

BAB III METODOLOGI PENELITIAN

3.1 Kerangka Pikir

Berikut pada gambar 3.1 adalah kerangka pikir dari penelitian.



Gambar 3.1 Kerangka Pikir.

3.2 Persiapan Pra-pengembangan

Sebelum melakukan pengembangan aplikasi, dilakukan terlebih dulu studi pustaka dari beberapa sumber untuk mengetahui prinsip-prinsip dari masing-masing teknologi yang hendak digunakan serta juga mempelajari berbagai macam *library* yang dapat berguna dalam proses pengembangan. Baru kemudian dilakukan eksperimen secara kecil dengan beberapa teknologi dan *library* yang hendak digunakan sampai setidaknya dapat melakukan fungsi sederhana dengan sesuai yang diharapkan.

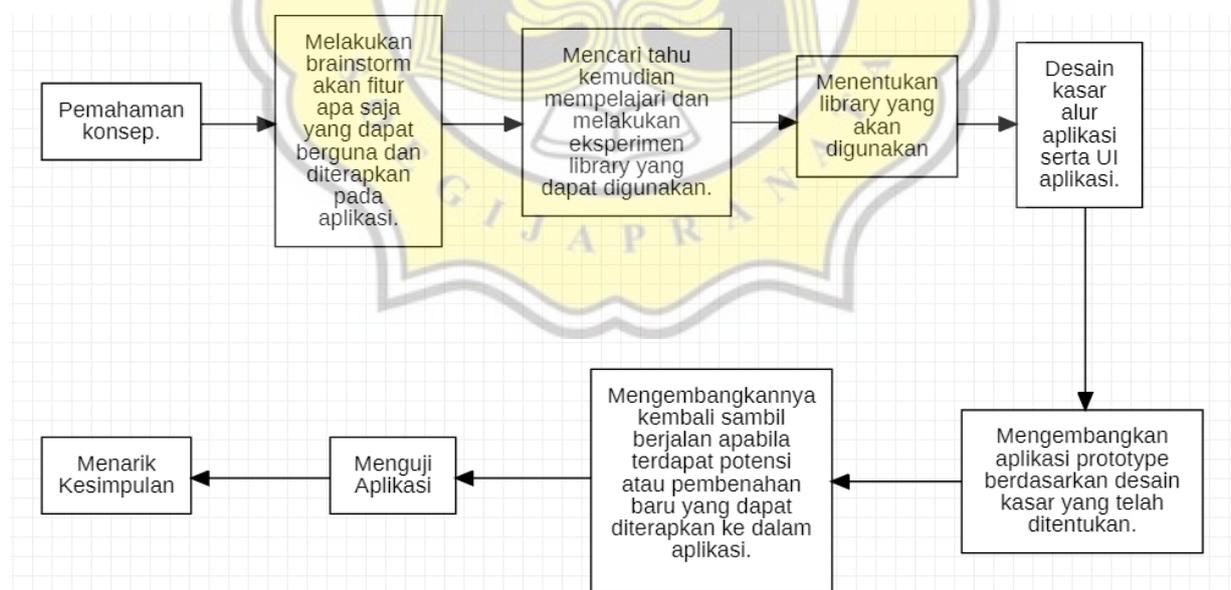
Baru setelah mengetahui apabila *library* yang digunakan sudah dapat dianggap layak dan dapat memberikan hasil sesuai yang diharapkan, dilakukan pemikiran dan perencanaan bagaimana alur dari proses program yang akan dibuat dengan memperhatikan juga potensi-potensi dari *library* yang digunakan. Baru kemudian dilanjutkan ke proses pembuatan aplikasi.

3.2 Pengembangan Aplikasi

Sebelum aplikasi dapat dibuat, diperlukan desain kasar berupa mockup sebagai referensi kira-kira seperti apa alur dan user *interface* dari aplikasi akan dibuat. Setelah dirancang kira-kira alur proses yang akan ada dalam aplikasi, pembuatan aplikasi dapat dilakukan berdasarkan alur kerja yang disasar. Pembuatan dilakukan secara paralel dengan mengerjakan sistem *back-end* dan sistem *front-end* secara bergantian. Dikarenakan karena beberapa proses memang harus melibatkan *front-end* agar dapat diujikan dengan mudah saat proses pengembangan. Tapi ada juga proses yang dapat diujikan menggunakan *tools* seperti Postman untuk menguji pengiriman dan penerimaan informasi yang tidak terlalu bergantung kepada aplikasi *front-end*.

Dalam proses pengembangan aplikasi tidak jarang juga terjadi perubahan akan alur yang dirancang dikarenakan beberapa pertimbangan selama pengerjaan dan juga makin bertambahnya pengetahuan dan wawasan akan prinsip-prinsip kerja masing-masing *library* yang digunakan dan juga akan teknologi yang sedang digunakan saat proses pengerjaan. Semua dilakukan dengan pertimbangan agar sistem dan aplikasi yang dikembangkan dapat menjadi lebih baik dari sebelumnya dan lebih menjawab kebutuhan.

Secara garis besar, berikut pada gambar 3.2 adalah runtut keseluruhan pada proses pengembangan aplikasi.



Gambar 3.2 Alur pengembangan aplikasi.

3.3 Pengujian Keamanan Aplikasi

Pengujian keamanan aplikasi dilakukan untuk melihat bagaimana kinerja maupun hasil keluaran dari aplikasi dalam menghadapi masalah yang mungkin dapat terjadi seperti serangan ataupun penyalahgunaan informasi atau data.

3.3.1 Pengujian terhadap konsistensi data Hyperledger Iroha

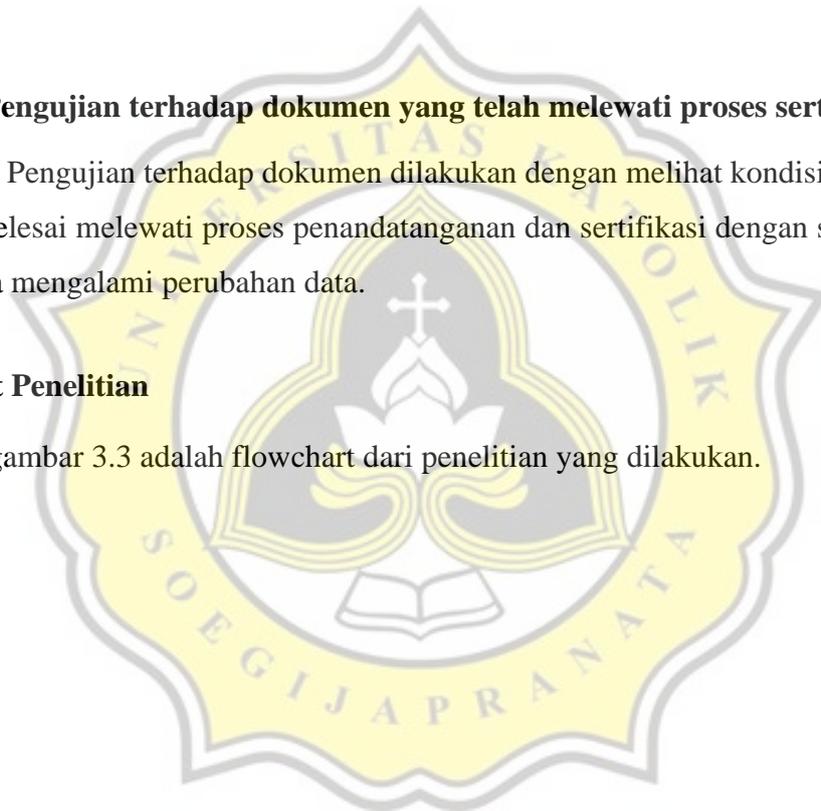
Pengujian terhadap Hyperledger Iroha dilakukan dengan melakukan simulasi perubahan data pada *blockstore* dan “*World State View*” yang mana adalah serangkaian tempat penyimpanan informasi pada masing-masing peer yang ada pada Hyperledger Iroha.

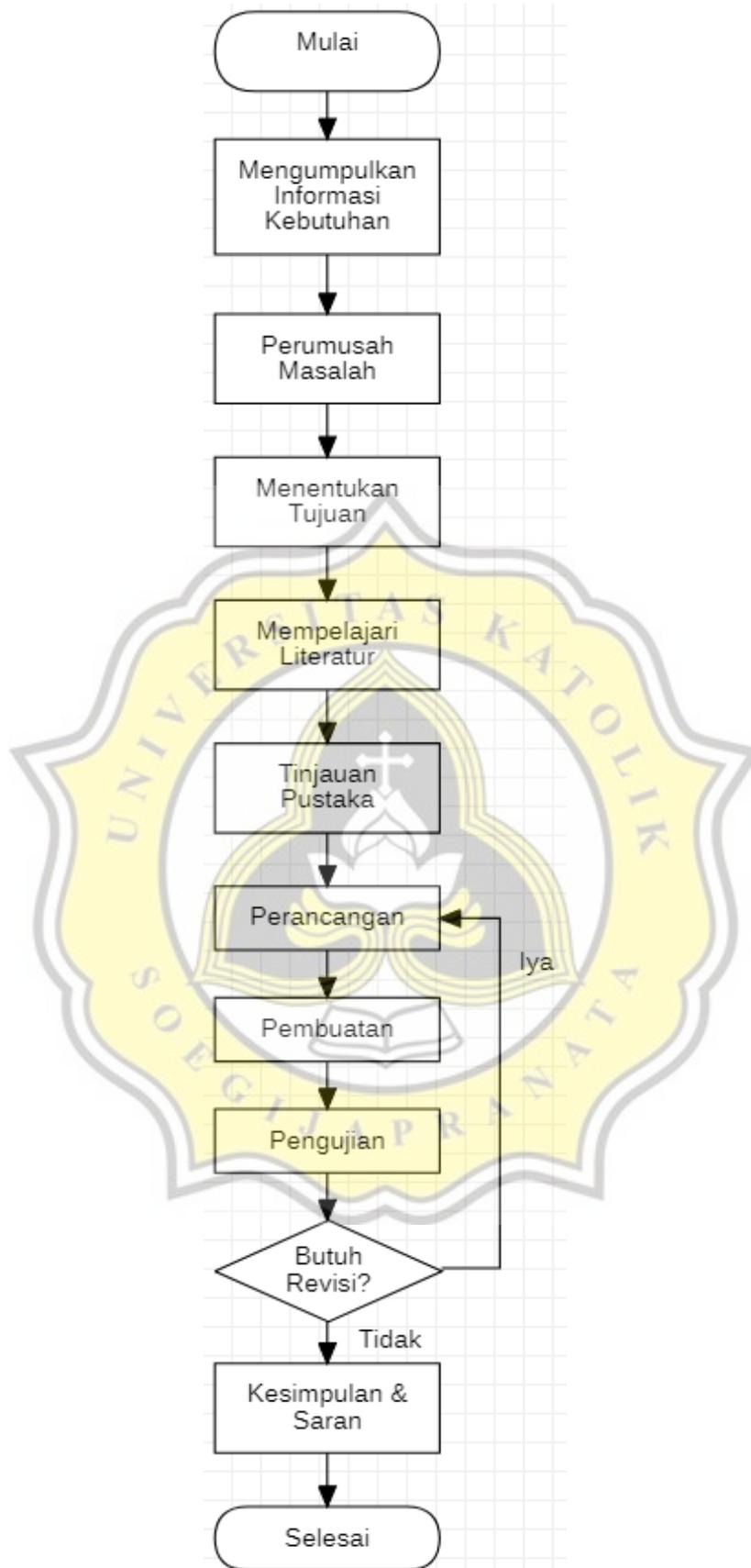
3.3.2 Pengujian terhadap dokumen yang telah melewati proses sertifikasi

Pengujian terhadap dokumen dilakukan dengan melihat kondisi dokumen yang telah selesai melewati proses penandatanganan dan sertifikasi dengan sertifikat *digital* apabila mengalami perubahan data.

3.4 Flowchart Penelitian

Berikut pada gambar 3.3 adalah flowchart dari penelitian yang dilakukan.





Gambar 3.3 Flowchart Penelitian.

3.5 Konsep Aplikasi

Aplikasi yang dibuat adalah aplikasi yang dapat menandatangani dokumen berformat PDF. Tanda tangan adalah tanda tangan dari pengguna aplikasi tersebut yang dapat disimpan ke dalam *smartphone* masing-masing pengguna, dan dapat digunakan untuk menandatangani sebuah dokumen yang diinginkan oleh pengguna. Kemudian setelah pengguna menyematkan tanda tangan mereka, dokumen tersebut akan dikirimkan ke sistem untuk kemudian dilakukan proses sertifikasi dan penguncian dokumen serta mencatat aktivitas (transaksi) ke dalam sistem *blockchain*, kemudian aplikasi akan mendapat balasan dari sistem yang akan mengembalikan hasil dokumennya ke pengguna setelah semua proses terlewati dan terselesaikan. Aplikasi difokuskan terlebih dahulu ke *platform mobile* untuk dapat menjangkau banyak pengguna.

Setiap transaksi harus dapat divalidasi, maka dari itu terdapat juga sistem untuk melihat transaksi yang telah dilakukan oleh pengguna serta juga memvalidasi dokumen yang telah ditandatanganinya. Oleh karena itu dibuat juga menu dalam aplikasi *mobile* untuk melakukan kegiatan validasi ini, dan juga aplikasi *website* sederhana untuk melakukan validasi transaksi sebagai opsi tambahan terhadap *user* yang tidak menggunakan aplikasi *mobile*.

Aplikasi *front-end* yang dikerjakan menggunakan framework berbasis *React*, yaitu menggunakan *Expo*. Alasannya adalah agar aplikasi kelak dapat dikembangkan apabila terdapat permintaan atau rencana untuk dikembangkan ke *platform* lain seperti *website* ataupun PC. Sedangkan *back-end* yang menangani *request* dari *front-end* dikembangkan menggunakan *Node.js*. *Database* menggunakan *MongoDB*, sistem yang menjadi CA (Certificate Authority), RA (Request Authority), dan VA (Validation Authority) menggunakan *OpenXPKI*, dan sistem blockchain menggunakan *Hyperledger Iroha*.

3.6 Alasan Penggunaan Teknologi

Ada beberapa alasan mengapa beberapa teknologi dipilih dibandingkan dengan beberapa pilihan lainnya yang ada dalam pembuatan sistem dan aplikasi ini.

3.6.1 Sistem REST API dan Front-end berbasis Javascript

Dalam pembuatan sistem REST API dan aplikasi *front-end*, *Node.js* dan *React* dipilih terutama karena dikembangkan menggunakan *JavaScript*. Karena dengan *JavaScript*, dapat membuat aplikasi secara *full stack* dengan cukup dengan mempelajari dan menggunakan satu bahasa pemrograman saja, tanpa harus bingung menggunakan berbagai bahasa pemrograman yang lain. Lain halnya apabila membuat aplikasi yang tidak berbasis ekosistem JavaScript

misalkan sebagai contoh harus menggunakan PHP sebagai sistem REST API atau *back-end*, kemudian mengembangkan *website* menggunakan HTML yang berisikan kode PHP, CSS, atau mungkin juga beberapa kode *JavaScript*, kemudian apabila hendak membuat aplikasi *mobile* harus mempelajari untuk menggunakan *Java* atau *Kotlin* apabila hendak menasar pengguna *Android*, dan mempelajari *Swift* atau *Objective-C* apabila hendak menasar pengguna *iOS*. *JavaScript* memberikan solusi yang cukup mempercepat proses baik dalam *learning curve* dan juga pengembangan aplikasi karena penggunaan bahasa yang sama dalam mengembangkan dua sisi aplikasi yang berbeda (*server side* dan *client side*). Karena dengan adanya *Node.js* seseorang dapat membuat *server side application*, dan dengan *framework* berbasis *React* seseorang dapat membuat *cross-platform client side application* yang berarti bahkan aplikasi *client-side* dapat lebih cepat dibuat baik dalam proses *development* dan *release* untuk beberapa *platform* yang berbeda-beda.

Selain itu *JavaScript* (dengan menggunakan *Node.js*) dapat memberikan respon yang sangat cepat dalam menangani *request* dibandingkan apabila menggunakan PHP yang umumnya juga digunakan dari dulu bahkan di beberapa sistem juga sampai sekarang, karena *JavaScript* dapat melakukan prosesnya secara *asynchronous* yang mana dapat menjalankan beberapa proses atau perintah tanpa harus menunggu satu proses harus terlewati terlebih dahulu (*synchronous*), tetapi juga dimungkinkan untuk membuat agar kode dapat berjalan secara *synchronous* dengan keyword *await* untuk menjalankan perintah secara berurutan (menunggu “*promise*” terlebih dahulu agar terselesaikan (*resolved*) baru kemudian melanjutkan ke proses selanjutnya). Secara teori, *JavaScript* dapat memberikan performa yang lebih cepat karena dapat melakukan perintah secara *asynchronous*, di mana PHP secara *native* hanya dapat melakukan perintah secara *synchronous* (berurutan).

3.6.2 MongoDB sebagai database

MongoDb dipilih karena *MongoDb* adalah *database* yang *non-relational*, yang berarti tidak perlu bergantung pada skema yang *fixed* dan kaku seperti pada database berbasis SQL. Hal ini bagus apabila hendak membuat sistem yang ekstensif dan dapat berkembang dan berubah sewaktu-waktu skema datanya. Selain itu pada satu *collection* (seperti *table* pada *database* berbasis SQL) dapat berisikan dokumen/data yang kolomnya berbeda-beda. Data pada *MongoDb* disimpan dalam bentuk BSON (*binary JSON*), dan datanya juga dapat ditampilkan dan disajikan dalam bentuk objek JSON yang akan membuat proses pengembangan aplikasi lebih mudah dan tidak terlalu menyusahkan (*convenient*). Selain itu *MongoDb* dapat mudah digunakan bersamaan dengan ekosistem aplikasi yang mayoritas

dikembangkan menggunakan *JavaScript* karena *MongoDb* menyediakan *driver/library* yang memungkinkan untuk dapat memberikan perintah ke *MongoDb* menggunakan bantuan kode *JavaScript*.

3.6.3 OpenxPKI

OpenxPKI dipilih karena mudah untuk disiapkan dalam uji coba testing dan telah menyediakan beberapa modul yang siap digunakan (seperti *website*, *RPC API*, dsb) bahkan dengan hanya menggunakan *script sample configuration*-nya saja yang membuat pengembangan aplikasi secara paralel tidak terlalu lambat. Selain itu juga telah disediakan beberapa *endpoint RPC API* yang siap untuk dapat digunakan seperti untuk melakukan *request certificate*, *revoke certificate*, dan sebagainya.

3.6.4 Hyperledger Iroha

Hyperledger Iroha menyediakan sistem *blockchain* yang sederhana, dapat ditentukan berapa jumlah *peer* yang dibutuhkan, dan menyediakan fitur seperti *user management*, *asset management*, dan *role management* yang membuat Hyperledger Iroha cocok untuk diterapkan tidak hanya untuk satu tujuan penggunaan (*one purpose usage*) saja. Hyperledger Iroha juga dapat dikembangkan menggunakan *JavaScript* karena juga terdapat *driver/library* Hyperledger Iroha (*library* “*iroha-helpers*”) yang dapat digunakan dalam proses pengembangan aplikasi *back-end* agar dapat berkomunikasi dengan Hyperledger Iroha.

Selain itu terdapat juga *project* Hyperledger Iroha yang dikembangkan oleh “*Diva.Exchange*” yang mempermudah dalam menyiapkan *setup* awal Hyperledger Iroha, serta juga terdapat “*blockchain explorer*” berbasis *web* yang dapat digunakan untuk menampilkan seluruh aktivitas transaksi *blockchain* yang telah dilakukan. Para pengembang dan pengguna dalam komunitas Hyperledger dan *Diva.Exchange* juga sangat kooperatif apabila terdapat pertanyaan-pertanyaan seputar tentang bagaimana cara menggunakan Hyperledger Iroha apabila mendapati kesulitan dalam mengembangkan Hyperledger Iroha beserta *library*-nya.

Pada Hyperledger Iroha dimungkinkan untuk melakukan transaksi yang dapat dicatat pada sistem *blockchain* misalkan seperti membuat akun, membuat aset, melakukan transfer aset yang dimiliki sebuah akun ke akun lain yang terdaftar pada Hyperledger Iroha, dan masih banyak lagi. Namun perlu diingat bahwa tidak dimungkinkan untuk merubah transaksi yang telah dilakukan tersebut. Dan walaupun apabila hendak merubah informasi, yang dapat dilakukan hanyalah mengganti informasi detail yang terdapat pada masing-masing akun yang terdaftar pada Hyperledger Iroha melalui perintah (*commands*) “*setAccountDetail*”, aktivitas

tersebut juga akan tercatat sebagai transaksi dan riwayat apa yang dilakukan tersebut dapat terlacak pada sistem *blockchain* karena setiap terjadi transaksi, riwayat dari transaksi yang telah dilakukan tersebut akan tersimpan menjadi sebuah *block* yang baru pada sistem *blockchain*.

Secara garis besar, sistem Hyperledger Iroha sudah memenuhi prinsip-prinsip pada *blockchain*, yang mana setiap transaksi harus melewati proses validasi terlebih dahulu berdasarkan apakah pengguna telah terdaftar sebagai pengguna, apakah pengguna mempunyai *role* yang berisikan hak akses untuk melakukan transaksi, dan juga bilamana kegiatan memerlukan *asset* pada pengguna untuk melakukan transaksi, serta kemudian apabila semua telah dilakukan *peers* akan menandatangani transaksi yang telah terjadi sebagai tanda bahwa transaksi telah dilakukan baik apabila transaksi dianggap valid maupun tidak valid (ditolak). Perlu diingat, bahwa penggunaan *blockchain* pada beberapa kasus tertentu tidak sepenuhnya menggantikan fungsi dari *database* karena pada kasus Hyperledger Iroha, terdapat beberapa keterbatasan seperti tipe data apa saja yang dapat disimpan, dan juga terdapat batas jumlah karakter yang dapat disimpan.

