

BAB 2

TINJAUAN PUSTAKA

2.1 Penyakit Tanaman Tomat

Penyakit pada tanaman tomat (*Solanum lycopersicum*) merupakan salah satu faktor utama yang mempengaruhi produktivitas dan kualitas buah tomat. Beberapa penyakit yang sering menyerang tanaman tomat antara lain penyakit busuk pangkal batang, layu fusarium, kudis daun (early blight), dan busuk buah. Berikut adalah penjelasan singkat mengenai beberapa penyakit tersebut:

1. Penyakit Busuk Pangkal Batang Penyakit ini disebabkan oleh jamur *Ralstonia solanacearum*. Gejala awal dari penyakit ini adalah terjadinya penurunan pertumbuhan pada tanaman dan layu pada daun-daun yang berada pada bagian bawah tanaman. Tanaman juga akan mengeluarkan lendir berwarna putih atau kuning yang berasal dari bagian pangkal batang. Dapat dilihat pada gambar 2.1.



Gambar 2.1 Penyakit Tanaman Tomat Busuk Pangkal Batang

2. Penyakit Layu Fusarium. Penyakit ini disebabkan oleh jamur *Fusarium oxysporum*. Gejala awal dari penyakit ini adalah terjadinya kekuningan pada daun-daun bagian bawah tanaman, kemudian daun-daun akan menjadi keriting, kecoklatan, dan akhirnya layu. Bagian akar yang terinfeksi juga akan mengeluarkan lendir berwarna merah kecoklatan. Dapat dilihat pada gambar 2.2.



Gambar 2.2 Penyakit Tanaman Tomat Penyakit Layu Fusarium

3. Penyakit Kudis Daun (Early Blight). Penyakit ini disebabkan oleh jamur *Alternaria solani*. Gejala awal dari penyakit ini adalah munculnya bercak-bercak kecil berwarna coklat pada daun yang kemudian semakin membesar dan mengeluarkan bercak-bercak kecil berwarna hitam di tengahnya. Penyakit ini dapat menyebabkan daun-daun mati dan jatuh. Dapat dilihat pada gambar berikut 2.3.



Gambar 2.3 Penyakit Tanaman Tomat Kudis Daun

4. Penyakit Busuk Buah (Fruit Rot). Penyakit ini disebabkan oleh beberapa jenis jamur seperti *Phytophthora infestans*, *Fusarium spp.*, dan *Rhizopus spp.* Gejala awal dari penyakit ini adalah munculnya bercak-bercak kecil berwarna coklat pada buah yang kemudian semakin membesar dan mengeluarkan bercak-bercak berwarna hitam di tengahnya. Buah yang

terinfeksi kemudian membusuk dan tidak dapat dimanfaatkan[6]. Dapat dilihat pada gambar 2.4.



Gambar 2.4 Penyakit Tanaman Tomat Buah Busuk

Penyakit pada tanaman tomat sangat penting untuk dikenali dan diatasi karena dapat berdampak pada produksi dan kualitas buah tomat. Pembangunan aplikasi pendeteksi penyakit tanaman tomat dengan memanfaatkan teknologi AI dan smartphone Android menjadi solusi yang efektif untuk mempermudah dan mempercepat pengenalan penyakit pada tanaman tomat

Menurut M. Asif et al. (2021). penggunaan teknologi AI dan Deep Learning untuk deteksi penyakit tanaman tomat. Studi ini menunjukkan bahwa teknologi AI dapat digunakan untuk mengidentifikasi jenis-jenis penyakit pada tanaman tomat dengan akurasi yang tinggi. Hal ini menjadi relevan dengan kajian dalam penelitian ini, karena aplikasi ini dapat menjadi solusi yang efektif untuk mendeteksi penyakit pada tanaman tomat. Aplikasi pendeteksi penyakit tanaman tomat yang dikembangkan dengan memanfaatkan teknologi diharapkan dapat membantu para petani dalam mengidentifikasi penyakit pada tanaman tomat dengan cepat dan akurat sehingga dapat mengurangi kerugian produksi dan meningkatkan kualitas buah tomat[7].

2.2 Smartphone

Smartphone Merupakan gabungan dari perangkat komunikasi dan perangkat penunjang kebutuhan gaya hidup dengan beberapa fitur multimedia. Seiring berkembangnya zaman, smartphone sekarang didukung dengan fitur kamera berkualitas tinggi, koneksi internet melalui WI-FI atau Jaringan seluler, layar

sentuh yang responsif, Sensor-sensor seperti GPS, accelerometer, dan gyroscope yang memungkinkan pengguna mengukur dan melacak proses aktifitas yang sedang berjalan pada smartphone. Pada umumnya smartphone memiliki prosesor yang sangat tinggi berkat adanya teknologi SoC (System on Chip) yang memiliki kemampuan hardware yang tinggi namun dalam ukuran yang kecil[8].

2.3 Android



Gambar 2.5 Logo Android

Android adalah sistem operasi open source yang dikembangkan oleh Google untuk perangkat mobile, seperti smartphone, tablet, televisi pintar, dan wearable devices. Sistem operasi ini dirancang berbasis pada kernel Linux dan menggunakan bahasa pemrograman Java untuk membuat aplikasi. Android memiliki beberapa fitur penting, antara lain:

1. Tersedia dalam berbagai bahasa dan mendukung berbagai jenis perangkat mobile.
2. Menggunakan antarmuka pengguna yang intuitif dan mudah digunakan.
3. Dapat diakses melalui Google Play Store, yang memungkinkan pengguna untuk mengunduh dan menginstal berbagai aplikasi.
4. Mendukung multitasking, sehingga pengguna dapat menjalankan beberapa aplikasi secara bersamaan.
5. Android terus ditingkatkan dan diperbarui oleh Google untuk meningkatkan performa dan keamanan.

Android juga memiliki beberapa versi yang telah dirilis, dimulai dari versi 1.0 pada tahun 2008 hingga versi terbaru saat ini, yaitu Android 12. Setiap versi memiliki fitur dan peningkatan yang berbeda untuk meningkatkan kinerja dan keamanan. Berikut adalah versi android yang sudah rilis hingga tahun 2023. [9].

Tabel 2.1 Versi Android

No	Nama Versi	Versi	Tanggal Rilis
1	Alpha	1.0	23 September 2008
2	Beta	1.1	9 Februari 2009
3	Cupcake	1.5	27 April 2009
4	Donut	1.6	15 September 2009
5	Eclair	2.0 – 2.1	26 Oktober 2009
6	Frozen Yogurt (Froyo)	2.2	20 Mei 2010
7	Gingerbread	2.3	6 Desember 2010
8	Honeycomb	3.0 – 3.2	22 Februari 2011
9	Ice Cream Sandwich	4	18 Oktober 2011
10	Jelly Bean	4.1 – 4.3	9 Juli 2012
11	KitKat	4.4	31 Oktober 2013
12	Lollipop	5.0	12 November 2014
13	Marshmallow	6.0	5 Oktober 2015
14	Nougat	7.0 – 7.1	22 Agustus 2016
15	Oreo	8.0 – 8.1	21 Agustus 2017
16	Pie	9.0	6 Agustus 2018
17	Android Q	Android 10	3 September 2019
18	Red Velvet Cake	Android 11	11 September 2020
19	Snow Cone	Android 12	18 Februari 2021
20	Tiramisu	Android 13	27 April 2022

2.4 Application Programming interface (API)

API (Application Programming Interface) adalah set instruksi yang mengizinkan berbagai aplikasi atau layanan web untuk berkomunikasi satu sama lain dan bertukar informasi. API memungkinkan pengembang untuk membuat aplikasi atau layanan web yang terintegrasi dengan sistem lain, tanpa perlu mengetahui seluk-beluk bagaimana sistem tersebut berfungsi.

Salah satu contoh penerapan API adalah pada layanan web seperti Facebook, Twitter, dan Instagram. Dengan menggunakan API, pengembang dapat membuat aplikasi yang terhubung dengan platform sosial media ini dan memanfaatkan data yang tersedia di dalamnya, seperti profil pengguna, postingan, foto, dan lain sebagainya. API juga digunakan pada layanan web seperti Google Maps, PayPal, dan banyak lagi.

API biasanya menggunakan protokol HTTP untuk mengirim permintaan dan menerima respons. Permintaan terdiri dari URL dan parameter-parameter yang diperlukan untuk melakukan operasi tertentu, seperti membaca data, menulis data, atau melakukan operasi lainnya. Respons biasanya dalam format JSON atau XML dan berisi data yang diminta oleh permintaan[10].

2.5 Teacheble Machine



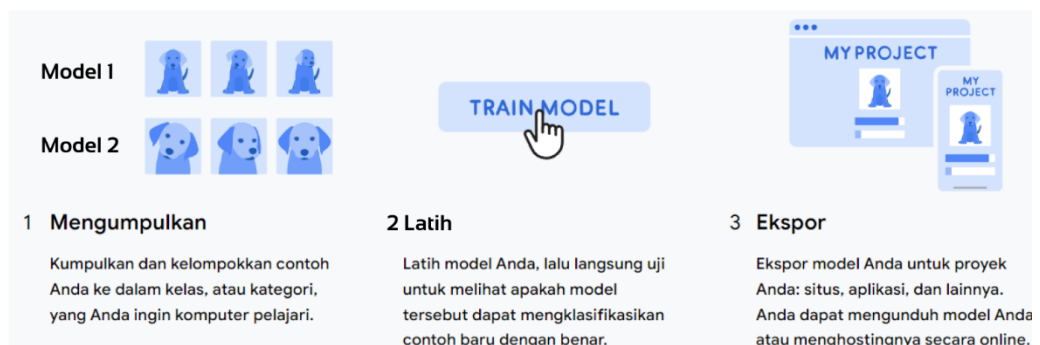
Gambar 2.6 Logo Teachable Machine

Teachable Machine adalah sebuah platform pembelajaran mesin yang dikembangkan oleh Google Creative Lab. Platform ini memungkinkan pengguna untuk melatih model pembelajaran mesin dengan mudah, tanpa harus memiliki pengetahuan mendalam tentang pemrograman atau kecerdasan buatan. Teachable Machine menggunakan teknik pengenalan pola visual yang memungkinkan pengguna untuk melatih model berdasarkan gambar atau video yang diberikan. Model yang telah dilatih dapat digunakan untuk berbagai aplikasi, termasuk deteksi penyakit pada tanaman.

Cara penggunaan Teachable Machine dalam pembangunan aplikasi pendeteksi penyakit tomat melibatkan beberapa langkah Seperti Berikut.

1. Pertama, pengguna perlu mengumpulkan dataset gambar tomat yang terdiri dari gambar tomat sehat dan gambar tomat yang terinfeksi penyakit. Dataset ini akan digunakan sebagai data latih untuk melatih model pembelajaran mesin.
2. Selanjutnya, pengguna perlu melatih dan label pada setiap gambar dalam dataset, misalnya "sehat" dan "penyakit", agar model dapat mempelajari perbedaan antara kedua kategori tersebut lalu melatih model tersebut.
3. Setelah itu, pengguna dapat mengekspor model yang sudah dilatih ke dalam aplikasi.

Beberapa langkah dapat digambarkan sebagai berikut :

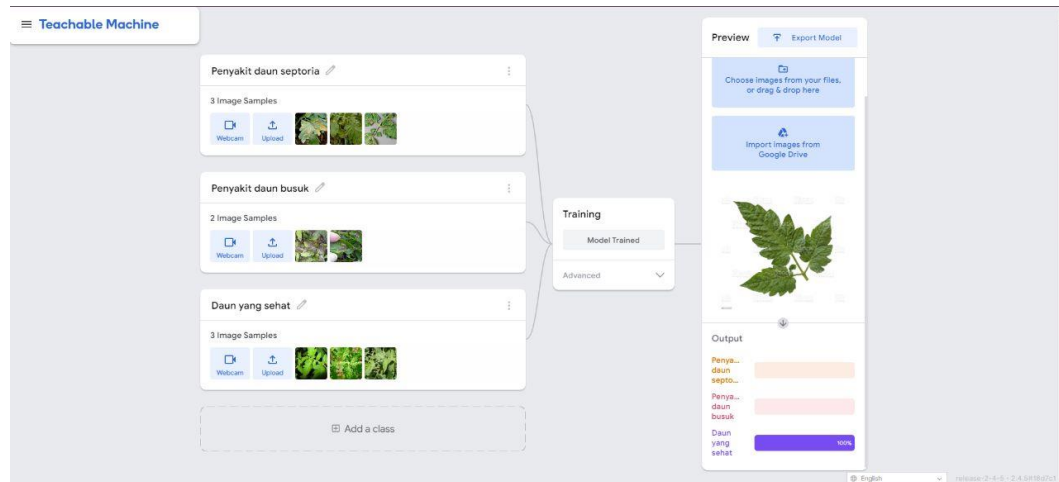


Gambar 2.7 Langkah-langkah penggunaan Teachable Machine

Setelah dataset dan label telah disiapkan, pengguna dapat melatih model menggunakan Teachable Machine. Platform ini akan memproses dataset dan menghasilkan model pembelajaran mesin yang dapat digunakan untuk mendeteksi penyakit pada tomat. Pengguna dapat menguji model dengan memberikan gambar tomat yang belum pernah dilihat sebelumnya dan melihat apakah model memberikan prediksi yang akurat berdasarkan label yang telah diberikan sebelumnya.

Integrasi model yang telah dilatih ke dalam aplikasi pendeteksi penyakit tomat dapat dilakukan dengan menggunakan API yang disediakan oleh Teachable Machine. Dengan menggunakan API ini, pengguna dapat menghubungkan aplikasi dengan model yang telah dilatih sehingga pengguna

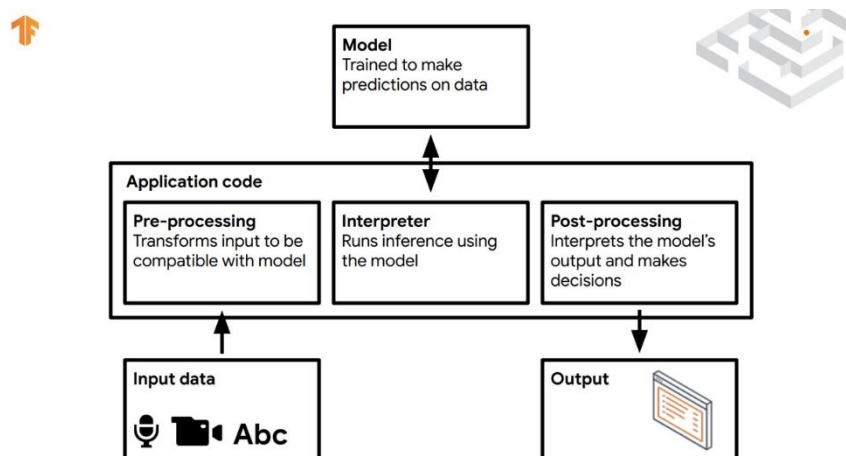
dapat memotret atau mengunggah gambar tomat melalui aplikasi dan mendapatkan prediksi apakah tomat tersebut sehat atau terinfeksi penyakit[11]



Gambar 2.8 Teacheble Machine

2.6 Tensorflow Lite API

TensorFlow Lite (TFLite) adalah versi ringan dari TensorFlow yang dirancang khusus untuk perangkat seluler, mikrokontroler, dan lingkungan sumber daya terbatas lainnya. TFLite memungkinkan pengguna untuk menjalankan model pembelajaran mesin yang telah dilatih menggunakan TensorFlow pada perangkat dengan sumber daya terbatas, sehingga cocok untuk aplikasi di perangkat mobile seperti smartphone. Berikut adalah arsitektur dari tensorflow lite.



Gambar 2.9 Arsitektur TensorFlow Lite

1. Model dalam TensorFlow Lite adalah representasi dari arsitektur dan parameter yang telah dipelajari selama proses pelatihan menggunakan data pelatihan. Model ini telah diadaptasi agar dapat dijalankan dengan cepat dan efisien pada perangkat seluler dan edge device, sehingga dapat digunakan dalam aplikasi-aplikasi yang berjalan di perangkat dengan keterbatasan sumber daya, seperti ponsel pintar, tablet, dan perangkat IoT.
2. Preprocessing dalam konteks TensorFlow Lite mengacu pada serangkaian langkah atau proses pra-pemrosesan yang dilakukan pada data masukan sebelum data tersebut diberikan ke model TensorFlow Lite untuk melakukan inferensi atau prediksi. Tujuan dari preprocessing adalah untuk mempersiapkan data masukan agar sesuai dengan kebutuhan model dan memastikan bahwa data masukan memiliki format yang benar sebelum dijalankan melalui model.
3. Input data pada TensorFlow Lite adalah data masukan yang diberikan ke model TensorFlow Lite saat menjalankan inferensi atau prediksi. Data masukan ini dapat berupa berbagai jenis data, tergantung pada jenis model dan tugas yang ingin diselesaikan.
4. Interpreter adalah komponen yang bertanggung jawab untuk memuat dan menjalankan model yang telah diubah formatnya ke dalam format TensorFlow Lite (.tflite). Interpreter berperan sebagai jembatan antara model TensorFlow Lite dan aplikasi yang digunakan untuk menjalankan inferensi atau prediksi.
5. Postprocessing Hasil: Hasil prediksi yang diterima dari model TensorFlow Lite mungkin perlu dipostproses sesuai dengan kebutuhan aplikasi. Misalnya, Anda dapat mengambil kelas dengan probabilitas tertinggi atau melakukan tindakan berdasarkan hasil prediksi.
6. Output dari TensorFlow Lite adalah hasil dari proses inferensi atau prediksi yang dilakukan oleh model yang telah dijalankan melalui interpreter TensorFlow Lite. Output ini adalah hasil dari pemrosesan data masukan yang diberikan ke model dan mencerminkan hasil prediksi atau informasi yang dihasilkan oleh model.[12].

2.7 Java



Gambar 2.10 Logo Java

Java adalah bahasa pemrograman yang dikembangkan oleh Sun Microsystems pada tahun 1995. Java awalnya dikembangkan untuk membuat aplikasi untuk consumer electronics, namun kini digunakan secara luas dalam pengembangan aplikasi perangkat lunak dan sistem informasi. Java dirancang agar mudah dipahami oleh manusia, portabel, dan memiliki kinerja yang tinggi. Java juga memiliki banyak library dan framework yang memudahkan pengembangan aplikasi.

Menurut penelitian yang dilakukan oleh Mulyawan dan Putri (2020), penggunaan bahasa pemrograman Java sangat luas dalam pengembangan perangkat lunak. Hal ini disebabkan oleh kemampuan Java yang dapat berjalan di berbagai platform, serta adanya banyak framework dan library yang mendukung pengembangan aplikasi. Selain itu, penggunaan bahasa pemrograman Java juga dianggap lebih aman dan dapat menghindari terjadinya kerentanan keamanan.

Java juga memiliki dukungan yang kuat dari perusahaan-perusahaan besar, seperti Oracle dan IBM. Perusahaan-perusahaan tersebut memiliki kontribusi dalam pengembangan bahasa pemrograman Java dan mendukung penggunaan Java dalam pengembangan aplikasi enterprise. Selain itu, banyak universitas

dan lembaga pendidikan juga mengajarkan bahasa pemrograman Java sebagai bahasa pemrograman dasar.

Java juga mendukung pengembangan aplikasi yang berorientasi objek (object-oriented programming). Hal ini memungkinkan pengembang untuk membuat aplikasi yang lebih mudah dipahami dan dapat di-maintain. Selain itu, Java juga mendukung pengembangan aplikasi yang berbasis web dengan framework seperti Spring dan Struts[13].

2.8 Kamera

Kamera pada smartphone adalah salah satu fitur utama yang paling banyak digunakan oleh pengguna smartphone. Dalam beberapa tahun terakhir, kamera smartphone telah mengalami perkembangan yang signifikan, mulai dari resolusi yang lebih tinggi hingga kualitas gambar yang lebih baik. Kamera pada smartphone juga memiliki kemampuan untuk mengambil gambar dalam kondisi cahaya yang minim, dengan menggunakan teknologi seperti OIS (Optical Image Stabilization) dan EIS (Electronic Image Stabilization) untuk menghasilkan gambar yang lebih stabil.

Saat ini, saat ini memiliki beberapa fitur seperti fitur pengenalan wajah, deteksi objek, dan fitur pengambilan gambar dalam mode malam hari. Fitur pengenalan wajah dan deteksi objek memungkinkan pengguna untuk mengambil gambar dengan mudah dan cepat. Dengan fitur pengambilan gambar dalam mode malam hari, kamera smartphone dapat menghasilkan gambar yang lebih terang dan detail dalam kondisi cahaya yang minim. Selain itu, beberapa smartphone juga memiliki kemampuan untuk merekam video dengan resolusi 4K dan bahkan 8K.

Selain itu, kamera pada smartphone juga dapat digunakan untuk membuat foto panorama, foto HDR (High Dynamic Range), dan bahkan foto 360 derajat. Kamera smartphone saat ini juga memiliki beberapa fitur editing foto yang memungkinkan pengguna untuk mengedit foto langsung dari perangkat mereka, seperti memotong, mengubah warna, dan menambahkan filter. Kamera pada smartphone juga dapat digunakan untuk merekam video dengan efek slow motion dan time-lapse[14].

2.9 Aplikasi

Aplikasi atau aplikasi perangkat lunak adalah program komputer yang dirancang untuk melakukan tugas tertentu pada perangkat elektronik, seperti smartphone, tablet, atau komputer. Aplikasi ini dapat digunakan untuk berbagai tujuan, seperti hiburan, belanja online, kesehatan, atau komunikasi.

Menurut Agarwal. (2020), aplikasi dibagi menjadi dua jenis, yaitu aplikasi web dan aplikasi seluler. Aplikasi web dirancang untuk diakses melalui browser web pada perangkat elektronik, sedangkan aplikasi seluler dirancang khusus untuk diunduh dan diinstal pada perangkat seluler seperti smartphone atau tablet.

Aplikasi memiliki antarmuka pengguna yang memudahkan pengguna untuk berinteraksi dengan program. Aplikasi memiliki fungsi yang spesifik dan biasanya dirancang untuk menyelesaikan tugas tertentu. Aplikasi dapat diunduh dan diinstal pada perangkat elektronik[15].

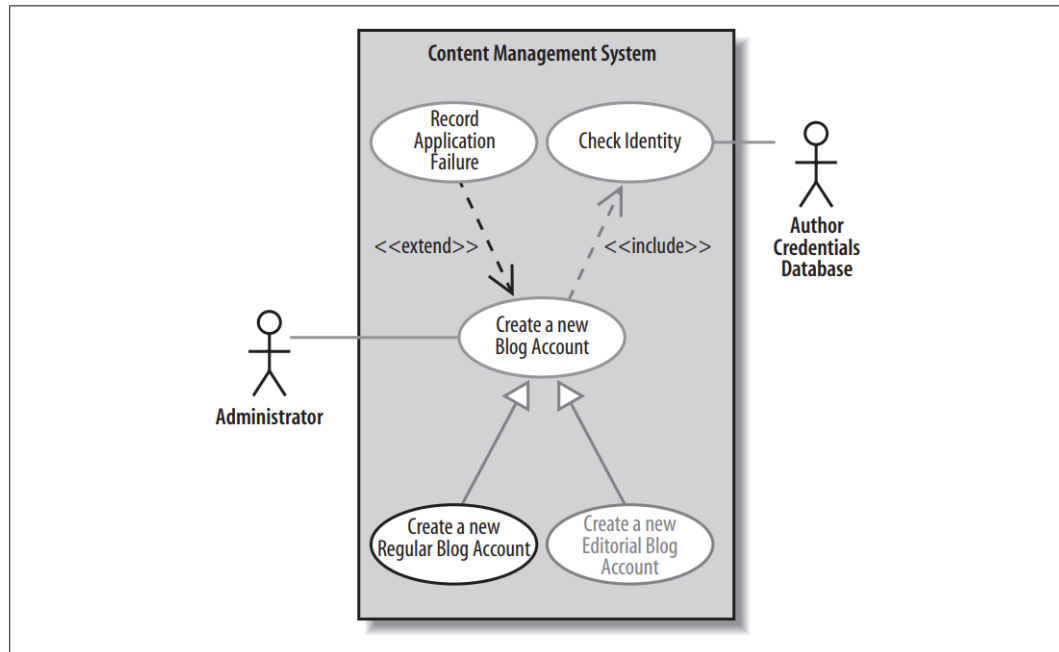
2.10 UML (*Unifies Modeling Language*)

UML adalah sebuah bahasa visual yang digunakan untuk memodelkan dan mengkomunikasikan sistem secara visual menggunakan diagram dan teks pendukung. UML memiliki beberapa jenis pemodelan, seperti diagram use case, diagram class, diagram aktivitas, dan diagram urutan[16].

2.10.1.1 Use Case Diagram

Use case diagram adalah sebuah pemodelan yang digunakan untuk menggambarkan perilaku suatu sistem informasi yang akan dibuat. Tujuan dari use case diagram adalah untuk mendeskripsikan fungsi-fungsi yang ada dalam sistem dari perspektif pengguna. Use case diagram bekerja dengan menggambarkan interaksi tipikal yang akan dilakukan oleh pengguna dalam menggunakan sistem melalui sebuah cerita yang menjelaskan bagaimana sistem dapat digunakan oleh

pengguna. Contoh dari penggunaan use case diagram dapat dilihat pada gambar 2.11 :



Gambar 2.11 Contoh Use Case Diagram

Untuk membuat sebuah use case diagram, terdapat beberapa simbol / notasi yang digunakan, yaitu:

1. Actor

Actor adalah peran yang dimainkan oleh individu, sistem, atau perangkat yang berperan dalam operasional sistem. Actor menunjukkan siapa yang menggunakan atau berinteraksi dengan sistem dan bagaimana mereka terlibat dalam menjalankan use case. Dalam diagram use case, actor digambarkan dengan simbol "stick man".

2. Use Case

Use case digunakan untuk mengilustrasikan tindakan atau aktivitas yang akan dilakukan oleh aktor dalam sistem untuk mencapai tujuan tertentu. Use case dapat diidentifikasi berdasarkan kebutuhan pengguna. Simbol yang digunakan untuk menggambarkan use case adalah oval dengan nama di dalamnya, yang mewakili aktivitas yang dilakukan oleh aktor.

3. Garis Komunikasi (Communication Line).

Communication lines dalam use case diagram adalah garis yang menghubungkan aktor dengan use case untuk menunjukkan interaksi atau pesan yang ditukar antara keduanya.

4. Batasan Sistem (Sistem Boundaries)

Batasan sistem adalah sebuah simbol berbentuk kotak digunakan untuk menunjukkan batas antara use case (yang berasal dari dalam sistem) dengan aktor (yang berasal dari luar sistem). Untuk menggambar batas sistem pada use case, buatlah sebuah kotak yang melingkupi use case dan biarkan aktor berada di luar kotak. Kemudian, berilah nama yang cocok untuk batas sistem.

5. Deskripsi Use Case (Use Case Description)

Deskripsi Use Case adalah sebuah tabel yang digunakan untuk menjelaskan use case dengan lebih jelas. Tabel ini berisi teks yang menguraikan langkah – langkah yang terjadi pada tiap use case dan informasi lain yang penting.

6. Relasi Use Case (Use Case Relationship)

Use case relationship menunjukkan bagaimana use case berinteraksi dengan use case lain. Dengan menggunakan use case relationship, kita dapat menghindari pengulangan pada use case description saat use case digunakan dalam situasi yang berbeda. Selain itu, use case relationship juga membantu menyederhanakan use case yang memiliki banyak alur dalam prosesnya.

Terdapat dua jenis use case yaitu:

1. <<include>>Relationship

Relasi <<include>> antar use case digunakan untuk menyatukan fungsionalitas use case yang sama pada beberapa skenario yang berbeda. Dengan menggunakan relasi <<include>> ini, kita dapat menghindari pengulangan fungsionalitas pada use case yang berbeda dan membuat use case lebih jelas dan mudah dikembangkan.

2. <<extends>>Relationship

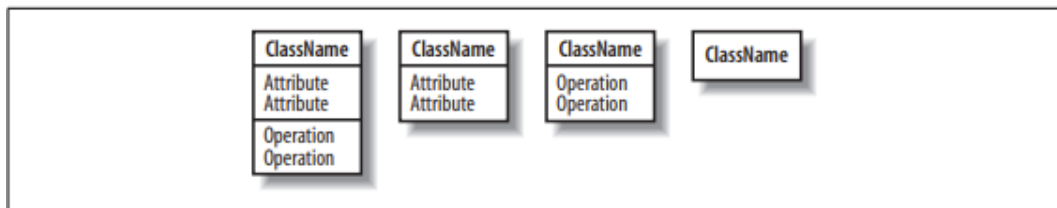
Relasi <<extends>> antar use case juga digunakan untuk mengulang use case pada skenario yang berbeda, tetapi pengulangan ini tidak

wajib (bisa terjadi atau tidak). Relasi <<extends>> ini mirip dengan relasi <<include>>, tetapi bersifat opsional.

2.10.1.2 Class Diagram

Diagram kelas, atau yang dikenal juga sebagai class diagram, adalah sebuah pemodelan yang mengilustrasikan struktur dari setiap kelas yang diperlukan dalam sistem. Dengan menggunakan diagram kelas, kita dapat memperoleh gambaran tentang hubungan antara kelas-kelas dalam sistem yang sedang dibangun, serta menetapkan aturan dan tanggung jawab dari setiap kelas tersebut. Contoh dari kelas diagram dapat dilihat pada gambar berikut:

Class diagram dapat diilustrasikan dengan persegi panjang yang terdiri dari 3 komponen. Komponen pertama berisi nama kelas, komponen kedua berisi atribut kelas dan komponen ketiga berisi operasi pada kelas.



Gambar 2.12 Bagian bagian class diagram

1. Nama Kelas

Nama kelas merupakan tipe untuk objek pada sebuah kelas

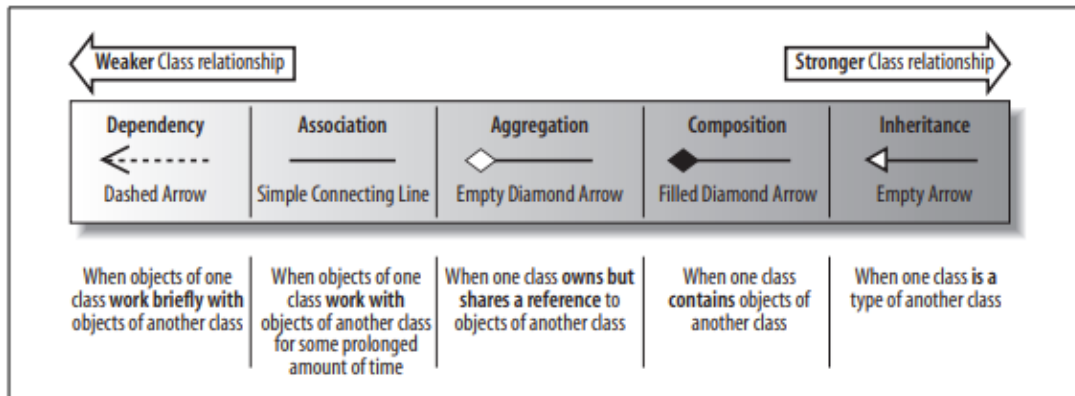
2. Atribut

Atribut merepresentasikan potongan informasi yang mewakili keadaan objek dalam sebuah kelas.

3. Operasi

Operasi mengilustrasikan apa saja yang dapat dilakukan oleh kelas tetapi tidak terlalu menggambarkan bagaimana cara untuk melakukannya.

Selain itu, untuk menunjukkan jenis hubungan yang berbeda antara kelas-kelas, digunakan panah yang menggambarkan dependency, association, aggregation, composition dan inheritance.



Gambar 2.13 Hubungan antar kelas pada class diagram

1. Dependency

Dependency antara dua kelas digunakan untuk menggambarkan sebuah kelas dapat menggunakan objek dari kelas lain.

2. Association

Association antar kelas digunakan untuk menunjukkan bahwa sebuah kelas memiliki atribut yang merujuk pada objek atau objek – objek dari kelas lain.

3. Aggregation

Aggregation antar kelas digunakan untuk menunjukkan bahwa sebuah kelas memiliki tapi dapat membagi objek dari kelas lain.

4. Composition

Composition antar kelas digunakan untuk menunjukkan bahwa sebuah objek yang termasuk dalam sebuah kelas tidak bisa eksis tanpa kelas tersebut.

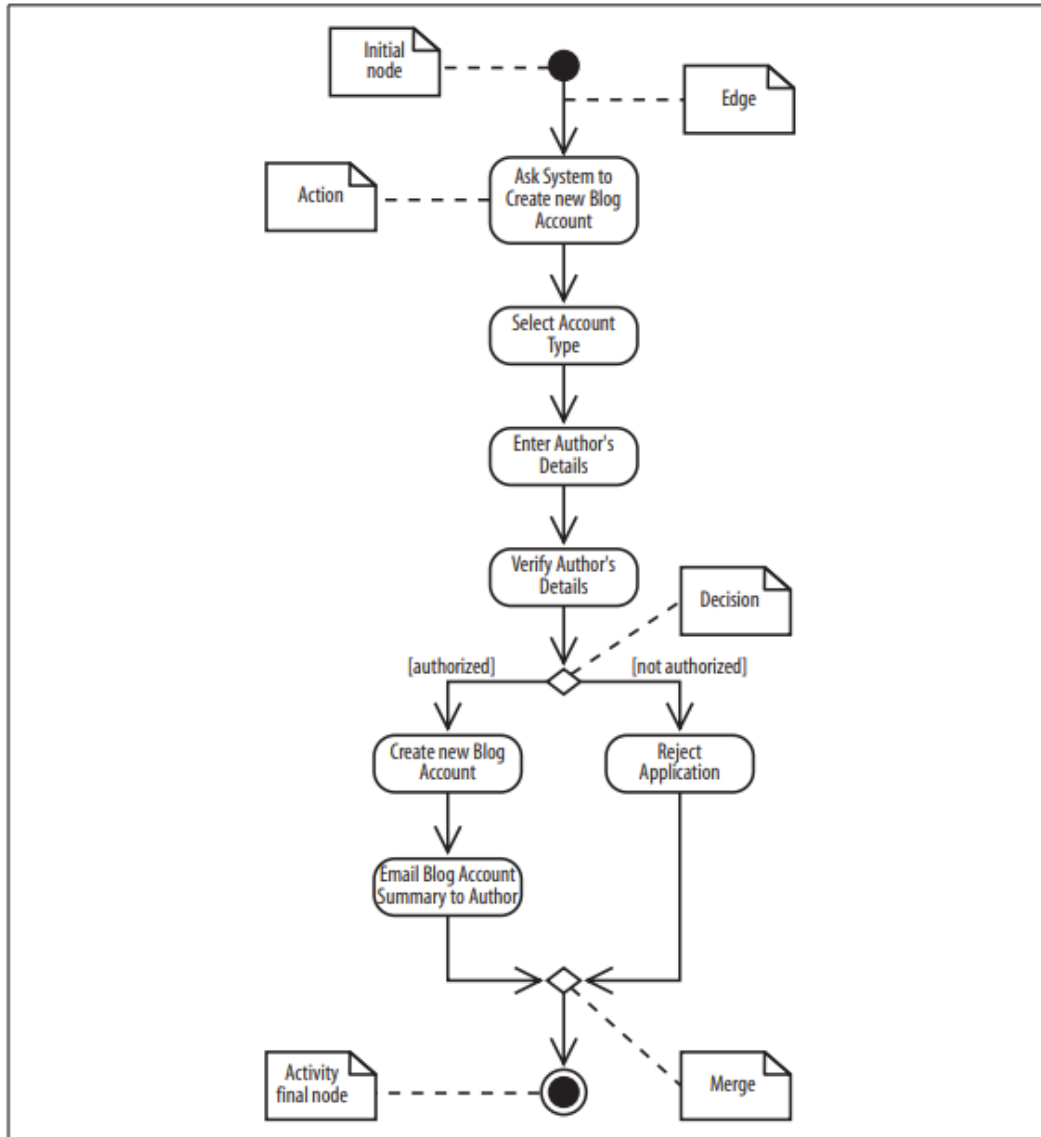
5. Inheritance

Inheritance antar kelas digunakan untuk menunjukkan bahwa sebuah kelas yang merupakan subkelas dari kelas lain.

2.10.1.3 Activity Diagram

Diagram aktivitas, atau activity diagram, adalah suatu model yang digunakan untuk menggambarkan alur kerja atau alur kerja dari suatu sistem yang mencakup aktivitas, proses bisnis, atau menu yang ada dalam perangkat lunak. Dengan menggunakan activity diagram, kita dapat memahami dengan mudah aliran kerja

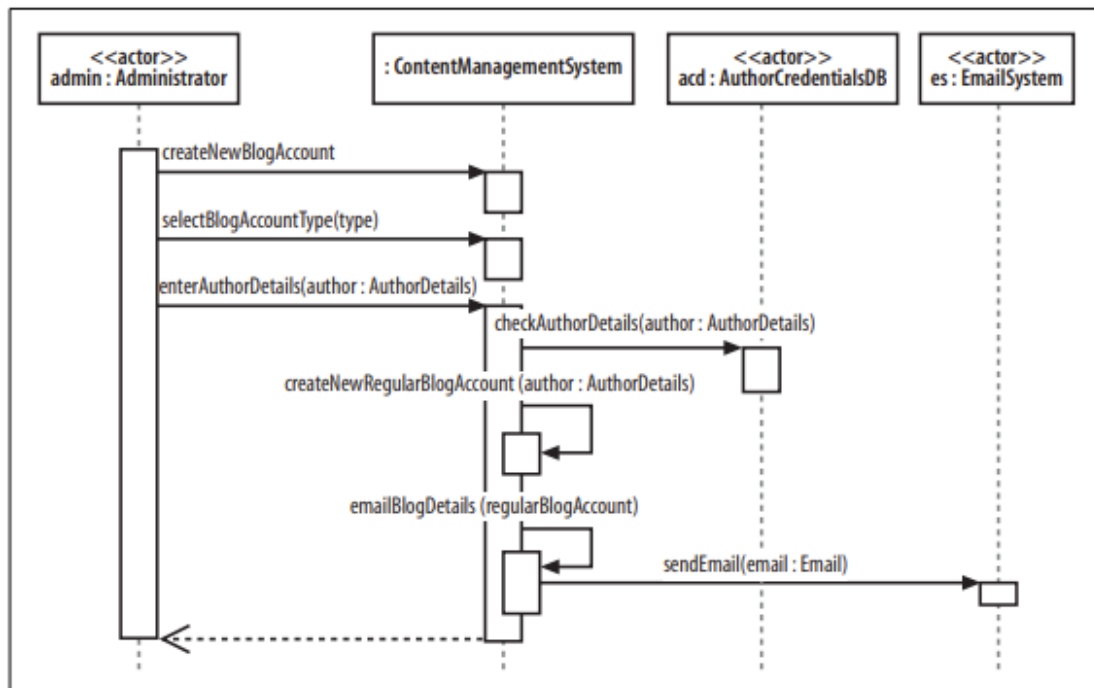
dalam suatu sistem perangkat lunak dan bagaimana berbagai aktivitas saling terkait. Contoh dari activity diagram dapat dilihat pada gambar 2.14.



Gambar 2.14 Contoh Activity Diagram

2.10.1.4 Sequence Diagram

Diagram urutan, atau sequence diagram, adalah sebuah model yang menggambarkan interaksi antara objek-objek di dalam dan sekitar sebuah sistem. Diagram urutan ini berfokus pada interaksi antara pengguna dan sistem, termasuk tampilan, pesan, dan gambar yang ada dalam sistem yang digambarkan berdasarkan urutan waktu. Contoh Sequence diagram dapat dilihat pada gambar berikut.



Gambar 2.15 Contoh Sequence Diagram