

BAB 2

TINJAUAN PUSTAKA

2.1 Absensi

Absen, menurut KBBI, adalah ketiadaan kehadiran seseorang di sebuah tempat. Absensi merupakan pencatatan kehadiran yang termasuk dalam proses pelaporan di sebuah lembaga. Sistem absensi memiliki peran krusial dalam rutinitas harian, khususnya di sekolah, perguruan tinggi, dan pabrik, kantor, rumah sakit dan lokasi-lokasi lainnya sebagai indikator kehadiran. Dengan majunya teknologi, umumnya sistem absensi di lingkungan pendidikan masih dilakukan secara konvensional. Masalah kesulitan dalam menentukan lokasi atau posisi Mahasiswa yang ingin melakukan absensi menjadi penting untuk didukung oleh teknologi [15].

2.2 GPS



Gambar 2.1 GPS

Sumber : <https://content.gps-trace.com>

Global Positioning System (GPS) (Greg Millete dan Adam Stroud, 2012) adalah suatu sistem navigasi dan penentuan lokasi yang dioperasikan. Tujuan dari sistem ini adalah untuk menyediakan informasi posisi, kecepatan dalam tiga dimensi, dan waktu secara terus-menerus di seluruh dunia, tanpa tergantung pada kondisi cuaca, kepada banyak pengguna secara bersamaan.

Setiap 24 jam, satelit GPS mengorbit Bumi dua kali dengan orbit yang presisi. Tiap satelit mengirimkan sinyal khas dan informasi orbit yang memungkinkan

perangkat GPS untuk menguraikan sinyal dan menentukan letak yang tepat dari satelit tersebut. Berikut cara mekanisme kerja dari GPS [6]:

1. GPS Menggunakan perhitungan “*triangulation*” dari satelit.
2. Untuk perhitungan, GPS pertama mengukur jarak menggunakan *travel* sinyal radio.
3. Untuk mengukur *travel time*, GPS memerlukan akurasi waktu yang tinggi.
4. Untuk pengukuran jarak diperlukan dengan pasti posisi satelit dan ketinggian orbitnya.
5. Dan pada akhirnya delay sinyal harus dikoreksi dikarenakan waktu perjalanan dari atmosfer sampai diterima oleh receiver.

2.3 Access Point



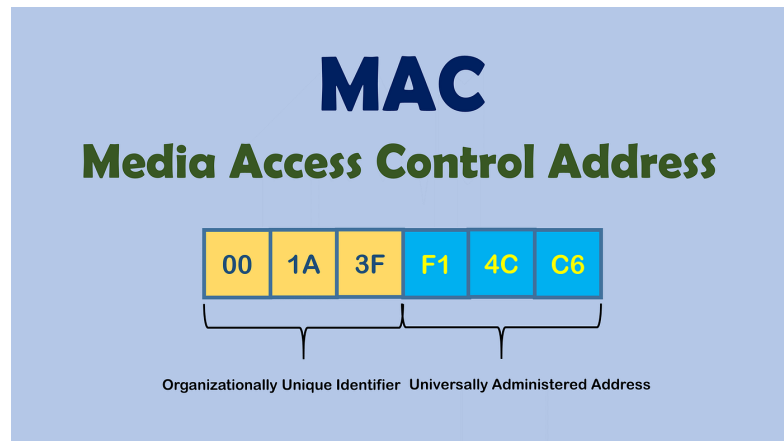
Gambar 2.2 Access Point

Sumber : <https://www.istockphoto.com>

Access Point adalah sebuah node yang telah diatur khusus dalam sebuah WLAN (Wireless Local Area Network). Fungsinya mirip dengan Hub/Switch yang bertugas menghubungkan jaringan lokal dengan klien wireless/nirkabel.

Access Point terdiri dari antena dan transceiver, berfungsi sebagai pemancar dan juga penerima sinyal untuk client server. Access Point tidak dapat mengatur aliran data seperti router, melainkan hanya menyambungkan atau memutuskan suatu perangkat yang mencoba terhubung ke dalam jaringan dari benar atau salahnya password yang dimasukkan pengguna [16].

2.4 MAC Address



Gambar 2.3 MAC Address

Sumber : <https://it.telkomuniversity.ac.id>

MAC Address (*Media Access Control Address*) adalah identifikasi unik pada sebuah antarmuka jaringan untuk memfasilitasi komunikasi melalui pengawasan pada tingkat jaringan. Ini juga merupakan alamat jaringan yang digunakan dalam lapisan data-link dari model OSI tujuh lapisan, mewakili sebuah node spesifik dalam jaringan. Dalam jaringan berbasis Ethernet, MAC Address terdiri dari 48-bit (6 byte) yang mengidentifikasi perangkat seperti komputer, antarmuka router, atau node lainnya dalam jaringan. MAC Address juga dikenal sebagai *Ethernet address*, *physical address*, atau *hardware address*. Selain itu, MAC Address memungkinkan perangkat dalam jaringan untuk saling berkomunikasi [5].

2.5 Google Maps API



Gambar 2.4 Logo Google Maps

Sumber : <https://maps.google.com>

Google Maps API adalah teknologi yang dikembangkan oleh Google untuk menyematkan Google Maps ke dalam aplikasi yang bukan dibuat oleh Google. Seiring dengan perkembangannya, Google Maps API memiliki kemampuan untuk

merender gambar peta statis, melakukan geocoding, dan memberikan petunjuk arah [17].

Google Maps API juga telah menjadi alat penting dalam industri teknologi dan bisnis. Banyak perusahaan mengandalkan API ini untuk meningkatkan pengalaman pengguna dengan menyediakan informasi lokasi yang akurat, navigasi yang tepat, dan visualisasi data yang mendalam.

2.6 Android



Gambar 2.5 Android

Sumber : <https://www.android.com>

Android adalah sebuah sistem operasi berbasis Linux yang berfungsi sebagai pengelola sumber daya perangkat keras, digunakan di ponsel, smartphone, dan PC tablet. Secara umum, Android merupakan platform terbuka (Open Source) yang memungkinkan pengembang untuk menciptakan aplikasi mereka sendiri yang dapat dijalankan di berbagai perangkat mobile. Telepon pertama yang menggunakan sistem operasi Android adalah HTC Dream, yang diluncurkan pada 22 Oktober 2008.

Pada akhir tahun 2009, diperkirakan ada setidaknya 18 jenis ponsel yang menggunakan Android di seluruh dunia. Sejak debutnya pada 9 Maret 2009, Android telah berkembang dengan versi 1.1, yang mencakup pembaruan estetis pada aplikasi seperti jam alarm, pencarian suara, pengiriman pesan melalui Gmail, dan pemberitahuan email. Selama perkembangannya, sistem operasi Android telah mengalami beberapa perbaikan dan perubahan. Yang menarik adalah penggunaan

nama-nama makanan untuk versi-versi Android yang dirilis. Berikut list perkembangan versi android dari awal sampat terkini [18]:

1. Android 1.0 dan 1.1 (Astro-Alpha dan Bender-Beta)

Kedua varian Android ini belum dirilis secara umum, sehingga masih terasa asing bagi sebagian orang. Kedua varian tersebut merupakan satu-satunya fase di mana mereka tidak menggunakan nama makanan manis. Google pertama kali memperkenalkan sistem operasi ini pada ponsel pintar HTC Dream, menjadikannya ponsel pertama yang menggunakan Android. Di versi ini, aplikasi Gmail, Youtube, Google Maps, dan Calendar sudah terpasang secara default. Play Store, tempat di mana pengguna dapat mengunduh dan memperbarui aplikasi, juga menjadi salah satu fitur yang sudah tersedia, menjadikan Android berbeda dari sistem operasi lainnya. Tujuan utamanya adalah membuat pengalaman menggunakan Android semakin ramah pengguna.

2. Android 1.5 (Cupcake)

Pada versi Android ini, yang merupakan versi ketiga dan memulai penggunaan penamaan nama makanan, dimulai dengan huruf "C" dan diberi nama "Cupcake". Beberapa fitur yang mengalami pembaharuan pada versi Android ini antara lain kemampuan merekam dan menonton video menggunakan kamera, penambahan fitur rotasi layar otomatis, kemampuan untuk terhubung melalui Bluetooth, penambahan beberapa widget baru, fitur untuk mengunggah video ke YouTube, dan adanya keyboard pihak ketiga pada layar.

3. Android 1.6 (Donut)

Pada versi Android ini, penggunaan dapat diperluas ke berbagai perangkat dengan berbagai ukuran layar. Beberapa fitur yang ditambahkan sangat berpengaruh, seperti adanya fitur pencarian baru di layar utama yang mempermudah pengguna dalam menggunakan smartphone mereka. Selain itu, ada juga dukungan jaringan berbasis CDMA yang memungkinkan ponsel untuk menggunakan layanan operator seluler dari berbagai negara. Fitur integrasi antara kamera dan galeri juga ditambahkan, memungkinkan

akses yang lebih cepat dari kamera ke galeri. Tambahan fitur teks ke ucapan dan lainnya juga turut hadir dalam versi ini.

4. Android 2.0-2.1 (Éclair)

Versi Android ini dilengkapi dengan fitur Google Maps, sebuah aplikasi peta lengkap dengan navigasi berputar dan panduan suara yang masih digunakan hingga saat ini. Selain itu, terdapat perubahan antarmuka pengguna (UI) dan browser baru dengan dukungan HTML5. Fitur tambahan termasuk dukungan untuk membuat lebih dari satu akun pada satu perangkat, live wallpaper, geser untuk membuka, kamus pintar yang dapat digunakan saat mengetik dengan keyboard, fitur kamera 3,2 MP dengan lampu kilat, zoom digital, dan Bluetooth versi 2.1.

5. Android 2.2 (Froyo/Frozen Yoghurt)

Fitur-fitur yang ditambahkan pada versi Android ini mencakup fitur Wifi hotspot portabel, pembaruan otomatis pada Android Market, kemampuan untuk memasang aplikasi ke kartu SD, dan peningkatan kinerja aplikasi hingga 2 hingga 5 kali lipat dari sebelumnya. Selain itu, terdapat juga fitur untuk meningkatkan keamanan ponsel, seperti penggunaan PIN dan kata sandi, fungsionalitas tethering USB, dan kemampuan Play Store untuk memperbarui aplikasi secara otomatis.

6. Android 2.3 (Gingerbread)

Gingerbread menghadirkan perubahan yang sangat signifikan pada Android, termasuk perubahan tampilan yang terkesan lebih modern pada ponsel. Perubahan antarmuka tidak hanya memperindah tampilan, tetapi juga membuatnya lebih sederhana dan responsif. Selain itu, terdapat peningkatan resolusi untuk mendukung layar yang lebih besar, penambahan fungsi NFC, peningkatan fungsi copy-paste, dukungan format video VP8 dan WebM, serta berbagai efek audio baru seperti reverb, equalization, dan bass boost. Android Gingerbread juga mendukung penggunaan lebih dari satu kamera.

Ponsel pertama yang dilengkapi dengan sistem operasi Gingerbread adalah Nexus S yang dikembangkan oleh Samsung. Beberapa fitur tambahan pada Android Gingerbread versi 2.3.3 termasuk fitur voice chat

dan video chat menggunakan Google Talk, pembaruan keamanan SSL, dan dukungan untuk Google Wallet pada beberapa ponsel.

7. Android 3.0/3.1 (Honeycomb)

Android ini dirancang khusus untuk meningkatkan pengalaman pengguna pada tablet dengan menambahkan beberapa fungsi yang hanya tersedia pada tablet. Perubahan signifikan terlihat pada penggunaan tombol nonfisik untuk navigasi, menggantikan tombol fisik seperti tombol home dan back yang digunakan sebelumnya. Pada versi ini, juga ditambahkan sistem bar yang memungkinkan pengguna untuk melihat notifikasi dan aksi yang dapat ditarik dari atas layar untuk menampilkan opsi dan fitur yang biasanya terdapat di bar aksi di bagian atas layar.

Android ini juga mendukung multi-processor dan memberikan perubahan warna tema pada smartphone serta memberikan tampilan preview pada widget. Fitur lain yang sangat berguna adalah kemampuan untuk mengenkripsi semua data pengguna guna meningkatkan keamanan ponsel.

8. Android 4.0 (Ice Cream Sandwich)

Beberapa fitur baru yang diperkenalkan pada versi Android ini meliputi pembukaan kunci dengan face recognition, pemantauan dan kontrol penggunaan data seluler, pencarian email secara offline, dan kemampuan berbagi informasi melalui NFC.

9. Android 4.1/4.2/4.3 (Jelly Bean)

Pada versi Android ini, fitur Google Now mulai diperkenalkan, yang membantu mempermudah pencarian langsung dari layar utama. Selain itu, fitur Photo Sphere juga hadir, yang memungkinkan pengguna untuk membuat foto panoramik yang imersif. Terdapat juga berbagai widget terbaru yang menambah fungsionalitas dan kemudahan penggunaan.

10. Android 4.4 (KitKat)

Versi Android ini memberikan optimisasi yang baik sehingga dapat mendukung perangkat smartphone dengan spesifikasi yang cukup rendah pada saat itu. Sebagai hasilnya, versi Android ini menjadi favorit di masanya. Fondasi dibuatnya smartphone flagship dengan spesifikasi yang

baik juga menggunakan versi ini. Selain itu, versi ini sudah mulai mendukung mode 64-bit, memungkinkan penggunaan RAM di atas 3GB.

11. Android 6.0 (Marshmallow)

Versi Android ini mulai memperkenalkan sistem sensor sidik jari sebagai fitur keamanan dan juga sistem pembayaran pada Google Play. Meskipun saat ini OS Android Nougat masih digunakan pada beberapa smartphone yang lebih lama, versi ini mengalami perubahan antarmuka dan juga mengimplementasikan mode layar terbagi, memungkinkan dua aplikasi untuk ditampilkan dalam satu layar secara bersamaan.

12. Android 8.0 & 8.1 (Oreo)

Oreo menjadi OS Android yang umum digunakan karena popularitasnya yang besar, dan mungkin Anda juga menggunakannya. Versi ini menjadi versi kedua setelah KitKat yang menggunakan merek dagang dari cemilan manis terkenal. OS Android ini memungkinkan pengguna untuk mendapatkan pembaruan lebih cepat.

13. Android 9.0 (Pie)

OS Android Pie memperkenalkan navigasi gesture yang mengubah tampilan tombol Home, Back, dan Recent Apps. Versi ini juga menyediakan berbagai fitur berguna seperti pengaturan kecerahan, pengaturan notifikasi, dan fitur screenshooting.

14. Android 10 (Q)

Versi Android ini masih terbatas pada sejumlah perangkat Android tertentu. Salah satu fitur yang ditawarkan dalam Android 10 adalah Dark Mode, yang diyakini dapat mengurangi konsumsi daya baterai. Selain itu, tema penggunaan nama-nama makanan manis telah dihentikan sejak versi Android ini.

15. Android 11

Sistem keamanan pada Android 11 lebih canggih dibandingkan dengan versi-versi sebelumnya. OS Android 11 juga memungkinkan screen recording tanpa perlu menggunakan aplikasi eksternal, serta menyediakan fitur screenshot panjang.

16. Android 12

Android 12 beta preview masih dalam tahap pengembangan dan membawa berbagai fitur baru yang belum ada pada versi sebelumnya, seperti One-Handed Mode, Customize Phone, dan Privacy Security Enhancement. Peningkatan fitur privasi menjadi mungkin dengan hadirnya Privacy Dashboard yang memberikan akses data pribadi seperti lokasi, penggunaan kamera, dan mikrofon dalam 24 jam terakhir.

Permintaan terhadap ponsel pintar terus meningkat dari tahun ke tahun, dan produsen akan terus bersaing dalam mengembangkan sistem operasi untuk ponsel. Android, sebagai sistem operasi dengan pangsa pasar sebesar 71,52%, memiliki

17. Android 13

Android 13 adalah versi terbaru sistem operasi Android yang diperkenalkan pada Februari 2022, sekitar 4 bulan setelah perilisan Android 12. Versi ini berfokus pada penyempurnaan fitur yang diperkenalkan dalam Android 12L, serta peningkatan privasi, keamanan, dan optimalisasi antarmuka pengguna.

Salah satu fitur baru yang diperkenalkan dalam Android 13 adalah pemilih foto baru yang memungkinkan pengguna untuk mengatur foto atau video tertentu yang dapat dibagikan dengan aplikasi. Selain itu, pengguna juga dapat menyesuaikan ikon aplikasi dengan warna yang sama seperti tema atau wallpaper yang digunakan.

2.7 Android Studio



Gambar 2.6 Android Studio

Sumber : <https://uxwing.com>

Android Studio adalah lingkungan pengembangan terintegrasi (Integrated Development Environment/IDE) resmi yang disediakan oleh Google untuk pengembangan aplikasi Android. Perangkat lunak ini digunakan oleh para pengembang untuk membuat aplikasi Android dengan berbagai fitur yang mendukung pengembangan, pengujian, dan distribusi aplikasi.

Android Studio menyediakan berbagai alat yang mempermudah pengembangan, termasuk editor kode yang cerdas, pengelola proyek, emulator Android bawaan, penyusun antarmuka pengguna (UI designer), pengujian otomatis, serta integrasi dengan layanan Google seperti Firebase untuk analitik, otentikasi, dan penyimpanan data.

Dengan Android Studio, pengembang dapat membuat aplikasi Android dari awal, mulai dari merancang antarmuka pengguna hingga menulis kode, menguji aplikasi dalam berbagai perangkat, dan mempublikasikannya ke Google Play Store. Ini merupakan lingkungan yang sangat populer dan kuat untuk pengembangan aplikasi Android [1].

2.8 Kotlin



Gambar 2.7 Kotlin

Sumber : <https://kotlinlang.org>

Kotlin adalah bahasa pemrograman modern yang diketik secara statis dan berjalan di platform Java Virtual Machine (JVM). Dengan menggunakan compiler LLVM, Kotlin dapat dikompilasi menjadi kode JavaScript. Bahasa pemrograman ini sangat populer di kalangan developer [19]. Kotlin disajikan secara statis dan bisa dioperasikan bersamaan (interoperable) dengan bahasa pemrograman Java, karena keduanya memiliki kompilasi kode byte yang sama. Artinya, kedua bahasa ini bisa saling berbagi dan menggunakan informasi, serta memudahkan migrasi antara aplikasi Java ke Kotlin atau sebaliknya.

Pada awalnya, Kotlin dirancang untuk meningkatkan bahasa pemrograman Java dan dioperasikan pada Java Virtual Machine (JVM), Java Class Libraries, dan Android. Namun karena Kotlin terintegrasi dengan Java, bahasa pemrograman ini jadi semakin populer digunakan untuk mengembangkan berbagai aplikasi.

2.9 PHP



Gambar 2.8 PHP

Sumber : <https://www.php.net>

PHP pertama kali dikembangkan oleh Rasmus Lerdoff pada tahun 1994 untuk memantau resume online dan informasi pribadi terkait, yang awalnya dinamai "Personal Home Page". Namun, dua programmer Zeev Suraski dan Andi Gutmans membangun ulang, memperbarui, dan merilis inti PHP pada tahun 1997 serta mengubah akronim PHP menjadi "Hypertext Processor".

PHP telah berkembang dan digunakan sebagai bahasa untuk World Wide Web (WWW) atau Internet, di mana pengembang menemukan PHP sebagai bahasa yang mudah dipelajari, ramah komunitas, tersedia secara gratis sebagai perangkat lunak open source, dan mudah diterapkan. Lingkungan PHP saat ini mengharuskan pengembang untuk membuat komponen antarmuka sistem, menghubungkan ke database, dan otentikasi pengguna. Penggunaan kerangka kerja dapat mengatasi masalah selama siklus pengembangan dengan penggunaan ulang kode yang dapat menghemat waktu dan biaya untuk desain, pengembangan kode, dan pengujian [20].

2.10 Laravel



Gambar 2.9 Laravel

Sumber : <https://academy.alterra.id>

Laravel adalah framework PHP yang sangat berguna untuk membuat aplikasi web dengan mudah. Menggunakan pola desain MVC (model–view–controller) dan berbasis sistem Symfony, Laravel memanfaatkan sistem paket modular yang memungkinkan pengembangan aplikasi dengan modul baru. Framework ini

menggunakan berbagai komponen yang sudah ada dari framework lain, yang membantu menciptakan aplikasi yang aman dan operasional dengan cepat. Beberapa fitur dan keunggulan Laravel termasuk dukungan untuk berbagai database, menyediakan utilitas untuk pengembangan aplikasi web, mesin routing yang sederhana dan cepat, serta skala aplikasi yang lebih besar.

Laravel adalah perangkat lunak open-source yang dilisensikan oleh MIT, dan sangat cocok untuk programmer pemula maupun berpengalaman. Dengan menggunakan metode PHP berorientasi objek modern, Laravel dapat mengurangi waktu pengembangan aplikasi web dan mempercepat proses pemasaran. Sintaks yang ekspresif dan fungsi modern dari Laravel menarik bagi pengembang yang ingin membuat aplikasi yang kuat, serta menyediakan lapisan abstraksi database yang ekspresif, injeksi ketergantungan, dan skala yang tinggi [21].

2.11 XAMPP



Gambar 2.10 XAMPP

Sumber : <https://badoystudio.com>

XAMPP adalah perangkat lunak bebas yang mendukung banyak sistem operasi, merupakan campuran dari beberapa program. Perangkat lunak ini berfungsi sebagai server yang berdiri sendiri (localhost), terdiri dari program MySQL database, Apache HTTP Server, dan penerjemah yang ditulis dalam bahasa pemrograman PHP dan Perl. Nama XAMPP adalah singkatan dari X (empat sistem operasi), Apache, MySQL, PHP, dan Perl [22].

2.12 MySQL

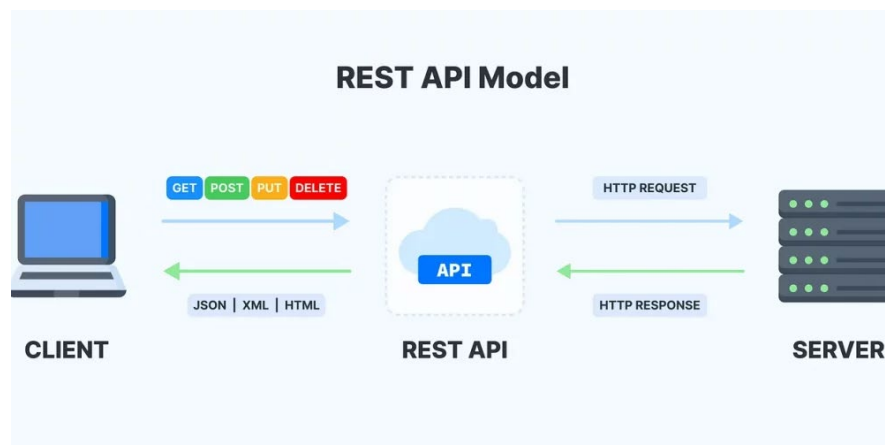


Gambar 2.11 MySQL

Sumber : <https://www.pngwing.com>

MySQL merupakan Database Management System (DBMS) tools open source yang mendukung multiuser, multithreaded, populer, dan free. Berdasarkan teori di atas, maka dapat disimpulkan bahwa SQL adalah bahasa permintaan database tertentu dimana subbahasa dapat membuat dan memanipulasi data di dalam database. SQL digunakan untuk melakukan tugas-tugas seperti melakukan update terhadap database, yang merujuk pada konsep Relational Database Management System (RDBMS) [23].

2.13 REST API



Gambar 2.12 REST API

Sumber : <https://course-net.com>

REST API adalah salah satu arsitektur yang digunakan dalam API (Application Programming Interface). REST, atau Representational State Transfer, menggunakan HTTP (Hypertext Transfer Protocol) sebagai media pertukaran data. Oleh karena itu, banyak aplikasi yang dikembangkan oleh para pengembang menggunakan REST API. Aplikasi yang menggunakan REST API diharapkan dapat melakukan pertukaran data secara efektif. Performa penggunaan REST API diharapkan cepat dan mudah untuk dikembangkan.

Cara kerja dari REST API melibatkan penggunaan Rest client yang bertugas untuk mengakses data dari Rest server. Proses akses data ini dapat berbeda-beda tergantung pada URLs atau Global ID yang digunakan. Dengan perbedaan dalam akses data, output yang dihasilkan juga dapat berbeda. Rest server dapat memberikan format output dalam bentuk XML (Extensible Markup Language), teks, atau bahkan JSON (JavaScript Object Notation). Setiap format tersebut biasanya memiliki kecepatan loading atau akses yang sedikit berbeda-beda. Arsitektur REST API ada tiga, dan masing-masing adalah hal yang berbeda berikut perbedaannya [24].

2.13.1 API

API (Application Programming Interface) adalah sebuah program yang digunakan untuk menghubungkan antara aplikasi satu dengan yang lainnya. Fungsinya adalah untuk memungkinkan pengiriman dan penerimaan data sehingga aplikasi dapat bertukar informasi. Dengan adanya API, sebuah aplikasi dapat bekerja secara efektif dan efisien.

2.13.2 REST API

REST API yaitu salah satu arsitektur yang ada di API (Application Programming Interface). Untuk REST atau Representational State Transfer sendiri menggunakan HTTP (Hypertext Transfer Protocol) sebagai media pertukaran data. Sehingga beberapa aplikasi tertentu yang sedang developer kembangan menggunakannya.

Aplikasi – aplikasi yang memakai hal tersebut nantinya yang mereka harapkan dapat melakukan pertukaran data secara efektif. Kemudian performa

setiap penggunaannya akan terasa cepat dan bisa dengan mudah nantinya ketika mengembangkannya. Berikutnya akan menjelaskan cara kerja dari REST API ini.

2.13.3 RESTful Api

Untuk RESTful API, penerapannya merupakan bagian dari Application Programming Interface (API). Dalam RESTful API, terdapat empat komponen yang sangat penting. Komponen-komponen tersebut adalah:

- 1. URL Design**

Nantinya akan terjadi pengaksesan menggunakan protokol HTTP (Hypertext Transfer Protocol). Pengembang aplikasi akan mudah memahaminya karena ada nama dan struktur URL yang sangat baik.

- 2. HTTP Verbs**

Dalam melakukan permintaan (request), terdapat beberapa metode yang digunakan agar server memahami permintaan dari klien.

- 3. HTTP Response Code**

Merupakan kode yang digunakan untuk menyampaikan hasil permintaan kepada klien. Terdapat tiga jenis kode respons, yaitu 2xx untuk keberhasilan, 4xx untuk kesalahan klien, dan 5xx untuk kesalahan server.

- 4. Format Response**

Ketika permintaan dari klien telah diproses, klien akan menerima data dari server. Setelah menerima respons, berupa data dalam format JSON atau XML, klien dapat menggunakannya sesuai kebutuhan.

Kemudian terdapat juga metode-metode dalam REST API, berikut metode HTTP yang biasa digunakan dalam REST API meliputi:

- 1. GET**

Metode pertama adalah GET. Metode ini merupakan salah satu yang paling sederhana dan langsung. GET digunakan untuk mengambil data atau informasi yang ada. Metode ini sangat penting karena menjadi kunci keberhasilan dalam setiap prosesnya.

- 2. POST**

Metode selanjutnya adalah POST. POST adalah metode HTTP yang digunakan untuk membuat data baru. Data tersebut dimasukkan ke dalam body saat proses permintaan (request).

3. PUT

Metode berikutnya adalah PUT. PUT digunakan untuk melakukan pembaruan data pada sumber daya (resource). Pembaruan ini penting untuk memastikan bahwa proses-proses yang sedang berlangsung dapat terselesaikan dengan baik.

4. DELETE

Metode berikutnya adalah DELETE yang digunakan untuk menghapus data tertentu pada sumber daya (resource). Pada metode ini, beberapa sumber daya tertentu pada URL akan dihapus atau dihilangkan.

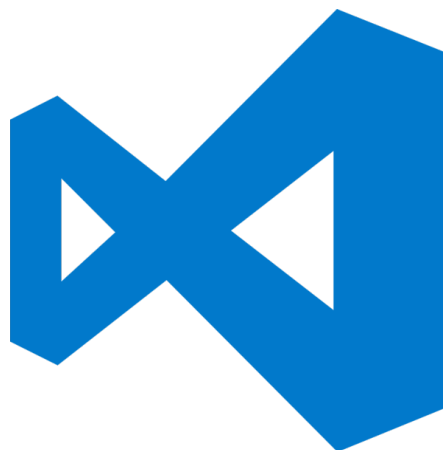
5. PATCH

Kemudian adalah PATCH, yang hampir sama dengan PUT dalam hal pembaruan data yang sudah ada. Namun, perbedaannya terletak pada fakta bahwa PATCH hanya melakukan pembaruan beberapa field dalam sebuah record, tidak seperti PUT yang memperbarui seluruh field.

6. OPTIONS

Metode OPTIONS akan berfungsi sebagai cara untuk mendapatkan daftar operasi yang didukung oleh sumber daya dari server.

2.14 Visual Studio Code (VSCode)



Gambar 2.13 Visual Studio Code

Sumber : <https://freebiesupply.com>

Visual Studio Code adalah editor kode sumber yang ringan namun kuat yang berjalan di desktop Anda dan tersedia untuk Windows, macOS, dan Linux. Editor ini dilengkapi dengan dukungan bawaan untuk JavaScript, TypeScript, dan Node.js serta memiliki ekosistem ekstensi yang kaya untuk bahasa dan runtime lainnya (seperti C++, C#, Java, Python, PHP, Go, .NET) [25].

2.15 Git



Gambar 2.14 Git

Sumber : <https://www.pngwing.com>

Git merupakan software berbasis Version Control System (VCS) yang bertugas untuk mencatat perubahan seluruh file atau repository suatu project. Developer software biasa menggunakan Git untuk distributed revision (VCS terdistribusi), hal ini bertujuan untuk menyimpan database tidak hanya ke satu tempat. Namun semua orang yang terlibat dalam penyusunan kode dapat menyimpan database ini. Prosedur yang diterapkan ini dapat membantu antar divisi project untuk memantau dan menghubungkan (merge) antar ekstensi yang berbeda dengan mudah. Sehingga aplikasi yang dibuat oleh sebuah tim project dapat berfungsi tanpa menghubungkan secara manual [26].

2.16 Postman



Gambar 2.15 Postman

Sumber : <https://apievangelist.com/>

Postman adalah aplikasi komputer yang digunakan untuk menguji API. Postman mengirimkan permintaan API ke server web dan menerima respons, apa pun itu. Tidak ada pekerjaan tambahan atau pengaturan kerangka kerja yang diperlukan saat mengirim dan menerima permintaan menggunakan Postman. Alat ini sering digunakan oleh penguji dan pengembang untuk meningkatkan pengujian aplikasi. Mudah diintegrasikan dengan Continuous Integration (CI) & Continuous Development Pipeline Anda.

Postman, dengan beragam fitur dan kemudahan penggunaannya, telah digunakan oleh jutaan penguji. Dengan antarmuka yang sederhana dan ramah pengguna, Anda dapat dengan mudah mengirim permintaan, mengisi data yang diperlukan, memilih metode HTTP, dan menekan tombol "Kirim". Salah satu fitur yang paling sering digunakan adalah Otomatisasi, yang memungkinkan Anda mengatur pengujian dan menulis rangkaian pengujian [27].

2.17 Unified Modelling Language (UML)

Menurut Sumirat, UML adalah bahasa berbasis grafik/gambar yang digunakan untuk memvisualisasikan, mendefinisikan, membangun, dan mendokumentasikan sistem perangkat lunak berorientasi objek (OO). UML tidak hanya sekadar bahasa pemrograman visual, tetapi juga dapat langsung terhubung ke beberapa bahasa pemrograman seperti JAVA, C++, Visual Basic, atau bahkan terhubung langsung ke database berorientasi objek [28]. Pemodelan menggunakan UML memiliki sejumlah keuntungan, yaitu :

1. UML adalah bahasa formal di mana setiap elemen memiliki makna spesifik, sehingga tidak ada kebingungan.
2. Memiliki representasi yang sederhana dan tidak rumit.
3. Mampu menggambarkan komponen-komponen utama dari suatu sistem.
4. Dapat digunakan untuk menangani sistem besar maupun kecil.
5. Banyak digunakan oleh berbagai kalangan, sehingga tersedia banyak tutorial cara menggunakannya.
6. UML adalah bahasa standar, oleh karena itu UML tidak terbatas pada produk tertentu.

Berikut adalah daftar diagram UML beserta penjelasannya:

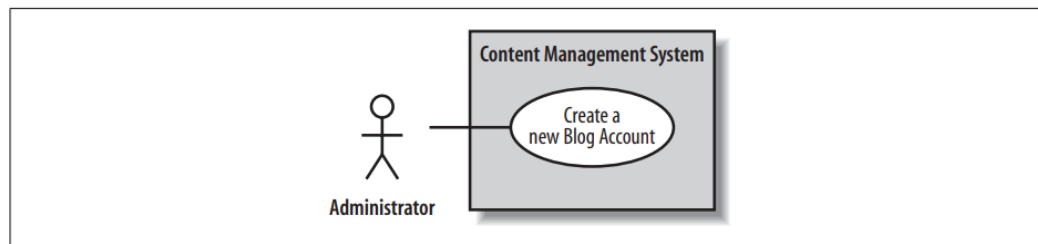
1. Use Case: Menggambarkan interaksi antara sistem dengan pengguna atau sistem lain, sangat berguna untuk perencanaan kebutuhan atau persyaratan sistem.
2. Activity: Menggambarkan fungsi sekuensial dan paralel dalam sistem.
3. Class: Menggambarkan kelas, tipe, antarmuka, dan hubungan di antara mereka.
4. Sequence: Menggambarkan interaksi antar objek di mana urutan interaksi sangat penting.

2.17.1 Use Case Diagram

Use case merupakan sebuah skenario atau situasi di mana sistem Anda digunakan untuk memenuhi kebutuhan pengguna. Ini menggambarkan fungsionalitas atau fitur spesifik yang disediakan oleh sistem untuk memenuhi kebutuhan pengguna tersebut. Dalam pengembangan perangkat lunak, use case sering digunakan sebagai alat untuk memodelkan interaksi antara pengguna dan sistem. Hal ini membantu dalam memahami kebutuhan pengguna secara lebih mendalam serta merancang solusi yang sesuai untuk memenuhi kebutuhan tersebut [29].

Use case adalah titik awal yang sangat baik untuk hampir semua aspek pengembangan sistem berbasis objek, perancangan, pengujian, dan dokumentasi. Mereka menggambarkan kebutuhan sistem secara ketat dari perspektif eksternal,

dan mendefinisikan nilai yang diberikan oleh sistem kepada pengguna [29]. Contoh use case dapat dilihat pada gambar berikut.



Gambar 2.16 Use Case Diagram

Sumber : Learning UML 2.0 [29]

Jika melihat gambar 2.19, Use Case Diagram memiliki beberapa elemen dasar penting, yaitu:

1. Aktor, juga dikenal sebagai actor, merupakan entitas dari luar sistem yang berinteraksi dengan sistem. Ini bisa berupa individu, kelompok, atau entitas lain yang menggunakan atau berinteraksi dengan sistem tersebut.
2. Use case adalah deskripsi fungsi atau tugas yang dapat dilakukan oleh sistem. Setiap use case menggambarkan skenario yang melibatkan interaksi antara aktor (pengguna atau entitas eksternal lain) dengan sistem untuk mencapai tujuan tertentu.
3. Garis komunikasi, atau communication line, adalah garis yang menunjukkan hubungan atau relasi antara aktor dan use case dalam diagram use case. Garis ini mencerminkan interaksi antara aktor dan use case serta menggambarkan keterkaitan di antara keduanya. Garis ini juga memperlihatkan hubungan "include" dan "extend" yang menunjukkan bagaimana use case saling terhubung dan dapat memperluas atau menyertakan fungsi tambahan.

Deskripsi kasus penggunaan adalah dokumen teks yang memberikan detail lengkap mengenai bagaimana suatu kasus penggunaan bekerja dalam sebuah sistem perangkat lunak. Dokumen ini mencakup informasi seperti aktivitas yang dilakukan dalam kasus penggunaan, siapa yang terlibat (aktor), dan bagaimana sistem merespons dalam situasi-situasi tertentu. Berikut ini adalah contoh dari deskripsi kasus penggunaan:

Use case name	Create a new Blog Account
Related Requirements	Requirement A.1.
Goal In Context	A new or existing author requests a new blog account from the Administrator.
Preconditions	The system is limited to recognized authors and so the author needs to have appropriate proof of identity.
Successful End Condition	A new blog account is created for the author.
Failed End Condition	The application for a new blog account is rejected.
Primary Actors	Administrator.
Secondary Actors	Author Credentials Database.
Trigger	The Administrator asks the CMS to create a new blog account.

Gambar 2.17 Use Case Description

Sumber: Learning UML 2.0 [29]

Main Flow	Step	Action
	1	The Administrator asks the system to create a new blog account.
	2	The Administrator selects an account type.
	3	The Administrator enters the author's details.
	4	The author's details are verified using the Author Credentials Database.
	5	The new blog account is created.
	6	A summary of the new blog account's details are emailed to the author.
Extensions	Step	Branching Action
	4.1	The Author Credentials Database does not verify the author's details.
	4.2	The author's new blog account application is rejected.

Gambar 2.18 Use Case Description

Sumber: Learning UML 2.0 [29]

Jika dilihat pada gambar 2.20 dan 2.21 di atas, terdapat beberapa elemen yang ada dalam use case description. Berikut adalah penjelasan mengenai elemen – elemen yang ada di gambar tersebut:

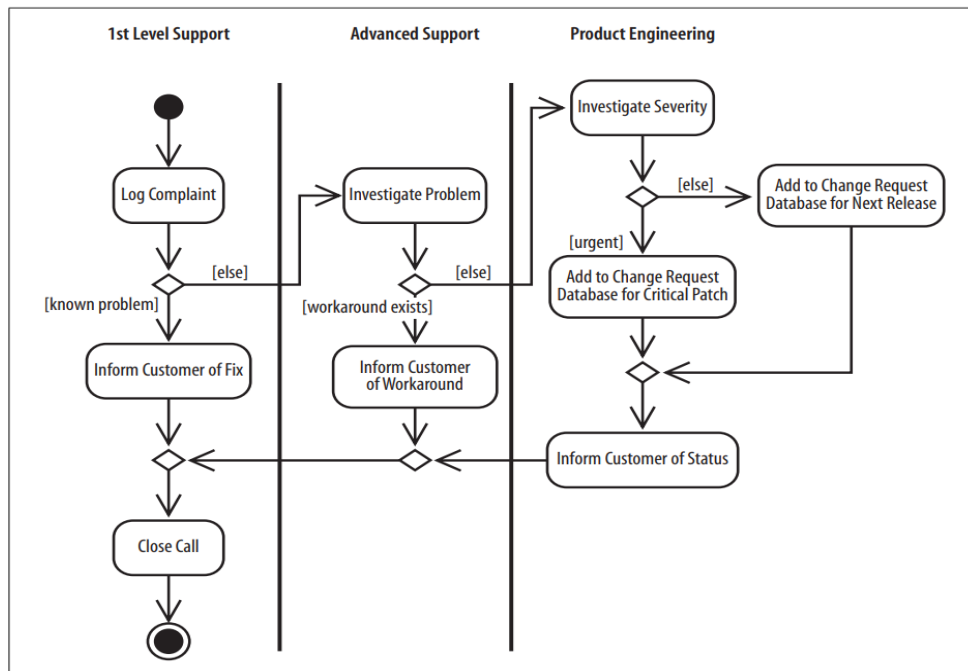
1. *Use Case Name*, nama use case, adalah identifikasi kasus penggunaan yang jelas di bagian paling atas dokumen.
2. *Related Requirements*, persyaratan terkait, adalah kode identifikasi yang menunjukkan persyaratan mana yang dipenuhi oleh kasus penggunaan ini.
3. *Goal in Context*, tujuan dalam konteks, adalah tujuan utama yang harus dicapai oleh suatu use case.
4. *Preconditions*, prekondisi, adalah kondisi yang harus terpenuhi sebelum use case dimulai.
5. *Successful End Condition*, kondisi akhir berhasil, adalah kondisi sistem setelah eksekusi sukses dari use case.
6. *Failed End Condition*, kondisi akhir gagal, adalah kondisi sistem setelah eksekusi gagal dari use case.
7. *Primary Actors*, aktor utama, adalah aktor atau entitas eksternal yang terlibat dalam kasus penggunaan ini.
8. *Secondary Actors*, aktor sekunder, adalah aktor tambahan atau sistem lain yang juga terlibat dalam kasus penggunaan ini, jika ada.
9. *Trigger*, pemicu, adalah kondisi yang dipicu oleh aktor yang memicu eksekusi use case.
10. *Main Flow*, aliran utama, adalah rangkaian langkah-langkah yang terjadi dalam skenario normal atau ideal dari kasus penggunaan.
11. *Extension*, adalah bagian yang digunakan untuk menjelaskan langkah-langkah alternatif dari alur utama yang telah dijelaskan.

2.17.2 Activity Diagram

Activity Diagram menunjukkan berbagai alur kerja dari sistem yang sedang dirancang, termasuk bagaimana setiap alur dimulai, keputusan yang diambil, dan bagaimana alur tersebut berakhir. Diagram aktivitas juga dapat menggambarkan proses paralel yang terjadi di berbagai disiplin ilmu. Di sisi lain, diagram fungsional adalah diagram keadaan diskrit di mana sebagian besar keadaannya berupa fungsi, dan transisi utamanya terjadi setelah kondisi sebelumnya (proses internal) selesai [30].

Activity Diagram digunakan untuk mengilustrasikan langkah-langkah atau fungsi dari suatu sistem. Setiap kasus memiliki satu atau lebih diagram aktivitas.

Diagram ini menggambarkan proses bisnis dan urutan aktivitas dalam proses tersebut [30]. Berikut ini adalah contoh dari diagram aktivitas:



Gambar 2.19 Activity Diagram

Sumber: Learning UML 2.0 [29]

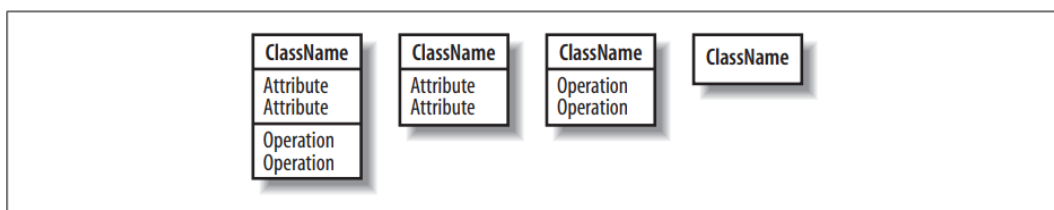
Terdapat beberapa elemen pada gambar 2.22 di atas, berikut adalah penjelasan setiap elemen - elemen yang ada di gambar:

1. *Initial Node*, menandakan titik awal dari aktivitas dalam proses. Awal ini direpresentasikan oleh sebuah lingkaran yang berisi tanda panah.
2. *Action*, merupakan suatu proses yang tidak dapat dibagi lagi menjadi subaktivitas lebih lanjut.
3. *Control Flow*, menggambarkan urutan atau alur dari satu aktivitas ke aktivitas lainnya dalam proses. Aliran kontrol direpresentasikan oleh panah.
4. *Fork*, menandakan titik dalam proses di mana dua atau lebih jalur aktivitas terpisah dimulai secara bersamaan. Garpu direpresentasikan oleh sebuah garis yang bercabang.
5. *Join*, menandakan titik dalam proses di mana dua atau lebih jalur aktivitas yang terpisah bergabung menjadi satu jalur tunggal. Gabungan direpresentasikan oleh sebuah garis yang bersatu.

6. *Final Node*, menandakan titik akhir dari aktivitas dalam proses. Akhir ini direpresentasikan oleh sebuah lingkaran yang berisi tanda lingkaran atau tanda silang.

2.17.3 Class Diagram

Class diagram atau diagram kelas adalah salah satu jenis diagram struktural dalam UML yang digunakan untuk menggambarkan struktur statis dari suatu sistem. Diagram ini menggambarkan kelas-kelas, properti, metode, dan hubungan antar objek. Class diagram tidak menggambarkan perilaku dinamis atau interaksi antar objek, tetapi fokus pada struktur statis system [28]. Berikut adalah contoh dari diagram kelas:



Gambar 2.20 Class Diagram

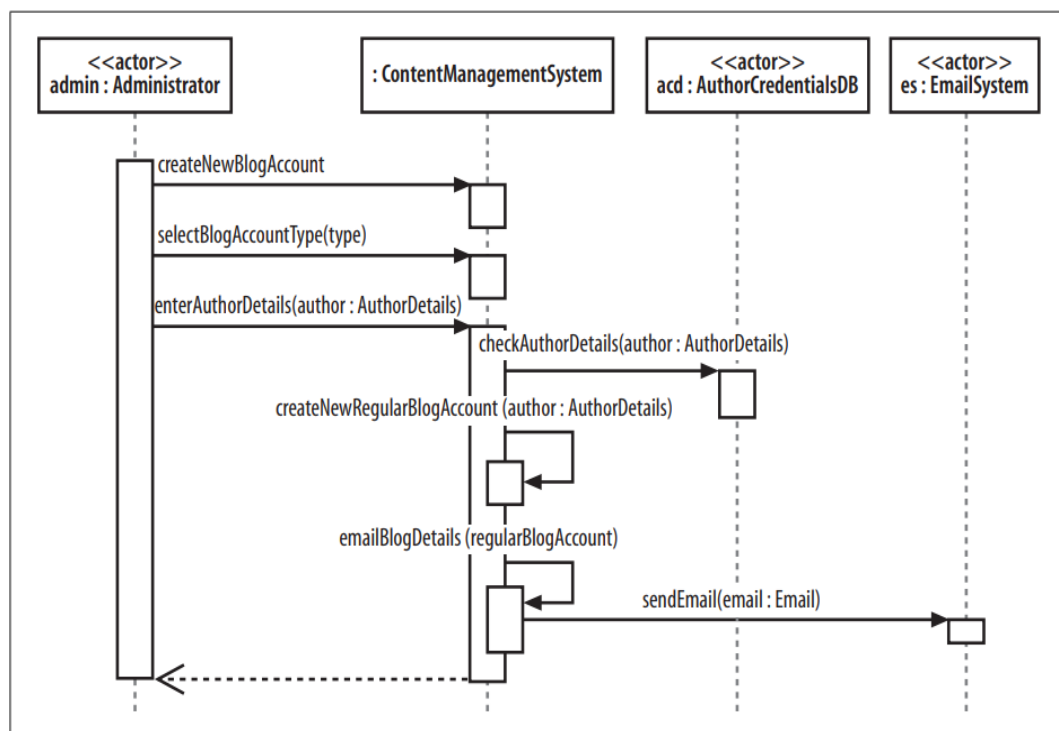
Sumber: Learning UML 2.0 [29]

Adapun elemen-elemen penting dalam class diagram antara lain:

1. *Class* adalah representasi dari entitas atau objek dalam sistem yang memiliki atribut dan metode.
2. *Attribute* adalah data atau informasi yang dimiliki oleh sebuah kelas.
3. *Method* adalah tindakan atau perilaku yang dapat dilakukan oleh sebuah kelas.
4. *Relationship* adalah hubungan antar kelas yang menunjukkan bagaimana kelas-kelas saling berinteraksi atau terkait satu sama lain dalam sistem.
5. *Association* adalah hubungan yang menunjukkan keterhubungan antara dua kelas.
6. *Aggregation* adalah jenis hubungan yang menunjukkan bagaimana bagian-bagian dari suatu kelas terhubung dengan kelas lain.
7. *Inheritance* adalah hubungan antara kelas yang menunjukkan pewarisan atau sub-kelas.

2.17.4 Sequence Diagram

Sequence diagram atau diagram urutan adalah sebuah representasi grafis yang digunakan untuk menggambarkan interaksi antara objek-objek dalam sebuah sistem dengan detail. Diagram ini menunjukkan bagaimana objek-objek saling berinteraksi melalui pertukaran pesan atau perintah, serta mencantumkan waktu eksekusi masing-masing pesan atau perintah tersebut. Sebagai contoh, berikut adalah sebuah sequence diagram:



Gambar 2.21 Sequence Diagram

Sumber: Learning UML 2.0 [29]

Berikut adalah penjelasan tentang elemen-elemen penting yang terdapat dalam diagram urutan (sequence diagram):

1. *Object*, merupakan representasi dari entitas dalam sistem yang terlibat dalam interaksi, digambarkan sebagai kotak vertikal dengan nama objek di bagian atas.
2. *Activation*, adalah saat di mana objek aktif dalam proses interaksi, ditunjukkan dengan garis vertikal yang menghubungkan objek dengan garis hidup.

3. *Message*, adalah pesan yang dikirim antara objek untuk memicu tindakan atau proses, digambarkan sebagai panah dengan nama pesan dan parameter.
4. *Control Fragment*, adalah fragmen untuk menggambarkan struktur kontrol seperti loop, kondisional, atau proses interaksi paralel.
5. *External Actor*, adalah entitas eksternal yang berinteraksi dengan sistem, digambarkan sebagai kotak vertikal dengan nama aktor.
6. *System Boundary*, adalah batas yang membatasi objek-objek sistem dari aktor eksternal atau lingkungan eksternal lainnya.
7. *Activity*, adalah tindakan yang dilakukan oleh objek dalam sistem, digambarkan sebagai kotak vertikal dengan nama aktivitas.
8. *Lifeline*, adalah garis vertikal yang menunjukkan waktu ketika objek aktif dalam proses interaksi, terhubung dengan aktivasi.