BAB II
LANDASAN TEORI

2.1. Sistem Presensi

Sistem presensi merupakan sebuah sistem yang digunakan untuk mencatat daftar kehadiran setiap anggota instansi. Sistem presensi mencatat identitas anggota dan hadir tidaknya anggota. Sistem presensi juga memiliki hasil akhir laporan pendataan kehadiran anggota instansi (Rubiati & Widya Harahap, 2019). Penerapan sistem presensi dewasa ini banyak jenisnya, seperti presensi manual yang dilakukan dengan cara menggunakan tanda tangan. Presensi dengan menggunakan sistem yang telah terkomputerisasi contohnya presensi menggunakan RFID, presensi menggunakan sidik jari, presensi menggunakan pengenalan wajah, dan lainnya.

2.2 *Geofencing*

Geofencing adalah teknologi yang mendefenisikan batas-batas virtual di sekitar wilayah geografis dunia (Azzami & Kusumaningrum, 2018). Pengurangan radius yang dapat mengurangi pemicu tindakan ditelepon atau perangkat elektronik portabel lainnya. Fungsi dari *geofencing* yang dibuat dengan lokasi terkini dari perangkat *mobile* yaitu: ketika pengguna memasuki atau meninggalkan area geografis yang telah dibuat dapat dideteksi secara otomatis, kemudian dari hasil deteksi tersebut dapat dihasilkan luaran yang diinginkan (Rahman & Putra, 2018). *Geofencing* memungkinkan lansiran otomatis yang akan dihasilkan berdasarkan koordinat yang dimasukkan ke dalam area yang telah ditentukan secara geografis yang digunakan untuk memantau objek bergerak seperti *smartphone*,kendaraan dan lain-lain dengan menggunakan jaringan satelit *Global Positioning Sistem* (GPS) *Geofencing* menggambarkan sebuah area (*geofence*) yang memiliki batas-batas geografis dari suatu peta. *Geofencing* pada umumnya dapat dimanfaatkan untuk membantu melacak pengiriman barang yang dibawa oleh suatu kendaraan, memantau posisi seseorang, menjalankan bisnis komersial tertentu, dan presensi otomatis di suatu perusahaan atau universitas. Ukuran wilayah dari

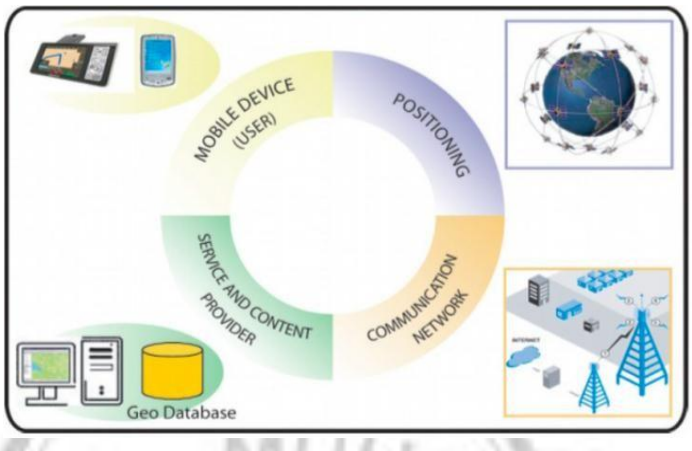
geofencing yaitu berkisar dari beberapa meter sampai beberapa kilometer tergantung dengan gps yang digunakan. Bentuk area sebuah *geofence* yaitu berbentuk sebuah lingkaran (circle) sedangkan mekanisme menentukan area ditentukan oleh *latitude, longitude,* dan radius dari titik yang ditentukan. Di dalam sistem operasi *Android geofencing* dibungkus dalam sebuah *library*. Fitur utama *geofencing* yang ada di sistem operasi *Android* adalah sistem dapat memberikan peringatan berupa notifikasi saat target masuk, tinggal, dan keluar dari area yang sudah ditentukan sebelumnya. Oleh karena itu, *geofencing* juga bisa digunakan untuk meningkatkan kesiapsiagaan terhadap suatu bencana.

2.3. *Location Based Service*(LBS)

Location based service adalah layanan informasi yang di akses menggunakan piranti *mobile* melalui jaringan internet dan seluler serta memanfaatkan kemampuan penunjuk lokasi pada piranti *mobile.* Konsep dari metode *Location Based Service* ini sendiri menggunakan database informasi geografis yang digabungkan dengan teknologi *Global Positioning System* (GPS) yang tertanam di *smartphone* pengguna untuk melacak suatu pergerakan *device* pengguna dan mengirimkan informasi yang dibutuhkan oleh *device* pengguna. (Susanti , Nanda, Thamrin, 2019).

1. *Mobile Device* yaitu sebuah alat yang digunakan oleh pengguna untuk meminta informasi yang dibutuhkan. Perangkat memungkinkan yaitu perangkat yang memiliki fasilitas navigasi seperti PDA, *mobile phone,* laptop dan lainnya.
2. *Communication Network* adalah jaringan selular yang mengirimkan datapengguna dan permintaan layanan.
3. *Positioning Component* biasanya posisi pengguna harus ditentukan untuk pengolahan layanan. Posisi pengguna dapat diperoleh menggunakan jaringan komunikasi atau dengan menggunakan *Global Positioning System* (GPS).
4. *Service and Content Provider* yaitu penyedia layanan informasi data yang dapatdi minta oleh pengguna. Komponen LBS dapat ditunjukan

pada gambar berikut:



Gambar 2. 1 Komponen *Location Based Service* (LBS)

(Sumber :Susanti , 2019)

a. Unsur utama pada *Location Based Service* (LBS) memiliki unsur utama yaitu:

1. *Location* (API Maps) menyediakan perangkat bagi sumber atau *source* untuk *location based service* (LBS), *Application Programming Interface* (API) maps menyediakan fasilitas untuk menampilkan dan memanipulasi peta.

2. *Location Provider* (API Location) menyediakan teknologi pencarian lokasi yang digunakan oleh perangkat. API Location berhubungan dengan data GPS (*Global Positioning System*) dan data lokasi *real-time*. API *Location* berada pada data *Android* yaitu data paket internet yang digunakan oleh perangkat.

b. Cara akses layanan *Location Based Service* (LBS) Pada *platform* ada dua cara yang berbeda untuk mengakses layanan LBS:

1. Inisiatif dari *platform*:
   Pengguna mengirimkan permintaan (teks) untuk informasi tentang layanan di daerah dekat sekitarnya
2. Inisiatif dari pengguna Pengguna register terlebih dahulu untuk menerima tertentu informasi setiap kali dekat dengan tempat pengguna. Pengguna menerima diminta informasi pada item baru apabila di dekat tempat sekitar tersebut.

2.4. *Haversine Formula*

Teorema *Haversine Formula* adalah sebuah persamaan yang penting dalam bidang navigasi, untuk mencari jarak busur antara dua titik pada bola dari *longitude* dan *latitude*. Ini merupakan bentuk persamaan khusus dari trigonometri bola, *law of haversines*, mencari hubungan sisi dan sudut pada garis segitiga dalam bidang bola. *Formula* ini pertama kali ditemukan oleh Jamez Andrew di tahun 1805, dan digunakan pertama kali oleh Josef de Mendoza y Rios di tahun 1801 (Sulistio, 2019). Rumus *Haversine* adalah metode yang digunakan untuk menghitung jarak dari suatu tempat ke tempat tujuan(Azdy & Darnis, 2020). Istilah *haversine* ini sendiri diciptakan pada tahun 1835 oleh Prof. James Inman. Josef de Mendoza y Rios menggunakan *haversine* pertama kali dalam penelitiannya tentang "Masalah Utama *Astronom Nautical*", Proc.Royal Soc, Dec 22. 1796. *Haversine* digunakan untuk menemukan jarak antar bintang. *Haversine formula* memberikan jarak lingkaran besar antara dua titik pada permukaan bola (bumi) berdasarkan bujur dan lintang dengan mengasumsikan jari-jari R 6.367, 45 km, dan lokasi dari 2 titik di koordinat bola (lintang dan bujur) masing - masing adalah lon1,lat1, dan lon2, lat2. Rumus *Haversine* dapat ditulis dengan persamaan sebagai berikut(Nugroho, 2020) :

Rumus *Haversine*:

$$
\begin{aligned}
&X = (lon2-lon1)*\cos((lat1+lat2)/2) \\
&y = (lat2-lat1) \\
&d = x * x + y * y * R
\end{aligned}
\qquad (2.1)
$$

Keteranagan:

X = *Longtitude*(Lintang)

y = *Lattitude*(Bujur)

d = Jarak

R = Radius Bumi = 6371km

1 derajat = 0.0174532925 radian

Haversine formula nantinya akan di gunakan dalam perhitungan jarak antara dua titik GPS. Dalam hal ini adalah titik GPS *user* dan titik GPS koordinat tempat presensi.

2.5. *Android*

Menurut Rana & Sung (2020) *Android* adalah kerangka kerja gabungan portabel sumber yang terbuka bekerja sehubungan dengan kernel *linux* dan desain yang dipisahkan menjadi lima bagian. *Android* memiliki pengertian sistem operasi bebasis *Linux* yang mengelola peragkat keras untuk komputer tablet, ponsel, dan *smartphone*. Pada saat perilisan perdana, *Android* menyatakan melakukan dukungan penuh pada perkembangan perangkat seluler secara standart. Perilisan tersebut dilakukan pada tanggal 5 November 2007 bersama dengan OPH (*Open Handset Alliance*)

2.1.1. Prinsip Kerja *Android*

Prinsip kerja *Android* dapat disamakan dengan system kerja yang ada pada komputer yaitu kemampuan untuk *input*, *proses*, dan *output*. Bukan hanya itu *Android* dan komputer memiliki kemampuan untuk mengelola data dan gambar. Pada ponsel *Android* terdapat komponen penting yang terdapat pada computer seperti RAM, CPU, GPU, dan lain sebagainya. *Operating* sistem pada ponsel *Android* adalah *Android*. *Android* dapat dilkasifikasikan operasi system *linux*, namun hanya saja sistem ini termasuk kedalam perkembangan oleh sistem *Linux*.

2.1.2. Keunggulan *Android*

1. Merupakan sistem operasi dengan *open source*, sehingga dapat mempermudah pengguna untuk mengembangkannya.
2. Os *Android* dalam penjalanannya terdapat banyak pilihan spesifikasi *hardware*.
3. Banyaknya dukungan dari berbagai aplikasi yang beragam.
4. Kerusakan system yang terjadi dapat segera diperbaiki karena OS *Android* mudah untuk dipahami.
5. Sistem operasi yang bekerja secara cepat dan *responsive*

2.6. *Dart*

Dart *language* merupakan salah satu bahasa pemrograman oleh Google yang merupakan bahasa *general-purpose* yang dapat digunakan untuk mengembangkan berbagai platform termasuk web, *mobile server*, dan IOT. Bahasa ini juga merupakan bahasa standar yang digunakan dari flutter(Hanif & Sinambela, 2020). *Dart* merupakan bahasa pemrograman yang mendukung adanya pendefinisian fungsi di luar kelas atau sering disebut dengan *toplevel function*. Dalam *Dar*t. Kode program utama disimpan di dalam fungsi main() sama halnya seperti C/C++. (Budi Raharjo, 2019).

2.7. *Flutter*

Flutter adalah SDK untuk pengembangan aplikasi *mobile* yang dikembangkan oleh Google untuk membangun antarmuka (*user interface*) aplikasi *Android* dan iOS. Rilis pada juni 2018 sama seperti *react native*, *framework* ini dapat digunakan untuk membuat atau mengembangkan aplikasi *mobile* yang dapat berjalan pada *device* iOS dan *Android*. Dibuat menggunakan bahasa C, C++, *Dart* and *Skia* membuat *Flutter* ini menjadi salah satu *framework* yang sangat menarik dan *worth* untuk di pelajari. Hal yang menarik pada *framework* ini adalah semua kode di *compile* dalam kode *native* nya (*Android* NDK, LLVM, AOT-compiled) tanpa ada *intrepeter* pada prosesnya sehingga proses *compile*-nya menjadi lebih cepat. Dari segi penulisan kodenya, *Flutter* ini sangat berbeda dari *react native* dan lebih cenderung mendekati Java *Android* jadi untuk *developer react native* agak sedikit kesulitan untuk memahami kode pada *Flutter* ini. Untuk membuat aplikasi Flutter, diperlukan untuk mengerti bahasa *Dart*. (Budi Sudrajat, 2021)

Salah satu keunggulan *flutter* terletak pada teknologi *multiplatform* lainnya seperti *react native* adalah dalam hal kinerja. *Flutter* menjanjikan aplikasi yang dibuat akan mendapatkan tingkat sebesar 60 frame per

second(Supiana, 2022). Kinerja ini bisa didapatkan karena cara kerja dari flutter sedikit berbeda. Kode-kode yang ditulis dengan menggunakan bahasa *dart* akan diubah menjadi kode C/C++ kemudian dikompilasi secara *native*. Hal inilah yang menyebabkan *flutter* memiliki performa yang hampir setara dengan aplikasi *native*. Flutter bisa berjalan pada sistem operasi *Android* mulai dari versi 4.1 dan *IOS* mulai dari versi 8. Serta dijalankan di perangkat asli maupun *simulator*.

2.8. *Database Firebase*

Firebase Realtime Database memungkinkan pengguna untuk membuat aplikasi kolaboratif dan kaya fitur dengan menyediakan akses yang aman ke *database*, langsung dari kode sisi *client*. Firebase merupakan platform untuk aplikasi *realtime*. Ketika data berubah, maka aplikasi yang terhubung dengan *firebase* akan meng-update secara langsung melalui setiap *device* (perangkat) baik *website* ataupun *mobile* (Sanad, 2019). Data disimpan di drive lokal. Bahkan saat *offline* sekalipun, peristiwa *realtime* terus berlangsung, sehingga pengguna akhir akan merasakan pengalaman yang *responsif*. Ketika koneksi perangkat pulih kembali, *Realtime Database* akan menyinkronkan perubahan data lokal dengan update jarak jauh yang terjadi selama *client offline*, sehingga setiap perbedaan akan otomatis digabungkan. *Realtime Database* adalah database NoSQL, sehingga memiliki pengoptimalan dan fungsionalitas yang berbeda dengan database terkait. (Ashok Kumar S, 2018)

.*Firebase* dapat digunakan untuk membuat aplikasi sesuai kebutuhan baik *mobile* maupun *web*. Adapun fitur yang dimiliki oleh *Firebase* diantaranya:

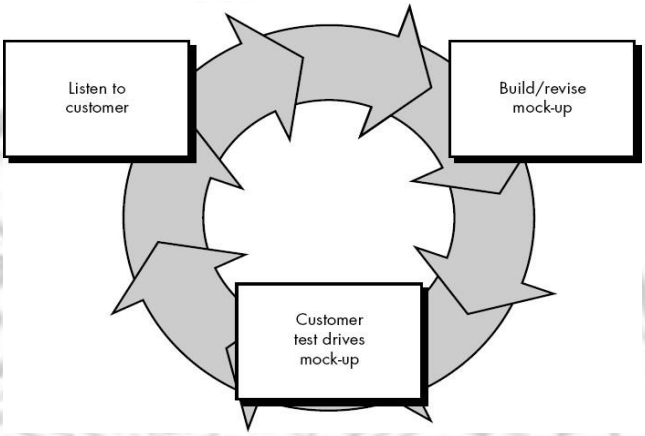
1. *Firebase Authentication Otentikasi Firebase* saat aplikasi perlu mengetahui identitas pengguna untuk memberikan data khusus pengguna, suatu bentuk masuk yang aman dan memungkinkan pengguna masuk dengan kredensial yang dikenal.

2. *The Realtime Database* ini adalah basis data berbasis NoSQL yang di *hosting* secara *cloud*. Ini menyediakan sinkronisasi saat perangkat terhubung dan tersedia saat tidak ada konektivitas jaringan melalui *cache* lokal.
3. *Cloud Storage for Firebase* untuk menyimpan data aplikasi, menyimpan file seperti foto, audio, maupun video.
4. *Firebase Hosting* mendapatkan ruang *hosting* yang dapat digunakan untuk meng-host aset statis seperti HTML, CSS, *JavaScript*, atau gambar.
5. *Firebase Crash Reporting* membantu dengan menyediakan jejak tumpukan semua kerusakan di konsol *Firebase* yang dapat digunakan untuk mencari tahu penyebab *crash*.
6. *The Grow Technologies* memiliki sejumlah teknologi yang dapat digunakan untuk membantu anda mengembangkan aplikasi secara sistematis.
7. *Firebase Cloud Messaging* memungkinkan untuk mengirim pesan dengan andal tanpa biaya. Perangkat yang terhubung menerima pesan-pesan ini dalam waktu kurang dari 500 ms.
8. *App Invites and Dynamic Links* memungkinkan untuk memilih siapa yang akan dikirimi undangan dari daftar yang disortir secara cerdas.
9. *Earning with AdMob* penggunakan iklan banner, iklan pengantara, video, atau iklan asli dalam aplikasi.
10. *Google Analytics for Firebase* membantu memahami *audiens* dengan baik, dapat digunakan untuk terus mengembangkan aplikasi dan mengendalikan teknologi seperti *Firebase Remote Config*, dan *Cloud Messaging*.

2.9. Metode Pengembangan Sistem *Prototype*

Prototype merupakan metodologi pengembangan *software* yang menitik pada pendekatan aspek desain, fungsi dan *user-interface*. *Developer*

dan user fokus pada *user-interface* dan bersama-sama mendefisikan spesifikasi, fungsi, desain dan bagaimana *software* bekerja. Metode *prototype* adalah versi dari sistem atau bagian dari sistem dan dikembangkan dengan cepat untuk memvalidasi persyaratan atau kelayakan dari beberapa keputusan desain yang diperlukan oleh pelanggan.(Gunawan dkk., 2022)



Gambar 2. 2 Metode Pengembangan Sistem *Prototype*

(Sumber : E. Ali, 2019)

Pengembangan dari perancangan sistem ini dalam pelaksanaannya menggunakan tiga tahap siklus pengembangan model *prototype* yaitu:

1. Mendengarkan pelanggan (*Listen* to *Customer*) merupakan tahap pertama dalam merancang sebuah sistem. Pada tahap ini akan menentukan informasi informasi yang dibutuhkan oleh pelanggan agar tercipta sebuah aplikasi sehingga mengarah pada tujuan dibuatnya aplikasi tersebut.
2. Membangun dan memperbaiki *prototype* (*Build*/*revice mockup*) dalam tahap ini dilakukan perancangan dan pengkodean untuk sistem yang diusulkan yang mana tahapannya meliputi perancangan proses-proses yang akan terjadi dalam sistem, perancangan diagram UML yang akan digunakan, perancangan antarmuka keluaran serta dilakukan tahap pengkodean terhadap rancangan rancangan yang telah didefinisikan, kelengkapan *software* dan *hardware*.

3. Pengujian *prototype* pada tahapan ini dilakukan pengujian terhadap sistem yang telah disusun dan melakukan pengenalan terhadap sistem yang telah diujikan serta evaluasi apakah sistem yang sudah jadi sudah sesuai dengan yang diharapkan.

2.10. Teori *Blackbox Testing*

Metode *Blackbox Testing* merupakan salah satu metode yang mudah digunakan karena hanya memerlukan batas bawah dan batas atas dari data yang di harapkan. Estimasi banyaknya data uji dapat dihitung melalui banyaknya *field* data entri yang akan diuji, aturan entri yang harus dipenuhi serta kasus batas atas dan batas bawah yang memenuhi. Dan dengan metode ini dapat diketahui jika fungsionalitas masih dapat menerima masukan data yang tidak diharapkan maka menyebabkan data yang disimpan kurang valid.(Made N, Febriyanti. dkk, 2021).

Blackbox testing adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Pengujian *blackbox*, mengevaluasi hanya dari tampilan luarnya (*interfacenya*), fungsionalitas tanpa mengetahui apa sesungguhnya yang terjadi dalam proses detilnya (hanya mengetahui *input* dan *output*)(Tresnawati & Pratama, 2021).

2.11. *Unified Modeling Language* (UML)

UML (*Unified Modelling Language*) merupakan sebuah model perancangan sistem yang mempunyai kelebihan dapat memudahkan *developer* sistem dalam merancang sistem yang akan dibuat karena sifatnya yang berorientasikan pada objek (M Teguh Prihandoyo, 2018). UML adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek. Hal ini disebabkan karena UML menyediakan bahasa pemograman visual yang memungkinkan bagi pengembang sistem untuk membuat cetak baru atas visi mereka dalam bentuk yang baku, mudah

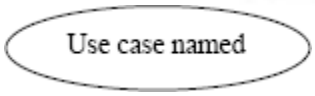
dimengerti, serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan.

Diagram *Unified Modelling Language* (UML) (Suendri, 2018) antara lain sebagai berikut:

1. *Use Case* Diagram, *Use case* menggambarkan *external view* dari sistem yang akan kita buat modelnya (E. Ali, 2019) Model *use case* dapat dijabarkan dalam diagram *use case*, tetapi perlu diingat, diagram tidak identik dengan model karena model lebih luas dari diagram. *Use case* harus mampu mengambarkan urutan *actor* yang menghasilkan nilai terukur (E. Ali, 2019).

Tabel 2. 1 Simbol-Simbol *Use Case*.

(Sumber : E. Ali, 2019)

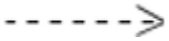
SIMBOL	NAMA	KETERANGAN
	***Actor***	*Actor* adalah pengguna sistem. *Actor* tidak terbatas hanya manusia saja, jika sebuah sistem berkomunikasi dengan aplikasi lain dan membutuhkan input atau memberikan *output*, maka aplikasi tersebutjuga bisa dianggap sebagai *actor*.
	***Use Case***	*Use case* digambarkan sebagai lingkaran *elips* dengan nama *usecase* dituliskan didalam *elips* tersebut.

	Associati on	Asosiasi digunakan untuk menghubungkan *actor* dengan *use case.* Asosiasi digambarkan dengan sebuah garis yang menghubungkan antara *actor* dengan *UseCase.*

2. *Class Diagram,* kelas sebagai suatu set objek yang memiliki atribut dan perilaku yang sama, kelas kadang disebut kelas objek(E. Ali, 2019). *Class* memiliki tiga area pokok yaitu :
    1. Nama, kelas harus mempunyai sebuah nama.
    2. Atribut, adalah kelengkapan yang melekat pada kelas. Nilai dari suatu kelas hanya bisa diproses sebatas atribut yang dimiliki.
    3. Operasi, adalah proses yang dapat dilakukan oleh sebuah kelas, baik pada kelas itu sendiri ataupun kepada kelas lainnya.

Tabel 2. 2 Simbol *Class Diagram*
(Sumber : E. Ali, 2019)

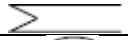
GAMBAR	NAMA	KETERANGAN
	Generalization	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
	***N-Ary Association***	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	***Class***	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.

	***Collaboration***	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu *actor*.
	***Realization***	Operasi yang benar-benar dilakukan oleh suatu objek.
	***Dependency***	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent*) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	Association	Apa yang menghubungkan antara objek satu dengan objek lainnya.

3. *Activity Diagram*, Diagram aktifitas menunjukkan aktifitas sistem dalam bentuk kumpulan aksi-aksi, bagaimana masing-masing aksi tersebut dimulai, keputusan yang mungkin terjadi hingga berakhirnya aksi. *Activity diagram* juga dapat menggambarkan proses lebih dari satu aksi salam waktu bersamaan. "Diagram *activity* adalah aktifitas-aktifitas, objek, *state*, transisi *state* dan*event*. Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktifitas".

Tabel 2. 3 Simbol *Activity Diagram*
(Sumber : E. Ali, 2019)

SIMBOL	KETERANGAN
	Titik Awal
	Titik Akhir
	*Activity*
	Pilihan untuk mengambil keputusan
	*Fork*; digunakan untuk menunjukkan kegiatan yangdilakukan secara parallel atau untuk menggabungkan dua kegiatan paralel menjadi satu.

	*Rake*; menunjukkan adanya dekomposisi
	Tanda waktu
	Tanda pengiriman
	Tanda penerimaan
	Aliran akhir (*Flow Final*)

2.12. Penelitian Terdahulu

Dalam penulisan skripsi ini peneliti menggali informasi dari beberapa penelitian sebelumnya sebagai bahan pertimbangan, baik mengenai kekurangan atau kelebihan yang sudah ada. Selain itu, peneliti juga menggali informasi dari buku – buku maupun skripsi dalam rangka mendapatkan suatu informasi yang ada sebelumnya tentang teori yang berkaitan dengan judul yang digunakan untuk memperoleh landasan teori ilmiah.

Sistem presensi *online* sebelumnya sudah pernah dibuat dan digunakan namun dalam program aplikasi yang berbeda – beda. Beberapa sistem presensi yang berhubungan dengan teknologi yang pernah dibuat diantaranya :

a. Shandy Tresnawati & Alfian Pratama (2021), membuat sebuah sistem presensi dengan metode *geolocation* berbasis web dengan studi kasus: PT. Codepolitan Integrasi Indonesia dengan menggunakan teknologi PHP dan MySQL. Aplikasi yang dibuat akan dijalankan pada *platfrom web mobile* dengan menggunakan bahasa pemrograman HTML5,CSS3,PHP,dan database MySQL.
b. Anggita Arfina Arfah dan Untung Suwardoyono (2022), membuat sistem presensi karyawan menggunakan *geolocation* dan *fingerprint* berbasis *Android* dimana sistem yang akan dibangun ini mampu memonitoring lokasi karyawan yang telah melakukan absen datang,

sehingga perusahaan dapat mengetahui apabila karyawan meninggalkan lokasi perusahaan pada jam kerja tanpa konfirmasi.

c. Nyoman Eddy Indrayana dkk (2020), membuat sistem pembatasan Area Virtual untuk pemantauan aktivitas anak – anak menggunakan *Smartphone* dan *smartwatch* dengan menggunakan bantuan *Geolocation* dan *Haversine Formula*. Aplikasi yang dibuat ini dapat mendeteksi koordinat pengguna *smartwatch* yang dapat dipantau melalui *Smartphone.*
d. Virgian Fajaryantoro dkk(2020), membuat sistem pencarian kios penyedia produk pertanian terdekat dengan menggunkan metode radius, *haversine formula* berbasis web.
e. Hermawan & Rosyid (2020), membuat sistem informasi lokasi wisata kabupaten gresik menggunakan metode *Item based collaborative filtering,* fitur aplikasi ini adalah navigasi lokasi wisata terdapat list tempat-tempat wisata yang terdapat pada Kabupaten Gresik, setelah memilih salah salah satu tempat wisata akan otomatis masuk ke google map dan mengarahkan *user* wisatawan menuju lokasi wisata yang dituju. Fitur ini akan sangat membantu para wisatawan yang tidak mengetahui arah tujuan ketika dia akan mengunjungi sebuah lokasi.
f. Mukhlasin dkk (2019),membuat sistem informasi *Tracer study* alumni pada prodi teknik informatika Universitas Muhammadiyah Gresik berbasis *Web.*
g. Mulyanti (2020), membuat sistem perancangan aplikasi presensi pegawai menggunakan metode *Geofencing* dan perhitungan jarak pada puskesmas inuman guna mengatasi masalah yang ditimbulkan seperti rusaknya buku ataupun hilangnya buku absen yang selayakya akan digunakan sebagai laporan absen bulanan maka dibuat aplikasi absensi mengunakan *Android* dan metode *geofencing.*
h. Syarifudin A. (2019), membuat sistem perancangan sistem informasi pengajuan dan pelaporan pembayaran tunjangan kinerja kementerian keuangan menggunakan metode *Prototype*. Salah satu upaya untuk

dapat meningkatkan *good governance* di lingkungan instansi pemerintah adalah dengan memanfaatkan teknologi informasi.

i. Syafar A., Kambau R., Mulayah N (2021), membuat sistem pencarian guru al qur'an dengan menggunakan metode *haversine formula* dengan memanfaatkan *smartphone* sebagai alat, selain itu perancangan aplikasi ini akan dilengkapi dengan fitur laporan hasil belajar siswa, dimana guru Al Qur'an tidak hanya akan melakukan proses pengajaran tetapi juga akan melakukan proses penilaian setiap pengembangan materi pembelajaran dilakukan oleh siswa yang tentunya dapat memudahkan orang tua untuk melakukan proses pemantauan perkembangan belajar anaknya tanpa harus ikut serta dalam proses belajar anaknya.
j. Gunawan Deny, Alfarizi S. dkk (2022), membuat sistem implementasi metode *Prototype* dalam perancangan sistem informasi upah pasang material konstruksi dengan *study* kasus PT. Cengkareng Permai yang dapat mencatat setiap informasi dari admin baja, direktur dapat langsung mengecek dan memberikan persetujuan serta bagian keuangan melakukan pembayaran.

Penelitian-penelitian yang berkaitan dengan sistem presensi di atas membahas tentang penggunaan *Haversine formula* dalam sistem presensi berbasis *Android* sehingga penulis ingin berkontribusi membuat sebuah sistem presensi yang akan dijalankan diatas *platfrom Android* dengan menggunakan bantuan teknologi SIG (sistem informasi Geografis) dengan judul penelitian penerapan sistem presensi guru menggunakan metode *geofencing* & *haversine formula* berbasis *andro*


UNIVERSITAS MUHAMMADIYAH
GRESIK


UNIVERSITAS MUHAMMADIYAH
GRESIK