

BAB 2

TINJAUAN PUSTAKA

2.1. Aplikasi

Menurut Hasan Abdurahman dan Asep Ririh Riswaya (2014), Aplikasi adalah Program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkann hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu tehnik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkann atau diharapakann maupun pemrosesan data yang diharapkan. [6]

2.2. Wearable Device

Wearable device atau *perangkat wearable* merupakan produk teknologi yang melekat atau dapat di tempelkan di tubuh manusia. Beberapa contoh wearable device adalah seperti kacamata, jam tangan, baju, badgets, gelang, atau sepatu yang mana di dalam perangkat wearable tersebut telah di tambahkan sentuhan teknologi canggih. [7]

Awal mula perangkat wearable mulai pada era 80-an. Saat itu, terdapat sebuah jam yang dilengkapi dengan kalkulator. Namun, istilah wearable sendiri baru populer setelah 2010, saat beberapa perusahaan teknologi seperti Google, Apple, dan Samsung turut mengeluarkan produk tersebut.

Jam kalkulator “pulsar”, Adalah Jam tangan pertama di dunia yang dirilis ke publik pada akhir tahun 1975 dan langsung mendapatkan popularitas dengan slogan “untuk pria yang memiliki segalanya sampai sekarang”. Presiden gerald ford dilaporkan menyatakan minatnya di harga \$ 3.950 untuk edisi terbatas, yang memicu pemberitaan sebagian kecil media massa.

Tiga belas tahun setelah menciptakan perangkat ransel fotografi, steve mann menciptakan webcam nirkabel yang dapat dipakai dan mulai meng-upload gambar ke web, sampai tahun 1996. Menurut mann, proyek ke 2 ini adalah kamera yang digunakan untuk tampil di web.

Selanjutnya pada tahun 2006 Nike dan Apple bekerja sama untuk menciptakan nike + ipod, kit sport yang memungkinkan pengguna melakukan sinkronisasi gerakan kedalam iPod mereka. dalam kerjasama dengan mitranya, nike merilis beberapa item pakaian dengan kantong khusus dirancang untuk ipod.

2.3. Smartwatch

Smartwatch adalah teknologi terbaru jam tangan yang sedang dimintati oleh banyak orang karena dapat berfungsi seperti *smartphone* yang banyak kegunaannya.[8]

Secara umum *smartwatch* diketahui sebagai perangkat pintar yang biasanya terhubung dengan *smartphone* untuk menjalankan beberapa fungsi disamping fungsinya sebagai jam tangan. Smartwatch dirancang untuk jaringan dan mengintegrasikan dengan perangkat pribadi lainnya melalui wifi dan teknologi *bluetooth*. Melalui *smartwatch* pengguna dapat mengakses berita, cuaca, olahraga, telepon, pesan, dan bahkan pada versi *smartwatch* terbaru pengguna dapat mengetahui informasi kesehatan badannya. *Smartwatch* menggunakan bacaan *biometric* melalui sensor *byonim* untuk mengidentifikasi detak jantung pengguna dan mengkalkulasi jumlah detak jantungnya, *Smartwatch* tidak dirancang untuk menggantikan fungsi ponsel, tetapi dirancang untuk memberi peringatan pada pengguna untuk pesan dan panggilan yang masuk pada ponsel.

2.4. Mi Band 3

Mi Band 3 merupakan Wearable device Mi Band 2 yang dirilis tahun 2016 lalu. Mi Band 3 berukuran lebih besar dan resolusi layarnya pun lebih tinggi (128 x 80 piksel) dengan panel OLED berukuran 0,78 inci. Mi Band 3 memiliki baterai berkapasitas 110 mAh, 50 persen lebih besar dari Mi Band 2. Xiaomi mengklaim baterai itu membuat Mi Band 3 mampu membuatnya menyala hingga 20 hari. Mi Band 3 memiliki fitur menjawab panggilan telepon langsung dari gelang pintar itu, tanpa harus menyentuh smartphone. Bluetooth 4.2 dan NFC untuk mobile payment secara nirkabel dimiliki oleh Mi Band 3. Ketahanan Mi Band 3 terhadap air hingga kedalaman 50 meter.[9]

2.5. Android

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat seluler layar sentuh seperti telepon pintar dan komputer tablet. Android awalnya dikembangkan oleh Android, Inc., Dengan dukungan finansial dari Google, yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007.[10]

2.6. Android SDK

Android Software Development Kit (SDK) merupakan *kit* yang bisa digunakan oleh para *developer* untuk mengembangkan aplikasi berbasis Android. Di dalamnya, terdapat beberapa *tools* seperti *debugger*, *software libraries*, *emulator*, dokumentasi, *sample code* dan tutorial.

Java SE Development kit adalah salah satu contoh Android SDK dan menjadi bahasa pemrograman yang paling sering digunakan untuk mengembangkan aplikasi Android. Di samping itu ada beberapa bahasa lainnya seperti C++, Go, dan Kotlin -bahasa yang ditetapkan Google pada tahun 2017 lalu.

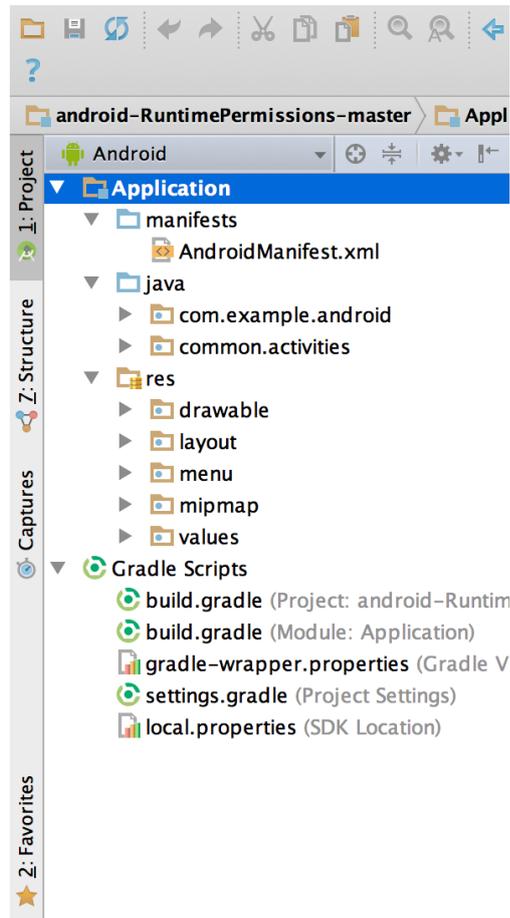
2.7. Android Studio

Android Studio adalah *Integrated Development Environment* (IDE) resmi untuk pengembangan *android*, yang didasarkan pada IntelliJ IDEA. Selain sebagai editor kode dan fitur *developer* IntelliJ yang andal, *Android Studio* menawarkan banyak fitur yang meningkatkan produktivitas dalam membuat aplikasi *android*, seperti [10]:

- a. Sistem versi berbasis Gradle yang fleksibel
- b. *Emulator* yang cepat dan kaya fitur
- c. Lingkungan terpadu untuk pengembangan bagi semua perangkat *Android*.
- d. Menerapkan perubahan untuk melakukan push pada perubahan kode dan *resource* ke aplikasi yang sedang berjalan tanpa memulai ulang aplikasi
- e. Template kode dan integrasi GitHub untuk membantu membuat fitur aplikasi umum dan mengimpor kode sampel
- f. *Framework* dan alat pengujian yang lengkap

- g. Alat lint untuk merekam performa, kegunaan, kompatibilitas versi, dan masalah lainnya
- h. Dukungan C++ dan NDK
- i. Dukungan bawaan untuk *Google Cloud Platform*, yang memudahkan integrasi *Google Cloud Messaging* dan *App Engine*.

2.7.1. Struktur Proyek



Gambar 2. 1 Struktur proyek pada Android Studio

Setiap project di Android Studio berisi satu atau beberapa modul dengan file kode sumber dan file *resource*. Jenis modul meliputi :

- a. Modul aplikasi *Android*
- b. Modul *library*
- c. Modul *Google App Engine*

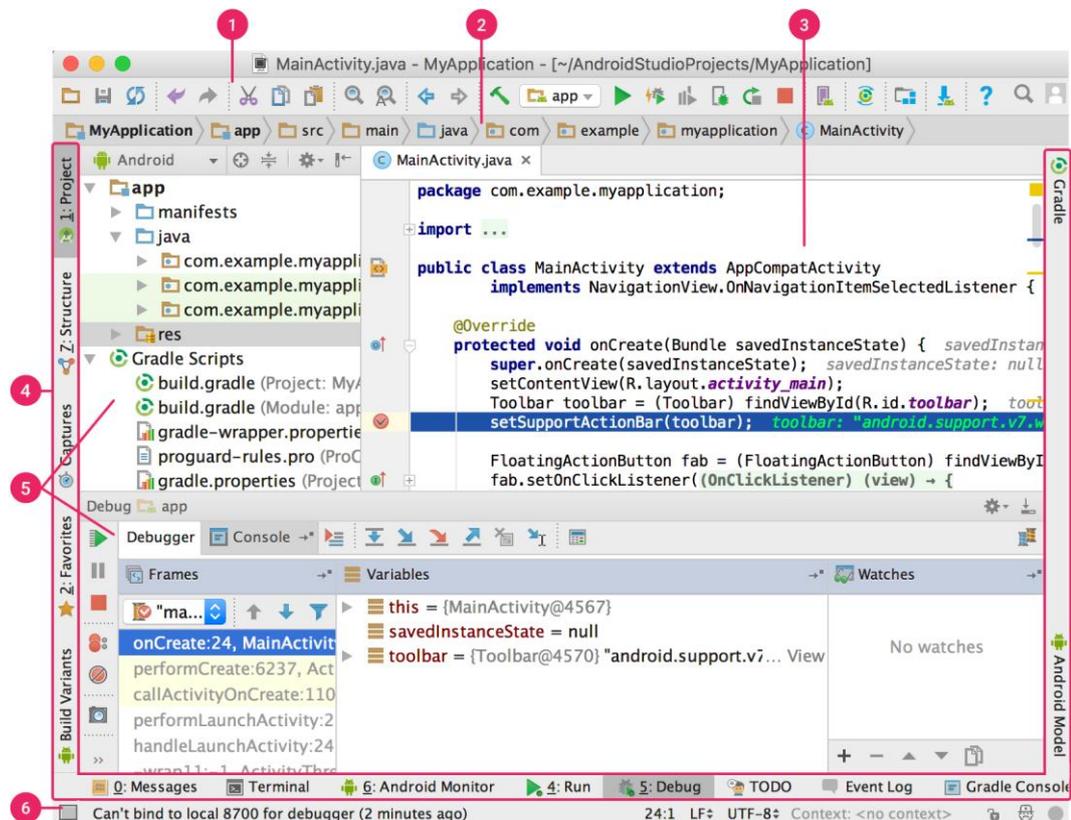
Secara default, Android Studio menampilkan file project dalam tampilan project Android, seperti yang ditunjukkan pada gambar 2.1. Tampilan disusun menurut modul untuk memberikan akses cepat ke file sumber utama project.[10]

Semua file build terlihat di tingkat teratas di bagian Gradle Script dan etuap modul aplikasi berisi folder berikut :

- a. `manifests` : Berisi file `AndroidManifest.xml`
- b. `java` : Berisi file kode sumber Java, termasuk kode pengujian Junit.
- c. `res` : Berisi semua resource non-kode, seperti tata letak XML, string UI, dan gambar bitmap.

Struktur *project Android* pada disk berbeda dengan representasi tersatukan ini, Untuk melihat struktur file *project* sebenarnya, pilih *Project* dari menu drop-down *Project* (pada gambar 2.1, ditampilkan sebagai *Android*). [10]

2.7.2. Antarmuka Pengguna



Gambar 2. 2 Jendela utama Android Studio

1. Toolbar memungkinkan untuk melakukan berbagai tindakan, termasuk menjalankan aplikasi dan meluncurkan alat Android.
2. Menu navigasi membantu untuk menjelajah project dan membuka file untuk diedit. Menu ini memberikan tampilan struktur yang lebih ringkas yang terlihat di jendela Project.
3. Jendela editor adalah tempat untuk membuat dan memodifikasi kode. Bergantung pada jenis file yang ada, editor ini dapat berubah. Misalnya, saat menampilkan file tata letak, editor akan menampilkan Layout Editor.
4. Panel jendela fitur berada di sisi luar jendela IDE dan berisi tombol-tombol yang memungkinkan untuk memperluas atau menciutkan setiap jendela fitur.
5. Jendela fitur memberi untuk akses ke tugas tertentu seperti pengelolaan project, penelusuran, kontrol versi, dan lainnya. Untuk dapat memperluas dan menciutkan jendela ini.
6. Status bar menampilkan status project dan IDE itu sendiri, serta semua peringatan atau pesan.

2.7.3. Jendela Alat

Sebagai ganti menggunakan perspektif preset, Android Studio mengikuti konteks dan otomatis menampilkan jendela alat yang relevan saat bekerja. Secara default, jendela alat yang paling umum digunakan disematkan ke kolom jendela fitur di tepi jendela aplikasi.[10]

1. Untuk memperluas atau menciutkan jendela alat, klik nama alat di kolom jendela alat. Dapat juga menarik, menyematkan, melepaskan sematan, memasang, dan melepas jendela fitur.
2. Untuk kembali ke tata letak jendela alat default saat ini, klik Window > Restore Default Layout atau sesuaikan tata letak default dengan mengklik Window > Store Current Layout as Default.
3. Untuk menampilkan atau menyembunyikan seluruh kolom jendela alat, klik ikon jendela di pojok kiri bawah jendela Android Studio.
4. Untuk menemukan jendela alat tertentu, arahkan kursor ke atas ikon jendela dan pilih jendela alat tersebut dari menu.

2.8. GPS

GPS (Global Positioning System) adalah system satellite navigasi dan penentuan posisi yang dimiliki dan dikelola oleh Amerika Serikat. Sistem ini didesain untuk memberikan posisi dan kecepatan tiga-dimensi serta informasi waktu, secara kontinyu di seluruh dunia tanpa bergantung waktu dan cuaca, kepada banyak orang secara simultan. Pada saat ini, system GPS sudah banyak diaplikasikan, terutama yang terkait dengan aplikasi-aplikasi yang menuntut informasi tentang posisi. [11]

Dibandingkan dengan sistem dan metode penentuan posisi lainnya, GPS mempunyai banyak kelebihan dan menawarkan lebih banyak keuntungan, baik dalam segi operasionalisasinya maupun kualitas posisi yang diberikan. Sebelum hal tersebut dijelaskan lebih lanjut, beberapa konsep dasar tentang posisi dan system koordinat serta metode-metode dalam penentuan posisi, akan dijelaskan terlebih dahulu secara singkat.[11]

2.9. Bluetooth

Bluetooth adalah sebuah teknologi komunikasi wireless (tanpa kabel) yang beroperasi dalam pita frekuensi 2,4 GHz unlicensed ISM (Industrial, Scientific and Medical) dengan menggunakan sebuah frequency hopping tranceiver yang mampu menyediakan layanan komunikasi data dan suara secara real-time antara host-host bluetooth dengan jarak jangkauan layanan yang terbatas (sekitar 10 meter). Bluetooth sendiri dapat berupa card yang bentuk dan fungsinya hampir sama dengan card yang digunakan untuk wireless local area network (WLAN) dimana menggunakan frekuensi radio standar IEEE 802.11, hanya saja pada bluetooth mempunyai jangkauan jarak layanan yang lebih pendek dan kemampuan transfer data yang lebih rendah. [12]

Pada dasarnya bluetooth diciptakan bukan hanya untuk menggantikan atau menghilangkan penggunaan kabel didalam melakukan pertukaran informasi, tetapi juga mampu menawarkan fitur yang baik untuk teknologi mobile wireless dengan biaya yang relatif rendah, konsumsi daya yang rendah, interoperability yang menjanjikan, mudah dalam pengoperasian dan mampu menyediakan layanan yang

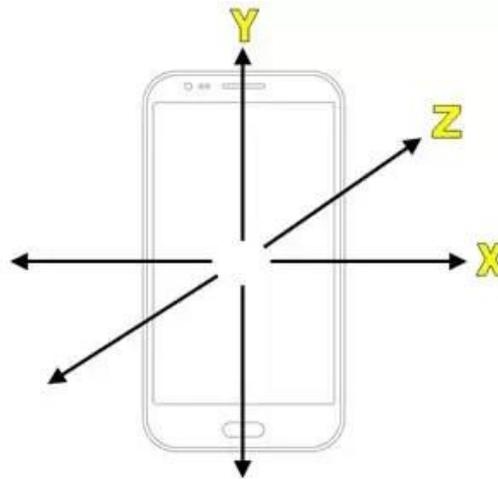
bermacam-macam. Untuk memberi gambaran yang lebih jelas mengenai teknologi bluetooth yang relatif baru ini kepada pembaca, berikut diuraikan tentang sejarah munculnya bluetooth dan perkembangannya, teknologi yang digunakan pada sistem bluetooth dan aspek layanan yang mampu disediakan, serta sedikit uraian tentang perbandingan metode modulasi spread spectrum FHSS (Frequency Hopping Spread Spectrum) yang digunakan oleh bluetooth dibandingkan dengan metode spread spectrum DSSS (Direct Sequence Spread Spectrum).

2.10. Accelerometer

Accelerometer adalah sebuah perangkat yang mampu mengukur sebuah kekuatan akselerasi. Kekuatan ini mungkin statis (diam) seperti halnya kekuatan konstan dari gravitasi Bumi, atau dapat juga bersifat dinamis karena gerakan atau getaran dari sebuah alat akselerometer.[13]

Accelerometer adalah sebuah transduser yang berfungsi untuk mengukur percepatan, mendeteksi dan mengukur getaran, ataupun untuk mengukur percepatan akibat gravitasi bumi. Accelerometer juga dapat digunakan untuk mengukur getaran yang terjadi pada kendaraan, bangunan, mesin, dan juga dapat digunakan untuk mengukur getaran yang terjadi di dalam bumi, getaran mesin, jarak yang dinamis, dan kecepatan dengan ataupun tanpa pengaruh gravitasi bumi.[13]

Fungsi *accelerometer* pada *smartphone* yaitu untuk *shake control*. Pada saat pengguna menggoyangkan perangkat, seketika fitur pada perangkat pun berubah. Misalnya saat pengguna memutar lagu kemudian dapat berganti ke playlist selanjutnya. Atau dapat juga disaat pengguna mengganti tampilan layar dari *vertical* ke *horizontal*. Sensor *accelerometer* mengukur percepatan bahwa perangkat mengalami perubahan yang relatif sesuai dengan tiga sumbu XYZ atau kanan, kiri, atas, bawah, dan datar. Sistem atau suatu aplikasi menggunakan data ini untuk mengetahui apakah *smartphone* dalam orientasi berdiri (*portrait*) atau memanjang (*landscape*). Sensor lain yang sama juga digunakan untuk menentukan orientasi perangkat bersama tiga sumbu. Gambar sumbu accelerometer dapat dilihat pada Gambar 2.3 Accelerometer.



Gambar 2. 3 Accelerometer

2.11. Vector Magnitude

Besarnya suatu vektor adalah panjang vektor tersebut. Besarnya vektor a dilambangkan dengan $\|a\|$. Rumus untuk besaran vektor dalam tiga dimensi adalah

$$\|a\| = \sqrt{a_1^2 + a_2^2 + a_3^2} . [14]$$

2.12. Firebase

Firebase merupakan layanan dari Google yang dapat digunakan untuk mempermudah pengembang aplikasi membuat, mengembangkan dan menjual aplikasi buatannya. Firebase adalah evolusi dari Envolv yang merupakan sebuah layanan pengembangan aplikasi yang memiliki fitur chat. Untuk membuat perangkat lunak yang bagus tidak harus menggunakan peralatan dengan harga mahal. Penyedia layanan untuk pengembang perangkat lunak secara gratis dan berbagai macam perangkat lunak open source dapat dimanfaatkan dengan bebas dari Internet. Membuat program yang disukai pengguna membutuhkan keahlian sekaligus tool yang tepat. Tanpa hal tersebut kecil kemungkinan dapat membuat perangkat lunak yang bagus dan mudah digunakan oleh pengguna. Dengan memanfaatkan fitur yang ada pada Firebase membuat mengembangkan dan menjual perangkat lunak lebih mudah. Panduan untuk menggunakan Firebase dapat diperoleh secara gratis melalui situs <https://firebase.google.com>. [15]

Firebase merupakan sebuah layanan yang ditunjukan untuk pengembang aplikasi perangkat lunak. Firebase memfasilitasi pengembang aplikasi agar dapat mengembangkan aplikasi berkualitas tinggi, Dengan menggunakan Firebase pengembang aplikasi dapat menngembangkan aplikasi ciptaannya dengan lebih mudah dan cepat. Pilihan penggunaannya pun bervariasi, sehingga dapat digunakan oleh berbagai pengembang dengan kondisi keuangan yang berbeda-beda. Fitur pilihan spark dapat digunakan untuk membuat dan menjual aplikasi secara gratis. Fitur pilihan flame dan blaze dapat dimanfaatkan dengan tarif biaya perbulan mulai dari \$25.[15]

2.13. Java

Pada tahun 1991, bahasa pemrograman Java pertama kali dikembangkan lewat sebuah proyek bernama “The Green Project” yang dibentuk oleh Sun Microsystems. Beranggotakan James Gosling, Patrick Naughton, dan Mike Sheridan, proyek ini memiliki tujuan untuk menciptakan bahasa pemrograman baru yang lebih andal dari C dan C++. Setelah proyek berjalan selama 18 bulan, James Gosling, sang developer, akhirnya berhasil mengembangkan bahasa pemrograman baru yang diberi nama Oak. Sayangnya, nama tersebut telah terdaftar sebagai merek dagang dari perusahaan Oak Technology.

Akhirnya, sekitar tahun 1995, Sun Microsystem kemudian resmi mengubah nama Oak menjadi Java yang kabarnya terinspirasi dari java coffee/kopi Jawa. Itu sebabnya, logo bahasa pemrograman Java bergambar secangkir kopi. Java dikenal memiliki moto “Write Once, Run Anywhere”. Hal ini menunjukkan bahwa setiap kode program yang ditulis menggunakan Java bisa langsung dijalankan pada platform berbeda-beda tanpa perlu menyusun ulang.

Penulisan kode program Java biasanya disimpan dalam file berekstensi java. Dengan menggunakan javac compiler, kamu bisa menyusun file berekstensi java menjadi file berekstensi class. File berekstensi class inilah yang disebut sebagai bytecode dan bisa dijalankan di seluruh Java Virtual Machine (JVM). Dilihat dari penggunaannya, sebagai bahasa pemrograman umum java bisa dimanfaatkan untuk membuat berbagai bentuk aplikasi. Hal itu berlaku mulai dari aplikasi berbasis

desktop, website, mobile, hingga aplikasi embedded device seperti perangkat pintar atau microprosesor. Sementara dalam praktiknya, bahasa Java tidak mendukung pemrograman prosedural melainkan menggunakan konsep pemrograman berbasis object.[16]

2.14. HyperText Markup Language (HTML)

HyperText Markup Language (HTML) adalah bahasa yang digunakan untuk menulis halaman web. HTML merupakan pengembangan dari standar pemformatan dokumen teks. Yaitu *Standar Generalized Markup Language* (SGML). HTML pada dasarnya merupakan dokumen ASCII atau teks biasa, yang dirancang untuk tidak tergantung pada suatu sistem operasi tertentu.

HTML dibuat oleh Tim Berners-Lee ketika masih bekerja untuk CERN, dan dipopulerkan pertama kali oleh *browser* Mosaic. Selama awal tahun 1990, HTML mengalami perkembangan yang sangat pesat. Setiap pengembangan HTML, pasti akan menambahkan kemampuan dan fasilitas yang lebih baik dari versi sebelumnya.

Kegunaan bahasa ini ialah untuk memanipulasi *browser* sehingga dapat menampilkan informasi yang dapat dibaca oleh pengguna komputer. [17]

2.15. Cascading Style Sheet (CSS)

CSS (*Cascading Style Sheet*) adalah suatu bahasa *stylesheet* yang digunakan untuk mengatur tampilan suatu *website*, baik tata letaknya, jenis huruf, warna, dan semua yang berhubungan dengan tampilan. Pada umumnya CSS digunakan untuk menformat halaman web yang ditulis dengan HTML atau XHTML.

Ada dua cara yang bisa diterapkan untuk menggunakan CSS pada web. Cara yang pertama dengan membuat CSS langsung di dalam satu file HTML (*internal/inline style sheet*). Cara yang kedua dengan memanggil CSS tersebut dari file CSS tersendiri (*external style sheet*). [17]

2.16. JavaScript

JavaScript adalah bahasa script berdasar pada objek yang memperbolehkan pemakai untuk mengendalikan banyak aspek interaksi pemakai pada suatu

dokumen HTML. Di mana objek tersebut dapat berupa suatu window, frame, URL, dokumen, form, button, atau item yang lain. Yang semuanya itu mempunyai properti yang saling berhubungan dengannya, dan masing-masing memiliki nama, lokasi, warna nilai, dan atribut lain. [17]

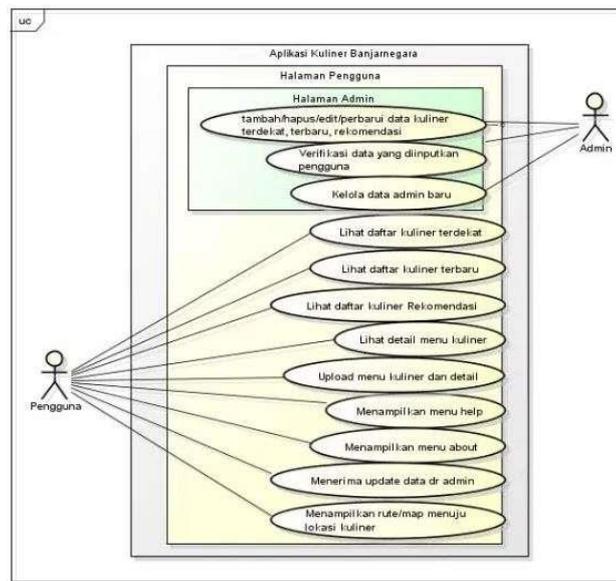
2.17. Unified Modelling Language (UML)

UML atau “Unified Modelling Language” adalah suatu metode permodelan secara visual yang berfungsi sebagai sarana perancangan sistem berorientasi objek. Definisi UML adalah sebagai suatu bahasa yang sudah menjadi standar pada visualisasi, perancangan, dan juga pendokumentasian sistem aplikasi. Saat ini UML menjadi bahasa standar dalam penulisan blue print software (arsitektur).

Unified Modelling Language atau UML merupakan salah satu standar bahasa yang banyak digunakan di dunia industri untuk menggambarkan kebutuhan (requirement), membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (PBO).[18]

2.17.1. Use Case Diagram

Use Case diagram adalah suatu urutan interaksi yang saling berkaitan antara sistem dan aktor. Use case dijalankan melalui cara menggambarkan tipe interaksi antara user suatu program (sistem) dengan sistemnya sendiri. Use case melalui sebuah cerita yang mana sebuah sistem itu dipakai. Use case juga dipakai untuk membentuk perilaku (behaviour) sistem yang akan dibuat. Sebuah use case menggambarkan sebuah interaksi antara pengguna (aktor) dengan sistem yang sudah ada. Contoh Use Case diagram terdapat pada Gambar 2.4 Use Case Diagram.



Gambar 2. 4 Contoh Use Case Diagram

Simbol dan penjelasan dari use case diagram terdapat pada Tabel 2.1 Simbol Usecase.

Tabel 2. 1 Simbol Usecase

Simbol	Nama	Keterangan
	Actor	Merupakan peran orang, sistem yang lain, atau alat ketika berhubungan dengan use case.
	Use Case	Abstraksi dari penghubung antara aktor dengan use case.
	Association	Abstraksi dari penghubung antara aktor dengan use case.

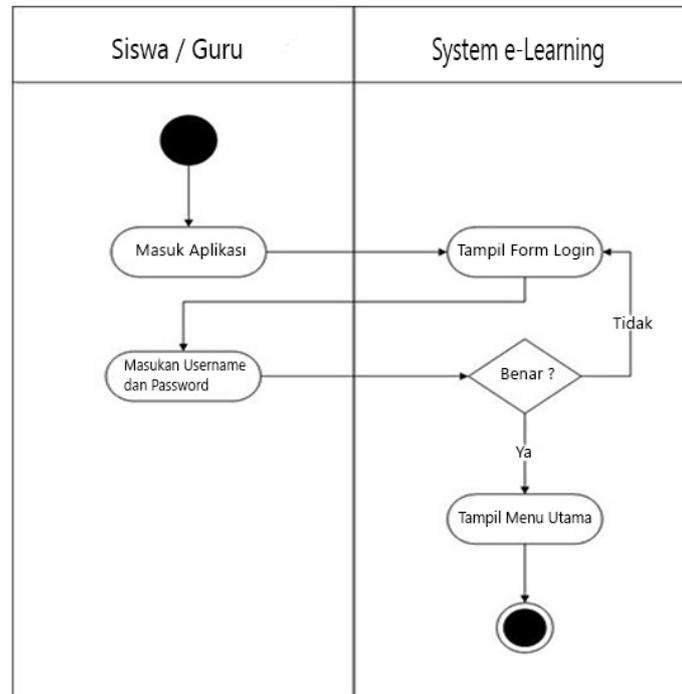
	Generalisasi	Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan use case.
	Include	Menunjukkan bahwa suatu use case seluruhnya merupakan fungsionalitas dari use case lainnya.
	Extend	Menunjukkan bahwa suatu use case merupakan tambahan fungsional dari use case lainnya jika suatu kondisi terpenuhi.

2.17.2. Activity Diagram

Activity diagram ialah sesuatu yang menjelaskan tentang alir kegiatan dalam program yang sedang dirancang, bagaimana proses alir berawal, keputusan yang mungkin terjadi, dan bagaimana sistem akan berakhir.

Activity diagram juga dapat menjelaskan metode paralel yang mungkin terjadi pada beberapa eksekusi. Activity diagram adalah state diagram khusus, yang mana state ini berfungsi sebagai action dan sebagian besar transisi ditrigger oleh akhir state sebelumnya (internal processing).

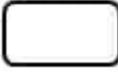
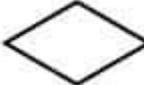
Contoh Activity Diagram dapat dilihat pada Gambar 2.5 Contoh Activity Diagram.



Gambar 2. 5 Contoh Activity Diagram

Simbol atau komponen dari activity diagram terdapat pada Tabel 2.2 Simbol Activity Diagram.

Tabel 2. 2 Simbol Activity Diagram

Simbol	Nama	Keterangan
	Status awal	Sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan / Decision	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
	Penggabungan / Join	Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.
	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir

	Swimlane	Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi
---	----------	--

2.17.3. Class Diagram

Class diagram adalah visual dari struktur sistem program pada jenis-jenis yang di bentuk. Class diagram merupakan alur jalannya database pada sebuah sistem.

Class diagram merupakan penjelasan proses database dalam suatu program. Dalam sebuah laporan sistem maka class diagram ini wajib ada.

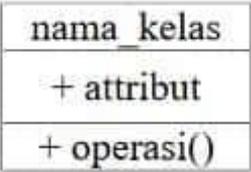
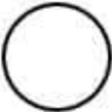
Menurut Para ahli Satzinger (2011:28) Diagram kelas atau class diagram menjelaskan struktur sistem dari segi pendefinisian class-class yang akan dibuat untuk membangun sebuah sistem. Contoh dari class diagram dapat dilihat pada Gambar 2.6 Contoh Class Diagram.



Gambar 2. 6 Contoh Class Diagram

Simbol dari class diagram dapat dilihat pada Tabel 2.3 Simbol Class Diagram

Tabel 2. 3 Simbol Class Diagram

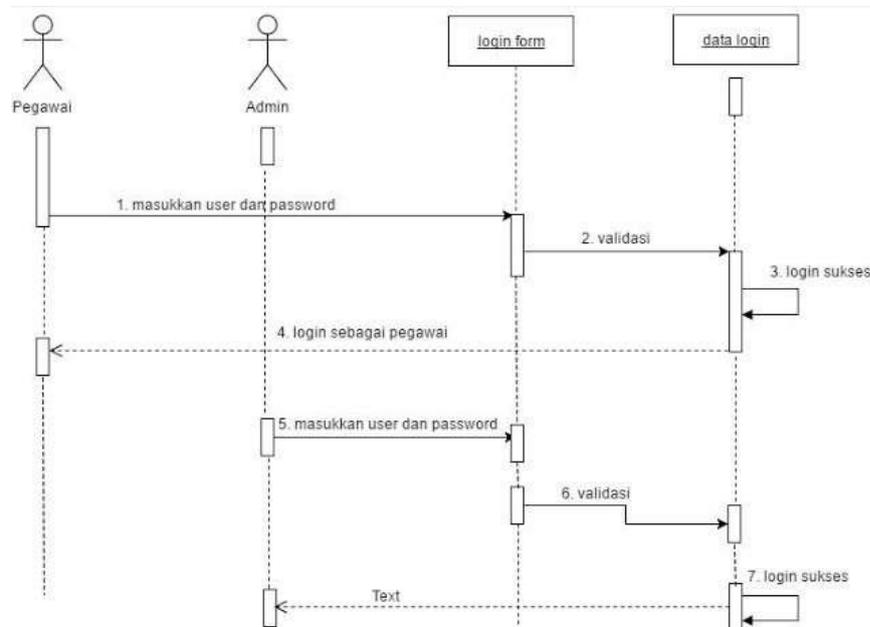
Simbol	Nama	Keterangan
	Kelas	Kelas pada struktur sistem.
	Interface	Sama dengan konsep interface dalam pemrograman berorientasi objek.
	Association	Relasi antarclass dengan arti umum, asosiasi biasanya juga disertai dengan Multiplicity.
	Directed Association	Relasi antarkelas dengan makna kelas yang atau digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity.
	Generalisasi	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus).
	Dependency	Relasi antarkelas dengan makna kebergantungan antarkelas'
	Aggregation	Relasi antarkelas dengan makna semua-bagian (whole-part)

2.17.4. Sequence Diagram

Sequence Diagram adalah suatu diagram yang menjelaskan interaksi objek dan menunjukkan (memberi tanda atau petunjuk) komunikasi diantara objek-objek tersebut.

Sequence diagram digunakan untuk menjelaskan perilaku pada sebuah skenario dan menggambarkan bagaimana entitas dan sistem berinteraksi, termasuk pesan yang dipakai saat interaksi. Semua pesan digambarkan dalam urutan pada eksekusi.

Sequence diagram berkaitan erat dengan Use Case Diagram, yang mana 1 Use Case akan menjadi 1 Sequence Diagram. Contoh sequence diagram dapat dilihat pada Gambar 2.7 Contoh Sequence Diagram

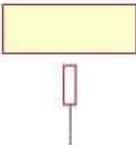
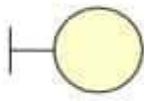


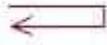
Gambar 2. 7 Contoh Sequence Diagram

Adapun simbol atau komponen sequence diagram terdapat pada Tabel 2.4 Simbol Sequence Diagram

Tabel 2. 4 Simbol Sequence Diagram

Simbol	Nama	Keterangan
	Aktor	merepresentasikan entitas yang berada di luar sistem dan berinteraksi dengan sistem

	Lifeline	menghubungkan objek selama sequence (message dikirim atau diterima dan aktifasinya).
	General	Merepresentasikan entitas tunggal dalam sequence diagram.
	Boundary	berupa tepi dari sistem, seperti user interface atau suatu alat yang berinteraksi dengan sistem yang lain.
	Control	element mengatur aliran dari informasi untuk sebuah skenario. Objek ini umumnya mengatur perilaku dan perilaku bisnis.
	Entitas	elemen yang bertanggung jawab menyimpan data atau informasi. Ini dapat berupa beans atau model object.
	Activation	suatu titik dimana sebuah objek mulai berpartisipasi di dalam sebuah sequence yang menunjukkan kapan sebuah objek mengirim atau menerima objek.
	Message	berfungsi sebagai komunikasi antar objek yang menggambarkan aksi yang akan dilakukan
	Message Entry	berfungsi untuk menggambarkan pesan/hubungan antar objek yang menunjukkan urutan kejadian yang terjadi.
	Message to Self	Simbol ini menggambarkan pesan/hubungan objek itu sendiri, yang menunjukkan urutan kejadian yang terjadi.

	Message Return	menggambarkan hasil dari pengiriman message dan digambarkan dengan arah dari kanan ke kiri.
---	-------------------	---