

BAB II **LANDASAN TEORI**

2.1 Tinjauan Pustaka

Tinjauan pustaka dari penelitian sebelumnya digunakan sebagai pendukung pada penelitian yang sedang dilakukan. Penelitian terdahulu yang telah dilakukan antara lain:

1. Penelitian oleh Wahyu Priyoatmiko dan Kapti (2020) dengan judul *Aplikasi Pembelajaran Doa Untuk Anak Berbasis Android*. Penelitian ini menggunakan metode penelitian *Multimedia Development Life Cycle (MDLC)* metode penelitian yang memiliki 6 tahapan, yaitu *Concept, Design, Material Collecting, Assembly, Testing* dan *Distribution*. Aplikasi yang digunakan dalam pembuatan sistem adalah Adobe Flash CS6 yang merupakan aplikasi pengembangan dari Adobe Flash CS5 yang mendukung Flash player 11.4 dan AIR 3.4. Pengujian aplikasi menggunakan *alpha testing*. Penelitian ini menghasilkan aplikasi doa harian islam dengan fitur teks arab, arti dengan teks latin dan gambar yang memudahkan *user*.
2. Penelitian oleh Nora Listiana dkk (2022) dengan judul *Aplikasi Doa-Doa Islam Untuk Pelajar Berbasis Android*. Penelitian ini dilakukan dengan tujuan untuk mempermudah mempelajari doa-doa islam yang bersumber pada Alquran dan Hadits. Metode pengembangan sistem yang digunakan adalah *waterfall* yaitu model pengembangan sistem yang menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengkodean, pengujian, dan tahap pendukung. Hasil

penelitian yaitu, sebuah aplikasi yang memiliki fitur untuk mempelajari doa dengan membaca dan mendengar.

3. Penelitian oleh Abdul Rokhim dan Siti Lailatul Rohmah (2020) dengan judul *Pembuatan Aplikasi mobile Pembelajaran Adab dan Doa Menggunakan Metode Addie*. Penelitian ini bertujuan untuk mengembangkan aplikasi android yang berisi materi adab dan doa harian dalam bentuk video. Metode penelitian yang digunakan pada penelitian ini adalah ADDIE (Analyze, Design, Development, Implement, dan Evaluation) yang merupakan metode yang telah umum digunakan untuk mengembangkan aplikasi multimedia. Pengujian pada penelitian ini menggunakan *black box testing*. Hasil dari penelitian ini adalah aplikasi berbasis android dengan fitur audio visual atau video yang berisi tentang adab dan do'a sehari-hari.
4. Penelitian oleh Fauziah Isnaini dan Muh Ariffudin Islam (2021) dengan judul *Perancangan Aplikasi Interaktif Islami Android "Doa sehari-Hari Bersama Zio & Aza"*. Penelitian ini dilakukan untuk melakukan perancangan sebuah aplikasi interaktif android yang diharapkan dapat dijadikan sebagai sarana edukasi digital untuk anak-anak. Metode perancangan yang digunakan pada penelitian ini adalah metode *Design Thinking*, yaitu proses berulang dimana kita berusaha memahami pengguna, menantang asumsi, dan mendefinisikan kembali masalah dalam upaya untuk mengidentifikasi strategi dan solusi alternatif yang mungkin tidak langsung terlihat pada tingkat pemahaman awal kita. Proses pengujian pada penelitian ini menggunakan *prototype* aplikasi oleh 2 orang validator dibidangnya dan seorang ibu rumah tangga beserta

anaknya. Hasil dari penelitian ini, yaitu aplikasi dengan fitur cerita berupa teks, doa teks arab dan permainan menyusun gambar dan tebak doa.

5. Penelitian oleh Ani Nur Aeni dkk (2022) dengan judul *Aplikasi BETA (Belajar dari Peta): Media Edukasi Doa-Doa Harian Siswa SD Kelas Rendah*. Penelitian ini bertujuan untuk membangun aplikasi yang dapat membantu menyempurnaan media belajar doa-doa harian bagi anak sekolah dasar kelas rendah. Metode penelitian ini menggunakan metode Design and Development (D&D) yaitu sebuah metode yang didefinisikan sebagai suatu pengkajian sistematis terhadap pendesainan, pengembangan dan evaluasi program, proses dan produk pembelajaran yang harus memenuhi kriteria validitas, kepraktisan, dan efektifitas. Hasil penelitian ini adalah aplikasi BETA yang mengenalkan doa-doa harian menggunakan metode belajar dan bermain.

Berdasarkan tinjauan pustaka, penelitian terkait aplikasi pembelajaran dan hafalan doa sudah banyak dilakukan pada penelitian terdahulu. Namun, perbedaan penelitian terdahulu dengan penelitian yang akan dilakukan yaitu metode penelitian yang akan digunakan. Pada penelitian ini metode yang digunakan yaitu *Extreme Programming (XP)* yang merupakan metode penelitian dengan 4 tahapan, yaitu perencanaan, perancangan, pengkodean dan pengujian. Fitur aplikasi dengan kompetensi belajar membaca, mendengar dan menyimak video animasi sesuai dengan anak usia dini. Metode pengujian yang digunakan untuk menguji aplikasi adalah *ISO/IEC 25010*. Pembuatan aplikasi menggunakan bahasa pemrograman Java menggunakan Android Studio.

2.2 Aplikasi

Aplikasi berasal dari kata *application* yang memiliki arti penerapan, penggunaan. Secara istilah aplikasi adalah program siap pakai yang dirancang untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh target yang dituju (Aziz, 2018). Aplikasi adalah perangkat lunak yang berfungsi sebagai *frontend* dalam suatu sistem yang digunakan untuk mengolah berbagai data menjadi informasi yang berguna bagi penggunanya dan sistem terkait (Soraya & Wahyudi, 2021). Aplikasi dirancang untuk melakukan tugas tertentu pada perangkat elektronik seperti komputer, *smartphone*, tablet, atau perangkat lainnya. Aplikasi merupakan *software* atau alat terapan yang dibuat untuk mengerjakan tugas-tugas khusus yang saling berinteraksi untuk tujuan tertentu (Wulan & Putra, 2022).

Berdasarkan uraian diatas, dapat disimpulkan bahwa aplikasi adalah program komputer yang dirancang untuk membantu kegiatan pengguna. Aplikasi juga memungkinkan pengguna untuk berinteraksi dengan sumber daya atau layanan yang tersedia di internet. Dalam dunia digital modern, aplikasi sangat penting karena memungkinkan pengguna untuk melakukan tugas-tugas secara efektif dan efisien, dan juga memungkinkan bisnis untuk menyediakan layanan dan produk yang lebih mudah diakses dan digunakan oleh konsumen.

2.3 Doa

Doa merupakan kajian yang sangat mendalam tentang kehidupan manusia, setiap doa telah diajarkan oleh Allah dan Rasul-Nya. Doa adalah salah satu cara untuk berkomunikasi sekaligus mensyukuri segala nikmat yang Allah SWT berikan kepada kita selaku hambanya. Dengan mengajarkan doa kepada anak-anak kita

mengajak mereka untuk belajar mensyukuri nikmat yang Allah SWT berikan kepada kita dan mereka dapat mengaplikasikannya dalam kehidupan sehari-hari (Rosyidah, 2019). Doa dalam Islam bukan hanya berupa permintaan, tetapi juga sebagai sarana untuk mengingat dan mengagungkan Allah SWT, serta sebagai wujud pengakuan manusia atas kekuasaan Allah SWT sebagai Tuhan semesta alam. Umat muslim percaya bahwa doa terbaik adalah yang bersumber dari ayat-ayat suci Al-Quran. Ada banyak contoh doa para nabi yang diabadikan didalamnya (Anwar & Priyatna, 2021).

Berdasarkan uraian diatas, dapat disimpulkan bahwa doa merupakan sebuah bentuk komunikasi dengan Tuhan yang dilakukan oleh orang-orang yang beragama untuk memohon bantuan, rahmat, dan perlindungan. Dalam islam, doa merupakan tindakan ibadah yang penting dan harus dilakukan dengan hati yang tulus dan penuh harapan kepada Allah SWT. Melalui doa, umat muslim dapat memperkuat hubungan spiritual dengan-Nya dan memperoleh keberkahan serta perlindungan dari-Nya.

2.4 Surah Pendek

Menurut situs website (theAsianparent, 2023) surah pendek adalah surah yang terdapat dalam Al-Qur'an juz 30. surah pendek memiliki jumlah ayat yang lebih sedikit dari surah lainnya. surah pendek biasa dibaca saat shalat setelah membaca surah Al-Fatihah. Dalam Islam salah satu cara untuk mempelajari surah pendek adalah dengan menghafal. Menghafal surah pendek adalah menghafal surah-surah pada Al-Qur'an sesuai dengan urutan yang terdapat dalam Mushaf Utsmani, dimulai dari surah An-Nass, dengan tujuan beribadah, menjaga dan memelihara kalam Allah yang merupakan mukjizat yang diberikan kepada Nabi

Muhammad SAW yang diturunkan melalui malaikat Jibril dan disampaikan dengan jalan Mutawatir (Suningsih, 2020).

Berdasarkan uraian diatas, dapat disimpulkan bahwa surah pendek adalah surah yang berada didalam Al-Qur'an dan terdiri dari sedikit ayat. Menghafal surah pendek menjadi salah satu cara untuk memperlajarinya, karena surah pendek memiliki keistimewaan dan keutamaan tersendiri dalam kehidupan sehari-hari bagi umat Muslim.

2.5 Android

Android adalah sistem operasi yang dibuat untuk perangkat *mobile* berbasis Linux termasuk sistem operasi, *middleware* dan aplikasi inti yang menggunakan sistem operasi Linux yang dimodifikasi, Android menyediakan platform terbuka bagi pengembang untuk membuat aplikasi mereka sendiri (Ramadhan et al., 2021).

Android merupakan generasi baru *platform mobile*, *platform* yang memberikan pengembangan sesuai dengan yang diharapkan. Sistem operasi yang mendasari android dilisensikan di bawah GNU, *General Public Lisensi* Versi 2 (GPLv2), yang sering dikenal dengan istilah *copyleft*. (Voutama & Novalia, 2021). Ikon Android berbentuk robot dengan dua antena pada kepalanya, melambangkan bahwa Android adalah ikon sistem operasi canggih untuk *gadget* dan *smartphone*. Sejak dirilis pada tahun 2007, Android telah beberapa kali memperbarui versinya. Versi Android memiliki nama unik yang kebanyakan menggunakan nama makanan manis (Komang Ratih Kumala et al., 2020).

Berdasarkan uraian diatas, dapat disimpulkan bahwa Android adalah sistem operasi yang dikembangkan dan dirancang untuk perangkat mobile seperti *gadget* dan *smartphone*. Android memiliki antarmuka pengguna yang responsif,

kemampuan *multi-tasking*, dan dukungan untuk aplikasi pihak ketiga yang besar. Android juga memiliki *platform* yang terbuka, yang memungkinkan pengembang untuk memodifikasi sistem operasi sesuai kebutuhan mereka. Hal ini membuat Android menjadi platform yang populer bagi pengembang aplikasi dan produsen perangkat *mobile*.

2.6 Media Pembelajaran

Media pembelajaran adalah media yang dapat membantu proses belajar mengajar sehingga pesan yang disampaikan menjadi lebih jelas dan tujuan pendidikan atau pembelajaran dapat tercapai dengan efektif dan efisien (Nurrita, 2018). Media pembelajaran merupakan sarana untuk menyalurkan pesan pembelajaran dan informasi. Media pembelajaran yang dirancang dengan baik akan sangat membantu dalam mencapai tujuan pembelajaran. Masing-masing jenis media pembelajaran memiliki karakteristik, keuntung-an, dan kerugian (Hamid et al., 2020).

Di era digital, media pembelajaran sangat penting bagi peserta didik untuk memperoleh konsep, keterampilan dan kompetensi baru. Tenaga pendidik dapat menggunakan berbagai jenis media dalam proses belajar mengajar, namun tenaga pendidik harus selektif dalam memilih media tersebut. Tenaga pendidik tidak hanya mengetahui cara menggunakan media belajar klasik, tetapi juga media belajar modern. Beberapa hasil penelitian juga menunjukkan pengaruh positif media sebagai bagian tak terpisahkan dari pengajaran tatap muka atau sebagai metode utama pembelajaran praktis (Hasan et al., 2021).

Berdasarkan uraian diatas, dapat disimpulkan bahwa media pembelajaran merupakan alat yang sangat berguna untuk menunjang proses pembelajaran agar

lebih efektif dan efisien. Media pembelajaran dapat membantu peserta didik memahami konsep yang kompleks, memperjelas informasi, meningkatkan keterlibatan peserta didik, dan membantu peserta didik mengingat materi dengan lebih baik.

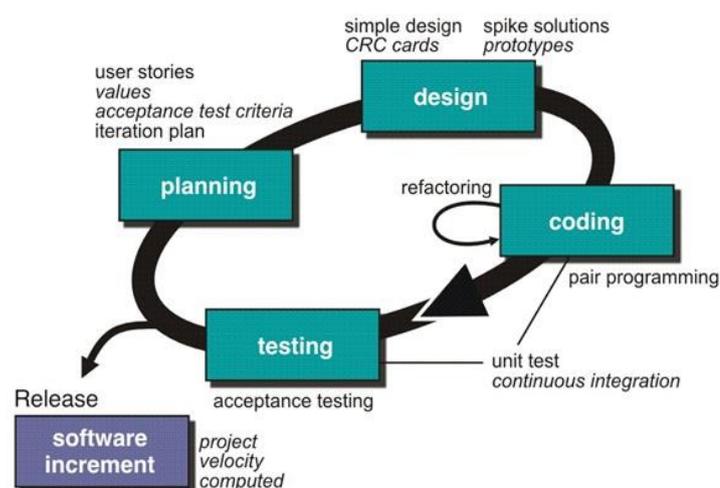
2.7 Anak Usia Dini

Anak Usia Dini adalah masa emas (*golden age*) yang berusia 0-6 tahun dan hanya ada satu tahap dalam hidup yang tidak bisa diulang. Pada saat ini, anak-anak sedang dalam masa sensitif dan sangat mudah menerima banyak hal yang berdampak pada mereka. Anak mulai berkembang dan cepat menerima apa yang dilihat dan didengar baik dari orang tua, guru maupun lingkungan sekitar (Suhendro & Syaefudin, 2020). Anak usia dini merupakan individu yang mengalami proses pertumbuhan dan perkembangan yang sangat pesat, bahkan sebagai lompatan dalam pertumbuhan anak. Anak usia dini memiliki usia yang sangat berharga dibandingkan dengan usia selanjutnya, karena perkembangan kecerdasannya sangat luar biasa. Usia ini merupakan masa kehidupan yang unik dan merupakan proses perubahan berupa pertumbuhan, perkembangan, pendewasaan dan kesempurnaan baik fisik maupun mental yang bertahap dan berkesinambungan sepanjang hidup (Khairi, 2018).

Berdasarkan uraian diatas, dapat disimpulkan bahwa Anak usia dini adalah istilah yang digunakan untuk menggambarkan anak-anak dari usia 0 hingga 6 tahun. Pada usia ini, anak berada dalam fase perkembangan fisik, mental, dan emosional yang sangat penting. Anak usia dini memerlukan perhatian khusus dan bimbingan yang tepat agar dapat tumbuh dan berkembang secara optimal.

2.8 Extreme Programming (XP)

Extreme Programming adalah metode yang berfokus pada pengembangan perangkat lunak untuk meningkatkan kualitas berdasarkan perubahan kebutuhan pelanggan (Ariyanti et al., 2020). XP berfokus pada pengcodingan, yang merupakan aktivitas utama pada semua tahapan atau siklus pengembangan sistem. Keunggulan XP adalah langkah-langkahnya membutuhkan waktu yang cepat dan dapat diulang di bagian yang berbeda tergantung tujuan pengembangan sistem. Selanjutnya, dengan pengembangan sistem yang menyederhanakan setiap proses dalam langkah-langkahnya, pendekatan XP menjadi lebih adaptif dan fleksibel (Ahmad et al., 2020). XP berfokus pada peningkatan kualitas perangkat lunak dengan menggunakan praktik-praktik yang berbeda, seperti pengujian otomatis, integrasi berkelanjutan, desain sederhana, dan komunikasi yang intensif antara para anggota tim. Pada metode XP terdapat tahapan-tahapan yang digunakan sebagai pengembangan sistem yang akan dilakukan. Terdapat 4 tahapan dalam metode XP yang ditunjukkan pada **Gambar 2.1**.



Gambar 2. 1 Tahapan-Tahapan *Extreme Programming*

Sumber:(Ahmad et al., 2020)

Dari 4 tahapan pengembangan yang ada pada XP, berikut penjelasan dari masing-masing tahapannya.

1. *Planning* (Perencanaan)

Tahapan ini merupakan tahap pertama pengembangan XP. Pendefinisian aliran proses dari perangkat lunak yang dihasilkan akan menentukan *output* perangkat lunak. Tahapan ini dilakukan untuk mengidentifikasi kebutuhan dan keinginan pengguna dalam pengembangan perangkat lunak. Pengguna dan pengembang aplikasi bekerja sama untuk menentukan fitur dan fungsi utama yang harus disertakan dalam perangkat lunak. Selain itu juga menentukan batas waktu dan biaya yang tersedia untuk pengembangan.

2. *Design* (Perancangan)

Pada tahap ini setelah fitur dan fungsi utama diidentifikasi, dilakukan pembuatan desain perangkat lunak dasar dan menentukan detail teknis. Desain ini biasanya sangat sederhana dan bertujuan untuk memenuhi kebutuhan pengguna. Pengembang harus melihat kemungkinan perubahan di masa mendatang dan memastikan bahwa perangkat lunak dapat diadaptasi dengan mudah.

3. *Coding* (Pengkodean)

Pada tahap ini melibatkan penulisan kode perangkat lunak sesuai dengan desain yang dibuat pada tahap *design*. Pengembang menggunakan metode pemrograman yang ketat, seperti unit berkelanjutan dan pengujian integrasi, untuk memastikan kualitas kode. Pengembang melakukan tinjauan kode secara teratur untuk meningkatkan kualitas dan keamanan kode. Tahap pengkodean ini dilakukan saat pembuatan aplikasi dengan menggunakan bahasa pemrograman untuk membuat pemodelan yang dibuat kedalam bentuk *user interface*.

4. *Testing* (Penguujian)

Pada tahap ini dilakukan penguujian yang melibatkan tes terus-menerus secara berkelanjutan selama dalam proses pengembangan, dari penguujian unit hingga integrasi dan penguujian sistem secara keseluruhan. Pengembang menggunakan otomatisasi untuk mempercepat penguujian dan memastikan kualitas perangkat lunak yang dihasilkan. Tahapan ini berfokus pada penguujian fungsional untuk mengetahui kesalahan pada sistem yang sedang berjalan dan memastikan bahwa perangkat lunak memenuhi kebutuhan pengguna.

2.9 Unified Modeling Language (UML)

Unified Modeling Language merupakan *tool* pengembangan perangkat lunak berorientasi objek. UML sendiri juga menyediakan standar untuk penulisan sistem blueprint yang mencakup konsep proses bisnis, kelas bahasa pemrograman, skema basis data, dan dasar-dasar sistem perangkat lunak (Sonata, 2019). Perancangan perangkat lunak menggunakan pemodelan UML banyak digunakan oleh para praktisi dan akademisi. Salah satu keuntungan menggunakan diagram UML adalah fleksibilitas dan kemampuan untuk menggambarkan sistem perangkat lunak secara lebih detail dan detail (Sumiati et al., 2021).

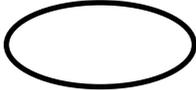
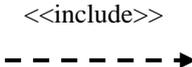
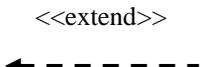
UML memungkinkan para pengembang perangkat lunak untuk menggambarkan struktur, perilaku, dan interaksi sistem perangkat lunak secara visual dan memudahkan para pengembang perangkat lunak untuk berkomunikasi satu sama lain dan juga untuk berkomunikasi dengan pelanggan atau pengguna. Dalam tahap analisis dan perancangan perangkat lunak, UML dapat digunakan untuk merancang, mengorganisir, dan menggambarkan model sistem yang

kompleks secara mudah dipahami oleh semua pihak yang terlibat dalam pengembangan perangkat lunak.

2.9.1 Use Case Diagram

Use case diagram dalam UML merupakan pemodelan dari perilaku (*behavior*). jenis diagram yang menggambarkan hubungan antara pengguna sistem dan sistem dalam kasus penggunaan. Diagram ini memberikan pandangan kinerja sistem dan menjelaskan skenario penggunaan dari sudut pandang pengguna. Berikut simbol-simbol yang digunakan pada *Use Case Diagram* ditunjukkan pada **Tabel 2.1** dan **Tabel 2.2**.

Tabel 2. 1 Simbol-Simbol *Use Case Diagram*

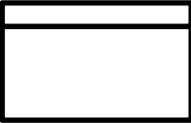
Simbol	Keterangan
	<i>Actor</i> : Mewakili peran orang, proses atau sistem lain yang berinteraksi dengan sistem yang akan dibuat.
	<i>Use Case</i> : Fungsionalitas yang disediakan untuk berinteraksi antara sistem dan <i>actor</i>
	<i>Association</i> : Digunakan untuk penghubung antara <i>actor</i> dengan <i>use case</i> yang menunjukkan koneksi antara keduanya.
	<i>Generalization</i> : Menunjukkan hubungan spesialisasi antar use case atau objek satu dengan lainnya.
	<i>Include</i> : Hubungan antara dua kasus penggunaan di mana satu <i>use case</i> penggunaan memerlukan tindakan dari <i>use case</i> penggunaan yang lain
	<i>Extend</i> : Hubungan antara dua <i>use case</i> penggunaan di mana satu <i>use case</i> penggunaan dapat menambahkan fitur atau tindakan pada <i>use case</i> penggunaan yang lain.

Sumber: (Rosa & Salahuddin, 2016)

2.9.2 Activity Diagram

Activity diagram adalah salah satu jenis diagram yang digunakan untuk memodelkan (*workflow*) aliran kerja atau aliran aktivitas dalam suatu proses bisnis atau sistem. *Activity diagram* menunjukkan aktivitas-aktivitas yang terjadi dalam suatu sistem, termasuk alur jalur yang diambil, keadaan atau kondisi sistem, tindakan yang diambil, dan objek-objek yang terlibat dalam setiap aktivitas. Berikut adalah simbol-simbol *activity diagram* ditunjukkan pada **Tabel 2.3**.

Tabel 2. 2 Simbol-Simbol *Activity Diagram*

Simbol	Keterangan
	<i>Start</i> : Sebuah diagram aktivitas harus memiliki status awal.
	<i>Activity</i> : Sebuah aktivitas yang dilakukan sistem biasanya diawali kata kerja.
	<i>Decision</i> : Percabangan apabila terdapat pilihan aktivitas yang lebih dari satu
	<i>Join</i> : Penggabungan apabila terdapat lebih dari satu aktivitas untuk digabungkan menjadi satu.
	<i>End</i> : Diagram aktivitas selalu memiliki status akhir.
	<i>Swimlane</i> : Digunakan untuk memisahkan organisasi bisnis yang bertanggung jawab pada aktivitas yang terjadi.

Sumber: (Rosa & Salahauddin, 2016)

2.10 Figma

Figma adalah salah satu *design tool* yang biasa digunakan untuk membuat tampilan aplikasi *mobile*, *desktop*, situs web, dan lainnya. Figma dapat digunakan pada sistem operasi Windows, Linux atau Mac dengan menghubungkan ke Internet. Umumnya, Figma digunakan oleh desainer *user interface/user experience* (UI/UX), desainer web, dan bidang serupa lainnya (Muhyidin et al., 2020). Keunggulan Figma adalah pekerjaan yang sama dapat dilakukan oleh beberapa orang secara bersamaan, bahkan di tempat yang berbeda. Ini bisa disebut kerja tim dan telah menjadi pilihan banyak desainer UI/UX karena fitur figma untuk membuat prototipe situs web atau aplikasi dengan cepat dan efisien (Al-Faruq et al., 2022).

Berdasarkan uraian diatas, dapat disimpulkan bahwa Figma merupakan alat desain grafis yang berbasis *cloud* yang dapat digunakan pada sistem operasi Windows, Linux atau Mac melalui browser dengan berbagai fitur kolaborasi dan prototyping yang memudahkan tim dalam membuat desain antarmuka pengguna. Figma memungkinkan para desainer dan tim proyek untuk bekerja bersama secara real-time, dengan fitur kolaborasi yang memungkinkan pengguna untuk melihat dan memberikan masukan pada desain.

2.11 Android Studio

Android Studio adalah *Integrated Development Environment* (IDE) untuk pengembangan Android yang diperkenalkan oleh Google pada acara Google I/O 2013. Android Studio merupakan pengembangan dari IDE Eclipse dan berbasis Java IDE populer, yaitu IntelliJ IDEA. Android Studio merupakan IDE resmi untuk pengembangan aplikasi Android dengan fitur yang berguna bagi pengembang (Alfarisi, 2019).

Menurut situs resmi (Developer Android, 2023) Android Studio memiliki fitur yang dapat meningkatkan produktivitas saat membangun aplikasi Android.

Adapun fitur yang diberikan sebagai berikut:

- a) Sistem build berbasis Gradle yang fleksibel
- b) Emulator yang cepat dan kaya fitur
- c) Lingkungan terpadu yang bisa mengembangkan aplikasi untuk semua perangkat Android
- d) Terapkan 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 (Native Development Kit)
- i) Dukungan bawaan untuk Google Cloud *platform*, yang memudahkan integrasi Google Cloud Messaging dan App Engine

Berdasarkan uraian diatas, dapat disimpulkan bahwa Android Studio dirancang khusus untuk membantu para pengembang dalam membuat, menguji, dan menerapkan aplikasi Android dengan lebih mudah dan efisien. Dengan fitur-fitur yang diberikan Android Studio menjadi alat yang sangat penting bagi para pengembang aplikasi Android untuk membuat aplikasi yang berkualitas dan bermanfaat bagi pengguna Android.

2.11.1 *Software Development Kit (SDK)*

Android SDK adalah sebuah *tools API (Application Programming Interface)* yang dibutuhkan untuk mengembangkan aplikasi pada platform Android dengan menggunakan bahasa pemrograman Java. Android SDK digunakan sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java (Mintoro et al., 2019). Android SDK memiliki seperangkat alat pengembangan yang komprehensif. Android SDK terdiri dari *debugger, software libraries, emulator*, dokumentasi, *sample code*, dan tutorial (Metafani et al., 2020).

Berdasarkan uraian diatas, dapat disimpulkan Android SDK adalah seperangkat alat yang diperlukan oleh para pengembang untuk membuat aplikasi Android. Android SDK berisi berbagai alat dan sumber daya yang digunakan untuk membuat aplikasi Android, termasuk emulator untuk menjalankan aplikasi secara virtual, alat untuk mengompilasi dan memecah kode sumber aplikasi menjadi file yang dapat dijalankan pada perangkat Android, serta berbagai API dan framework untuk memperluas kemampuan aplikasi.

2.11.2 *Java Development Kit (JDK)*

JDK adalah paket fungsi API untuk bahasa pemrograman Java, termasuk *Java Runtime Environment (JRE)* dan *Java Virtual Machine*. JDK adalah perangkat lunak yang digunakan untuk menerjemahkan kode Java menjadi bytecode yang dapat dipahami dan dijalankan oleh JRE (Chia & Susanti, 2019). JDK adalah seperangkat perangkat lunak yang digunakan untuk mengembangkan perangkat lunak berbasis Java, sedangkan JRE adalah implementasi dari *Java Virtual Machine* yang digunakan untuk menjalankan program Java. Setiap JDK berisi satu atau lebih

JRE dan berbagai alat pengembangan lainnya, seperti compiler java, bundling, debuggers, development libraries dan lain sebagainya (Meidika, 2020).

Berdasarkan uraian diatas, dapat disimpulkan JDK adalah seperangkat perangkat lunak yang digunakan untuk mengembangkan aplikasi Java yang menjadi komponen penting dalam pengembangan aplikasi Android menggunakan bahasa pemrograman Java. Android JDK memungkinkan para pengembang untuk mengambil keuntungan dari banyak fitur dan sumber daya yang tersedia dalam JDK untuk membuat aplikasi Android yang efektif dan efisien.

2.12 ISO/ICE 25010

ISO/IEC adalah suatu standar yang digunakan secara internasional untuk mengevaluasi dan mengukur kualitas perangkat lunak. ISO/IEC versi 25010 merupakan versi terbaru dari ISO/IEC 9126. Versi tersebut telah diperluas dengan penambahan beberapa struktur dan bagian dari model standar kualitas (Rauf Abdur & Prastowo Agung Tri, 2021). ISO/IEC 25010 menyajikan karakteristik kualitas yang perlu dipertimbangkan dalam evaluasi perangkat lunak. Dalam standar tersebut terdapat dua model yang dapat digunakan untuk mengukur kualitas sistem, yaitu model kualitas dalam penggunaan (*quality in use*) dan model kualitas produk perangkat lunak (*software product quality*) (Mulyawan et al., 2021).

Berdasarkan uraian diatas, dapat disimpulkan ISO/IEC 25010 adalah standar internasional yang mengatur tentang kualitas perangkat lunak dan sistem informasi. Standar ini memberikan panduan untuk pengujian dan evaluasi kualitas perangkat lunak berdasarkan serangkaian atribut yang didefinisikan. ISO/IEC 25010, yang juga dikenal sebagai SQuaRE (Software Product Quality Requirements and Evaluation), menggantikan standar sebelumnya ISO/IEC 9126.