

BAB 2

TINJAUAN PUSTAKA

2.1. Multibahasa

Multibahasa merupakan suatu tindakan menggunakan banyak bahasa oleh individu atau masyarakat. Multibahasa menjadi salah satu fenomena sosial yang disebabkan oleh pengaruh globalisasi dan keterbukaan budaya. Berkat kemudahan akses informasi yang diberikan oleh Internet, semakin banyak orang yang terpapar oleh berbagai jenis bahasa. Multibahasa dapat ditemukan baik dalam individu maupun dalam suatu komunitas.

Di sisi lain, multibahasa dapat didefinisikan sebagai keahlian dan kefasihan berbahasa yang tinggi. Fenomena ini mencerminkan kemampuan seseorang untuk menggunakan berbagai bahasa dengan lancar dan tepat, menciptakan jaringan komunikasi yang lebih luas dan kompleks. Bahasa memiliki peran krusial dalam kehidupan manusia sebagai individu dan masyarakat. Melalui bahasa, manusia dapat saling berkomunikasi dan terhubung satu sama lain[14].

2.2. Translasi

Dalam konteks bahasa, translasi merujuk pada proses mengubah teks atau ucapan dari satu bahasa ke bahasa lain dengan tetap mempertahankan makna yang setidaknya seakurat mungkin. Proses translasi melibatkan pilihan kata, struktur kalimat, dan konteks budaya untuk memastikan pesan asli dapat dipahami dengan benar dalam bahasa sasaran. Proses translasi dapat dilakukan oleh manusia (penerjemah) atau menggunakan teknologi seperti program penerjemah otomatis. Penerjemahan pada dasarnya adalah tindakan linguistik yang bertujuan untuk menuliskan ulang pesan dari Bahasa sumber ke Bahasa sasaran[15].

Proses translasi dengan manusia (penerjemah) dapat dilakukan secara lisan maupun tulisan. Translasi lisan sering disebut dengan penerjemahan. Penerjemahan lisan biasanya dilakukan oleh penerjemah profesional dalam suatu acara, seperti konferensi atau rapat. Penerjemah lisan harus memiliki keterampilan mendengarkan, berbicara, dan menerjemahkan secara cepat dan akurat. Translasi

tulisan sering disebut dengan penerjemahan. Penerjemahan tulisan biasanya dilakukan oleh penerjemah profesional untuk berbagai keperluan, seperti dokumen hukum, dokumen medis, atau karya sastra. Penerjemah tulisan harus memiliki keterampilan membaca, menulis, dan menerjemahkan secara cermat dan akurat.

2.3. Voice Recognition

Voice Recognition adalah teknologi yang memungkinkan komputer atau sistem lain untuk mengenali suara manusia. Teknologi ini dapat diterapkan untuk berbagai tujuan, termasuk translasi bahasa. Dengan menggunakan teknologi voice recognition, pengguna dapat memasukkan teks menggunakan suara, yang membantu individu yang mengalami keterbatasan dalam mengetik teks secara manual. Voice Recognition adalah proses di mana komputer dapat mendengar kata-kata yang diucapkan oleh seseorang dan kemudian mengubah kata-kata tersebut ke dalam format yang dapat dipahami oleh mesin. Setelah itu, informasi tersebut diterjemahkan ke dalam teks, suara, atau format lain yang diperlukan, tergantung pada tujuan akhir[16].

Voice Recognition juga dikenal sebagai Speech Recognition. Voice recognition adalah istilah yang lebih luas yang mencakup semua kemampuan mengenali suara manusia, sedangkan speech recognition adalah bagian spesifik dari voice recognition yang fokus pada mengenali kata-kata atau frasa yang diucapkan. Speech Recognition atau yang biasa dikenal dengan automatic speech recognition (ASR) merupakan suatu pengembangan teknik dan sistem yang memungkinkan komputer untuk menerima masukan berupa kata yang diucapkan. Sistem automatic speech recognition merubah sinyal suara menjadi bentuk digital, contohnya spectrum[17].

Terdapat lima jenis pengenalan kata berdasarkan kemampuan dalam mengenal kata yang diucapkan, yaitu :

6. Kata-kata yang terisolasi

Proses pengidentifikasian kata yang hanya dapat mengenal kata yang diucapkan jika kata tersebut memiliki jeda waktu pengucapan antar kata.

7. Kata-kata yang berhubungan

Proses pengidentifikasian kata yang mirip dengan kata-kata terisolasi, namun membutuhkan jeda waktu pengucapan antar kata yang lebih singkat.

8. Kata-kata yang berkelanjutan

Proses pengidentifikasian kata yang sudah lebih maju karena dapat mengenal kata-kata yang diucapkan secara berkesinambungan dengan jeda waktu yang sangat sedikit atau tanpa jeda waktu. Proses pengenalan suara ini sangat rumit karena membutuhkan metode khusus untuk membedakan kata-kata yang diucapkan tanpa jeda waktu. Pengguna perangkat ini dapat mengucapkan kata-kata secara natural.

9. Kata-kata spontan

Proses pengidentifikasian kata yang dapat mengenal kata-kata yang diucapkan secara spontan tanpa jeda waktu antar kata.

10. Verifikasi atau identifikasi suara

Proses pengidentifikasian kata yang tidak hanya mampu mengenal kata, namun juga mengidentifikasi siapa yang berbicara.

Proses kerja alat speech recognition memiliki 4 tahap atau langkah utama dalam prosesnya, yaitu :

1. Tahap penerimaan masukan, sumber suara diterima melalui media perantara.
2. Tahap ekstraksi, penyimpanan masukan yang berupa suara dan sekaligus pembuatan basis data sebagai pola.
3. Tahap perbandingan, tahap ini merupakan tahap pencocokan data baru dengan data suara (pencocokan tata bahasa) pada pola.
4. Tahap validasi identitas pengguna, alat pengenalan ucapan yang sudah memiliki sistem verifikasi/identifikasi suara akan melakukan identifikasi orang yang berbicara berdasarkan kata yang diucapkan setelah menerjemahkan suara tersebut menjadi tulisan atau komando.

Kelebihan dari penggunaan teknologi speech recognition adalah :

1. Cepat, teknologi ini mempercepat transmisi informasi dan umpan balik dari transmisi tersebut.

2. Mudah digunakan, kemudahan teknologi ini juga dapat dilihat dalam aplikasi komando suara. Komando yang biasanya dimasukkan ke dalam komputer dengan menggunakan keyboard kini dapat dengan mudahnya kita lakukan tanpa perangkat keras, yakni dengan komando suara.

Kekurangan dari penggunaan teknologi speech recognition adalah:

1. Rawan terhadap gangguan, hal ini disebabkan oleh proses sinyal suara yang masih berbasis frekuensi. Ketika sebuah informasi dalam sinyal suara mempunyai komponen frekuensi yang sama banyaknya dengan komponen frekuensi gangguannya, akan sulit untuk memisahkan gangguan dari sinyal suara.
2. Jumlah kata yang dapat dikenal terbatas, hal ini disebabkan pengenalan ucapan bekerja dengan cara mencari kemiripan dengan basis data yang dimiliki.

2.4. Aplikasi

Aplikasi adalah subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Aplikasi secara umum dibagi menjadi tiga platform yaitu desktop, web, dan mobile. Aplikasi mobile adalah teknologi yang paling banyak digunakan, hal ini didukung dengan perkembangan smartphone yang semakin canggih[18]. Secara sederhana, Aplikasi dibagi menjadi tiga tipe yaitu :

1. Native Apps

Native apps adalah aplikasi yang dikembangkan khusus untuk satu sistem operasi tertentu, seperti Android atau iOS. Aplikasi ini ditulis menggunakan bahasa pemrograman yang native untuk sistem operasi tersebut, seperti Java untuk Android dan Swift untuk iOS. Keuntungan dari Native apps adalah memiliki kinerja tinggi dan memastikan pengalaman pengguna yang baik karena pengembang menggunakan UI perangkat asli. Selain itu, Native apps dapat diakses dari berbagai API yang tidak membatasi penggunaan aplikasi, dan mereka memiliki yang jelas untuk menjangkau pelanggan sasaran.

2. Web Apps

Web apps adalah aplikasi yang dapat diakses melalui browser web, baik di perangkat desktop maupun mobile. Aplikasi ini ditulis menggunakan bahasa pemrograman web, seperti HTML, CSS, dan JavaScript. Selain itu, web apps memerlukan memori perangkat minimum, sebagai aturan. Karena semua basis data pribadi disimpan di server, jadi pengguna dapat mengaksesnya dari perangkat apa pun yang terhubung ke internet. Oleh karena itu, jika menggunakan web apps yang memiliki koneksi yang buruk, maka akan memiliki pengalaman pengguna yang buruk. Kekurangannya adalah bahwa pengembang tidak memiliki akses ke banyak API (selain geolokasi dan beberapa lainnya).

3. Hybrid Apps

Hybrid apps adalah aplikasi yang menggabungkan fitur-fitur native apps dan web apps. Aplikasi ini ditulis menggunakan bahasa pemrograman web, tetapi dapat diakses melalui native apps. Aplikasi hybrid berkembang dengan cepat dan relatif mudah dibuat. Pembaruan lancar dan perawatan murah dijamin oleh basis kode tunggal untuk semua platform. API geolokasi, akselerometer, dan giroskop tersedia.

2.5. Android

Android merupakan sistem operasi berbasis linux yang digunakan untuk telepon seluler (mobile), seperti telepon pintar (smartphone) dan Komputer Tablet. Android awalnya dikembangkan oleh Android Inc., dengan dukungan finansial dari Google, yang kemudian membelinya pada tahun 2005. Setelah itu, Android resmi liris pada tanggal 5 November 2007 bersamaan dengan berdirinya Open Handset Alliance yang mana merupakan perusahaan telekomunikasi yang bertujuan untuk dapat memajukan standar dari perangkat seluler yang ada. Di sisi lain, pihak Google merilis kode-kode pada Android di bawah naungan lisensi Apache yang mana merupakan lisensi perangkat lunak serta standar terbuka untuk perangkat seluler. Terdapat dua jenis distributor resmi dari sistem Android, yang pertama memiliki dukungan penuh Google (Google Mail Services) dan yang kedua adalah distributor yang memang benar benar bebas tanpa adanya dukungan dari Google yang sering disebut Open Handset Distribution (OHD)[19].

Android memiliki sejumlah besar komunitas pengembang aplikasi yang memperluas fungsionalitas perangkat, umumnya ditulis dalam versi kustomisasi bahasa pemrograman Java. Pada bulan Oktober 2013, Android memiliki lebih dari satu juta aplikasi yang dapat diakses, dan Google Play, toko aplikasi utama Android, memiliki komunitas pengembang aplikasi yang membantu meningkatkan fungsionalitas perangkat. Sebuah survei yang dilakukan pada bulan April-Mei 2013 menemukan bahwa Android adalah platform pengembang paling populer, dengan lebih dari 50 miliar aplikasi yang diunduh. Di Google I/O 2014, Google melaporkan peningkatan dari 583 juta pengguna bulanan aktif Android menjadi lebih dari satu miliar.

Faktor-faktor di atas telah memengaruhi perkembangan Android, menjadikannya sebagai sistem operasi telepon pintar yang paling populer di dunia pada tahun 2010, mengalahkan Symbian. Android juga tersedia untuk perusahaan teknologi yang menginginkan sistem operasi murah, dapat disesuaikan, dan ringan untuk perangkat berteknologi tinggi tanpa perlu mengembangkannya dari awal. Sifat Android yang terbuka juga telah mendorong sejumlah besar komunitas pengembang aplikasi untuk menggunakan kode sumber terbuka sebagai dasar proyek pembuatan aplikasi, dengan menambahkan fitur baru untuk pengguna tingkat lanjut atau menjalankan Android pada perangkat yang secara resmi dirilis dengan sistem operasi lain. Uniknya, Android menggunakan nama-nama makanan untuk membedakan versi sistem android yang diluncurkannya. Android menggunakan huruf depan dari nama makanan tersebut sebagai penanda peningkatan versi sistemnya[19]. Perkembangan versi android dari pertama kali rilis hingga sekarang dapat dilihat pada Tabel 2.1 berikut.

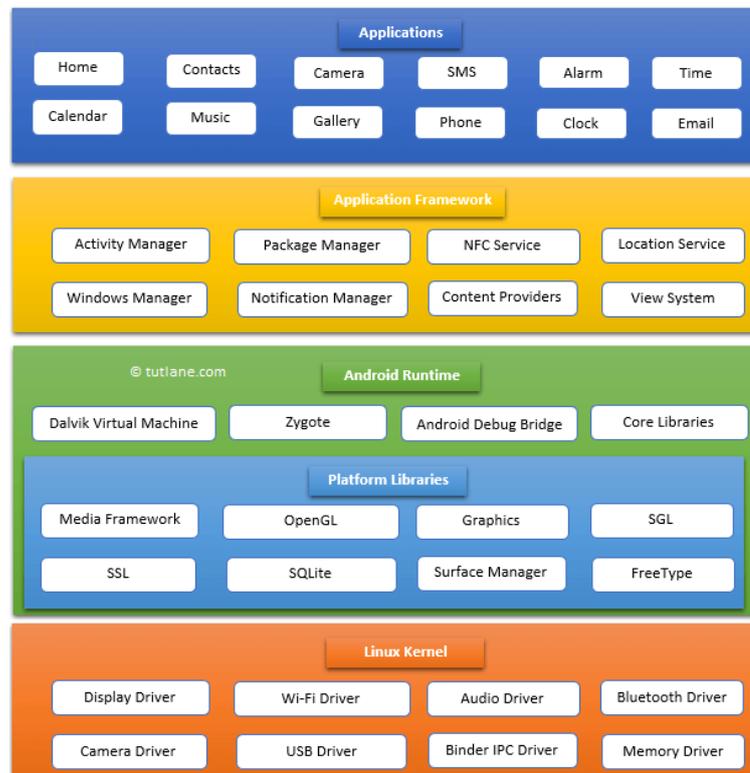
Tabel 2.1 Versi Android

No	Versi	Nama Kode	Tanggal rilis	Level API
1	1.5	Cupcake	30 April 2009	3
2	1.6	Donut	15 September 2009	4
3	2.0-2.1	Eclair	26 Oktober 2009	7
4	2.2	Froyo	20 Mei 2010	8

5	2.3-2.3.2	Gingerbread	6 Desember 2010	9
6	2.3.3-2.3.7	Gingerbread	9 Februari 2011	10
7	3.1	Honeycomb	10 Mei 2011	12
8	3.2	Honeycomb	15 Juli 2011	13
9	4.0.3–4.0.4	Ice Cream Sandwich	16 Desember 2011	15
10	4.1.x	Jelly Bean	9 Juli 2012	16
11	4.2.x	Jelly Bean	13 November 2012	17
12	4.3.x	Jelly Bean	24 Juli 2013	18
13	4.4.x	Kitkat	31 Oktober 2013	19
14	5.0	Lolipop	15 Oktober 2014	21
15	5.1	Lolipop	9 Maret 2015	22
16	6.0	Marshmallow	19 Agustus 2015	23
17	7.0	Nougat	22 Agustus 2016	24
18	8.0	Oreo	21 Agustus 2017	26
19	9.0	Pie	6 Agustus 2018	28
20	10	10/Q	3 September 2019	29
21	11	Red Velvet	8 September 2020	30
22	12	Snow Cone	14 Oktober 2020	31
23	13	Tiramisu	15 Agustus 2022	33
24	14	Upside Down Cake	4 Oktober 2023	34

2.5.1. Arsitektur Android

Android disusun dalam bentuk software stack yang terdiri dari applications layer, applications framework, Android Runtime, Libraries, Linux Kernel. Arsitektur ini ditampilkan pada Gambar 2.2.



Gambar 2.1 Arsitektur Sistem Operasi Android

Sumber: https://www.tutlane.com/images/android/android_architecture_diagram.png

Setiap lapisan dari tumpukan, dan unsur-unsur yang sesuai dalam setiap lapisan, saling terintegrasi untuk memberikan pengembangan aplikasi dan lingkungan eksekusi yang optimal untuk perangkat mobile[20]. Penjelasan dari tiap lapisan tersebut adalah sebagai berikut :

1. Application layer

Application Layer adalah lapisan yang paling terlihat oleh pengguna saat menjalankan program. Pengguna hanya akan melihat lapisan ini saat program digunakan, tanpa mengetahui prosesnya terjadi di balik lapisan aplikasi. Lapisan ini berjalan dalam Android runtime dengan menggunakan kelas dan service yang tersedia di framework aplikasi.

2. Application Framework

Kerangka aplikasi menyediakan kelas-kelas yang dapat digunakan untuk mengembangkan aplikasi android. Selain itu, memungkinkan abstraksi umum untuk mengakses perangkat dan mengatur sumber daya aplikasi dan tampilan user Interface.

3. Android Runtime

Sebagian besar fungsi android disediakan oleh paket pustaka inti yang tertanam dalam android. Ini adalah aspek yang membedakan Android dari sistem operasi lain yang menggunakan Linux. Android Runtime adalah mesin virtual yang meningkatkan kemampuan aplikasi Android dengan paket pustaka yang tersedia.

4. Libraries

Libraries merupakan lapisan dimana fitur-fitur android berada. Dengan standar distribusi software Berkeley (BSD), Android menggunakan beberapa paket pustaka C/C++, dan jumlah paket yang ditanamkan pada kernel Linux hanya setengah dari yang pertama. Beberapa Pustaka diantaranya :

- a. Media Library untuk memutar dan merekam berbagai format audio dan video.
- b. Surface Manager untuk mengatur hak akses lapisan untuk berbagai aplikasi.
- c. Graphic Library, yang mencakup SGL dan OpenGL untuk tampilan 2D dan 3D.
- d. SQLite untuk mengatur hubungan database yang digunakan oleh aplikasi.
- e. SSL dan WebKit untuk browser dan keamanan internet.

5. Linux Kernel

Linux Kernel menyediakan layanan sistem dasar seperti manajemen memori, manajemen proses, dan manajemen perangkat keras. Android dibangun di atas kernel Linux 2.6. namun, secara keseluruhan, Android bukanlah Linux karena tidak ada paket standar Linux lainnya. Karena Linux adalah sistem operasi terbuka yang baik untuk manajemen proses dan memori, android hanya memiliki beberapa layanan yang diperlukan: keamanan, manajemen memori, manajemen proses, jaringan, dan driver. Untuk mengatur aplikasi dan lubang keamanan, Kernel Linux

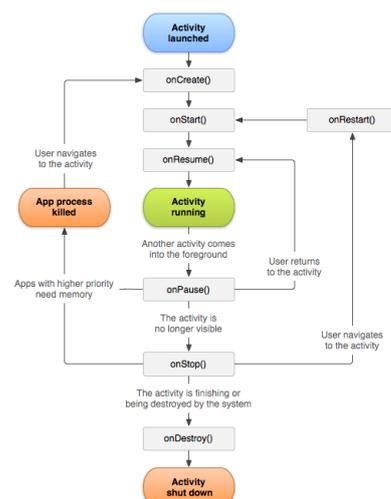
menyediakan driver untuk layar, kamera, keypad, WiFi, Memori Flash, audio, dan IPC.

2.5.2. Komponen Android

Komponen Android adalah blok bangunan dasar yang digunakan untuk membangun aplikasi Android. Terdapat enam komponen utama yang dapat digunakan dalam aplikasi android diantaranya[21] :

1. Activity

Setiap user interface diwakili oleh kelas activity. Sebuah aplikasi dapat terdiri dari satu atau lebih activity yang diproses dalam Linux. Setiap activity memiliki siklus hidup yang digunakan untuk menavigasi antara tahap siklus activity. Selama siklus hidupnya, activity dapat memiliki lebih dari dua status. Karena sistem secara otomatis menangani semua status, pengguna tidak dapat mengontrol setiap status. Namun, saat terjadi perubahan status melalui metode callback, pengguna akan menerima pesan. Saat pengguna meninggalkan dan kembali ke activity, perilaku activity dapat ditetapkan dalam siklus hidup metode callback. Itu adalah fungsi metode callback sebuah aplikasi pada platform Android. Gambar berikut menunjukkan alur activity untuk platform Android.



Gambar 2.2 Siklus hidup activity

Sumber : <https://static.javatpoint.com/images/androidimages/Android-Activity-Lifecycle.png>

Dari siklus tersebut dapat diketahui bahwa sebuah kelas Activity menyediakan tujuh method inti callback, Adapun penjelasan dari tiap method callback tersebut yaitu sebagai berikut :

- a. onCreate(), method ini adalah method utama dari setiap Activity. Saat aplikasi dijalankan, method ini dipanggil. method ini biasanya digunakan saat deklarasi variabel atau membuat user interface.
- b. onStart(), Ketika method onCreate() digunakan, method ini menunjukkan activity yang ditampilkan ke pengguna (user).
- c. onResume(), Ketika method onStart() selesai, method ini dipanggil. method ini adalah cara aplikasi berinteraksi dengan pengguna. Aplikasi akan tetap dalam keadaan ini sampai ada tindakan yang dilakukan olehnya, seperti menerima panggilan telepon atau mematikan layar smartphone.
- d. onPause(), method ini digunakan ketika pengguna meninggalkan activity, meskipun tidak selalu berarti activity dihancurkan. Ini berguna untuk menghentikan suatu fungsi, seperti menjeda pemutaran musik, dan sebagainya.
- e. onStop(), method ini digunakan ketika aplikasi tidak terlihat lagi oleh pengguna, atau ketika aplikasi berhenti berjalan. Ini dapat terjadi ketika aplikasi berjalan di belakang layar dalam waktu yang lama. Selain itu, sistem dapat menghubungi method ini ketika aplikasi selesai berjalan dan akan segera dihentikan.
- f. onDestroy, method ini disebut ketika activity telah selesai, yang berarti aplikasi benar-benar berhenti bekerja, atau ketika proses yang mengandung activity tersebut dihancurkan oleh sistem untuk menghemat ruang memori.
- g. onRestart(), Ketika activity kembali menampilkan user interface setelah status stop, method ini digunakan.

2. Service

Service digunakan untuk menjalankan proses aplikasi di belakang layar. Misalnya, ia dapat memainkan musik di latar belakang saat pengguna berada dalam beberapa aplikasi, atau dapat mengambil data melalui jaringan tanpa mengganggu interaksi pengguna dengan aktivitas.

3. Content Provider

Dalam aplikasi, Content Provider digunakan untuk mengakses atau membagikan data (berhubungan dengan database) yang diperlukan untuk aktivitas tertentu.

4. Resource

Resource digunakan untuk mengatur sumber daya aplikasi dengan menyimpan file coding dan non-coding yang diperlukan dalam aplikasi ke dalam folder tertentu. File coding, seperti file XML yang digunakan untuk membuat User Interface, disimpan dalam folder res/layout, sedangkan file non-coding, seperti icon, gambar, dan audio, disimpan dalam folder res/drawable ldpi.

5. Views

Views digunakan untuk membuat antarmuka pengguna untuk komponen activity.

6. Notification

Notification digunakan sebagai mekanisme sinyal untuk memberikan pemberitahuan kepada pengguna secara acak pada waktu yang telah ditentukan. Ini dapat mencakup pesan masuk, telepon masuk, dan hal-hal lainnya yang ditampilkan pada status bar.

2.6. Android Studio

Android Studio adalah Integrated Development Environment (IDE) resmi untuk sistem operasi Android, yang dibangun di atas perangkat lunak JetBrains IntelliJ IDEA dan didesain khusus untuk pengembangan Android. IDE ini merupakan pengganti dari Eclipse Android Development Tools (ADT) yang sebelumnya merupakan IDE utama untuk pengembangan aplikasi android. Eclipse dulunya menjadi IDE favorit dan pilihan bagi banyak pengembang Java, wajar jika ini juga menjadi IDE pilihan bagi pengembang Android[22].

Pada konferensi Google I/O 16 Mei 2013, Android studio sendiri pertama kali diumumkan. Ini adalah tahap preview dari versi 0.1 yang dirilis pada Mei 2013. Setelah memasuki tahap beta dari versi 0.8, rilis stabil pertama dimulai pada Desember 2014 dan dimulai dengan versi 1.0. Versi stabil saat ini, versi 4.0, dirilis

pada Mei 2020. Perkembangan versi android studio dari pertama kali rilis hingga sekarang dapat dilihat pada Tabel 2.2 berikut.

Tabel 2.2 Versi Android Studio

No	Versi	Bulan dan Tahun Rilis
1	1.0	Desember 2014
2	1.1	Februari 2015
3	1.2	April 2015
4	1.3	Juli 2015
5	1.4	September 2015
6	1.5	November 2015
7	2.0	April 2016
8	2.1	April 2016
9	2.2	September 2016
10	2.3	Maret 2017
11	3.0	Oktober 2017
12	3.1	Maret 2018
13	3.2	September 2018
14	3.3	Januari 2019
15	3.4	April 2019
16	3.5	Agustus 2019
17	3.6	Februari 2020
18	4.0	Mei 2020
19	4.1	Oktober 2020
20	4.2	Mei 2021
21	Arctic Fox (2020.3.1)	Juli 2021
22	Bumblebee (2021.1.1)	Januari 2022
23	Chipmunk (2021.2.1)	Mei 2022
24	Dolphin (2021.3.1)	September 2022
25	Electric Eel (2022.1.1)	Januari 2023
26	Flamingo (2022.2.1)	April 2023
27	Giraffe (2022.3.1)	Juli 2023

Android studio memiliki fitur yang sangat membantu developer untuk membangun aplikasi android. Beberapa Fitur yang tersedia saat ini dalam versi stabil yaitu :

- a. Dukungan untuk build berbasis-Gradle
- b. refactoring dan perbaikan cepat yang khusus untuk Android.
- c. Alat Lint digunakan untuk mengukur masalah seperti kinerja, kegunaan, kompatibilitas versi, dan lainnya.
- d. Kemampuan penanda tangan aplikasi serta integrasi Proguard.
- e. Template-based wizards untuk membuat desain templat umum seperti drawer atau empty activity.
- f. Mendukung untuk pengembangan aplikasi Android Wear.
- g. Editor tata letak yang memungkinkan pengguna untuk menyeret dan menjatuhkan (drag-and-drop) komponen UI, opsi untuk melihat tata letak pada beberapa konfigurasi layar.
- h. Dukungan bawaan untuk Google Cloud Platform, memungkinkan integrasi dengan Firebase Cloud Messaging ('Perpesanan Google Cloud' Sebelumnya) dan Google App Engine.
- i. Android Virtual Device (Emulator) untuk menjalankan dan men-debug aplikasi di studio Android.

2.7. Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini pertama kali diciptakan oleh James Gosling ketika dia bekerja di Sun Microsystems—yang sekarang menjadi bagian dari Oracle—tahun 1995. Bahasa ini banyak menggunakan sintaksis C dan C++, tetapi dengan sintaksis model objek yang lebih sederhana dan dukungan rutin-rutin aras bawah yang sedikit. Aplikasi berbasis Java biasanya dikompilasi ke dalam p-code, juga dikenal sebagai bytecode, dan dapat dijalankan pada berbagai mesin virtual Java (JVM).

Java merupakan salah satu bahasa pemrograman dengan konsep OOP. Konsep sentral dari pemrograman berorientasi objek adalah objek, yang merupakan sejenis

modul berisi data dan subrutin. Sudut pandang dalam OOP adalah bahwa suatu objek adalah sejenis entitas mandiri yang memiliki keadaan internal (data yang dikandungnya) dan dapat merespons pesan (panggilan ke subrutinnya). Pendekatan OOP untuk rekayasa perangkat lunak dimulai dengan mengidentifikasi objek yang terlibat di dalamnya[23].

Bahasa pemrograman Java lahir dari proyek bernama The Green Project yang berlangsung selama 18 bulan, dari awal 1991 hingga musim panas 1992 di Sun Microsystems. Dipimpin oleh Patrick Naughton, Mike Sheridan, dan James Gosling, proyek ini menghasilkan maskot Duke yang dibuat oleh Joe Palrang. Proyek awalnya menggunakan nama Oak, tetapi setelah proyek ini ditutup pada musim panas 1992, Java Oak pertama diluncurkan untuk mengendalikan peralatan dengan layar sentuh. Setelah era Star Seven selesai, proyek beralih ke anak perusahaan TV kabel di Palo Alto. Pada awal 1990-an, mereka memanfaatkan Internet untuk kolaborasi. Mereka mengembangkan perambah pertama, Web Runner, yang kemudian berganti nama menjadi Hot Java pada Maret 1995. Kode sumber Java versi 1.0a2 pertama kali dibuka pada waktu tersebut. Pada Mei 1995, kesuksesan Java diumumkan oleh surat kabar San Jose Mercury News. Namun, terjadi perpecahan pada pukul 04.00 di sebuah hotel Sheraton Palace, di mana Eric Schmidt, George Paolini, dan Marc Andreessen meninggalkan proyek untuk membentuk Netscape. Awalnya dinamai Oak karena pohon di depan jendela kerja James Gosling, namun, karena masalah merek dagang, nama Java diambil. Nama Java berasal dari kopi murni kesukaan Gosling yang berasal dari Pulau Jawa. Sehingga, nama bahasa pemrograman Java terinspirasi dari kata "Java" yang merujuk pada kopi Jawa.

Bahasa pemrograman java memiliki beberapa kelebihan, berikut adalah kelebihan dari Bahasa pemrograman java :

- a. Multiplatform, Keunggulan utama Java adalah dapat dijalankan di berbagai platform dan sistem operasi komputer karena prinsip tulis sekali jalankan di mana saja. Dengan demikian, programmer hanya perlu menulis program Java dan kemudian dikompilasi (diubah dari bahasa manusia menjadi bytecode atau

bahasa mesin), dan kemudian program tersebut dapat dijalankan di berbagai platform tanpa mengalami perubahan.

- b. Library kelas yang lengkap, Java terkenal dengan kelengkapan library atau perpustakaan, yang merupakan kumpulan program program yang terintegrasi dalam pemrograman Java dan sangat memudahkan bagi programmer untuk membangun aplikasi. Kelengkapan ini dibantu oleh komunitas Java yang besar, yang terus membuat perpustakaan baru untuk memenuhi semua kebutuhan pembangunan aplikasi.
- c. Bergaya C++, memiliki sintaks yang mirip dengan bahasa pemrograman C++, yang menarik banyak pemrogram C++ untuk beralih ke Java. Saat ini, banyak pengguna Java, sebagian besar adalah pemrogram C++.

Selain memiliki kelebihan, Bahasa pemrograman java juga memiliki kekurangan yang diantaranya :

- a. Mudah didekompilasi, Dekompilasi adalah proses membalikkan dari kode jadi menjadi kode sumber. Ini dimungkinkan karena kode jadi Java merupakan bytecode yang menyimpan banyak atribut bahasa tingkat tinggi, seperti nama-nama kelas, metode, dan tipe data. Hal yang sama juga terjadi pada Microsoft .NET Platform. Dengan demikian, algoritma yang digunakan program akan lebih sulit disembunyikan dan mudah.
- b. Penggunaan memori yang banyak, Program berbasis Java menggunakan memori yang lebih banyak daripada bahasa tingkat tinggi generasi sebelumnya seperti C/C++ dan Pascal—terutama Delphi dan Object Pascal. Karena memori terpasang semakin murah, ini biasanya tidak menjadi masalah bagi mereka yang menggunakan teknologi terbaru. Sebaliknya, ini menjadi masalah bagi mereka yang masih menggunakan komputer berusia lebih dari empat tahun.

2.8. Kotlin

Kotlin adalah bahasa pemrograman dengan sumber terbuka yang diketik secara statis yang berjalan pada Mesin Virtual Java dan menggunakan kompiler LLVM. Selain itu, ia memiliki kemampuan untuk dikompilasikan ke dalam bentuk kode sumber JavaScript. Pengembang utamanya berasal dari tim programmer dari

perusahaan Rusia JetBrains. Meskipun sintaksis Kotlin tidak sesuai dengan bahasa Java, ia dirancang untuk bekerja dengan kode Java dan bergantung pada kode dari Kelas Pustaka Java, termasuk berbagai framework. Tim pengembang memutuskan untuk menamakannya Kotlin karena itu mengambil nama sebuah pulau di Rusia, seperti halnya Java mengambil nama sebuah pulau di Indonesia yang disebut Jawa. Dengan pengumuman Google bahwa Kotlin akan menjadi bahasa utama untuk Android, bersama dengan Java dan C++, Kotlin menjadi bahasa resmi untuk pengembangan aplikasi Android.

Bahasa Kotlin pertama kali dirilis pada bulan Februari 2016 dengan versi 1.0, Versi ini secara resmi ditetapkan sebagai versi rilis stabil pertama dan JetBrains telah menetapkan dukungan versi sebelumnya untuk jangka panjang dengan versi ini. Pada Google I/O 2017, Google mengumumkan dukungan kelas pertama untuk Kotlin pada Android. Semenjak event tersebut popularitas bahasa pemrograman ini mengalami kemajuan yang pesat. Di balik kemajuan pesatnya Kotlin, terdapat pengembang - pengembang hebat dan kreatif dari JetBrains. Kotlin dikembangkan oleh lebih dari 50 developer pimpinan Andrey Breslav.

Kotlin dapat digunakan untuk mengembangkan aplikasi android, baik server-side maupun client-side. Karena itu, Kotlin dapat digunakan bersama Java dalam proyek aplikasi. NullPointerException (NPE) adalah masalah lain yang sering terjadi dengan Java. Kotlin didesain null-safety, sehingga masalah tersebut tidak lagi ditemui dalam Kotlin. Berbeda dengan Java, Kotlin mengetahui NPE pada saat compile time. Selain itu, Kotlin didukung oleh fitur programming fungsional seperti penggunaan ekspresi lambda, fungsi pengorderan yang lebih tinggi, penilaian yang tidak teliti, dan metode pengumpulan seperti filtering, mapping, dan pengorderan.

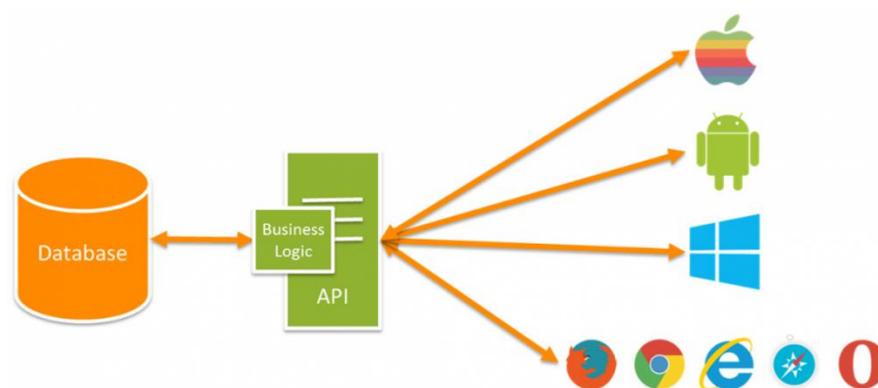
2.9. Application Programming Interface (API)

Application Programming Interface (API) adalah seperangkat aturan dan protocol yang memungkinkan berbagai aplikasi dan perangkat lunak berkomunikasi satu sama lain. API berfungsi sebagai perantara dan memungkinkan aplikasi berinteraksi dan berbagi data tanpa mengetahui informasi internal satu sama lain. Didalam Api terdapat istilah open API, dikenal juga dengan API publik.

API publik adalah sebuah antarmuka pemrograman yang bersifat publik yang tersedia untuk semua orang. Restful API adalah layanan web populer untuk open API dan JSON merupakan format data yang paling banyak digunakan[24]. Umumnya API terbagi menjadi empat jenis, yaitu :

1. Open API (Public API), yaitu jenis API terbuka yang dapat digunakan oleh siapa saja, baik pengembang aplikasi maupun perusahaan eksternal.
2. Internal API (Private API), yaitu jenis API yang digunakan di dalam suatu organisasi dan tidak diakses secara publik.
3. Partner API, yaitu jenis API yang dibagikan dengan pihak ketiga atau mitra bisnis tertentu. Akses ke Partner APIs biasanya memerlukan identifikasi dan otorisasi yang diberikan oleh penyedia API.
4. Composite API, yaitu jenis API yang memungkinkan penggabungan atau pengumpulan data dari berbagai sumber, yang memberikan antarmuka tunggal untuk mengakses fungsionalitas yang tersebar.

API digunakan untuk mempercepat pengembangan dengan memberikan fitur terpisah sehingga pembangun tidak perlu membuat fitur serupa. API biasanya berfungsi sebagai sumber data untuk kebutuhan sistem atau aplikasi tertentu. Ini memungkinkan pertukaran data dari satu aplikasi ke aplikasi lainnya. Gambar 2.4 berikut menunjukkan cara kerja API.



Gambar 2.3 Cara kerja API

Sumber : [https://www.codepolitan.com/mengenal-apa-itu-web-api-](https://www.codepolitan.com/mengenal-apa-itu-web-api-5a0c2855799c8)

5a0c2855799c8

Untuk melakukan permintaan (request) ke sebuah API, beberapa syarat umum yang perlu dipenuhi, diantaranya :

- a. Endpoint URL, Endpoint adalah alamat spesifik di mana API menyediakan layanan atau informasi tertentu.
- b. HTTP Method, Tentukan metode HTTP yang sesuai untuk jenis operasi yang ingin dilakukan. Metode umumnya melibatkan GET untuk mendapatkan data, POST untuk mengirim data baru, PUT untuk memperbarui data, dan DELETE untuk menghapus data.
- c. Request Headers, Beberapa API memerlukan pengaturan header tertentu dalam permintaan HTTP. Ini dapat mencakup header seperti Content-Type atau Authorization
- d. Request Body, berisikan data yang ingin dikirimkan oleh client ke server, sehingga server dapat melakukan tindakan sesuai dengan data yang dikirimkan oleh client.

2.10. Google Cloud Translation API

Cloud Translation API adalah layanan terjemahan yang disediakan oleh Google Cloud Platform (GCP). Google Cloud API yang didukung AI dapat membantu menerjemahkan dokumen, situs web, aplikasi, file audio, video, dan lebih banyak lagi secara besar-besaran dengan kualitas terbaik dan kontrol keamanan dan keamanan tingkat perusahaan. Google Cloud Translation API dapat menerjemahkan bahasa lebih dari 100 bahasa yang ada di dunia. Dengan adanya google cloud translation sekarang ini dapat membuat orang lebih mudah dan cepat dalam menerjemahkan kata-kata dan sekaligus menggantikan kamus tebal yang sering orang pakai dalam menerjemahkan kata.

Cloud Translation API menggunakan teknologi terjemahan mesin neural dari Google yang memungkinkan menerjemahkan teks secara dinamis melalui API dengan menggunakan melalui API menggunakan model kustom da terlatih Google, atau model bahasa besar khusus terjemahan (LLM)[25]. Cloud Translation dapat menerjemahkan teks antara ribuan pasangan bahasa secara dinamis. Dengan Cloud

Translation, situs dan program dapat terintegrasi secara terprogram dengan layanan terjemahan.

Untuk mengakses API ini dapat melalui beberapa cara seperti Pustaka library klien, REST atau GPRC (Cloud Translation – Hanya Tingkat lanjut). Sebelum dapat mulai menggunakan Cloud Translation API, developer diharuskan mengaktifkan project Cloud Translation API dan kredensial untuk dapat melakukan panggilan yang diautentikasi. setelah mendapatkan kredensial maka langkah selanjutnya melakukan request. Pada implementasinya pemanggilan fungsinya dibutuhkan tiga parameter di antaranya teks yang ingin di translasi, bahasa asal, bahasa tujuan. `Translation translation = translate.translate("Hola Mundo!", Translate.TranslateOption.sourceLanguage("es"), Translate.TranslateOption.targetLanguage("de"));`. Implementasi tersebut menggunakan metode pemanggilan dengan cara Pustaka library Klien.

2.11. Mikrofon

Mikrofon atau microphone, perangkat yang mengubah energi akustik (gelombang suara) menjadi energi listrik, yang kemudian dapat diperkuat, direkam, atau disiarkan. Gelombang suara terbentuk ketika udara di sekitar sumber suara bergetar dan menyebar ke udara sekitarnya dalam bentuk gelombang longitudinal. Mikrofon memiliki transduser yang mengubah getaran suara ini menjadi sinyal listrik. Transduser dalam mikrofon umumnya berupa membran tipis yang digerakkan oleh gelombang suara, dan pergerakan membran ini menghasilkan arus listrik yang sebanding dengan intensitas getaran tersebut. Cara kerja mikrofon mirip dengan fungsi telinga manusia.

Mikrofon digunakan dalam berbagai perangkat seperti telepon, alat perekam, alat bantu dengar, penyiaran radio dan televisi, dan lain-lain. Pada dasarnya, mikrofon berfungsi untuk mengubah suara menjadi getaran listrik dalam bentuk sinyal analog. Sinyal ini kemudian diperkuat dan diolah sesuai kebutuhan, biasanya melalui Power Amplifier, sehingga suara dengan intensitas rendah dapat diperkuat menjadi lebih keras sebelum akhirnya diumpankan ke speaker.

Pemilihan mikrofon harus dilakukan dengan hati-hati untuk memastikan bahwa performa mikrofon tetap optimal. Agar lebih efektif, mikrofon yang digunakan harus sesuai dengan kebutuhan dan seimbang dengan sumber suara yang ingin direkam, seperti suara manusia, alat musik, suara kendaraan, atau lainnya. Selain itu, sistem tata suara yang digunakan, seperti sistem suara untuk musik live, alat perekam, atau arena balap motor GP, juga harus dipertimbangkan.

Terdapat beberapa jenis mikrofon yang umum digunakan. Mikrofon dinamik adalah jenis yang paling umum dan paling tahan lama. Mikrofon dinamik bekerja berdasarkan prinsip elektromagnetik untuk menghasilkan sinyal listrik. Mikrofon kondensor, di sisi lain, adalah jenis yang lebih sensitif dan akurat dibandingkan mikrofon dinamik. Mikrofon kondensor menggunakan dua pelat yang dipisahkan oleh jarak sangat kecil, dan ketika suara masuk, satu pelat bergerak, menghasilkan perubahan kapasitansi yang kemudian diubah menjadi sinyal listrik. Mikrofon ribbon, yang dikenal sebagai jenis mikrofon paling sensitif dan akurat dalam merekam suara, juga paling rentan terhadap kerusakan.

2.12. Speech to Text

Speech to Text adalah bagian dari teknologi voice recognition, yang memungkinkan konversi bahasa dengan ucapan menjadi teks melalui komputasi linguistik. Teknologi ini sering juga disebut speech recognition. Aplikasi, alat, dan perangkat tertentu dapat melakukan transkripsi audio secara real-time untuk menampilkan teks serta merespons perintah suara. Teknologi ini berfungsi dengan mendengarkan audio dan menghasilkan transkrip yang dapat diedit secara langsung pada perangkat yang digunakan. Program komputer menggunakan algoritma linguistik untuk mengurutkan sinyal auditori dari kata-kata yang diucapkan dan mentransfer sinyal tersebut ke dalam teks menggunakan karakter yang disebut Unicode[26].

Penerapan teknologi speech-to-text telah berkembang dari penggunaan sehari-hari di telepon rumah menjadi berbagai aplikasi di industri seperti pemasaran, perbankan, dan medis. Aplikasi pengenalan suara ini menunjukkan bagaimana teknologi konversi suara ke teks dapat meningkatkan efisiensi tugas-tugas

sederhana serta memperluas cakupannya ke tugas-tugas yang biasanya dilakukan oleh manusia.

2.13. Speech Recognizer

Speech Recognizer merupakan kelas yang disediakan oleh Android, yang merupakan sistem operasi yang dikembangkan oleh Google. Penggunaan kelas Speech Recognizer memungkinkan pengembang untuk mengintegrasikan fitur pengenalan suara ke dalam aplikasi Android. Implementasi dari API ini kemungkinan akan mengalirkan audio ke server jarak jauh untuk melakukan pengenalan suara. Oleh karena itu, API ini tidak dimaksudkan digunakan untuk pengenalan kontinu, yang dapat mengonsumsi daya baterai dan bandwidth yang signifikan. Harap dicatat bahwa aplikasi harus memiliki izin Manifest.permission.RECORD_AUDIO untuk menggunakan kelas ini.

Speech Recognizer adalah teknologi yang memungkinkan komputer untuk mengenali dan menerjemahkan ucapan manusia menjadi teks tertulis. Ini memungkinkan interaksi suara antara manusia dan mesin, memungkinkan perintah suara, pengenalan teks dari audio, dan aplikasi lain yang melibatkan input suara. Speech Recognizer beroperasi menggunakan algoritma pemrosesan sinyal suara dan model machine learning untuk mengenali pola dan fitur dalam ucapan, kemudian menerjemahkannya menjadi teks yang dapat dipahami oleh komputer.

2.14. Google Cloud Text-to-Speech

Text-to-Speech memungkinkan untuk membuat ucapan manusia yang terdengar alami dan sintetis sebagai audio yang dapat diputar. developer dapat menggunakan file data audio yang dibuat menggunakan Text-to-Speech untuk mendukung aplikasi atau media peningkatan seperti rekaman video atau audio (sesuai dengan Persyaratan Layanan Google Cloud Platform) termasuk kepatuhan terhadap semua hukum yang berlaku). Text-to-Speech mengonversi input Bahasa Markup Sintesis Ucapan (SSML) atau teks menjadi data audio seperti MP3 atau LINEAR16 (encoding yang digunakan dalam file WAV). Text-to-Speech cocok untuk aplikasi apa pun yang memutar audio ucapan manusia kepada pengguna. Alat

ini memungkinkan untuk mengubah string, kata, dan kalimat arbitrer menjadi suara seseorang yang mengucapkan hal yang sama.

Proses menerjemahkan input teks menjadi data audio disebut sintesis dan output sintesis disebut ucapan sintesis. Text-to-Speech menggunakan dua jenis input: teks mentah atau data berformat SSML (dibahas di bawah). Untuk membuat file audio baru, panggil endpoint `synthesize` dari API-nya. Proses sintesis ucapan menghasilkan data audio mentah sebagai string berkode base64. developer harus melakukan dekode string berkode base64 menjadi file audio sebelum aplikasi dapat memutarinya. Sebagian besar platform dan sistem operasi memiliki alat untuk mendekode teks base64 menjadi file media yang dapat diputar. Keunggulan dari Google Cloud Text-to-Speech adalah Ucapan berakurasi tinggi yaitu menghasilkan ucapan dengan intonasi layaknya manusia. Dibangun berdasarkan keahlian sintesis ucapan DeepMind, API tersebut memberikan suara yang mendekati kualitas suara manusia, Keunggulan lain yaitu didukung lebih dari 50 bahasa dan 380 varian suara[27].

2.15. Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah bahasa pemodelan visual yang digunakan untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. UML menyediakan serangkaian diagram yang dapat digunakan untuk menggambarkan berbagai aspek dari sistem, termasuk struktur, perilaku, dan interaksinya. Tujuan utama UML adalah untuk menyediakan standar untuk pemodelan sistem perangkat lunak.

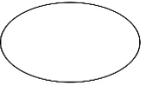
Dengan tujuan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan sistem, UML mencakup konsep semantik, notasi, dan pedoman yang bersifat universal. UML mendefinisikan struktur statis dan perilaku dinamis suatu sistem, memungkinkan pemodelan dari berbagai sudut pandang yang terpisah namun terkait. Selain itu, UML juga menyediakan konstruksi organisasi untuk mempartisi sistem besar, mengelola ketergantungan, dan mewakili keputusan implementasi.

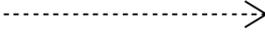
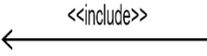
UML bukanlah bahasa pemrograman. Alat ini dapat menyediakan generator kode dari UML ke dalam berbagai bahasa pemrograman, serta membangun model rekayasa balik dari program yang sudah ada. UML bukanlah bahasa formal yang dimaksudkan untuk pembuktian teorema. Ada sejumlah bahasa seperti itu, namun tidak mudah dipahami atau digunakan untuk sebagian besar tujuan. UML adalah bahasa pemodelan tujuan umum. Untuk domain khusus, seperti tata letak GUI, desain sirkuit VLSI, atau kecerdasan buatan berbasis aturan, alat yang lebih khusus dengan bahasa khusus mungkin sesuai. UML adalah bahasa pemodelan diskrit. Hal ini tidak dimaksudkan untuk memodelkan sistem kontinu seperti yang ditemukan dalam bidang teknik dan fisika. UML dimaksudkan untuk menjadi bahasa pemodelan tujuan umum universal untuk sistem diskrit seperti yang dibuat dari perangkat lunak, firmware, atau logika digital[28]. Dalam UML terdapat banyak diagram, empat diantaranya adalah sebagai berikut:

2.15.1. Use Case Diagram

Use case diagram adalah teknik untuk merekam persyaratan fungsional sebuah system, menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Use case diagram menekankan kepada “apa” yang diperbuat oleh sistem, dan bukan “bagaimana”. Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem. Use case merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-create sebuah daftar belanja, dan sebagainya. Seorang atau sebuah aktor adalah sebuah entitas dapat berupa manusia atau mesin yang berinteraksi dengan system untuk melakukan pekerjaan-pekerjaan tertentu[29]. Simbol-simbol pada use case dapat dilihat pada Tabel 2.4 berikut [30].

Tabel 2.3 Simbol Use Case Diagram

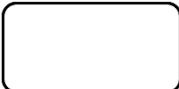
No	Simbol	Keterangan
1		Aktor : Mewakili peran orang, prose, sistem yang lain atau alat berkomunikasi dengan use case.
2		Use Case : abstraksi dan interaksi antara sistem dan aktor.

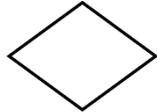
3		Association: abstraksi dari penhubung antara aktor dengan use case.
4		Generalisasi : Menggambarkan spesialisasi aktor untuk dapat berpartisipasi dengan use case.
5		Include : menunjukkan bahwa suatu use case seluruhnya merupakan fungsionalitas dari use case lainnya.
6		Extend : Menunjukkan bahwa suatu use case merupakan tambahin fungsionalitas dari use case lainnya jika suatu kondisi terpenuhi.

2.15.2. Activity Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Activity diagram merupakan state diagram khusus, di mana sebagian besar state adalah action dan sebagian besar transisi di-trigger oleh selesainya state sebelumnya (internal processing). Oleh karena itu activity diagram tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum[29]. Simbol-simbol pada activity diagram dapat dilihat pada Tabel 2.5 berikut[30].

Tabel 2.4 Simbol Activity Diagram

No	Simbol	Nama	Keterangan
1		Status awal	Menunjukkan awal dari sebuah proses
2		Aktivitas	Menggambarkan aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.

3		Percabangan / Decision	Percabangan Dimana ada pilihan lebih dari satu aktivitas.
4		Penggabungan / Join	Penggabungan Dimana aktivitas yang lebih dari satu digabungkan menjadi satu aktivitas.
5		Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6		Swimlane	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

2.15.3. Sequence Diagram

Sequence diagram digunakan untuk menjelaskan dan menunjukkan interaksi antar objek dalam sebuah sistem. Sequence diagram juga menunjukkan pesan atau perintah yang dikirim, serta waktu pelaksanaannya. Objek-objek yang berhubungan dengan berjalannya proses operasi biasanya diurutkan dari kiri ke kanan. Tujuan utama pembuatan sequence diagram adalah untuk mengetahui urutan kejadian yang dapat menghasilkan hasil yang diinginkan[31]. Contoh simbol-simbol sequence diagram dapat dilihat pada Tabel 2.6 berikut[30].

Tabel 2.5 Simbol Sequence Diagram

No	Simbol	Nama	Keterangan
1	 Actor	Aktor	Orang, poses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang
2		Garis hidup / Lifeline	Menyatakan kehidupan suatu objek.

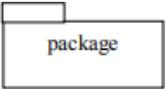
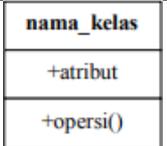
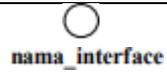
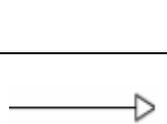
3		Objek	Menyatakan objek yang berinteraksi pesan
4		Pesan tipe create	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
No	Simbol	Nama	Keterangan
6		Pesan tipe call	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri
7		Pesan tipe send	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
8		Pesan tipe return	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
9		Pesan tipe destroy	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy

10		Pesan kembali	Mengindikasikan komunikasi kembali kedalam sebuah objek itu sendiri.
----	---	---------------	--

2.15.4. Class Diagram

Class diagram atau diagram kelas adalah salah satu jenis diagram struktur pada UML yang menggambarkan dengan jelas struktur serta deskripsi class, atribut, metode, dan hubungan dari setiap objek. Ia bersifat statis, dalam artian diagram kelas bukan menjelaskan apa yang terjadi jika kelas-kelasnya berhubungan, melainkan menjelaskan hubungan apa yang terjadi[31]. Contoh simbol-simbol class diagram dapat dilihat pada Tabel 2.7 berikut[30].

Tabel 2.6 Simbol Class Diagram

No	Simbol	Nama	Keterangan
1		Package	Package merupakan sebuah bungkus dari satu atau lebih kelas
2		Kelas	Kelas pada struktur sistem
3		Antarmuka / interface	Sama dengan konsep interface dalam pemrograman berorientasi objek
4		Asosiasi / association	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity
5		Asosiasi berarah	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity
6		Generalisasi	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
7		Kebergantungan / dependency	Relasi antar kelas dengan makna kebergantungan antar kelas

8		Agregasi / aggregation	Relasi antar kelas dengan makna semua-bagian (whole-part)
9		Composition	Relasi antar kelas yang menunjukkan bahwa suatu kelas memiliki bagian-bagian dari kelas lain. child class tidak dapat hidup tanpa parent class. Jika kelas utama dihapus, maka kelas-kelas bagian juga akan terhapus.