Smart Contract Blockchain pada *E-Voting*

Teresa Enades Hari Setia[1], Ajib Susanto[2]
[1,2,3]*Jurusan Informatika, Fakultas ILKOM, Universitas Dian Nuswantoro Semarang*
Gedung H, Jl. Imam Bonjol No.207 Semarang
E-mail : hari.setia0612@gmail.com[1], ajib.susanto@dsn.dinus.ac.id[2]

Abstract— The technology that is increasingly developing in the current era of globalization is e-voting. What is often done by the people of Indonesia in conducting e-voting is to have a chairman in a position the biggest example is the president. With e-voting it will be very easy to do. But there are always problems that occur in the results of the election of candidates who can be manipulated by anyone. So that from the problems that occur conducted research on the results of the election to reduce fraud in the election. Therefore, the authors conducted a security test of election results, by testing with the programming language Solidity and smart contract to produce a unique code in each new election. So that manipulation of election results will not be possible because each voter has only one account and one address blockchain. From the results of the test the author can conclude that the smart contract blockchain can be used to prove the results of a safe choice and create e-voting that is honest.

***Abstrak*— Teknologi yang semakin berkembang di era globalisasi saat ini salah satunya yaitu *e-voting.* Hal yang sering dilakukan oleh masyarakat indonesia dalam melakukan *e-voting* adalah memiliki ketua dalam suatu jabatan contoh paling besar yaitu presiden. Dengan adanya *e-voting* akan sangat mudah dilaukan. Tetapi selalu ada permasalahan yang terjadi pada hasil pemilihan kandidat yang bisa dimanipulasi oleh siapa saja. Sehingga dari permasalahan yang terjadi dilakukan penelitian terhadap hasil pemilihan untuk mengurangi kecurangan pada pemilihan. Oleh kaena itu, penulis melakukan pengujian pengamanan hasil pemilihan, Dengan melakukan pengujian dengan bahasa pemrogaman *Solidity* dan *smart contract* menghasilkan *code* unik pada setiap pemilihan baru. Sehingga manipulasi hasil pemilihan tidak akan bisa dilakukan karena setiap pemilih hanya memiliki satu *account* dan satu address *blockchain.* Dari hasil pengujian penulis lakukan dapat disimpulkan bahwa *smart contract blockchain* bisa digunakan untuk membuktikan hasil peilihan yang aman dan menciptakan *e-voting* yang bersifat jurdil.**

Kata Kunci—Smart contract, blockchain, Solidity, e-voting

I. PENDAHULUAN

Salah satu tahap pelaksanaan pemilihan umum adalah pemungutan suara atau *voting.* Secara umum pemungutan suara di laksanakan banyak negara secara rahasia pada tempat khusus yang di persiapkan untuk pelaksanaan pemungutan suara [1]. *E-voting* atau *electronic voting* secara sederhana dapat diartikan sebagai penggunaan hak pilih dalam sebuah pemilu namun dengan menggunakan teknologi secara elektronik[2]. *E-Voting* digunakan untuk menghimpun aspirasi dari seluruh elemen masyarakat, dan kemudian menemukan jalan keluar yang dianggap paling baik untuk menyelesaikan permasalahan pada manipulasi pada hasil *e-voting* [3].

Blockchain adalah *database* yang mengamankan penyimpanan berbagai jenis data dalam jaringan terdesentralisasi. Data pada *blockchain* disimpan di dalam *block*. Setiap *block* mempunyai hash dari *block* sebelumnya. Dengan hash tersebut akan mudah untuk mendeteksi jika ada perubahan [4]. *Blockchain* menggunakan banyak sekali fungsi hash dalam prosesnya. *Hash* tersebut membantu *Blockchain* untuk mendeteksi apakah ada data yang diubah oleh seseorang atau ada data yang berubah karena kesalahan jaringan. Dari *hash* yang sudah digunakan dalam *blockchain, hash* selalu mempunyai ukuran yang sama, dua string yang identik akan menghasilkan *hash* yang sama, dua *string* yang berbeda akan menghasilkan *hash* yang berbeda, dan membuat *string* yang cocok dengan *hash* yang diberikan itu sangatlah sulit [4].

Blockchain sendiri telah digunakan dalam sistem *Ethereum. Ethereum* adalah salah satu yang paling cocok, karena konsistensi penggunaannya yang luas, dan penyediaan logika *smart contract* [5]. *Smart Contract* adalah aplikasi yang digunakan pada buku besar *blockchain* dan dieksekusi secara mandiri sebagai bagian dari validasi transaksi. Untuk menggunakan *smart contract* di *Ethereum*, transaksi pembuatan khusus dijalankan, yang memperkenalkan *contract* ke *blockchain.* Selama prosedur ini, kontrak diberikan alamat unik, dalam bentuk pengidentifikasi 160-bit, dan kodenya diunggah ke *blockchain*. Setelah berhasil dibuat, kontrak pintar terdiri dari alamat kontrak, saldo *contract*, kode yang dapat dieksekusi yang ditentukan sebelumnya, dan status. Pihak yang berbeda kemudian dapat berinteraksi dengan kontrak tertentu dengan mengirimkan transaksi yang melibatkan kontrak ke alamat *contract*, yang diketahui [6].

Solidity digunakan dalam bahasa untuk menggunakan kontrak dengan *Ethereum blockchain*. *Solidity* adalah bahasa tingkat tinggi yang digunakan untuk menerapkan smart *contract* [7]. Bahasa *Solidity* ditulis dalam *file .sol*,

dan *Solidity* sangat mirip dengan javascript seperti *syntax* nya yang sangat mirip dengan javascript *Solidity* juga mirip *JavaScript*, diketik secara statis, mendukung pewarisan dan polimorfisme, serta perpustakaan dan tipe kompleks yang ditentukan pengguna. Ketika menggunakan *Solidity* untuk pengembangan *contract*, *contract* disusun mirip dengan kelas dalam bahasa pemrograman berorientasi objek. Kode *contract* terdiri dari variabel dan fungsi yang membaca dan memodifikasi ini, seperti dalam pemrograman imperatif tradisional [8]. Dalam jaringan *Ethereum*, semua operasi setidaknya secara real time, dan semua blok ditulis dalam rantai pamungkas dengan imbalan beberapa Eter (mata uang dari jaringan *Ethereum*). Ini diberikan sebagai hadiah kepada para penambang, yang melakukan operasi penulisan dan validasi ini, yang mahal dalam hal waktu dan kekuasaan perhitungan. Peneliti telah mendefinisikan kontrak pintar seperti yang sudah di sebutkan secara singkat di atas [9].

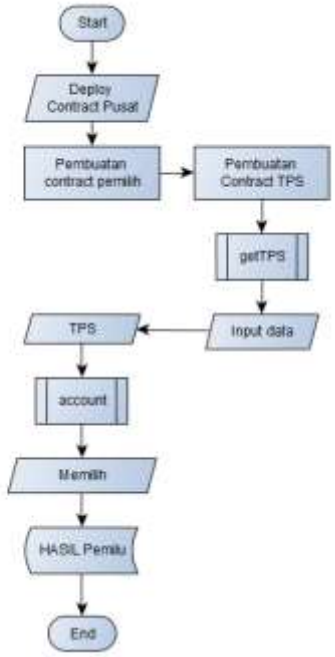
Smart *Contract* adalah prosedur tertanam yang disimpan dengan data yang ditindak lanjuti. Men-*debug* kontrak pintar adalah tugas yang sangat sulit karena sekali dikerahkan, kode tidak dapat dieksekusi kembali dan memeriksa atribut sederhana tidak mudah dilakukan karena data dikodekan [10].

II. METODE PENELITIAN

Pada penelitian ini akan dilakukan pengujian keamanan menggunakan smart contracr *blockchain* pada e-voting dengan menggunakan bahasa solidty. Langkah pertama dalam proses ini adalah dengan men-debug *contract* pusat pada *Solidity*. Setelah itu pemilih melakukan pengisian identitas, setiap pemilih yang ingin memilih akan mendapat account address *blockchain*. Langkah terakhir yang dilakukan adalah dengan memilih kandidat menggunakan alamat *blockchain* pada masing-masing kandidat.

Metode yang digunakan yaitu menerapkan teknologi *blockchain* dengan *smart contract* pada *e-voting*. Pada penelitian ini, ada beberapa komponen penting yang harus ada yaitu *account* untuk setiap pemilih. Komponen tersebut adalah bahan utama untuk melakukan proses *e-voting*. Pada saat proses disimpan disitulah transaksi akan dicatat segala informasi yang sudah dilakukan yaitu id, nama dan kode unik. Dan memeriksa apakah orang yang sama bisa memilih lagi atau tidak. Adanya Prosedur Pemilihan Kandidat dengan *Smart Contract :*

1. Melakukan *deploy contract* Pusat
2. Melakukan pembuatan *contract* Pemilih untuk setiap pemilih. dilakukan pada *contract* PUSAT.
3. Melakukan pembuatan *contract* TPS untuk mengarahkan setiap *contract* Pemilih yang telah di generasi ke TPS yang telah ditentukan. dilakukan pada *contract* TPS
4. Melakukan pembuatan *contract* kandidat untuk setiap kandidat yang termasuk pada pemilihan. dilakukan pada *contract* Pusat
5. Melakukan pengambilan alamat *contract* TPS pada get tps yang ada pada *contract* Pusat.
6. Melakukan penginputan data mengisi identitas diri pada menu pemilihan baru dengan memasukkan alamat *blockchain* tps.
7. Melakukan pengecekan pada *contract* TPS akan berisi list array pemilih yang termasuk pada TPS tersebut hal ini sama pada proses yang dilakukan. pemilih telah ditentukan lokasi TPSnya.
8. Account merupakan address yang unik yang dimiliki setiap pemilih yang tidak generasi oleh *contract* manapun. sedangkan yang di generasi adalah *contract* Pemilih untuk setiap pemilih. hal ini sama dengan proses bisnis yang ada, pemilih hanya berhak dengan satu surat suara.
9. Melakukan pemilihan kandidat
10. Melihat hasil pemilu, hasil yang didapat dengan melakukan perhitungan setiap *contract* kandidat. setiap pemilih menentukan pilihannya maka suara yang dimiliki dari setiap *contract* kandidat akan bertambah dan variabel suara tersebut akan dibandingkan dengan *contract* kandidat yang ada, dan diambil pemenang berdasarkan suara terbanyak.



Gambar 1 Prosedur pemilihan pada *smart contract*

III. HASIL DAN PEMBAHASAN

A. Contract pusat

Contract pusat dari program ini digunakan untuk menjalankan proses dari *e-voting* dengan alur proses:

1. Mengisi tempat TPS agar mendapatkan address tps pada get tps
2. Mengisi identitas pemilih dengan ktp, nama, dan alamat *blockchain* tps
3. Mengisi nama calon kandidat pada *contract* kandidat
4. gettps untuk memanggil alamat *blockchain* tps yang sudah terdaftar pada *contract* tps (TPS1 dimulai dari angka nol, dikarenakan meggunakan array).

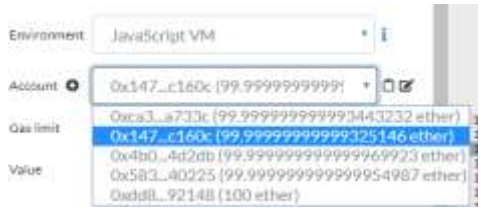
5. getkandidat untuk memanggil memanggil alamat *blockchain* kandidat yang sudah terdaftar pada *contract* kandidat (kandidat satu dimulai dari angka nol, dikarenakan meggunakan *array*).
6. getpemilih untuk memanggil memanggil alamat *blockchain* dari setiap pemilih yang sudah terdaftar pada *contract* pemilih (pemilih pertama dimulai dari angka nol, dikarenakan meggunakan *array*).

Gambar 2 Pengisian data pada *contract* pusat

Untuk pemilih baru digunakan untuk pemilih mengisi identitas. Untuk tambah kandidat kita menambahkan calon kandidat yang akan dipilih. Pada tambahtps mengisi tempat tps pemilih seperti gambar 2. Untuk kolom *get* kandidat, gettps dan get pemilih adalah tempat dimana pengambilan alamat *blockchain*. Dan untuk kolom pemenang tempat dimana hasil akan dilihat siapa yang akan terpilih.

B. Contract Pemilih

Pada *contract* pemilih, pemilih juga memiliki kode unik dan yang berbeda – beda agar tidak terjadi pemilihan lebih dari satu kali. sepeti pada gambar menunjukkan pemilih pertama dengan alamat *blockchain* dan *account* yang didapat. Seperti gambar 3. dan pada gambar 4 adalah bentuk dari kode unik setiap pemilih. Setiap pemilih diberi kode unik satu untuk satu kali pemilihan, mencegah adanya manipulasi hasil *voting* yang sudah ada.



Gambar 3 Account yang dimiliki setiap pemilih

Pemilih pertama dengan account 0x147

Pemilih kedua dengan account 0x4b0

Pemilih ketiga dengan account 0x583

Gambar 4. Setiap pemilih memiliki *account* yang berbeda-beda

C. Pengujian

Berdasarkan pengujian *smart contract* dengan bahasa pemrogaman *Solidity* hasil yang didapat dari teknologi *blockchain* akan dilakukan pada *contract* pusat. pemilihan dilakukan dengan cara memasukkan *address blockchain* kandidat pada *contract* pemilih. Pada alamat *blockchain* kandidat terdapat pada pemilih pertama dan pemilih kedua ditunjukkan untuk kandidat Jokowi, sedangkan pemilih ketiga ditunjukkan untuk kandidat Prabowo. Hasil pemilihan tidak dihitung dan tidak menunjukkan nomor urut yang terpilih melainkan langsung menampilkan nama calon kandidat. Pembuktian yang dilakukan bahwa kandidat nomor urut satu terpilih terlihat pada pemilih pertama dan kedua lebih banyak suara dari pada pemilih ketiga.

Gambar 5. Hasil *e-voting*

IV. KESIMPULAN

Berdasarkan pengujian yang sudah penulis lakukan, dapat disimpulkan bahwa *e-voting* berbasis teknologi *blockchain* ini mampu menjadi penyimpan data voting yang aman. manipulasi hasil dan pemilihan tidak bisa dilakukan lebih dari satu kali oleh siapapun dikarenakan pegujian menggunakan smart *contract* dengan bahasa pemrogaman *Solidity*. Maka teknologi *blockchain* pada *e-voting* tidak mudah diretas karena pada setiap pemilih memiliki alamat unik *blockchain* masing-masing yang tidak akan bisa digunakan dua kali. Dan pada pengujian dibuktikan bahwa sistem yang diimplementasikan dengan *smart contract* mampu menangani proses pada *e-voting.*

DAFTAR PUSTAKA

[1] W. Y. Dio Lavarino, "RANCANG BANGUN E – VOTING BERBASIS WEBSITE DI UNIVERSITAS NEGERI SURABAYA," vol. 6, pp. 72–81, 2016.

[2] F. P. Juniawan, "RSA implementation for data transmission security in BEM chairman E-voting Android based application," *Proc. - 2016 1st Int. Conf. Inf. Technol. Inf. Syst. Electr. Eng. ICITISEE 2016*, pp. 93–98, 2016.

[3] M. Ridwan, Z. Arifin, and Y. Yulianto, "Rancang Bangun E-Voting Dengan Menggunakan Keamanan Algoritma Rivest Shamir Adleman (RSA) Berbasis Web (Studi Kasus : Pemilihan Ketua Bem Fmipa)," *Inform. Mulawarman J. Ilm. Ilmu Komput.*, vol. 11, no. 2, p. 22, 2016.

[4] Emurgo indonesia, *Pengantar Teknologi Blockchain*. Dicoding Indonesia, 2018.

[5] A. D. E. Rayendra, "Rancang Bangun Sistem E-Voting dengan menggunakan teknologi *blockchain*," *Ranc. Bangun Sist. E-Voting dengan menggunakan Teknol. blockchain*, 2017.

[6] M. Wohrer and U. Zdun, "Smart *contracts*: Security patterns in the *Ethereum* ecosystem and *Solidity*," *2018 IEEE 1st Int. Work. Blockchain Oriented Softw. Eng. IWBOSE 2018 - Proc.*, vol. 2018-Janua, pp. 2–8, 2018.

[7] B. K. Mohanta, S. S. Panda, and D. Jena, "An Overview of Smart *Contract* and Use Cases in *Blockchain* Technology," *2018 9th Int. Conf. Comput. Commun. Netw. Technol. ICCCNT 2018*, pp. 1–4, 2018.

[8] *Solidity*, "*Solidity*." 2016.

[9] A. K. Koç, E. Yavuz, U. C. Çabuk, and G. Dalkiliç, "Towards secure e-voting using *Ethereum blockchain*," *6th Int. Symp. Digit. Forensic Secur. ISDFS 2018 - Proceeding*, vol. 2018-Janua, pp. 1–6, 2018.

[10] S. Bragagnolo, H. Rocha, M. Denker, and S. Ducasse, "SmartInspect: *Solidity* smart *contract* inspector," *2018 IEEE 1st Int. Work. Blockchain Oriented Softw. Eng. IWBOSE 2018 - Proc.*, vol. 2018-Janua, pp. 9–18, 2018.