

BAB 2

TINJAUAN PUSTAKA

2.1 TensorFlow

Google melakukan open source pada framework machine learning-nya, TensorFlow, pada akhir tahun 2015 di bawah lisensi Apache 2.0. Sebelumnya, ini digunakan secara eksklusif oleh Google dalam pengenalan ucapan, Penelusuran, Foto, Gmail, dan aplikasi lainnya. Pustaka ini diimplementasikan dalam C++ dan memiliki API Python yang mudah digunakan, serta API C++ yang kurang dihargai. Karena dependensinya yang lebih sederhana, TensorFlow dapat dengan cepat di-deploy ke berbagai arsitektur [3].

Mirip dengan Theano (perpustakaan komputasi numerik populer untuk Python yang mungkin sudah Anda kenal), komputasi digambarkan sebagai diagram alur, yang memisahkan desain dari implementasi. Dengan sedikit atau tanpa kerumitan, dikotomi ini memungkinkan desain yang sama diimplementasikan tidak hanya pada sistem pelatihan berskala besar dengan ribuan prosesor, namun juga pada perangkat seluler. Sistem tunggal ini mencakup berbagai platform. Salah satu properti TensorFlow yang paling menarik adalah kemampuan diferensiasi otomatisnya. Anda dapat bereksperimen dengan jaringan baru tanpa harus mendefinisikan ulang banyak kunci perhitungan [3].

2.1.1 TensorFlow Lite

TensorFlow Lite adalah versi ringan dari TensorFlow yang dirancang untuk perangkat mobile dan perangkat dengan sumber daya terbatas. TensorFlow Lite memfasilitasi pengguna model pembelajaran mesin di perangkat edge seperti ponsel, mikrokontroler, dan perangkat Internet of Things (IoT) [4]. Beberapa karakteristik utama TensorFlow Lite adalah:

- 1) Efisiensi: TensorFlow Lite dirancang untuk efisiensi dalam penggunaan memori dan daya komputasi, sehingga cocok untuk perangkat dengan keterbatasan sumber daya.
- 2) Kecepatan: TensorFlow Lite memiliki optimasi tertentu yang memungkinkan inferensi (proses menjalankan model untuk membuat prediksi) berjalan lebih cepat.

- 3) **Fleksibilitas:** Mendukung berbagai arsitektur perangkat keras, termasuk CPU, GPU, dan akselerator khusus seperti TPU (Tensor Processing Unit).
- 4) **Kompatibilitas:** Mudah diintegrasikan dengan berbagai platform dan mendukung beragam bahasa pemrograman seperti Python, Java, dan C++.

Untuk mengimpelentasikan TensorFlow Lite, kita perlu mengikuti beberapa langkah dasar, mulai dari pelatihan model, konversi model ke format TensorFlow Lite, dan akhirnya menjalankan model di perangkat mobile atau embedded. Berikut adalah langkah-langkah dasarnya:

1) Pelatihan Model di TensorFlow

Langkah pertama adalah melatih model pembelajaran mesin menggunakan TensorFlow. Ini biasanya dilakukan di lingkungan komputasi yang kuat seperti server atau desktop. Proses pelatihan melibatkan penyusunan arsitektur model (seperti jaringan neural), mempersiapkan data pelatihan, dan melatih model menggunakan data tersebut. Setelah model dilatih hingga mencapai performa yang diinginkan, model tersebut disimpan dalam format yang dapat digunakan kembali (seperti format HDF5 untuk model Keras).

2) Konversi Model ke Format TensorFlow Lite

Setelah model dilatih dan disimpan, langkah berikutnya adalah mengonversi model tersebut ke format TensorFlow Lite. Konversi ini dilakukan menggunakan TensorFlow Lite Converter, alat yang disediakan oleh TensorFlow. Konverter ini mengubah model dari format TensorFlow yang berat ke format TensorFlow Lite yang lebih ringan dan efisien. Hasil konversi adalah file dengan ekstensi '.tflite', yang siap digunakan di perangkat dengan sumber daya terbatas.

3) Menjalankan Model di Perangkat Mobile atau Embedded

Langkah terakhir adalah menjalankan model TensorFlow Lite di perangkat target, seperti ponsel atau perangkat embedded. Untuk ini, TensorFlow Lite menyediakan interpreter yang bisa digunakan dalam berbagai bahasa pemrograman (seperti Java, Swift, dan C++).

Interpreter ini memuat model '.tflite' dan menjalankan inferensi menggunakan input data yang diberikan. Proses ini memungkinkan aplikasi untuk membuat prediksi atau mengambil keputusan berdasarkan model yang dilatih.

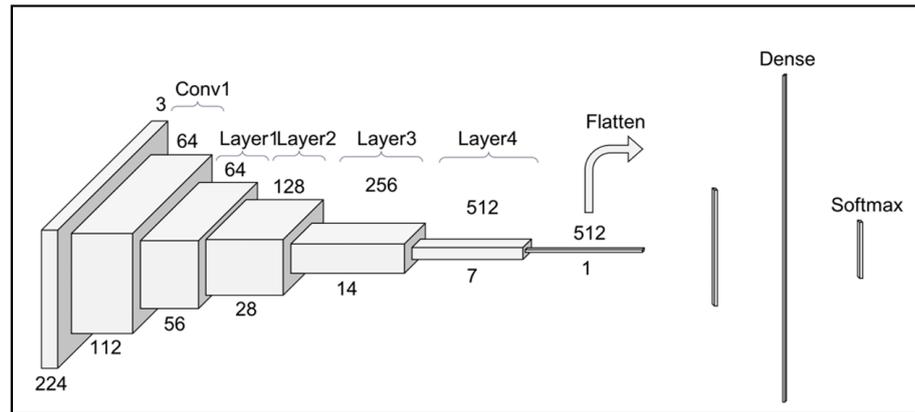
TensorFlow Lite adalah alat yang kuat untuk membawa model pembelajaran mesin ke perangkat dengan keterbatasan sumber daya. Proses implementasinya melibatkan pelatihan model di TensorFlow, konversi ke format TensorFlow Lite, dan menjalankan model di perangkat target menggunakan interpreter TensorFlow Lite. Dengan fitur optimasi dan kuantisasi, TensorFlow Lite memungkinkan pengembangan aplikasi yang efisien dan cepat di berbagai platform [4].

2.2 Image Classification

Image Classification merupakan proses yang dapat mengklasifikasikan gambar sesuai dengan kategori tertentu. Dalam image classification mungkin dirancang untuk mengetahui apakah suatu gambar mengandung sosok manusia atau tidak. Saat mendeteksi suatu objek sepele bagi manusia, klasifikasi gambar yang kuat masih merupakan tantangan dalam visi komputer aplikasi [5].

2.3 ResNet50

Residual Network (ResNet) memenangkan ILSVRC pada tahun 2015 dengan menggunakan skip connection dan features batch normalization. Jaringan ini memungkinkan pelatihan Neural Network dengan 152 layer dan mampu mengurangi kompleksitas dibandingkan dengan VGGNet. Tingkat kesalahan yang dicapai dari arsitektur ini adalah sebesar 3,57% pada Top-5, sehingga mampu mengalahkan kinerja human-level pada dataset yang diberikan. Contoh jaringan ResNet dapat dilihat pada Gambar 2.1.



Gambar 2.1 Contoh Jaringan ResNet

ResNet50 adalah salah satu arsitektur deep learning yang digunakan untuk image classification. Berikut adalah langkah-langkah umum untuk menggunakan ResNet50:

1. Mengimpor Pustaka yang Dibutuhkan

Pertama, impor pustaka pustaka yang diperlukan, seperti TensorFlow atau PyTorch. Dalam contoh ini, kita akan menggunakan TensorFlow dan Keras, seperti pada Gambar 2.2.

```

1 import tensorflow as tf
2 from tensorflow.keras.applications import ResNet50
3 from tensorflow.keras.preprocessing.image import ImageDataGenerator
4 from tensorflow.keras.models import Sequential
5 from tensorflow.keras.layers import Dense, Flatten
6 from tensorflow.keras.optimizers import Adam

```

Gambar 2.2 Mengimpor Pustaka

2. Menggunakan Model Pre-trained

Unduh model ResNet50 yang sudah dilatih sebelumnya pada dataset ImageNet. Model ini dapat digunakan sebagai fitur extractor. Seperti pada Gambar 2.3.

```

8
9 base_model = ResNet50(weights='imagenet', include_top=False, input_shape=(224, 224, 3))
10

```

Gambar 2.3 Menggunakan Model Pre-trained

3. Membangun Model

Tambahkan layer tambahan di atas base model sesuai dengan kebutuhan klasifikasi. Seperti pada Gambar 2.4.

```

12 model = Sequential([
13     base_model,
14     Flatten(),
15     Dense(256, activation='relu'),
16     Dense(num_classes, activation='softmax') # Sesuaikan dengan jumlah kelas yang ingin diklasifikasi
17 ])

```

Gambar 2.4 Membangun Model

4. Mengompilasi Model

Tentukan loss function, optimizer, dan metrics yang akan digunakan. Seperti pada Gambar 2.5.

```

21 model.compile(optimizer=Adam(learning_rate=0.0001),
22               loss='categorical_crossentropy',
23               metrics=['accuracy'])

```

Gambar 2.5 Mengompilasi Model

5. Mempersiapkan Data

Gunakan 'ImageDataGenerator' untuk mempersiapkan data latihan dan validasi. Seperti pada Gambar 2.6.

```

27 train_datagen = ImageDataGenerator(rescale=1./255,
28                                   shear_range=0.2,
29                                   zoom_range=0.2,
30                                   horizontal_flip=True)
31
32 test_datagen = ImageDataGenerator(rescale=1./255)
33
34 train_generator = train_datagen.flow_from_directory('path_to_train_data',
35                                                    target_size=(224, 224),
36                                                    batch_size=32,
37                                                    class_mode='categorical')
38
39 validation_generator = test_datagen.flow_from_directory('path_to_validation_data',
40                                                        target_size=(224, 224),
41                                                        batch_size=32,
42                                                        class_mode='categorical')
43
44

```

Gambar 2.6 Mempersiapkan Data

6. Melatih Model

Latih model dengan data yang sudah dipersiapkan. Seperti pada Gambar 2.7.

```

47 history = model.fit(train_generator,
48                       epochs=10,
49                       validation_data=validation_generator)
--

```

Gambar 2.7 Melatih Model

7. Evaluasi Model

Setelah model dilatih, evaluasi performa model dengan data validasi atau data test. Seperti pada Gambar 2.8.

```

54 score = model.evaluate(validation_generator)
55 print(f"Accuracy: {score[1] * 100}%")

```

Gambar 2.8 Evaluasi Model

8. Menyimpan Model

Simpan model yang sudah dilatih untuk digunakan di masa depan. Seperti pada Gambar 2.9.

```

60 model.save('resnet50_model.h5')
61

```

Gambar 2.9 Menyimpan Model

Itulah langkah-langkah dasar untuk menggunakan ResNet50 dalam klasifikasi gambar. Anda dapat menyesuaikan parameter dan struktur model sesuai dengan kebutuhan spesifik aplikasi anda.

2.4 Buah

Buah adalah hasil reproduksi antara putik dan serbuk sari pada tumbuhan. Buah termasuk organ pada tumbuhan berbunga yang merupakan perkembangan lanjutan dari bakal buah (ovarium). Buah biasanya membungkus dan melindungi biji . aneka rupa dan bentuk buah tidak terlepas kaitannya dengan fungsi utama buah, yakni sebagai pemencar biji tumbuhan [6]. Indonesia merupakan suatu negara yang beruntung karena terletak di daerah katulistiwa yang mempunyai tipe hutan hujan tropik cukup unik dengan tidak kurang dari 329 jenis buah-buahan (terdiri dari 61 suku dan 148 marga) baik yang merupakan jenis asli Indonesia maupun pendatang [7].

2.5 Healthy Drink

Healthy drink didefinisikan sebagai minuman yang mengandung nutrisi yang bermanfaat bagi tubuh dan tidak mengandung bahan berbahaya. Nutrisi ini bisa berupa vitamin, mineral, protein, serat, antioksidan, dan fitokimia. *Healthy drink* umumnya bertujuan untuk meningkatkan hidrasi tubuh, mendukung kesehatan secara menyeluruh atau fungsi organ tertentu, dan membantu mengatasi penyakit tertentu misalnya minuman jahe untuk meredakan mual [8].

Manfaat mengonsumsi *healthy drink* secara teratur bagi kesehatan sangatlah banyak, antara lain:

- 1) Meningkatkan Energi: Minuman seperti air kelapa dan jus buah menyediakan energi alami dari gula alami.
- 2) Meningkatkan Kekebalan Tubuh: Minuman kaya vitamin C seperti jus jeruk dan jambu membantu meningkatkan sistem kekebalan tubuh.
- 3) Menjaga Kesehatan Pencernaan: Minuman probiotik seperti yogurt dan kefir membantu menjaga kesehatan pencernaan.
- 4) Detoksifikasi: Minuman seperti air putih dan teh hijau membantu membuang racun dalam tubuh.
- 5) Meningkatkan Kesehatan Kulit: Minuman kaya antioksidan seperti teh hijau dan jus delima membantu menjaga kesehatan kulit.
- 6) Menurunkan Risiko Penyakit Kronis: Minuman sehat dapat membantu menurunkan risiko penyakit kronis seperti penyakit jantung, stroke, diabetes, dan kanker [9].

Ada banyak jenis *healthy drink* yang tersedia, berikut beberapa contohnya:

- 1) Air Putih: Minuman paling penting untuk hidrasi tubuh. Pastikan untuk minum air putih minimal 8 gelas per hari.
- 2) Jus Buah dan Sayur: Kaya vitamin, mineral, dan serat. Pilihlah jus yang terbuat dari buah dan sayur segar tanpa tambahan gula.
- 3) Susu Rendah Lemak: Sumber kalsium dan protein yang baik. Pilihlah susu rendah lemak atau tanpa lemak untuk pilihan yang lebih sehat.

- 4) Teh Hijau: Kaya antioksidan dan membantu meningkatkan metabolisme. Konsumsi teh hijau tanpa gula untuk mendapatkan manfaat maksimal.
- 5) Minuman Berbahan Rempah: Seperti kunyit, jahe, atau temulawak, memiliki sifat anti-inflamasi dan membantu pencernaan.
- 6) Kopi: Mengandung kafein yang dapat meningkatkan kewaspadaan dan fokus. Konsumsi kopi secukupnya, dan hindari menambahkan gula berlebihan.
- 7) Air Kelapa: Kaya elektrolit dan membantu rehidrasi tubuh. Minumlah air kelapa setelah berolahraga atau saat cuaca panas [10].

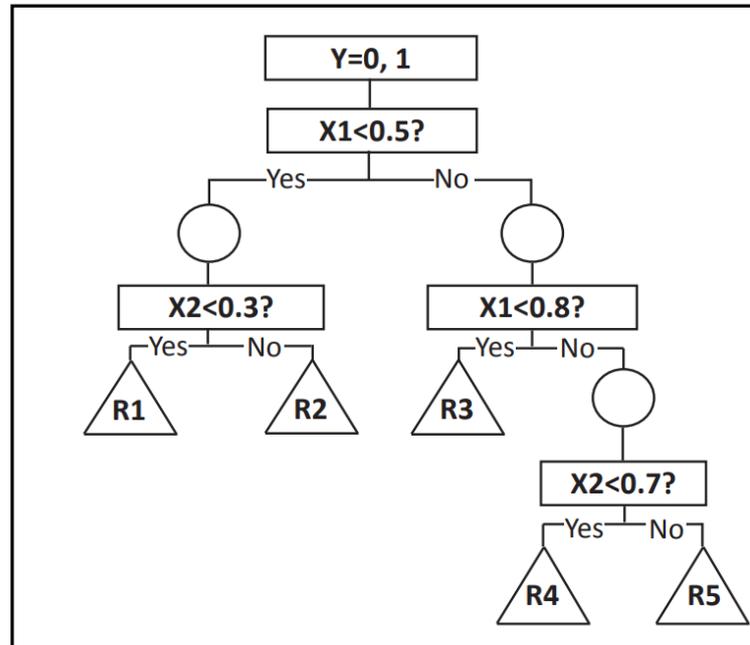
2.6 Rekomendasi

Rekomendasi adalah saran atau pendapat yang diberikan untuk membantu seseorang dalam membuat keputusan atau mengambil tindakan tertentu. Rekomendasi dapat muncul dalam berbagai konteks, seperti akademik, profesional, medis, dan banyak lagi. Dalam konteks akademik atau penelitian, rekomendasi biasanya didasarkan pada analisis data, hasil penelitian, dan tinjauan literatur yang komprehensif. Untuk memberikan rekomendasi yang kredibel dan valid, sering kali merujuk pada jurnal ilmiah dan buku penting. Rekomendasi adalah kesimpulan yang ditarik berdasarkan analisis dan evaluasi data yang bertujuan untuk memberikan saran atau tindakan yang harus diambil untuk mencapai hasil yang diinginkan [11].

2.6.1 Decision Tree

Metodologi decision tree adalah metode penambangan data yang umum digunakan untuk membangun sistem klasifikasi berdasarkan beberapa kovariat atau untuk mengembangkan algoritma prediksi untuk variabel target. Metode ini mengklasifikasikan suatu populasi ke dalam segmen-segmen seperti cabang yang membentuk pohon terbalik dengan simpul akar, simpul internal, dan simpul daun. Algoritme ini bersifat non-parametrik dan dapat secara efisien menangani kumpulan data yang besar dan rumit tanpa menerapkan struktur parametrik yang rumit. Jika ukuran sampel cukup besar, data penelitian dapat dibagi menjadi

kumpulan data pelatihan dan validasi. Menggunakan kumpulan data pelatihan untuk membangun model pohon keputusan dan kumpulan data validasi untuk menentukan ukuran pohon yang sesuai yang diperlukan untuk mencapai model akhir yang optimal [12].



Gambar 2.10 Contoh Decision Tree

Gambar 2.10 mengilustrasikan model pohon keputusan sederhana yang mencakup satu variabel target biner Y (0 atau 1) dan dua variabel kontinu, x_1 dan x_2 , yang berkisar dari 0 hingga 1. Komponen utama model pohon keputusan adalah simpul dan cabang dan langkah terpenting dalam membangun model adalah pemisahan, penghentian, dan pemangkasan.

Node. Ada tiga jenis node. (a) Simpul akar, juga disebut simpul keputusan, mewakili pilihan yang akan menghasilkan pembagian semua catatan menjadi dua atau lebih subset yang saling eksklusif. (b) Node internal, juga disebut node peluang, mewakili salah satu kemungkinan pilihan yang tersedia pada titik tersebut dalam struktur pohon; tepi atas node terhubung ke node induknya dan tepi bawah terhubung ke node anak atau node daunnya. (c) Node daun, juga disebut node akhir, mewakili hasil akhir dari kombinasi keputusan atau peristiwa.

Ranting. Cabang mewakili hasil atau kejadian kebetulan yang berasal dari simpul akar dan simpul internal. Model pohon keputusan dibentuk menggunakan

hierarki cabang. Setiap jalur dari node akar melalui node internal ke node daun mewakili aturan keputusan klasifikasi. Jalur pohon keputusan ini juga dapat direpresentasikan sebagai aturan ‘jika-maka’. Misalnya, “jika kondisi 1 dan kondisi 2 dan kondisi... dan kondisi k terjadi, maka hasil j terjadi.”

Pemisahan. Hanya variabel masukan yang terkait dengan variabel target yang digunakan untuk membagi node induk menjadi node anak yang lebih murni dari variabel target. Variabel masukan diskrit dan variabel masukan kontinu (yang dipecah menjadi dua kategori atau lebih) dapat digunakan .

2.7 Aplikasi Mobile

Aplikasi mobile adalah aplikasi dari sebuah perangkat lunak yang digunakan pada perangkat mobile seperti Smartphone atau Tablet. Terdapat dua jenis pada aplikasi mobile yaitu mobile native dan mobile web. Aplikasi mobile native merupakan aplikasi yang dibangun khusus untuk mobile dengan sistem operasi seperti iOS atau Android. Sedangkan untuk aplikasi mobile web aplikasi yang berinteraksi melalui browser namun dirancang untuk mobile dan membutuhkan koneksi internet.

Aplikasi mobile dapat berinteraksi dengan berbagai fitur yang tersedia pada smartphone seperti GPS, kamera, sensor akselerometer, dan fitur-fitur lainnya. Dengan banyaknya fitur yang tersedia tentunya aplikasi mobile dapat membantu dan menjadi penting dari kehidupan sehari-hari sehingga aplikasi mobile menjadi teknologi yang paling banyak digunakan [13].

2.8 Android

Android merupakan sebuah sistem operasi untuk perangkat mobile pertama pada tahun 2008 yang diperkenalkan oleh Google. Android bersifat *open-source* yang di mana pengguna dapat mengakses dan memodifikasi kode dengan bebas. Banyak developer dalam suatu organisasi yang melakukan modifikasi sistem operasi Android seperti LineageOS yang merupakan sistem operasi kustom berbasis Linux dengan menambahkan banyak fitur tambahan. Android dirancang untuk beberapa perangkat seperti *smartphone*, *smartwatch*, tablet, *smart TV*, dan sebagainya.

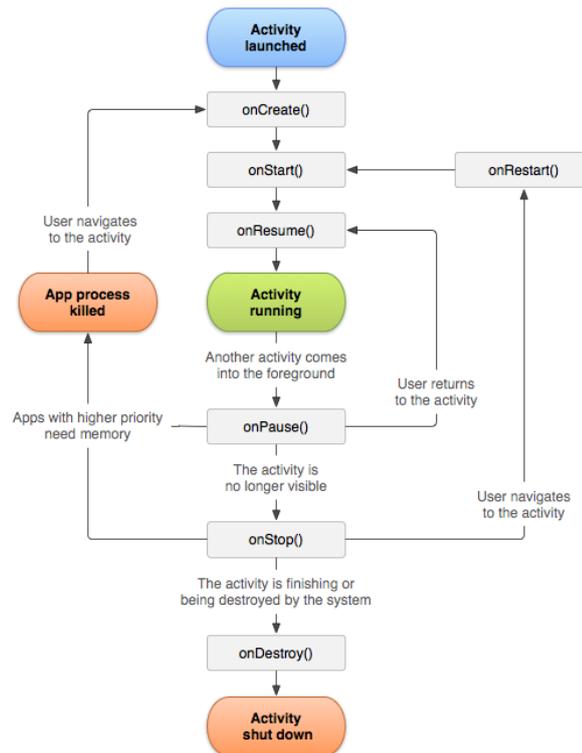
Sistem operasi Android terus berkembang dari tahun ke tahun, dari muali versi pertama yang dikenalkan yaitu Android 1.0 (Alpha) hingga Android terbaru saat ini yaitu Android 13 (Tiramisu) yang mempunyai segudang fitur seperti *Material Design* atau *Material U* yang pertama kali diperkenalkan pada Android 12 (*Snow Cone*) dan berfokus pada sistem keamanan untuk penggunanya.

Sistem Android menggunakan kernel Linux yang mempunyai arsitektur berbasis aplikasi yang memungkinkan untuk menjalankan lebih dari satu aplikasi secara bersamaan. Android juga memiliki SDK atau *Software Development Kit* yang memungkinkan pengembang untuk membuat dan mengembangkan aplikasi Android menggunakan bahasa pemrograman seperti Java atau Kotlin.

Aplikasi yang dijalankan pada Android berformat *Android Package Kit* (APK) yang di dalamnya berisi semua kode program, data dan sumber daya yang diperlukan. Penggunaanya dapat memasang aplikasi dengan berbagai cara seperti melalui toko resmi Google Play Store atau sumber lain [14].

2.8.1 Android Life Cycle

Android Life Cycle merupakan siklus hidup pada aplikasi yang digunakan oleh sistem operasi Android untuk mengatur dan mengelola aktivitas pada aplikasi. Aktivitas merupakan komponen utama yang menunjukkan interface dari aplikasi. Aktivitas memiliki siklus hidup yang terdiri dari serangkaian metode yang digunakan oleh sistem sesuai dengan status dari aktivitas tersebut. Pengembang aplikasi berbasis Android perlu memahami konsep dari siklus hidup ini untuk membangun aplikasi yang mempunyai performa baik, minim kesalahan, dan stabil. Selain itu, pengembang juga harus mengetahui jika terjadi perubahan pada siklus hidup seperti rotasi layar, masuk atau keluar dari suatu aktivitas, dan sebagainya. Berikut ilustrasi sederhana dari siklus proses aktivitas [14] dapat dilihat pada Gambar 2.11.



Gambar 2.11 Ilustrasi *Android Life Cycle*

Penjelasan pada gambar mengenai *Android Life Cycle* dari mulai aktivitas dibuat hingga aktivitas dimatikan adalah sebagai berikut:

1. onCreate

Metode yang akan dijalankan saat awal aktivitas dijalankan atau sebuah aktivitas dibuat.

2. onStart

Metode yang dijalankan setelah metode *onCreate()* selesai dieksekusi. Pada metode ini aktivitas dapat dilihat oleh pengguna

3. onPause

Metode yang terjadi karena pengguna aplikasi meninggalkan aktivitas atau menghentikan aplikasi secara sementara karena pengguna membuka aplikasi lain. Metode ini tidak menjalankan perintah apa pun dan aktivitas masih bisa berjalan di belakang layar.

4. onResume

Metode untuk memasuki atau melanjutkan aktivitas yang sebelumnya dihentikan sementara. Metode aplikasi ini juga bisa disebut mengembalikan aktivitas yang berada pada belakang layar ke latar depan agar dapat terlihat oleh pengguna.

5. onStop

Metode yang dijalankan baik secara paksa atau tidak oleh sistem Android saat aktivitas tidak dapat dilihat oleh pengguna. Metode ini digunakan untuk menghentikan aktivitas yang sudah tidak perlu dijalankan saat komponen-komponen tidak terlihat pada layar.

6. onDestroy

Metode yang dipanggil sebelum aktivitas ditutup dan akan menghapus aktivitas dan komponen-komponen di dalamnya. Metode ini terjadi saat pengguna keluar aplikasi dan menghapus aplikasi dari belakang layar.

2.8.2 Android Studio

Android Studio adalah software *Integrated Development Environment* (IDE) resmi untuk sistem operasi Android untuk pengembangan aplikasi Android. Android Studio dibangun oleh JetBrains yang dirancang khusus untuk pengembangan Android. Android Studio juga memungkinkan pengembang untuk membuat dan mengelola proyek Android yang kompleks, juga terdapat berbagai fitur seperti penyelesaian kode secara otomatis, *testing*, dan *debugging* [14].

Berdasarkan penelitian yang dilakukan bahwa para pengembang Android pada umumnya memiliki pengetahuan dan kesadaran yang terbatas dalam membangun aplikasi yang sesuai dengan model *lifecycle*. Hal tersebut tentunya memengaruhi performa aplikasi yang dibangun. Android Studio dapat mengelola kesesuaian *lifecycle* secara otomatis untuk menjaga performa aplikasi yang dibangun terutama bagi pengembang aplikasi Android baru yang masih kurang pengalaman [14].

2.8.3 Android Architecture

Arsitektur Android merupakan sebuah kerangka kerja yang digunakan untuk mengembangkan aplikasi Android dengan terstruktur. Dengan menggunakan arsitektur Android, pengembang lebih mudah dalam mengembangkan aplikasi Android karena dapat memisahkan tugas-tugas aplikasi menjadi beberapa lapisan serta mengurangi duplicate code [15]. Terdapat beberapa arsitektur yang sering digunakan dalam pengembangan aplikasi Android antara lain sebagai berikut:

- 1) *Model-View-Controller* (MVC), memisahkan kode menjadi tiga komponen utama yaitu *model* (data), *view* atau tampilan, dan *controller*.
- 2) *Model-View-Presenter* (MVP), memisahkan kode menjadi tiga komponen utama yaitu *model* (data), *view* atau tampilan, dan *presenter* yang bertugas sebagai pengendali antara data dan tampilan.
- 3) *Model-View-ViewModel* (MVVM), memisahkan kode menjadi tiga komponen utama yaitu *model* (data), *view* atau tampilan, dan *ViewModel* yang bertugas sebagai perantara antara tampilan dan model yang memungkinkan pengikatan atau *data binding* agar lebih mudah dan terstruktur.
- 4) *Clean Architecture*, membagi kode ke dalam beberapa lapisan yang terpisah seperti *presentasi*, *domain*, dan *data*. Setiap lapisan berkomunikasi yang dilakukan melalui *interface*.

Penggunaan arsitektur MVVM menjadikan pengembangan aplikasi lebih mudah karena terpisah dari *Android Lifecycle* yang memungkinkan untuk digunakan diberbagai tampilan yang memiliki struktur data yang sama. Secara keseluruhan, MVVM dapat mengurangi keterikatan kode, mengurangi kode duplikat [15].

2.8.4 Services

Services merupakan sebuah komponen aplikasi yang dapat beroperasi lama di background dan tidak menampilkan tampilan antarmuka. Komponen aplikasi ini akan terus berjalan di background meskipun pengguna mengeluarkan dan menutup aplikasi atau berpindah ke aplikasi lain. Service ini memungkinkan untuk

menjalankan transaksi jaringan, media player, melakukan input atau output, dan lain sebagainya. Beberapa jenis service diantaranya adalah sebagai berikut:

1) Foreground Service

Service yang memungkinkan untuk melakukan beberapa operasi yang terlihat oleh pengguna melalui notifikasi. Misalnya aplikasi pemutar musik [16].

2) Background Service

Service yang melakukan operasi di background yang tidak akan terlihat oleh pengguna. Misalnya aplikasi yang mendeteksi atau membersihkan penyimpanan perangkat [16].

3) Bound Service

Service yang akan terikat bila komponen aplikasi mengikat dan memanggil fungsi *bindService()*. *Service* ini menawarkan tampilan antarmuka *client-server* yang memungkinkan komponen ini berinteraksi dengan *service* tersebut [16].

Service dapat memberikan keuntungan yang signifikan dalam pengembangan sebuah aplikasi Android. Aplikasi memungkinkan untuk melakukan tugas jangka panjang di background tanpa membebani pengguna. Hal yang harus diperhatikan dari *service* yaitu dalam penggunaannya karena dapat mengganggu kinerja perangkat dan menguras baterai [16].

2.9 Visual Studio Code

Visual Studio Code merupakan kode editor yang ringan dan dapat dijalankan di berbagai sistem operasi seperti Windows, MacOS, dan Linux. Kode editor ini mendukung banyak bahasa pemrograman, plug in, framework dan library. Beberapa bahasa pemrograman diantaranya adalah JavaScript, C++, Python, Java, dan lain-lain. Kode editor ini digunakan oleh banyak pengembang karena ringantetapi powerfull untuk mengembangkan suatu aplikasi atau sistem dengan sifatnya yang open source sehingga dapat dikembangkan oleh siapa saja [17].

2.10 Kotlin

Kotlin merupakan salah satu pemrograman yang berjalan pada Java Virtual Machine (JVM) yang dirilis tahun 2011 dan dikembangkan oleh JetBrains. Kotlin mendapatkan popularitas dalam waktu singkat dengan fitur-fitur yang modern seperti null-safety, operation overloading, extension function, lambda expressions, dan lainnya yang membantu pengembang untuk menuliskan kode agar mudah dibaca dan lebih efisien [18].

Kotlin menjadi bahasa pemrograman utama untuk pengembangan Android yang diumumkan oleh Google secara resmi pada tahun 2017. Kotlin memiliki keuntungan interoperabilitas dengan Java yang di mana Kotlin dapat menggunakan kode Java yang sudah ada serta mempunyai syntax dan kemampuan untuk mengurangi jumlah kode yang ditulis [18].

2.11 JavaScript

JavaScript atau dikenal dengan singkatan JS, merupakan bahasa pemrograman script tingkat tinggi yang dikompilasi secara cepat Just-in-time Compilation (JIT). JavaScript adalah bahasa pemrograman yang memungkinkan untuk membuat halaman web interaktif dan merupakan bagian penting dari web. JavaScript merupakan bahasa Multi-paradigma yang mendukung gaya pemrograman berbasis fungsional dan imperatif [19].

JavaScript dapat digunakan untuk pengembangan aplikasi web baik frontend untuk tampilan interface maupun back-end untuk pengolahan data server dan database. Awalnya JavaScript dibuat untuk lingkungan browser dan membuat website menjadi interaktif. Namun, saat ini JavaScript dapat berjalan di luar browser seperti menggunakan Node.js [19].

2.12 NodeJs

NodeJs adalah platform runtime JavaScript yang dibangun untuk mengeksekusi kode JavaScript di luar browser. NodeJs telah banyak digunakan oleh perusahaan-perusahaan besar seperti Netflix, Paypal, eBay, dan lainnya. NodeJs menggunakan model pemrosesan non-blocking I/O yang memungkinkan untuk menangani banyak permintaan I/O secara bersamaan tanpa menghentikan

proses permintaan yang lain. Selain itu, NodeJs mendukung pengembangan aplikasi secara real-time dan streaming [19].

NodeJs memiliki banyak modul yang sudah terpasang di dalamnya serta mengandung modul pihak ketiga yang dapat dipasang melalui Node Package Manager (NPM) yang dapat dengan mudah diintegrasikan dengan teknologi lain. NodeJs digunakan untuk pembangunan aplikasi web, server, desktop, mobile, dan lainnya [19].

2.13 Extensible Markup Language (XML)

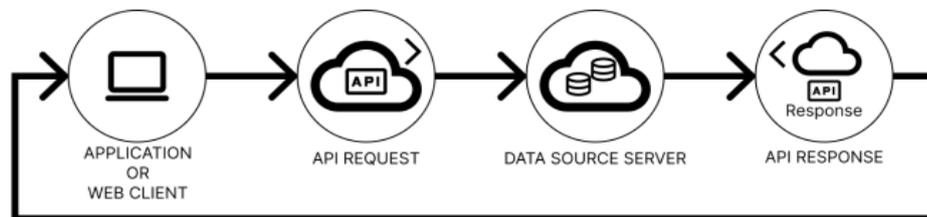
Extensible Markup Language atau disingkat XML merupakan bahasa markup yang sering digunakan untuk menampilkan data dalam bentuk teks yang terstruktur. XML biasanya digunakan saat pertukaran data antar aplikasi yang berbeda bahasa pemrograman serta platform yang digunakan.

XML digunakan di berbagai bidang seperti pada web servis, pengembangan game, pengembangan antarmuka aplikasi mobile, dan lain sebagainya. Struktur bahasa XML direpresentasikan dalam bentuk tag, atribut, dan nilai. Tag digunakan untuk menentukan jenis data yang digunakan, atribut digunakan untuk memberikan informasi terkait dengan data yang digunakan, dan nilai merupakan isi atau data itu sendiri.

2.14 Application Programming Interface (API)

Application Programming Interface atau disingkat API yang memungkinkan aplikasi atau web berkomunikasi dengan server. API memungkinkan pengembang untuk membuat aplikasi yang dapat menggunakan fungsi dari aplikasi lain atau web service dengan cara mengintegrasikannya. Tujuannya adalah untuk saling bertukar data antar aplikasi dan mempercepat proses pengembangan aplikasi. REST API merupakan desain arsitektur yang terdapat di dalam API dan RESTful API merupakan REST client yang mengakses data pada REST server dari masing-masing sumber daya. Dengan tujuan dan cara kerja tersebut tentunya memudahkan pengembang dalam mengembangkan aplikasi karena pengembang dapat memakai fitur yang sudah ada tanpa perlu mengetahui

detail bagaimana sumber daya itu diimplementasikan. Berikut merupakan alur cara kerja dari API [20] dapat dilihat pada Gambar 2.12.



Gambar 2.12 Cara Kerja API

Cara kerja API yaitu pertama aplikasi atau web client melakukan request untuk mengakses data pada sumber data di server. Lalu, server akan merespons permintaan dan mengembalikan data yang diminta yang biasanya berformat teks, JSON atau XML. Metode yang digunakan secara umum yaitu menggunakan metode HTTP yang dipakai dalam REST API seperti GET, POST, PUT, DELETE, dan OPTIONS [20].

Penjelasan metode HTTP yang dipakai dalam REST API adalah sebagai berikut:

- 1) GET, digunakan untuk mendapatkan data dari REST server.
- 2) POST, digunakan untuk membuat suatu data/sumber daya baru pada REST server.
- 3) PUT, digunakan untuk mengubah dan memperbaharui data/sumber daya pada REST server.
- 4) DELETE, digunakan untuk menghapus data/sumber daya pada REST API.
- 5) OPTIONS, digunakan untuk mendapatkan operasi yang didukung pada sumber daya dari REST server.

2.15 JavaScript Object Notation (JSON)

JavaScript Object Notation atau disingkat JSON merupakan format data yang mudah dibaca dan ditulis oleh manusia maupun mesin. Biasanya JSON digunakan sebagai format dalam pertukaran data antar aplikasi atau sistem yang

menggunakan API. Struktur data pada JSON yaitu berbentuk objek (object) dan array [21].

Objek merupakan kumpulan pasangan informasi data berupa nama dan nilai ditulis dengan cara dibungkus dengan “{}” seperti contoh berikut “{nama: nilai}”. Sedangkan array ditulis dengan cara dibungkus dengan “[]” dan pada setiap data 22 yang dikirimkan dipisahkan dengan tanda koma “,”. Tipe data JSON mendukung berbagai tipe data seperti String, Boolean, Integer, null, dan lain sebagainya. Keuntungan menggunakan JSON yaitu bahasa yang mudah dibaca oleh manusia dan komputer serta ringan sehingga proses atau operasi yang dijalankan dapat berjalan dengan cepat dan dapat memproses gigabyte data per detik pada satu inti prosesor yang umum [21].

2.16 Web Service

Web service adalah layanan web yang digunakan untuk berkomunikasi satu sama lain antara perangkat elektronik satu ke perangkat elektronik lain melalui World Wide Web (WWW) atau server yang berjalan pada komputer, menerima permintaan di port tertentu melalui jaringan yang melayani dokumen web seperti HTML, JSON, XML, dan sebagainya, lalu membuat layanan web yang berfungsi untuk menyelesaikan masalah domain tertentu (WWW, HTTP).

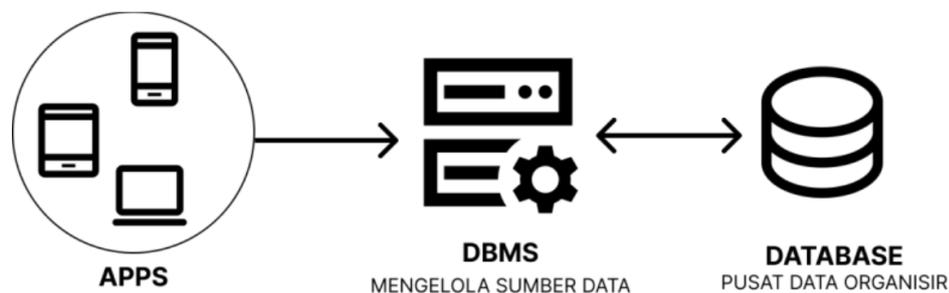
Salah satu cara untuk melakukan permintaan tersebut yaitu dengan memasukkan informasi ke dalam metode HTTP yang di mana memiliki lima metode yang umum digunakan seperti GET, HEAD, PUT, DELETE, dan POST. Tentunya HTTP sudah terstandarisasi meskipun beberapa web service memilih untuk menggunakan nama metode yang lebih spesifik untuk aplikasi seperti menggunakan jalur URI atau dokumen permintaan [22].

2.17 Basis Data MySQL (Database)

Basis dapat diartikan sebagai tempat yang digunakan untuk mengumpulkan dan menyimpan barang atau benda. Data merupakan representasi fakta pada dunia nyata yang dapat berupa objek seperti manusia, benda, dan lainnya yang disimpan dalam bentuk teks, angka, bunyi gambar, simbol, atau kombinasi dari semuanya. Jadi, basis data merupakan kumpulan data yang saling berhubungan satu sama lain

yang diorganisasikan berdasarkan skema atau struktur tertentu, kemudian disimpan di dalam perangkat keras komputer dan digunakan dengan perangkat lunak untuk mengoperasikannya seperti memperbaharui data, mencari data, menambah data, dan menghapus data dengan tujuan tertentu [23].

MySQL adalah sebuah sistem pengorganisasian dan pengolahan database pada komputer atau bisa disebut Database Management System (DBMS). Berikutnya yaitu Relation Database Management System (RDBMS) yang sama dengan DBMS dan mengorganisasikan data dalam suatu struktur serta menghubungkan basis data yang disimpan. MySQL itu sendiri merupakan Structured Query Language (SQL) yang merupakan perangkat lunak bersifat open source yang digunakan oleh banyak pengembang untuk membangun aplikasi yang berbasis database. Berikut merupakan gambaran dari database dapat dilihat pada gambar 2.13 .



Gambar 2.13 Database Management System (DBMS)

2.18 Jetpack Compose

Jetpack Compose adalah toolkit modern untuk membangun antarmuka pengguna (UI) asli (native) di Android. Toolkit ini dikembangkan oleh Google dan merupakan bagian dari Android Jetpack, sebuah koleksi pustaka yang membantu pengembang dalam membangun aplikasi Android. Jetpack Compose dirancang untuk menyederhanakan dan mempercepat proses pengembangan UI dengan menggunakan paradigma deklaratif, mirip dengan React dalam pengembangan web [24]. Jetpack Compose memiliki karakteristik Utama yaitu:

- 1) Deklaratif, Dalam jetpack compose UI didefinisikan menggunakan fungsi-fungsi deklaratif. Alih-alih memanipulasi antarmuka pengguna langsung, pengembang cukup mendeklarasikan tampilan UI yang diinginkan berdasarkan keadaan saat ini.
- 2) Reaktif, jetpack compose secara otomatis memperbarui UI saat data berubah. Jika ada perubahan data, compose akan memperbarui komponen UI yang relevan tanpa perlu campur tangan manual.
- 3) Kompak dan Mudah Dibaca, kode yang ditulis dengan jetpack compose biasanya lebih ringkas dan mudah dibaca dibandingkan dengan pendekatan imperative tradisional seperti XML layout dan 'findViewById'.
- 4) Interoperabilitas Dengan View Lama, jetpack compose dapat bekerja bersama dengan UI berbasis View yang sudah ada. Anda dapat menggabungkan compose dengan komponen View lama dalam satu aplikasi.
- 5) Tooling Yang Kuat, jetpack compose didukung oleh alat-alat pengembangan yang kuat, termasuk pratinjau real-time di android studio, yang memungkinkan pengembang melihat perubahan UI secara langsung saat mereka menulis kode.

2.19 Skala Likert

Skala Likert digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang fenomena sosial. Dalam penelitian, fenomena sosial ini telah ditetapkan secara spesifik oleh peneliti, yang selanjutnya disebut sebagai variabel penelitian. Dengan skala likert, maka variabel yang akan diukur dijabarkan menjadi indikator variabel. Kemudian indikator tersebut dijadikan sebagai titik tolak untuk menyusun item-item instrumen yang dapat berupa pernyataan atau pertanyaan [25].

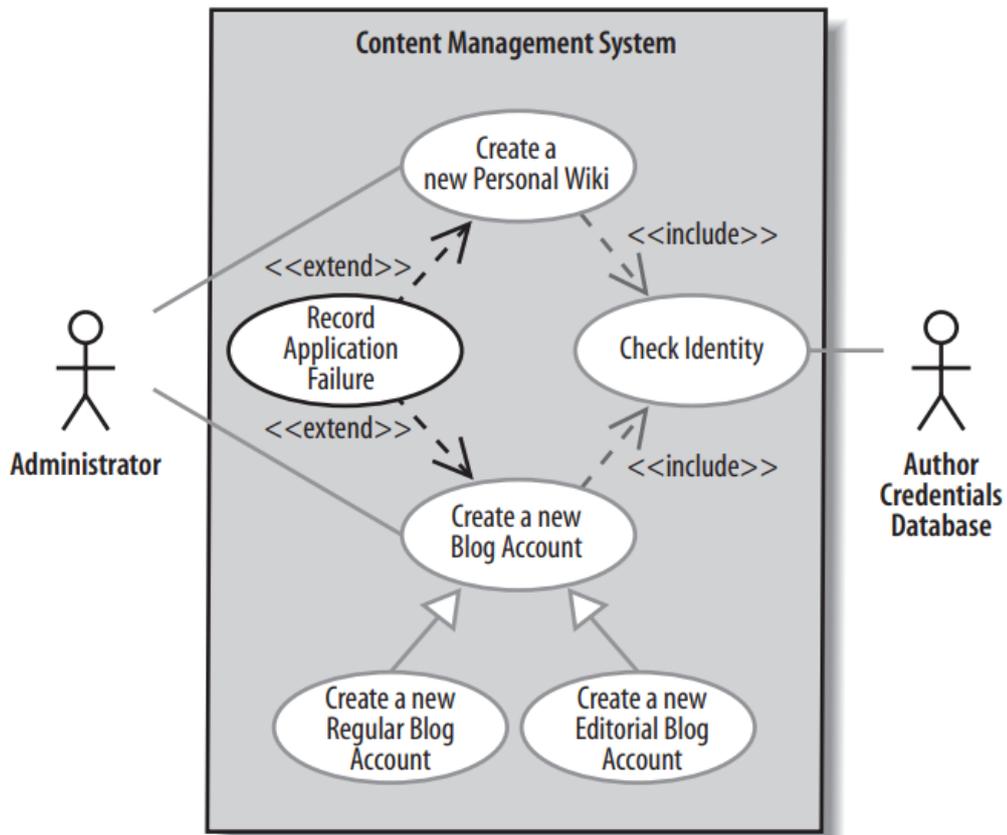
2.19 Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah bahasa untuk memvisualisasikan, menspesifikasikan, memodelkan, dan membangun dengan cara

mendokumentasikan sistem perangkat lunak berorientasi objek [26]. UML 2.0 memiliki berbagai diagram, empat diagram diantaranya yang akan digunakan pada penelitian ini yaitu Activity Diagram, Use Case Diagram, Class Diagram, dan Sequence Diagram [27].

2.19.1 Use Case Diagram

Use case adalah kasus (atau situasi) di mana sistem Anda digunakan untuk memenuhi satu atau beberapa persyaratan pengguna Anda; kasus penggunaan menangkap sebagian fungsionalitas yang disediakan sistem. Kasus penggunaan merupakan inti dari model Anda, karena kasus tersebut memengaruhi dan memandu semua elemen lain dalam desain sistem Anda. Use Case merupakan titik awal yang sangat baik untuk hampir setiap aspek pengembangan, desain, pengujian, dan dokumentasi sistem berorientasi objek. Use Case menggambarkan persyaratan sistem secara ketat dari sudut pandang luar, use case menentukan nilai yang diberikan sistem kepada pengguna. Karena use case merupakan persyaratan fungsional sistem Anda, use case harus menjadi keluaran serius pertama dari model Anda setelah proyek dimulai. Lagi pula, bagaimana Anda dapat mulai merancang sistem jika Anda tidak tahu apa yang harus dilakukan? [27]. Berikut contoh dari use case dapat dilihat pada Gambar 2.14.



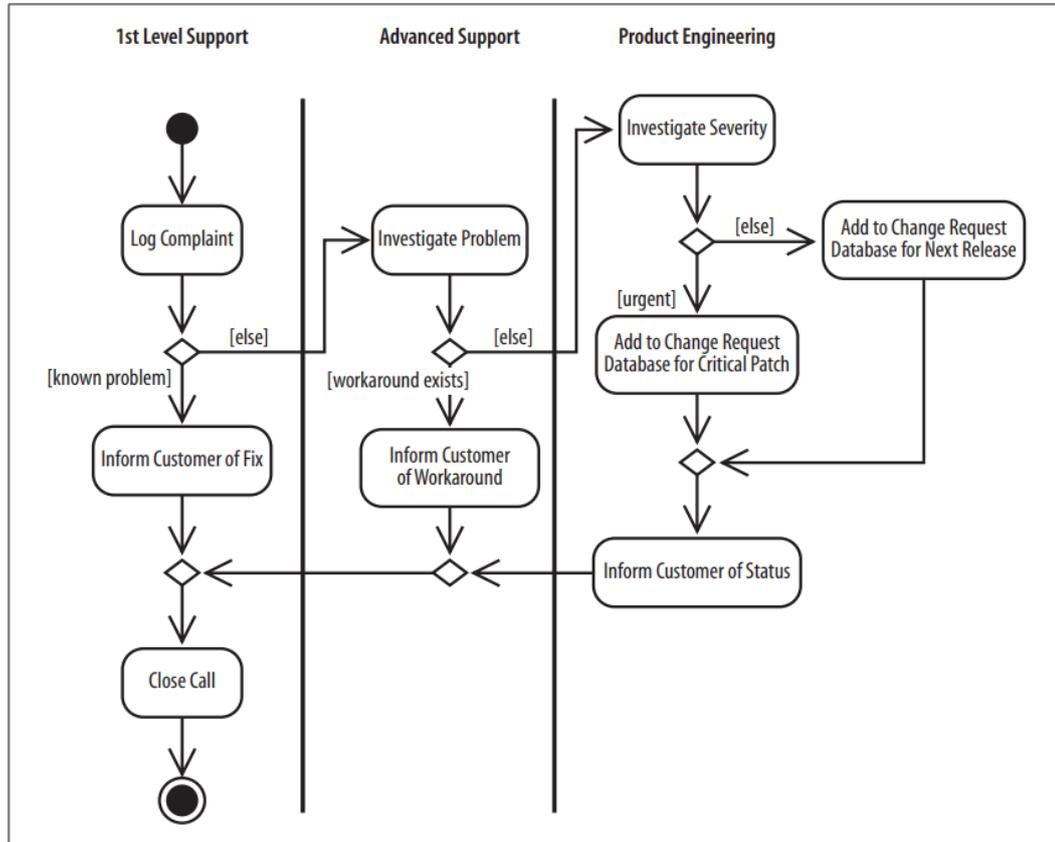
Gambar 2.14 Contoh Use Case

2.19.2 Activity Diagram

Activity Diagram sangat baik dalam memodelkan proses bisnis. Proses bisnis adalah serangkaian tugas terkoordinasi yang mencapai tujuan bisnis, seperti mengirimkan pesanan pelanggan. Beberapa alat manajemen proses bisnis (BPM) memungkinkan Anda untuk mendefinisikan proses bisnis menggunakan activity diagram, atau notasi grafis serupa, lalu menjalankannya. Ini memungkinkan Anda untuk mendefinisikan dan menjalankan, misalnya, proses persetujuan pembayaran di mana salah satu langkahnya memanggil layanan web persetujuan kartu kredit—menggunakan notasi grafis yang mudah seperti activity diagram.

Activity Diagram adalah salah satu diagram UML yang paling mudah diakses karena menggunakan simbol yang mirip dengan notasi flowchart yang dikenal luas. Oleh karena itu, activity diagram berguna untuk menggambarkan proses kepada khalayak luas. Faktanya, activity diagram berakar pada flowchart,

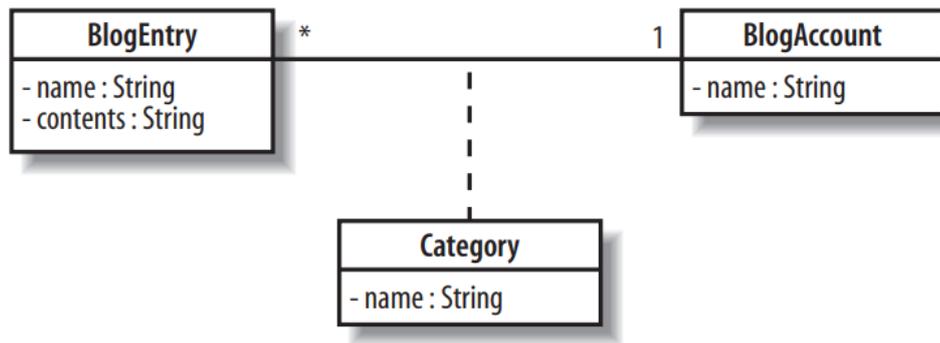
serta UML state diagram, data flow diagram, dan Petri Nets [27]. Berikut Contoh dari Activity Diagram dapat dilihat pada Gambar 2.15.



Gambar 2.15 Contoh Activity Diagram

2.19.3 Class Diagram

Class merupakan inti dari setiap sistem berorientasi objek; oleh karena itu, diagram UML yang paling populer adalah class diagram. Struktur sistem terdiri dari kumpulan bagian yang sering disebut sebagai objek. Class menggambarkan berbagai jenis objek yang dapat dimiliki sistem Anda, dan class diagram menunjukkan kelas-kelas ini dan hubungan di antara mereka. Kasus penggunaan menggambarkan perilaku sistem Anda sebagai serangkaian masalah. Kelas menggambarkan berbagai jenis objek yang dibutuhkan dalam sistem Anda untuk memenuhi masalah tersebut. Kelas merupakan bagian dari tampilan logis model Anda [27]. Berikut contoh dari class diagram dapat dilihat pada Gambar 2.16.

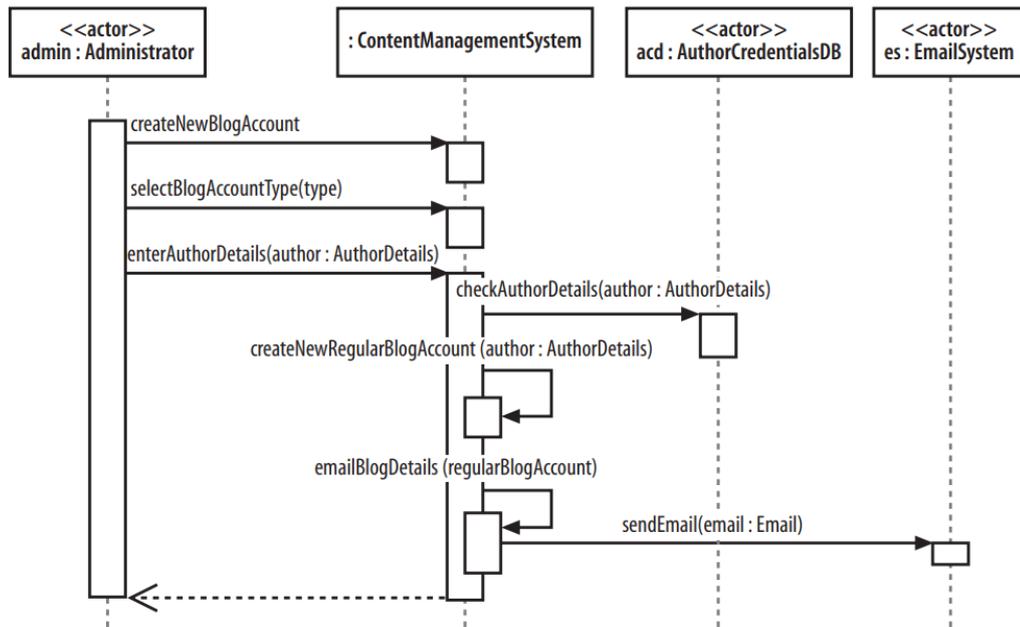


Gambar 2.16 Contoh Class Diagram

2.19.4 Sequence Diagram

Use Case memungkinkan model Anda untuk menjelaskan apa yang harus dapat dilakukan oleh sistem Anda, sedangkan class memungkinkan model Anda untuk menjelaskan berbagai jenis komponen yang membentuk struktur sistem Anda. Ada satu bagian besar yang hilang dari teka-teki ini, dengan use case dan class saja, Anda belum dapat memodelkan bagaimana sistem Anda benar-benar menjalankan tugasnya. Di sinilah diagram interaksi, dan khususnya sequence diagram, berperan.

Sequence Diagram berfungsi untuk menangkap urutan interaksi antara komponen sistem Anda. Dengan menggunakan sequence diagram, Anda dapat menjelaskan interaksi mana yang akan dipicu ketika kasus penggunaan tertentu dijalankan dan dalam urutan apa interaksi tersebut akan terjadi. Sequence Diagram menunjukkan banyak informasi lain tentang interaksi, tetapi kelebihanannya adalah cara yang sederhana dan efektif untuk mengomunikasikan urutan kejadian dalam suatu interaksi. Berikut contoh sederhana dari sequence diagram dapat dilihat pada Gambar 2.17.



Gambar 2.17 Contoh Sequence Diagram