Penerapan *Multi Smart Contract* pada *Naive Chain* untuk Meningkatkan Integritas Bukti Digital dan *Chain of Custody*

Arif Surya Putra
16917202

Tesis diajukan sebagai syarat untuk meraih gelar Magister Komputer
Konsentrasi Forensika Digital
Program Studi Magister Teknik Informatika
Jurusan Teknik Informatika Fakultas Teknologi Industri
Universitas Islam Indonesia
2020

Lembar Pengesahan Pembimbing

Penerapan *Multi Smart Contract* pada *Naive Chain* untuk Meningkatkan Integritas Bukti Digital dan *Chain of Custody*

Arif Surya Putra

16917202

Yogyakarta, Desember, 2020

Pembimbing

Dr. Yudi Prayudi, S.Si, M.Kom

Lembar Pengesahan Penguji

Penerapan *Multi Smart Contract* pada *Naive Chain* untuk Meningkatkan Integritas Bukti Digital dan *Chain of Custody*

Arif Surya Putra
16917202

Yogyakarta, Desember 2020

Tim Penguji,

Dr. Yudi Prayudi, S.Si, M.Kom
Ketua

Dr. Imam Riadi, M.Kom
Anggota I

Dr. Ir. Bambang Sugiantoro, S.Si., M.T.
Anggota II

Mengetahui,
Ketua Program Pascasarjana Fakultas Teknologi Industri
Universitas Islam Indonesia

Izzati Muhimmah, ST.,M.Sc., Ph.D

Pernyataan Keaslian Tulisan

Dengan ini saya menyatakan bahwa tesis ini merupakan tulisan asli dari penulis, dan tidak berisi material yang telah diterbitkan sebelumnya atau tulisan dari penulis lain terkecuali referensi atas material tersebut telah disebutkan dalam tesis. Apabila ada kontribusi dari penulis lain dalam tesis ini, maka penulis lain tersebut secara eksplisit telah disebutkan dalam tesis ini.

Dengan ini saya juga menyatakan bahwa segala kontribusi dari pihak lain terhadap tesis ini, termasuk bantuan analisis statistik, desain survei, analisis data, prosedur teknis yang bersifat signifikan, dan segala bentuk aktivitas penelitian yang dipergunakan atau dilaporkan dalam tesis ini telah secara eksplisit disebutkan dalam tesis ini.

Segala bentuk hak cipta yang terdapat dalam material dokumen tesis ini berada dalam kepemilikan pemilik hak cipta masing-masing. Apabila dibutuhkan, penulis juga telah mendapatkan izin dari pemilik hak cipta untuk menggunakan ulang materialnya dalam tesis ini.

Yogyakarta, Desember 2020

Arif Surya Putra, S.Kom

Abstrak

Penerapan *Multi Smart Contract* pada *Naive Chain* untuk Meningkatkan Integritas Bukti Digital dan *Chain of Custody*

Bukti digital yang disimpan dalam blok mengurangi kinerja blok, mengurangi kecepatan akses blok dan membuat media penyimpanan data cepat habis karena data yang akan tersimpan ke dalam blok akan terus bertambah meskipun bukti digital tersebut dihapus dari blok. Informasi-informasi metadata yang diambil dari bukti digital yang hanya berupa informasi dasar terkait bukti digital dapat mengurangi integritas bukti digital dan juga menyulitkan penyidik dalam mengidentifikasi bukti digital yang ditemukan. Pengurangan integritas bukti digital dapat menyebabkan bukti digital ditolak dalam persidangan. Upaya menyelesaikan permasalahan tersebut maka diajukan sebuah model *multi-smart contract* dalam mengelola bukti digital dan *chain of cusotdy* agar kekurangan atau kelemahan untuk memperoleh informasi terkait pengelolaan bukti digital tersebut bisa lebih lengkap dan sempurna. Pemanfaatan multi smart contract juga diharapkan mampu meningkatkan kinerja blok dan juga bisa membuat media penyimpanan bukti digital lebih optimal. Langkah-langkah untuk membuat sistem *multi smart contract* dimulai dengan melakukan identifikasi masalah yang dihadapi terkait kekurangan atau kelemahan metode yang digunakan pada pengelolaan bukti digital yang sudah ada sebelumnya serta alasan pentingnya menerapkan model baru tersebut. Kedua, melakukan studi literatur untuk memperkaya informasi dalam rangka pengembangan dan perancangan metode baru yang diajukan dalam penelitian ini. Ketiga, merancang dan membangun sebuah sistem *multi-smart contract* untuk memisahkan bukti digital berdasarkan jenis atau tipenya agar setelah diproses dapat menghasilkan informasi yang lebih lengkap dan lebih informatif dalam kegiatan pengelolaan bukti digital dan meningkatkan integritas bukti digital. Ketiga, melakukan pengujian terhadap implementasi, integritas, dan performa sistem *multi smart contract* yang sudah dibangun. Keempat, melakukan analisa terhadap desain *multi-smart contract* yang sudah dibuat tersebut untuk mengetahui dan mengambil kesimpulan dengan cara membandingkan dengan metode sebelumnya. Hasil dari penerapan *multi cmart* contract ditemukan bahwa bukti digital memiliki karakteristik dan detail informasi yang berbeda-beda antara satu jenis bukti digital gambar, audio, video, dan dokumen atau jenis bukti digital lainnya. Informasi yang lebih detail mampu meningkatkan integritas bukti digital. Otomatisasi dalam membuat hash dan ekstraksi informasi dari suatu bukti digital dapat mempersingkat waktu first responder dalam menginputkan form isian pada sistem *multi smart contract*. Penyimpanan bukti digital di luar blok dapat meningkatkan performa sistem *multi smart contract*. Penyimpanan bukti yang terbatas pada alat bukti persidangan mampu mengoptimalkan media penyimpanan bukti digital.

Kata kunci

multi-smart contract, chain of custody, blockchain, file properties

Abstract

Implementation of Multi Smart Contracts on the Naive Chain to Improve the Integrity of Digital Evidence and Chain of Custody

Digital evidence stored in blocks reduces block performance, reduces block access speed and makes data storage run out quickly because the data that will be stored in blocks will continue to increase even if the digital evidence is deleted from the block. Metadata information taken from digital evidence which is only basic information related to digital evidence can reduce the integrity of digital evidence and also make it difficult for investigators to identify digital evidence found. A reduction in the integrity of digital evidence can result in digital evidence being rejected in court. In an effort to solve this problem, a multi-smart contract model is proposed in managing digital evidence and the chain of customer so that any shortcomings or weaknesses in obtaining information related to digital evidence management can be more complete and perfect. The use of multi smart contracts is also expected to improve block performance and also make digital evidence storage more optimal. The steps to create a multi smart contract system begin by identifying the problems faced regarding the shortcomings or weaknesses of the methods used in managing existing digital evidence and the reasons for the importance of implementing the new model. Second, conducting literature studies to enrich information in the context of developing and designing the new methods proposed in this study. Third, design and build a multi-smart contract system to separate digital evidence based on its type or type so that after processing it can produce more complete and more informative information in digital evidence management activities and improve the integrity of digital evidence. Third, testing the implementation, integrity and performance of the multi smart contract system that has been built. Fourth, analyze the multi-smart contract design that has been made to find out and draw conclusions by comparing with the previous method. The results of the application of multi-cmart contracts found that digital evidence has different characteristics and details of information between one type of digital evidence, images, audio, video, and documents or other types of digital evidence. More detailed information can improve the integrity of digital evidence. Automation in creating hashes and extracting information from digital evidence can shorten the time for first responders to input fields in a multi-smart contract system. Storage of digital evidence outside the block can improve the performance of a multi smart contract system. The limited storage of evidence on trial evidence can optimize digital evidence storage.

Keywords:

multi smart contract,chain of custody, blockchain, file properties

Pernyataan Keaslian Tulisan

Dengan ini saya menyatakan bahwa tesis ini merupakan tulisan asli dari penulis, dan tidak berisi material yang telah diterbitkan sebelumnya atau tulisan dari penulis lain terkecuali referensi atas material tersebut telah disebutkan dalam tesis. Apabila ada kontribusi dari penulis lain dalam tesis ini, maka penulis lain tersebut secara eksplisit telah disebutkan dalam tesis ini.

Dengan ini saya juga menyatakan bahwa segala kontribusi dari pihak lain terhadap tesis ini, termasuk bantuan analisis statistik, desain survei, analisis data, prosedur teknis yang bersifat signifikan, dan segala bentuk aktivitas penelitian yang dipergunakan atau dilaporkan dalam tesis ini telah secara eksplisit disebutkan dalam tesis ini.

Segala bentuk hak cipta yang terdapat dalam material dokumen tesis ini berada dalam kepemilikan pemilik hak cipta masing-masing. Apabila dibutuhkan, penulis juga telah mendapatkan izin dari pemilik hak cipta untuk menggunakan ulang materialnya dalam tesis ini.

Yogyakarta, Desember 2020

Arif Surya Putra, S.Kom

Daftar Publikasi

Putra, A.S., Prayudi, Y. (2021). Implementasi Multi Smart Contract pada Bukti Digital dan Chain of Custody dalam Meningkatkan Keamanan dan Integritas Bukti Digital. Jurnal Sistem dan Teknologi Informasi (JUSTINDO). Jember. Jawa Timur.

Publikasi yang menjadi bagian dari tesis

Sitasi publikasi 1

Kontributor	Jenis Kontribusi
Author Arif Surya Putra	Mendesain eksperimen (60%) Menulis *paper* (70%)
Author Yudi Prayudi	Mendesain eksperimen (20%) Menulis dan mengedit *paper* (20%)

Halaman Kontribusi

Tidak ada kontribusi dari pihak lain

Halaman Persembahan

Sujud Syukur atas segala anugerah dan nikmat yang sudah diberikan Alloh S.W.T yang telah mempermudah saya dalam menyelesaikan tesis ini. Tesis ini saya persembahkan pertama untuk Bapak dan Ibu saya, yang senantiasa istiqomah selalu mend'oakan dan mendukung saya dalam menyelesaikan studi ini. Tidak lupa pula tesis ini saya persembahkan kepada Istri dan Anak saya yang luar biasa selalu setia mendampingi saya melewati segala proses untuk mencapai titik akhir pada proses penyelesaian tesis ini.

Tesis ini juga saya persembahkan kepada dosen pembimibing dan kaprodi Magister Informatika yang selalu memberikan semangat dan dukungan dalam penyelesaian tesis ini. Saya mohon maaf jika banyak kesalahan yang disengaja ataupun tidak disengaja selama saya menempuh pendidikan di Universitas Islam Indonesia.

Persembahan ini juga saya berikan kepada pimpinan/atasan, seluruh rekan kerja dan semua kolega yang memberikan support dan keluangan waktu dalam menyelesaikan tesis ini. Semoga ini menjadi langkah awal dalam memajukan perusahaan dan mengembangkan karir saya..

Tesis ini saya persembahkan kepada keluarga, adik-adik saya, saudara dan rekan rekan yang sudah memberikan kontirbusi sehingga bisa membantu saya menyelesaikan studi saya.

Kata Pengantar

Puji syukur kehadirat Allah SWT atas rahmat serta karunia-Nya, sehingga Tesis ini berhasil diselesaikan tepat pada waktunya. Sholawat serta salam tercurah untuk Baginda Rosulullah Terima kasih penulis ucapkan kepada:

1. Rektor Universitas Islam Indonesia, Bapak Prof. Fathul Wahid, S.T., M.Sc., Ph.D yang memberikan kesempatan kepada penulis untuk menimba ilmu di Universitas Islam Indonesia.
2. Dekan Fakultas Teknologi Indudtri Universitas Islam Indonesia, Bapak, Prof. Dr. Ir. Hari Purnomo M.T. yang memberikan fasilitas dan bantuan untuk belajar.
3. Ketua Program Pascasarjana Magister Teknologi Industri Universitas Islam Indonesia, Ibu Izzati Muhimmah, S.T., M.Sc., Ph.D. dengan segala kebijaksanannya dan perhatiannya.
4. Dosen Pembimbing Tesis, Bapak Dr. Yudi Prayudi, S.Si, M.Kom atas segala bimbingan, arahan, motivasi, ilmu dan kebaikannya.
5. Dosen penguji Bapak Dr. Imam Riadi, M.Kom, dan Dr. Ir. Bambang Sugiantoro, S.Si., M.T. atas arahan dan kebaikannya.
6. Bapak Ahmad Muslim, S.Pd dan Ibu Nurhasanah, S.Pd yang telah mencurahkan segenap cinta, kasih sayang, perhatian dan dukungan baik moril maupun materil.
7. Istri saya Mufidatul Ummah, S.Kom dan anakku ananda Ayuna Almahira Elsyanum selalu setia mendampingi dan menjadi penyemangat dalam hidup saya.
8. Seluruh keluarga besar, Bapak/Ibu mertua, adik-adik yang selalu mendoakan saya.
9. Bapak Ronal Rivandi dan seluruh keluarga PT Citraweb Digital Multisolusi yang selalu mensupport kemajuan saya.
10. Semua pihak, baik individu maupun kelompok yang ikut serta menorehkan warna dalam kanvas kehidupan penulis.

Kritik dan saran dari semua pihak yang bersifat membangun selalu penulis harapkan. Akhir kata, penulis sampaikan terima kasih kepada semua pihak yang telah berperan serta dalam penyusunan Tesis ini dari awal sampai akhir.

Yogyakarta, Desember 2020

Penulis

Daftar Isi

Lembar Pengesahan Pembimbing i
Lembar Pengesahan Penguji ii
Abstrak i
Abstract ii
Pernyataan Keaslian Tulisan iii
Daftar Publikasi iv
Halaman Kontribusi v
Halaman Persembahan vi
Kata Pengantar vii
Glosarium xiii
BAB 1 PENDAHULUAN 1
1.1. Latar Belakang 1
1.2. Rumusan Masalah 3
1.3. Tujuan Penelitian 4
1.4. Manfaat Penelitian 4
1.5. Batasan Masalah 4
BAB 2 KAJIAN PUSTAKA 5
2.1. Landasan Teori 5
2.1.1 Bukti Digital 5
2.1.2 *Chain of Custody* 5
2.1.3 *Blockchain* 6
2.1.4 *Smart Contract* 7
2.1.5 *Naivechain* 7
2.1.6 GetID3 10
2.2. Penelitian Sebelumnya dan Kontribusi 11

BAB III METODE PENELITIAN 14
3.1. Identifikasi Masalah 18
3.2. Studi Literatur 18
3.3. Rancangan Sistem *Multi Smart Contract* 18
3.3.1 Analisa Sistem 18
3.3.2 Alur Penyimpanan Bukti Digital dan Chain of Custody pada Naive Chain 19
3.3.3 Rancangan Multi Smart Contract pada Naive Chain 21
3.3.4 Rancangan Antar Muka 22
3.4. Implementasi Sistem *Multi Smart Contract* 27
3.4.1 Membangun Naive Chain 28
3.4.2 Membangun Multi Smart Contract 28
3.4.3 Membangun *Middleware* 28
3.4.4 Membangun *Front End* 29
3.5. Pengujian Sistem 29
3.5.1 Pengujian Kerja Sistem 29
3.5.2 Pengujian implementasi *Multi Smart Contract* 29
3.5.3 Pengujian Integritas Bukti Digital dan *Chain of Custody* 30
3.5.4 Pengujian Performa *Multi Smart Contract* 30
BAB IV HASIL DAN PEMBAHASAN 31
4.1. Implementasi Sistem Multi Smart Contract 31
4.1.1 Membangun Naive Chain 31
4.1.2 Membangun *Multi Smart Contract* 32
4.1.3 Membangun Middleware 33
4.1.4 Membangun Antarmuka Sistem 33
4.2. Pengujian Sistem 38
4.2.1 Pengujian Kerja Sistem 38
4.2.2 Pengujian Implementasi *Multi Smart Contract* 38

4.2.3 Pengujian Integritas Bukti Digital dan *Chain of Custody* 39
4.2.4 Pengujian Performa *Multi Smart Contract* 41
4.3. Analisa 42
4.3.1 Analisa Implementasi *Multi Smart Contract* 42
4.3.2 Analisa Pengujian Integeritas Bukti Digital 43
4.3.3 Analisa Pengujian Performa *Multi Smart Contract* 44
BAB V KESIMPULAN DAN SARAN 45
5.1. Kesimpulan 45
5.2. Saran 46

Daftar Tabel

Tabel 2.1 Tren Kejahatan Siber pada Masa Covid 19 di Dunia 11
Tabel 2.2 Studi Literatur 12
Tabel 3.1 Rancangan Pengujian Kerja Sistem 29
Tabel 3.2 Rancangan Pengujian Implementasi Sistem 30
Tabel 3.3 Rancangan Pengujian Performa Multi Smart Contract 30
Tabel 4.1 Hasil Pengujian Kerja Sistem 38
Tabel 4.2 hasil Pengujian Implementasi Multi Smart Contract 38
Tabel 4.3 Detail Komponen Informasi Berdasarkan TipeBukti Digital 39
Tabel 4.4 Hasil Pengujian Performa Multi Smart Contract 41
Tabel 4.5 Analisa Impelmentasi Multi Smart Contract 42

Daftar Gambar

Gambar 1.1 Tren Kejahatan Siber pada Masa Covid 19 di Dunia 1
Gambar 1.2 Kejahatan Siber Terbanyak Di Indonesia 2
Gambar 2.1 Model Dasar *Blockchain* 8
Gambar 2.2 Model penanganan *Redudant Block* 9
Gambar 2.3 Model Komunikasi Antar *Node* 9
Gambar 2.4 Komponen Utama *Naivechian* 10
Gambar 3.1 Alur Penelitian 17
Gambar 3.2 Alur Penyimpanan Bukti Digital dan Chain of Custody Pada Naive Chain 20
Gambar 3.3 Alur Akses Bukti Digital dan *Chain of Custody* pada *Blockchain* 21
Gambar 3.4 Desain *Multi Smart Contract* untuk Bukti Digital dan *Chain of Custody* 22
Gambar 3.5 Rancangan Halaman *Login* 23
Gambar 3.6 Rancangan Halaman *List Evidence* 23
Gambar 3.7 Rancangan Halaman *Add Evidence* 24
Gambar 3.8 Rancangan Halaman *Chain* 25
Gambar 3.9 Rancangan Halaman *Add Peer* 25
Gambar 3.10 Rancangan Halaman *List Peer* 26
Gambar 3.11 Rancangan Tampilan Halaman *User* 26
Gambar 3.12 Rancangan Halaman *Add Peer* 27
Gambar 3.13 Alur Implementasi Multi Smart Contract 27
Gambar 3.14 Source Kode Helper Smart Contract 28
Gambar 4.1 Menjalankan Naive chain 31
Gambar 4.2 Middleware Naive Chain, Multi Smart Contract dan Frontend 33
Gambar 4.3 Tampilan Halaman *Login* 34
Gambar 4.4 Tampilan Halaman *Add Evidence* 34
Gambar 4.5 Tampilan Halaman *List Evidence* 35
Gambar 4.6 Tampilan Halaman *Chain* 35
Gambar 4.7 Tampilan Hamalam *Add Peer* 36
Gambar 4.8 Tampilan Halaman *List Peer* 36
Gambar 4.9 Tampilan Halaman *Add User* 37
Gambar 4.10 Tampilan Halaman *List User* 37

Glosarium

BDEC : *Blockchain Digital Evidence Cabinet*

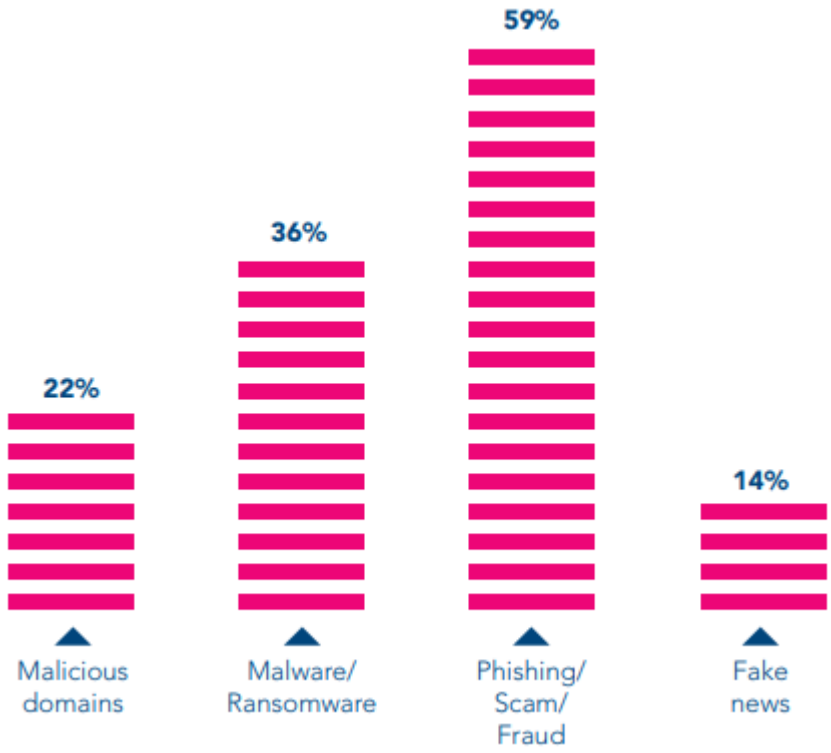
CoC : *Chain of Custody*

MSC : *Multi Smart Contract*

BAB 1
PENDAHULUAN

1.1. Latar Belakang

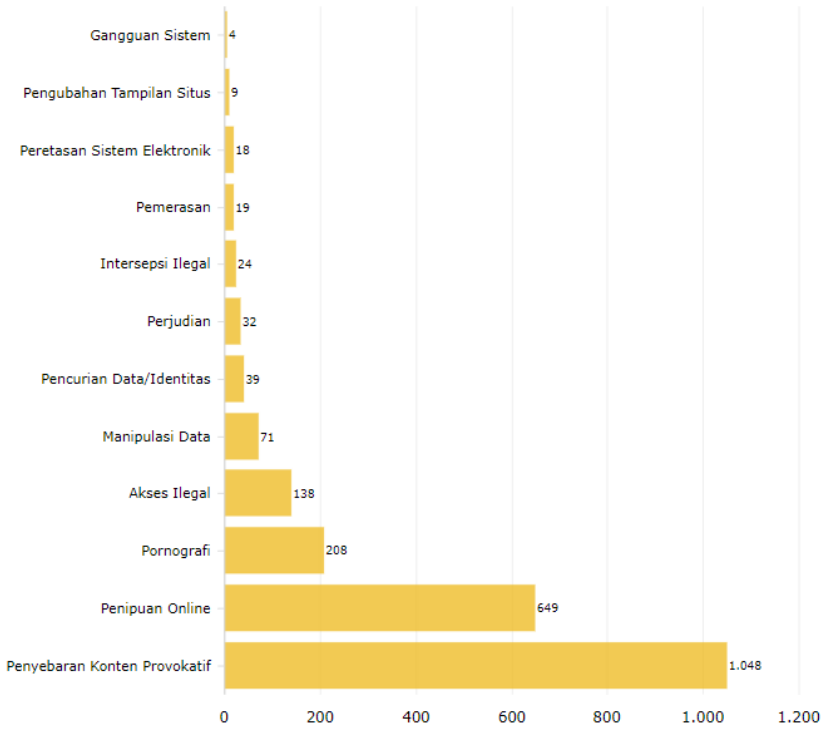
Bertambahnya jumlah dan ukuruan bukti digital yang di simpan dalam suatu media penyimpanan onine, dapat mengurangi performa dalam menyimpan bukti digital. Selain itu penyimpanan bukti digital akan membutuhkan media yang daya tampungnya lebih besar. (Tian Zhihong:2019)



Sumber : Interpol 2020

Gambar 1.1 Tren Kejahatan Siber pada Masa Covid 19 di Dunia

Penambahan jumlah bukti digital ini bisa disebabkan oleh merebaknya kejahatan siber. Meskipun di masa pandemi Covid-19 ternyata tidak mengurangi terjadinya kejahatan siber. Kejahatan penipuan, *phising* dan *scam* secara global tercatat paling banyak yaitu 59 %. Kemudian kejahatan dalam bentuk malware 36 %. Domain yang mencurigakan sebanyak 22% dan berita palsu sebanyak 14 %.



Sumber : katadata 2020

Gambar 1.2 Kejahatan Siber Terbanyak Di Indonesia

Di Indonesia, Kasus penyebaran konten provokatf menjadi tranding sebanyak 1048 kasus, kasus penipuan online sebanyak 649 kasus, kasus pornografi sebanyak 208 kasus, akses ilegal sebanyak 138 kasus, dan manipulasi data sebanyak 71 kasus. Laporan kasus ini diambil dalam rentang bulan Januari 2020 hingga september 2020 yang dipublikasikan oleh Kepolisian Republik Indonesia.

Peningkatan kejahatan di dunia maya memberi dampak pada meningkatnya volume bukti digital yang ditangani oleh para penyelidik. Ini juga dapat menyebabkan lebih banyak dokumentasi dan kompleksitas manajemen bukti digital. (Prayudi, Ashari, & K Priyambodo, 2014). Bukti digital yang tidak dikelola dengan baik maka integritasnya biasanya akan dipertanyakan dan bukti digital tersebut akan mengalami penolakan dalam persidangan (Bonomi, Casini, & Ciccotelli, 2018).

Solusi untuk memelihara integritas, keaslian, dan keutuhan bukti yaitu dengan dikelompokkannya bukti digital dalam satu rak (kabinet) yang berisi bukti digital dan dokumen catatan *chain of custody* dari bukti digital tersebut (Prayudi et al., 2014). Pemanfaatan *cabinet* untuk menyimpan file bukti digital beserta file *chain of custody* masih memiliki kelemahan yaitu rentannya file *chain of custody* dari perubahan dan hilangnya dokumen *chain of custody* tersebut. Penelitian berikutnya diajukan model

konsep *forensic chain* untuk menyimpan bukti digital beserta dokumen *chain of custody* dalam suatu rantai blok (*blockchain*) (Lone & Mir, 2017). Lone yakin dengan menggunakan *blockchain* maka bukti digital dan dokumen *chain of custody* bisa diamankan dari kehilangan dan perubahan data karena *blockchain* memiliki sifat data yang terdistribusi dan terenkripsi. Hanya saja konsep ini hanya sebataas konsep, dan tidak ada parameter yang jelas untuk menguji konsep ini.

Dari konsep yang diajukan Lone, kemudian dikembangkanlah pengelolaan *chain of custody* berbasis *ethereum permission blockchain* (B-CoC) oleh Bonomi (2018). Kekurangan dari model ini, adalah satu investigator hanya bisa mengakses *chain of custody* untuk satu bukti digital. Ketika satu bukti digital dikerjakan oleh banyak investigator maka model ini tidak direkomendasikan. Masalah ini kemudian, di selesaikan dalam penelitian tentang *Blockchain Digital Evidence Cabinet* (B-DEC) (Yunianto, 2019) dan juga penelitian lanjutan yang dilakukan oleh Lone dalam penelitiannya mengenai *forensic chain* untuk *chain of custody* menggunakan *hyperledger* (Lone & Mir, 2019). Pada penelitian tersebut satu smart contract untuk mengelola semua jenis bukti digital. Jadi penggunaanya masih bersifat general, belum ada pengelompokkan.

Berdasarkan penelitian-penelitian yang telah dilakukan tentang pengelolaan bukti digital dan *chain of custody*, maka akan mengembangkan sebuah model blockchain yang bisa memisahkan *chain of custody* berdasarkan jenis atau tipe filenya dalam *smart contract* yang berbeda-beda. Pengembangan blockchain ini diharapkan bisa menjadi sebuah framework, sehingga bisa digunakan diseluruh dunia.

1.2. Rumusan Masalah

Berdasarkan beberapa masalah yang sudah dipaparkan dalam latar belakang, maka dapat dirumuskan masalah yang akan dibahas dalam penelitian ini, antara lain:

1. Bagaimana rancangan sistem *multi smart contract* pada *Naive Chain* untuk meningkatkan integritas bukti digital dan *chain of custody*
2. Bagaimana penerapan *multi smart contract* pada *Naive Chain* untuk meningkatkan integritas bukti digital dan *chain of custody* ?
3. Bagaimana hasil uji penerapan *multi smart contract* pada *Naive Chain* untuk meningkatkan integritas bukti digital dan *chain of custody*?

1.3. Tujuan Penelitian

Dari masalah-masalah yang telah dirumuskan, maka tujuan dari penelitian ini antara lain :

1. Merancang sistem multi smart contract pada Naive Chain untuk meningkatkan integritas bukti digital dan chain of custody
2. Menerpakan metode multi smart contract pada Naive Chain untuk meningkatkan integritas bukti digital dan chain of custody
3. Menguji metode multi *smart contract* dan *naive chain* dalam meningkatkan integritas bukti digital dan *chain of custody*.

1.4. Manfaat Penelitian

Manfaat manfaat yang diperoleh dari penelitian ini antara lain :

1. Penerapan metode *multi smart contract* diharapkan mampu mengelompokkan bukti digital berdasarkan jenisnya sehingga integritas bukti digital bisa meningkat
2. Penggunaan *Naive Chain* dapat meringakan proses penerapan multi smart contract.
3. Dengan pengelompokkan bukti digital menggunakan *multi smart contract* diharapkan dapat menyimpan khusus bukti digital yang menjadi alat bukti, bukan hasil akuisisi barang bukti sehingga membantu mengoptimalkan kapasitas penyimpanan pada *naive chain*

1.5. Batasan Masalah

Batasan Masalah dari penelitian ini antara lain :

1. Pengumpulan data dilakukan melalui studi litreatur
2. Multi smart contract hanya digunakan untuk menyimpan informasi chain of custody bukti digital ke dalam naive chain.
3. Bukti digitial hanya dikelompokkan menjadi gambar, video, audio, dokumen, dan file lainnya.

BAB 2
KAJIAN PUSTAKA

2.1. Landasan Teori

2.1.1 Bukti Digital

Menurut (Maulitasari & Passarella, 2020 : 55) dalam buku berjudul "Teori dan Sejarah Citra Forensik", Bukti Digital adalah kumpulan data yan diperoleh dari semua jenis penyimpanan digital yang dijadikan subyek untuk pemeriksaan forensik komputer. Maka dari itu segala sesuatu yang mengandung informasi digital dapat dijadikan subjek untuk proses penyelidikan. Dalam kegiatan pengumpulan data yang akan digunakan sebagai bukti digital yang legal sesuai dengan undang-undang yang berlaku maka hal tersebut harus dilakukan oleh pakar forensika digital yang terlatih dan berpengalaman pada bidang ini.

Dalam Undang-Undang Nomor 11 Tahun 2008 tentang Informasi dan Transaksi Elektronik (UU ITE) yang menjadi pedoman hukum siber di Indonesia, terdapat istilah alat bukti elektronik dan tidak mencantumkan istilah bukti digital. Bukti elektronik yang dimaksud dalam Undang-Undang Nomor 11 Tahun 2008 terdiri dari informasi elektronik dan dokumen elektronik. Istilah bukti elektronik tersebut oleh para pakar forensika digital lebih dikenal sebagai bukti digital.

Menurut pakar digital Eoghan Casey (2011:1), Bukti Digital didefinisikan sebagai data yang disimpan atau dikirimkan menggunakan perangkat komputer yang digunakan untuk mendukung atau menyangkal suatu teori mengenai bagaimana suatu pelanggaran terjadi atau elemen-elemen penting dari pelanggaran tersebut. Data yang dimaksud dalam pernyataan tersebut adalah kombinasi dasar dari angka-angka yang merepresentasikan berbagai jenis informasi seperti teks, gambar, audio, dan video.

2.1.2 *Chain of Custody*

Chain of Custody didefinsikan sebagai prosedur pencatatan atau dokumentasi kronologis barang bukti sejak barang bukti ditemukan, proses duplikasi, penyimpanan barang bukti baik itu secara fisik ataupun digital hingga sampai pada presentasi dan keputusan akhir

terhadap barang bukti. *Chain of Custody* digunakan untuk memastikan integritas dan orisinalitas dari barang bukti (Prayudi & SN, 2015).

Dokumentasi *Chain of Custody* selama ini tidak memiliki standar yang baku. Sehingga setiap penegak hukum dapat memiliki *form* dokumentasi *Chain of Custody* yang berbeda-beda. Namun untuk dapat diterima di persidangan, sebuah *form chain of custody* setidaknya mencakup informasi "5W dan 1H" untuk mencatat setiap proses investigasi diantaranya (Cosic, 2017):

b. Siapa yang terlibat dalam penanganan barang bukti.
c. Kapan waktu setiap proses penanganan barang bukti dilakukan.
d. Bagaimana proses penanganan yang dilakukan terhadap barang barang bukti.
e. Kemana saja alur perjalanan proses penanganan barang bukti itu dibawa dan dimana disimpan.
f. Mengapa pihak tersebut menanganinya.
g. Apa saja barang bukti yang telah dikumpulkan.

2.1.3 *Blockchain*

Awal munculnya *Blockchain* diperkenalkan oleh Satoshi Nakamoto sebagai teknologi yang digunakan pada *Bitcoin* di jurnalnya pada tahun 2008/2009 (Shorish, 2018).

Walaupun dikenalkan sebagai mata uang, tetapi *Blockchain* tidak serta merta merekam transaksi keuangan saja (Singhal, Dhameja, & Panda, 2018). Menurut (Mougayar, 2016) *Blockchain* sendiri merupakan teknologi baru yang pada dasarnya mempunyai *ledger* atau buku besar yang didistribusikan secara terbuka (*database*) dan mencatat seluruh transaksi yang ada secara detil pada sebuah *blocks* yang setiap *block* mempunyai waktu dan saling terhubung ke *block* sebelumnya dan tahan terhadap modifikasi.

Berdasarkan jenis cakupan layanan, menurut Laurence (2017) ada 3 macam jenis *Blockchain* yaitu :

a. *Public Blockchain* merupakan sebuah jaringan terdistribusi dalam skala besar yang dijalankan menggunakan token asli (mata uang asli). Jenis *Blockchain* ini terbuka untuk siapa saja yang ingin berpartisipasi dan memiliki kode sumber terbuka (open source) yang dikelola oleh komunitas mereka. Sebagai contoh adalah *Bitcoin*.
b. *Permission Blockchain* adalah jenis *Blockchain* yang dapat mengelola siapa saja yang boleh menggunakannya di jaringan. Akan tetapi jenis *Blockchain* ini masih dalam skala

besar dan terdistribusi dengan menggunakan token asli, akan tetapi kode sumber tertutup untuk umum. Sebagai contonya adalah *Ripple*.

c. *Private Blockchain* adalah jenis *Blockchain* yang lebih kecil dan tidak menggunakan token. Seluruh pengguna di kontrol secara penuh dan tipe blockchain ini biasanya digunakan oleh kelompok untuk bertukar informasi secara internal.

Proses transaksi pada *Blockchain* dihubungkan antara satu *block* ke *block* lainnya. Untuk menjaga integritas setiap transaksinya, maka nilai *hash block* sebelumnya akan dimasukkan ke dalam *block* berikutnya. Sedangkan untuk nilai *hash* yang digunakan sudah pasti harus memenuhi persyaratan tertentu dengan tingkat kesulitan yang disepakati oleh konsensus agar dianggap transaksi tersebut sah. Proses pencarian nilai *hash* ini yang sering disebut dengan PoW (Proof on Work). Untuk itu proses percobaan pengubahan informasi akan mendapatkan tantangan kesulitan yang lebih karena harus mengubah blok-blok berikutnya (Hanifatunnisa Rifa, 2017).

2.1.4 *Smart Contract*

Smart Contracst adalah sebuah program komputer yang dapat menterjemahkan perjanjian antar bagian. *Smart Contracts* dapat bekerja pada jaringan terdesentralisasi (Garcia-alfaro,Navarro-arribas, Eds, & Hutchison, 2017). *Smart Contracts* sendiri dapat berjalan di *Blockchain* sebagai *Sidechain* dimana jika sesuatu terjadi di *Sidechain* maka tidak akan memberikan dampak pada *Blockchain* utama (Hegadekatti, 2017).

Kebanyakan *Blockchain* memiliki kemampuan untuk secara otomatis mengeksekusi *Smart Contracts* ketika kondisi tertentu terpenuhi. Sementara *platform* jaringan terpusat (tersentralisasi) juga dapat memberikan eksekusi otomatis, *Smart Contracts* pada dasarnya memberikan transparansi bagi semua pihak untuk melihat apa aturan otomasi. Sebaliknya, kode yang menggerakkan otomatisasi yang disediakan oleh *platform* terpusat biasanya tersembunyi di balik layar (ChainLink, 2018).

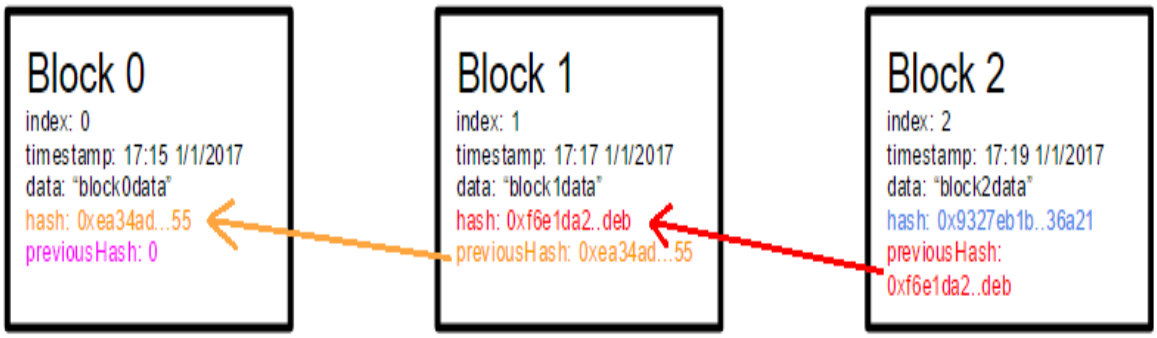
2.1.5 *Naivechain*

(Hartikka Lauri., 2018). *Naivechain* muncul sebagai solusi untuk memahami dan mengimpelementasikan *Blockchain* dengan mudah, karena *Blockchain* yang sudah ada seperti *Ethereum* dan *Hyperledger* cukup sulit untuk didalami dan diimplementasikan.

Naivechain hanya membutuhkan 200 baris kode dalam membentuk sebuah *Blockchain*. Berikut ini beberapa hal yang perlu diketahui mengenai *Naivechain:*

a. Struktur *Block*

Langkah pertama adalah menentukan struktur blok. Untuk membuatnya sesederhana mungkin, komponen yang paling diperlukan yaitu: indeks, stempel waktu, data, *hash*, dan *hash* sebelumnya.



Gambar 2.1 Model Dasar *Blockchain.*

b. Membuat *Block*

Untuk menghasilkan sebuah blok kita harus mengetahui *hash* dari blok sebelumnya dan membuat sisa konten yang dibutuhkan (indeks, *hash*, data dan stempel waktu). Data blok adalah sesuatu yang disediakan oleh pengguna akhir.

c. Menyimpan *Block*

Array Javascript dalam memori digunakan untuk menyimpan *Blockchain*. Blok pertama dari *Blockchain* selalu disebut "blok genesis", yang merupakan kode keras

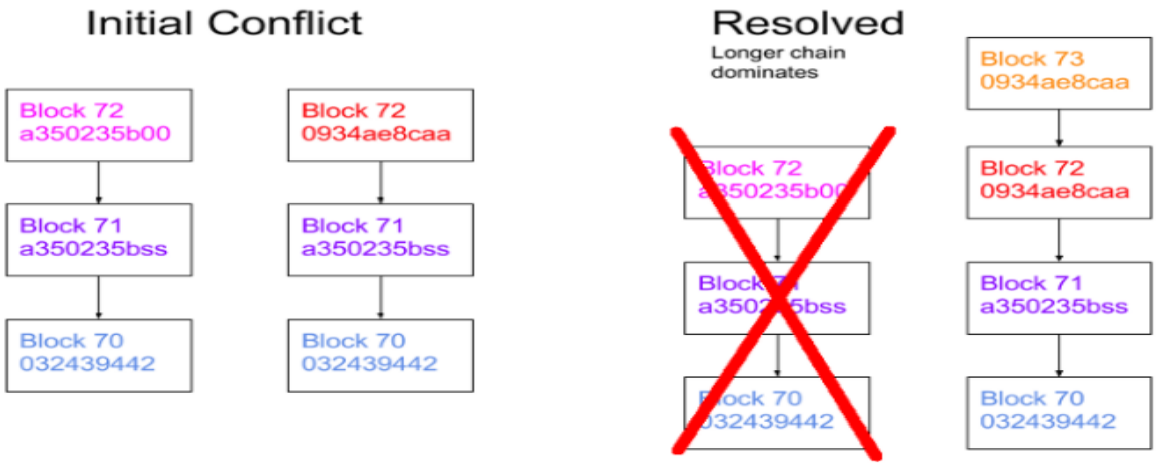
.

d. Validasi Integritas *Block*

Pada waktu tertentu kita harus dapat memvalidasi jika sebuah blok atau rangkaian blok valid dalam hal integritas. Ini benar terutama ketika kita menerima blok baru dari node lain dan harus memutuskan apakah akan menerimanya atau tidak.

e. Memilih Rantai Terpanjang

Hanya boleh ada satu kumpulan blok eksplisit dalam rantai pada waktu tertentu. Jika terjadi konflik (misalnya dua node sama-sama menghasilkan nomor blok 72) kita memilih rantai yang memiliki jumlah blok terpanjang.

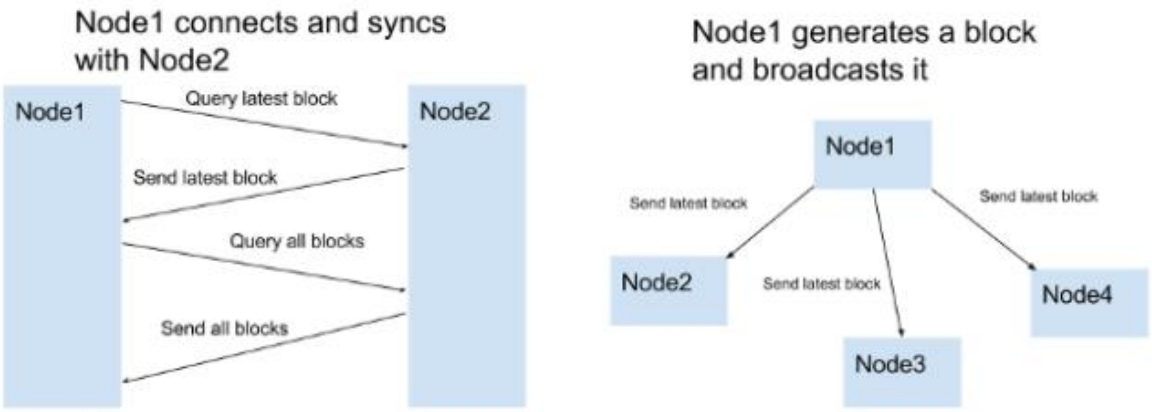


Gambar 2.2 Model penanganan Redudant *Block*.

f. Berkomunikasi dengan Node lain

Bagian penting dari sebuah node adalah membagikan dan menyinkronkan blockchain dengan node lain. Aturan berikut digunakan untuk menjaga jaringan tetap sinkron.

- Ketika sebuah node menghasilkan blok baru, ia menyiarkannya ke jaringan
- Saat sebuah node terhubung ke peer baru, node meminta blok terbaru
- Ketika sebuah node menemukan sebuah blok yang memiliki indeks lebih besar dari blok yang diketahui saat ini, ia akan menambahkan blok tersebut ke rantai saat ini atau menanyakan ke seluruh *Blockchain*.



Gambar 2.1 Model Komunikasi Antar Node.

Tidak ada integrasi antar *node* secara otomatis yang digunakan. Lokasi *node* lainnya harus ditambahkan secara manual.

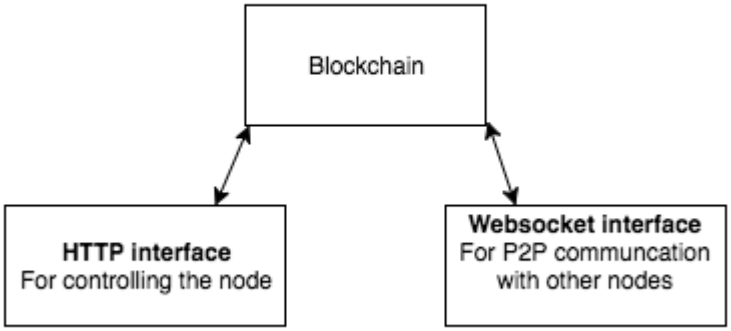
g. Mengontrol Node

Pengguna harus dapat mengontrol node dengan cara tertentu. Ini dilakukan dengan menyiapkan server HTTP. pengguna dapat berinteraksi dengan node dengan cara berikut:

- Buat daftar semua blok
- Buat blok baru dengan konten yang diberikan oleh pengguna
- Buat daftar atau tambahkan rekan

h. Arsitektur

Perlu dicatat bahwa node sebenarnya mengekspos dua server web: Satu untuk pengguna untuk mengontrol node (*server* HTTP) dan satu untuk komunikasi *peer-to-peer* antara node. (*Websocket* HTTP *server*).



Gambar 2.2 Komponen Utama *Naivechian.*

2.1.6 GetID3

GetID3 merupakan salah satu *plugin* berbasis PHP yang bisa digunakan untuk mengekstrak detail informasi dari suatu file. *Plugin* GetID3 telah diuji pada banyak sistem, pada banyak jenis *file*, pada banyak sistem operasi, dan secara umum diyakini stabil dan aman. Beberapa jenis atau tipe *file* yang sudah bisa diekstrak informasi tentang *file* tersebut antara lain di jelaskan pada tabel berikut.

Tabel 2.1 Tipe *file* yang mendukung GetID3

Image	Audio	Video	File
• BMP • GIF • JPEG • PNG • TIFF	**Audio Lossy** • MP3/MP2/MP1 • MPC / Musepack • Ogg (Vorbis, OggFLAC, Speex,	• ASF: ASF, Windows Media Audio (WMA), Windows Media Video	• ISO-9660 CD-ROM image (directory structure) • SZIP (limited support)

• SWF (Flash) • PhotoCD	Opus) • AAC / MP4 • AC3 • DTS • RealAudio • Speex • DSS • VQF **Audio Lossles** • AIFF • AU • Bonk • CD-audio (*.cda) • FLAC • LA (Lossless Audio) • LiteWave • LPAC • MIDI • Monkey's Audio • OptimFROG • RKAU • Shorten • Tom's lossless Audio • Kompressor (TAK) • TTA • VOC • WAV (RIFF) • WavPack	(WMV) • AVI (RIFF) • Flash • Matroska (MKV) • MPEG-1 / MPEG-2 • NSV (Nullsoft Streaming Video) • Quicktime (including MP4) • RealVideo	• ZIP (directory structure) • TAR • CUE

Tabel 2.1 menjelaskan bahwa beberapa tipe data bukti digital yang bisa diekstraksi secara detail menggunakan plugin GetID3. GetID3 telah mensupport sebagian besar tipe data yang sifatnya umum digunakan sehari-hari sehingga bisa digunakan untuk membantu ekstraksi bukti digital secara otomatis.

2.2. Penelitian Sebelumnya dan Kontribusi

Penelitian tentang pemanfaatan teknologi blockchain dalam mengelola bukti digital sudah bisa kita jumpai diberbagai jurnal online maupun offline. Ketika melakukan penelitian cara mudah untuk memahami penelitian-penelitan sebelumnya yaitu dengan bantuan tabel literatur review. Tabel review disajikan untuk mengetahui relevansi antar penelitian dan juga untuk mengetahui kontirbusi yang diberikan pada setiap penelitian..

Tabel 2.2 Studi Literatur

No	Peneliti	Metode	*Smart Contract*	*Type of Storage*	Hasil Penelitian
1	Lone., (2017)	*Forensic Chain*	*Single*	*No Classification*	Model *Forensic Chain* untuk menyimpan rekaman data *Chain of Custody* dari suatu bukti digital dengan menggunakan *Blockchain*.
2	Nizamuddin et al (2019)	*Document Version Control with Ethereum*	*Single*	*No Classification*	Pengelolaan dokumen menggunakan ethereum dan IPFS untuk melakukan *tracking* dari pengolahan suatu dokumen
3	Bonomi et al., (2018)	*Blockchain based Chain of Custody* (B-CoC).	*Single*	*No Classification*	Penerapan arsitektur *Blockchain based Chain of Custody* (B-CoC) dalam mengelola dan menyimpan bukti digital beserta dokumen *chain of custody*.
4	Eko Yunianto., (2019)	*Blockchain Digital Evidence Cabinet*. (B-	*Single*	*No Classification*	*Framework* dan implementasi *Blockchain* dalam mengelola barang bukti digital beserta *history Chain of Custody* dari bukti digital tersebut

		DEC).			
5	Lone (2019)	*Forensic Chain*	*Single*	*No Classification*	Penerapan konsep *Forensic Chain* pada *Blockchain Hyperledger Composer*.
6	Gopalan et al (2019)	*Blocchain Chain of Custody* BCOC	*Single*	*No Classification*	Penerapan konsep BCOC pada perangkat Andorid
7	Tian et al (2019)	Block-DEF	*Single*	*No Classification*	Framework yang digunakan dalam mengatasi masalah penyimpanan bukti digital yang berukuran besar.
8	Ahmad et al (2019)	Blockchain based Smart Lock	*Single*	*No Classification*	Penggunaan *smart lock* pada blockchain diharapkan mampu meningkatkan keamanan terhadap akses bukti digital.
9	Chopade Mrunali et al (2019)	Hash to Verify Transfering Digital Evidence	*Single*	*No Classification*	Penerapan base64 dalam verifikasi transfer bukti digital dari satu pengguna ke pengguna lainnya
10	Usulan	*Multi Smart Contract* BCOC	*Multi*	*Images, Video, Audio, Document, Others Files.*	Penerapam multi *smart contract* untuk meningkatkan integritas bukti digital dan chain of custody

Beberapa penelitian yang membahas mengenai pemanfaatan *Blockchain* dalam mengelola *digital chain of custody* telah berkembang tiga tahun terakhir ini. Dimulai dari penelitian yang dilakukan oleh Lone yang dipublikasi pada tahun 2017 mengajukan sebuah model yang disebut *forensic chain*. Adanya peningkatan pelanggaran integritas dan penolakan pada suatu bukti digital di persidangan, disebabkan oleh tidak jelasnya sumber, cara penanganan, dan cara mengamankan bukti digital yang melatarbelakangi Lone dalam mengajukan model tersebut. Model ini menggambarkan tentang cara *Blockchain* dengan *Smart Contract*-nya mampu mengelola dan mengamankan suatu rekaman aktivitas (C*hain of Custody*) dari suatu barang bukti digital dengan menyimpannya pada blok-blok yang terenkripsi. Model ini kemudian diimplementasikan pada tahun 2019 menggunakan *Blockchain Hyperledger Composer*. Menurut Lone implementasi model ini sudah bisa digunakan untuk menjaga integritas, keaslian, keamanan, dan kemampuan mengaudit dokumen chain of custody dari suatu bukti digital. Selain itu dari hasil pengujian performa, sistem ini overheadnya sudah bisa diterima dan bisa diimplementasikan di dunia nyata. Namun Lone masih menggunakan skema standar untuk mencatat aktivitas dari pengelolaan bukti digital. Tidak ada acuan penelitian lainnya dalam membuat instrumen chain of custody untuk smart contractnya. Smart contract yang digunakan untuk menyimpan dokumen C*hain of Custody* masih menggunakan satu format untuk semua bukti digital.

Perubahan yang terjadi pada suatu dokumen sering kali tidak terdokumentasi dengan baik, sehingga mengakibatkan pelacakan terhadap perubahan dokumen tersebut sulit dilakukan. Pemanfaatan *ethereum* dan IPFS diharapkan mampu melakukan tracking terhadap perubahan suatu dokumen.

Penelitian berikutnya dilakukan juga oleh Bonomi et al., (2018). Banyaknya investigator ataupun ahli yang ikut berpartisipasi dalam menganalisa bukti digital dan cara mempertanggungjawabkan bukti digital agar diterima di pengadilan yang melatarbelakangi penelitian oleh Bonomi tentang pemanfaatan *Blockchain Ethereum* dalam mengelola dokumen C*hain of Custody*. *Ethereum* digunakan sebagai *private* atau *permissioned Blockchain*, sehingga hanya pengguna dengan otoritas tertentu yang mampu mengakses dokumen C*hain of Custody*. Penggunaan ethereum dalam mengelola dokumen C*hain of Custody* dianggap sudah cukup efektif karena mampu mempertahankan beban kerja yang realistis dengan *overhead* yang dapat diterima. Akan tetapi dalam penelitian Bonomi, beberapa investigator belum bisa bekerja sama untuk menganalisa satu bukti digital. Tidak

seperti pada penelitian Lone yang sudah mampu memberikan fasilitas beberapa investigator bisa menganalisa satu bukti digital yang sama.

Menurut Eko Yunianto., (2019), barang bukti digital tidak bisa digunakan didalam persidangan karena masih digunakannya dokumen fisik untuk C*hain of Custody* bukti digital dan tidak kompetennya investigator. Pemanfaatan *framework Digital Evidence Cabinet* (DEC) dalam mengelola dan menyimpan dokumen C*hain of Custody* beserta bukti digitalnya (Prayudi et al., 2014) dikombinasikan dengan teknologi *Blockchain Ethereum,* diharapkan mampu memberikan integritas, keaslian, keutuhan terhadap barang bukti digital, sehingga mampu dipertanggungjawabkan dalam persidangan. Dari penelitian yang dilakukan Eko menghasilkan *framework* baru yaitu *Blockchain Digital Evidence Cabinet* (B-DEC) yang mampu mengelola dan menyimpan dokumen *Chain of Custody* dan bukti digital dalam blok yang terenkripsi. Penelitian ini sudah mampu memberikan hak akses kepada beberapa investigator terhadap suatu barang bukti seperti penelitian yang telah dilakukan oleh Lone.

Kekhawatiran terhadap rusaknya bukti digital dan tidak diterimanya bukti digital di persidangan melatar belakangi munculnya blockchain berbasis andorid. Pembuatan aplikasi blockchain berbasis android menjadi solusi dalam menjaga keamanan bukti digital. Fitur yang ditawarkan dari aplikasi android ini berupa aplikasi yang bisa mencatat pengguna yang telah mengkases bukti digital. (Gopalan:2019)

Ukuran bukti digital yang bisa saja sangat besar, dan bukti digital yang terus bertambah mengurangi performa dalam menyimpan bukti digital. Model block-DEF menawarkan solusi untuk mengatasi masalah ukuran penyimpanan bukti digital. Cara kerjanya yaitu dengan memisahkan bukti digital dengan informasi metadata ketika dilakukan penyimpanan bukti digital. Hasil model block-DEF menawarkan solusi framework yang sifatnya skalabilitas. Selain itu model block-DEF menawarkan solusi untuk meningkatkan integritas dan validasi bukti digital. (Tian Zhihong:2019)

Penolakan bukti digital bisa saja terjadi dipersidangan ketika bukti digital tersebut keamanan bukti digital yang lemah: integritas bukti digital yang kurang, hilangnya kepercayaan terhadap bukti digital. Penggunaan *smart lock* pada bukti digital menjadi solusi dalam mengamankan bukti digital.(Ahmad Liza:2019)

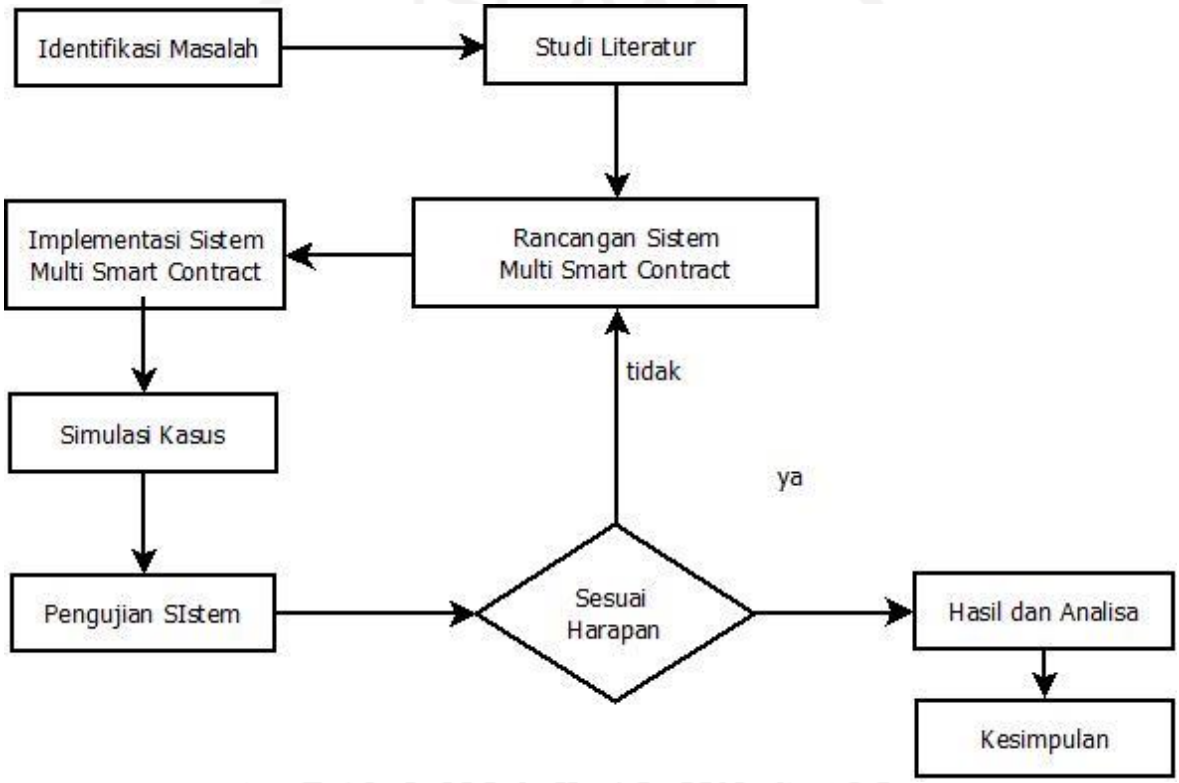
Adanya aktifitas bukti digital dari satu user ke user lainnya mengakibatkan rentannya adanya perubahan yang dilakukan terhadap bukti digital. Solusi melakukan enkripsi data ketika melakukan transfer bukti digital menjadi pilihan untuk mengamankan

bukti digital. Aplikasi yang dibangun mampu melakukan enkripsi bukti digital ketika dikrimkan dari satu pengguna ke pengguna lainnya. (Chopade Mrunali :2019).

Penelitian yang ada sudah mampu memanfaatkan suatu *Smart Contract* dalam mengelola dan menyimpan suatu *Chain of Custody* bukti digital pada blok-blok yang ternkripsi secara baik. Akan tetapi belum ada penelitian mengidentifikasi *Chain of Custody* bukti digital dalam format S*mart Contract* yang berbeda-beda dan bisa disesuaikan dengan kebutuhan. Oleh karena itu dalam penelitian ini, saya akan menyajikan model *Multi Smart Contract* yang bisa mengelola bukti digital yang memiliki tipe dengan properti/*tag* yang berbeda-beda

BAB III
METODE PENELITIAN

Dalam melakukan penelitian diperlukan suatu kerangka penelitian yang sistematis untuk menjadi acuan langkah-langkah penelitian secara terstruktur agar mempermudah penulis dalam melakukan proses penelitian yang dimulai dari identifikasi masalah hingga penyusunan laporan.



Gambar 3.1 Alur Penelitian.

.

Pada Gambar 3.1 dijelaskan tentang tahapan penelitian yang akan dilakukan mulai dari identifikasi masalah, studi literatur, rancangan sistem *multi smart contract,* implementasi *Multi Smart Contract,* simulasi kasus, pengujian sistem. Jjika belum sesuai harapan kembali untuk memperbaiki rancangan sistem *multi smart contract* dan jika sudah sesuai harapan maka lanjut ke tahapan membuat hasil dan analisa dari penerapan *Multi Smart Contract* pada *naive chain* untuk meningkatkan integritas bukti digital dan *Chain of Custody*.

3.1. Identifikasi Masalah

Identifikasi masalah dilakukan dengan cara mengenali dan menandai masalah pengelolaan dokumen *Chain of Custody* pada suatu bukti digital yang dialami oleh aparatur penegak hukum khususnya kasus kejahatan siber. Selain itu peneliti juga mengidentifikasi masalah yang ditemukan pada penelitian-penelitian sebelumnya. Dengan melakukan identifikasi masalah, peneliti berharap bisa mengetahui bahwa penelitian tentang penerapan *Multi Smart Contract* dalam meningkatkan integritas bukti digital dan *Chain of Custody* memang dibutuhkan.

3.2. Studi Literatur

Pengumpulan literatur yang berkaitan dengan entitas terlibat dalam pengelolaan bukti digital, tema *Chain of Custody, Blockchain*, jenis-jenis bukti digital dan desain model *Multi Smart Contract* dalam mengelola bukti digital dan *Chain of Custody* diperlukan dalam membantu menyusun penelitian ini. Dari literatur yang diperoleh kemudian dipetakan menjadi beberapa bagian mencari poin penting dari masing-masing penelitian. Dari poin penting tersebut, kemudian penulis bisa mencari posisi dalam kontribusi pada penelitian serupa.

3.3. Rancangan Sistem *Multi Smart Contract*

3.3.1 Analisa Sistem

Sebelum merancang sistem *multi smart* pertama harus kebutuhan sistem harus didefinisikan terlebih dahulu. Kebutuhan Sistem tersebut antara lain :

a. Analisa Otoritas Pengguna Sistem

   Penentuan otoritas terhadap pengguna sistem diperlukan untuk mengamankan sistem dan membatasi akses ke dalam sistem. Dengan otoritas pengguna setiap aktifitas yang dilakukan terhadap bukti digital dapat tercatat ke dalam block. Otoritas pengguna yang dimaksud antara lain seperti first responder, investigator dari kepolisian, pengacara, hakim, dan pihak berwenang lainnya.

b. Analisa Kebutuhan Input Sistem

   Kebutuhan input sistem yang diperlukan antara lain :

   - Input form isian data profil pengguna dan input otoritas (hak akses pengguna)
   - Input file bukti digital dan form isian chain of custody
   - Input form isian tambahan informasi tentang bukti digital

- Input dalam bentuk form isian tentang aktifitas yang dilakukan oleh pihak berwenang terhadap bukti digital.

c. Kebutuhan Proses Sistem

Proses yang akan dilakukan dalam meningkatkan integritas bukti digital menggunakan *multi smart contract* pada *naive chain* antara lain :

- Proses pembuatan akun dan otoritas pengguna
- Proses input data kasus dan *chain of custody*
- Proses unggah bukti digital dan ekstraksi informasi metadata bukti digital dengan bantuan GetID3
- Proses menampilkan bukti digital, chain of custody, dan metadata bukti digital
- Proses unduh bukti digital dan chain of custody
- Proses pemutakhiran dan penghapusan bukti digital
- Proses penyimpanan log aktifitas yang dilakukan terhadap bukti digital yang dimulai dari pembuatan kasus, penyimpanan bukti digital hingga penghapusan bukti digital ke dalam blok.
- Proses menampilkan log aktifitas.

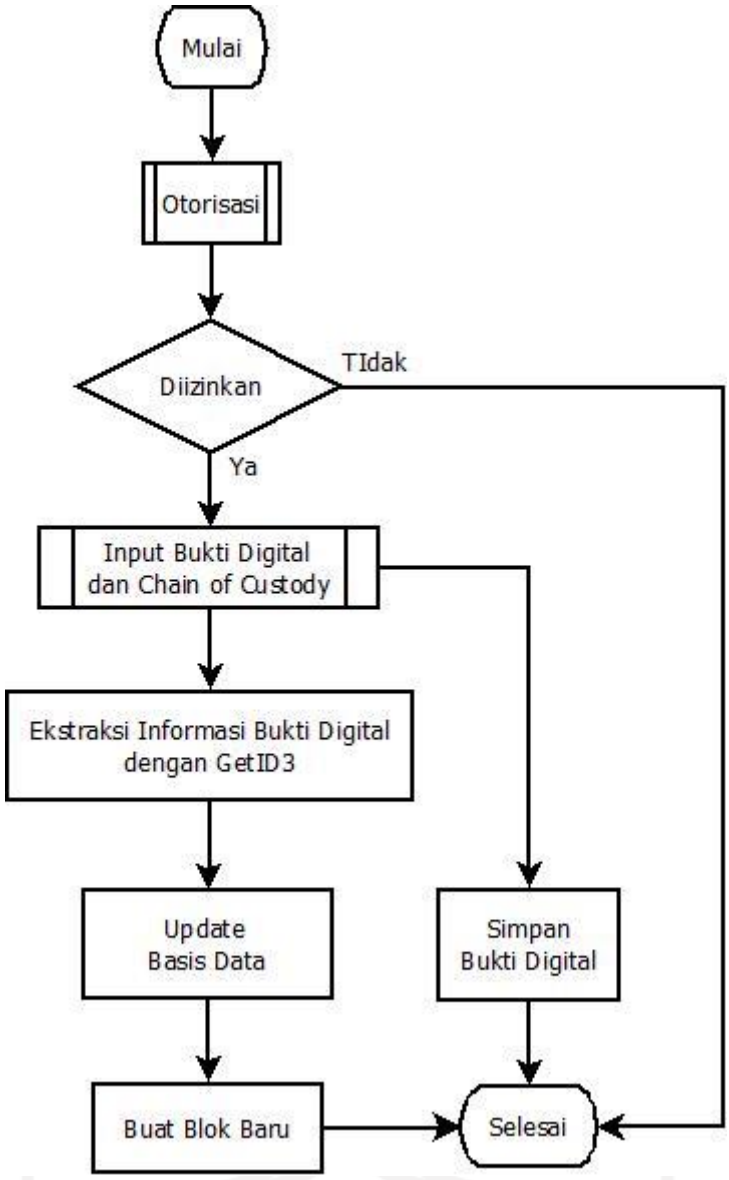
d. Analisa Kebutuhan Output Sistem

Dari rangkaian proses penggunaan *multi smart contract* dalam mengelola bukti digital dan *chain of custody* dihasilkan beberapa output antara lain :

- File Bukti digital dan dokumen chain of custody tersimpan dalam sistem dan bisa diakses dan diunduh oleh petugas yang memiliki otoritas.
- Semua aktifitas yang dilakukan terhadap bukti digital dan *chain of custody* tercatat dalam blok dan bisa dilihat oleh petugas yang berwenang.

3.3.2 Alur Penyimpanan Bukti Digital dan Chain of Custody pada Naive Chain

Dalam rancangan *multi smart contract*, data pengguna dan chain of custody disimpan dalam basis data. Kemudian data bukti digital akan dismpan pada server naive chain. Sedangkan untuk menjaga integritas data log aktifitas terhadap bukti digital beserta chain of custody disimpan dalam blok *naive chain*.

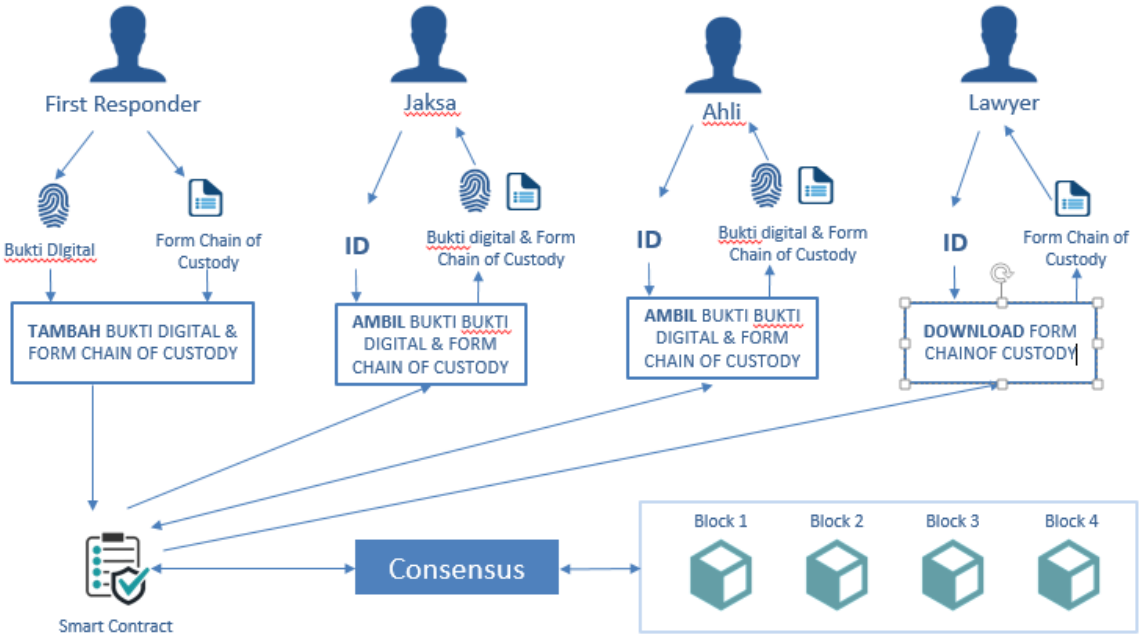


Gambar 3.2 Alur Penyimpanan Bukti Digital dan Chain of Custody Pada Naive Chain

Pertama pengguna sistem harus login berdasarkan akses yang telah diberikan, Kemudian pengguna dapat melakukan penambahan kasus, penambahan bukti digital, dan input form chain of custody. Kemudian ketika melakukan penyimpanan bukti digital, akan dilakukan eksraksi informasi tentang bukti digital tersebut dibantu dengan plugin GetID3. Data tentang kasus, chain of custody, dan informasi bukti digital akan disimpan ke dalam basis data. Kemudian rekaman data dari basis data akan disimpan ke dalam *naive chain* beserta log aktifitas yang dilakukan terhadap bukti digital. Sedangkan File bukti digital akan disimpan ke dalam server sistem naive chain.

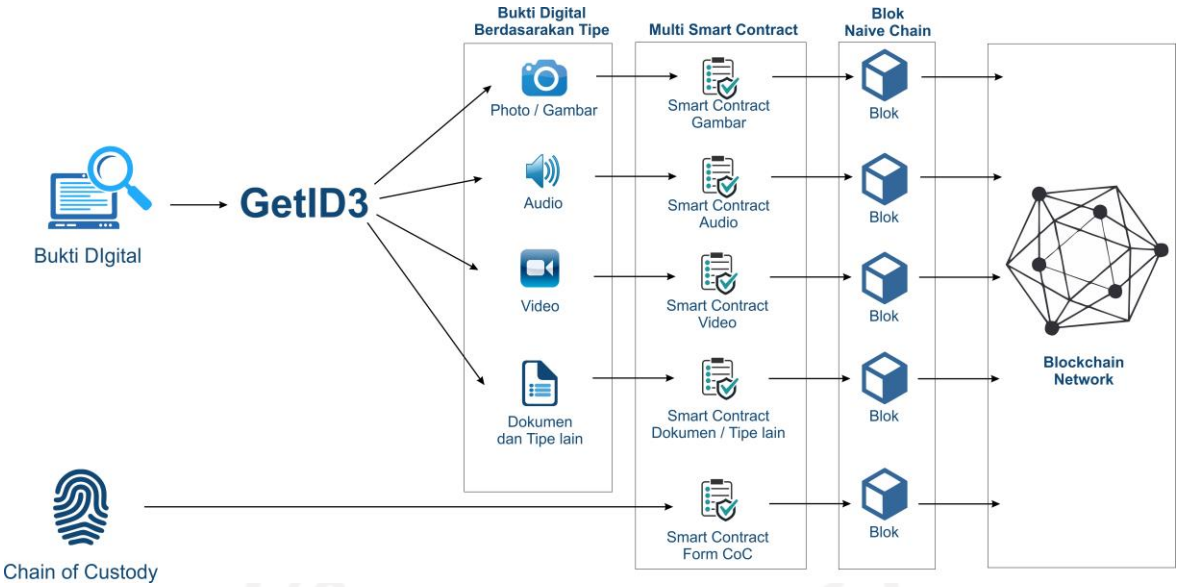
3.3.3 Rancangan Multi Smart Contract pada Naive Chain

Alur penggunaan *Blockchain* dalam mengelola bukti digital dan *Chain of Custody* dimulai dari *First responder* menginputkan bukti digital dan mengisi *form Chain of Custody* ke dalam *Blockchain* dengan *Multi Smart Contract*. Kemudian ketika bukti digital akan digunakan untuk keperluan investigasi, maka jaksa dan ahli diberi izin untuk mengakses bukti digital. Semua aksi dilakukan jaksa dan ahli, dan semua hasil temuan yang diperoleh dari hasil investigasi kemudian dicatat dalam *form Chain of Custody*. Selanjutnya bukti digital dan hasil analisanya akan disajikan dalam persidangan.



Gambar 3.3 Alur Akses Bukti Digital dan *Chain of Custody* pada *Blockchain.*

Smart Contract yang akan digunakan dalam mengelola bukti digital dan *Chain of Custody* akan dipisahkan berdasarkan jenis bukti digitalnya. Pembagian *Smart Contract* akan dibagi menjadi gambar, audio, video dan dokumen atau jenis file lainnya.



Gambar 3.4 Desain *Multi Smart Contract* untuk Bukti Digital dan *Chain of Custody*.

Dalam rangka mendapatkan informasi yang lengkap tentang suatu bukti digitial, GetID3 membantu sistem mengekstraksi informasi dari bukti digital yang diunggah secara otomatis. Hasil ekstraksi infomrasi tersebut akan dikelompokkan berdasarkan jenis bukti digital disimpan ke dalam basis data dan juga disimpan dalam naive chain dengan bantuan *multi smart contract*. Selain itu, isian form chain of custody dari bukti digital tersebut juga akan disimpan pada basis data dan naive chain dengan bantuan *multi smart contract*.

Rancangan *Chain of Custody* yang digunakan dalam penelitian ini mengacu pada penelitian sebelumnya yaitu penelitian tentang *Blockchain Digital Evidence Chain of Custody* (Yunianto., 2019) dan ditambahkan dengan informasi detail bukti digital sesuai tipe atau jenisnya.

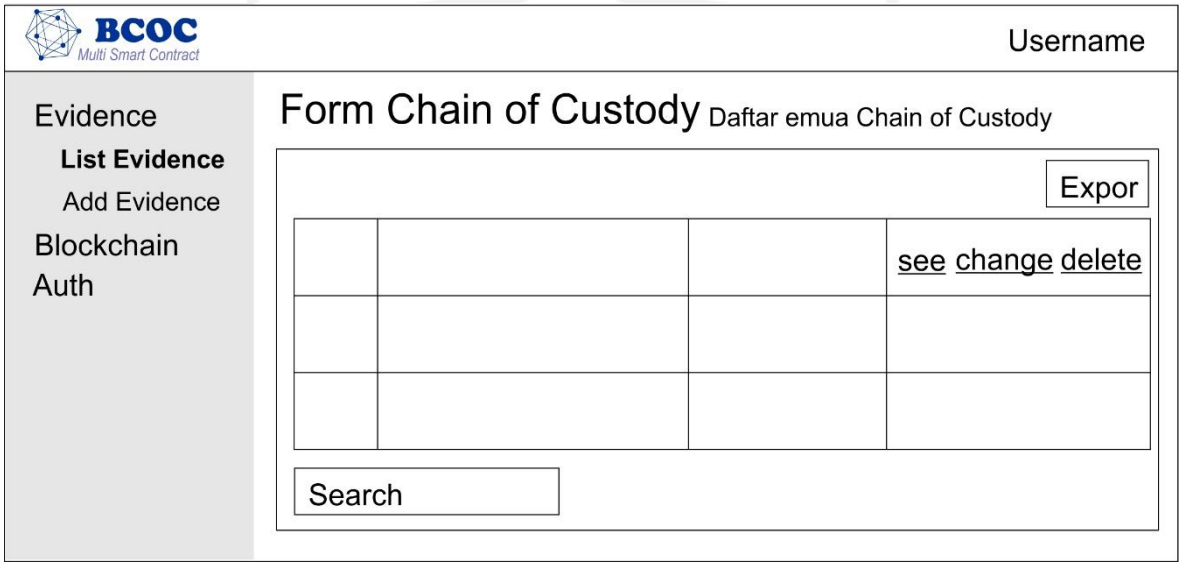
3.3.4 Rancangan Antar Muka

Pengelolaan bukti digital dan *Chain of Custody* menggunakan *Blockchain*, dibantu menggunakan *frontend* yang berbasis *website*. Pemilihan *website* didasarkan karena bukti digital dan *Chain of Custody* akan diakses oleh beberapa pihak yang melakukan penanganan bukti digital sehingga jika menggunakan website, aplikasi akan bersifat terpusat dan tidak perlu instalasi aplikasi pada perangkat masing-masing pengguna.

Sebelum melakukan proses pengelolaan bukti digital dan *Chain of Custody*, pengguna (*First responder*, jaksa/investogator, ahli, dan lawyer) harus memasukkan *username* dan *password* sesuai dengan ketentuan yang sudah di konfigurasi oleh admin.

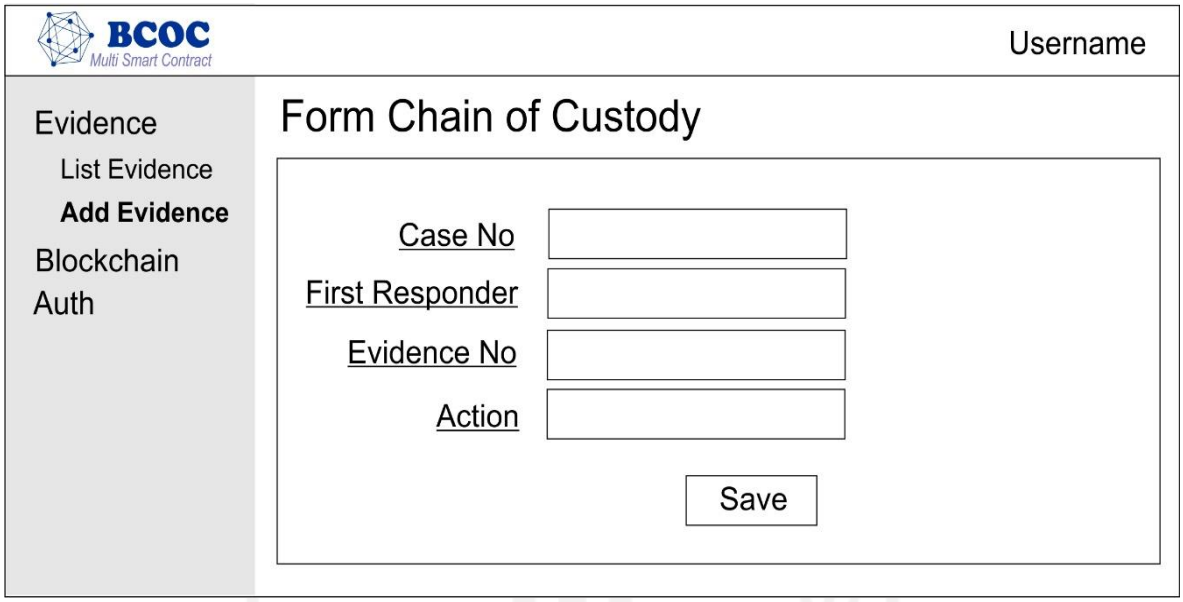
Gambar 3.5 Rancangan Halaman Login.

Setelah pengguna berhasil melakukan *login* maka selanjutnya akan diarahkan ke halaman Daftar Bukti Digital sesuai hak akses masing-masing pengguna. Bagi pengguna yang memiliki hak akses atau otoritas untuk dapat mengunduh bukti digital, bisa diunduh melalui halaman ini. Tampilan halaman Daftar Bukti Digital dapat dilihat pada **Gambar** berikut:



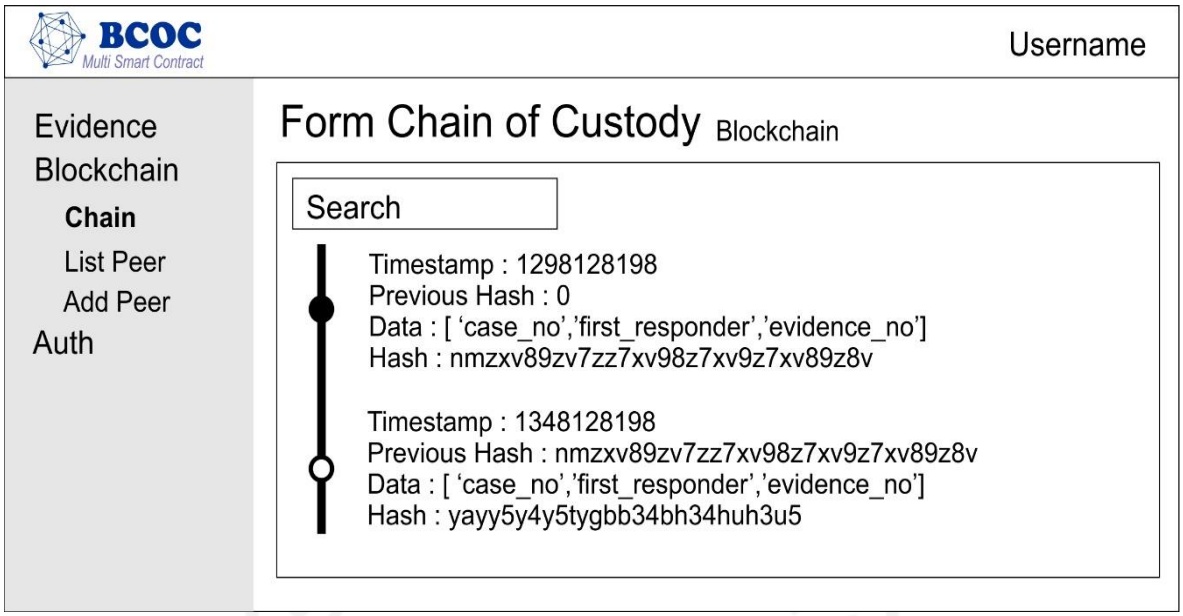
Gambar 3.6 Rancangan Halaman *List Evidence*.

Pengguna yang bisa menambahkan daftar bukti digital dan *form Chain of Custody* adalah *first responder*. Bukti digital beserta *file properties*-nya serta form *Chain of Custody* akan disimpan pada *database* dan juga pada *Blockchain*. Begitu juga dengan hasil aksi *update* bukti digital dan *update form Chain of Custody* akan disimpan ke dalam *Blockchain*. Ketika dilakukan penghapusan bukti digital beserta *form Chain of Custody*, maka *log*-nya akan disimpan ke dalam *Blockchain*, sehingga ketika dibutuhkan penelusuran semua aksi tersebut tercatat didalam *Blockchain*.



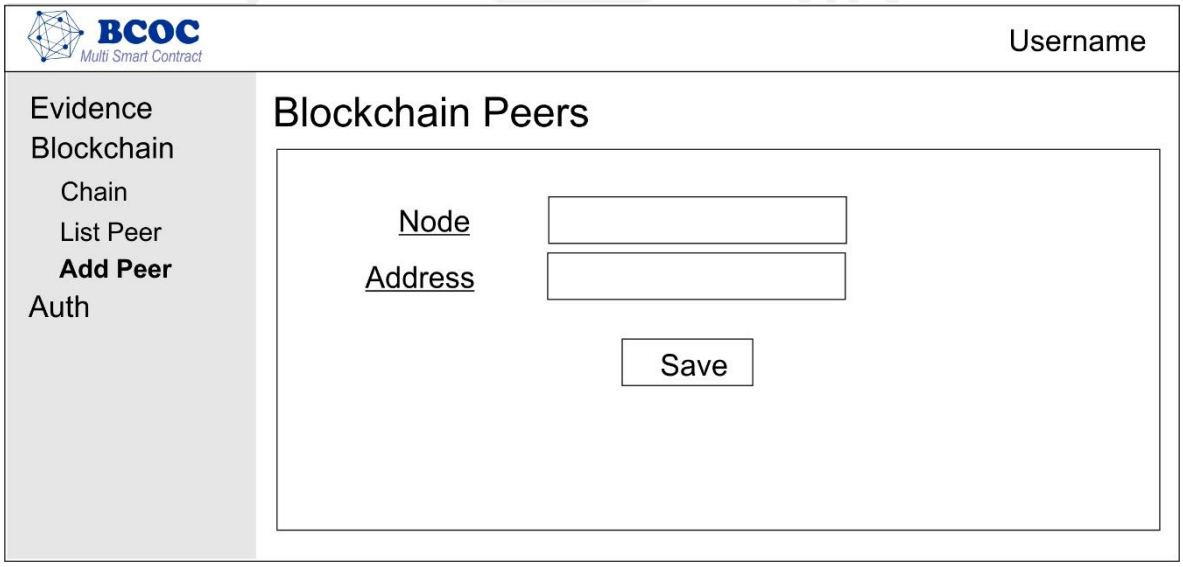
Gambar 3.7 Rancangan Halaman *Add Evidence*.

Data *Blockchain* hasil proses penyimpanan, pengubahan, dan penghapusan bukti digital beserta *Chain of Custody* bisa diakses oleh *First Responder* pada menu *Chain*. Data *Blockchain* ini akan terus bertambah dan tidak bisa dihapus untuk menjaga otentifikasi data bukti digital.



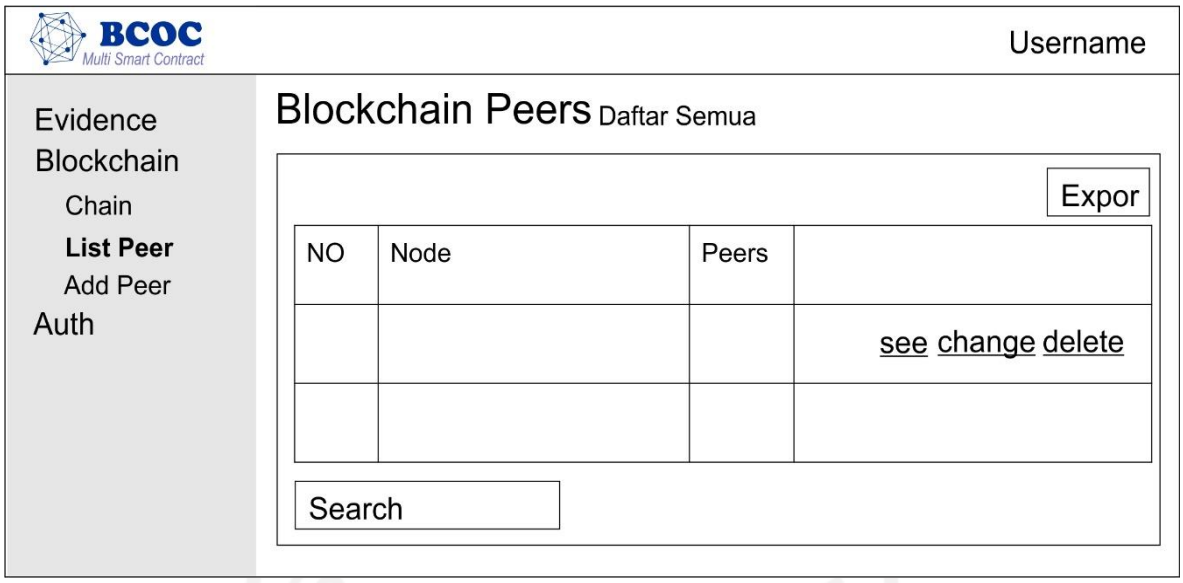
Gambar 3.8 Rancangan Halaman *Chain.*

Selain itu untuk menjaga keamanan data, maka data *Blockchain* ini bisa disimpan pada *server* atau *node* lainnya yang ada pada jaringan yang berbeda. Penambahan *node* ini biasa disebut dengan *peers*. Penambahan *peer* ini bisa dilakukan melalui menu *Add Peer*.



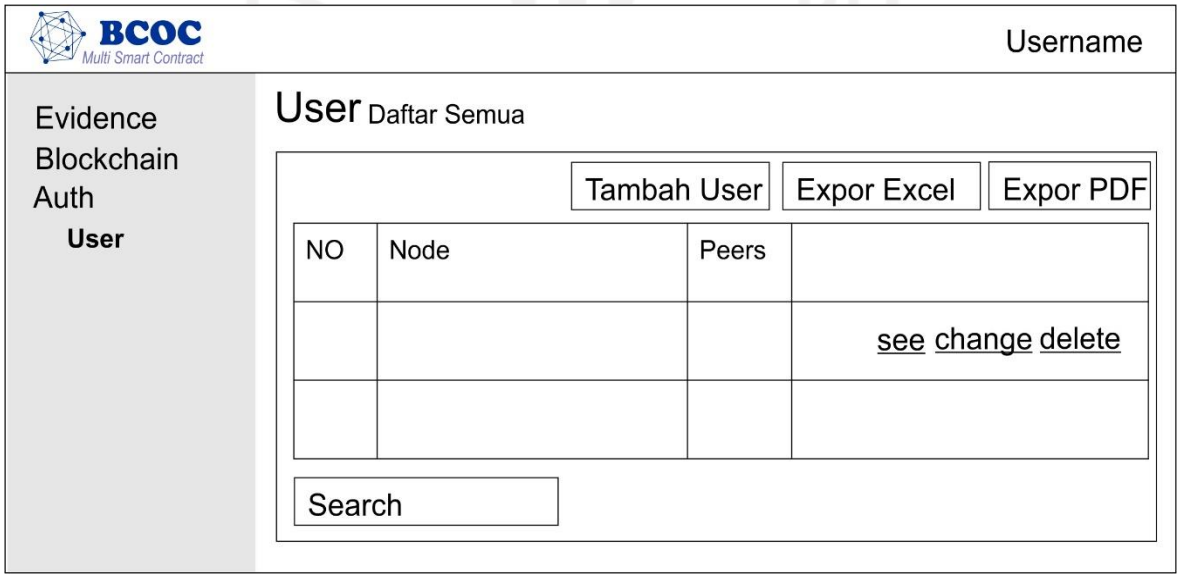
Gambar 3.9 Rancangan Halaman *Add Peer*.

Daftar *peer* yang berhasil ditambahkan bisa dilihat pada menu *List Peer*. Pengubahan dan *update peer* menunya juga bisa ditemukan pada halaman *List peer*.



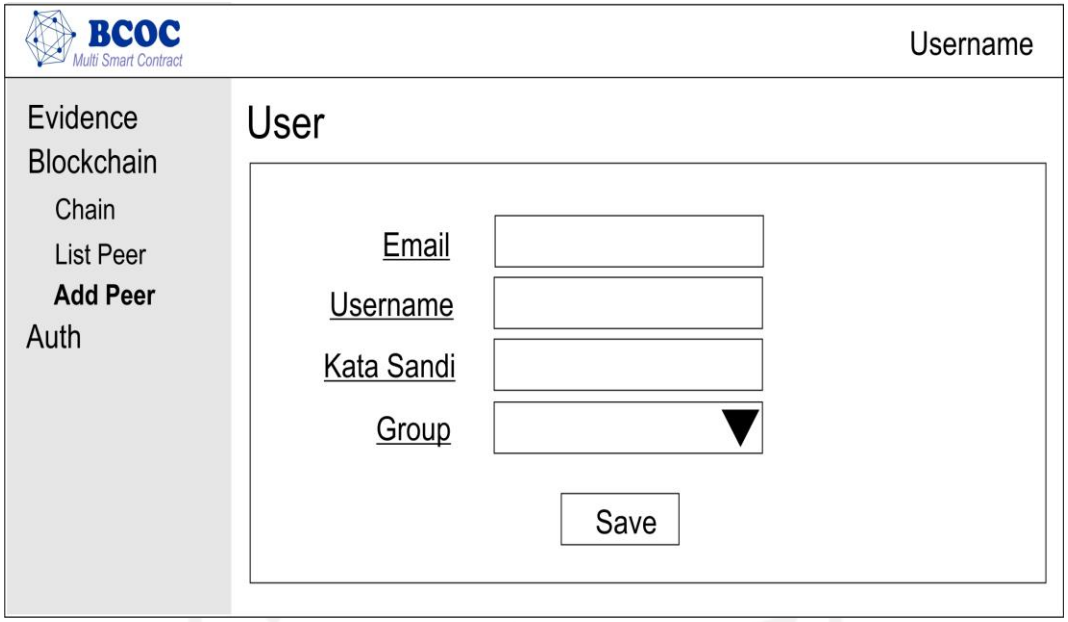
Gambar 3.10 Rancangan Halaman *List Peer*.

Menu *List User* digunakan untuk melihat data pengguna yang terlibat didalam pengelolaan bukti digital dan *Chain of Custody*. Disini akan ditemukan menu untuk melakukan pengubahan dan penghapusan data pengguna. Data *user* ini tidak disimpan dalam *Blockhain* namun disimpan dalam *database* MySQL..



Gambar 3.11 Rancangan Tampilan Halaman *User*.

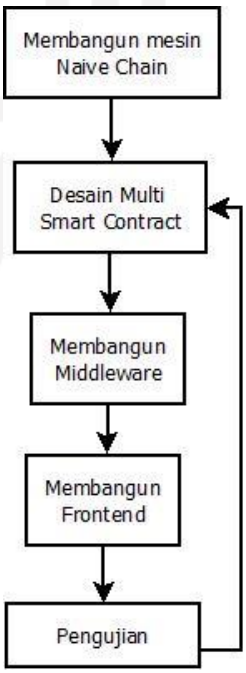
Menu *Add User* digunakan untuk menambahkan daftar pengguna yang terlibat dan mempunyai otoritas dalam pengelolaan bukti digital dan *Chain of Custody*.



Gambar 3.12 Rancangan Halaman *Add Peer*.

3.4. Implementasi Sistem *Multi Smart Contract*

Tahapan yang dilakukan dalam membangun sistem *multi smart contract* pada *naive chain* untuk meningkatkan integritas bukti digital dan chain of custody antara lain dimulai dari membangun mesin untuk *naive chain*, membangun *multi smart contract*, membangun *middleware,* membangun *fornt end*, dan integrasi sistem.



Gambar 3.13 Alur Implementasi Multi Smart Contract

3.4.1 Membangun Naive Chain

Naive chain akan dibangun dengan spesifikasi sebagai berikut :

a. VPS 2 Core, RAM 4 GB, HDD 100 GB.

b. Sistem operasi Ubuntu 18.4

c. *Apache* dan PHP sebagai *webserver* untuk aplikasi GUI, dan konfigurasi *Smart Contract.*

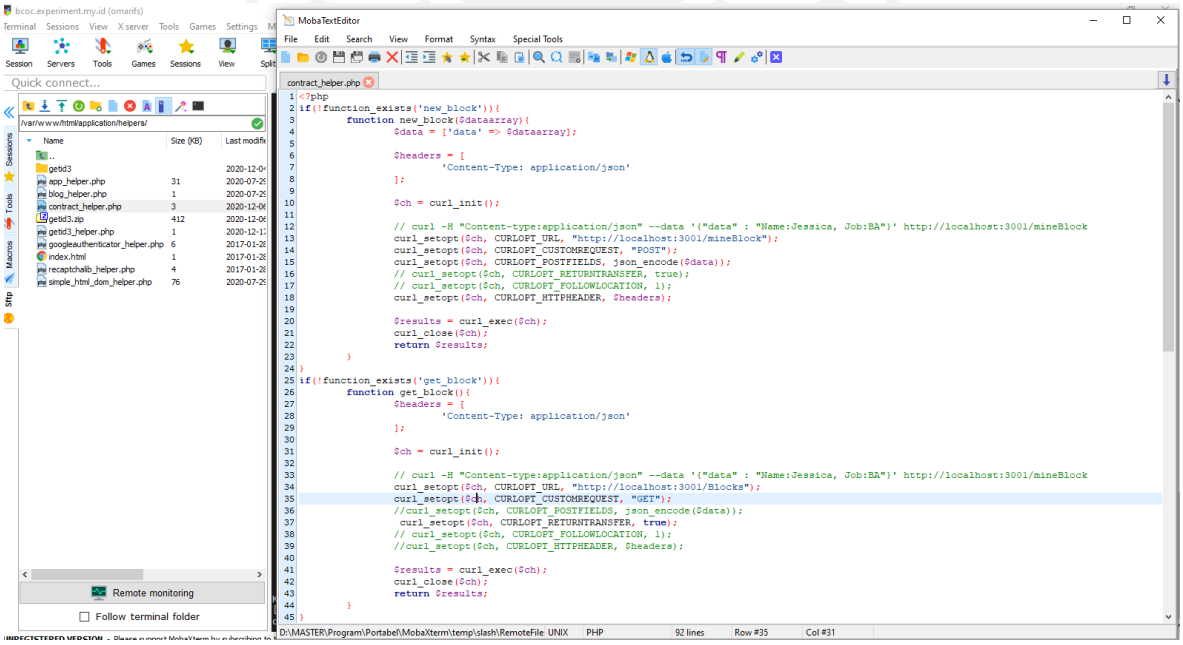
d. *MySQL* untuk menyimpan data user, kasus dan *chain of custody*.

e. Node JS dan Docker untuk menyimpan data *Blockchain*.

f. *Composer* untuk menjalankan *Blockchain*.

3.4.2 Membangun Multi Smart Contract

Multi smart contract dibangun menggunakan bahasa pemerograman PHP. Impelemntasinya diwujudkan dalam bentuk suatu fungsi helper, sehingga penggunaannya bisa disesuaikan sesuai kebutuhan.



Gambar 3.14 Source Kode Helper Smart Contract

3.4.3 Membangun *Middleware*

Dalam menghubungkan antara *naive chain* dengan smart contract dan *frontend,* naive chain menyediakan API yang diwujudkan dalam bentuk url dan bisa dipanggil menggunakan perintah curl. Contohnya untuk memanggil semua data blok API urlnya " http://localhost:3001/blocks", membuat blok url apinya

“http://localhost:3001/mineBlock”. Dengan metode API seperti ini diharapkan naive chain bisa digunakan pada banyak platform pemrograman.

3.4.4 Membangun *Front End*

Pembuatan *frontend* tidak terbatas pada bahasa pemrograman tertentu. Namud *frontend* bisa dibangun dengan banyak bahasa pemrograman karena *naive chain* sudah memudahkan aksesnya menggunakan API url. Penggunaan website berbasis PHP menjadi pilihan dalam membangun *forntend* karena sistem ini berbasis server site dan bersifat multi platform.

3.5. Pengujian Sistem

3.5.1 Pengujian Kerja Sistem

Pengujian kerja sistem diperlukan untuk memastikan sistem berjalan sesuai alur yang dimulai dari memastikan naive chain berjalan sesuai rancangan, *multi smart contract* berjalan sesuai rancangan, dan *frontend* berjalan sesuai rancangan. Selain itu juga perlu dipastikan bahwa alur kerja sistem mulai dari input bukti digital dan input form digital chain of custody hingga berhasil disimpan dalam naive chain bisa berjalan sesuai rancangan. Pengujian kinerja sistem dilakukan dengan black box testing dan white box testing.

Tabel 3.1 Rancangan Pengujian Kerja Sistem

No	Skema Pengujian	Otoritas Pengguna			
		First Respon	Jaksa	Ahli	Lawyer

3.5.2 Pengujian implementasi *Multi Smart Contract*

Pengujian implementasi *multi smart contract* dengan cara memastikan fungsi smart contract sesuai dengan fungsi smart contract yang ada pada penelitian sebelumnya. Fungsi yang dimaksud yaitu rancangan alur sistem multi smart contract sudah sama dengan rancangan smart contract sebelumnya.

Tabel 3.2 Rancangan Pengujian Implementasi Sistem

No	Alur / Proses	Terpenuhi	Keterangan

3.5.3 Pengujian Integritas Bukti Digital dan *Chain of Custody*

Dengan *multi smart contract,* diharapkan dalam penyimpanan bukti digital dapat membantu meningkatkan integirtas bukti digital. Integritas bukti digital yang diharapkan bisa tercapai jika informasi tentang suatu bukti digital bisa disajikan secara lebih detail.

3.5.4 Pengujian Performa *Multi Smart Contract*

Pengujian performa *multi smart contract* meliputi pengujian kemampuan meningkatkan integritas bukti digital, pengujian penyimpanan bukti digital, dan pengujian kecepatan akses *multi smart contract*. Dalam meningkatkan integritas bukti digital dapat dilakukan dengan cara berusaha menyajikan informasi yang lengkap tentang bukti digital. Peningkatan performa penyimpanan bukti digital ke dalam sistem dapat dilakukan dengan cara hanya menyimpan bukti digital yang hanya digunakan dalam pembuktian dipersidangan. Sedangkan penggunaan naive chain dilakukan untuk mencapai optimalisasi kecepatan akses *multi smart contract*.

Tabel 3.3 Rancangan Pengujian Performa *Multi Smart Contract*

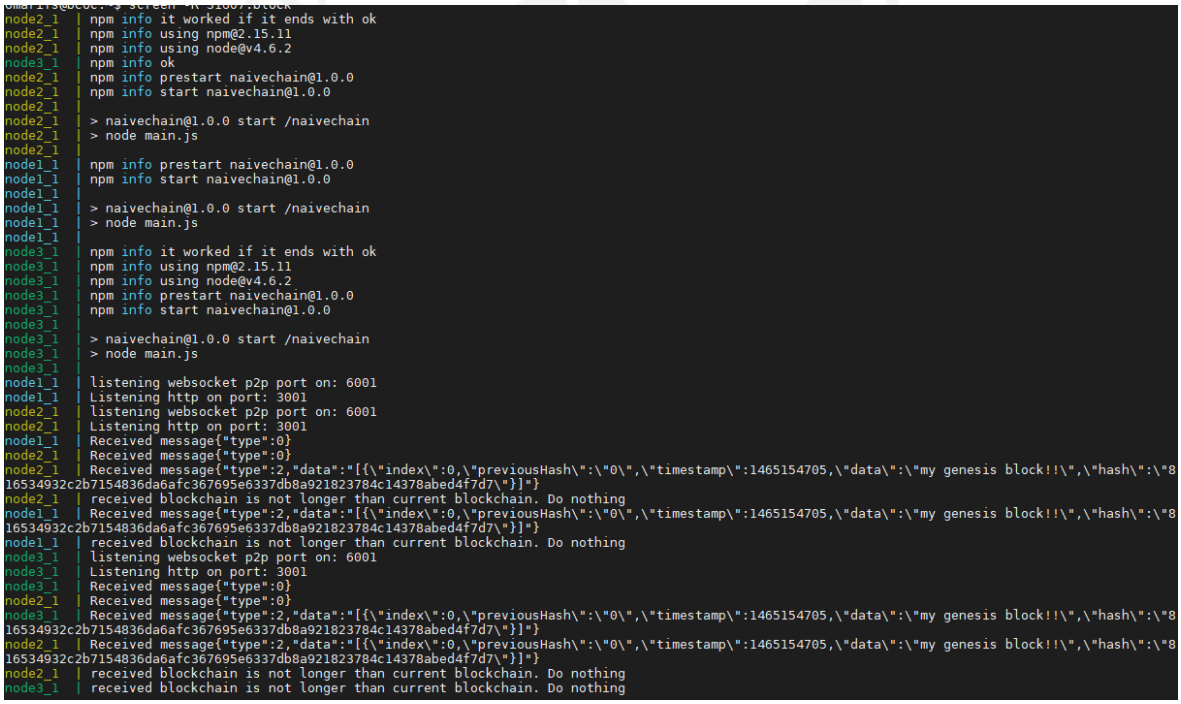
No	Skema Pengujian	Jumlah Komponen yang Disimpan	Ukuran File	Kecepatan Akses

BAB IV
HASIL DAN PEMBAHASAN

4.1. Implementasi Sistem Multi Smart Contract

4.1.1 Membangun Naive Chain

Langkah membangun naive chain dimulai dari menyiapkan sistem operasi, menginstall docker, composer, dan nodejs. Kemudian mengunduh file master naive chain di https://github.com/lhartikk/naivechain . Selanjutnya menjalankan naive chain pada server dengan perintah “docker-compose up” .Jika naive chain berjalan dengan baik maka tampilannya seperti Gambar 4.1.



Gambar 4.1 Menjalankan Naive chain

Sebelum naive chain digunakan untuk transaksi data,maka perlu dilakukuan inisiasi blok 0, yang biasa disebut file genesis. File genesis pada naive chain dibuat dengan kode berikut :

```
var getGenesisBlock = () => {
    return new Block(0, "0", 1465154705, "my genesis block!!",
"816534932c2b7154836da6afc367695e6337db8a921823784c14378abed4f7d7");
};
```

4.1.2 Membangun *Multi Smart Contract*

Sebelum membangun *multi smart contract* hal yang perlu dilakukan yaitu menginstall webserver apache, PHP dan database server MySQL yang nanti digunakan dalam mengelola *smart contract*. Smart contract akan dijadikan sebagai helper pada framework Codeigniter. Helper smart contract ini juga akan bekerja sama dengan fungsi dari library GetID3 untuk mengekstraksi informasi bukti digital yang akan disertakan bersama chain of custody untuk dimasukkan ke dalam blok.

Langkah pertama dalam membuat *multi smart contract* adalah mengekstraksi informasi dari bukti digital dengan bantuan helper GetID3 . Hasil ekstraksi informasi akan disimpan dalam satu variabel array.

```
/*metadata */
 $metadata=[];$hash=[];
        if(count($listed_image)>0){
          for($i=0;$i<count($listed_image);$i++){
              /* get metadata */
                 $filelocation='uploads/bcoc/'.$listed_image[$i];
              $metadata[]=json_encode(get_fileinfo($filelocation));
              $hash[]=get_hash($filelocation);

              /* get metadata */
          }
          $save_data['hash']=json_encode($hash);
          $save_data['file_properties']=json_encode($metadata);
        }
 /* end metadata*/
```

Langkah kedua, variabel array dari hasil ekstraksi informasi bukti digital digabungkan menjadi satu dengan variabel dari form isian chain of custody. Yaitu variabel $save_data. Langkah ketiga variabel ini akan disimpan ke dalam *naive chain* dengan bantuan *multi smart contract* melalui pemanggilan fungsi new_block yang sudah didefinisikan di dalam helper smart contract..

```
$save_data['operation']='insert data';
/* save to blockchain */
      new_block($save_data);
/* end save to blockchain*/
```

Fungsi new_blok ini dijabarkan dalam file contract_helper.php.

```
if(!function_exists('new_block')){
   function new_block($dataarray){
      $data = ['data' => $dataarray];
      $headers = [
                  'Content-Type: application/json'
                ];
      $ch = curl_init();
      curl_setopt($ch, CURLOPT_URL, "http://localhost:3001/mineBlock");
      curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "POST");
      curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($data));
```

```
        curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
        $results = curl_exec($ch);
        curl_close($ch);
        return $results;
    }
}
```

Fungsi untuk menampilkan isi blok adalah dengan memanggil fungsi get_block. Data yang ditampilkan dalam bentuk data json.

```
if(!function_exists('get_block')){
    function get_block(){
        $headers = [
            'Content-Type: application/json'
        ];
        $ch = curl_init();
        curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "GET");
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
        $results = curl_exec($ch);
        curl_close($ch);
        return $results;
    }
}
```

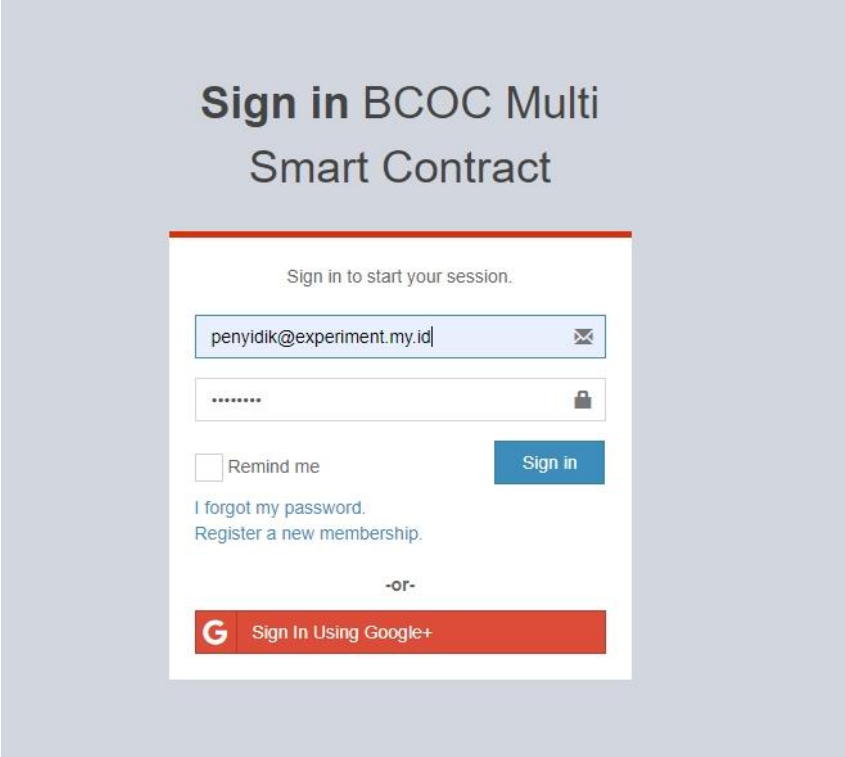
4.1.3 Membangun Middleware

Pemrograman PHP digunakan untuk menghubungkan antara naive chain dengan multi smart contract dan antarmuka (*frontend*). *Naive chain* akan terhubung ke *multi smart contract* mengunakan API dalam bentuk url yang diakses menggunakan fungsi curl pada *multi smart contract*. Sedangkan penghubung antara multi smart contract dengan antar muka atau *frontend* yaitu dengan menggunakan kode PHP.

Gambar 4.2 Middleware Naive Chain, Multi Smart Contract dan Frontend

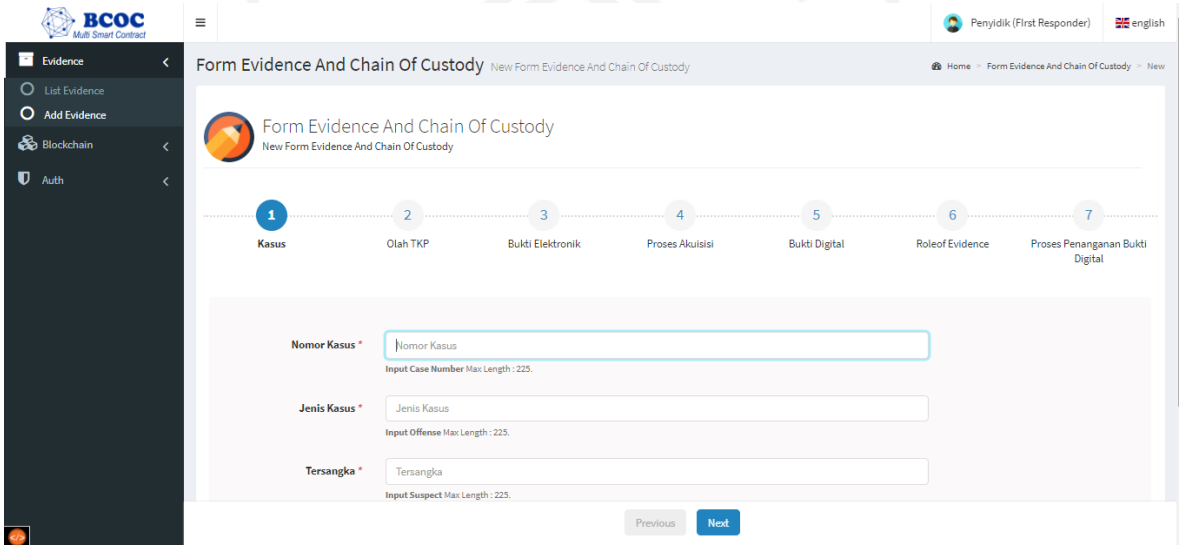
4.1.4 Membangun Antarmuka Sistem

Langkah langkah membangun aplikasi *Multi Smart Contract* dimulai dari penggunaan *Service Apache, Mysql, Docker dan Composer*. Setelah itu pengguna melakukan *login*



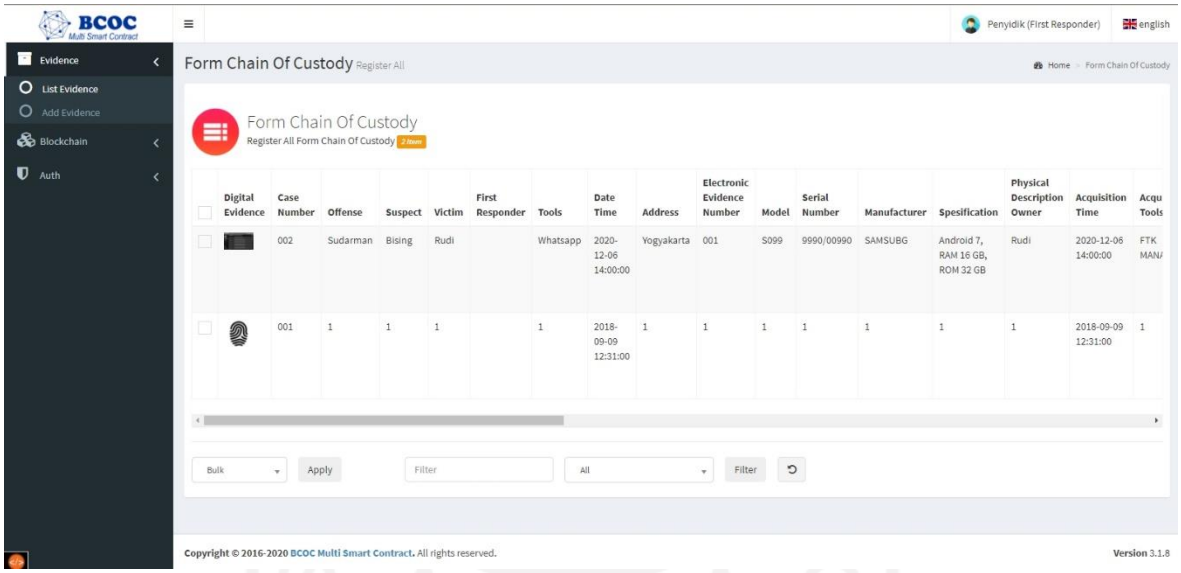
Gambar 4.3 Tampilan Halaman *Login.*

Setelah berhasil melakukan *login* untuk masuk ke aplikasi, penyidik bisa melakukan *upload* bukti digital dan mengisi *form Chain of Custody* Pada menu *Add* Evidence. Bukti digital bisa diinputkan lebih dari satu, karena bukti digital dari suatu kasus bisa saja terdiri dari beberapa buah.



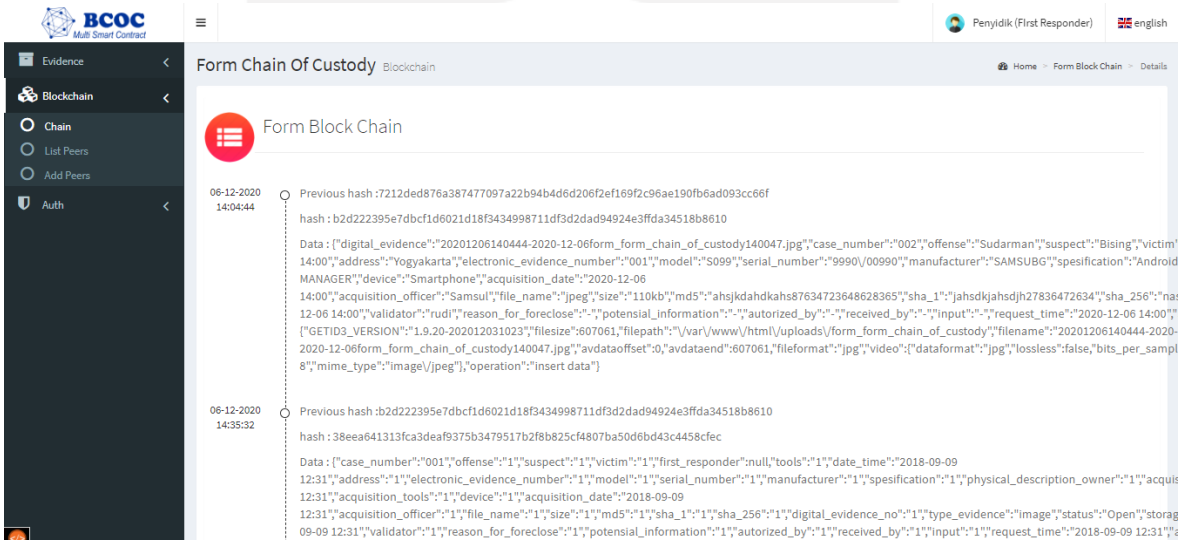
Gambar 4.4 Tampilan Halaman *Add Evidence.*

Hasil *upload* bukti digital tersebut selanjutnya bisa dilihat pada menu *List Evidence*. Halaman yang ditampilkan disini akan menyesuaikan hak akses dari masing masing *user* atau pengguna.



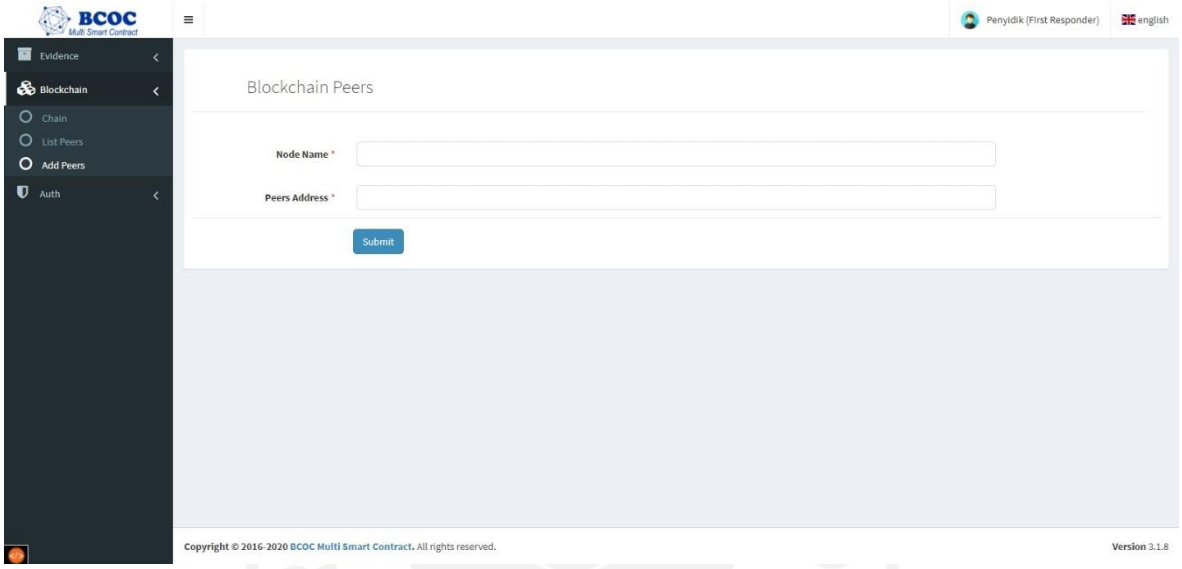
Gambar 4.5 Tampilan Halaman *List Evidence*.

Berikutnya data bukti digital dan *form Chain of Custody* disimpan pada *database* dan juga pada *Blockchain*. Data yang ada pada *Blockchain* akan bertambah terus dan apabila ada penghapusan data, maka akan bisa terdeteksi dari urutan *hash* yang ada didalam *block*. Data yang ada pada *Blockchain log*-nya bisa dilihat pada menu *Chain.*



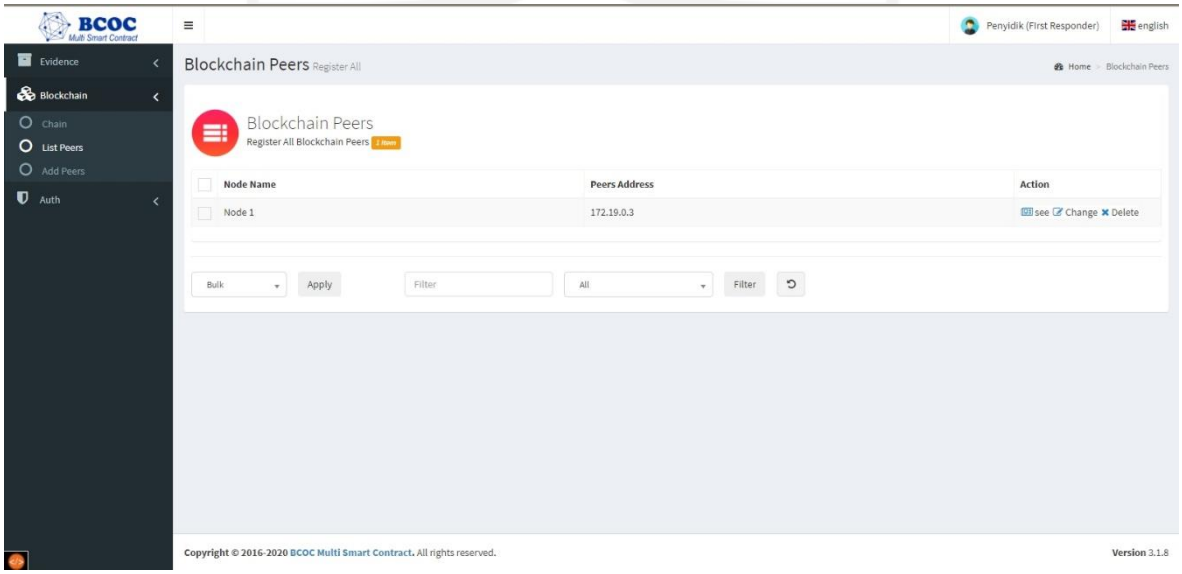
Gambar 4.6 Tampilan Halaman *Chain.*

Untuk meningkatkan keamanan penyimpanan bukti digital dan *Chain of Custody* maka *Blockchain* ini bisa dibuatkan *node* atau *server* baru yang akan terus saling berintegrasi satu sama lain. Untuk mengintegrasikan antar *Blockchain*, bisa menggunakan menu *Add Peer*.



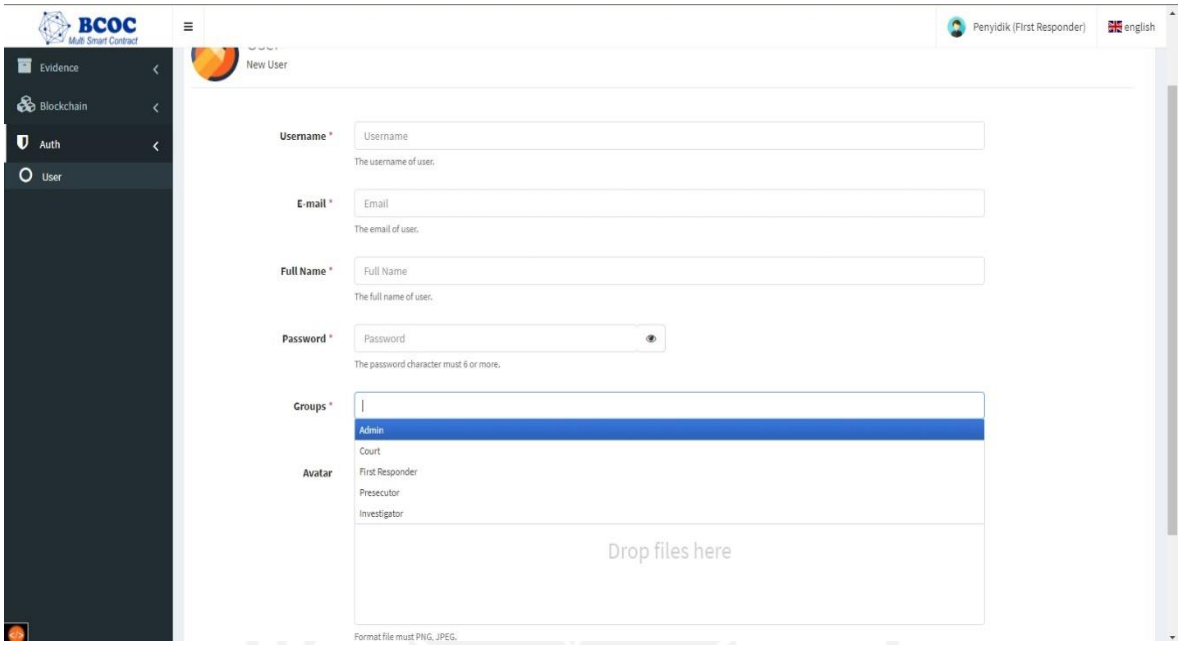
Gambar 4.7 Tampilan Hamalam *Add Peer*.

Sedangkan daftar *peer* atau *node* yang saling berintegrasi bisa dilihat pada menu *List Peer*. Suatu saat ketika salah satu node sudah tidak dibutuhkan lagi, koneksi bisa dihapus melalui menu ini.

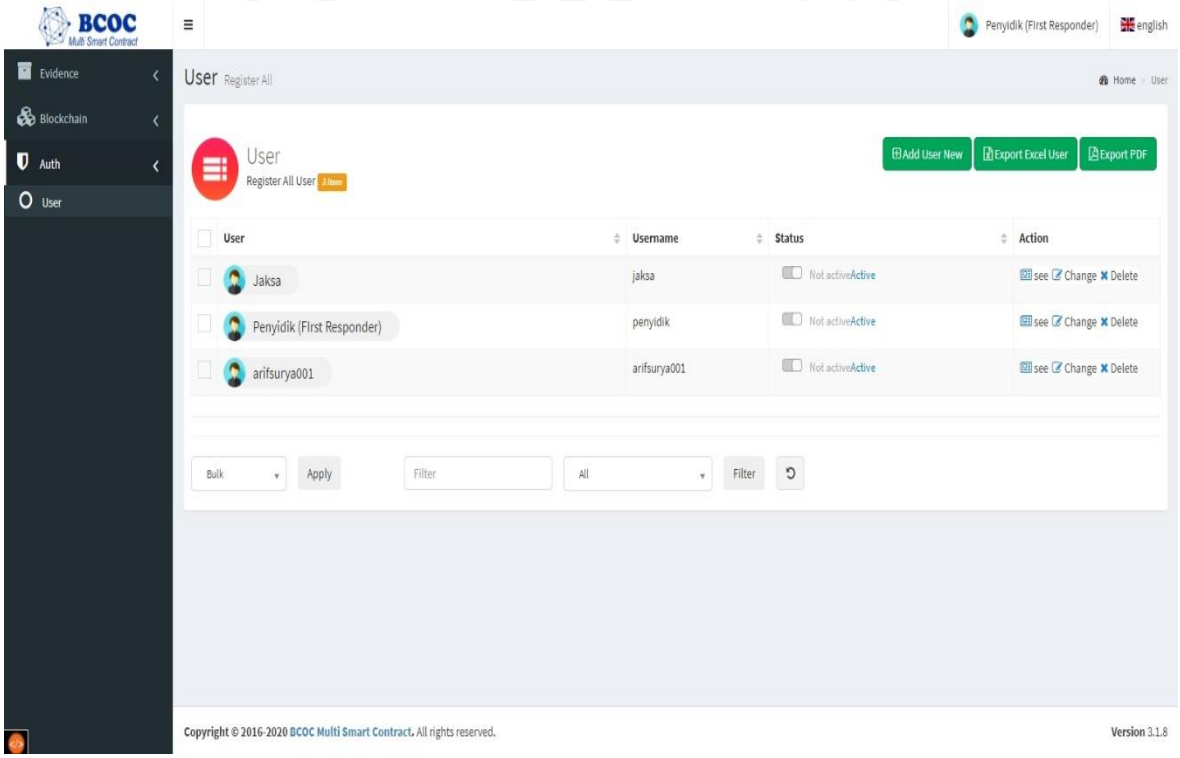


Gambar 4.8 Tampilan Halaman *List Peer*.

Selanjutnya menu *user* bisa digunakan untuk menambah pengguna, mengubah detail dan hak akses pengguna serta menghapus data pengguna.



Gambar 4.9 Tampilan Halaman *Add User*.



Gambar 4.10 Tampilan Halaman *List User*.

4.2. Pengujian Sistem

4.2.1 Pengujian Kerja Sistem

Pengujian kerja sistem dilakukan dengan cara memastikan alur sistem dan hak akses sudah bisa berjalan sesuai rancangan multi smart contract pada *naive chain* untuk meningkatkan integritas bukti digital dan *chain of custody*.

Tabel 4.1 Hasil Pengujian Kerja Sistem

No	Skema Pengujian	Otoritas Pengguna			
		First Respon	Jaksa	Ahli	Lawyer
1	Input detail kasus	✓			
2	Lihat detail kasus	✓	✓		
3	Unggah Bukti Digital	✓			
4	Input *Chain of Custody*	✓			
5	Unduh Bukti Digital	✓		✓	
6	Lihat Chain of Custody	✓		✓	✓
7	Unduh form Chain of Custody	✓	✓		
8	Lihat Log akses data pada naive chain	✓			

Tabel ini menunjukkan bahwa sistem sudah berjalan sesuai rancangan yang telah dibuat. Sistem sudah bisa menangani keperluan untuk menyimpan bukti digital dan *chain of custody*.

4.2.2 Pengujian Implementasi *Multi Smart Contract*

Dalam melakukan pengujian terhadap smart contract , perlu dipastikan bahwa kemampuan sistem *multi smart contract* dapat terpenuhi sesuai alur proses sistem *smart contract* yang sudah ada sebelumnya.

Tabel 4.2 hasil Pengujian Implementasi Multi Smart Contract

No	Alur / Proses	BCOC	BDEC	Multi Smart Contract
1	Login sistem Blockchain	✓	✓	✓
2	Penyimpanan data user dan chain of custody ke database	✓	✓	✓
3	Penyimpanan Bukti digital ke dalam sistem	✓	✓	✓
4	Penyimpanan Hash bukti digital	✓	✓	✓
5	Penyimpanan Log aktivitas terhadap bukti digital dan *chain ofcustody*	✓	✓	✓
6	Ekstraksi informasi untuk meningkatkan	-	-	✓

	integritas bukti digital			
7	Penggunaan multi smart contract secara otomatis okeh sistem.	-	-	✓

Dari tabel di atas , bisa dilihat bahwa beberapa fungsi dasar dari konsep blockchain digital chain of custody yang digunakan pada penelitian ini sudah sama dengan penelitian sebelumnya. Selain itu juga dari penerapan multi smart contract ini, informasi yang lebih detail diberikan untuk menjaga integritas bukti digital dan chain of custody.

4.2.3 Pengujian Integritas Bukti Digital dan *Chain of Custody*

Salah satu cara untuk meningkatan integritas bukti digital dan chain of custody selain dengan menyajikan hash dari bukti digital yaitu dengan menyajikan informasi yang lebih detail terkait bukti digital yang disimpan. Penggunaan plugin GetID3 menjadi solusi untuk melakukan ekstraksi informasi dari bukti digital. Dari hasil ekstraksi data tersebut kemudian dilakukan pengelompokkan bukti digital berdasarkan tipe bukti digital yang digolongkan menjadi bukti digtial gambar, bukti digital audio, bukti digital video, dan bukti digital dokumen atau file bukti lainnya.

Tabel 4.3 Detail Komponen Informasi Berdasarkan TipeBukti Digital

Komponen	Image	Audio	Video	Document/Other File
Avdataend	✓	✓	✓	✓
Avdataoffset	✓	✓	✓	✓
Encoding	✓	✓	✓	✓
Fileformat	✓	✓	✓	✓
Filename	✓	✓	✓	✓
Filenamepath	✓	✓	✓	✓
Filepath	✓	✓	✓	✓
Filesize	✓	✓	✓	✓
mime type	✓	✓	✓	✓
bits per sample	✓	-	✓	-
compression ratio	✓	✓	-	-
Dataformat	✓	✓	✓	-
Lossless	✓	✓	-	-
pixel aspect ratio	✓	-	-	-
resolution x	✓	-	✓	-
resolution y	✓	-	✓	-
Bitrate	-	✓	✓	-

bitrate mode	-	✓	✓	-
channelmode	-	✓	✓	-
Channels	-	✓	✓	-
encoder setting	-	✓	✓	-
encoder options	-	✓	✓	-
sample rate	-	✓	✓	-
compatible brands	-	✓	✓	-
major brands	-	✓	✓	-
playtime seconds	-	-	✓	-
playtime string	-	-	✓	-
Controller	-	-	✓	-
display scale	-	-	✓	-
free hierarchy	-	-	✓	-
free name	-	-	✓	-
free offset	-	-	✓	-
free size	-	-	✓	-
ftyp fourcc	-	-	✓	-
ftype hierarchy	-	-	✓	-
ftype name	-	-	✓	-
ftype offset	-	-	✓	-
ftype signature	-	-	✓	-
ftype size	-	-	✓	-
ftype unknown	-	-	✓	-
hinting	-	-	✓	-
mdat hierarchy	-	-	✓	-
mdat name	-	-	✓	-
mdat offset	-	-	✓	-
mdat size	-	-	✓	-
Fourcc	-	-	✓	-
fourcc_lookup	-	-	✓	-
frame_rate	-	-	✓	-
Rotate	-	-	✓	-

Dari tabel menyajikan bahwa masing masing bukti digital memiliki karakteristik sendiri dan memiliki jumlah karakter yang berbeda antara satu tipe bukti digital dengan tipe bukti digital lainnya. Hasil ekstarksi informasi secara otomatis yang dilakukan oleh GetID3 akan disimpan kedalam blo sehingga ketika ada perubahan yang dilakukan pada bukti digital, maka metadatanya akan ikut berubah dan history perubahan data bisa dilihat melalui data yang tercatat di dalam blok. Oleh karena itu penyajian informasi yang lebih detail akan meningkatkan integritas bukti digital.

4.2.4 Pengujian Performa *Multi Smart Contract*

Selain pengujian integritas data, diperlukan pula pengujian performa untuk mengukur kemampuan sistem multi smart contract dalam menangani kemungkinan penambahan data yang lebih banyak dan lebih intens, yang mengakibatkan daya tampung bisa cepat penuh.Skenario pengujian performa Multi smart contract dilakukan dengan cara mengupload bukti digital dengan tipe berbeda dengan kombinasi jumlah yang berbeda – beda. Hasil pengujian performa dengan upload bukti digital bisa dilihat pada tabel.

Tabel 4.4 Hasil Pengujian Performa *Multi Smart Contract*

No	Skema Pengujian	Jumlah Komponen yang Disimpan	Ukuran File	Kecepatan Akses
1	Upload 1 Bukti Digital Gambar	1	1011 KB	2.18 detik
2	Upload 2 Bukti Digital Gambar	2	260 KB	1.99 detik
3	Upload 3 Bukti Digital Gambar	3	1271 KB	2.36 detik
4	Upload 1 Bukti Digital Audio	1	1106 KB	2.09 detik
5	Upload 2 Bukti Digital Audio	2	91 KB	1.91 detik
6	Upload 3 Bukti Digital Audio	3	1197 KB	2.16 detik
7	Upload 1 Bukti Video	1	1998 KB	2.08 detik
8	Upload 2 Bukti Video	2	2939 KB	2.02 detik
9	Upload 3 Bukti Video	3	4937 KB	2.06 detik
10	Upload 1 Dokumen	1	1777 KB	2.10 detik
11	Upload 2 Dokumen	2	859 KB	1.98 detik
12	Upload 3 Dokumen	3	2636 KB	2.09 detik
13	Upload 1 Gambar 2 Audio	3	1102 KB	2.58 detik
14	Upload 1 Gambar 2 Video	3	3950 KB	2.16 detik
15	Upload 1 Gambar 2 Dokumen	3	1870 KB	2.11 detik
16	Upload 2 Gambar 1 Audio	3	1366 KB	2.14 detik
17	Upload 2 Gambar 1 Video	3	2258 KB	2.15 detik
18	Upload 2 Gambar 1 Dokumen	3	2037 KB	2.22 detik
19	Upload 1 Audio 2 Video	3	4045 KB	2.51 detik
20	Upload 1 Audio 2 Dokumen	3	1965 KB	2.27 detik
21	Upload 2 Audio 1 Video	3	2089 KB	2.14 detik
22	Upload 2 Audio 1 Dokumen	3	1868 KB	1.99 detik
23	Upload 1 Video 2 Dokumen	3	2857 KB	1.97 detik
24	Upload 2 Video 1 Dokumen	3	3775 KB	1.88 detik
25	Upload 1 Gambar 1 Audio 1 Video dan 1 Dokumen	3	4786 KB	2.16 detik
26	Unduh Bukti Digital	1	1011 KB	0.5 detik
28	Hapus bukti digital	1	1011 KB	1.85 detik

Dari hasil pengujian, jumlah file bukti digital , ukuran file belum mempengaruhi performa sistem *multi smart contract*. Namun perbandingan dengan hasil penelitian lain diperlukan agar bisa mengetahui performa dari sistem *multi smart contract*. Langkah pertama yaitu kita harus mengetahui nilai rata-rata dari ukuran file.

$$\overline{Size} = \frac{\sum size}{\sum jumlah\ skema} \tag{4.1}$$

Dengan rumus ini maka diperoleh nilai rata rata ukuran file perhitungannya total ukuran dibagi dengan jumlah skema ujian, 56072 KB dibagi 27 mnejadi 2076.74 KB. Kemudian akan dihtiung juga rata-rata dari waktu . umusanya sebagai berikut:

$$\overline{time} = \frac{\sum time}{\sum jumlah\ skema} \tag{4.2}$$

Nilai rata-rata waktu yang diperoleh dari perhitungan total waktu dibagi dengan jumlah jumlah skema ujian , 55.65 detik dibagi 27 adalah 2.06 detik. Selanjutnya perlu dicari nilai rasio antara rata-rata waktu dengan rata-rata ukuran file,Sehingga hasilnya akan menjadi, 2.06 : 2076.74 hasilnya rasionya adalah 0.000991939 .

4.3. Analisa

4.3.1 Analisa Implementasi *Multi Smart Contract*

Analisa penggunaan aplikasi dilakukan untuk mengetahui komponen informasi apa saja yang bisa digali dengan menggunakan metode *Multi Smart Contracts* yang selanjutnya digunakan untuk membandingkan dengan metode lain yang dijadikan acuan pada penelitian yang sudah ada sebelumnya dengan menggunakan metode *Blockchain based Chain of Custody* (BCOC) dan *Block Chain Digital Evidence Cabinet* (BDEC).

Tabel 4.5 Analisa Impelmentasi *Multi Smart Contract*

No	Klausul	BCOC	BDEC	Multi Smart Contract	Keterangan
1	Mampu mengakomodir lebih dari satu bukti digital	✓	✓	✓	Dalam satu kasus memiliki banyak bukti digital
2	Bukti digital lebih spesifik terhadap alat bukti persidangan	-	-	✓	Hanya alat bukti yang disimpan ke dalam sistem

3	Space penyimpanan yang dibutuhkan lebih kecil	-	-	✓	Yang disimpan alat bukti bukan barang bukti hasil akuisisi.
4	Informasi Bukti Digital Lengkap	-	-	✓	Informasi diekstrak otomatis langsung dari bukti digital.
5	Smart contract dinamis sesuai tipe bukti digital	-	-	✓	Mampu menyimpan variasi data ke blok berdsarkan tipe
6	Memiliki log aktifitas terhadap bukti digital dan *chain of custody*	✓	✓	✓	Aksi input, update, akses, dan delete disimpan dalam blok
7	Log aktifitas bukti digital tetap terjaga di dalam blok walaupun kasus, bukti digital dan *chain of custody* dihapus	✓	✓	✓	File yang dihapus bisa dilihat aktivitasnya dari rantai bloknya

Beberapa pengembangan yang dilakukan dalam penggunaan metode *Multi Smart Contract* ini antara lain, yaitu mengurangi penggunaan media penyimpanan bukti digital dengan hanya menyimpan bukti digital yang menjadi alat bukti dipersidangan. Kemudian pemanfaatan plugin GetID3 berhasil menyajikan informasi yang lebih detail terhadap bukti digital secara otomatis oleh sistem. Penambahan informasi ini bisa meningkatkan integritas bukti digital. Selanjutnya konsep *multi smart contract* yang dibangun bersifat dinamis sehingga diharapkan mampu mengatasi adanya perubahan-perubahan pada bukti digital di masa yang akan datang.

4.3.2 Analisa Pengujian Integeritas Bukti Digital

Adanya penambahan komponen informasi tentang detail bukti digital yang dikelola sesuai tipenya diharapkan bisa membantu menyajikan informasi yang lebih lengkap bagi investigator, ahli, dan pihak lain yang membutuhkan informasi tersebut. Penambahan komponen informasi ini bisa digunakan untuk mempermudah menentukan tindakan yang harus dilakukan dalam proses penyelidikan.

Hasil penerapan *Multi Smart Contract* pada pengelolaan bukti digital dan *Chain of Custody*, berupa sebuah model dengan menambahkan informasi rinci tentang bukti digital ke dalam rantai *block* bersamaan dengan *form Chain of Custody*,. Penambahan informasi

rinci tentang bukti digital ini disajikan dalam bentuk *file properties* yang bisa diperoleh otomatis dari bantuan *plugin* getID3.

Pada penelitian sebelumnya yang menggunakan metode *Block Chain Digital Evidence Cabinet* (BDEC) penyajian informasi yang disimpan ke dalam *block* hanya berupa informasi dasar tentang bukti digital seperti ukuran *file*, nama *file*, lokasi penyimpanan bukti digital, dan *hash file*. Sedangkan pada penelitian ini, penulis menggunakan metode *Block Chain of Custody – Multy Smart Contracts* (BCOC-MSC) dan dapat memberikan kontribusi berupa adanya temuan tambahan komponen informasi yang lebih detail terkait bukti digital. Penambahan informasi detail tentang bukti digital ini disertai dengan penyesuaian berupa penerapan *Multi Smart Contract* untuk mengelola bukti digital dengan tipe yang berbeda-beda.

Dengan penambahan informasi yang lebih detail pada bukti digital dibantu dengan *Multi Smart Contract* maka meningkatkan integritas dan akurasi bukti digital . Selain itu juga akan membantu mempermudah dan mempercepat investigator atau ahli dalam memilih solusi atau tindakan dalam memeriksa bukti digital yang dikelola.

Berikut ini adalah detail komponen informasi dari masing-masing tipe *file* bukti digital yang diproses menggunakan metode *Block Chain of Custody – Multy Smart Contracts* (BCOC-MSC):

4.3.3 Analisa Pengujian Performa *Multi Smart Contract*

Dari hasil pengujian performa didaptkan bahwa nilai rasio dari penerapan multi smart contract dalam meningkatkan integritas bukti digital dan chain of custody menghasilkan nilai 0.000991939. Mari bandingkan dengan penelitan Blockchain Digital Evidence Cabinet (BDEC) (Yunianto, E:2019). Jika dilakukan perhitungan rasio yang sama dengan menyamakan satuan dari ukuran file dan waktu maka rasionya 0.970812 : 5.261833333 dan jika dihitung maka akan menjadi 0.184500713. Dari perbandingan ini bisa kita lihat bahwa performa multi smart contract performanya hampir sama dengan penelitan sebelumnya.

BAB V
KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan penerapan Multi Smart Contract dalam meningkatkan integritas bukti digital dan *Chain of Custody*, maka dapat disimpulkan :

1. Rancangan sistem multi smart contract meliputi proses analisa kebutuhan sistem, mmerancang alur penyimpanan bukti digital, merancang sistem multi smart contract pada naive chain, dan merancang antarmuka. Hasil dari rancangan sistem multi smart contract berupa alur proses dan deskripsi komponen-komponen yang dibutuhkan dalam membangun aplikasi berbasis *naive chain* menggunakan *multi smart contract* dalam meningkatkan integritas bukti digital.
2. Proses implementasi sistem *multi smart contract* dimulai dari membangun naive chain, membangun sistem *multi smart contract*, membangun *middleware* untuk menghubungkan *naive chain* dengan *multi smart contract*, membangun antarmuka, dan melakukan pengujian. Hasil dari proses implementasi sistem berupa aplikasi berbasis *naive chain* menggunakan *multi smart contract* dalam meningkatkan integritas bukti digital.
3. Tahapan pengujian implementasi sistem meliputi pengujian kerja sistem, pengujian implementasi sistem, pengujian integritas bukti digital dan *chain of custody*, dan pengujian performa sistem. Hasil pengujian sistem menunjukkan bahwa implementasi sistem sudah sesuai dengan sistem-sistem yang dibangun pada penelitian sebelumnya. Hasil dari pengujian integritas bukti digital menampilkan bahwa bukti digital memiliki karakteristik dan detail informasi yang berbeda-beda antara satu jenis bukti digital gambar, audio, video, dan dokumen atau jenis bukti digital lainnya. Selain itu Informasi yang lebih detail dari hasil ekstraksi bukti digital mampu meningkatkan integritas bukti digital. Hasil pengujian performa membuktikan bahwa otomatisasi dalam membuat hash dan ekstraksi informasi dari suatu bukti digital dapat mempersingkat waktu first responder dalam menginputkan form isian pada sistem *multi smart contract*. Penyimpanan bukti digital di luar blok dapat meningkatkan performa sistem *multi smart contract*. Penyimpanan bukti

yang terbatas pada alat bukti persidangan mampu mengoptimalkan media penyimpanan bukti digital.

5.2. Saran

Dalam menyelesaikan penelitian ini, peneliti tidak luput dari beberapa keterbatasan, sehingga perlu dilakukannya pengembangan. Saran untuk pengembangan dari penelitian ini antara lain :

1. Pengembangan dan implementasi dari rancangan *Multi Smart Contract* ini perlu dilakukan untuk mengikuti perkembangan dari pembaharuan jenis-jenis *file* yang suatu saat bisa saja menjadi bukti digital.
2. Peningkatan efisiensi akses terhadap *block* perlu dilakukan untuk mengatasi data yang terus bertambah banyak dan bertambah besar. Peningkatan efisiensi ini perlu dilakukan karena data pada *block* tidak akan pernah dihapus untuk menjaga esensi dari *Blockchain*.

DAFTAR PUSTAKA

Ahmad Liza, Khanji Salam, Iqbal Farkhund, Kamoun Fauzi. 2020. Blockchain-based Chain of Custody: Towards Real-time Tamper-proof Evidence Management. ARES '20: Article No.: 48 Pages 1–8. https://doi.org/10.1145/3407023.3409199

An-Nur C.M. (2020). Daftar Kejahatan Siber yang Paling Banyak Dilaporkan ke Polisi. https://databoks.katadata.co.id/datapublish/2020/09/08/daftar-kejahatan-siber-yang-paling-banyak-dilaporkan-ke-polisi. Diakses pada 25 Januari 2021

Bonomi, S., Casini, M., & Ciccotelli, C. (2018). *B-CoC: A Blockchain-based Chain of Custody for Evidences Management in Digital Forensics*. Retrieved from http://arxiv.org/abs/1807.10359

Casey, E. (2011). Digital Evidance And Computer Crime. Computer. https://doi.org/10.1017/CBO9781107415324.004

ChainLink. (2018). Blockchain ’ s Role in the Produce Supply Chain.

Chopade Mrunali. (2019). Digital Forensics : Maintaining Chain of Custody Using Blockchain. Proceedings of the Third International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)

Cosic, J. (2017). Formal Acceptability of Digital Evidence. Springer International Publishing. http://doi.org/10.1007/978-3-319-44270-9

Hanifatunnisa, Rifa. (2017). Perancangan Dan Implementasi Sistem Pencatatan E-Voting Berbasis Blockchain Tesis. Retrieved from http://budi.rahardjo.id/files/students/rifa/thesis.pdf

Hartikka Lauri. (2018). https://github.com/lhartikk/naivechain. Diakses tanggal 1 Desember 2020

Hegadekatti, K. (2017). Legal Systems and Blockchain Interactions, (66085).

Heinrich James. (2020). https://github.com/JamesHeinrich/getID3. DIakses tanggal 3 Desember 2020

interpol. (2020). Cybercrime: COVID-19 Impact. https://www.interpol.int/content/download/15526/file/COVID-19%20Cybercrime%20Analysis%20Report-%20August%202020.pdf. Diakses pada 25 januari 2021

Garcia-alfaro, J., Navarro-arribas, G., Eds, J. H., & Hutchison, D. (2017). Data Privacy Management, Cryptocurrencies and Blockchain Technology (Vol. 10436). https://doi.org/10.1007/978-3-319-67816-0

Gopalan. H. (2019). Digital Forensics Using Blockchain. IJRTE

Laurence, T. (2017). Blockchain for dummies. John Wiley & Sons, Inc. Hoboken.

Lone, A. H., & Mir, R. N. (2017). Forensic-Chain: Ethereum Blockchain Based Digital Forensics. *Scientific and Practical Cyber Security Journal (SPCSJ) 1(2):21-27*, *1*(2), 21–27.

Lone, A. H., & Mir, R. N. (2019). Forensic-chain: Blockchain based digital forensics chain of custody with PoC in Hyperledger Composer. *Digital Investigation*, *28*, 44–55. https://doi.org/10.1016/j.diin.2019.01.002

Mougayar, W. (2016). The Business Blockchain. Wiley.

Hartikka Lauri. (2020). Naive chain. https://github.com/lhartikk/naivechain. Diakses pada September 2020

Nizamuddin, N., Salah, K., Ajmal Azad, M., Arshad, J., & Rehman, M. H. (2019). *Decentralized document version control using ethereum blockchain and IPFS. Computers & Electrical Engineering*, 76, 183–197. doi:10.1016/j.compeleceng.2019.03.014

Prayudi, Y., & SN, A. (2015). Digital Chain of Custody: State of The Art. International Journal of Computer Applications, 114(5), 1–9. https://doi.org/10.5120/19971-1856

Prayudi, Y., Ashari, A., & K Priyambodo, T. (2014). Digital Evidence Cabinets: A Proposed Framework for Handling Digital Chain of Custody. *International Journal of Computer Applications*, *107*(9), 30–36. https://doi.org/10.5120/18781-0106

Ratnasari, D., Prayudi, Y., & Sugiantoro, B. (2018). XML Approach for the Solution of Chain of Custody of Digital Evidence. *International Journal of Computer Applications*, *179*(23), 20–25. https://doi.org/10.5120/ijca2018916445

Republik Indonesia. (2016). Undang-Undang No. 19 Tahun 2016 tentang Perubahan Atas Undang-Undang No. 11 Tahun 2008 tentang Informasi dan Transaksi Elektronik.

Shorish, J. (2018). Blockchain State Machine Representation. https://doi.org/10.17605/OSF.IO/EUSXG

Singhal, B., Dhameja, G., & Panda, P. S. (2018). Beginning Blockchain : A Beginner's Guide to Building Blockchain Solutions. Apress. https://doi.org/10.1007/978-1-4842-3444-0

Sitompul, Josua. 2012. Cyberspace, Cybercrimes, Cyberlaw : Tinjauan Aspek Hukum Pidana, Tatanusa, Jakarta.

Jakarta: Sekretariat Negara

Yunianto, E. (2019). *Blockchain Digital Evidence Cabinet (B-DEC) : Manajemen Bukti Digital Berbasis Blockchain*.

LAMPIRAN

1. Source Code /var/www/html/application/helpers/contract_helper.php

```
<?php
if(!function_exists('new_block')){
        function new_block($dataarray){
                $data = ['data' => $dataarray];

                $headers = [
                        'Content-Type: application/json'
                ];

                $ch = curl_init();

                // curl -H "Content-type:application/json" --data '{"data" :
"Name:Jessica, Job:BA"}' http://localhost:3001/mineBlock
                curl_setopt($ch, CURLOPT_URL,
"http://localhost:3001/mineBlock");
                curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "POST");
                curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($data));
                // curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
                // curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1);
                curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);

                $results = curl_exec($ch);
                curl_close($ch);
                return $results;
        }
}
if(!function_exists('get_block')){
        function get_block(){
                $headers = [
                        'Content-Type: application/json'
                ];

                $ch = curl_init();

                // curl -H "Content-type:application/json" --data '{"data" :
"Name:Jessica, Job:BA"}' http://localhost:3001/mineBlock
                curl_setopt($ch, CURLOPT_URL, "http://localhost:3001/Blocks");
                curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "GET");
                //curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($data));
                 curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
                // curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1);
                //curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);

                $results = curl_exec($ch);
                curl_close($ch);
                return $results;
        }
}
if(!function_exists('new_peer')){
        function new_peer($dataarray){
                $data = ['peer' => $dataarray];

                $headers = [
                        'Content-Type: application/json'
```

```
                ];

                $ch = curl_init();

                // curl -H "Content-type:application/json" --data '{"data" :
"Name:Jessica, Job:BA"}' http://localhost:3001/mineBlock
                curl_setopt($ch, CURLOPT_URL, "http://localhost:3001/addPeer");
                curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "POST");
                curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($data));
                // curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
                // curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1);
                curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);

                $results = curl_exec($ch);
                curl_close($ch);
                return $results;
        }
}
if(!function_exists('get_peer')){
        function get_peer(){
                $headers = [
                        'Content-Type: application/json'
                ];

                $ch = curl_init();

                // curl -H "Content-type:application/json" --data '{"data" :
"Name:Jessica, Job:BA"}' http://localhost:3001/mineBlock
                curl_setopt($ch, CURLOPT_URL, "http://localhost:3001/peers");
                curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "GET");
                //curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($data));
                 curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
                // curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1);
                //curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);

                $results = curl_exec($ch);
                curl_close($ch);
                return $results;
        }
}

?>
```

2. Source Code /var/www/html/application/helpers/getid3_helper.php

```
<?php
require_once('getid3/getid3.php');

if(!function_exists('get_fileinfo')) {
  function get_fileinfo($filename=''){
        // Initialize getID3 engine
            $ci =& get_instance();
        $getID3 = new getID3;

        // Analyze file and store returned data in $ThisFileInfo
        return  $filename!='' ? $getID3->analyze($filename) : FALSE;
  }
}
if(!function_exists('hashfile')) {
```

```
  function get_hash($filename=''){
        // Initialize getID3 engine
             $hash['MD5']=hash_file('md5',$filename);
        $hash['SHA-1']=hash_file('sha1',$filename);
        $hash['SHA-256']=hash_file('sha256',$filename);
        // Analyze file and store returned data in $ThisFileInfo
        return  $filename!='' ? $hash : FALSE;
  }
}
?>
```

3. Source Code /var/www/html/modules/bcoc/controllers/backend/Bcoc.php

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');


/**
*| --------------------------------------------------------------------------
*| Bcoc Controller
*| --------------------------------------------------------------------------
*| Bcoc site
*|
*/
class Bcoc extends Admin
{

       public function __construct()
       {
              parent::__construct();

              $this->load->model('model_bcoc');
              $this->lang->load('web_lang', $this->current_lang);
    $this->load->helpers(array('getid3','contract'));
       }

       /**
       * show all Bcocs
       *
       * @var $offset String
       */
       public function index($offset = 0)
       {
              $this->is_allowed('bcoc_list');

              $filter = $this->input->get('q');
              $field       = $this->input->get('f');

              $this->data['bcocs'] = $this->model_bcoc->get($filter, $field,
$this->limit_page, $offset);
              $this->data['bcoc_counts'] = $this->model_bcoc->count_all($filter,
$field);

              $config = [
                     'base_url'     => 'administrator/bcoc/index/',
                     'total_rows'   => $this->data['bcoc_counts'],
                     'per_page'     => $this->limit_page,
```

```
                'uri_segment'  => 4,
            ];

            $this->data['pagination'] = $this->pagination($config);

            $this->template->title('Form Evidence And Chain Of Custody List');
            $this->render('backend/standart/administrator/bcoc/bcoc_list',
$this->data);
      }

      /**
      * Add new bcocs
      *
      */
      public function add()
      {
            $this->is_allowed('bcoc_add');

            $this->template->title('Form Evidence And Chain Of Custody New');
            $this->render('backend/standart/administrator/bcoc/bcoc_add',
$this->data);
      }

      /**
      * Add New Bcocs
      *
      * @return JSON
      */
      public function add_save()
      {
            if (!$this->is_allowed('bcoc_add', false)) {
                  echo json_encode([
                        'success' => false,
                        'message' =>
cclang('sorry_you_do_not_have_permission_to_access')
                        ]);
                  exit;
            }

            $this->form_validation->set_rules('case_number', 'Nomor Kasus',
'trim|required|max_length[225]');
            /*$this->form_validation->set_rules('offense', 'Jenis Kasus',
'trim|required|max_length[225]');
            $this->form_validation->set_rules('suspect', 'Tersangka',
'trim|required|max_length[225]');
            $this->form_validation->set_rules('victim', 'Korban',
'trim|required|max_length[225]');
            $this->form_validation->set_rules('tools', 'Alat Olah TKP',
'trim|required');
            $this->form_validation->set_rules('date_time', 'Waktu Olah TKP',
'trim|required');
            $this->form_validation->set_rules('address', 'Lokasi Olah TKP',
'trim|required');
            $this->form_validation->set_rules('electronic_evidence_number',
'Bukti Elektronik', 'trim|required|max_length[225]');
            $this->form_validation->set_rules('model', 'Model',
'trim|required|max_length[225]');
            $this->form_validation->set_rules('serial_number', 'Serial
Number', 'trim|required');
```

```
            $this->form_validation->set_rules('manufacturer', 'Manufacturer',
'trim|required|max_length[225]');
            $this->form_validation->set_rules('spesification', 'Spesifikasi
Teknis', 'trim|required|max_length[225]');
            $this->form_validation->set_rules('physical_description_owner',
'Deskripsi Fisik', 'trim|required');
            $this->form_validation->set_rules('acquisition_time', 'Waktu
Akuisisi', 'trim|required');
            $this->form_validation->set_rules('acquisition_tools', 'Tools
Akuisisi', 'trim|required|max_length[225]');
            $this->form_validation->set_rules('device', 'Perangkat Akuisisi',
'trim|required|max_length[225]');
            $this->form_validation->set_rules('acquisition_officer', 'Petugas
Akuisisi', 'trim|required|max_length[225]');
            $this->form_validation->set_rules('bcoc_digital_evidence_name[]',
'Bukti Digital', 'trim|required');
            $this->form_validation->set_rules('digital_evidence_no', 'No Bukti
Digital', 'trim|required|max_length[225]');
            $this->form_validation->set_rules('status', 'Status Disposal',
'trim|required');
            $this->form_validation->set_rules('time_stored', 'Tanggal
Disposal', 'trim|required');
            $this->form_validation->set_rules('validator', 'Validator
Disposal', 'trim|required|max_length[225]');
            $this->form_validation->set_rules('potensial_information',
'Potensial Informasi Yang Tersimpan', 'trim|required');
            $this->form_validation->set_rules('autorized_by', 'Otorisasi
Permintaan', 'trim|required|max_length[225]');
            $this->form_validation->set_rules('received_by', 'Penerima',
'trim|required|max_length[225]');*/


            if ($this->form_validation->run()) {

                  $save_data = [
                        'case_number' => $this->input->post('case_number'),
                        'offense' => $this->input->post('offense'),
                        'suspect' => $this->input->post('suspect'),
                        'victim' => $this->input->post('victim'),
                        'first_responder' => $this->input-
>post('first_responder'),
                        'tools' => $this->input->post('tools'),
                        'date_time' => $this->input->post('date_time'),
                        'address' => $this->input->post('address'),
                        'electronic_evidence_number' => $this->input-
>post('electronic_evidence_number'),
                        'model' => $this->input->post('model'),
                        'serial_number' => $this->input-
>post('serial_number'),
                        'manufacturer' => $this->input->post('manufacturer'),
                        'spesification' => $this->input-
>post('spesification'),
                        'physical_description_owner' => $this->input-
>post('physical_description_owner'),
                        'acquisition_time' => $this->input-
>post('acquisition_time'),
                        'acquisition_tools' => $this->input-
>post('acquisition_tools'),
                        'device' => $this->input->post('device'),
```

```
                        'acquisition_officer' => $this->input-
>post('acquisition_officer'),
                        'digital_evidence_no' => $this->input-
>post('digital_evidence_no'),
                        'status' => $this->input->post('status'),
                        'time_stored' => $this->input->post('time_stored'),
                        'validator' => $this->input->post('validator'),
                        'storage_location' => $this->input-
>post('storage_location'),
                        'reason_for_foreclose' => $this->input-
>post('reason_for_foreclose'),
                        'potensial_information' => $this->input-
>post('potensial_information'),
                        'autorized_by' => $this->input->post('autorized_by'),
                        'received_by' => $this->input->post('received_by'),
                        'request_time' => $this->input->post('request_time'),
                        'approve_time' => $this->input->post('approve_time'),
                        'received_time' => $this->input-
>post('received_time'),
                        'action' => $this->input->post('action'),
                    ];

                    if (!is_dir(FCPATH . '/uploads/bcoc/')) {
                        mkdir(FCPATH . '/uploads/bcoc/');
                    }
      $listed_image = [];
                    if (count((array) $this->input-
>post('bcoc_digital_evidence_name'))) {
                        foreach ((array) $_POST['bcoc_digital_evidence_name']
as $idx => $file_name) {
                            $bcoc_digital_evidence_name_copy =
date('YmdHis') . '-' . $file_name;

                            rename(FCPATH . 'uploads/tmp/' .
$_POST['bcoc_digital_evidence_uuid'][$idx] . '/' .  $file_name,
                                FCPATH . 'uploads/bcoc/' .
$bcoc_digital_evidence_name_copy);

                            $listed_image[] =
$bcoc_digital_evidence_name_copy;

                            if (!is_file(FCPATH . '/uploads/bcoc/' .
$bcoc_digital_evidence_name_copy)) {
                                echo json_encode([
                                    'success' => false,
                                    'message' => 'Error uploading
file'
                                    ]);
                                exit;
                            }
                        }
        /*metadata */
                    $metadata=[];$hash=[];
        if(count($listed_image)>0){
          for($i=0;$i<count($listed_image);$i++){
              /* get metadata */
             $filelocation='uploads/bcoc/'.$listed_image[$i];
              $metadata[]=json_encode(get_fileinfo($filelocation));
              $hash[]=get_hash($filelocation);
```

```
            /* get metadata */
          }
          $save_data['hash']=json_encode($hash);
          $save_data['file_properties']=json_encode($metadata);
        }
        /* end metadata*/
                        $save_data['digital_evidence'] =
implode($listed_image, ',');
                    }


                    $save_data['operation']='insert data';
          /* save to blockchain */
      new_block($save_data);
                    /* end save to blockchain*/
                    $save_bcoc = $this->model_bcoc->store($save_data);


                    if ($save_bcoc) {
                        if ($this->input->post('save_type') == 'stay') {
                            $this->data['success'] = true;
                            $this->data['id']      = $save_bcoc;
                            $this->data['message'] =
cclang('success_save_data_stay', [
                                anchor('administrator/bcoc/edit/' .
$save_bcoc, 'Edit Bcoc'),
                                anchor('administrator/bcoc', ' Go back
to list')
                            ]);
                        } else {
                            set_message(
                                cclang('success_save_data_redirect', [
                                anchor('administrator/bcoc/edit/' .
$save_bcoc, 'Edit Bcoc')
                            ]), 'success');

                    $this->data['success'] = true;
                            $this->data['redirect'] =
base_url('administrator/bcoc');
                        }
                    } else {
                        if ($this->input->post('save_type') == 'stay') {
                            $this->data['success'] = false;
                            $this->data['message'] =
cclang('data_not_change');
                        } else {
                    $this->data['success'] = false;
                    $this->data['message'] = cclang('data_not_change');
                            $this->data['redirect'] =
base_url('administrator/bcoc');
                        }
                    }

             } else {
                    $this->data['success'] = false;
                    $this->data['message'] = 'Opss validation failed';
                    $this->data['errors'] = $this->form_validation-
>error_array();
```

```
		}

		echo json_encode($this->data);
	}

		/**
	* Update view Bcocs
	*
	* @var $id String
	*/
	public function edit($id)
	{
		$this->is_allowed('bcoc_update');

		$this->data['bcoc'] = $this->model_bcoc->find($id);

		$this->template->title('Form Evidence And Chain Of Custody
Update');
		$this->render('backend/standart/administrator/bcoc/bcoc_update',
$this->data);
	}

	/**
	* Update Bcocs
	*
	* @var $id String
	*/
	public function edit_save($id)
	{
		if (!$this->is_allowed('bcoc_update', false)) {
			echo json_encode([
				'success' => false,
				'message' =>
cclang('sorry_you_do_not_have_permission_to_access')
				]);
			exit;
		}

		$this->form_validation->set_rules('case_number', 'Nomor Kasus',
'trim|required|max_length[225]');
		/*$this->form_validation->set_rules('offense', 'Jenis Kasus',
'trim|required|max_length[225]');
		$this->form_validation->set_rules('suspect', 'Tersangka',
'trim|required|max_length[225]');
		$this->form_validation->set_rules('victim', 'Korban',
'trim|required|max_length[225]');
		$this->form_validation->set_rules('tools', 'Alat Olah TKP',
'trim|required');
		$this->form_validation->set_rules('date_time', 'Waktu Olah TKP',
'trim|required');
		$this->form_validation->set_rules('address', 'Lokasi Olah TKP',
'trim|required');
		$this->form_validation->set_rules('electronic_evidence_number',
'Bukti Elektronik', 'trim|required|max_length[225]');
		$this->form_validation->set_rules('model', 'Model',
'trim|required|max_length[225]');
		$this->form_validation->set_rules('serial_number', 'Serial
Number', 'trim|required');
```

```
            $this->form_validation->set_rules('manufacturer', 'Manufacturer',
'trim|required|max_length[225]');
            $this->form_validation->set_rules('spesification', 'Spesifikasi
Teknis', 'trim|required|max_length[225]');
            $this->form_validation->set_rules('physical_description_owner',
'Deskripsi Fisik', 'trim|required');
            $this->form_validation->set_rules('acquisition_time', 'Waktu
Akuisisi', 'trim|required');
            $this->form_validation->set_rules('acquisition_tools', 'Tools
Akuisisi', 'trim|required|max_length[225]');
            $this->form_validation->set_rules('device', 'Perangkat Akuisisi',
'trim|required|max_length[225]');
            $this->form_validation->set_rules('acquisition_officer', 'Petugas
Akuisisi', 'trim|required|max_length[225]');
            $this->form_validation->set_rules('bcoc_digital_evidence_name[]',
'Bukti Digital', 'trim|required');
            $this->form_validation->set_rules('digital_evidence_no', 'No Bukti
Digital', 'trim|required|max_length[225]');
            $this->form_validation->set_rules('status', 'Status Disposal',
'trim|required');
            $this->form_validation->set_rules('time_stored', 'Tanggal
Disposal', 'trim|required');
            $this->form_validation->set_rules('validator', 'Validator
Disposal', 'trim|required|max_length[225]');
            $this->form_validation->set_rules('potensial_information',
'Potensial Informasi Yang Tersimpan', 'trim|required');
            $this->form_validation->set_rules('autorized_by', 'Otorisasi
Permintaan', 'trim|required|max_length[225]');
            $this->form_validation->set_rules('received_by', 'Penerima',
'trim|required|max_length[225]');*/

            if ($this->form_validation->run()) {

                  $save_data = [
                        'case_number' => $this->input->post('case_number'),
                        'offense' => $this->input->post('offense'),
                        'suspect' => $this->input->post('suspect'),
                        'victim' => $this->input->post('victim'),
                        'first_responder' => $this->input-
>post('first_responder'),
                        'tools' => $this->input->post('tools'),
                        'date_time' => $this->input->post('date_time'),
                        'address' => $this->input->post('address'),
                        'electronic_evidence_number' => $this->input-
>post('electronic_evidence_number'),
                        'model' => $this->input->post('model'),
                        'serial_number' => $this->input-
>post('serial_number'),
                        'manufacturer' => $this->input->post('manufacturer'),
                        'spesification' => $this->input-
>post('spesification'),
                        'physical_description_owner' => $this->input-
>post('physical_description_owner'),
                        'acquisition_time' => $this->input-
>post('acquisition_time'),
                        'acquisition_tools' => $this->input-
>post('acquisition_tools'),
                        'device' => $this->input->post('device'),
```

```
                        'acquisition_officer' => $this->input-
>post('acquisition_officer'),
                        'digital_evidence_no' => $this->input-
>post('digital_evidence_no'),
                        'status' => $this->input->post('status'),
                        'time_stored' => $this->input->post('time_stored'),
                        'validator' => $this->input->post('validator'),
                        'storage_location' => $this->input-
>post('storage_location'),
                        'reason_for_foreclose' => $this->input-
>post('reason_for_foreclose'),
                        'potensial_information' => $this->input-
>post('potensial_information'),
                        'autorized_by' => $this->input->post('autorized_by'),
                        'received_by' => $this->input->post('received_by'),
                        'request_time' => $this->input->post('request_time'),
                        'approve_time' => $this->input->post('approve_time'),
                        'received_time' => $this->input-
>post('received_time'),
                        'action' => $this->input->post('action'),
                    ];

                    $listed_image = [];
                    if (count((array) $this->input-
>post('bcoc_digital_evidence_name'))) {
                        foreach ((array) $_POST['bcoc_digital_evidence_name']
as $idx => $file_name) {
                            if
(isset($_POST['bcoc_digital_evidence_uuid'][$idx]) AND
!empty($_POST['bcoc_digital_evidence_uuid'][$idx])) {
                                $bcoc_digital_evidence_name_copy =
date('YmdHis') . '-' . $file_name;

                                rename(FCPATH . 'uploads/tmp/' .
$_POST['bcoc_digital_evidence_uuid'][$idx] . '/' .  $file_name,
                                        FCPATH . 'uploads/bcoc/' .
$bcoc_digital_evidence_name_copy);

                                $listed_image[] =
$bcoc_digital_evidence_name_copy;

                                if (!is_file(FCPATH . '/uploads/bcoc/' .
$bcoc_digital_evidence_name_copy)) {
                                    echo json_encode([
                                        'success' => false,
                                        'message' => 'Error
uploading file'
                                        ]);
                                    exit;
                                }

                            }else
                $listed_image[]=$file_name;
                        }
                    }

                    $save_data['digital_evidence'] = implode($listed_image,
',');
                /*tambahan data */
```

```
                $metadata=[];$hash=[];
      if(count($listed_image)>0){
        for($i=0;$i<count($listed_image);$i++){
            /* get metadata */
             $filelocation='uploads/bcoc/'.$listed_image[$i];
            $metadata[]=json_encode(get_fileinfo($filelocation));
            $hash[]=get_hash($filelocation);
            /* get metadata */
        }
        $save_data['hash']=json_encode($hash);
        $save_data['file_properties']=json_encode($metadata);
      }
                $save_data['operation']='update data';
      /* save to blockchain */
      new_block($save_data);
                /* end save to blockchain*/
      $save_bcoc = $this->model_bcoc->change($id, $save_data);

                if ($save_bcoc) {
                      if ($this->input->post('save_type') == 'stay') {
                            $this->data['success'] = true;
                            $this->data['id']      = $id;
                            $this->data['message'] =
cclang('success_update_data_stay', [
                                  anchor('administrator/bcoc', ' Go back
to list')
                            ]);
                      } else {
                            set_message(
                                  cclang('success_update_data_redirect', [
                            ]), 'success');

                $this->data['success'] = true;
                            $this->data['redirect'] =
base_url('administrator/bcoc');
                      }
                } else {
                      if ($this->input->post('save_type') == 'stay') {
                            $this->data['success'] = false;
                            $this->data['message'] =
cclang('data_not_change');
                      } else {
                $this->data['success'] = false;
                $this->data['message'] = cclang('data_not_change');
                            $this->data['redirect'] =
base_url('administrator/bcoc');
                      }
                }
            } else {
                  $this->data['success'] = false;
                  $this->data['message'] = 'Opss validation failed';
                  $this->data['errors'] = $this->form_validation-
>error_array();
            }

            echo json_encode($this->data);
      }

      /**
```

```
	* delete Bcocs
	*
	* @var $id String
	*/
	public function delete($id = null)
	{
		$this->is_allowed('bcoc_delete');

		$this->load->helper('file');

		$arr_id = $this->input->get('id');
		$remove = false;

		if (!empty($id)) {
			$remove = $this->_remove($id);
		} elseif (count($arr_id) >0) {
			foreach ($arr_id as $id) {
				$remove = $this->_remove($id);
			}
		}

		if ($remove) {
		/* save to blockchain */
			$save_data['operation']='delete data';
		new_block($save_data);
			/*save to blockchain*/
		set_message(cclang('has_been_deleted', 'bcoc'), 'success');
	  } else {
		set_message(cclang('error_delete', 'bcoc'), 'error');
	  }

		redirect_back();
	}

		/**
	* View view Bcocs
	*
	* @var $id String
	*/
	public function view($id)
	{
		$this->is_allowed('bcoc_view');

		$this->data['bcoc'] = $this->model_bcoc->join_avaiable()-
>filter_avaiable()->find($id);

		$this->template->title('Form Evidence And Chain Of Custody
Detail');
		$this->render('backend/standart/administrator/bcoc/bcoc_view',
$this->data);
	}

	/**
	* delete Bcocs
	*
	* @var $id String
	*/
	private function _remove($id)
	{
```

```
            $bcoc = $this->model_bcoc->find($id);


            if (!empty($bcoc->digital_evidence)) {
                  foreach ((array) explode(',', $bcoc->digital_evidence) as
$filename) {
                        $path = FCPATH . '/uploads/bcoc/' . $filename;

                        if (is_file($path)) {
                              $delete_file = unlink($path);
                        }
                  }
            }

            return $this->model_bcoc->remove($id);
      }


      /**
      * Upload Image Bcoc *
      * @return JSON
      */
      public function upload_digital_evidence_file()
      {
            if (!$this->is_allowed('bcoc_add', false)) {
                  echo json_encode([
                        'success' => false,
                        'message' =>
cclang('sorry_you_do_not_have_permission_to_access')
                        ]);
                  exit;
            }

            $uuid = $this->input->post('qquuid');

            echo $this->upload_file([
                  'uuid'                  => $uuid,
                  'table_name' => 'bcoc',
            ]);
      }

      /**
      * Delete Image Bcoc *
      * @return JSON
      */
      public function delete_digital_evidence_file($uuid)
      {
            if (!$this->is_allowed('bcoc_delete', false)) {
                  echo json_encode([
                        'success' => false,
                        'error' =>
cclang('sorry_you_do_not_have_permission_to_access')
                        ]);
                  exit;
            }

            echo $this->delete_file([
           'uuid'              => $uuid,
           'delete_by'         => $this->input->get('by'),
```

```
            'field_name'        => 'digital_evidence',
            'upload_path_tmp'   => './uploads/tmp/',
            'table_name'        => 'bcoc',
            'primary_key'       => 'id',
            'upload_path'       => 'uploads/bcoc/'
        ]);
    }
    /**
    * Get Image Bcoc    *
    * @return JSON
    */
    public function get_digital_evidence_file($id)
    {
          if (!$this->is_allowed('bcoc_update', false)) {
                echo json_encode([
                      'success' => false,
                      'message' => 'Image not loaded, you do not have
permission to access'
                      ]);
                exit;
          }
          $bcoc = $this->model_bcoc->find($id);
          echo $this->get_file([
          'uuid'              => $id,
          'delete_by'         => 'id',
          'field_name'        => 'digital_evidence',
          'table_name'        => 'bcoc',
          'primary_key'       => 'id',
          'upload_path'       => 'uploads/bcoc/',
          'delete_endpoint'   =>
'administrator/bcoc/delete_digital_evidence_file'
        ]);
    }
    /**
    * Export to excel
    *
    * @return Files Excel .xls
    */
    public function export()
    {
          $this->is_allowed('bcoc_export');
          $this->model_bcoc->export(
                'bcoc',
                'bcoc',
                $this->model_bcoc->field_search
          );
    }
    /**
    * Export to PDF
    *
    * @return Files PDF .pdf
    */
    public function export_pdf()
    {
          $this->is_allowed('bcoc_export');
          $this->model_bcoc->pdf('bcoc', 'bcoc');
    }

    public function single_pdf($id = null)
```

```
        {
              $this->is_allowed('bcoc_export');

              $table = $title = 'bcoc';
              $this->load->library('HtmlPdf');
          $config = array(
              'orientation' => 'p',
              'format' => 'a4',
              'marges' => array(5, 5, 5, 5)
          );
          $this->pdf = new HtmlPdf($config);
          $this->pdf->setDefaultFont('stsongstdlight');
          $result = $this->db->get($table);
          $data = $this->model_bcoc->find($id);
          $fields = $result->list_fields();
          $content = $this->pdf->loadHtmlPdf('core_template/pdf/pdf_single', [
              'data' => $data,
              'fields' => $fields,
              'title' => $title
          ], TRUE);
          $this->pdf->initialize($config);
          $this->pdf->pdf->SetDisplayMode('fullpage');
          $this->pdf->writeHTML($content);
          $this->pdf->Output($table.'.pdf', 'H');
        }
}
/* End of file bcoc.php */
/* Location: ./application/controllers/administrator/Bcoc.php */
```

4. File untuk pengujian *Multi Smart Contract*

no	Nama File	Tipe	Ukuran
1	Gambar 1	jpg	1011 KB
2	Gambar 2	png	120 KB
3	Gambar 3	gif	140 KB
4	Audio 1	mp3	1106 KB
5	Audio 2	ogg	28 KB
6	Audio 3	wav	63 KB
7	Video 1	mp4	1998 KB
8	Video 2	3gp	1450 KB
9	Video 3	mkv	1489 KB
10	File 1	pdf	799 KB
11	File 2	doc	60 KB
12	File 3	ppt	1777 KB