diaktifkan, seperti yang ditunjukan pada Gambar 4.16.



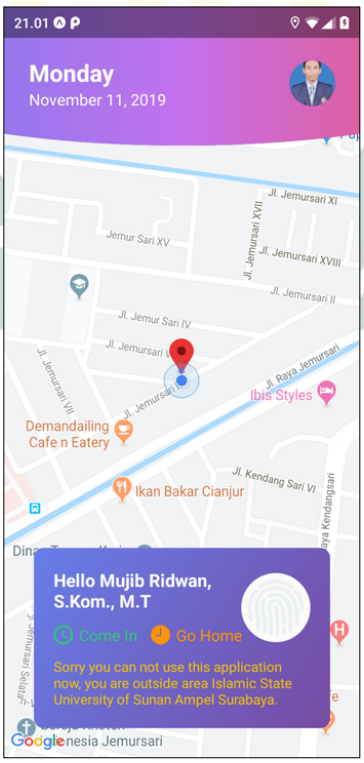
Gambar 4.16: UI *Home Screen* Saat Posisi Pengguna di Dalam Poligon

Desain tampilan *home screen* dibuat untuk menjalankan fungsionalitas utama aplikasi dan mempermudah pengguna untuk menjalankan aplikasi ini. Tampilan ini terdiri dari 3 bagian, yang pertama adalah *header*, kemudian *body* dan yang terakhir adalah *floating container*. Bagian *header* terdapat informasi hari dan tanggal ketika menggunakan aplikasi, serta *user image* di sebelah kanan layer. Desain *body* pada *home screen* ini diisi oleh peta digital, dengan menggambarkan wilayah poligon UINSA Surabaya. Tidak lupa dibuat penggambaran posisi pengguna dengan simbol *marker* merah. Dalam *floating container* terdapat nama pengguna di bagian atas, kemudian waktu kedatangan di bawah nama pengguna, dengan warna hijau, serta waktu kepergian dengan warna jingga. Perbedaan warna ditujukan untuk mempermudah pengguna untuk membedakan informasi waktu yang diberikan. Di sebelah kanan terdapat tombol *fingerprint*. Ketika tombol ini ditekan akan memunculkan *pop up* pemberitahuan untuk meletakan jari pada sensor *fingerprint smartphone*. Pengguna sudah dapat melakukan *fingerprint scanner* ketika *pop up*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

ini muncul. Pada bagian bawah sendiri terdapat informasi tentang posisi pengguna, ketika menggunakan aplikasi ini. Ketika posisi pengguna berada di dalam poligon saat menggunakan aplikasi ditunjukan dalam kalimat peringatan berikut, *"you are inside area Islamic State University of Sunan Ampel Surabaya, check your attendance now"*, serta tombol *fingerprint* berwarna biru, seperti yang ditunjukan pada Gambar 4.16.

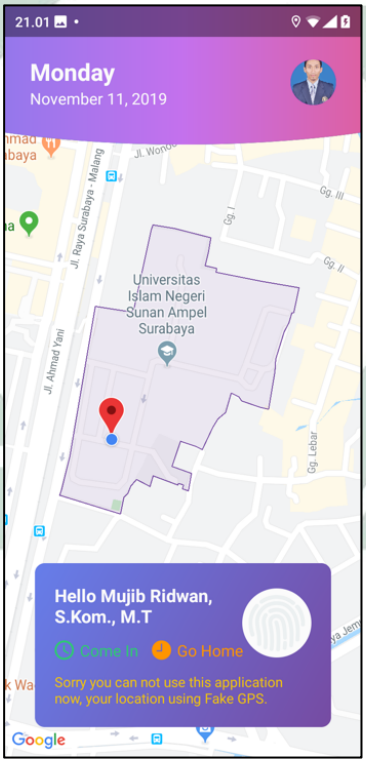
Tampilan *floating container* berbeda ketika posisi pengguna berada di dalam poligon atau sebaliknya. Saat posisi pengguna berada diluar poligon, bagian informasi dan tombol *fingerprint* yang terdapat di *floating container* akan berubah. Kalimat informasi akan berganti seperi berikut, *"sorry you can not use this application now, you are outside Islamic State University of Sunan Ampel Surabaya"*. Sedangkan tombol *fingerprint* akan di *disable* dan berubah warna menjadi abu-abu. Penggunaan aplikasi diluar wilayah poligon ditunjukan dalam Gambar 4.17.



Gambar 4.17: UI *Home Screen* Saat Posisi Pengguna di Luar Poligon

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Untuk membatasi penyalahgunaan aplikasi oleh pengguna untuk dijalankan diluar batas poligon yang sudah ditentukan, maka dibuatlah fungsi untuk mendeteksi penggunaan *mock location*. Ketika pengguna memaksa menggunakan *mock location*, seperti halnya aplikasi *fake GPS*, sistem akan menutup fungsi *fingerprint*, ditandai dengan berubahnya tombol *fingerprint* menjadi *disabled* dan warnanya menjadi abu-abu, serta kalimat informasi pada *floating container* berubah menjadi kalimat peringatan seperti berikut, *"sorry you can not use this application now, your location using Fake GPS"*. Tampilan aplikasi pendataan kehadiran pegawai berbasis *mobile* ketika menggunakan aplikasi *mock location* ditunjukan pada Gambar 4.18.



Gambar 4.18: UI *Home Screen* Saat Menggunakan *Fake GPS*

Desain antarmuka atau UI pada aplikasi ini dibuat sederhana dan simpel untuk mempermudah pengguna dalam menggunakan aplikasi, serta memberikan efisiensi waktu penggunaan. Objek dasar pada tampilan *home* dibuat menjadi peta digital dengan tujuan untuk memudahkan pengguna dalam mengetahui posisinya saat menggunakan aplikasi. Tombol *fingerprint* di desain dalam ukuran besar berguna dalam memudahkan melakukan *fingerprint*, walaupun fungsi utama

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

tombol ini untuk memicu fungsi *fingerprint scanner* pada perangkat *mobile*. Jadi pembuatan desain aplikasi ini difokuskan untuk *user experience* (UX) yang baik dan dapat digunakan oleh pengguna baru.

4.13 Pengujian Fungsionalitas Aplikasi

Sebagai salah satu tahap pengembangan aplikasi, tahap pengujian ini dijalankan untuk menguji fungsionalitas aplikasi dapat berjalan secara benar. Jika dalam pengujian ditemukan *bug*, akan dilakukan perbaikan sebelum aplikasi siap di *release*. Pengujian dilakukan sesuai dengan skenario pengujian yang sudah dibuat pada Tabel 3.1, hasilnya dimuat dalam Tabel 4.1.

Tabel 4.1: Hasil Pengujian Berdasarkan *Test Case Scenario*

No.	***Timestamp***	***Untitled Question***	***Expected Result***	***Actual Result***	***Status***
1.	*Check device*	*Valid*	Tampilan dialihkan ke *home screen*	Tampilan dialihkan ke *home screen*	Pass
		Invalid	Tampil *pop up* pemberitahuan bahwa perangkat *smartphone* tidak mempunyai sensor *fingerprint*	Tampil *pop up* pemberitahuan bahwa perangkat *smartphone* tidak mempunyai sensor *fingerprint*	Pass
		Invalid	Tampilan dialihkan ke *register screen*	Tampilan dialihkan ke *register screen*	Pass
		Invalid	Tampilan dialihkan ke *register screen*	Tampilan dialihkan ke *register screen*	Pass
2.	*Register device*	*Valid*	Muncul *pop up* informasi sukses, kemudian tampilan dialihkan ke *home screen*	Muncul *pop up* informasi sukses, kemudian tampilan dialihkan ke *home screen*	Pass
		Invalid	Muncul *pop up* pemberitahuan *register* gagal dilakukan karena NIP kurang dari 7 karakter	Muncul *pop up* pemberitahuan *register* gagal dilakukan karena NIP kurang dari 7 karakter	Pass

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Tabel 4.1: Hasil Pengujian Berdasarkan *Test Case Scenario* lanjutan

No.	***Timestamp***	***Untitled Question***	***Expected Result***	***Actual Result***	***Status***
		Invalid	Muncul *pop up* pemberitahuan *register* gagal dilakukan karena NIP tidak ditemukan dalam *database employee*	Muncul *pop up* pemberitahuan *register* gagal dilakukan karena NIP tidak ditemukan dalam *database employee*	Pass
		Invalid	Tombol "Daftar" akan tetap menjadi ikon *loading*	Tombol "Daftar" akan tetap menjadi ikon *loading*	Pass
3.	*Shows Map*	*Valid*	Tampil peta digital	Tampil peta digital	Pass
		Invalid	Peta digital tidak dapat ditampilkan, dan berwarna abu-abu	Peta digital tidak dapat ditampilkan, dan berwarna abu-abu	Pass
4.	*GPS Position*	*Valid*	Tampil posisi pengguna pada peta digital, posisi pengguna bergerak mengikuti langkah	Tampil posisi pengguna pada peta digital, posisi pengguna bergerak mengikuti langkah	Pass
		Invalid	Muncul *pop up* pemberitahuan untuk menyalakan GPS	Muncul *pop up* pemberitahuan untuk menyalakan GPS	Pass
5.	*Mock Location*	*Valid*	Tampil pemberitahunan bahwa lokasi terdeteksi menggunakan *mock location* dan tombol *fingerprint* dinonaktifkan	Tampil pemberitahunan bahwa lokasi terdeteksi menggunakan *mock location* dan tombol *fingerprint* dinonaktifkan	Pass
6.	*Fingerprint Attendance*	*Valid*	Muncul *pop up* pemberitahuan sukses, lalu tampil *attendance time* pada *home screen*	Muncul *pop up* pemberitahuan sukses, lalu tampil *attendance time* pada *home screen*	Pass

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Tabel 4.1: Hasil Pengujian Berdasarkan *Test Case Scenario* lanjutan

No.	***Timestamp***	***Untitled Question***	***Expected Result***	***Actual Result***	***Status***
		Invalid	Ikon *fingerprint* tetap berubah mendaji *loading* dan data tidak dapat tersimpan	Ikon *fingerprint* tetap berubah mendaji *loading* dan data tidak dapat tersimpan	Pass

Hasil yang ada pada Tabel 4.1 merupakan hasil pengujian terahir yang dilakukan pada aplikasi. Pada pengujian terakhir ini sudah tidak ditemukan *bug* yang menggangu jalanya aplikasi berdasarkan dengan *test screnario* yang ada. Dalam 15 pengujian yang dilakukan didapatkan hasil bahwa aplikasi sudah berhasil melewati pengujian dalam beberapa skenario yang dilakukan.

4.14 Evaluasi Aplikasi

Evaluasi dilakukan untuk menguji aplikasi ini berdasarkan beberapa tindakan yang dilakukan. Pengujian ini meliputi, pengujian aplikasi untuk dapat digunakan di sekitar batas-batas poligon UINSA Surabaya. Pengujian kedua dilakukan untuk melihat *latency* jaringan seluler ketika menggunakan aplikasi ini, dengan indikator jaringan dan sistem operasi yang digunakan. Indikator jarigan dibagi menjadi beberapa bagian, diantaranya adalah 4G, 3G dan 2G, sedangkan sistem operasi yang diuji terbatas pada Android 8, Android 9, IOS 12 dan IOS 13.

Didapatkan data hasil pengujian yang berbeda-beda di setiap tindakan yang diberikan. Pada pengujian pertama aplikasi diuji pada batas poligon untuk mendapatkan data koordinat pengguna sesuai yang diharapkan. Pengujian ini menghasilkan data mentah berupa titik koordinat pengguna, sesuai yang ditunjukan pada Tabel 4.2 di bawah.

Tabel 4.2: *Raw Data* Hasil Pengujian Metode PNPOLY

No	**Titik Koordinat**	***Expected***	***Actual***	**Tempat**	**Sistem Operasi**
1	[-7.3203298, 112.7340211]	True	False	Pagar Barat Laut (bagian dalam)	Android 9
2	[-7.3206151, 112.7341895]	True	True	Pagar Barat Laut (bagian dalam)	Android 8

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Tabel 4.2: *Raw Data* Hasil Pengujian Metode PNPOLY lanjutan

No	Titik Koordinat	*Expected*	*Actual*	Tempat	Sistem Operasi
3	[-7.320472053258978, 112.7340920549936]	True	True	Pagar Barat Laut (bagian dalam)	IOS 12
4	[-7.320466588622292, 112.73412821853168]	True	True	Pagar Barat Laut (bagian dalam)	IOS 13
5	[-7.3210081,112.7348703]	True	True	Pagar Utara (bagian dalam)	Android 9
6	[-7.321044720094862, 112.7348669147534]	True	True	Pagar Utara (bagian dalam)	IOS 13
7	[-7.3211165,112.7346289]	True	True	Pagar Utara (bagian dalam)	Android 8
8	[-7.321232593849719, 112.73453620713646]	True	True	Pagar Utara (bagian dalam)	IOS 12
9	[-7.3212348,112.7359668]	True	True	Pagar Timur Laut (bagian dalam)	Android 9
10	[-7.32122386806395, 112.73592618304559]	True	True	Pagar Timur Laut (bagian dalam)	IOS 12
11	[-7.3213666,112.735975]	True	True	Pagar Timur Laut (bagian dalam)	Android 8
12	[-7.321322287842681, 112.73579220086194]	True	True	Pagar Timur Laut (bagian dalam)	IOS 13
13	[-7.3215686,112.7359779]	True	True	Pagar Timur (bagian dalam)	Android 9
14	[-7.321736756719017, 112.73580254997381]	True	True	Pagar Timur (bagian dalam)	IOS 12
15	[-7.322175502987337, 112.73580593609684]	True	False	Pagar Timur (bagian dalam)	IOS 13
16	[-7.3221012,112.7357606]	True	True	Pagar Timur (bagian dalam)	Android 8
17	[-7.3230468,112.7330661]	True	True	Pagar Barat (bagian dalam)	Android 9
18	[-7.3229797,112.7330098]	True	True	Pagar Barat (bagian dalam)	Android 8
19	[-7.323119990075146, 112.73281810790625]	True	False	Pagar Barat (bagian dalam)	IOS 13
20	[-7.32294111047622, 112.73309737454423]	True	True	Pagar Barat (bagian dalam)	IOS 12
21	[-7.321623,112.7332184]	True	True	Pagar Samping Masjid (bagian luar)	Android 9

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Tabel 4.2: *Raw Data* Hasil Pengujian Metode PNPOLY lanjutan

No	Titik Koordinat	*Expected*	*Actual*	Tempat	Sistem Operasi
22	[-7.3215164,112.7332053]	True	False	Pagar Samping Masjid (bagian luar)	Android 8
23	[-7.3214890844079665, 112.73324647696359]	True	True	Pagar Samping Masjid (bagian luar)	IOS 13
24	[-7.3214782588252065, 112.7332418785549]	True	True	Pagar Samping Masjid (bagian luar)	IOS 12
25	[-7.3203289,112.7340148]	True	False	Pagar Barat Laut (bagian dalam)	Android 9
26	[-7.3206151,112.7341895]	True	True	Pagar Barat Laut (bagian dalam)	Android 8
27	[-7.320390664979183, 112.73406548436054]	True	True	Pagar Barat Laut (bagian dalam)	IOS 12
28	[-7.320286252666553, 112.73397126674318]	True	False	Pagar Barat Laut (bagian dalam)	IOS 13
29	[-7.3210058,112.7348728]	True	True	Pagar Utara (bagian dalam)	Android 9
30	[-7.3210536089056175, 112.73487104733317]	True	True	Pagar Utara (bagian dalam)	IOS 13
31	[-7.3211165,112.7346289]	True	True	Pagar Utara (bagian dalam)	Android 8
32	[-7.320994371555113, 112.73485782566735]	True	True	Pagar Utara (bagian dalam)	IOS 12
33	[-7.3212385,112.7359686]	True	True	Pagar Timur Laut (bagian dalam)	Android 9
34	[-7.321192645474636, 112.73603498014876]	True	True	Pagar Timur Laut (bagian dalam)	IOS 12
35	[-7.3211727,112.7360671]	True	True	Pagar Timur Laut (bagian dalam)	Android 8
36	[-7.321280064477779, 112.73596878174114]	True	True	Pagar Timur Laut (bagian dalam)	IOS 13
37	[-7.3215742,112.7359668]	True	True	Pagar Timur (bagian dalam)	Android 9
38	[-7.321931677877271, 112.73577329713174]	True	True	Pagar Timur (bagian dalam)	IOS 12

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Tabel 4.2: *Raw Data* Hasil Pengujian Metode PNPOLY lanjutan

No	Titik Koordinat	*Expected*	*Actual*	Tempat	Sistem Operasi
39	[-7.322182276417008, 112.73581102054287]	True	False	Pagar Timur (bagian dalam)	IOS 13
40	[-7.3220931,112.7356721]	True	True	Pagar Timur (bagian dalam)	Android 8
41	[-7.3230469,112.7329876]	True	True	Pagar Barat (bagian dalam)	Android 9
42	[-7.3229797,112.7330098]	True	True	Pagar Barat (bagian dalam)	Android 8
43	[-7.323119990075146, 112.73281810790625]	True	False	Pagar Barat (bagian dalam)	IOS 13
44	[-7.32294111047622, 112.73309737454423]	True	True	Pagar Barat (bagian dalam)	IOS 12
45	[-7.3216039,112.7332128]	True	True	Pagar Samping Masjid (bagian luar)	Android 9
46	[-7.3215526,112.7332331]	True	True	Pagar Samping Masjid (bagian luar)	Android 8
47	[-7.321456499680142, 112.73325718613904]	True	True	Pagar Samping Masjid (bagian luar)	IOS 13
48	[-7.321484922438228, 112.73324179473587]	True	True	Pagar Samping Masjid (bagian luar)	IOS 12

Data mentah dari hasil pengujian dilakukan pengecekan ulang untuk menentukan kenyataan kebenaran perhitungan metode. Harapan yang diinginkan adalah semua pengujian bernilai *true*, karena pengujian dilakukan di dalam poligon yang sudah ditentukan. Hasil dari perhitungan ulang didapatkan kenyataan yang berbeda dari yang diharapkan. Terdapat 8 koordinat yang berindikasi berada diluar poligon, dan sisanya bekerja dengan baik. Hasil pengecekan ulang dikelompokan kedalam Tabel 4.3 berikut.

Tabel 4.3: Hasil Pengujian Metode PNPOLY Berdasarkan Koordinat GPS

	Expected Result	***Actual Result***
Sukses	48	40
Gagal	0	8
Total	48	48
Persentase Kesuksesan	100%	83%
Persentase Kegagalan	0%	17%

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Pengujian pertama ini dilakukan sebanyak 48 kali dengan koordinat pengguna berada di dalam poligon. Dari 48 pengujian yang dilakukan, metode PNPOLY pada aplikasi dapat memproses semua koordinat tersebut dan menghasilkan nilai *true*. Sehingga aplikasi dapat digunakan untuk *attendance system* sebanyak pengujian yang dilakukan.

Setelah tahap pengujian pertama selesai dilakukan, didapatkan hasil yang tidak sesuai dengan yang diharapkan. Kenyataan yang didapat dari pengujian menunjukan 40 titik koordinat yang diuji berada di dalam poligon, tetapi 8 titik sisanya berada diluar poligon. Berdasarkan perhitungan persentase didapatkan hasil sebanyak 83% titik koordinat berhasil di hitung secara benar oleh metode PNPOLY yang diimplementasikan kedalam aplikasi, dan 17% sisanya menjadi margin eror. Karena titik koordinat yang terambil dari GPS berbeda-beda setiap perangkat, hal ini didasari oleh 8 koordinat (gagal dalam pengujian) yang terambil diidentifikasi berada diluar poligon, sedangkan semua pengujian dilakukan di dalam wilayah poligon. Dengan adanya margin error tersebut, maka dilakukan kembali pengelompokan data mentah berdasarkan sistem operasi yang digunakan, terlihat seperti Tabel 4.4.

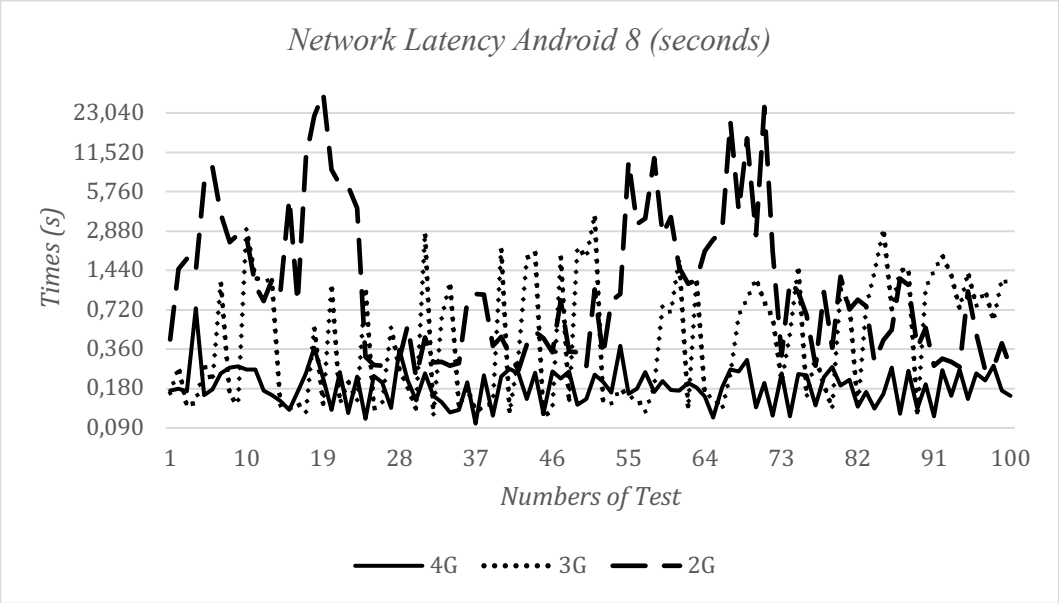
Tabel 4.4: Pengelompokan Kegagalan Pengujian Berdasarkan Sistem Operasi

No.	Sistem Operasi	Tempat
1.	Android 9	Pagar Barat Laut (bagian dalam)
2.	IOS 13	Pagar Timur (bagian dalam)
3.	IOS 13	Pagar Barat (bagian dalam)
4.	Android 8	Pagar Samping Masjid (bagian luar)
5.	Android 9	Pagar Barat Laut (bagian dalam)
6.	IOS 13	Pagar Barat Laut (bagian dalam)
7.	IOS 13	Pagar Timur (bagian dalam)
8.	IOS 13	Pagar Barat (bagian dalam)

Hasil yang didapat memperlihatkan banyakya sistem operasi yang gagal melakukan perhitungan PNPOLY. Pada Android 8 hanya terdapat satu kali kegagalan, Android 9 terdapat 2 kali dan di IOS 13 tingkat kegagalan mencapai 5 kali. Banyaknya kegagalan ini dipengarui oleh tingkat akurasi GPS dari masing-masing perangkat, dalam kasus ini adalah akurasi GPS perangkat pengujian yang menggunakan IOS 13 memiliki tingkat akurasi tidak terlalu baik.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Pengujian kedua dilakukan di dalam poligon dengan posisi pengguna berbeda-beda. Aplikasi ini diuji sebanyak 100 kali di jaringan 2G, 3G dan 4G, serta sistem operasi Android 8, Android 9, IOS 12 dan IOS 13. Gambar 4.19 menunjukan *network latency* ketika menggunakan aplikasi ini di Android 8.



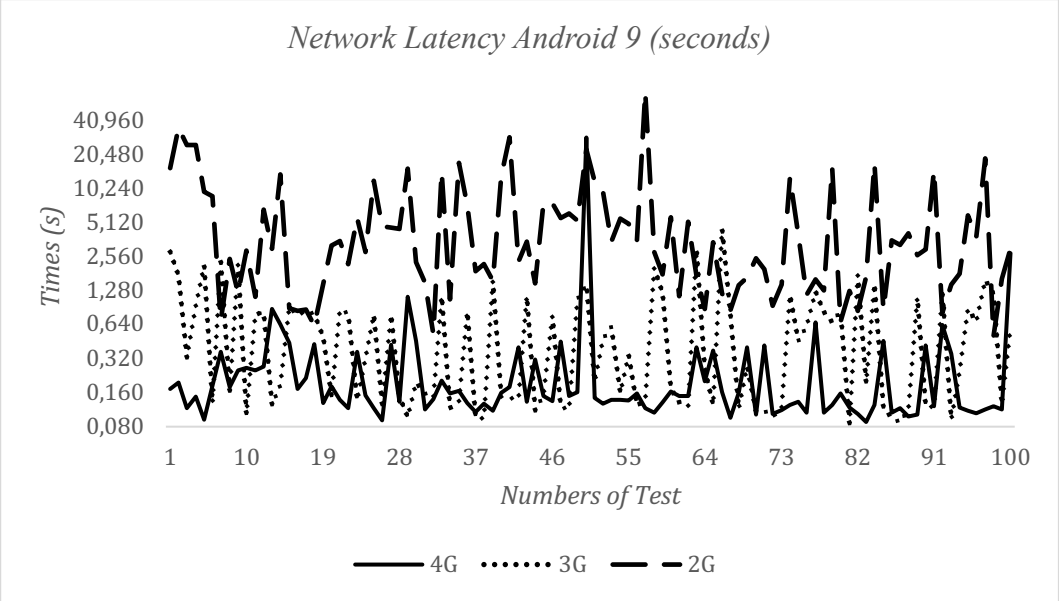
Gambar 4.19: *Network Latency* di Android 8

Hasil pengujian *network lateny* di Android 8 dihasilkan data yang bervariatif seperti terlihat pada Gambar 4.19. Jaringan 2G memiliki *latency* yang besar dibandingkan dengan jaringan 3G maupun 4G, dengan *latency* paling tinggi mencapai 32,497 detik. Tetapi *latency* jaringan ini juga bisa mencapai titik rendah, bahkan bisa lebih rendah dari pada jaringan 3G pada beberapa pengujian. Pada jaringan 3G ringgi-rendah *latency* hampir merata di setiap pengujian. Di beberapa kesempatan *latency* jaringan 3G dapat melampaui jaringan 2G, akan tetapi tidak sampai menyentuh *latency* tertinggi yang dimiliki jaringan 2G. *Latency* terendah untuk jaringan ini juga tidak dapat melampaui jaringan 4G, dimana *latency* terendah mencapai 0,111 detik. Pengujian *latency* pada jarigan 4G Android 8 dihasilkan *network latency* berada di bawah angka 0,800 detik dengan *latency* tertinggi mencapai 0,740 detik dan terendah berada pada 0,097 detik.

Setiap pengujian yang dilakukan pada sistem operasi berbeda menghasilkan *network latency* yang berbeda. Pengujian yang dilakukan di Android 9 jauh berbeda

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

dibandingkan dengan Android 8, meskipun sama-sama satu sistem operasi, hanya saja berbeda versi. Berikut ini hasil pengujian *network latency* di Android 9 dimuat dalam Gambar 4.20.



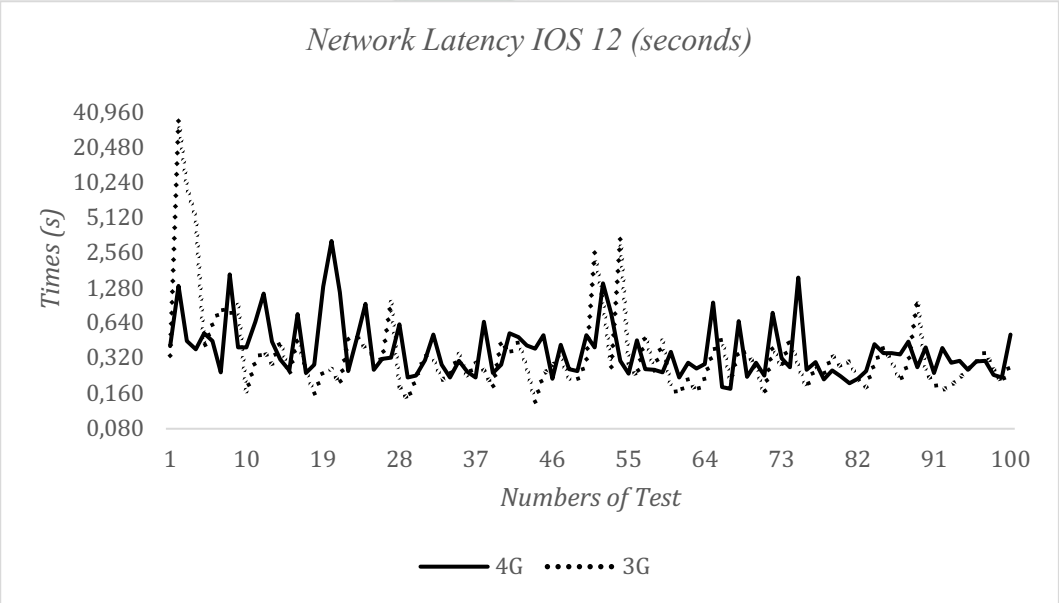
Gambar 4.20: *Network Latency* di Android 9

Hasil pengujian network latency di Android 9 menunjukan peningkatan *latency* di jaringan 2G sangat besar, dengan nilai *latency* tertinggi mencapai 73,346 detik dan beberapa *latency* yang berada mencapai angka 10 detik. Kebanyakan *latency* jaringan 2G pada sistem operasi ini hampir tidak mencapai ketinggian jaringan 3G. Pada jaringan 3G tingkat ketinggian *latency* menunjukan kemiripan, jangkauan latency setiap pengujian tidak terlalu jauh. Tetapi ada beberapa pengujian dengan tingkat *latency* yang tinggi, melebihi *latency* pada jaringan 2G, meskipun tidak sampai mencapai tingkat paling tinggi jaringan 2G. Tinggi-rendah *latency* di jaringan 4G Android 9 cukup stabil, meskipun pada beberapa pengujian menunjukan peningkatan *latency*. Tingkat terendah *latency* di jaringan ini tidak dapat mengalahkan *latency* jaringan 3G, dimana jaringan 3G dapat mencapai 0,85 detik.

Pengujian yang dilakukan pada IOS mendapatkan hasil yang sangat jauh berbeda dibandingkan dengan sistem operasi Android. Pada IOS pengujian yang dilakukan pada jaringan 2G tidak dapat dilakukan. Hasil yang didapat ketika

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

melakukan pengujian di jaringan ini membutuhkan waktu yang sangat lama, bahkan tidak dapat mengirim *request* ke *server*. Hasil studi terkait menghasilkan hipotesis bahwa arsitektur sistem IOS membatasi penggunaan jaringan 2G pada aplikasi, dikarenakan IOS berfokus pada peningkatan performa. Hipotesis ini didukung dengan kenyataan beberapa aplikasi hiburan milik provider besar juga tidak dapat dijalankan pada jaringan 2G milik IOS. Pengujian *network latency* pada IOS dibagi menjadi dua bagian, yaitu IOS 12 dan IOS 13, Gambar 4.21 menunjukan hasil pengujian *network latency* pada IOS 12.



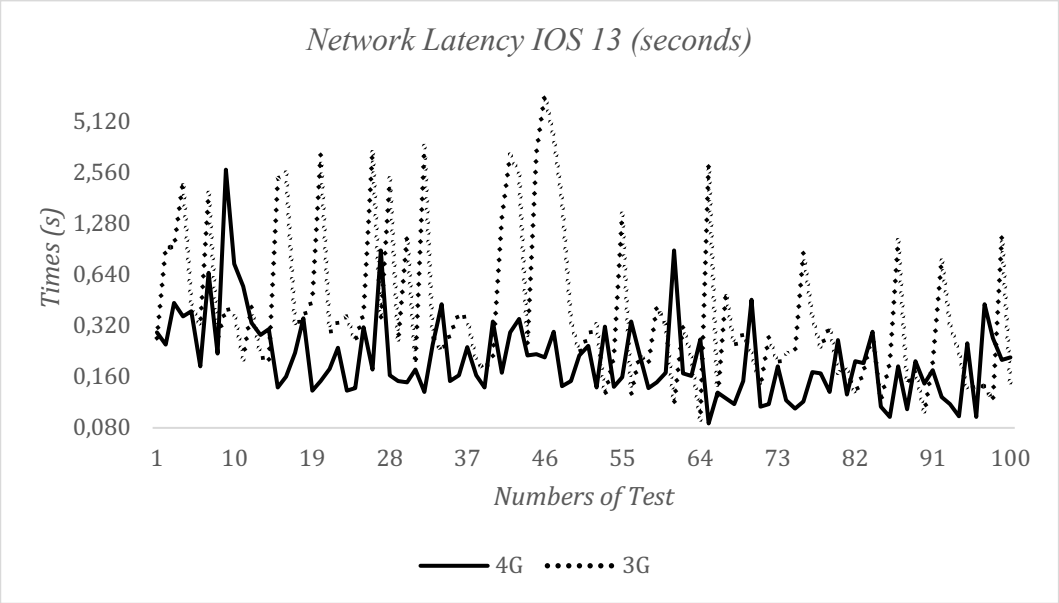
Gambar 4.21: *Network Latency* di IOS 12

Latency jaringan 3G dan 4G pada IOS 12 hampir mirip satu sama lain, ketinggian *latency* dari kedua jaringan ini saling naik turun. Tetapi ada pengujian yang menghasilkan *latency* tertinggi di jaringan 3G mencapai 35,667 detik. Meskipun begitu ketinggian *latency* pada jaringan 4G ada juga yang mencapai 3,255 detik. Ketinggian yang hampir sama ini menjadikan rata-rata *latency* jaringan pada IOS 12 tidak terlalu jauh antara jaringan 3G dan jaringan 4G.

Pengujian yang dilakukan pada IOS 13 juga hanya terbatas pada jaringan 3G dan jaringan 4G sesuai yang ditunjukan pada Gambar 4.22. Hasil dari pengujian didapatkan kenaikan *network latency* terdapat pada jaringan 3G, dengan *latency* tertinggi mencapai 7,430 detik. Beberapa *latency* tinggi dari hasil pengujian pada

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

sistem operasi ini berada diatas 1 detik. Pengujian *latency* pada jaringan 4G hampir sejajar dengan hasil *latency* jaringan 3G, meskipun kenaikan *latency* jaringan ini masih dibawah jaringan 3G. Tetapi ada pengujian jaringan 4G yang menghasilkan *latency* sama dengan jaringan 3G, dengan nilai sebesar 2,579 detik.



Gambar 4.22: *Network Latency* di IOS 13

Hasil dari pengujian *network latency* di setiap jaringan pada sistem operasi Android dan IOS dilakukan perhitungan rata-rata untuk mendapatkan pilihan jaringan terbaik ketika menggunakan aplikasi ini. Tabel 4.5 menunjukan rata-rata *network latency* dari hasil pengujian.

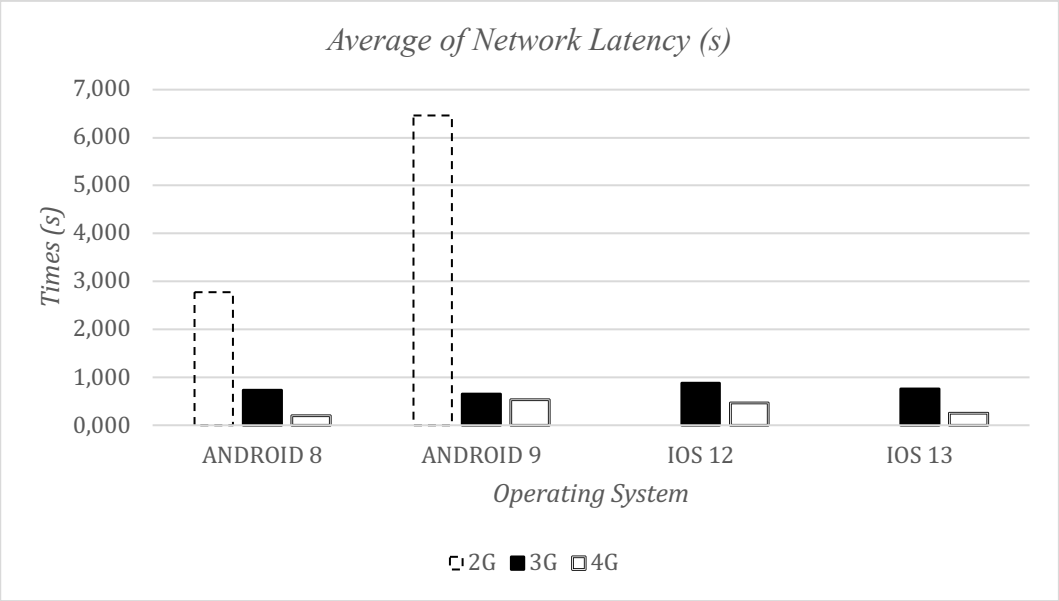
Tabel 4.5: Rata-Rata *Network Latency (Seconds)*

	Android 8	**Android 9**	**IOS 12**	**IOS 13**
2G	2,776s	6,459s	NULL	NULL
3G	0,732s	0,657s	0,875s	0,761s
4G	0,198s	0,533s	0,461s	0,254s

Untuk mempermudah klasifikasi informasi, data pada Tabel 4.5 dibuat grafik seperti yang terlihat pada Gambar 4.23. Hasil dari perbandingan rata-rata *network latency* didapatkan bahwa jaringan 2G Android 9 menempati urutan pertama untuk *network latency* paling besar, disusul dengan Android 8, dengan rata-rata *latency* sebesar 6,459 detik dan 2,776 detik. Untuk *latency* jaringan 3G hampir sama di semua sistem operasi. *Latency* jaringan 3G di Android 8 sebesar 0,732 detik,

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Android 9 sebesar 0, 657 detik, IOS 12 sebesar 0,875 detik dan IOS 13 sebesar 0,761 detik. Rata-rata *latency* pada jaringan 4G mendapatkan hasil yang memuaskan dari setiap sistem operasi. Urutan *latency* terkecil dipegang oleh sistem operasi Android 8 dengan nilai sebesar 0,198 detik, lalu disusul oleh IOS 13 sebesar 0,254 detik, kemudian IOS 12 sebesar 0,461, dan yang terakhir adalah Android 9 dengan nilai 0,533 detik.



Gambar 4.23: Grafik Rata-Rata *Network Latency*

Rata-rata *network latency* berpengaruh terhadap performa koneksi saat menggunakan aplikasi ini. Dari pengujian yang dilakukan diperoleh hasil bahwa jaringan 4G memiliki *latency* paling kecil. Sedangkan jaringan 2G menghasilkan *latency* yang besar, bahkan tidak dapat digunakan pada IOS. Untuk jaringan 3G diperoleh hasil yang hampir sama di semua rata-rata pengujian *network latency* pada sistem operasi Android maupun IOS.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

BAB V
PENUTUP

5.1 Kesimpulan

Dari hasil penelitian yang dilakukan untuk mengembangkan aplikasi pendataan kehadiran pegawai berbasis *mobile*, dapat ditarik kesimpulan sebagai berikut.

1. Pengembangan aplikasi menggunakan metode *waterfall*. Pada tahap analisis didapatkan kebutuhan sistem dengan arsitektur dasar pengembangan menggunakan *framework react native* yang berbasis pada bahasa pemrograman *javascript*. Di tahap desain dibuat rancangan arsitektur sistem, desain UML dan desain *database* yang digunakan. Implementasi metode *Point Inclusion in Polygon Test* (PNPOLY) ke dalam bahasa *javascript* digunakan untuk menentukan posisi pengguna berada di dalam poligon. Hasil pengujian fungsionalitas menyatakan bahwa aplikasi ini dapat melewati semua *scenario test* yang diujikan.
2. Hasil percobaan untuk menguji keberhasilan metode dalam menentukan posisi pengguna memberikan hasil bahwa metode PNPOLY dapat menentukan posisi pengguna berada di dalam poligon atau diluar poligon, dengan persentase keberhasilan sebesar 83% dan margin eror sebesar 17%, tergantung dengan perangkat yang digunakan. Hasil evaluasi ulang memperlihatkan kegagalan paling banyak terdapat pada perangkat yang menggunakan IOS 13 mencapai 5 kali dari total keseluruhan pengujian pertama.
3. Hasil evaluasi kedua yang dilakukan untuk mengukur *network latency* dalam jaringan dan sistem operasi yang berbeda didapatkan informasi bahwa penggunaan maksimal aplikasi ini berada pada jaringan 4G di Android 8 dengan rata-rata *latency* sebesar 0,198 detik.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

5.2 Saran

Saran yang dapat penulis berikan dalam penggunaan aplikasi ini atau pengembangan penelitian yang akan datang antara lain.

1. Disarankan untuk menggunakan jaringan 4G pada Android 8 ketika menggunakan aplikasi ini. Karena hasil pengujian *network latency* menunjukan *latency* terendah berada pada jaringan di sistem operasi tersebut.
2. Mengembangkan sistem pendataan kehadiran berbasis biometrik *mobile* dengan indikator lain seperti *face ID*, *voice ID*, maupun *iris ID*. Karena perkembangan teknologi *mobile* ke masa depan akan lebih banyak menggunakan sistem keamanan berbasis biometrik yang lain, seperti yang terdapat pada perangkat iPhone yang terbaru menghilangkan sensor *fingerprint* dan menggantinya dengan sensor *face ID*.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

DAFTAR PUSTAKA

Adetokunbo, A., & Basirat, A. (2014). Software Engineering Methodologies: A Review of The Waterfall Model and Object-Oriented Approach. *International Journal of Scientific & Engineering Research*, *4*(7), 427–434.

Airbnb. (2017). React Native Map. Retrieved October 29, 2019, from https://github.com/react-native-community/react-native-maps

Al Fawareh, H. M. azer, & Jusoh, S. (2017). The use and effects of smartphones in higher education. *International Journal of Interactive Mobile Technologies*, *11*(6), 103–111. https://doi.org/10.3991/ijim.v11i6.7453

Alsaadi, I. M. (2015). Physiological Biometric Authentication Systems Advantages Disadvantages And Future Development A Review. *International Journal of Scientific & Technology Research*, *4*(8), 285–289.

AlShehri, H., Hussain, M., AboAlSamh, H., & AlZuair, M. (2018). A large-scale study of fingerprint matching systems for sensor interoperability problem. *Sensors (Switzerland)*, *18*(4), 1–18. https://doi.org/10.3390/s18041008

B'Far, R., Fielding, R. T., Brady, D., B'Far, R., & Fielding, R. T. (2009). Introduction to UML. *Mobile Computing Principles*, 155–228. https://doi.org/10.1017/cbo9780511546969.005

Cachia, E. (2004). *UML FUNDAMENTALS*. Retrieved from http://staff.um.edu.mt/ecac1/files/csa2130_UML.pdf

Chang, Y. (2018). *Fake GPS Defender : A Server-side Solution to Detect Fake GPS*. (c), 36–41.

ESOA. (2019). Latency in Communications Networks. Retrieved October 5, 2019, from https://www.esoa.net/Resources/1527-ESOA-Latency-Update-Proof4.pdf

Expo. (2018). React Native Expo. Retrieved October 29, 2019, from https://github.com/expo/expo/commit/274cb10331fae2343315227a78b36c7dca2d74d6

Facebook Inc. (2019). React Native. Retrieved October 29, 2019, from https://facebook.github.io/react-native/docs/getting-started

Fang, C., & Zhang, S. (2018). Geographic information retrieval method for geography mark-Up language data. *ISPRS International Journal of Geo-Information*, *7*(3). https://doi.org/10.3390/ijgi7030089

Foster, E. C., Godbole, S., Foster, E. C., & Godbole, S. (2016). Overview of SQL. In *Database Systems*. https://doi.org/10.1007/978-1-4842-1191-5_10

Franklin, W. R. (2006). PNPOLY - Point Inclusion in Polygon Test - WR Franklin (WRF). Retrieved from http://www.ecse.rpi.edu/Homepages/wrf/Research/Short_Notes/pnpoly.html

Garcia-Molina, H., Ullman, J. D., & Widom, J. (2009). *DATABASE SYSTEMS The Complete Book* (2nd ed.; M. J. Horton, T. Dunkelberger, M. Haggerty, M. Waples, C. Kelly, S. Disanno, … A. Fischer, Eds.). New Jersey: Pearson Education Inc.

Gilmore, W. J. (2010). *Begining PHP & MySQL*. Retrieved from http://dialnet.unirioja.es/servlet/libro?codigo=318317

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Google Inc. (2019). Google Map API. Retrieved October 29, 2019, from https://developers.google.com/maps/documentation

Grässle, P., Baumann, H., & Baumann, P. (2005). UML 2.0 in Action. In *Packt Publishing* (1st ed.). Birmingham: Packt Publishing Ltd.

Herbei, M. V., Ciolac, V., & Smuleac, A. (2010). Georeferencing of topographical maps using the software arcgis. *Research Journal of Agricultural Science*, *42*(3), 595–606.

James Rumbaugh, Ivar Jacobson, G. B. (2013). The Unified Modeling Language Reference Manual. In *Journal of Chemical Information and Modeling* (Vol. 53). https://doi.org/10.1017/CBO9781107415324.004

Jo, Y. H., Jeon, S. Y., Im, J. H., & Lee, M. K. (2016). Security analysis and improvement of fingerprint authentication for smartphones. *Mobile Information Systems*, *2016*(Krait 400). https://doi.org/10.1155/2016/8973828

Jones, C. B., & Purves, R. S. (2008). Geographical information retrieval. *International Journal of Geographical Information Science*, *22*(3), 219–228. https://doi.org/10.1080/13658810701626343

Kaner, C. (2003). An introduction to scenario testing. *Center for Software Testing Education and Research*, (June), 1–10. Retrieved from http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:An+Introduction+to+Scenario+Testing#1

Kumar, G. N., & Bangi, M. (2018). An Extension to Winding Number and Point-in-Polygon Algorithm. *IFAC-PapersOnLine*. https://doi.org/10.1016/j.ifacol.2018.05.092

Malathi, R., & Jeberson Retna Raj, R. (2016). An Integrated Approach of Physical Biometric Authentication System. *Procedia Computer Science*. https://doi.org/10.1016/j.procs.2016.05.271

Mil'shtein, S. (2005). *Final Report for the period 1 st*. Retrieved from www.ukstrandings.org

OpenStreetMap Foundation. (2019). OpenStreetMap. Retrieved October 29, 2019, from https://www.openstreetmap.org/way/470103964

Padmanabhan, B. (2012). Unified Modeling Language (UML) Overview. *Principles of Software Engineering*, 1–20.

Purves, R. S., Clough, P., Jones, C. B., Hall, M. H., & Murdock, V. (2018). Geographic information retrieval: Progress and challenges in spatial search of text. *Foundations and Trends in Information Retrieval*, *12*(2–3), 164–318. https://doi.org/10.1561/1500000034

Ramadhani, W. (2018). Bintu Syati' Dan Penafsirannya Terhadap Surah Al-'asr Dalam Kitab At-tafsir Al-bayani Lil Qur'anil Karim. *Jurnal At-Tibyan*. https://doi.org/10.32505/tibyan.v3i2.717

Rauschmayer, A. (2019). *JavaScript for impatient programmers* (F. Caye, Ed.). Retrieved from exploringjs.com

S., S., D., V., & Waghmare, S. (2018). Remote Biometric Authentication System using Android Phone. *International Journal of Computer Applications*, *180*(33), 6–12. https://doi.org/10.5120/ijca2018916811

Soewito, B., Gaol, F. L., Simanjuntak, E., & Gunawan, F. E. (2015). Attendance system on Android smartphone. *ICCEREC 2015 - International Conference*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

on Control, Electronics, Renewable Energy and Communications, 208–211. https://doi.org/10.1109/ICCEREC.2015.7337046

Sultana, S., Enayet, A., & Mouri, I. J. (2015). A Smart, Location Based Time and Attendance Tracking System using Android Application. *International Journal of Computer Science, Engineering and Information Technology*, *5*(1), 01–05. https://doi.org/10.5121/ijcseit.2015.5101

Sunday, D. (2012). Inclusion of a Point in a Polygon. Retrieved October 5, 2019, from http://geomalgorithms.com website: http://geomalgorithms.com/a03-_inclusion.html

Wadhwa, P. (2017). Attendance System Using Android integrated Biometric Fingerprint Recognition. *International Research Journal of Engineering and Technology(IRJET)*, *4*(6), 1069–1073. Retrieved from https://irjet.net/archives/V4/i6/IRJET-V4I6197.pdf

Wood, L. E. (2011). Locating People, Places, and Things: Situating GIS in the Intelligent Network Landscape. *Global Media Journal*, *10*(19), 1–11.

Ye, Y., Guangrui, F., & Shiqi, Q. (2013). An algorithm for judging points inside or outside a polygon. *Proceedings - 2013 7th International Conference on Image and Graphics, ICIG 2013*, *1*, 690–693. https://doi.org/10.1109/ICIG.2013.140

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id