

BAB II TINJAUAN PUSTAKA

2.1. Studi Literatur

Sumber referensi penelitian terdahulu yang relevan dengan penelitian ini adalah:

1. Hasil penelitian Nugraha dan Tresnawati (2015) [4] yang berjudul pengembangan aplikasi pengenalan kesenian daerah Indonesia sebagai media pembelajaran berbasis android, memberikan kesimpulan bahwa, aplikasi Pengenalan Kesenian Daerah ini telah berfungsi dengan baik dan sesuai dengan tujuan yaitu membuat aplikasi sebagai media pembelajaran yang menarik dan mempermudah dalam belajar kesenian daerah. Fitur-fitur didalam aplikasi ini dapat berjalan sesuai dengan kebutuhan pengguna yaitu terdapat materi pemutar lagu daerah dan demo alat musik.
2. Hasil kajian Tyas dan Suryawan (2014) [5] yang berjudul aplikasi android sebagai media pembelajaran interaktif budaya provinsi di Indonesia, menyatakan bahwa terciptanya aplikasi *android* sebagai media pembelajaran interaktif provinsi di Indonesia Berdasarkan pengujian yang dilakukan pada responden melalui kuisioner didapatkan hasil yaitu sebanyak 87% siswa SMP, 85% siswa SMA dan 90% masyarakat umum menyatakan tampilan aplikasi menarik. Sebanyak 85% siswa SMP, 92% siswa SMA dan 86% masyarakat umum menyatakan bahwa aplikasi ini dapat membantu dalam pencarian informasi provinsi di indonesia. Sebanyak 88% siswa SMP, 87% siswa SMA dan 90% masyarakat umum menyatakan bahwa aplikasi ini sangat baik dan bermanfaat. Sehingga tujuan dari penelitian ini untuk membuat dan merancang aplikasi pembelajaran interaktif informasi budaya provinsi di indonesia yang menarik dan bermanfaat dalam membantu pencarian tentang kebudayaan yang ada di tiap provinsi di Indonesia bagi semua kalangan masyarakat luas telah berhasil tercapai.

2.2. Studi Pustaka

2.2.1. Seni Ukir Jepara

Seni selalu berada di tengah-tengah lapisan masyarakat tanpa memdeveloerpng perbedaan status, waktu maupun tempat. Dalam pegertian dari kalangan atas atau kalangan bawah, dari zaman primitif hingga zaman modern, seni senantiasa hadir dalam berbagai bentuknya.

Kata “ukir” dalam bahasa Inggris “*carving*” yang berarti ukiran. Kata pahatan atau mengukir kayu dapat persamaan bahasa *woodcarving*. Ada pekerjaan *carve*, bagian bahan yang tidak diperlukan akan dibuang, seperti halnya kalau kita bekerja dengan batu atau marmer, untuk memperoleh bentuk yang dikehendaki, Sunaryo (2012) [6]. Tujuan untuk mengurangi atau membuang bagian yang tidak diperlukan yaitu untuk membentuk ornament sesuai dengan gagasan seniman sehingga menimbulkan artistik dalam ukiran tersebut.

Ukiran kayu adalah hasil suatu gambaran yang dilaksanakan dengan alat-alat pahat kayu, sehingga permukaan yang asal mulanya rata menjadi tidak rata (*kruwikan* dan *buledan*), Syafi'i dan Rohidi (2012) [7].

Ciri utama ukiran adalah bentuk permukaan yang tadinya rata menjadi tidak rata, setelah mengalami proses pengurangan terdapat tonjolan motif yang mempunyai kedalaman bervariasi dan akan menimbulkan nilai artistik bagi yang menikmati. Seni ukir sebagai bagian seni rupa seringkali diklasifikasikan ke dalam seni kerajinan atau kriya. Menurut kata harafiahnya, kerajinan dilahirkan oleh sifat rajin manusia. Tetapi titik berat perbuatan manusia bukan dikarenakan sifat rajin, tetap lahir dari sifat terampil atau *keprigelan* tangan. Kata lain dari seni kerajinan adalah seni kria atau kriya hasta yang dalam bahasa asing *handycraft*, Kustandi (2012) [8].

Menurut Sunaryo (2012) [9] jenis seni ukir berdasarkan tinggi rendah ukiran yang dihasilkan oleh para seniman atau pengrajin ada enam macam, yaitu:

1. Ukir rendah (*bas relief*), disebut ukir rendah karena gambar yang timbul kurang dari separo belah bentuk utuhnya. Contoh: ukir gaya Bali. Seniman Bali pada umumnya mahir membuat ukir rendah ini pada dinding-dinding pura.

2. Ukir sedang (*mezzo relief*), disebut ukir sedang karena gambar yang timbul tepat separo belah bentuk utuhnya.
3. Ukir tinggi (*haut relief*), disebut ukir tinggi karena gambar yang timbul lebih dari separo belah bentuk utuhnya.
4. Ukir cekung atau ukir tenggelam (*encreux relief*), disebut ukir cekung karena gambarnya tenggelam lebih rendah daripada bidang dasarnya.
5. Ukir tembus atau ukir krawangan (*ayour relief*), disebut demikian karena gambarnya menembus bidang dasar, sehingga berupa lubang-lubang gambar atau krawangan. Ada kalanya juga yang tembus bukan gambarnya tetapi dasarnya. Ukir tumpang, disebut demikian karena gambarnya tumpang tindih di atas bidang dasar. Ukir tumpang serupa dengan relief patung karena gambarnya utuh seperti patung. Contoh: relief patung terkenal adalah relief kamadhatu di kaki candi Borobudur.
6. Ukir Berundak, disebut demikian karena ukiran dibuat berundak-undak. Jenis-jenis seni ukir tersebut di atas terutama terdapat di Jepara. Sedangkan daerah-daerah lain pada umumnya menghasilkan seni ukir rendah atau ukir sedang saja.

2.2.2. Android

Pengembangan aplikasi ini berbasis pada sistem operasi Android. Android merupakan sistem operasi untuk perangkat mobile yang bersifat open-source berbasis Linux. Platform ini memungkinkan pengembang untuk membuat program atau aplikasi dengan menggunakan bahasa pemrograman Java. Android sendiri memiliki berbagai fitur diantaranya *storage*, *connectivity*, *messaging*, *web browser*, *media*, *supporting hardware*, *multitouch*, *multitasking*, serta *tethering*. Beberapa peralatan yang dibutuhkan untuk mengembangkan sebuah aplikasi berbasis Android yaitu *Android Studio*, *Android SDK*, *Gradle*, dan *Android Virtual Devices*, Lee (2011) [10].

1. Android Studio.

Android Studio adalah Lingkungan Pengembangan Terpadu - Integrated Development Environment (IDE) untuk pengembangan aplikasi

Android, berdasarkan IntelliJ IDEA . Selain merupakan editor kode IntelliJ dan alat pengembang yang berdaya guna, Android Studio menawarkan fitur lebih banyak untuk meningkatkan produktivitas Developer saat membuat aplikasi Android, misalnya:

- a. Sistem versi berbasis Gradle yang fleksibel
- b. Emulator yang cepat dan kaya fitur
- c. Lingkungan yang menyatu untuk pengembangan bagi semua perangkat Android
- d. Instant Run untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru
- e. Template kode dan integrasi GitHub untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh
- f. Alat pengujian dan kerangka kerja yang ekstensif
- g. Alat Lint untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah-masalah lain
- h. Dukungan C++ dan NDK
- i. Dukungan bawaan untuk Google Cloud Platform, mempermudah pengintegrasian Google Cloud Messaging dan App Engine

2. Android SDK

Perlengkapan Pengembangan Perangkat Lunak Android (dalam Perjanjian Lisensi disebut sebagai "SDK" dan secara khusus termasuk file sistem Android, API terpaket, dan add-on API Google) yang lisensinya diberikan kepada Developer tunduk pada Perjanjian Lisensi. Perjanjian Lisensi merupakan kontrak yang mengikat secara hukum antara Developer dan Google sehubungan dengan penggunaan SDK.

Android SDK merujuk pada tumpukan perangkat lunak Android untuk perangkat, yang disediakan oleh Proyek Sumber Terbuka Android, yang terdapat dalam URL berikut: <http://source.android.com/>, dan diperbarui dari waktu ke waktu. "Penerapan yang kompatibel" adalah perangkat Android yang (i) sesuai dengan dokumen Definisi Kompatibilitas Android, yang terdapat di situs web kompatibilitas Android

(<http://source.android.com/compatibility>) dan dapat diperbarui dari waktu ke waktu; dan (ii) lulus Rangkaian Uji Kompatibilitas Android (CTS). "Google" adalah Google LLC, sebuah perusahaan di Delaware dengan lokasi bisnis utama di 1600 Amphitheater Parkway, Mountain View, CA, 94043, Amerika Serikat.

Untuk menggunakan SDK, Developer harus menyetujui Perjanjian Lisensi terlebih dahulu. Developer tidak dapat menggunakan SDK jika tidak menerima Perjanjian Lisensi. Dengan mengklik terima, berarti Developer menyatakan setuju terhadap persyaratan Perjanjian Lisensi. Developer tidak dapat menggunakan SDK dan tidak dapat menerima Perjanjian Lisensi jika Developer adalah orang yang dilarang menerima SDK menurut undang-undang Amerika Serikat atau negara lain, termasuk negara domisili atau tempat asal Developer mengakses SDK. Jika Developer bersedia untuk terikat oleh Perjanjian Lisensi atas nama atasan atau entitas lain, Developer mewakili dan menjamin bahwa Developer memiliki otoritas hukum penuh untuk mengikat atasan atau entitas semacam ini dengan Perjanjian Lisensi. Jika tidak memiliki kewenangan yang diperlukan, Developer tidak dapat menerima Perjanjian Lisensi atau menggunakan SDK atas nama atasan atau entitas lain.

Sesuai dengan persyaratan Perjanjian Lisensi, Google memberi Developer lisensi terbatas, berlaku di seluruh dunia, bebas royalti, tidak dapat dialihkan, tidak eksklusif, dan tidak dapat disublisensikan, untuk menggunakan SDK semata-mata guna mengembangkan aplikasi untuk penerapan yang kompatibel bagi Android. Developer tidak dapat menggunakan SDK untuk mengembangkan aplikasi bagi platform lain (termasuk penerapan yang tidak kompatibel untuk Android) atau untuk mengembangkan SDK lain.

Developer bebas mengembangkan aplikasi untuk platform lain, termasuk penerapan yang tidak kompatibel untuk Android, asalkan SDK ini tidak digunakan untuk tujuan tersebut. Developer setuju bahwa Google atau pihak ketiga memiliki semua hak hukum, judul, dan minat dalam dan untuk

SDK, termasuk Hak atas Kekayaan Intelektual yang terdapat dalam SDK. "Hak atas Kekayaan Intelektual" adalah segala dan semua hak di bawah undang-undang paten, undang-undang hak cipta, undang-undang rahasia dagang, undang-undang merek dagang, dan semua hak kekayaan lain. Google memiliki semua hak yang tidak diberikan kepada Developer secara tersurat.

Developer tidak diperbolehkan menggunakan SDK untuk tujuan apa pun selain yang diizinkan oleh Perjanjian Lisensi secara tersurat. Kecuali sejauh yang diperlukan oleh lisensi pihak ketiga yang berlaku, Developer tidak diperbolehkan menyalin (kecuali untuk tujuan pencadangan), mengubah, menyesuaikan, mendistribusikan ulang, mendekompilasi, merekayasa balik, membongkar, atau membuat karya turunan SDK atau bagian SDK. Penggunaan, produksi ulang, dan distribusi komponen SDK yang diberi lisensi di bawah lisensi perangkat lunak sumber terbuka diatur semata-mata oleh persyaratan lisensi perangkat lunak sumber terbuka tersebut dan bukan Perjanjian Lisensi. Developer setuju bahwa bentuk dan sifat SDK yang diberikan Google dapat berubah tanpa pemberitahuan sebelumnya kepada Developer, dan versi SDK mendatang mungkin tidak kompatibel dengan aplikasi yang dikembangkan di versi SDK sebelumnya. Developer setuju bahwa Google dapat berhenti (baik permanen atau sementara) memberikan SDK (atau fitur apa pun dalam SDK) kepada Developer atau kepada pengguna secara umum berdasarkan kebijakan Google sendiri, tanpa pemberitahuan sebelumnya kepada Developer.

Perjanjian Lisensi tidak memberi Developer hak untuk menggunakan nama produk, merek dagang, tdeveloper layanan, logo, nama domain, atau fitur merek khusus Google yang lain. Developer setuju bahwa Developer tidak akan menghapus, mengaburkan, atau mengubah pemberitahuan hak kekayaan apa pun (termasuk pemberitahuan hak cipta dan merek dagang) yang dapat disertakan atau terdapat dalam SDK.

Google setuju bahwa Google tidak memiliki hak, hak milik, atau minat dari Developer (atau pemberi lisensi Developer) di bawah Perjanjian Lisensi pada atau untuk aplikasi perangkat lunak yang Developer

kembangkan menggunakan SDK, termasuk hak atas kekayaan intelektual yang terdapat dalam aplikasi tersebut.

Developer harus setuju untuk menggunakan SDK hanya untuk tujuan yang diizinkan menurut (a) Perjanjian Lisensi dan (b) undang-undang, peraturan, pedoman, atau praktik apa pun yang diterima secara umum dan berlaku di wilayah hukum terkait (termasuk undang-undang apa pun tentang ekspor data atau perangkat lunak ke dan dari Amerika Serikat, maupun negara lainnya yang relevan). Developer setuju bahwa jika Developer menggunakan SDK untuk mengembangkan aplikasi, Developer akan melindungi privasi dan hak hukum pengguna. Apabila pengguna memberikan nama pengguna, sandi, atau informasi proses masuk atau informasi pribadi lainnya, Developer harus mengingatkan mereka bahwa informasi akan tersedia bagi aplikasi Developer, dan Developer harus menyediakan pemberitahuan dan perlindungan privasi yang memadai secara hukum bagi mereka. Jika aplikasi Developer menyimpan informasi pribadi atau sensitif yang diberikan oleh pengguna, hal itu harus dilakukan dengan aman.

Jika pengguna memberikan informasi Akun Google-nya kepada aplikasi Developer, aplikasi Developer hanya boleh menggunakan informasi tersebut untuk mengakses Akun Google pengguna apabila, serta untuk tujuan terbatas dan pengguna telah mengizinkan Developer untuk melakukannya. Developer setuju bahwa Developer tidak akan terlibat dalam aktivitas apa pun dengan SDK, termasuk pengembangan dan distribusi aplikasi, yang mengganggu, mengacaukan, merusak, atau mengakses tanpa izin server, jaringan, atau properti atau layanan lain dari pihak ketiga yang mencakup, tetapi tidak terbatas pada, Google atau operator komunikasi seluler mana pun. Developer setuju untuk bertanggung jawab penuh (dan Google tidak bertanggung jawab kepada Developer atau pihak ketiga mana pun) terhadap data, konten, atau sumber daya yang dibuat, ditransmisikan, atau ditampilkan melalui Android dan/atau aplikasi untuk Android, dan konsekuensi atas tindakan Developer (termasuk kerugian maupun kerusakan apa pun yang mungkin dialami Google) karena melakukannya. Developer setuju bahwa

Developer bertanggung jawab penuh (dan Google tidak bertanggung jawab kepada Developer atau pihak ketiga mana pun) atas semua pelanggaran kewajiban berdasarkan Perjanjian Lisensi, kontrak atau persyaratan layanan pihak ketiga apa pun yang berlaku, atau undang-undang atau peraturan apa pun yang berlaku, dan atas konsekuensi (termasuk kerugian maupun kerusakan apa pun yang mungkin dialami Google atau pihak ketiga mana pun) akibat pelanggaran tersebut.

3. Gradle

Android menggunakan Gradle sebagai dasar sistem versi, dengan kemampuan khusus Android yang disediakan oleh Plugin Android untuk Gradle. Sistem ini bisa dijalankan sebagai alat terpadu dari menu Android Studio dan secara independen dari baris perintah. Anda bisa menggunakan fitur-fitur sistem versi untuk melakukan yang berikut:

- a. Menyesuaikan, mengonfigurasi, dan memperluas proses pembangunan.
- b. Membuat beberapa APK untuk aplikasi Android Anda, dengan aneka fitur menggunakan proyek dan modul yang sama. Menggunakan kembali kode dan sumber daya pada seluruh set sumber.

Dengan menerapkan fleksibilitas Gradle, Anda dapat mencapai semua ini tanpa mengubah file sumber inti aplikasi. File versi Android Studio diberi nama `build.gradle`. File ini adalah teks biasa yang menggunakan Groovy mengonfigurasi versi dengan elemen yang disediakan oleh plugin Android untuk Gradle. Masing-masing proyek memiliki file versi level atas untuk seluruh proyek dan file versi level modul terpisah untuk setiap modul. Saat Anda mengimpor proyek saat ini, Android Studio otomatis menghasilkan file versi yang diperlukan.

Untuk mengetahui selengkapnya tentang versi sistem dan cara mengonfigurasi, lihat [Mengonfigurasi Versi](#). Sistem versi dapat membantu Anda membuat versi berbeda dari aplikasi yang sama dari satu proyek. Ini berguna ketika Anda sama-sama memiliki versi gratis dan versi berbayar dari aplikasi, atau jika Anda ingin mendistribusikan beberapa APK untuk perangkat berbeda di Google Play. Untuk informasi selengkapnya tentang

mengonfigurasi varian versi, lihat Mengonfigurasi Versi Gradle.

4. Android Virtual Device

Android Virtual Device (AVD) memungkinkan Anda mendefinisikan karakteristik dari ponsel Android, tablet, Android Wear, atau perangkat Android TV yang ingin Anda simulasikan di Android Emulator. AVD Manager memudahkan Anda dalam membuat dan mengelola AVD.

Untuk secara efektif menguji aplikasi, Anda harus membuat sebuah AVD yang memodelkan setiap tipe perangkat yang didesain untuk didukung oleh aplikasi. Misalnya, kami sarankan Anda membuat AVD untuk setiap tingkat API yang setara dan lebih tinggi dari nilai yang telah ditetapkan untuk `minSdkVersion`. Pengujian dengan tingkat API yang lebih tinggi dari yang dibutuhkan oleh aplikasi Anda akan memastikan kompatibilitas aplikasi di masa mendatang ketika pengguna mengunduh pemutakhiran sistem.

Sebuah AVD berisi profil perangkat keras, citra sistem, tempat penyimpanan, skin, dan properti lainnya. Profil perangkat keras mendefinisikan karakteristik perangkat ketika didistribusikan dari pabrik. AVD Manager didistribusikan dengan profil perangkat keras yang dipramuat, seperti perangkat ponsel Nexus, dan Anda bisa menentukan dan mengimpor profil perangkat keras yang diperlukan. Anda bisa mengganti beberapa setelan di AVD, jika diperlukan.

AVD Manager membantu Anda memilih citra sistem untuk AVD dengan memberikan rekomendasi. Ini juga memungkinkan Anda untuk mengunduh citra sistem, beberapa dengan pustaka pengaya, seperti Google API, yang mungkin diperlukan oleh aplikasi. Perhatikan bahwa citra sistem x86 berjalan paling cepat di emulator.

AVD memiliki tempat penyimpanan khusus pada mesin pengembangan Anda. Ini menyimpan data pengguna perangkat, seperti aplikasi yang dipasang dan setelan, serta kartu SD yang diemulasikan. Skin emulator menentukan tampilan perangkat. AVD Manager menyediakan beberapa skin yang telah ditentukan. Anda juga bisa menentukan sendiri, atau menggunakan skin yang disediakan oleh pihak ketiga.

Sama seperti dengan perangkat sungguhan, agar aplikasi bisa menggunakan fitur tertentu yang didefinisikan dalam AVD, misalnya kamera, aplikasi harus memiliki setelan `<uses-feature>` yang sesuai dalam manifest aplikasi.

Sistem operasi Android merupakan sistem operasi populer di dunia yang dapat menjalankan berbagai macam perangkat mulai dari smartphone, jam tangan, tablet, televisi dan kaca mata (Rio, 2016) [11]. Android memiliki beberapa versi yang telah resmi rilis tertera pada tabel versi platform Android bersumber dari situs resmi pengembang Android dalam tabel berikut.



Version	Codename	API	Distribution
2.2	Froyo	8	0.2%
2.3.3 - 2.3.7	Gingerbread	10	3.0%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	2.7%
4.1.x	Jelly Bean	16	9.0%
4.2.x		17	12.2%
4.3		18	3.5%
4.4	KitKat	19	36.1%
5.0	Lollipop	21	16.9%
5.1		22	15.7%
6.0	Marshmallow	23	0.7%

Gambar 2.1. Versi platform Android

Android ini bersifat open-source, karena sistem informasi ini banyak mendapat perhatian pengembang. Pengembang yang melakukan pengembangan aplikasi pada platform Android telah disediakan tata cara mekanisme pengembangan melalui halaman website resmi di <http://developer.android.com>.

Banyaknya versi pada platform Android yang telah resmi rilis, pengembangan sistem informasi gaya belajar menggunakan versi android 2.3 ke atas sebagai versi pengembangannya agar distribusi pengguna dapat menggunakan sistem informasi gaya belajar bersifat luas dan menyeluruh, serta telah didukungnya penggunaan konektivitas yang lebih baik mulai versi ini.

2.2.3. Kriteria Penilaian Kelayakan

Pengukuran kualitas perangkat lunak dapat dilakukan dengan menggunakan berbagai macam model. Model penilaian yang digunakan dalam penelitian ini adalah ISO 25010 Model. Model ISO dipilih karena merupakan merupakan stdeveloperr internasional yang berlaku saat ini. Model ISO ini menentukan delapan karakteristik termasuk functional suitability, reliability, usability, maintainability, security, compability, dan portability yang dibagi lagi menjadi serangkaian sub-karakteristik (Wagner, 2013) [12].

Dalam penelitian ini, pengujian produk dilakukan sesuai stdeveloperr ISO 25010. Pengujian untuk mobile application meliputi empat aspek yaitu functional testing, compatibility testing, dan usability testing. Maka dari itu peneliti hanya mengambil ketiga aspek tersebut untuk pengujian produk ini demi kelayakan aplikasi.

Adapun keempat aspek yang digunakan untuk pengujian adalah sebagai berikut:

1. Functional Testing

Functional testing digunakan untuk memvalidasi fungsi aplikasi sesuai dengan syarat yang dibutuhkan. Dalam stdeveloperr ISO 25010 mewakili aspek functional suitability. Sebuah perangkat lunak memenuhi aspek fuctional suitability jika perangkat tersebut mampu menampilkan halaman utama dengan muncul semua fitur seperti fitur kamera, hasil pendeteksian kamera, fitur materi- materi, fitur petunjuk penggunaan dan fitur keluar untuk meninggalkan perangkat yang digunakan.

2. Compatibility Testing

Compatibility testing dilakukan dengan cara menjalankan aplikasi menggunakan berbagai macam variasi browser, OS, jenis device, ukuran device, dan kecepatan koneksi. Dalam stdeveloperr ISO 25010 mewakili aspek compatibility. Sebuah perangkat lunak memenuhi aspek compatibility jika perangkat tersebut mampu dijalankan dengan baik tanpa ada kesalahan pada berbagai macam perangkat smartphone berbasis android

seperti Asus Zenfone 5, Sony Xperia, Xioami Redmi Note 3, Samsung Grand Dous dan perangkat smartphone android yang lain.

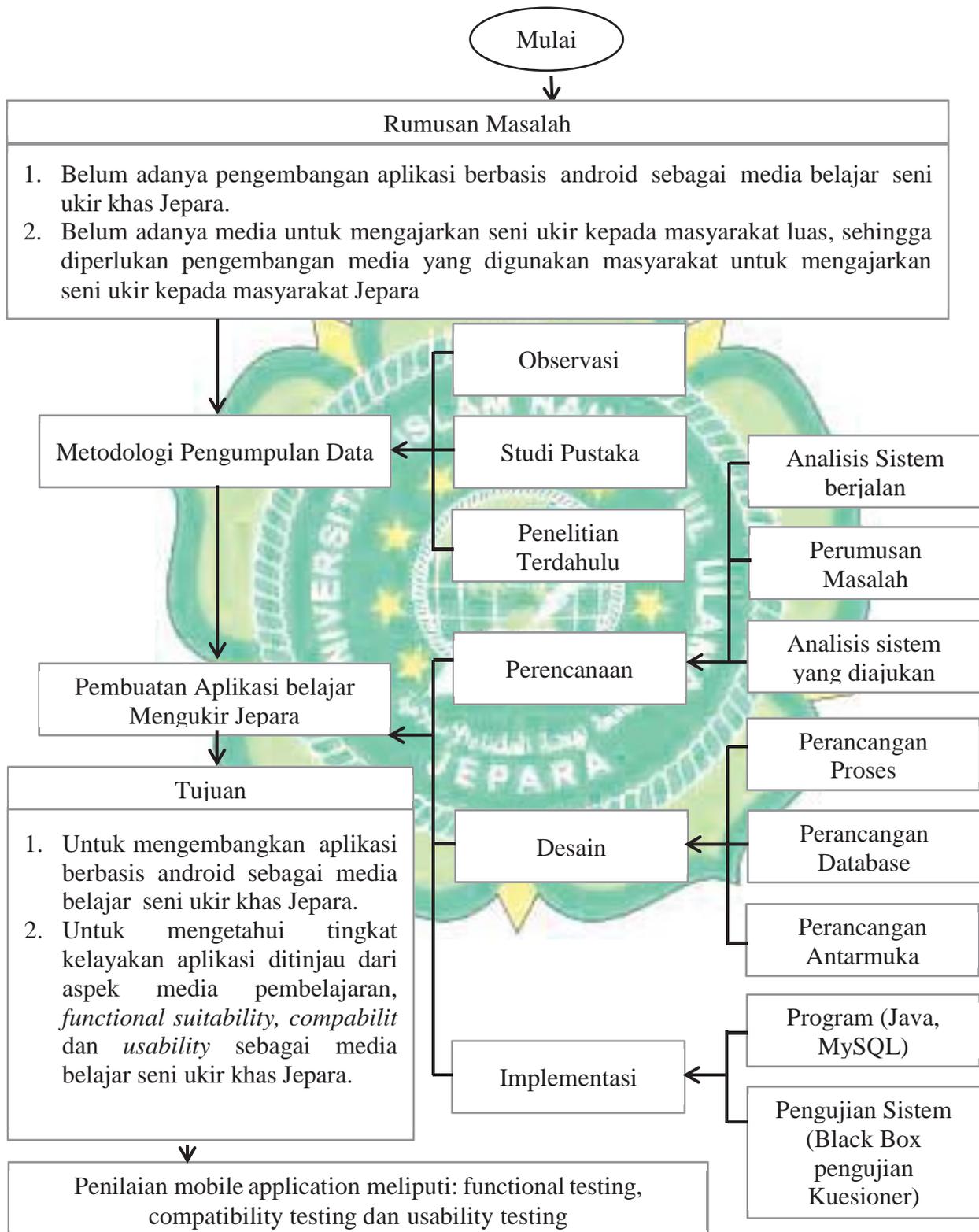
3. Usability Testing

Usability testing digunakan untuk menguji kepada pengguna akhir mengenai penggunaan aplikasi. Dalam stdeveloperr ISO 25010 mewakili aspek usability. Sebuah perangkat lunak memenuhi aspek usability jika perangkat tersebut memenuhi 4 kriteria yaitu usefulness atau kegunaan, easy of use atau kemudahan penggunaan, ease of learning atau kemudahan belajar, dan *satisfaction* atau kepuasan.

2.3.Kerangka Pemikiran

Aplikasi belajar mengukir khas Jepara ini bertujuan untuk mengembangkan aplikasi berbasis android sebagai media belajar seni ukir khas Jepara. Aplikasi dibangun dengan beberapa tahap yakni tahap analisis, tahap desain, tahap implementasi, dan tahap pengujian. Dalam tahap analisis dilakukan pengumpulan data/informasi yang dibutuhkan untuk membangun Aplikasi belajar mengukir khas Jepara. Hasil analisis tersebut kemudian dijadikan pedoman untuk melakukan tahap desain. Tahap desain meliputi desain UML dan desain interface. Berikut diagram yang menggambarkan kerangka pikir:

Kerangka Pemikiran



Gambar 2.2. Kerangka Pemikiran