

BAB 2

TINJAUAN PUSTAKA

2.1. Masjid

Masjid adalah tempat ibadah bagi umat Muslim dan merupakan salah satu unsur penting dalam agama Islam yang berasal dari kata dalam bahasa Arab yaitu Ismi makan (yang menunjukkan pada sebuah tempat) yang diambil dari *fi'il masdar sulasi mujarrad* yang berarti tempat bersujud atau tempat ibadah shalat, sehingga masjid mengandung pengertian tempat melaksanakan kewajiban bagi umat Islam untuk melaksanakan shalat lima waktu yang diperintahkan Allah SWT. Pengertian ini mengacu pada masjid biasanya digunakan untuk melaksanakan shalat lima waktu, salat Jumat, dan salat Idul Fitri dan Idul Adha. Selain itu, masjid juga digunakan untuk berbagai macam kegiatan sosial dan keagamaan, seperti pengajian, bimbingan belajar, dan kegiatan amal [6]. Gambar 2.1 di bawah merupakan gambar salah satu masjid.



Gambar 2.1 Masjid

Sumber : <https://images.app.goo.gl/Pyyl1aDxNjUMasNRM6>

2.2. Aplikasi

Aplikasi adalah program di komputer atau ponsel yang digunakan untuk menjalankan program yang sudah dibuat. Aplikasi merupakan program yang digunakan oleh orang untuk melakukan sesuatu pada sistem komputer, sistem aplikasi seluler adalah aplikasi yang dapat digunakan bahkan ketika pengguna dengan mudah berpindah dari satu tempat ke tempat lain tanpa gangguan atau terputusnya komunikasi [7].

Keunikan dari sistem aplikasi seluler terletak pada kemampuannya untuk memberikan akses dan fungsionalitas yang lengkap kepada pengguna bahkan ketika mereka dengan mudah berpindah dari satu tempat ke tempat lain tanpa mengalami gangguan atau terputusnya komunikasi. Dalam perkembangan teknologi yang terus berlanjut, aplikasi seluler terus menjadi bagian penting dalam kehidupan sehari-hari masyarakat modern. Baik untuk tujuan produktivitas, hiburan, kesehatan, keuangan, atau kebutuhan khusus lainnya, aplikasi seluler memainkan peran yang signifikan dalam memenuhi kebutuhan dan keinginan pengguna dengan cara yang efisien dan inovatif.

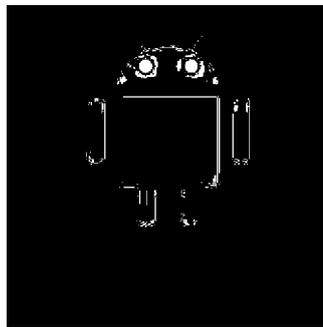
Pembangunan Aplikasi adalah suatu kegiatan yang memiliki tujuan untuk mendesain dan menciptakan sistem atau aplikasi baru yang diharapkan dapat menyelesaikan masalah – masalah yang dihadapi dan diperoleh dari pemilihan alternatif sistem yang baik. Aplikasi yang dibangun dapat berupa aplikasi desktop, aplikasi seluler (*mobile*), atau aplikasi web. [8]. Proses pembangunan aplikasi dimulai dengan perencanaan dan analisis kebutuhan, dilanjutkan dengan desain dan pengembangan aplikasi, dan kemudian dilanjutkan dengan pengujian, peluncuran, dan pemeliharaan aplikasi.

Pembangunan aplikasi adalah proses yang rumit dan kompleks. Namun, dengan perencanaan yang baik, desain yang baik, pengembangan yang hati-hati, dan pengujian yang cermat, pengembang dapat menciptakan aplikasi yang berfungsi dengan baik dan memenuhi kebutuhan pengguna dan bisnis.

2.3. Android

Android adalah sistem operasi mobile yang berbasis pada kernel Linux dan dirancang untuk digunakan pada perangkat mobile seperti smartphone, tablet, dan komputer jinjing. Android dikembangkan oleh Google dan dirilis pertama kali pada tahun 2008. Sistem operasi Android ini ditawarkan secara gratis dan bersifat open-source, yang memungkinkan pengguna dan pengembang untuk memodifikasi dan mengembangkan sistem operasi Android sesuai dengan kebutuhan mereka. Android menyediakan lingkungan pengembangan yang lengkap, termasuk SDK (*Software Development Kit*) yang dapat digunakan oleh

para pengembang untuk membuat aplikasi Android. Aplikasi Android dapat dibuat dengan menggunakan bahasa pemrograman Java, Kotlin, atau C++. Sistem operasi Android juga mendukung berbagai macam fitur seperti layar sentuh, koneksi internet, kamera, GPS, dan masih banyak lagi. Android telah menjadi sistem operasi mobile yang sangat populer dan saat ini telah menjadi sistem operasi mobile terbesar di dunia [11]. Gambar 2.2 di bawah merupakan gambar logo Android.



Gambar 2.2 Logo Sistem Operasi Android

Sumber : <https://images.app.goo.gl/QS8wnTfsbwzU2w5Q9>

2.3.1. Arsitektur Android

Arsitektur Android adalah model arsitektur perangkat lunak yang digunakan pada sistem operasi Android. Arsitektur ini terdiri dari beberapa komponen, antara lain [12]:

1. *Application and Widgets*

Application and widgets merupakan layer di mana kita dapat terhubung dengan aplikasi saja, dimana biasanya *user* men-download aplikasi kemudian melakukan instalasi dan menjalankan aplikasi tersebut.

2. *Application Frameworks*

Merupakan “*Open Development Platform*” dimana Android menawarkan atau memberi kemampuan kepada pengembang untuk

membangun aplikasi yang bagus dan inovatif. Disini pengembang bebas mengakses perangkat keras, akses informasi resource, menjalankan service background, menambah status notification dan sebagainya.

3. *Libraries*

Merupakan layer dimana Fitur-fitur dari android berada, biasanya para developer aplikasi mengakses libraries untuk menjalankan aplikasinya.

4. *Android Run Time*

Merupakan layer yang membuat aplikasi pada android dapat dijalankan, dalam pemrosesanya menggunakan Implementasi Linux yang terbagi dalam *Core Libraries* dan *Dalvik Virtual Machine*.

5. *Linux Kernel*

Merupakan layer dimana inti dari sistem operasi Android itu berada.

2.3.2. Silent Mode

Fitur Silent Mode pada smartphone adalah fitur yang memungkinkan pengguna untuk menonaktifkan suara panggilan, pesan, notifikasi, dan alarm. Dalam mode ini, ponsel hanya akan bergetar atau tidak memberikan suara sama sekali ketika ada panggilan atau pesan masuk. Fitur ini sering digunakan ketika pengguna ingin menghindari gangguan suara yang tidak diinginkan di lingkungan yang tenang, seperti saat berada di tempat ibadah, rapat, atau tengah tidur.

Dengan menggunakan Android *AudioManager API* yang disediakan oleh sistem operasi Android memungkinkan untuk mengatur pengaturan audio pada perangkat Android, termasuk pengaturan volume, mode nada dering, dan mode getar. Pertama, perlu diinisialisasi *AudioManager* dengan memanggil metode *getSystemService(Context.AUDIO_SERVICE)*. Setelah itu, panggil metode *adjustStreamVolume()* dengan parameter *ADJUST_MUTE* untuk mengatur mode silent pada perangkat. Metode *adjustStreamVolume()* akan mengubah mode nada dering pada perangkat

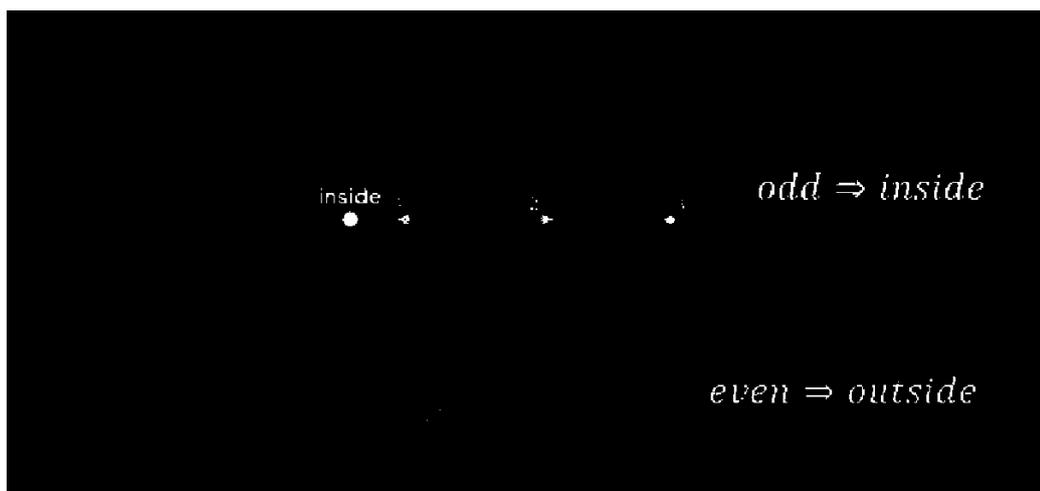
menjadi mode silent. Untuk mengatur kembali perangkat ke mode nada dering, maka dapat memanggil metode `adjustStreamVolume()` dengan parameter `ADJUST_UNMUTE`. Terakhir, aplikasi perlu memiliki izin untuk mengakses `AudioManager API` dengan menambahkan izin pada file `AndroidManifest.xml`. Gambar 2.3 merupakan code untuk menambahkan izin di `AndroidManifest.xml`.

```
<user-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
```

Gambar 2.3 Kode Untuk Menambahkan Izin Audiomanager

2.4. Algoritma Ray Casting

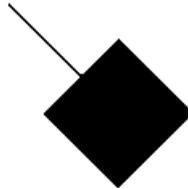
Ray casting adalah sebuah teknik rendering yang digunakan dalam grafik komputer dan geometri komputasi. Tujuan dari *ray casting* adalah untuk menciptakan perspektif tiga dimensi dalam ruang dua dimensi. Metode ini menggunakan algoritma geometrik yang serupa dengan ray tracing. Analoginya, algoritma ini bekerja dengan mengirimkan sinar cahaya yang dimulai dari tepi sebuah poligon dan melacak jalannya hingga titik akhir [19]. Gambar 2.4 dibawah merupakan Gambaran dari perpotongan garis *Ray Casting* dengan poligon.



Gambar 2.4 Visualisasi Algoritma Ray Casting

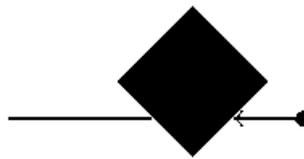
Sumber : <https://www.youtube.com/watch?v=RSXM9bgqxJM>

Algoritma ini didasarkan pada pertimbangan bahwa, jika suatu titik berada di dalam poligon, maka setiap sinar yang keluar darinya berpotongan dengan poligon dengan bilangan ganjil seperti 1,3,5,7,9,11 dan seterusnya, contoh perpotongan berada dalam poligon dapat dilihat pada gambar 2.5 berikut.



Gambar 2. 5 Titik Di Dalam Poligon

Jika sebaliknya, titik berada di luar poligon, maka setiap sinar berpotongan dengan poligon dengan bilangan genap seperti 0,2,4,6,8,10 dan seterusnya, contoh perpotongan berada di luar poligon dapat dilihat pada gambar 2.6 berikut.



Gambar 2. 6 Titik Di Luar Poligon

Untuk menentukannya mula – mula kita ambil 3 parameter: “*latitude*”, “*longitude*” yang merupakan koordinat titik yang akan diperiksa, dan “*polygon*” yang berisikan koordinat titik awal dan titik akhir dari sisi poligon ditentukan menggunakan variabel x_i , y_i , x_j , dan y_j .

Untuk pengecekan apakah titik yang akan diperiksa berpotongan dengan sisi poligon adalah sebagai berikut:

1. Pertama periksa apakah sinar dari titik melintasi garis di antara dua titik poligon, y_i dan y_j (*longitude* titik awal dan akhir garis poligon) pada sumbu y. *Longitude* dari titik yang diperiksa harus lebih kecil dari y_i dan lebih besar dari y_j (*longitude* $< y_i$ dan *longitude* $> y_j$) atau sebaliknya (*longitude* $> y_i$ dan *longitude* $< y_j$).

2. Kedua periksa apakah titik “**latitude**” lebih kecil dari garis yang menghubungkan **xi** dan **xj**. Dengan rumus $(\text{latitude} < (\text{xi} - \text{xj}) \times (\text{longitude} - \text{yj}) / (\text{yi} - \text{yj}) + \text{xj})$, Jika persamaan ini benar, berarti garis yang melalui titik yang diperiksa memotong garis poligon antara **xi** dan **xj** (latitude titik awal dan akhir garis poligon).

Jika kedua kondisi tersebut terpenuhi, berarti terjadi perpotongan antara garis poligon dengan garis yang melalui titik yang diperiksa. Lakukan perulangan pada seluruh bagian poligon, jika jumlah perpotongan adalah “**ganjil**”, maka titik berada di dalam poligon. Jika jumlah Perpotongan adalah “**genap**” maka titik berada di luar poligon.

2.5. GPS (Global Positioning System)

Global Positioning System (GPS) merupakan sistem radio navigasi yang menggunakan satelit untuk menentukan lokasi dengan koordinat tiga dimensi (x, y, z), serta informasi waktu dan kecepatan. Unit GPS menghitung posisinya berdasarkan titik koordinat latitude dan longitude pengguna yang diukur. GPS digunakan untuk menemukan koordinat latitude dan longitude suatu target, terutama dengan mengaktifkan GPS pada smartphone saat dalam perjalanan [12].

Untuk mengakses lokasi GPS pada perangkat Android, Perlu meminta izin lokasi pada perangkat Android. GPS digunakan sebagai titik point untuk menentukan posisi dari smartphone apakah berada pada daerah masjid atau di luar daerah masjid yang akan terhubung ke OpenStreetMap. Code untuk meminta izin dan Request lokasi pengguna dapat di lihat pada Gambar 2.7 dan 2.8 di berikut.

```
<user-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

Gambar 2. 7 permission untuk mengakses lokasi pengguna

```

LocationCallback locationCallback = new LocationCallback() {
    @Override
    public void onLocationResult(LocationResult locationResult) {
        if (locationResult == null) {
            return;
        }
        Location location = locationResult.getLocations().get(0);
        double latitude = location.getLatitude();
        double longitude = location.getLongitude();
        OSMRRequest osmRequest = new OSMRRequest(getApplicationContext(), InitialRingMode);
        osmRequest.execute(latitude, longitude);
        Log.d("TAG", msg "OSMRRequest executed. Latitude: " + latitude + ", Longitude: " + longitude);
    }
};

```

Gambar 2. 8 Code Untuk Mendapatkan Lokasi Pengguna

Berikut ini adalah contoh Respon yang didapat dari code request lokasi pengguna, Respon yang didapatkan yaitu koordinat Latitude dan Longitude dari pengguna dapat di lihat pada gambar 2.9 berikut.

```
Latitude: -6.9120249, Longitude: 107.7058101
```

Gambar 2. 9 Hasil Yang Di Dapat Dari Request lokasi

2.6. API

Application Programming Interface (API) adalah seperangkat prosedur, fungsi, protokol, dan aturan. Ini menentukan bagaimana komponen atau paket perangkat lunak yang berbeda harus berinteraksi satu sama lain. API berfungsi sebagai antarmuka virtual antara dua fungsi perangkat lunak yang bekerja sama menggunakan bahasa yang sangat berbeda [9].

2.6.1. API OpenStreetMap

OpenStreetMap (OSM) adalah sebuah proyek pemetaan digital global yang dibangun oleh komunitas pengguna yang terdiri dari sukarelawan dari seluruh dunia. Proyek ini menggunakan teknologi pemetaan partisipatif dengan konsep *Volunteered Geographic Information* (VGI) yang memungkinkan siapa saja untuk ikut serta dalam mengumpulkan dan membagikan data geografis secara online. Data geografis yang dikumpulkan meliputi informasi tentang jalan, bangunan, rel kereta api, dan lain-lain[13]. OSM menyediakan data masjid terbuka dan bebas lisensi yang dapat

digunakan oleh siapa saja untuk keperluan apapun dengan batasan request 10.000 per hari untuk satu user. Data Masjid OSM dapat diakses dan dimanfaatkan oleh pengguna melalui aplikasi dan layanan web yang memanfaatkan API (*Application Programming Interface*) yang disediakan oleh proyek OSM [14].

Untuk mengambil data geografis dari *OpenStreetMap* (OSM) digunakan layanan dari OSM yaitu *Overpass*. Permintaan *Overpass* dibentuk menggunakan sintaks *Overpass QL*, yaitu bahasa *query* yang digunakan dalam *Overpass*. Permintaan ini menggambarkan kriteria pencarian data masjid yang diinginkan. Pencarian data masjid dalam *OpenStreetMap* dilakukan berdasarkan kunci (*Key*) dan nilai (*Value*) yang tertera pada setiap tempat ibadah yang terdaftar di dalamnya. Setiap tempat ibadah, termasuk masjid, memiliki atribut-atribut yang membedakannya dari jenis tempat lainnya, seperti tipe fasilitas (*amenity*) dan agama yang dianut (*religion*). Berikut atribut yang dimiliki dari setiap tempat ibadah yang terdaftar di *OpenStreetMap* dapat dilihat di gambar 2.10.

Tempat Ibadah

Objek yang digunakan untuk melakukan peribadatan ataupun kegiatan yang berkaitan dengan agama dan keyakinan masyarakat. Berikut objek yang termasuk tempat beribadah di Indonesia:

No.	Nama Objek	Jenis Objek	Deskripsi	Key	Value	Simbol di OSM	Foto Contoh
1.	Masjid / Mushala		Tempat untuk melakukan kegiatan peribadatan untuk umat muslim	<ul style="list-style-type: none"> amenity religion 	<ul style="list-style-type: none"> place_of_worship muslim 		
2.	Gereja / Chapel		Tempat untuk melakukan kegiatan peribadatan untuk umat kristiani	<ul style="list-style-type: none"> amenity religion 	<ul style="list-style-type: none"> place_of_worship christian 		
3.	Pura		Tempat untuk melakukan kegiatan peribadatan untuk umat hindu	<ul style="list-style-type: none"> amenity religion 	<ul style="list-style-type: none"> place_of_worship hindu 		
4.	Vihara		Tempat untuk melakukan kegiatan peribadatan untuk umat budha	<ul style="list-style-type: none"> amenity religion 	<ul style="list-style-type: none"> place_of_worship buddhist 		
5.	Klenteng		Tempat untuk melakukan kegiatan peribadatan untuk umat khonghuou	<ul style="list-style-type: none"> amenity religion 	<ul style="list-style-type: none"> place_of_worship taoist 		

Gambar 2. 10 Atribut-Atribut Tempat Ibadah Di OSM

Sumber: https://wiki.openstreetmap.org/wiki/Id:Indonesian_Tagging_Guidelines

Untuk sintaks request dikirim ke *Overpass API* melalui *URL* dengan menggunakan metode *HTTP GET* dari sebuah koordinat. Untuk cara request *overpass* dapat dilihat pada Gambar 2.11 berikut.:

```
String overpassUrl = "https://overpass-api.de/api/interpreter?data=[out:json]" +  
    ";way[\"amenity\"=\"/>"place_of_worship\"][\"religion\"=\"/>"muslim\"]" +  
    "(around:10," + lat + "," + lon + ");out geom;";
```

Gambar 2. 11 Potongan Kodingan Untuk Mendapatkan Informasi Posisi Pengguna

Permintaan ini mencari dan mengambil data masjid terdekat dengan atribut *Key* "amenity" yang bernilai "place_of_worship" dan atribut "religion" yang bernilai "muslim" dalam jarak 10 meter dari koordinat yang ditentukan (lat, lon). Permintaan menggunakan format JSON dengan mengatur parameter "out:json" dan mengembalikan geometri objek dengan menggunakan "out geom" (*OpenStreetMap Wiki*, 2021). Untuk hasil respon dari request dapat dilihat pada gambar 2.12 berikut.:

```

{
  "version": 0.6,
  "generator": "Overpass API 0.7.60 f4c14d41",
  "osm3s": {
    "timestamp_osm_base": "2023-05-31T08:52:24Z",
    "copyright": "The data included in this document is from www.openstreetmap.org. The data is made available under ODbL."
  },
  "elements": [
    {
      "type": "way",
      "id": 546598302,
      "bounds": {
        "minlat": -6.8864585,
        "minlon": 107.6149306,
        "maxlat": -6.8862941,
        "maxlon": 107.6151378
      },
      "nodes": [
        5282112062,
        5282112063,
        5297143921,
        5282112064,
        5282112065,
        5282112066,
        5282112067,
        5282112062
      ],
      "geometry": [
        { "lat": -6.8863163, "lon": 107.6149306 },
        { "lat": -6.8864585, "lon": 107.6149715 },
        { "lat": -6.8864230, "lon": 107.6150547 },
        { "lat": -6.8863074, "lon": 107.6151378 },
        { "lat": -6.8862956, "lon": 107.6150970 },
        { "lat": -6.8863316, "lon": 107.6150077 },
        { "lat": -6.8862941, "lon": 107.6149915 },
        { "lat": -6.8863163, "lon": 107.6149306 }
      ],
      "tags": {
        "amenity": "place_of_worship",
        "building": "yes",
        "name": "Masjid Baiturrahman",
        "religion": "muslim"
      }
    }
  ]
}

```

Gambar 2. 12 Respon Dari Overpass

Respon dari *Overpass* akan berupa data *geografis* dalam format JSON yang berisi data masjid terdekat yang memenuhi kriteria yang telah ditentukan. Data tersebut akan mencakup informasi mengenai masjid seperti atribut dan geometri yang mencakup koordinat (longitude dan latitude) yang menggambarkan poligon masjid tersebut, ambil data “*geometry*” Kemudian dari koordinat tersebut diperiksa apakah titik point berada didalam atau diluar poligon

Untuk pengecekan apakah Koordinat berada di dalam atau di luar poligon masjid caranya dengan menggunakan Algoritma *ray casting*. Ray casting berfungsi dengan melintasi setiap tepi poligon dan menghitung

berapa banyak kali garis yang ditarik dari titik ke arah positif sumbu-x memotong tepi. Jika jumlah ganjil maka titik berada di dalam poligon, jika genap maka titik berada di luar poligon. Code untuk menghitung berapa banyak perpotongan dapat dilihat pada gambar 2.13 berikut.

```
// Lakukan pengecekan apakah titik berada di antara dua titik poligon
boolean intersect = ((yi > lon) != (yj > lon)) &&
    (lat < (xj - xi) * (lon - yi) / (yj - yi) + xi);
```

Gambar 2. 13 Potongan Kodingan Untuk Cek Posisi Terhadap Poligon Masjid

2.6.2. API AudioManager

API AudioManager di Android adalah antarmuka untuk mengakses dan mengontrol volume dan mode nada pada perangkat Android. Salah satu fungsinya adalah untuk mengatur mode senyap pada perangkat, yaitu dengan mengubah pengaturan suara dari mode normal ke mode Silent[10].

Jika hasil yang di dapat dari *Raycasting* bernilai *true* maka *API AudioManager* akan mengubah pengaturan audio di smartphone pengguna menjadi mode senyap menggunakan metode *setStreamVolume()*. Untuk mengubah audio menjadi silent dapat dilihat pada gambar 2.14 berikut.

```
AudioManager.FLAG_REMOVE_SOUND_AND_VIBRATE
```

Gambar 2. 14 Potongan Kodingan Mengatur Audio Mode Silent

Kemudian ketika hasil yang di dapat dari Ray Casting bernilai *false*, aplikasi akan mengembalikan mode dering ke mode sebelumnya menggunakan metode *setRingerMode()*. Untuk mengubah audio menjadi unsilent dapat dilihat pada gambar 2.15 berikut.

```
setRingerMode(AudioManager.RINGER_MODE_NORMAL)
```

Gambar 2. 15 Potongan Kodingan Mengatur Audio Kembali Mode Normal

2.7. Android Studio

Menurut buku "Pemrograman Aplikasi Android dengan Android Studio, Photoshop, dan Audition" karya Herlinah S.Kom M.Si dan Susliadi KH S.Kom, dijelaskan bahwa Android Studio adalah *Integrated Development Environment* (IDE) yang digunakan oleh pengembang aplikasi Android untuk membuat, mengedit, dan menguji aplikasi Android. IDE ini menyediakan berbagai macam alat pengembangan, seperti emulator Android, debugger, dan code editor yang memudahkan pengembang dalam mengembangkan aplikasi Android. Android Studio juga menyediakan fitur untuk menganalisis kinerja aplikasi, seperti penggunaan memori dan CPU, serta membantu dalam mengoptimalkan kode. Android Studio didukung oleh Google dan telah menjadi IDE yang populer di kalangan pengembang aplikasi Android. [15]. Gambar 2.16 di bawah merupakan gambar logo Android Studio.



Gambar 2. 16 Logo Android Studio

Sumber : <https://images.app.goo.gl/LzDF8LjHz8xkcLkZ6>

2.8. Java

Java adalah bahasa pemrograman yang *multi platform* dan *multi device*. Java dapat berjalan di hampir semua Perangkat komputer atau perangkat yang lainnya yang sudah support Java. Java dapat berjalan dengan platform sistem operasi yang berbeda – beda, karena sifatnya yang umum dan non-spesifik. Kelebihan Java yang pertama tentu saja multiplatform. Java dapat berjalan di

beberapa platform komputer dan sistem operasi yang berbeda - beda. Kedua adalah Java merupakan bahasa yang bersifat OOP atau *Object Oriented*

Programming. Java digunakan untuk desktop komputasi, komputasi seluler, pengembangan game, pengembangan Web, dan komputasi numerik[16]. Gambar 2.17 di bawah merupakan gambar logo Java.



Gambar 2. 17 Logo Java

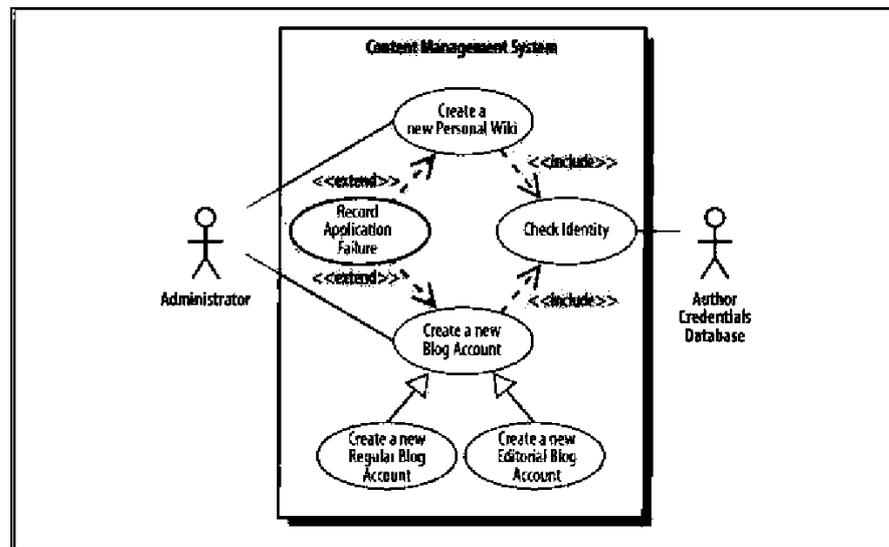
Sumber : <https://images.app.goo.gl/4XPz6f1PgxeDCDfZ7>

2.9. Unified Modeling Language

UML (*Unified Modeling Language*) adalah bahasa berbasis grafik/gambar untuk memvisualisasikan, menentukan, menulis, dan mendokumentasikan sistem pengembangan perangkat lunak berorientasi objek (OO). UML sendiri juga menyediakan standar penulisan sistem cetak biru yang mencakup konsep proses bisnis, penulisan kelas dalam bahasa pemrograman tertentu, skema basis data, dan komponen yang diperlukan dalam sistem perangkat lunak [17].

1. Use Case Diagram

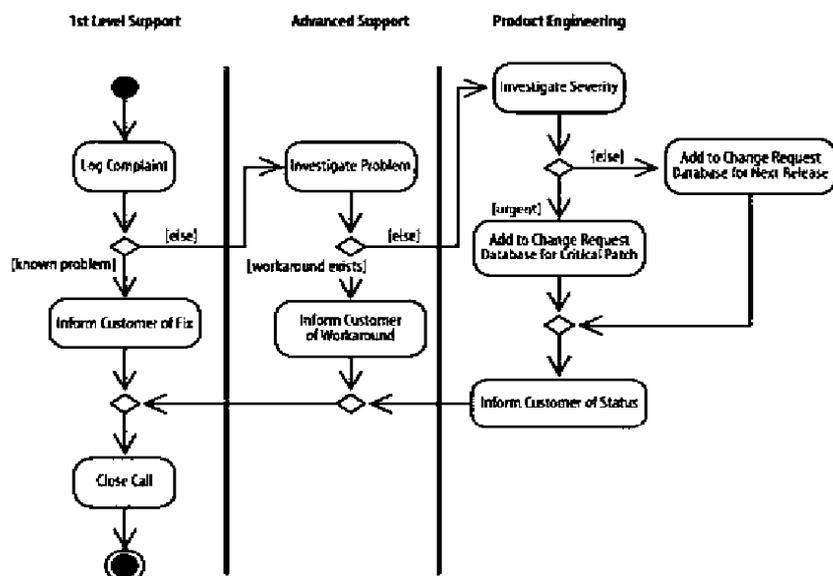
Use case diagram ialah diagram yang menggambarkan kelakuan sistem yang dibuat. Use case diagram menggambarkan suatu interaksi antara satu ataupun lebih aktor dengan sistem yang dibuat. Sederhananya, use case diagram digunakan untuk mengontrol fitur apa saja yang termasuk dalam sistem dan siapa yang dapat menggunakan fitur tersebut. Gambar 2.18 di bawah merupakan gambar contoh sebuah *Use Case Diagram* [20].



Gambar 2. 18 Use Case Diagram

2. Activity Diagram

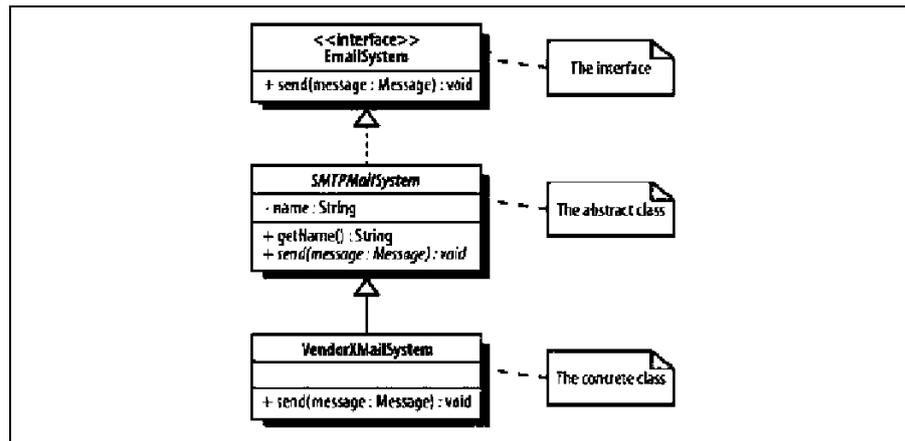
Diagram aktivitas adalah diagram yang menggambarkan Alur kerja model berdasarkan urutan aktivitas di sebuah proses yang merferensikan diagram use case yang ada. Gambar 2.19 di bawah merupakan gambar contoh sebuah *Activity Diagram* [20].



Gambar 2. 19 Activity Diagram

3. Class Diagram

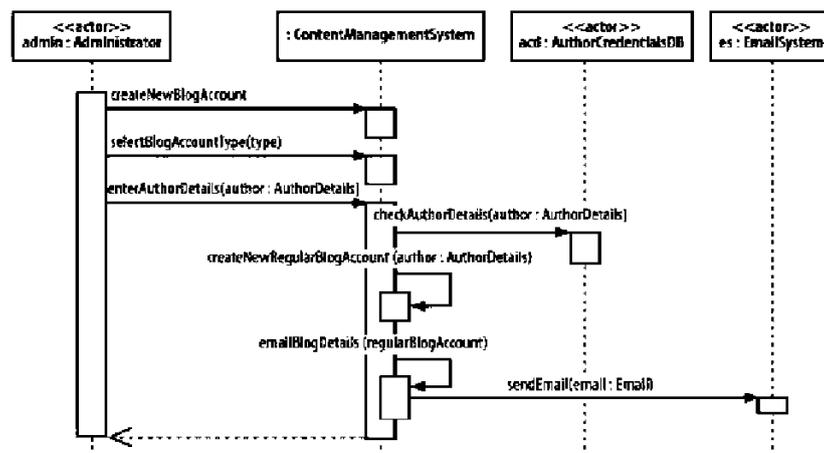
Diagram kelas adalah diagram yang menggambarkan relasi atau hubungan antara kelas-kelas yang terkandung di dalamnya. Ada atribut dan fungsi dari suatu objek. Gambar 2.20 di bawah merupakan gambar contoh sebuah *Class Diagram* [20].



Gambar 2. 20 Class Diagram

4. Sequence Diagram

Sequence diagram merupakan diagram yang menggambarkan pengaplikasian dari masing-masing skenario use case yang ada pada sistem. Gambar 2.21 di bawah merupakan gambar contoh sebuah *Sequence Diagram* [20].



Gambar 2. 21 Sequence Diagram