**SISTEM VOTING ELEKTRONIK BERBASIS ETHEREUM SMART CONTRACT**

**TUGAS AKHIR**

Diajukan guna memenuhi sebagian persyaratan dalam rangka menyelesaikan Pendidikan Sarjana Strata Satu (S1) Program Studi Teknologi Informasi



**I DEWA GEDE DIRGA YASA**

**NIM: 1705551062**

**PROGRAM STUDI TEKNOLOGI INFORMASI**

**FAKULTAS TEKNIK**

**UNIVERSITAS UDAYANA**

**2021**

**SISTEM VOTING ELEKTRONIK BERBASIS ETHEREUM SMART CONTRACT**

**TUGAS AKHIR**

Diajukan guna memenuhi sebagian persyaratan dalam rangka menyelesaikan Pendidikan Sarjana Strata Satu (S1) Program Studi Teknologi Informasi



**I DEWA GEDE DIRGA YASA**

**NIM: 1705551062**

**PROGRAM STUDI TEKNOLOGI INFORMASI**

**FAKULTAS TEKNIK**

**UNIVERSITAS UDAYANA**

**2021**

# KATA PENGANTAR

Puji dan syukur saya panjatkan kehadapan Tuhan Yang Maha Esa sehingga penulis dapat menyelesaikan tugas akhir dengan judul **“Sistem Voting Elektronik Berbasis Ethereum Smart Contract”**. Dalam penyusunan Tugas Akhir ini, penulis mendapatkan masukan dan bimbingan dari berbagai pihak. Sehubungan dengan hal tersebut pada kesempatan ini penulis menyampaikan ucapan terima kasih kepada:

1. Prof. Ir. Ngakan Putu Gede Suardana, MT, Ph.D selaku Dekan Fakultas Teknik Universitas Udayana.
2. Dr. Ir. I Made Oka Widyantara, ST., MT., IPM, ASEAN Eng. selaku Koordinator Program Studi Teknologi Informasi Fakultas Teknik Universitas Udayana.
3. Bapak I Putu Arya Dharmadi, S.T.,M.T. selaku dosen pembimbing I yang telah banyak meluangkan waktu memberikan dorongan, bimbingan, tuntunan dan kesabaran selama penyusunan Tugas Akhir ini.
4. Bapak A. A. Kt. Agung Cahyawan Wiranatha, S.T.,M.T. selaku dosen pembimbing II yang juga telah banyak meluangkan waktu memberikan dorongan, bimbingan, tuntunan dan kesabaran selama penyusunan Tugas Akhir ini.
5. Segenap dosen pengajar yang turut memberikan ilmu dan pengetahuan selama penulis menempuh perkuliahan di Program Studi Teknologi Informasi Fakultas Teknik Universitas Udayana.
6. Bapak dan Ibu pegawai di Program Studi Teknologi Informasi Fakultas Teknik Universitas Udayana yang telah membantu selama proses perkuliahan dan administrasi.
7. Orang tua serta anggota keluarga yang telah banyak memberikan motivasi dan dukungan baik jasmani maupun rohani selama penulis menyusun Tugas Akhir ini.
8. Teman-teman seperjuangan dan segenap civitas di Program Studi Teknologi Informasi Universitas Udayana yang telah memberikan sumbangan ide dan dukungan dalam penyusunan Tugas Akhir ini.
9. Berbagai pihak yang belum dapat disebutkan satu-persatu yang juga berperan penting dalam membatu, memberikan sumbangan ide, pemikiran dan dukungan dalam penyusunan Tugas Akhir ini

Penulis menyadari bahwa laporan ini jauh dari sempurna baik dalam materi maupun penulisannya. Berkaitan dengan hal tersebut, maka kritik dan saran yang bersifat membangun dari semua pihak sangat diharapkan. Akhir kata, semoga laporan ini dapat memberikan manfaat bagi semua pihak sesuai dengan yang diharapkan.

Denpasar, November 2020

I Dewa Gede Dirga Yasa

# ABSTRAK

# ABSTRACT

# DAFTAR ISI

# DAFTAR GAMBAR

# DAFTAR KODE PROGRAM

# DAFTAR TABEL

# BAB I PENDAHULUAN

Bab I berisi bahasan mengenai pendahuluan dari penelitian ini. Pendahuluan dijabarkan menjadi beberapa bagian, diantaranya latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika penulisan.

## Latar Belakang

Pengambilan keputusan bersama memiliki beberapa cara dalam pelaksanaannya, diantaranya musyawarah mufakat, voting dan aklamasi. Masing-masing metode pengambilan keputusan memiliki kelebihan dan kekurangan. Metode pengambilan keputusan yang paling sering digunakan adalah voting karena keputusan dapat dihasilkan dalam jangka waktu yang relatif cepat. Kegiatan yang paling sering menggunakan metode voting adalah pemilihan kepala daerah karena voting sesuai dengan prinsip demokrasi yang memberikan kesempatan setiap kalangan untuk memberikan hak suara. Seiring dengan berkembangnya teknologi, kecepatan dalam mengakses informasi adalah prioritas sehingga cara voting konvensional yang menggunakan kertas dan paku bukan menjadi pilihan yang optimal.

Berdasarkan permasalahan di atas, negara-negara di dunia mulai mengembangkan voting elektronik atau yang biasa disebut dengan *e-voting*. *E-voting* memiliki beberapa model yang digunakan yaitu *internet poll site voting*, *kiosk voting* dan *internet voting*. *Internet poll site voting* merupakan model *e-voting* yang menggunakan komputer di lokasi pemilihan dan menggunakan internet untuk mengirim data dari lokasi pemilihan ke otoritas penyelenggara pemilihan. Sehingga pemilih masih harus datang ke lokasi pemilihan untuk melakukan voting. *Kiosk voting* adalah model *e-voting* yang pemilihnya bisa menggunakan komputer di lokasi yang sudah menjalankan kerja sama dengan otoritas penyelenggara pemilihan misalnya komputer di perpustakaan daerah, sekolah dan *mall*. Sedangkan *internet voting* adalah model *e-voting* yang sepenuhnya menggunakan internet sehingga pemilih dapat melakukan voting dari perangkat masing-masing (Muhammad Habibi, 2018.). Model *e-voting internet poll site voting* dan *kiosk voting* dirasa kurang optimal karena walaupun sudah menggunakan teknologi internet, pemilih masih belum bisa menggunakan perangkat yang dimiliki sehingga tetap harus datang ke tempat pemilihan untuk melakukan voting. Saat ini, era dimana kecepatan dan kemudahan akses informasi menjadi prioritas, internet voting menjadi model *e-voting* yang paling optimal untuk dikembangkan.

Beberapa negara yang sudah mengembangkan *e-voting* model internet voting diantaranya Australia, Kanada, Jepang, Jerman dan Inggris (Muhammad Habibi, 2018). *E-voting* berjalan dengan lancar di Australia, Kanada dan Jepang namun gagal di Jerman dan Inggris. *E-voting* juga memiliki beberapa masalah yang membuat *e-voting* mendapatkan penolakan oleh masyarakat yaitu masalah kerahasiaan dan kepercayaan terhadap data. Salah satu teknologi yang dapat mengatasi masalah tersebut adalah *blockchain*.

*Blockchain* merupakan teknologi terdistribusi yang memungkinkan pengguna atau komunitas pengguna untuk menambahkan data yang akan tercatat dalam sebuah jaringan, namun data yang telah ditambahkan tidak dapat diubah kembali. Hal tersebut tentunya menyebabkan *blockchain* dianggap menjadi teknologi yang tangguh dan tidak ada celah untuk melakukan modifikasi ataupun manipulasi data.

## Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan sebelumnya, maka diperoleh beberapa rumusan masalah sebagai berikut.

1. Bagaimana proses kerja *Ethereum Smart Contract* pada Sistem Voting Elektronik*.*
2. Bagaimana perancangan dan proses pembangunan Sistem Voting Elektronik berbasis *Ethereum smart contract*.
3. Bagaimana hasil penelitian dan kinerja Sistem Voting Elektronik berbasis *Ethereum smart contract*.

## Tujuan

Berdasarkan rumusan masalah yang telah dipaparkan sebelumnya, maka diperoleh tujuan sebagai berikut.

1. Memahami proses kerja Sistem Voting Elektronik berbasis *Ethereum smart contract*.
2. Mengetahui perancangan dan proses pembangunan Sistem Voting Elektronik berbasis *Ethereum smart contract*.
3. Mengetahui hasil penelitian dan kinerja Sistem Voting Elektronik berbasis *Ethereum smart contract*.

## Manfaat Penelitian

Penelitian ini dalam pelaksanaannya memiliki beberapa manfaat yang dijabarkan sebagai berikut.

1. Menghilangkan *trust issue* dalam pelaksanaan voting.
2. Meningkatkan keamanan data hasil voting.
3. Menghemat waktu dalam proses pengambilan data voting.
4. Meminimalisir biaya yang perlu dikeluarkan dalam proses pengambilan data voting.
5. Mempermudah proses pengambilan data voting karena dapat dilakukan dari mana saja melalui perangkat pengguna yang terkoneksi internet.

## Batasan Masalah

Penelitian ini dalam pelaksanaannya memiliki beberapa batasan yang bertujuan agar penggunaan aplikasi tidak terlalu luas. Batasan masalah yang dijabarkan adalah sebagai berikut.

1. Penelitian yang dilakukan hanya berfokus pada Sistem Voting Elektronik.
2. *Platform blockchain* yang digunakan adalah *Ethereum.*

## Sistematika Penulisan

Sistematika penulisan merupakan bagian yang menjelaskan gambaran laporan dari semua dasar teori dan metode yang digunakan serta hasil yang diperoleh selama pengerjaan tugas akhir. Laporan tugas akhir ini dibagi menjadi lima bab sebagai berikut.

BAB I : **Pendahuluan**

Bagian ini membahas mengenai latar belakang, perumusan masalah, tujuan, manfaat, batasan masalah, serta sistematika penulisan laporan tugas akhir Sistem Voting Elektronik Berbasis *Ethereum Smart Contract*.

BAB II : **Tinjauan Pustaka**

Bagian ini menjelaskan tentang dasar teori, pengetahuan, dan referensi yang digunakan untuk memperkuat projek Tugas Akhir Sistem Voting Elektronik Berbasis *Ethereum Smart Contract*.

BAB III : **Metodologi dan Perancangan Sistem**

Bagian ini memuat metode yang digunakan dalam penelitian tugas akhir Sistem Voting Elektronik Berbasis *Ethereum Smart Contract* beserta dengan proses pembuatan laporan yang meliputi tempat dan waktu penelitian, sumber data dan metode pengumpulan, instrumen perancangan, algoritma pemrograman, hingga perancangan sistem dimulai dari rancangan prosedural hingga antarmuka (*interface*) sistem.

BAB IV : **Pengujian dan Analisa Hasil**

Bagian ini memuat rancangan yang telah dibuat, implentasi dari perangkat yang digunakan serta pengujian sistem dari Sistem Voting Elektronik Berbasis *Ethereum Smart Contract*.

BAB V : **Penutup**

Bagian ini memuat kesimpulan yang mengacu pada rumusan masalah dan tujuan penelitian beserta saran yang diberikan peneliti untuk kepentingan pengembangan aplikasi selanjutnya.

# BAB II TINJAUAN PUSTAKA

Bab II berisi bahasan mengenai teori atau materi pendukung yang digunakan sebagai acuan atau dasar teori dalam Sistem Voting Elektronik Berbasis *Ethereum Smart Contract*.

## *State of the Art*

Perancangan untuk pengembangan Sistem Voting Elektronik memiliki persamaan dan atau keterikatan dari beberapa penelitian yang dilakukan beberapa diantaranya adalah sebagai berikut.

Penelitian pertama yang digunakan sebagai acuan dalam penelitian ini adalah penelitian mengenai pengembangan sistem E-Voting berbasis *web* menggunakan keamanan algoritma Rivest Shamir Adleman (RSA). Aplikasi ini dibangun menggunakan bahasa pemrograman PHP dan *database* MySQLi sebagai *database server*. Fokus dari sistem voting elektronik ini adalah melakukan enkripsi hak suara pemilih pada sistem menggunakan *public key* dan *private key* untuk proses verifikasi sehingga tidak ada manipulasi data hak suara pemilih. Hasil yang diperoleh adalah sistem dapat melakukan validasi pemilih dan melakukan verifikasi hak suara apakah telah terjadi manipulasi data selama pengiriman atau tidak (Ridwan, et al., 2016). Penelitian selanjutnya adalah penelitian mengenai sistem *e-voting* berbasis SMS *gateway* yang dirancang menggunakan *software* Gammu dan menggunakan bahasa pemrograman PHP dan *database* MySQL. Uji coba pemungutan suara dilakukan dengan mengirim pemberitahuan melalui SMS kepada pemilih sesuai dengan data yang telah disimpan dan pemilihan dilakukan dengan cara mengirim SMS balasan sesuai dengan format yang telah ditentukan (Adhi & Harjono, 2014).

Perancangan sistem *voting* elektronik juga diterapkan dalam penelitian menggunakan *smart contract blockchain*. Sistem dirancang menggunakan bahasa pemrogaman *solidity* dan *smart contract* menghasilkan kode unik pada setiap pemilihan baru. Pengujian menghasilkan bahwa manipulasi hasil pemungutan suara tidak dapat dilakukan karena setiap pemilih hanya dapat memiliki satu akun dan satu *address blockchain* (Setia & Susanto, 2019). Perancangan sistem *e-voting* juga diterapkan menggunakan metode enkripsi *blockchain.* Enkripsi *blockchain* merupakan salah satu cara pengamanan data yang tidak memungkinkan untuk dilakukan penyusupan data ke dalamnya serta menjaga kerahasiaan pada data pemilih karena dilakukan proses enkripsi pada data pemilih dan apa yang dipilih, akan tetap dapat diperoleh informasi pemilik suara terbanyak (Ardilla, 2018). Penelitian lain terhadap sistem *e-voting* menggunakan teknologi *blockchain* adalah penelitian oleh Kurnia Hu, dkk menggunakan *multichain tools.* Penelitian ini menghasilkan sistem voting elektronik yang dapat menyimpan data yang transparan dan dapat diakses oleh publik, menjaga identitas pemilih, menyimpan data suara yang tidak dapat diubah, digandakan, atau dihapus. Berdasarkan hasil kuesioner diperoleh persentase sebesar 55,6 % pengguna menilai baik dan 44,4 % pengguna menilai baik sekali mengenai keseluruhan aplikasi (Kurnia Hu, et al., 2019). Aplikasi *e-voting* selanjutnya adalah SEVA: Secure E-Voting Application. Sistem dirancang bertujuan untuk memungkinan bagi para pemilih dapat melakukan pemilihan melalui *web* tanpa terbatas oleh lokasi. Aplikasi dikembangkan dengan Azure Service Fabric menggunakan arsitektur Microservice. Aplikasi ini diterapkan pada cluster Five-Node menggunakan Server Fabric Local Cluster (Johari, et al., 2020).

Penelitian selanjutnya adalah penerapan teknologi *blockchain* dalam layanan *E-commerce* pada OJS (*Open Journal System*)*.* Penggunaan teknologi *blockchain* dalam penelitian ini bertujuan untuk mengefisiensi manajemen identitas, membangun sistem pelacakan dan mengidentifikasi keaslian produk, dapat menyinkronkan data yang tersimpan dalam *blockchain* ke semua jaringan pengguna, dapat menjadikan sistem pembayaran yang lebih mudah, efisien dan terdokumentasikan dengan baik. Serta kemudahan bagi mahasiswa, dosen dan pihak lain dalam melakukan proses transaksi (Rahardja, et al., 2020)*.*

## Blockchain

Blockchain adalah sebuah pendekatan baru untuk konsep *database* terdistribusi. Blockchain juga bisa dikatakan sebagain sebuah struktur data yang memungkinkan untuk membuat *digital ledger* dari suatu data dan membagikannya di antara pihak independen. Ada beberapa jenis *blockchain* berdasarkan ijin keikutsertaan sebuah *node* di suatu jaringan *blockchain*. Yang pertama adalah Public Blockchain, *blockchain* ini mengijinkan siapapun untuk berpartisipasi dalam jaringan. Kedua adalah Permissioned Blockchain, *blockchain* ini terdapat *role* yang mengatur jalannya *network*. Dan yang ketiha adalah Private Blockchain, *blockchain* ini memiliki aturan tersendiri dalam menjalankan *network* dan umumnya berskala kecil. Ketiga jenis *blockchain* tersebut menggunakan kriptografi untuk memungkinkan setiap peserta di jaringan untuk mengelola buku besar dengan aman tanpa perlu *central authority* untuk menegakkan peraturan. Penghapusan *central authority* dari struktur *database* adalah sebuah aspek penting dari *blockchain*.

Blockchain membuat catatan permanen dan *history* dari semua transaksi yang terjadi. Permanen dalam konteks *blockchain* bergantung dari keberadaan partisipan dalam sebuah jaringan *blockchain* yang artinya data yang terekam pada *blockchain* akan benar-benar hilang jika semua partisipan meninggalkan jaringan *blockchain*. Ketika seorang partisipan ingin menambahkan suatu data ke *blockchain*, maka partisipan yang mempunyai *validation control* akan melakukan validasi transaksi tersebut. Jika transaksi sudah valid, maka data transaksi akan disebarkan ke seluruh partisipan di jaringan. Maka dari itu data telah terekam dalam *blockchain* akan sangat sulit untuk menghapus atau mengubah data tersebut. Setiap jaringan blockchain memiliki cara yang berbeda dalam menentukan bagaimana sebuah transaksi di validasi dan siapa yang berhak memiliki *validation control*.

## Ethereum

Ethereum adalah sebuah general purpose blockchain yang di inisiai oleh Vitalik Buterin. Blockchain ini di rilis pada tanggal 30 juli 2015. Tujuan Vitalik Buterin membuat Ethereum adalah untuk menyediakan blockchain yang tidak hanya bisa menyimpan digital asset seperti yang dilakukan oleh Bitcoin, namun juga bisa memberikan kemampuan bahasa pemrograman untuk berinteraksi dengan blockchain. Hal ini nantinya disebut dengan Smart Contract. Dengan adanya kemampuan ini, Ethereum dapat digunakan untuk mengembangkan aplikasi yang memiliki kelebihan blockchain didalamnya.

Ethereum memiliki banyak persamaan dengan Bitcoin. Namun ada perbedaan yang dimiliki oleh Ethereum. Perbedaan utama antara Ethereum dan Bitcoin adalah dari segi arsitektur. Tidak seperti Bitcoin yang hanya menyimpan salinan daftar transaksi, Ethereum menyimpan salinan transaksi, state terbaru di jaringan blockchain, block number dan tingkat kesulitan. Algoritma validasi secara umum di Ethereum adalah

1. Periksa apakah blok sebelumnya yang direferensikan ada dan valid.
2. Periksa apakah stempel waktu blok lebih besar dari pada blok sebelumnya yang direferensikan dan kurang dari 15 menit ke depan
3. Periksa apakah nomor blok, kesulitan, root transaksi, root paman, dan batas gas (berbagai konsep khusus Ethereum tingkat rendah) valid.
4. Periksa apakah bukti pekerjaan di blok itu valid.
5. Misalkan S [0] menjadi negara bagian pada akhir blok sebelumnya.
6. Misalkan TX menjadi daftar transaksi blok, dengan n transaksi. Untuk semua i dalam 0 ... n-1, setel S [i + 1] = APPLY (S [i], TX [i]). Jika ada aplikasi yang mengembalikan kesalahan, atau jika total gas yang dikonsumsi di blok sampai titik ini melebihi GASLIMIT, kembalikan kesalahan.
7. Misalkan S\_FINAL menjadi S [n], tetapi menambahkan hadiah blok yang dibayarkan ke penambang.
8. Periksa apakah akar pohon Merkle dari negara S\_FINAL sama dengan akar negara akhir yang disediakan di header blok. Jika ya, blok tersebut valid; jika tidak, itu tidak valid.

Pendekatan ini mungkin tampak sangat tidak efisien pada pandangan pertama, karena ia perlu menyimpan seluruh negara bagian dengan setiap blok, tetapi pada kenyataannya efisiensi harus sebanding dengan Bitcoin. Alasannya adalah karena status disimpan dalam struktur pohon, dan setelah setiap blok hanya sebagian kecil dari pohon yang perlu diubah. Jadi, secara umum, antara dua blok yang berdekatan, sebagian besar pohon harus sama, dan oleh karena itu data dapat disimpan sekali dan direferensikan dua kali menggunakan pointer (mis. Hash subpohon). Jenis pohon khusus yang dikenal sebagai "pohon Patricia" digunakan untuk mencapai hal ini, termasuk modifikasi pada konsep pohon Merkle yang memungkinkan simpul untuk disisipkan dan dihapus, dan tidak hanya diubah, secara efisien. Selain itu, karena semua informasi negara adalah bagian dari blok terakhir, tidak perlu menyimpan seluruh riwayat blockchain - sebuah strategi yang, jika dapat diterapkan pada Bitcoin, dapat dihitung untuk memberikan penghematan 5-20x di ruang penyimpanan.

## VSCode

Visual Studio Code adalah sebuah teks editor ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi *multiplatform*, artinya tersedia juga untuk versi Linux, Mac, dan Windows. Teks editor ini secara langsung mendukung bahasa pemrograman JavaScript, Typescript, dan Node.js, serta bahasa pemrograman lainnya dengan bantuan *plugin* yang dapat dipasang melalui *marketplace* Visual Studio Code (seperti C++, C#, Python, Go, Java, dst).

Fitur-fitur yang disediakan oleh Visual Studio Code, diantaranya *Intellisense, Git Integration, Debugging,* dan fitur ekstensi yang menambah kemampuan teks editor. *Editor* ini dirancang untuk pengembang yang bekerja dengan teknologi *open source* *cloud* serta menggunakan NET untuk memberikan dukungan untuk ASP. Antarmukanya tergolong mudah untuk digunakan karena didasarkan pada gaya *explorer* umum, dengan *panel* di sebelah kiri yang menunjukkan semua *file* dan *folder* serta *panel editor* di sebelah kanan, yang menunjukkan isi dari *file* yang telah dibuka.

|  |
| --- |
| https://code.visualstudio.com/opengraphimg/opengraph-home.png |

Gambar 2. Visual Studio Code

Microsoft telah menyediakan dokumentasi yang lengkap untuk membantu dan memudahkan penggunaan bagi para *developer*. Visual Studio Code ditargetkan pada pengembang scripting *server-side* dan dapat digunakan oleh siapa saja untuk membangun aplikasi berbasis *web* (Code, 2021)*.*

## NodeJS

Node.js merupakan salah satu *platform* pengembang yang dapat digunakan untuk membuat aplikasi berbasis *Cloud*. Node.js dikembangkan dari *engine* JavaScript yang dibuat oleh Google untuk *browser* Chrome ditambah dengan *libuv* serta beberapa pustaka lainnya. Node.js menggunakan JavaScript sebagai bahasa pemrograman dan *event-driven, non-blocking I/O (asynchronous)* model yang membuatnya ringan dan efisien. Node.js memiliki fitur built-in HTTP *server library* yang menjadikannya mampu menjadi sebuah *web server* tanpa bantuan *software* lainnya seperti Apache dan Nginx.

|  |
| --- |
| https://miro.medium.com/max/724/1*ODU5V_oAmYmzvZ1wIw3rDw.png |

Gambar 2. Node.js

Sumber : https://ichi.pro/id

Node.js adalah sebuah *runtime environment* dan *script library*. Sebuah *runtime environment* adalah sebuah *software* yang berfungsi untuk mengeksekusi, menjalankan dan mengimplementasikan fungsi-fungsi serta cara kerja inti dari suatu bahasa pemrograman. Sedangkan *script library* adalah kumpulan, kompilasi atau bank data berisi skrip/kode-kode pemrograman. Node.js dibangun menggunakan JavaScript dan C++, terdapat arsitektur serta fungsi dari Google V8 di dalamnya yang berfungsi sebagai *compiler* ditulis dalam C++ dan library Libuv bekerja untuk menangani operasi *asynchronous I/O* dan *main event loop* (Wilson, 2013).

### NestJS

Next.js adalah salah satu *framework* React yang sangat sederhana tetapi tidak terbatas. Next.js menyediakan struktur umum yang memungkinkan untuk membangun aplikasi React *frontend* dengan mudah, dan secara transparan menangani *rendering server-side*. Berikut ini adalah beberapa fitur utama Next.js (Copes, 2019) :

1. *Hot Code Reloading:* Next.js memuat ulang (*reload*) halaman ketika mendeteksi perubahan apa pun yang disimpan ke disk.
2. *Automatic Routing:* Setiap URL dipetakan ke sistem *file*, ke *file* yang dimasukkan ke *folder* dan tidak memerlukan konfigurasi apa pun.
3. *Single File Components:* Menggunakan style-jsx, yang sepenuhnya terintegrasi karena dibangun oleh tim yang sama, dan sangat mudah untuk menambahkan *style* yang tercakup ke komponen.
4. *Server Rendering*: Dapat membuat komponen React di sisi *server*, sebelum mengirim HTML ke klien
5. *Ecosystem Compatibility*: Next.js berfungsi baik dengan ekosistem JavaScript, Node, dan React lainnya.
6. *Automatic Code Splitting:* Aplikasi dipecah secara otomatis oleh Next.js di beberapa sumber daya yang berbeda. Hanya memuat JavaScript yang diperlukan untuk halaman tersebut. Next.js melakukannya dengan menganalisis sumber daya yang diimpor.
7. *Prefetching*: Komponen *Link*, digunakan untuk menghubungkan halaman yang berbeda, mendukung *prefetch prop* yang secara otomatis mengambil sumber halaman (termasuk kode yang hilang karena pemisahan kode) di latar belakang.
8. *Dynamic Components*: Dapat mengimpor modul JavaScript dan Komponen React secara dinamis.
9. *Static Exports*: Menggunakan perintah ekspor berikutnya, Next.js memungkinkan untuk mengekspor situs yang sepenuhnya statis dari aplikasi.
10. *TypeScript Support*: Next.js ditulis dalam *TypeScript* dan dengan demikian dilengkapi dengan dukungan *TypeScript* yang sangat baik.

### Web3JS

Web3js memungkinkan untuk berinteraksi dengan *node* Ethereum lokal atau jarak jauh, menggunakan koneksi HTTP atau IPC. Web3js merupakan pustaka yang memungkinkan untuk melakukan tindakan pengiriman seperti mengirim *ether* dari akun satu ke akun lain, membaca dan menulis *smart contract* dan lain sebagainya.

|  |
| --- |
|  |

Gambar 2. Interaksi Web3.JS dengan Node Ethereum

Web3.js dapat digunakan untuk terhubung ke jaringan Ethereum melalui *node* Ethereum yang memungkinkan akses melalui HTTP. Akses tersebut dapat berupa *node* lokal, *node* yang di-*hosting* oleh penyedia DApp, atau *gateway* publik seperti Infura, yang mengoperasikan titik akses Ethereum gratis. Salah satu cara umum untuk mengintegrasikan aplikasi *browser web* dengan Ethereum adalah dengan menggunakan ekstensi *browser* Metamask dengan kombinasi Web3js (Web3JS, 2016).

## MySQL

MySQL merupakan *software* yang tergolong sebagai DBMS (*Database Management System*) yang bersifat *open source*. *Open source* menyatakan bahwa *software* ini dilengkapi dengan *source code* selain bentuk *executable*-nya atau kode yang dapat dijalankan secara langsung dalam sistem operasi. MySQL memiliki fungsi-fungsi MySQL yang digunakan untuk mengakses *database server* MySQL. Fungsi ini berguna untuk mengantarkan perintah SQL pada PHP menuju ke *server* sehingga perintah tersebut dapat dieksekusi oleh semua *server* MySQL.

Tabel 2. MySQL Function

| **No.** | **Function** | **Kegunaan** |
| --- | --- | --- |
|  | *Mysql\_connect()* | Membuat hubungan ke *database* MySQL yang terdapat pada suatu *host* |
|  | *Mysql\_close()* | Menutup hubungan ke *database MySQL* |
|  | *Mysql\_select\_db()* | Memilih *database* |
|  | *Mysql\_query()* | Mengeksekusi permintaan terhadap sebuah tabel atau sejumlah tabel |
|  | *Mysql\_db\_query()* | Menjalankan suatu permintaan terhadap suatu *database* |
|  | *Mysql\_num\_rows()* | Memperoleh jumlah baris dari suatu hasil permintaan yang menggunakan *SELECT* |
|  | *Mysql\_affected\_rows()* | Memperoleh jumlah baris yang dikenai operasi *INSERT, DELETE, UPDATE* |
|  | *Mysql\_num\_fields()* | Memperoleh jumlah kolom pada suatu hasil permintaan |
|  | *Mysql\_fecth\_row()* | Menghasilkan *array*/baris yang berisi seluruh kolom dari sebuah baris pada suatu himpunan hasil |
|  | *Mysql\_fecth\_array()* | Menghasilkan *array*/baris yang berisi seluruh kolom dari sebuah baris pada suatu himpunan hasil yang akan disimpan dua kali pada *array* hasil |
|  | *Mysql\_fecth\_field()* | Menghasilkan informasi suatu kolom |
|  | *Mysql\_data\_seek()* | Memindahkan pointer pada suatu himpunan hasil supaya menunjuk ke baris tertentu |
|  | *Mysql\_field\_seek()* | Memindahkan pointer pada suatu himpunan hasil supaya menunjuk ke kolom tertentu |
|  | *Mysql\_create\_db()* | Membuat *database MySQL* |
|  | *Mysql\_drop\_db()* | Menghapus *database MySQL* |
|  | *Mysql\_list\_dbs()* | Menghasilkan daftar *database MySQL* |
|  | *Mysql\_list\_tables()* | Memperoleh daftar nama tabel dalam suatu *database* |
|  | *Mysql\_list\_fields()* | Memperoleh daftar nama kolom dalam suatu *database* |
|  | *Mysql\_fetch\_assoc()* | Mendapatkan array baris dari suatu *recordset* |
|  | *Mysql\_fetch\_lengths()* | Mendapatkan panjang baris pada setiap isi *field* |
|  | *Mysql\_fetch\_object()* | Menghasilkan baris dari *recordset* sebagai sebuah objek |
|  | *Mysql\_field\_len()* | Mendapatkan informasi panjang maksimum *field* dalam sebuah *recorset* |
|  | *Mysql\_field\_name()* | Mendapatkan informasi nama *field* dalam *recordset* |
|  | *Mysql\_ping()* | Memeriksa koneksi *server* dan akan mencoba untuk melakukan koneksi ulang jika koneksi terputus |

Tabel 2.1 merupakan fungsi-fungsi dalam MySQL yang dapat digunakan untuk mengantarkan perintah SQL pada PHP menuju ke *server* sehingga perintah tersebut dapat dieksekusi oleh semua *server* MySQL.

Selain fungsi-fungsinya, MySQL juga memiliki beberapa keunggulan, seperti dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lainnya. MySQL didistribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi GPL sehingga dapat digunakan secara gratis serta dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik. MySQL memiliki kecepatan yang tinggi dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu. MySQL memiliki ragam tipe data yang sangat kaya, seperti *signed / unsigned integer, float, double, char, text, date, timestamp*, dan lain-lain. MySQL memiliki antar muka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*) (Kadir, 2008).

## JSON

JSON (JavaScript Object Notation) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari bahasa pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh *programmer* keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dan lainnya (Derizal, 2011).

## Google Cloud Platform

Google Cloud Platform merupakan layanan *public cloud computing* dari Google yang terdiri dari beragam layanan. *Platform* dari Google ini menyediakan beragam layanan *hosting* mulai dari komputasi, *storage* dan *aplication development* yang berjalan pada *hardware* Google. Google Cloud Platform Service dapat diakses oleh pengembang *software, administrator cloud* dan profesional IT lainnya menggunakan internet publik atau melalui koneksi jaringan *dedicated* (Google, 2021).

### Compute Engine

Google Compute Engine memungkinkan untuk menjalankan beban kerja komputasi skala besar dengan infrastruktur yang sama yang menjalankan Google Search, Gmail dan Ads. Pengguna dapat meluncurkan *virtual machines* sesuai permintaan, mengelola konektivitas jaringan menggunakan solusi jaringan yang sederhana namun fleksibel, dan mengakses berbagai alternatif penyimpanan data dari *virtual machines*. Fitur-fitur Google Compute Engine antara lain (Google, 2021) :

1. *Full Virtual Machines:* Mesin virtual yang di-*hosting* kernel yang menjalankan Ubuntu atau CentOs. Meluncurkan 1, 2, 4, atau 8 *virtual core instances* dengan memori 3.75 GB per *virtual core*.
2. *Flexible Storage:* menawarkan opsi penyimpanan yang berbeda untuk memenuhi berbagai kebutuhan.
3. *Flexible networking:* menawarkan solusi jaringan yang memudahkan untuk menghubungkan *virtual machines* satu sama lain dan ke Internet.
4. *Open tooling:* menawarkan UI sederhana dan *command-line tooling* untuk mengonfigurasi dan meluncurkan *virtual machines*.

# BAB III METODOLOGI DAN PERANCANGAN SISTEM

BAB III ini memuat metode yang digunakan dalam penelitian beserta dengan proses pembuatan laporan yang meliputi tempat dan waktu penelitian, sumber data dan metode pengumpulan, instrument perancangan, dan gambaran umum sistem.

## Tempat dan Waktu Penelitian

Penelitian ini bertempat di Kampus Bukit Universitas Udayana, Fakultas Teknik, Program Studi Teknologi Informasi yang berlokasi di Jalan Kampus Bukit UNUD Jimbaran*,* Kabupaten Badung, Bali. Kampus Bukit Universitas Udayana dipilih karena memiliki semua aspek pendukung agar penelitian dapat berjalan dengan baik. Penelitian dilaksanakan pada semester genap tahun ajaran 2019-2020 yaitu antara bulan Januari sampai dengan bulan Juni dan semester ganjil tahun ajaran 2020-2021 yaitu antara bulan Juli sampai dengan bulan Desember.

## Data

Penelitian ini memerlukan data untuk mendukung keberhasilan dari sistem yang dihasilkan. Data tersebut dibagi menjadi beberapa jenis data yang diperoleh dari berbagai sumber dengan menggunakan beberapa metode pengumpulan data.

### Sumber Data

Sumber data pada penelitian ini bersumber dari studi literatur, survei dan pengumpulan data langsung di lapangan yang dalam penelitian ini adalah x. Studi literatur yang digunakan diperoleh dari penelitian terdahulu, jurnal ilmiah, buku, *e-book*, forum, dan paper publikasi. Pengumpulan data di lapangann dilakukan dengan cara x.

### Jenis Data

Data yang digunakan pada penelitian ini terdiri dari dua jenis data yaitu data primer dan sekunder. Data primer merupakan jenis data yang didapatkan dari hasil pengumpulan data di lapangan dan survei melalui kuesioner. Data sekunder merupakan jenis data yang didapatkan dari penelitian terdahulu, jurnal ilmiah, buku, e-book, forum, dan paper publikasi.

### Metode Pengumpulan Data

Metode pengumpulan data dalam penelitian Sistem Voting Elektronik Berbasis *Ethereum Smart Contract* yang digunakan, antara lain sebagai berikut:

1. Metode Kepustakaan

Metode kepustakaan merupakan suatu metode pengumpulan data yang bersumber pada penelitian terdahulu, jurnal ilmiah, buku, *e-book*, forum, dan paper publikasi.

1. Metode Observasi Langsung

Metode observasi langsung yaitu dengan pengumpulan data yang dilakukan dengan pengumpulan secara langsung terkait pencarian data x dari x.

## Metode Penelitian

Metode penelitian yang digunakan pada pembuatan Sistem Voting Elektronik Berbasis *Ethereum Smart Contract* menggunakan metode *Waterfall.* Metode *Waterfall* pertama kali diperkenalkan pada tahun 1970 oleh Royce dengan 7 (tujuh) tahapan yang berurut (Petersen, et al., 2009). Kemudian pada tahun 2011 metode ini mengalami banyak perbaikan dan perubahan diantaranya pada langkah 7 (tujuh) dan menjadi 5 (lima) tahapan (Sommerville, 2011). Metode *Waterfall* mempunyai tahapan-tahapan seperti pada Gambar 3.1.

|  |
| --- |
| metode waterfall |

Gambar 3. 1 Metode Waterfall

Gambar 3.1 merupakan gambaran tahapan dari metode *waterfall* yang akan digunakan dalam pengembangan Sistem Voting Elektronik Berbasis *Ethereum Smart Contract*.

### Requirements Analysis and Definition

Tahap ini melakukan pengumpulan dan analisa data. Studi literatur dilakukan terhadap jurnal penelitian internasional, tesis dan disertasi nasional dan internasional, serta buku-buku teori pendukung nasional dan internasional. Peneliti melakukan analisis, interpretasi dan generalisasi fakta-fakta dari literatur yang didapatkan. Studi juga dilakukan terhadap kondisi saat ini melalui pengumpulan data yang tersedia. Pengumpulan data dilakukan secara langsung pada x melalui x.

### System and Software Design

Tahapan selanjutnya yaitu mengalokasikan kebutuhan-kebutuhan sistem seperti perangkat keras maupun perangkat lunak serta melakukan identifikasi masalah untuk menentukan desain sistem. Identifikasi dilakukan terhadap masalah yang ada terutama berdasarkan teknologi dan kondisi yang ada. Proses ini dilakukan kajian untuk memahami dan menentukan masalah yang akan diselesaikan berdasarkan hasil dari studi literature. Perancangan perangkat lunak melibatkan identifikasi dan penggambaran abstraksi sistem dasar perangkat lunak dan hubungannya.

### Implementation and Unit Testing

Tahap ini dilakukan implementasi dengan *software* x untuk pembuatan x*.* Bahasa pemrograman yang digunakan yaitu x.

### Integration and System Testing

Langkah terakhir pada proses pengembangan Sistem Voting Elektronik Berbasis *Ethereum Smart Contract* ini adalah pengujian aplikasi. *Testing* Aplikasi langsung dilakukan pada *software* x.

### Operation and Maintenance

Tahapan ini merupakan tahapan yang paling panjang. Sistem di-*build* dan sudah diinstal pada *x* serta dilakukan pengujian secara langsung di lapangan atau dalam penelitian ini adalah x.

Setelah rancangan dan demonstrasi/simulasi didapatkan, pengujian terhadap aplikasi dilakukan dalam 2 (dua) tahapan. Tahapan pertama adalah dengan melakukan *debugging* terhadap aplikasi menggunakan metode *blackbox*. Tahapan kedua adalah dievaluasi terhadap kelompok pengguna (beta tester) sebagai bagian dari evaluasi dengan pengguna sekaligus menilai pencapaian tujuan dari aplikasi

*Maintenance* melibatkan perbaikan kesalahan yang tidak ditemukan pada tahapan-tahapan sebelumnya, meningkatkan implementasi dari unit sistem, dan meningkatkan informasi sebagai kebutuhan baru.

## Gambaran Umum Sistem

Sistem Voting Elektronik berbasis *Ethereum Smart Contract* merupakan sistem yang digunakan sebagai sarana pemilihan elektronik dengan menerapkan teknologi *blockchain*. Gambar 3.2 merupakan gambaran umum arsitektur sistem *e-voting* berbasis *ethereum smart contract*. Gambaran umum arsitektur teknologi merupakan bagan yang menjelaskan alur kerja teknologi yang digunakan secara menyeluruh secara umum.

|  |
| --- |
| *C:\Users\sarah\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Gambaran Umum.png* |

Gambar 3. 2 Gambaran Umum Sistem

Gambar 3.2 merupakan gambaran umum dari Sistem Voting Elektronik berbasis *Ethereum Smart Contract*. Pertama *super admin* akan membuat akun untuk *election authority*. *Election authority* akan membuat membuat eleksi dan menambahkan kandidat pada eleksi. Eleksi akan di *review* oleh *super admin* apakah diijinkan untuk di-*deploy* ke *blockchain* karena setiap transaksi di *blockchain* memerlukan sumber daya *Ether*. Dalam sistem ini hanya *super admin* yang memiliki sumber daya *Ether* tersebut. *Smart Contract* yang telah di-*deploy* oleh *super admin* sudah siap digunakan untuk merekam suara *voter* yang terdaftar dalam eleksi.

### Use Case Diagram

Use case diagram merupakan representasi interaksi *user* dengan sistem dan menggambarkan spesifikasi dari *use case*. *Use case* untuk sistem *e-voting* berbasis *Ethereum Smart Contract* ditunjukkan oleh Gambar 3.3.

|  |
| --- |
| C:\Users\sarah\AppData\Local\Microsoft\Windows\INetCache\Content.Word\usecase.png |

Gambar 3. 3 *Use Case Diagram* Aplikasi AR Taman Ayun

Gambar 3.3 menunjukkan *use case* dari sistem *e-voting* berbasis *Ethereum Smart Contract*. *Use case* tersebut menggambarkan interaksi pengguna terhadap sistem *e-voting* dan interaksi antara satu elemen dengan elemen lainnya. *Use case* aplikasi sistem *e-voting* memiliki 3 aktor, diantaranya *super admin, election authority* dan *voter*.

Masing-masing aktor memiliki peran dan akses yang berbeda-beda. *Super admin* dapat melakukan penambahan atau mendaftarkan *election authority* ke dalam sistem, melakukan proses *send ether* danproses *deploy* atau aktivasi *election* sehingga dapat digunakan, serta melihat hasil *election*. *Election authority* dapat melakukan pembuatan *election* baru, menambahkan kandidat dalam *election* yang dibuat, menentukan awal dan berakhirnya *election,* serta melihat hasil *election. Voter* dapat melakukan pendaftaran dalam setiap *election* yang diikuti dan melakukan proses *vote* dalam *election* yang dipilih, serta melihat hasil *election.*

### Rancangan Alur Sistem

Rancangan alur penggunaan aplikasi sistem *e-voting* berbasis *Ethereum Smart Contract* digunakan sebagai acuan untuk membuat antarmuka dari aplikasi.

1. Rancangan Alur Sistem *(Super Admin)*

Aplikasi sistem *e-voting* memiliki 3 jenis *user* yang memiliki peran dan akses yang berbeda-beda. Jenis *user* yang pertama adalah *super admin*. Rancangan alur sistem dengan *super admin* sebagai *user* dapat dilihat pada Gambar 3.4.

|  |
| --- |
|  |

Gambar 3. Rancangan Alur Sistem *(Super Admin)*

Gambar 3.4 menggambarkan rancangan alur keseluruhan sistem dengan *super admin* sebagai *user*. Pertama-tama, *super admin* harus *login* terlebih dahulu ke dalam sistem. Setelah berhasil *login, super admin* dapat melakukan pembuatan akun bagi *election authority.* Apabila terdapat *election* baru yang siap di­-*deploy,* maka *super admin* bertugas melakukan proses *deploy* ke *blockchain.*

1. Rancangan Alur Sistem *(Election Authority)*

Jenis *user* selanjutnya adalah *election authority,* yaitu *user* yang bertanggung jawab untuk membuat dan menangani satu atau lebih *election.* Rancangan alur sistem dengan *election authority* sebagai *user* dapat dilihat pada Gambar 3.5.

|  |
| --- |
| C:\Users\sarah\Downloads\Telegram Desktop\flowchart-election-authority.png |

Gambar 3. Rancangan Alur Sistem *(Election Authority)*

Gambar 3.5 menggambarkan rancangan alur keseluruhan sistem dengan *election authority* sebagai *user.* Alur sistem diawali dengan proses *login* oleh *election authority* menggunakan akun yang telah didaftarkan atau dibuat oleh *super admin* sebelumnya.Setelah berhasil *login,* maka *election authority* dapat melakukan proses pembuatan *election* baru, diikuti dengan peambahan daftar kandidatnya. Setelah itu, apabila election yang dibuat telah di-*deploy* oleh *super admin,* maka *election authority* dapat memulai *election.* Namun, apabila *election* belum di-*deploy,* maka *election authority* harus menunggu sampai *election* berhasil di­-*deploy* oleh *super admin* untuk dapat memulai *election.*

1. Rancangan Alur Sistem *Voter*

Jenis *user* yang terakhir adalah *voter,* yaitu *user* yang memiliki hak suara dan berhak memilih dalam satu atau lebih *election.* Rancangan alur sistem dengan *voter* sebagai *user* dapat dilihat pada Gambar 3.6.

|  |
| --- |
|  |

Gambar 3. Rancangan Alur Sistem *(Voter )*

Gambar 3.6 menggambarkan rancangan alur keseluruhan sistem dengan *voter* sebagai *user.* Langkah pertama yang harus dilakukan oleh *voter* adalah *login* ke dalam sistem. Setelah berhasil *login,* maka *voter* dapat mendaftar ke dalam *election* yang akan diikuti. Pendaftaran *voter* dalam *election* harus terlebih dahulu disetujui untuk selanjutnya dapat melihat daftar kandidat dalam *election,* namun apabila pendaftaran ditolak, maka *voter* harus melakukan pendaftaran ulang sampai pendaftaran diterima untuk dapat mengikuti *election* tersebut. Proses pemungutan suara atau *vote* hanya dapat dilakukan oleh *voter* apabila *election* telah dimulai.

1. Rancangan Alur Proses *Login*

Proses *login* dapat dilakukan oleh 3 aktor yang terlibat dalam mengakses sistem. Rancangan alur untuk proses *login* pada sistem *e-voting* dapat dilihat pada Gambar 3.7.

|  |
| --- |
| C:\Users\sarah\AppData\Local\Microsoft\Windows\INetCache\Content.Word\flowchart-login .png |

Gambar 3. Rancangan Alur Proses *Login*

Gambar 3.7 memperlihatkan aktivitas untuk melakukan proses *login* untuk masuk atau mengakses sistem. Proses *login* juga menentukan hak akses dari masing-masing aktor yang terlibat. Setelah melakukan pengecekan data *login* yang dimasukkan maka dilakukan pengecekan terhadap *role* dari *user* yang akan *login.* Apabila dalam proses pengecekan tidak ada kesalahan, maka *user* akan dibawa menuju halaman sesuai dengan *user role.* Apabila terdapat kesalahan, maka *user* akan diminta memasukkan data *login* kembali.

1. Rancangan Alur Penambahan *Election Authority*

Rancangan alur untuk proses penambahan *election authority* pada sistem *e-voting* dapat dilihat pada Gambar 3.8.

|  |
| --- |
| C:\Users\sarah\AppData\Local\Microsoft\Windows\INetCache\Content.Word\super-admin_create-ea.png |

Gambar 3. Rancangan Alur Penambahan *Election Authority*

Gambar 3.8 memperlihatkan proses penambahan atau pembuatan *election authority* baru oleh *super admin.* Data yang dimasukkan untuk pembuatan *election authority* adalah nama, *usernme* dan *password. Username* harus bersifat *unique* atau belum pernah digunakan sebelumnya.

1. Rancangan Alur Pembuatan *Election*

Rancangan alur untuk proses pembuatan *election* baru pada sistem *e-voting* dapat dilihat pada Gambar 3.9.

|  |
| --- |
| C:\Users\sarah\AppData\Local\Microsoft\Windows\INetCache\Content.Word\election-authority_create-election.png |

Gambar 3. 9 Rancangan AlurPembuatan *Election* Baru

Gambar 3.9 memperlihatkan aktivitas untuk melakukan penambahan atau pembuatan *election* baru. *Election authority* yang bertugas melakukan pembuatan *election* baru meng-*input*-kan nama*,* deskripsi, serta tanggal berlangsungnya pemilihan. Setelah proses pembuatan *election,* maka *election* baru berhasil dibuat.

1. Rancangan Alur Tambah Kandidat

Rancangan alur untuk proses penambahan kandidat yang dapat dipilih dalam sebuah *election* pada sistem *e-voting* dapat dilihat pada Gambar 3.10.

|  |
| --- |
| C:\Users\sarah\AppData\Local\Microsoft\Windows\INetCache\Content.Word\election-authority_add-candidate.png |

Gambar 3. 10 Rancangan AlurPenambahan Kandidat

Gambar 3.10 memperlihatkan aktivitas untuk melakukan proses penambahan kandidat yang dapat dipilih dalam sebuah *election*. Sebelum melakukan penambahan kandidat, *election authority* harus terlebih dahulu membuat *election. Election* yang telah berhasil dibuat kemudian dapat dipilih untuk melakukan penambahan kandidat. *Election authority* memasukkan data nama, visi, misi dan pengalaman dari kandidat. Proses penambahan kandidat dapat dilakukan secara berulang sebanyak jumlah kandidat dalam *election* yang dipilih.

1. Rancangan Alur *Deploy Election*

Rancangan alur untuk proses aktivasi *election* yang telah dibuat sehingga dapat digunakan atau *deploy election* dapat dilihat pada Gambar 3.11.

|  |
| --- |
| C:\Users\sarah\AppData\Local\Microsoft\Windows\INetCache\Content.Word\super-admin_deploy-election.png |

Gambar 3. Rancangan Alur *Deploy Election*

Gambar 3.11 memperlihatkan aktivitas untuk melakukan proses mengaktifkan *election* yang telah dibuat oleh *election authority* sebelumnya atau proses membawa *smart contract* ke *blockchain* sehingga dapat digunakan. Setelah *election* yang akan di-*deploy* dipilih, maka akan dilakukan proses pengiriman *ether* dari *super admin* ke *wallet address election authority.* Selanjutnya dilakukan proses *deploy smart contract* ke *private blockchain* dan penentuan kandidat di *smart contract* bedasarkan data yang tersimpan di basis data. Setelah semua proses selesai dilakukan maka *smart contract* telah berhasil di-*deploy* dan sudah dapat digunakan.

1. Rancangan Alur Registrasi *Voter*

Sebelum dapat melakukan proses *voting, s*etiap *voter* harus terdaftar pada sistem terlebih dahulu. Rancangan alur untuk proses pendaftaran *voter* dapat dilihat pada Gambar 3.12.

|  |
| --- |
| C:\Users\sarah\AppData\Local\Microsoft\Windows\INetCache\Content.Word\umum-register-voter.png |

Gambar 3. Rancangan Alur Registrasi *Voter*

Gambar 3.12 memperlihatkan aktivitas untuk melakukan proses pendaftaran masing-masing *voter* dalam sistem *e-voting*. *Voter* memasukkan *username* yang bersifat unik dan *password* untuk melakukan proses pembuatan akun yang terdaftar dalam sistem.

1. Rancangan Alur Pendaftaran *Voter* dalam *Election*

Setiap *voter* yang telah terdaftar di sistem akan melakukan *voting* dan juga harus terdaftar pada masing-masing *election* yang akan diikuti. Rancangan alur untuk proses pendaftaran *voter* dapat dilihat pada Gambar 3.13.

|  |
| --- |
| C:\Users\sarah\AppData\Local\Microsoft\Windows\INetCache\Content.Word\voter_voter-register-to-election.png |

Gambar 3. Rancangan Alur Pendaftaran *Voter* dalam *Election*

Gambar 3.13 memperlihatkan aktivitas untuk melakukan proses pendaftaran masing-masing *voter* dalam setiap *election* yang diikuti. Setelah melakukan pemilihan *election* yang diikuti, maka *election authority* akan memverifikasi apakah *voter* tersebut memiliki hak suara atau tidak dalam *election* yang dipilih.

1. Rancangan Alur Proses *Vote*

Rancangan alur untuk pemberian suara atau *vote* oleh setiap *voter* dalam sebuah *election* pada sistem *e-voting* dapat dilihat pada Gambar 3.14.

|  |
| --- |
| C:\Users\sarah\AppData\Local\Microsoft\Windows\INetCache\Content.Word\voter_voter-vote.png |

Gambar 3. Rancangan Alur Proses *Vote*

Gambar 3.14 memperlihatkan aktivitas untuk melakukan proses *vote* oleh masing-masing *voter. Voter* terlebih dahulu melakukan pemilihan *election* dan selanjutnya memilih kandidat yang akan di *vote.* Setelah itu dilakukan proses pengiriman *ether* dari *super admin* ke *wallet address voter* dilanjutkan dengan proses pemanggilan fungsi *vote* pada *smart contract* oleh *voter.* Apabila semua proses telah selesai dilakukan maka suara dari *voter* telah berhasil terekam di *blockchain.*

## Rancangan Basis Data

Perancangan basis data merupakan suatu langkah dalam merancang basis data yang akan digunakan pada sistem. Perancangan basis data digunakan sebagai pendekatan struktur dengan menggunakan prosedur, teknik, alat serta bantuan dokumen untuk membantu dan memudahkan dalam proses perancangan.

### Implementasi Basis Data

Implementasi basis data diambil berdasarkan perancangan basis data yang telah dibuat, secara fisik, implementasi basis data diimplementasikan menggunakan perangkat lunak MySQL. Struktur tabel basis data dapat dilihat pada Gambar 3.12.

|  |
| --- |
|  |

Gambar 3. 15 Implementasi Basis Data

Gambar 3.12 merupakan implementasi basis data dari *e-voting* berbasis *Ethereum Smart Contract*. Basis data tersebut digunakan sebagai media penyimpanan informasi data *user*, *election* dan *candidate*.

### Struktur Data Tabel

Basis data merupakan kumpulan data yang disatukan di dalam kumpulan tabel. Struktur data dari masing-masing tabel dalam perancangan basis data dijelaskan sebagai berikut.

1. Tabel X

aaaaaaa.

Tabel 3. 1 Struktur Data Tabel X

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Tabel 3.1 menjabarkan struktur data dari tabel x. Contoh data atau isi dari setiap *fields* dalam tabel x dapat dilihat pada Tabel 3.2.

Tabel 3. 2 Contoh Data Tabel x

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Tabel 3.2 merupakan contoh data pada tabel x. Tabel x memilikix.

## Bahasa Pemrograman

Implementasi Sistem Voting Elektronik Berbasis *Ethereum Smart Contract* merupakan aplikasi berbasis x. X dibangun dengan bahasa pemrograman X. Adapun *database* yang digunakan adalah database MySQL..

## Instrumen Perancangan Aplikasi

Instrumen Perancangan Aplikasi adalah media pendukung untuk melakukan perancangan aplikasi dalam penelitian ini. Instrumen perancangan dibagi menjadi dua bagian yaitu perangkat lunak dan perangkat keras. Kebutuhan minimum perangkat lunak dan perangkat keras untuk melakukan perancangan aplikasi pada penelitian ini dijelaskan pada subbab dibawah ini.

### Kebutuhan Perangkat Lunak Perancangan Aplikasi

Kebutuhan perangkat lunak merupakan kebutuhan minimal perangkat lunak yang harus dipenuhi agar suatu aplikasi dapat dikembangkan. Kebutuhan minimal perangkat lunak untuk merancang Sistem Voting Elektronik Berbasis *Ethereum Smart Contract* disebutkan pada Tabel 3.7

Tabel 3. 3 Kebutuhan Perangkat Lunak

|  |  |
| --- | --- |
| PC | Smartphone |
|  |  |
|  |
|  |
|  |
|  |
|  |
|  |

Tabel 3.7 merupakan kebutuhan *software* dalam pembuatan Sistem Voting Elektronik Berbasis *Ethereum Smart Contract*. Kebutuhan *software* yang harus dipenuhi baik dari sisi *software* pada komputer maupun pada *smartphone*. *Software* komputer seperti x, sangat diperlukan dalam pembuatan aplikasi Sistem Voting Elektronik Berbasis *Ethereum Smart Contract* sedangkan pada Smartphone minimum sistem operasi x.

### Kebutuhan Perangkat Keras Perancangan Aplikasi

Kebutuhan perangkat keras merupakan kebutuhan perangkat keras yang diperlukan agar suatu aplikasi dapat dikembangkan. Berikut ini merupakan kebutuhan perangkat keras untuk merancang Sistem Voting Elektronik Berbasis *Ethereum Smart Contract*.

1. *a*
2. *a*
3. *a*

## Rancangan Antarmuka Aplikasi

Rancangan antarmuka aplikasi merupakan rancangan terhadap desain *user interface* pembentuk dan pendukung Sistem Voting Elektronik Berbasis *Ethereum Smart Contract*.

|  |
| --- |
|  |

Gambar 3. 16 Rancangan Antarmuka Halaman *x*

Gambar 3.19 menunjukkan tampilan *x.*

# BAB IV PENGUJIAN DAN ANALISA HASIL

BAB IV menjelaskan hasil dari penelitian ini yaitu berupa hasil pengujian aplikasi, dan analisa dari aplikasi yang telah dikembangkan serta survei terhadap perangkat lunak yang telah dibuat dan implementasinya ke dalam perangkat keras yaitu *smartphone* berbasis Android.

## X

aaaaaaaa.

### x

aaaaa.

### X

Aaaa

# BAB V PENUTUP

BAB V membahas mengenai kesimpulan yang mengacu pada rumusan masalah dan tujuan penelitian beserta saran yang diberikan peneliti untuk kepentingan pengembangan aplikasi selanjutnya.

## Kesimpulan

## Saran

# DAFTAR PUSTAKA