# KATA PENGANTAR

# DAFTAR ISI

[KATA PENGANTAR i](#_heading=h.1fob9te)

[DAFTAR ISI ii](#_heading=h.3znysh7)

[DAFTAR GAMBAR iv](#_heading=h.2et92p0)

[DAFTAR TABEL v](#_heading=h.tyjcwt)

[DAFTAR ISTILAH vi](#_heading=h.3dy6vkm)

[DAFTAR LAMPIRAN vii](#_heading=h.1t3h5sf)

[BAB I PENDAHULUAN 8](#_heading=h.4d34og8)

[1.1. Latar Belakang 8](#_heading=h.2s8eyo1)

[1.2. Rumusan Masalah 11](#_heading=h.17dp8vu)

[1.3. Tujuan Penelitian 11](#_heading=h.3rdcrjn)

[1.4. Manfaat Penelitian 11](#_heading=h.26in1rg)

[1.5. Batasan Masalah 12](#_heading=h.35nkun2)

[1.6. Kerangka Pemikiran Penelitian 12](#_heading=h.1ksv4uv)

[1.7. Sistematika Penulisan 13](#_heading=h.2jxsxqh)

[BAB II KAJIAN LITERATUR 15](#_heading=h.z337ya)

[2.1. *State of the Art* 15](#_heading=h.3j2qqm3)

[2.2. Dasar Teori 21](#_heading=h.4i7ojhp)

[2.2.1. Blockchain 21](#_heading=h.2xcytpi)

[2.2.2. Pemilihan Elektronik 23](#_heading=h.1ci93xb)

[2.2.3. Ethereum 24](#_heading=h.2bn6wsx)

[2.2.4. *Smart Contract* 26](#_heading=h.qsh70q)

[2.2.5. *Solidity* 27](#_heading=h.1pxezwc)

[BAB III METODOLOGI PENELITIAN 29](#_heading=h.49x2ik5)

[3.1. *Plan* (Perencanaan) 29](#_heading=h.2p2csry)

[3.2. *Design* (Desain) 29](#_heading=h.147n2zr)

[3.3. *Develop* (Pengembangan) 30](#_heading=h.3o7alnk)

[3.4. *Test* 30](#_heading=h.23ckvvd)

[3.5. *Deploy* 30](#_heading=h.ihv636)

[BAB IV HASIL DAN PEMBAHASAN 31](#_heading=h.32hioqz)

[4.1. *Plan* (Perencanaan) 31](#_heading=h.1hmsyys)

[4.1.1. Kebutuhan Fungsional 32](#_heading=h.2grqrue)

[4.1.2. Kebutuhan Non-Fungsional 33](#_heading=h.vx1227)

[4.2. *Design* (Desain) 33](#_heading=h.4f1mdlm)

[3.2.1. Arsitektur Sistem 33](#_heading=h.3tbugp1)

[3.2.1. *Class Diagram* 35](#_heading=h.2lwamvv)

[3.2.1. *Use Case Diagram* 36](#_heading=h.2nusc19)

[3.2.1. *Sequence Diagram* 37](#_heading=h.1302m92)

[3.2.1. *Data Flow Diagram* 38](#_heading=h.3mzq4wv)

[4.3. *Development* (Pengembangan) 39](#_heading=h.2250f4o)

[4.3.1. *Smart Contract* 39](#_heading=h.haapch)

[4.3.2. Pengembangan Antarmuka 41](#_heading=h.319y80a)

[4.4. *Testing* (Pengujian) 44](#_heading=h.1gf8i83)

[4.4.1. Pengujian *Smart Contract* 44](#_heading=h.40ew0vw)

[4.4.2. *Mock Election* 46](#_heading=h.2fk6b3p)

[BAB V KESIMPULAN DAN SARAN 52](#_heading=h.upglbi)

[4.3. Kesimpulan 52](#_heading=h.3ep43zb)

[4.3. Saran 52](#_heading=h.1tuee74)

[DAFTAR PUSTAKA 53](#_heading=h.4du1wux)

# DAFTAR GAMBAR

[Gambar 1.1. Kerangka Pemikiran Penelitian 13](#_heading=)

[Gambar 2.1. Struktur Blok dalam sebuah Blockchain 23](#_heading=h.1y810tw)

[Gambar 2.2. Struktur Smart Contract 26](#_heading=h.3whwml4)

[Gambar 3.1. Metode Agile 29](#_heading=)

[Gambar 4.1. Arsitektur Sistem 34](#_heading=)

[Gambar 4.2 Class Diagram 36](#_heading=)

[Gambar 4.3 Use Case Diagram 37](#_heading=)

[Gambar 4.4 Sequence Diagram 38](#_heading=)

[Gambar 4.5 DFD Level 0 39](#_heading=)

[Gambar 4.6 DFD Level 1 39](#_heading=)

[Gambar 4.7 Smart Contract 40](#_heading=)

[Gambar 4.8 Rancangan Antarmuka Home 41](#_heading=)

[Gambar 4.9 Rancangan Antarmuka Admin-Login 42](#_heading=)

[Gambar 4.10 Rancangan Antarmuka Admin-Dashboard 42](#_heading=)

[Gambar 4.11 Rancangan Antarmuka Admin-Candidate 42](#_heading=)

[Gambar 4.12 Rancangan Antarmuka Admin-Usermng 43](#_heading=)

[Gambar 4.13 Rancangan Antarmuka User-Dashboard 44](#_heading=)

[Gambar 4.14 Rancangan Antarmuka User-Vote 45](#_heading=)

[Gambar 4.15 Ganache Server Detail 45](#_heading=)

[Gambar 4.16 Konfigurasi Metamask 46](#_heading=)

[Gambar 4.17 Truffle Migrate 47](#_heading=)

[Gambar 4.18 Admin-dahsboard 47](#_heading=)

[Gambar 4.20 Add User 48](#_heading=)

[Gambar 4.19 Add Candidate 48](#_heading=)

[Gambar 4.21 Add Election 49](#_heading=)

[Gambar 4.22 Add Phase 49](#_heading=)

[Gambar 4.23 Election 50](#_heading=)

[Gambar 4.24 Election-detail 50](#_heading=)

[Gambar 4.25 User Identification 51](#_heading=)

[Gambar 4.26 Metamask Confirmation 51](#_heading=)

[Gambar 4.27 Result 52](#_heading=)

# DAFTAR TABEL

[Tabel 2.1 State of the Art 20](#_heading=h.44sinio)

[Tabel 4.1. Kebutuhan Fungsional 33](#_heading=)

[Tabel 4.2. Kebutuhan Non-Fungsional 33](#_heading=)

# DAFTAR ISTILAH

# DAFTAR LAMPIRAN

# BAB I PENDAHULUAN

## 1.1. Latar Belakang

Pemilu merupakan kesempatan bagi masyarakat Indonesia untuk memilih pemimpin dan wakil nasional pilihannya. Pemilu di Indonesia masih menggunakan metode tradisional yaitu pemungutan suara di atas kertas dengan mendatangi tempat pemungutan suara yang telah ditentukan 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,
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 hambatan dalam pelaksanaannya, proses pemilu tidak berhasil serta kurang efektif dan efisien. Di sisi lain, perkembangan teknologi telah membawa banyak manfaat besar karena menawarkan berbagai kemudahan baik dari segi akses maupun biaya. Penggunaan teknologi informasi dalam penyelenggaraan pemilu akan meningkatkan efektivitas dan efisiensi penerapan transisi ke model pemungutan suara elektronik tradisional. Saat ini, teknologi pemilihan elektronik (*e-voting)* dapat menjadi pilihan yang tepat untuk menyelenggarakan pemilu, salah satu pilar terpenting demokrasi. Selain itu, penerapan metode tradisional yang telah memiliki kelemahan [1].

Beberapa negara maju dan berkembang juga menerapkan teknologi pemilihan elektronik (*e-voting)*, seperti India, pemilu India adalah sebuah perhelatan demokrasi paling besar di dunia, pada hari pemilu, setiap pemilih harus membawa kartu pemilih, setelah diperiksa oleh panitia pemungutan suara, dan sesuai dengan data daftar pemilih tetap, pemilih diizinkan untuk memilih pada mesin elektronik yang disediakan [2]. Selain di India, beberapa negara yang sudah menerapkan sesuai dengan *e-voting map* 2015 yang dikeluarkan *Competence Center for Electronic Voting and Participation* di antaranya, kategori legally binding electronic voting with voting machines yaitu: Australia, Brazil, Kanada, Prancis, India, Jepang, Kazakhstan, Peru, Rusia, Amerika, Arab Saudi, dan Venezuela, kategori legally binding internet voting yaitu: Austria, Australia, Kanada, Estonia, Prancis Jepang, dan Swiss, kategori *Planning trials*, *non-legally binding e-voting* yaitu: Argentina, Azerbaijan, Belarus, Bulgaria, Chile, Republik Czech, Finlandia, Greece, Italia, Latvia, Lithuania, Mexico, Nepal, Nigeria, Norwegia, Polandia, Portugal, Romania, Slovakia, Slovenia, Afrika Selatan, Spanyol, Korea Selatan, Swedia [2].

*Blockchain* adalah teknologi yang memiliki potensi untuk mengatasi masalah keamanan, integritas, dan transparansi dalam sistem pemilihan elektronik. *Blockchain* adalah suatu sistem distribusi data yang dapat dipercayai yang memungkinkan transaksi dan data secara distribusi melalui jaringan komputer yang terhubung. *Blockchain* memungkinkan transaksi yang tidak dapat diubah dan tidak dapat dihapus, serta memungkinkan transaksi yang dilakukan oleh satu pihak dapat dilihat oleh semua pihak [5]. Dalam penelitian yang dilakukan oleh Gavin Wood (2014), 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 [3]. 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* (Ernest, 2014) yang dapat digunakan dalam mendesain sistem yaitu [4]:

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.

Pemilihan elektronik adalah salah satu cara yang digunakan untuk mempercepat dan mempermudah proses pemilihan. Namun, keberhasilan sistem pemilihan elektronik tergantung pada keamanan, integritas, dan transparansi sistem. Kebanyakan sistem pemilihan elektronik yang ada saat ini menggunakan arsitektur *client-server*, yang mungkin menjadi sumber dari kegagalan sistem pemilihan elektronik. Penelitian ini akan mencakup perancangan dan implementasi sistem pemilihan elektronik berbasis *blockchain* menggunakan *Solidity*. Tujuan dari penelitian ini adalah untuk mengembangkan sistem pemilihan elektronik yang aman, terintegrasi, dan transparan. Penelitian ini akan mencakup perancangan sistem, implementasi sistem, dan evaluasi kinerja sistem pemilihan elektronik berbasis blockchain menggunakan Solidity.

## 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 mengatasi pemilihan tradisional yang rawan dimanipulasi?
2. Bagaimana mendesain dan mengimplementasikan sistem pemilihan elektronik berbasis *blockchain* menggunakan *solidity*?

## 1.3. Tujuan Penelitian

Tujuan dari pembuatan sistem ini adalah sebagai berikut:

1. Merancang dan mengimplementasikan sistem pemilihan elektronik yang aman dan transparan berbasis teknologi *blockchain* menggunakan *solidity*.
2. Memberikan data pemilihan elektronik yang aman, akurat, transparan, otonom, anonim, adil dan efisien.

## 1.4. Manfaat Penelitian

Berdasarkan tujuan diatas maka manfaat dari penelitian ini adalah sebagai berikut:

1. Dapat mengimplementasikan sistem pemilihan elektronik berbasis *blockchain* dengan keamanan yang baik.
2. Dengan melalui jaringan *blockchain* aman, akurat, transparan, otonom, anonim, adil dan efisien, diharapkan akan meningkatkan kepercayaan umum terhadap data yang tersimpan karena setiap server dan bahkan semua orang pun dapat ikut terhubung ke dalam jaringan lalu menyimpan salinan dari *blockchain* secara utuh serta dapat dengan mudah untuk melakukan audit terhadap data untuk menentukan mana data yang valid dan telah diverifikasi. Selain meningkatkan keamanan dalam penyimpanan data, dari sifat *blockchain* sendiri menawarkan keamanan, akurat, transparan, otonom, anonim, adil dan efisien. di mana sifat ini dapat disesuaikan dengan asas pemilu di Indonesia yaitu langsung, umum, bebas, rahasia, jujur dan adil.

## 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.

## 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*

Dalam penulisan penelitian ini penulis menggali informasi dari beberapa penelitian sebelumnya sebagai bahan perbandingan, baik mengenai kekurangan atau kelebihan yang sudah ada. Selain itu, penulis juga menggali informasi dari buku-buku, jurnal ilmiah maupun skripsi dalam rangka mendapatkan suatu informasi yang ada sebelumnya tentang teori yang berkaitan dengan judul yang digunakan untuk memperoleh landasan teori ilmiah. 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, PhD (2018) [0] 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. Penulis mengeksplorasi penggunaan teknologi *blockchain* untuk mengatasi tantangan-tantangan ini, dengan fokus pada potensinya untuk pemungutan suara yang aman, dapat diandalkan, dan *anonim*. Poin-poin penting dari penelitian ini meliputi:
  1. Membangun sebuah sistem pemungutan suara elektronik yang aman itu sulit, seperti yang ditunjukkan oleh sistem pemungutan suara online yang ditinggalkan oleh Pentagon AS.
  2. Teknologi blockchain berpotensi untuk menjawab tantangan keamanan dan keandalan dalam sistem pemungutan suara elektronik.
  3. Penelitian ini membahas berbagai penelitian dan implementasi sistem pemungutan suara elektronik berbasis *blockchain*, seperti *Bitcoin*, *Ethereum*, *Hyperledger Fabric*, dan *Hyperledger Sawtooth*.
  4. Penelitian ini juga membahas masalah yang berkaitan dengan skalabilitas, privasi, dan keamanan dalam sistem pemungutan suara elektronik berbasis blockchain.
     1. Uzma Jafar, Mohd Juzaiddin Ab Aziz, dan Zarina Shukur (2021) [0] 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. Para penulis mendiskusikan manfaat potensial dari sistem *e-voting* berbasis *blockchain*, seperti keamanan, integritas, dan transparansi, serta tantangan yang perlu diatasi untuk implementasi yang sukses. Poin-poin penting dari penelitian tersebut meliputi:

1. Pemungutan suara online semakin populer karena potensinya untuk mengurangi biaya penyelenggaraan dan meningkatkan jumlah pemilih.
2. Teknologi *blockchain* dapat meningkatkan keamanan dan integritas sistem pemungutan suara elektronik.
3. Penelitian ini membahas berbagai penelitian dan implementasi sistem *e-voting* berbasis *blockchain*, seperti *DecentraVote*, Sistem *E-Voting* Berbasis *Blockchain*, dan *framework* Int & Inter.
4. Para penulis menyoroti manfaat dari sistem *e-voting* berbasis *blockchain*, termasuk keamanan, integritas, dan transparansi.
5. Penelitian ini juga membahas tantangan yang dihadapi dalam mengimplementasikan sistem *e-voting* berbasis *blockchain*, seperti skalabilitas, privasi, dan kecepatan transaksi.

Para penulis 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*.

* + 1. Basit Shahzad dan Jon Crowcroft (2019) [0] 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. Para penulis mengusulkan sebuah *framework* yang menjawab tantangan sistem *e-voting* berbasis *blockchain*, seperti skalabilitas, privasi, dan kecepatan transaksi. Poin-poin penting dari penelitian ini meliputi:

1. Para penulis 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.
2. *Framework* yang diusulkan berfokus pada keefektifan proses pemungutan suara, algoritma *hashing*, pembuatan dan penyegelan blok, akumulasi data, dan banyak lagi.
3. Para penulis menyoroti manfaat dari sistem *e-voting* berbasis *blockchain*, termasuk keamanan, integritas, dan transparansi.
4. Penelitian ini juga membahas tantangan yang dihadapi dalam mengimplementasikan sistem *e-voting* berbasis *blockchain*, seperti skalabilitas, privasi, dan kecepatan transaksi.

Para penulis 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. Para penulis 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.

* + 1. Setiawan Restu Aji, Wahyuningdiah Trisari Harsanti Putri (2023) [0] "Implementasi Teknologi *Blockchain* dalam Aplikasi *E-Voting* Berbasis *Mobile*" oleh membahas tentang implementasi teknologi *blockchain* dalam sistem pemungutan suara elektronik berbasis *mobile*. Para penulis berfokus pada penggunaan *blockchain* untuk mengatasi masalah keamanan dan integritas pada aplikasi *e-voting* dan pengembangan aplikasi *e-voting* berbasis *mobile* untuk meningkatkan aksesibilitas. Poin-poin penting dari penelitian ini meliputi:

1. Para penulis mendiskusikan tantangan dalam membangun sistem pemungutan suara elektronik yang aman.
2. Solusi yang diusulkan berfokus pada penggunaan teknologi blockchain untuk meningkatkan keamanan dan integritas sistem pemungutan suara elektronik.
3. Penelitian ini menyoroti manfaat dari sistem e-voting berbasis blockchain, termasuk keamanan, integritas, dan transparansi.
4. Para penulis membahas tantangan yang dihadapi dalam mengimplementasikan sistem e-voting berbasis blockchain, seperti skalabilitas, privasi, dan kecepatan transaksi.
   * 1. Fadli Fajri, Singgih Jatmiko, dan Missa Lamsani (2020) [0] “*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. Para penulis 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.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Judul** | **Penulis** | **Tahun** | **Hasil** |
| 1 | *E-Voting on the Blockchain* | Kevin Curran, PhD | 2018 | Penulis 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 | Para penulis 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 | Para penulis membahas keefektifan penggunaan teknologi *blockchain* untuk meningkatkan keamanan dan integritas sistem pemungutan suara elektronik. Para penulis mengusulkan sebuah *framework* yang menjawab tantangan sistem *e-voting* berbasis *blockchain*, seperti skalabilitas, privasi, dan kecepatan transaksi. |
| 4 | Implementasi Teknologi *Blockchain* dalam Aplikasi *E-Voting* Berbasis *Mobile* | Setiawan Restu Aji, Wahyuningdiah Trisari Harsanti Putri | 2023 | Para penulis membahas tentang implementasi teknologi *blockchain* dalam sistem pemungutan suara elektronik berbasis *mobile*. Para penulis 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 | Para penulis mengusulkan sebuah sistem terdesentralisasi yang menggunakan blockchain Ethereum untuk memastikan keamanan, integritas, dan transparansi dalam proses pemungutan suara. |

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

## 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 [7]. (Nasir Mumtaz, 2021).

*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 (Zheng et al., 2017) [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 (Lewis, 2015). Dalam teknologi blockchain, blockchain dapat dibagi menjadi 3 jenis yaitu (Purwantoro & Wibowo, 2020) [8]:

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 (Purwantoro & Wibowo, 2020) [9].

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

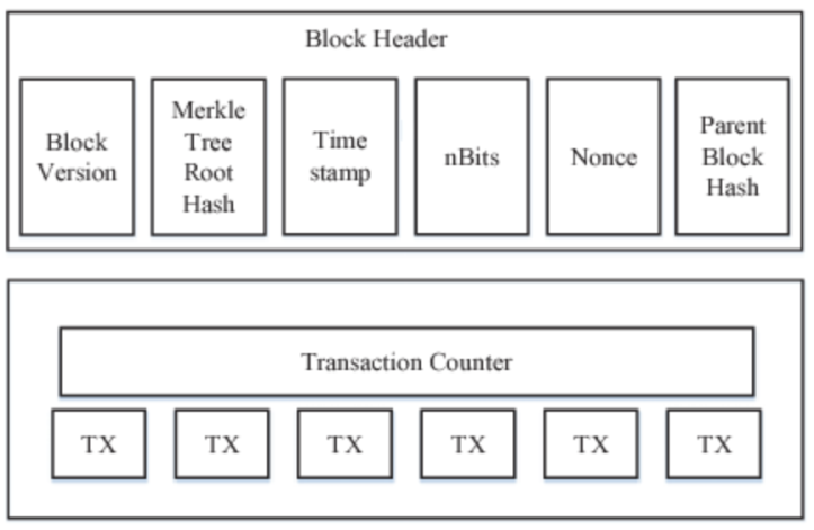
*Blockchain* adalah urutan blok, yang menyimpan daftar lengkap catatan transaksi seperti buku besar publik konvensional (Lee Kuo Chuen, 2015). Dalam sebuah blok yang terdapat di blockchain terdapat dua bagian yaitu (Zheng et al., 2017) [8]:

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*

### 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 (M.Ilyas Gultom & Saripurna, 2019) [10]. 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*) (Hu et al., 2019) . *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 (Ben Ayed, 2017) [11].

### 2.2.3. Ethereum

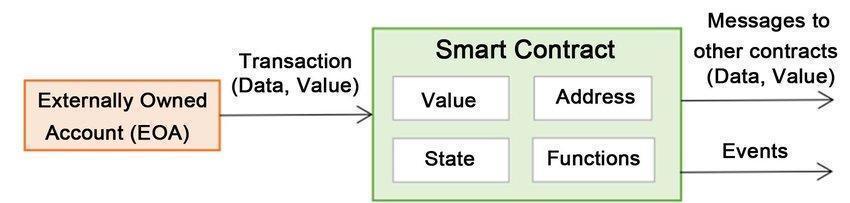
*Ethereum* adalah *platform* perangkat lunak terdesentralisasi yang berjalan di atas teknologi *blockchain*. Diciptakan oleh *Vitalik Buterin* pada tahun 2013 dan diluncurkan pada tahun 2015, *Ethereum* memungkinkan pengembang untuk membangun dan menjalankan aplikasi terdesentralisasi (*DApps*) serta 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 [12]. *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 immutability dari transaksi. Smart contract dapat digunakan dalam berbagai aplikasi, seperti sistem pemilihan elektronik, sistem keuangan, dan sistem logistik.



#### *Gambar 2.2. Struktur Smart Contract*

*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 [13] :

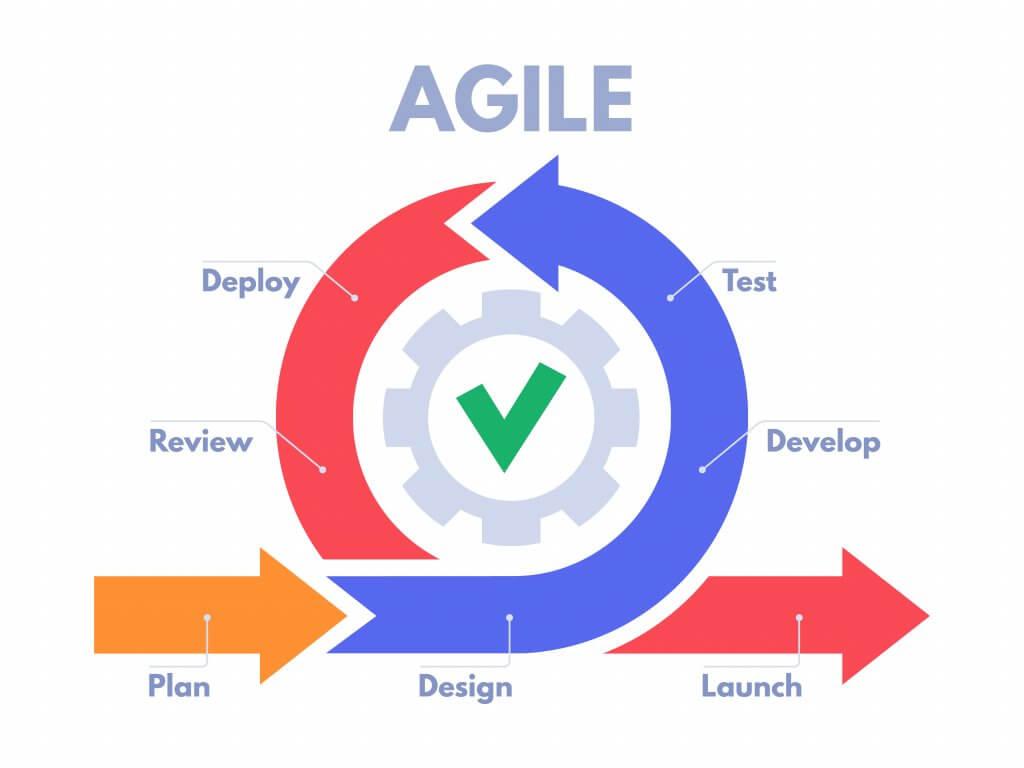
1. *Supply chain*. Dalam *supply chain*, *smart contract* digunakan untuk membuat sistem menjadi otomatis, aman dan juga transparan.
2. *Internet of Things* (IoT). Penerapan teknologi smart contract pada IoT yaitu untuk membuat sistem menjadi otomatis.
3. *Healthcare system*. Penerapan *smart contract* 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. (Mohanta et al., 2018).

### 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 kontrak smart 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[14].

# BAB III METODOLOGI PENELITIAN

Metode penelitian yang digunakan dalam mengimplementasikan Sistem Pemilihan Elektronik Berbasis Blockchain Menggunakan Solidity yaitu *System Development Life Cycle* (SDLC) yang bertujuan untuk merancang, mengembangkan, dan mengimplementasikan sistem pemilihan elektronik berbasis blockchain. Penelitian ini menggunakan metodologi *Agile* untuk mengembangkan sistem *e-voting* berbasis *blockchain* dengan *Solidity*. Model *Agile* dipilih karena sifatnya yang fleksibel, memungkinkan pengembangan sistem dilakukan dalam beberapa tahapan yang dapat disesuaikan.:



#### *Gambar 3.1. Metode Agile*

## 3.1. *Plan* (Perencanaan)

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*.

## 3.2. *Design* (Desain)

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.

## 3.3. *Develop* (Pengembangan)

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.

## 3.4. *Test*

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

## 3.5. *Deploy*

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.

# BAB IV HASIL DAN PEMBAHASAN

## 4.1. *Plan* (Perencanaan)

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.

* + - 1. Analisis Berjalan

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.

* + - 1. Analisisi Usulan

Berdasarkan permasalahan yang diangkat, maka blockchain dapat menjadi solusi dalam pembuatan sistem pemilihan elektronik yang dapat membuat sistem pemilihan menjadi lebih aman *blockchain*, 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 memberikan data pemilihan yang aman, akurat, transparan, otonom, anonim, adil, dan efisien. 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. Hal ini mencakup sistem yang akan dibuat, analisis kebutuhan fungsional dan non-fungsional.

### 4.1.1. Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan yang berisi tentang proses apa saja yang akan dilakukan oleh sistem. Kebutuhan fungsional jugan berisi tentang informasi apa saja yang harus ada dan dihasilkan oleh sistem. Berikut merupakan kebutuhan fungsional dari perancangan sistem yang akan dibuat sebagai berikut:

|  |  |
| --- | --- |
| Kode | Deskripsi |
| KF-01 | Sistem dapat menampilkan kandidat pemilihan. |
| KF-02 | Sistem dapat menampilkan sesi pemlihan yang berlangsung. |
| KF-03 | Sistem dapat menampilkan hasil pemilihan. |
| KF-04 | Admin dapat mendaftarkan *user*. |
| KF-05 | Admin dapat mendaftarkan kandidat. |
| KF-06 | Admin dapat membuat sesi pemilihan. |
| KF-07 | *User* dapat melakukan *voting.* |
| KF-08 | *User* dapat melihat pemilihan yang berlangsung. |
| KF-09 | *User* dapat melihat hasil pemilihan yang berlangsung. |
| KF-10 | *User/Voter* bersifat *anonym.* |

##### Tabel 4.1. Kebutuhan Fungsional

### 4.1.2. Kebutuhan Non-Fungsional

Kebutuhan non-fungsional merupakan

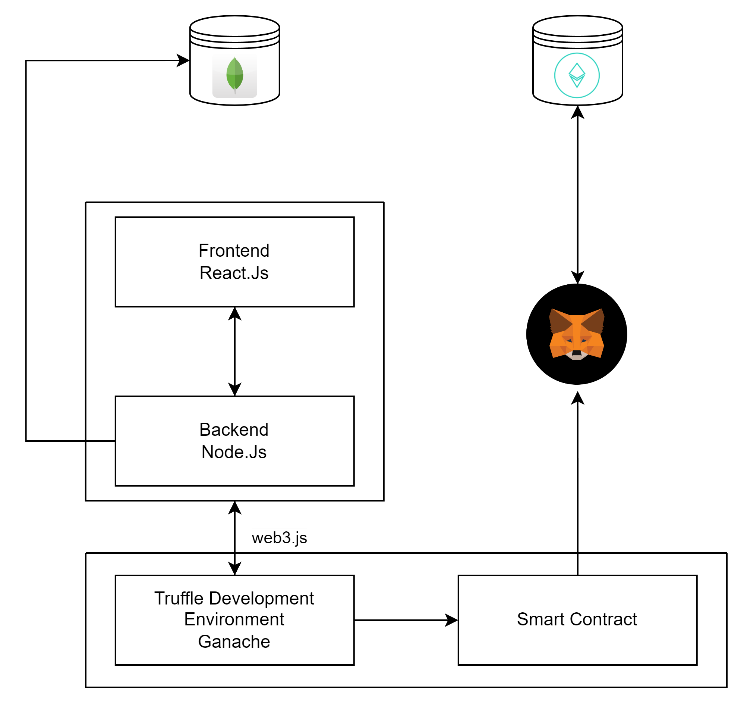
|  |  |
| --- | --- |
| Kode | Deskripsi |
| KNF-01 | Web memiliki antarmuka yang *user friendly.* |
| KNF-02 | Hanya admin yang punya wewenang untuk mendaftarkan pemilih, kandidat, dan sesi pemilihan |
| KNF-03 | Hanya admin yang dapat mengubah sesi pemilihan. |
| KNF-04 | Data-data yang ditampilkan harus bersifat benar adanya. |

##### Tabel 4.2. Kebutuhan Non-Fungsional

## 4.2. *Design* (Desain)

### 4.2.1. Arsitektur Sistem

Arsitektur Sistem Pemilihan Elektronik Berbasis *Blockchain* menggunakan *Solidity* 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. *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*:

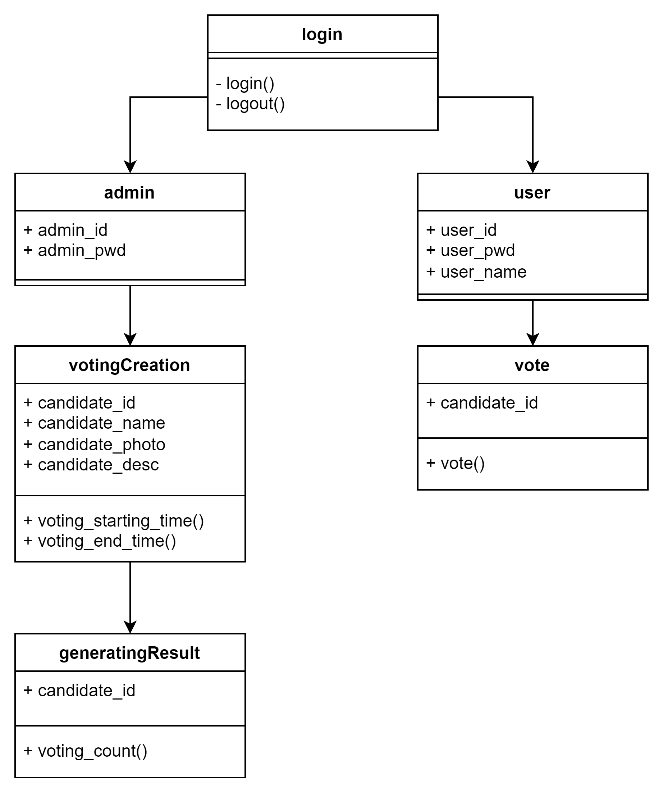


#### *Gambar 4.1. Arsitektur Sistem*

* + - 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. *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.
      7. *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*.
      8. *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. *Class Diagram*

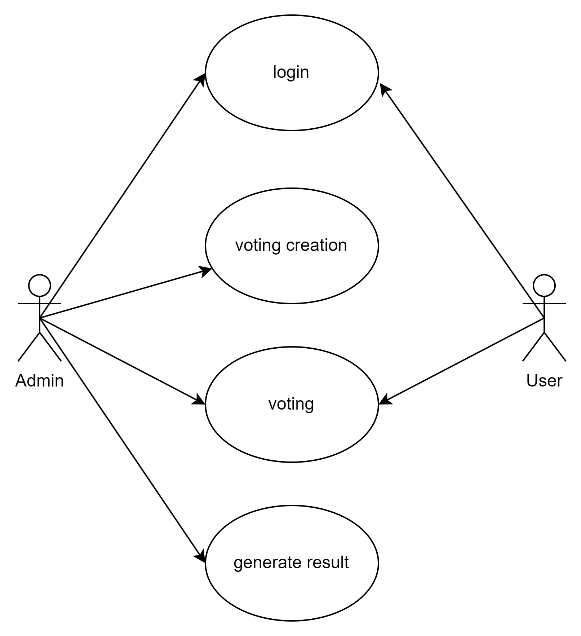
*Class diagram* atau diagram kelas adalah salah satu jenis diagram struktur pada *UML* yang menggambarkan dengan jelas struktur serta deskripsi *class*, atribut, metode, dan hubungan dari setiap objek. *Class diagram* ini bersifat statis, dalam artian diagram ini bukan menjelaskan apa yang terjadi jika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



#### Gambar 4.2 *Class Diagram*

### 4.2.3. *Use Case Diagram*

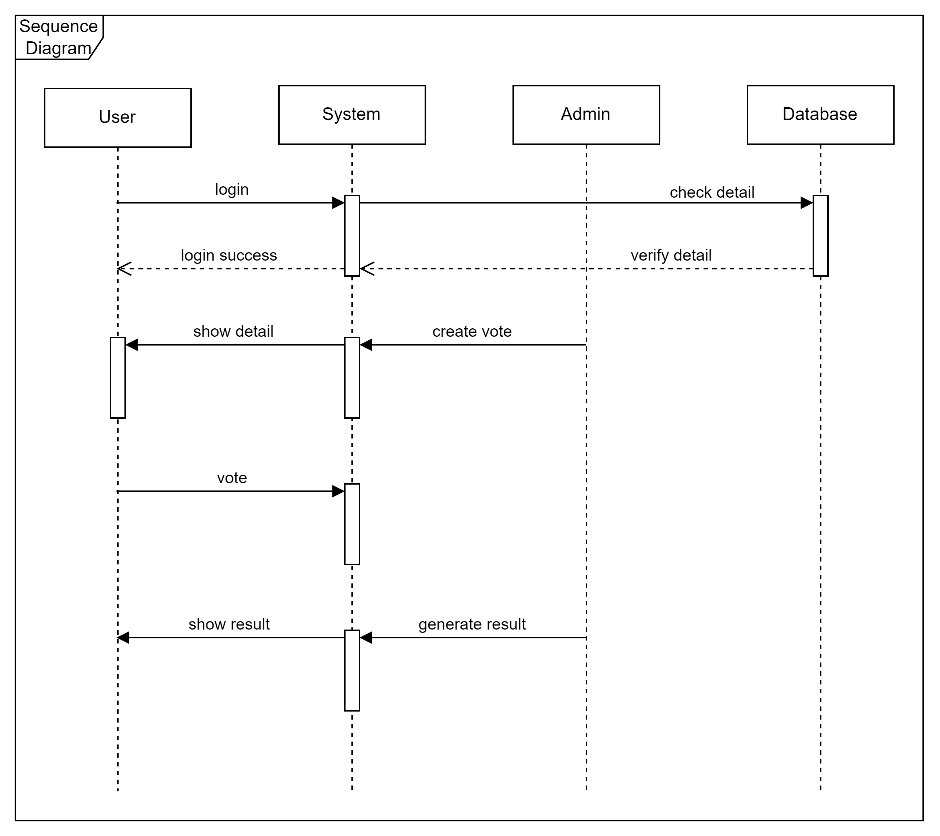
*Use case diagram* adalah cara umum untuk mengomunikasikan fungsi-fungsi utama sistem perangkat lunak. Use-case diagram yang paling sederhana adalah representasi interaksi pengguna dengan sistem yang menunjukkan hubungan antara pengguna dan berbagai kasus penggunaan di mana pengguna terlibat. *Use-case 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.



#### Gambar 4.*3 Use Case Diagram*

### 4.2.4. *Sequence Diagram*

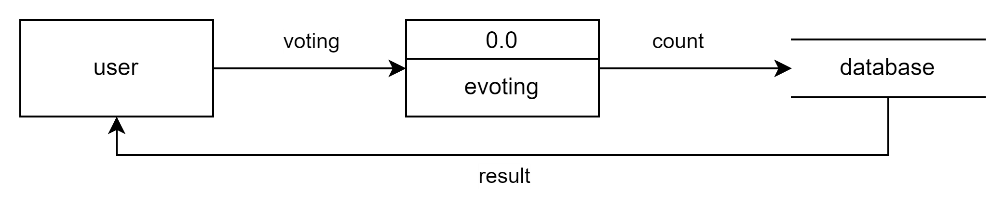
*Sequence diagram* atau diagram urutan adalah sebuah diagram yang digunakan untuk menjelaskan dan menampilkan interaksi antar objek-objek dalam sebuah sistem secara terperinci. Selain itu *sequence diagram* juga akan menampilkan pesan atau perintah yang dikirim, beserta waktu pelaksanaannya. Tujuan utama 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 4.4 *Sequence Diagram*

### 4.2.5. *Data Flow Diagram*

*DFD* memberikan gambaran fungsional dari sebuah sistem. Representasi grafis dengan mudah mengatasi kesenjangan antara 'pengguna dan analis sistem' dan 'analis dan perancang sistem' dalam memahami sebuah sistem. Dimulai dari gambaran umum sistem, *DFD* mengeksplorasi desain rinci sebuah sistem melalui sebuah hirarki.



#### Gambar 4.5 *DFD* Level 0

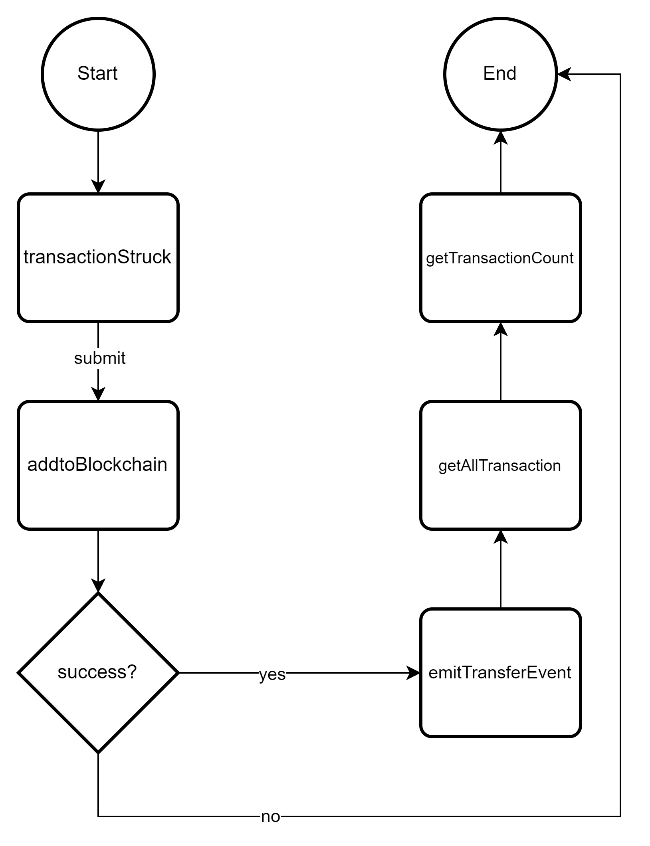
#### 

#### Gambar 4.6 *DFD* Level 1

## 4.3. *Development* (Pengembangan)

### 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 dan 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).

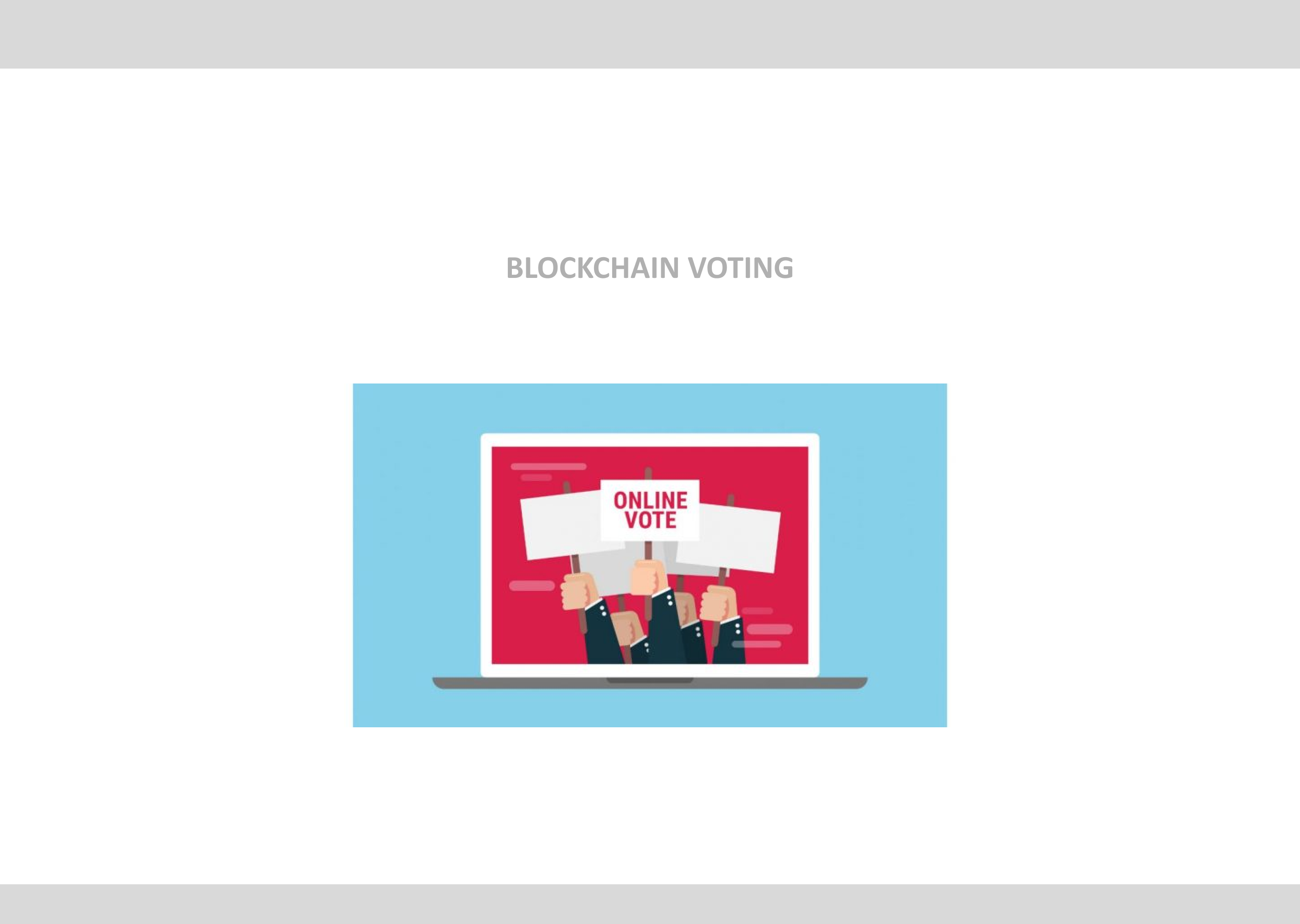


#### Gambar 4.7 *Smart Contract*

Berdasarkan gambar 4.7 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.

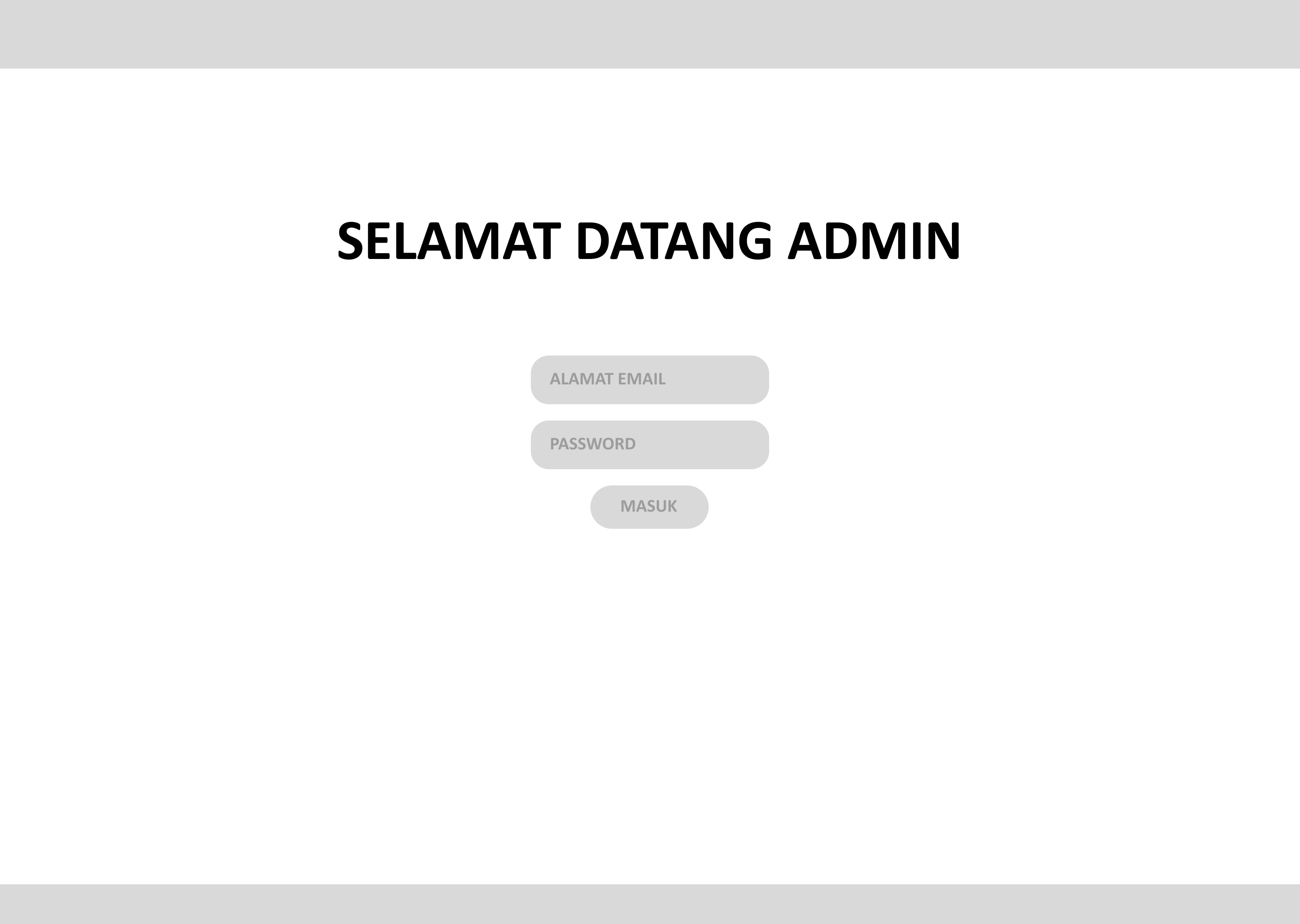
### 4.3.2. Pengembangan Antarmuka

1. Halaman *Home*



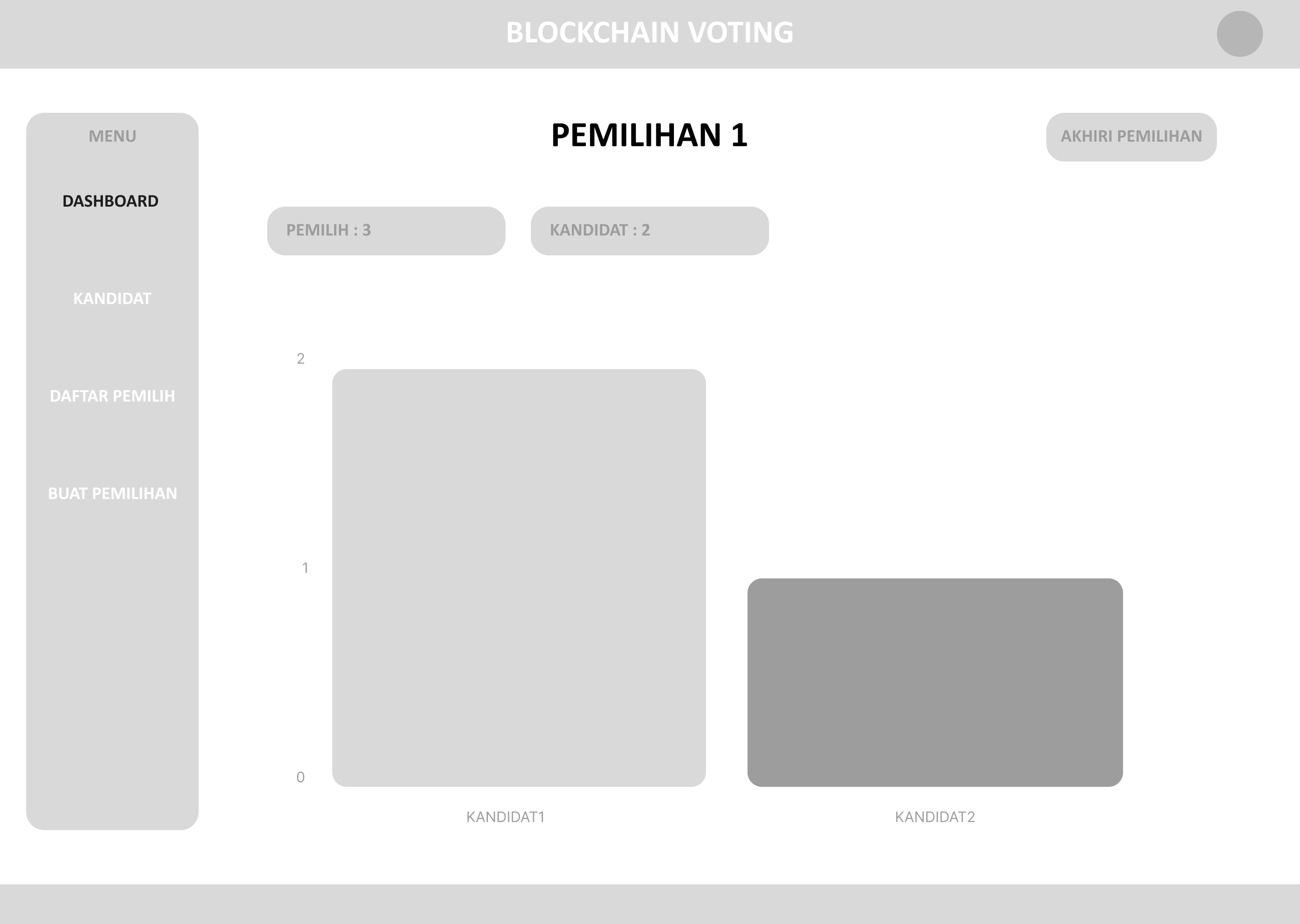
#### Gambar 4.8 Rancangan Antarmuka *Home*

1. Halaman Admin-Login



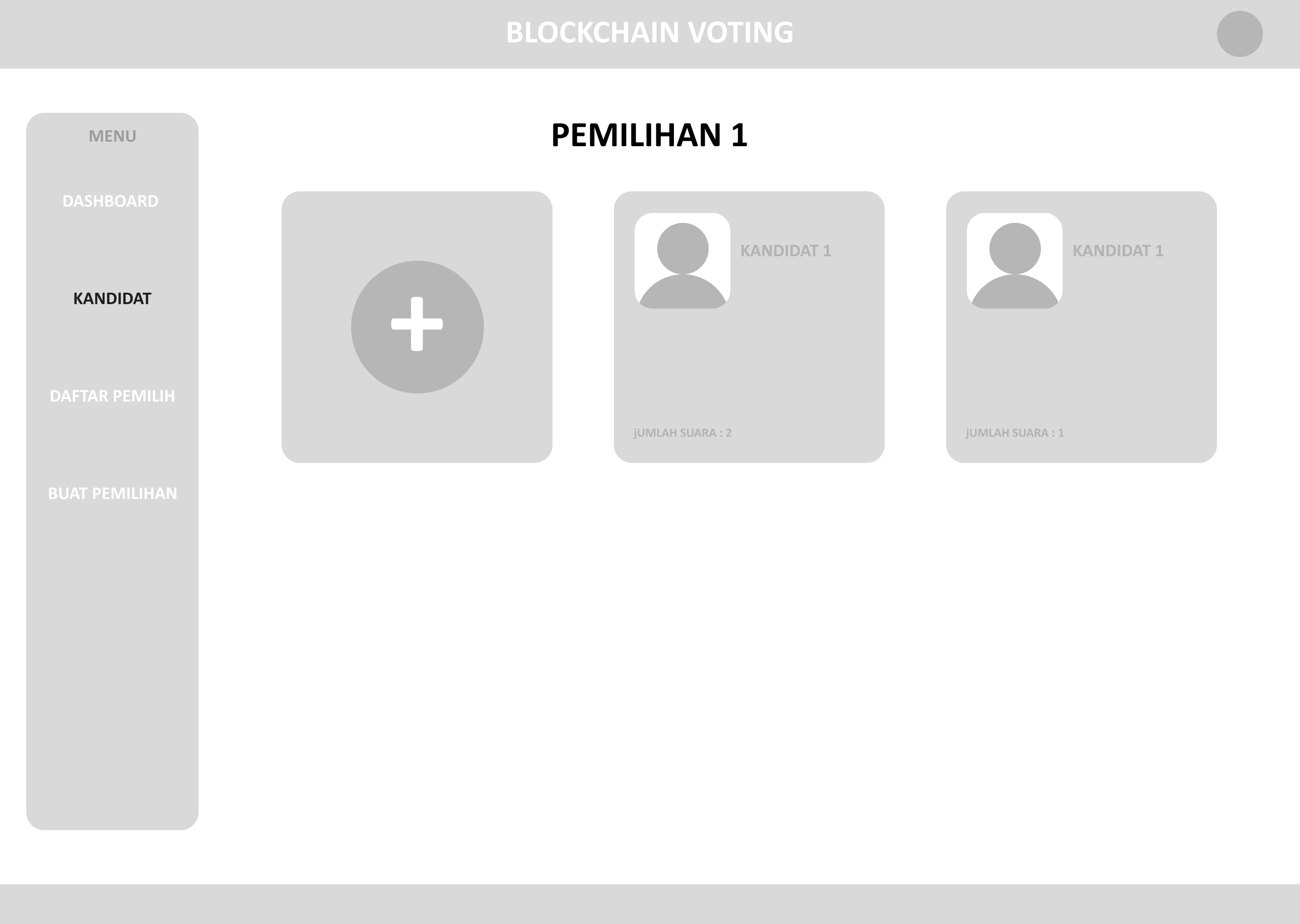
#### Gambar 4.9 Rancangan Antarmuka *Admin-Login*

1. Halaman Admin-Dashboard



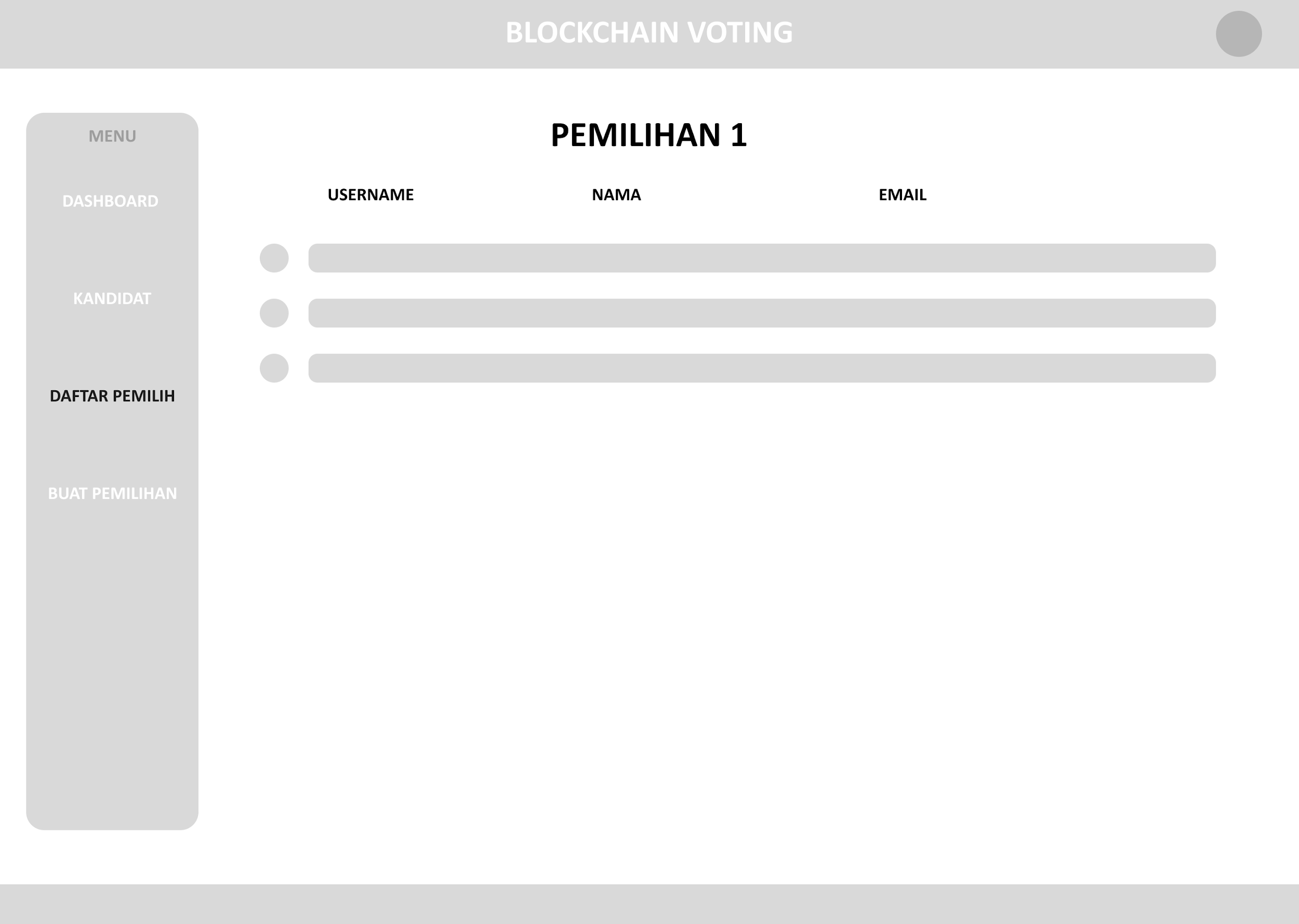
#### Gambar 4.10 Rancangan Antarmuka *Admin-Dashboard*

1. Halaman *Admin-Candidate*



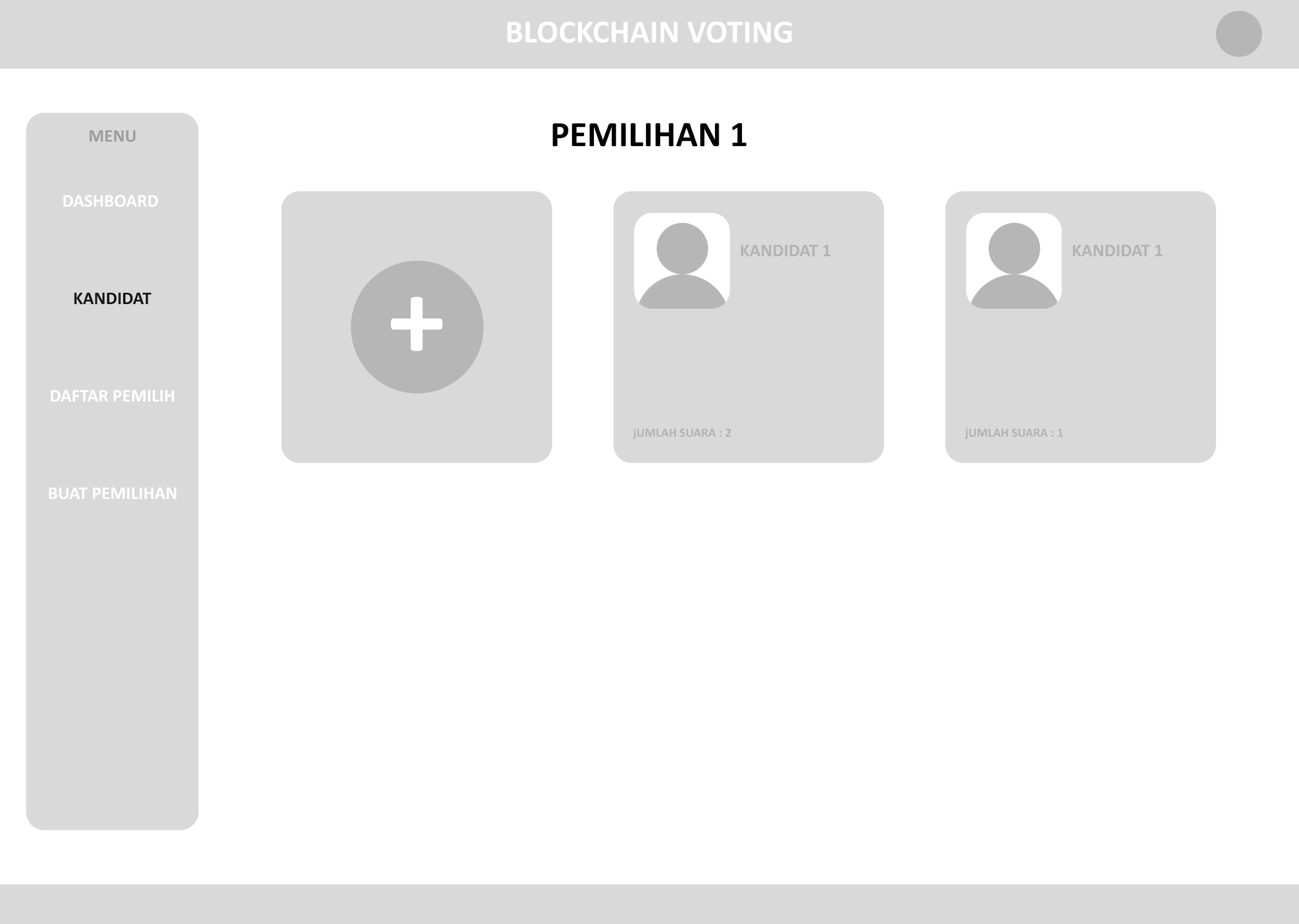
#### Gambar 4.11 Rancangan Antarmuka *Admin-Candidate*

1. Halaman *Admin-Usermng*

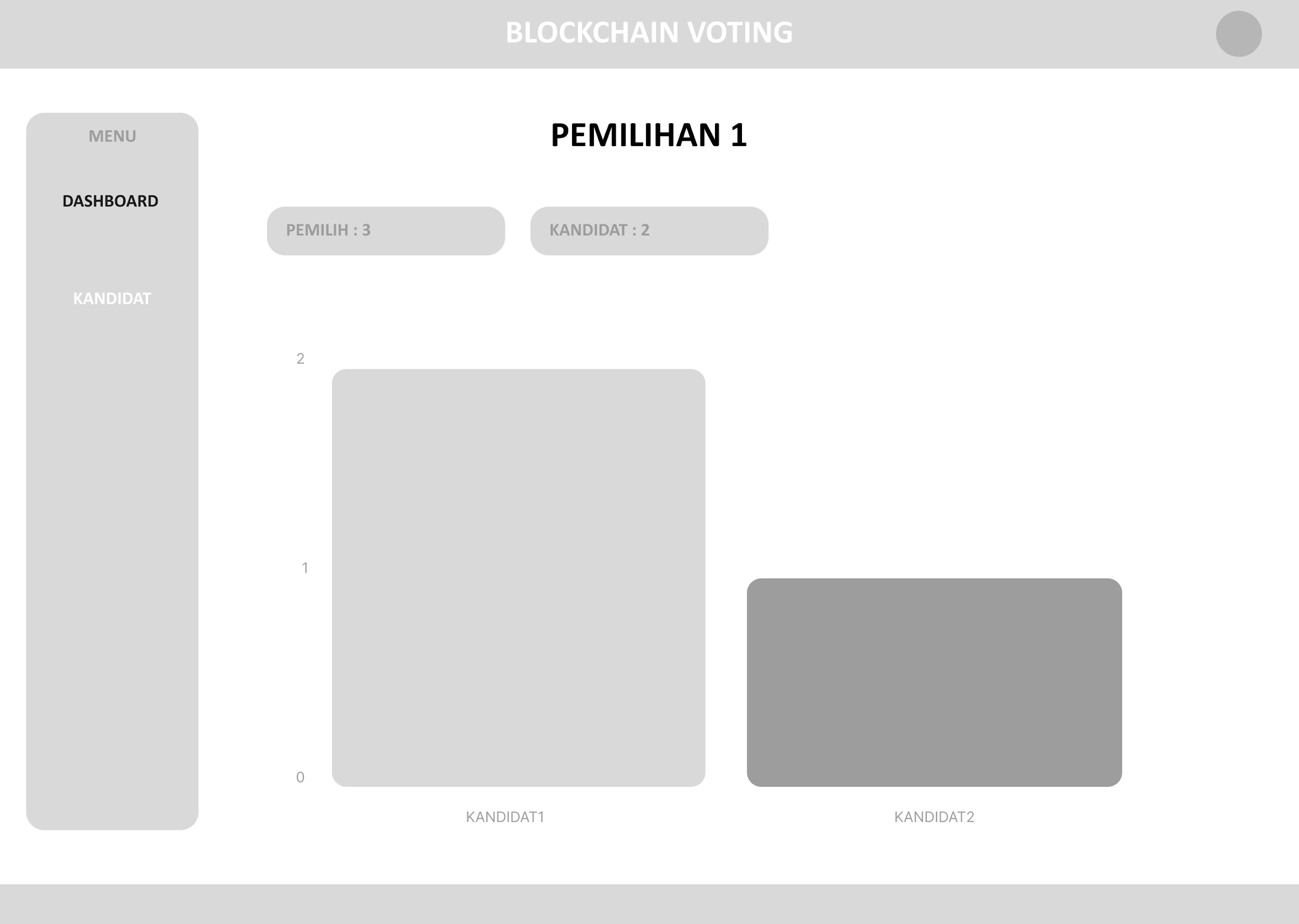


#### Gambar 4.12 Rancangan Antarmuka *Admin-Usermng*

1. Halaman *Admin-Election*

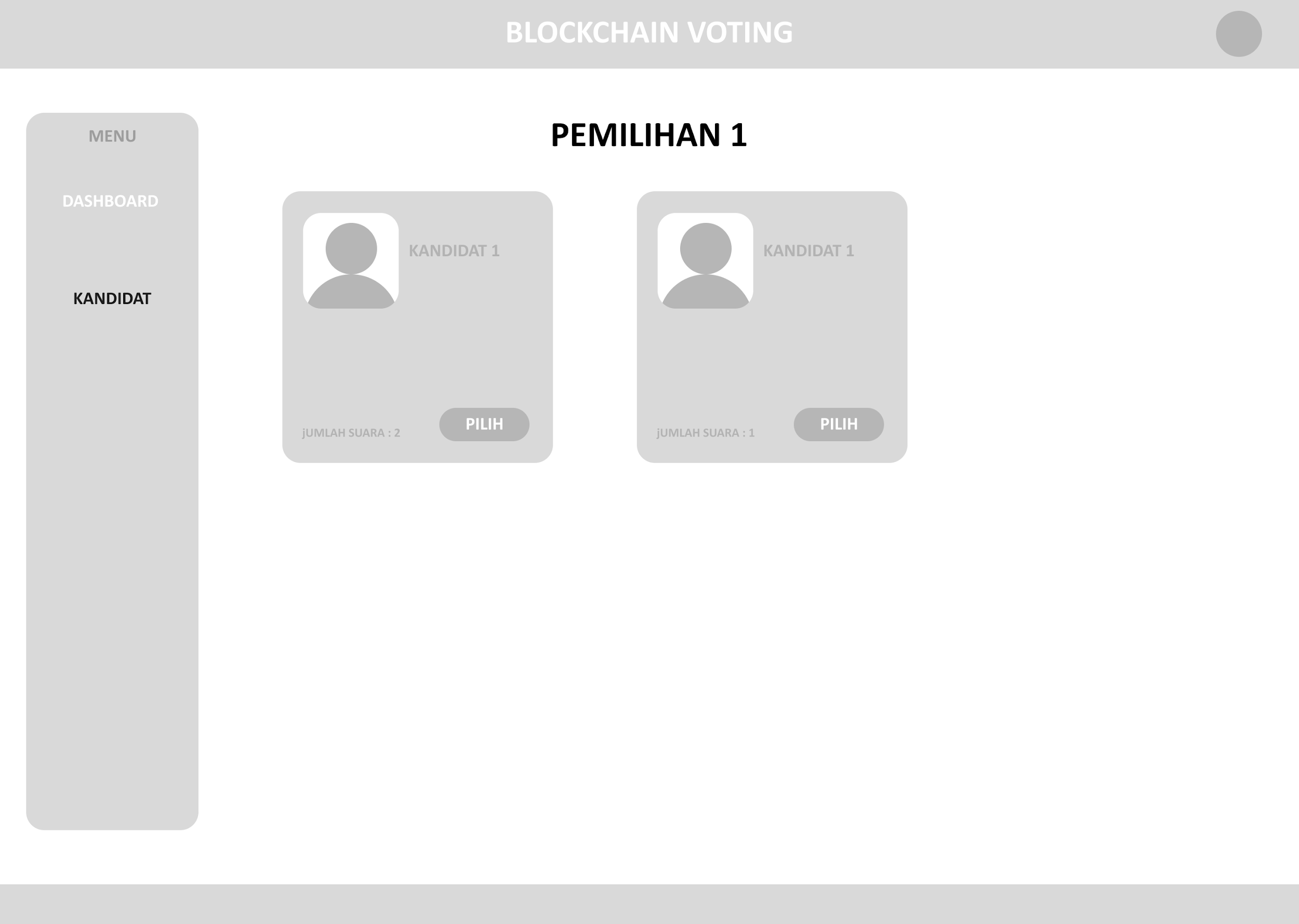


1. Halaman *User-Dashboard*



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

1. Halaman *User-Vote*



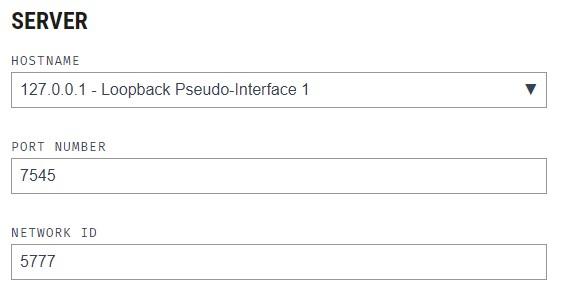
#### Gambar 4.14 Rancangan Antarmuka *User-Vote*

## 4.4. *Testing* (Pengujian)

### 4.4.1. Pengujian *Smart Contract*

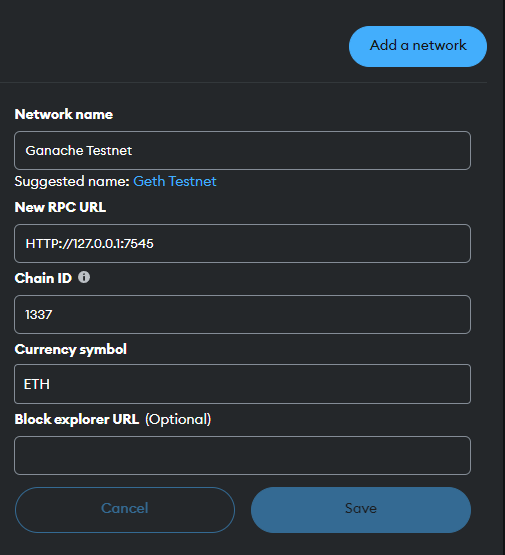
*Smart Contract Testing* bertujuan untuk memastikan bahwa kode *smart contract* yang ditulis dalam *Solidity* bekerja sesuai spesifikasi. Pengujian dilakukan dengan menggunakan *Node.js v.18.40, Truffle v5.11.5 , Ganache v2.7.1 , Solidity, dan web3.js.*

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 4.8:



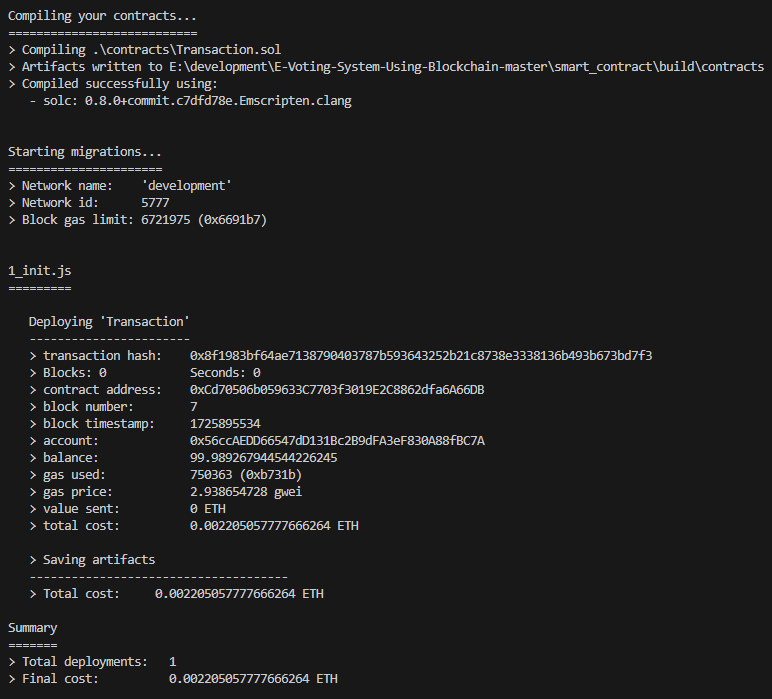
#### Gambar 4.15 *Ganache Server Detail*

Selanjutnya konfigurasi *metamask* untuk jaringan *local blockchain* seperti pada gambar 4.9:



#### Gambar 4.16 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 4.10:

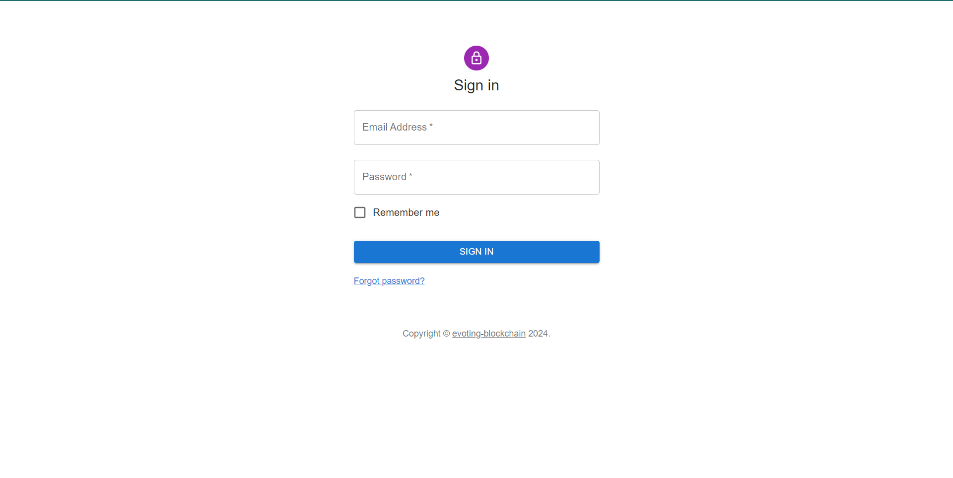


#### Gambar 4.17 *Truffle Migrate*

### 4.4.2. *Mock Election*

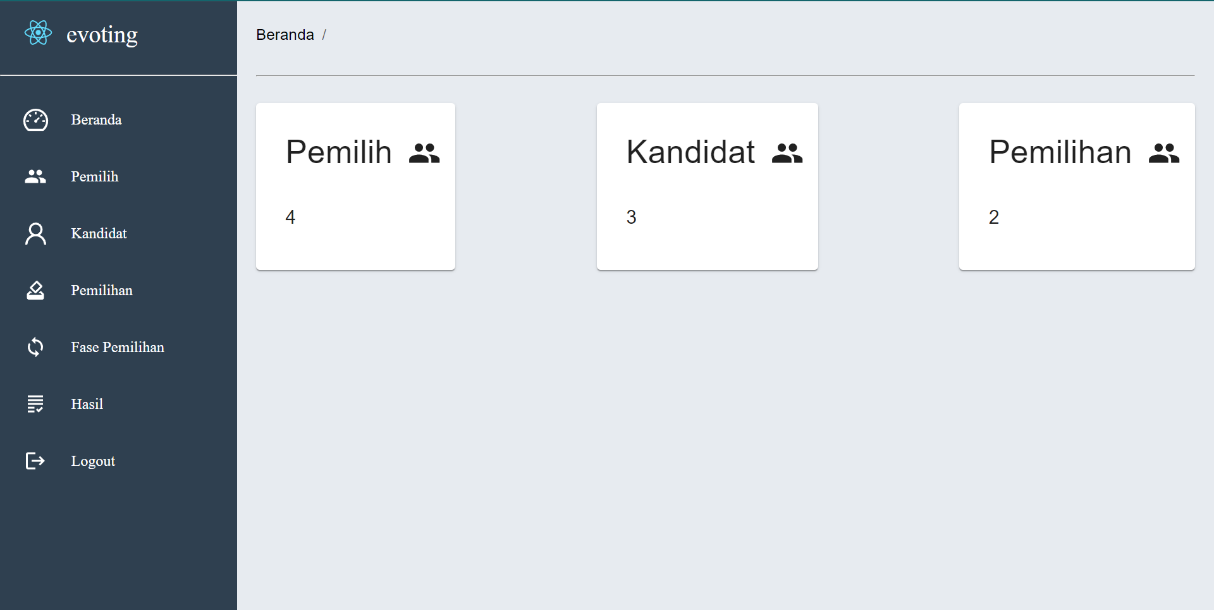
*Mock Election* adalah simulasi pemilihan elektronik yang dilakukan untuk menguji sistem dalam skenario nyata. Dalam uji coba ini, sejumlah pemilih melakukan pendaftaran, pemungutan suara, dan hasil pemungutan suara dihitung secara otomatis oleh sistem. Tujuannya adalah untuk menguji pengalaman pengguna,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

1. Halaman *login* untuk admin.



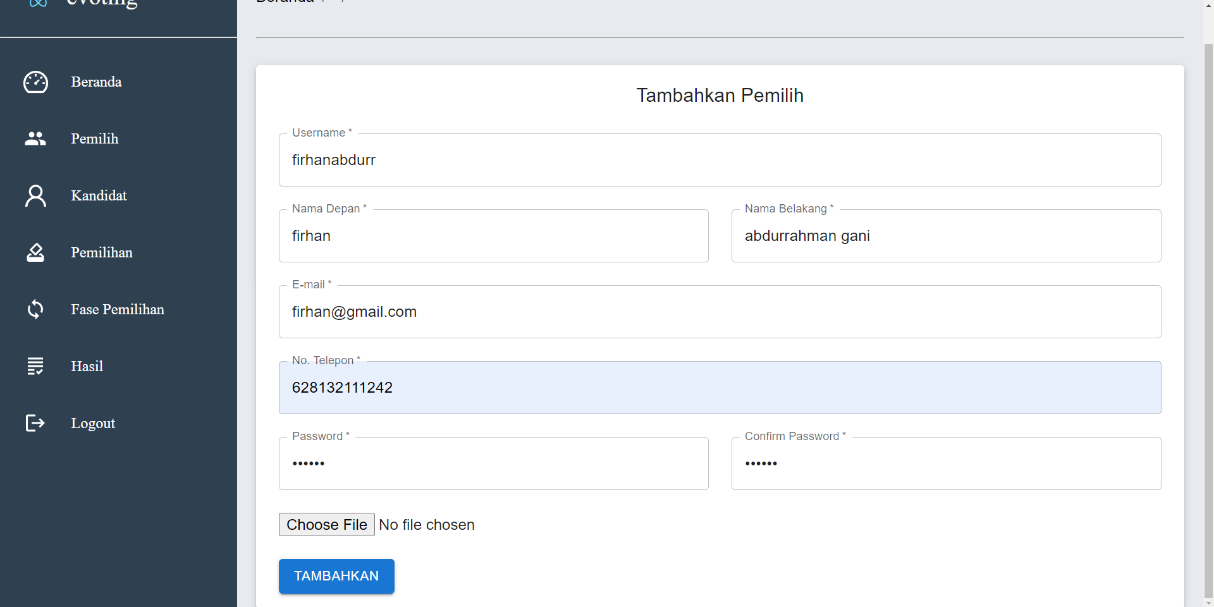
#### Gambar 4.18 *Admin-login*

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.



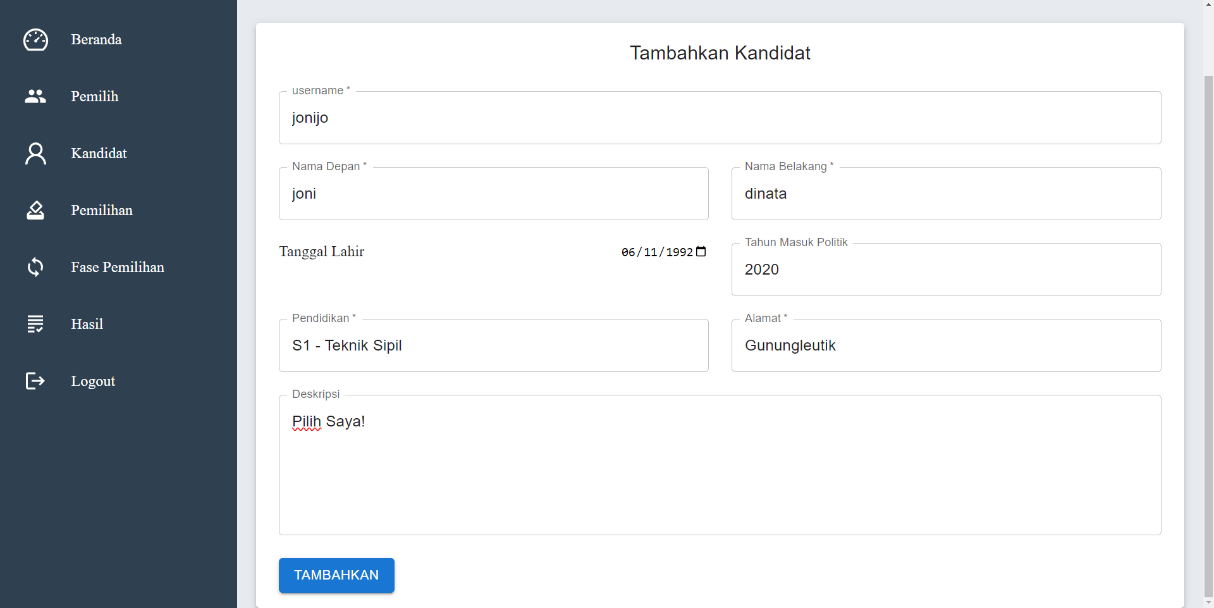
#### Gambar 4.19 *Admin-dahsboard*

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



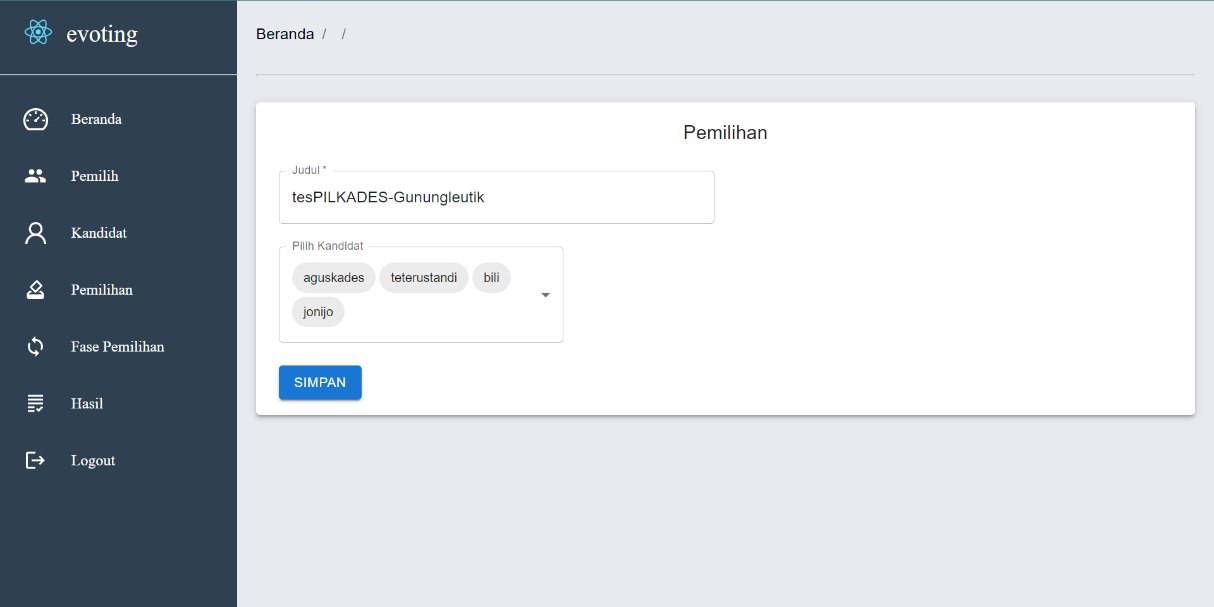
#### Gambar 4.20 *Add User*

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



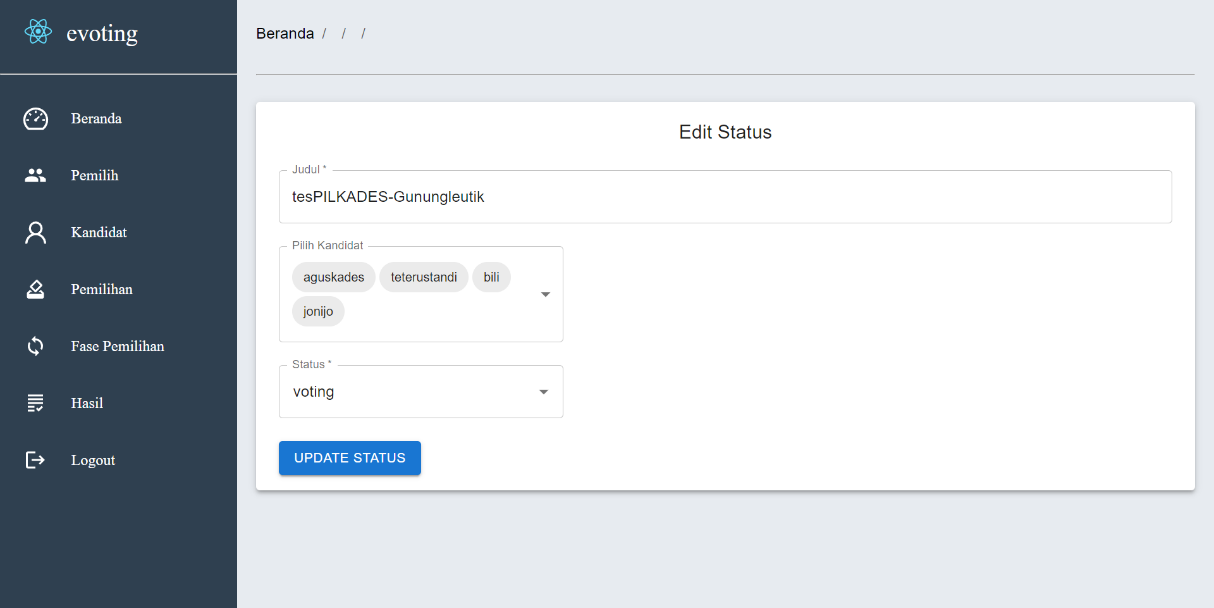
#### Gambar 4.21 *Add Candidate*

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



#### Gambar 4.22 *Add Election*

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



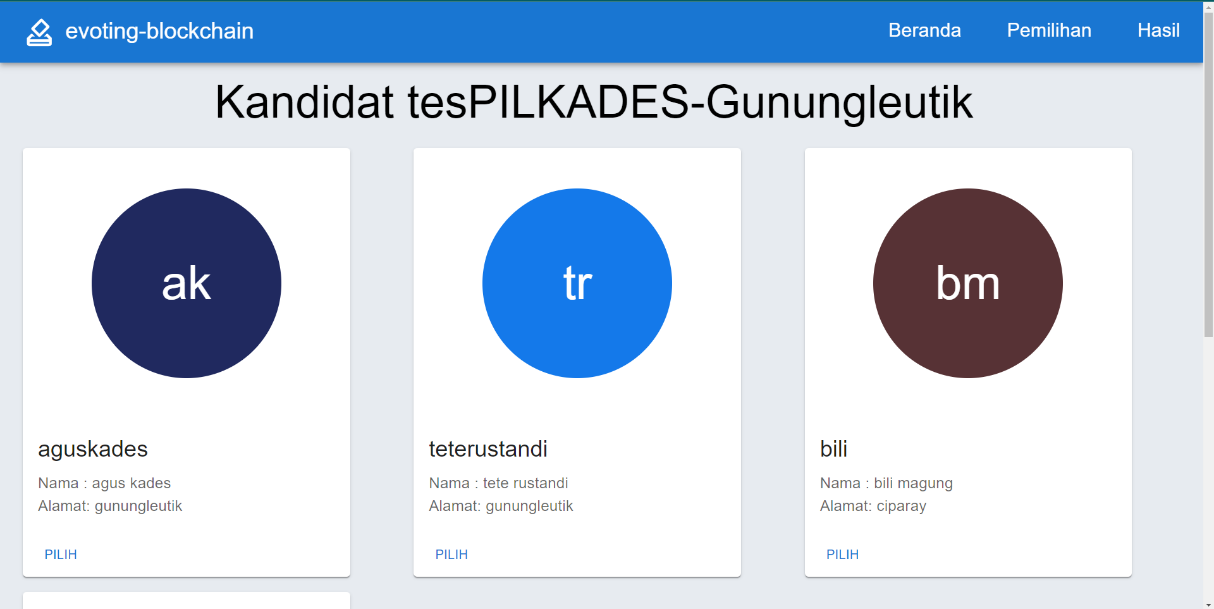
#### Gambar 4.23 *Add Phase*

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



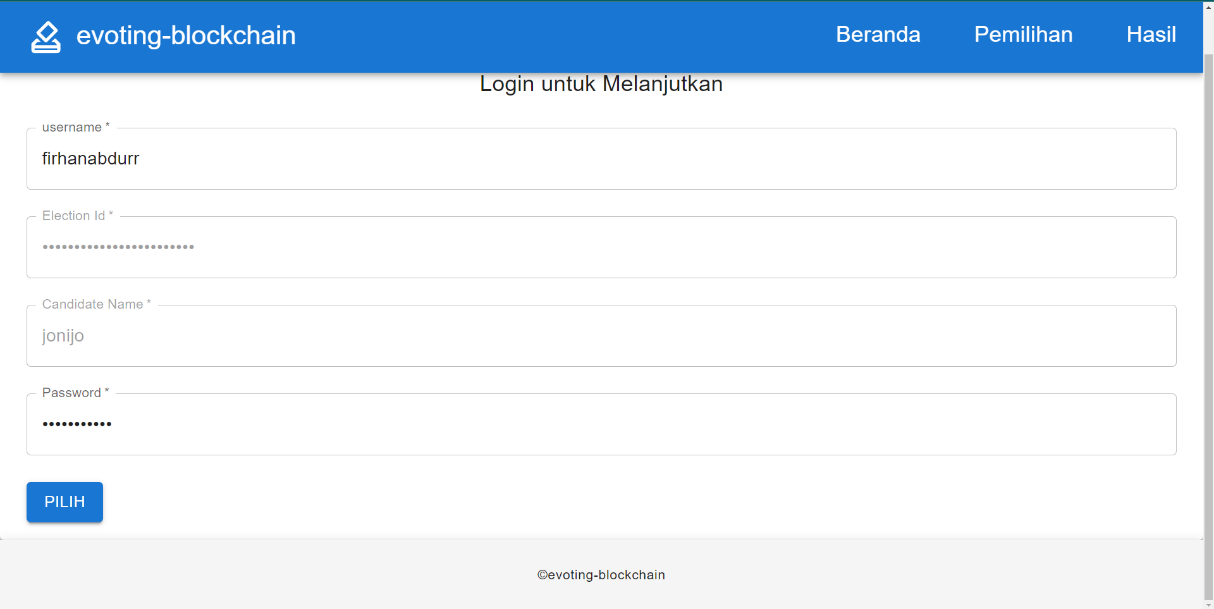
#### Gambar 4.24 *Election*

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



#### Gambar 4.25 *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.



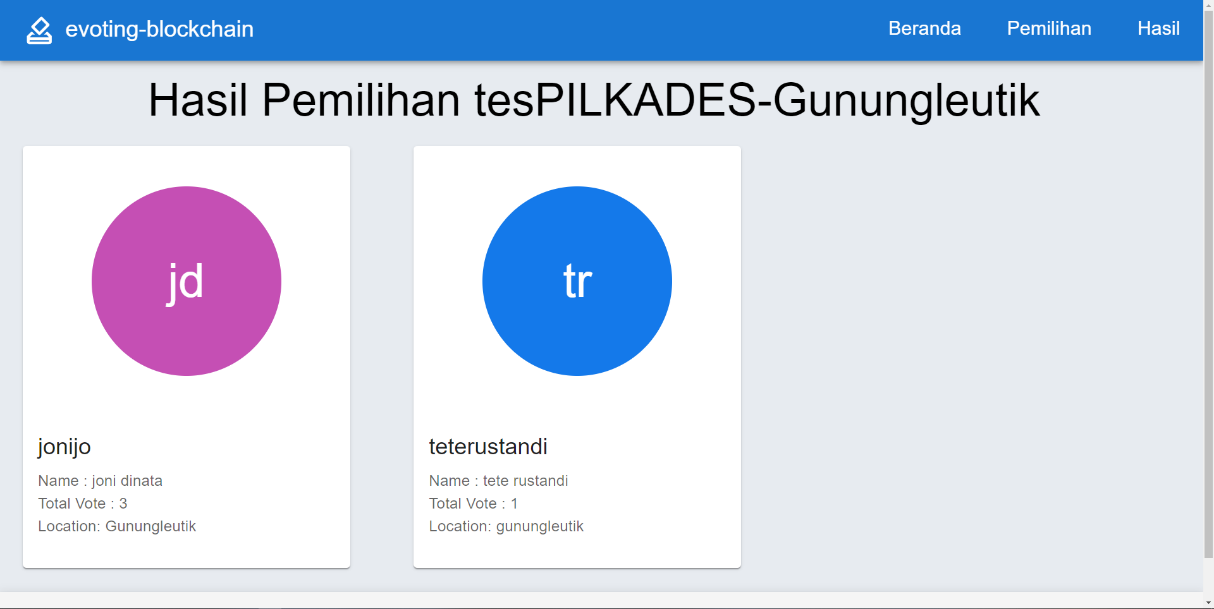
#### Gambar 4.26 *User Identification*

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



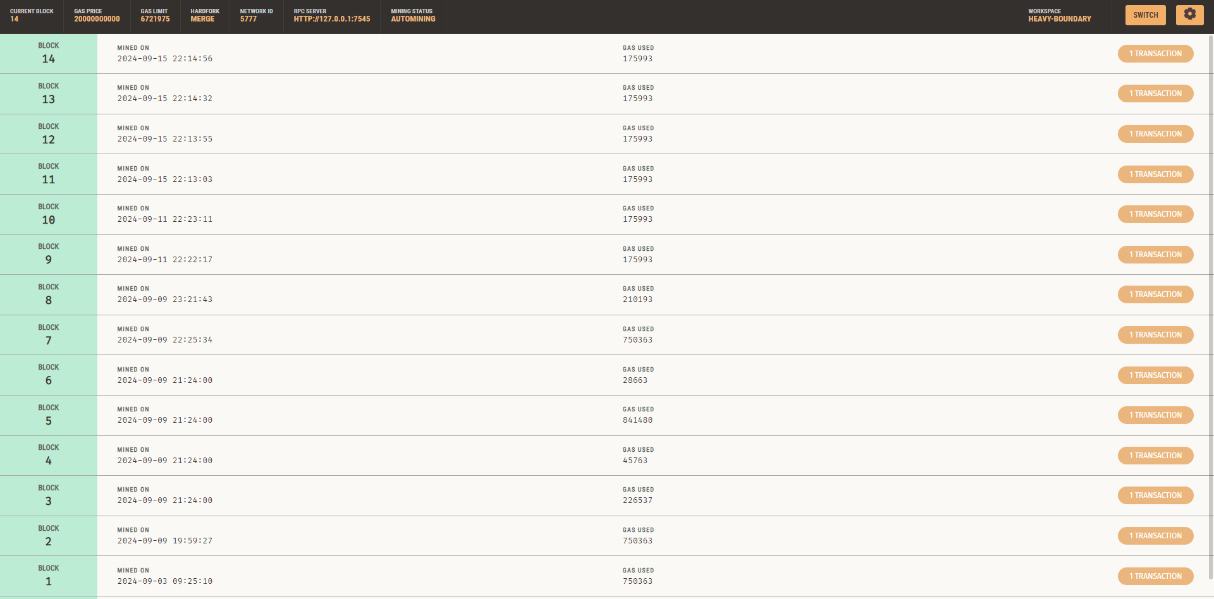
#### Gambar 4.27 *Metamask Confirmation*

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

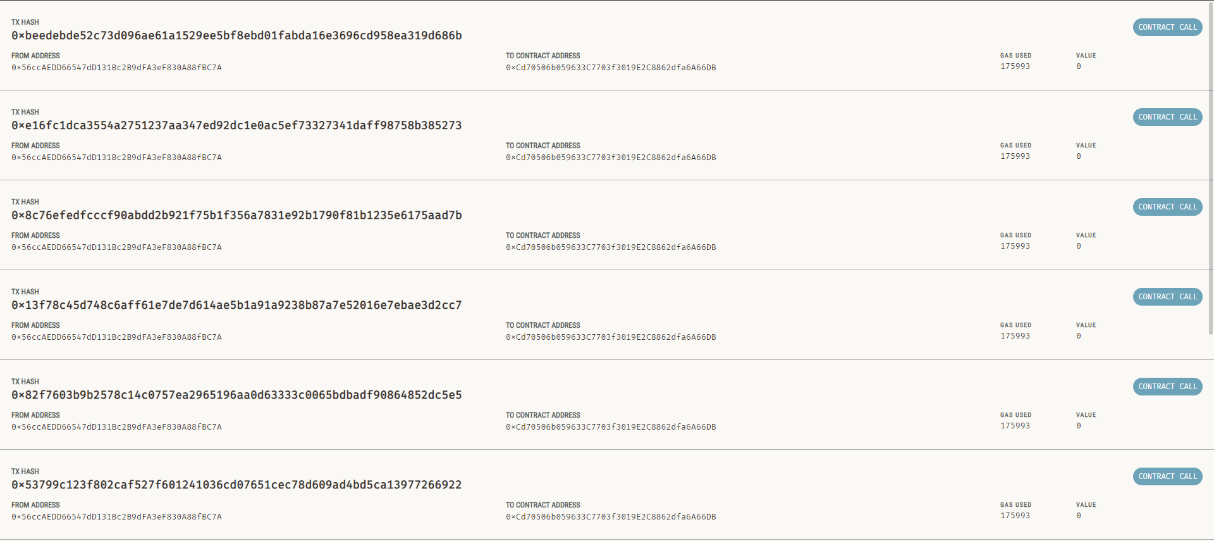


#### Gambar 4.28 *Result*

1. Untuk keperluan audit, setiap transaksi direkam ke dalam jaringan blockchain, gambar 4.29 dan 4.30 merupakan rincian transaksi yang sudah terenkripsi menggunakan kriptografi *blockchain*.



#### Gambar 4.29 *Mining Block*



#### Gambar 4.30 *Record Transaction*

# BAB V KESIMPULAN DAN SARAN

## 4.3. Kesimpulan

## 4.3. Saran

Lampiran 1. Lembar Bimbingan Skripsi

Lampiran 1. Surat Ijin Penelitian Lapangan

Lampiran 1. Daftar Pertanyaan Kuisioner

Lampiran 1. Hasil Pengolahan Data

Lampiran 1. *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 1. *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.

# DAFTAR PUSTAKA

[1] Hardjaloka, L., & Simarmata, V. M. (n.d.). *E-Voting: Kebutuhan vs. Kesiapan (Menyongsong) E-Demokrasi*.

[2] Falah, A. I., Rheza, K., & Adinegoro, R. (n.d.). PELUANG DAN TANTANGAN ADOPSI E-VOTING INDIA DALAM PEMILIHAN KEPALA DAERAH DI INDONESIA. In *Humaniora Dan Kebijakan Publik* (Vol. 5).

[3] *Wood, Gavin. ETHEREUM: A SECURE DECENTRALISED GENERALISED TRANSACTION LEDGER EIP-150 REVISION*. (n.d.).

[4] Wu, Kean & Liu, Manlu & Xu, Jennifer. (2019). How Will Blockchain Technology Impact Auditing and Accounting: Permissionless Vs. Permissioned Blockchain. Current Issues in Auditing. 13. 10.2308/ciia-52540.

[5] Setiawan Restu Aji, and Wahyuningdiah Trisari Harsanti Putri. “Implementasi Teknologi Blockchain Dalam Aplikasi E-Voting Berbasis Mobile.” *Digital Zone: Jurnal Teknologi Informasi Dan Komunikasi*, vol. 14, no. 2, 1 Dec. 2023,<https://doi.org/10.31849/digitalzone.v14i2.16682>.

[6] Hutahaean, J. (2015). Konsep Sistem Informasi. Deepublish. https://books.google.co.id/books?id=o8LjCAAAQBAJ

[7] Bhutta, Muhammad Nasir Mumtaz, et al. “A Survey on Blockchain Technology: Evolution, Architecture and Security.” *IEEE Access*, vol. 9, no. 2169-3536, 2021, pp. 1–1,<https://doi.org/10.1109/access.2021.3072849>.

[8] Zheng, Zibin, et al. “An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends.” *2017 IEEE International Congress on Big Data (BigData Congress)*, June 2017, https://doi.org/10.1109/bigdatacongress.2017.85. Accessed 29 Mar. 2024.

[9] H Purwantoro and M Y Wibowo, Teknologi Blockchain dan Cara Kerjanya. APTIKOM, 2020.

[10] M.Ilyas Gultom, M., & Saripurna, D. (2019). Perancangan Sistem Keamanan Aplikasi Evoting Untuk Pemilihan Ketua Badan Eksekutif Mahasiswa Fakultas Teknik UISU Dengan Menggunakan Algoritma MD5. Algoritma : Jurnal Ilmu Komputer Dan Informatika, 3(2), 70.<https://doi.org/10.30829/algoritma.v3i2.6438>

[11] Ben Ayed, A. (2017). A Conceptual Secure Blockchain Based Electronic voting System. International Journal of Network Security & Its Applications, 9(3), 01–09. https://doi.org/10.5121/ijnsa.2017.9301

[12] Andreas M Antonopoulos and Dr Gavin Wood, Mastering Ethereum: Building Smart Contracts and DApps 1st Edition, 1st ed. O’Reilly Media, 2018.

[13] Fadli, Fajri, et al. *Electronic Voting Using Decentralized System Based on Ethereum Blockchain*. 1 Mar. 2020,<https://doi.org/10.32409/jikstik.19.1.2718>.

[14] Curran, Kevin. “E-Voting on the Blockchain.” *The Journal of the British Blockchain Association*, vol. 1, no. 2, 4 Dec. 2018, pp. 1–6,<https://doi.org/10.31585/jbba-1-2-(3)2018>.

[15] Adekeye, Taiwo. (2023). Implementing a secure electronic voting system using Blockchain smart contract. 10.13140/RG.2.2.27008.10242.

[16] Hjálmarsson, Friðrik, and Gunnlaugur Hreiðarsson. Blockchain-Based E-Voting System.

[17] Jafar, Uzma, et al. “Blockchain for Electronic Voting System—Review and Open Research Challenges.” Sensors, vol. 21, no. 17, 31 Aug. 2021, p. 5874, https://doi.org/10.3390/s21175874.

[18] Kshetri, Nir, and Jeffrey Voas. “Blockchain-Enabled E-Voting.” IEEE Software, vol. 35, no. 4, July 2018, pp. 95–99, https://doi.org/10.1109/ms.2018.2801546.

[19] Mizzi, Jerome, and Frankie Inguanez. “Blockchain Based E-Voting System.” Journal of E-Technology, vol. 9, no. 2, 1 May 2018, p. 44, https://doi.org/10.6025/jet/2018/9/2/44-48.

[20] Shahzad, Basit, and Jon Crowcroft. “Trustworthy Electronic Voting Using Adjusted Blockchain Technology.” IEEE Access, vol. 7, 2019, pp. 24477–24488, https://doi.org/10.1109/access.2019.2895670.

[21] ---. “Blockchain Challenges and Opportunities: A Survey.” International Journal of Web and Grid Services, vol. 14, no. 4, 2018, p. 352, https://doi.org/10.1504/ijwgs.2018.095647.

[22] “Use of E-Voting around the World | International IDEA.” Www.idea.int, www.idea.int/news-media/media/use-e-voting-around-world.