

Electronic Voting (e-voting) sebagai Aplikasi Terdesentralisasi pada Vexanium *Blockchain*

Suryo Bramasto¹, Sandriana Febia Savitri², Endang Ratnawati D³

^{1,2,3}Department of Informatics, Institut Teknologi Indonesia, Indonesia
suryo.bramasto@iti.ac.id¹

Article Info

Article history:

Received Aug 24, 2023

Revised Mar 27, 2024

Accepted May 6, 2024

Keywords:

blockchain
decentralized application
electronic voting
smart contract
Vexanium

ABSTRACT

Decentralized applications, or DApps, are software programs that run on a blockchain or peer-to-peer (P2P) network of computers instead of on a single computer. Thus, DApps are outside of the purview and control of a single authority. In this research, a DApps for electronic voting (e-voting) that runs on a Vexanium blockchain was built. In addition to the Vexanium platform and toolchain, PHP 8.2, Apache web server, and PostgreSQL were used to build e-voting DApps in this research. Common public blockchain application development steps are implemented for building applications on the Vexanium Blockchain. For the next step in development, a Ricardian contract will be implemented. Security at the blockchain level should also be implemented, such as building a re-entrancy attack mechanism, improving the source of randomness for the nonce, and ad safeguards against frontrunning.

Copyright © 2024 Universitas Indraprasta PGRI.
All rights reserved.

Corresponding Author:

Suryo Bramasto,
Department of Informatics,
Institut Teknologi Indonesia,
Jl. Raya Puspipetek Serpong, Kota Tangerang Selatan, Banten.
Email: suryo.bramasto@iti.ac.id

1. PENDAHULUAN

Blockchain adalah buku besar digital terdesentralisasi dan terdistribusi yang mencatat transaksi di seluruh jaringan komputer. *Blockchain* menggunakan kriptografi untuk mengamankan dan memverifikasi transaksi serta untuk mengontrol pembuatan unit baru dari mata uang kripto tertentu. Setiap *block* dalam rantai berisi sejumlah transaksi, dan setiap kali transaksi baru ditambahkan ke jaringan, catatan transaksi tersebut ditambahkan berikut tanda tangan digital terenkripsi dan *timestamp* tervalidasi setiap peserta ke buku besar setiap peserta pada rantai. Dengan demikian tercipta catatan yang tidak dapat diubah dari semua transaksi di *blockchain*, sehingga menyulitkan satu pengguna untuk memanipulasi atau mengubah catatan. Desentralisasi dan kekekalan inilah yang membuat *blockchain* sangat cocok untuk berbagai aplikasi, seperti *cryptocurrency*, *smart contracts*, dan manajemen rantai pasokan. Desentralisasi dan kekekalan merupakan aspek-aspek yang dapat diimplementasikan dalam keamanan informasi, selain ketidakjelasan. Dengan demikian *blockchain* juga dapat diimplementasikan dalam ekosistem yang sangat bergantung kepada prinsip-prinsip keamanan informasi seperti voting elektronik (*e-voting*).

E-voting mengacu pada penggunaan sistem elektronik untuk memberikan dan menghitung suara dalam pemilihan. Sistem ini dapat mencakup pemungutan suara melalui internet, pemungutan suara melalui telepon, dan mesin pemungutan suara. *E-voting* dimaksudkan agar proses pemungutan suara lebih efisien dan mudah diakses, serta mengurangi potensi kesalahan dan kecurangan. Namun, ada juga kekhawatiran tentang keamanan dan keakuratan sistem *e-voting*, serta potensi pemaksaan dan manipulasi pemilih. Oleh karena itu, sistem *e-voting* harus dirancang, diuji, dan diaudit dengan benar untuk memastikan integritas proses pemungutan suara [2].

Penelitian serupa yang pernah dilakukan yakni membangun aplikasi terdesentralisasi guna *cryptocurrency* (Desentralized Finance/DeFi), pada Ethereum *blockchain* [1], sedangkan penelitian pada

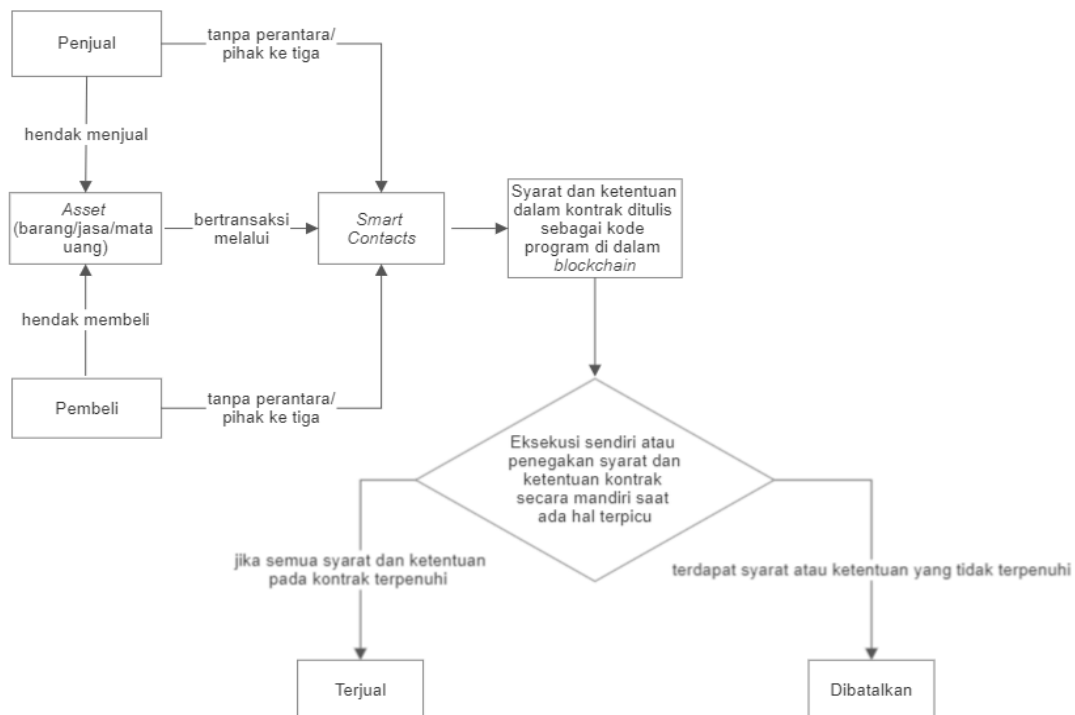
artikel ini mencoba membangun aplikasi terdesentralisasi yakni sistem *e-voting* diatas jaringan *blockchain* Vexanium, dengan tujuan untuk menyampaikan bagaimana membangun sekaligus *deployment* untuk aplikasi terdesentralisasi pada jaringan *blockchain* Vexanium, berikut bagaimana memenuhi kebutuhan-kebutuhan pendukungnya seperti pemenuhan aspek keamanan dengan penerapan enkripsi. Vexanium merupakan protokol publik *blockchain* dan platform kontrak pintar (*smart contract*) yang memungkinkan pengembang guna membuat proyek-proyek berbasis *blockchain*. Vexanium telah mendukung DApps (*Decentralized Applications*), DeFi (*Decentralized Finance*), serta berbagai kasus nyata penerapan *blockchain* untuk *startups*, korporasi, dan bisnis. *Blockchain* Vexanium menerapkan *smart contract* yang merupakan kontrak otomatis dalam bentuk barisan kode tanpa memerlukan pihak ke-tiga. *Smart contract* akan tersimpan di setiap *node* yang ada pada seluruh jaringan *blockchain* [3].

1.1 Smart Contracts dan penerapannya pada aplikasi terdesentralisasi (e-voting)

Smart contract merupakan kontrak transaksional tanpa perantara, layanan, maupun dokumen tambahan, menggunakan alat tukar *cryptocurrency*. *Smart contract* dan *cryptocurrency* memungkinkan aturan untuk didefinisikan dan ditegakkan *by system*. Pada penerapan *smart contract* pada *cryptocurrency*, aset atau *currency* atau mata uang diubah menjadi program yang menjalankan dan mengevaluasi kode berdasarkan tugas-tugas yang menentukan apakah aset atau *currency* berpindah ke pihak lain atau kembali ke pemiliknya semula [4]. Ketentuan-ketentuan pada *smart contract* yakni:

- Kontrak dibuat antara dua pihak yang senantiasa anonim.
- Kontrak disimpan pada *public ledger*.
- Terdapat hal-hal pemicu yang ditentukan misalnya tenggat waktu.
- Smart contract* mengeksekusi dirinya sendiri sesuai dengan kode program tertulis.
- Regulator dan pengguna dapat melakukan analisis terhadap semua aktivitas.
- Mampu memprediksi tren dan ketidakpastian pasar.

Ilustrasi penerapan transaksi menggunakan *blockchain* dan *smart contract* ditunjukkan pada Gambar 1.



Gambar 1. Ilustrasi transaksi menggunakan *smart contract* dan *blockchain* [5]

Sistem *smart contracts* bekerja pada klausa *if-then*, dimana tidak ada yang dapat meretas sistem tersebut tanpa sepengetahuan setiap pihak yang terlibat di dalam *blockchain* [6]. Jika terdapat seseorang yang mencoba melakukan akses terhadap kode program di dalam *blockchain* maka semua yang terikat *smart contract* di dalam *blockchain* akan segera memperoleh peringatan. Teknologi transaksi berbasis *blockchain* dan *smart contract* ini dapat digunakan untuk penegakan kredit, layanan finansial, premi asuransi, hukum properti, dan sebagainya [7].

1.2 Delegated Proof of Stake (DPoS) [8]

Inti dari Vexanium adalah DPoS, dimana merupakan algoritma konsensus yang dikembangkan guna mengamankan *blockchain* dengan memastikan representasi transaksi di dalamnya. DPoS didesain sebagai implementasi demokrasi berbasis teknologi, menggunakan proses voting dan pemilihan guna melindungi *blockchain* dari sentralisasi dan penggunaan jahat. DPoS diciptakan sebagai alternatif terhadap konsensus *blockchain Proof-of-Work* yang tidak efisien dalam konsumsi energi dan konsensus *blockchain Proof-of-Stake* yang kurang aman. DPoS juga lebih *scalable* dibandingkan kedua konsensus tersebut. Dengan DPoS, semua *block* tervalidasi guna mengurangi penggunaan sumber daya sekaligus meningkatkan kecepatan transaksi pada setiap *block* yang dihasilkan. Piranti lunak Vexanium memungkinkan *block* dihasilkan tepat per setengah detik dan tepat satu produser terotorisasi untuk menghasilkan *block* pada waktu yang ditentukan. Jika *block* tidak dihasilkan pada waktu yang ditentukan/terjadwal, maka *block* untuk slot waktu tersebut dilewati. Jika terdapat satu atau lebih *block* dilewati akan terdapat celah pada *blockchain* sebesar paling tidak setengah detik. *Block* dihasilkan dalam 126 ronde, yakni 6 *block* dikalikan 21 produser. Di awal setiap ronde, 21 produser *block* terpilih secara unik dengan mekanisme voting sesuai keinginan pemegang *token*. Produser-produser yang terpilih terjadwal dalam urutan yang disetujui oleh 15 atau lebih produser. Jika sebuah produser melewatkan sebuah *block* dan tidak menghasilkan *block* dalam 24 jam terakhir, maka produser tersebut akan dihapus hingga produser tersebut mengirimkan notifikasi ke *blockchain* jika bermaksud melakukan produksi *block* kembali. Hal ini memastikan kehandalan operasi jaringan *blockchain*.

Pada kondisi normal, tidak akan pernah terjadi *forks* pada sebuah *blockchain* DPoS karena pada *blockchain* DPoS produser-produser *block* saling bekerja sama dalam menghasilkan *block*, bukan saling berkompetisi seperti halnya pada *blockchain* lain. Namun jika terjadi *fork*, maka konsensus secara otomatis akan beralih ke rantai terpanjang. Dengan metode ini maka laju *block-block* mana yang ditambahkan ke sebuah *blockchain fork* terkait langsung dengan persentase produser-produser *block* yang berbagi konsensus yang sama. Atau dengan kata lain *blockchain fork* dengan jumlah produser yang lebih banyak akan membuat panjang *blockchain* tumbuh lebih cepat dengan persentase *block* meleset lebih rendah. DPoS Vexanium juga menerapkan ketentuan dimana tidak ada produser yang dapat menghasilkan *block* pada dua *fork* secara bersamaan. Jika ada *block* yang melakukan hal ini maka akan dikeluarkan dari *blockchain*. Bukti kriptografis diterapkan sebagai proteksi ganda guna penghapusan penyalahgunaan-penyalahgunaan secara otomatis. DPoS Vexanium menerapkan *Byzantium Fault Tolerance* sehingga memungkinkan semua produser menandatangani semua *block*, selama tidak ada produser yang menandatangani dua atau lebih *block* dengan *timestamp* yang sama atau pada level yang sama. Jika ada *block* yang melakukan pelanggaran yakni menandatangani dua *block* atau lebih pada *timestamp* atau level yang sama, maka setiap *block* sebagai *byzantine producer* akan menghasilkan bukti kriptografis sebagai bukti pelanggarannya. Setelah 15 produser menandatangani suatu *block*, maka *block* tersebut dinyatakan *irreversible*. *Byzantium Fault Tolerance* memungkinkan konsensus *irreversible* tercapai dalam 1 detik.

1.3 Vexwallet

Vexwallet merupakan *wallet* terdesentralisir yang berjalan pada Vexanium *blockchain*, dimana setiap asset dan akun pengguna menjadi tanggung jawab masing-masing pengguna tersebut, serta hanya pengguna yang merupakan pemilik akun dan asset yang memiliki akses terhadap akun dan akses. Vexwallet memiliki versi MS Windows, MAC, IOS, dan Android. Fitur *basic* dari Vexwallet antara lain menyimpan Vex; transfer Vex; serta jual beli RAM, CPU, dan NET guna transaksi di Vexanium *blockchain*. Vexwallet memungkinkan penggunaannya untuk mengelola Vex dan menggunakan semua Vexanium *decentralized applications* (Dapp) dalam satu platform. *Block Producers* merupakan entitas-entitas terdesentralisasi yang biasanya terdiri atas grup, perusahaan, atau organisasi yang dipilih melalui *e-voting blockchain system* di dalam DPoS Consensus yang mengatur Vexanium *blockchain*. Tugas *Block Producers* adalah menghasilkan *block-block* baru di dalam protocol Vexanium *blockchain* sekaligus melakukan verifikasi terhadap transaksi. DPoS merupakan suatu mekanisme konsensus yang mengatur Vexanium *blockchain* dimana hanya pengurus organisasi (delegasi) yang dipilih melalui *e-voting* dapat memverifikasi transaksi. *Block producers* memiliki fungsi dasar serupa dengan *miner* pada Bitcoin *blockchain* dengan tambahan beberapa fungsi. Mekanisme konsensus DPoS yang dijalankan oleh *block producers* sebagai representatif dari Vexanium yang dipilih oleh semua pengguna dalam jaringan melalui *voting*. *Block producers* memiliki tanggung jawab untuk membuat dan memvalidasi *blocks* dalam jaringan. *Block producers* memperoleh imbalan terhadap tanggung jawab dan kerjanya.

1.4 VEX token model dan penggunaan sumber daya

Vexanium *blockchain* menerapkan model *blockchain as a service* dimana secara model bisnis, bisnislah yang membiayai keseluruhan operasi. Dengan demikian aplikasi-aplikasi terdesentralisasi tidak memaksa pelanggannya untuk membayar *blockchain* secara langsung terkait penggunaan *blockchain*. Terdapat pembatasan penggunaan sumber daya (*storage*, CPU, dan RAM) secara *default* terkait penggunaan Vexanium

blockchains, serta terdapat sistem bawaan guna mencegah penyalahgunaan. Sedangkan untuk konsumsi *bandwidth* dan komputasi dikelompokkan menjadi penggunaan instan dan penggunaan jangka panjang. Setiap Vexanium *blockchain* menerapkan *log* aksi yang disimpan dan dapat diunduh oleh semua *nodes*. Dengan demikian dimungkinkan untuk melakukan rekonstruksi *state* dari semua aplikasi.

1.5 Blockchain block

Block merupakan struktur data di dalam basis data *blockchain*, dimana transaksi data di dalam *blockchain* direkam secara permanen [9]. Sebuah *block* merekam semua data transaksi-transaksi terbaru yang belum divalidasi jaringan *blockchain*. Setelah data divalidasi maka *block* akan ditutup dan dibuat *block* baru guna input dan validasi transaksi baru. Penyimpanan *records* di dalam *block* bersifat permanen dan tidak dapat atau dihapus. Ciri utama dari *blockchain block* adalah sebagai berikut:

- a. *Block* merupakan sebuah tempat di dalam *blockchain* dimana informasi tersimpan dan terenkripsi.
- b. *Block* teridentifikasi dengan deretan angka panjang yang mencakup informasi transaksi terenkripsi dari *block* sebelumnya sekaligus informasi transaksi baru.
- c. *Block* berikut keseluruhan informasi di dalamnya harus terverifikasi oleh jaringan *blockchain* sebelum *block* baru dapat diciptakan.

Sebuah *block* menyimpan berbagai informasi, dimana pada umumnya sebuah *block* terdiri atas beberapa elemen sebagai berikut:

- a. *Magic number*, yakni sebuah angka dengan nilai spesifik yang mengidentifikasi bahwa *block* tersebut merupakan bagian dari suatu jaringan *blockchain*.
- b. *Block size*, menentukan batasan ukuran *block* guna memastikan ukuran spesifik dari informasi yang dapat disimpan dalam *block*.
- c. *Block header*, berisi informasi tentang *block*.
- d. *Transaction counter*, merupakan angka yang merepresentasikan jumlah transaksi yang tersimpan di dalam *block*.
- e. *Transactions*, merupakan daftar transaksi di dalam *block*.

Elemen *transactions* dari *block* merupakan elemen terbesar, dimana mengandung informasi paling banyak.

Elemen *block header* dari *block* memiliki sub-elemen sebagai berikut:

- a. *Version*, versi dari apapun jaringan *blockchain* yang digunakan.
- b. *Previous block hash*, berisi *hash* dari *block's header* sebelumnya.
- c. *Hash Merkle root*, *hash* dari transaksi pada pohon *merkle* dari *block* saat ini.
- d. *Time*, dimana merupakan *timestamp* dari *block deployment* dalam *blockchain*.
- e. *Bits*, yang menunjukkan tingkat kesulitan dari target *hash* dimana menyatakan tingkat kesulitan penyelesaian *nonce*.
- f. *Nonce*, angka 32-bit terenkripsi yang harus diselesaikan setiap anggota dari *blockchain* guna verifikasi dan penutupan *block*.

2. METODE

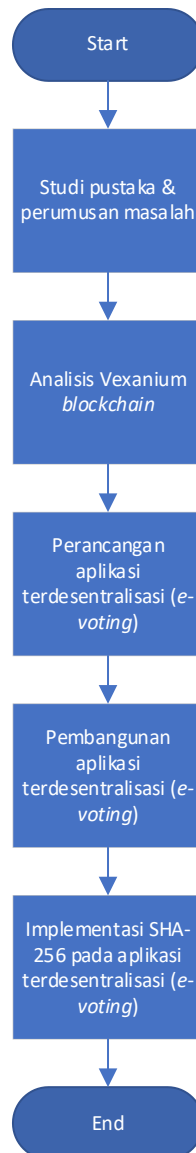
Metode penelitian yang diterapkan ditunjukkan dalam *flowchart* pada gambar 2.

2.1 Studi pustaka dan perumusan masalah

Dalam penelitian ini dilakukan studi pustaka dari jurnal ilmiah, buku, dan internet terkait *blockchain* dan penerapannya. Kemudian dilakukan perumusan masalah tentang bagaimana *blockchain* dapat diterapkan dalam suatu studi kasus selain *cryptocurrency* dengan tujuan menelaah manfaat dari *blockchain* selain yang terkait dengan *cryptocurrency*.

2.2 Analisis Vexanium blockchain [10]

Pemenuhan kebutuhan terkait penerapan Vexanium *blockchain* dilakukan dengan bahasa pemrograman C++ sebagai lingkungan pengembangan, kemudian menerapkan konstruksi kanonikal C++ *code construct* dalam pengembangan *blockchain* dan membangun *smart contract*. Dipersiapkan juga infrastruktur lingkungan pengembangan dengan spesifikasi yang lebih memadai dari lingkungan pengembangan untuk perangkat lunak berbasis web pada umumnya dikarenakan kebutuhan VEX VM yaitu mesin virtual WebAssembly (*wasm*) yang mengeksekusi kode *smart contract* secara hirarkis. Selanjutnya dilakukan rancangan penyesuaian terhadap *smart contract* guna tata kelola sumber daya. Harus dipastikan juga mekanisme interaksi terhadap *blockchains*, baik melalui *Command Line Interface* (CLI) *cleos*, *RPC APIs*, atau aplikasi pemroses akses sumber daya (CPU dan NET) yang melakukan *staking* terhadap *VEX token*. Skema otorisasi dengan berbagai tingkat otoritas dapat diterapkan pada *smart contract* secara komprehensif.



Gambar 2. Metode penelitian

2.3 Perancangan aplikasi terdesentralisasi (*e-voting*)

Terlebih dahulu dirumuskan skenario penerapan aplikasi *e-voting*, dimana merupakan aplikasi yang digunakan untuk pemilihan kepala negara yang diselenggarakan oleh masing-masing negara bagian. Di masing-masing negara bagian, warga setempat menjadi pemilih sekaligus penyelenggara pemilihan kepala negara. Proses pemungutan suara dilakukan di masing-masing negara bagian. Kemudian rekapitulasi perhitungan suara diperoleh dari pemungutan suara yang telah berlangsung, menggunakan aplikasi *e-voting* terdesentralisasi yang menerapkan Vexanium *blockchain*. Detail mekanisme *e-voting* dijabarkan sebagai berikut:

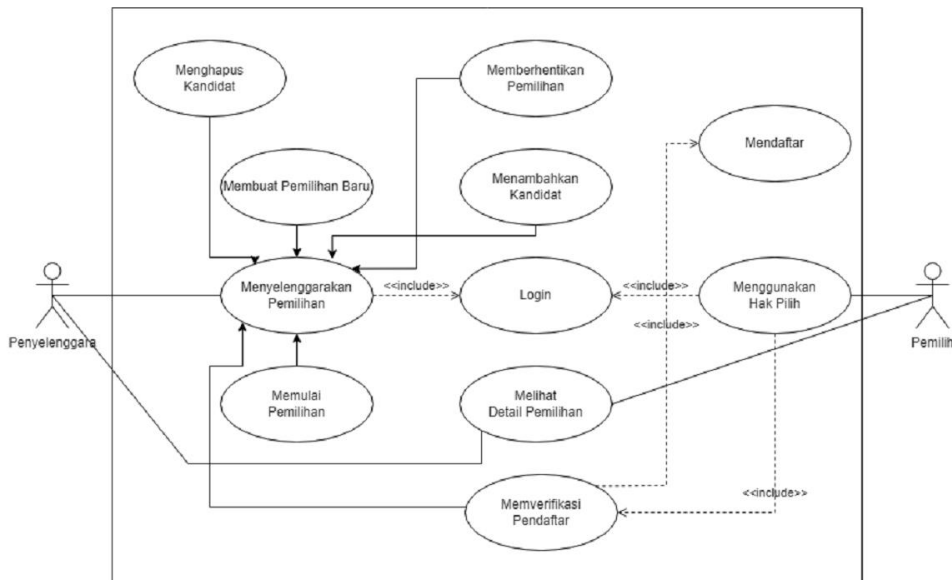
1. Tahap persiapan
 - a. Penyelenggara harus memiliki alamat dan private key pada jaringan Vexanium *blockchain*.
 - b. Data calon kepala negara diinputkan oleh masing-masing penyelenggara pada aplikasi terdesentralisasi (*e-voting*).
 - c. Agar dapat menjadi pemilih, warga harus melakukan pendaftaran pada aplikasi terdesentralisasi (*e-voting*). Setelah terdaftar maka pemilih akan memiliki otoritas untuk menggunakan alamat pada jaringan Vexanium *blockchain*.
 - d. Pada saat pemilihan, terlebih dahulu pemilih melakukan input ID yang telah terdaftar sebelumnya.

- e. Penyelenggara melakukan verifikasi pemilih sesuai ID yang dimasukkan. Jika ID terverifikasi maka pemilih dapat menggunakan hak suaranya.
- 2. Tahap proses pemungutan suara
 - a. Pemilih menggunakan suara dengan memilih calon kepala negara.
 - b. Hak suara yang sudah digunakan tidak dapat digunakan kembali.
- 3. Tahap penyelesaian voting
 - a. Penyelenggara menghentikan proses pemungutan suara.
 - b. Pemilih yang belum menggunakan hak suara tidak lagi dapat menggunakan hak suaranya.
 - c. Hasil rekapitulasi penghitungan suara ditampilkan pada aplikasi terdesentralisasi (*e-voting*).

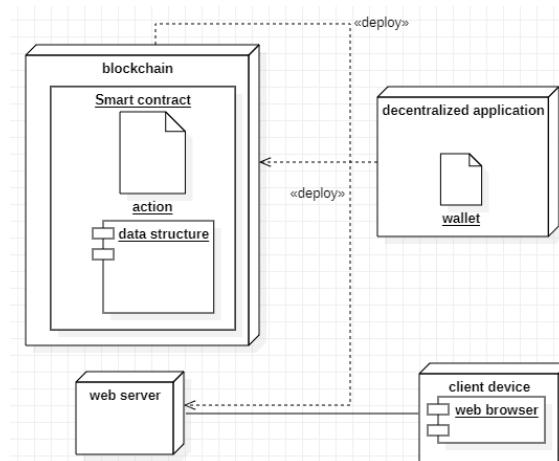
Interaksi yang dimungkinkan dari pengguna terhadap aplikasi terdesentralisasi (*e-voting*) berikut eksekusi arsitektur aplikasi terdesentralisasi (*e-voting*) pada menerapkan Vexanium *blockchain* ditunjukkan dengan UML *use case diagram* dan *deployment diagram* pada gambar 3 dan gambar 4.

2.4. Pengembangan aplikasi terdesentralisasi (*e-voting*)

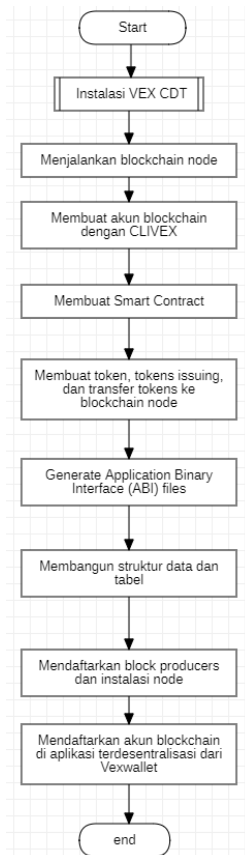
Tahapan pengembangan aplikasi terdesentralisasi (*e-voting*) ditunjukkan pada gambar 5.



Gambar 3. *Use case diagram* aplikasi terdesentralisasi (*e-voting*)



Gambar 4. *Deployment diagram* arsitektur aplikasi terdesentralisasi (*e-voting*)



Gambar 5. Tahapan pengembangan aplikasi terdesentralisasi (e-voting)

2.4.1 Pembangunan struktur data pada smart contract

Struktur data *smart contract* terdiri atas tabel dan *struct* [11]. Pada kasus *e-voting* dengan scenario pemilihan kepala negara ini dibangun satu tabel dan empat *struct*. *Struct* dalam *smart contract* nantinya akan tersimpan di dalam *blockchain*. Tabel akan berisi *record entity* yang melakukan transaksi (*action*) pada *smart contract* [11]. Salah satu dari pembangunan *struct* pada *smart contract* aplikasi terdesentralisasi (*e-voting*) ditunjukkan pada gambar 6, dimana terlihat bahwa *struct* terindeks oleh *daemon core* dari Vexanium (EOS.IO).

```

struct [[eosio::table]] pemilihan
{
    name          id_pemilihan;
    name          account_penyelenggara;
    string        nama_penyelenggara;
    string        judul_pemilihan;
    string        deskripsi;

    info_s       info_status;

    int8_t       kategori_pemilih;

    int8_t       kategori_hak_suara;
    int8_t       kuantitas_hak_suara;
    int8_t       sisa_hak_suara;

    dataopt      data;

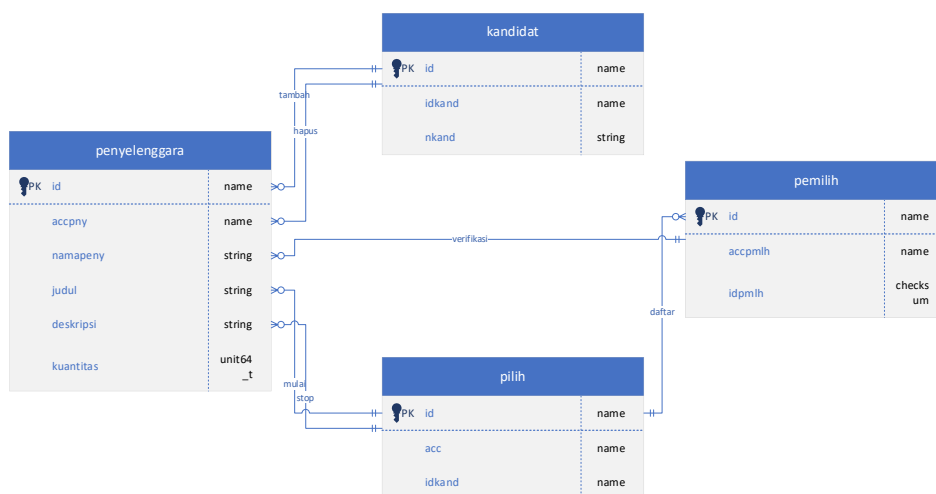
    uint64_t primary_key() const { return id_pemilihan.value; }
    uint64_t by_secondary() const { return account_penyelenggara.value; }
};

typedef eosio::multi_index<"evoting"_n, pemilihan, eosio::indexed_by<"byacc"_n, eosio::const_mem_fun<pemilihan, uint64_t, &pemilihan::by_secondary>> pemilihan_index;
    
```

Gambar 6. Contoh pembangunan struct

2.4.2 Action pada smart contract

Action di dalam *smart contract*, pada prinsipnya serupa dengan *query* pada basis data. *Action* terjadi pada *struct* di dalam *smart contract*. *Action* memiliki nama dan parameter, dimana parameter merupakan *fields* pada *struct*. *Fields* sendiri pada prinsipnya merupakan atribut pada tabel basis data. *Action* berikut parameter pada *smart contract* dari aplikasi terdesentralisasi (*e-voting*) ditunjukkan pada Entity Relationship Diagram (ERD) gambar 7. Sedangkan salah satu *action* pada *smart contract* dari aplikasi terdesentralisasi (*e-voting*) ditunjukkan pada gambar 8.



Gambar 7. Action berikut parameter pada smart contract dari aplikasi terdesentralisasi (e-voting)

```

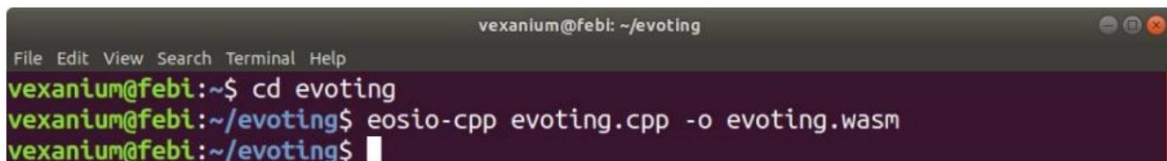
30
31 class [[eosio::contract("evoting")]] evoting : public eosio::contract {
32
33     public:
34
35         evoting(name receiver, name code, datastream<const char *> ds) : contract(receiver, code,
36             ds), bysecond(receiver, receiver.value)
37         {}
38
39         // action : membuat pemilihan baru
40         [[eosio::action]]
41         void buatpmlhn(
42             name id,
43             name accpeny,
44             string namapeny,
45             string judul,
46             string deskripsi,
47             int8_t katpem,
48             int8_t katsu,
49             uint64_t kuantitas
50         ){
51             require_auth(accpeny); // butuh hak akses sebagai actor action : account tersebut yang
52             // akan melakukan action ini
53
54             auto i = bysecond.find(id.value); //mencari : row id pemilihan pada table menggunakan
55             // parameter id
56
57             check(i == bysecond.end(), "Tidak dapat menggunakan id pemilihan yang sama, masukkan id
58             pemilihan baru!"); // cek : id pemilihan terdapat dalam table atau tidak
59         }
60     };
61 };

```

Gambar 8. Salah satu *action* pada *smart contract* dari aplikasi terdesentralisasi (*e-voting*)

2.4.3 Kompilasi kode *smart contract* dari aplikasi terdesentralisasi (*e-voting*)

Kompilasi kode *smart contract* dilakukan pada VEX.CDT. Keluaran dari proses kompilasi kode *smart contract* adalah file-file ABI dan wasm yang selanjutnya akan *deployed* di *contract account* pada Vexanium *blockchain*. Kompilasi kode *smart contract* dari aplikasi terdesentralisasi (*e-voting*) pada VEX.CDT ditunjukkan pada gambar 9.



```

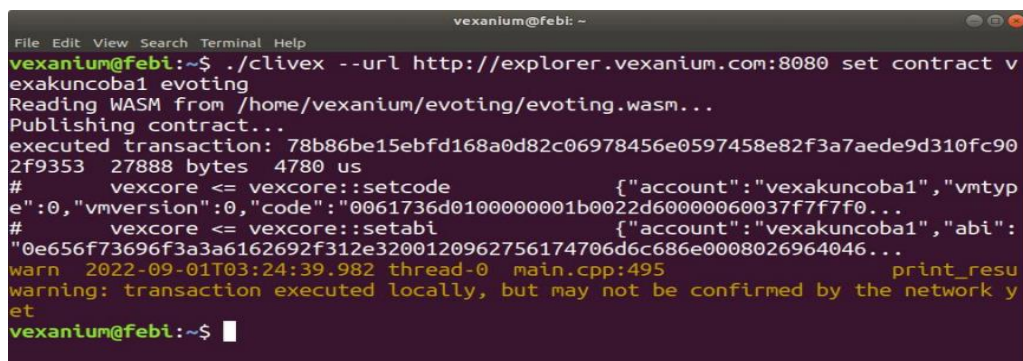
vexanium@febi: ~/evoting
File Edit View Search Terminal Help
vexanium@febi:~$ cd evoting
vexanium@febi:~/evoting$ eosio-cpp evoting.cpp -o evoting.wasm
vexanium@febi:~/evoting$

```

Gambar 9. Kompilasi kode *smart contract* dari aplikasi terdesentralisasi (*e-voting*)

2.4.4 *Smart contract deployment* ke Vexanium *blockchain*

Smart contract deployment ke Vexanium *blockchain* dilakukan menggunakan *./clivex terminal*, seperti ditunjukkan pada gambar 10.



```

vexanium@febi: ~
File Edit View Search Terminal Help
vexanium@febi:~$ ./clivex --url http://explorer.vexanium.com:8080 set contract vexakuncoba1 evoting
Reading WASM from /home/vexanium/evoting/evoting.wasm...
Publishing contract...
executed transaction: 78b86be15ebfd168a0d82c06978456e0597458e82f3a7aede9d310fc90
2f9353 27888 bytes 4780 us
#   vexcore <= vexcore::setcode           {"account":"vexakuncoba1","vmtype":0,"vmversion":0,"code":"0061736d0100000001b0022d60000060037f7f7f0..."}
#   vexcore <= vexcore::setabi            {"account":"vexakuncoba1","abi":"0e656f73696f3a3a6162692f312e3200120962756174706d6c686e0008026964046..."}
warn 2022-09-01T03:24:39.982 thread-0 main.cpp:495 print_resu
warning: transaction executed locally, but may not be confirmed by the network yet
vexanium@febi:~$

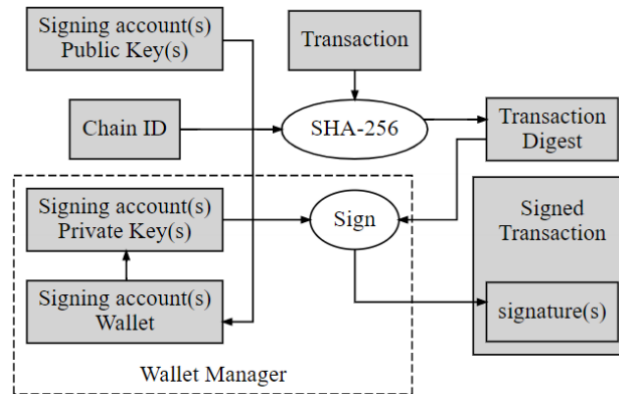
```

Gambar 10. *Smart contract deployment* ke Vexanium *blockchain*

Seperti ditunjukkan pada gambar 9 dan gambar 12, *smart contract* memiliki nama (*evoting*), akun (*vexakuncoba1*), dan *blockchain node* dengan alamat <http://explorer.vexanium.com:8080>. Akun dibuat pada aplikasi Vexwallet Desktop, dimana akun tersebut memiliki *public key* dan *private key*.

2.5 Implementasi SHA-256 pada aplikasi terdesentralisasi (*e-voting*) [12]

SHA-256 diterapkan pada aplikasi terdesentralisasi (*e-voting*) yang *deployed* di jaringan Vexanium *blockchain* untuk transaksi dan *chain ID* yang menghasilkan *Transaction Digest*. *Transaction Digest* selanjutnya ditandatangani secara digital untuk menghasilkan *Signed Transaction*. Akun, yang ditunjukkan pada gambar 11 ditandatangani secara digital menggunakan *public key* untuk menjadi *wallet*, dimana kemudian *wallet* ditandatangani kembali secara digital menggunakan *private key* guna menghasilkan *Signed Transaction*. Skema penerapan SHA-256 pada Vexanium *blockchain* ditunjukkan pada gambar 11.

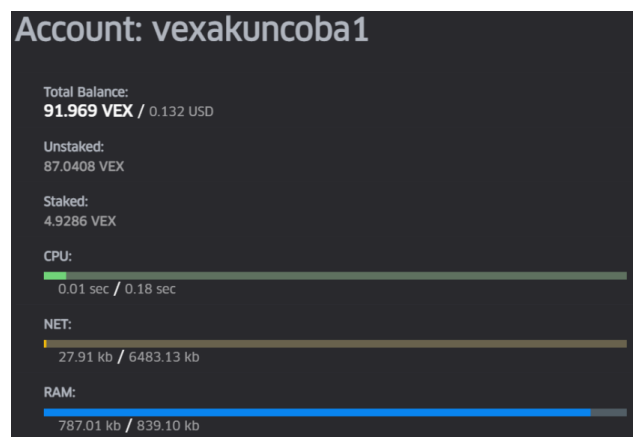


Gambar 11. Penerapan SHA-256 pada Vexanium *blockchain*

Selain itu, SHA-256 juga digunakan untuk enkripsi identitas pemilih (*voters*) pada aplikasi terdesentralisasi (*e-voting*) menjadi bentuk *message digest* dari fungsi *hash*. Hal ini merupakan persyaratan agar identitas pemilih dapat diverifikasi oleh penyelenggara.

3. HASIL DAN PEMBAHASAN

Aplikasi terdesentralisasi (*e-voting*), dikembangkan sebagai aplikasi berbasis *web*, sehingga sebelum meletakkan aplikasi ke jaringan Vexanium *blockchain* terlebih dahulu dilakukan *deployment* ke *server* aplikasi (Apache) dan *server* basis data (PostgreSQL). Aplikasi terdesentralisasi (*e-voting*) diletakkan ke jaringan Vexanium *blockchain* dengan akses menggunakan aplikasi Vexwallet Desktop dan akun. Jaringan Vexanium *blockchain* dan jaringan *blockchain* pada umumnya serupa dengan layanan *virtual server* atau layanan *cloud*, dimana setiap akun yang dibuat dengan aplikasi Vexwallet Desktop akan diberi sumber daya terbatas seperti halnya pada layanan *virtual server* atau layanan *cloud*. Sumber daya yang dimiliki akun *vexakuncoba1* pada Vexanium *blockchain* ditunjukkan pada gambar 12.



Gambar 12. Sumber daya akun *vexakuncoba1* pada Vexanium *blockchain*

Sumber daya berupa CPU, NET, dan RAM diperoleh dan dapat ditingkatkan dengan melakukan *staking* atau melakukan pembelian baik dengan *cryptocurrency* maupun *traditional currency*. Halaman muka (*home*) dari aplikasi terdesentralisasi (*e-voting*) ditunjukkan pada gambar 13. Detail dari Data Pemilihan, Data Kandidat, dan Data Pemilih dapat ditampilkan pada aplikasi terdesentralisasi (*e-voting*), seperti ditunjukkan pada gambar 14. Pada laman Lihat Detail yang ditunjukkan pada gambar 14 tersebut juga merupakan tampilan

dimana pemilih dapat melakukan pemilihan. *Dashboard* untuk penyelenggara pemilihan ditunjukkan pada gambar 15, dimana pada aplikasi terdesentralisasi (*e-voting*) dapat digunakan untuk menyelenggarakan lebih dari satu proses pemilihan sekaligus menambahkan Data Pemilih, Data Kandidat, serta Verifikasi Pemilih. Saat proses pemilihan/voting dilakukan dimana aplikasi tersentralisasi telah *deployed* ke Vexanium *blockchain*, *state* transaksi pada aplikasi terdesentralisasi pada sisi *blockchain data block* ditunjukkan pada gambar 16.

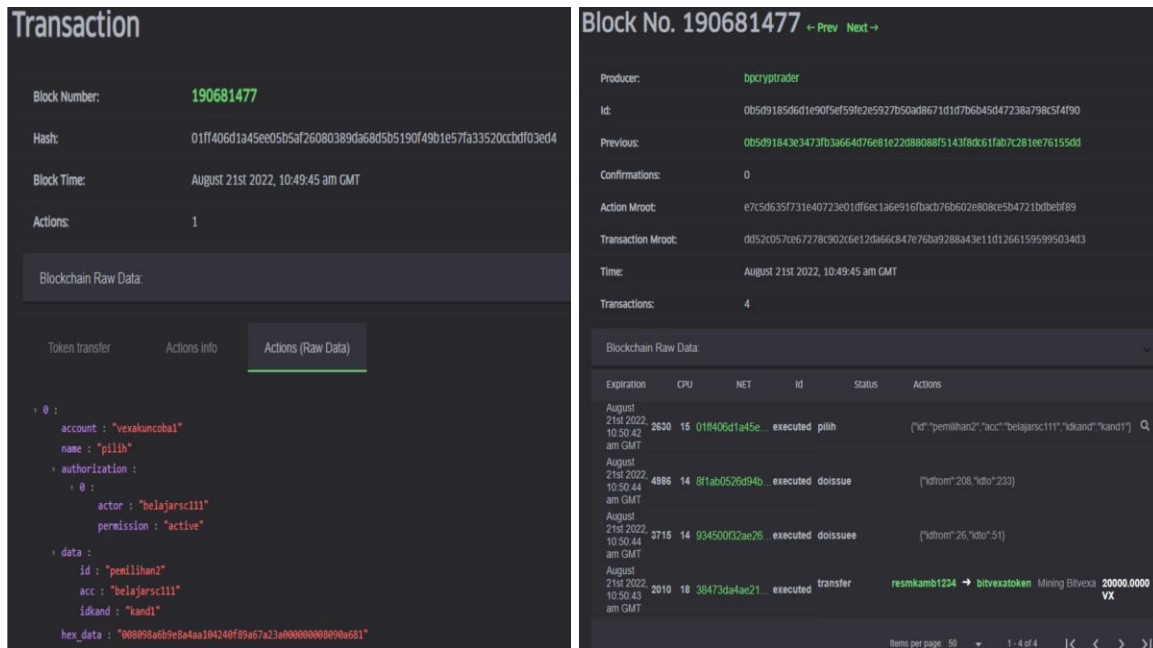
Tabel : evoting

Nomor	ID Pemilihan	Judul Pemilihan	Status Pemilihan
	pem11	w	Selesai
	pem2	b	Sedang berlangsung
	pem22	wa	Sedang dalam persiapan
	pemil3	Judul	Sedang dalam persiapan
	pemilihan1	Pemilihan member BTS Terfavorit	Sedang berlangsung
	pemilihan2	Pemilihan Ketua Organisasi AAAAA	Sedang berlangsung
	pemilihann	Pemilihan Ketua Organisasi PPP	Sedang dalam persiapan
	votingan	g	Sedang dalam persiapan

Gambar 13. Halaman muka (*home*) dari aplikasi terdesentralisasi (*e-voting*)

Gambar 14. Halaman Lihat Detail dari aplikasi terdesentralisasi (*e-voting*)

Gambar 15. *Dashboard* penyelenggara pemilihan



Gambar 16. Hubungan transaksi *e-voting* dan *blockchain data block* dari aplikasi terdesentralisasi pada Vexanium *blockchain*

Tabel 1. Keterangan dari elemen-elemen *blockchain data block*

No.	Elemen	Keterangan
1.	Block number	Merupakan elemen <i>magic number</i> pada <i>block</i>
2.	Hash	<i>Nonce</i>
3.	Block Time	Elemen <i>time</i> pada <i>block</i>
4.	Actions	Merupakan <i>actions counter</i>
5.	hex_data	<i>Hash</i> dari <i>action</i>
6.	Producer	Nama dari <i>block producer</i> dimana ditentukan oleh Vexanium <i>blockchain</i>
7.	id	<i>Hash</i> dari id <i>block</i> , dimana setiap <i>block</i> pada Vexanium <i>blockchain</i> memiliki id tersendiri. Id dihasilkan saat <i>block</i> diciptakan
8.	Previous	Merupakan <i>previous block hash</i> dari transaksi
9.	Action Mroot	<i>Hash Merkle root</i> dari <i>actions</i>
10.	Transaction Mroot	<i>Hash Merkle root</i> dari <i>transactions</i>
11.	Blockchain Raw Data	Merupakan elemen <i>transactions</i> dari <i>block</i> , berikut data <i>actions</i> , dan penggunaan sumber daya

4. PENUTUP

Aplikasi terdesentralisasi (*e-voting*) pada jaringan Vexanium *blockchain* telah berhasil dikembangkan dengan penerapan *smart contract*, sedemikian rupa dirumuskan teknik dan prosedur guna pengembangan aplikasi terdesentralisasi sekaligus *deployment* ke Vexanium *blockchain*. Guna pengembangan pada tahap selanjutnya maka Ricardian *contract* akan diterapkan. Kemudian diperlukan juga peningkatan keamanan pada level *block* dalam jaringan Vexanium *blockchain* setelah dilakukan pengujian terhadap aspek keamanan pada jaringan Vexanium *blockchain* tersebut, terutama pada hal terkait transaksi *e-voting*. Keamanan dapat ditingkatkan yakni dengan membangun mekanisme yang mencegah serangan *reentrancy*, menerapkan *true random generator* (TRNG) sebagai sumber keacakan *nonce*, serta menerapkan *safeguards* terhadap *frontrunning*. Sedangkan untuk fungsi dari *e-voting* itu sendiri juga akan ditingkatkan terutama pada hal terkait transparansi dan pembatasan fungsi-fungsi.

DAFTAR PUSTAKA

- [1] B. Wang *et al.*, "BLOCKKEYE: Hunting for DeFi Attacks on Blockchain," in *Proceedings - International Conference on Software Engineering*, IEEE Computer Society, May 2021, pp. 17–20. doi: 10.1109/ICSE-Companion52605.2021.00025.
- [2] Mulyati, U. Rahardja, M. Hardini, A. L. Al Nasir, and Q. Aini, "Taekwondo sports test and training data management using blockchain," in *2020 5th International Conference on Informatics and*

- Computing, ICIC 2020*, Institute of Electrical and Electronics Engineers Inc., Nov. 2020. doi: 10.1109/ICIC50835.2020.9288598.
- [3] A. Abubashim and C. C. Tan, "Smart Contract Designs on Blockchain Applications," *2020 IEEE Symposium on Computers and Communications (ISCC)*, Dec. 2020.
- [4] Institute of Electrical and Electronics Engineers, *2020 IEEE Symposium on Computers and Communications (ISCC)*.
- [5] S. Kitzler, F. Victor, P. Saggese, and B. Haslhofer, "Disentangling Decentralized Finance (DeFi) Compositions," *ACM Transactions on the Web*, vol. 17, no. 2, Mar. 2023, doi: 10.1145/3532857.
- [6] D. Suyitno, B. R. Aladhirus, and R. W. Wardhani, "Design and Implementation of Smart Card based Secure Key Storage the Blockchain E-voting Application," in *Proceeding - 1st International Conference on Information Technology, Advanced Mechanical and Electrical Engineering, ICITAMEE 2020*, Institute of Electrical and Electronics Engineers Inc., Oct. 2020, pp. 259–264. doi: 10.1109/ICITAMEE50454.2020.9398390.
- [7] I. Grigg, "The Ricardian Contract," *Proceedings of the First International Workshop on Electronic Contracting (WEC'04)*, 2004.
- [8] Vexanium, "Vexanium-whitepaper," Aug. 2019. Accessed: Jul. 24, 2023. [Online]. Available: <https://www.vexanium.com/files/whitepaper-vexanium-indonesian.pdf>
- [9] A. Levina, A. Plotnikov, and E. Ashmarov, "New Method of Hash Functions Analysis," in *2023 12th Mediterranean Conference on Embedded Computing (MECO)*, IEEE, Jun. 2023, pp. 1–5. doi: 10.1109/MECO58584.2023.10154990.
- [10] Vexanium, "Technical Feature," Jul. 2020. <https://belajar.vexanium.com/article/technical-feature/> (accessed Jul. 24, 2023).
- [11] N. Grech, S. Lagouvardos, I. Tsatiris, and Y. Smaragdakis, "Elipmoc: Advanced decompilation of Ethereum smart contracts," *Proceedings of the ACM on Programming Languages*, vol. 6, no. OOPSLA1, Apr. 2022, doi: 10.1145/3527321.
- [12] X. Yang, Y. Chen, and X. Chen, "Effective scheme against 51% attack on proof-of-work blockchain with history weighted information," in *Proceedings - 2019 2nd IEEE International Conference on Blockchain, Blockchain 2019*, Institute of Electrical and Electronics Engineers Inc., Jul. 2019, pp. 261–265. doi: 10.1109/Blockchain.2019.00041.