**IMPLEMENTASI BLOCKCHAIN DAN SMART CONTRACT SEBAGAI SISTEM PENYIMPANAN SERTIFIKAT VAKSIN DAN TES COVID-19**

*IMPLEMENTATION OF BLOCKCHAIN AND SMART CONTRACT AS COVID-19 TEST AND VACCINE CERTIFICATE STORAGE SYSTEM*

**TUGAS AKHIR**

Disusun sebagai syarat untuk memperoleh gelar

Sarjana Teknik pada Program Studi S1 Teknik Komputer

Universitas Telkom

**Oleh :**

**Dendi Arya Raditya Prawira Putra**

**1103184150**

****

**FAKULTAS TEKNIK ELEKTRO**

**UNIVERSITAS TELKOM**

**BANDUNG**

**2022**

**LEMBAR PENGESAHAN**

**TUGAS AKHIR**

**IMPLEMENTASI BLOCKCHAIN DAN SMART CONTRACT SEBAGAI SISTEM PENYIMPANAN SERTIFIKAT VAKSIN DAN TES COVID-19**

***IMPLEMENTATION OF BLOCKCHAIN AND SMART CONTRACT AS COVID-19 TEST AND VACCINE CERTIFICATE STORAGE SYSTEM***

**Telah disetujui dan disahkan sebagai Buku Tugas Akhir**

**Program Studi Teknik Elektro**

**Fakultas Teknik Elektro**

**Universitas Telkom**

**Disusun oleh:**

**DENDI ARYA RADITYA PRAWIRA PUTRA**

**1103184150**

|  |  |
| --- | --- |
| Pembimbing I  **Dr. Yudha Purwanto S.T., M.T.**  NIP. 02770066 | Pembimbing II  **Dr. Marisa W. Paryasto S.T., M.T**  NIP. 13750017 |

**Bandung, Juli 2022**

# LEMBAR PERNYATAAN ORISINALITAS

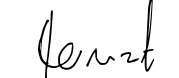
|  |  |  |
| --- | --- | --- |
| Nama | : | Dendi Arya Raditya Prawira Putra |
| NIM | : | 1103184150 |
| Alamat | : | Semarang, Jawa Tengah |
| No. Telepon | : | 081339419724 |
| Email | : | dendiaryar@gmail.com |

Menyatakan bahwa Tugas Akhir ini merupakan karya orisinal saya sendiri, dengan judul :

**IMPLEMENTASI BLOCKCHAIN DAN SMART CONTRACT SEBAGAI SISTEM PENYIMPANAN SERTIFIKAT VAKSIN DAN TES COVID-19**

***IMPLEMENTATION OF BLOCKCHAIN AND SMART CONTRACT AS COVID-19 TEST AND VACCINE CERTIFICATE STORAGE SYSTEM***

Atas pernyataan ini, saya siap menanggung resiko / sanksi yang dijatuhkan kepada saya apabila kemudian ditemukan adanya pelanggaran terhadap kejujuran akademik atau etika keilmuan dalam karya ini, atau ditemukan bukti yang menunjukkan ketidak aslian karya ini.

Bandung, 22 Juni 2022

(Dendi Arya Raditya Prawira Putra)

1103184150

# ABSTRAK

Keberadaan virus Covid-19 yang pertama kali diumumkan pada akhir tahun 2019, dengan waktu singkat mampu menyebar keseluruh dunia dan mengakibatkan pandemi global. Untuk mengatasi hal tersebut pemerintah menerapkan protokol yang mengintegrasikan hasil tes Covid-19 dan sertifikat vaksinasi ke dalam sebuah aplikasi, tujuannya agar individu dapat membuktikan bahwa dirinya terbebas dari infeksi Covid-19 dan dapat kembali berkegiatan dengan normal. Namun sistem yang tersentralisasi rentan terhadap *single point of failure* danmanipulasi data dari campur tangan pihak tertentu dikarenakan kurangnya transparansi. Oleh karenanyadiperlukan adanya alternatif solusi akan persoalan tersebut dan dapat menjamin keamanan serta memberi lebih banyak pilihan kepada pengguna untuk mengatur kerahasiaan data pribadi mereka.

Penggunaan *blockchain* dan *smart contract* dapat menjadi solusi dari persoalan tersebut. Dengan penggunaan teknologi *blockchain* dan *smart contract* proses pengelolaan data akan lebih transparan, dikarenakan *blockchain* hanya menggunakan proses *read* dan *write* dan setiap transaksi pada *blockchain* tercatat oleh setiap *node.* *Blockchain* juga dapat mencegah terjadinya *single point of failure* dikarenakan terdapat lebih dari satu sumber penyedia data.

Tugas akhir ini menghasilkan sebuah aplikasi berbasis *website* yang dibangun diatas jaringan *blockchain ethereum* dan memanfaatkan *smart contract* untuk mengeksekusi transaksi pada sistem. Berdasarkan pengujian *white box* dan *black box* dapat dipastikan bahwa aplikasi dan *smart contract* dapat berjalan dengan baik dan dari hasil *user acceptance testing* didapatkan hasil bahwa sistem dapat memenuhi kebutuhan *user.* Namun performa masih menjadi masalah bila dibandingkan dengan sistem tersentralisasi dikarenakan mekanisme konsesus pada sistem *blockchain*.

**Kata Kunci:** *Blockchain, Sertifikat digital,Blockchain, Smart contract*

# ABSTRACT

The existence of the Covid-19 virus, which was first announced at the end of 2019, in a short time was able to spread throughout the world and cause a global pandemic. To overcome this, the government implemented a protocol that integrates Covid-19 test results and vaccination certificates into an application, with the aim that individuals can prove that they are free from Covid-19 infection and can return to normal activities. However, a centralized system is prone to single point of failure and data manipulation from the intervention of certain parties due to a lack of transparency. Therefore, it is necessary to have alternative solutions to these problems and can ensure security and give users more choices to manage the confidentiality of their personal data.

The use of blockchain and smart contracts can be a solution to this problem. With the use of blockchain technology and smart contracts, the data management process will be more transparent, because the blockchain only uses a read and write process and every transaction on the blockchain is recorded by each node. Blockchain can also prevent a single point of failure because there is more than one source of data providers.

This final project produces a website-based application that is built on the ethereum blockchain network and utilizes smart contracts to execute transactions on the system. Based on white box and black box testing, it can be ascertained that the application and smart contract can run well and from the results of user acceptance testing, it is found that the system can meet user needs. However, performance is still a problem when compared to a centralized system due to the consensus mechanism in the blockchain system.

**Keywords**: Blockchain, Digital certificate, Blockchain, Smart contract

# Kata Pengantar

Assalamu'alaikum Wr. Wb.

Segala puji dan syukur penulis ucapkan kepada Allah SWT atas segala limpahan kasih, karunianya, dan kehendaknya sehingga penulis dapat menyelesaikan penelitian tugas akhir ini dengan judul “**IMPLEMENTASI BLOCKCHAIN DAN SMART CONTRACT SEBAGAI SISTEM PENYIMPANAN SERTIFIKAT VAKSIN DAN TES COVID-19”.**

Tugas akhir ini disusun sebagai syarat kelulusan program S1 Teknik Komputer, Fakultas Teknik Elektro, Universitas Telkom. Penulis mengucapkan banyak terima kasih kepada semua pihak yang telah mendukung penulis dalam menyusun Tugas Akhir ini.

Penulis menyadari bahwa Buku Tugas Akhir ini masih memiliki kekurangan dan sangat jauh dari kata sempurna. Oleh karena itu, penulis mengharpkan kritik maupun saran yang membangun. Demikian Buku Tugas Akhir ini disusun, penulis berharap Buku Tugas Akhir ini dapat memberikan manfaat bagi para pembaca.

|  |
| --- |
| Bandung, |
|  |
| Penulis |

Wassalamualaikum warahmatullahi wabarakatuh.

# UCAPAN TERIMAKASIH

Puji syukur penulis panjatkan kepada Allah SWT, atas segala petunjuk, rahmat, karunia, dan hidayahnya sehingga penulis dapat menyelsaikan Tugas Akhir ini dan lulus sebagai sarjana teknik dari prodi S1 Teknik Komputer Telkom University. Pada kesempatan ini penulis juga ingi berterima kasih sebesar – besarnya kepada :

1. Keluarga tercinta, terutama ayah, mamah dan bung atas dukungannya secara materi, moral dan bantuan lainnya yang tidak dapat disebutkan satu persatu.
2. Dosen pembimbing yaitu Bapak Dr. Yudha Purwanto S.T., M.T., selaku pembimbing satu dan juga Ibu Dr. Marisa W. Paryasto S.T., M.T., selaku pembimbing dua yang telah membimbing penulis selama pengerjaan tugas akhir ini agar dapat menghasilkan tugas akhir yang baik. Terima kasih atas seluruh ilmu yang telah diberikan selama masa studi dan proses bimbingan pengerjaan tugas akhir.
3. Seluruh dosen Telkom University yang telah mengajarkan pembimbing banyak ilmu selama menempuh studi di Telkom University.
4. Teman – teman Fajri, Damas, Nabil, dan keluarga cigan boys, yang sering mendengarkan keluh kesah penulis dan dapat memberikan saran masukan yang memotivasi. Terima kasih juga telah berbagi cerita semasa kuliah.

# DAFTAR ISI

[LEMBAR PENGESAHAN ii](#_Toc107162013)

[LEMBAR PERNYATAAN ORISINALITAS iii](#_Toc107162014)

[ABSTRAK iv](#_Toc107162015)

[ABSTRACK v](#_Toc107162016)

[Kata Pengantar vi](#_Toc107162017)

[UCAPAN TERIMAKASIH vii](#_Toc107162018)

[DAFTAR ISI viii](#_Toc107162019)

[DAFTAR GAMBAR x](#_Toc107162020)

[DAFTAR TABEL xi](#_Toc107162021)

[BAB 1 Pendahuluan 1](#_Toc107162022)

[1.1. Latar Belakang Masalah 1](#_Toc107162023)

[1.2. Rumusan Masalah 2](#_Toc107162024)

[1.3. Tujuan dan Manfaat 2](#_Toc107162025)

[1.4. Batasan Masalah 3](#_Toc107162026)

[1.5. Metode Penelitian 3](#_Toc107162027)

[BAB 2 Dasar Teori 4](#_Toc107162028)

[2.1. Blockchain 4](#_Toc107162029)

[2.2. Ethereum 4](#_Toc107162030)

[2.3. Akun Ethereum 5](#_Toc107162031)

[2.4. Transaksi pada Ethereum 5](#_Toc107162032)

[2.5. Smart Contract dan Solidity 6](#_Toc107162033)

[*2.6.* *Decentralized Application* 7](#_Toc107162034)

[2.7. *Brownie Python* 7](#_Toc107162035)

[*2.8.* *Digital Wallet dan Metamask* 8](#_Toc107162036)

[2.9. *Web3* JS 8](#_Toc107162037)

[BAB 3 Perancangan Sistem 9](#_Toc107162038)

[3.1. Analisis Kebutuhan Sistem 9](#_Toc107162039)

[3.1.1. Gambaran Alur Sistem Saat ini 9](#_Toc107162040)

[3.1.2. Perbandingan Sistem Tersentralisasi Dengan Sistem Terdesentralisasi Blockchain 11](#_Toc107162041)

[3.1.3. Kebutuhan Pengguna 11](#_Toc107162042)

[3.1.4. Peran Pengguna Pada Sistem Yang Diusulkan 12](#_Toc107162043)

[3.1.5. Analisis Kebutuhan Fungsional Dari Sistem Yang Diusulkan 13](#_Toc107162044)

[3.2. Desain Sistem 14](#_Toc107162045)

[3.2.1. Diagram Blok 14](#_Toc107162046)

[3.2.2. Perancangan *Smart Contract* 15](#_Toc107162047)

[3.2.3. Struktur *Smart Contract* 16](#_Toc107162048)

[3.2.4. *Entity Relational Diagram* Pada *Smart Contract* 17](#_Toc107162049)

[*3.2.5.* *Authentication* dan *Authorization Issuer* 18](#_Toc107162050)

[3.2.6. Proses Pendaftaran Issuer Oleh Regulator 22](#_Toc107162051)

[3.2.7. Proses Pendaftaran Sertifikat 24](#_Toc107162052)

[3.2.8. Proses Verifikasi Sertifikat 27](#_Toc107162053)

[3.2.9. *Data Flow* Diagram Level 0 29](#_Toc107162054)

[3.2.10. *Data* Flow Diagram Level 1 29](#_Toc107162055)

[3.2.11. *Data flow Diagram* level 2 30](#_Toc107162056)

[BAB 4 Pengujian dan Hasil 33](#_Toc107162057)

[4.1. Pengujian 33](#_Toc107162058)

[4.2. Uji Validitas Fungsional 33](#_Toc107162059)

[4.3. Pengujian performa 41](#_Toc107162060)

[4.4. UAT (User Acceptance Testing) 43](#_Toc107162061)

[BAB 5 Kesimpulan dan Saran 46](#_Toc107162062)

[5.1. Kesimpulan 46](#_Toc107162063)

[5.2. Saran 46](#_Toc107162064)

[Lampiran 49](#_Toc107162065)

# DAFTAR GAMBAR

[Gambar 2.1 Transaksi pada Ethereum merubah state pada EVM (Sumber : Dokumentasi Ethereum) 5](file:///D:\TUGAS%20AKHIR\TUGAS%20AKHIR%20Revisi%20Bimbingan.docx#_Toc107051467)

[Gambar 3.1 Alir Pendaftaran Existing vaksinasi covid-19 pada sentra vaksinasi dan tes covid-19 puskesmas kedungmundu ……………………………………………………………………………………………………..10](file:///D:\TUGAS%20AKHIR\TUGAS%20AKHIR%20Revisi%20Bimbingan.docx#_Toc107140036)

[Gambar 3.2 Use Case Diagram 13](#_Toc107140037)

[Gambar 3.3 Diagram blok dari sistem yang disusulkan 15](#_Toc107140038)

[Gambar 3.4 Diagram block inheritance dari smart contract yang dirancang 16](file:///D:\TUGAS%20AKHIR\TUGAS%20AKHIR%20Revisi%20Bimbingan.docx#_Toc107140039)

[Gambar 3.5 Entity Relational Diagram 17](#_Toc107140040)

[Gambar 3.6 ER Diagram Issuer Registry Smart Contract 17](file:///D:\TUGAS%20AKHIR\TUGAS%20AKHIR%20Revisi%20Bimbingan.docx#_Toc107140041)

[Gambar 3.7 ER diagram data pada certificate registry smart contract 18](file:///D:\TUGAS%20AKHIR\TUGAS%20AKHIR%20Revisi%20Bimbingan.docx#_Toc107140042)

[Gambar 3.8 Contract deployment oleh Regulator 19](file:///D:\TUGAS%20AKHIR\TUGAS%20AKHIR%20Revisi%20Bimbingan.docx#_Toc107140043)

[Gambar 3.9 Sequence Diagram pendaftaran issuer oleh regulator 20](file:///D:\TUGAS%20AKHIR\TUGAS%20AKHIR%20Revisi%20Bimbingan.docx#_Toc107140044)

[Gambar 3.10 Prosesn autentikasi issuer 21](file:///D:\TUGAS%20AKHIR\TUGAS%20AKHIR%20Revisi%20Bimbingan.docx#_Toc107140045)

[Gambar 3.11 Verifikasi signature 22](file:///D:\TUGAS%20AKHIR\TUGAS%20AKHIR%20Revisi%20Bimbingan.docx#_Toc107140046)

[Gambar 3.12 Proses pembuatan public private key oleh issuer 23](file:///D:\TUGAS%20AKHIR\TUGAS%20AKHIR%20Revisi%20Bimbingan.docx#_Toc107140047)

[Gambar 3.13 Flowchart registrasi issuer oleh regulator 24](file:///D:\TUGAS%20AKHIR\TUGAS%20AKHIR%20Revisi%20Bimbingan.docx#_Toc107140048)

[Gambar 3.14 Flowchart pendaftaran sertifikat Covid-19 25](file:///D:\TUGAS%20AKHIR\TUGAS%20AKHIR%20Revisi%20Bimbingan.docx#_Toc107140049)

[Gambar 3.15 Flowchart pendaftaran sertifikat Covid-19 26](file:///D:\TUGAS%20AKHIR\TUGAS%20AKHIR%20Revisi%20Bimbingan.docx#_Toc107140050)

[Gambar 3.16 Flowchart verifikasi sertifikat 27](file:///D:\TUGAS%20AKHIR\TUGAS%20AKHIR%20Revisi%20Bimbingan.docx#_Toc107140051)

[Gambar 3.17 Flowchart verifikasi sertifikat 28](file:///D:\TUGAS%20AKHIR\TUGAS%20AKHIR%20Revisi%20Bimbingan.docx#_Toc107140052)

[Gambar 3.18 Data flow diagram level 0 29](file:///D:\TUGAS%20AKHIR\TUGAS%20AKHIR%20Revisi%20Bimbingan.docx#_Toc107140053)

[Gambar 3.19 Data Flow Diagram level 1 30](file:///D:\TUGAS%20AKHIR\TUGAS%20AKHIR%20Revisi%20Bimbingan.docx#_Toc107140054)

[Gambar 3.20 Data flow diagram level 1 pengesahan issuer 30](file:///D:\TUGAS%20AKHIR\TUGAS%20AKHIR%20Revisi%20Bimbingan.docx#_Toc107140055)

[Gambar 3.21 Data Flow Diagram level 2 pembuatan sertifikat 31](file:///D:\TUGAS%20AKHIR\TUGAS%20AKHIR%20Revisi%20Bimbingan.docx#_Toc107140056)

[Gambar 3.22 Data flow diagram level 2 verifikasi sertifikat 32](file:///D:\TUGAS%20AKHIR\TUGAS%20AKHIR%20Revisi%20Bimbingan.docx#_Toc107140057)

[Gambar 4.1 Grafik performansi pembuatan sertifikat………………………………………………...37](#_Toc106562023)

[Gambar 4.2 Grafik performansi verifikasi sertifikat. 38](file:///D:\TUGAS%20AKHIR\TUGAS%20AKHIR%20Revisi%20Bimbingan.docx#_Toc106562024)

# DAFTAR TABEL

[Tabel 3.1 Tabel Kebutuhan Pengguna 11](#_Toc106041923)

[Tabel 4.1 Pengujian WhiteBox Deployment Smart Contract ……………………………………………………24](#_Toc106042391)

[Tabel 4.2 Pengujian *White Box* Pendaftaran *Issuer* 25](#_Toc106042392)

[Tabel 4.3 Pengujian Whitebox Pendaftaran Serifikat 27](#_Toc106042393)

[Tabel 4.4 Verifikasi Sertifikat 31](#_Toc106042394)

[Tabel 4.5 Pilihan jawaban kuisioner 34](#_Toc106042395)

[Tabel 4.6 User Acceptance Testing 35](#_Toc106042396)

# BAB 1 Pendahuluan

## Latar Belakang Masalah

Keberadaan virus Covid-19 yang pertama kali diumumkan pada akhir tahun 2019, dengan waktu singkat mampu menyebar keseluruh dunia dan mengakibatkan pandemi globa[1]. Akibat dari pandemi tersebut sangatlah besar salah satunya dalam bidang ekonomi[2]. Pengembangan vaksin dan tes imunitas terhadap Covid-19 menjadi prioritas masyarakat global untuk mengurangi dampak dari pandemi tersebut[3]. Dengan adanya vaksin dan tes terhadap imunitas Covid-19 resiko penyebaran dari virus Covid-19 dapat dikurangi sehingga masyarakat dapat kembali berativitas seperti normalnya[4].

Melihat pentingnya sertifikasi hasil tes maupun vaksinasi Covid-19 diperlukan solusi yang dapat menjamin keaslian dari sertifikat tersebut. Pemerintah kita sangat menyadari akan hal tersebut. Oleh karenanya pemerintah menerapkan protokol yang mengintegrasikan hasil tes Covid-19 dan vaksinasi dalam skala besar dalam sebuah sistem yang memudahkan pengguna untuk membutikan dirinya telah menerima vaksin atau melakukan tes Covid-19 [5]. Namun tentunya selalu ada kontroversi terkait keamanan dan privasi pengguna pada sistem yang tersentralisasi, salah satunya kebocoran data pribadi pengguna[6]. Oleh karenanya diperlukan sebuah sistem yang dapat menyelesaikan persoalan terkait keaslian sertifikat, pembuktian kepemilikan, dan tetap dapat menjaga privasi pengguna.

Persoalan lain dari sistem tersentralisasi Single point of failure. Sistem yang centralized akan mengalami down time dimana server tidak dapat diakses oleh client. Hal tersebut dapat terjadi karena adanya serangan dari luar ataupun error pada server penyedia layanan. Selain itu pada sistem yang tersentralisasi akan ada suatu pihak yang dapat melakukan perubahan dari data. Dikarenakan hanya terdapat satu sumber data, peembuktian atas sumber dan keaslian data akan sulit untuk dilakukan oleh pihak lain.

Sebuah sistem terdistribusi berbasis *blockchain* dapat menjadi solusi dari permasalahan tersebut, dikarenakan blockchain bersifat *imutable* sehingga mencegah adanya perubahan data atau dapat disebut menjaga data *integrity,* dan menjaga data dengan model konsesnsus yang aman dam jaringan yang bersifat *trustless*[7][8][9]. *Blockhain* seperti Ethereum juga memungkinkan adanya mekanisme yang memberikan kewenangan pada pengguna untuk mengatur identitas apa saja yang bersifat publik.

Pada penelitian ini penulis mencoba melakukan penelitian untuk membangun sebuah sistem untuk menyimpan sertifikat hasil tes dan vaksin Covid-19 pada jaringan terdistribusi *blockchain* berbasis *Ethereum*. Selain menyimpan sistem nantinya akan  ada fitur registrasi *issuer* sebagai pembuat sertifikat, membuat sertifikat, memverivikasi integritas dari sertifikat, dan menampilkan data sertifikat dan pengguna berdasar persetujuan pengguna. Sistem diharapkan dapat menyelesaikan permasalahan terkait integritas data, dan privasi serta keamanan pengguna.

## Rumusan Masalah

Sistem penyimpanan yang tersentralisasi memiliki berbagai kelebihan dan kekurangan sehingga penulis merasa perlu sistem penyimpanan yang terdistribusi atau terdesentralisasi untuk melakukan perbaikan dari kekurangan sistem yang tersentralisasi. Secara spesifik dalam penelitian tugas akhir ini penulis merumuskan masalah dan tantangan terkait penyimpanan sertifikat hasil tes dan vaksin Covid-19 yang terdesentralisasi sebagai berikut:

1. Bagaimana proses pembuatan sistem penyimpanan berbasis *blockhain* dan *smart contract* pada jaringan *Ethereum* untuk menyimpan sertifikat hasil tes atau vaksin Covid-19 secara digital?
2. Bagaimana mengimplementasikan sistem terdistribusi *blockchain* dan *smart contract* pada jaringan *Ethereum* dengan aplikasi untuk membuat sertifikat hasil tes dan vaksin Covid-19 secara digital?

## Tujuan dan Manfaat

Penulis berharap hasil akhir dari penelitian ini diharapkan tercapainya tujuan sebagai berikut ini :

1. Merancang sistem berbasis *blockchain* dan *smart contract* pada jaringan *Ethereum* untuk melakukan pendataan sertifikat hasil tes dan vaksin Covid-19 secara digital.
2. Merancang aplikasi yang dibagun diatas jaringan *blockchain* *Ethereum* untuk melakukan penyimpanan sertifikat hasil tes dan vaksin Covid-19 secara digital.

## Batasan Masalah

Pada penelitian tugas akhir ini penulisi mendefinisikan batasan masalah sebagai berikut:

1. Sistem dibangun diatas *smart contract* pada jaringan *blockchain Ethereum.*
2. Komponen aplikasi yang dibagun diatas jaringan *blockchain* yang berbasis website.
3. Penelitian berfokus pengembangan *smart contract* pada jaringan *Ethereum* dalam membuat, menyimpan dan memverifikasi sertifikat. Proses verifikasi data pribadi ke basis data pemerintah ataupun institusi formal lainnya di luar lingkup penelitian.

## Metode Penelitian

1. Studi literatur dan pengumpulan data

Merupakan tahap awal, dimana pada tahap ini penulis mencari dan mempelajari referensi yang akan digunakan. Hal tersebut meliputi jurnal, dokumen, dan materi pada website yang membahas terkait teknologi *blockchain*.

1. Analisis dan Perancangan sistem

Pada tahap ini penulis melakukan analisis terhadap kebutuhan sistem. Selanjutnya melakukan perancangan sistem dengan menggunakan *flow diagram* dan *use case diagram*.

1. Pengujian *Prototype* sistem dan analisis hasil pengujian

*Prototype* sistem akan diuji dengan beberapa metode pengujian. Selanjutnya dilakukan analisis untuk memastikan sistem telah menyelesaikan permasalahan dan mencapai tujuan yang diinginkan.

1. Penyusunan laporan Tugas Akhir

Menyusun buku tugas akhir sesuai pedoman yang diberikan.

# BAB 2 Dasar Teori

## Blockchain

Pada November tahun 2008 seseorang dengan *presudonym* Satoshi Nakamoto mengembangkan sebuah mata uang digital yang diberi nama Bitcoin tanpa ada pihak yang tersentralisasi untuk mengontrol mata uang tersebut [9]. Bersamaan dengan dirilisnya mata uang digital tersebut Satoshi Nakamoto mengeluarkan sebuah artikel yang berjudul “Bitcoin: A Peer-to-Peer Electronic Cash System”. Artikel tersebut menjelaskan hal yang diperlukan dalam pembayaran elektronik secara langsung antara dua pihak tanpa memerlukan pihak ketiga[10]. Teknologi yang mendasari sistem mata uang *peer to peer* tersebutt adalah *Blockchain*.

*Blockchain* merupakan sebuah sistem terdistribusi, dan basis data terdesentralisasi dalam sebuah jaringan *peer-to-peer* yang menggunakan mekanisme consensus agar dapat memungkinkan transaksi tanpa adanya pihak ketiga yang dipercaya[11]. Jaringan *blockchain* terdiri dari kumpulan *node* yang menyimpan rekaman data[10]. *Node* yang dimaksud disini adalah kumpulan komputer yang terhubungan dengan jaringan *peer-to-peer* dan menyimpan salinan dari data pada *blockchain*.

## Ethereum

*Ethereum* merupakan teknologi yang berdasar pada *blockchain* sama seperti *Bitcoin* pada pembahasan sebelumnya. *Blockchain* sendiri pertama kali dikenalkan oleh Vitalik Buterin pada 2013, dalam [9] *Ethereum* ditunjukan sebagai protocol alternatif untuk membangun aplikasi yang terdesentralisasi. *Ethereum* menyediakan sebuah fungsiniolitas *smart contract* untuk mendukung hal tersebut. *Ethereum* juga mendukung eksekusi skrip yang ditulis pada bahasa pemrograman berlevel tinggi dengan menyediakan sebuah *virtual machine* yang dikenal sebagai *Ethereum Virtual Machine*.

Dalam *Ethereum* dikenal *Ethereum account* dimana semua *state* pada *Ethereum* tersusun atas akun yang memiliki alamat 20-byte dan segala bentuk perubahan dari *state* pada *blockchain* merujuk pada transfer informasi dan nilai antara dua akun[12]. Akun dari Ethereum tersusun atas :

1. Nonce, penghitung yang menunjukan jumlah transaksi yang dikirim sebuah akun.
2. transaksi hanya dapat diproses satu kali.
3. Saldo ether akun saat ini.
4. Kode kontrak akun, jika ada.
5. Penyimpanan akun (kosong secara bawaan).

## Akun Ethereum

Akun pada Ethereum secara umum dibagi menjadi yaitu akun yang dimiliki secara external, dan akun kontrak yang dikontrol oleh kode kontrak. Akun eksternal tidak memiliki kode dan melakukan pengiriman informasi dengan menandatangani transaksi, oleh karenanya sebuah akun dimiliki oleh manusia. EOA terdiri atas pasangan public dan private key, memiliki saldo eter, dapat mengirim transaksi. Dalam [9], *Public key* dari dibangkitkan dari kunci privat menggunakan Elliptic Curve Digital Signature Algorithm. Anda mendapatkan alamat publik untuk akun Anda dengan mengambil 20 byte terakhir dari kunci publik Keccak-256 dan menambahkan 0x ke awal.

Sedangkan akun kontak merupakan akun yang secara otomatis akan melakukan proses pembacaan dan penulisan data yang ditulis secara internal pada kotnrak tersebut apabila menerima pesan dari akun eksternal[12]. Pembahasan terkait transaksi dan *smart contract* masing – masing dibahas pada sub bab 2.4 dan 2.5

## Transaksi pada Ethereum

A picture containing diagram

Description automatically generatedDalam [9], *Transaction* atau transaksi pada Ethereum merupakan istilah yang merujuk pada segala data yang menyimpan pesan dan telah ditandatangani oleh pemilik akun eksternal.

Gambar 2.1 Transaksi pada Ethereum merubah state pada EVM (Sumber : https://ethereum.org/id/developers/docs/transactions/)

Transaksi pada *Ethereum* akan merubah *state* dari EVM yang nantinya akan disebarkan ke seluruh jaringan *Ethereum*.Berdasar pada dokumentasi *Ethereum* [9] komponen pada sebuah transaksi *Ethereum* terdiri atas :

1. Recipient/penerima

*Recipient* merupakan Alamat penerima (jika akun milik eksternal, transaksi akan mentransfer nilai. Jika akun kontrak, transaksi akan mengeksekusi kode kontrak)

1. *Signature*

Sebuah *signature* merupakan identifikasi dari pengirim untuk membuktikan bahwa transaksi telah terautorisasi.

1. *Value*

Jumlah Ethereum yang dikirim.

1. Data

Berisi data dari transaksi.

1. gasLimit

Jumlah maksimum unit gas yang dapat dikonsumsi oleh transaksi. Satuan gas mewakili langkah-langkah komputasi.

1. maxPriorityFeePerGas

Jumlah maksimum gas yang akan dimasukkan sebagai tip untuk penambang.

1. maxFeePerGas

Jumlah maksimum gas yang bersedia dibayarkan untuk transaksi (termasuk baseFeePerGas dan maxPriorityFeePerGas).

*Gas* dalam Ethereum merujuk pada satuan unit untuk mengukur usaha komputasional untuk memproses operasi pada jaringan Ethereum[9]. Gas digunakan sebagai *fee* kepada *miner* untuk memproses transaksi tersebut dan memasukannya kedalam *block* selanjutnya.

## Smart Contract dan Solidity

Dalam [13], “Kontrak pintar adalah program yang berjalan di blockchain dan memiliki eksekusi yang benar yang ditegakkan oleh protokol konsensus”. Dalam sebuah kontrak pintar terdapat kode yang menjadi aturan yang mana akan dieksekusi oleh sebuah *event* dalam hal ini transaksi seperti pembayaran, pembacaan data, dan penulisan data.

Dalam dokumentasi *Ethereum[9],* sifat dari *smart contract* pada Ethereum adalah sebagai berikut :

1. *Permisionless*

*Permisionless* artinya setiap pengguna dapat menulsi *smart contract* mereka kedalam jaringan *Ethereum*.

1. *Composability*

*Smart contract* bersifat public pada *ehtereum blockchain* sehingga dapat diakses oleh *smart contract* sampai batasan tertentu sesuai aturan yang diterapkan pada *smart contract* yang akan diakses.

Solidity merupakan bahasa pemrograman tingkat tinggi yang diperuntukan untuk menulis kontrak yang ditargetkan pada *ethereum virtual machine*[14]. Solidity digunakan untuk menulis kode dari *smart contract* yang nantinya akan mengeksekusi transaksi pada *blockchain*.

## *Decentralized Application*

Dalam dokumentasi ethereum “Aplikasi terdesentralisasi (dapp) adalah aplikasi yang dibangun di atas jaringan terdesentralisasi yang menggabungkan kontrak pintar dan antarmuka pengguna frontend. Di Ethereum, kontrak cerdas dapat diakses dan transparan – seperti API terbuka – sehingga dapp Anda bahkan dapat menyertakan kontrak cerdas yang telah ditulis orang lain.”[9]. Secara sederhana sebuah *decentralized application* merupakan aplikasi yang mengintegrasikan *tech stack* dengan jaringan *blockchain*.

## *Brownie Python*

Brownie merupakan sebuah *framework* yang ditulis dengan bahasa *python*. Kegunaan dari *brownie* adalah untuk melakukan kompilasi dan membantu proses *deployment smart contract* yang menargetkan *Ethereum Virtual Machine*[15]*.* *Brownie* mendukung penulisan *smart contract* dalam bahasa *solidity*. *Brownie* mempermudah proses pengujian *smart contract* dan sangat membantu dalam pengembangan *smart contract.*

## *Digital Wallet dan Metamask*

*Digital wallet* merupakan aplikasi yang digunakan untuk menyimpan akun pengguna pada jaringan *blockchain*. Umumnya *digital wallet* menyimpan sebuah *private key* dari akun yang terdaftar pada jaringan *blockchain. Digital wallet* dijadikan sebagai medium antara pengguna dengan jaringan *blockchain*, *digital wallet* dapat digunakan untuk melakukan penandatangan dari sebuah transaksi yang dibuat oleh pengguna.

*Metamask* sendiri merupakan salah satu teknologi *digital wallet* yang bebentuk aplikasi yang mudah digunakan karena berupa e*xtension* pada web browser. Menjadikan *metamask* sangat cocok untuk penggembangan *DApp* berasis *web application.*

## *Web3* JS

*Web3js* merupakan sekumpulan library *javascript* yang dikembangkan untuk berinteraksi dengan *node ethereum* dengan protokol HTTP,IPC,atau web socket[16]*.* *Web3js* dapat digunakan sebaga medium yang digunakan *DApp* untuk berinteraksi dengan *smart contract*  pada jaringan *ethereum.*

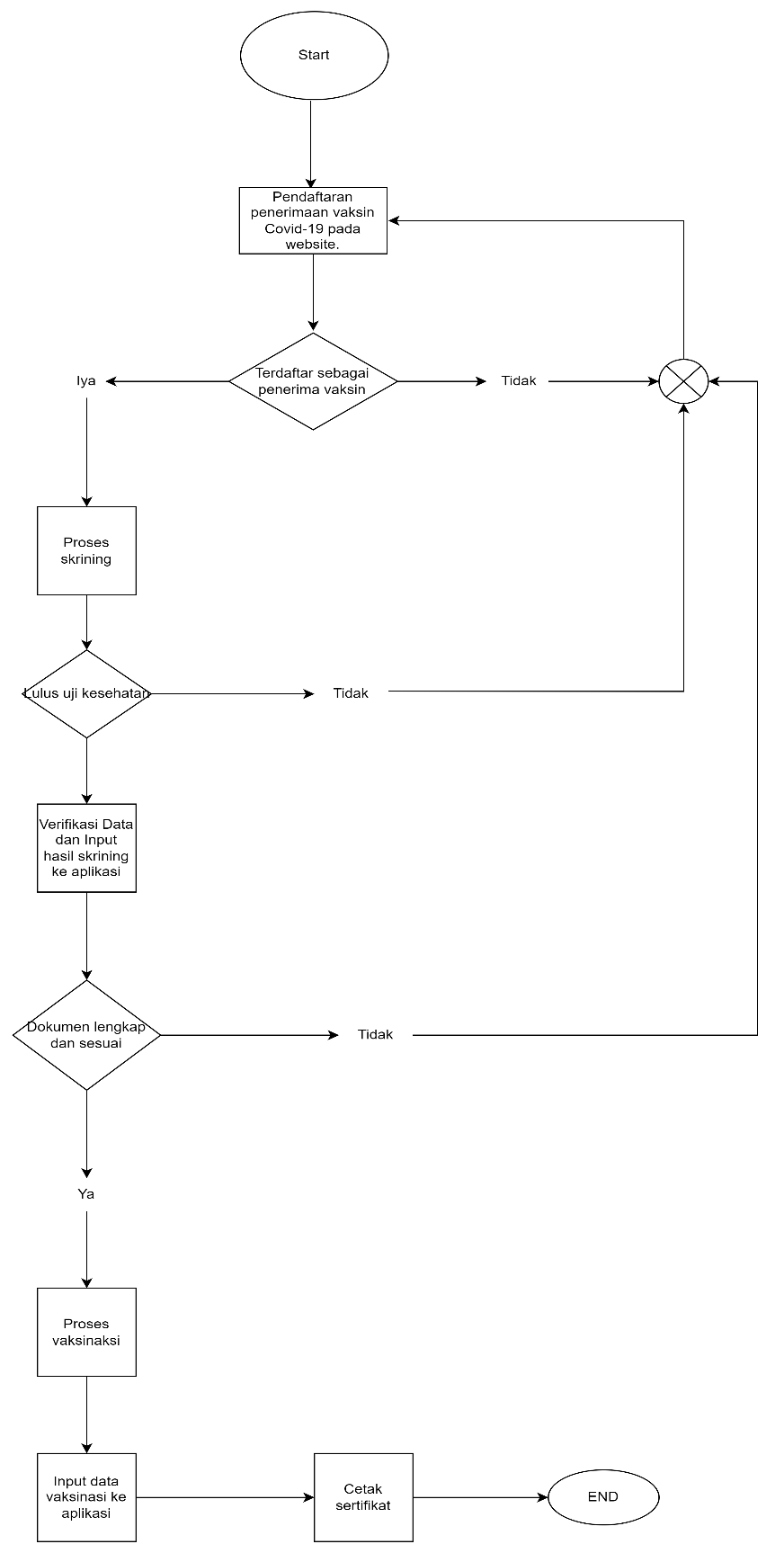
# BAB 3 Perancangan Sistem

## Analisis Kebutuhan Sistem

Pada bagian ini dilakukan analisis untuk menentukan kebutuhan sistem. Selanjutnya data akan digunakan untuk merancang aplikasi.

### **Gambaran Alur Sistem Saat ini**

Pada gambar 3.1 adalah diagram alir atau *flowchart* proses penerbitan vaksinasi. Data berikut diperoleh dari sentra vaksinasi Puskesmas Kedungmundu Kota Semarang. proses dimulai dari proses pendaftaran pada website victori.semarangkota.go.id nantinya calon penerima vaksinasi dapat memilih sentra vaksinasi yang diinginkan. Sistem selanjutnya akan mengecek apakah calon penerima vaksin telah terdaftar pada sistem sebagai penerima vaksin. Pada tahapan selanjutnya calon penerima vaksin akan mendatangani sentra vaksin yang telah dipilih sebelumnya. Selanjutnya dilakukan proses skrining untuk menguji apakah calon penerima dalam keadaan sehat untuk menerima vaksin. Hasil proses skrining akan dimasukan kedalam aplikasi bernama **P-Care** bersamaan dengan proses verifikasi data pengguna**.** Selanjutnya dilakukan pemberian vaksinasi kepada calon penerima yang telah lulus skrining kesehatan dan verifikasi data. Setelah proses vaksinasi penerima vaksin menunggu proses input data vaksinasi yang dilakukan oleh petugas. Selanjutnya penerima vaksin akan menerima sertifikat yang telah dicetak dan sertifikat dalam bentuk digital yang dapat diakses merek melalui aplikasi **PeduliLindungi**.

****

Gambar 3.1 Alir Pendaftaran Existing vaksinasi covid-19 pada sentra vaksinasi dan tes covid-19 puskesmas kedungmundu

### **Perbandingan Sistem Tersentralisasi Dengan Sistem Terdesentralisasi Blockchain**

Sistem yang telah tersedia sekarang merupakan sistem yang tersentralisasi dan memiliki beberapa kekurangan antara lain:

* *Single point of failure*. Sistem yang *centralized* akan mengalami *down time* dimana server tidak dapat diakses oleh *client*. Hal tersebut dapat terjadi karena adanya serangan dari luar ataupun *error* pada server penyedia layanan.
* *Asset provenance*. Pada sistem yang tersentralisasi suatu pihak dapat melakukan perubahan dari data yang disimpan. Dikarenakan hanya ada satu sumber data sangat sulit untuk mendeteksi inkonsistensi data dan pembuktian sumber data ke pihak lain.

Pada sistem terdesentralisasi *blockchain* dikarenakan terdapat lebih dari satu *node* yang bertugas menyediakan data maka permasalahan terkait *single point failure* dapat teratasi. Pada *blockchain* setiap peserta menyimpan salinan dari data dan perubahan yang dilakukan. Melalui metode konsensus *blockchain* dapat langsung mengidentifikasi apabila ada inkonsistensi pada data dan setiap *user* dapat membuktikan asal data tersebut.

### **Kebutuhan Pengguna**

Selanjutnya dilakukan juga wawancara untuk kebutuhan pengguna dalam hal ini adalah tenaga kesehatan yang melakukan proses vaksinasi kepada masyarakat. Data kebutuhan pengguna ini dapat membantu dalam menyusun kebutuhan fungsional dari sistem. Kebutuhan pengguna dapat dilihat pada tabel berikut :

|  |  |
| --- | --- |
| No | Kebutuhan |
| 1 | Terdapat pengawas yang memberikan akses kepada petugas untuk melakukan pembuatan serifikat. |
| 2 | Sistem menyimpan data petugas yang telah diberikan akses. |
| 3 | Petugas memiliki akses terhadap riwayat vaksinasi/tes calon penerima vaksin/tes Covid-19. |
| 4 | Sistem mencatat data nama pasien penerima vaksin atau penerima hasil uji tes covid-19. |
| 5 | Sistem mencatat data nomor kependudukan pasien penerima vaksin atau hasil uji tes covid-19. |
| 6 | Sistem mencatat data umur pasien penerima vaksin atau hasil uji tes covid-19 |
| 7 | Sistem mencatat data gender pasien penerima vaksin atau penerima hasil uji tes covid-19. |
| 8 | Untuk data vaksin yang diterima pasien meliputi jenis vaksin, dosis, dan waktu penerimaan vaksin. |
| 9 | Untuk data hasil tes Covid-19 data yang perlu disimpan meliputi jenis tes, hasil, dan masa berlaku tes tersebut. |
| 10 | Sertifikat yang telah dibuat dapat dikirimkan ke whatsapp agar lebih mudah bagi pasien. |
| 11 | Sertifikat yang telah dibuat harus dapat diverifikasi melalui sistem. |
| 12 | Sistem menampilkan data dari sertifikat yang telah diverifikasi data dari sertifikat tersebut. |

Tabel 3.1 Tabel Kebutuhan Pengguna

### **Peran Pengguna Pada Sistem Yang Diusulkan**

Pada bagian ini penulis akan menjabarkan klasifikasi pengguna dari aplikasi yang dirancang oleh penulis. Pengguna dari aplikasi terbagi menjadi empat aktor, yaitu *Issuer, patient, checker,* dan *regulator*. Masing – masing pernanan mereka dapat dijabarkan sebagai berikut :

1. *Issuer*

*Issuer* merupakan aktor yang berperan untuk menerbitkan sertifikat tes/vaksin Covid-19.

1. *Patient*

*Patient* merupakan pemilk dari sertifikat.

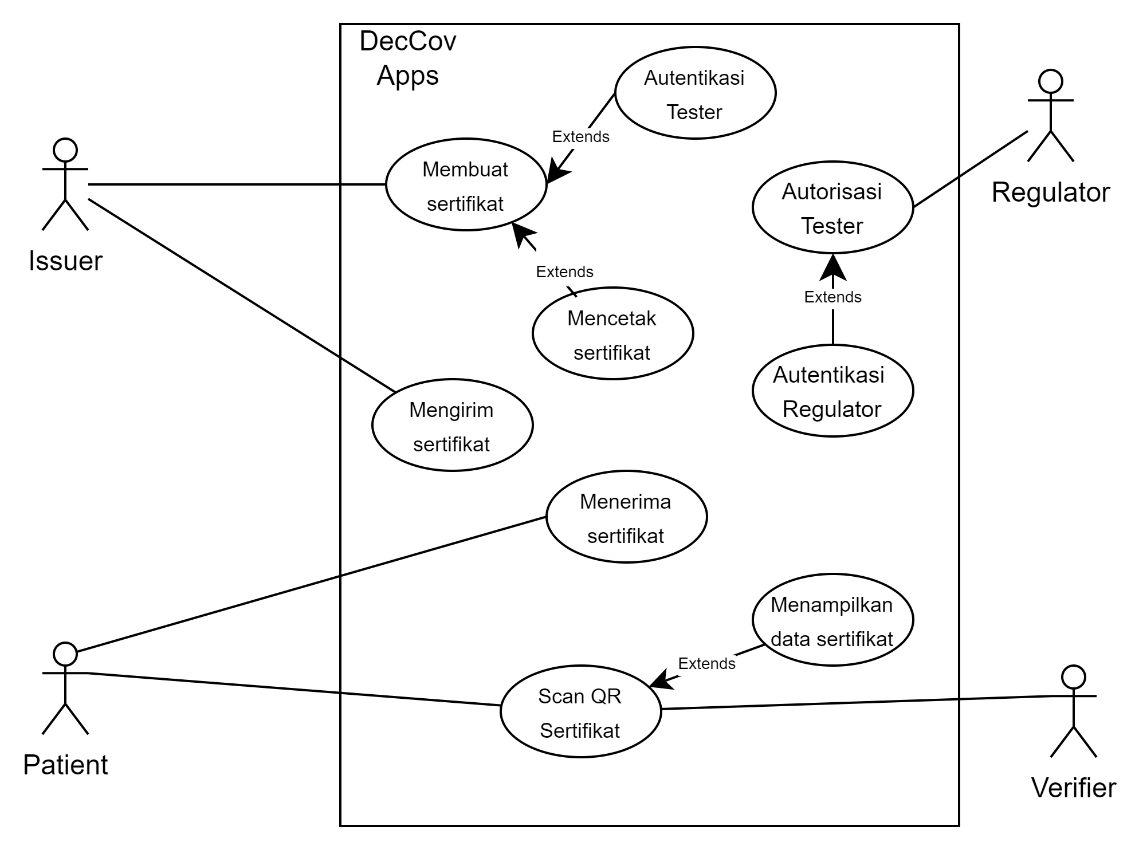
1. *Verifier*

*Verifier* merupakan pihak yang mengakses aplikasi untuk melakukan pengecekan apakah sebuah sertifkat valid.

1. *Regulator*

*Regulator* berperan sebagai pemilik dari *Smart Contract* dan sistem. *Regulator* berperan dalam mengesahkan *issuer*.

Gambar berikut merupakan *use* case diagram yang menggambarkan peran dari masing – masing aktor



Gambar 3.2 Use Case Diagram

### **Analisis Kebutuhan Fungsional Dari Sistem Yang Diusulkan**

Pada dasarnya sistem yang akan dirancang harus bisa menyelesaikan kebutuhan yang diingiinkan oleh pengguna. Penulis menjabarkan kebutuhan fungsional dari aplikasi sebagai berikut :

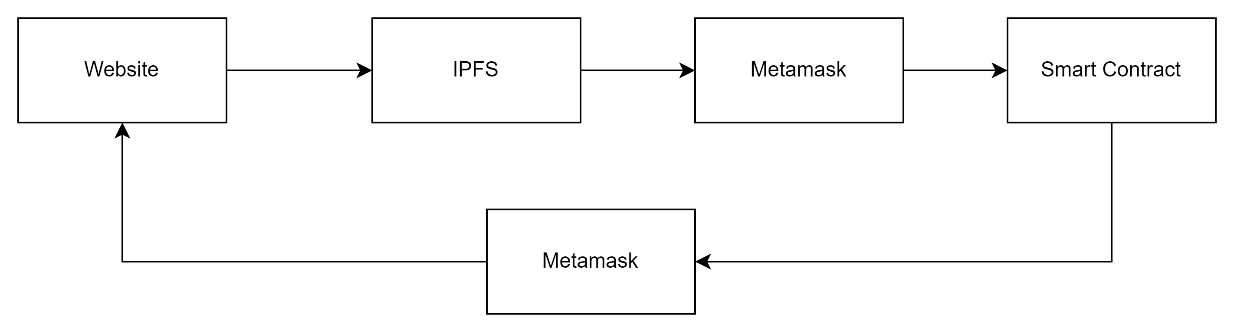
1. Sistem memiliki tiga portal berbeda yang masing – masing akan digunakan oleh aktor *verifier*, *issuer, regulator*.
2. *Regulator* dapat menyetujui pengguna lain sebagai *issuer*.
3. Sistem dapat melakukan pengecekan riwayat vaksinasi dan hasil tes calon penerima vaksin.
4. Sistem memberikan akses kepada *issuer* yang telah tervalidasi oleh *regulator* untuk mendaftarkan sertifikat vaksin atau hasil tes Covid-19 kedalam *blockchain*.
5. Sertifikat vaksin atau hasil tes Covid-19 yang berbentuk *QR Code* dapat diberikan kepada *Patient* (penerima vaksin) *Whatsapp*.
6. Sertifikat juga menyimpan data foto milik pengguna yang nantinya akan digunakan untuk membuktikan kepemilikan sertifikat tersebut.
7. Data sertifikat yang disimpan pada *database* dienkripsi dengan pin yang dibuat oleh pengguna dengan tujuan untuk melindungi data pengguna.
8. *Verifier* dapat melakukan verifikasi sertifikat vaksin dengan melakukan scan pada *QR code* milik *patient* dengan persetujuan *patient* sebagai pemilik dari sertifikat*.*
9. Pasien dapat mengakses data sertifikat menggunakan *QR Code*.

## Desain Sistem

Pada bagian ini penulis akan menjelaskan secara umum rancangan dari sistem yang diusulkan. Perancangan ini didasarkan analisis kebutuhan yang telah dijelaskan pada bagian sebelumnya. Sistem nantinya akan dibagun diatas jaringan *blockchain Ethereum* yang mana mendukung pengembangan *smart* *contract* yang dapat dijalankan pada EVM (*Ethereum Virtual Machine)*. Penggunaan *smart contract* akan menjadi dasar pembangunan *Decentralized Application* untuk penyimpanan sertifikat vaksin / hasil tes Covid-19. Sistem yang diusulkan juga akan memanfaatkan penyimpanan data tedesentralisasi yaitu *IPFS.*

### **Diagram Blok**

Implementasi penyimpanan sertifikat tes/vaksin Covid-19 pada jaringan *blockchain* akan menggunakan sebuah *Decentralize Application*. Secara umum sistem dibagi menjadi tiga fungsi utama yaitu :

1. Proses registrasi *Issuer*
2. Proses pembuatan dan penyimpanan sertifikat hasil tes atau vaksin Covid-19
3. Proses verifikasi sertifikat vaksin.

Gambar 3.3 Diagram blok dari sistem yang disusulkan

Proses akan dimulai dari website sebagai *Decentralize Application* lalu dilakukan proses pada IPFS sesuai dengan fungsionalitas yang dilakukan. Apabila fungsi yang dipanggil merupakan proses registrasi *Issuer* maka akan dilakukan penyimpanan data *Issuer.* Sedangkan untuk proses pembuatan dan penyimpanan dilakukan penyimpanan data sertifikat pada IPFS. Proses verifikasi sendiri nantinya akan membaca sertifikat dari IPFS.

Selanjutnya penggunaan *Metamask* untuk menyimpan private key dari pengguna. *Metamask* merupakan aplikasi yang berjalan pada client sehingga dipilih agar aman. *Metamask* nantinya akan mengirim transasksi ke *Blockchain* untuk proses pembuatan sertifikat tes/vaksin Covid-19.

### **Perancangan *Smart Contract***

Penggunaan *Ethereum* mendukung pemabngunan *decentralized application* dengan memanfaatkan *Smart Contract.* Pada penelitian ini dibangun dua *Smart Contract* yang akan di-*deploy* sebelum aplikasi akan diluncurkan. Nantinya seluruh proses yang dilakukan pada sistem akan mengacu hanya kepada dua *Smart Contract* tersebut. *Smart Contrat* digunakan pada sistem terdiri atas :

1. *Certificate Registry Smart Contract*

*Certificate Registry Contract* merupakan *smart contract* yang digunakan untuk meyimpan hash dari sertifikat yang diterbitkan oleh *issuer*.Fungsi yang terdapat pada *certificate registry smart contract* adalah sebagai berikut :

1. registerCertificate

Fungsi ini digunakan oleh *issuer* untuk mendaftarkan sertifikat ke *smart contract.* Fungsi ini melakukan proses *write* pada *smart contract* dan hanya bisa diakses oleh *issuer* yang telah terdaftar sebelumnya.

1. verifyCertificate

Fungsi ini digunakan untuk melakukan verifikasi sertifikat. Fungsi ini dapat dibaca oleh semua pengguna.

1. getCertificatesByUser

Fungsi ini digunakan oleh *issuer* untuk mendapatkan riwayat vaksinasi milik pasien. Fungsi ini melakukan pembacaan data pada *smart contract* dan hanya dapat dilakukan oleh *issuer* yang sebelumnya telah mendaftarkan akunnya ke *regulator.*

1. grantAccess

Merupakan sebuah *function modifier* yang membatasi akses pada fungsi yang hanya diperuntukan untuk *issuer.*

1. *Issuer Registry Smart Contract*

*Registry Smart Contract* digunakan untuk menyimpan address dari *issuer* yang telah didaftarkan oleh *regulator*. Dengan begitu *Registry Smart Contract* berperan dalam memberikan autorisasi pendaftaran sertifikat pada *Certificate Smart Contract.* Fungsi pada *issuer Registry Smart Contract* adalah sebagai berikut :

1. registerIssuer

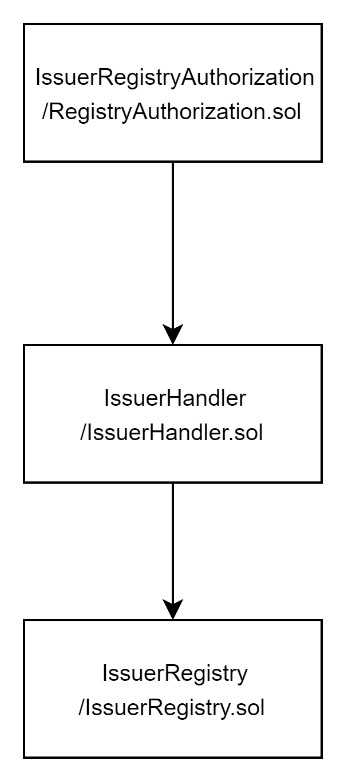
Fungsi ini digunakan oleh *regulator* untuk mendaftarkan alamat *ethereum* milik *issuer.* Fungsi ini melakukan *write* pada *smart contract* dan hanya dapat diakes oleh *owner* dari *smart contract* yang mana adalah aktor *regulator* itu sendiri.

1. checkIssuerExist

Fungsi ini digunakan untuk melakukan pengecekan apakah alamat *ethereum* terdaftar sebagai *issuer* pada *issuer registry smart contract.*

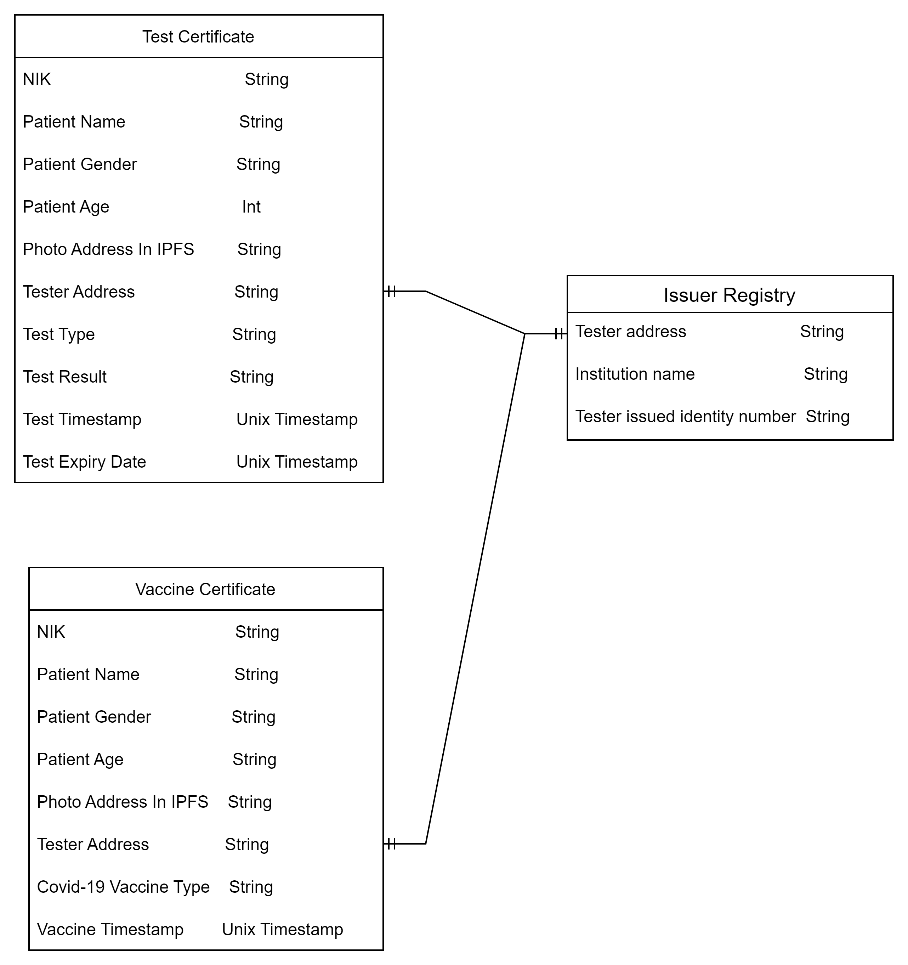
### **Struktur *Smart Contract***

Perancangan *smart contract* dipecah kedalam beberapa file solidity dengan konsep *linear contract inheritance.* Setiap file akan menurunkan sifatnya sampai ke kontrak utama. Berikut adalah diagram bloknya.

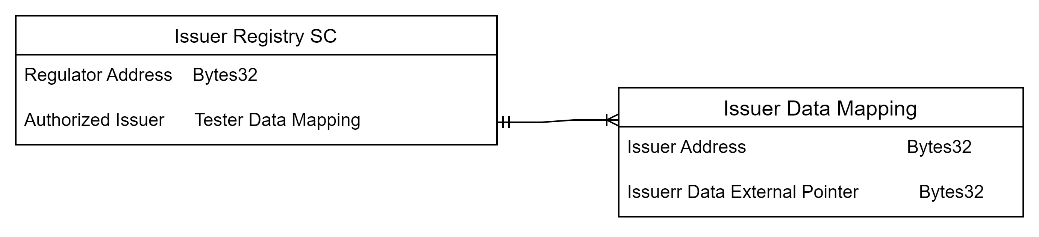


Gambar 3.4 Diagram block inheritance dari smart contract yang dirancang

### ***Entity Relational Diagram* Pada *Smart Contract***

****Untuk memenuhi kebutuhan struktur data untuk sertifikat vaksin dan hasil tes Covid-19 terlampir pada gambar berikut :

Gambar 3.5 Entity Relational Diagram

Lalu penyimpanan data pada *Issuer Registry Smart Contract* terdiri atas *regulator address* yang menyimpan alamat milik *regulator* dan *authorized user* yang merupakan sebuah array yang menyimpan sekumpul *stuct* data *issuer\_data. Stuct issuer\_data* menyimpan alamat *ethereum account* milik *issuer* dan *pointer data external* yang merujuk pada alamat penyimpanan data *issuer* pada IPFS. Semua data pada *smart contract* bersifat *private,* sehingga tidak dapat diakses dari luar *smart contract.*

Gambar 3.6 ER Diagram Issuer Registry Smart Contract

Pada certificate registry smart contract strutktur penyimpanan sertifikat milik pasien dapat dilihat pada ER diagram berikut.



Gambar 3.7 ER diagram data pada certificate registry smart contract

Pada gambar 3.7, data yang disimpan pada *certificate registry smart contract* meliputi *owner* yang menyimpan alamat *ethereum account* milik *regulator*, *issuerRegistrySC* menyimpan alamat dari *IssuerRegistrySC*, *dan holder\_to\_certificate* yang merupakan sebuah *array* dari tipe data *map*. Tipe data *map* pada *solidity* merupakan pasangan *key-value*, pada *Certificate Registry Smart Contract* yang menjadi *key* adalah *hash* dari nomor NIK milik pasien dengan *value-*nya sebuah *array* dari *structure* bernama *cov\_certificate.*

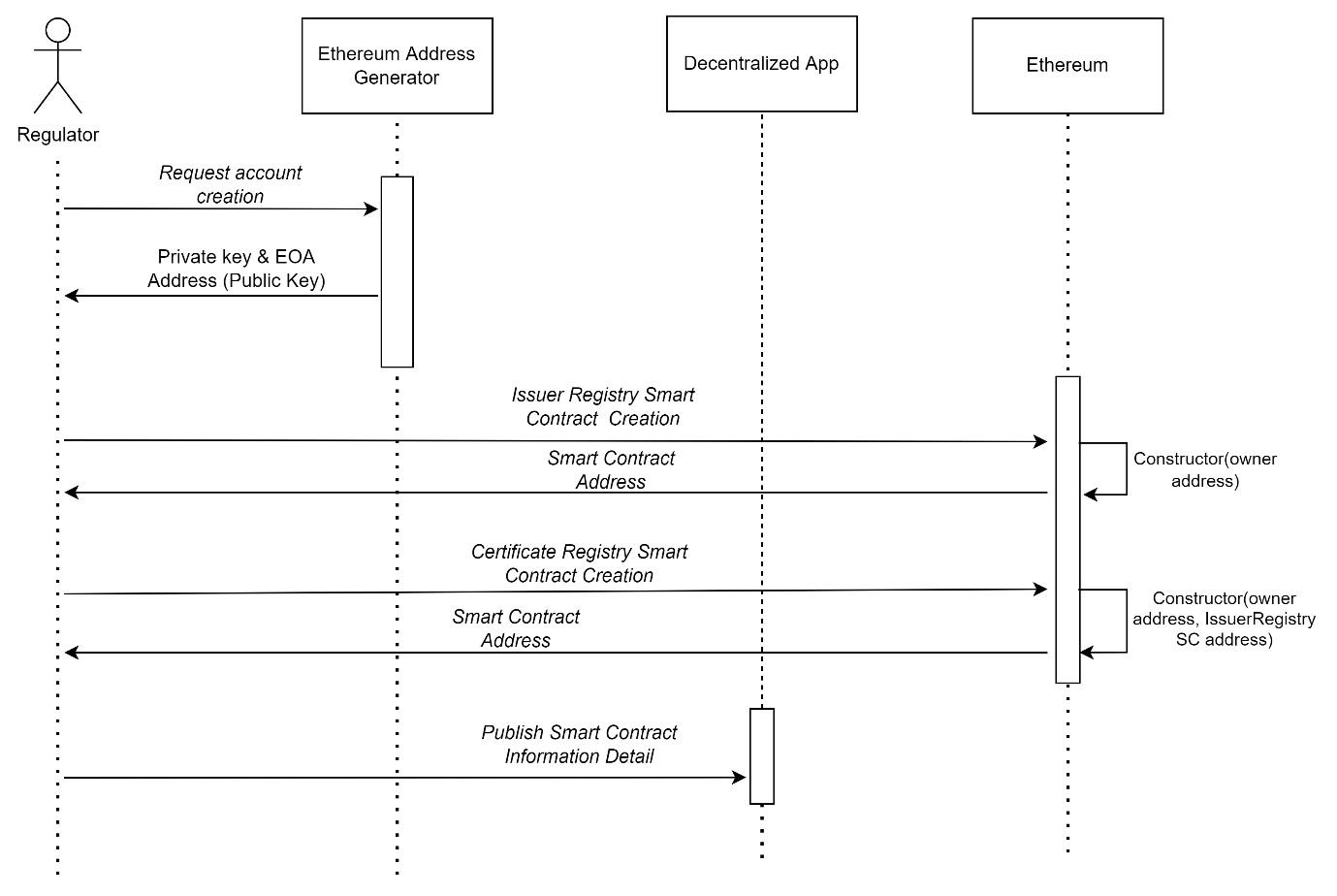
Data *Struct cov\_certificate* sendiri menyimpan data sertifikat milik pasien penjelasannya sebagai berikut.

1. *cov\_certificate\_identifier* merupakan *pointer data external* yang merujuk pada alamat data pada IPFS. Variabel ini digunakan sebagai *value* unik dari data sertifikat untuk menghindari adanya duplikat data milik pasien. Selain itu digunakan oleh *issuer* untuk melakukan pemeriksaan riwayat vaksin/tes milik pasien.
2. *certificate\_data* merupakan *pointer* data external yang merujuk pada alamt data pada IPFS. Data yang disimpan merupakan data sertifikat pasein yang telah dienkripsi.
3. *Issuer\_address* menyimpan alamat penerbit *ethereum account* penerbit sertifikat.
4. Timestamp menyimpan waktu penerbitan sertifikat dalam format ISO.

### ***Authentication* dan *Authorization Issuer***

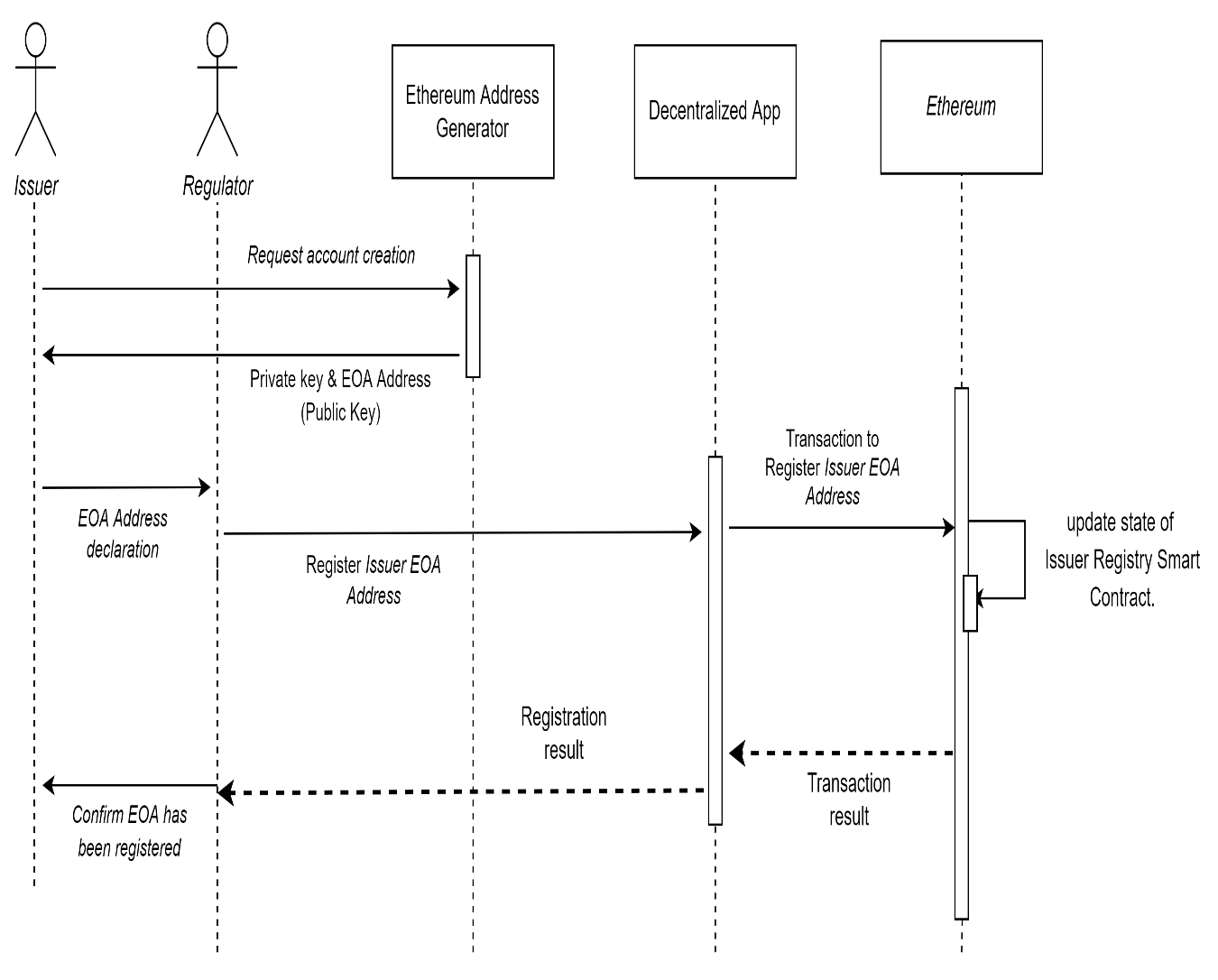
*Ethereum* merupakan *public permissionless blockchain* dimana setiap orang dapat melakukan transaksi selama memiliki akun *Ethereum*. Pada penilitian ini, sistem yang dirancang memerlukan mekanisme untuk melakukan *autorrization* dengan tujuan untuk membatasi akses *write* ke dalam *smart contract*. Selanjutnya proses *authentication* dilakukan untuk membutikan bahwa *user* memiliki akses untuk melakukan *write* kedalam *smart contract*.

Sebelumnya pada sub bab 3.1.4 telah didefinisikan aktor yang terdapat pada sistem yang diusulkan beserta *role-*nya masing - masing. Dari empat aktor yang telah didefinisikan sebelumnya aktor *patient* dan *verifier* hanya akan diperbolehkan untuk membaca data pada *smart contract* sedangkan proses *write* akan dilakukan oleh *regulator* dan *issuer*. Proses ini nantinya akan melibatkan verifikasi *signature* dari sebuah *message* yang ditandatangani oleh pengguna.

*Regulator* akan berperan dalam melakukan *deployment* dari *smart contract*. Untuk melakukan hal tesebut *regulator* perlu membuat pasangan *Private Key* dan *Public Key* (*EOA)*. Diagram pada gambar 3.7 berikut menunjukan proses yang dilakukan oleh aktor *regulator* dalam melakukan *deployment smart contract*.

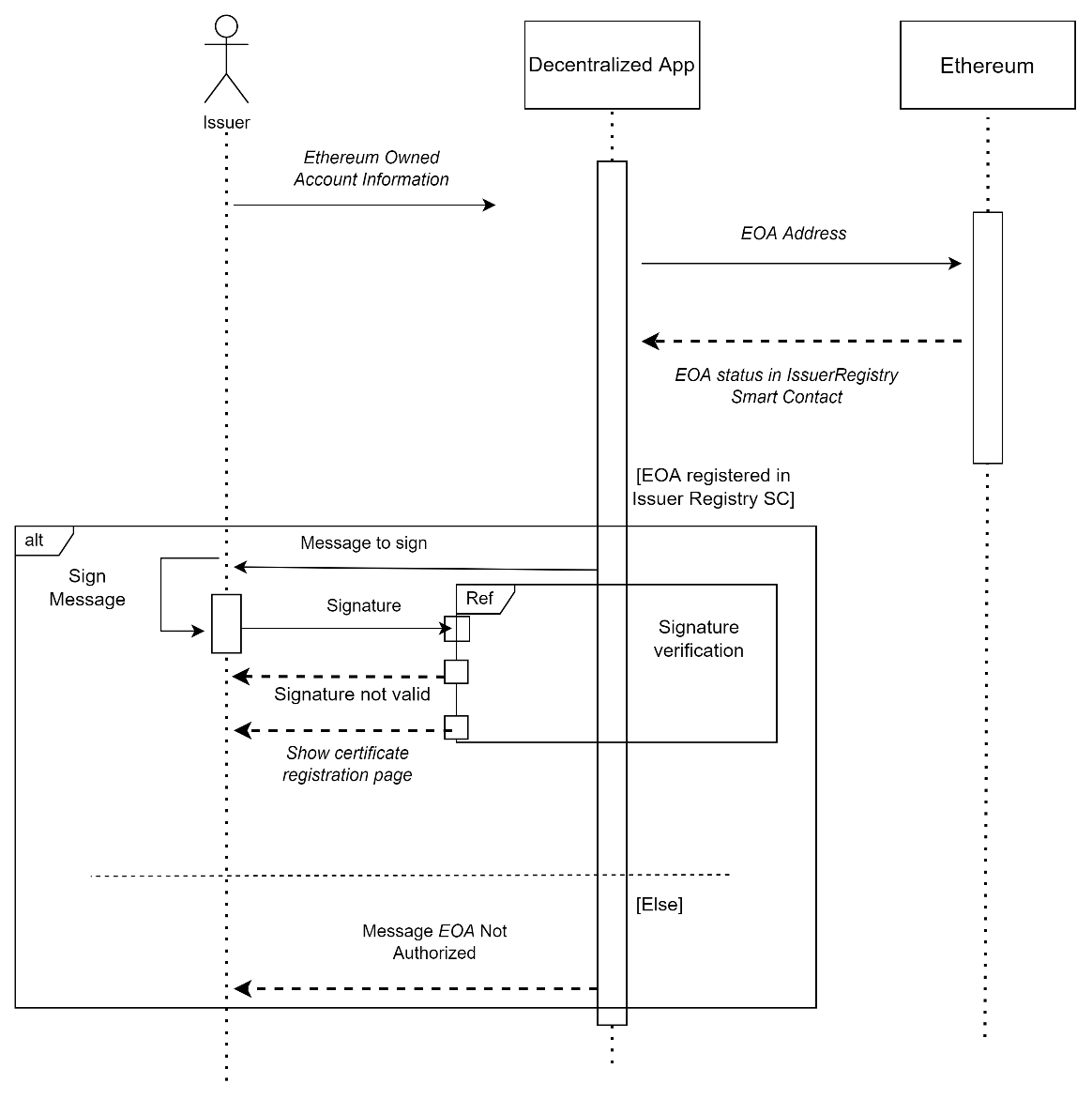
Gambar 3.8 Contract deployment oleh Regulator

Sama halnya dengan *regulator, issuer* perlu membuat pasangan *Private Key* dan *Public Key* (*EOA*). Selanjutnya *issuer* menyerahkan *Ethereum Owned Account address* miliknya ke *Regulator* untuk didaftarkan ke dalam *IssuerRegisitry Smart Contract* sebagai bagian dari autorisasi untuk mengakses *CertificateRegistry Smart Contract.* Melalui fungsi *registerIssuer* *regulator* dapat mendaftarkan *EOA* milik *issuer* kedalam *IssuerRegistry Smart Contract*. Proses tersebut dapat dilihat pada gambar 3.8 berikut.



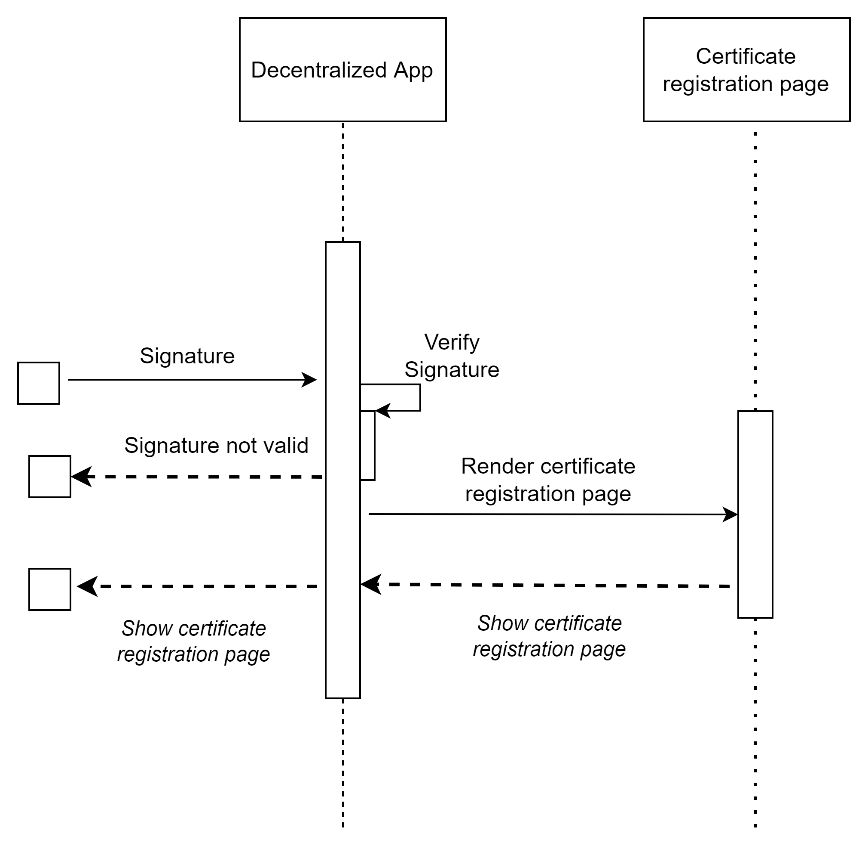
Gambar 3.9 Sequence Diagram pendaftaran issuer oleh regulator

Melalui *modifier grantAccess* membatasi transaksi pada *IssuerRegistry smart contract* hanya akan dapat dilakukan oleh *regulator*. Hal tersebut dapat dilakukan karena *Ethereum* dapat melakukan komputasi untuk mendapatkan informasi *EOA* dari sebuah transaksi atau *signer* dari transaksi. Sama halnya dengan transaksi pada *CertificateRegistry smart contract* setiap transaksi akan dilakukan pengecekan alamat publik dari pembuat transaksi apakah alamat tersebut terdaftar pada *IssuerRegistry smart contract.* Sehingga fungsi pembuatan sertifikat hanya dapat dilakukan oleh *issuer.* Namun pihak lainnya masih dapat melakukan pembacaan data pada *smart contract*.

Pada gambar 3.10 menunjukan proses autentikasi *issuer* saat mengakses *decentralized application.* Aplikasi terlebih dahulu akan melakukan pengecekan pada alamat *ethereum account* yang diakui oleh *issuer.* Selanjutnya apabila alamat akun terdaftar maka dilakukan metode *chalenge-response authentication* untuk membuktikan kepemilikan *user* terhadap akun tersebut*.*

Gambar 3.10 Prosesn autentikasi issuer

Proses tersebut melibatkan proses pembuktian bahwa user memiliki *private key* dari *Ethereum Owned Account* (*EOA*). Dengan menggunakan bantuan *module web3js* dapat dilakukan pembuatan *signature* dengan cara berikut *Signature* = *sign(User.privateKey, User.address, web3.sha3(message)).* Dengan *module web3JS* juga dapat dilakukan perhitungan ulang terkait informasi *EOA* dari sebuah *signature,.*  Sehingga *ownership* dari sebuah akun dapat dibuktikan oleh pihak yang mengaku memiliki akun tersebut. Gambar 3.11 menunjukan proses verification dari *siganture.*



Gambar 3.11 Verifikasi signature

### **Proses Pendaftaran Issuer Oleh Regulator**

Proses registrasi *Issuer* dilakukan oleh *regulator*. Proses dimulai dari *issuer* membuat pasangan *private/public key*. Proses tersebut dapat dilakukan dengan bantuan *metamask,* sebagai *wallet ethereum*.Selanjutnya *issuer* menyerahkan alamat publik dari *ethereum owned account* miliknya ke *regulator*. Regulator selanjutnya akan mendaftarkan *ethereum owned account* milik *issuer* pada *issuer registry smart contract*. Data *issuer* akan disimpan pada *IPFS,* nantinya alamat data pada *ipfs.*

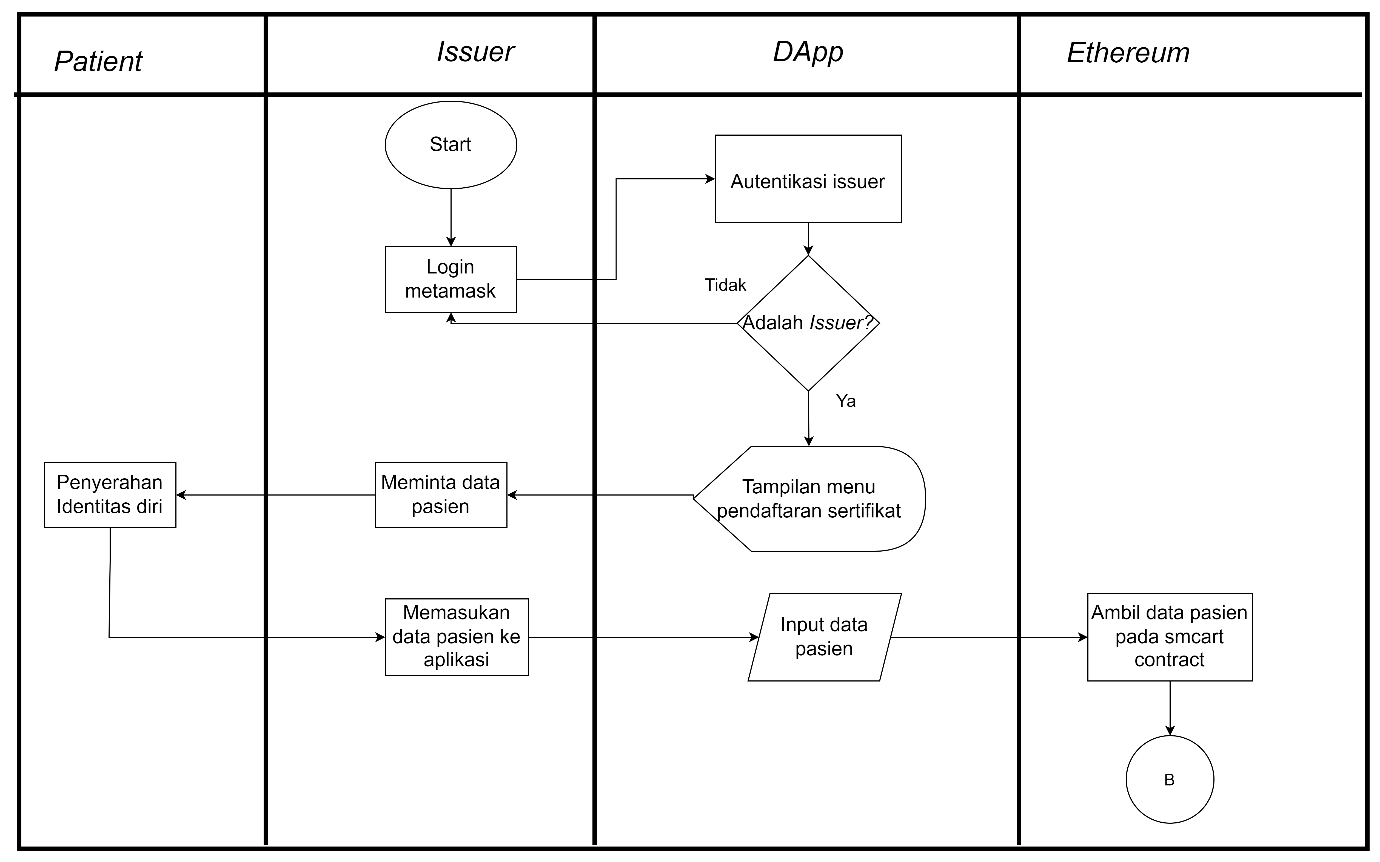
*****Flowchart* pada gambar 3.10 menggambarkan proses yang dilakukan calon *issuer* dalam membuat pasangan *public*/*private key* miliknya. Selanjutnya *issuer* menyerahkan alamat *EOA* miliknya ke *regulator*. Gambar 3.11 merupakan *flowchart* proses pendaftaran *issuer* oleh regulator.

Gambar 3.12 Proses pembuatan public private key oleh issuer

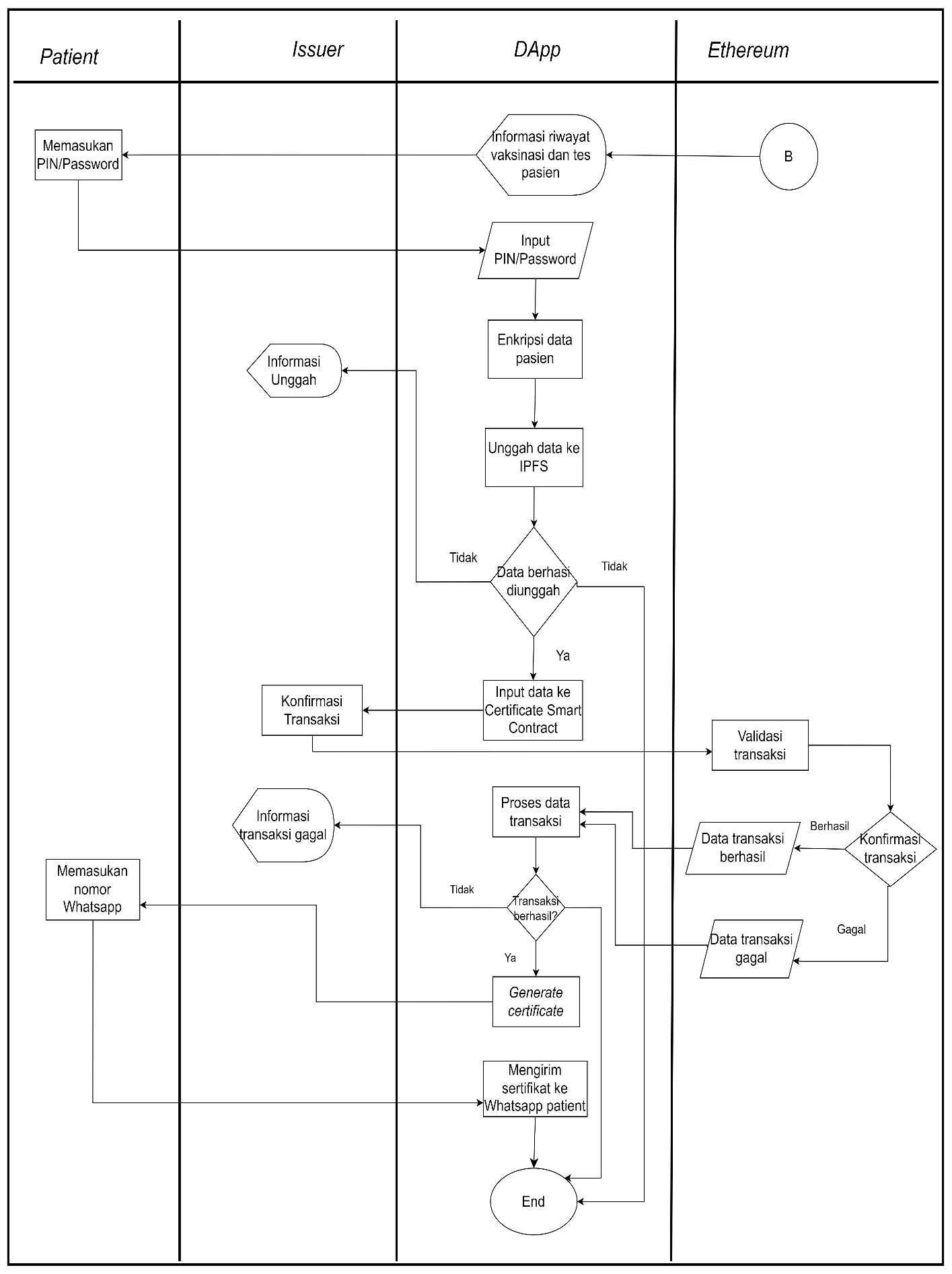
### **Proses Pendaftaran Sertifikat**

Gambar 3.13 Flowchart registrasi issuer oleh regulator

Proses ini dimulai dengan *patient* memberikan identitasnya kepada *issuer* dengan menunjukan kartu identitas dapat berupa KTP, SIM, dan karu identitas lain yang diterbitkan oleh insitusi pemerintah. Selanjutnya proses tes dan vaksin dapat dilakukan dan *Issuer* akan memasukan data *patient* yang telah dienkripsi kedalam IPFS. Apbila sukses sertifikat akan didaftarkan ke *Certificate Smart Contract.*

Nantinya *patient* akan menerima data hasil tes maupun vaksinasi covid-19 dalam bentuk sertifikat. Sertifikat yang diberikan ke *patient* akan berupa *QR Code*. Pada *QR Code* juga telah tertanam foto dari *patient* dengan begitu *patient* dapat membuktikan kepemilikandari sertifikat tersebut. *Flowchart* dari proses pendaftaran *issuer* dapat dilihat pada gambar berikut.

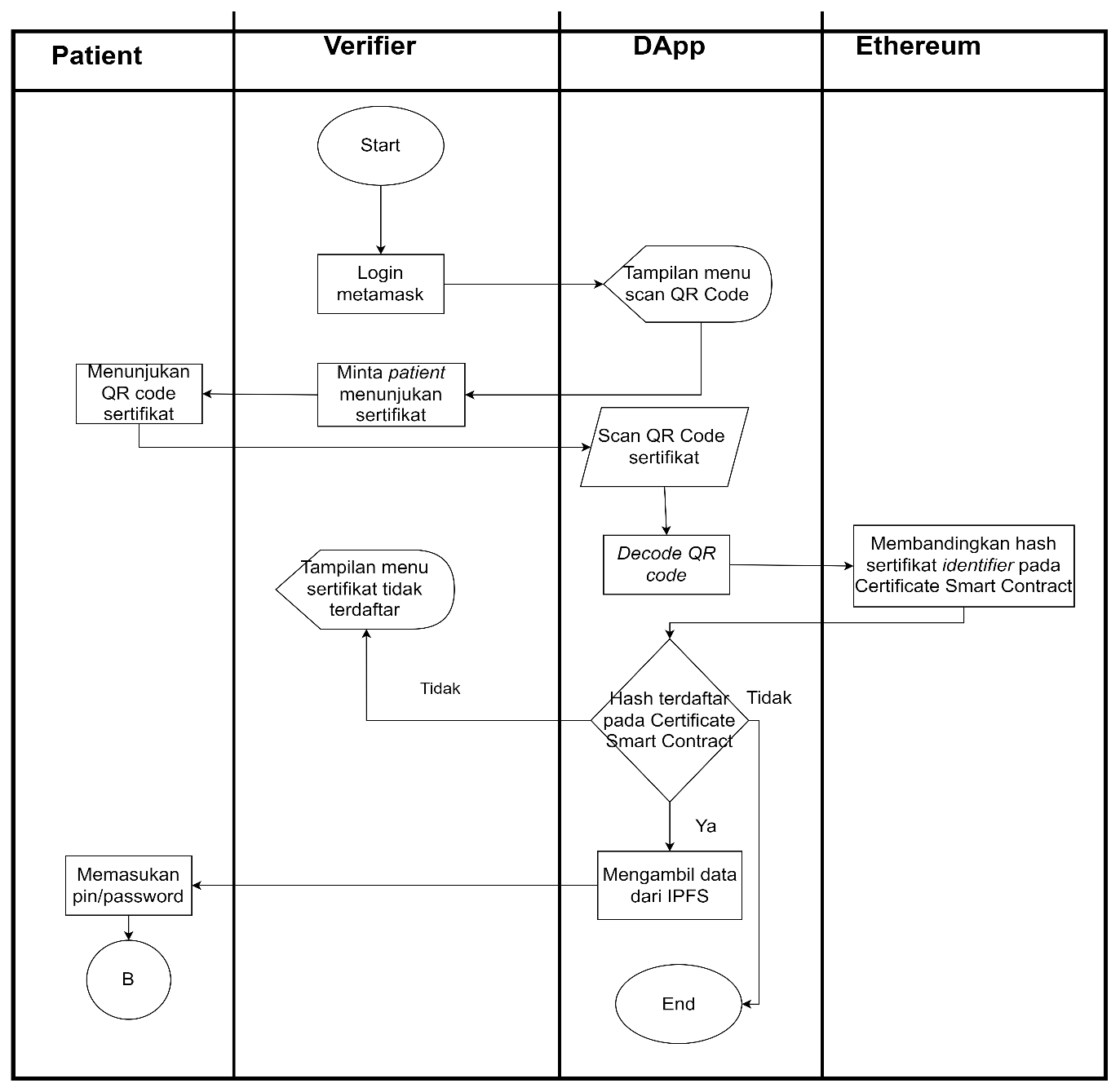
Gambar 3.14 Flowchart pendaftaran sertifikat Covid-19



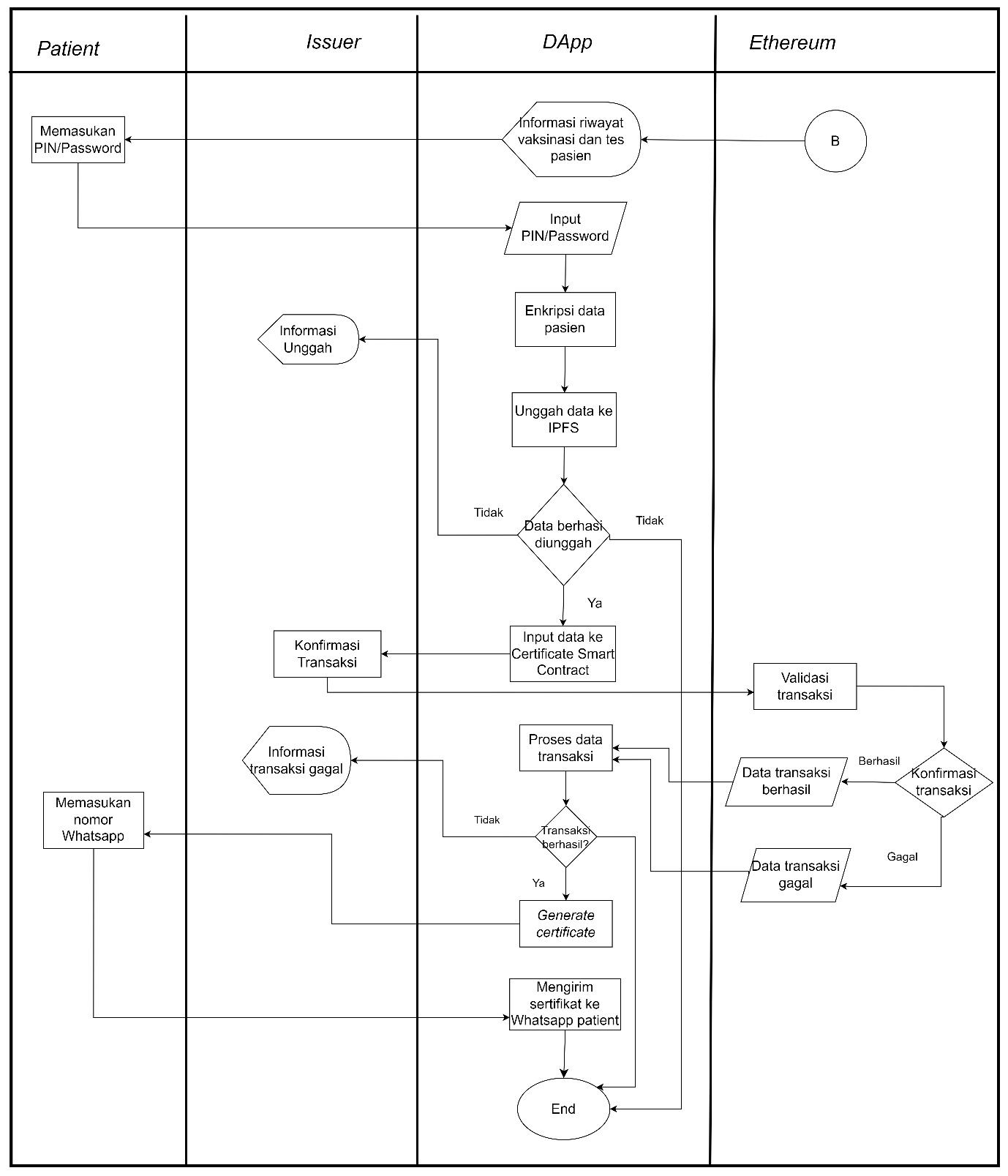
Gambar 3.15 Flowchart pendaftaran sertifikat Covid-19

### **Proses Verifikasi Sertifikat**

Dalam proses verifikasi *patient* hanya perlu menunjukan *QR code* yang telah dimiliki. Lalu melalui aplikasi *verifier*, *patient* akan diminta memasukan pin yang digunakan untuk mendekripsi data sertifikat. Selanjutnya aplikasi akan membandingkan *hash* dari data sertifikat *patient* ke *certificate smart contract*. Apabila data terdaftar maka sertifikat dinyatakan *valid*. Sebagai bukti kepemilikan aplikasi nantinya akan menampilkan foto *patient* yang telah tertanam pada sertifikat tersebut.

****

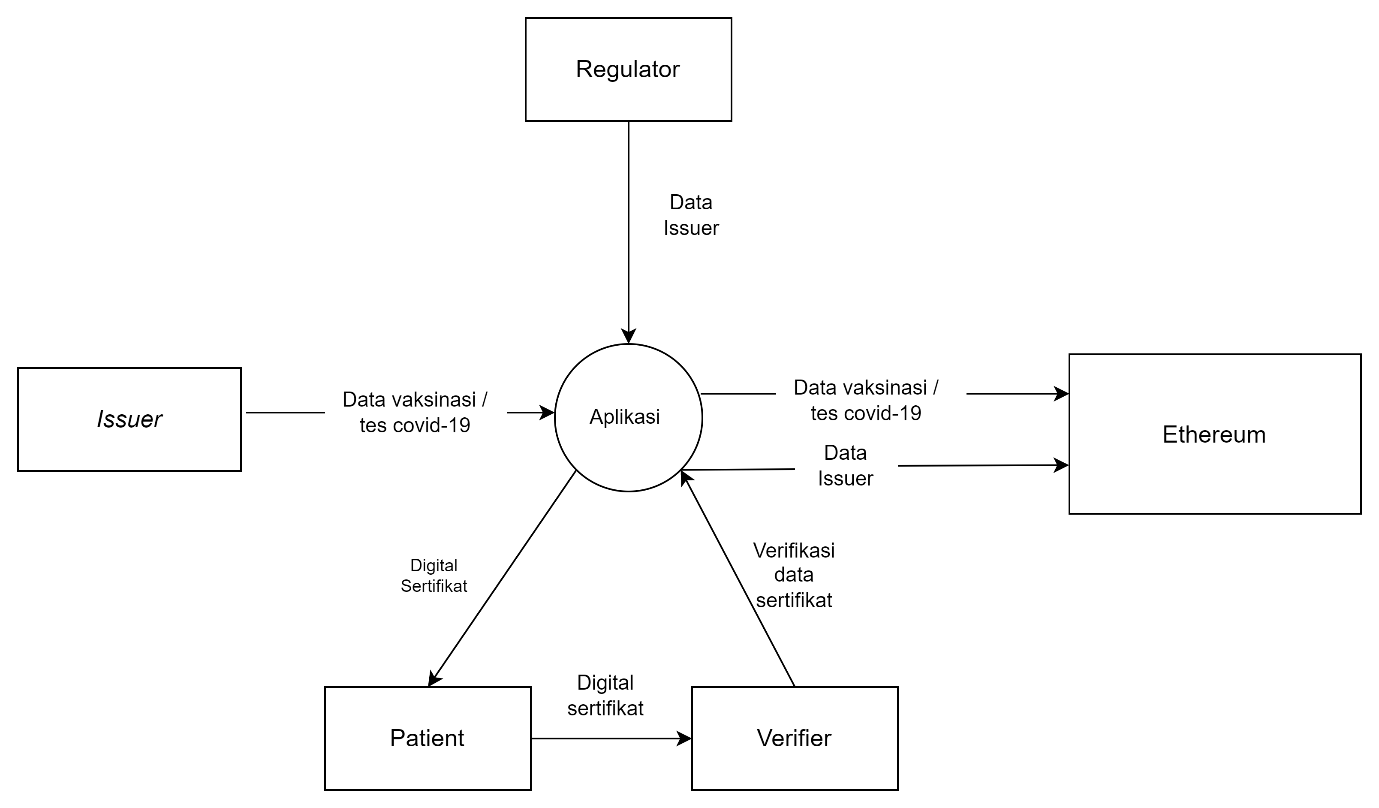
Gambar 3.16 Flowchart verifikasi sertifikat

****

Gambar 3.17 Flowchart verifikasi sertifikat

### ***Data Flow* Diagram Level 0**

Secara umum proses aliran data pada sistem dapat dilihat pada gambar berikut:

Pada gambar 3.18 adalah gambaran umum aliran data untuk proses aliran data pada sistem yang dirancang. Termasuk didalamnya proses input data vaksinasi/tes Covid-19, input data *issuer* oleh aktor *regulator,* penerimaan sertifikat vaksin digital oleh *patient*, verifikasi data sertifikat dan penyimpanan data pada *blockchain*.

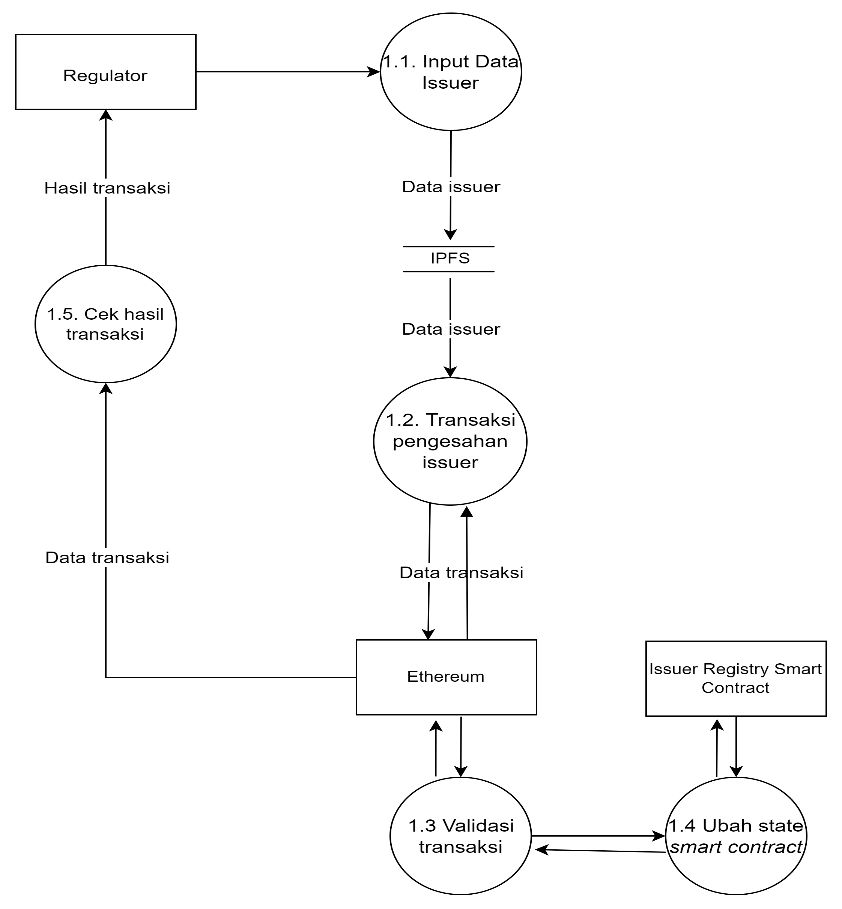
Gambar 3.18 Data flow diagram level 0

### ***Data* Flow Diagram Level 1**

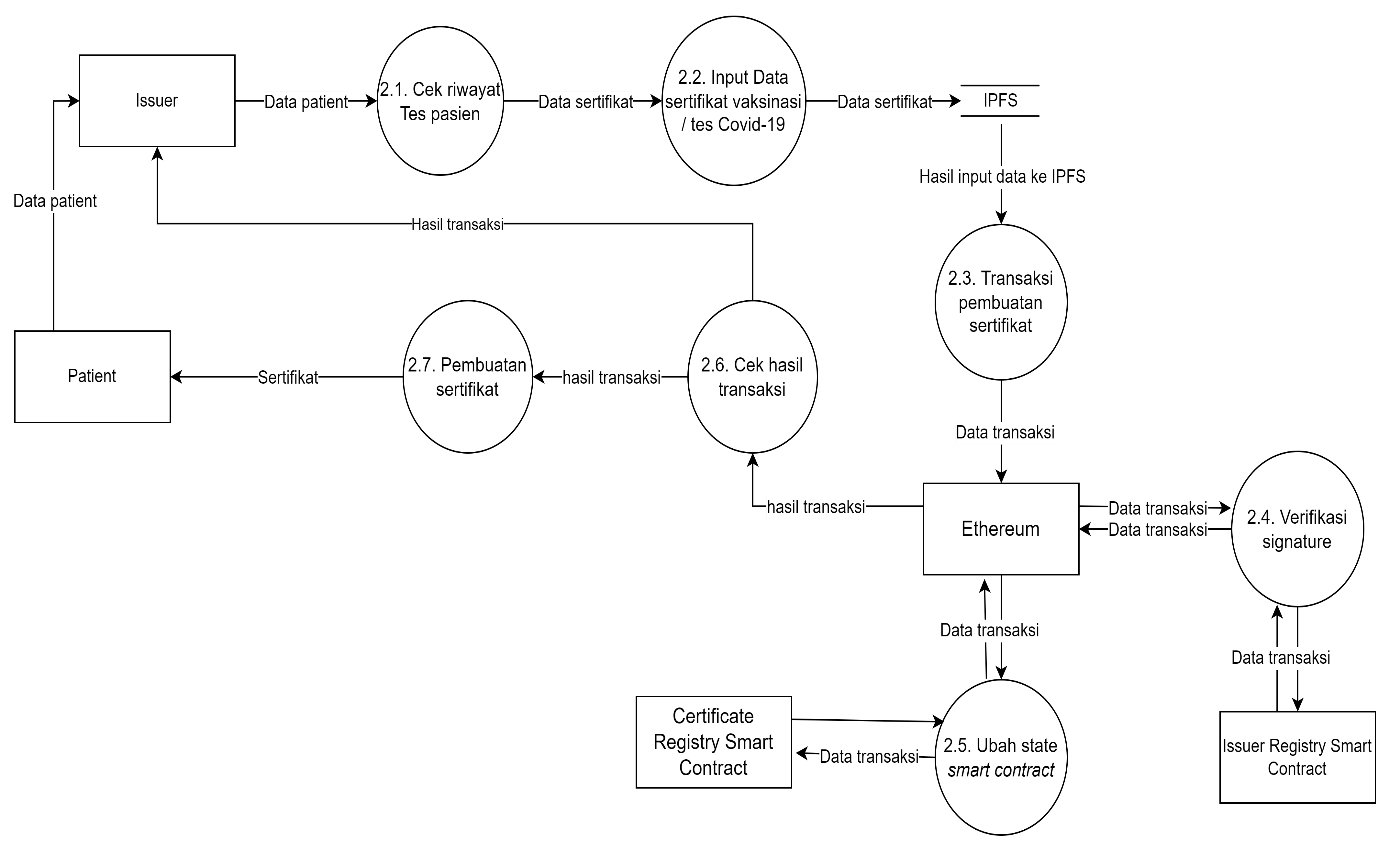
Pada *data flow diagram* level satu dijelaskan proses apa saja yang terjadi pada aplikasi. Pada *data flow diagram level* 1, proses tersebut dipecah ke dalam beberapa bagian yang lebih rinci. Pada gambar 3.19 ditunjukan detail dari proses – proses tersebut. Dimana proses dipecah menjadi : 1) Registrasi *issuer;* 2) Pembuatan sertifikat vaksin; 3) Verifikasi sertifikat.

### ***Data flow Diagram* level 2**

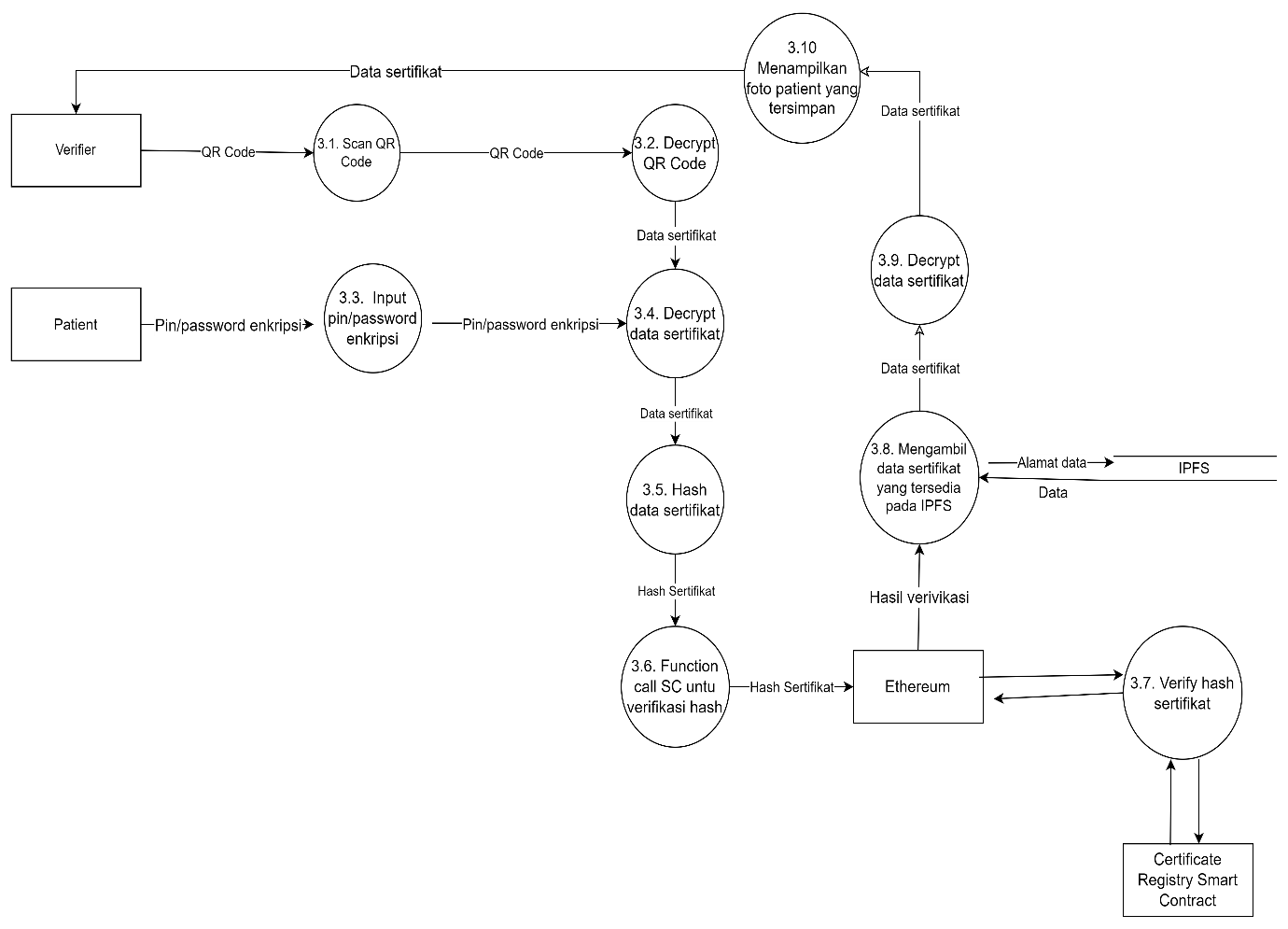
Gambar 3.19 Data Flow Diagram level 1

*Data* *flow diagram* level dua akan menjelaskan secara lebih detail proses aliran data yang ada pada *data flow diagram* level satu. Masing – masing proses akan dijabarkan menjadi sub-proses yang lebih detail.

Gambar 3.20 Data flow diagram level 1 pengesahan issuer

Pada gambar 3.20 ditunjukan proses pengesahan *issuer* oleh *regulator*. *Regulator* akan memasukan data *issuer* dan selanjutnya akan melakukan transaksi untuk menulis data tersebut ke dalam *issuer registry smart contract.* Pada *issuer registry smart contract* akan dilakukan proses untuk mengecek apakah *signer* dari transasksi benar adalah *regulator*.Selanjutnya data akan divalidasi dan ditulis ke dalam *blockchain.* Hasil dari proses penulisan tersebut akan dikirim kembali ke aplikasi dan dilakukan pengecekan hasil dari transaksi tersebut. Hasil transasksi akan ditunjukan kepada regulator.

Gambar 3.21 Data Flow Diagram level 2 pembuatan sertifikat

Pada gambar 3.21 menunjukan alur data pada proses pembuatan sertifikat. Proses dimulai dari pemberian data diri *patient* ke *issuer*, selanjutnya *issuer* akan memasukan data pasien selanjutnya data tersebut akan dienkripsi menggunakan *password* yang dimasukan oleh *patient.* Data yang telah dienkripsi selanjutnya akan diunggah ke IPFS. Setelah data berhasil diunggah ke IPFS, *issuer* akan melakukan transaksi ke *smart contract* untuk menuliskan data ke dalam *certificate registry smart contract.* Pada *certificate registry smart contract* dilakukan validasi data dari transaksi untuk mengecek apakah *signer* terdaftar *issuer registry smart contract*. *Smart contract* selanjutnya akan mengembalikan hasil dari proses penulisan data ke dalam *certificate registry smart contract*. Hasil tersebut akan diproses dan ditunjukan kepada *issuer* dan akan diproses menjadi sebuah sertifikat digital dalam bentuk QR Code.

Gambar 3.22 Data flow diagram level 2 verifikasi sertifikat

Pada gambar 3.22 ditunjukan proses verifikasi data sertifikat vaksin. *Verifier* akan melakukan *scan* pada *QR Code* yang ditunjukan oleh *patient*. Hasil dari scan akan diproses untuk mendekripsi data yang tersimpan dalam bentuk *QR Code*. Selanjutnya pasien akan diminta untuk memasukan *password/pin* miliknyauntuk mendekripsi data dari sertifikat. Data dari sertifikat tersebut selanjutnya akan diproses menjadi sebuah *hash.* *Hash* dari sertifikat akan dibandingkan dengan *hash* yang terdaftar pada *certificate registry smart contract*. *Smart contract* selanjutnya akan mengembalikan hasil dari verifikasi data tersebut.

# BAB 4 Pengujian dan Hasil

## Pengujian

Setelah proses perancangan sistem selesai dilakukan proses pengujian untuk mengetahui apakah sistem yang dibuat sudah berjalan sesuai kondisi yang diperlukan. Dari pengujian yang dilakukan dapat ditentukan tingkat keberhasilan dari sistem yang telah dirancang. Pada penelitian ini dilakukan beberapa pengujian pada sistem yaitu :

1. Uji validatas fungsional dengan metode *whitebox.*
2. Pengujian performa pada proses pembuatan sertifikat, pendaftaran *issuer,* verifikasi sertifikat.
3. Uji validatas kebutuhan pengguna dengan metode *UAT* ( *user acceptance test* ) dilaksanakan oleh petugas kesehatan pada Puskesmas Kedungmundu.

## Uji Validitas Fungsional

Validasi fungsional dari sistem pada penelitian ini dilakukan dengan metode *white* box dengan tujuan untuk memastikan *code* dan *logic* pada sistem telah berjalan sesuai yang diharapkan. Harapannya aplikasi akan memiliki *bug* yang minim atau tidak ada sama sekali. Berikut alur dari pengujian *white box* yang dilakukan.

Tabel 4.1 Validasi fungsionalitas Deployment Smart Contract

|  |  |  |  |
| --- | --- | --- | --- |
| **Data masukan** | **Hasil yang diperoleh** | **Hasil yang diharapkan** | **Kesimpulan** |
| Deployment *Certificate Registry Smart Contract* ke *Rinkeby Test Netwrok* | *Source Code Certificate Registry Smart Contract* terkompilasi tanpa error. | *Source Code Certificate Registry Smart Contract* terkompilasi tanpa error. | Diterima |
|  | Mendapatkan alamat *Certificate Registry smart contract* pada *rinkeby test network*. | Mendapatkan alamat *Certificate Registry smart contract* pada *rinkeby test network*. | Diterima |
| Deployment *issuer Registry Smart Contract* ke *Rinkeby Test Netwrok* | *Source Code issuer Registry Smart Contract* terkompilasi tanpa error. | *Source Code issuer Registry Smart Contract* terkompilasi tanpa error. | Diterima |
|  | Mendapatkan alamat *issuer Registry smart contract* pada *rinkeby test network*. | Mendapatkan alamat *issuer Registry smart contract* pada *rinkeby test network*. | Diterima |

Berdasarkan hasil pengujian *Whitebox* pada tabel 4.1, dapat disimpulkan bahwa *smart contract* berhasil untuk di-*deploy* ke *rinkeby test network.* Selanjutnya *smart contract* dapat digunakan untuk berinteraksi dengan aplikasi klien.

Tabel 4.2 Validasi fungsionalitas Pendaftaran *Issuer*

| **Data masukan** | **Hasil yang diperoleh** | **Hasil yang diharapkan** | **Kesimpulan** |
| --- | --- | --- | --- |
| Buka halaman *website* registrasi *issuer* | Halaman registrasi *issuer* terbuka, menampilkan sejumlah form dan permintaan untuk login *Metamask.* | Halaman registrasi *issuer* terbuka, menampilkan sejumlah form dan permintaan untuk login *Metamask* | Diterima |
| Tidak melakukan *login* pada *Metamask* | *Form* pada *website* tidak dapat diklik dan terdapat *overlay loading* untuk meminta pengguna masuk ke *Metamask* | *Form* pada *website* tidak dapat diklik dan terdapat *overlay loading* untuk meminta pengguna masuk ke *Metamask* | Diterima |
| *Login* pada *Metamask* | Website dapat diakses dan *overlay loading* hilang | Website dapat diakses dan *overlay* *loading* hilang | Diterima |
| Isi *form* data *issuer* | *Form* dapat diisi | *Form* dapat diisi | Diterima |
| Melakukan transaksi ke *Blockchain* | *Metamask* muncul untuk melakukan konfirmasi transaksi | *Metamask* muncul untuk melakukan konfirmasi transaksi | Diterima |
| Tranasksi dilakukan oleh *non regulator* | Transaksi gagal, pada *metamask* status transaksi menunjukan status gagal dengan keterangan ada kesalahan pada kontra. | Transaksi gagal dan muncul *pop up modal* dengan informasi bahwa transaksi ditolak | Diterima |
| Transaksi dilakukan oleh *Regulator* | Transaksi sukses dan muncul *pop up* modal transaksi suskes | Transaksi sukses dan muncul *pop up* modal transaksi sukses | Diterima |

Pada tabel 4.2 ditunjukan bahwa fitur untuk melakukan transaksi ke *smart contract* untuk mendaftarkan *issuer*. Selanjutnya aplikasinya juga telah berhasil menampilkan apakah alamat *ethereum address issuer* telah berhasil didaftarkan atau tidak. Selain itu pengujian juga menunjukan hasil bahwa hanya *regulator* atau pihak yang memiliki akses ke *issuer registry smart contract* yang dapat melakukan penulisan ke dalam *smart contract*.

Tabel 4.3 Validasi fungsionalitas Pendaftaran Serifikat

| **Data masukan** | **Hasil yang diperoleh** | **Hasil yang diharapkan** | **Kesimpulan** |
| --- | --- | --- | --- |
| Buka halaman website untuk pendaftaran sertifikat | Halaman pendaftaran serifikat dapat dibuka. | Halaman pendaftaran serifikat dapat dibuka. | Diterima |
| Tidak login *metamask* | Halaman menampilkan *overlay loading* untuk meminta user login ke *metamask.* | Halaman menampilkan *overlay loading* untuk meminta user login ke *metamask.* | Diterima |
| Login *metamask* | Halaman dapat diakses dan *overlay loading* hilang dari tampilan website. | Halaman dapat diakses dan *overlay loading* hilang dari tampilan website. | Diterima |
| Isi *form data* pasien | *Form* dapat diisi. | *Form* dapat diisi | Diterima |
| Menekan tombol cek pada halaman data diri pasien | Website menampilkan riwayat vaksinasi/tes covid-19 pasien apabila telah terdata. | Website menampilkan riwayat vaksinasi/tes covid-19 pasien apabila telah terdata. | Diterima |
| Menekan tombol *next* pada halaman data diri pasien setelah mengisi data pasien | Website beralih ke halaman data sertifikat. | Website beralih ke halaman data sertifikat | Diterima |
| Memilih opsi sertifikat vaksin covid-19 | Website menampilkan form untuk data sertifikat vaksin. | Website menampilkan form untuk data sertifikat vaksin. | Diterima |
| Data pasien yang diinputkan sebelumnya memiliki data vaksin covid-19 yang terdaftar pada sistem | Pada halaman data sertifikat vaksin hanya dapat dilakukan input dosis vaksin selanjutnya | Pada halaman data sertifikat vaksin hanya dapat dilakukan input dosis vaksin selanjutnya | Diterima |
| Memasukan data sertifikat vaksin | Form dapat diisi | Form dapat diise | Diterima |
| Memilih opsi sertifikat tes covid-19 | Website menampilkan form untuk data tes covid-19 | Website menampilkan form untuk data tes covid-19 | Diterima |
| Memasukan data sertifikat tes Covid-19 | Form dapat diisi | Form dapat diisi | Diterima |
| Menekan tombol *next* pada halaman data sertifikat setelah mengisi form data sertifikat | Halaman berpindah ke halaman pengambilan foto | Halaman berpindah ke halaman pengambilan foto | Diterima |
| Menekan tombol *next* pada halaman data sertifikat sebelum mengisi data sertifikat | Tombol tidak dapat ditekan | Tombol tidak dapat ditekan | Diterima |
| Pada halaman pengambilan foto pasien, dilakukan pengambilan foto pasien | Foto pasien yang telah muncul ditampilkan pada halaman | Foto pasien yang telah muncul ditampilkan pada halaman | Diterima |
| Menekan tombol *next* pada halaman pengambilan foto | Website menampilkan halaman untuk transaksi ke *smart contract* | Website menampilkan halaman untuk transaksi ke *smart contract* | Diterima |
| Ditampilkan data pasien | Ditampilkan data pasien | Diterima |
| Menekan tombol *add pin* | Muncul *modal* untuk memasukan pin untuk enkripsi data | Muncul *modal* untuk memasukan pin untuk enkripsi data | Diterima |
| Memasukan pin yang sama pada kedua kolom input | Pin dapat dimasukan dan modal tertutup | Pin dapat dimasukan dan modal tertutup | Diterima |
| Memasukan pin yang berbeda pada kedua kolom input | Pin tidak dapat dimasukan dan muncul warning pada modal | Pin tidak dapat dimasukan dan muncul warning pada modal | Diterima |
| Menekan tombol *Submit* pada halaman untuk transaksi ke *smart contract* dengan akun *issuer* | Tombol tidak dapat diklik apabila pin/password belum dimasukan | Tombol tidak dapat diklik apbila pin/password belum dimasukan | Diterima |
| Tombol dapat diklik apabila pin/password telah dimasukan | Tombol dapat diklik apabila pin/password telah dimasukan | Diterima |
| Pada tabel data pasien muncul hash dari data pasien beserta data sertifikat | Pada tabel data pasien muncul hash dari data pasien beserta data sertifikat | Diterima |
| Muncul notifikasi dari *metamask* untuk mengkonfirmasi transaksi | Muncul notifikasi dari *metamask* untuk mengkonfirmasi transaksi | Diterima |
| Tombol tidak dapat diklik apabila pin/password belum dimasukan | Tombol tidak dapat diklik apbila pin/password belum dimasukan | Diterima |
| Apabila transaksi sukses akan muncul modal menampilkan sertifikat dalam bentuk QR Code | Apabila transaksi sukses akan muncul modal menampilkan sertifikat dalam bentuk QR Code | Diterima |
| Apabila transasaksi gagal muncul modal menunjukan transaksi gagal | Apabila transaksi gagal muncul modal menunjukan transaksi gagal | Diterima |
| Memasukan nomor whatsapp pengguna dan menekan tombaol send to whatsapp | Membuka halaman whatsapp dan file QR code dapat terdownload serta dapat dikirimkan melalui halaman whatsapp | Membuka halaman whatsapp dan file QR code dapat terdownload serta dapat dikirimkan melalui halaman whatsapp | Diterima |
| Menekan tombol submit dengan akun *non issuer* | Transaksi ditolak dapat dilihat pada riwayat aktivitas pada *metamask* | Transaksi ditolak dapat dilihat pada riwayat aktivitas pada *metamask* | Diterima |

Hasil dari pengujian *white box* pada tabel 4.3 menunjukan bahwa fitur pedaftaran sertifikat telah bekerja dengan baik. Seluruh komponen yang ada pada fitur tersebut telah bekerja dengan semestinya. Selain itu *smart contract* telah berhasil untuk mengautentikasi apakah transaksi dibuat oleh *issuer* yang tardaftar atau tidak.

Tabel 4.4 Validasi fungsionalitas Verifikasi Sertifikat

|  |  |  |  |
| --- | --- | --- | --- |
| **Data masukan** | **Hasil yang diperoleh** | **Hasil yang diharapkan** | **Kesimpulan** |
| Menekan tombol *switch camera* | Sumber input kamera berubah | Sumber input kamera berubah | Diterima |
| Scan sertifikat yang terdaftar | Muncul modal untuk memasukan pin/password | Muncul modal untuk memasukan pin/password | Diterima |
| Memasukan pin/password yang benar | Data sertifikat beserta foto pasien ditampilkan | Data sertifikat beserta foto pasien ditampilkan | Diterima |
| Memasukan pin/password yang salah | Data sertifikat tidak ditampilkan | Data sertifikat tidak ditampilkan | Diterima |
| Scan sertifikat yang tidak terdaftar | Muncul *alert* memberitahukan bahwa sertifikat tidak terdaftar | Muncul *alert* memberitahukan bahwa sertifikat tidak terdaftar | Diterima |

Berdasrkan hasil dari pengujian pada tabel 4.4 menunjukan bahwa fitur verifikasi dapat bekerja sesuai yang dibutuhkan. Fitur dapat memverifikasi apakah sertifikat covid-19 terdaftar. Selain itu melalui tes yang telah dilakukan dapat memastikan bahwa data pengguna telah terenkripsi.

## Pengujian performa

Pada bagian ini akan diuji waktu yang dibutuhkan untuk melakukan transaksi dengan *smart contract* pada *Rinkeby Test Network.* Perhitungan dilakukan dengan menghitung waktu yang dibutuhkan dari fungsi untuk transaksi dipanggil hingga *callback* diterima. Selain itu untuk proses registrasi sertifikat vaksin dan tes covid-19 dilakukan perhitungan waktu yang diperlukan untuk meproses fungsi yang berperan dalam mengolah data hingga diperloah sertifikat yang telah selesai, dalam fungsi tersebut didalamnya juga memanggil fungsi untuk melakukan transaksi dengan *smart contract*.

Gambar 4.1 Grafik performansi pembuatan sertifikat

Pada gambar 4.1 ditunjukan performa dari proses pembuatan sertifikat. Secara rata – rata transaksi memerlukan waktu 33,43 detik dan rata -rata waktu yang diperlukan aplikasi dalam memerlukan waktu 36,039. Dengan rata – rata waktu diatas 30 detik, proses registrasi sertifikat dapat dikategorikan sangat lama, bahkan terdapat proses yang memakan waktu hingga 180 detik. Grafik menunjukan performa pembuatan sertifikat sangat dipengaruhi oleh waktu yang diperlukan untuk melakukan transaksi dengan *smart contract*. Faktor yang mempengaruhi waktu transaksi pada jaringan *blockchain* adalah kemampuan komputasi pada jaringan *rinkeby* dan jumlah transaksi yang diprosesdi waktu yang bersamaan. Oleh karenanya terdapat lonjakan pada grafik yang sangat mungkin diakibatkan karena adanya peningkatan jumlah transaksi yang terjadi pada jaringan *blockchain.*

Gambar 4.2 Grafik performansi verifikasi sertifikat.

Pada gambar 4.2 ditunjukan performa verifikasi sertifikat. Proses verifikasi sertifikat ke *smart contract*  pada jaringan *blockchain* membuthkan waktu rata – rata 0,89 detik dan untuk rata – rata waktu yang diperlukan untuk melakukan proses verifikiasi sertifikat adalah 3,73 detik. Pada proses verifikasi waktu yang diperlukan jauh lebih cepat dibandingkan dengan proses pembuatan sertifikat. Waktu pembacaan ke *smart contract* rata – rata hanya memerlukan waktu 0,89 detik, berbanding jauh dari proses yang diperlukan untuk melakukan transaksi. Hal ini dikarenakan *node* pada jaringan *blockchain* tidak perlu melakukan verifikasi seperti yang dilakukan pada proses transaksi. Sehingga tidak memerlukan banyak sumber daya komputasi.

Gambar 4.3 Grafik perfomansi transaksi registrasi isssuer

Pada gambar 4.3 dan gambar 4.4 masing masing menunjukan total waktu yang dibutuhkan untuk melakukan proses registrasi *issuer* dan waktu yang diperlukan untuk melakukan transaksi registrasi *issuer.* Rata – rata total waktu yang diperlukan untuk melakukan registrasi *issuer* adalah 26,648 detik sedangkan untuk melakukan transaksi diperlukan waktu rata – rata 26,56 detik. Hasil tersebut menunjukan hal yang sama dengan proses pembuatan sertifikat dimana proses sangat dipengaruhi oleh waktu yang diperlukan untuk melakukan transaksi dengan *smart contract*. Namun dengan hasil rata – rata waktu yang lebih rendah dibanding proses pembuatan sertifikat, kita dapat menganalisa bahwa kompleksitas dari proses yang dilakukan pada *smart contract* juga mempengaruhi waktu yang diperlukan untuk melakukan transaksi dengan *smart contrat.*

Gambar 4.4 Grafik performansi registrasi issuer.

## UAT (User Acceptance Testing)

Pengujian UAT dilakukan dengan memberikan kusioner kepada petugas kesehatan yang bertugas dalam pembuatan sertifikat vaksin dan tes Covid-19 di Puskesmas Kedungmundu Kota Semarang. Tujuan dari tes UAT adalah untuk melakukan verifikasi terhadap *user requirement.* Berikut adalah tabel pengujiannya.

Tabel 4.5 Pilihan jawaban kuisioner

|  |  |  |
| --- | --- | --- |
| Pilihan | Nilai Jawaban | Bobot |
| A | Baik | 5 |
| B | Cukup | 3 |
| C | Kurang | 1 |

Terdapat 25 pertanyaan yang harus diisi oleh petugas kesehatan pada Puskesma Kedungmundu Kota Semarang. Dimana untuk baik memiliki bobot penilaian 5, cukup bernilai 3 dan kurang bernilai 1. UAT dilakukan oleh bapak Miko sebagai penanggung jawab saya selama melakukan pengumpulan data di Puskesmas Kedungmundu Kota Semarang. Berikut adalah tabel pertanyaan beserta hasil dari UAT yang telah dilakukan.

Tabel 4.6 User Acceptance Testing

| No | Pertanyaan | Hasil |
| --- | --- | --- |
| 1 | Bagaimana tanggapan anda terkait fitur pengisian nama pasien? | A |
| 2 | Bagaimana tanggapan anda terkait fitur pengisian nomor kependudukan pasien? | A |
|  | Bagaimana tanggapan anda terkait fitur input gender pasien? | A |
| 3 | Bagaimana tanggapan anda terkait fitur input umur pasien? | A |
| 4 | Bagaimana tanggapan anda terkait fitur formulir data sertifikat vaksin/tes covid-19? | A |
| 5 | Bagaimana tanggapan anda terkait fitur pengambilan foto pasien? | A |
| 6 | Bagaimana tanggapan anda terkait fitur penambahan sertifikat? | A |
| 7 | Bagaimana tanggapan anda terkait fitur input pin untuk enkripsi sertifikat? | A |
| 8 | Bagaimana tanggapan anda terkait fitur submit data sertifikat covid-19? | A |
| 9 | Bagaimana tanggapan anda terkait fitur untuk mecegah pembuatan sertifikat oleh pihak yang tidak memiliki otoritas*?* | A |
| 10 | Bagaimana tanggapan anda terkait fitur untuk mengirim sertifikat ke nomor *whatsapp* pasien? | A |
| 11 | Bagaimana menurut anda terkait fitur untuk mengubah kamera? | A |
| 12 | Apakah halaman *website* untuk registrasi sertifikat menarik dah mudah digunakan? | A |
| 13 | Bagaimana tanggapan anda terkait fitur *scan QR Code* sertifikat vaksin? | A |
| 14 | Bagaimana menurut anda terkait fitur mengubah kamera QR *scanner*? | A |
| 15 | Bagaimana tanggapan anda terkait *output* dari verifikasi sertifikat vaksin? | A |
| 16 | Apakah halaman *website* untuk verifikasi sertifikat vaksin Covid-19 menarik dan mudah digunakan? | A |
| 17 | Bagaimana tanggapan anda terkait fitur formulir pengisian data calon *issuer*  ? | A |
| 18 | Bagaimana tanggapan anda terkait fitur input alamat publik *Ethereum Account Address?* | A |
| 19 | Bagaimana tanggapan anda terkait fitur input nama institusi asal *issuer?* | A |
| 20 | Bagaimana tanggapan anda terkait fitur input nomor kepegawaian? | A |
| 21 | Bagaimana tanggapan anda terkait fitur input nomor kontak *issuer?* | A |
| 22 | Bagaimana tanggapan anda terkait fitur tombol *submit* registrasi *issuer*? | A |
| 23 | Bagaimana tanggapan anda terkait proses registrasi *issuer*? | A |
| 24 | Apakah halaman *website* untuk regristasi *issuer* menarik dan mudah digunakan? | A |
| 25 | Bagaimana tanggapan anda terkait fitur pengisian nama pasien? | A |

Pada tabel 4.6 ditunjukan hasil dari kuisioner yang diisi oleh petugas kesehatan pada Puskesmas Kedungmundu. Semua pertanyaan telah diisi dengan opsi A atau nilai baik. Skor rata – rata setiap pertanyaan dapat dihitung dengan cara dimana variable x merupakan bobot nilai masing – masing pertanyaan. Dengan bobot skor lima untuk opsi A maka nilai rata – rata dari setiap pertanyaan adalah lima (baik). Dengan rata – rata skor lima meunjukan bahwa setiap fungsionalitas pada sistem telah dianggap baik dari sudut pandang pengguna.

# BAB 5 Kesimpulan dan Saran

## Kesimpulan

Berdasarkan hasil pengujian dan analisis yang telah dilakukan dalam penelitian tugas akhir ini dapat ditarik kesimpulan sebagai berikut :

1. Untuk membuat sistem penyimpanan pada jaringan *blockchain ethereum* dapat diawali dengen penulisan *smart contract* yang ditulis dengan bahasa *solidity.* Selanjutnya dilakukan proses *deployment smart cotnract*  ke jaringan *Ethereum.*
2. *Blockchain* dan *smart contract* dapat digunakan sebagai penyimpanan sertifikat digital hasil tes atau vaksinasi Covid-19. Aplikasi yang dirancang dapat berinteraksi dengan *smart contract* pada jaringan *Testnet Rinkeby* dengan medium Metamask sebagai *wallet* yang menyimpan akun *Ethereum*.
3. Berdasarkan hasil validasi fungsiniolitas dengan metode *white box testing* fitur yang dirancang telah berjalan sesuai skenario algoritma yang dirancang.
4. Transaksi ke *smart contract* sangat dipengaruhi beberapa variabel seperti kemampuan komputasi pada jaringan *rinkeby* dan jumlah transaksi yang diprosesdi waktu yang bersamaan. Proses *smart contract* yang lebih rumit juga cenderung membutuhkan waktu yang lebih lama untuk diproses.
5. Sistem yang diajukan dapat dinyatakan telah memunuhi kebutuhan pengguna.

## Saran

Berdasarkan hasil penelitian tugas akhir sarat yang dapat diberikan sebagai berikut :

1. Dalam pengujian selanjutnya dapat lebih mempertimbangkan waktu yang diperlukan untuk melakukan transaksi pada blockchain.
2. Ethereum tidak mendukung enkripsi pada *smart contract* oleh karenanya enkripsi perlu dilakukan di sisi klien sebelum transasksi dilakukan. Selanjutnya perlu dipastikan setiap kliens menggunakan metode enkripsi yang sama.
3. Diperlukan penelitian lebih lanjut untuk melakukan tracking pada batch vaksin yang digunakan untuk menghindari adanya pemalsuan data vaksin.

**Daftar Pustaka**

[1] *WHO Director-General's opening remarks at the media briefing on COVID-19 - 11 March 2020*. (2020, Maret 11). Retrieved from WHO: <https://www.who.int/director-general/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19---11-march-2020>

[2] Arianto, B. (2020). Dampak Pandemi COVID-19 terhadap Perekonomian Dunia. *JURNAL EKONOMI PERJUANGAN*.

[3] *‘Landmark collaboration’ to make COVID-19 testing and treatment available to all*. (2020,April24).Retrieved from United Nation: <https://news.un.org/en/story/2020/04/1062512>

[4] Kate Proctor, I. S. (2020, Maret 30). *'Immunity passports' could speed up return to work after Covid-19*. Retrieved from The Guardian: https://www.theguardian.com/world/2020/mar/30/immunity-passports-could-speed-up-return-to-work-after-covid-19

[5] Rokom. (2021, Juli 4). *Pemerintah Integrasikan Data Kesehatan dengan Aplikasi Pedulilindungi untuk Mencegah Pemalsuan Hasil Tes COVID-19 sebagai Syarat Perjalanan*. Retrieved October,5 ,021 from Sehat Negeriku: <https://sehatnegeriku.kemkes.go.id/baca/rilis-media/20210704/0738021/pemerintah-integrasikan-data-kesehatan-dengan-aplikasi-pedulilindungi-untuk-mencegah-pemalsuan-hasil-tes-covid-19-sebagai-syarat-perjalanan/>

[6] CNN Indonesia.(n.d) *Pakar Kritisi Kebocoran Data Sertifikat Vaksin PeduliLindungi*. (2021, September 5). Retrieved October,5 ,021 from CNN INDONESIA: <https://www.cnnindonesia.com/teknologi/20210905085011-185-689866/pakar-kritisi-kebocoran-data-sertifikat-vaksin-pedulilindungi>

[7] Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System.

[8] Vitalik Buterin, e. (2013). Ethereum Whitepaper.

[9] Ethereum.org. (n.d.). *Ethereum Documentation*. Retrieved November 28, 2021, from Ethereum.org: https://ethereum.org/en/developers/docs/

[10] iani Wu, N. K. (2018). Application of Blockchain Technology in Sustainable. Sustainability 2018, 10, 3067

[11] Janvi Dattani, H. S. (2019). Overview of Blockchain Technology. Asian Journal of Convergence in Technology .

[12] Jani, S. (2017). An Overview of Ethereum & Its Comparison with. International Journal of Scientific & Engineering Research.

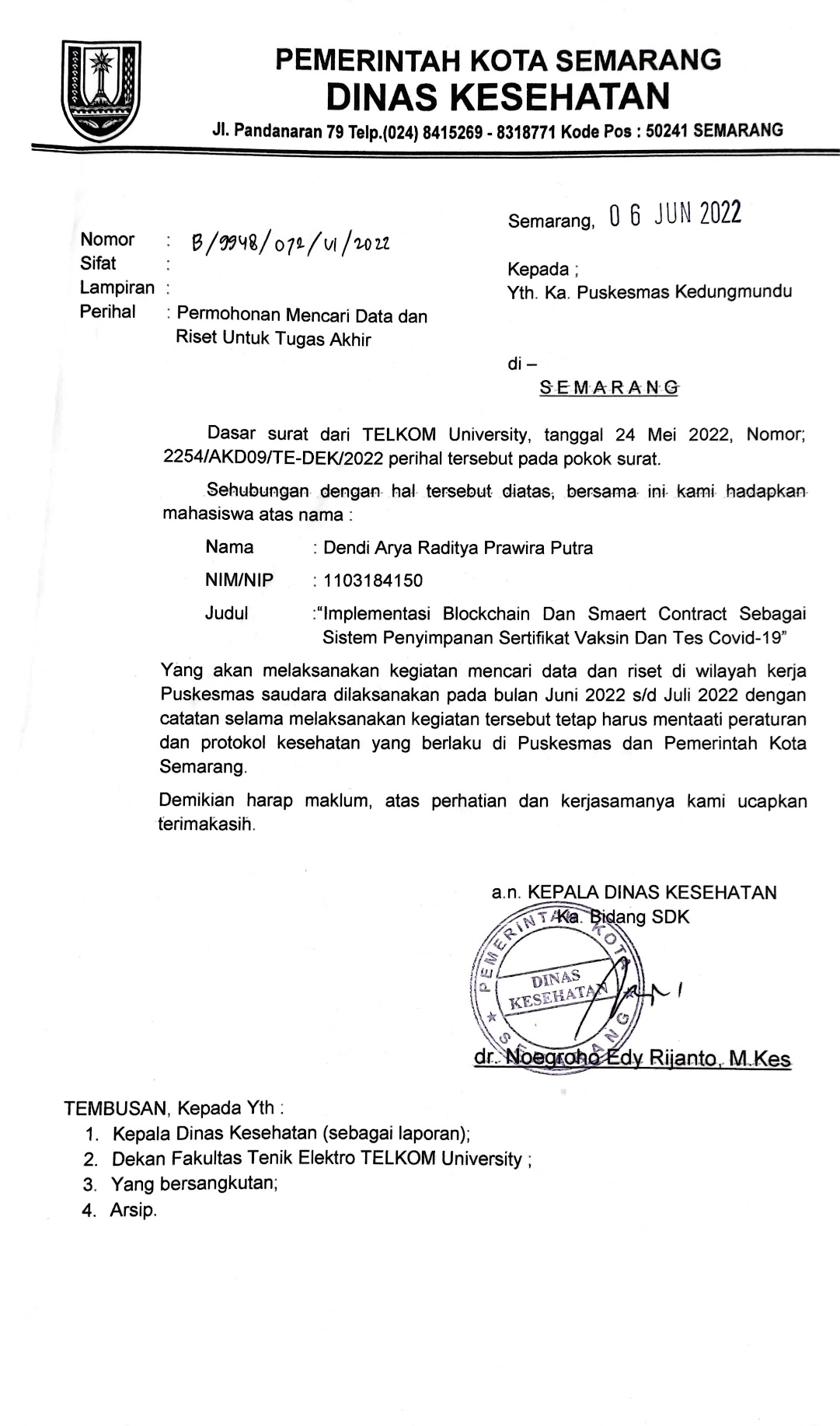
[13] Loi Luu, D.-H. C. (2016). Making Smart Contracts Smarter. CCS '16: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (pp. 254-269). ACM library digita

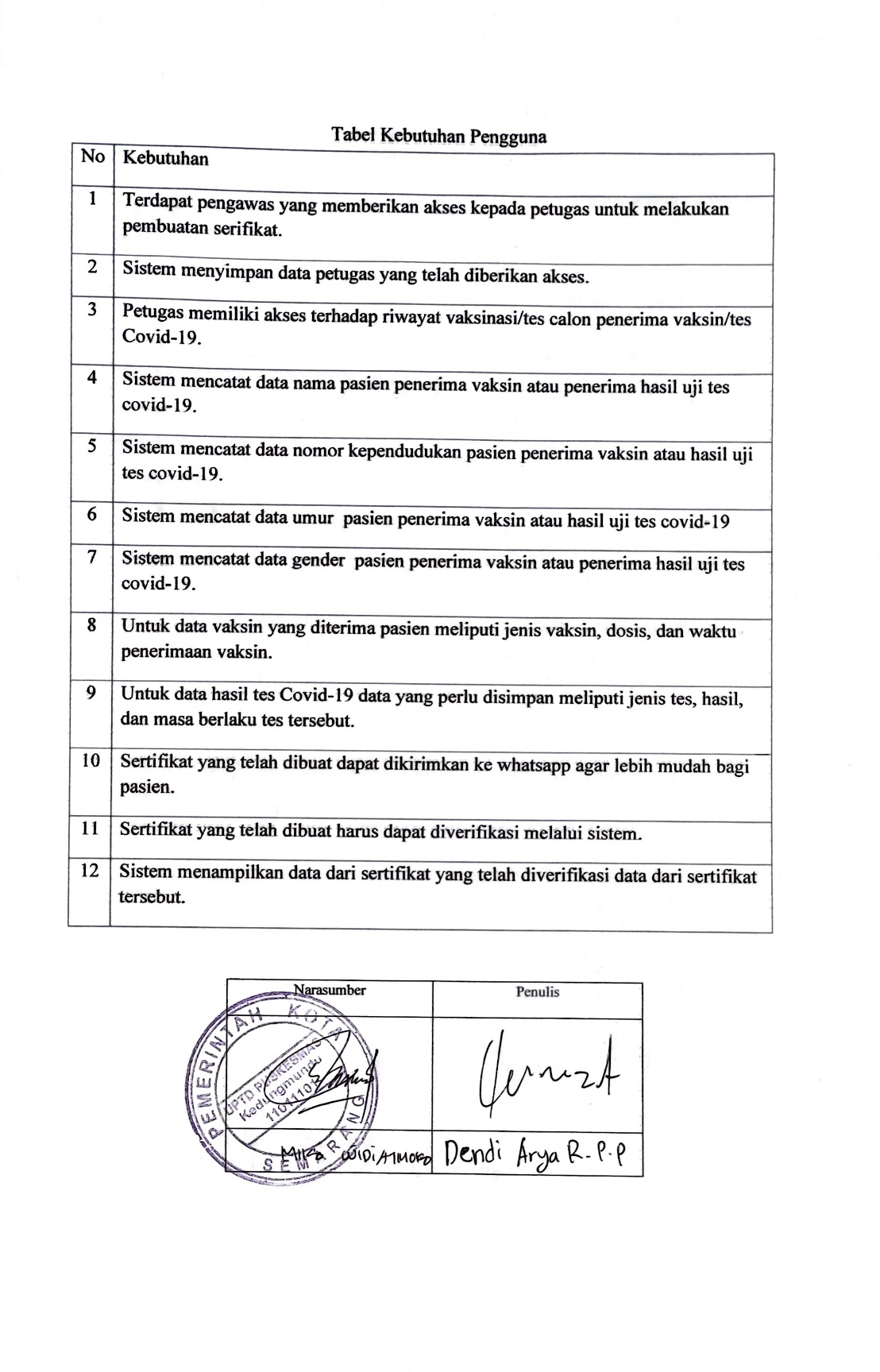
[14] Soliditylang. (n.d.). *Solidity*. Retrieved January 5, 2022, from Solidity: <https://docs.soliditylang.org/en/v0.8.15/>

[15] EtheBrownie. (n.d.). *Brownie*, c01ff902. Retrieved June 24, 2022, from Brownie: <https://eth-brownie.readthedocs.io/en/stable/>

[16] Web3js. (n.d.). *Web3js Docs*, 77005c07. Retrieved January 5, 2022, from https://web3js.readthedocs.io/en/v1.5.2/

# Lampiran

1. Surat izin penelitian
2. Tabel User Requirement



1. User Accpetance Testing

