

BAB 2

TINJAUAN PUSTAKA

2.1 Pengenalan *Cyberbullying*

Cyberbullying merupakan tindakan yang dilakukan oleh seseorang atau sekelompok orang terhadap seseorang melalui text, gambar, atau video yang cenderung merendahkan dan melecehkan [4]. *Cyberbullying* dapat dilakukan oleh pelaku secara *anonymous* atau tanpa nama, hal ini menyebabkan banyak kasus *cyberbullying* yang tidak dapat dilacak sumbernya. Media yang digunakan oleh pelaku *cyberbullying* pun terus berkembang searah dengan perkembangan teknologi informasi, mulai dari *Email* hingga *Online Games*.

Beberapa tipe *Cyberbullying*, antara lain :

1. *Flaming* merupakan tindakan mengirim pesan dengan nada kasar dan vulgar tentang seseorang kepada sebuah kelompok *online*.
2. *Online harrasment* merupakan tindakan mengirim pesan yang melecehkan secara berulang kali melalui media pesan teks
3. *Cyberstalking* merupakan tindakan mengancam , melukai, dan mengintimidasi yang dilakukan secara berlebihan dengan mengikuti gerak – gerak korban dalam dunia maya.
4. *Denigration* merupakan tindakan yang bertujuan untuk merusak reputasi korban dengan mengirim pesan yang berisi tuduhan terhadap korban.
5. *Outing* merupakan tindakan menyebarkan informasi pribadi sensitif sang korban ke dunia maya.
6. *Exclusion* merupakan tindakan mengeluarkan atau menyingkirkan seseorang dari kelompok *online* secara kasar.

Secara umum korban *cyberbullying* mengalami penderitaan yakni depresi, harga diri yang rendah, gelisah, konsentrasi berkurang, melukai diri sendiri, hingga memiliki ide untuk bunuh diri [5]. Keadaan tersebut juga diperburuk

dengan sulitnya orang tua sang korban untuk mengetahui apakah anaknya adalah korban *cyberbullying*.

2.2 Media Sosial

2.2.1 Definisi Media Sosial

Media sosial terdiri dari dua kata yakni media yang berarti alat, sarana, perantara, atau penghubung dan sosial yang berarti berkenaan dengan masyarakat. Menurut Van Dijk dalam [6], media sosial merupakan fasilitator *online* yang menguatkan hubungan antar pengguna. Menurut Boyd dalam [6], media sosial adalah kumpulan perangkat lunak yang memungkinkan individu maupun komunitas untuk berkumpul, berbagi, dan berkomunikasi. Media sosial memiliki kekuatan pada *user-generated content* dimana konten dihasilkan oleh pengguna, bukan oleh editor sebagaimana di media massa.

2.2.2 Karakteristik Media Sosial

Menurut Rulli dalam [6], media sosial memiliki beberapa karakteristik, antara lain :

1. Jaringan, sesuai dengan terminologi dalam bidang teknologi yang berarti infrastruktur yang menghubungkan komputer maupun perangkat keras lainnya.
2. Informasi, merupakan hal terpenting pada media sosial. Dalam media sosial, pengguna mengkreasikan representasi identitas, membuat konten, dan melakukan interaksi berdasarkan informasi. Informasi diproduksi, dipertukarkan dan dikonsumsi oleh setiap individu dalam media sosial
3. Arsip, menjelaskan bahwa informasi telah tersimpan dan dapat diakses kapan saja melalui perangkat apapun. Informasi yang telah diunggah di media sosial tidak hilang begitu saja saat hari, bulan, ataupun tahun berganti.
4. Interaksi, merupakan karakteristik dasar dari media sosial dimana terbentuknya jaringan antar pengguna.

5. Simulasi Sosial, saat pengguna berinteraksi dengan sesama melalui antarmuka di media sosial, pengguna menjalani simulasi dari interaksi sosial pada dunia nyata
6. Konten oleh pengguna atau *user-generated content* merupakan karakter media sosial yang menunjukkan konten dalam media sosial sepenuhnya milik dan berdasarkan kontribusi pengguna.

2.2.3 Aplikasi – aplikasi media sosial

Saat ini, media sosial sudah menjadi bagian yang penting dalam interaksi sosial. Hal ini juga diperkuat dengan banyaknya aplikasi – aplikasi media sosial yang muncul. Berikut adalah media sosial yang populer dikalangan masyarakat khususnya anak – anak dan remaja :

1. Facebook

Facebook dibuat pada tahun 2004 oleh Mark Zuckerberg dan Eduardo Saverin, saat ini Facebook berhasil menjadi media sosial terpopuler dengan 2.9 miliar pengguna aktif setiap bulannya di seluruh dunia.

Fitur – fitur utama facebook diantaranya :

- a. *Feed*, merupakan halaman utama saat pengguna masuk ke Facebook. Halaman ini berisi konten – konten yang diunggah oleh teman , grup, atau halaman yang diikuti oleh pengguna
- b. *Chat*, melakukan pesan langsung dengan teman pengguna dalam Facebook
- c. *Facebook Live*, fitur untuk berbagi secara momen secara langsung. Penonton juga dapat berinteraksi langsung dengan pengguna yang melakukan siaran melalui kolom mentions yang tersedia.
- d. Foto dan Album, pengguna dapat membagikan beberapa foto dan bahkan membuat album foto pribadi.

2. Instagram

Instagram adalah media sosial populer yang didirikan pada tahun 2010 oleh Kevin Systrom dan Michel Krieger. Saat ini Instagram telah

menjadi media sosial berbagi foto terbesar dengan 2 miliar pengguna aktif setiap bulannya di seluruh dunia

Berikut adalah fitur – fitur utama dari Instagram :

- a. *Home Page*, merupakan halaman utama saat pengguna masuk ke Instagram. Halaman ini berisi konten – konten yang diunggah oleh teman, brand, ataupun artis yang pengguna ikuti
- b. *Explore*, merupakan halaman sugesti konten berdasarkan apa yang disukai pengguna. Dalam halaman ini pengguna pun dapat mencari sebuah konten dengan menuliskan *keyword* pada *search bar*
- c. *Comments*, pengguna dapat bermentions pada foto atau video yang diunggah oleh pengguna lain.
- d. *Direct Message*, pengguna dapat menuliskan pesan secara langsung kepada akun yang telah pengguna *follow*.

3. WhatsApp

WhatsApp adalah media sosial populer yang banyak digunakan karena *Instant Messaging*-nya. WhatsApp didirikan pada Januari 2009 oleh Brian Acton dan Jan Kourm dan menjadi bagian dari Facebook pada tahun 2014.

Berikut adalah fitur – fitur utama dari Instagram :

- a. *Chats*, fungsi utama WhatsApp adalah untuk melakukan komunikasi secara langsung. Pengguna dapat melakukan percakapan secara langsung dengan orang baik yang ada dalam kontak maupun tidak.
- b. *Status*, pengguna dapat berbagi momen berupa foto atau video dan dipasang sebagai status agar terlihat oleh pengguna lain
- c. *Calls*, pengguna melakukan panggilan suara maupun video dengan pengguna lain. WhatsApp juga dapat melakukan

panggilan suara maupun video secara masal dalam sebuah grup.

4. Twitter

Twitter adalah media sosial yang didirikan pada maret 2006 oleh Jack Dorsey dan rekan – rekannya. Twitter memiliki konsep dimana pengguna dapat mengirimkan pesan singkat sepanjang 280 yang dapat dilihat oleh pengguna lain.

Berikut adalah fitur – fitur utama dari Twitter :

- a. *Home*, halaman ini berisi konten – konten yang diunggah oleh teman maupun rekomendasi dari Twitter.
- b. *Tweet*, fungsi utama Twitter adalah untuk mengirimkan pesan singkat yang dapat dilihat, dimentionasi, disukai oleh pengguna lain.
- c. *Direct Message*, pengguna dapat menuliskan pesan secara langsung terhadap pengguna lain
- d. *Trends*, pengguna dapat melihat apa yang sedang ramai dibicarakan oleh pengguna lain.

2.3 Android

Android adalah sistem operasi terbuka yang menggunakan kernel sistem operasi Linux. Android berawal pada tahun 2003 sebagai sistem operasi untuk kamera digital [7]. Pada tahun 2005, Google memutuskan untuk membeli Android dan pada tahun 2007 Google memulai proyek untuk membangun sistem operasi Android. Pada tahun 2008, HTC dirilis sebagai smartphone pertama yang menggunakan sistem operasi Android. Tidak lama dari dirilisnya HTC, dunia memiliki 400 manufaktur smartphone dengan sistem operasi Android dan 1.5 juta aktivasi Android setiap harinya [8]. Hingga akhir 2021 terdapat beberapa versi Android yang tersedia, diantaranya sebagai berikut:

Tabel 2.1 Tabel Versi Android

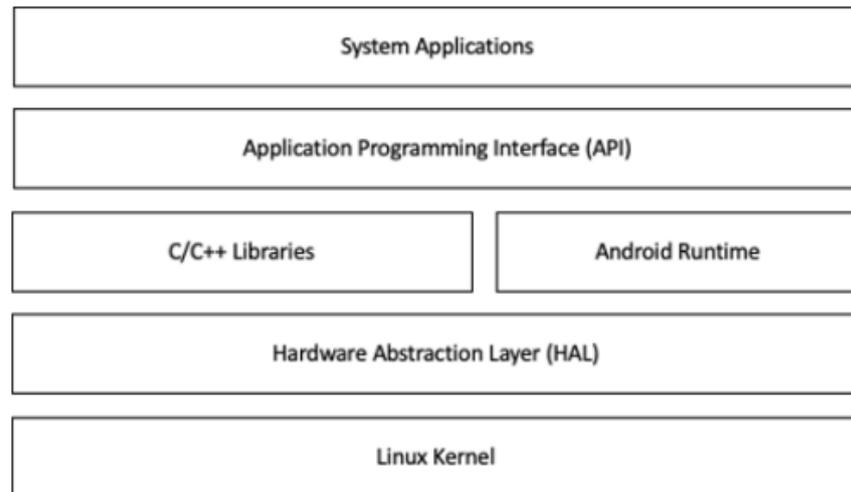
Nama kode	Versi	API
Android 12L	12.1	API level 32

Android 12	12	API level 31
Android 11	11	API level 30
Android 10	10	API level 29
Pie	9	API level 28
Oreo	8.1	API level 27
Oreo	8.0	API level 26
Nougat	7.1	API level 25
Nougat	7.0	API level 24
Marshmallow	6.0	API level 23
Lollipop	5.1	API level 22
Lollipop	5.0	API level 21
KitKat	4.4 – 4.4.4	API level 19
Jelly Bean	4.3.x	API level 18
Jelly Bean	4.2.x	API level 17
Jelly Bean	4.1.x	API level 16
Ice Cream Sandwich	4.0.3 – 4.0.4	API level 15
Ice Cream Sandwich	4.0.1 – 4.0.2	API level 14
Honeycomb	3.2.x	API level 13
Honeycomb	3.1	API level 12
Honeycomb	3.0	API level 11
Gingerbread	2.3.3 – 2.3.7	API level 10
Gingerbread	2.3 – 2.3.2	API level 9
Froyo	2.2.x	API level 8
Éclair	2.1	API level 7
Éclair	2.0.1	API level 6
Ecliar	2.0	API level 5
Donut	1.6	API level 4
Cupcake	1.5	API level 3
(tidak ada kode nama)	1.1	API level 2
(tidak ada kode nama)	1.0	API level 1

2.3.1 Arsitektur Android

Sistem operasi adalah program yang besar dan mengatur berjalannya sebuah perangkat. Hampir seluruh sistem operasi dibangun dengan bentuk lapisan,

dimulai dari lapisan terluar yang dapat muncul di depan muka pengguna hingga lapisan terdalam yang tidak dapat dilihat oleh pengguna. Penting untuk seorang pengembang aplikasi Android untuk mengetahui arsitektur sistem operasi Android.



Gambar 2.1 Arsitektur Sistem Operasi Android

Dapat dilihat pada Gambar 2.1 Arsitektur Sistem Operasi Android bahwa Android dibangun dengan dasar kernel linux. Berikut adalah penjelasan setiap lapisan arsitektur android [8] :

1. System Applications

Lapisan teratas ini adalah tempat aplikasi – aplikasi seperti kontak, kamera, permainan, dan lainnya. Pengembang dan pengguna berinteraksi pada lapisan ini. Pengembang menulis kode untuk dijalankan pada lapisan ini dan pengguna melihat tampilan aplikasi pada layar.

2. Application Programming Interface (API)

Lapisan ini adalah tempat aplikasi – aplikasi melakukan *permintaan* terhadap kode – kode pada lapisan API. Lapisan ini adalah yang dibutuhkan oleh pengembang untuk membuat sebuah aplikasi Android.

3. C/C++ Libraries dan Android Runtime

Lapisan ini terdiri dari Libraries yang dapat mendukung lapisan API seperti Open GL untuk proses grafis. Pada lapisan ini juga terdapat Android Runtime yang berfungsi untuk melakukan *compiling* kode dari aplikasi saat pertama dijalankan, hal ini dilakukan supaya mulainya aplikasi tidak memakan waktu yang lama.

4. Hardware Abstraction Layer (HAL)

Lapisan ini adalah lapisan yang diperlukan oleh Android Runtime. Android Runtime tidak dibuat untuk segala macam perangkat, oleh karena itu lapisan ini bertindak sebagai penerjemah agar Android Runtime dapat berjalan pada perangkat tertentu.

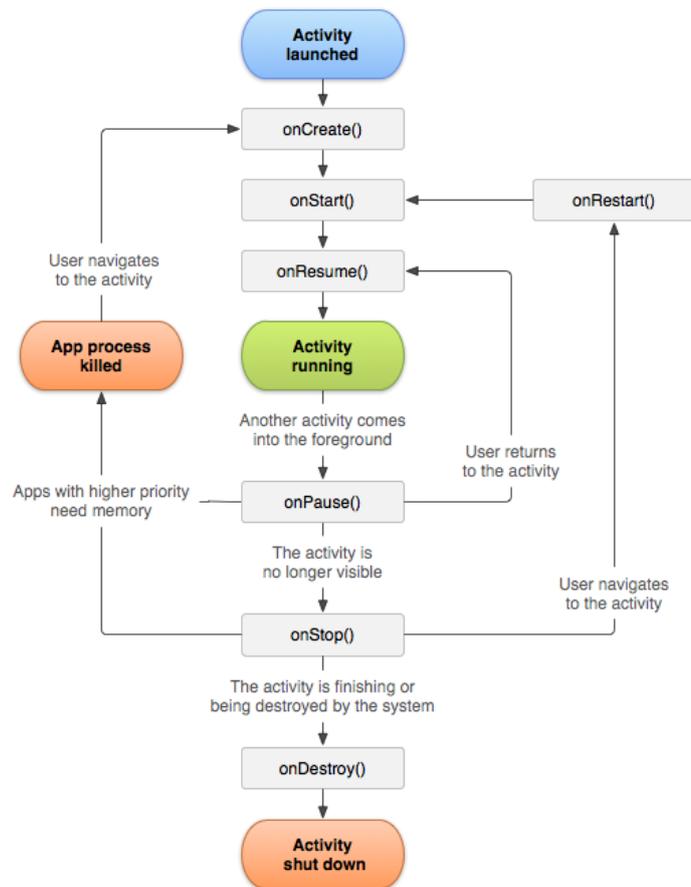
5. Linux Kernel

Lapisan adalah fondasi dari Android, kernel berperan untuk mengatasi seluruh komunikasi dalam proses yang berjalan.

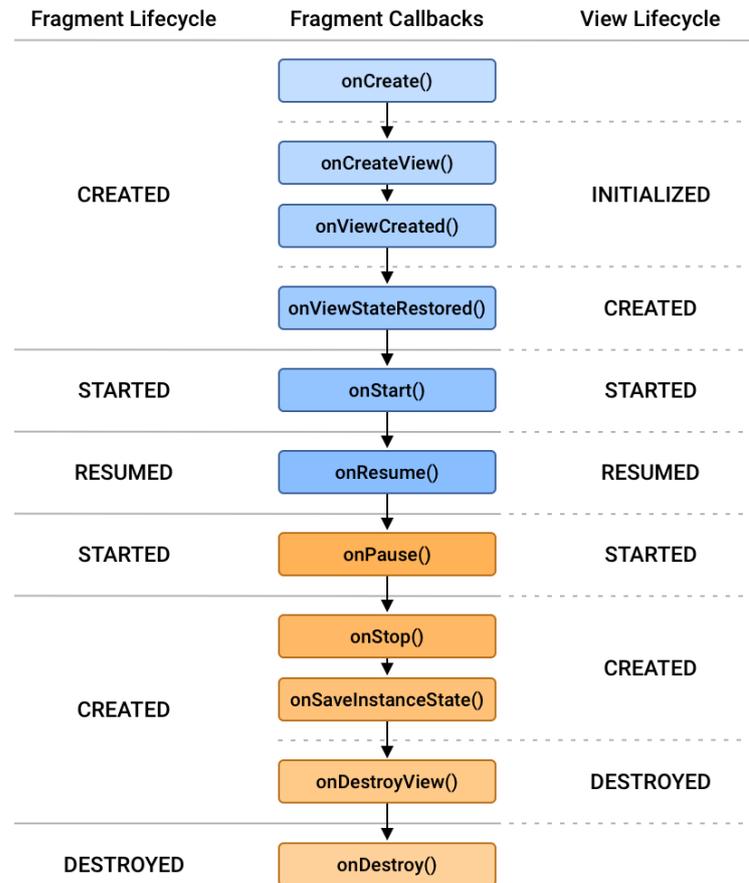
2.4 Android Life-Cycle

Kata Android Life-Cycle sering didengar oleh para pengembang aplikasi android karena berkaitan dengan hal yang penting dalam setiap pengembangan aplikasi Android yakni Activity dan Fragment. Life-cycle adalah keadaan saat itu untuk sebuah komponen dan cakupan dari keadaan tersebut selama komponen tersebut digunakan [8]. Sederhananya, Life-cycle adalah tahapan yang dilalui oleh Activity dan Fragment mulai dari hidup, mati, hingga dihidupkan lagi.

Seluruh Activity dan Fragment memiliki Life-Cycle yang bertujuan untuk mencegah penggunaan sumber daya perangkat yang berlebihan. Hal ini disebabkan karena Activity dan Fragment membutuhkan sumber daya perangkat yang cukup besar untuk dijalankan. Jika tidak dimatikan, maka akan terus memakan sumber daya perangkat hingga perangkat tidak sanggup menjalankan aplikasi tersebut.



Gambar 2.2 Life-cycle sebuah Activity



Gambar 2.3 Life-cycle sebuah Fragment

Ilustrasi pada Gambar 2.2 dan Gambar 2.3 memperlihatkan bagaimana tahapan life-cycle yang dilalui oleh Activity dan Fragment saat pengembangan ataupun berjalannya aplikasi. Tahap onCreate(), onStart(), onResume(), onPause(), onStop(), dan onDestroy() adalah saat activity atau fragment memasuki keadaan yang baru. Hal ini memungkinkan pengembang untuk menyisipkan kode program yang akan berjalan selama Life-cycle.

2.5 Android Studio

Android Studio adalah *Integrated Development Environment* (IDE) yang diperkenalkan oleh Google pada tahun 2003. Android Studio dibuat berdasarkan pada IntelliJ IDEA yang telah dirapikan sehingga fitur – fitur yang tidak pernah digunakan oleh pengembang aplikasi Android dihapus. Android Studio adalah

IDE yang dikhususkan untuk pengembangan Android dan diharapkan menjadi alat yang dapat membantu pengembangan aplikasi Android lebih produktif.

Android Studio dapat digunakan dalam beberapa sistem operasi diantaranya

1. Windows 7, 8, 10 atau 11 (32-bit, 64-bit)
2. MacOS 10.14 atau lebih tinggi
3. Seluruh distribusi Linux 64-bit yang mendukung Gnome, KDE, atau Unity DE

2.6 Java

Java adalah bahasa pemrograman yang umum digunakan. Java memiliki fitur untuk mendukung pemrograman dengan paradigma pemrograman berbasis objek, prosedural, dan fungsional. Pemrograman Java tidak khusus untuk paradigma berorientasi objek, melainkan hanya mendukung paradigma tersebut, pengembang dapat menulis program dengan Java tanpa sedikitpun menggunakan paradigma berorientasi objek.

Versi awal Java dirilis oleh Sun Microsystems pada 1995. Pengembangan bahasa pemrograman Java ini dimulai dari 1991 dengan nama awal Oak. Tidak lama dari rilisnya Java menjadi bahasa pemrograman paling populer karena fitur *Write once, run anywhere* (WORA). Fitur tersebut memungkinkan program yang dibuat dengan Java dapat dijalankan dalam sistem operasi apapun tanpa memerlukan modifikasi pada kode program, baik itu Mac, Windows atau UNIX [9]. Berkat fitur WORA tersebut, Java juga dapat digunakan untuk pengembangan pada platform Android dan menjadi bahasa pemrograman yang populer untuk pengembangan aplikasi Android.

2.7 Object Oriented Programming (OOP)

Paradigma OOP muncul dengan berkembangnya kapasitas komputer dan kompleksitas dari aplikasi. Kompleksitas aplikasi yang meningkat berarti kode program yang ditulis akan menjadi lebih banyak dan besar, sehingga sulit untuk menghindari duplikasi, melakukan debug, bahkan memelihara kode tersebut. Untuk memecahkan dilema ini para pengembang ahli mulai memecah

permasalahan yang besar menjadi beberapa permasalahan kecil. Ide tentang objek pun mulai diperkenalkan oleh Alan Curtis Kay yang dijuluki sebagai bapak dari OOP [9].

OOP memiliki beberapa prinsip utama yakni [10].

1. Kelas dan Objek

Merupakan inti dari OOP. Kelas adalah rancangan atau *template* untuk sebuah objek. Sedangkan Object adalah *instance* dari sebuah Kelas. Setiap objek memiliki keadaan, perilaku, dan identitas.

2. Enkapsulasi

Dalam OOP, data tidak diperbolehkan untuk mengalir dengan bebas kedalam sistem. Oleh karena itu, data dan fungsi di enkapsulasi dalam suatu unit (dalam Kelas). Enkapsulasi memiliki tujuan diantaranya untuk menempatkan batasan sehingga komponen dari object tidak bisa diakses secara langsung dan mengikat data dengan sebuah *method* yang akan bereaksi pada data tersebut.

3. Abstraksi

Tujuan dari abstraksi adalah untuk menunjukkan fitur yang diperlukan dan menyembunyikan detail dari implementasi.

4. Pewarisan

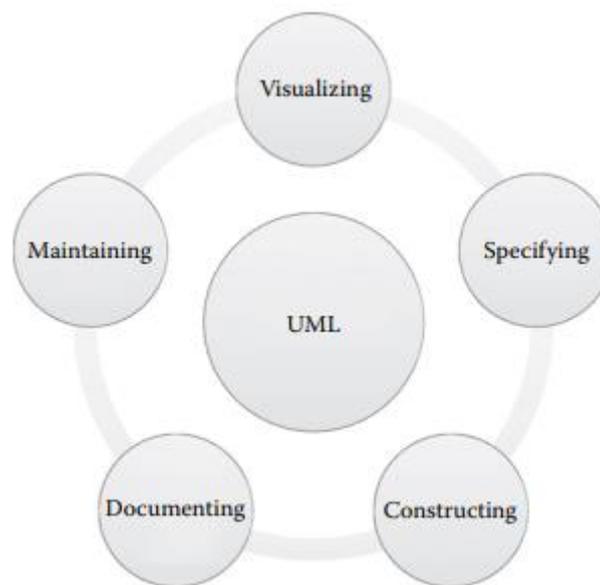
Merupakan proses dimana sebuah objek mendapatkan sebuah ciri dari objek lain. Dalam pemrograman, pewarisan menghasilkan Kelas amal dari sebuah Kelas orang tua. Contoh, kelas Bus adalah warisan dari kelas Kendaraan dan modifikasi pada kelas Bus tidak akan mempengaruhi kelas Kendaraan.

5. Polimorfisme

Polimorfisme biasanya diasosiasikan dengan satu nama tetapi banyak rupa. Dalam OOP, polimorfisme adalah suatu kemampuan objek untuk digunakan dalam keadaan yang berbeda.

2.8 Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah standar bahasa modeling dalam pengembangan software. Hingga saat ini UML adalah salah satu bahasa modeling berorientasi objek yang sering digunakan. UML menyediakan cara pada pengembang untuk menyampaikan informasi desain kelas dan ketergantungannya [11].



Gambar 2.4 Tujuan UML dalam pengembangan

Gambar 2.4 adalah ringkasan dari tujuan UML dalam modeling, pengembangan, dan memelihara sebuah program. Ada 5 cara penggunaan UML dalam pengembangan sebuah software [12] :

1. *Visualizing* (Visualisasi)
Visualisasi adalah tujuan utama dari UML karena notasi dan diagram UML menyediakan mekanisme dengan standar industri untuk menggambarkan kebutuhan, proses, desain solusi, dan arsitektur.
2. *Specifying* (Menentukan)
UML dapat memfasilitasi untuk penentuan alat dalam modeling seperti penentuan aktor, *use case*, kelas, atribut, dan lainnya.
3. *Constructing* (Membangun)

UML digunakan untuk membangun sebuah software karena memungkinkan untuk mengeluarkan kode program (C++, Java) tergantung dari *tools* yang digunakan. Tetapi fitur ini memiliki keterbatasan karena pengembang tetap harus memodifikasi kode yang dihasilkan.

4. *Documenting* (Pencatatan)

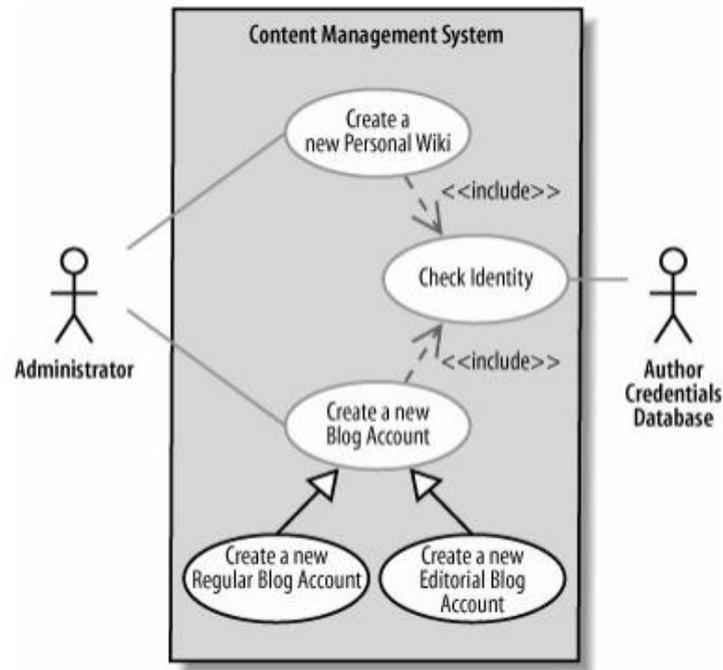
Dengan UML, pencatatan mendetail untuk sebuah kebutuhan, arsitektur, desain, dan rencana proyek, test, dan prototype disediakan untuk meningkatkan spesifikasi dan representasi visual.

5. *Maintaining* (Pemeliharaan)

Model UML yang baik dapat menjadi pertolongan yang handal saat melakukan pemeliharaan dari software. Model UML memungkinkan pengembang untuk melihat keseluruhan sistem dengan mudah.

2.9 Use Case Diagram

Use case adalah serangkaian interaksi aktor terhadap sistem. Interaksi ini dimaksudkan untuk mendeskripsikan apa yang sebuah sistem dapat lakukan tetapi tidak spesifik dalam bagaimana sistem melakukan hal tersebut. Use case tidak hanya digunakan untuk mencatat interaksi antara aktor dan sistem tetapi bisa juga digunakan untuk menjelaskan prakondisi dan kondisi sesudah sebuah use case [12].

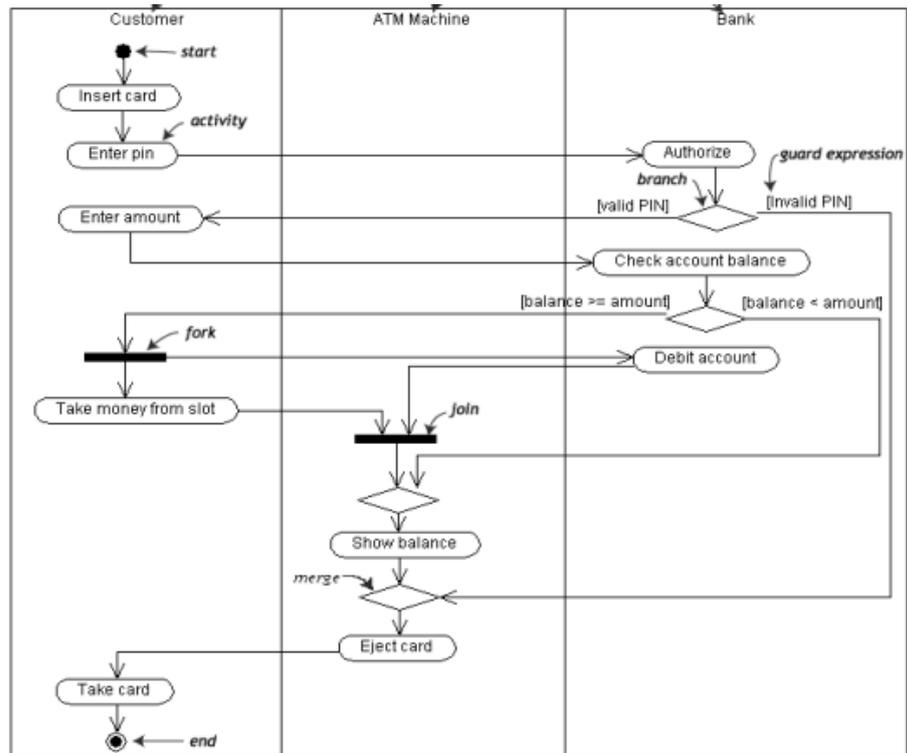


Gambar 2.5 Contoh Use Case Diagram

Gambar 2.5 memperlihatkan dua bagian penting yaitu Aktor sebagai pengguna dan *Use Case* sebagai fungsi dalam sistem. Use case adalah pemodelan dari kebutuhan fungsional suatu sistem sehingga akan menjadi dasar pemodelan selanjutnya. Yaitu *Activity Diagram* dan *Sequence Diagram*.

2.10 Activity Diagram

Jika Use Case memperlihatkan apa yang seharusnya dijalankan oleh sistem, maka Activity Diagram memungkinkan pengembang untuk menjelaskan bagaimana sistem akan menjalankan use case tersebut. Activity Diagram memperlihatkan aktivitas - aktivitas yang saling terikat untuk menggambarkan proses yang sedang terjadi dalam sistem [13].

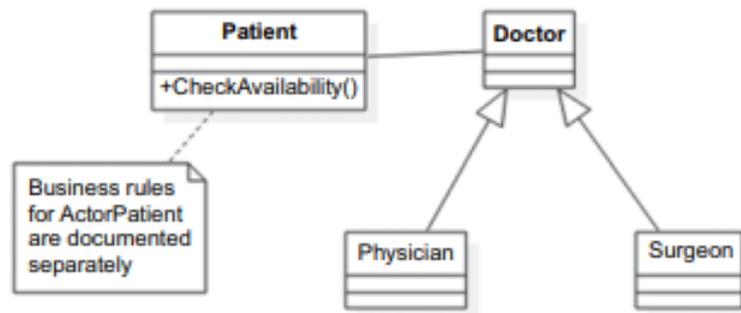


Gambar 2.6 Contoh Activity Diagram

Gambar 2.6 memperlihatkan susunan aktifitas dari suatu *Use Case* saat aktor melakukan penarikan tunai pada sebuah ATM. *Activity Diagram* menggambarkan aktifitas – aktifitas dalam use case secara detail.

2.11 Class Diagram

Class Diagram merupakan diagram yang paling populer dalam pengembangan perangkat lunak. *Class Diagram* menggambarkan *entities* ini baik dalam logika bisnis ataupun teknis serta menggambarkan relasi yang dimiliki oleh kelas tersebut. *Class Diagram* merupakan diagram yang statis. [12].

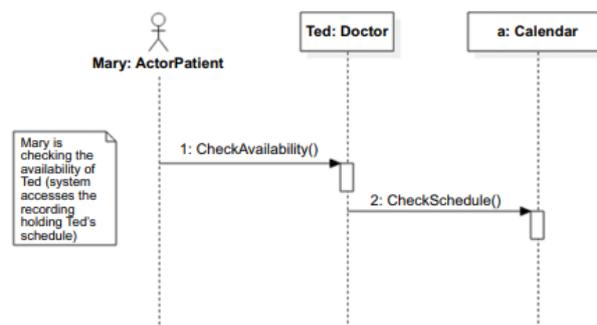


Gambar 2.7 Contoh Class Diagram

Gambar 2.10 memperlihatkan *Class Diagram* sederhana yang menggambarkan kelas *Patient*, *Doctor*, *Surgeon*, dan *Physician*. Relasi dari setiap kelas pun terlihat mulai dari relasi asosiasi antara *Patient* dengan *Doctor* juga relasi *inheritance* *Doctor* dengan *Physician* dan *Surgeon*.

2.12 Sequence Diagram

Sequence Diagram merupakan diagram yang digunakan untuk dokumentasi perilaku dalam sebuah *Use Case*. *Sequence Diagram* juga disebut *Scenario Diagram* karena diagram ini menggambarkan sebuah skenario dalam sebuah *Use Case*. [12].



Gambar 2.8 Contoh Sequence Diagram

Gambar 2.8 memperlihatkan *Sequence Diagram* sederhana yang menggambarkan bagaimana sebuah aktor memeriksa jadwal praktik seorang dokter. Dalam diagram ini juga diperlihatkan, objek *Doctor* harus melalui objek *Calendar* untuk memeriksa jadwal praktiknya.

2.13 *Application Programming Interface (API)*

Application Programming Interface (API) adalah cara bagaimana sistem komputer berinteraksi. Dengan banyaknya sistem yang dibangun API juga banyak bermunculan, baik itu dalam *library* ataupun kode yang ditulis untuk kepentingan sendiri [14]. API menyediakan suatu fitur untuk digunakan oleh perangkat lunak lain.

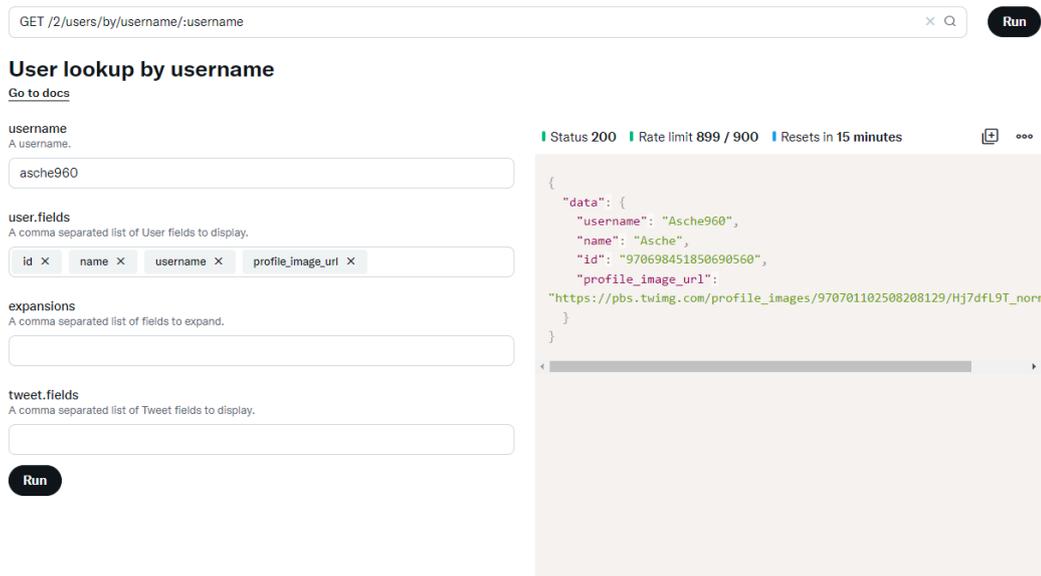
Ditengah banyaknya API, Web API adalah API yang terbuka dan dapat diakses melalui internet juga digunakan oleh berbagai orang. API tersebut dapat diakses dengan menggunakan URL yang telah disediakan oleh penyedia API tersebut. API ini memungkinkan pengguna untuk mengakses, menambah, mengubah, dan menghapus data melalui jaringan internet.

Dengan Web API dan terus berkembangnya internet, pengembangan sebuah perangkat lunak sudah erat hubungannya dengan Web API. Dengan menggunakan Web API, perangkat lunak yang dibangun dapat mengakses, ataupun mengolah data yang berada di luar sistem perangkat lunak yang dibangun.

2.14 *Twitter API*

Twitter memungkinkan setiap penggunanya untuk berinteraksi pesan singkat yang disebut *tweet* dimana pengguna lainnya dapat ikut serta membalas, atau mengulang (*retweet*) pesan tersebut. Twitter sendiri menyediakan API untuk pengembang agar dapat mengakses data yang ada dalam media sosial tersebut. Untuk menggunakan API tersebut, pengembang wajib memiliki Akun Developer Twitter dan melakukan permintaan akses untuk memakai API yang akan digunakan.

Dalam penelitian ini, fitur Twitter API yang digunakan diantaranya adalah *get users by username* dan *user mention timeline*. Twitter juga menyediakan API *Playground* untuk mencoba fitur-fitur API nya.



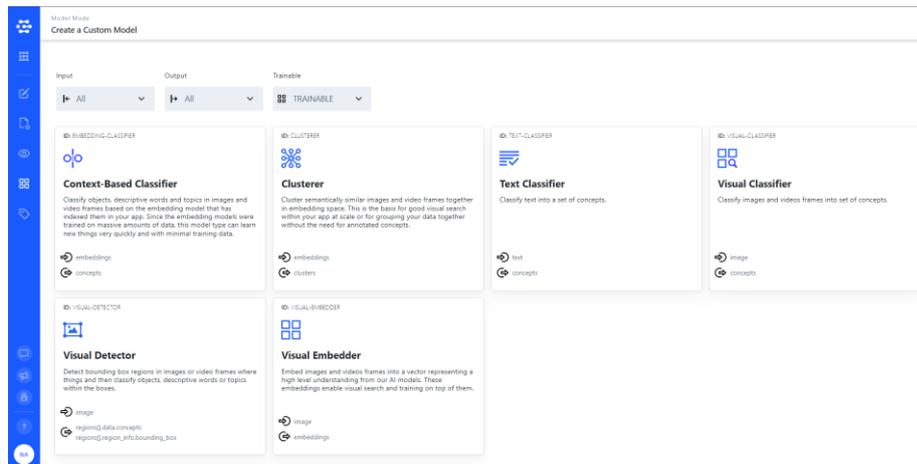
Gambar 2.9 Halaman Twitter API Playground

Gambar 2.9 menunjukkan halaman dimana pengembang dapat mencoba endpoint *User lookup by username* dalam API Playground yang disediakan oleh Twitter. Setiap fitur dapat dilakukan percobaan *request* hingga mendapatkan contoh *response* yang akan didapatkan jika menggunakan fitur API tersebut.

2.15 Clarifai API

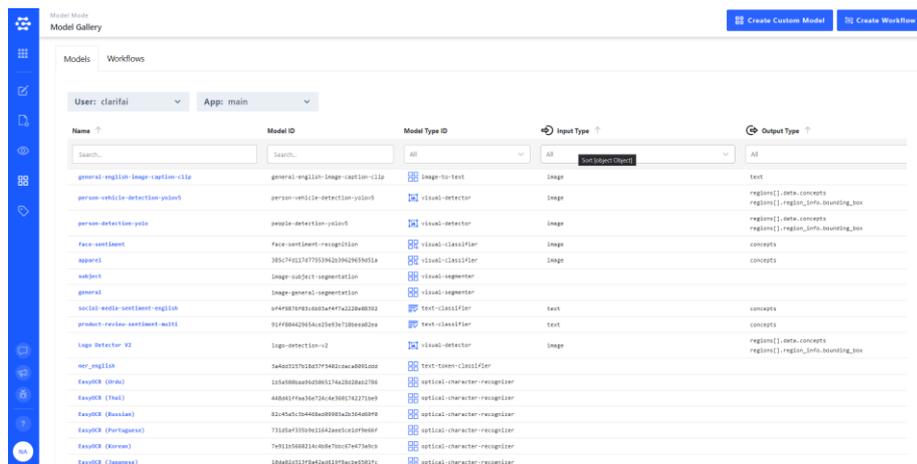
Clarifai merupakan API yang memiliki spesialisasi dalam model *deep learning* yang digunakan untuk memahami gambar, video, audio, ataupun teks yang tidak terstruktur. Clarifai dapat membantu memecahkan masalah kompleks dengan menggunakan *object classification*, *detection*, *tracking*, *geolocation*, *visual search*, dan *natural language processing*.

Clarifai menyediakan dua pilihan untuk pengembang dalam penggunaannya. Pengembang dapat membuat *workflow* sendiri yang dimulai dengan pembuatan model dan melakukan training data terlebih dahulu. Pengembang juga dapat menggunakan model siap pakai yang sudah dilakukan training untuk keperluan pengembangan aplikasi.



Gambar 2.10 Membuat Custom Model

Gambar 2.10 menunjukkan halaman dimana pengembang dapat membuat custom model sesuai kebutuhan pada saat pengembangan aplikasi. Setiap jenis model harus menjalani training data terlebih dahulu sebelum dapat digunakan.

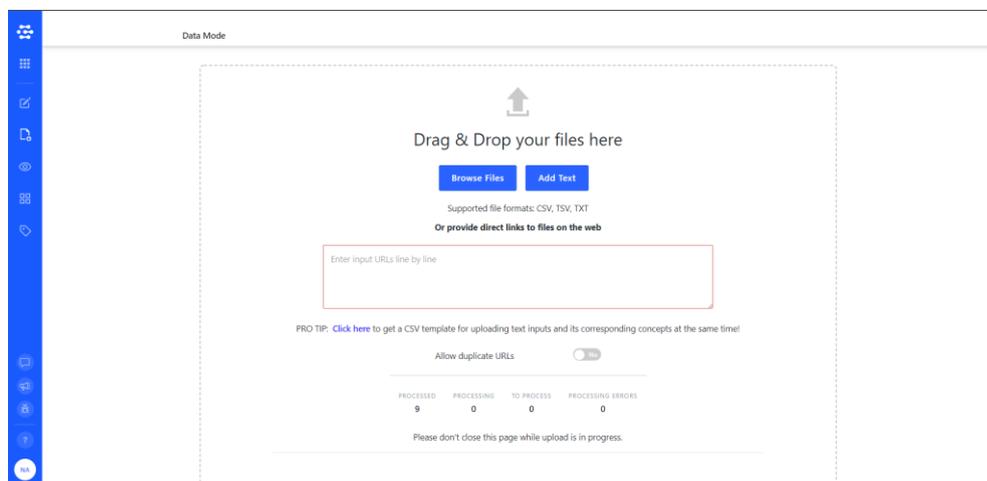


Gambar 2.11 Gallery Model Clarifai

Gambar 2.11 menunjukkan halaman dimana pengembang dapat memilih model yang sudah siap pakai sesuai kebutuhan pada saat pengembangan aplikasi. Model yang ada pada gallery ini sudah dilakukan training data sehingga pengembang dapat langsung menggunakan model tersebut.

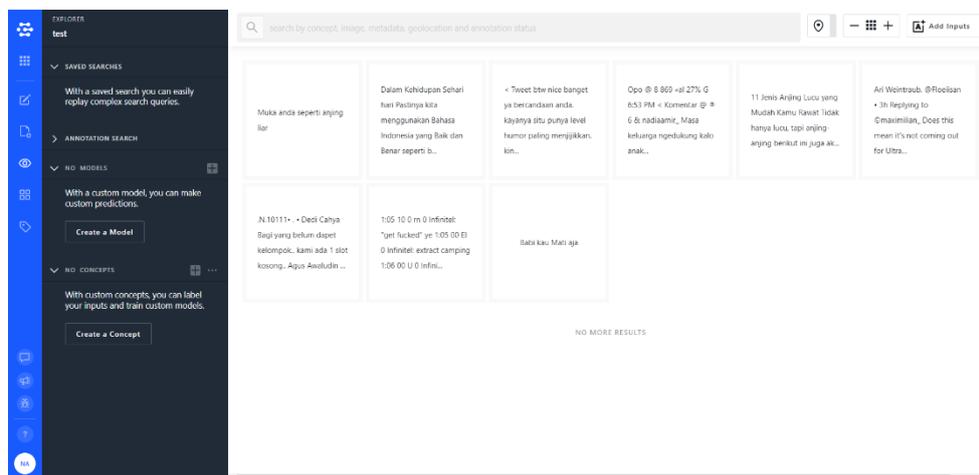
Penelitian ini akan menggunakan model yang sudah siap pakai, yaitu multilingual-moderation-text-classifier. Dimana fungsi dari model ini adalah melakukan klasifikasi teks berdasarkan beberapa label yaitu : *toxic* (perilaku

negatif), *threat* (ancaman), *identity hate* (ujaran kebencian), *insult* (pelecehan), *obscene* (vulgar), dan *severe toxicity* (perilaku negatif yang berlebih).



Gambar 2.12 Halaman untuk Mengunggah File Input Clarifai

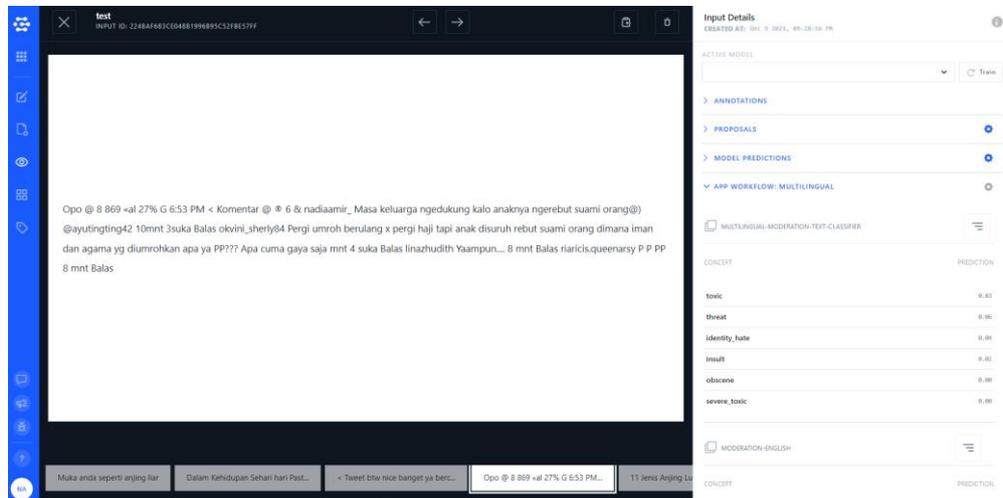
Pengembang dapat mencoba model klasifikasi yang dipilih dengan mengunggah file text dengan format CSV, TSV, dan TXT seperti pada Gambar 2.12. File tersebut digunakan sebagai *input* untuk dilakukan klasifikasi sesuai dengan model yang sudah dipilih sebelumnya



Gambar 2.13 Halaman Explorer Clarifai

File yang telah diunggah kemudian akan terlihat di halaman *Explorer* seperti pada Gambar 2.13. Pengembang dapat memilih teks mana yang akan

dilakukan klasifikasi sesuai dengan model yang telah dipilih sebelumnya dan melihat hasil dari klasifikasi pada halaman ini.



Gambar 2.14 Hasil Klasifikasi Teks

Gambar 2.14 menunjukkan halaman hasil klasifikasi dari teks yang telah dimasukkan. Dapat dilihat bahwa hasil klasifikasi teks ini memiliki skor *toxic* sebesar 0.83. Dari hasil tersebut, cukup terbukti jika model ini memiliki kinerja yang akurat sehingga digunakan dalam penelitian ini.

2.16 Mean Aritmetika (Rata-Rata Hitung)

Mean merupakan ukuran gejala pusat yang dapat dikatakan sebagai wakil dari sekumpulan data. Mean dapat ditentukan dengan cara menjumlah seluruh nilai pada data kemudian membagi dengan banyaknya data yang digunakan [15]. Bila data peubah acak X sebanyak n buah dinotasikan sebagai $x_1, x_2, x_3, \dots, x_n$ maka Mean dari data tersebut dapat dituliskan sebagai berikut :

$$\tilde{x} = \frac{x_1 + x_2 + x_3 + \dots + x_n}{n}$$

Dengan adanya landasan teori mengenai perhitungan Mean Aritmetika akan membantu penelitian untuk membangun aplikasi monitoring *cyberbullying* ini yang kemudian akan dijadikan acuan untuk menentukan rata – rata skor hasil klasifikasi *cyberbullying*.

2.17 MariaDB

Hampir seluruh sistem yang dibangun memiliki sebuah basis data yang bekerja dibelakang layar. Basis data digunakan untuk menyimpan, mengurutkan, dan menganalisa data yang dimiliki. Data mulai dari katalog sebuah produk, buku, mentions media sosial, dan lain sebagainya. Basis data merupakan fondasi dari dunia modern.

Basis data yang paling populer digunakan adalah MySQL yang dibuat oleh Michael “Monty” Widenius pada tahun 1995. Pada tahun 2009 Monty juga merilis MariaDB dengan harapan menjadi pengganti dan generasi berikutnya dari MySQL. MariaDB mulai menggantikan MySQL dalam pengembangan sistem karena performa yang lebih baik [16].

MariaDB merupakan basis data *open source* yang berarti sumber kode dari basis data ini bebas diakses oleh publik secara gratis. MariaDB secara cepat meraih popularitas lebih cepat dari basis data lainnya. Hingga saat ini MariaDB mendukung banyak *website* dan menjadi pilihan banyak perusahaan untuk digunakan sebagai basis data dalam perusahaan tersebut [17].