KATA PENGANTAR

Puji syukur kehadirat Allah Subhanahu Wata’ala. Dzat Maha penolong yang hanya kepada-Nya memohon pertolongan, juga atas rahmat dan karunia-Nya sehingga penelitian tugas akhir dengan judul “*Sistem Pemilihan Elektronik (e-voting) Berbasis Blockchain*” ini dapat diselesaikan. Shalawat serta salam kepada Rasulullah Shallallahu ‘Alaihi Wasallam yang senantiasa menjadi teladan dan inspirasi terbaik bagi umatnya.

Adapun latar belakang penelitian ini adalah kebutuhan akan sistem pemilu yang lebih transparan, efisien, dan aman, khususnya di tingkat lokal. Dengan memanfaatkan teknologi blockchain, penelitian ini bertujuan untuk memberikan solusi inovatif yang dapat meningkatkan kepercayaan masyarakat terhadap proses pemilihan.

Pada penelitian ini banyak pihak yang memberi dukungan dan bantuan selama menyelesaikan studi dan tugas akhir ini. Oleh karena itu, sudah sepantasnya penulis memberikan hormat dan ucapan terimakasih semoga Allah memberikan balasan terbaik kepada:

* Prof. Dr. H. Rosihon Anwar, M.Ag., selaku Rektor UIN Sunan Gunung Djati.
* Prof. Dr. Hasniah Aliah, M. Si., selaku Dekan Fakultas Sains dan Teknologi.
* Dr. Dian Sa’adilah Maylawati M. T., selaku Ketua Program Studi Teknik Informatika.
* Bapak Dr. Wisnu Uriawan, ST., M Kom., dan bapak Ichsan Taufik M T., selaku pembimbing yang telah meluangkan waktu ditengah kesibukanya memberikan kritik, saran, dan pengarahan kepada penulis.
* Orang tua penulis yang selalu memberi dukungan baik secara moril maupun materil.
* Seluruh staff dosen jurusan Teknik Informatika UIN Sunan Gunung Djati yang sudah memberikan ilmu yang tak ternilai harganya, masukan, pemikiran dan tenaga selama proses pembelajaran yang dapat menambah wawasan bagi penulis
* Teman-teman di jurusan Teknik Informatika UIN Sunan Gunung Djati angkatan 2020, khususnya untuk kelas B.
* Kepala Desa Gunungleutik beserta jajaranya.

Terimakasih juga penulis haturkan untuk semua pihak yang telah membantu dalam menyelesaikan penelitian tugas akhir ini yang tidak dapat sebutkan satu persatu. Penulis berharap semoga penelitian tugas akhir ini dapat bermanfaat bagi pembaca dan dapat dijadikan referensi demi pengembangan ke arah yang lebih baik. Semoga Allah SWT senantiasa melimpahkan Rahmat dan Ridho-Nya kepada kita semua.

|  |  |  |
| --- | --- | --- |
|  |  | Bandung,  Penulis  **Firhan Abdurrahman Gani** |

DAFTAR ISI

[KATA PENGANTAR i](#_Toc184231982)

[DAFTAR ISI iii](#_Toc184231983)

[DAFTAR GAMBAR v](#_Toc184231984)

[DAFTAR TABEL vi](#_Toc184231985)

[DAFTAR LAMPIRAN vii](#_Toc184231986)

[BAB I PENDAHULUAN 8](#_Toc184231987)

[1.1. Latar Belakang 8](#_Toc184231988)

[1.2. Rumusan Masalah 9](#_Toc184231989)

[1.3. Tujuan Penelitian 10](#_Toc184231990)

[1.4. Manfaat Penelitian 10](#_Toc184231991)

[1.5. Batasan Masalah 11](#_Toc184231992)

[1.6. Kerangka Pemikiran Penelitian 12](#_Toc184231993)

[1.7. Sistematika Penulisan 12](#_Toc184231994)

[BAB II KAJIAN LITERATUR 14](#_Toc184231995)

[2.1. *State of the Art* 14](#_Toc184231996)

[2.2. Dasar Teori 25](#_Toc184231997)

[2.2.1. Blockchain 25](#_Toc184231998)

[2.2.2. Pemilihan Elektronik 28](#_Toc184231999)

[2.2.3. Ethereum 29](#_Toc184232000)

[2.2.4. *Smart Contract* 31](#_Toc184232001)

[2.2.5. *Solidity* 32](#_Toc184232002)

[2.2.6. *UML (Unified Modeling Language)* 33](#_Toc184232003)

[2.2.6. *Agile* 33](#_Toc184232004)

[2.2.6. *Black Box Testing* 34](#_Toc184232005)

[BAB III METODE PENELITIAN 35](#_Toc184232006)

[3.1. *Plan* 35](#_Toc184232007)

[3.2. *Design* 36](#_Toc184232008)

[3.3. *Develop* 37](#_Toc184232009)

[3.4. *Test* 38](#_Toc184232010)

[3.4.1. *Black Box Testing* 38](#_Toc184232011)

[3.4.2. *Mock Election* 40](#_Toc184232012)

[3.5. *Deploy* 41](#_Toc184232013)

[BAB IV HASIL DAN PEMBAHASAN 43](#_Toc184232014)

[4.1. *Plan* 43](#_Toc184232015)

[4.2. *Design* 44](#_Toc184232016)

[4.2.1. Arsitektur Sistem 44](#_Toc184232017)

[4.2.2. *Use Case Diagram* 46](#_Toc184232018)

[4.2.3. *Activity Diagram* 48](#_Toc184232019)

[4.2.4. *Class Diagram* 53](#_Toc184232020)

[4.2.4. *Sequence Diagram* 57](#_Toc184232021)

[4.3. *Develop* 59](#_Toc184232022)

[4.3.1. *Smart Contract* 59](#_Toc184232023)

[4.3.2. Pengembangan Antarmuka 61](#_Toc184232024)

[4.4. *Test* 63](#_Toc184232025)

[4.4.2. *Black Box Testing* 63](#_Toc184232026)

[4.4.3. *Mock Election* 66](#_Toc184232027)

[4.4. *Deploy* 69](#_Toc184232028)

[4.4.1. *Smart Contract* 69](#_Toc184232029)

[BAB V SIMPULAN DAN SARAN 72](#_Toc184232030)

[4.3. Kesimpulan 72](#_Toc184232031)

[4.3. Saran 72](#_Toc184232032)

[DAFTAR PUSTAKA 74](#_Toc184232033)

[LAMPIRAN 77](#_Toc184232034)

DAFTAR GAMBAR

[Gambar 1.1. Kerangka Pemikiran Penelitian 11](#_heading=)

[Gambar 2.1. Struktur Blok dalam sebuah Blockchain 28](#_heading=)

[Gambar 2.2. Struktur Smart Contract 31](#_heading=)

[Gambar 3.1. Metode Agile 35](#_heading=)

[Gambar 4.1. Arsitektur Sistem 44](#_heading=)

[Gambar 4.2 Use Case Diagram 47](#_heading=)

[Gambar 4.3 Activity diagram login 48](#_heading=)

[Gambar 4.4 Activity diagram pendaftaran pemilih 48](#_heading=)

[Gambar 4.5 Activity diagram pendaftaran kandidat 49](#_heading=)

[Gambar 4.6 Activity diagram mengadakan pemilihan 50](#_heading=)

[Gambar 4.7 Activity diagram pemilihan 51](#_heading=)

[Gambar 4.8 Activity diagram hasil pemilihan 52](#_heading=)

[Gambar 4.9 Class Diagram 53](#_heading=)

[Gambar 4.10 Sequence Diagram 58](#_heading=)

[Gambar 4.11 Smart Contract 60](#_heading=)

[Gambar 4.12 Rancangan Antarmuka Admin-Dashboard 61](#_heading=)

[Gambar 4.13 Rancangan Antarmuka Voting Creation 62](#_heading=)

[Gambar 4.14 Rancangan Antarmuka User-Vote 63](#_heading=)

[Gambar 4.15 Add Election 66](#_heading=)

[Gambar 4.16 Election-detail 66](#_heading=)

[Gambar 4.17 User Login 67](#_heading=)

[Gambar 4.18 Metamask Confirmation 67](#_heading=)

[Gambar 4.19 Ganache Server Detail 69](#_heading=)

[Gambar 4.20 Konfigurasi Metamask 70](#_heading=)

[Gambar 4.21 Truffle Migrate 71](#_heading=)

DAFTAR TABEL

[Tabel 2.1 State of the Art 20](#_heading=)

[Tabel 3.1 Skenario Black box Pemilih 38](#_heading=)

[Tabel 3.2 Skenario Black box Admin 39](#_heading=)

[Tabel 4.1 Identifikasi aktor 46](#_heading=)

[Tabel 4.2 Deskripsi class diagram pemilih 53](#_heading=)

[Tabel 4.3 Deskripsi class diagram admin 54](#_heading=)

[Tabel 4.4 Deskripsi class diagram 54](#_heading=)

[Tabel 4.5 Deskripsi class diagram pendaftaran kandidat 55](#_heading=)

[Tabel 4.6 Deskripsi class diagram pendaftaran pemilih 55](#_heading=)

[Tabel 4.7 Deskripsi class diagram memilih 56](#_heading=)

[Tabel 4.8 Deskripsi class diagram hasil pemilihan 56](#_heading=)

[Tabel 4.9 Deskripsi class diagram login 57](#_heading=)

[Tabel 4.10 Hasil skenario Black box Pemilih 63](#_heading=)

[Tabel 4.11 Hasil skenario Black box Admin 64](#_heading=)

[Tabel 4.12 Hasil Mock Election 68](#_heading=)

DAFTAR LAMPIRAN

[Lampiran Hasil Penelitian 77](#_heading=h.36ei31r)

[Lampiran Source Code Smart Contract 83](#_heading=)

[Lampiran Source Code Program 84](#_heading=)

[Lampiran Foto Penelitian 90](#_heading=)

[Lampiran SK Pembimbing Skripsi 92](#_heading=)

[Lampiran Surat Ijin Penelitian Lapangan 93](#_heading=)

[Lampiran SK Telah Menyelesaikan Penelitian 94](#_heading=)

[Lampiran Produk Backlog 95](#_heading=)

BAB I PENDAHULUAN

## **1.1. Latar Belakang**

Pemilu merupakan kesempatan bagi masyarakat Indonesia untuk memilih pemimpin dan wakil rakyat pilihannya. Pemilu di Indonesia sendiri masih menggunakan metode tradisional yaitu pemungutan suara di atas kertas dengan mendatangi tempat pemungutan suara pada waktu yang telah ditentukan selama pemilu. Setelah proses pemungutan suara selesai, surat suara dibuka satu per satu dan suara dihitung. Proses tradisional ini memiliki beberapa kelemahan, diantaranya [1]:

1. Besarnya anggaran biaya yang diperlukan untuk tiap pemungutan suara, mulai dari kertas, biaya cetak, gaji untuk panitia dan pengawas, sampai uang transport.
2. Dapat terjadi kecacatan seperti salah cetak, kertas suara robek, data pemilih yang tidak sesuai, hingga data pemilih ganda.
3. Dapat terjadi kecurangan seperti kertas suara yang telah tercoblos sebelum waktu pemilihan.
4. Lambatnya tabulasi hasil perhitungan suara dikarenakan perhitungan dilakukan dengan secara manual, dan keterbatasan infrastruktur.
5. Banyaknya surat suara yang tidak digunakan menimbulkan sampah kertas yang lama kelamaan hal ini akan berdampak pada kelestarian alam.

Karena kompleksitas dan berbagai kendala dalam pelaksanaannya, proses pemilu tradisional ini sering kali berjalan secara kurang efektif dan efisien. Di sisi lain, kemajuan teknologi telah memberikan berbagai manfaat signifikan dengan menawarkan kemudahan dalam aksesibilitas dan pengurangan biaya. Pemanfaatan teknologi informasi dalam penyelenggaraan pemilu berpotensi meningkatkan efektivitas dan efisiensi. Saat ini, teknologi pemilihan elektronik (*e-voting*) dianggap sebagai alternatif yang tepat untuk mendukung penyelenggaraan pemilu sebagai salah satu pilar utama demokrasi. Pemilihan elektronik juga dikenal sebagai *e-voting*, adalah proses pemilihan umum atau pemilihan lainnya di mana pemilih menggunakan sistem elektronik, seperti perangkat komputer atau perangkat *mobile* untuk memberikan suara mereka, yang bertujuan untuk mempercepat dan mempermudah proses pemungutan dan perhitungan suara pada pemilihan, serta dapat menggantikan kertas suara[2]. Desa Gunungleutik terletak di Kecamatan Ciparay, Kabupaten Bandung, Jawa Barat. Desa ini merupakan salah satu dari beberapa desa yang berada di wilayah Ciparay, dan dikenal sebagai daerah dengan potensi pertanian serta kegiatan masyarakat yang aktif. Dengan total jumlah penduduk 562 jiwa, 287 laki-laki dan 275 perempuan, Karakteristik ini menjadikannya lokasi yang ideal untuk melakukan penelitian terkait penerapan sistem pemilihan elektronik berbasis *blockchain* (*e-voting*).

*Blockchain* memungkinkan setiap transaksi (suara) dicatat secara publik, sehingga dapat diverifikasi oleh semua pihak tanpa memerlukan pihak ketiga. Hal ini menciptakan transparansi penuh dalam proses pemilihan, setiap suara yang diberikan dapat dilacak secara anonim, memungkinkan pemilih memverifikasi bahwa suara mereka telah dihitung tanpa mengetahui identitas pemilih lainnya. *Blockchain*, terutama jaringan publik, dapat menghadapi masalah skalabilitas ketika digunakan untuk jumlah pemilih yang sangat besar. Ini bisa memperlambat proses dan meningkatkan biaya transaksi, terutama di jaringan seperti *ethereum* [2]. Teknologi *blockchain* dapat mengatasi beberapa masalah kritis dalam sistem pemilihan elektronik, seperti *non-repudiation*, transparansi, dan desentralisasi. Namun, masalah-masalah lain seperti kecepatan transaksi dan perlindungan privasi masih menjadi tantangan utama. Misalnya, skalabilitas *blockchain* saat ini tidak cukup untuk pemilu karena ini bisa memperlambat proses dan meningkatkan biaya transaksi,dalam menangani transaksi dalam jumlah besar[3].

## **1.2. Rumusan Masalah**

Membangun sistem pemilihan elektronik menggunakan *blockchain* ini. Berdasarkan latar belakang yang telah dipaparkan, maka rumusan masalahnya adalah sebagai berikut:

1. Bagaimana mendesain dan mengimplementasikan sistem pemilihan elektronik (*e-voting*) berbasis *blockchain*?
2. Bagaimana pengujian melalui *mock election* dan implementasi *smart contract* dalam sistem pemilihan elektronik (*e-voting*) pada jaringan *blockchain*?

## **1.3. Tujuan Penelitian**

Tujuan dari pembuatan sistem ini adalah sebagai berikut:

1. Membangun sistem pemilihan elektronik yang aman berbasis teknologi *blockchain*.
2. Melakukan pengujian melaui *mock election* dan implementasi *smart contract* dalam sistem pemilihan elektronik (*e-voting*).

## **1.4. Manfaat Penelitian**

Berdasarkan tujuan di atas, manfaat penelitian ini dapat dibagi menjadi dua kategori, yaitu manfaat bagi individu (pribadi) dan bagi masyarakat secara umum:

Manfaat bagi pribadi :

1. Peningkatan pengetahuan dan keterampilan. Penelitian ini akan memberikan pengetahuan dan keterampilan baru tentang implementasi teknologi blockchain, terutama dalam sistem pemilihan elektronik. Individu yang mempelajari hasil penelitian ini akan memahami cara kerja blockchain dalam konteks keamanan data dan proses pemilihan yang transparan serta efisien.
2. Pengembangan kompetensi teknologi. Penelitian ini dapat membantu individu yang terlibat dalam pengembangan teknologi informasi, khususnya yang berkaitan dengan keamanan siber, dan sistem terdistribusi untuk mengembangkan kompetensi dalam penerapan teknologi *blockchain* pada sistem yang lebih kompleks seperti pemilu.

Manfaat bagi Masyarakat:

1. Meningkatkan Kepercayaan terhadap pemilu. Dengan adanya sistem pemilihan elektronik berbasis blockchain yang aman, transparan, dan efisien, kepercayaan masyarakat terhadap proses pemilu diharapkan meningkat. Teknologi ini memungkinkan audit yang mudah dilakukan oleh masyarakat luas, sehingga transparansi proses pemilu lebih terjamin.
2. Penyelenggaraan pemilihan menjadi lebih adil dan efisien. *Blockchain* menawarkan sistem yang aman, akurat, anonim, dan adil, yang dapat disesuaikan dengan asas pemilu di Indonesia (langsung, umum, bebas, rahasia, jujur, dan adil). Dengan demikian, masyarakat dapat merasakan manfaat dari proses pemilu yang lebih efisien dan lebih sulit dimanipulasi.

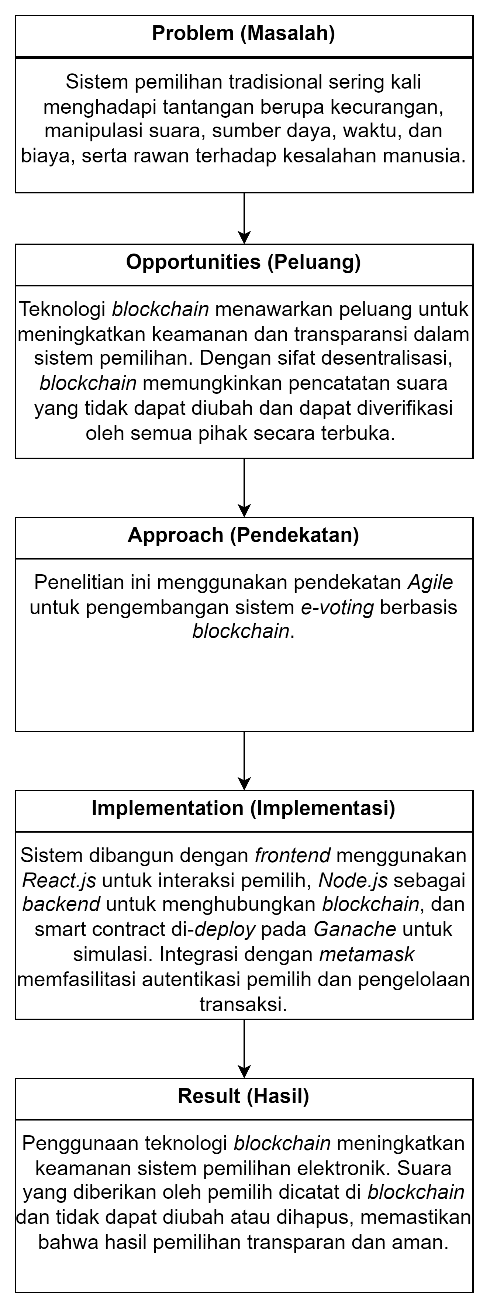
## **1.5. Batasan Masalah**

Penelitian ini mengevaluasi kerangka kerja *blockchain* dan penerapannya sebagai layanan untuk sistem pemilihan elektronik sehingga penelitian yang dilakukan akan dibatasi menjadi berikut ini:

1. Penelitian ini terbatas pada penggunaan teknologi *Blockchain Ethereum* dan bahasa pemrograman *Solidity* untuk merancang dan mengimplementasikan sistem pemilihan elektronik.
2. Sistem yang dikembangkan akan memungkinkan pemilih untuk memberikan suara mereka secara elektronik, menyimpan dan mencatat suara secara aman di dalam *blockchain*.
3. Penelitian ini akan memperhatikan skala yang cocok untuk implementasi pemilihan elektronik berbasis *blockchain*. Namun, skala ekstensif atau implementasi pada tingkat nasional tidak akan menjadi fokus pada tahap ini, penelitian ini akan fokus pada skala pemilihan kepala desa, yaitu di desa Gunungleutik, Kecamatan Ciparay.
4. Implementasi pada skala desa memungkinkan pengujian sistem dalam lingkungan yang lebih terkendali dan terukur sebelum diterapkan di tingkat yang lebih besar.
5. Pada *blockchain*, terutama jaringan publik, dapat menghadapi masalah skalabilitas ketika digunakan untuk jumlah pemilih yang sangat besar. Ini bisa memperlambat proses dan meningkatkan biaya transaksi, terutama di jaringan seperti *ethereum*.
6. Sistem ini masih terbatas penggunaanya di jaringan *blockchain* lokal.

## **1.6. Kerangka Pemikiran Penelitian**

Pada Gambar 1.1 merupakan kerangka pemikiran yang digunakan dalam penelitian ini:



#### Gambar 1.1. Kerangka Pemikiran Penelitian

## **1.7. Sistematika Penulisan**

Penelitian ini disusun dengan sistematika penulisan yang bertujuan untuk memberikan gambaran umum. Sistematika penulisan pada tugas akhir ini adalah sebagai berikut:

**BAB I: Pendahuluan**

Bab ini menguraikan pendahuluan penelitian, yang meliputi latar belakang penelitian, rumusan masalah, tujuan penelitian, batasan masalah, kerangka pemikiran, dan sistematika penulisan.

**BAB II: Kajian Literatur**

Bab ini membahas kajian literatur yang meliputi tinjauan terhadap penelitian-penelitian sebelumnya serta penjelasan mengenai sejumlah teori yang relevan untuk mengatasi permasalahan penelitian.

**BAB III: Metodologi Penelitian**

Bab ini membahas metode penelitian yang diimplementasikan dalam kajian ini, disertai dengan penjelasan terperinci mengenai setiap langkah dan teknik yang digunakan dalam penelitian.

**BAB IV: Hasil dan Pembahasan**

Bab ini menguraikan tentang proses dan hasil yang telah diperoleh dalam penelitian ini.

**BAB V: Simpulan dan Saran**

Bab ini menjelaskan tentang bagian akhir dari penelitian, seperti kesimpulan secara keseluruhan yang dapat menjawab rumusan masalah dengan saran untuk penelitian selanjutnya dan bertujuan untuk mengembangkan penelitian lebih lanjut.

BAB II KAJIAN LITERATUR

## **2.1. *State of the Art***

Penelitian ini menggali informasi dari beberapa penelitian sebelumnya sebagai bahan perbandingan, baik mengenai kekurangan maupun kelebihan yang sudah ada. Selain itu, informasi juga diperoleh dari buku-buku, jurnal ilmiah, dan skripsi untuk mendapatkan pemahaman tentang teori-teori yang relevan dengan topik penelitian, sehingga dapat membangun landasan teori ilmiah yang kuat. Sistem Pemilihan Elektronik sebelumnya sudah pernah dibuat dan digunakan, namun dalam program aplikasi yang berbeda-beda. Beberapa sistem Pemilihan Elektronik yang berhubungan dengan yang pernah dibuat adalah :

* + 1. Kevin Curran [4], membuat sebuah penelitian yang berjudul "*E-Voting on the Blockchain*", membahas potensi penggunaan teknologi *blockchain* untuk sistem pemungutan suara elektronik yang aman dan andal. penelitian ini menyoroti tantangan dalam membangun sistem pemungutan suara elektronik yang aman dan mengutip sistem pemungutan suara online yang ditinggalkan oleh Pentagon, Amerika Serikat sebagai contoh. Penelitian ini mengeksplorasi penggunaan teknologi *blockchain* untuk mengatasi tantangan-tantangan ini, dengan fokus pada potensinya untuk pemungutan suara yang aman, dapat diandalkan, dan *anonim*. Membangun sebuah sistem pemungutan suara elektronik yang aman itu sulit, seperti yang ditunjukkan oleh sistem pemungutan suara online yang ditinggalkan oleh Pentagon AS. Teknologi blockchain berpotensi untuk menjawab tantangan keamanan dan keandalan dalam sistem pemungutan suara elektronik. Penelitian ini membahas berbagai penelitian dan implementasi sistem pemungutan suara elektronik berbasis *blockchain*, seperti *Bitcoin*, *Ethereum*, *Hyperledger Fabric*, dan *Hyperledger Sawtooth*. Penelitian ini juga membahas masalah yang berkaitan dengan skalabilitas, privasi, dan keamanan dalam sistem pemungutan suara elektronik berbasis blockchain.
    2. Uzma Jafar, Mohd Juzaiddin Ab Aziz, dan Zarina Shukur [5], dengan penelitian yang berjudul "*Blockchain for Electronic Voting System—Review and Open Research Challenges*" memberikan tinjauan yang komprehensif mengenai penggunaan teknologi *blockchain* pada sistem pemungutan suara elektronik. Penelitian ini mendiskusikan manfaat potensial dari sistem *e-voting* berbasis *blockchain*, seperti keamanan, integritas, dan transparansi, serta tantangan yang perlu diatasi untuk implementasi yang sukses. Pemungutan suara online semakin populer karena potensinya untuk mengurangi biaya penyelenggaraan dan meningkatkan jumlah pemilih. Teknologi *blockchain* dapat meningkatkan keamanan dan integritas sistem pemungutan suara elektronik. Penelitian ini membahas berbagai penelitian dan implementasi sistem *e-voting* berbasis *blockchain*, seperti *DecentraVote*, Sistem *E-Voting* Berbasis *Blockchain*, dan *framework* Int & Inter. Penelitian ini menyoroti manfaat dari sistem *e-voting* berbasis *blockchain*, termasuk keamanan, integritas, dan transparansi. Penelitian ini juga membahas tantangan yang dihadapi dalam mengimplementasikan sistem *e-voting* berbasis *blockchain*, seperti skalabilitas, privasi, dan kecepatan transaksi. Penelitian ini menyimpulkan bahwa meskipun teknologi *blockchain* menjanjikan sistem pemungutan suara elektronik yang aman dan dapat diandalkan, masih ada tantangan yang harus dihadapi, seperti perlindungan privasi, skalabilitas, dan kecepatan transaksi. Mereka menyarankan bahwa penelitian lebih lanjut diperlukan untuk mengatasi tantangan-tantangan ini dan sepenuhnya menyadari potensi sistem *e-voting* berbasis *blockchain*.
    3. Basit Shahzad dan Jon Crowcroft [6], dengan penelitian yang berjudul "*Trustworthy Electronic Voting Using Adjusted Blockchain Technology*" membahas keefektifan penggunaan teknologi *blockchain* untuk meningkatkan keamanan dan integritas sistem pemungutan suara elektronik. Penelitian ini mengusulkan sebuah *framework* yang menjawab tantangan sistem *e-voting* berbasis *blockchain*, seperti skalabilitas, privasi, dan kecepatan transaksi. Penelitian ini mendiskusikan tantangan dalam membangun sebuah sistem pemungutan suara elektronik yang aman dan mengutip sistem pemungutan suara online yang ditinggalkan oleh Pentagon Amerika Serikat sebagai contoh. *Framework* yang diusulkan berfokus pada keefektifan proses pemungutan suara, algoritma *hashing*, pembuatan dan penyegelan blok, akumulasi data, dan banyak lagi. Penelitian ini menyoroti manfaat dari sistem *e-voting* berbasis *blockchain*, termasuk keamanan, integritas, dan transparansi. Penelitian ini juga membahas tantangan yang dihadapi dalam mengimplementasikan sistem *e-voting* berbasis *blockchain*, seperti skalabilitas, privasi, dan kecepatan transaksi. Penelitian ini menyimpulkan bahwa meskipun teknologi *blockchain* menjanjikan sistem pemungutan suara elektronik yang aman dan dapat diandalkan, masih ada tantangan yang harus diatasi, seperti perlindungan privasi, skalabilitas, dan kecepatan transaksi. Penelitian ini menyarankan bahwa penelitian lebih lanjut diperlukan untuk mengatasi tantangan-tantangan ini dan sepenuhnya menyadari potensi sistem *e-voting* berbasis *blockchain*. Penelitian ini berkontribusi pada penelitian yang sedang berlangsung di bidang sistem *e-voting* berbasis *blockchain* dan memberikan implementasi praktis dari teknologi untuk pemilihan elektronik.
    4. Setiawan Restu Aji, Wahyuningdiah Trisari Harsanti Putri [7], "Implementasi Teknologi *Blockchain* dalam Aplikasi *E-Voting* Berbasis *Mobile*" oleh membahas tentang implementasi teknologi *blockchain* dalam sistem pemungutan suara elektronik berbasis *mobile*. Penelitian ini berfokus pada penggunaan *blockchain* untuk mengatasi masalah keamanan dan integritas pada aplikasi *e-voting* dan pengembangan aplikasi *e-voting* berbasis *mobile* untuk meningkatkan aksesibilitas. Penelitian ini mendiskusikan tantangan dalam membangun sistem pemungutan suara elektronik yang aman. Solusi yang diusulkan berfokus pada penggunaan teknologi blockchain untuk meningkatkan keamanan dan integritas sistem pemungutan suara elektronik. Penelitian ini menyoroti manfaat dari sistem e-voting berbasis blockchain, termasuk keamanan, integritas, dan transparansi. Penelitian ini membahas tantangan yang dihadapi dalam mengimplementasikan sistem e-voting berbasis blockchain, seperti skalabilitas, privasi, dan kecepatan transaksi.
    5. Fadli Fajri, Singgih Jatmiko, dan Missa Lamsani [8], “*Electronic Voting Using Decentralized System Based on Ethereum Blockchain*” merupakan penelitian yang mengeksplorasi implementasi sistem pemungutan suara elektronik yang terdesentralisasi dengan menggunakan teknologi *blockchain* *ethereum*. Penelitian ini bertujuan untuk menjawab tantangan sistem pemungutan suara elektronik tradisional, seperti keamanan, integritas, dan ketergantungan pada otoritas pusat. Penelitian ini mengusulkan sebuah sistem terdesentralisasi yang menggunakan blockchain Ethereum untuk memastikan keamanan, integritas, dan transparansi dalam proses pemungutan suara. Sistem ini dirancang untuk dapat diakses melalui perangkat seluler, yang meningkatkan partisipasi dan aksesibilitas pemilih. Penelitian ini menyoroti manfaat penggunaan teknologi blockchain untuk mengatasi masalah keamanan dan integritas dalam sistem pemungutan suara elektronik. Penelitian ini juga membahas pengembangan aplikasi *e-voting* berbasis seluler untuk meningkatkan aksesibilitas dan kontribusi penelitian terhadap pengembangan aplikasi blockchain dalam berbagai konteks pemilu.
    6. Hjálmarsson dan Gunnlaugur K. Hreiðarsson [9] "*Blockchain-Based E-Voting System*" Tujuan dari penelitian ini adalah mengevaluasi penggunaan teknologi blockchain untuk membangun sistem e-voting yang aman dan terdistribusi. Penelitian ini mengidentifikasi batasan teknologi dan hukum terkait penggunaan blockchain dan mengusulkan sistem baru yang dapat mengatasi batasan tersebut. Tujuan ini jelas dan sesuai dengan isu keamanan pemilu yang sering menjadi perhatian dalam demokrasi modern. Penelitian ini menggunakan metode evaluasi terhadap beberapa framework blockchain populer seperti Exonum, Quorum, dan Geth. kemudian penelitian mengusulkan sistem yang menggunakan blockchain "*permissioned*" berbasis konsensus *Proof of Authority* (PoA) yang dirancang untuk memenuhi persyaratan pemilu. Penelitian ini juga menunjukkan bahwa sistem blockchain dapat memperbaiki keamanan pemilu dan mengurangi biaya operasional. Misalnya, penggunaan kontrak pintar dalam sistem mereka memungkinkan perhitungan suara yang lebih cepat dan terverifikasi secara otomatis oleh beberapa node. Penelitian ini menyimpulkan bahwa teknologi blockchain menawarkan potensi besar untuk pemilu demokratis yang lebih aman, transparan, dan hemat biaya. Namun, sistem ini masih memerlukan pengembangan lebih lanjut dalam hal skalabilitas dan ketahanan terhadap serangan.
    7. Zheng, Zibin, Xie, Shaoan, Dai, dan Hong-Ning. Dalam penelitianya yang berjudul “*Blockchain challenges and opportunities*: *A survey*” [10]. Penelitian ini membahas tentang teknologi blockchain, yang memiliki banyak kelebihan seperti decentralisasi, persistensi, anonimitas, dan auditabilitas. Meskipun banyak penelitian yang memfokuskan pada penggunaan teknologi blockchain dalam berbagai aspek aplikasi, belum ada survei yang komprehensif tentang teknologi *blockchain* dari perspektif teknis dan aplikatif. Penelitian ini menjelaskan bahwa blockchain dapat dianggap sebagai buku besar publik, di mana semua transaksi yang dikomitmen disimpan dalam rantai blok yang terus tumbuh ketika blok baru ditambahkan. diakhiri dengan membahas arah masa depan teknologi *blockchain*, termasuk kebutuhan untuk mengembangkan algoritma konsensus yang lebih efisien dan solusi untuk menangani tantangan yang diidentifikasi.
    8. Wu, K., Liu, M., & Xu, J. Pada penelitianya yang berjudul “*How Will Blockchain Technology Impact Auditing and Accounting: Permissionless Vs. Permissioned Blockchain”* [11]. Menjelajahi dampak potensial teknologi blockchain terhadap praktik audit dan akuntansi. Penelitian ini membedah perbedaan antara *permissionless blockchain* dan *permissioned blockchain*, serta mengulas bagaimana variasi ini mungkin mempengaruhi proses audit dan akuntansi. *Permissionless blockchain* terbuka untuk siapa saja dan tidak memerlukan izin sebelumnya untuk berpartisipasi. Contoh termasuk *Bitcoin* dan *Ethereum*. Penelitian ini membahas tantangan yang terkait masalah skalabilitas dan potensi aktivitas berbahaya. Sedangkan *permissioned blockchain* dikontrol oleh kelompok tertentu atau organisasi dan memerlukan izin untuk bergabung ke jaringan.
    9. Mohanta, Bhabendu & Panda, Soumyashree & Jena, Debasish. Dalam penelitianya yang berjudul “*An Overview of Smart Contract and Use Cases in Blockchain Technology”* [12]. Memberikan gambaran menyeluruh tentang *smart contract* dan aplikasinya dalam teknologi *blockchain*. Penelitian ini menjelaskan konsep dasar dari *smart contract*, cara kerjanya, serta berbagai kasus penggunaan yang relevan dalam berbagai industri. Penelitian ini menjelaskan bahwa *smart contract* adalah program komputer yang secara otomatis mengeksekusi, mengontrol, atau mendokumentasikan peristiwa dan tindakan sesuai dengan ketentuan kontrak. Kontrak ini disimpan di dalam *blockchain* dan dapat dieksekusi tanpa memerlukan pihak ketiga. Penelitian ini mengurai beberapa karakteristik dari *smart contract* yaitu otomatisasi, transparansi dan keamanan, dengan beberapa pengguanaan seperti pada sektor keungan, kesehatan dan pendidikan. Dengan menjelaskan karakteristik utama, kasus penggunaan yang relevan, serta tantangan yang dihadapi, penelitian ini menjadi sumber informasi yang berharga bagi peneliti, pengembang, dan profesional di bidang teknologi informasi yang tertarik pada inovasi *blockchain*.
    10. Salman, S. A. B., Al-Janabi, S., & Sagheer, A. M. [13]. Dalam penelitianya yang berjudul “*A Review on E-Voting Based on Blockchain Models*.” memberikan gambaran menyeluruh tentang penggunaan *blockchain* dalam sistem *e-voting*. Penelitian ini menjelaskan apakah *blockchain* dapat digunakan untuk membangun sistem *e-voting* yang efisien, serta mengidentifikasi beberapa teknologi *blockchain* yang relevan dan kelebihan serta kekurangannya. Sistem *e-voting* adalah cara modern untuk melakukan pemungutan suara secara *online* menggunakan kriptografi. Pemungutan suara dapat dilakukan dari kenyamanan rumah menggunakan komputer atau perangkat *mobile*. Hasil pemungutan suara akhir dapat dihitung secara otomatis oleh *server* pusat. Kesulitan dalam sistem e-voting meliputi integritas data, keandalan, privasi suara, konsekuensi gangguan, pemungut suara yang tidak terdidik, keterampilan IT yang spesialis, penyimpanan peralatan, keamanan, efek penipuan, dan biaya. Dalam beberapa tahun terakhir, penggunaan *blockchain* dalam sistem *e-voting* telah menjadi pilihan yang lebih penting untuk mengatasi banyak masalah keamanan yang mungkin timbul dengan sistem *e-voting*. *Blockchain* memungkinkan pemerintah untuk mengimplementasikan sistem pemungutan suara cerdas yang terintegrasi dengan informasi keberlanjutan untuk memastikan bahwa semua anggota memiliki informasi yang akurat tentang properti yang sedang berlangsung. Penelitian ini menyimpulkan dengan membahas potensi masa depan dari penggunaan *blockchain* dalam sistem *e-voting*. Mereka menekankan bahwa seiring dengan kemajuan teknologi *blockchain* dan peningkatan pemahaman tentang penggunaannya, blockchain akan semakin banyak diterapkan di berbagai industri untuk meningkatkan keamanan, transparansi, dan efisiensi dalam pemungutan suara.

Secara umum, penelitian sebelumnya lebih berfokus pada tantangan penelitian terbuka dan peluang masa depan dalam *blockchain* untuk *e-voting*, sedangkan penelitian perancangan dan implementasi sistem elektronik berbasis blockchain, menawarkan solusi implementasi teknis langsung dalam bentuk *smart contract* untuk *e-voting*. Penelitian sebelumnya menekankan kebutuhan akan inovasi dalam hal privasi dan skalabilitas, sementara penelitian ini menunjukkan bagaimana teknologi ini bisa diimplementasikan secara nyata dalam sistem pemilu meskipun masih terbatas dalam skala kecil. Kebaruan dari penelitian ini terletak pada desain sistem nyata yang menggunakan *solidity* untuk mengembangkan sistem *e-voting* yang dapat diuji dan digunakan. Penelitian ini menunjukkan bagaimana *smart contract* dapat menggantikan peran otoritas pemilihan dalam menghitung suara dan menjaga transparansi, memberikan solusi langsung dan praktis untuk beberapa tantangan yang disebutkan dalam penelitian sebelumnya seperti transparansi dan otomasi proses. Penelitian ini memberikan kontribusi baru dengan mengusulkan penggunaan *blockchain* *permissioned* berbasis *PoA* untuk sistem *e-voting* dan mengeksplorasi aplikasi *smart contract* untuk pemilu. Selain itu, integrasi *face recognition* untuk menjaga privasi suara menunjukkan pendekatan inovatif terhadap masalah privasi dalam pemilu.

##### Tabel 2.1 *State of the Art*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Judul** | **Penulis** | **Tahun** | **Pembahasan** |
| 1 | *E-Voting on the Blockchain* | Kevin Curran | 2018 | Penelitian ini mengeksplorasi penggunaan teknologi *blockchain* untuk mengatasi tantangan-tantangan ini, dengan fokus pada potensinya untuk pemungutan suara yang aman, dapat diandalkan, dan *anonim*. |
| 2 | *Blockchain for Electronic Voting System—Review and Open Research Challenges* | Uzma Jafar, Mohd Juzaiddin Ab Aziz, dan Zarina Shukur | 2021 | Penelitian ini mendiskusikan manfaat potensial dari sistem *e-voting* berbasis *blockchain*, seperti keamanan, integritas, dan transparansi, serta tantangan yang perlu diatasi untuk implementasi yang sukses. |
| 3 | *Trustworthy Electronic Voting Using Adjusted Blockchain Technology* | Basit Shahzad dan Jon Crowcroft | 2019 | Penelitian mengusulkan sebuah *framework* yang menjawab tantangan sistem *e-voting* berbasis *blockchain*, seperti skalabilitas, privasi, dan kecepatan transaksi. |
| **No** | **Judul** | **Penulis** | **Tahun** | **Pembahasan** |
| 4 | Implementasi Teknologi *Blockchain* dalam Aplikasi *E-Voting* Berbasis *Mobile* | Setiawan Restu Aji, Wahyuningdiah Trisari Harsanti Putri | 2023 | Penelitian ini berfokus pada penggunaan *blockchain* untuk mengatasi masalah keamanan dan integritas pada aplikasi *e-voting* dan pengembangan aplikasi *e-voting* berbasis *mobile* untuk meningkatkan aksesibilitas. |
| 5 | *Electronic Voting Using Decentralized System Based on Ethereum Blockchain* | Fadli Fajri, Singgih Jatmiko, dan Missa Lamsani | 2020 | Penelitian ini mengusulkan sebuah sistem terdesentralisasi yang menggunakan blockchain Ethereum untuk memastikan keamanan, integritas, dan transparansi dalam proses pemungutan suara. |
| 6 | *Blockchain-Based E-Voting System* | Hjálmarsson dan Gunnlaugur K. Hreiðarsson | 2018 | Teknologi blockchain menawarkan potensi besar untuk pemilu demokratis yang lebih aman, transparan, dan hemat biaya. Namun, sistem ini masih memerlukan pengembangan lebih lanjut dalam hal skalabilitas dan ketahanan terhadap serangan. |
| **No** | **Judul** | **Penulis** | **Tahun** | **Pembahasan** |
| 7 | *Blockchain challenges and opportunities*: *A survey* | Zheng, Z., Xie, S., Dai, H-N., Chen, X., & Wang, H. | 2018 | Membahas tentang teknologi blockchain, yang memiliki banyak kelebihan seperti decentralisasi, persistensi, anonimitas, dan auditabilitas. |
| 8 | *How Will Blockchain Technology Impact Auditing and Accounting: Permissionless Vs. Permissioned Blockchain* | Wu, K., Liu, M., & Xu, J. | 2019 | Menjelajahi dampak potensial teknologi blockchain terhadap praktik audit dan akuntansi. Penelitian ini membedah perbedaan antara *permissionless blockchain* dan *permissioned blockchain*, serta mengulas bagaimana variasi ini mungkin mempengaruhi proses audit dan akuntansi. |
| 9 | *An Overview of Smart Contract and Use Cases in Blockchain Technology* | Mohanta, Bhabendu & Panda, Soumyashree & Jena, Debasish | 2018 | Memberikan gambaran menyeluruh tentang *smart contract* dan aplikasinya dalam teknologi *blockchain*. Penelitian ini menjelaskan konsep dasar dari *smart contract*, cara kerjanya, serta berbagai kasus penggunaan yang relevan dalam berbagai industri. |
| **No** | **Judul** | **Penulis** | **Tahun** | **Pembahasan** |
| 10 | *A Review on E-Voting Based on Blockchain Models.* | Salman, S. A. B., Al-Janabi, S., & Sagheer, A. M. | 2022 | Penelitian ini menjelaskan apakah *blockchain* dapat digunakan untuk membangun sistem *e-voting* yang efisien, serta mengidentifikasi beberapa teknologi *blockchain* yang relevan dan kelebihan serta kekurangannya. |
| 11 | *Perancangan dan Implementasi Sistem Pemilihan Elektronik Berbasis Blockhchain* |  | 2024 | Penelitian ini menawarkan solusi implementasi teknis langsung dalam bentuk *smart contract* untuk *e-voting*. penelitian ini menunjukkan bagaimana teknologi ini bisa diimplementasikan secara nyata dalam sistem pemilu meskipun masih terbatas dalam skala kecil. Kebaruan dari penelitian ini terletak pada desain sistem nyata yang menggunakan *solidity* untuk mengembangkan sistem *e-voting* yang dapat diuji dan digunakan. |

## **2.2. Dasar Teori**

Dasar teori adalah landasan atau struktur konseptual yang dipergunakan untuk menguraikan teori-teori yang relevan dalam mendukung suatu penelitian. Berikut adalah dasar teori yang bisa mendukung penelitian ini:

### **2.2.1. Blockchain**

*Blockchain* adalah teknologi revolusioner yang memberikan dampak besar pada masyarakat modern karena transparansi, desentralisasi, dan keamanannya. *Blockchain* mendapatkan perhatian yang cukup besar karena aplikasi pertama dari mata uang digital, misalnya *Bitcoin*. Dalam waktu dekat, teknologi *Blockchain* bertekad untuk mengubah cara kita hidup, berinteraksi, dan menjalankan bisnis. Baru-baru ini, para akademisi, industrialis, dan peneliti secara agresif menyelidiki berbagai aspek *Blockchain* sebagai teknologi yang sedang berkembang [14], tidak hanya transfer uang tetapi juga semua jenis informasi struktural dapat disimpan dalam *blockchain* ini, dan dengan bantuan beberapa metode *kriptografi*, sistem dapat dipertahankan dengan aman [15]. Banyak informasi dapat direkam dengan sistem ini dengan modifikasi yang relevan. *ethereum* misalnya, *cryptocurrency* lain dengan lingkungan pengembangan multiguna, yang muncul beberapa tahun setelah *bitcoin*, membedakan *blockchain* secara nyata, mengungkapkan bahwa teknologi ini dapat menghasilkan perangkat lunak yang dapat menyimpan informasi yang terstruktur seperti dijelaskan di atas. Selain itu, ada manfaat yang dimiliki *blockchain* yang dapat digunakan dalam mendesain sistem yaitu [11]:

1. *Security*, dengan menggunakan enkripsi asimetris. Kunci yang dihasilkan adalah dalam bentuk dua pasang kunci publik dan privat yang terkait secara matematis. Kunci publik akan dipublikasikan dalam sistem dan kunci pribadi hanya diakses oleh pengguna. Hanya orang yang memiliki kunci pribadi yang dapat menggunakan akun.
2. *Accuracy*, dalam pemilihan setiap pengguna akan diverifikasi oleh lembaga dan mendaftar sehingga pengguna yang dapat memilih adalah pengguna yang sudah diverifikasi
3. *Transparency*, Semua surat suara dalam sistem disimpan di *blockchain* sehingga pengguna dapat melihat berapa banyak suara yang dimiliki masing masing kandidat, selain itu sistem ini dapat dan mudah diaudit.
4. *Autonomy*, hal besar yang mengikuti sistem pemungutan suara ini adalah bahwa memiliki keuntungan di bidang sistem desentralisasi.
5. *Anonymity*, Sistem pemungutan suara yang digunakan adalah anonim dimana penyimpanan data yang terdapat di dalam *blockchain* tidak ada data mengenai pemilih melainkan hanya berupa alamat publik dari setiap dompet digital karena dalam *blockchain* itu sendiri hanya mencatat
6. *Fairness*, setiap pengguna dapat memilih sesuai pilihan mereka.
7. *Efficiency*, target yang diharapkan dari sistem ini adalah untuk meminimalkan biaya yang digunakan untuk operasi.

*Blockchain* merupakan sebuah buku besar yang dapat diakses oleh publik dan setiap transaksi yang terjadi dicatat dalam blok yang saling terhubung satu dengan yang lainnya sehingga berbentuk seperti rantai [8]. *Blockchain* juga bisa disebut sebuah basis data terdistribusi dari catatan atau buku besar dari semua transaksi yang telah dieksekusi dan dibagikan kepada pihak-pihak yang berpartisipasi. Setiap transaksi dalam buku besar diverifikasi oleh konsensus mayoritas peserta dalam sistem. Dan sekali masuk, informasi tidak akan pernah bisa dihapus. Dalam teknologi blockchain, blockchain dapat dibagi menjadi 3 jenis yaitu [10]:

1. *Public Blockchain,* merupakan jaringan yang terdistribusi secara besar serta dapat diakses bagi siapa saja untuk berpartisipasi dan memiliki kode *open source* yang dikelola oleh komunitas pendiri.
2. *Permissioned Blockchain*, adalah jenis *blockchain* yang dapat dijalankan secara individu pada jaringan besar yang terdistribusi tetapi kode intinya ada yang o*pen source* dan tidak.
3. *Private Blockchain*, merupakan tipe *blockchain* yang keanggotaannya dikendalikan secara terpusat atau biasa disebut *Centralized*.

Setiap jenis *blockchain* menggunakan teknologi kriptografi yang memungkinkan setiap anggota di jaringan tertentu untuk mengelola struktur data dengan cara yang aman tanpa perlu otoritas pusat untuk menegakkan aturan. Teknologi Blockchain memiliki cara kerja dengan mendistribusikan buku besar ke dalam jaringan komputer dan melakukan verifikasi. Setiap komputer memiliki buku besar (*Blockchain*) dari semua transaksi yang dicatat untuk diumumkan. Transaksi yang telah melalui verifikasi akan dimasukan ke dalam blok. Selain itu, blok secara permanen akan saling terhubung dengan transaksi yang terjadi sebelum dan sesudahnya dan membentuk sebuah rantai yang dinamakan blockchain. Karena *blockchain* terdiri dari dua jenis record, transaksi dan blok. Tidak ada server terpusat untuk transaksi karena setiap blok harus memenuhi syarat dan juga tidak akan menimpa transaksi sebelumnya [16].

Blockchain sendiri terbagi menjadi dua yaitu bitcoin based dan ethereum based dimana ethereum based ini lebih mementingkan keperluan pengembangan blockchainya,

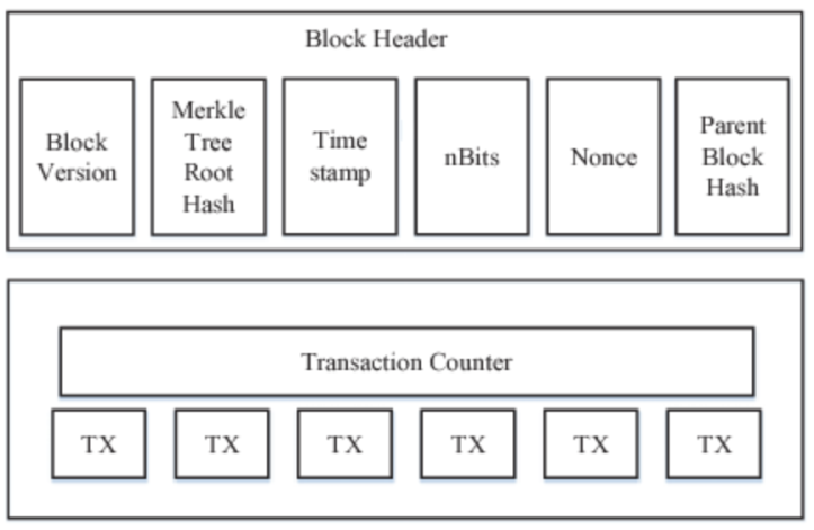
*Blockchain* adalah urutan blok, yang menyimpan daftar lengkap catatan transaksi seperti buku besar publik konvensional. Dalam sebuah blok yang terdapat di blockchain terdapat dua bagian yaitu [10]:

1. *Block header*

Pada block header terdapat:

1. *Block version*, berisikan tentang aturan mana yang harus diikuti.
2. *Merkle Tree Root Hash,* yang berisikan nilai hash dari semua transaksi di blok.
3. *Time Stamp,* yang berisikan waktu saat ini dalam satuan waktu universal.
4. *nBits,* yang berisikan batas dari target blok yang valid.
5. *Nonce,* yang berisikan 4 *byte* yang biasa dimulai dari angka 0 dan meningkat setiap terjadi perhitungan *hash*.
6. *Parent Block Hash,* yang berisikan nilai dari hash sebelumnya.
7. *Block body*

*Block Body* berisikan perhitungan transaksi dan transaksi itu sendiri. Jumlah dari banyaknya transaksi yang dapat ditampung dalam sebuah blok tergantung pada ukurannya.



#### Gambar 2.1. Struktur Blok dalam sebuah *Blockchain* [10]

### **2.2.2. Pemilihan Elektronik**

Pemilihan elektronik, juga dikenal sebagai *e-voting*, adalah proses pemilihan umum atau pemilihan lainnya di mana pemilih menggunakan sistem elektronik, seperti komputer atau perangkat mobile, untuk memberikan suara mereka. Ini berbeda dengan pemilihan tradisional di mana pemilih memberikan suara mereka secara manual dengan menggunakan kertas dan pensil atau alat lainnya. Pemilihan elektronik (*e-voting*) adalah sebuah proses pemungutan suara yang dilakukan dengan bantuan media teknologi informasi yang bertujuan untuk mempercepat dan mempermudah proses pemungutan dan perhitungan suara pada pemilihan umum, serta dapat menggantikan kertas suara [2]. Pemilihan elektronik (*e-voting*) ini pertama kali diperkenalkan oleh *David Shaum* pada awal tahun 1980. Sistem *e-voting* ini menggunakan *cryptography-key* yang dapat membantu para *voter* untuk tetap tidak terdeteksi (*anonym*). *Washington D. C*. pernah mengembangkan pemilihan elektronik pada tahun 2010. Tetapi terjadi banyak sekali masalah keamanan pada saat melakukan pengujian pada sistemnya. Sehingga proyek tersebut gagal untuk diimplementasikan [17].

### **2.2.3. Ethereum**

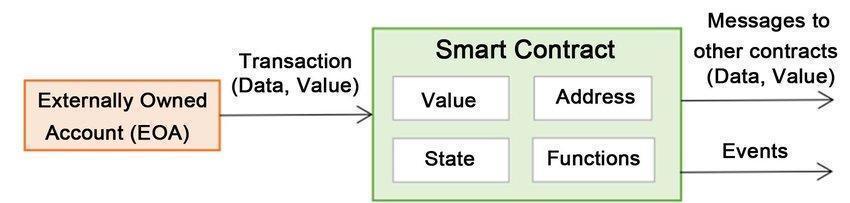
*Ethereum* adalah *platform* perangkat lunak terdesentralisasi yang berjalan di atas teknologi *blockchain*. Diciptakan oleh *Vitalik Buterin* pada tahun 2013 dan diluncurkan pada tahun 2015, *Ethereum* memungkinkan pengembang untuk membangun dan menjalankan aplikasi terdesentralisasi (*DApps*) serta kontrak pintar (*smart contracts*) di atas *blockchain Ethereum*. Dalam beberapa hal, *ethereum* ini tidak berbeda dengan *bitcoin*, tetapi ada beberapa hal yang berbeda diantara keduanya. *Bitcoin* menawarkan sistem *digital currency* yang *peer to peer*, sementara *ethereum* biasanya berfokus untuk penggunaan *smart contract* dari berbagai aplikasi terdesentralisasi [18]. *Ethereum* memberikan kebebasan kepada pengembang untuk membuat program mereka sendiri ke dalam kontrak, sehingga pengembang bebas untuk menerapkan logika bisnis ke dalam *blockchain* (Vujicic et al., 2018). Salah satu keunggulan utama *Ethereum* dibandingkan *blockchain* lainnya adalah dukungan untuk *Turing-complete smart contracts*. Ini memungkinkan pembuatan logika pemrograman yang kompleks, yang sangat penting Sistem Pemilihan Elektronik (*e-voting)* yang membutuhkan berbagai aturan dan kondisi. *Ethereum* memiliki komunitas pengembang yang besar dan ekosistem yang luas. Banyak *tools*, *library*, dan *framework* yang dikembangkan untuk *Ethereum*, membuatnya lebih mudah untuk membangun dan memelihara Sistem Pemilihan Elektronik (*e-voting)*. *Ethereum* ini terus dikembangkan dan ditingkatkan oleh komunitasnya, dengan pembaruan dan peningkatan berkelanjutan. Ini memastikan bahwa jaringan *Ethereum* tetap aman, efisien, dan mampu mengakomodasi kebutuhan aplikasi masa depan.

*Ethereum Virtual Machine* (EVM) adalah jantung dari *ethereum*. Setiap *node* dalam jaringan *ethereum* menjalankan EVM yang dapat mengeksekusi kode algoritma yang kompleks yang ditulis dalam bahasa pemrograman seperti *javascript* dan *python* (P.et al., 2018). Penggunaan *Ethereum* dalam sistem pemilihan elektronik (*e-voting*) didasarkan pada keunggulan teknologi *blockchain* yang menawarkan transparansi, keamanan, dan imutabilitas transaksi. *Smart contract* di *Ethereum* dapat digunakan untuk menjalankan program yang berisi kode yang dieksekusi otomatis ketika kondisi dan persyaratan tertentu terpenuhi. Dalam konteks *e-voting*, *smart contract* dapat mengatur dan mengelola transaksi suara secara otomatis tanpa membutuhkan perantara seperti pengguna hukum atau pihak pengawas. Dengan menggunakan teknologi *Ethereum*, sistem *e-voting* dapat memanfaatkan kontrak pintar (*smart contracts*) untuk memastikan bahwa hanya suara yang sah yang dihitung, suara diberikan secara anonim, dan hasil pemilihan dihitung dengan cepat juga akurat. beberapa keuntungan menggunakan Ethereum dalam Sistem Pemilihan Elektronik:

1. **Transparansi:** Semua transaksi suara dapat dilacak dan diverifikasi di *blockchain*, sehingga memastikan transparansi penuh dalam proses pemilihan.
2. **Keamanan:** *Smart contract* memastikan bahwa data suara tidak dapat diubah setelah dicatat di *blockchain*, sehingga menjamin keamanan dan kejujuran hasil pemilihan.
3. **Imutabilitas:** Data yang dicatat di *blockchain* tidak dapat diubah atau dihapus, menjamin bahwa hasil pemilihan tetap utuh dan bebas dari manipulasi.
4. **Efisiensi:** Dengan menghilangkan perantara dan menjalankan proses pemilihan secara otomatis, *e-voting* berbasis *Ethereum* dapat mengurangi biaya dan meningkatkan efisiensi proses pemilihan.

### **2.2.4. *Smart Contract***

*Smart contract* adalah program yang berjalan di jaringan *blockchain*. *Smart contract* adalah transaksi yang berisi kode yang akan dieksekusi otomatis ketika kondisi dan persyaratan tertentu terpenuhi. *Smart contract* tidak membutuhkan intermediari seperti pengguna hukum atau pihak pengawas untuk menjamin perjanjian. *Smart contract* dapat mengatur dan mengatur transaksi, mengatur proses, dan mengelola aset digital dalam sistem yang tidak tergantung pada pihak tertentu. *Smart contract* yang berjalan di jaringan *blockchain* dapat menjamin transparansi, keamanan, dan imutabilitas dari transaksi. Smart contract dapat digunakan dalam berbagai aplikasi, seperti sistem pemilihan elektronik, sistem keuangan, dan sistem logistik.



#### Gambar 2.2. Struktur *Smart Contract* [12]

*Smart contract* terdiri dari nilai, alamat, status dan fungsi. *Smart contract* membutuhkan sebuah transaksi sebagai *input*, mengeksekusi kode yang sesuai, dan memicu *output*. Sebuah transaksi *Smart contract* dapat dilacak dan tidak dapat diubah. Semua informasi tentang transaksi ada dalam *Smart contract* dilakukan secara otomatis. Karena sistemnya yang dilakukan secara otomatis penggunaan *smart contract* ini dapat menghilangkan pihak ketiga, *smart contract* sendiri dapat digunakan di berbagai bidang seperti [8] :

1. *Supply chain*. Dalam *supply chain*, *smart contract* digunakan untuk membuat sistem menjadi otomatis, aman dan juga transparan.
2. *Internet of Things* (IoT). Penerapan teknologi smart contract pada IoT yaitu untuk membuat sistem menjadi otomatis.
3. *Healthcare system*. Penerapan *smart contract* pada sistem kesehatan yaitu untuk menjaga privasi dari pasien dan juga membuat sistem yang lebih baik dan otomatis serta dengan *smart contract* manusia dapat menulis beberapa syarat dan ketentuan yang dapat diterapkan setelah data dikumpulkan.
4. *Digital right management*. Penerapan *smart contract* dalam manajemen hak cipta yaitu untuk memastikan royalti diterima oleh pembuat karya terkait dengan menggunakan hak kepemilikan dalam teknologi *blockchain*.
5. *Insurance*. Penerapan *smart contract* pada asuransi yaitu untuk membuat sistem yang lebih sederhana dan juga lebih transparan tanpa ada pihak ketiga.
6. *Financial system*. Dengan menggunakan teknologi *smart contract* pada sistem keuangan membuat pengguna dapat mentransfer uang melalui jaringan *peer to peer* tanpa keterlibatan ada pihak ketiga [12].

### **2.2.5. *Solidity***

*Solidity* adalah bahasa pemrograman yang digunakan untuk mengembangkan aplikasi berbasis *blockchain*. *Solidity* memungkinkan pengembang untuk mengembangkan aplikasi yang kompatibel dengan Ethereum, yang merupakan platform *blockchain* yang paling populer. *Solidity* adalah *high-level programming language* yang digunakan untuk menulis kontrak pintar (*smart contracts*) di *platform Ethereum*. *Smart contracts* adalah program komputer yang berjalan di atas *blockchain Ethereum* dan dirancang untuk secara otomatis mengeksekusi perjanjian atau kondisi yang telah diprogram saat kondisi tertentu terpenuhi. *Solidity* digunakan untuk mengembangkan *smart contract* yang dapat menyediakan fungsionalitas yang diperlukan untuk aplikasi *blockchain*, seperti menangani transaksi, menyimpan data, dan mengatur kebijakan. *Solidity* memiliki fitur-fitur yang unik, seperti *support* untuk *transient storage* (EIP-1153), *shard blob (binary large object) transactions* (EIP-4844), dan lain-lain [4].

### **2.2.6. *UML (Unified Modeling Language)***

*UML* atau*Unified Modeling Language*, adalah bahasa pemodelan standar yang digunakan untuk menggambarkan, merancang, dan mendokumentasikan sistem perangkat lunak dan non-perangkat lunak. Tujuannya adalah untuk menyediakan notasi yang terstandarisasi sehingga semua pihak yang terlibat dalam pengembangan perangkat lunak termasuk pengembang, analis sistem, dan pemangku kepentingan dapat berkomunikasi dengan lebih efektif. Fungsi dan kegunaan dari UML diantaranya visualisasi, perancangan, dokumentasi dan komunikasi. UML merupakan alat penting dalam pengembangan perangkat lunak modern. Dengan menyediakan notasi standar untuk berbagai aspek desain dan analisis, UML membantu memastikan bahwa semua orang yang terlibat memiliki pemahaman yang sama tentang arsitektur sistem, sehingga meningkatkan efisiensi dan efektivitas proses pengembangan perangkat lunak [19].

### **2.2.6. *Agile***

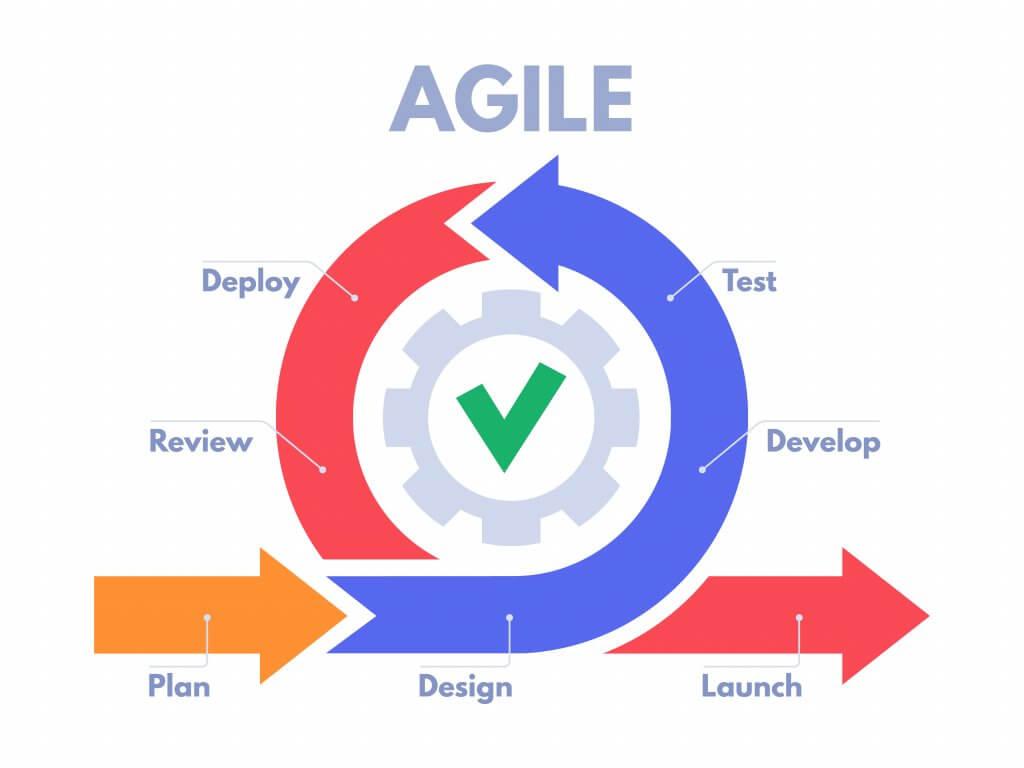
Metode *Agile* adalah sebuah pendekatan manajemen proyek yang digunakan dalam pengembangan perangkat lunak, yang berfokus pada iterasi, kolaborasi tim, dan fleksibilitas dalam merespon perubahan. Metode *Agile* memecah proyek besar menjadi serangkaian iterasi atau *sprint* yang lebih kecil dan terukur. Setiap *sprint* biasanya berlangsung selama dua hingga empat minggu. Manfaat yang didapatkan dari sisi developer adalah peningkatan produktivitas. Karena setiap tim dapat mengerjakan setiap tugas tanpa harus menunggu tim lain untuk menyelesaikan tugasnya. Selain itu, developer menjadi memiliki banyak waktu untuk berkembang sesuai dengan keinginan mereka. Dengan demikian, metode Agile merupakan pendekatan manajemen proyek yang efektif dalam pengembangan perangkat lunak, dengan fokus pada iterasi, kolaborasi tim, dan fleksibilitas dalam merespon perubahan [20].

### **2.2.6. *Black Box Testing***

*Black box testing* adalah metode pengujian perangkat lunak yang fokus pada evaluasi fungsionalitas sistem tanpa mengetahui struktur internal atau cara kerja dari sistem tersebut. Dalam pengujian ini, tester berinteraksi dengan aplikasi hanya melalui antarmuka pengguna, mengamati input dan output yang dihasilkan, tanpa akses ke *source code* atau logika internal. *Black box testing* dibuat untuk memvalidasi fungsional dari suatu sistem apakah sistem tersebut sudah sesuai dengan kebutuhan atau belum tanpa perlu mengetahui *source code* dari sistem tersebut. Black box testing berfokus pada tampilan dan pengujan fungsional sistem. Diawali dengan membuat test case pengujian dengan membagi masukan dan keluaran dari sebuah program [21].

BAB III METODE PENELITIAN

Metode penelitian yang digunakan dalam mengimplementasikan Sistem Pemilihan Elektronik Berbasis *Blockchain* yaitu metode pengembangan sistem yang bertujuan untuk merancang, mengembangkan, dan mengimplementasikan sistem pemilihan elektronik berbasis blockchain. Penelitian ini menggunakan metodologi *agile* untuk mengembangkan sistem *e-voting* berbasis *blockchain*. Model *Agile* dipilih karena sifatnya yang fleksibel, memungkinkan pengembangan sistem dilakukan dalam beberapa tahapan yang dapat disesuaikan.:



#### Gambar 3.1. Metode *Agile* [20]

## **3.1. *Plan***

Pada tahap ini, langkah-langkah strategis disusun untuk menentukan kebutuhan sistem dan mengidentifikasi fitur-fitur utama. Fokus utama pada tahap perencanaan adalah memahami kebutuhan pengguna, menentukan fitur-fitur utama dari sistem *e-voting* berbasis *blockchain*. Berikut merupakan beberapa elemen kunci pada tahap perencanaan:

* + - 1. Identifikasi Tujuan, Pada tahap awal perencanaan, bekerja sama dengan stakeholder untuk memahami dan mendefinisikan tujuan penelitian ini. Fokusnya adalah untuk menciptakan hasil yang dapat memenuhi kebutuhan pengguna akhir.
      2. Pembentukan *product backlog*, adalah daftar prioritas fitur atau tugas yang perlu diselesaikan dalam penelitian ini. Pada tahap ini, *backlog* dibuat berdasarkan input dari stakeholder dan pengguna.
      3. Penyusunan sprint plan, sprint Plan dibuat untuk menentukan apa yang dapat diselesaikan dalam waktu tertentu [20].

## **3.2. *Design***

Pada tahap desain, arsitektur sistem mulai dirancang secara rinci. Desain meliputi *frontend* yang akan digunakan oleh pemilih, *backend* untuk pengolahan data, serta desain *smart contract* di *blockchain*. Selain itu, alur proses pengguna akan disusun untuk memastikan sistem berjalan lancar dari pendaftaran pemilih hingga penghitungan suara. Berikut merupakan beberapa elemen kunci pada tahap desain:

* + - 1. Desain iteratif dan inkremental, desain dilakukan secara bertahap atau **inkremental**, di mana setiap iterasi (sprint) menghasilkan versi desain yang terus disempurnakan. Ini memastikan bahwa desain dapat beradaptasi dengan perubahan kebutuhan atau umpan balik dari pengguna setelah setiap iterasi.
      2. Agile sering melibatkan pembuatan prototipe cepat atau model minimum dari produk untuk memvalidasi ide desain dan memastikan bahwa sistem yang dirancang dapat memenuhi kebutuhan pengguna. Prototipe ini diuji pada pengguna untuk memeriksa fungsionalitas, kegunaan, dan efisiensi sistem.
      3. Desain dalam Agile cenderung bersifat modular dan fleksibel, sehingga memudahkan perubahan pada bagian tertentu tanpa harus merombak keseluruhan sistem. Ini juga memungkinkan pengembangan paralel dari berbagai komponen sistem.
      4. Umpan balik yang cepat dari pengujian dan stakeholder sangat penting dalam tahap desain Agile. Setiap iterasi desain diuji dan umpan baliknya digunakan untuk memperbaiki dan mengoptimalkan desain sebelum bergerak ke sprint berikutnya.
      5. Berbeda dengan pendekatan tradisional di mana desain selesai sebelum pengembangan dimulai, Agile mengintegrasikan desain dengan pengembangan. Desainer dan pengembang bekerja secara bersamaan, memastikan bahwa keputusan desain dapat diimplementasikan dengan cepat dan efisien.

## **3.3. *Develop***

Tahap pengembangan adalah implementasi desain ke dalam kode yang dapat dijalankan. Pengembangan dilakukan secara bertahap, dimulai dari fitur paling kritis seperti autentikasi pemilih, pencatatan suara, dan penghitungan hasil. dalam pendekatan *Agile*, pengembangan perangkat lunak dilakukan secara bertahap dan inkremental, di mana setiap *sprint* menghasilkan fitur yang berfungsi dan dapat digunakan atau diuji. Tujuan utama dari tahap ini adalah untuk memberikan hasil yang cepat dan bisa diterapkan, sambil tetap fleksibel untuk perubahan berdasarkan umpan balik pengguna dan stakeholder.

Pengembangan dilakukan dalam siklus pendek atau *sprint*. Setiap *sprint* berfokus pada pengembangan fitur atau modul tertentu. Fitur-fitur ini harus dapat berfungsi di akhir sprint sehingga hasil dapat segera diuji dan divalidasi.

Agile mendorong *Continuous Integration*, yaitu praktik menggabungkan kode baru yang ditulis oleh pengembang ke dalam repositori kode utama secara berkala. Setiap perubahan diuji secara otomatis untuk memastikan bahwa tidak ada masalah yang mengganggu fungsi fitur yang sudah ada.

Setiap iterasi atau *sprint* bertujuan untuk menghasilkan perangkat lunak yang bisa digunakan, meskipun belum lengkap secara keseluruhan. Setiap fitur yang selesai dapat langsung diuji oleh pengguna atau stakeholder untuk mendapatkan umpan balik, yang kemudian diterapkan pada iterasi berikutnya.

Agile mendorong penerimaan umpan balik yang cepat dari pengguna dan stakeholder selama proses pengembangan. Jika ada perubahan kebutuhan, tim pengembangan dapat segera menyesuaikan prioritas dalam sprint berikutnya. Hal ini memungkinkan fleksibilitas dalam proses pengembangan.

Pengujian dilakukan secara berkelanjutan sepanjang siklus pengembangan. Setiap fitur yang selesai langsung diuji untuk memastikan fungsionalitasnya sebelum melanjutkan ke fitur berikutnya. Pengujian ini mencakup pengujian unit, integrasi, dan pengujian otomatis.

## **3.4. *Test***

Pengujian dilakukan untuk memastikan bahwa setiap bagian dari sistem berfungsi dengan baik. dua metode pengujian utama akan digunakan: *Black Box Testing,* dan *Mock Election*.

### **3.4.1. *Black Box Testing***

*Black box testing* merupakan metode pengujian perangkat lunak yang berfokus pada fungsi-fungsi dari aplikasi tanpa memperhatikan struktur internal atau kode sumber. Pengujian ini bertujuan untuk memastikan bahwa setiap fungsi dalam sistem berjalan sesuai dengan spesifikasi dan kebutuhan pengguna. Pada sistem pemilihan elektronik (*e-voting*) berbasis *blockchain* ini, pengujian dilakukan dengan memeriksa apakah setiap fitur utama dalam aplikasi dapat berjalan sesuai yang diharapkan. Pengujian dilakukan berdasarkan berbagai skenario yang sudah dirancang. Berikut merupakan skenario yang akan diujikan:

##### Tabel 3.1 Skenario *Black box* Pemilih

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Skenario** | **Tujuan** | **Langkah Pengujian** | **Hasil yang Diharapkan** |
| 1 | **Login Pengguna** | Memastikan bahwa sistem memungkinkan pengguna untuk login dengan menggunakan kredensial yang valid. | 1. Masukkan username dan password yang valid. 2. Klik tombol "Login". | Pengguna berhasil login dan diarahkan ke halaman dashboard. |
| **No** | **Skenario** | **Tujuan** | **Langkah Pengujian** | **Hasil yang Diharapkan** |
| 2 | **Pemilihan Kandidat** | Memastikan bahwa sistem memungkinkan pemilih untuk memilih kandidat yang diinginkan. | 1. Login sebagai pemilih yang sudah terdaftar. 2. Pilih kandidat dari daftar kandidat yang tersedia. 3. Klik tombol "Submit". | Suara berhasil disimpan dan pesan konfirmasi ditampilkan. |
| 4 | **Hasil Pemilihan** | Memastikan bahwa sistem memungkinan pemilih untuk melihat hasil pemilihan | 1. Klik halaman "Hasil" di antarmuka pengguna 2. Lihat hasil pemilihan | Pengguna berhasil diarahkan menuju halaman hasil dan sistem memperlihatkan hasil pemilihan. |
| 5 | **Pemilihan Ganda** | Memastikan bahwa akan menolak pemilih menggunakan hak nya lebih dari satu kali. | 1. Login sebagai pemilih yang sudah terdaftar. 2. Pilih kandidat dari daftar kandidat yang tersedia. 3. Klik tombol "Submit". | Pemilih ditolak saat mencoba memberikan suara kedua kali |

##### Tabel 3.1 Skenario *Black box* Admin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Skenario** | **Tujuan** | **Langkah Uji** | **Output yang Diharapkan** |
| 1 | **Login Admin** | Memastikan bahwa sistem memungkinkan admin untuk login dengan menggunakan kredensial yang valid. | 1. Masukkan username dan password yang valid. 2. Klik tombol "Login". | Admin berhasil login dan diarahkan ke halaman dashboard. |
| 2 | **Pendaftaran Pemilih** | Memastikan bahwa sistem memungkinkan pemilih untuk mendaftar dengan informasi yang lengkap dan valid. | 1. Masukkan data diri lengkap pada form pendaftaran. 2. Klik tombol "Daftar". | Pesan konfirmasi bahwa pendaftaran berhasil dan data pemilih disimpan di database. |
| **No** | **Skenario** | **Tujuan** | **Langkah Uji** | **Output yang Diharapkan** |
| 3 | **Pendaftaran Kandidat** | Memastikan bahwa admin dapat mendaftarkan kandidat pemilihan menggunakan kredensial yang valid | 1. Klik menu "Kandidat", 2. Klik tombol "Tambah Kandidat", 3. Isi form dengan informasi kandidat seperti nama, tanggal lahir, dan deskripsi. | Admin berhasil menambahkan kandidat kedalam daftar kandidat |
| 4 | **Membuat Pemilihan** | Membuat pemilihan baru oleh admin | 1. Klik tombol "Buat Pemilihan Baru" pada antarmuka admin, 2. Isi form dengan informasi pemilihan seperti nama pemilihan, dan kandidat. 3. Klik tombol "Simpan" | Pemilihan baru berhasil dibuat dan muncul di daftar pemilihan |
| 5 | **Logout** | Memastikan bahwa sistem memungkinkan admin untuk logout. | 1. Klik tombol "Logout" di antarmuka pengguna, 2. Verifikasi bahwa pengguna diarahkan ke halaman login atau beranda. | Pengguna berhasil logout dan diarahkan ke halaman login atau beranda. |

### **3.4.2. *Mock Election***

*Mock election* dalam konteks *e-voting* adalah simulasi pemilihan yang dilakukan secara elektronik untuk menguji dan memastikan keamanan, efisiensi, dan akurasi sistem pemilihan elektronik. Tujuan utama dari *mock election* adalah untuk [22]:

Menguji keamanan, mengidentifikasi dan memperbaiki celah keamanan yang mungkin ada dalam sistem, seperti potensi serangan hacking atau manipulasi data.

Mengoptimalkan proses, menguji kecepatan dan efisiensi dalam proses pemilihan, penghitungan suara, dan pengiriman hasil.

Menguji penggunaan, menguji bagaimana sistem dapat digunakan oleh berbagai jenis pengguna, termasuk mereka yang tidak terbiasa dengan teknologi, seperti pemilih buta huruf atau lansia.

## **3.5. *Deploy***

Tahap *Deployment* adalah proses di mana produk yang telah selesai dikembangkan diimplementasikan ke lingkungan produksi atau disediakan untuk pengguna akhir. Dalam metodologi Agile, proses ini dilakukan secara iteratif setiap kali fitur baru berhasil dikembangkan. Pendekatan ini memungkinkan untuk menerima umpan balik langsung dari pengguna sejak tahap awal, sehingga penyesuaian atau perbaikan dapat dilakukan dengan cepat sesuai kebutuhan. Setelah pengujian selesai, sistem akan di-deploy ke lingkungan yang lebih besar, seperti *testnet Ethereum* atau jaringan *blockchain* lokal (seperti *Ganache*) untuk memastikan kelayakannya di skenario nyata. Pendekatan ini memastikan bahwa perangkat lunak dapat digunakan oleh pengguna akhir dengan cepat dan efisien, sambil tetap memberikan fleksibilitas untuk perubahan dan pembaruan di masa mendatang. *Smart contract* testing pada *e-voting* merujuk pada proses pengujian dan verifikasi kode *smart contract* yang digunakan dalam sistem pemungutan suara elektronik. Implementasi ini penting untuk memastikan bahwa *smart contract* berfungsi sebagaimana mestinya, aman, dan dapat diandalkan dalam mengelola proses pemilihan. Berikut adalah beberapa aspek penting dari *smart contract testing* dalam konteks *e-voting* [22]:

Keamanan, mengidentifikasi dan memperbaiki potensi kerentanan dalam kode *smart contract* untuk mencegah manipulasi data dan kecurangan selama pemilihan.

Fungsionalitas, memastikan bahwa semua fungsi yang diharapkan dari *smart contract* seperti pendaftaran pemilih, penghitungan suara, dan penyimpanan hasil berfungsi dengan baik dan sesuai dengan spesifikasi yang ditetapkan.

Akurasi, memverifikasi bahwa hasil pemungutan suara dihitung dengan benar dan tidak ada kesalahan dalam proses penghitungan.

Transparansi, menjamin bahwa semua transaksi yang dilakukan oleh *smart contract* dapat diaudit dan diverifikasi oleh pihak ketiga, sehingga meningkatkan kepercayaan publik terhadap sistem *e-voting*.

# **BAB IV**

**HASIL DAN PEMBAHASAN**

## **4.1. *Plan***

Tahap awal untuk memahami tujuan, masalah dan peluang yang perlu dipecahkan, masalah dan peluang yang perlu dipecahkan adalah pada fase perencanaan, Ini melibatkan pengumpulan informasi dari *stakeholder* untuk memahami persyaratan yang diinginkan, seperti keamanan, transparansi, dan efisiensi dalam proses pemilihan. Mengacu pada banyaknya permasalahan dalam pelaksanaan pemilihan tradisional pada latar belakang, maka perlu adanya analisis terhadap sistem pemilihan elektronik yang akan berjalan.

Analisis masalah

Dalam pembuatan sistem ini mengacu pada pelaksanaan pemilihan yang sudah ada saat ini terbagi menjadi dua, yaitu pemilihan tradisional dan pemilihan elektronik. Dalam pemilihan tradisional, berbagai masalah seperti kecurangan suara, kurangnya transparansi, dan tingginya biaya operasional masih menjadi tantangan besar. Hal ini berakibat pada rendahnya kepercayaan publik terhadap hasil pemilihan. Sistem manual yang digunakan sulit memastikan bahwa data pemilihan tersebut aman, akurat, transparan, otonom, dan anonim, karena manipulasi suara, *human error*, dan pengawasan yang terbatas.

Mendefinisikan peluang

Berdasarkan permasalahan yang diangkat, maka *blockchain* dapat menjadi solusi dalam pembuatan sistem pemilihan elektronik yang dapat membuat sistem pemilihan menjadi lebih aman, dengan sifat desentralisasi dan kriptografi, memberikan jaminan keamanan dan akurasi tinggi dalam mencatat suara. Menurut *Vitalik Buterin*, teknologi *blockchain* memastikan "integritas data yang kuat, karena setiap transaksi atau suara tercatat dalam blok yang tidak dapat diubah setelah dikonfirmasi di jaringan" (*Buterin*, 2017). Dengan demikian, tidak ada aktor tunggal yang bisa memanipulasi hasil pemilihan, memastikan data pemilihan elektronik yang aman dan akurat. transparan, setiap suara yang tercatat di *blockchain* dapat diverifikasi oleh publik, memastikan transparansi penuh.

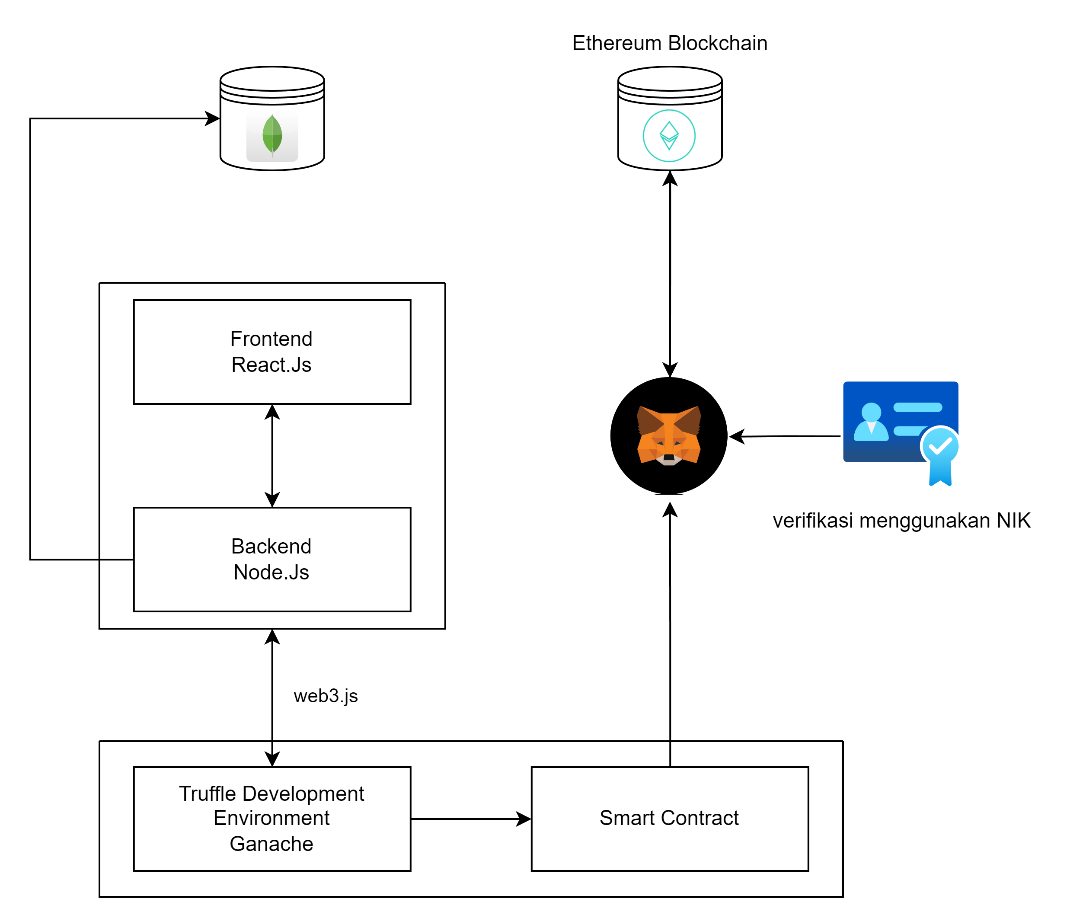
Dengan pendekatan ini, penggunaan *blockchain* dalam pemilihan elektronik mampu memenuhi tujuan penelitian, yaitu membuat sistem pemilihan elektronik (*e-voting*) yang aman dan transparan, berbasis *blockchain*. Pandangan para ahli mendukung implementasi *blockchain* sebagai teknologi yang mampu mengatasi berbagai masalah yang dihadapi dalam pemilihan tradisional, serta meningkatkan kepercayaan publik terhadap sistem pemilihan yang baru.

## **4.2. *Design***

Pada tahap desain, arsitektur sistem mulai dirancang secara rinci. Desain meliputi *frontend* yang akan digunakan oleh pemilih, *backend* untuk pengolahan data, serta desain *smart contract* di *blockchain*.

### **4.2.1. Arsitektur Sistem**

Arsitektur Sistem Pemilihan Elektronik Berbasis *Blockchain* terdiri dari beberapa lapisan yang saling berinteraksi untuk menyediakan proses pemilihan yang aman dan transparan. Pada lapisan antarmuka pengguna (*frontend*), pemilih dan admin dapat berinteraksi dengan sistem melalui web yang dibangun menggunakan *HTML*, *CSS*, dan *React.js*. Pemilih menggunakan *crypto wallet* seperti *MetaMask* untuk mendaftar, mengautentikasi diri, dan memberikan suara, pada saat pemilihan pemilih akan menggukanan NIK untuk memilih, sehingga dapat mencegah terhadinya suara ganda. *Backend*, yang dibangun menggunakan *Node.js*, mengelola logika bisnis dan berkomunikasi dengan *frontend* melalui *middleware*. *Backend* juga berinteraksi dengan *smart contract* di *blockchain* menggunakan *library* *web3.js*. *Smart contract* yang ditulis dalam *Solidity* mengelola pendaftaran pemilih, pemungutan suara, dan penghitungan hasil pemilihan. Semua transaksi dicatat di *blockchain* *Ethereum*, memastikan keamanan dan transparansi. *Database* tradisional dapat digunakan untuk menyimpan data yang tidak sensitif. Berikut adalah arsitektur aplikasi Pemilihan Elektronik Berbasis *Blockchain* menggunakan *Ethereum* menggunakan *Solidity*:



#### Gambar 4.1. Arsitektur Sistem

* + - 1. *Frontend*, menyediakan antarmuka bagi pengguna untuk berinteraksi dengan sistem *e-voting*. *React.js* digunakan sebagai *framework* untuk pengembangan aplikasi web yang dinamis dan cepat, sementara *HTML* dan *CSS* digunakan untuk membangun struktur dan desain visual aplikasi.
      2. *Backend*, *Node.js* digunakan untuk mengelola logika server, termasuk autentikasi pengguna, validasi input, dan interaksi dengan *database.*
      3. *Smart contract*, *Smart contract* yang ditulis dalam *Solidity* bertindak sebagai logika inti untuk *e-voting*, memastikan bahwa semua aturan pemilihan (seperti validasi suara dan pencatatan suara) dijalankan secara otomatis dan transparan di jaringan *blockchain* *ethereum*.
      4. *Ethereum Blockchain*, berfungsi sebagai dasar infrastruktur tempat *smart contract* dieksekusi. Semua transaksi pemilihan, termasuk pendaftaran suara dan pencatatan hasil, dilakukan melalui Ethereum untuk memastikan transparansi dan keamanan.
      5. *Crypto Wallet,* pemilih menggunakan *crypto wallet* seperti *MetaMask* yang memungkinkan pengguna untuk berinteraksi dengan aplikasi berbasis *blockchain* melalui *browser*. Dalam arsitektur ini, *Metamask* digunakan untuk mengelola identitas pengguna, memungkinkan mereka untuk melakukan transaksi dan memberikan suara dalam *e-voting* melalui jaringan *blockchain*.
      6. Verifikasi identitas pemilih menggunakan NIK, sehingga tidak akan ada suara ganda.
      7. *MongoDB* digunakan sebagai database *NoSQL* untuk menyimpan data yang tidak sensitif, seperti daftar kandidat, data pengguna (tanpa informasi pribadi yang terkait dengan identitas *blockchain*), dan *metadata* terkait pemilihan.
      8. *Truffle Development Environment, truffle* mempermudah penulisan dan kompilasi *smart contract* yang ditulis dalam *Solidity*. Truffle menangani proses kompilasi kontrak menjadi *bytecode* yang siap di-*deploy* ke *blockchain*. *Truffle* juga digunakan untuk menguji kontrak di *blockchain* lokal, seperti *Ganache*.
      9. *Ganache,* berfungsi sebagai blockchain lokal yang dapat digunakan oleh pengembang untuk melakukan pengujian dan simulasi *smart* *contract* dalam lingkungan yang aman dan terkendali tanpa harus menggunakan jaringan *blockchain* publik.

### **4.2.2. *Use Case Diagram***

*Use case diagram* adalah cara umum untuk mengomunikasikan fungsi-fungsi utama sistem perangkat lunak. Use-case diagram yang paling sederhana adalah representasi interaksi pengguna dengan sistem yang menunjukkan hubungan antara pengguna dan berbagai kasus penggunaan di mana pengguna terlibat. *Use-case diagram* dapat mengidentifikasi berbagai jenis pengguna sistem dan kasus penggunaan yang berbeda dan sering kali disertai dengan jenis diagram lainnya juga. *Use-case diagram* tidak lain adalah fungsi sistem yang ditulis dengan cara yang terorganisir. Hal lain yang relevan dengan use case adalah aktor. Aktor dapat didefinisikan sebagai sesuatu yang berinteraksi dengan sistem.

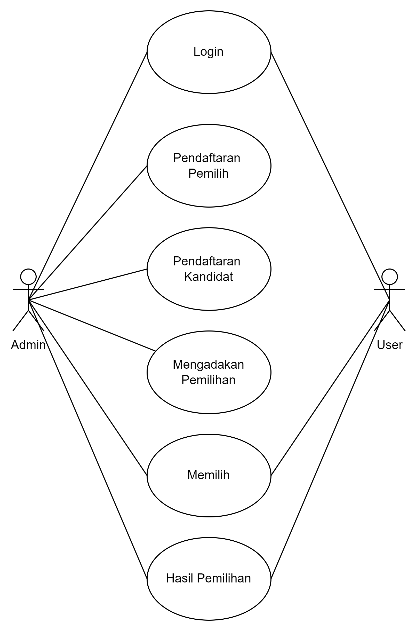
Identifikasi aktor

Pada sistem pemilihan elektronik (*e-voting*) berbasis *blockchain*. Setiap aktor akan di deskripsikan pada tabel 4.1.

##### Tabel 4.1 Identifikasi aktor

|  |  |  |
| --- | --- | --- |
| No | Aktor | Deskripsi |
| 1 | Admin | Pihak yang bertanggung jawab untuk mengelola sistem e-voting, termasuk mengelola data pemilih, mengatur pemilihan, dan memverifikasi identitas pemilih. |
| 2 | Pemilih | Warga Desa yang terdaftar untuk mengikuti pemilihan dan memberikan suaranya melalui sistem *e-voting.* |

Use case diagram



#### Gambar 4.*2 Use Case Diagram*

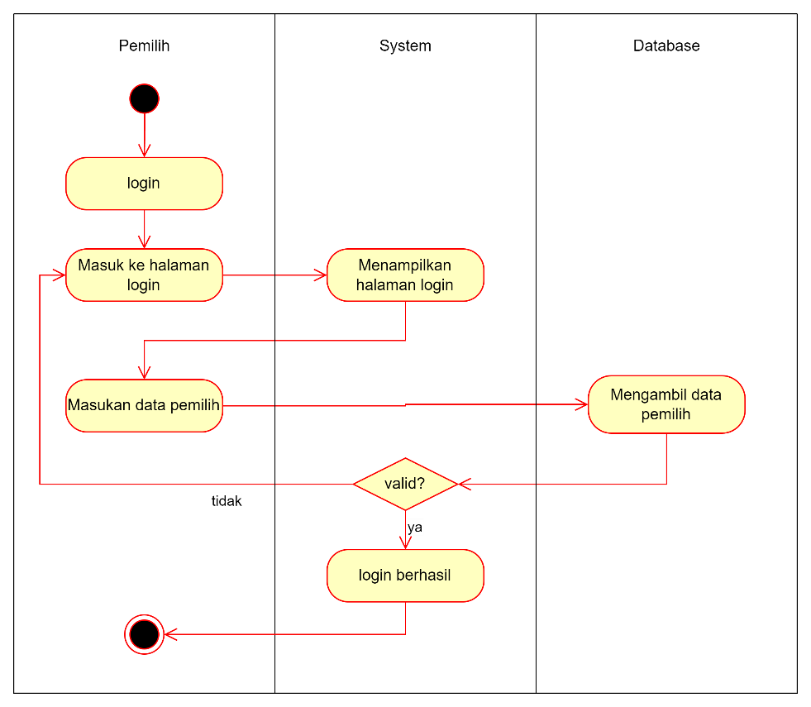
*Use case diagram* pada gambar 4.2 menggambarkan proses dalam sistem *e-voting* yang mencakup empat aktivitas utama: login, mengadakan pemilihan, memilih, dan hasil pemilihan. Tahap pertama adalah login, di mana pengguna (pemilih atau admin) mengakses sistem dengan autentikasi. Kemudian, admin menggunakan fitur mengadakan pemilihan untuk menyiapkan dan menyusun acara pemilihan, termasuk pengaturan kandidat dan jadwal. Selanjutnya, pada tahap memilih, pemilih yang telah terdaftar dapat memberikan suara mereka kepada kandidat yang diinginkan, dan sistem mencatat suara tersebut dengan aman. Terakhir, hasil pemilihan memungkinkan sistem menampilkan hasil pemilihan yang transparan dan akurat. Semua aktivitas ini dirancang untuk memastikan keamanan, transparansi, dan efisiensi dalam proses pemilihan elektronik berbasis blockchain.

### **4.2.3. *Activity Diagram***

*Activity diagram* adalah salah satu jenis diagram yang digunakan dalam UML untuk memodelkan aliran kerja atau proses bisnis. Diagram ini menggambarkan urutan aktivitas, tindakan, dan aliran keputusan dalam suatu sistem atau proses. *Activity* *diagram* sangat berguna untuk menggambarkan proses dinamis dalam sistem, seperti aliran kontrol dari satu aktivitas ke aktivitas lain dan bagaimana keputusan diambil di sepanjang proses.

*Activity diagram login*

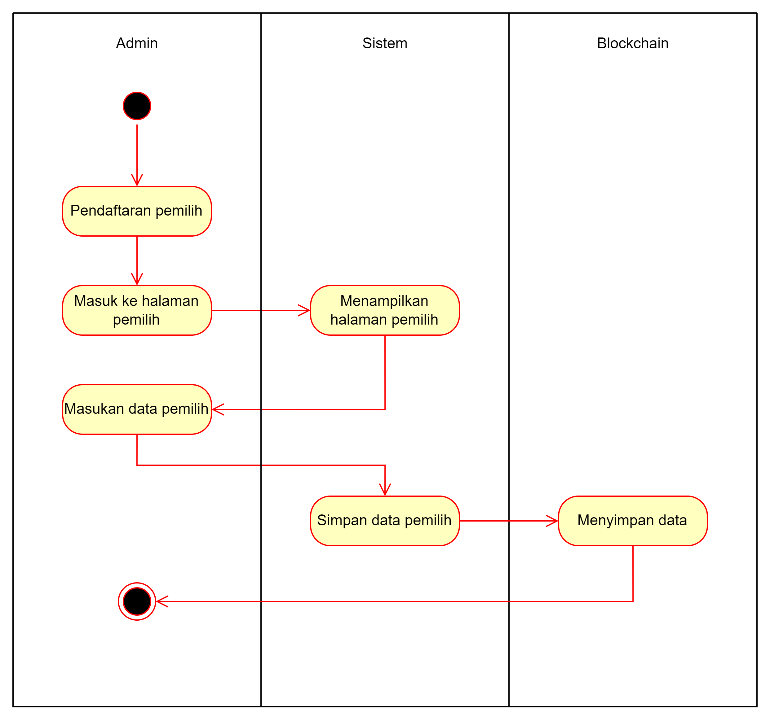
Aktivitas diagram pada Gambar 4.3 menjelaskan aktivitas yang dilakukan oleh pemilih untuk *login*, dimulai dengan masuk halaman *login* dan sistem akan menampilkan antarmukanya. Lalu pemilih akan mengisi data seperti NIK dan *password* untuk dicocokan dalam *database* melakukan validasi.



#### Gambar 4.3 *Activity diagram login*

1. *Activity diagram* pendaftaran pemilih

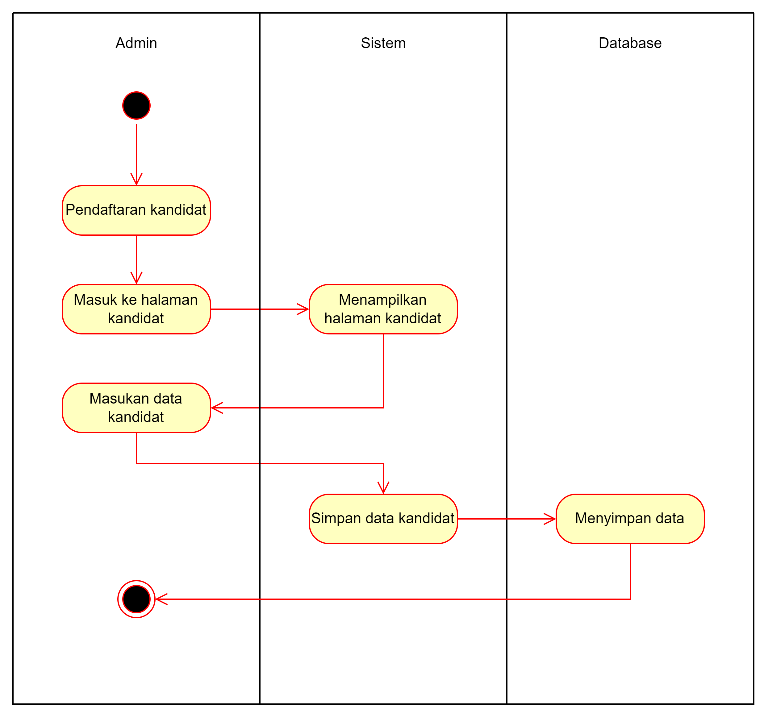
Pada Gambar 4.4 menjelaskan aktivitas yang dilakukan oleh admin untuk melakukan pendaftaran pemilih, dimulai dengan masuk ke halaman pendaftaran pemlih lalu admin akan mengisi data seperti NIK, nama, alamat, tanggal lahir dan *password.*



#### Gambar 4.4 *Activity diagram* pendaftaran pemilih

1. *Activity diagram* pendaftaran kandidat

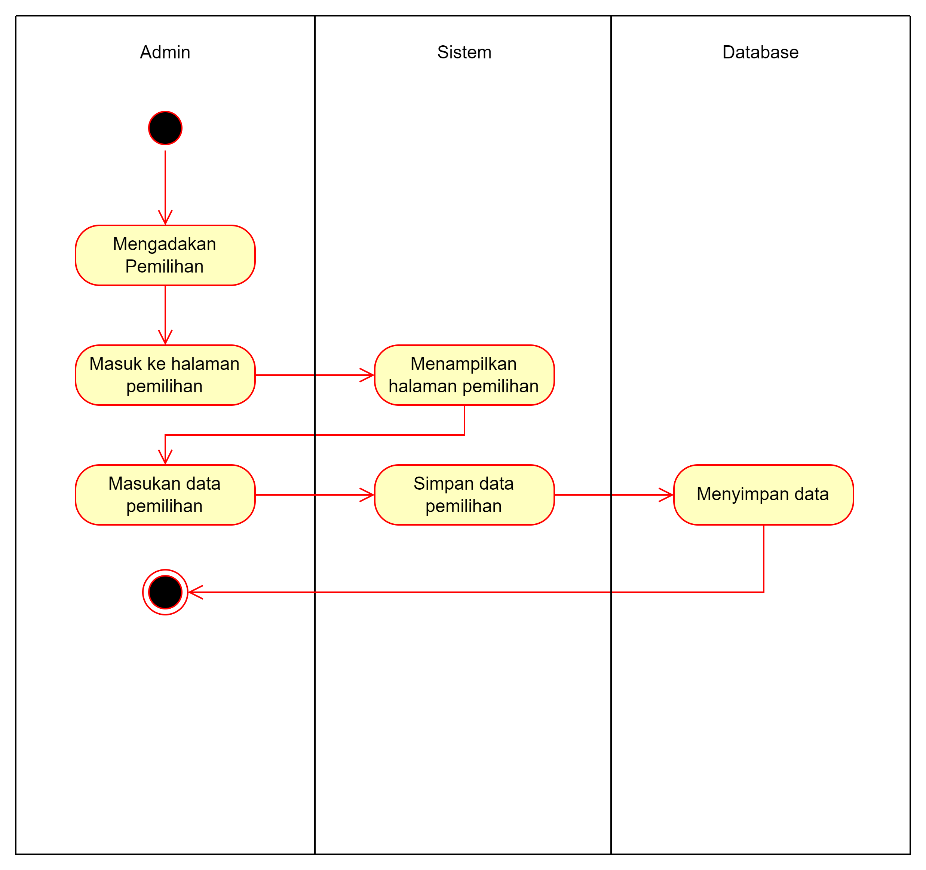
Gambar 4.5 menjelaskan aktivitas yang dilakukan oleh admin untuk melakukan pendaftaran kandidat, dimulai dengan masuk ke halaman pendaftaran kandidat lalu admin akan mengisi data seperti username, nama, alamat, tanggal lahir, tahun masuk politik, pendidikan, dan deskripsi untuk kandidat*.*



#### Gambar 4.5 *Activity diagram* pendaftaran kandidat

1. *Activity diagram* mengadakan pemilihan

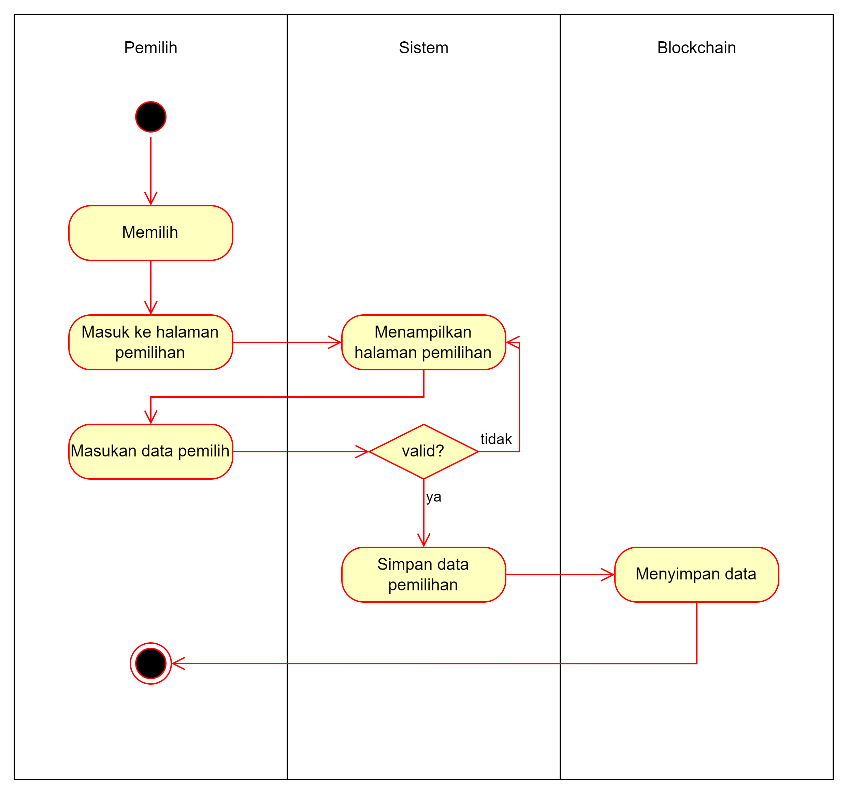
Aktivitas diagram pada Gambar 4.5 menjelaskan aktivitas yang dilakukan oleh admin untuk melakukan pembuatan pemilihan, dimulai dengan masuk halaman buat pemilihan dan sistem akan menampilkan antarmukanya. Lalu admin akan mengisi data seperti judul dan nama kandidat untuk disimpan dalam *database* dan *blockchain*.



#### Gambar 4.6 *Activity diagram* mengadakan pemilihan

*Activity diagram* pemilihan

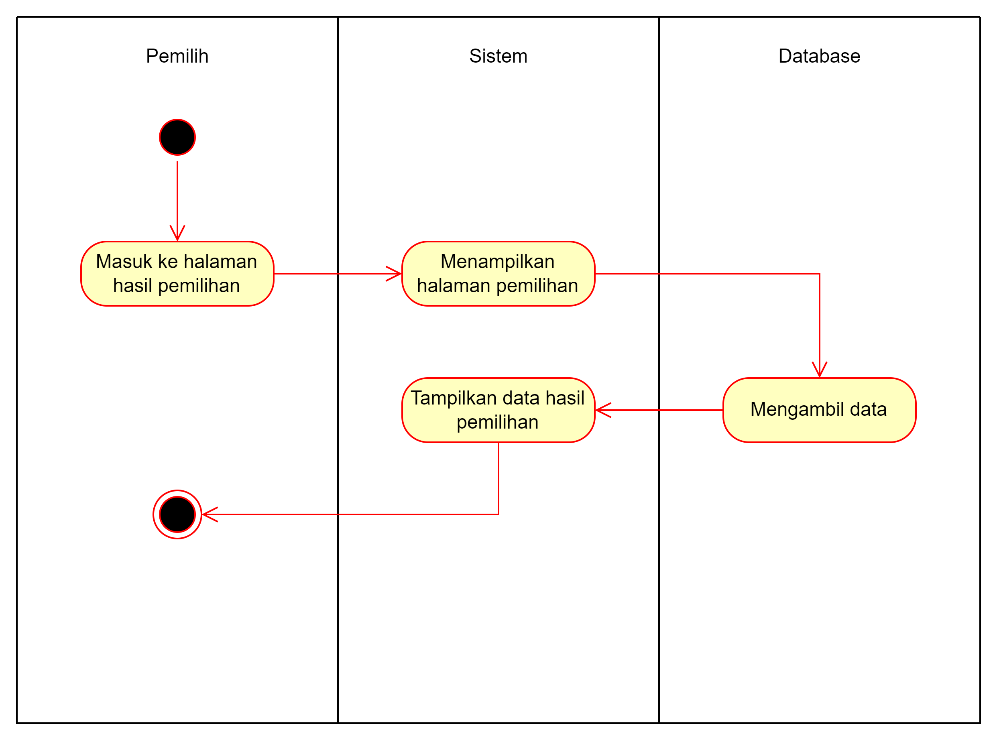
Aktivitas diagram pada Gambar 4.6 menjelaskan aktivitas yang dilakukan oleh pemilih untuk melakukan *voting* mulai dari *login* hingga mendapat umpan balik dari request yang dikirim. pemilih melakukan *login* dengan cara memasukkan NIK dan *password*. Lalu setelah mengisi data kredensial sistem akan melakukan validasi, Jika data valid sistem akan melanjutkan transaksi dengan *metamask*. Hasil voting dimasukkan dalam *database* dan *blockchain.* Pemilih belum bisa melihat hasilnya agar pemilih dapat menjaga kerahasiaannya.



#### Gambar 4.7 *Activity diagram* pemilihan

*Activity diagram* hasil pemilihan

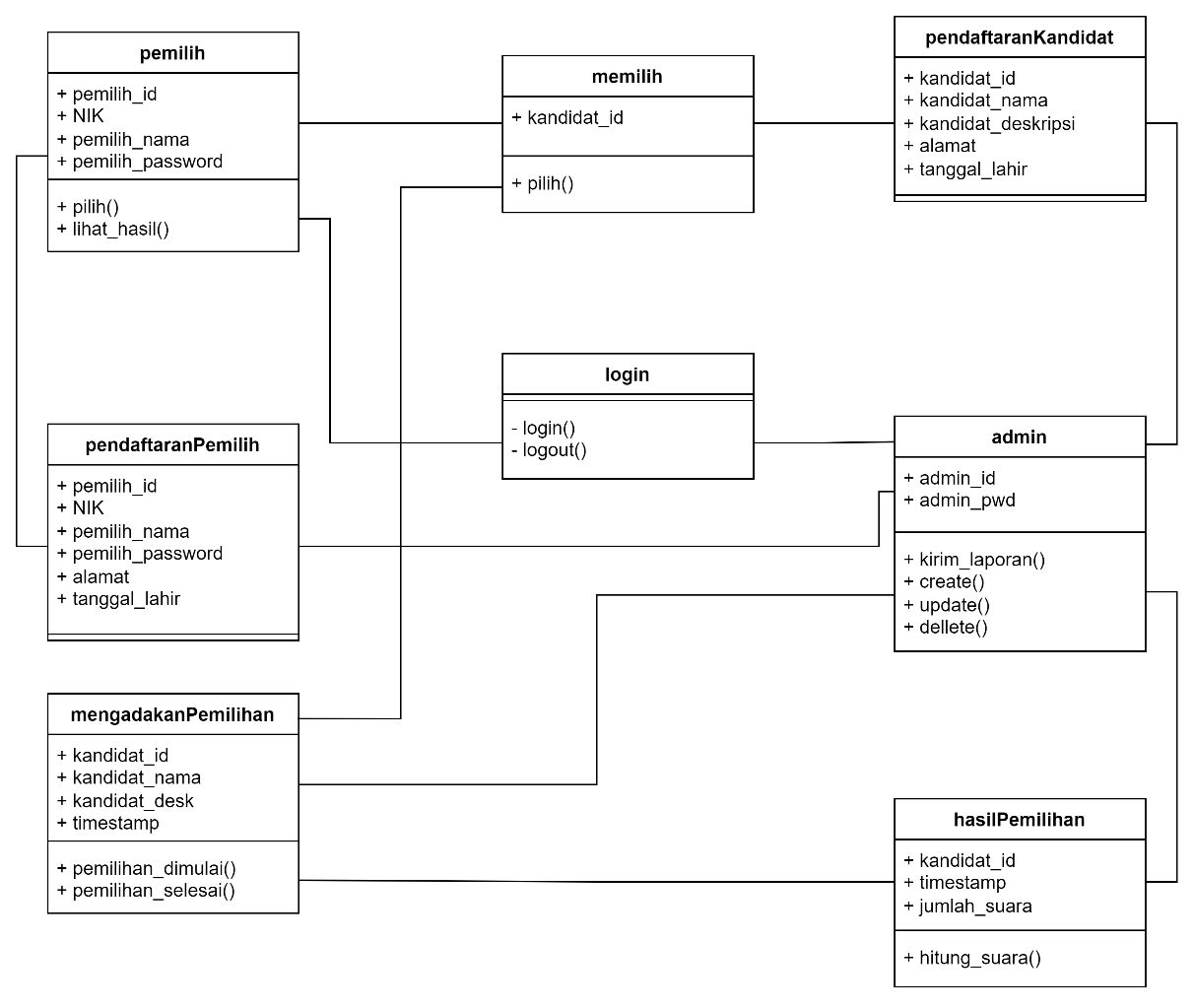
Aktivitas diagram pada Gambar 4.7 menjelaskan aktivitas yang dilakukan oleh *admin* untuk melakukan pembuatan pemilihan, dimulai dengan masuk halaman buat pemilihan dan sistem akan menampilkan antarmukanya. Lalu *admin* akan mengisi data seperti judul dan nama kandidat untuk disimpan dalam database dan *blockchain*.



#### Gambar 4.8 *Activity diagram* hasil pemilihan

### **4.2.4. *Class Diagram***

*Class diagram* atau diagram kelas adalah salah satu jenis diagram struktur pada *UML* yang menggambarkan dengan jelas struktur serta deskripsi *class*, atribut, metode, dan hubungan dari setiap objek. *Class diagram* ini bersifat statis, dalam artian diagram ini bukan menjelaskan apa yang terjadi jika kelas-kelasnya berhubungan, melainkan menjelaskan hubungan apa yang terjadi. *Class diagram* ini memiliki beberapa fungsi, fungsi utamanya yaitu menggambarkan struktur dari sebuah sistem.



#### Gambar 4.9 *Class Diagram*

Berikut ini penjelasan *class diagram* pada gambar 4.9.

##### Tabel 4.2 Deskripsi class diagram pemilih

|  |  |  |
| --- | --- | --- |
| Nama | Pemilih | |
| Deskripsi | Merepresentasikan pemilih dalam sistem *e-voting*. | |
| Atribut | + pemilih\_id  + NIK  + pemilih\_nama  + pemilih\_password | |
| Methods | Pilih() | Method untuk melakukan pemilihan oleh pemilih. |
| Lihat\_hasil() | Method untuk melihat hasil pemilihan. |

##### Tabel 4.3 Deskripsi class diagram admin

|  |  |  |
| --- | --- | --- |
| Nama | Admin | |
| Deskripsi | Mengelola pemilihan dan kandidat. | |
| Atribut | + admin\_id  + admin\_password | |
| Methods | Kirim\_laporan() | Mengirimkan laporan hasil pemilihan. |
|  | Create() | Membuat atau menambahkan pemilih, kandidat dan pemilihan. |
|  | Update() | Memperbaharui data pemilih, kandidat dan pemilihan. |
|  | Delete() | Menghapus pemilih, kandidat dan pemilihan. |

##### Tabel 4.4 Deskripsi class diagram

|  |  |  |
| --- | --- | --- |
| Nama | mengadakanPemilihan | |
| Deskripsi | Merepresentasikan pengelolaan pemilihan yang dilakukan oleh admin. | |
| Atribut | + kandidat\_id  + kandidat\_nama  + kandidat\_desk  + timestamp | |
| Methods | Pemilihan\_dimulai() | Method untuk memulai pemilihan. |
|  | Pemilihan\_selesa() | Method untuk mengakhiri pemilihan. |

##### Tabel 4.5 Deskripsi class diagram pendaftaran kandidat

|  |  |  |
| --- | --- | --- |
| Nama | pendaftaranKandidat | |
| Deskripsi | Menyimpan data kandidat yang akan dipilih dalam pemilihan. | |
| Atribut | + kandidat\_id  + kandidat\_nama  + kandidat\_deskripsi  + alamat  + tanggal\_lahir | |
| Methods | - | - |

##### Tabel 4.6 Deskripsi class diagram pendaftaran pemilih

|  |  |  |
| --- | --- | --- |
| Nama | pendaftaranPemilih | |
| Deskripsi | Kelas untuk menyimpan data pemilih yang sudah terdaftar. | |
| Atribut | + pemilih\_id  + NIK  + pemilih\_nama  + pemilih\_password  + alamat  + tanggal\_lahir | |
| Methods |  |  |

##### Tabel 4.7 Deskripsi class diagram memilih

|  |  |  |
| --- | --- | --- |
| Nama | memilih | |
| Deskripsi | Kelas untuk pemilih melakukan proses pemilihan | |
| Atribut | + kandidat\_id | |
| Methods | Pilih() | Method untuk pemilih bisa melakukan pemilihan. |

##### Tabel 4.8 Deskripsi class diagram hasil pemilihan

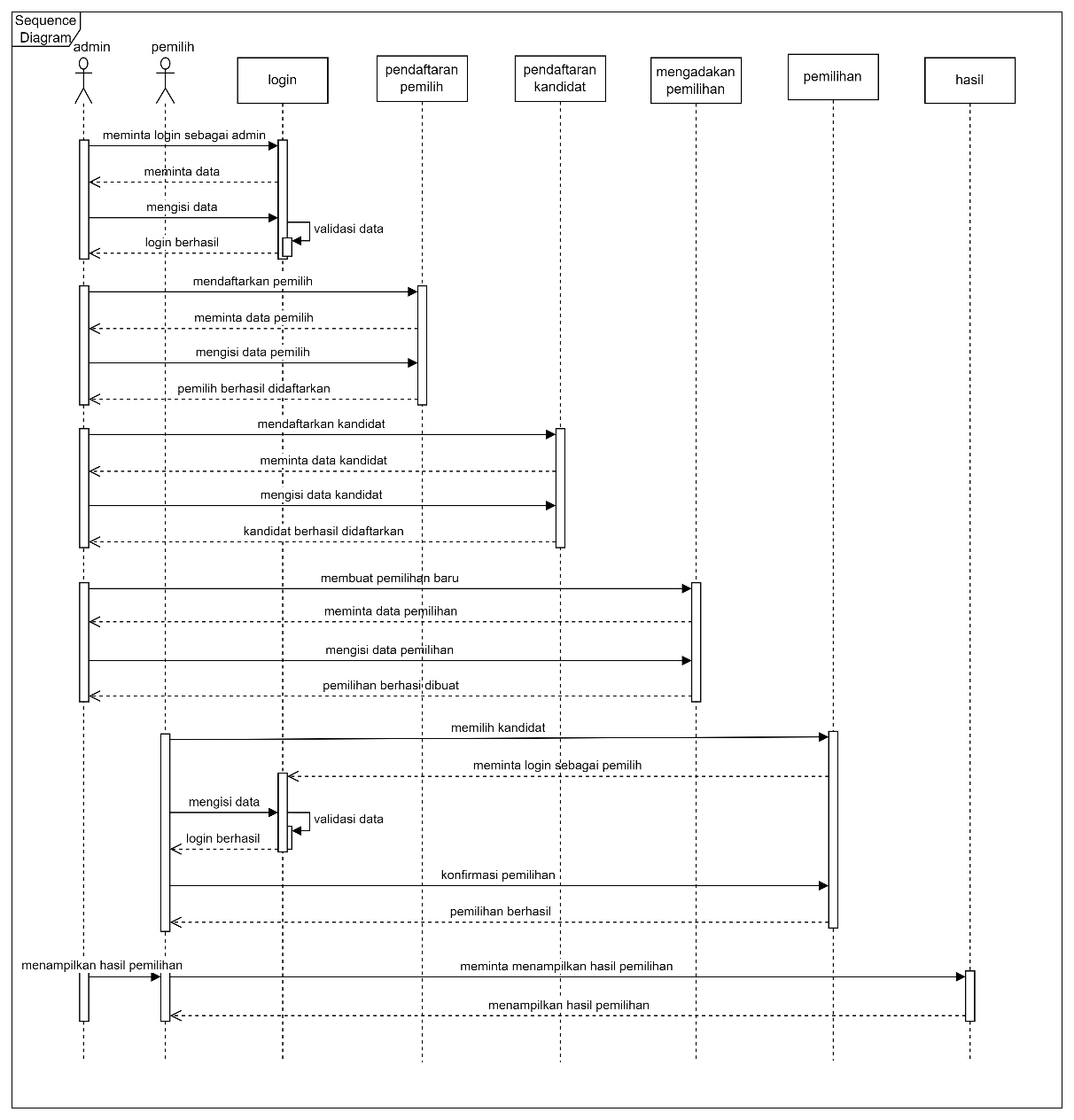
|  |  |  |
| --- | --- | --- |
| Nama | hasilPemilihan | |
| Deskripsi | Kelas untuk menyimpan hasil dari proses pemilihan. | |
| Atribut | + kandidat\_id  + timestamp  + jumlah\_suara | |
| Methods | Terima\_laporan() | Menerima laporan hasil pemilihan. |
|  | Hitung\_suara() | Menghitung jumlah suara untuk menentukan hasil. |

##### Tabel 4.9 Deskripsi class diagram login

|  |  |  |
| --- | --- | --- |
| Nama | login | |
| Deskripsi | Kelas ini menangani proses autentikasi bagi pengguna, baik pemilih maupun admin, untuk mengakses sistem e-voting. | |
| Atribut | - | |
| Methods | Login() | Autentikasi pengguna untuk mengakses sistem e-voting. |
|  | Logout() | Mengakhiri sesi login dan mengeluarkan pengguna dari sistem. |

### **4.2.4. *Sequence Diagram***

*Sequence diagram* atau diagram urutan adalah sebuah diagram yang digunakan untuk menjelaskan dan menampilkan interaksi antar objek-objek dalam sebuah sistem secara terperinci. Selain itu *sequence diagram* juga akan menampilkan pesan atau perintah yang dikirim, beserta waktu pelaksanaannya. Tujuan utama dari pembuatan *sequence diagram* adalah untuk mengetahui urutan kejadian yang dapat menghasilkan output yang diinginkan. Selain itu, tujuan dari diagram urutan ini mirip dengan *activity diagram*, seperti menggambarkan alur kerja dari sebuah aktivitas, serta dapat menggambarkan aliran data dengan lebih detail, termasuk data atau perilaku yang diterima atau dikirimkan.



#### Gambar 4.10 *Sequence Diagram*

Sequence diagram ini menunjukkan alur proses dalam sistem e-voting mulai dari login sebagai admin hingga penampilan hasil pemilihan.

1. Login sebagai Admin: Admin memulai dengan meminta login, kemudian memasukkan data untuk validasi. Setelah data valid, admin berhasil login.
2. Pendaftaran Pemilih: Admin mendaftarkan pemilih dengan mengirim permintaan data pemilih, mengisi data tersebut, dan kemudian memvalidasinya hingga pemilih berhasil didaftarkan.
3. Pendaftaran Kandidat: Proses pendaftaran kandidat juga dimulai dengan permintaan data kandidat, pengisian, dan validasi data. Setelah data kandidat berhasil divalidasi, kandidat dinyatakan berhasil terdaftar.
4. Pembuatan Pemilihan: Admin membuat pemilihan baru dengan mengisi data pemilihan yang diperlukan. Setelah data pemilihan divalidasi, pemilihan dinyatakan berhasil dibuat.
5. Login sebagai Pemilih: Pemilih melakukan login dengan mengisi data untuk divalidasi. Jika validasi berhasil, pemilih berhasil login ke sistem.
6. Memilih Kandidat: Pemilih memilih kandidat yang tersedia dan mengonfirmasi pilihannya. Setelah konfirmasi, pilihan pemilih dinyatakan berhasil.
7. Menampilkan Hasil Pemilihan: Admin atau pemilih dapat meminta untuk menampilkan hasil pemilihan. Sistem mengambil data hasil pemilihan dan menampilkannya kepada pengguna.

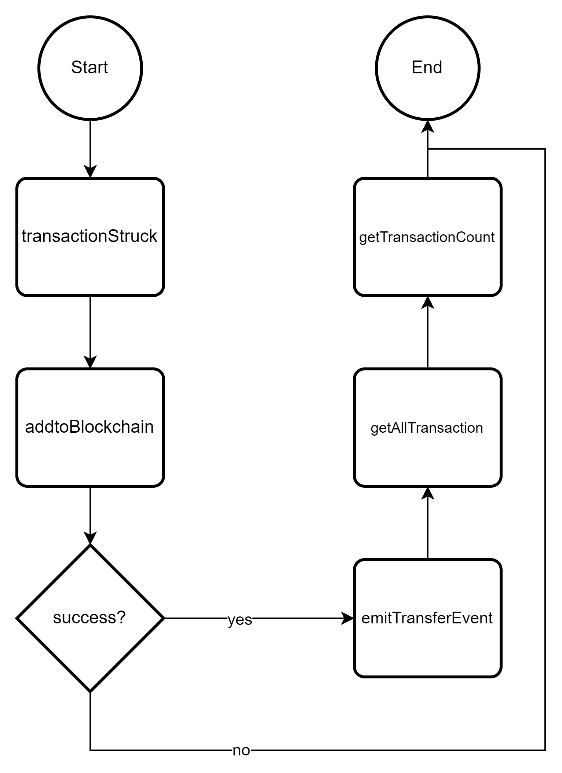
Diagram ini memperlihatkan bagaimana sistem *e-voting* memastikan alur kerja yang terstruktur dengan validasi di setiap tahap, mulai dari login hingga penyelesaian pemilihan dan penampilan hasil.

## **4.3. *Develop***

Tahap pengembangan adalah implementasi desain ke dalam kode yang dapat dijalankan. Pengembangan dilakukan secara bertahap, dimulai dari fitur paling kritis seperti autentikasi pemilih, pencatatan suara, dan penghitungan hasil.

### **4.3.1. *Smart Contract***

*Smart contract* adalah transaksi yang berisi kode yang akan dieksekusi otomatis ketika kondisi dan persyaratan tertentu terpenuhi. *Smart contract* tidak membutuhkan intermediari seperti pengguna hukum atau pihak pengawas untuk menjamin perjanjian. *Smart* *contract* dapat mengatur transaksi, mengatur proses, dan mengelola aset digital dalam sistem yang tidak tergantung pada pihak tertentu. *Smart contract* ini ditulis dalam berbagai macam bahasa pemrograman, diantaranya *solidity.* *Solidity* memungkinkan pengembang untuk mengembangkan aplikasi yang kompatibel dengan *ethereum*. Setiap komputer dalam jaringan (atau “*node*”) menyimpan salinan semua *smart contract* yang ada dan statusnya saat ini bersama dengan *blockchain* dan data transaksi. Ketika *smart contract* menerima data dari pengguna, kodenya dieksekusi oleh semua *node* dalam dalam jaringan untuk mencapai konsensus tentang hasil dan aliran nilai yang dihasilkan. Inilah yang memungkinkan *smart contract* berjalan dengan aman tanpa otoritas pusat, bahkan ketika pengguna melakukan transaksi keuangan yang kompleks dengan entitas yang tidak dikenal. Untuk mengeksekusi *smart contract* di jaringan *Ethereum*, biasanya harus membayar biaya yang disebut “*gas*” (dinamakan demikian karena biaya ini membuat *blockchain* tetap berjalan).



#### Gambar 4.11 *Smart Contract*

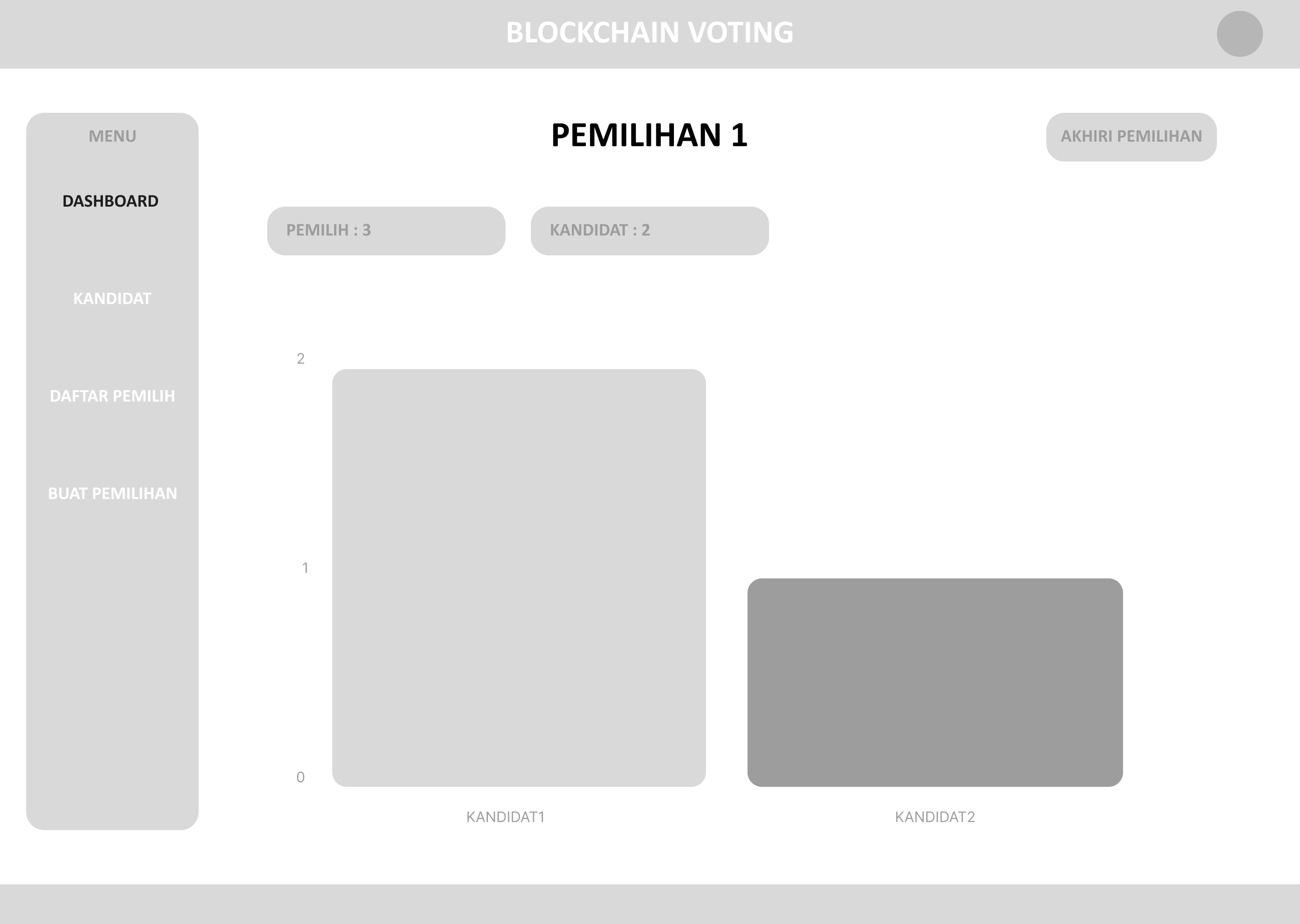
Berdasarkan gambar 4.10 Titik awal dari proses. Pada langkah ini, sistem siap menerima input untuk memulai proses transaksi. *transactionStruct* ini adalah langkah di mana transaksi (*voting*) didefinisikan dan siap untuk dicatat. Di sini, data seperti *receiver*, *user\_id*, *election\_id*, dan *candidate\_id* disiapkan. *addtoBlockchain* pada langkah ini, sistem menambahkan transaksi baru ke dalam *blockchain*, memperbarui *transactionCounter*, dan menyimpan transaksi ke dalam array *transactions*. Selanjutnya ada *decision* *node* yang menentukan apakah transaksi berhasil atau tidak, Jika transaksi berhasil, *eventTransfer* akan dipancarkan untuk memberitahu jaringan bahwa transaksi baru telah terjadi, mencatat rincian transaksi yang sudah disimpan., namun jiga gagal proses tidak akan dilanjutkan, dan transaksi akan berhakhir. Setelah transaksi dicatat, pengguna dapat memanggil fungsi *getAllTransaction* untuk mendapatkan semua transaksi yang tersimpan. Pengguna juga dapat memanggil fungsi *getTransactionCount* untuk mengetahui jumlah total transaksi yang telah dilakukan.

### **4.3.2. Pengembangan Antarmuka**

Pengembangan antarmuka pengguna merupakan tahap penting dalam proses pembangunan sistem *e-voting* berbasis *blockchain*. Antarmuka pengguna yang baik harus sederhana, intuitif, dan mudah digunakan oleh pengguna dengan berbagai tingkat pemahaman teknologi, pembuatan antarmuka ini menggunakan software figma.

1. Halaman Admin-Dashboard

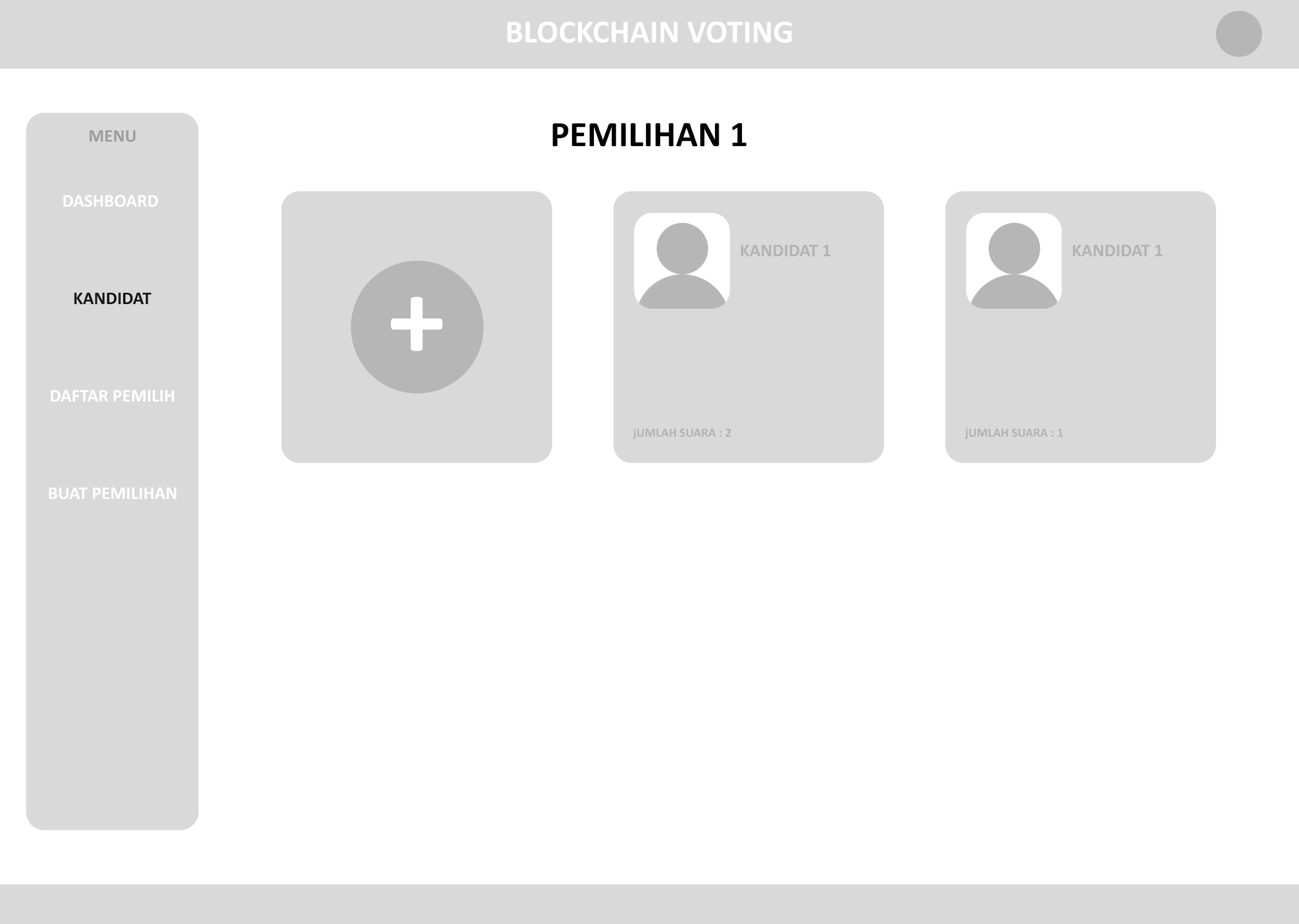
Pada Gambar 4.11 merupakan halaman *dashboard* untuk admin, terdapat informasi dasar berupa jumlah pemilih, dan jumlah kandidat. Terdapat juga navigasi untuk pengelolaan pemilih, kandidat, dan pemilihan.



#### Gambar 4.11 Rancangan Antarmuka *Admin-Dashboard*

1. Halaman Admin*-*Pemilihan

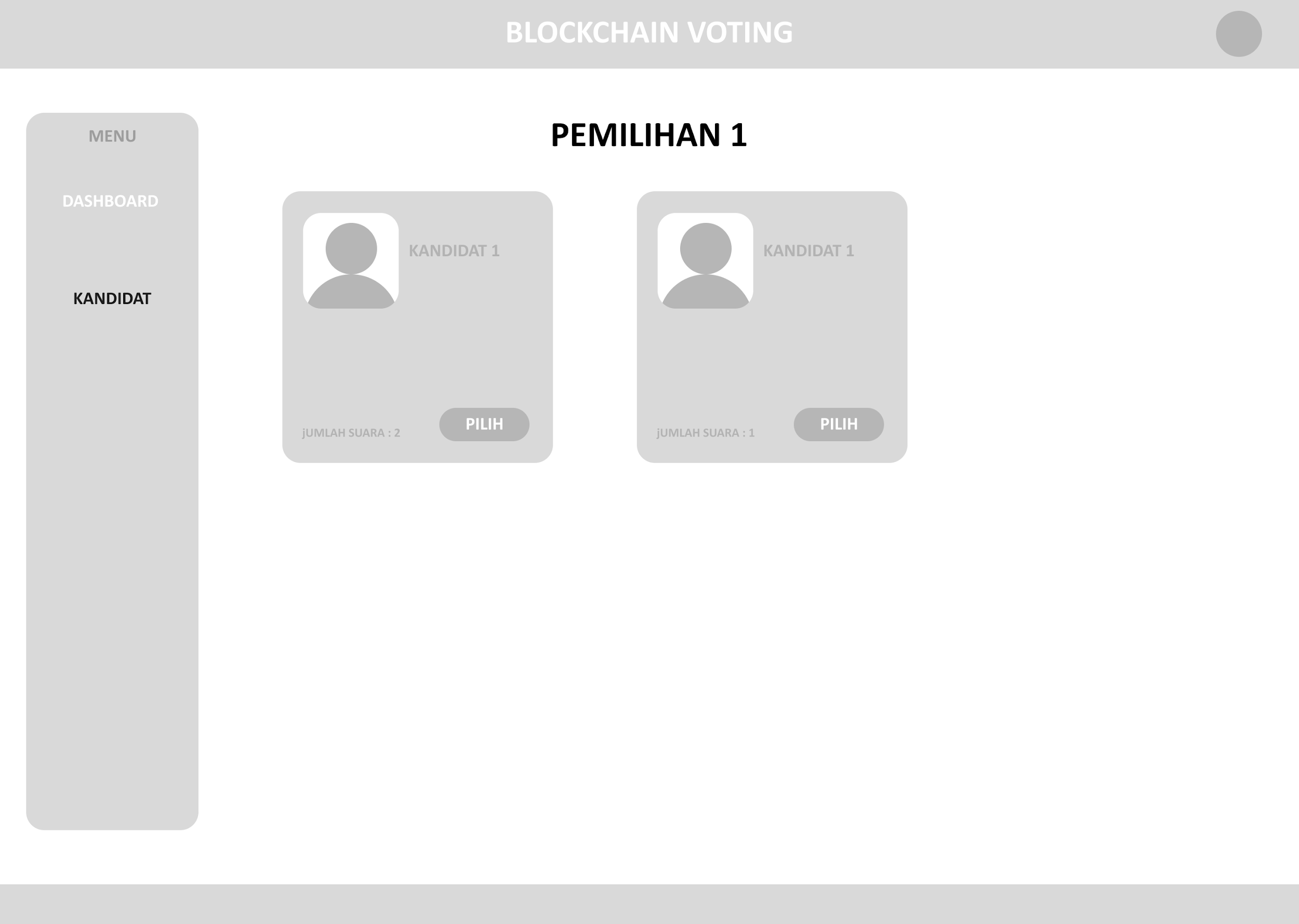
Gambar 4.12 merupakan halaman *voting creation* untuk admin membuat pemilihan dengan menambahkan kandidat yang telah terdaftar.



#### Gambar 4.12 Rancangan Antarmuka *Voting Creation*

1. Halaman Pemilihan

Gambar 4.13 merupakan halaman pemilihan untuk *user*, dimana ada daftar kandidat yang terdaftar dan juga sekaligus halaman untuk result, yaitu hasil pemilihan yang akan telah dilaksanakan.



#### Gambar 4.13 Rancangan Antarmuka *User-Vote*

## **4.4. *Test***

Pengujian dilakukan untuk memastikan bahwa setiap bagian dari sistem berfungsi dengan baik. dua metode pengujian utama akan digunakan: *Black Box Testing* dan *Mock Election*.

### **4.4.2. *Black Box Testing***

Tujuan dari pengujian black box pada sistem e-voting berbasis blockchain ini adalah untuk memastikan bahwa fungsi-fungsi utama, seperti autentikasi, pembuatan pemilihan, proses pemilihan, dan tampilan hasil pemilihan, bekerja dengan benar sesuai dengan kebutuhan. Hal ini penting untuk menjamin bahwa pengguna, baik itu pemilih maupun admin, dapat menggunakan aplikasi tanpa kendala. Setelah melakukan pengujian berdasarkan skenario yang telah ditentukan, hasil dari setiap skenario dicatat untuk dievaluasi. Jika ada skenario yang tidak berhasil, analisis dilakukan untuk menemukan penyebab kesalahan dan memperbaiki fungsi yang bermasalah.

##### Tabel 4.1 Hasil skenario *Black box* Pemilih

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Skenario** | **Langkah Pengujian** | **Hasil yang Diharapkan** | **Hasil Pengujian** |
| 1 | **Login Pengguna** | 1. Masukkan username dan password yang valid. 2. Klik tombol "Login". | Pengguna berhasil login dan diarahkan ke halaman dashboard. | **Berhasil** |
| 2 | **Pemilihan Kandidat** | 1. Login sebagai pemilih yang sudah terdaftar. 2. Pilih kandidat dari daftar kandidat yang tersedia. 3. Klik tombol "Submit". | Suara berhasil disimpan dan pesan konfirmasi ditampilkan. | **Berhasil** |
| 4 | **Hasil Pemilihan** | 1. Klik halaman "Hasil" di antarmuka pengguna 2. Lihat hasil pemilihan | Pengguna berhasil diarahkan menuju halaman hasil dan sistem memperlihatkan hasil pemilihan. | **Berhasil** |
| 5 | **Pemilihan Ganda** | 1. Login sebagai pemilih yang sudah terdaftar. 2. Pilih kandidat dari daftar kandidat yang tersedia. 3. Klik tombol "Submit". | Pemilih ditolak saat mencoba memberikan suara kedua kali | **Berhasil** |
| 6 | **Keamanan Sistem** | 1. Coba akses halaman yang dilindungi tanpa login. | Sistem mengarahkan pengguna ke halaman login. | **Berhasil** |

##### Tabel 4.2 Hasil skenario *Black box* Admin

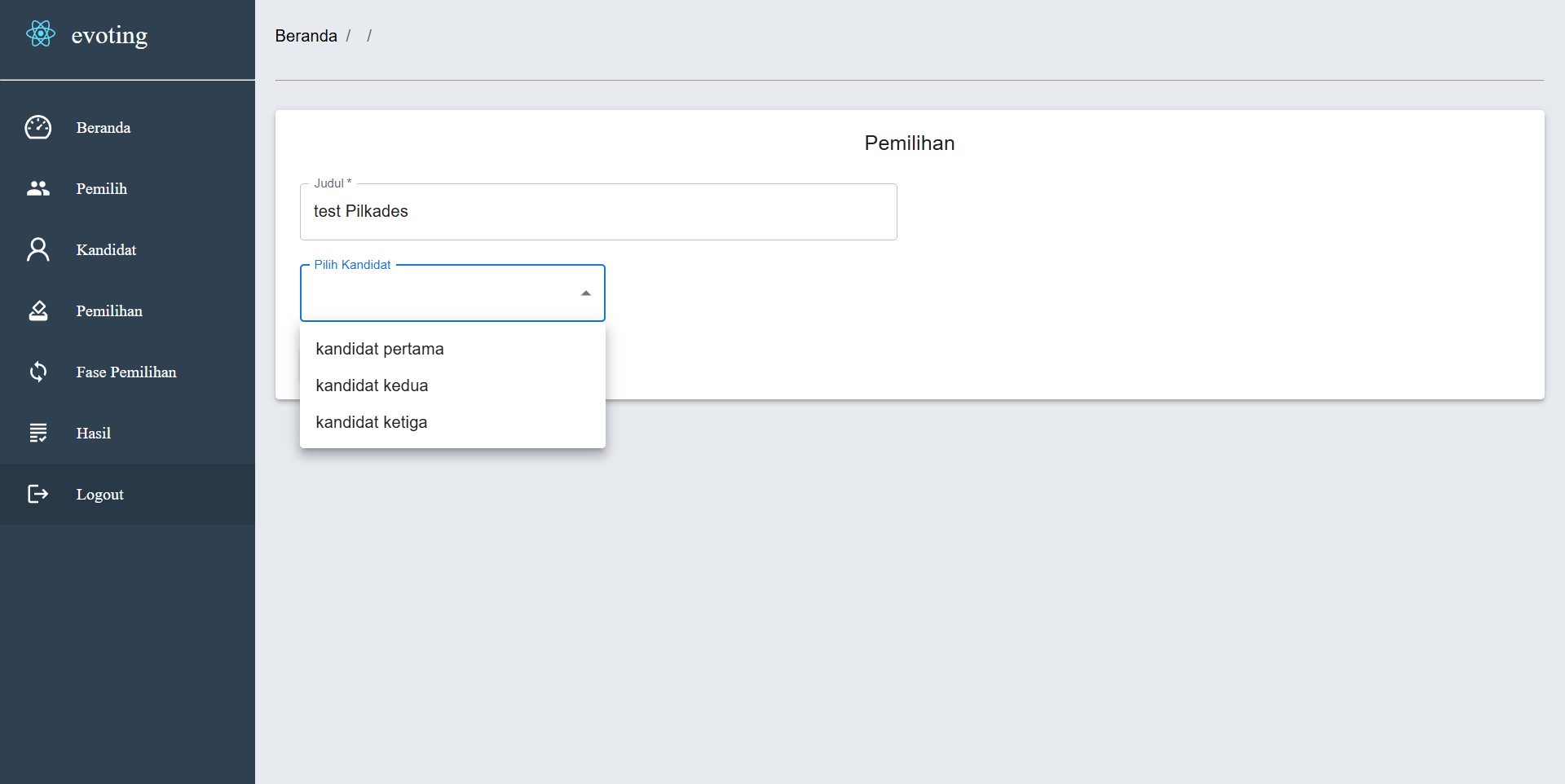
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Skenario** | **Langkah Uji** | **Output yang Diharapkan** | **Hasil Pengujian** |
| 1 | **Login Admin** | 1. Masukkan username dan password yang valid. 2. Klik tombol "Login". | Admin berhasil login dan diarahkan ke halaman dashboard. | **Berhasil** |
| 2 | **Pendaftaran Pemilih** | 1. Masukkan data diri lengkap pada form pendaftaran. 2. Klik tombol "Daftar". | Pesan konfirmasi bahwa pendaftaran berhasil dan data pemilih disimpan di database. | **Berhasil** |
| 3 | **Pendaftaran Kandidat** | 1. Klik menu "Kandidat" 2. Klik tombol "Tambah Kandidat" 3. Isi form dengan informasi kandidat seperti nama, email, deskripsi dan foto. | Admin berhasil menambahkan kandidat kedalam daftar kandidat | **Berhasil** |
| 4 | **Membuat Pemilihan** | 1. Klik tombol "Buat Pemilihan Baru" pada antarmuka admin. 2. Isi form dengan informasi pemilihan seperti nama pemilihan, tanggal, dan kandidat. 3. Klik tombol "Simpan" | Pemilihan baru berhasil dibuat dan muncul di daftar pemilihan | **Berhasil** |
| 5 | **Logout** | 1. Klik tombol "Logout" di antarmuka admin 2. Verifikasi bahwa pengguna diarahkan ke halaman login atau beranda. | Admin berhasil logout dan diarahkan ke halaman login atau beranda. | **Berhasil** |

Setelah melakukan pengujian berdasarkan skenario yang telah ditentukan, hasil dari setiap skenario dicatat untuk dievaluasi. Jika ada skenario yang tidak berhasil, analisis dilakukan untuk menemukan penyebab kesalahan dan memperbaiki fungsi yang bermasalah. Berdasarkan hasil pengujian *black box* di atas, dapat disimpulkan bahwa sistem e-voting berbasis blockchain berhasil menjalankan fungsinya sesuai dengan kebutuhan yang telah dirancang. Semua fitur utama dapat berjalan dengan baik dan memberikan hasil yang sesuai dengan yang diharapkan.

### **4.4.3. *Mock Election***

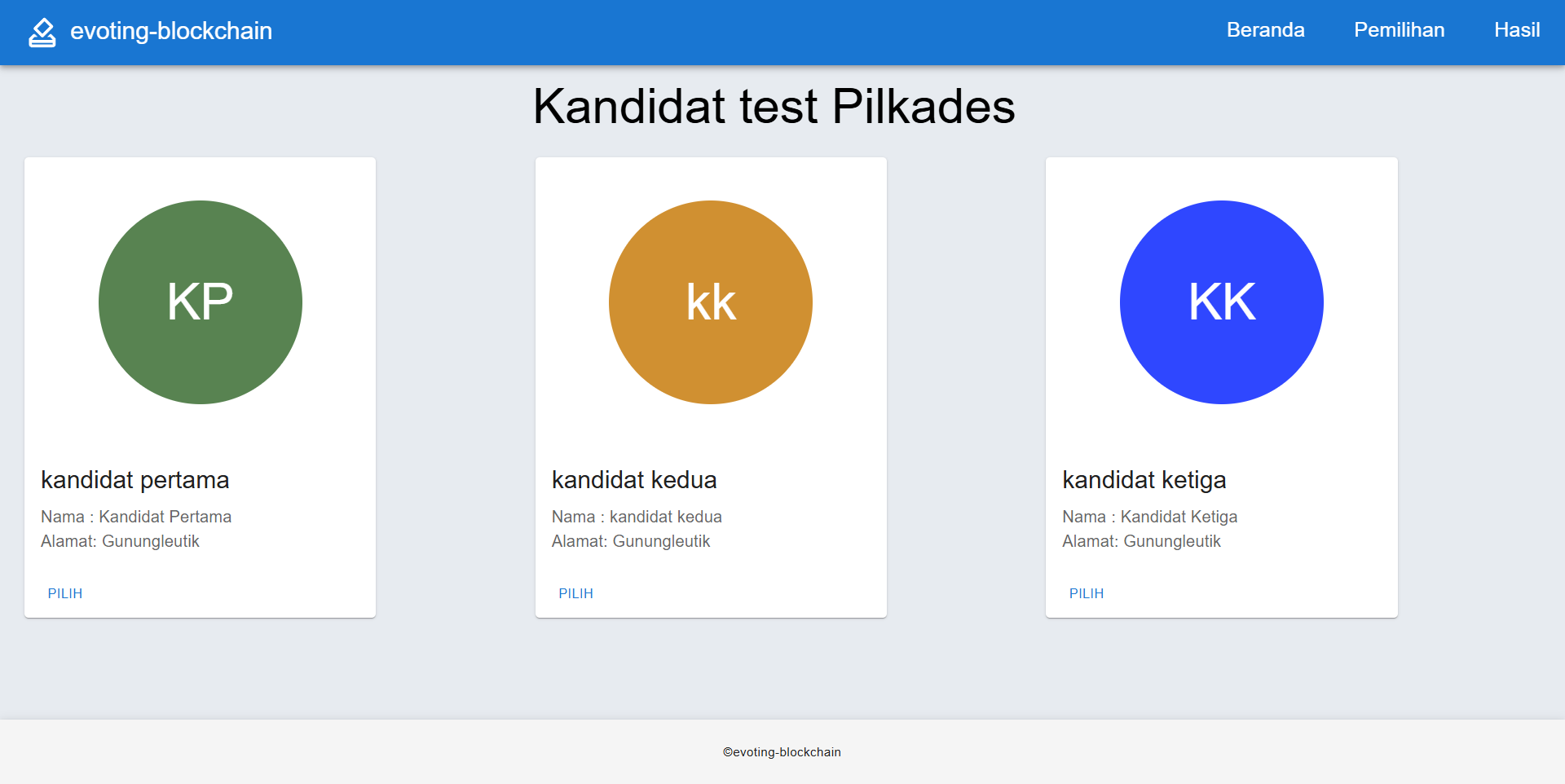
*Mock Election* adalah simulasi pemilihan elektronik yang dilakukan untuk menguji sistem dalam skenario nyata. Dalam uji coba ini, sejumlah pemilih melakukan pendaftaran, pemungutan suara, dan hasil pemungutan suara dihitung secara otomatis oleh sistem. Tujuannya adalah untuk menguji pengalaman pengguna, stabilitas sistem, dan kinerja secara keseluruhan.

1. Admin dapat menambahkan Pemilihan dalam sistem, dimana admin mengisi judul dan kandidat setelah klik tombol simpan, data pemilihan akan ditambahkan ke dalam database.



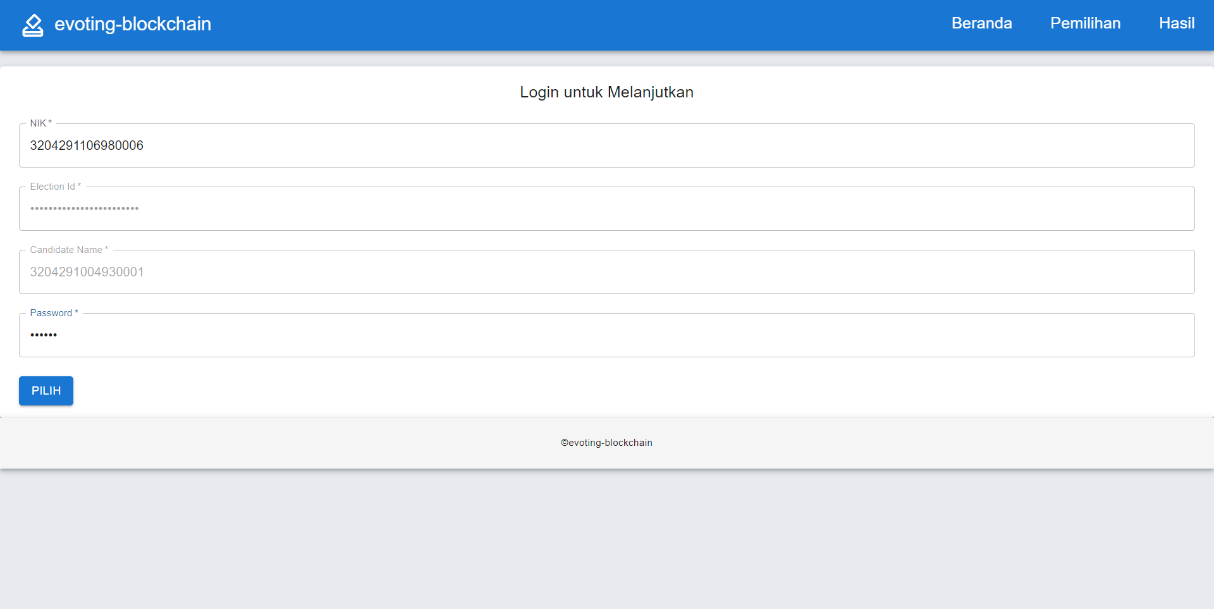
#### Gambar 4.14 *Add Election*

1. Pada halaman detail pemilihan, pemilihdapat memilih para kandidat yang tersedia, dengan klik tombol pilih.



#### Gambar 4.15 *Election-detail*

1. Setelah pemilih memilih pada halaman pemilihan, pemilih akan diarahkan menuju halam verifikasi data diri, dan *user* memasukan *username* dan *password* untuk mengonfirmasi kandidat yang dipilihnya.



#### Gambar 4.16 *User Login*

1. Setelah mengisi data verifikasi pemilih akan mendapati *pop-up* ekstensi *metamask* untuk mengonfirmasi transaksi dengan *ethereum blockchain*.



#### Gambar 4.17 *Metamask Confirmation*

1. Percobaan simulasi telah dilakukan oleh 26 pemilih, dan hasilnya disajikan pada Tabel 4.3.

##### Tabel 4.3 Hasil Mock Election

|  |  |  |  |
| --- | --- | --- | --- |
| **Percobaan Pengguna** | **Login Pengguna** | **Pemilihan Kandidat** | **Hasil Pemilihan** |
| **1** | Berhasil | Berhasil | Berhasil |
| **2** | Berhasil | Berhasil | Berhasil |
| **3** | Berhasil | Berhasil | Berhasil |
| **4** | Berhasil | Berhasil | Berhasil |
| **5** | Berhasil | Berhasil | Berhasil |
| **6** | Berhasil | Berhasil | Berhasil |
| **7** | Berhasil | Berhasil | Berhasil |
| **8** | Berhasil | Berhasil | Berhasil |
| **9** | Berhasil | Berhasil | Berhasil |
| **10** | Berhasil | Berhasil | Berhasil |
| **11** | Berhasil | Berhasil | Berhasil |
| **12** | Berhasil | Berhasil | Berhasil |
| **13** | Berhasil | Berhasil | Berhasil |
| **Percobaan Pengguna** | **Login Pengguna** | **Pemilihan Kandidat** | **Hasil Pemilihan** |
| **14** | Berhasil | Berhasil | Berhasil |
| **15** | Berhasil | Berhasil | Berhasil |
| **16** | Berhasil | Berhasil | Berhasil |
| **17** | Berhasil | Berhasil | Berhasil |
| **18** | Berhasil | Berhasil | Berhasil |
| **19** | Berhasil | Berhasil | Berhasil |
| **20** | Berhasil | Berhasil | Berhasil |
| **21** | Berhasil | Berhasil | Berhasil |
| **22** | Berhasil | Berhasil | Berhasil |
| **23** | Berhasil | Berhasil | Berhasil |
| **24** | Berhasil | Berhasil | Berhasil |
| **25** | Berhasil | Berhasil | Berhasil |
| **26** | Berhasil | Berhasil | Berhasil |

Penelitian ini melakukan *black box testing* dan simulasi pemilihan (*mock election*), di mana hasil pengujian sistem pemilihan elektronik (*e-voting*) dapat memenuhi kebutuhan untuk skala pemilihan kepala desa. Dalam proses pengujian ini, total sebesar 0,09503622 ETH digunakan, mencakup mulai dari deploy smart contract hingga 26 transaksi untuk proses voting.

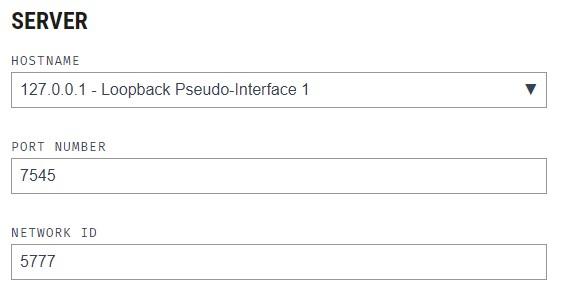
## **4.4. *Deploy***

Tahap *Deployment* adalah proses di mana produk yang telah selesai dikembangkan diimplementasikan ke lingkungan produksi atau disediakan untuk pengguna akhir.

### **4.4.1. *Smart Contract***

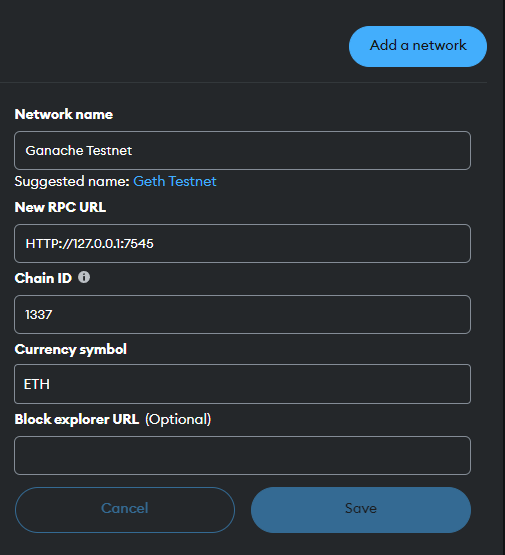
Implementasi *Smart Contract* bertujuan untuk memastikan bahwa kode *smart contract* yang ditulis dalam *Solidity* bekerja sesuai spesifikasi. Pengujian dilakukan dengan menggunakan *Node.js v.18.40, Truffle v5.11.5 , Ganache v2.7.1 , Solidity, dan web3.js.*

Untuk menginstall *truffle* bisa menggunakan *command* “npm install -g truffle” pada *terminal.* Lalu untuk menginstall ganache bisa dengan mengunjungi official website dan mengikuti instruksi yang tertera. Selanjutnya adalah melakukan konfigurasi ganache pada menu *setting*, seperti yang tertera pada gambar 4.14:



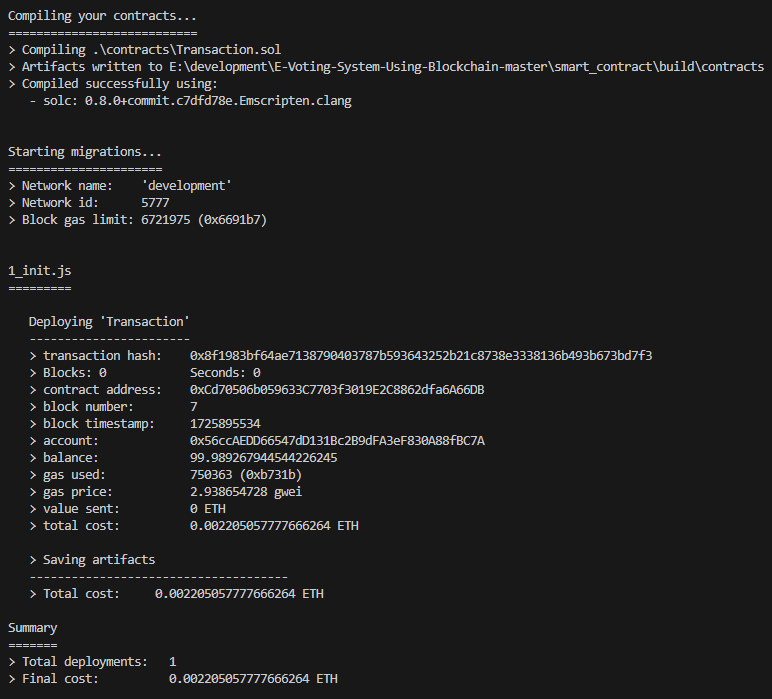
#### Gambar 4.18 *Ganache Server Detail*

Selanjutnya konfigurasi *metamask* untuk jaringan *local blockchain* seperti pada gambar 4.15:



#### Gambar 4.19 Konfigurasi *Metamask*

Setelah selesai konfigurasi *ganache* dan *metamask*, langkah selanjutnya adalah masuk ke direktori *smart contract* dan menjalankan *command* “*truffle migrate*” pada *terminal*. Perintah ini berfungsi untuk men-*deploy* *smart contract* ke jaringan *local blockchain*. Seperti pada gambar 4.16:



#### Gambar 4.20 *Truffle Migrate*

Proses kompilasi dan migrasi *smart contract* "*Transaction*" berhasil dilakukan pada jaringan pengembangan lokal dengan menggunakan *Truffle*. Kompilasi *smart contract* ini menggunakan *Solidity* *versi 0.8.0* tanpa mengalami kesalahan. Proses migrasi dilakukan pada jaringan dengan *Network ID* 5777 menggunakan server *ganache,* dengan batas gas maksimum sebesar 6.721.975. Setelah kontrak berhasil dideploy, *hash* transaksi yang dihasilkan adalah 0x8f9183bf64ae713879004378b599643252b21c8738e333816b493b67b3df7f3, dan kontrak di *assign* ke alamat 0xcd7d80b65936327f03f0192c8862df6a4606DB di blok nomor 7.

Transaksi ini menggunakan 75.963 *gas*, dengan harga *gas* sebesar 2.93864728 *gwei*, dan total biaya untuk melakukan *deploy* kontrak ini adalah 0.00220557777666264 *ETH*, dan setelah *deploy* berhasil, artifacts yang diperlukan untuk interaksi selanjutnya disimpan. Secara keseluruhan, *deploy* kontrak berjalan sukses dan siap untuk digunakan pada jaringan *blockchain*.

BAB V SIMPULAN DAN SARAN

## **4.3. Kesimpulan**

Berdasarkan penelitian yang telah dilakukan mengenai **Perancangan dan Implementasi Sistem Pemilihan Elektronik (*E-Voting*) Berbasis *Blockchain***, dapat disimpulkan bahwa sistem *e-voting* berbasis blockchain menawarkan solusi yang signifikan untuk mengatasi berbagai permasalahan yang sering dihadapi dalam pemilihan tradisional. Teknologi *blockchain*, dengan fitur desentralisasi dan keamanan berbasis kriptografi, mampu memberikan keamanan, transparansi, anonimitas, dan efisiensi yang lebih baik dibandingkan sistem manual.

Pengujian melalui *mock election* atau simulasi dan implementasi *smart contract* menunjukkan bahwa sistem dapat berjalan dengan baik dalam mencatat suara secara elektronik, data pemilih dan hasil suara terlindungi dari manipulasi maupun akses tidak sah. Penggunaan *smart contract* memastikan bahwa semua suara yang diberikan tidak dapat diubah setelah tercatat di *blockchain*, serta hanya pemilih yang terverifikasi yang dapat memberikan suara. Pengujian juga menunjukkan bahwa sistem ini mampu menjaga integritas data pemilih dan suara dengan tingkat keamanan yang tinggi, memastikan proses pemilihan berjalan adil dan tidak rentan terhadap kecurangan.

## **4.3. Saran**

Berdasarkan penelitian yang telah dilakukan mengenai Perancangan dan Implementasi Sistem Pemilihan Elektronik (*E-Voting*) Berbasis *Blockchain*, beberapa saran yang diperoleh untuk penelitian masa depan:

* + - 1. *Blockchain* memiliki potensi besar dalam banyak sektor, dan penerapannya dalam *e-voting* hanya satu contoh kasus. Penelitian lebih lanjut bisa difokuskan pada penerapan teknologi *blockchain* di sektor lain *selain e-voting*, seperti sektor keuangan, pendidikan (misalnya, sistem sertifikasi digital), atau manajemen rantai pasok (*supply chain*). Ini akan membantu memahami bagaimana *blockchain* dapat meningkatkan transparansi dan keamanan di berbagai bidang.
      2. Skalabilitas sistem, Untuk memastikan sistem dapat digunakan dalam skala yang lebih besar seperti pemilihan umum nasional, perlu dilakukan uji coba pada jaringan *blockchain* yang lebih besar dan desentralisasi penuh. Hal ini akan memastikan bahwa sistem tetap responsif meskipun digunakan oleh jutaan pemilih khususnya di Indonesia.
      3. Penelitian lanjutan dapat meneliti cara-cara untuk meningkatkan usability dan aksesibilitas sistem *e-voting* berbasis *blockchain* bagi pemilih yang tidak memiliki literasi teknologi yang tinggi, seperti integrasi dengan aplikasi seluler sederhana atau penggunaan teknologi biometrik.
      4. Kolaborasi dengan pemerintahan dan institusi terkait, Implementasi *e-voting* berbasis *blockchain* sebaiknya melibatkan kerjasama dengan pihak pemerintah, komisi pemilihan, serta ahli keamanan siber untuk memastikan bahwa sistem yang dibangun dapat memenuhi standar keamanan dan regulasi yang ada.

DAFTAR PUSTAKA

[1] L. Hardjaloka and V. M. Simarmata, “E-Voting: Kebutuhan vs. Kesiapan (Menyongsong) E-Demokrasi,” *J. Konstitusi*, vol. 8, no. 4, p. 579, 2016, doi: 10.31078/jk847.

[2] M. M.Ilyas Gultom and D. Saripurna, “Perancangan Sistem Keamanan Aplikasi E-Voting Untuk Pemilihan Ketua Badan Eksekutif Mahasiswa Fakultas Teknik UISU Dengan Menggunakan Algoritma MD5,” *Algoritm. J. Ilmu Komput. Dan Inform.*, vol. 3, no. 2, p. 70, 2019, doi: 10.30829/algoritma.v3i2.6438.

[3] T. Adekeye, “Implementing a secure electronic voting system using Blockchain smart contract,” no. March, 2023, doi: 10.13140/RG.2.2.27008.10242.

[4] K. Curran, “E-Voting on the Blockchain,” *J. Br. Blockchain Assoc.*, vol. 1, no. 2, pp. 1–6, 2018, doi: 10.31585/jbba-1-2-(3)2018.

[5] U. Jafar, M. J. A. Aziz, and Z. Shukur, “Blockchain for electronic voting system—review and open research challenges,” *Sensors*, vol. 21, no. 17, 2021, doi: 10.3390/s21175874.

[6] B. Shahzad and J. Crowcroft, “Trustworthy Electronic Voting Using Adjusted Blockchain Technology,” *IEEE Access*, vol. 7, pp. 24477–24488, 2019, doi: 10.1109/ACCESS.2019.2895670.

[7] S. Restu Aji and W. Trisari Harsanti Putri, “Implementasi Teknologi Blockchain dalam Aplikasi E-Voting Berbasis Mobile,” *Digit. Zo. J. Teknol. Inf. dan Komun.*, vol. 14, no. 2, pp. 219–231, 2023, doi: 10.31849/digitalzone.v14i2.16682.

[8] D.Jyothi Reddy, B. Vineetha Lakshmi, G.Smily Srinidhi, B.Udayasri, and Md.Shakeel Ahmed, “Electronic Voting Using Decentralized System Based on Ethereum Blockchain,” *EPRA Int. J. Res. Dev.*, vol. 19, pp. 363–370, 2024, doi: 10.36713/epra16278.

[9] T. Hanne, S. Deepak, B. Dhil Rohith, S. P. Sri Hari Nivas, and S. T. Shobika, “Blockchain Based E Voting System,” *Blockchain Intell. Syst. Protoc. Appl. Approaches Futur. Gener. Comput.*, pp. 193–208, 2024, doi: 10.48175/ijarsct-19471.

[10] Z. Zheng, S. Xie, H. N. Dai, X. Chen, and H. Wang, “Blockchain challenges and opportunities: A survey,” *Int. J. Web Grid Serv.*, vol. 14, no. 4, pp. 352–375, 2018, doi: 10.1504/IJWGS.2018.095647.

[11] M. Liu, K. Wu, and J. J. Xu, “How Will Blockchain Technology Impact Auditing and Accounting: Permissionless versus Permissioned Blockchain,” *Curr. Issues Audit.*, vol. 13, no. 2, pp. A19–A29, 2019, doi: 10.2308/ciia-52540.

[12] B. K. Mohanta, S. S. Panda, and D. Jena, “An Overview of Smart Contract and Use Cases in Blockchain Technology,” *2018 9th Int. Conf. Comput. Commun. Netw. Technol. ICCCNT 2018*, no. November, pp. 1–4, 2018, doi: 10.1109/ICCCNT.2018.8494045.

[13] S. A. B. Salman, S. Al-Janabi, and A. M. Sagheer, “A Review on E-Voting Based on Blockchain Models,” *Iraqi J. Sci.*, vol. 63, no. 3, pp. 1362–1375, 2022, doi: 10.24996/ijs.2022.63.3.38.

[14] M. N. M. Bhutta *et al.*, “A Survey on Blockchain Technology: Evolution, Architecture and Security,” *IEEE Access*, vol. 9, no. April, pp. 61048–61073, 2021, doi: 10.1109/ACCESS.2021.3072849.

[15] G. Wood, “Ethereum: A Secure Decentralised Generalised Transaction Ledger Eip-150 Revision,” 2019. [Online]. Available: https://files.gitter.im/ethereum/yellowpaper/VIyt/Paper.pdf

[16] APTIKOM, “Teknologi Block Chain dan Cara Kerjanya,” pp. 1–32, 2020.

[17] A. Ben Ayed, “A Conceptual Secure Blockchain Based Electronic Voting System,” *Int. J. Netw. Secur. Its Appl.*, vol. 9, no. 3, pp. 01–09, 2017, doi: 10.5121/ijnsa.2017.9301.

[18] G. Antonopoulos, A. M., Wood, *Mastering Ethereum [Book]*. 2018. [Online]. Available: https://www.oreilly.com/library/view/mastering-ethereum/9781491971932/

[19] R. Julianto, “UML.” Accessed: Sep. 28, 2024. [Online]. Available: https://www.dicoding.com/blog/apa-itu-uml

[20] R. Setiawan, “Konsep Agile Pada Software Development.” Accessed: Sep. 28, 2024. [Online]. Available: https://www.dicoding.com/blog/konsep-agile-pada-software-development/

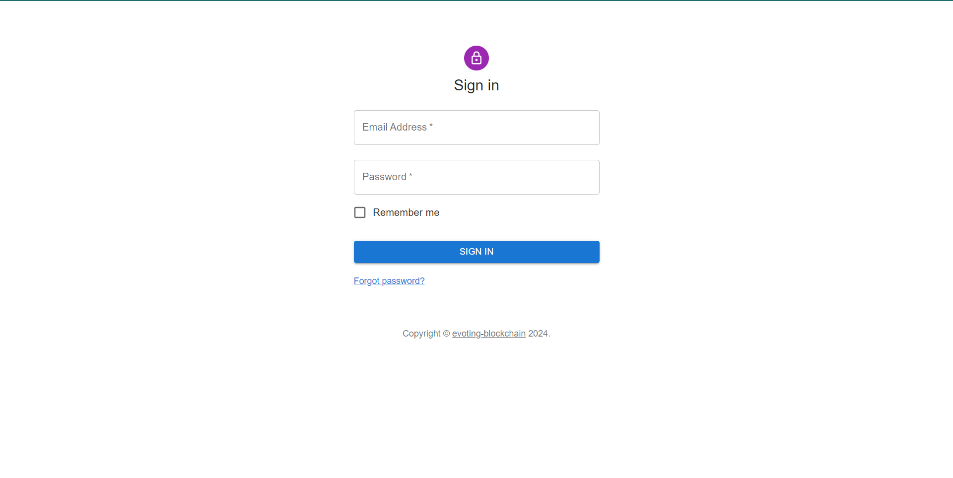
[21] M. Destiningrum and Q. J. Adrian, “Sistem Informasi Penjadwalan Dokter Berbassis Web Dengan Menggunakan Framework Codeigniter (Studi Kasus: Rumah Sakit Yukum Medical Centre),” *J. Teknoinfo*, vol. 11, no. 2, p. 30, 2017, doi: 10.33365/jti.v11i2.24.

[22] A. Singh, A. Ganesh, R. R. Patil, S. Kumar, R. Rani, and S. K. Pippal, “Secure Voting Website Using Ethereum and Smart Contracts,” *Appl. Syst. Innov.*, vol. 6, no. 4, pp. 1–22, 2023, doi: 10.3390/asi6040070.

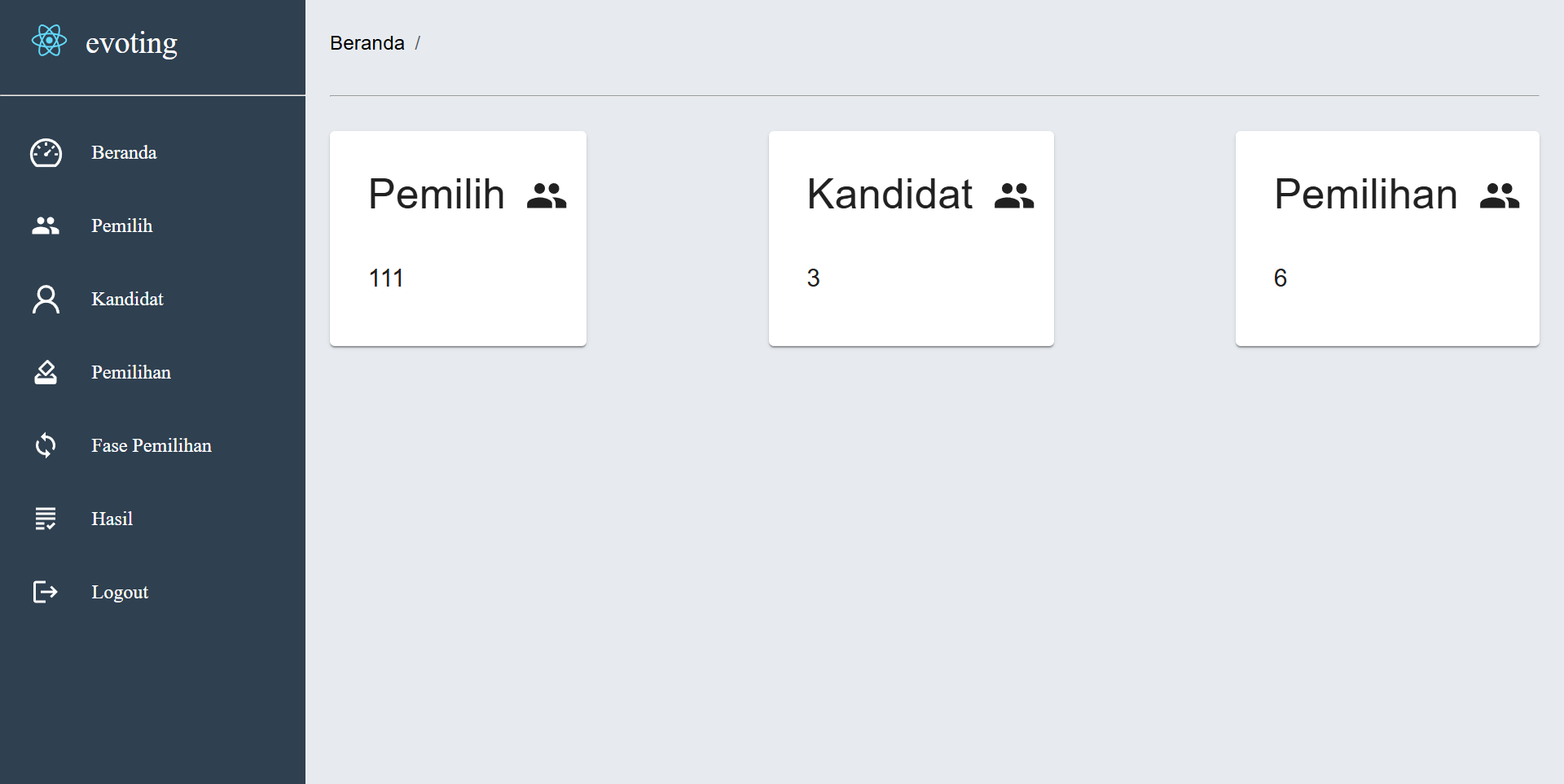
# **LAMPIRAN**

###### **Lampiran Hasil Penelitian**

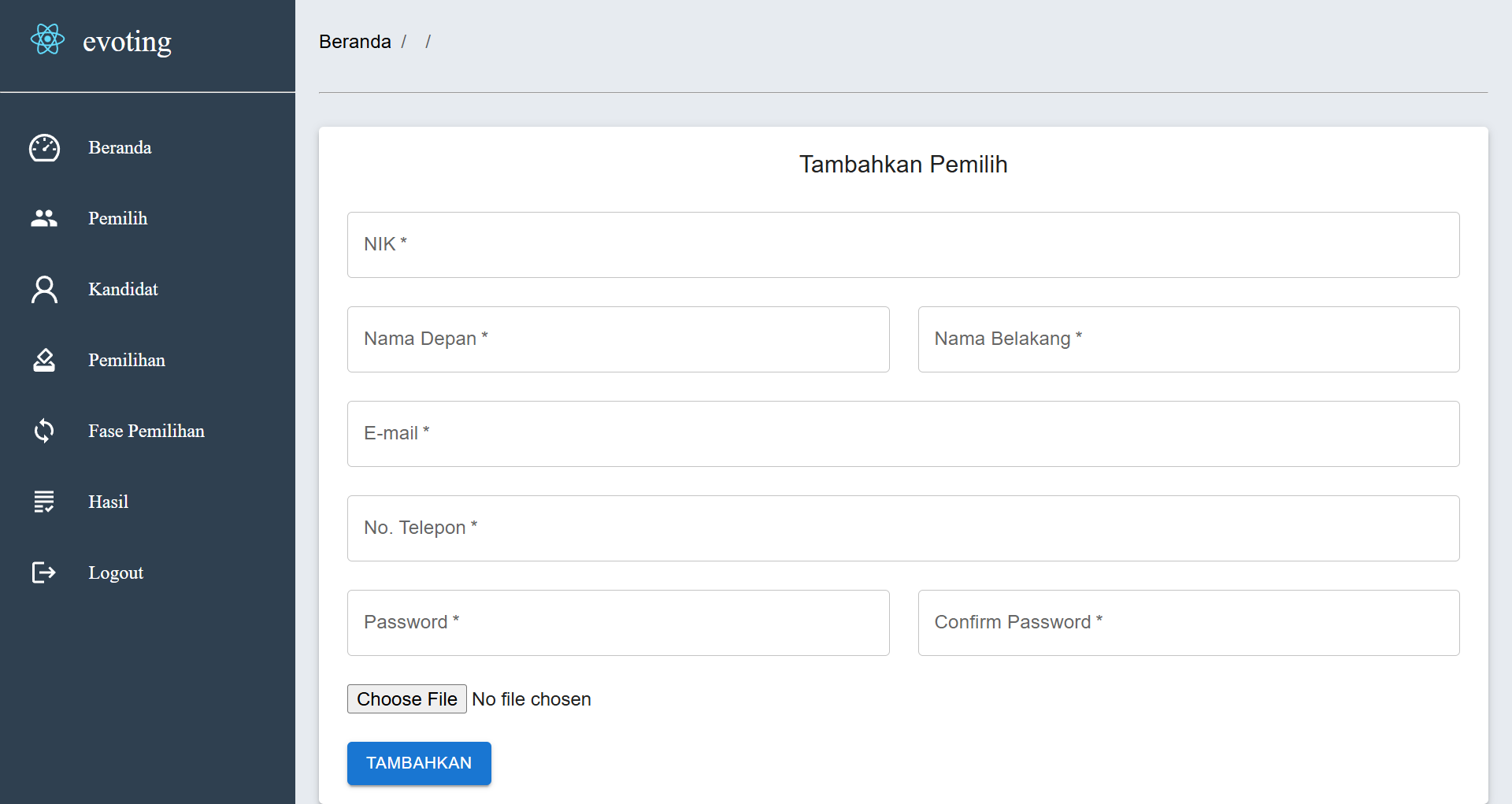
1. Halaman *login* untuk admin.



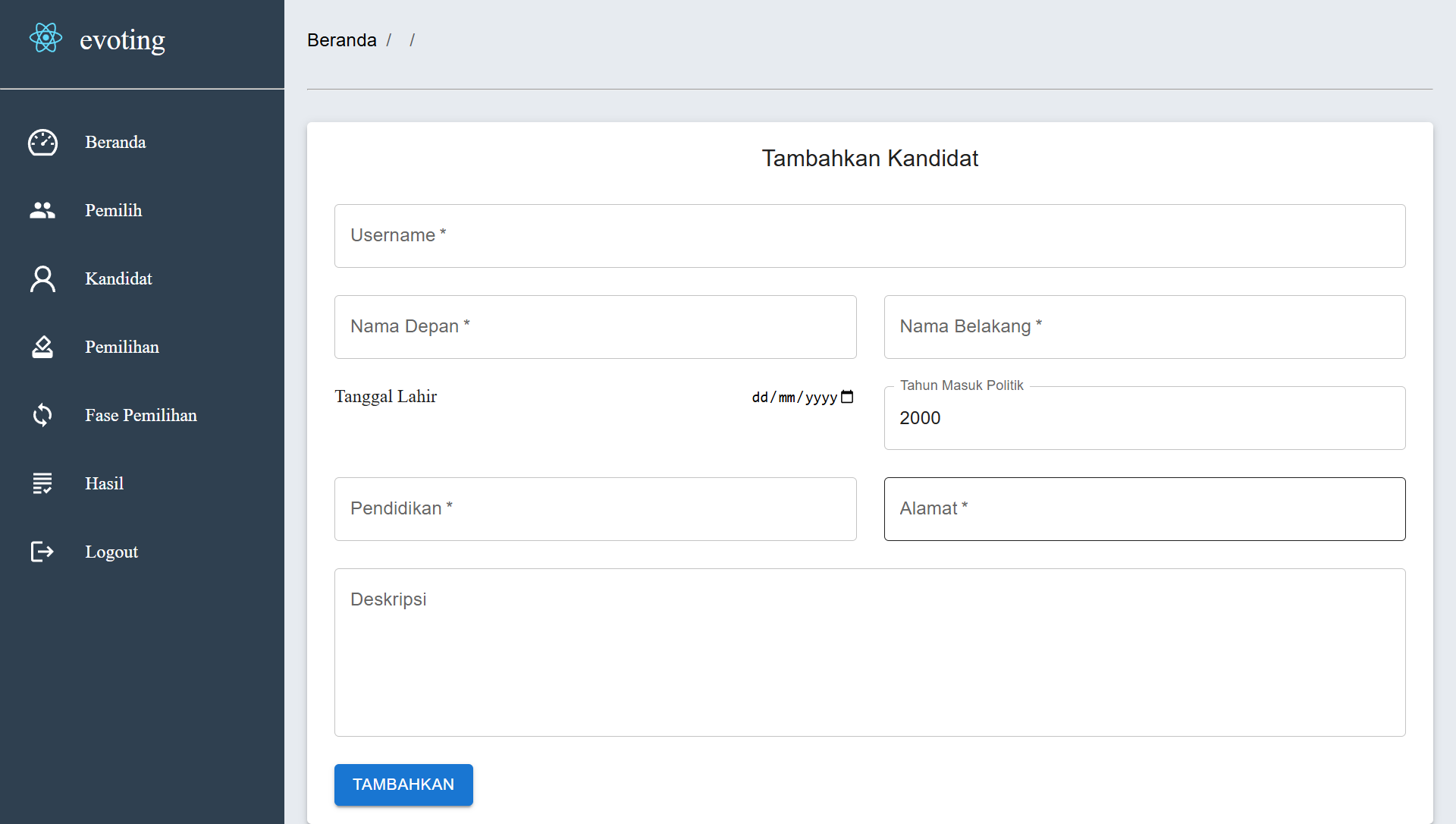
1. Ditampilkan *dahsboard admin* dimana semua informasi seperti jumlah pemilih, kandidat, dan pemilihan yang tersedia pada sistem, serta di sampingnya ada menu-menu seperti Pemilih , kandidat, pemilihan, fase pemilihan, dan hasil pemilihan.



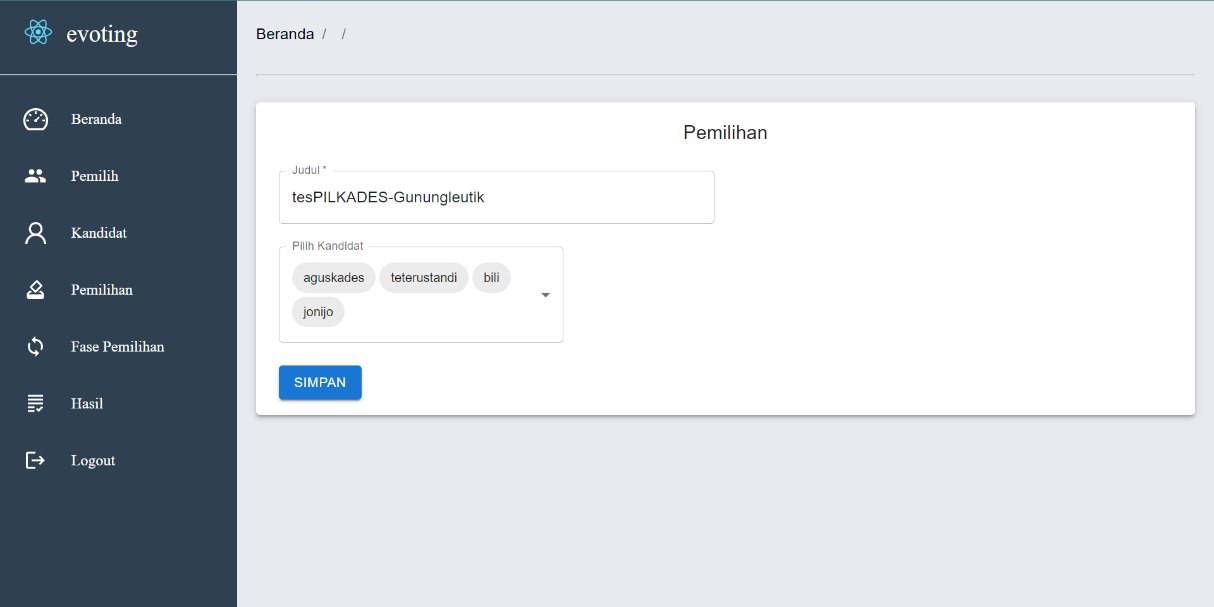
1. Admin dapat menambahkan Pemilih dalam sistem dimana admin harus mengisi detail seperti NIK, nama depan dan detail lainnya setelah klik tambah pengguna orang akan mendapatkan nama pengguna dan *password* dalam *email* yang dimasukkan dalam formulir yang diberikan.



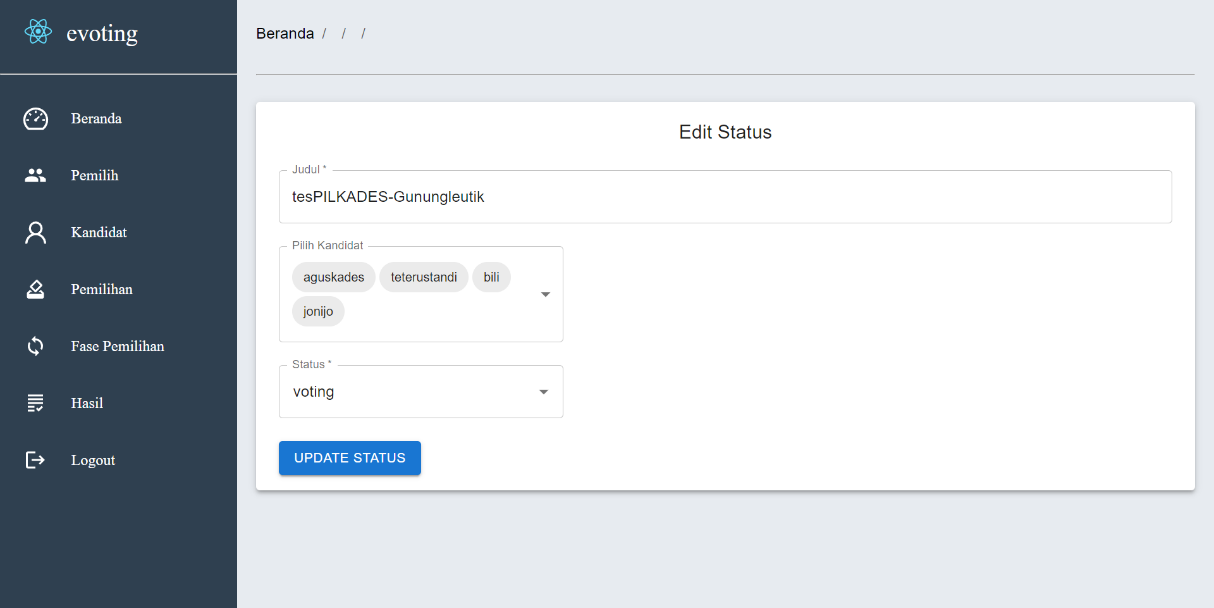
1. Admin dapat menambahkan Kandidat dalam sistem dimana admin harus mengisi detail seperti NIK, nama depan dan detail lainnya setelah klik Tambahkan, data kadidat akan ditambahkan ke dalam *database*.



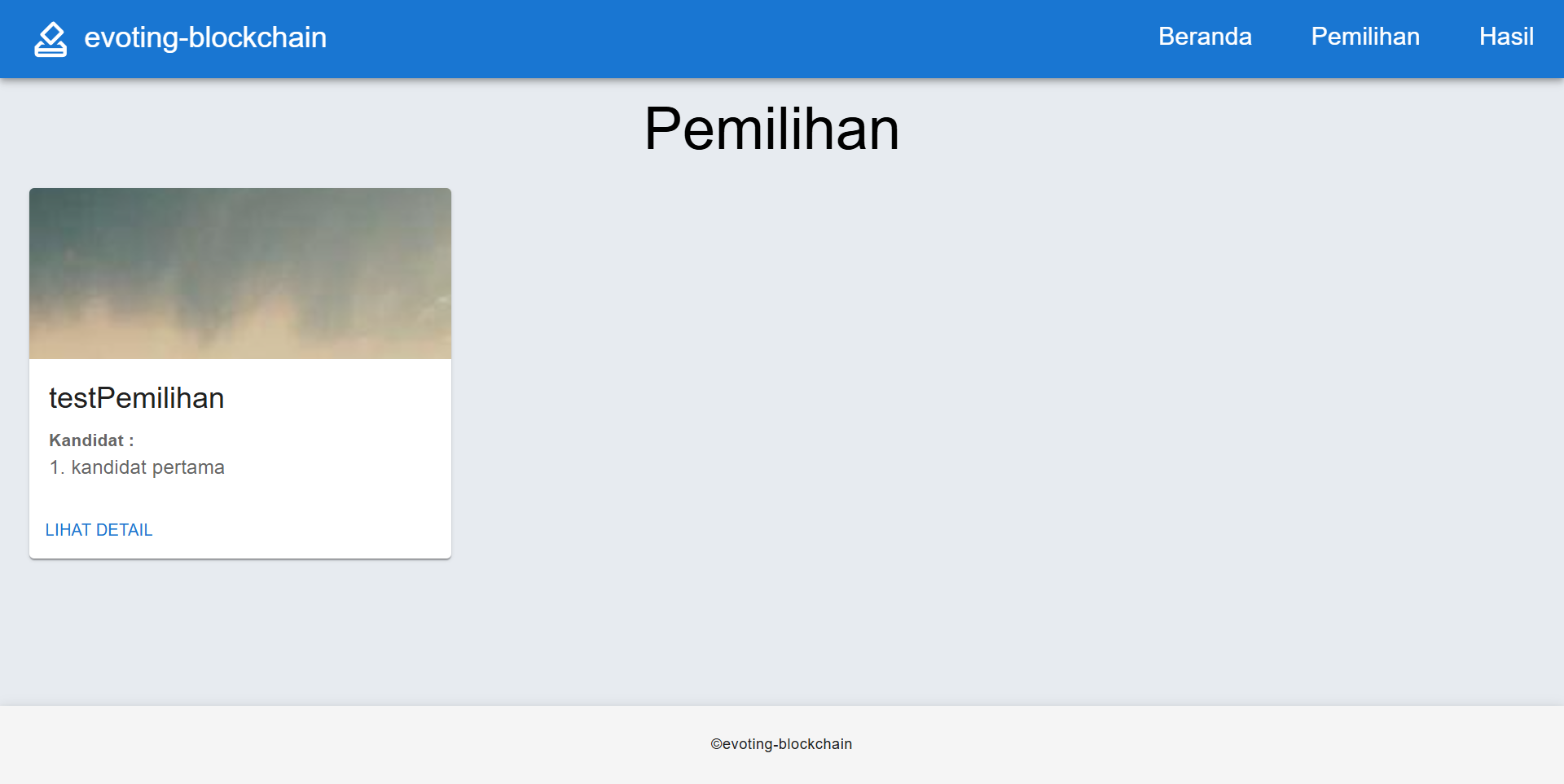
1. Admin dapat menambahkan Pemilihan dalam sistem, dimana admin mengisi judul dan kandidat setelah klik tombol simpan, data pemilihan akan ditambahkan ke dalam database.



1. Admin dapat mengubah status fase pemilihan dimana ada tiga fase yaitu *init*, *voting* dan *counting*.



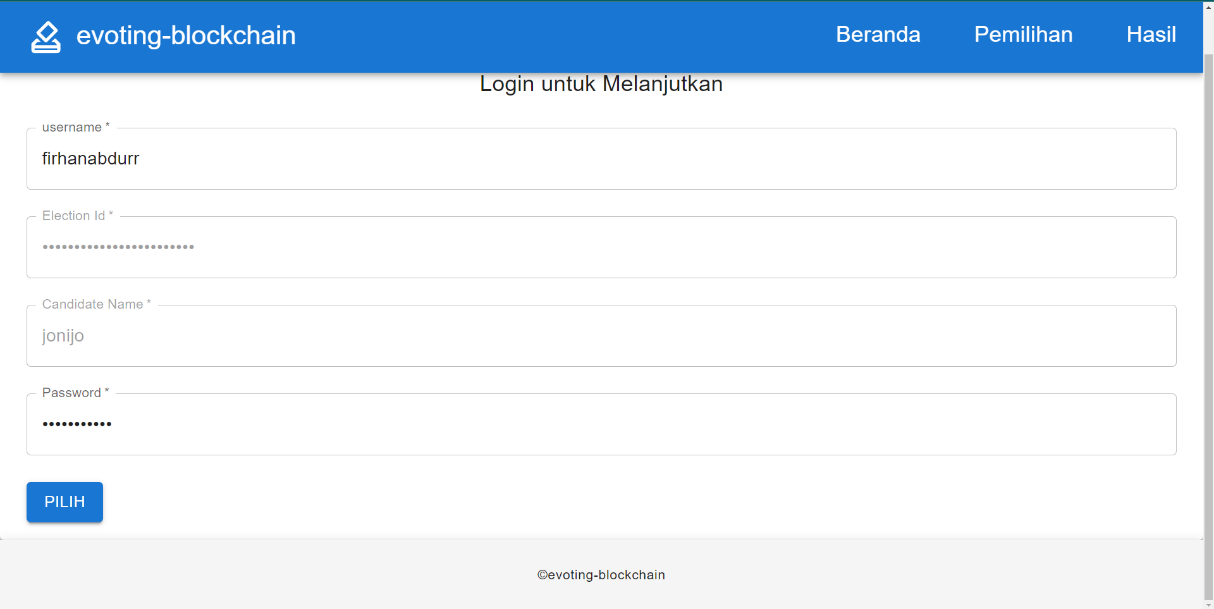
1. Pada halaman Pemilihan, pemilih dapat melihat pemilihan yang sedang berlangsung.



1. Pada halaman detail pemilihan, pemilihdapat memilih para kandidat yang tersedia, dengan klik tombol pilih.



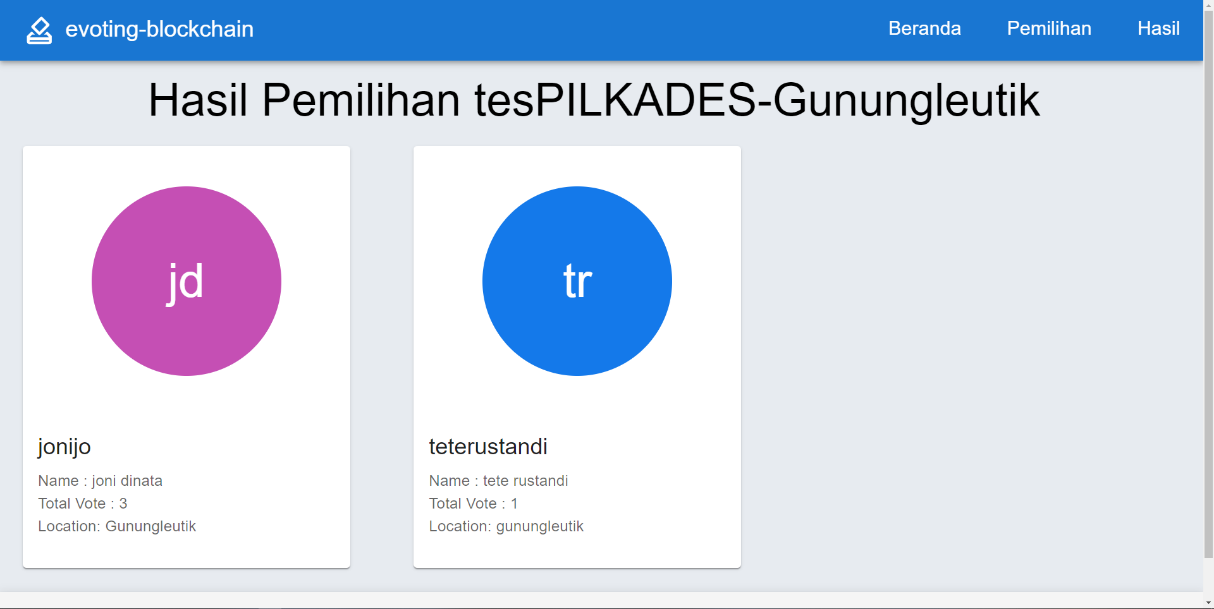
1. Setelah pemilih memilih pada halaman pemilihan, pemilih akan diarahkan menuju halam verifikasi data diri, dan *user* memasukan *username* dan *password* untuk mengonfirmasi kandidat yang dipilihnya.



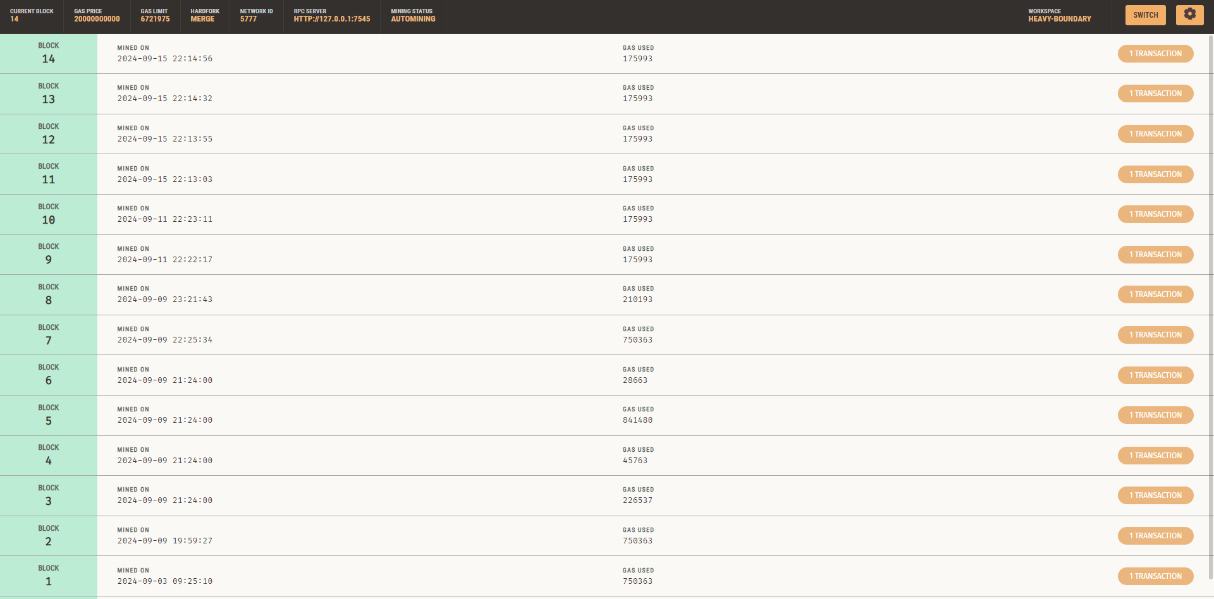
1. Setelah mengisi data verifikasi pemilih akan mendapati *pop-up* ekstensi *metamask* untuk mengonfirmasi transaksi dengan *ethereum blockchain*.

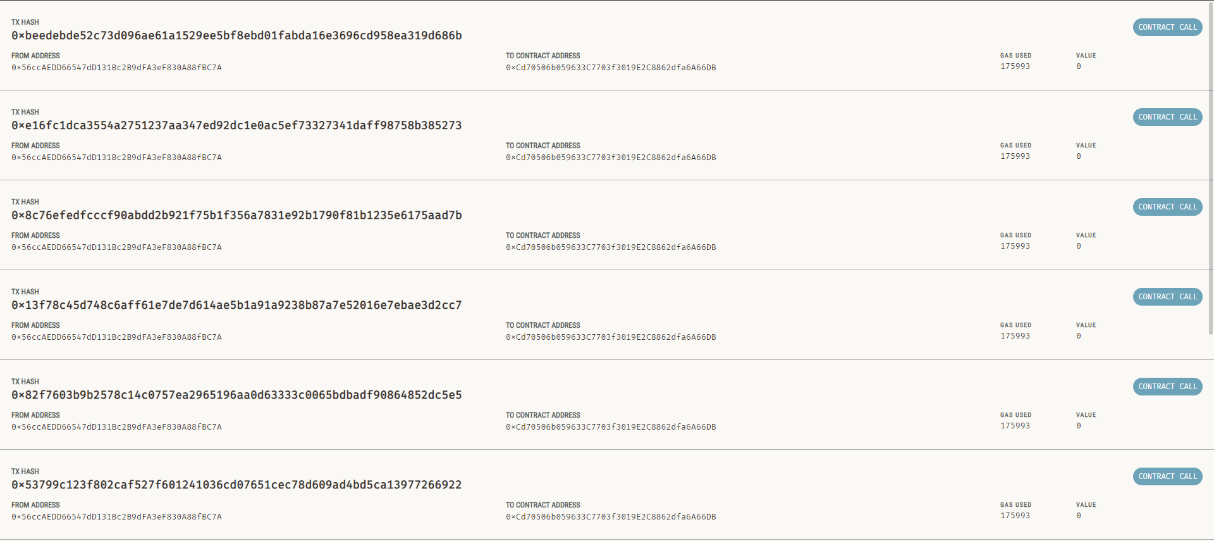


1. Terakhir untuk melihat hasil pemilihan terdapat halaman hasil, yang berisi hasil pemilihan dan jumlah suara yang didapat.

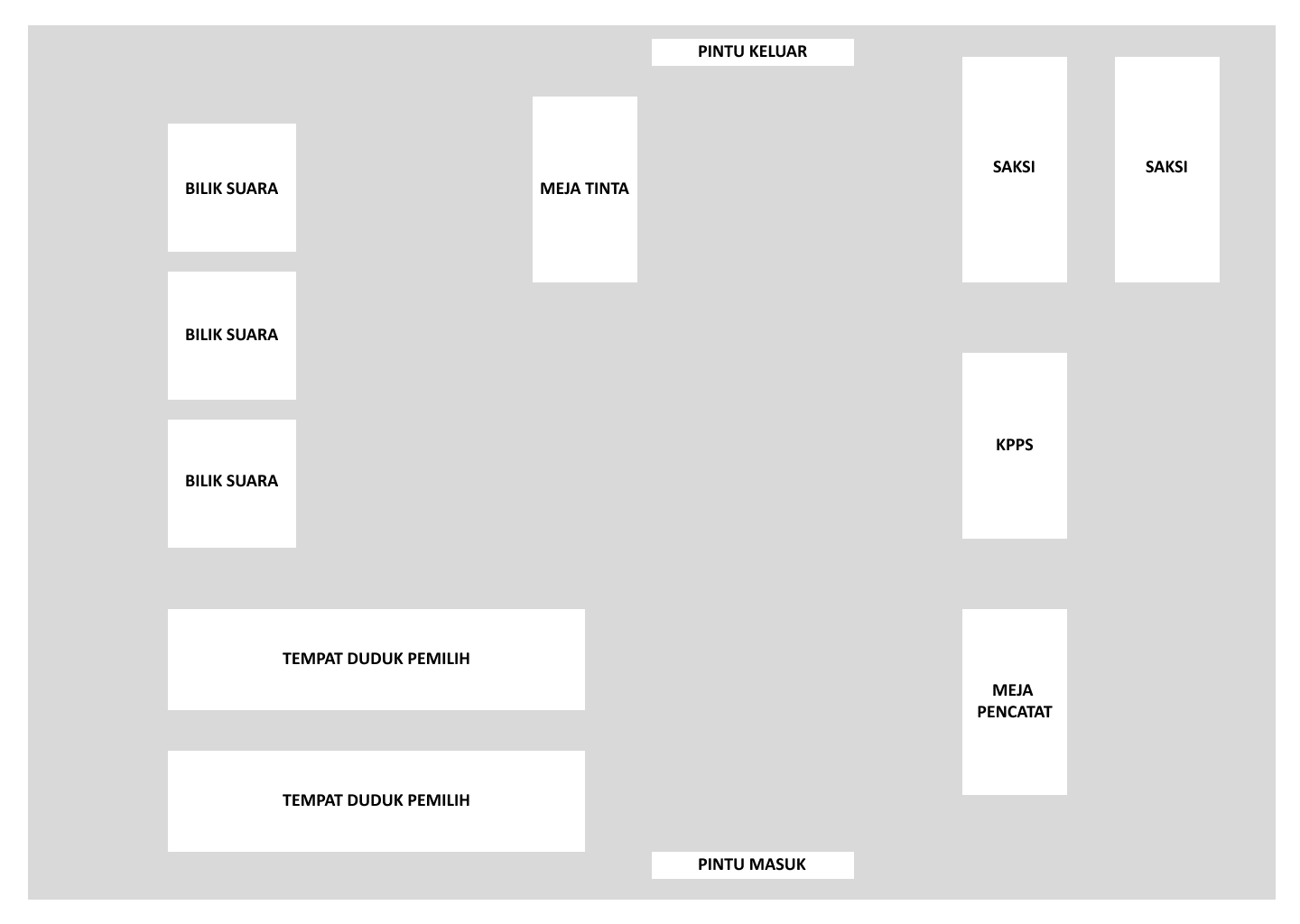


1. Untuk keperluan audit, setiap transaksi direkam ke dalam jaringan blockchain, gambar diatas merupakan rincian transaksi yang sudah terenkripsi menggunakan kriptografi *blockchain*.





1. Denah untuk mock election



###### **Lampiran Source Code Smart Contract**

|  |
| --- |
| pragma solidity ^0.8.0;  contract Transaction{      uint256 transactionCounter;      event Transfer(address from,address receiver,string user\_id,string election\_id,string candidate\_id,uint timestamp);      struct TransactionStruct {          address from;          address receiver;          string user\_id;          string election\_id;          string candidate\_id;          uint timestamp;      }      TransactionStruct[] transactions;      function addToBlockchain(address payable receiver,string memory user\_id,string memory election\_id,string memory candidate\_id) public {          transactionCounter+=1;          transactions.push(TransactionStruct(msg.sender,receiver,user\_id,election\_id,candidate\_id,block.timestamp));          emit Transfer(msg.sender,receiver,user\_id,election\_id,candidate\_id,block.timestamp);      }      function getAllTransaction() public view returns (TransactionStruct[] memory){          return transactions;      }      function getTransactionCount() public view returns (uint256){          return transactionCounter;      }  } |

###### **Lampiran Source Code Program**

Code program untuk add user

|  |
| --- |
| export const users = {    getUsers: async (req, res) => {      try {        const tmp = await User.find();        return res.status(201).send(tmp);      } catch (e) {        return res.status(500).send("Error");      }    },    getUser: async (req, res) => {      try {        const tmp = await User.findById(req.params.id);        return res.status(201).send(tmp);      } catch (e) {        console.log(e);        return res.status(500).send("Error!");      }    },    getUserByName: async (req, res) => {      try {        const tmp = await User.find({ username: req.params.id });        return res.status(201).send(tmp);      } catch (e) {        console.log(e);        return res.status(500).send("Error!");      }    },    delete: async (req, res) => {      try {        const tmp = await User.findByIdAndDelete(req.params.id);        const isPhotoDeleted = users.deleteUserProfile(tmp);        if (isPhotoDeleted) {          return res            .status(201)            .send("Election and photo file deleted successfully");        } else {          return res.status(500).send("Error deleting photo file");        }      } catch (e) {        console.log(e);        return res.status(500).send("Error!");      }    },    edit: async (req, res) => {      const tmp = await User.findById(req.params.id);      const isPhotoDeleted = users.deleteUserProfile(tmp);      if (!isPhotoDeleted) {        return res.status(500).send("Error updating User");      }      upload(req, res, async function (err) {        if (err instanceof multer.MulterError) {          return res.status(500).json(err);        } else if (err) {          return res.status(500).json(err);        }        try {          const user = {            username: req.body.username,            email: req.body.email,            mobile: req.body.mobile,            fname: req.body.fname,            lname: req.body.lname,          };          const tmp = await User.findByIdAndUpdate(req.params.id, user);          return res.status(201).send("User Updated Successfully");        } catch (e) {          console.log(e);          return res.status(500).send("error");        }      });    },  }; |

Kode ini adalah kumpulan metode untuk mengelola operasi terkait *User*, termasuk mengambil, menghapus, dan mengedit profil user di *database* *MongoDB* dengan menggunakan model *User*. *getUsers(req, res):* Mendapatkan semua user berdasarkan id dari database. Jika berhasil, mengembalikan daftar user dengan status 201. Jika terjadi error, mengembalikan status 500.

Code program untuk add candidate

|  |
| --- |
| export const candidateRegister = {    validator: async (req, res, next) => {      next();    },    controller: async (req, res) => {      const candidate = await Candidate.create({        username: req.body.username,        firstName: req.body.firstName,        lastName: req.body.lastName,        dob: req.body.dob,        qualification: req.body.qualification,        join: req.body.join,        location: req.body.location,        description: req.body.description,      });      return res.status(201).send("Candidate Added");    },  };  export const candidates = {    getCandidates: async (req, res) => {      const data = await Candidate.find();      return res.status(201).send(data);    },    register: async (req, res) => {      const candidate = await Candidate.create({        username: req.body.username,        firstName: req.body.firstName,        lastName: req.body.lastName,        dob: req.body.dob,        qualification: req.body.qualification,        join: req.body.join,        location: req.body.location,        description: req.body.description,      });      return res.status(201).send("Candidate Added");    },    getCandidate: async (req, res) => {      const data = await Candidate.findOne({ username: req.params.username });      if (data == null) {        return res.status(500).send("Candidate Not Found");      }      return res.status(201).send(data);    },    delete: async (req, res) => {      try {        const data = await Candidate.findByIdAndDelete(req.params.id);        return res.status(201).send("Candidate Deleted Successfully");      } catch (e) {        return res.status(500).send("Error");      }    },  }; |

Kode ini merupakan kumpulan fungsi yang digunakan untuk mengelola data kandidat dalam aplikasi. Fungsi-fungsi ini mengelola operasi seperti menambah, mengambil, dan menghapus data kandidat.

Code program untuk election

|  |
| --- |
| export const elections = {    controller: async (req, res) => {      try {        const tmp = await Election.find();        return res.status(201).send(tmp);      } catch (e) {        return res.status(500).send("Error");      }    },    register: async (req, res) => {      try {        const newElection = await Election.create({          name: req.body.name,          candidates: req.body.candidates,        });        return res.status(201).send("Election Successfully Added");      } catch (e) {        return res.status(500).send("Internal Error" + e);      }    },    getElection: async (req, res) => {      try {        const data = await Election.findById(req.params.id);        return res.status(201).send(data);      } catch (e) {        return res.status(500).send("Error");      }    },    voting: async (req, res) => {      try {        const tmp = await Election.find({ currentPhase: "voting" });        return res.status(201).send(tmp);      } catch (e) {        return res.status(500).send("Error");      }    },    result: async (req, res) => {      try {        const tmp = await Election.find({ currentPhase: "result" });        return res.status(201).send(tmp);      } catch (e) {        return res.status(500).send("Error");      }    },    delete: async (req, res) => {      try {        const tmp = await Election.findByIdAndDelete(req.params.id);        return res.status(201).send("Election Deleted Successfully");      } catch (e) {        return res.status(500).send("Error");      }    },  }; |

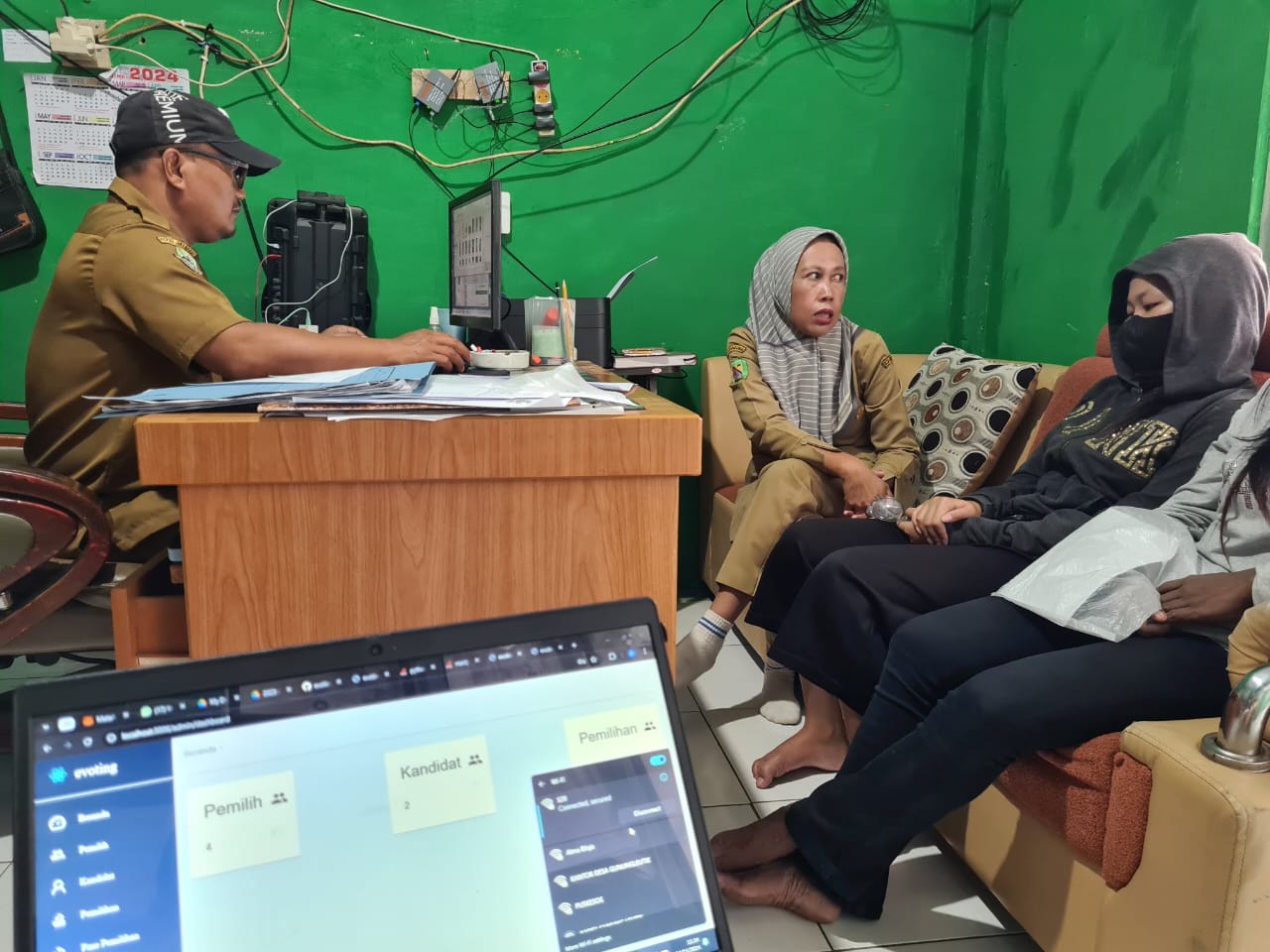
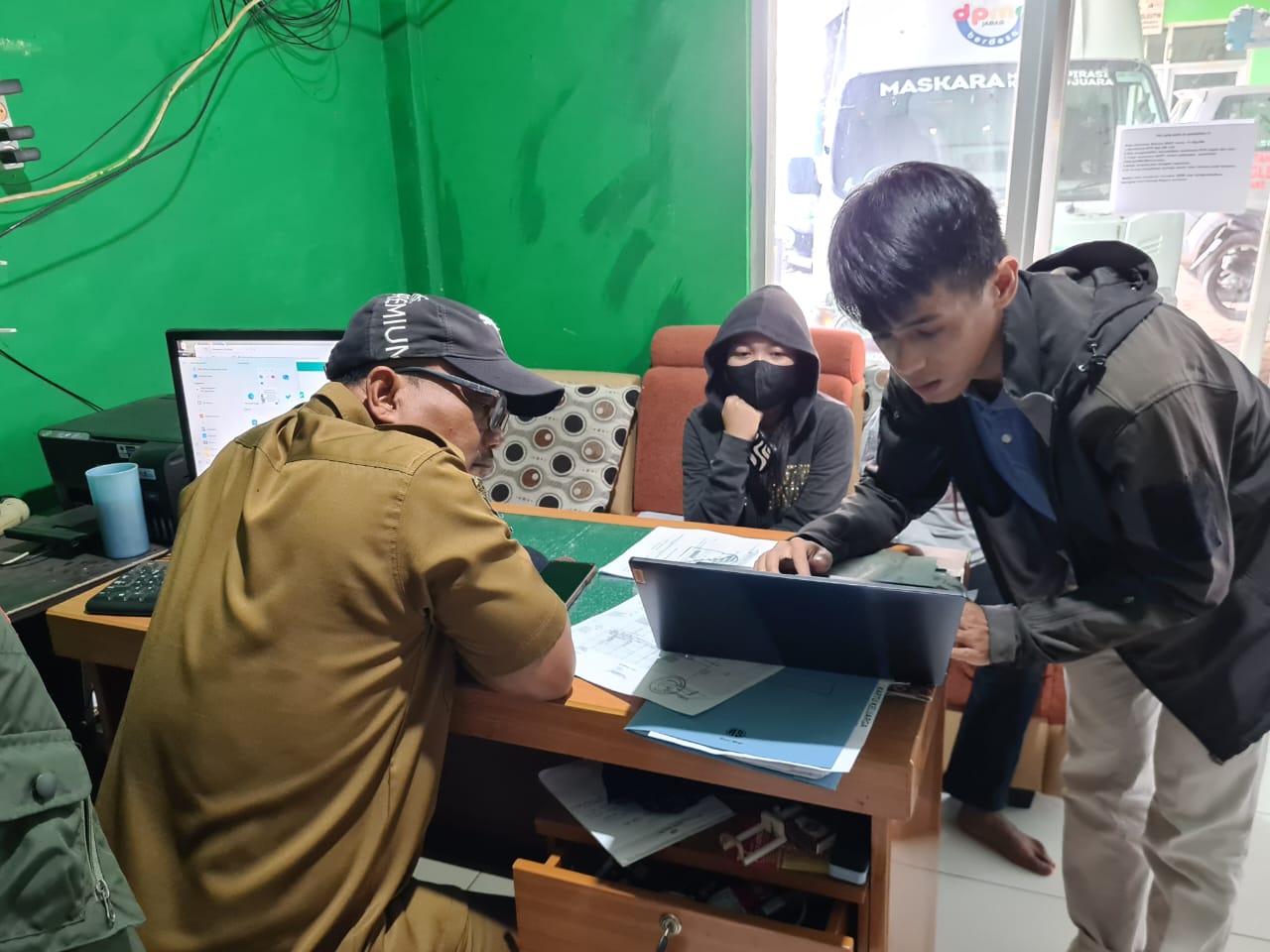
Kode ini berisi fungsi-fungsi untuk mengelola *election,* didalamnya ada fungsi controller untuk mengambil semua data pemilihan yang ada pada database menggunakan *Election.find(),* ada juga *register* untuk mendaftrakan pemilihan baru dengan memasukan judul dan daftar kandidat. fungsi *voting* mengambil semua pemilihan yang berada pada fase *voting*. Fungsi *result* mengambil semua pemilihan yang berada pada fase *result*.

Code program untuk add phase

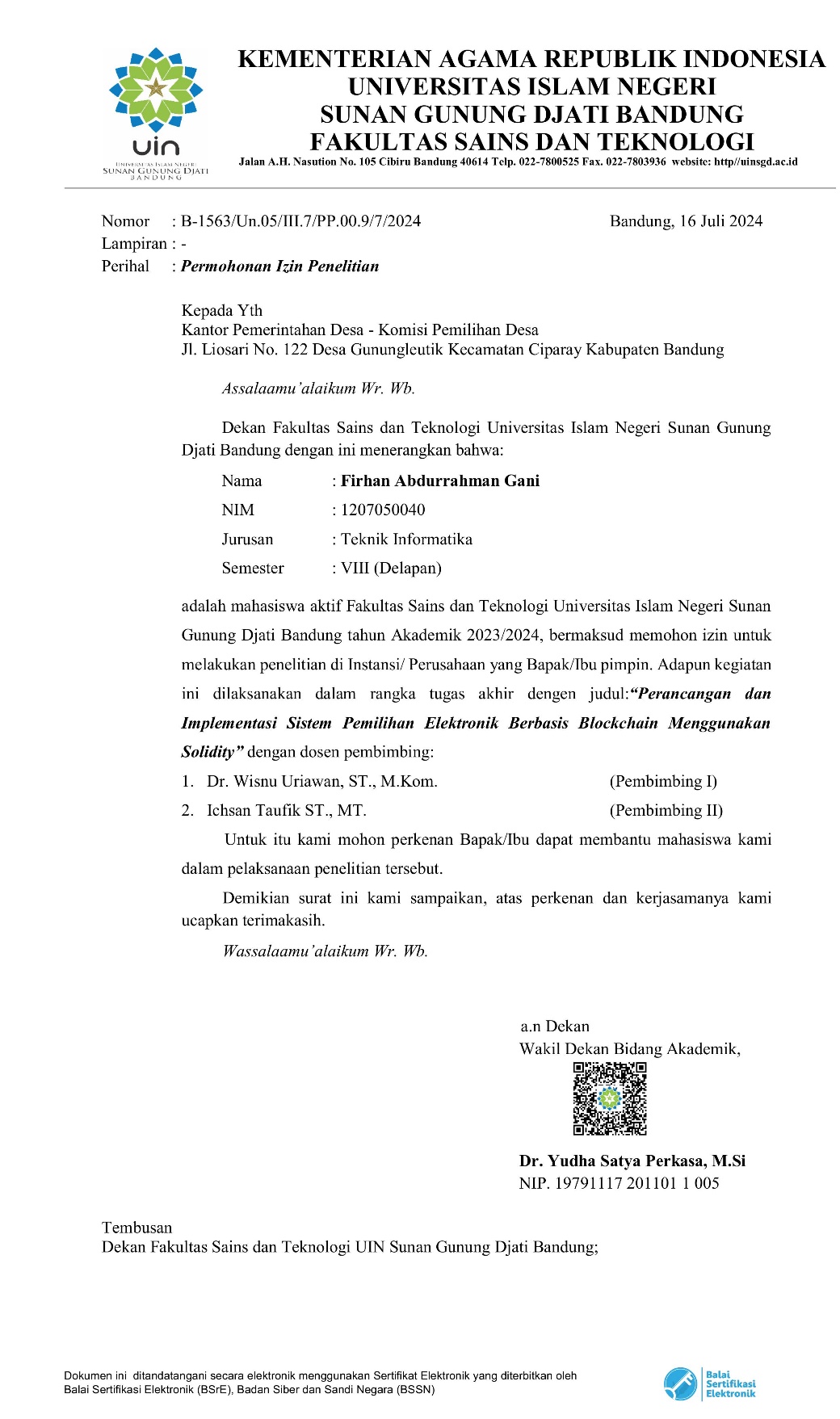
|  |
| --- |
| export const phase = {    controller: async (req, res) => {      const data = await Election.findByIdAndUpdate(req.params.id, {        currentPhase: req.body.currentPhase,      });      return res.status(201).send(data);    },  }; |

Kode ini berfungsi untuk memperbarui fase pemilihan menggunakan currenPhase yang diambil dari req.body.

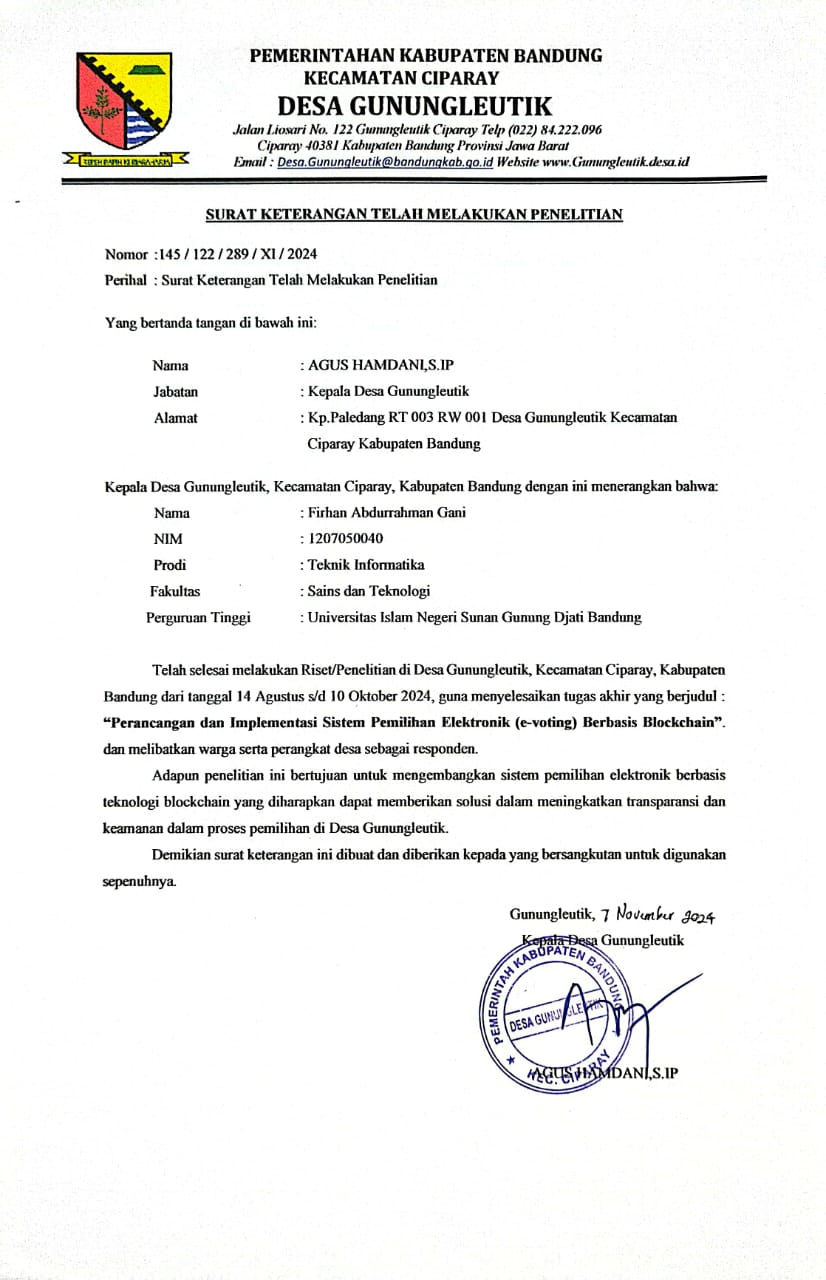
###### **Lampiran Foto Penelitian**



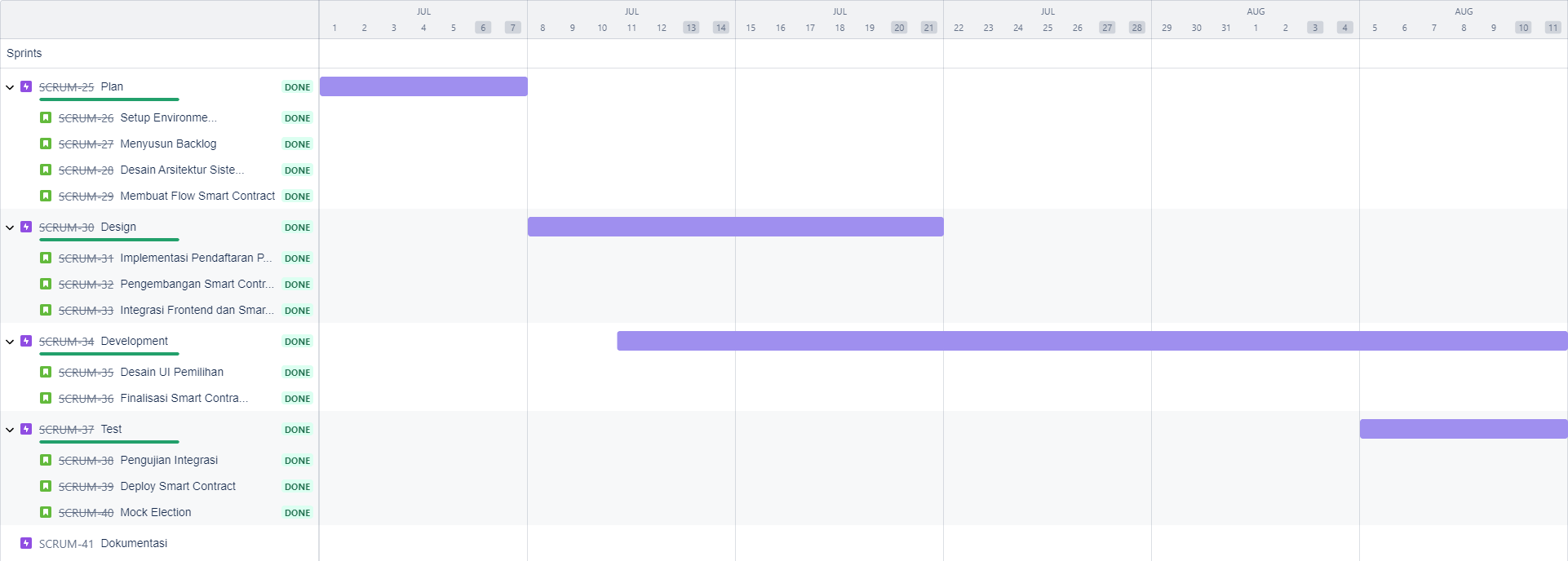
###### **Lampiran Surat Ijin Penelitian Lapangan**



###### **Lampiran SK Telah Menyelesaikan Penelitian**



###### **Lampiran Produk Backlog**



Sprint Planning

|  |  |  |
| --- | --- | --- |
| **Sprint** | **Durasi** | **Tugas Utama** |
| Sprint: Plan | 1 Minggu | Menyelesaikan setup environment, menyusun backlog, desain arsitektur, dan membuat flow smart contract. |
| Sprint: Design | 2 Minggu | Desain UI pemilihan dan finalisasi smart contract. |
| Sprint: Development | 4 Minggu | Implementasi fitur-fitur utama, pengembangan smart contract, dan integrasi frontend-backend. |
| Sprint: Test | 1 Minggu | Melakukan pengujian integrasi, deployment smart contract, dan mock election. |

* + - 1. Sprint Plan. **Deskripsi:** Sprint ini fokus pada perencanaan awal pengembangan sistem, memastikan semua kebutuhan dan spesifikasi teknis jelas.

|  |  |  |
| --- | --- | --- |
| **ID** | **Nama Backlog** | **Deskripsi** |
| SCRUM-26 | Setup Environment | Menyiapkan lingkungan pengembangan seperti penginstalan software (Node.js, Ganache, MetaMask, Truffle). |
| SCRUM-27 | Menyusun Backlog | Membuat daftar backlog produk dengan fitur utama yang diperlukan untuk sistem e-voting. |
| SCRUM-28 | Desain Arsitektur Sistem | Merancang arsitektur sistem, termasuk komponen frontend, backend, dan blockchain. |
| SCRUM-29 | Membuat Flow Smart Contract | Merancang alur kerja smart contract untuk proses pendaftaran, voting, dan penghitungan suara. |

* + - 1. Sprint Design. **Deskripsi:** Sprint ini bertujuan untuk membuat desain teknis dan implementasi awal fitur sistem.

|  |  |  |
| --- | --- | --- |
| **ID** | **Nama Backlog** | **Deskripsi** |
| SCRUM-35 | Desain UI Pemilihan | Membuat antarmuka pengguna yang intuitif dan menarik untuk proses pemilihan suara. |
| SCRUM-36 | Finalisasi Smart Contract | Menyelesaikan pengembangan smart contract, memastikan semua fungsi bekerja sesuai kebutuhan. |

* + - 1. Sprint Develop. Sprint ini mencakup pengembangan sistem secara menyeluruh dan memastikan fungsionalitas utama dapat dijalankan.

|  |  |  |
| --- | --- | --- |
| **ID** | **Nama Backlog** | **Deskripsi** |
| SCRUM-31 | Implementasi Pendaftaran Pemilih | Membuat fitur yang memungkinkan admin mendaftarkan pemilih dengan data valid ke dalam sistem. |
| SCRUM-32 | Pengembangan Smart Contract | Menulis kode smart contract menggunakan Solidity untuk mendukung fitur e-voting. |
| SCRUM-33 | Integrasi Frontend dan Backend | Menghubungkan antarmuka pengguna (frontend) dengan backend untuk menjalankan logika aplikasi. |

* + - 1. Sprint Test. Sprint ini bertujuan untuk menguji sistem untuk memastikan kualitas dan fungsionalitas sebelum digunakan.

|  |  |  |
| --- | --- | --- |
| **ID** | **Nama Backlog** | **Deskripsi** |
| SCRUM-38 | Pengujian Integrasi | Menguji integrasi antara frontend, backend, dan blockchain untuk memastikan sistem berjalan dengan lancar. |
| SCRUM-39 | Deploy Smart Contract | Melakukan deployment smart contract ke jaringan blockchain (contoh: Ethereum Testnet). |
| SCRUM-40 | Mock Election | Melakukan simulasi pemilu untuk menguji proses pendaftaran, voting, dan penghitungan suara. |