Smart Contract Sebagai Proof-of-concept DRMChain Hak Cipta Buku Digital Ditinjau Dari Segi Hukum Positif Indonesia

Jehan Afwazi Ahmad
18917115

Tesis diajukan sebagai syarat untuk meraih gelar Magister Komputer
Konsentrasi Sistem Informasi Enterprise
Program Studi Informatika Program Magister
Fakultas Teknologi Industri
Universitas Islam Indonesia
2023

Lembar Pengesahan Pembimbing

Smart Contract Sebagai Proof-of-concept DRMChain Hak Cipta Buku Digital Ditinjau Dari Segi Hukum Positif Indonesia

Jehan Afwazi Ahmad
18917115

Yogyakarta, Januari, 2023

Pembimbing

Dr. Raden Teduh Dirgahayu, S.T., M.Sc.

Lembar Pengesahan Penguji

Smart Contract Sebagai Proof-of-concept DRMChain Hak Cipta Buku Digital Ditinjau Dari Segi Hukum Positif Indonesia

Jehan Afwazi Ahmad

18917115

Yogyakarta, Januari, 2023

Tim Penguji,

Dr. R Teduh Dirgahayu, ST., M.Sc _____

Ketua

Irving Vitra Paputungan, ST., M.Sc., Ph.D _____

Anggota I

Dr. Mukhammad Andri Setiawan, ST., M.Sc _____

Anggota II

Mengetahui,

Ketua Program Studi Informatika Program Magister

Fakultas Teknologi Industri

Universitas Islam Indonesia

Irving Vitra Paputungan, ST., M.Sc., Ph.D.

Abstrak

Smart Contract Sebagai Proof-of-concept DRMChain Hak Cipta Buku Digital Ditinjau Dari Segi Hukum Positif Indonesia

Kasus pelanggaran hak cipta digital di indonesia cukup menyita perhatian karena syarat kerugian yang cukup besar terhadap pihak terkait. Meskipun begitu, beberapa kasus pelanggaran tersebut secara hukum belum dapat ditegakkan. Hal ini karena kasus pelanggaran hak cipta secara hukum bersifat delik aduan, sehingga penegakan hukum baru dapat dilakukan seketika terdapat laporan dari pihak terkait. Bahkan teknologi DRM juga belum secara efektif dapat menyelesaikan permasalahan tersebut. Teknologi DRM saat ini masih memiliki beberapa kekurangan, seperti tidak mampu melacak kebocoran data dan membuktikan pelanggaran, tidak dapat meng-autentikasi konten ketika sumber informasi kontennya tidak lagi diketahui, baik karena rusak atau proses manipulasi, sistem DRM juga tidak dapat memberikan batasan unduh bagi pengguna terhadap konten digital, dan sistem tidak dapat memindahkan hak kepemilikan konten kepada orang lain. Untungnya blockchain adalah teknologi terdesentralisasi, bersifat *immutable*, dan menjamin keamanan transaksi, sehingga memungkinkan digabungkan (DRMChain) dan menjadi solusi yang ideal untuk menyelesaikan permasalahan pada sistem DRM yang ada. Penelitian ini akan membuktikan apakah konsep DRMChain dapat menyelesaikan permasalahan hak cipta digital di Indonesia sesuai hukum positif yang berlaku. Penelitian ini menggunakan metode *Proof-of-Concept* (PoC) untuk membuktikan konsep tersebut. Dengan melibatkan pengujian secara *blackbox* untuk menunjukkan kelayakan dari teori/konsep yang dibuat. Tahapan yang dilakukan dalam penelitian adalah (i) mengidentifikasi masalah hak cipta digital di Indonesia, (ii) mengidentifikasi kebutuhan sistem, (iii) implementasi aplikasi, dan (iv) evaluasi. Dari hasil identifikasi masalah menunjukkan, bahwa terdapat beberapa kriteria ideal agar sistem dapat menyelesaikan permasalahan hak cipta. Kriteria tersebut antara lain: (i) sistem harus dapat membatasi mengunduh, distribusi, dan modifikasi, (ii) sistem juga harus dapat mengidentifikasi pencipta, (iii) sistem harus dapat membutikan keaslian konten, (iv) sistem dapat membuktikan pelanggaran, dan (v) sistem dapat memindahkan kepemilikan. Hasil penelitian ini menunjukkan bahwa sistem DRMChain dapat menjadi solusi permasalahan hak cipta digital. Berkenaan dengan kriteria membuktikan pelaku pelanggaran (pembajakan), sistem tidak mampu menanganinya karena perilaku tersebut umumnya

dilakukan di luar sistem, sehingga tidak dapat dilakukan tindakan apapun terhadap perilaku tersebut. Namun dengan fitur log transaksi pada blockchain, sistem ini dapat membantu melacak kemungkinan pelaku pembajakan terhadap konten yang dilaporkan oleh pihak yang berkepentingan. Meskipun begitu, konsep DRMChain yang diusulkan ini dapat dikatakan layak menjadi solusi yang ideal untuk menyelesaikan permasalahan hak cipta digital di Indonesia.

Kata kunci

DRM, Blockchain, Desentralisasi, Perlindungan Hak Cipta, Hak Kekayaan Intelektual

Abstract

Smart Contract as Proof-of-concept DRMChain Digital Book Copyrights Viewed from Indonesia's Positive Legal Perspective

Cases of digital copyright infringement in Indonesia are concerning because they have a considerable loss impact on related parties. However, several violations have not been enforced effectively because the cases of copyright infringement are legal complaints, so law enforcement can prosecute as soon as there is a report from the relevant party. Even DRM technology has not been able to solve these problems effectively. Today's DRM technology still has some weaknesses, such as not being able to track data leaks and prove violations, not being able to authenticate content when the source of content information is no longer known, either due to a malfunction or manipulation process, the DRM system also cannot provide user download restrictions on digital content, and the system cannot transfer the ownership rights of the content to others. Fortunately, blockchain is a decentralized technology, immutable, and guarantees transaction security, making it possible to combine (DRMChain) and become an ideal solution to solve problems with existing DRM systems. This research will prove whether the DRMChain concept can solve digital copyright problems in Indonesia by applying positive law. This research uses the Proof-of-Concept (POC) method to prove the concept. By involving BlackBox testing to show the feasibility of the theory/concept created. The stages carried out in the study are (i) identifying digital copyright problems in Indonesia, (ii) identifying system requirements, (iii) implementation, and (iv) evaluation. The result identifying the digital copyright problem in Indonesia shows that there are several ideal criteria for the system to solve copyright problems. These criteria include: (i) the system must be able to restrict downloading, distribution, and modification, (ii) the system must also be able to identify the creator, (iii) the system must be able to determine the authenticity of the content, (iv) the system can prove infringement, (v) the system can transfer ownership. The results of this study show that the DRMChain system can be a solution to digital copyright problems. The results of this study show that the DRMChain system can be a solution to digital copyright problems. Concerning the criteria for proving the perpetrator of the violation (piracy), the system cannot handle it because the violation behavior generally carried out outside the system, so there are no action can be taken against the behavior. But with the transaction log feature on the blockchain, this system

can help track possible perpetrators of piracy against content reported by interested parties. Even so, the proposed DRMChain concept can be said to be an ideal solution to solve digital copyright problems in Indonesia.

Keywords

DRM, Blockchain, Decentralization, Copyrigh Protection

Pernyataan Keaslian Tulisan

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, 24 Januari 2023

Jehan Afwazi Ahmad, S.Kom.

Daftar Publikasi

Ahmad, J. A., & Dirgahayu, T. (2023). *PERANAN BLOCKCHAIN UNTUK MENGATASI KEKURANGAN TEKNOLOGI PENGELOLA HAK CIPTA DIGITAL (DRM)*.

Publikasi berikut menjadi bagian dari Bab 2

Sitasi publikasi 1

Kontributor	Jenis Kontribusi
Jehan Afwazi Ahmad	Mereview dan menulis artikel
Teduh Dirgahayu	Memberi ide dan masukan

Halaman Kontribusi

Tidak ada kontribusi dari pihak lain

Halaman Persembahan

Dengan rasa syukur yang mendalam kepada Allah SWT, tesis ini saya persembahkan kepada:

1. Kedua orang tua saya, tanpa mu semua ini menjadi tidak berguna.
2. Istri saya tercinta, terimakasih atas semua dukungan yang tidak terhitung, *I love you.*
3. *My lovely* Humaira, ini salah satu hasil perjuangan bapak, semoga bisa jadi motivasi buat mu kelak saat dewasa.
4. Semua teman seperjuangan, pokoknya kalian luar biasa.
5. Semua citivitas akademika kampus UII, khususnya jurusan teknik informatika, semoga karya ini dapat menginspirasi.

Kata Pengantar

Assalamu'alaikum Wr. Wb.

Alhamdulillahirabbila'lamiin, Puji syukur kehadirat Allah SWT yang tidak henti melimpahkan rahmat dan hidayah-Nya, sehingga penyusunan tesis yang berjudul **Smart Contract Sebagai Proof-of-concept DRMChain Hak Cipta Buku Digital Ditinjau Dari Segi Hukum Positif Indonesia**, dapat diselesaikan secara baik tepat waktu.

Pada kesempatan ini saya ingin menyampaikan rasa terima kasih kepada semua pihak yang telah mendukung dan membantu menyelesaikan tugas akhir ini, khususnya kepada:

1. Kedua orang tua saya, yang telah memberikan banyak dukungan, mendo'akan, serta merestui setiap langkah dan kepetusan saya.
2. Istri dan anak saya tercinta yang telah memberikan semangat dan dukungan moral, sehingga dapat terus berjuang hingga akhir.
3. Dr. Raden Teduh Dirgahayu, S.T., M.Sc, sebagai dosen pembimbing yang telah bermurah hati membimbing dan mengarahkan saya dalam menyusun tesis ini.
4. Para dosen penguji proposal, progress, dan pendadaran yang memberikan kritikan dan masukan.
5. Sarah, yang memberikan dukungan, pikiran, kritikan, masukan, dan tenaganya hingga akhir.
6. Aziz, Fatma, Ting, Salmuasih, dan kawan-kawan pejuang tesis terakhir sebagai teman curhat, berbagi dan diskusi.
7. Segenap dosen Informatika UII yang telah mengajarkan banyak hal dan ilmu yang bermanfaat.
8. Teman seangkatan 18-1 Magister Teknik Informatika.
9. Kos Bernadia, dan semua Kafe daerah tiyasan jogja sebagai tempat yang nyaman untuk mengerjakan tesis.

Semoga Allah SWT membalas semua kebaikan kalian dan senantiasa melimpahkan rahmat dan kasih sayang-Nya kepada kita semua Amin.

Saya menyadari bahwa tesis ini masih memiliki banyak kekurangan dan keterbatasan baik dari penyusunan maupun tata bahasa yang disampaikan. Oleh karena itu, saya dengan rendah hati menerima kritik dan saran dari para pembaca agar dapat memperbaiki tesis ini.

Semoga tesis ini dapat memberikan manfaat bagi semua kalangan, khususnya kepada mahasiswa teknik informatika UII dan bidang sejenisnya. Saya juga berharap bahwa tesis ini dapat memberikan kontribusi terhadap ilmu pengetahuan, khususnya di bidang teknologi informasi dan komunikasi.

Wassalamu'alaikum Wr. Wb.

Yogyakarta, 24 Januari 2023

Daftar Isi

Daftar Gambar

Daftar Tabel

BAB I
Pendahuluan

1.1. Latar Belakang

Kebutuhan manusia akan informasi dan pesatnya perkembangan teknologi melahirkan era digital saat ini. Hadirnya era digital membawa implikasi yang luar biasa terhadap kehidupan manusia dalam berbagai sektor (Irawati, 2019). Salah satunya adalah perubahan gaya konsumsi manusia terhadap pemanfaatan dan penggunaan suatu karya cipta seperti buku. Pada era sebelumnya, buku dibuat dengan media kertas atau sejenisnya sehingga memiliki wujud fisik yang dapat disentuh. Namun, di era digital saat ini buku fisik telah banyak ditinggalkan dan beralih ke buku berbentuk digital. Hal ini karena buku digital memiliki beberapa kelebihan, yaitu: kemudahan akses, modifikasi, penyimpanan, dan distribusi. Wujud dari buku digital dapat berupa *ebook* dalam format PDF atau *Kindle*. Sedangkan pemanfaatan dan penggunaan buku digital membutuhkan suatu perangkat elektronik berupa komputer, *smartphone*, dan tablet yang memiliki spesifikasi tertentu.

Salah satu bagian penting yang tidak lepas dari lahirnya era digital adalah teknologi internet. Teknologi internet menjadi sarana informasi dan komunikasi yang menawarkan kemudahan dalam proses pertukaran data. Namun, teknologi internet juga memberikan dampak yang cukup besar terhadap eksistensi suatu karya cipta yang menjadi hak atas Kekayaan Intelektual (KI) bagi penciptanya. Berbagai aktivitas dapat dilakukan dalam memanfaatkan internet seperti mengunduh, mengunggah, berbagi file, dan sebagainya. Hal ini menimbulkan berbagai masalah hak cipta. Salah satu masalah yang muncul adalah pelanggaran hak cipta melalui pembajakan karya cipta. Hak Cipta merupakan bagian dari Hak Kekayaan Intelektual (HKI) yang timbul secara otomatis sesuai prinsip deklaratif dan menjadi dasar perlindungan hukum terhadap karya-karya hasil kemampuan intelektual manusia. HKI berperan penting dalam kehidupan masyarakat modern karena di dalamnya terkandung aspek ekonomi, seni, dan teknologi yang bernilai bagi manusia.

Persoalan pelanggaran hak cipta di Indonesia menyita perhatian yang cukup serius karena syarat kerugian yang sangat besar. Seperti diungkapkan oleh (Puput, 2017), pada tahun 2017 dirgen HKI menangani sebanyak 60 kasus pelanggaran hak cipta dengan perkiraan kerugian mencapai 100 milyar rupiah. Jumlah kasus ini kian bertambah hingga tahun 2021 pelanggaran teridentifikasi mencapai 243 kasus (Pangestu Pratama, 2021).

Badan Ekonomi Kreatif (Bekraf) menaksir kerugian negara akibat pembajakan mencapai angka yang fantastis yaitu 100 Trilun rupiah per tahun (Rachmawati, 2019). Berkaitan dengan buku digital, ketua Lembaga Manajemen Kolektif (LMK) Perkumpulan Reproduksi Cipta Indonesia (PRCI) Kartini Nurdin mengungkapkan, terdapat banyak buku digital berbentuk PDF yang dibagikan secara gratis atau dijual di berbagai platform *marketplace*. Riset yang dilakukan oleh Ikatan Penerbit Indonesia (IKAPI) pada tahun 2019 mengungkapkan, bahwa kerugian yang diterima oleh sebanyak 11 penerbit mencapai hingga 119 milyar rupiah yang sebagian besar penyebarannya melalui media online (Silfia, 2021). Banyaknya kasus pelanggaran hak cipta yang terjadi, Indonesia menyandang status *Priority Watch List* (PWL) yang membuat investasi menjadi terhambat. Meningkatnya kasus pelanggaran hak cipta ini tidak lain karena konten digital menawarkan kemudahan distribusi, penggandaan, dan modifikasi tanpa mengurangi kualitas asilnya (Simatupang, 2021). Oleh karena itu sudah semestinya diperlukan upaya hukum dan inovasi teknologi yang tepat untuk mencegah semakin maraknya kasus pelanggaran.

Kasus pembajakan buku digital dengan cara distribusi secara ilegal adalah kasus yang banyak ditemui pada media sosial dan *merketplace,* seperti diungkapkan oleh (Anshary & Labetubun, n.d.) dan (Simangunsong et al., 2020). Modus yang dilakukan adalah dengan membagikan buku secara gratis setelah pembeli bergabung atau mengikuti akun media sosial mereka. Sedangkan pada *merketplace,* modus yang dilakukan adalah dengan menggandakan dan menjual kembali buku asli yang dibeli. Kasus pembajakan lain adalah terkait plagiasi, klaim, dan penerjemahan yang diungkapkan oleh (Lauren, 2019) dan (Risky & Bintang, 2019) yang merujuk pada kasus yang dilakukan pemerintah aceh terhadap buku berjudul "Kilas Balik Pembangunan Aceh Setelah MoU Helsinki *Flashback on The Development of Aceh After Helinski MoU*" dan kasus plagiasi melalui situs fanfiksi oleh salah satu pengguna dengan menerbitkan ulang ke situs lain tanpa mencantumkan penulis aslinya. Disamping itu, kasus yang tidak kalah penting dan banyak dilakukan oleh masyarakat luas seperti diungkapkan oleh (Wicaksono & Urumsah, 2017) dan (Setiawan et al., n.d.) adalah mengunduh secara ilegal. Fenomena unduh secara ilegal ini tidak lepas dari peran pelaku pembajakan yang menyebarkan konten bajakan melalui situs tidak resmi dengan memanfaatkan *cyberlocker (file hosting)* (Mike, 2019). Meskipun secara eksplisit, aktivitas unduh tidak diatur dalam undang-undang. Akan tetapi, aktivitas unduh dikiaskan dengan proses penggandaan suatu ciptaan seperti dijelaskan pada pasal 1 ayat (12) UUHC (Abi Jam'an Kurnia, 2018). Dengan begitu seseorang yang mengunduh suatu konten ciptaan

secara ilegal masuk dalam unsur pasal 9 ayat (2) dan (3). Unduh secara ilegal termasuk pelanggaran hak moral terhadap penggunaan karya ciptaan tanpa izin. Tindakan pelanggaran dalam beberapa kasus yang diungkapkan tersebut terangkum dalam Tabel 1.1.1.

Tabel 1.1.1 Tindakan Pelanggaran Hak Cipta Digital

Jenis Tindakan	**Bentuk Tindakan**	**Hak Dilanggar**
Unduh	Mengunduh melalui situs/platform lain	Hak ekonomi: hak penggunaan / pemanfaatan
Modifikasi karya atau karya turunan	Memplagiasi, mengklaim, menerjemahkan	Hak ekonomi: hak pengumuman, penerjemahan Hak Moral: modifikasi dan distorsi ciptaan
Penyebaran / mengunggah	Menjual lewat platform media sosial atau *marketplace*	Hak ekonomi: hak pengumuman, komersialisasi ciptaan

Upaya hukum di Indonesia sebetulnya telah mengakomodir beberapa tindakan pelanggaran hak cipta digital. Upaya hukum tersebut bertumpu pada tiga pendekatan yang dikemukakan oleh Jacques de Werra, meliputi: (1) perlindungan dengan ketentuan hukum konvensional, (2) perlindungan dengan teknologi pengaman, (3) perlindungan hukum atas teknologi pengaman (Simatupang, 2021). Dalam Pasal Undang-Undang Hak Cipta (UUHC) Nomor 28 Tahun 2014 dijelaskan mengenai aspek hukum konvensional dan aspek perlindungnan hukum atas perlindungan teknologi pengaman. Sedangkan aspek teknologi pengaman sesuai penjelasan Pasal 53 UUHC menyatakan bahwa tidak ada batasan khusus dalam proses pengembangannya. Namun dalam pelaksanaanya, wajib memenuhi aturan perizinan dan persyaratan yang ditetapkan instansi berwenang serta kententuan lanjutan dari peraturan pemerintah baik berkaitan dengan sarana produksi, penyimpanan data, dan sebagainya. Sayangnya, upaya hukum perlindungan hak cipta digital yang diterapkan belum efektif menekan pelaku pelanggaran.

Salah satu cara efektif untuk menekan pelanggaran hak cipta dalam ranah digital adalah memberlakukan batasan melalui teknologi seperti *Digital Right Management* (DRM). DRM merupakan sebuah sistem keamanan yang memiliki mekanisme tertentu untuk

melindungi suatu asset digital yang bernilai (karya cipta) dengan cara mengontrol distribusi serta penggunaannya (Irawati, 2019). Karya cipta digital tersebut dapat berupa musik, sinematografi, ebook, perangkat lunak, dan sebagainya (Kementrian Hukum Hak Asasi Manusia Tim, 2020). Hingga saat ini banyak pegiat hak cipta dan pakar teknologi terus berupaya mengembangkan teknologi pengaman hak cipta. Namun sayangnya, kebanyakan teknologi DRM seperti *Silverlight*, *Flash Air*, *Real Network*, *Windows DRM*, dan *Apple DRM*, berfokus pada enkripsi konten digital dan manajemen lisensi sehingga memiliki beberapa kekurangan, di antaranya: (1) tidak mampu melacak kebocoran data dan membuktikan pelanggaran, (2) tidak dapat meng-autentikasi konten ketika sumber informasinya tidak lagi diketahui, baik disebabkan karena rusak atau dimanipulasi. (3) tidak dapat memberikan batasan akses bagi pengguna terhadap konten digital, (4) tidak dapat memindahkan hak kepemilikan suatu karya cipta kepada orang lain, (5) menggunakan model sentralisasi yang syarat dengan masalah kepercayaan (*trust issue*) (Garba et al., 2021). Padahal tujuan diterapkannya teknologi DRM selain untuk melindungi konten digital adalah untuk menjamin keamanan distribusi, memastikan originalitas konten, menyediakan transaksi *non-repudiation* (prinsip tak terbantahkan), serta dapat mendukung identifikasi pencipta (Simatupang, 2021). Dengan begitu teknologi DRM saat ini masih belum dapat mencapai tujuan yang dimaksud. Oleh kerena itu diperlukan suatu konsep inovasi teknologi yang mampu menyelesaikan masalah DRM demi mencapai sistem yang ideal. Secara ringkas DRM ideal harus memenuhi kriteria yang ditunjukkan pada Tabel 1.1.2.

Tabel 1.1.2 Kriteria DRM Ideal

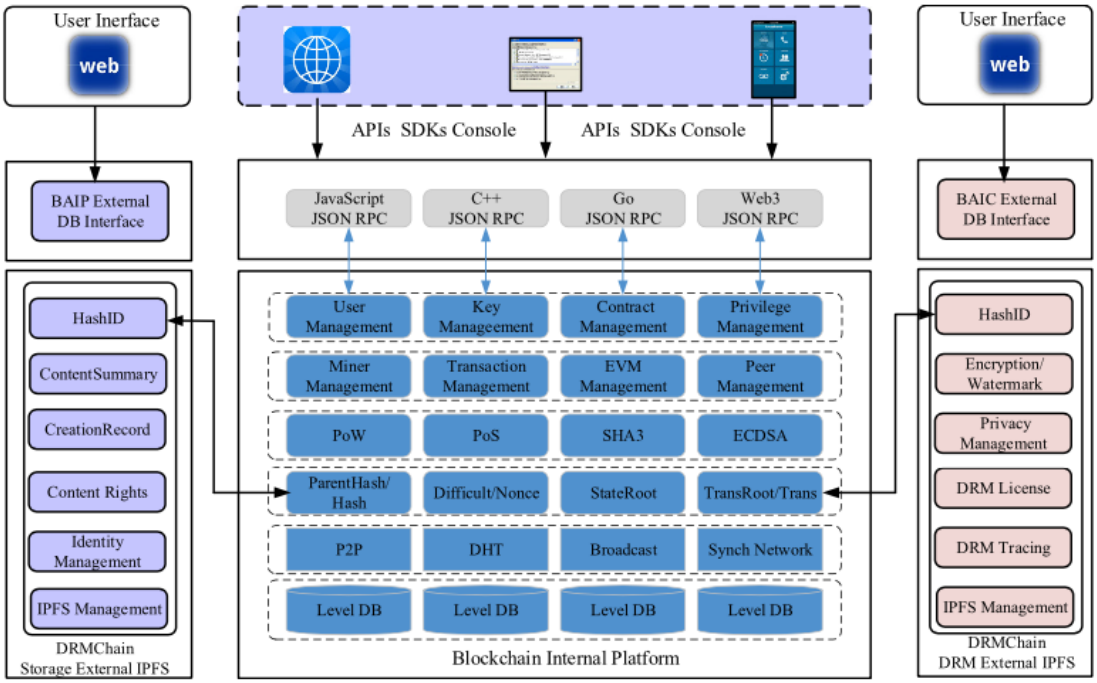
No.	Tindakan
1.	Mengidentifikasi pencipta
2.	Membuktikan keaslian
3.	Membuktikan pelanggaran
4.	Membatasi mengunduh
5.	Memindahkan hak kepemilikan

Menurut (Garba et al., 2021), *Blockchain* adalah salah satu teknologi yang secara teknis cocok digunakan untuk menyelesaikan permasalahan DRM. Hal ini karena Blockchain mengusung konsep sistem terdesentralisasi yang berjalan dalam lingkungan peer-to-peer (P2P). *Blockchain* sediri merupakan daftar catatan terdistribusi yang terus

berkembang secara historis seiring waktu (Zheng et al., 2018). *Blockchain* akan menyimpan setiap transaksinya ke dalam sebuah blok yang saling terhubung membentuk sebuah rantai yang setiap blok pada blockchain tersebut tersemat kode terenkripsi (*hash*) dari blok sebelumnya. Konsep terdesentralisasi pada blockchain memungkinkan tidak ada satu perangkat pusat yang akan mengontrol atau mengendalikan data, melainkan setiap blok yang dibuat akan disalin dan didistribusikan ke setiap pengguna sistem dalam waktu bersamaan sehingga otoritas sistem sepenuhnya dikendalikan oleh pengguna sistem (Kusumawardhani, 2021). Dengan mekanisme seperti itu, manipulasi data akan sulit dilakukan, pendistribusian data menjadi jelas, dan setiap transaksi akan ter-*monitor* dan mudah untuk dilacak (*traceable*).

Terdapat satu bagian penting yang tidak dapat dipisahkan dalam diskusi mengenai *blockchain* yaitu teknologi *Smart Contract*. *Smart Contract* merupakan sebuah program yang berjalan pada jaringan blockchain dan dieksekusi secara otomatis seketika kondisi tertentu terpenuhi. Beberapa hal yang dapat diterapkan pada *smart contract* adalah melakukan proses *agreement*, transfer dana, pencairan dana, pendaftaran kendaraan, pengiriman notifikasi, menerbitkan tiket, voting, dan sebagainya. Berkaitan dengan Hak Cipta, smart contract dapat digunakan untuk memberikan hak akses, mengubah hak kepemilikan karya cipta, mendistribusikan *reward*/Royalti, bahkan dapat memberikan penalti tertentu pengguna akun sesuai ketentuan yang dibuat dalam program.

Salah satu penelitian oleh (Ma et al., 2018) telah mengusulkan sebuah konsep yang menyatukan teknologi DRM dan *Blockchain* yang disebut DRMChain. Konsep DRMChain yang diusulkan secara umum terbagi atas tiga komponen utama, yaitu: jaringan IPFS *(InterPlanetary File System)*, platform *Blockchain*, dan platform DRM. Implementasi DRMChain dilakukan dengan menggunakan dua aplikasi antarmuka *blockchain*, yaitu *Application Interface Plain* (BAIP) untuk menangani ringkasan metadata konten asli dan *Application Interface Chiper* (BAIC) untuk menangani layanan DRM. Kedua antarmuka tersebut akan saling terhubung dengan platform *blockchain* melalui hash dari konten. Skema *blockchain* dari penelitian tersebut secara utuh disajikan pada Gambar 1.1.1 Model DRMChain (Ma et al., 2018).



Gambar 1.1.1 Model DRMChain (Ma et al., 2018)

Sebelum menjadi konsumsi bisnis, metadata dari konten asli dibuat dan disimpan ke dalam jaringan P2P *blockchain* melalui BAIP. Konten asli tersebut menjadi data utama yang akan digunakan dalam pemrosesan DRM dan sebagai bukti keaslian konten. Selanjutnya, data konten asli akan dienkripsi dan disimpan ke dalam penyimpanan eksternal IPFS melalui BAIC yang terdiri dari tiga jenis informasi di antaranya adalah informasi kepemilikan, hak cipta, dan lisensi. Kemudian platform blockchain akan memverifikasi dan mengkonfirmasi transaksi terkait dan menyimpannya ke dalam blockchain.

Dari penjelasan yang diuraikan tersebut, secara ringkas diketahui bahwa banyak kasus pelanggaran hak cipta buku digital di Indonesia yang belum ditegakkan dengan sistem hukum konvensional. Bahkan teknologi DRM juga belum secara efektif dapat menyelesaikan masalah tersebut karena memiliki beberapa kekurangan. Maka dari itu, gabungan teknologi DRM dan blockchain (DRMChain) dibuat untuk menyelesaikan permasalahan tersebut. Namun, belum ditemukan bukti bahwa teknologi tersebut mampu menjadi solusi permasalahan hak cipta di Indonesia. Oleh karena itu, penelitian ini dilakukan untuk memberikan bukti apakah sistem DRMChain dapat menyelesaikan permasalahan hak cipta buku digital di Indonesia sesuai hukum positif yang berlaku. Pembuktian dalam penelitian ini menggunakan metode *Proof-of-Concept* (PoC) dengan melibatkan pengujian tertentu untuk menunjukkan kelayakan dari teori/konsep yang dibuat.

Dari penelusuran makalah yang dilakukan, penelitian terkait perlindungan hak cipta digital di Indonesia dengan teknologi pengaman (DRM) masih jarang ditemui. Khususnya teknologi pengaman yang menerapkan *blockchain* sebagai basis teknologinya. Adanya penelitian ini, diharapkan dapat memberikan solusi yang efektif dan efisien bagi pihak-pihak yang berkepentingan seperti pencipta, penerbit, atau lembaga hak cipta, untuk menyelesaikan permasalahn hak cipta buku digital di Indonesia dengan menerapkan teknologi pengaman. Kemudian, penelitian ini diharapkan dapat memberikan masukan kepada pengembang perangkat lunak dalam mengembangkan teknologi pengaman hak cipta berbasis *blockchain*. Meskipun begitu, penelitian ini tentu memiliki keterbatasan yang mungkin tidak dapat diterapkan sepenuhnya untuk menyelesaikan permasalahan hak cipta dalam berbagai kasus dan bentuk media digital. Hal ini karena objek penelitian terbatas pada karya cipta buku digital dalam format PDF, begitu juga kasus yang diterapkan terbatas pada kasus yang terindetifikasi.

1.2. Rumusan Masalah

Berdasarkan uraian pada latar belakang, maka masalah yang menjadi fokus dalam penelitian ini adalah *"Membuktikan apakah konsep DRMChain dapat menyelesaikan permasalahan hak cipta buku digital ditinjau dari segi hukum positif Indonesia yang berlaku?"*

1.3. Batasan Masalah

Dalam melaksanakan penelitian ini, agar materi atau isi dalam tulisan tidak menyimpang dari pokok permasalahan, maka ditentukan batasan sebagai berikut:

1. Penelitian hanya dilakukan pada buku digital berupa PDF.
2. Implementasi aplikasi berbentuk aplikasi Smart Contract.
3. Pengujian akan dilakukan pada hasil identifikasi pelanggaran hak cipta digital di Indonesia secara *blackbox* pada aplikasi.
4. Skema model utama yang dibuat akan merujuk pada model penelitian oleh (Ma et al., 2018).

1.4. Tujuan Penelitian

Berdasarkan uraian latar belakang, penelitian ini bertujuan untuk *"Membuktikan bahwa konsep DRMChain dapat menyelesaikan masalah hak cipta dokumen digital di Indonesia sesuai hukum positif yang berlaku".*

1.5. Manfaat Penelitian

Hasil dari penelitian ini akan memberikan manfaat barupa:

1. Sebagai solusi yang efektif dan efisien dalam menyelesaikan permasalah hak cipta digital di Indonesia dengan teknologi pengaman.
2. Sebagai ilmu pengetahuan yang dapat digunakan para pengembang untuk mengembangkan perangkat lunak teknologi pengaman perlindungan karya cipta digital berbasis *blockchain*.

1.6. Sistematika Penulisan

Sistematika penulisan pada penelitian ini, sebagai berikut:

1. Bab 1 Pendahuluan

   Pada bab ini membahas mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

2. Bab 2 Tinjauan Pustaka

   Bab ini membahas mengenai studi literatur terkait DRM, Blockchain, DRMChain, dan berbagai permasalah hak cipta dan hukum positif hak cipta di Indonesia dari berbagai sumber referensi dan jurnal terkait. Bab ini juga akan membahas mengenai teori yang menjadi landasan dalam penelitian ini.

3. Bab 3 Metodologi Penelitian

   Bab ini akan membahas mengenai langkah-langkah penelitian yang digunakan berupa identifikasi permasalahan HKI, pembuatan aplikasi serta pengujian, dan evaluasi.

4. Bab 4 Perancangan Sistem

   Bab ini akan membahas mengenai hasil perancangan sistem dari penelitian yang dilakukan.

5. Bab 5 Implementasi dan Pembahasan

   Bab ini akan membahas mengenai implementasi, pengujian, dan evaluasi sistem.

6. Bab 6 Kesimpulan dan Saran

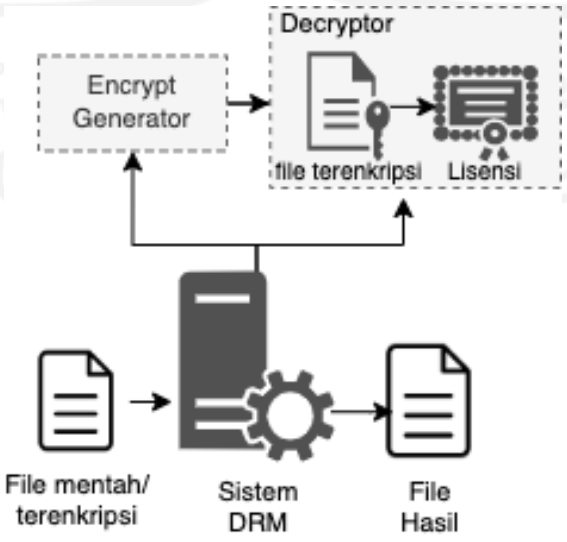
   Bab ini akan berisi kesimpulan hasil akhir keseluruhan proses analisis penelitian yang dihasilkan.

BAB 2
Tinjauan Pustaka

Penelitian ini dimulai dari identifikasi masalah yang berkaitan dengan HKI digital di Indonesia. Beberapa literatur yang merujuk kepada beberapa permasalahn hukum HKI digital, teknologi pengamanan digital dan pemanfaatan teknologi untuk membuktikan teknologi DRMChain dapat menyelesaikan permasalahan hak cipta digital di Indonesia. Selain itu, akan dijelaskan juga mengenai landasan penelitian yang digunakan dalam penelitian ini.

2.1. Digital Right Management (DRM)

DRM merupakan teknologi pengamanan digital yang digunakan untuk melindungi aset digital (karya cipta) dengan cara mengontrol distribusi serta penggunaan aset digital (Irawati 2019). DRM adalah sekumpulan sistem yang melindungi hak cipta berupa konten digital, seperti dokumen, musik, video dan konten elektronik yang tersimpan dan dapat dipindahkan melalui media elektronik (Simatupang 2021). DRM merupakan kombinasi dari teknologi, hukum dan kebijakan hak cipta serta model bisnis yang dibuat untuk mengontrol pendistribusian HKI. Tujuan penerapan DRM adalah memberikan perlindungan konten digital, memastikan keaslian konten, menyediakan transaksi *non-repudiation* (tak terbantahkan) dan mendukung proses identifikasi sumber konten. Secara sederhana arsitektur sistem DRM ditunjukkan pada Gambar 2.1.1 .



Gambar 2.1.1 Arsitektur Sistem DRM

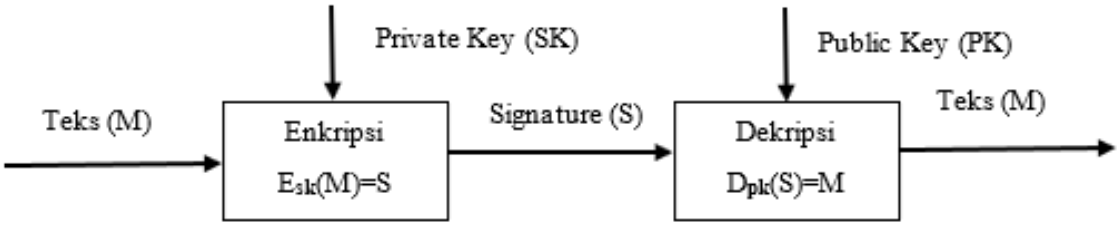
Terdapat dua proses dalam arsitektur sistem DRM, yaitu proses enkripsi file dan dekripsi file. Pada proses enkripsi, masukan berupa file mentah yang kemudian diolah pada mesin encrypt generator yang menghasilkan file terenkripsi dan lisensi. File terenkripsi adalah file yang sudah dienkripsi dengan kunci tertentu, sedang file lisensi adalah file yang berisi informasi yang umumnya berupa metadata, kunci, dan aturan hak akses dari file yang dilisensi. Pada proses dekripsi, masukan berupa file terenkripsi dan lisensi yang selanjutnya diproses pada mesin decryptor dan menghasilkan file yang diharapkan. Pada praktiknya, implementasi arsitektur sistem DRM ini dapat dilakukan dengan cara yang berbeda-beda, tapi konsep mekanismenya pada umumnya tetap sama.

Berdasarkan fungsinya, sistem DRM terbagi menjadi 3 bagian (Muslim Ijtihadie, Ciptaningtyas, and Zabo n.d.), yaitu: (i) DRM manager yang bertugas sebagai melakukan validasi dan dekripsi konten, (ii) DRM security bertugas untuk menangani proses kriptografi pada konten, (iii) aplikasi DRM yang bertugas untuk menjalankan (rendering) konten. Dalam hukum perlindungan teknologi, DRM atau bisa disebut juga Technologies Protection Measure (TPM) memiliki beberapa metode untuk melakukan kontrol dan menjaga keamanan data (Simatupang 2021).

2.1.1. Encryption

Encryption adalah suatu algoritma kriptografi untuk mengkodekan suatu konten dengan sebuah sandi. Algoritma ini secara umum akan mengacak data sehingga menjaga kerahasian sebuah data/pesan. Lawan dari Encryption adalah Decryption yang merupakan proses kebalikan dari proses encryption yang akan mengembalikan data terenkripsi menjadi data asli.

Selain teknik enkripsi yang telah dijelaskan, terdapat pula teknik enkripsi dengan menggunakan sepasang kunci public (PK) dan private (SC) untuk melakukan kriptografi yang biasa disebut kriptografi kunci publik. PK dan SC memiliki korespondensi yang sama, sehingga sebuah pesan yang dienkripsi dengan SC dapat didekripsi dengan PK. Dengan metode ini selain dapat menjaga kerahasiaan sebuah pesan, juga dapat digunakan untuk mengotorisasi pesan serta memastikan bahwa pesan tersebut tidak dapat disangkal oleh pengirim maupun penerima (*non-repudation*). Secara teknis, proses kriptografi public key dan private key ditunjukan pada Gambar 2.1.2



Gambar 2.1.2 Kriptografi Kunci Publik

Sebuah pesan akan dienkripsi oleh pengirim pesan menggunakan SC, kemudian pesan tersebut akan diteruskan kepada penerima pesan. Penerima pesan dapat membuka pesan tersebut jika hanya pengirim pesan memberikan PK kepada penerima pesan. Dengan cara tersebut, maka seorang pengirim pesan tidak akan bisa menyangkal bahwa dia benar-benar mengirim pesan.

2.1.2. Watermark

Watermark adalah proses menyisipkan atau menempatkan sebuah informasi ke dalam sebuah data baik berupa dokumen, gambar, musik atau video yang berisi informasi tertentu tanpa mengubah kualitas dari konten. Terdapat beberapa syarat umum untuk melakukan teknik watermarking, antara lain (Saelan, Bandung, and Bandung 2011):

1. *Imperceptible*, *watermark* yang disisipkan tidak bisa dideteksi panca indra manusia
2. *Robustness*, *watermark* yang disispuan tahan dari kerusakan baik karena proses perubahan atau manipulasi serta tidak mudah untuk diekstrak.
3. *Secure, watermark* yang disisipkan seharusnya tidak mudah diekstrak, dihilangkan, atau dirusak.

Kategori *watermarking* jika dilihat dari kenampakan secara kasat mata terbagi menjadi dua hal, yaitu *visible watermarking* dan *invisible watermarking*. Penjelasan terkait hal ini, sebagai berikut:

1. *Visible watermarking*

Pada kategori ini memungkinkan panca indra dapat melihat *watermark* pada sebuah dokumen. Hal ini sengaja ditampakan karena alasan tertentu. Dengan begitu *watermark* ini tidak memenuhi syarat *imperceptible*. *Watermark* jenis ini memiliki kelemahan yaitu mudah dihapus karena kebanyakan memiliki kontras yang berbeda yang diberi *watermark*, sehingga tidak memenuhi syarat *secure*. Namun, *watermark* jenis ini memenuhi syarat *robusness*, karena memiliki ketahanan yang bagus.

2. Invisible watermarking

Kebalikan dengan *visible watermarking*, *watermark* jenis ini tidak dapat dilihat oleh panca indra. *Watermark* jenis ini lebih sulit dibuat, karena harus mempertahankan kualitas dokumen dan harus mempertimbangakan persyaratan *imperceptible*, *secure* dan *robusness*. *Watermark* jenis ini memiliki ketahanan yang kurang bagus.

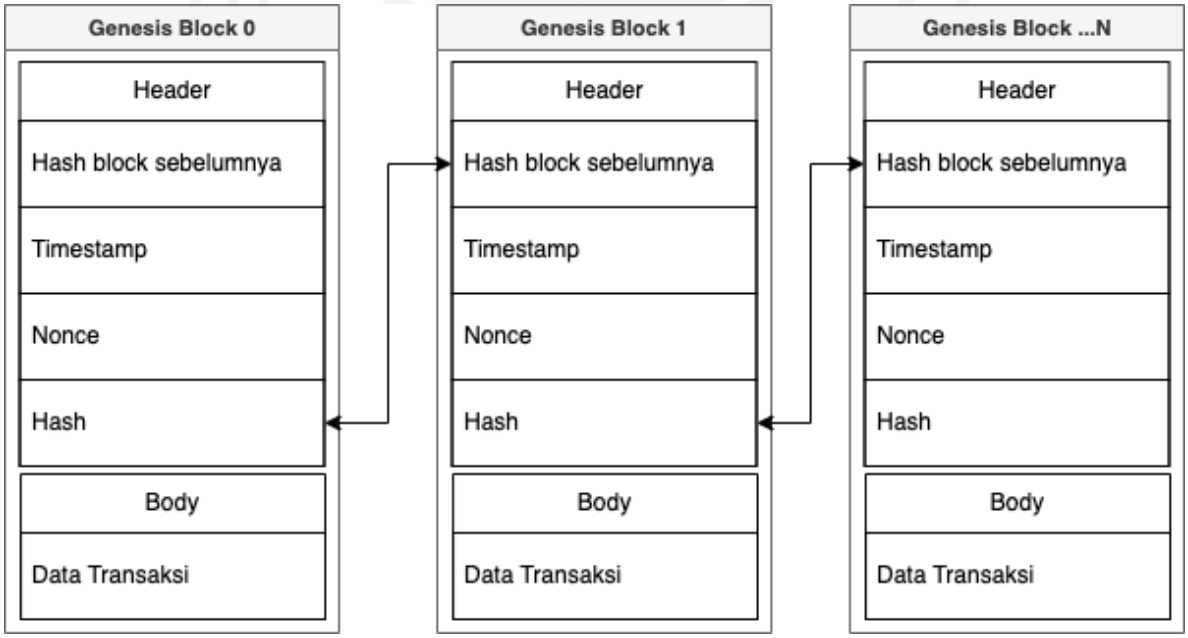
Dari penjelasan mengenai sistem DRM diketahui bahwa komponen yang bekerja pada sistem DRM hanya dapat digunakan untuk melindungi konten digital dengan proses enkripsi untuk kepentingan autentikasi dan pemberian *watermark* untuk melacak dan membuktikan keaslian (Ma et al. 2018). Melacak dan membuktikan keaslian ini dapat dilakukan secara manual dengan melihat tanda lisensinya (*watermark*) atau meng-autentikasi melalui sistem DRM. Namun, ketika konten tersebut telah dimanipulasi baik dengan cara merusak atau menghilangkan tanda lisensinya, maka sistem DRM tidak lagi berfungsi karena sumber konten tidak dapat dikenali atau di-autentikasi. Dengan begitu, penyebaran konten menjadi susah dikendalikan (Garba et al. 2021). Peran *non-repudiation* yang diterapkan pada sistem DRM ini masih dilakukan oleh perantara (*arbitrase*) atau dapat juga dilakukan dengan mekanisme kriptografi kunci publik jika hanya mekanisme tersebut diterapkan. Di samping itu, teknologi DRM yang beredar saat ini hanya berfokus pada enkripsi konten dan managemen lisensi, sehingga tidak ada cara untuk memeriksa dan melacak pelaku pelanggaran (Ma et al. 2018). Selain itu, fasilitas untuk memindahkan hak kepemilikan baik dengan meminjamkan, memberikan, atau menjual konten tidak dapat dilakukan dengan sistem DRM yang ada, padahal hal tersebut merupakan hal yang penting (Rosenblatt 2018). Di samping hal tersebut, sistem DRM yang ada saat ini umumnya diterapkan pada sistem terpusat yang mensyaratkan informasi data dan transaksi tidak transparan dan terdapat kemungkinan terjadi kerusakan pusat sistem baik karena kegagalan sistem ataupun serangan cyber (Zhang and Zhao 2018).

2.2. Blockchain

Penerapan teknologi blockchain secara teknis dapat digunakan untuk melengkapi kekurangan DRM dengan mengadopsi beberapa karakteristiknya. Salah satunya adalah karakter terdesentralisasi. Karakter ini memungkinkan data transaksi disebarkan ke setiap pengguna sistem melalui jaringan P2P (Ma et al., 2018). Dengan konsep ini, tidak ada pusat

sistem yang akan mengendalikan data (*ownerless*) sehingga apabila terjadi kegagalan sistem pada salah satu pengguna tidak akan berdampak pada pengguna lain. Selain itu juga, penerapan konsep ini akan meningkatkan kepercayaan pengguna terhadap kontrol sistem.

Dalam sistem blockchain, mekanisme konsensus atau kesepakatan bersama untuk menambahkan blok ke dalam blockchain diwakili oleh node. Node merupakan sebuah komputer yang terhubung dengan jaringan blockchain (Rezkitha 2021). Node tersebut akan berkomunikasi dan saling berbagi informasi dengan komputer lain yang terhubung. Peran utama node adalah menyebarkan dan memvalidasi transaksi, menyimpan riwayat transaksi blockchain, dan menjaga konsensus satu sama lain. Setiap transaksi yang selesai dibuat akan diterima oleh node kemudian disebarkan ke setiap node lain melalui jaringan peer-to-peer (P2P) (Ma et al. 2018). Transaksi yang diterima oleh node, kemudian akan divalidasi dan disepakati sebelum dibuat menjadi blok baru. Transaksi tersebut dianggap gagal apabila lebih dari 50% node yang berpartisi tidak menyepakatinya. Semakin banyak node yang berpartisipasi dan mengkonfirmasi transaksi, keamanan dan integritas data menjadi semakin kuat.



Gambar 2.2.1 Arsitektur Blockchain

Secara umum arsitektur blockchain ditunjukkan pada Gambar 2.1.1. Terdiri dari sejumlah blok yang terkait satu dengan yang lain. Setiap blockchain memiliki blok pertama yang disebut genesis blok sebagai acuan referensi blok setelahnya. Struktur blok dalam blockchain terdiri dari *block header* dan *body*. *Block header* berguna untuk identifikasi blok

tertentu di dalam keseluruhan blockchain (Lee 2019). Komponen *block header* terdiri dari beberapa komponen antara lain:

1. Hash block sebelumnya, adalah sebuah kode unik yang digunakan untuk mereferensi blok sebelumnya pada blockchain.
2. Timestamp, adalah nilai yang menunjukkan waktu ketika blok ditambahkan ke blockchain.
3. Nonce (*Number Use Only One*), adalah nilai atau nomor dari proses penambangan (mining) yang digunakan untuk mendapatkan hash block yang diterima.
4. Hash, adalah sebuah kode unik yang mengidentifikasi blok pada blockchain.

Kode hash pada sebuah blok dibuat melalui fungsi hash. Fungsi hash akan memetakan sejumlah data menjadi satu string karakter yang bersifat deterministik. Block hash hasil dari fungsi hash ini kemudian disimpan atau disematkan ke dalam blok selanjutnya. Apabila data di dalam blok sebelumnya diubah, maka hash dari blok sebelumnya juga akan berubah sehingga blok yang mengikutinya menjadi tidak valid atau terputus. Dengan mekanisme rumit seperti ini, data yang telah disematkan ke dalam blockchain sangat sulit untuk diubah (Chingath and Babu 2020). Kemampuan seperti ini membuat blockchain disebut bersifat immutable. Kelebihan yang dihasilkan dari sifat ini adalah kemampuannya untuk mendeteksi gangguan yang menyebabkan kerusakan data (Tamper-proof). Sebagaimana dijelaskan bahwa hash adalah sebuah kode unik yang diumpamakan sebagai rantai blok di dalam blockchain. Apabila terdapat seseorang berusaha untuk mengubah data pada salah satu block, maka dia harus mengubah setiap hash pada blok yang mengikutinya. Belum juga dia harus mengubahnya pada seluruh jaringan blockchain yang terhubung. Oleh karena itu, setiap gangguan yang dapat menyebabkan kerusakan menjadi mudah untuk dideteksi.

Dalam mencetak sebuah hash, fungsi hash memiliki beberapa karakteristik yaitu: (i) deterministik artinya sebuah data atau pesan yang sama akan selalu menghasilkan kode hash yang sama, (ii) proses searah artinya ketika sebuah pesan dienkripsi akan tidak dapat dikembalikan ke data semula, (iii) Perbedaan satu karakter pada sebuah pesan akan selalu menghasilkan kode hash yang berbeda sahingga kecil kemungkinan akan menghasilkan kode hash yang sama. Dengan cara ini, akan sulit membedakan antara kode hash yang satu dengan yang lain.

Salah satu proses di dalam kerja blockchain adalah proses mining (penembangan). Mining adalah sebuah proses komputasi matematis yang dilakukan oleh mesin (miner) untuk

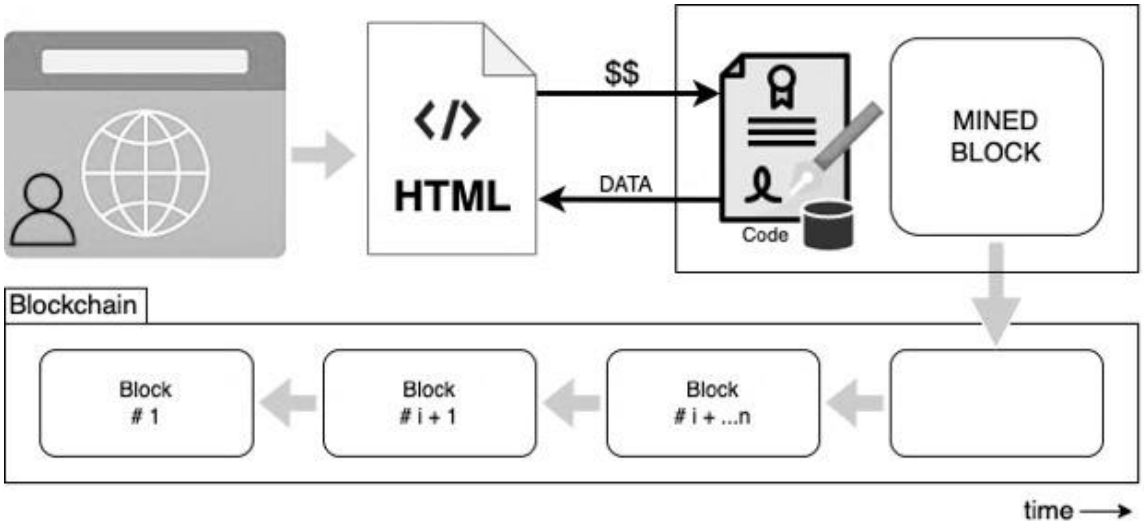
mendapatkan kode hash yang terverifikasi sesuai kriteria yang ditentukan. Sedangkan proses untuk mendapatkan kode hash dengan mekanisme konsensus dalam jaringan blockchain tersebut disebut *proof of work* (PoW) (Hamilton 2021). Proses penambangan dengan PoW membutuhkan perangkat daya dengan komputasi yang tinggi, semakin banyak perangkat yang digunakan untuk komputasi semakin besar pula peluang untuk mendapatkan hash sebagai blok baru. Oleh karena PoW memiliki beberapa masalah terkait keamanan serta memakan konsumsi energi yang sangat besar, algoritma Proof-of-Stack (PoS) dibuat sebagai alternatif mekanisme konsensus yang dapat digunakan. Dengan mekanisme PoS penambang diharuskan mempertaruhkan sejumlah asset koin mereka agar dapat menjadi validator untuk menghasilkan blok baru. Semakin besar jumlah koin yang dipertaruhkan, semakin besar peluang untuk menjadi validator. Apabila validator berhasil memvalidasi transaksi akan diberi imbalan berupa koin. Sebaliknya, jika validator tidak berhasil memvalidasi transaksi, koin yang dipertaruhkan akan dipotong dan validator tidak dapat menjadi validator untuk beberapa waktu tertentu.

2.2.1. Smart Contract

Smart Contract (SC) adalah sebuah program yang disimpan dalam blockchain yang secara otomatis dieksekusi ketika sebuah kondisi yang ditentukan terpenuhi dan terverifikasi (Garba et al. 2021). Secara teknis SC menggunakan konsep "jika…maka" dalam mengeksekusi sebuah transaksi. Adanya SC menghilangkan proses tradisional yang mensyaratkan formalitas dengan birokrasi administrasi yang panjang tanpa mengurangi keaslian dan kredibilitas. Keuntungan dari SC ini yaitu: (i) mengurangi ketergantungan pada pihak ketiga (perantara), (ii) proses transaksi realtime, artinya transaksi yang dilakukan memiliki waktu yang singkat setelah kriteria terpenuhi, (iii) transparansi dan keamanan transaksi, oleh karena SC didasarkan pada blockchain yang menjamin data bersifat transparan dan immutable, memungkinkan kesepakatan yang dibuat dapat dilakukan tanpa perlu mengenal satu sama lain. Hal ini juga dapat menghindari terjadinya pelanggaran seperti manipulasi klausul kontrak atau kesalahan dalam pengelolaan dan pelaksanaan kesepakatan.

Sebagaimana SC adalah sebuah program yang berarti sebuah kode yang disimpan, diverifikasi, dan dieksekusi di dalam platform blockchain. Selain itu kode ini yang digunakan untuk mengeksekusi dan menyelesaikan sebuah ketentuan dalam kontrak kesepakatan. Meskipun SC dieksekusi secara otomatis oleh komputer, beberapa bagian SC

mensyaratkan masukan dari pengguna / manusia. Secara umum sistem SC ditunjukkan pada Gambar 2.2.2.



Gambar 2.2.2 Sistem Smart Contract

SC terdiri dari saldo akun, penyimpanan personal (*personal storage*), dan *excutable code* (Alharby and Moorsel 2017). Bagian-bagian SC tersebut akan saling berkolaborasi sesuai dengan fungsinya. SC dapat membaca dan menuliskan data ke dalam penyimpanan personal, menyimpan uang di dalam kontrak saldo, dan juga dapat membuat kontrak baru. Selain itu, dengan kemampuannya untuk mengidentifikasi pengguna melalui alamat akun, SC dapat memberikan kontrol akses kepada pengguna. Secara lebih jauh, di dalam SC terdapat sebuah *state* yang nilainya akan berubah ketika suatu transaksi dilakukan. Misalnya, seseorang mengirimkan sejumlah uang ke rekening orang lain, maka akan terjadi perubahan nilai *state* pada kedua sisi orang tersebut, yaitu dari sisi pengirim dan penerima. *State* terdiri dari penyimpanan personal dan saldo kontrak yang disimpan di dalam salah satu blok dan akan diperbarui setiap kali kontrak dipanggil. Setiap kontrak yang dibuat (di-*deploy*) di dalam blockchain, akan ditetapkan dengan sebuah alamat sepanjang 20 *bytes* yang tidak dapat diubah. Untuk menjalankan sebuah kontrak, secara sederhana pengguna harus membuat sebuah transaksi kepada alamat kontrak tersebut. Setelah transaksi tersebut dibuat, kemudian akan dieksekusi oleh setiap node yang terlibat dalam jaringan blockchain dengan menyebarkan ke setiap node dalam jaringan dan divalidasi, sedemikian *state* kontrak juga diperbaharui.

Menurut (Sillaber and Waltl 2017) terdapat 4 fase yang terjadi dalam siklus SC:

1. Penciptaan (*Creation*), fase ini dibagi menjadi dua tahapan yaitu negosiasi dan implementasi. Tahap negosiasi akan dilakukan pertama kali bersama dengan para pihak terkait untuk menyepakati isi dan tujuan kontraknya. Pada tahap implementasi, kontrak yang telah disepakati tersebut diubang menjadi sebuah kode. Tahapan ini akan terus berulang sampai kesepakatan kontrak dan implementasi kode yang dibuat selesai. Hal ini dilakukan, karena ketika SC telah di-*deploy* di blockchain, kontrak tersebut tidak dapat diperbaharui.
2. Pembekuan (*Freeze*), fase ini dimulai setelah SC di-*deploy* ke dalam blockchain. Dalam fase ini SC bersifat publik sehingga para pihak yang terlibat dapat melakukan transaksi seperti pada umumnya. Transaksi SC dalam blockchain menggunakan metode escrow yang memberlakukan penahanan pembayaran sampai seluruh klausul pada kontrak terpenuhi dan node berperan untuk memastikan semua prasyarat kontrak terpenuhi.
3. Pelaksanaan (*Execute*), setelah semua prasayarat terpenuhi, selanjutnya kontrak disimpan di dalam blockchain dan didistribusikan pada semua node. Setelah itu, kontrak akan divalidasi dan mesin penerjemah SC akan mengeksekusi kode. Eksekusi SC menghasilkan sebuah transaksi baru dan state baru. Hasil dari SC dan state selanjutnya diserahkan ke node blockchain yang selanjutnya dilakukan konsensus untuk menghasilkan blok baru.
4. Finalisasi (*Finalize*), setelah SC dieksekusi, asset digital dipindahkan, dan semua transaksi telah dikonfirmasi, maka kontrak telah terpenuhi.

Selain itu, sistem kerja SC yang melibatkan blockchain dalam suatu transaksi terbagi menjadi dua (Oktaviani and Kenotariatan 2021):

1. On-Chain, merupakan transaksi SC yang terjadi di dalam blockchain. Transaksi ini sama halnya yang telah dijelaskan pada siklus SC. Berawal dari penciptaan, pembekuan, pelaksanaan, dan finalisasi. Artinya semua transaksi yang terjadi tidak melibatkan pihak di luar blockchain.
2. Off-Chain, berbeda dengan transaksi on-chain, transaksi ini melibatkan pihak ketiga, seperti konversi uang crypto ke nilai rupiah, memberikan *watermark* pada sebuah dokumen, dan sebagainya. Hal ini umumnya dilakukan sebelum informasi tersebut digunakan oleh SC. Atau dapat pula informasi tersebut diverfikasi atau disaring melalui

suatu perangkat yang disebut *oracle* sebelum selanjutnya dinyatakan dapat memasuki blockchain dan digunakan oleh SC.

2.2.2. InterPlanetary File System (IPFS)

IPFS merupakan filesystem yang mengusung sistem terdesentralisasi yang terinsipirasi dari beberapa ide sukses penerapan sistem P2P seperti BitTorrent, Git, DHTs, dan SFS (Benet 2014). IPFS adalah salah satu teknologi pendukung blockchain yang secara spesifik menangani penyimpanan konten media dan referensi yang terdesentralisasi dan dikhususkan untuk blockchain Ethereum (Apa Itu IPFS Dan Kegunaannya Di NFT Project - Diginews.id 2022). Konten yang dapat disimpan dan dibagikan pada IPFS ini antara lain, video, audio, dan teks dokumen. Sistem penyimpanan dan pendistribusian pada IPFS tidak tergantung pada satu node, melainkan disebarkan sejumlah node sehingga aplikasi menjadi lebih aman, cepat, tahan, handal, dan transparan. Response yang dikembalikan setelah mengunggah file ke IPFS berupa hash.

2.3. DRMChain

DRMChain adalah sebuah konsep yang menyatukan teknologi DRM dan Blockchain. Penelitian atau framework DRMChain yang diusulkan oleh (Ma et al., 2018) dan (Garba et al., 2021) bertujuan untuk mencegah penyebaran konten ilegal, mengontrol konten yang diakses, melindungi hak cipta konten, perlindungan privasi, lisensi, dan pelacakan penyalahgunaan konten. Komponen yang digunakan dalam penelitian mereka terdiri dari blockchain platform dan penggunaan cloud storage IPFS sebagai penyimpanan data/konten digital. Keduanya juga menggunakan hashID metadata konten untuk menghubungkan IPFS dengan platform blockchain. Hanya saja penelian oleh (Garba et al., 2021) lebih berfokus pada masalah skalabilitas transaksi untuk mencegah kemacetan transaksi pada transaksi yang memiliki intensitas dan skala yang banyak.

Dalam rangka mengkonfirmasi atau membuktikan keaslian konten serta agar dapat dilakukan pelacakan pelanggaran HKI terhadap konten tersebut, dilakukan pinyisipan *watermark* pada konten asli. Dalam penelitian (Garba et al., 2021), *Watermark* yang disisipkan dienkripsi terlebih dahulu sebelum kemudian disisipkan pada konten asli. Hal ini dilakukan untuk meningkatkan keamanan dan integritas konten. Autentikasi dokumen digital yang digunakan adalah *digital signature* dengan *Elliptic Curve Digital Signature Algotitm* (ECDSA). Auntentkasi dokumen ini bertujuan untuk memberikan akses kepada

pengguna yang tepat. Untuk proses hash untuk menghasilkan hash unik dari data pada sebuah blok digunakan algoritma *Secure Hash Algorithm* (SHA-256).

2.4. Dasar Hukum

Hak Cipta adalah hak eksklusif pencipta terhadap karya-karyannya yang secara otomatis timbul sesuai prinsip deklaratif. Hak Cipta sendiri merupakan bagian dari Hak Kekayaan Intelektual (HKI) yang menjadi dasar perlindungan hukum terhadap karya-karya dari hasil kemampuan intelektual manusia. Hal itu menjadi sangat penting karena menjadi jaminan keamanan nilai dari suatu karya dan bukti dari peradaban dan martabat manusia (Nareswari Manuaba & Sukihana, 2020). Hak Cipta jika ditinjau dari segi hak-hak yang tercakup di dalamnya merupakan hak eksklusif yang terdiri dari hak moral dan hak ekonomi (Kementrian Hukum Hak Asasi Manusia Tim, 2020). Hak moral adalah hak yang mutlak melekat pada diri pencipta sehingga secara umum tidak dapat dipindahalihkan kepada pihak lain kecuali dengan kondisi tertentu. Sedang hak ekonomi adalah hak yang bertujuan untuk pemanfaatan atau komersialisasi. Hak ekonomi ini secara otomatis juga melekat pada diri pencipta, akan tetapi dapat dipindahalihkan haknya kepada pihak lain. Untuk mendapatkan hak ekonomi, diperlukan izin dari pencipta atau pemegang hak cipta. Izin tersebut berupa lisensi. Lisensi ini menjadi pegangan utama bagi pihak yang diberi izin untuk dapat memanfaatkan, mengumumkan, memperbanyak, atau mengedarkan karya dari penciptanya berdasarkan kesepakatan tertentu.

Hukum positif perlindungan hak cipta di Indonesia telah mengakomodir tiga pendekatan yang direkomendasikan oleh Jacques de Werra dalam melindungi karya cipta digital, yaitu: (1) perlindungan dengan ketentuan hukum konvensional, (2) perlindungan secara teknis dengan teknologi pengaman, (3) perlindungan hukum atas teknologi pengaman (Simatupang, 2021). Berkaitan dengan ketentuan hukum konvensional, hak cipta di Indonesia khususnya mengenai literatur telah dijelaskan dalam pasal peraturan perundang-undangan Hak Cipta (UUHC) Nomor 28 Tahun 2014 Pasal 40 Ayat (1) Huruf a. Pasal tersebut menyatakan bahwa ciptaan yang dilindungi meliputi buku, pamflet, perwajahan karya tulis yang diterbitkan, dan semua hasil karya tulis lainnya. Sedangkan bentuk tindakan yang melanggar hukum dijelaskan dalam Pasal 9 Ayat (3) yang menyatakan bahwa "Setiap orang yang tanpa izin pencipta atau pemegang hak cipta dilarang melakukan penggandaan dan/atau penggunaan secara komersial ciptaan". Pasal 10 UUHC juga menyebutkan bahwa

“Pengelola tempat perdagangan dilarang membiarkan penjualan dan/atau pengadaan barang hasil pelanggaran hak cipta dan/atau hak terkait di tempat perdagangan yang dikelolanya”.

Berkenaan dengan hukum untuk melindungi teknologi pengaman, secara umum mencakup aspek kepentingan publik, di antaranya adalah adanya pelanggaran merusak teknologi pengaman dari ciptaan, ketentuan aspek hukum pidana, pembatasan, dan pengecualian terhadap pelaku perusakan teknologi. Ketentuan-ketentuan tersebut secara tegas dijelaskan dalam Pasal 6 dan 7 UUHC yang menyatakan bahwa seorang pencipta dapat memiliki informasi manajemen Hak Cipta berupa metode untuk mengidentifikasi originalitas dan kode akses informasi. Pencipta juga dapat memiliki informasi elektronik Hak Cipta yang berupa informasi mengenai suatu ciptaan yang melekat secara elektronik, nama pencipta, pencipta sebagai pemegang Hak Cipta, masa dan kondisi penggunaan, nomor, dan kode informasi. Informasi manajemen dan informasi elektronik Hak Cipta tersebut dilarang dihilangkan, dirubah, atau dirusak. Selanjutnya mengenai penggunaan teknologi pengaman sebagai perlindungan hak cipta dijelaskan dalam Pasal 52 dan 53 UUHC yang secara umum menyatakan bahwa seseorang dilarang merusak, memusnahkan, menghilangkan, atau membuat tidak berfungsi sarana kontrol yang digunakan untuk melindungi suatu ciptaan. Sarana kontrol atau teknologi pengaman tersebut wajib memenuhi aturan perizinan dan persyaratan yang ditetapkan instansi berwenang sesuai ketentuan lanjutan dari peraturan pemerintah.

Berkaitan dengan sanksi pidana yang diterima oleh pelaku pelanggaran yang dimaksud dalam pasal 7 dan 52 dengan tujuan untuk penggunaan secara komersial dijelaskan pada pasal 112, berupa pidana penjara paling lama dua tahun dan/atau pidana denda paling banyak Rp.300.000.000 (tiga ratus juta rupiah). Sedangkan sanksi pidana berkaitan dengan pasal 9 ayat (1) huruf a dijelaskan pada pasal 113 yaitu berupa pidana penjara paling lama empat tahun dan/atau pidana denda paling banyak Rp.1.000.000.000 (satu milyar rupiah).

2.5. *Proof of Concept* (Poc)

PoC adalah suatu metode yang digunakan menunjukkan kelayakan dari sebuah konsep (*Proof of Concept Definition & Meaning - Merriam-Webster*, n.d.). Dalam definisi lain dikatakan bahwa PoC merupakan realisasi sebuah metode atau ide tertentu yang tujuannya untuk memastikan suatu parameter ilmiah atau teknologinya. PoC harus benar-benar dipahami secara keseluruhan agar dapat diidentifikasi dan prototipe dapat dirancang (*Partnerships for Innovation: Accelerating Innovation Research- Technology Translation*

(PFI: AIR-TT) (Nsf14569), n.d.). Dalam sumber lain dikatakan bahwa PoC adalah membuat suatu bukti atau dokumentasi tentang kelayakan sebuah ide yang menguraikan bagaimana produk atau layanan tersebut diidealkan dan siap dipasarkan (MacPherson, 2021). Poc tidak memiliki metode atau tahapan yang baku dalam pembuatannya, akan tetapi proses PoC setidaknya meliputi (1) definisi kriteria yang jelas dalam mencapai kesuksesan atau tujuan, (2) dokumentasi bagaimana PoC dilakukan, (3) komponen evaluasi (Pratt, 2020).

2.6. Metode Evaluasi *Grey Box*

Metode evaluasi *Black Box* merupakan metode pengujian pada fungsi perangkat lunak tanpa melihat struktur dan cara kerja yang dilakukan di dalam sistem (Boulton, 2017). Evaluasi yang dilakukan adalah memberikan masukan (*input*) data pada sistem, kemudian mengamati hasil (*output*) yang dilakukan oleh sistem. Pengujian ini dapat dilakukan oleh orang awam tanpa perlu mengerti hal teknis yang terjadi di dalam sistem yang dievaluasi. Tujuan dari dari metode ini adalah untuk mengidentifikasi bagaimana sistem bekerja atau merespon sesuatu dari tindakan pengguna, serta melihat seberapa jauh sistem bekerja, seperti melihat berapa banyak waktu respon yang dibutuhkan, kegunaan, keamanan, dan kehandalan sistem. Dengan pengujian ini penguji dapat mensimulasikan aktivitas pengguna dan melihat bagaimana sistem ini sesuai dengan yang diharapkan. Metode *Black Box* akan mengevaluasi semua subsistem yang relevan termasuk *User Interface* (UI), aplikasi, database, dependensi dan sistem yang terintegrasi.

Metode *White Box* adalah proses evaluasi secara mendetail pada sistem baik dari segi fungsi, arsitektur dan konfigurasi, sehingga dengan evaluasi ini dapat melihat secara keseluruhan permasalahan yang terjadi di dalam sistem, baik terkait keamanan, alur sistem, integrasi dan permasalahan lainnya. Pengujian pada metode ini tidak dapat dilakukan oleh sembarang penguji karena mensyaratkan seseorang yang memiliki pengetahuan mendalam terhadap sistem tersebut, seperti *programmer* atau *Quality Assuracne Enginneer* (QAE).

Metode evaluasi *Grey Box* merupakan gabungan dari metode evaluasi Black Box dan White Box. Penguji dalam metode ini dapat berupa seorang profesional sistem, pakar, atau arang awam. Dengan berbagai sudut pandang penguji tersebut diharapkan mampu memberikan gambaran dan hasil yang komprehensif dari produk sistem yang dibuat serta mendapatkan kualitas sistem yang lebih baik.

BAB 3
Metodologi

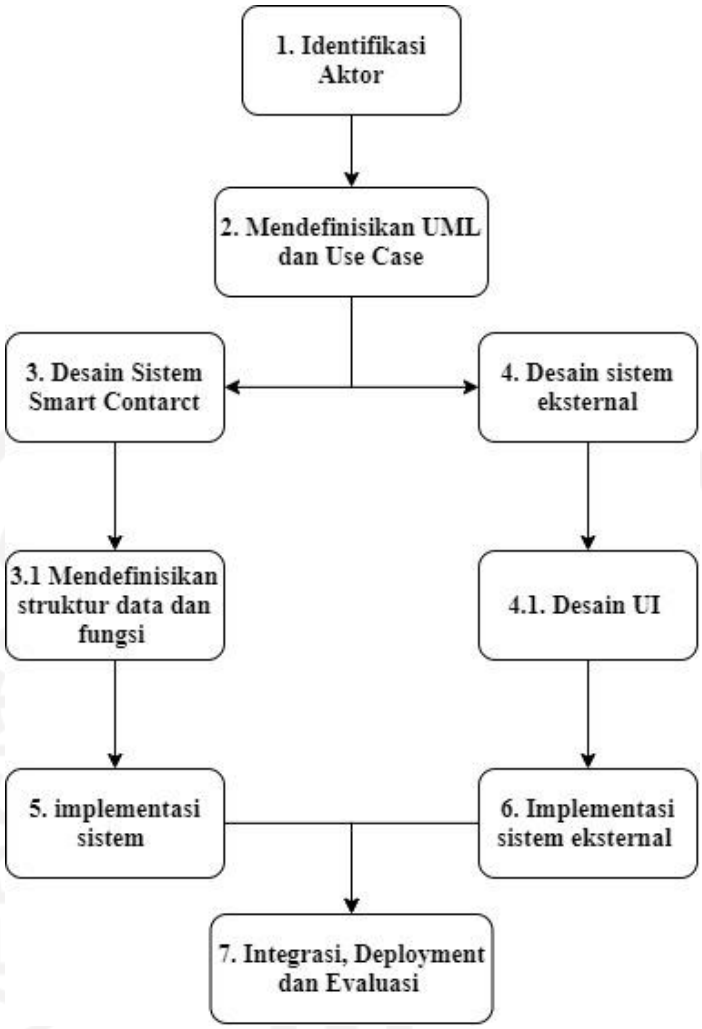
Dalam penelitian ini, dilakukan beberapa tahapan penelitian, yaitu: Identifikasi masalah, pembuatan aplikasi dan pengujian, tahap terakhir adalah evaluasi. Penjelasan mengenai tiap tahapan tersebut adalah sebagai berikut:

3.1. Identifikasi masalah HKI

Identifikasi masalah HKI dilakukan dengan cara studi pustaka dari berbagai sumber, baik artikel, situs, buku dan sumber pendukung lainnya. Literatur yang dirujuk untuk penelitian ini banyak menggunakan sumber dari artikel dan buku mengenai pelanggaran HKI dan sumber perlindungan hukum konvensional di Indonesia. Tujuan dari identifikasi ini adalah untuk mendapatkan data permasalah HKI digital di Indonesia dan sebagai alat evaluasi pembuktian sistem DRMChain yang dikembangkan.

3.2. Pembuatan Aplikasi

Pada bagian ini akan dilakukan proses identifikasi kebutuhan sistem atau teknologi yang mendukung proses implementasi. Proses pengembangan sistem ini, akan dilakukan dengan beberapa tahapan sebagaimana ditunjukkan pada Gambar 3.2.1. Tahap pertama adalah identifikasi aktor yang terlibat di dalam sistem yang dibangun. Selanjutnya mendefinisikan atau membuat usecase dari sistem yang akan dibangun. Sampai tahap ini, proses akan dibagi menjadi dua bagian, yaitu: Desain Sistem SC, dan desain aplikasi eksternal, seperti: aplikasi yang bertugas berinteraksi dengan SC dan pengguna. Desain sistem SC, akan mendefinisikan struktur data dan fungsinya. Sedangkan desain sistem aplikasi DRMChain akan fokus berkomunikasi dengan SC.



Gambar 3.2.1 Alur Pengembangan Sistem

Alur pengembangan sistem yang ditunjukan pada Gambar 3.2.1, sebagai berikut:

1. Identifikasi aktor bertujuan untuk mengidentifikasi pengguna yang menggunakan sistem.
2. Membuat desain sisitem dalam bentuk usecase dan UML, pada alur ini sistem yang dikembangkan harus mempertimbangkan semua aspek sistem secara keseluruhan.
3. Mendefinisikan ulang kebutuhan aktor dan *user story* pada sistem *smart contract* yang terdiri dari struktur data, fungsi dan *modifier*.
4. Mendefinisikan ulang kebutuhan aktor dan *user story* pada sitem eksternal yang berfokus pada desain *user interface* (UI).
5. Membuat dan menguji struktur dan fungsi sistem smart contract.
6. Membuat dan menguji sistem eksternal yang berfokus pada UI.
7. Integrasi, *deploy* dan mengevaluasi keseluruhan sistem.

Berkenaan dengan kebutuhan implementasi yang ditunjukkan pada Tabel 3.2.1, kebutuhun tersebut meliputi teknologi yang digunakan untuk mengimplementasi rancangan penelitian ini.

Tabel 3.2.1 Kebutuhan Teknologi

Teknologi	Keterangan
Blockchain	Ethereum
Development Tool	Truffle v5.6.2 (core: 5.6.2), Ganache v7.4.4, Solidity v0.5.16 (solc-js), Node v18.7.0, Web3.js v1.7.4, Metamask, react js, Python 3, Flask
DRM Protection	*Visible Watermark* (QRCode)
Storing and Sharing P2P Network	Infura IPFS

3.3. Evaluasi

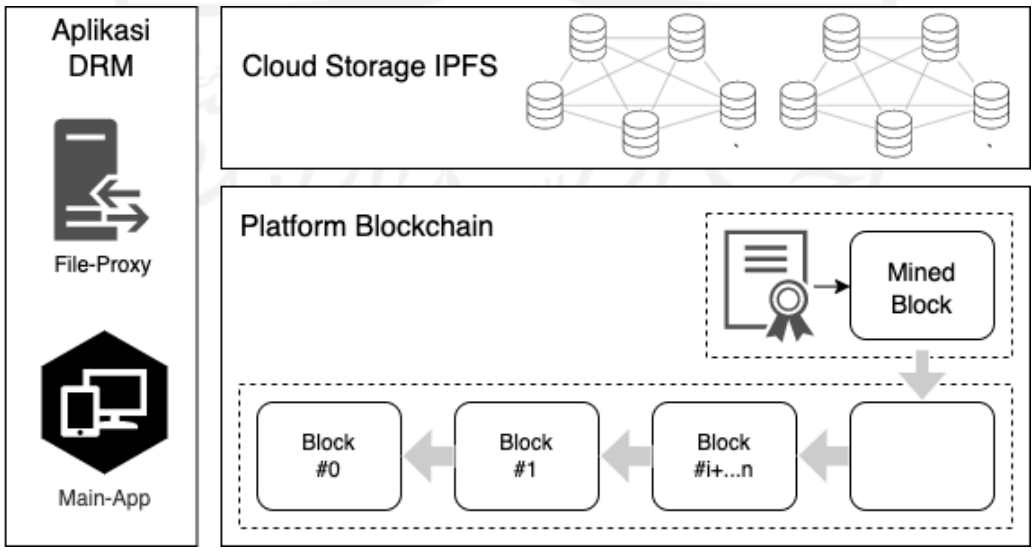
Tahap pertama pada bagian evaluasi ini adalah menguji kinerja sistem secara keseluruhan. Pengujian ini berguna untuk menguji kinerja dan fungsi sistem berjalan sesuai harapan. Pengujian ini menggunakan metode *Gray Box* yang merupakan kombinasi dari metode *Black Box* dan *White Box*. Pengujian *Black Box* akan dilakukan dengan memberikan masukan data dan menganalisa hasil keluarannya. Sedangkan pengujian secara *White Box* akan menguji sisi internal sistem berupa fungsi dan bebera subsistem yang terintegrasi di dalamnya. Tahap selanjutnya adalah menganalisa hasil penilitian dengan hasil identifikasi yang didapatkan. Hasil analisa ini digunakan untuk menjawab pertanyaaan penelitian.

BAB 4
Perancangan Sistem

Setelah dilakukan identifikasi permasalahan HKI, tahap selanjutnya dilakukan perancangan simulasi sistem sebagai sarana pembuktian konsep. Sistem yang dirancang sebatas pada aktivitas untuk membuktikan konsep DRMChain buku digital sesuai kebutuhan hasil identifikasi yang didapatkan. Perancangan ini terdiri dari arsitektur sistem dan desain sistem yang terdiri dari desain dari sistem SC, desain aplikasi DRMChain, serta desain antarmuka aplikasi.

4.1. Arsitektur Sistem

Sistem yang dirancang adalah sistem terdesentralisasi yang berjalan di atas blockchain. Menggunakan SC untuk menyimpan informasi hak cipta, mencatat setiap transaksi yang terjadi ke dalam blockchain, serta memindahkan hak kepemilikan konten. Dalam arsitektur sistem ini, file dokumen digital akan disimpan ke *Cloud Storage IPFS* (CSIPFS) melalui aplikasi File-Proxy dan kriptografi kunci publik digunakan untuk mengamankan hash file asli yang didapatkan diri mengunggah file ke CSIPFS sebelum disimpan ke SC. Hal ini dilakukan agar hanya pengguna yang memiliki kunci privat yang terkorespondensi dengan kunci publiknya yang dapat membuka dokumen. Sistem ini, terbagi menjadi tiga komponen utama, yaitu: aplikasi DRM, blockchain, dan storage seperti yang ditunjukkan pada Gambar 4.1.1.



Gambar 4.1.1 Arsitektur Sistem DRMChain

1. Aplikasi DRM

Komponen ini terdiri dari dua aplikasi yaitu Main-App dan File-Proxy. Main-App adalah aplikasi yang secara langsung berinteraksi dengan pengguna, SC, dan File-Proxy. Sedangkan File-Proxy adalah aplikasi terpercaya yang digunakan mengelola file seperti memberikan *watermarking* pada file, mengunggah dan mengunduh file dari CSIPFS. Aktivitas yang dilakukan pada komponen ini adalah mendaftarkan/mencatat konten karya cipta digital, menyisipkan *watermark* serta mengunggah dan mengunduh file dari CSIPFS, melakukan validasi dan verifikasi dengan proses enkripsi dan dekripsi *hash document* yang disimpan, melakukan permintaan dan penerimaan hak kepemilikan konten.

2. Storage

Komponen ini digunakan untuk menyimpan file konten digital asli. Penyimpanan pada *storage* ini dilakukan untuk menangani ukuran file yang besar sehingga tidak ada kekhawatiran akan ukuran file yang akan disimpan. Semua file yang disimpan akan di enkripsi dan harus melewati aplikasi File-Proxy untuk mendapatkan alamat file asli. Dengan begitu pemilik konten tidak perlu khawatir file-nya akan dicuri oleh orang lain.

3. Blockchain

Komponen ini merupakan bagian utama yang digunakan untuk menyimpan transaksi. Bagian ini akan berkomunikasi dengan SC untuk mengeksekusi transaksi, mendistribusikan ke setiap node yang terhubung dalam jaringan, dan melakukan proses validasi dan verifikasi sebelum kemudian dibuat blok baru dan disematkan ke dalam blockchain. Platform yang digunakan pada bagian ini adalah ethereum.

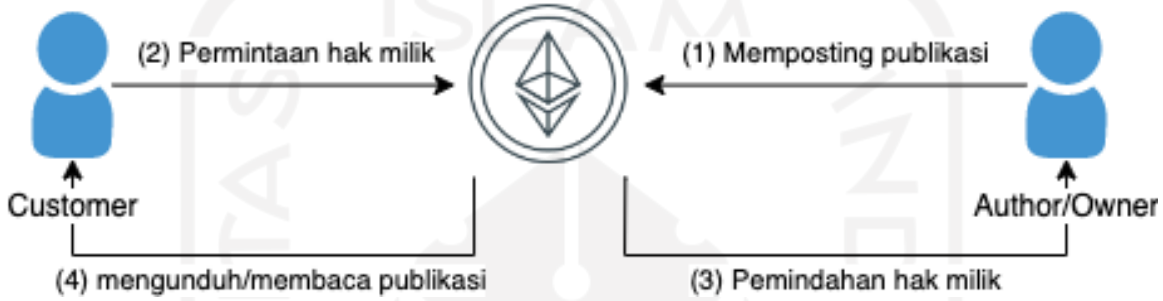
4.2. Desain Sistem

Aplikasi ini mengimplementasi proses publikasi dan pemindahan hak kepemilikan sebuah buku yang melibatkan dua partisipan, yaitu *author* dan *customer*. Gambar 4.2.1 mengilustrasikan desain secara umum interaksi antar kedua partisipan dengan membuat kesepakatan melalui SC. Setiap partisipan yang terlibat harus memiliki akun ethereum dengan sebuah alamat, kunci publik, dan kunci privat. Dua partisipan tersebut dijelaskan sebagai berikut:

1. *Author/Owner*: *Author* adalah seorang penulis buku yang membuat buku asli berwujud digital. *Author* dapat memposting atau mempublikasi buku dan menerima permintaan pemindahan buku melalui SC dengan memenuhi syarat dan ketentuan kontrak tertentu agar transaksi bisa dilanjutkan. Berkenaan dengan hak kepemilikan, *author* merupakan

pemilik buku (*owner*) dari buku yang dipublikasi. Setelah hak kepemilikan dipindahkan, *author* tidak lagi menjadi pemilik buku.

2. *Customer*: *Customer* adalah seseorang yang menjadi pelanggan untuk meminta/mendapatkan hak kepemilikan buku. *Customer* ini berinteraksi dengan SC untuk meminta hak milik buku. *Customer* dapat mengunduh/membaca buku dan memindahkan hak kepemilikannya kepada orang lain jika hanya hak buku tersebut sudah menjadi miliknya.



Gambar 4.2.1 Interaksi Sistem

Gambar 4.2.1 menunjukan empat aktivitas dalam sebuah siklus transaksi sukses yang terjadi di dalam sistem. Dimulai dari *author* memposting/mempublikasi buku digital yang mencakup nilai hash dari file yang telah diberi *watermark* dan diunggah ke CSIPFS melalui aplikasi File-Proxy. Nilai hash file tersebut telah dienkripsi dengan kunci publik *author/owner* sebelum dikirim ke SC. Setelah buku terpublikasi, *customer* yang berminat untuk mendapatkan buku, dapat meminta hak kepemilikan buku tersebut kepada *author* melalui aplikasi dengan mengirimkan data akun dan kunci publiknya ke SC. Permintaan pemindahan hak tersebut diteruskan ke *author* dan selanjutnya dapat dilakukan persetujuan permintaan pemindahan hak kepemilikan buku oleh *customer*. Setelah itu, author akan menyetujui hak pemindahan kepemilikan buku tersebut dengan mengambil dan mendekripsi nilai hash terenkripsi dari SC untuk mendapatkan nilai hash file asli dan mengenkripsinya kembali dengan kunci publik *customer* yang dikirimkan saat melakukan permintaan. Setelah hak milik buku disetujui, buku akan berpindah hak kepemilikannya kepada *customer*. Sampai disini, *customer* telah menjadi pemilik/*owner* buku dan dapat mengunduh/membaca buku tersebut dengan mendekripsi nilai hash file terenkripsi dari SC dengan kunci privatnya. Proses dekripsi nilai hash file akan mendapatkan nilai hash file asli yang dapat digunakan untuk mengambil/mengunduh/membaca dokumen dari aplikasi File-Proxy. Sistem ini

mengharuskan pengguna untuk login terlebih dahulu ke akun ethereum supaya sistem dapat mengenali pengguna dan transaksi dapat dilakukan. Skenario usecase pada sistem secara lebih detail ditunjukkan oleh Tabel 4.2.1 .

Tabel 4.2.1 Deskripsi Skenario Usecase

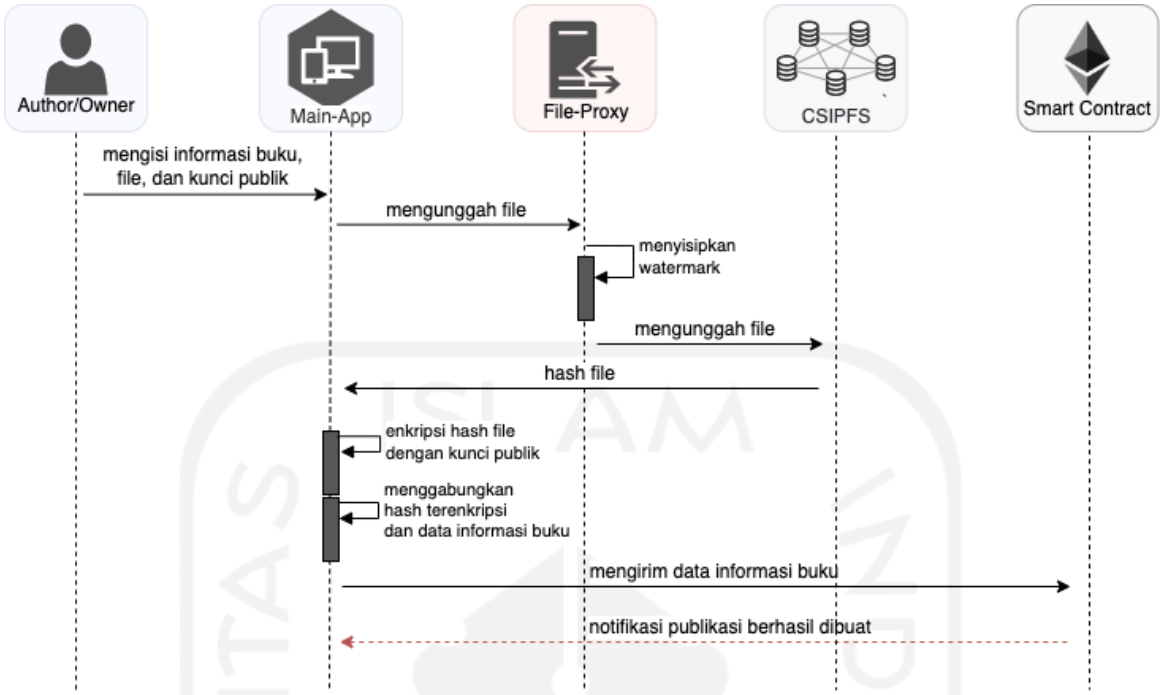
Judul	Aktor	Skenario
Memposting publikasi	Author (Owner)	• Author mengisi form informasi buku dan mengunggah file buku digital asli. • Sistem mengunggah file ke aplikasi file-proxy kemudian memberi *watermark* dan mengunggah ke ipfs cloud server. • Sistem mengenkripsi hash file respon dari aplikasi file-proxy. • Sistem menggabungkan data informasi dan hash file hasil dari file-proxy. • Sistem mengirimkan data ke SC. • Sistem akan menampilkan pesan berhasil disimpan.
Permintaan hak milik	Customer	• Customer memilih buku. • Customer melakukan permintaan hak kepemilikan buku dengan menyertakan kunci publiknya. • Sistem mengirimkan data customer dengan kunci publik ke SC. • Sistem menampilkan pesan berhasil disimpan.
Penerimaan dan pemindahan hak milik	Author (Owner)	• Author melihat daftar permintaan hak milik. • Author memilih salah satu permintaan. • Author menerima permintaan. • Sistem mengambil hash dokumen terenkripsi dari SC. • Sistem mendekripsi hash dokumen untuk mendapatkan hash file asli

		• Sistem mengenkripsi hash file asli dengan kunci publik customer. • Sistem mengirim hash file ke SC. • SC memindahkan hak milik ke customer.
Mengunduh/membaca buku	Author (Owner)	• Author menekan tombol baca. • Sistem mengambil hash dokumen terenkripsi dari SC. • Sistem mendekripsi hash dokumen terenkripsi. • Sistem menampilkan dokumen.

Sebagaimana diketahui, skenario usecase pada Tabel 4.2.1 memuat empat aktivitas utama yaitu: memposting publikasi oleh *author*, melakukan permintaan oleh *customer*, menerima dan memindahkan hak milik oleh *author*, serta mengunduh/membaca publikasi oleh *customer*. Setiap aktivitas di dalam sistem yang terjadi, akan sesuai urutan kejadian waktu dan objek. Namun, urutan kejadian pada setiap aktivitas pada sistem yang dirancang tidak akan dijelaskan mengenai akun pengguna. Hal ini karena fitur SC sudah mencakup fitur tersebut. Penjelasan mengenai urutan aktivitas di dalam sistem dengan komponen terkait, diuraikan sebagai berikut:

1. Memposting Publikasi oleh *Author*

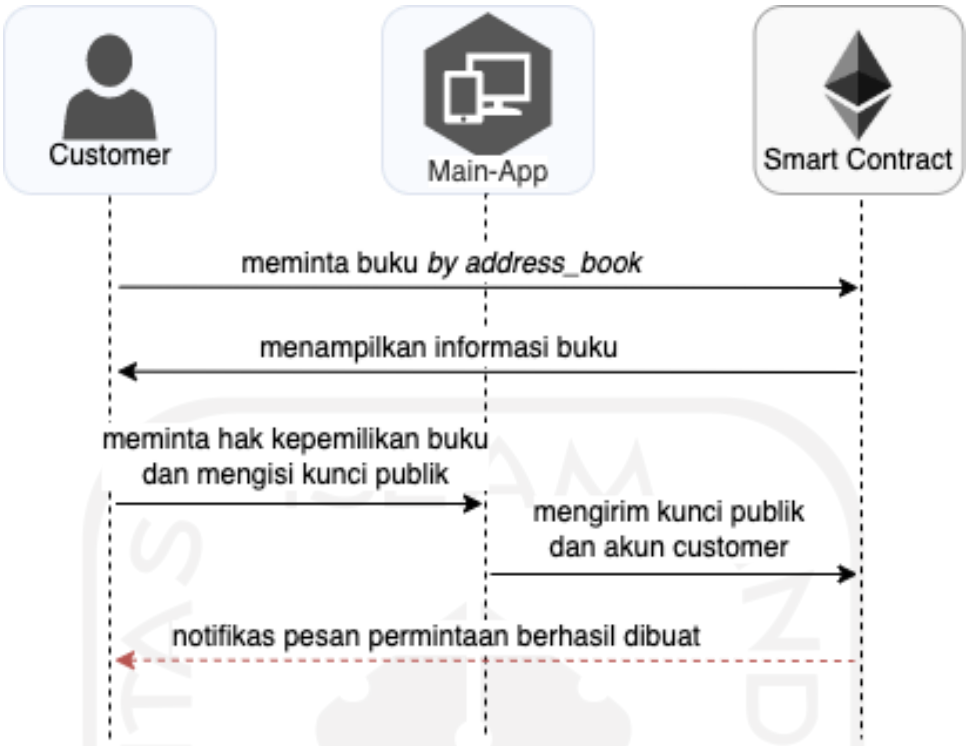
Mula-mula *author* membuka halaman *create book* pada aplikasi Main-App. Setelah itu, *author* memasukkan data informasi buku dan file dokumen yang mau dipublikasikan pada formulir (*form*) yang tersedia. Formulir tersebut terdiri dari beberapa kolom data terkait buku yang akan dipublikasikan dan terdapat satu kolom kunci privat pengguna (*author*) untuk kebutuhan enkripsi. Setelah selesai mengisi data, *author* akan menekan tombol proses untuk mengeksekusi fungsi mempublikasikan buku. Setelah tombol proses ditekan, sistem akan mengirimkan data buku dan mengunggah file ke aplikasi File-Proxy. Selanjutnya, aplikasi File-Proxy akan menyisipkan *watermark* dari data buku yang dikirimkan dan mengunggahnya ke CSIPFS. CSIPFS akan mengembalikan nilai hash dari lokasi file tersebut dan diteruskan ke Main-App. Aplikasi Main-App akan melakukan mengenkripsi nilai hash tersebut dengan kunci publik *author* yang diperoleh dari hasil ekstraksi kunci privat yang dimasukkan *author*. Selanjutnya Main-App akan mengirimkan semua data termasuk nilai hash yang telah dienkripsi dengan kunci publik author ke SC untk disimpan ke dalam blockchain. Diagram sequence publikasi ditunjukkan pada Gambar 4.2.2.



Gambar 4.2.2 Diagram Sequence Untuk Sukses Posting Publikasi

2. Permintaan Hak Milik Oleh *Customer*

Setelah buku yang dipublikasi oleh *author* terpublikasi, buku tersebut dapat dilihat oleh pengguna lain melalui halaman katalog buku. Dengan begitu, *customer* yang berminat untuk memiliki bukunya, dapat melakukan permintaan hak kepemilikan buku melalui halaman detail buku pada aplikasi Main-App. Pada halaman detail buku, *customer* dapat membuat permintaan hak kepemilikan buku dengan menekan tombol *request Owner*. Sebelum permintaan diteruskan ke SC, *customer* diminta untuk mengisi data kunci privat miliknya terlebih dahulu. Setelah itu Main-App akan mengekstrak kunci privat tersebut menjadi kunci publik dan mengirimkan beserta alamat akun customer ke SC. Diagram sequence permintaan hak milik ditunjukkan pada Gambar 4.2.3.

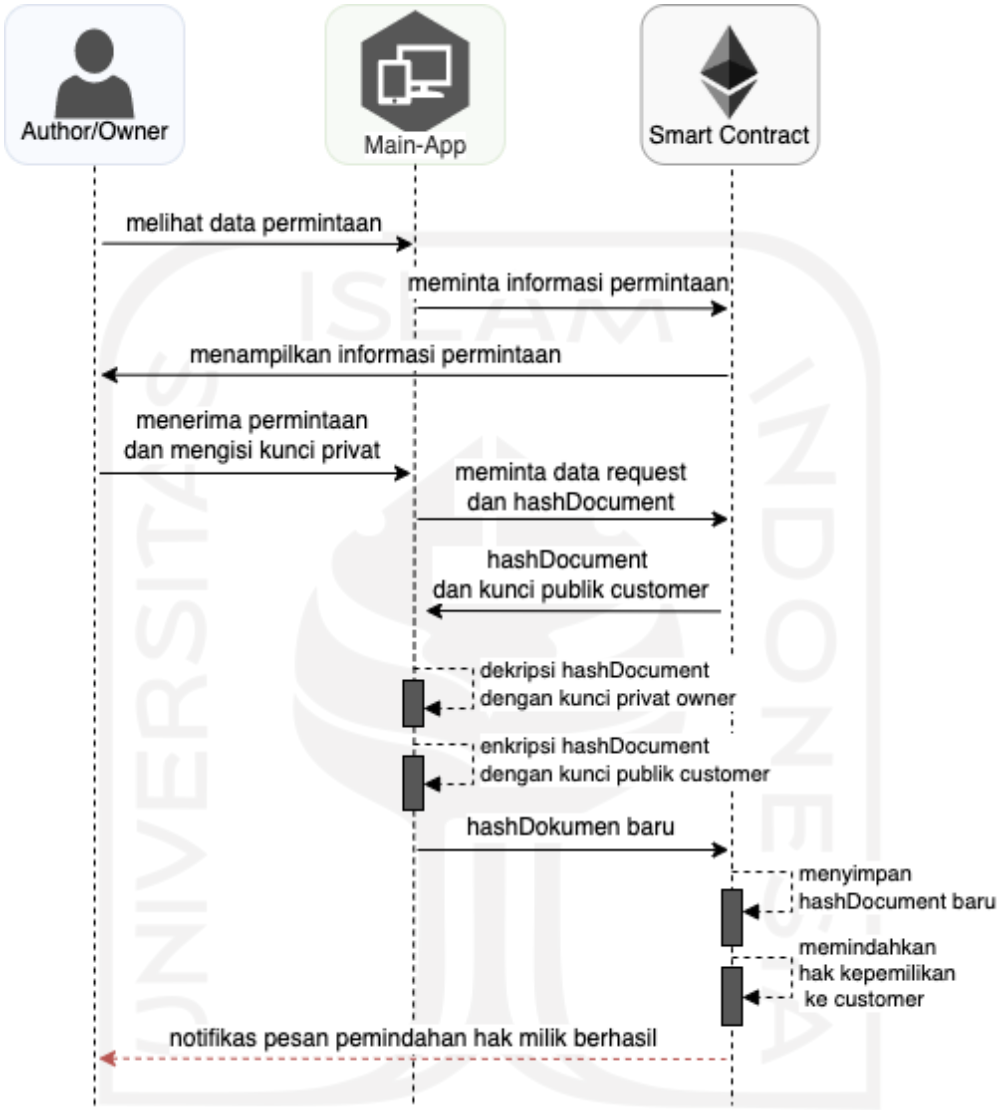


Gambar 4.2.3 Diagram Sequence Untuk Sukses Permintaan Hak Milik

3. Penerimaan dan Pemindahan Hak Milik Oleh *Author*

Proses penerimaan dan pemindahan hak milik diawali dengan *author/owner* yang mula-mula melihat daftar permintaan hak milik pada Main-App untuk memilih permintaan yang akan disetujui. Selanjutnya *author/owner* milih salah satu permintaan tersebut. Setelah itu, *author/owner* menekan tombol terima dan mengisi data kunci privatnya pada *field* yang diminta oleh sistem. Selanjutnya Sistem meminta data *requestOwner* dari SC dan SC mengembalikan data berupa *hashDocument*, alamat akun *customer*, kunci publik *customer*. Untuk memindahkan hak kepemilikan kepada pengguna lain tentu diperlukan pengaman agar lokasi file (*hashFile* asli) tidak dapat dibuka oleh sembarang pengguna. Oleh karena itu, *documentHash* yang diambil akan dienkripsi menggunakan kunci publik *customer*, sehingga hanya customer tersebut yang dapat membuka dengan kunci privatnya. Skema yang ditunjukkan pada Gambar 4.2.4, *dokumentHash* yang diambil, didekripsi terlebih dahulu menggunakan kunci privat *owner* untuk mendapatkan *hasFile* asli (lokasi file asli). Setelah *hashFile* asli didapatkan, *hashFile* tersebut kemudian dienkripsi kembali menggunakan kunci publik *customer*. Setelah proses enkripsi ulang selesai dilakukan, sistem mengirimkan *hashDocument* baru ke SC dan mengubah *hashDocument* yang lama. Setelah *hashDocument* diubah, SC akan memindahkan hak kepemilikan buku tersebut ke *customer*. Setelah proses berhasil dilakukan, SC akan mengirimkan notifikasi pesan kepada

author bahwa proses pemindahan hak kepemilikan berhasil dilakukan. Diagram sequence proses ini ditunjukkan pada Gambar 4.2.4.

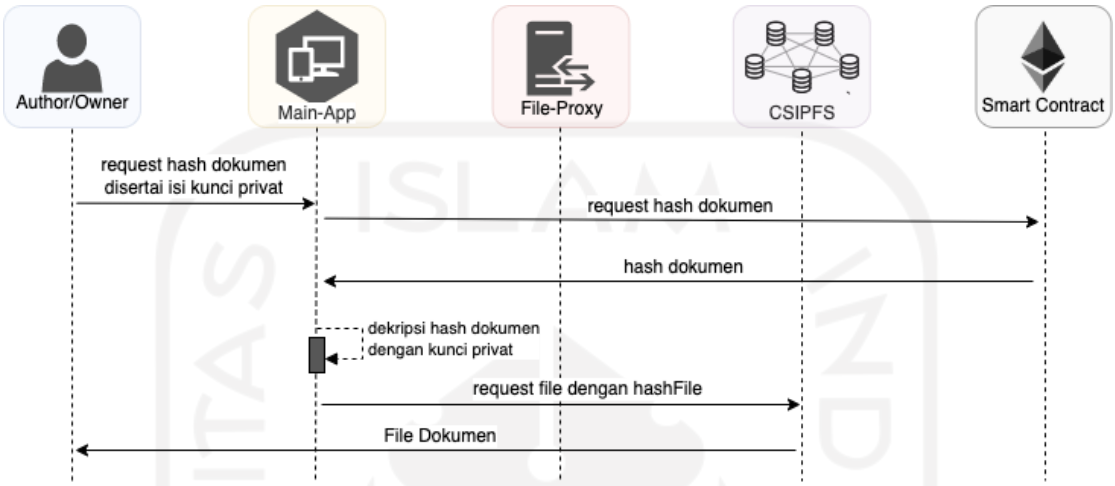


Gambar 4.2.4 Diagram Sequence Untuk Sukses Penerimaan dan Pemindahan Hak Kepemilikan Buku

4. Mengunduh atau Membaca Buku Oleh *Customer*

Proses mengunduh / membaca buku, diawali dengan owner mula mula masuk dalam detail buku pada aplikasi Main-App, kemudian menekan tombol read (baca) untuk mendapatkan hash dokumen. Setelah itu sistem meminta *owner* untuk memasukkan kunci privat sebelum melanjutkan permintaan hash dokumen oleh Main-App. Kunci privat ini berguna untuk mendekripsi hash dokumen pada tahap berikutnya. Selanjutnya, sistem akan meminta hash dokumen ke SC dan SC akan mengembalikan data hash dokumen. Setelah

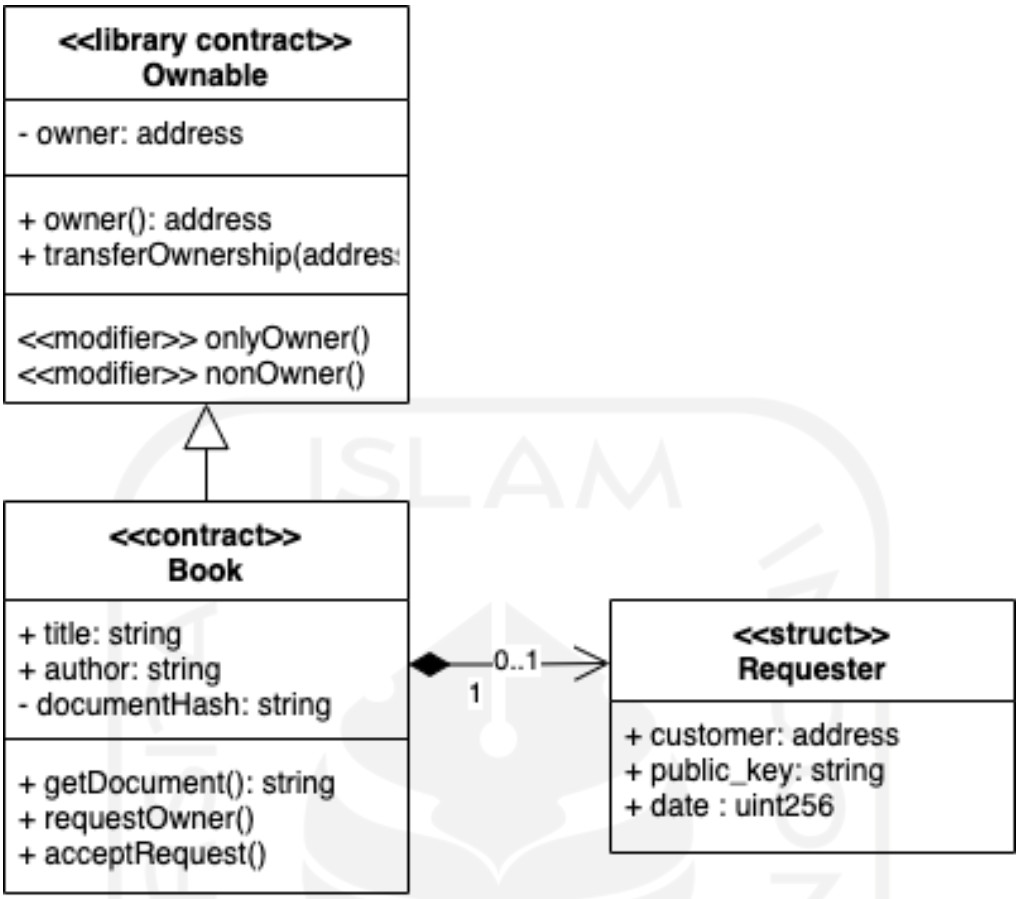
hash dokumen didapatkan, Main-App akan melakukan dekripsi dengan kunci privat owner. Setelah itu, sistem akan meminta file dokumen asli ke CSIPFS melalui aplikasi File-Proxy. Kemudian, CSIPFS akan mengembalikan file dokumen yang dapat dibaca/diunduh oleh *owner*. Diagram sequence proses ini ditunjukkan pada Gambar 4.2.5.



Gambar 4.2.5 Diagram Sequence Untuk Sukses Unduh / Baca Dokumen

4.2.1. Desain Sistem Smart Contract

Berdasarkan ilustrasi dan skenario usecase yang dijelaskan, diketahui domain yang terlibat di dalam perancangan sistem tersebut adalah buku, pengguna, pemilik/*owner* buku, dan transaksi permintaan atau pemindahan hak kepemilikan buku. Dengan begitu, keempat domain tersebut menjadi model rujukan dari pembentukan struktur data dan operasinya. Oleh karena sistem yang dirancang akan diimplementasi pada SC yang pada dasarnya memuat fitur identifikasi pengguna dan transaksi, maka domain pengguna dan transaksi tidak diperlukan. Sebagai gantinya, domain transaksi untuk melakukan permintaan dan persetujuan dibuat dan dinamakan *requester*. Secara detail struktur data dan operasi diwujudkan dalam diagram kelas seperti Gambar 4.2.6.



Gambar 4.2.6 Diagram Kelas Smart Contract

Sebelum membahas lebih jauh, perlu diketahui bahwa perancangan sistem ini akan diimplementasi pada SC ethereum menggunakan bahasa solidity. Bahasa solidity tersebut memuat beberapa entitas utama (Nizamuddin et al., 2019), antara lain:

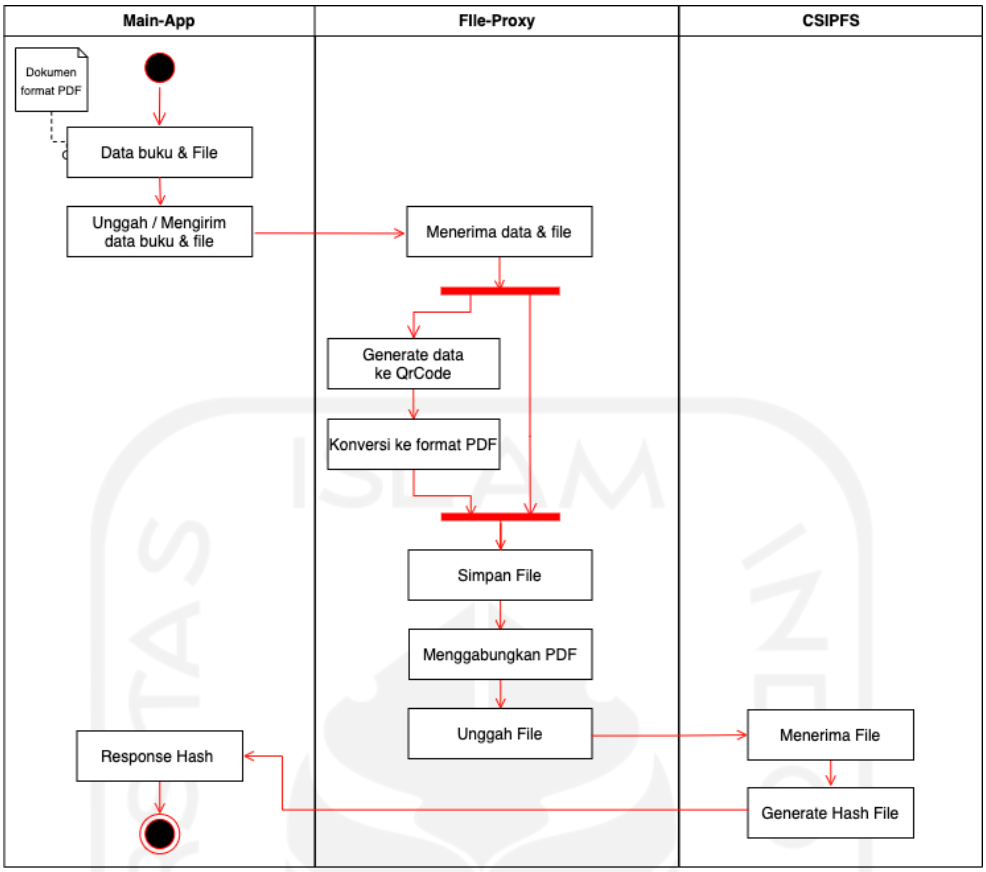
1. *Variables*: *Variables* adalah suatu tempat yang digunakan untuk menampung data dalam memori yang memiliki nilai yang dapat berubah tergantung kondisinya. *Variables* ini dapat menyimpan tipe data tertentu.
2. *Functions*: *Functions*/Fungsi adalah metode yang digunakan untuk menyelesaikan tugas.
3. *Modifiers*: *Modifiers* digunakan untuk mengubah atau membatasi fungsi dalam SC. Dengan *modifier* ini untuk memeriksa kondisi secara otomatis sebelum menjalankan fungsi.
4. *Events*: *Events* digunakan untuk mencatat suatu kejadian sehingga dapat membantu melacak suatu kejadian/perubahan *state* di dalam SC. Event juga dapat digunakan untuk menyebarkan pesan notifikasi ke semua entitas aktif dalam jaringan blockchain.
5. *Structs*: *Structs* adalah tipe data yang memuat data dengan tipe lain.

Dari diagram yang ditunjukkan pada Gambar 4.2.6 kontrak *book* akan menurunkan semua fungsi dan *modifier* dari library kontrak *ownable*. Dengan begitu kontrak *book* dapat digunakan untuk mengidentifikasi pemilik melalui fungsi *owner()* dan memindahkan hak kepemilikan melalui fungsi *transferOwnership*(). Fungsi *getDocument()* digunakan oleh pengguna (*owner*) untuk mengambil data *hashDocument* dari SC. *Struct requester* merupakan bagian dari kontrak *book* yang berfungsi untuk menampung data permintaan hak kepemilikan buku dari *customer*. Data yang ditampung berupa alamat identitas *customer,* kunci publik customer, dan tanggal permintaan dibuat. *Struct requester* akan diisi ketika pengguna (customer) memanggil dan mengirimkan data ke fungsi *requestOwner().* Fungsi *acceptRequest()* digunakan oleh *owner,* untuk menyetujui permintaan dan memindahkan hak kepemilikan buku melalui fungsi *transferOwnership()*.

4.2.2. Desain Aplikasi Main-App dan File-Proxy

Seperti yang telah dijelaskan sebelumnya, kedua aplikasi ini merupakan komponen utama yang mendukung sistem DRMChain. Bagian utama dari komponen ini adalah proses penyisipan *watermark* dan pengunggahan file ke CSIPFS oleh aplikasi File-Proxy dan proses enkripsi serta dekripsi hash file oleh aplikasi Main-App. *Watermark* yang diimplementasikan dalam sistem ini berupa *visibily watermark* berupa QRCode. Pertimbangan utama dengan membuat aplikasi File-Proxy adalah untuk keamanan dan perlindungan konten dari pencurian. Pada subbab ini hanya dipresentasikan terkait aktivitas dalam beberapa proses tersebut.

Gambar 4.2.7 menunjukkan diagram aktivitas untuk proses *watermarking* dan pengunggahan ke CSIPFS. Proses ini dimulai dari Aplikasi Main-App yang mengirim data dan mengunggah file dokumen PDF ke File-Proxy. Setelah aplikasi File-Proxy menerima data dan file, sistem akan memisahkan antara data dan file. Data diubah menjadi QRCode kemudian dikonversi ke format PDF. Setelah itu kedua QRCode dan File disimpan digabungkan menjadi file ter-*watermark.* Selanjutnya, file tersebut diunggah ke CSIPFS dan mengembalikan nilai hash lokasi file dari CSIPFS ke Main-App.



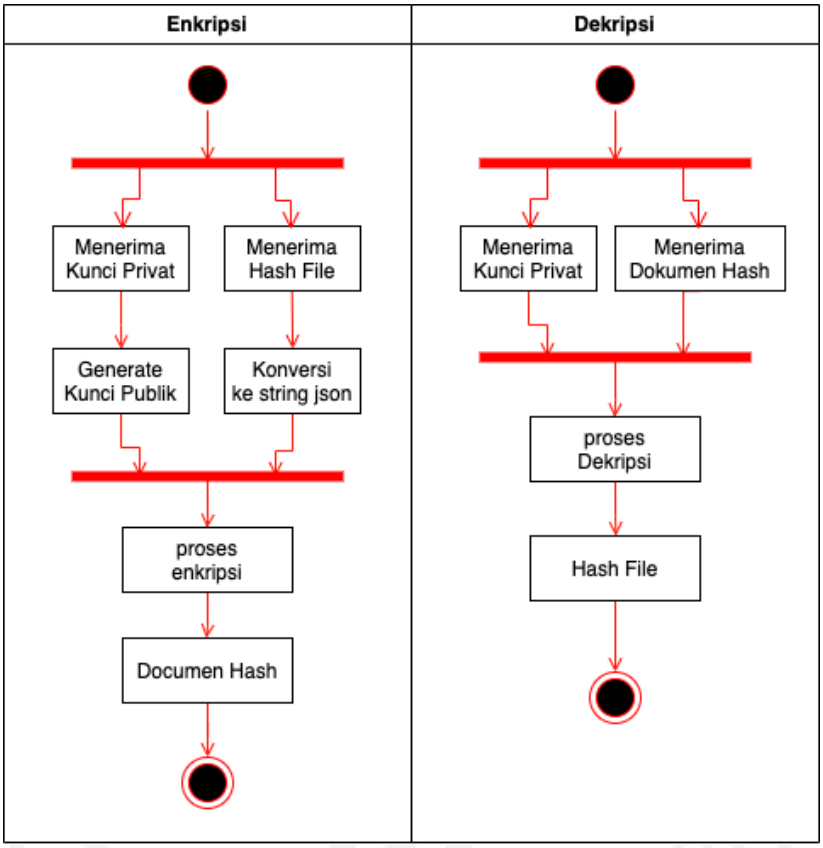
Gambar 4.2.7 Diagram Aktivitas Sukses Penyisipan *Watermark* QRCode dan Pengunggahan File ke CSIPFS

Aplikasi lain dari sistem DRMChain adalah Main-App. Fitur utama aplikasi ini adalah proses enkripsi dan dekripsi hash file. Fitur ini digunakan dalam beberapa aktivitas, seperti aktivitas sebelum hash file dikirim ke SC ketika publikasi dibuat, aktivitas ketika menerima permintaan hak kepemilikan, dan ketika membaca/mengunduh file dokumen.

Proses enkripsi membutuhkan dua masukan berupa kunci privat yang dimasukkan oleh pengguna melalui form dan hash file dari proses unggah ke aplikasi File-Proxy yang diubah ke bentuk string JSON. Kebutuhan untuk proses enkripsi sebenarnya adalah kunci publik. Namun, karena kunci publik pada akun ethereum tidak dipublikasikan, maka kunci publik didapatkan dari proses *generate* kunci privat. Setelah mendapatkan kunci publik, selanjutnya dilakukan proses enkripsi untuk menghasilkan dokumen hash.

Berkebalikan dari proses enkripsi, proses dekripsi membutuhkan masukan berupa *hash document* dari proses enkripsi dan kunci privat. Dari kedua masukan ini, selanjutnya dilakukan proses dekripsi dan dihasilkan Hash File. Hash File ini adalah lokasi file yang

berada di CSIPFS. Aktivitas enkripsi dan dekripsi pada aplikasi Main-App ditunjukkan pada Gambar 4.2.8.



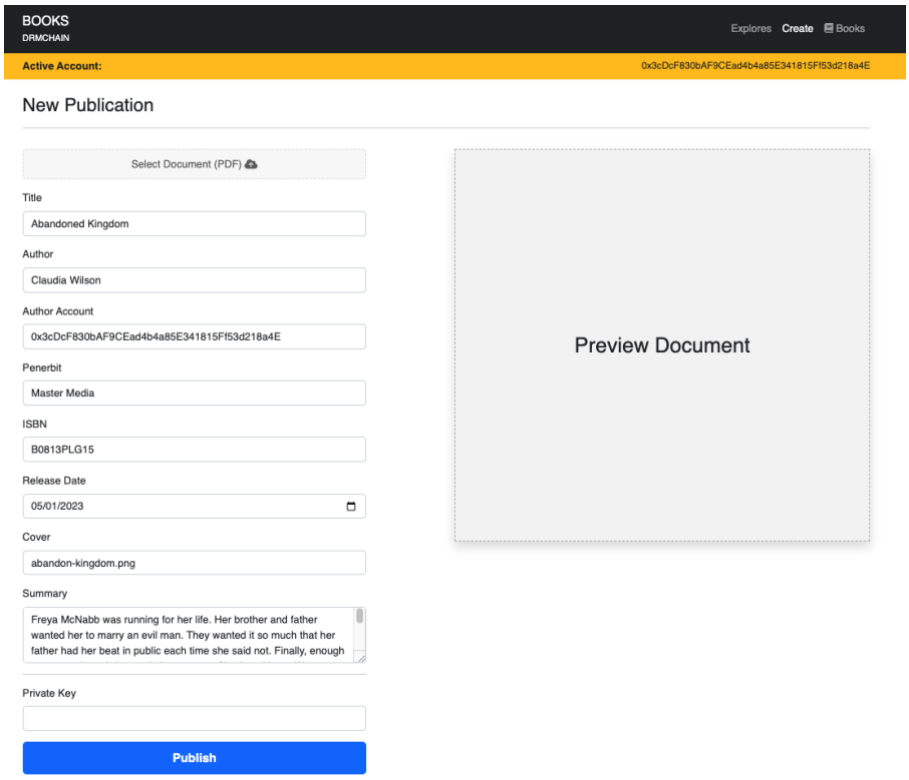
Gambar 4.2.8 Diagram Aktivitas Sukses Proses Enkripsi dan Dekripsi

4.2.3. Desain Antarmuka (*User Interface*)

Bagian ini menjelaskan mengenai rancangan antarmuka sistem yang dibuat. Rancangan ini menjelaskan bagian antarmuka yang diperlukan pada setiap fase aktivitas. Berikut penjelasan selengkapnya:

1. Antarmuka Mempublikasikan Buku

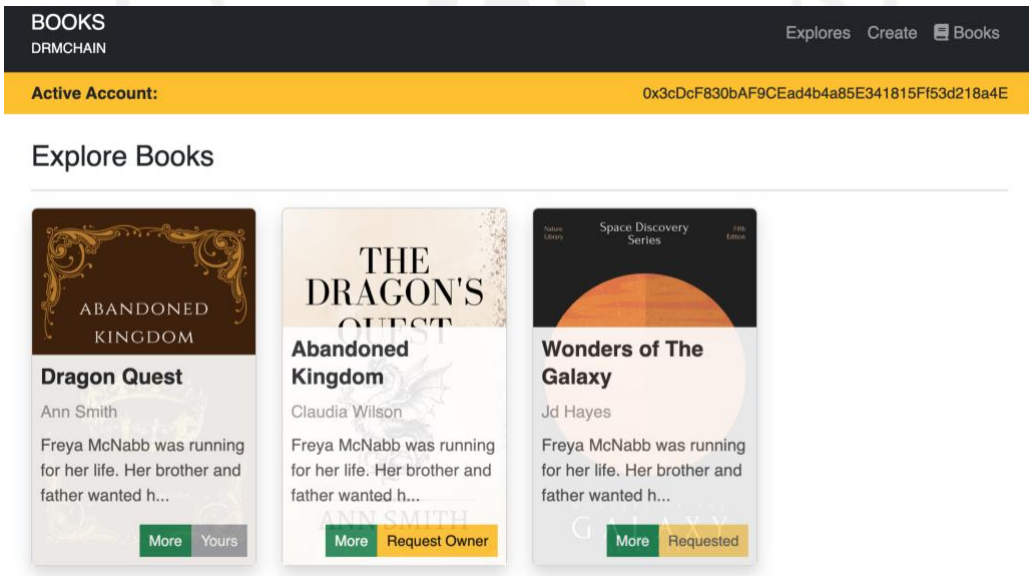
Antarmuka ini berfungsi bagi pengguna/author untuk mempublikasikan publikasi baru. Setiap pengguna di dalam sistem dapat menjadi *author* maupun *customer*, sehingga tidak ada batasan bagi pengguna untuk mengakses halaman ini. Halaman ini terdiri dari form dan preview PDF yang diupload. Untuk memastikan semua aspek data publikasi terpenuhi, semua *field* di dalam form tersebut diwajibkan. Terdapat *field* masukan private key yang digunakan untuk mengenkripsi hash file dari mengunggah file ke CSIPFS. Antarmuka membuat publikasi ini ditunjukkan pada Gambar 4.2.9.



Gambar 4.2.9 Antarmuka Mempublikasikan Buku

2. Antarmuka Katalog Buku

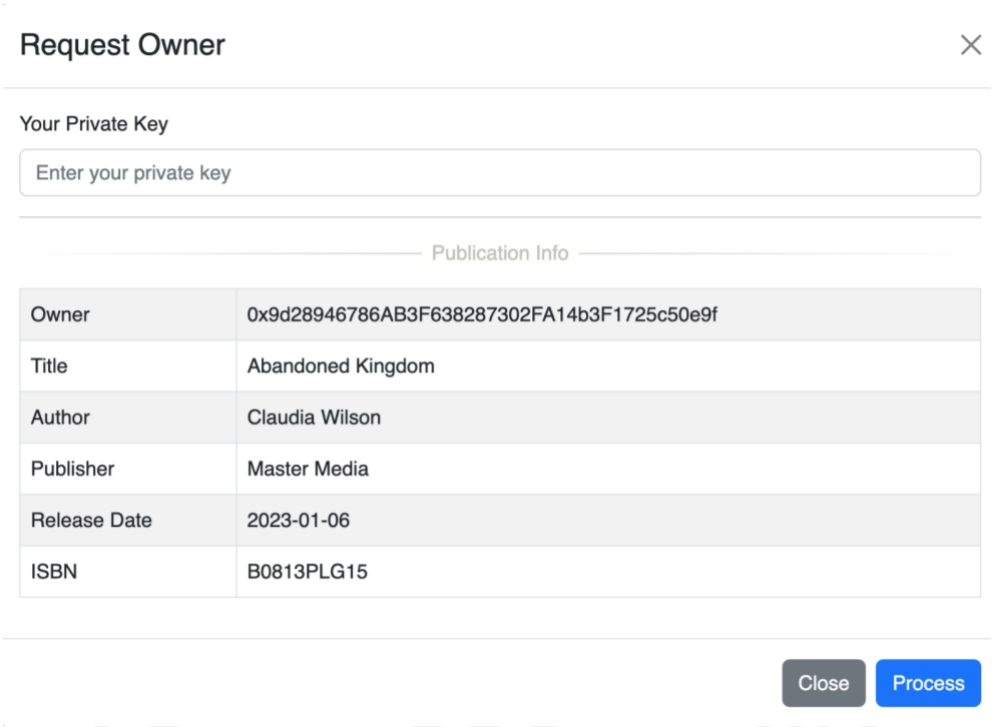
Antarmuka ini dibuat agar setiap pengguna dapat melihat daftar buku yang tersedia. Dengan begitu pengguna melakukan permintaan hak kepemilikan pada buku yang diinginkan. Antarmuka ini ditunjukkan pada Gambar 4.2.10.



Gambar 4.2.10 Antarmuka Katalog Buku

3. Antarmuka Konfirmasi Permintaan Hak Kepemilikan Buku

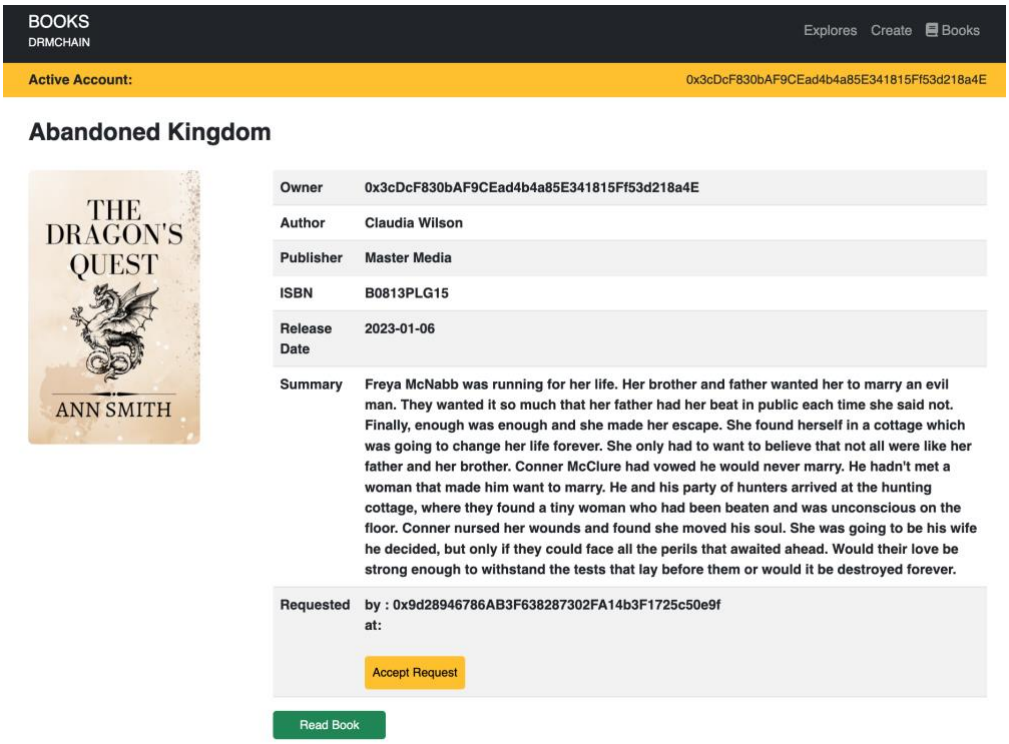
Antarmuka ini akan muncul ketika pengguna menekan tombol *Request Owner*. Berfungsi untuk mengkonfirmasi permintaan. Terdapat *field* kunci privat pengguna sebagai prasyarat data permintaan hak milik buku. Terdapat pula Informasi buku terkait. Antarmuka ini ditunjukkan pada Gambar 4.2.11.



Gambar 4.2.11 Antarmuka Konfirmasi Permintaan Hak Kepemilikan Buku

4. Antarmuka Detail Buku

Antarmuka ini berfungsi untuk menampilkan detail informasi buku yang dipilih oleh pengguna. Terdapat tombol *Accept Request* yang berfungsi untuk menerima permintaan pemindahan hak kepemilikan buku dari *customer*. Terdapat juga tombol *Read Book* yang berfungsi untuk membuka dokumen buku tersebut. Antarmuka ini ditunjukkan pada Gambar 4.2.12.



Gambar 4.2.12 Antarmuka Detail Buku

5. Antarmuka Konfirmasi Menerima Permintaan

Antarmuka ini muncul ketika pengguna menekan tombol *Accept Request* pada Halaman Detail Buku. Berfungsi untuk mengkonfirmasi dan form data sebagai prasyarat proses penerimaan buku. Terdapat *field* untuk memasukkan kunci privat pengguna dan juga informasi mengenai *attribute customer* (pihak yang meminta). Antarmuka ini ditunjukkan pada Gambar 4.2.13.

Gambar 4.2.13 Antarmuka Menerima Permintaan

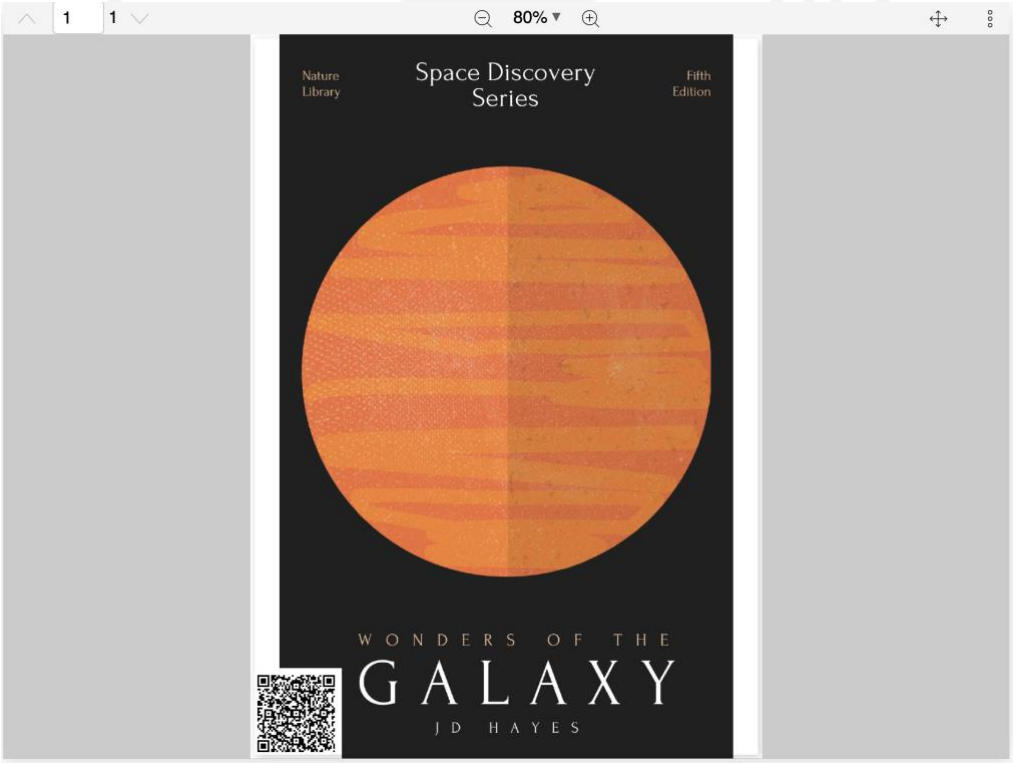
6. Antarmuka Konfirmasi Membaca Buku

Antarmuka ini muncul ketika pengguna menekan tombol *Read Book* pada Halaman Detail Buku. Berfungsi untuk mengkonfirmasi dan form data sebagai prasyarat proses membaca buku. Terdapat *field* untuk memasukkan kunci privat pengguna. Antarmuka ini ditunjukkan pada Gambar 4.2.14.

Gambar 4.2.14 Antarmuka Konfirmasi Membaca Buku

7. Antarmuka Penampil Dokumen PDF

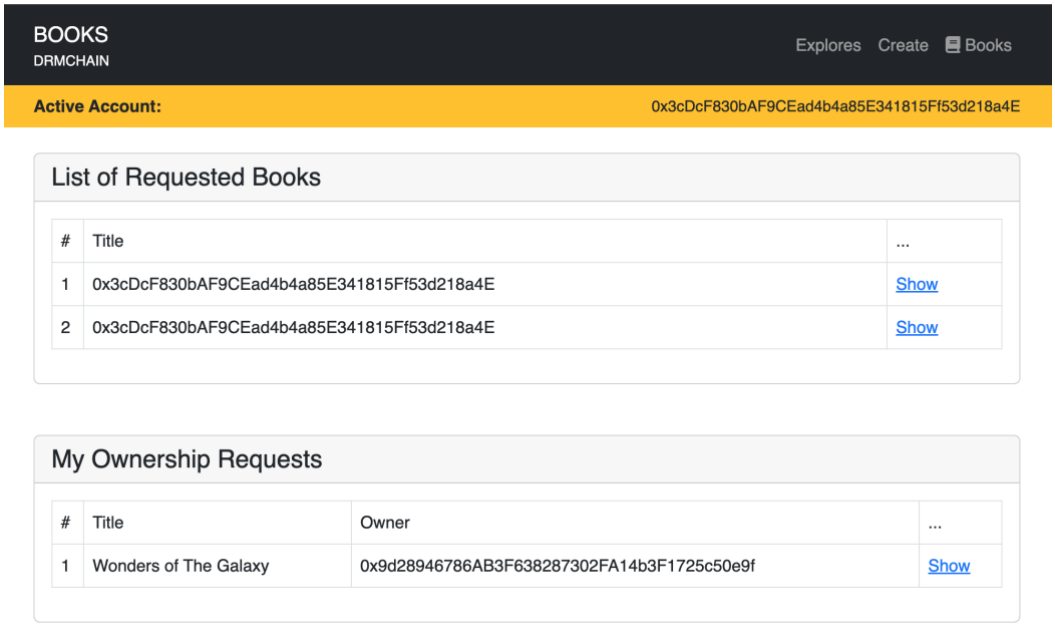
Antarmuka ini berada pada halaman detail buku. Halaman ini akan muncul jika proses konfirmasi membaca buku berhasil dilakukan. Berfungsi untuk menampilkan dokumen PDF yang diambil dari URL. Antarmuka ini ditunjukkan pada Gambar 4.2.15.



Gambar 4.2.15 Antarmuka Penampil Dokumen PDF

8. Antarmuka Permintaan Buku

Antarmuka ini terdiri dari dua informasi yaitu, daftar permintaan, dan daftar buku yang diminta hak kepemilikannya. Antarmuka ini berguna agar pengguna dapat memantau dan melihat permintaan. Dari Antarmuka ini juga memungkinkan pengguna dapat melihat detail informasi dengan menekan tombol *show*. Antarmuka ini ditunjukkan pada Gambar 4.2.16.



Gambar 4.2.16 Antarmuka Permintaan Buku

BAB 5
Implementasi dan Pembahasan

Bagian ini akan dibagi menjadi beberapa subbab, yaitu: implementasi sistem, pengujian sistem, dan evaluasi. Subbab implementasi sistem akan fokus membahas bagaimana sistem yang telah dirancang ini diimplementasi. Subbab pengujian akan fokus membahas pengujian sistem yang dirancang dengan mengukur tingkat keberhasilan dalam setiap alur. Sedangkan evaluasi akan fokus membahas PoC dari sistem yang dibangun dengan menilai dengan kriteria yang telah diidentifikasi dalam Bab 4.

5.1. Implementasi Sistem

Sebagaimana dengan arsitektur sistem yang ditunjukkan pada Gambar 4.1.1, penjelasan implementasi sistem ini akan dibagi dalam dua komponen utama, yaitu: implementasi pada SC dan implementasi aplikasi pada DRMChain. Pada bagian ini, akan disajikan implementasi sistem dengan beberapa potongan kode utama.

5.1.1. Implementasi Smart Contract

Implementasi kode ini ditulis menggunakan bahasa pemrograman solidity yang memuat beberapa variabel, fungsi, *modifier*, dan *event*. Implementasi ini mengacu pada rancangan kelas diagram yang dibuat susuai Gambar 4.2.6. Tiga entitas utama yang diimplementasi, antara lain: *library* kontrak *Ownable*, kontrak *Book*, dan *struct requester*.

Kontrak *Library kontrak Ownable* menangani beberapa hal yang berkaitan dengan peran pengguna sebagai pemilik buku yang dipertukarkan. Kontrak ini didefinisikan sebagai abstrak karena berisi variabel dan fungsi dasar yang dapat diturunkan dan digunakan secara fleksibel. Kontrak ini memiliki variabel *_owner* untuk menampung alamat pengguna sebagai pemilik buku dan diinisialisasi dengan alamat pengguna yang melakukan permintaan terhadap kontrak. Implementasi kontrak dan variabel *_owner* ditunjukkan pada Gambar 5.1.1.

```
abstract contract Ownable {
    address private _owner;

    constructor() {
        _owner = msg.sender;
    }
```

Gambar 5.1.1 Kode Kontrak *Ownable* Variabel dan Konstruktor

Beberapa fungi yang diimplementasi pada kontrak ini adalah fungsi *owner()* yang digunakan untuk mendapatkan informasi pemilik dari kontrak. Fungsi ini menjadi bagian utama dalam identifikasi pemilik kontrak/buku yang diakses pengguna. Dengan fungsi ini sistem dapat membedakan antara pengguna yang menjadi pemilik/bukan pemilik kontrak. Fungsi lain dalam kontak ini adalah *transferOwnership()*. Fungsi ini tidak kalah penting karena salah satu kebutuhan utama sistem yang dibangun adalah memindahkan hak kepemilikan buku. Peran fungsi ini adalah mengubah variable *_owner* dengan parameter yang dikirim oleh pengguna. Potongan kedua fungsi tersebut ditunjukkan pada Gambar 5.1.2.

```
function owner() public view virtual returns (address) {
    return _owner;
}

function _transferOwnership(address newOwner) public virtual onlyOwner {
      _owner = newOwner;
}
```

Gambar 5.1.2 Kode fungsi *owner()* dan *transferOwnership()*

Kontrak *Ownable* juga mendefinisikan dua *modifier* yang digunakan untuk membatasi hak akses pengguna terhadap suatu fungsi tertentu di dalam kontrak. Dua *modifier* tersebut antara lain: *onlyOwner()* yang digunakan untuk membatasi akses fungsi hanya kepada pemilik buku. Dengan kata lain, selain pemilik buku tidak dapat mengakses fungsi tersebut. Lain halnya dengan *modifier nonOwner()* yang hanya mengizinkan pengguna selain pemilik kontrak yang dapat mengakses fungsi tersebut. Potongan kode kedua *modifier* ini ditunjukkan pada Gambar 5.1.3.

```
modifier onlyOwner() {
  require(owner() == msg.sender, "the guest not allowed.");
  _;
}

modifier nonOwner() {
  require(owner() != msg.sender, "the owner not allowed");
  _;
}
```

Gambar 5.1.3 Kode *modifier onlyOwner()* dan *nonOwner()*

Kontrak *book* adalah bagian utama kode yang dieksekusi oleh luar SC. Kontrak ini terdiri dari variabel dan fungsi yang mengimplementasi *Library* kontrak *Ownable,* sehingga semua fungsi, *modifier*, dan variabel yang bersifat publik atau internal dapat digunakan dalam kontrak *book*. Terdapat beberapa variabel yang diidentifikasi dalam kotrak ini, tetapi variabel utama yang digunakan untuk bertransaksi adalah *documentHash*. Variabel ini berisi nilai terenkripsi dari proses enkripsi hash file asli yang diunggah ke CSIPFS dengan kunci publik pemilik. Gambar 5.1.4 menunjukkan potongan kode kontrak buku dengan beberapa variabel yang dideklarasikan.

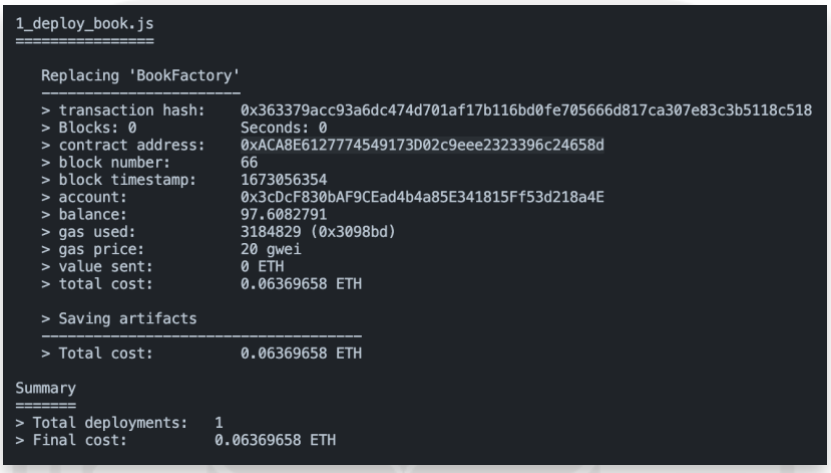
```
import "./Ownable.sol";

contract Book is Ownable {
    string public title;
    string public author;
    address public author_account;
    string public publisher;
    string public releaseDate;
    string public isbn;
    string public cover;
    string public description;
    string private documentHash;
```

Gambar 5.1.4 Definisi Kontrak Book dan Variabel

Kontrak ini menginisialisasi semua variabel dalam fungsi konstruktor yang diekekusi pertama kali secara otomatis ketika kontrak dibuat/di-*deploy*. Dalam fungsi ini, akan dipanggil fungsi *_transferOwnership()* untuk memindahkan hak kepemilikan kontrak kepada akun pembuat kontrak. Hal ini dilakukan karena pada pada dasarnya pemilik kontrak adalah akun yang melakukan *deploy* terhadap kontrak tersebut. Pada Gambar 5.1.5

ditunjukkan bahwa terdapat informasi mengenai *contract address* dan *account*. *Contract address* adalah akun yang melakukan *deploy* kontrak. Akun ini secara bawaan menjadi pemilik kontrak. Padahal seharusnya *account*-lah yang menjadi pemilik kontrak tersebut. Oleh karena itu, pemilik kontrak tersebut seketika kontrak dibuat pertama kali, perlu dilakukan pemindahan kepemilikan kontrak dari *contract address* ke *account* pengirim. Berkaitan dengan fungsi konstruktor pada kontrak ini ditunjukkan pada Gambar 5.1.6



Gambar 5.1.5 Informasi Kontrak

```
constructor(
        string memory _title,
        string memory _author,
        address _author_account,
        string memory _publisher,
        string memory _releaseDate,
        string memory _isbn,
        string memory _cover,
        string memory _description,
        string memory _hashDocument,
        address _owner
    ) {
        title = _title;
        author = _author;
        author_account = _author_account;
        publisher = _publisher;
        releaseDate = _releaseDate;
        isbn = _isbn;
        cover = _cover;
        description = _description;
        documentHash = _hashDocument;

        //transfer ownership to specific account
        _transferOwnership(_owner);
    }
```

Gambar 5.1.6 Konstruktor Kontrak *Book*

Entitas utama *selanjutnya* adalah *struct Requester*. *Struct* dalam solidity adalah sebuah tipe data yang dapat berisi beberapa variabel yang berbeda. Oleh karena itu, *Requester* adalah sebuah tipe data yang digunakan untuk menampung permintaan hak kepemilikan buku *customer*. Berisi beberapa informasi berupa alamat *customer*, kunci publik *customer*, dan tanggal permintaan tersebut dibuat. Oleh karena *Requester* merupakan sebuah tipe data, maka diperlukan variabel untuk dapat mengakses data dalam *Requester*. Dalam hal ini *struct Requester* dideklarasikan dalam variabel *_requester* yang ditunjukkan pada Gambar 5.1.7.

```
1  struct Requester {
2    address customer;
3    string publicKey;
4    uint256 date;
5  }
6
7  Requester private _requester;
```

Gambar 5.1.7 Deklarasi Variabel dengan Tipe Data Requester

Terdapat dua fungsi utama dalam kontrak ini yang digunakan untuk proses pertukaran hak kepemilikan buku, yaitu fungsi pemintaan dan penerimaan hak kepemilikan. Fungsi permintaan didefinisikan dengan fungsi *requestOwner*. Fungsi ini menerima parameter dari luar berupa kunci publik *customer*. Selanjutnya kunci publik tersebut akan disimpan pada variabel *requester* bersamaan dengan akun pengirim dan waktu blok transaksi dibuat. Setelah proses penyimpanan data selesai, *event Requested* akan dipanggil sebagai bukti bahwa proses penyimpanan *requester* berhasil. Fungsi ini hanya dapat diakses oleh pengguna selain pemilik kontrak yang didefinisikan dengan *modifier nonOwner* dan akan membatasi pengguna dengan hanya dapat melakukan satu kali permintaan. Setelah permintaan dibuat, selanjutnya pengguna akan menerima permintaan tersebut melalui fungsi *acceptRequest* yang hanya dapat diakses oleh pemilik kontrak/buku yang ditandai dengan *modifier onlyOwner*. Fungsi ini menerima parameter dari luar berupa *documentHash*. Parameter ini berisi data lokasi file berupa hash yang telah dienkripsi dengan kunci publik customer. Parameter ini selanjutnya akan disimpan dalam variable *documentHash* sebagai data blok baru dalam transaksi yang dilakukan. Setelah itu, pemilik buku dipindahkan kepada *customer* melalui fungsi *transferOwnership()* kemudian *event RequestAccepted()*

dipanggil sebagai bukti bahwa proses berhasil dilakukan. Tahap terahir dalam fungsi ini adalah mengosongkan kembali variabel *requester* karena kedepan variabel tersebut akan diisi kembali oleh data *customer* yang meminta hak kepemilikan terhadap buku tersebut. Potongan kode kedua fungsi ini ditunjukkan pada Gambar 5.1.8.

```
function requestOwner(string memory _customerPublicKey) external nonOwner {
    require(_requester.customer != msg.sender, "you requested");
    require(_requester.customer == address(0x0), "the book requested");

    _requester.customer = msg.sender;
    _requester.publicKey = _customerPublicKey;
    _requester.date = block.timestamp;

    //emit event
    emit Requested(_requester.customer, title, _requester.date);
}

  function acceptRequest(string memory _documentHash) external onlyOwner {
      require(
          _requester.customer != address(0x0),
          "the request is not available"
      );

      documentHash = _documentHash;
      _transferOwnership(_requester.customer);

      emit RequestAccepted(_requester.customer, title, block.timestamp);

      delete _requester;
  }
```

Gambar 5.1.8 Fungsi *requestOwner* dan *acceptRequest*

5.1.2. Implementasi Sistem DRMChain

Bagian ini memuat dua komponen aplikasi yang saling mendukung satu sama lain, yaitu: aplikasi Main-App dan File-Proxy. Aplikasi Main-App berfokus interaksi antara pengguna dan SC. Sedangkan aplikasi File-Proxy berfokus pada proses pengolahan file dan interaksi antara Main-App dengan CSIPFS.

Aplikasi Main-App dibangun menggunakan *framework reactjs* di atas bahasa pemrograman typescript. Menggunakan *Library web3* untuk berkomunikasi dengan SC dan menggunakan library *axios* untuk berkomunikasi dengan aplikasi File-Proxy. Aplikasi ini terdiri dari beberapa halaman utama, yaitu: *Explore /* Katalog, *Create Book*, *My Books*, *Book Detail*, dan *Request List.* Sedangkan proses utama yang dilakukan dalam aplikasi ini adalah proses publikasi buku/pembuatan kontrak, permintaan hak kepemilikan buku, penerimaan

permintaan, dan mengunduh atau membaca dokumen. Dalam subbab ini, hanya akan dijelaskan bagian utama pada setiap proses yang dilakukan.

Dalam proses publikasi buku, terdapat dua proses utama yang dilakuan ketika mempublikasikan buku, yaitu proses pengiriman dan pengunggahan data ke aplikasi File-Proxy untuk memberi *watermark* pada buku dan mendapatkan nilai hash file, dan proses mengirimkan data dan dokumen yang terenkripsi ke SC dan disimpan di dalam blockchain. Dalam proses pengunggahan data ke aplikasi File-Proxy, langkah pertama adalah mengisi data yang dibutuhkan. Data tersebut berupa judul, penulis, penerbit, tanggal terbit, isbn, gambar sampul, dan dokumen file yang dibungkus dalam format JSON. Data tersebut kemudian dikirim ke aplikasi File-Proxy dan diteruskan ke CSIPFS. Apabila proses tersebut berhasil, aplikasi File-Proxy akan mengembalikan data hash file yang merupakan lokasi file asli. Potongan kode pada proses ini ditunjukkan pada Gambar 5.1.9.

```
const handleSubmit = async (event: React.FormEvent<HTMLFormElement>) => {
  event.preventDefault();

  // validasi eksistensi variabel file
  if (files.file === undefined) return;

  // deklarasi variabel file
  const file: File = files.file;

  // membuat objek formdata
  var formData = new FormData();

  // membuat / mengisi data yang dibutuhkan dalam format json
  const jsonMetaData = {
    title: title,
    author: author,
    publisher: publisher,
    releaseDate: releaseDate,
    isbn: isbn,
    cover: cover,
  };

  // memasukkan nilai formdata
  formData.append("metadata", JSON.stringify(jsonMetaData));
  formData.append("document", file);

  // inisialisasi nilai hashFile
  let hashFile = "";

  // proses pengiriman data dan file ke aplikasi File-Proxy
  await axios
    .post(`${fileProxyEndPoint}/upload`, formData, {
      headers: {
        "Content-Type": "multipart/form-data",
      },
    })
    .then((response) => {
      // memasukkan hash file setelah proses pengiriman data berhasil
      // jika tidak berhasil, akan ditampilkan ke log
      if (response.status === 200) hashFile = response.data;
    })
    .catch(console.error);
```

Gambar 5.1.9 Potongan Kode Proses Pengiriman Data ke Aplikasi File-Proxy

Setelah mendapatkan nilai hash file, langkah berikutnya adalah melakukan proses enkripsi hash file dengan kunci publik pengguna untuk mendapatkan string hash terenkripsi. Implementasi kode yang dilakukan, memanfaatkan *library EthCrypto* untuk mengenkripsi nilai hash file dengan kunci publik pengguna. Gambar 5.1.10 menunjukkan potongan kode fungsi enkripsi hash file.

```
export const Encrypt = async (hashFile, publicKey) => {
  try {
    //membuat string payload
    const payloadString = JSON.stringify({
      hashFile: hashFile,
    });

    //enkripsi payload dengan kunci publik
    const encrypted = await EthCrypto.encryptWithPublicKey(
      publicKey,
      payloadString
    );

    //konversi hasil enkripsi ke string
    const hashFileEncrypted = EthCrypto.cipher.stringify(encrypted);

    //menembalikan variabel hashFileEncrypted
    return hashFileEncrypted;
  } catch (error) {
    console.info(error);
    return;
  }
};
```

Gambar 5.1.10 Potongan Kode Fungsi Enkripsi Hash File

Selanjutnya apabila proses enkripsi berhasil dilakukan, langkah terakhir dalam proses ini adalah pembuatan kontrak/publish buku dengan mengirimkan data ke SC. Potongan kode pada proses ini ditunjukkan pada Gambar 5.1.11.

```
    // proses enkripsi hashfile menggunakan konci publik pengguna
    let encryptedHash = await Encrypt(hashFile, GetPublicKey(privateKey));

    // validasi jika proses enkripsi gagal
    if (!encryptedHash) Alert("encryption process failed.");

    // mengirim data ke Smart Contract / menyimpan data dalam blockchain
    const tx = await contract.methods
      .createBook(
        title,
        author,
        authorAccount,
        publisher,
        releaseDate,
        isbn,
        cover,
        description,
        encryptedHash
      )
      .send({ from: account });
```

Gambar 5.1.11 Proses enkripsi Hash File dan Pengiriman Data ke Smart Contract

Proses selanjutnya pada aplikasi Main-App adalah melakukan permintaan hak kepemilikan buku. Kode yang dibutuhkan dalam proses ini adalah mengirimkan data ke fungsi *requestOwner* dalam SC. Data yang butuhkan hanya berupa kunci publik *customer*. Potongan kode fungsi dalam proses ini ditunjukkan pada Gambar 5.1.12.

```
// mendapatkan kunci publik dari kunci privat pengguna
const publicKey = GetPublicKey(privateKey);

// membuat permintaan hak kepemilikan pada kontrak
const result = await props.contract.methods
  .requestOwner(publicKey)
  .send({ from: props.account });
```

Gambar 5.1.12 Potongan Kode Untuk Membuat Permintaan Hak Kepemilikan

Berkenaan dengan penerimaan permintaan hak kepemilikan dalam aplikasi Main-App sebagaimana ditunjukkan oleh diagram sequence pada Gambar 4.2.4, tahap pertama dalam proses ini adalah mengambil hash dokumen dari SC yang kemudian hash tersebut

didekripsi untuk mendapatkan hash file asli. Potongan kode untuk proses dekripsi hash dokumen ditunjukkan pada Gambar 5.1.13.

```
export const Decrypt = async (hashDocument, privateKey) => {
  try {

    // konversi string hashDocument ke bentuk objek
    const encryptedObject = EthCrypto.cipher.parse(hashDocument);

    // proses dekripsi objek hashDocument menggunakan kunci private pengguna
    const decrypted = await EthCrypto.decryptWithPrivateKey(
      privateKey,
      encryptedObject
    );

    // konversi hasil string dekripsi ke format json
    const decryptedPayload = JSON.parse(decrypted);

    // mengembalikan data hashFile
    return decryptedPayload.hashFile;
  } catch (error) {
    console.info(error);
    return;
  }
};
```

Gambar 5.1.13 Potongan Kode Fungsi Dekripsi *Hash Document*

Setelah hash file asli berhasil didapatkan, hash file tersebut kemudian dienkripsi kembali menggunakan kunci publik *customer* yang melakukan permintaan (*requester)* dan selanjutnya hash dokumen terenkripsi tersebut dikirim kembali ke SC. Potongan kode proses penerimaan permintaan hak kepemilikan ditunjukkan pada Gambar 5.1.14.

```
// inisialisasi variabel request pada reactjs
const [request, setRequest]  = useState({});

// memanggil data requester ke SC
await contract.methods
  .getRequest()
  .call({ from: owner })
  .then((resp: any) => {

  //data response berhasil dimasukkan ke variabel request
  setRequest({
    account: resp[0],
    publicKey: resp[1],
    documentHash: resp[2],
  });

})
  .catch((error: any) => {
  // dipanggil jika terjadi error pada proses request ke SC
  console.log(error);
});

// --- berbeda lokasi kode ---

// dekripsi hashDocument dengan kunci private pengguna
let hashFile = await Decrypt(props.requestData.documentHash, privateKey);

// enkripsi hash file dengan kunci publik customer
let newHashDocument = await Encrypt(hashFile, props.requestData.publicKey);

// mengirimkan hash dokumen baru ke Smart Contract
const tx = await props.contract.methods
.acceptRequest(newHashDocument)
.send({ from: props.owner });
```

Gambar 5.1.14 Potongan Kode Proses Penerimaan Permintaan Hak Kepemilikan

Proses terakhir dalam siklus sistem melalui aplikasi Main-App ini adalah membaca/mengunduh dokumen. Proses membaca/mengunduh dokumen, terdiri dari empat tahapan, yaitu: (i) mengambil hash dokumen dari SC, (ii) mendekripsi hash dokument menjadi hash file, (iii) mengambil file dari CSIPFS melalui File-Proxy, dan terakhir (iv) menampilkan file dokumen pada penampil PDF. Potongan kode proses ini ditunjukkan pada Gambar 5.1.15.

```
// mengambil hashDocument data dari SC
const hashDocument = await props.contract.methods
.getDocument()
.call({ from: props.owner });

if (!hashDocument) {
  Alert("Load document failed.");
  return;
}

// dekripsi hashDocument
const hashFile = await Decrypt(hashDocument, privateKey);

// mengambil file dari aplikasi File-Proxy
await axios({
  method: "post",
  url: `${fileProxyEndPoint}/get`,
  data: { hash: hashFile.hash },
  responseType: "arraybuffer",
  headers: {
    "Content-Type": "application/json",
  },
})
  .then((response) => {

  //membuat objek dari response
  var file = new Blob([response.data], { type: "application/pdf" });
  //membuat objek url
  var fileURL = URL.createObjectURL(file);
  //memanggil fungsi menampilkan dokumen
  props.onSetDocumentFile(fileURL);
  handleClose();
})
  .catch(console.error);
```

Gambar 5.1.15 Potongan Kode Proses Membaca/Mengunduh Dokumen

Berbeda dengan Aplikasi Main-App, aplikasi File-Proxy dibangun menggunakan *framework Flask* di atas bahasa pemrograman *Python3* mengimplementasi CSIPFS dengan platform Infura. Fungsi utama dari aplikasi ini adalah menghasilkan QRCode dari JSON data yang dikirim dari Main-App. Proses ini menggunakan *library qrcode* yang ditunjukkan pada Gambar 5.1.16. Selanjutnya melakukan penyisipan *watermark* QRCode pada file dokumen. Proses ini menggunakan *library PyPDF2* yang ditunjukkan pada Gambar 5.1.17. Proses terakhir adalah mengunggah/mengunduh file dari CSIPFS. Proses tersebut ditunjukkan pada Gambar 5.1.18. Semua proses tersebut dilakukan secara berurutan seperti alur yang ditunjukkan pada diagram sequence.

```python
def __generate_qrcode(self, metadata):
        watermark_path = os.path.join(
            self.tmpdir, f"watermark-{uuid.uuid4()}.pdf")
        qr = qrcode.QRCode(
            version=1,
            error_correction=qrcode.constants.ERROR_CORRECT_L,
            box_size=10,
            border=4,
        )
        qr.add_data(metadata)
        qr.make(fit=True)
        img = qr.make_image(fill_color="black", back_color="white")
        type(img)
        converted = img.convert('RGB')
        converted.save(watermark_path)

        return watermark_path
```

Gambar 5.1.16 Potongan Kode Fungsi Untuk Menghasilkan QRCode dari Data dari Klien

```python
# mempersiapkan pembuatan file pdf baru
output_pdf = PyPDF2.PdfWriter()
# proses ekstraksi dan penggabungan watermark ke file utama
for index in list(range(0, len(input_pdf.pages))):
  content_page = input_pdf.pages[index]
  mediabox = content_page.mediabox
  if (index == 0):
    content_page.merge_page(watermark_page)
    content_page.mediabox = mediabox
    output_pdf.add_page(content_page)
```

Gambar 5.1.17 Potongan Kode Proses Penyisipan *Watermark* ke Dokumen Utama

```python
def upload(self, file) -> str:

  # membuat objek files yang berisi parameter file
  files = {
    'file': file,
  }

  # mengirim file ke ipfs dengan basic autentikasi
  # menggunakan projek id dan secret id dari platform infura ipfs
  response = requests.post(
    self.__endpoint + '/api/v0/add',
    files=files,
    auth=(self.__projectId, self.__projectSecret)
  )

  # menangkap respose dengan format json
  # dan mengambil nilai hash
  body = response.json()
  hash = body['Hash']

  # mengembalikan nila hash
  return hash
```

Gambar 5.1.18 Potongan Kode Fungsi Upload File ke IPFS

5.2. Pengujian

Pada bagian ini akan dilakukan pengujian interaksi dan fungsionalitas oleh penguji dari sisi pengguna pada sistem yang dibangun. Tabel 5.2.1 menunjukkan empat skenario pengujian dalam satu siklus sukses aktivitas terhadap sistem yang dilakukan, untuk memastikan fungsi dan interaksi pada sistem berjalan sesuai harapan.

Tabel 5.2.1 Skenario Pengujian Pada Sistem DRMChain

Kasus	Deskripsi	Ekspektasi	Langkah-langkah
1	Mempublikasikan buku oleh *Author*	• Muncul pesan '*Book Created*' setelah proses berhasil • Pada halaman *mybooks* dan detail buku, *Author* dapat melihat informasi	• Masuk halaman *create* book • Masukkan data buku • Tekan tombol *'Publish'*

		buku yang sama dengan masukan dan informasi *owner* adalah akun *Author* • Informasi *console log* berisi *event* dengan nilai *'BookCreated'*, dan nilai kembalian *event* berupa judul dan pemilik sama dengan masukan	
2	Request hak kepemilikan buku oleh *Customer*	• Author mendapat pesan *'Requested'* setelah sukses melakukan permintaan • *Auhor* dapat melihat buku yang diminta oleh *customer* pada halaman *myrequests* • *Customer* dapat melihat buku yang diminta pada halaman *myrequests*	• Masuk halaman *explore* • Memilih menekan tombol *'Request Owner'* pada salah satu buku • Masukkan kunci privat *customer* pada dialog konfirmasi • Tekan tombol *'process'*
3	Menerima Permintaan buku oleh *Author*	• Informasi *owner* buku adalah *customer* • Buku Author pada halaman *mybooks* kosong • Data *Customer* pada halaman *mybooks* berisi buku yang dipindahkan	• Masuk halaman *myrequests* • Tekan tombol *'show'* pada salah satu buku yang diminta • Tekan tombol *'Accept Request'* pada halaman detail buku • Masukkan kunci privat pada dialog konfirmasi penerimaan • Tekan tombol *'process'*

4	Membaca buku oleh *Customer*	• Customer dapat membuka dokumen pada halaman detail melalui penampil PDF	• Masuk halaman *mybooks* • Tekan tombol *'more'* pada salah satu buku • Tekan tombol *'Read Book'* pada halaman detail buku • Masukkan kunci privat pada dialog konfirmasi • Tekan tombol *'Confirm'*

Pengujian ini menggunakan akun ethereum pada jaringan lokal Ganache. Sampel data yang dibutuhkan pada pengujian ini adalah dua data akun ethereum yag memuat alamat dan kunci privat serta data buku sebagai objek transaksi. Sampel data pengujian ini ditunjukkan pada Tabel 5.2.2.

Tabel 5.2.2 Sampel Data Pengujian

Nama	Keterangan
Akun 1 (*Author*)	• Alamat*: 0x3cDcF830bAF9CEad4b4a85E341815Ff53d218a4E* • Kunci Privat: *452de3bf245afa918e6c40e7811424a3e951195cf93b773a631591fdd8e81808*
Akun 2 (*Customer*)	• Alamat*: 0x9d28946786AB3F638287302FA14b3F1725c50e9f* • Kunci Privat: *3cc14abad65841533c86c1e63d184f97260fe6ed99bc6bf8cedf9fbf223df49e*
Buku	• Title: *Abandon Kingdom* • Author: *Claudia Wilson* • Author Account: *0x3cDcF830bAF9CEad4b4a85E341815Ff53d218a4E* • Publisher: *Media Publisher* • Release Date: *2023-01-01*

	• ISBN: *B0813PLG15* • Cover: *abandon-kingdom.png* • Description = *Freya McNabb was running for her life. Her brother and father wanted her to marry an evil man. They wanted it so much that her father had her beat in public each time she said not. Finally, enough was enough and she made her escape. She found herself in a cottage which was going to change her life forever. She only had to want to believe that not all were like her father and her brother. Conner McClure had vowed he would never marry. He hadn't met a woman that made him want to marry. He and his party of hunters arrived at the hunting cottage, where they found a tiny woman who had been beaten and was unconscious on the floor. Conner nursed her wounds and found she moved his soul. She was going to be his wife he decided, but only if they could face all the perils that awaited ahead. Would their love be strong enough to withstand the tests that lay before them or would it be destroyed forever.* • File: *abandon-kingdom.pdf*

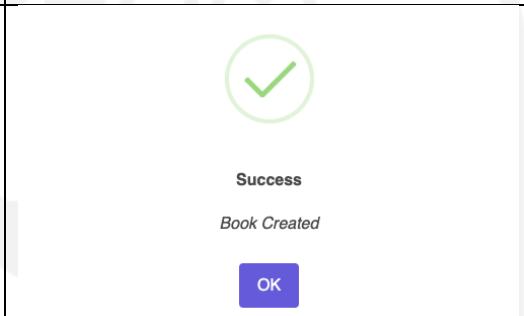
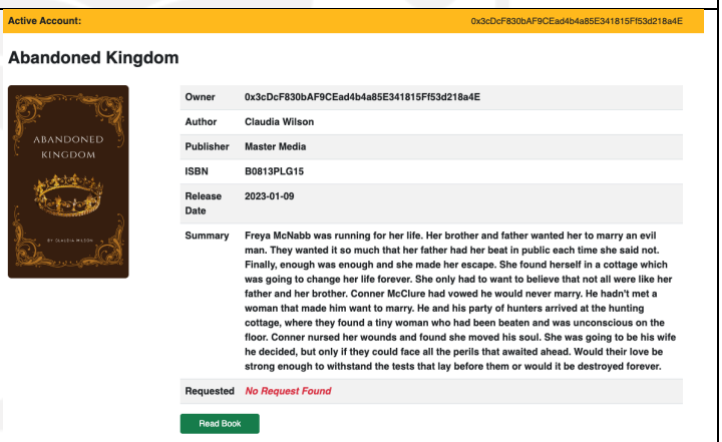
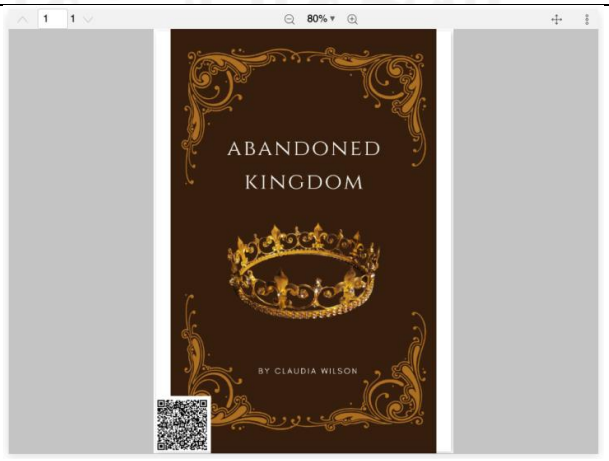
1. Hasil pengujian kasus 1

Pengujian yang pertama ini adalah menguji keberhasilan sistem dalam proses mempublikasi buku oleh *author*. Tabel 5.2.3 menunjukkan bahwa semua skenario yang diujikan berhasil dilakukan. Hal tersebut terbukti dengan informasi log pada *console browser* menampilkan informasi *event* "*BookCreated*". Sebagaimana fungsi *createBook()* pada SC memanggil *event* "*BookCreated*" ketika proses pembuatan buku berhasil dilakukan. Selain itu sistem juga memunculkan pesan sukses berupa "*Book Created*" setelah proses dilakukan. Pada halaman *mybooks* dan detail buku juga menampilkan buku yang baru saja dipublikasikan dengan nilai *attribute owner* pada buku tersebut adalah akun *author*. Selain itu, ditunjukkan juga bahwa terdapat sisipan QRCode pada buku yang dibuka.

Tabel 5.2.3 Hasil Pengujian Kasus 1

Ekspektasi	Status	Hasil

Author console log berisi *event* dengan nilai 'BookCreated', dan nilai kembalian event berupa judul, pemilik, dan tanggal buku dibuat	OK	"events": {
		"BookCreated": {
		"logIndex": 0,
		"transactionIndex": 0,
		"transactionHash": "0x19fd72382f2e0ee63f24fcd0d263f7f31dc97d992e170a57d209bfdaa80ac513",
		"blockHash": "0xa0d8491bbf2237b2c2c77819c1cd76992623984062b23fcfa88c91b97d51e39c",
		"blockNumber": 85,
		"address": "0x7c24256CA6D9B364ce180Cec63e479c6c68D4936",
		"type": "mined",
		"id": "log_f1f353bd",
		"returnValues": {
		"0": "Abandoned Kingdom",
		"1": "0x3cDcF830bAF9CEad4b4a85E341815Ff53d218a4E",
		"2": "1673238550",
		"title": "Abandoned Kingdom",
		"owner": "0x3cDcF830bAF9CEad4b4a85E341815Ff53d218a4E",
		"date": "1673238550"
		},
		"event": "BookCreated",
		"signature": "0x2c3d8654c722907d677c4b23bb45791dd5537752c42883afd760f076d69436b4",

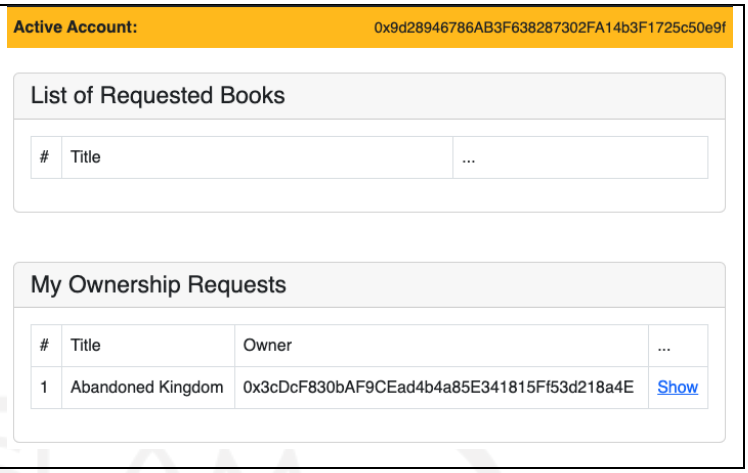
Muncul pesan 'Book Created' setelah proses berhasil	OK	
Pada halaman mybooks dan detail buku, Author dapat melihat informasi buku yang sama dengan masukan dan informasi owner adalah akun *Author*	OK	
Author dapat buka dokumen dan dokumen berisi watermark QRCode	OK	

2. Hasil pengujian kasus 2

Pengujian kasus 2 adalah menguji keberhasilan sistem terhadap *customer* yang melakukan permintaan pemindahan hak kepemilikan buku kepada *author*/*owner*. Tabel 5.2.4 menunjukkan bahwa semua skenario pengujian pada kasus ke 2 berhasil dilakukan. Terbukti ketika *customer* selesai melakukan permintaan, informasi log pada *console browser* menampilkan informasi *event* "*Requested*". Hal ini sudah sesuai dengan fungsi *requestOwner* pada SC yang akan menjalankan *event Requested* setelah semua proses dalam fungsi tersebut berhasil dieksekusi. Hal ini juga ditunjukkan dengan munculnya pesan sukses "*The book is requested*". Bukti lain adalah munculnya data buku yang diminta pada halaman myrequests bagian "*My Ownership Requests"*

Tabel 5.2.4 Hasil Pengujian Kasus 2

Ekspektasi	Status	Hasil
Customer console log berisi *event* dengan nilai 'Requested, dan nilai kembalian event berupa akun customer, judul, dan tanggal request	OK	
Customer mendapat pesan 'The Book is Requested' setelah sukses melakukan permintaan	OK	

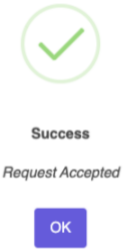
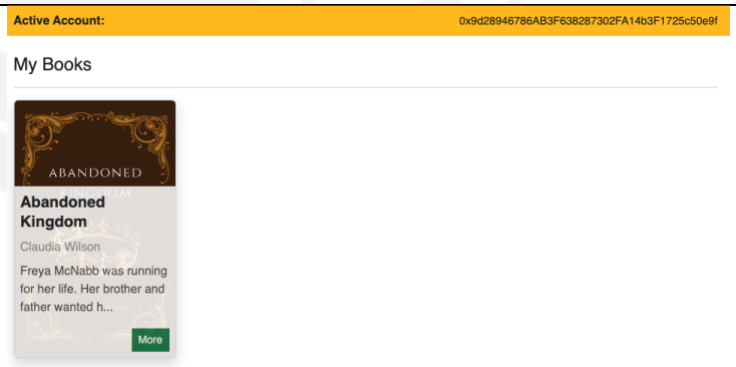
Customer dapat melihat buku yang diminta pada halamalan myrequests	OK	

3. Hasil pengujian kasus 3

Pada pengujian kasus ketiga ini, dilakukan pengujian dalam proses menerima permintaan hak kepemilikan oleh *author* terhadap permintaan *customer*. Dari hasil pengujian kasus yang ditunjukkan pada Tabel 5.2.5, diketahui bahwa semua skenario yang diujikan berhasil dilakukan. Bukti yang ditunjukkan pada hasil pengujian tersebut adalah informasi log transaksi yang ditampilkan pada *console browser*. Pada log terdapat informasi *event* yang berisi "*RequestAccepted*". Hal ini sesuai dengan fungsi *acceptRequest* pada SC yang mengeksekusi *event* tersebut setelah semua kode berhasil dieksekusi. Sistem juga menampilkan pesan sukses "*Request Accepted*" serta data buku dalam halaman *mybooks* juga sudah tidak ada lagi. Sedangkan data buku yang telah dipindahkan oleh *author* tersebut, muncul pada halaman *mybooks* akun *customer*.

Tabel 5.2.5 Hasil Pengujian Kasus 3

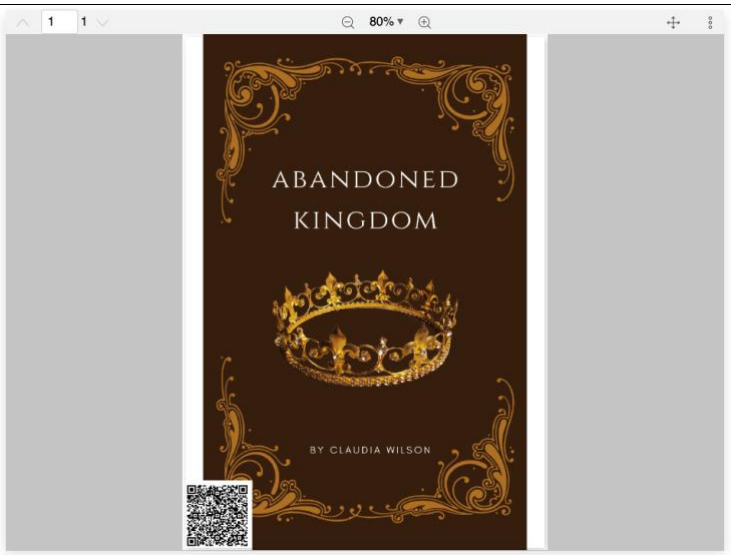
Ekspektasi	Status	Hasil
Author console log berisi *event* dengan nilai 'RequestAccepted, nilai kembalian event berupa akun *customer*, judul, dan tanggal terima	OK	

Author Mendapat Pesan Sukses "Request Accepted"	OK	
Daftar Author pada halaman mybooks kosong	OK	
Informasi *owner* buku yang setelah dipindahkan adalah *customer*	OK	
Data *Customer* pada halaman mybooks berisi buku yang dipindahkan	OK	

4. Hasil pengujian kasus 4

Pengujian terakhir pada sistem ini adalah menguji apakah buku yang telah dipindahkan berhasil dibuka oleh *customer*. Tabel 5.2.6 menunjukkan bahwa buku yang telah dimiliki *customer* tersebut berhasil dibuka pada penampil PDF. Pada dokumen tersebut juga menunjukkan QRCode yang disisipkan.

Tabel 5.2.6 Hasil Pengujian Kasus 4

Ekspektasi	Status	Hasil
Customer dapat membuka dokumen pada halaman detail melalui penampil pdf	OK	

5.3. Evaluasi

Bagian evaluasi ini akan dilakukan analisis dan pembuktian terhadap rancangan yang dibuat berdasarkan hasil identifikasi pada Bab 1. Tujuan evaluasi ini adalah untuk memastikan apakah konsep DRMChain dapat menyelesaikan permasalahan hak cipta digital. Pada hasil identifikasi yang didapatkan, diketahui beberapa tindakan yang menjadi kendala dan permasalahan sistem DRM, termasuk di dalamnya memuat permasalahan kasus pelanggaran hak cipta digital di Indonesia. Tabel 1.1.2 menunjukkan beberapa kriteria tindakan yang menjadi alat ukur evaluasi sistem DRMChain yang dibuat untuk mencapai sistem DRM yang ideal.

Pada sistem DRMChain yang dibangun, pembatasan mengunduh, distribusi, dan modifikasi sama dengan sistem DRM biasa yaitu dengan proses enkripsi dan *watermarking*. Enkripsi pada konten memungkinkan seseorang tidak dapat membuka konten tersebut tanpa memiliki kunci yang tepat untuk membukanya. Enkripsi pada konten menjadi fitur yang berguna untuk membatasi seseorang mengunduh file. Hal ini karena dengan memberi enkripsi pada konten, akan membutuhkan upaya yang cukup keras untuk dapat membuka enkripsi tersebut tanpa memiliki kuncinya. Walaupun file tersebut pada akhirnya dapat dibuka, masih tersemat *watermark* di dalamnya sehingga memerlukan upaya lagi untuk merusak/menghilangkan *watermark* pada konten tersebut agar selanjutnya dapat didistribusikan dengan aman. Dengan kedua teknik tersebut, upaya perlindungan konten

digital dari tindakan mengunduh, dan distribusi secara ilegal, cukup dapat dilakukan dengan baik.

Pada sistem yang dibangun, penggunaan enkripsi dan *watermarking* juga diimplementasi untuk membatasi tindakan mengunduh, distribusi, dan modifikasi secara ilegal. Teknik enkripsi yang digunakan adalah menggunakan kriptografi kunci publik dan menggunakan *visibility watermarking* dalam bentuk QRCode yang disisipkan di dalam konten. Enkripsi dalam sistem ini diterapkan pada *hashfile* konten asli. *Hashfile* tersebut merupakan sebuah alamat file yang diunggah ke CSIPFS, sehingga untuk mengaksesnya melalui protokol *http*. Tabel 5.3.1 menunjukkan hasil enkripsi *hashfile* pada sistem menggunakan kunci publik pemilik konten.

Tabel 5.3.1 Enkripsi *Hashfile* Pada Sistem DRMChain Menggunakan Kunci Publik.

Akun *(owner)*	*0x88949744700f697259274f766815970d8581556F*
Hashfile	*QmZnFtvTXGeckyNxoGSuzTjZ4xL2o4yi1nZ8c9sC8oJj56*
Kunci Publik	*7d63acaa9be0fddf7c4351b075b456928b14c9723d87bc1b1c9eced3abe c73b7fc28f6ea772ad85ef3544a80e487c642a8bea9a5b6f300e4b7ca073 8b0cca853*
Hashfile Terenkripsi	*45813ea274bda155ad0b11f671efd28803ab326cb691353dd9b0c9b73f1 185081311b9c0f27242ff536ceeb1850908f509fd9625509e3ed59919236 bbdf029d3d44df2d53eafb5678e6c1156b1b790f1c952ea230956da923c b4902178927eb6497dc70cad3464fdf0fc2310d2b6c5a41b5c8aa24f5c9d 9c48399f2b30ddad848974aa477ef747683263293bf85072c614977be12 203a5ae09569ab89583b2b73f*

Hasil enkripsi *hashfile* yang ditunjukkan dalam tabel Tabel 5.3.1 hanya dapat dibuka oleh orang yang memiliki kunci privat dari kunci publik tersebut. Seseorang yang hanya memegang *hashfile* terenkripsi tersebut akan sangat sulit baginya untuk mendapatkan hashfile aslinya tanpa kunci privatnya. Selanjutnya Tabel 5.3.2 menunjukkan hasil dekripsi *hashfile* terenkripsi dengan kunci privat. Hasil yang didapatkan dari proses dekripsi tersebut adalah nilai *hashfile* asli.

Tabel 5.3.2 Dekripsi *Hashfile* Terenkripsi Pada Sistem DRMChain Menggunakan Kunci Privat.

Akun *(owner)*	*0x88949744700f697259274f766815970d8581556F*
Hashfile Terenkripsi	*45813ea274bda155ad0b11f671efd28803ab326cb691353dd9b0c9b73f1 185081311b9c0f27242ff536ceeb1850908f509fd9625509e3ed59919236 bbdf029d3d44df2d53eafb5678e6c1156b1b790f1c952ea230956da923c b4902178927eb6497dc70cad3464fdf0fc2310d2b6c5a41b5c8aa24f5c9d 9c48399f2b30ddad848974aa477ef747683263293bf85072c614977be12 203a5ae09569ab89583b2b73f*
Kunci Privat	*92cabb64d8104e0590e3568363207ad4a8659ab0088fd9e2b83e86cb8b 215417*
Hashfile	*QmZnFtvTXGeckyNxoGSuzTjZ4xL2o4yi1nZ8c9sC8oJj56*

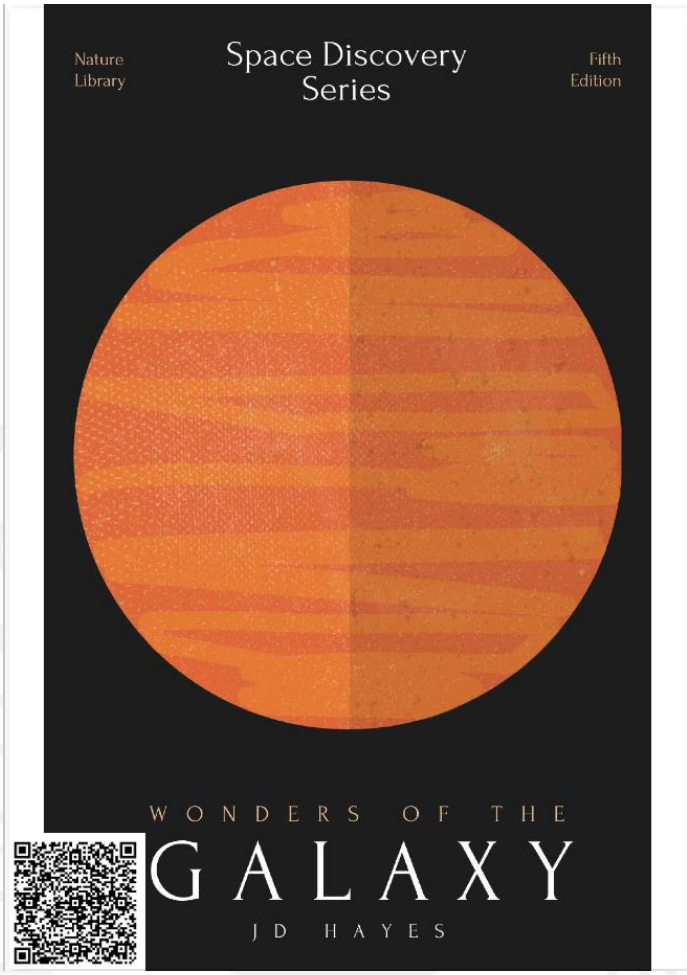
Selanjutnya bagaimana jika seseorang memasukkan kunci privat yang salah dalam proses dekripsi hashfile. Tabel 5.3.3 menunjukkan hasil berupa *error Bad Mac* ketika satu karakter terakhir pada kunci privat diganti. Artinya fungsi tidak dapat mendekripsi *hashfile* terenkripsi dengan sembarang kunci privat/kunci privat masukan tidak sesuai. Dengan begitu dapat dikatakan bahwa mekanisme enkripsi yang dibuat berjalan dengan baik, sehingga dapat digunakan untuk melindungi konten digital dari pencurian.

Tabel 5.3.3 Mendekripsi *Hashfile* Terenkripsi pada Sistem DRMChain Menggunakan Kunci Privat yang Salah.

Akun *(owner)*	*0x88949744700f697259274f766815970d8581556F*
Hashfile Terenkripsi	*45813ea274bda155ad0b11f671efd28803ab326cb691353dd9b0c9b73f1185081311b9c0f27242ff536ceeb1850908f509fd9625509e3ed59919236bbdf029d3d44df2d53eafb5678e6c1156b1b790f1c952ea230956da923cb4902178927eb6497dc70cad3464fdf0fc2310d2b6c5a41b5c8aa24f5c9d9c48399f2b30ddad848974aa477ef747683263293bf85072c614977be12203a5ae09569ab89583b2b73f*
Kunci Privat Diganti	*92cabb64d8104e0590e3568363207ad4a8659ab0088fd9e2b83e86cb8b215418*

Hasil	`Error: Bad MAC`
	`at assert (browser.js:12:1)`
	`at browser.js:241:1`
	`at async Decrypt (Security.js:39:1)`
	`at async handleVerify (DialogReadConfirm.tsx:33:1)`

Selain melakukan enkripsi pada konten digital, sistem DRMChain yang dibuat juga mengimplementasi *watermarking.* Proses *watermarking* dilakukan melalui aplikasi File-Proxy. Sebagaimana disinggung sebelumnya bahwa teknik *watermarking* yang diterapkan pada sistem ini adalah *visibility watermarking* berbentuk QRCode. QRCode ini berisi informasi mengenai buku yang diunggah. Gambar 5.3.1 Dokumen yang Telah Diberi *Watermark* menunjukkan hasil *watermarking* pada dokumen. Sedangkan Gambar 5.3.2 menunjukkan *watermark* dan hasil ekstraksi *watermark* menggunakan aplikasi barcode reader.



Gambar 5.3.1 Dokumen yang Telah Diberi Watermark

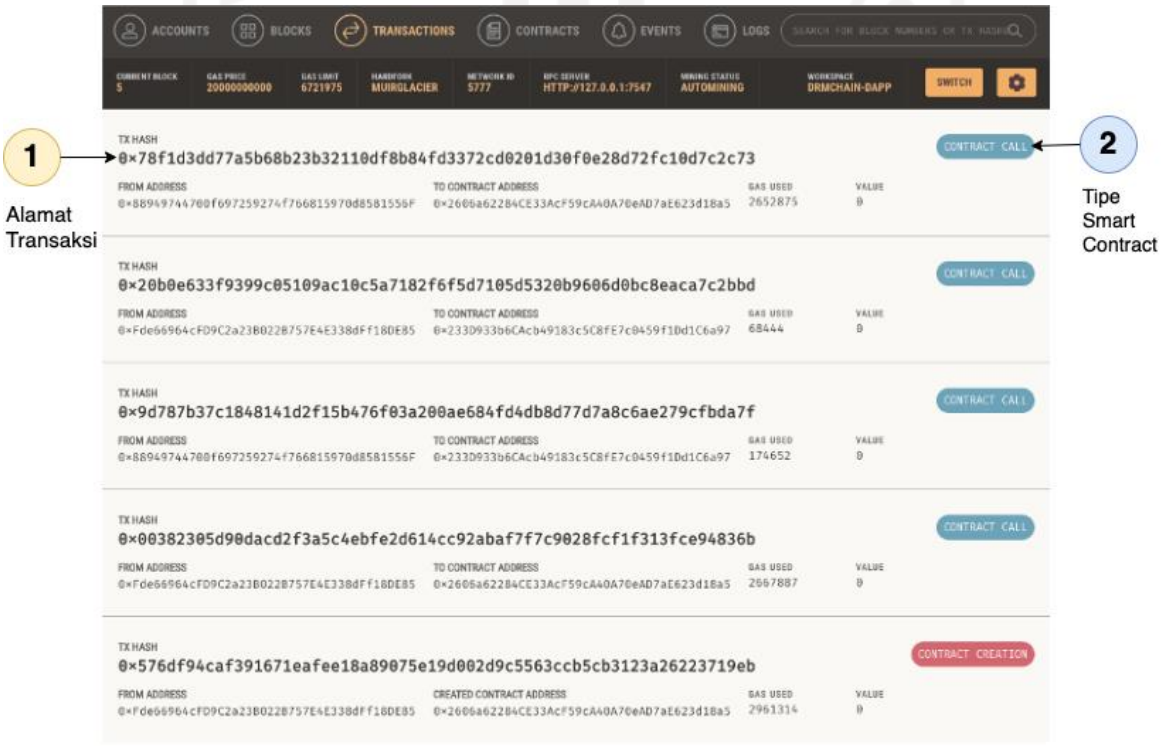

```
{
  "title": "Wonders of The Galaxy",
  "author": "JD Hayes",
  "author_account": "0x88949744700f697259274f766815970d8581556F",
  "publisher": "Galaxy Media Publisher",
  "releaseDate": "2023-01-12",
  "isbn": "B0813PLG15",
  "cover": "galaxy.png"
}
```


Gambar 5.3.2 Gambar *Watermark* pada Dokumen dan Hasil Ekstraksi

Penyematan *watermark* pada konten ini berguna untuk mengidentifikasi pencipta dan membuktikan keaslian konten. Namun, seperti yang telah disinggung sebelumnya, apabila bukti keaslian konten tersebut dirusak atau dihilangkan oleh pihak yang tidak bertanggung jawab, maka membuktikan keaslian dan identifikasi pencipta tentu tidak lagi

dapat dilakukan. Oleh karena itu, sistem blockchain diadopsi untuk menyelesaikan permasalahan tersebut. Sebagaimana telah dijelaskan, bahwa transaksi di dalam blockchain akan disimpan secara kronologis sesuai urutan blok yang dibuat dengan mereferensi blok sebelumnya. Apabila salah satu data di dalam blok diubah, maka akan memutuskan rantai blockchain. Selain itu, mekanisme kunci publik dengan menyematkan tanda tangan pengirim pada setiap transaksi di dalam SC, menjadi bukti yang sah bahwa transaksi benar-benar dilakukan oleh pihak terkait. Dengan kedua mekanisme tersebut, pelacakan transaksi di dalam blockchain dapat dimanfaatkan untuk membuktikan keaslian, mengidentifikasi pencipta, serta untuk mengidentifikasi pelaku pelanggaran.

Pada sistem yang dibangun ini akan ditunjukkan bagaimana melacak transaksi yang terjadi di dalam sistem DRMChain. Sistem ini memanfaatkan fitur *event* pada solidity untuk menyebarkan pesan atau argumen melalui log transaksi yang disimpan di dalam blockchain setelah sebuah transaksi dilakukan. Meskipun setiap transaksi di dalam blockchain dapat di lihat, akan tetapi dengan fitur *event* ini memungkinkan informasi transaksi lebih jelas, sehingga mempermudah dalam proses pelacakan transaksi. Fitur *event* ini dapat di akses menggunakan alamat dari kontrak yang di-*deploy*. Gambar 5.3.3 menunjukkan informasi transaksi yang dilakukan pada uji skenario pada Tabel 5.2.1. Informasi tersebut dipresentasikan menggunakan Ganache.



Gambar 5.3.3 Ganache Menampilkan Informasi Keseluruhan Transaksi yang Dilakukan

Transaksi pertama pada kontrak yang ditunjukkan adalah membuat kontrak (*contract creation*). Ganache menunjukkan bahwa kontrak tersebut dibuat oleh akun *0xFde66964cFD9C2a23B022B757E4E338dFf18DE85*. Transaksi selanjutnya adalah mempublikasi buku. Gambar 5.3.4 menunjukkan informasi detail transaksi yang dilakukan. Sayangnya, ganache hanya menampilkan informasi *encoded event*, sehingga memerlukan proses *decode* untuk dapat melihat informasi di dalamnya. Hasil *decoded event* ditunjukkan pada Gambar 5.3.5. Dalam informasi tersebut terdapat attribut *event* "*BookCreated*" dengan nilai kembalian berupa *title*, *owner*, dan *date*. Hal ini menandakan bahwa proses publikasi berhasil dilakukan.

Gambar 5.3.4 Informasi *Encoded Event* pada Ganache


```
{
  "BookCreated": [
    {
      "logIndex": 0,
      "transactionIndex": 0,
      "transactionHash": "0x00382305d90dacd2f3a5c4ebfe2d614cc92abaf7f7c9028fcf1f313fce94836b",
      "blockHash": "0x58c498cafad3a5583d5bcd99f2d18a641a196e9119ec2626264105d656a00e1e",
      "blockNumber": 2,
      "address": "0x233D933b6CAcb49183c5C8fE7c0459f1Dd1C6a97",
      "type": "mined",
      "id": "log_b6e5d6ea",
      "returnValues": {
        "0": "Abandoned Kingdom",
        "1": "0xFde66964cFD9C2a23B022B757E4E338dFf18DE85",
        "2": "1673401373",
        "title": "Abandoned Kingdom",
        "owner": "0xFde66964cFD9C2a23B022B757E4E338dFf18DE85",
        "date": "1673401373"
      },
      "event": "BookCreated",
      "signature": "0x2c3d8654c722907d677c4b23bb45791dd5537752c42883afd760f076d69436b4",
      "raw": {
        "data": "0x0000000000000000000000000000000000000000000000000000000000000060000000000000000000000000fde66964cfd9c2a23b022b757
        "topics": [
          "0x2c3d8654c722907d677c4b23bb45791dd5537752c42883afd760f076d69436b4"
        ]
      }
    }
  ],
```


Gambar 5.3.5 Informasi Hasil dari Proses *Decoded Event*

Transaksi lain yang dapat dilihat pada kontrak buku adalah permintaan hak kepemilikan dengan mengeksekusi *event* "*Requested*" setelah transaksi permintaan berhasil dilakukan. Begitu juga dengan transaksi penerimaan dengan mengeksekusi *event* "*RequestAccepted*". Gambar 5.3.6 menunjukkan informasi event yang telah dilakukan.


```
"RequestAccepted": [
 {
  "logIndex": 0,
  "transactionIndex": 0,
  "transactionHash": "0x20b0e633f9399c05109ac10c5a7182f6f5d7105d5320b9606d0bc8eaca7c2bbd",
  "blockHash": "0x08a40baa4019dba0bf7ff7a8f7cbe77666b6097e8ea46ef3c485cdeca1dd109f",
  "blockNumber": 4,
  "address": "0x233D933b6CAcb49183c5C8fE7c0459f1Dd1C6a97",
  "type": "mined",
  "id": "log_5f42c0c9",
  "returnValues": {
   "0": "0xFde66964cFD9C2a23B022B757E4E338dFf18DE85",
   "1": "0x88949744700f697259274f766815970d8581556F",
   "2": "Abandoned Kingdom",
   "3": "1673406940",
   "from": "0xFde66964cFD9C2a23B022B757E4E338dFf18DE85",
   "to": "0x88949744700f697259274f766815970d8581556F",
   "title": "Abandoned Kingdom",
   "date": "1673406940"
  },
  "event": "RequestAccepted",
  "signature": "0x05ed72fb067c347f6105353bae5a89a2a6007f55cc0cca8d3ae682196e553444",
  "raw": {
   "data": "0x000000000000000000000000fde66964cfd9c2a23b022b757e4e338dff18de850000000000000000000000008894974
   "topics": [
    "0x05ed72fb067c347f6105353bae5a89a2a6007f55cc0cca8d3ae682196e553444"
   ]
  }
 }
],
"Requested": [
 {
  "logIndex": 0,
  "transactionIndex": 0,
  "transactionHash": "0x9d787b37c1848141d2f15b476f03a200ae684fd4db8d77d7a8c6ae279cfbda7f",
  "blockHash": "0xfdc29beab06eab4b3b2baae019e763c822ebc087e2313bd56e69a6216dea9ee4",
  "blockNumber": 3,
  "address": "0x233D933b6CAcb49183c5C8fE7c0459f1Dd1C6a97",
  "type": "mined",
  "id": "log_f8a1a1a4",
  "returnValues": {
   "0": "0x88949744700f697259274f766815970d8581556F",
   "1": "Abandoned Kingdom",
   "2": "1673406845",
   "customer": "0x88949744700f697259274f766815970d8581556F",
   "title": "Abandoned Kingdom",
   "date": "1673406845"
  },
  "event": "Requested",
  "signature": "0x129f67783f1fc54762392a484603d038c3007571edd583280de130581c11f1c2",
  "raw": {
   "data": "0x000000000000000000000000008894974700f697259274f766815970d8581556f000000000000000000000000000000
   "topics": [
    "0x129f67783f1fc54762392a484603d038c3007571edd583280de130581c11f1c2"
   ]
  }
```


Gambar 5.3.6 Informasi Event Requested dan RequestAccepted

Berdasarkan penjelasan tersebut, maka proses identifikasi pencipta dan pembuktian keaslian konten dapat dilakukan dengan cara menelusuri transaksi yang dilakukan oleh *author* melalui fitur log pada blockchain. Akan tetapi pembuktian terhadap pelaku tindak pelanggaran (pembajakan) hak cipta digital akan sulit dilakukan oleh sistem. Hal ini karena tindakan pelanggaran umumnya dilakukan di luar sistem, seperti menggandakan atau memanipulasi konten. Apabila tindakan tersebut dilakukan di luar sistem, maka sistem tidak lagi punya kendali atas tindakan pelanggaran tersebut. Sebagai gantinya, segala permasalahan di luar sistem ini dapat diserahkan ke pihak berwenang. Dalam hal ini, sistem dapat menjadi alat bantu untuk melacak pelaku palanggaran seraya berasumsi bahwa setiap pengguna yang pernah memegang hak kepemilikan terhadap konten berpotensi melakukan pelanggaran. Dengan begitu, penelusuran pelaku pelanggaran dapat diseleksi secara lebih mudah. Atas dasar evaluasi ini, disimpulkan bahwa konsep DRMChain ini layak menjadi solusi yang ideal untuk melindungi hak cipta konten digital.

BAB 6
Kesimpulan dan Saran

6.1. Kesimpulan

Penelitian ini diajukan untuk membuktikan apakah sistem DRMChain dapat menyelesaikan permasalahan hak cipta digital di Indonesia sesuai hukum positif yang berlaku. Kriteria pembuktian pada sistem DRMChain yang dilakukan dalam penelitian ini adalah (i) mampu melindungi konten digital dari unduh, distribusi, dan modifikasi secara ilegal, (ii) mampu mengidentifikasi pencipta, (iii) mampu membuktikan keaslian, (iv) membuktikan pelanggaran, dan (v) dapat memindahkan hak kepemilikan konten ke orang lain. Implementasi dalam sistem ini dibagi dalam beberapa komponen, yaitu: komponen aplikasi DRM yang terdiri dari aplikasi Main-App dan File-Proxy, selanjutnya menggunakan CSIPFS infura sebagai tempat penyimpanan file, dan platform. Aplikasi SC ditulis menggunakan bahasa pemrograman solidity yang dibuat menggunakan *Truffle Framework*. Hasil evaluasi penelian ini mengungkapkan bahwa pembuktian konsep berhasil dilakukan. Akan tetapi, pembuktian pelanggaran tidak dapat dilakukan karena pelanggaran tersebut berada di luar sistem yang dibuat. Namun, sistem yang dibuat dapat diandalkan dalam proses penelusuran pelaku pelanggaran (pembajakan). Dengan begitu konsep penggabungan sistem DRM dengan blockchain (DRMChain) ini dapat dikatakan layak menjadi solusi yang ideal untuk melindungi hak cipta digital di Indonesia.

6.2. Saran

Penelitian ini masih memiliki berbagai kekurangan yang perlu diperbaiki atau dioptimasi, berikut beberapa hal yang disarankan untuk penelitian lebih lanjut:

1. Berkenaan dengan penyisipan *watermark* pada file PDF, dalam penelitian ini masih sebatas implementasi dasar teknik penyisipan *watermark* berupa *visibility watermark*, sehingga belum dipastikan kehandalan dan ketahanannya. Penelitian lebih lanjut, mekanisme implementasi ini dapat lebih optimalkan.
2. Dalam penelitian ini juga masih menggunakan kunci privat pengguna secara manual untuk mengenkripsi dokumen, sehingga secara *experience* tentu tidak nyaman bagi pengguna. Akan lebih baik apabila kunci privat tersebut disimpan pada tempat penyimpanan data yang aman.

3. Setiap pembuatan kontrak dalam sistem ini, memakan biaya gas yang cukup besar. Hal ini mungkin akan menjadi kendala dimasa mendatang dan tentu saja pengguna akan mempertimbangkan keuntungan yang dapat didapatkan. Oleh karena itu, disarankan untuk merancang ulang atau me-*refactor* kode yang ada dengan mempertimbangkan biaya gas yang akan dikeluarkan setiap pembuatan buku.
4. Pada penelitian lebih lanjut dapat diterapkan sistem royalti. Hal ini sangat mungkin dilakukan dengan cara menarik biaya pada setiap pemindahan hak kepemilikan. Dari royalti tersebut dapat dipertimbangan perhitungan biaya yang masuk ke akun penjual dan *author*.

Daftar Pustaka

Abi Jam'an Kurnia, S. H. (2018). *Aspek Hukum Unduh Lagu dari Internet - Klinik Hukumonline*. Kamis, 29 November 2018. https://www.hukumonline.com/klinik/a/aspek-hukum-unduh-lagu-dari-internet-lt4b202fd2a0be8/

Anshary, M., & Labetubun, H. (n.d.). *Aspek Hukum Hak Cipta Terhadap Buku Elektronik (E-Book) Sebagai Karya Kekayaan Intelektual*. *24*, p-ISSN.

Apa Itu IPFS Dan Kegunaannya Di NFT Project - Diginews.id. (2022). April 1, 2022. https://diginews.id/apa-itu-ipfs-dan-kegunaannya-di-nft-project/

Boulton, I. (2017). *Black Box Testing*. Ptc. https://www.imperva.com/learn/application-security/application-security-testing/%0Ahttps://www.fiixsoftware.com/maintenance-strategies/predictive-maintenance/%0Ahttps://www.imperva.com/learn/application-security/phishing-attack-scam/%0Ahttps://www.ptc.com

Chingath, V., & Babu, R. (2020). *Advantage Blockchain Technology for the Libraries Open access and Resource sharing View project*. https://www.researchgate.net/publication/341725555

Garba, A., Dwivedi, A. D., Kamal, M., Srivastava, G., Tariq, M., Hasan, M. A., & Chen, Z. (2021). A digital rights management system based on a scalable blockchain. *Peer-to-Peer Networking and Applications*, *14*(5), 2665–2680. https://doi.org/10.1007/s12083-020-01023-z

Irawati. (2019). *DIGITAL RIGHT MANAGEMENTS (TEKNOLOGI PENGAMAN) DALAM PERLINDUNGAN TERHADAP HAK CIPTA DI ERA DIGITAL*. https://ejournal2.undip.ac.id/index.php/dplr/article/view/5022

Kementrian Hukum Hak Asasi Manusia Tim. (2020). Modul Kekayaan Intelektual Tingkat Dasar Bidang Hak Cipta (Edisi 2020). *Kementrian Hukum Hak Asasi Manusia Tim*.

Kukkuru, M. G. (n.d.). *Role of Smart Contracts on Blockchain Explained - Insights | Infosys*. Retrieved November 10, 2022, from https://www.infosys.com/insights/digital-future/smart-contracts.html

Kusumawardhani, N. Q. (2021). *Mengenal Teknologi Desentralisasi yang Ada di Blockchain | Republika Online*. 07 Mei 2021. https://www.republika.co.id/berita/qspu3q368/mengenal-teknologi-desentralisasi-yang-ada-di-blockchain

Lauren, A. (2019). *4 Kasus Pelanggaran Hak Cipta Buku yang Pernah Ada di Indonesia - Hukamnas.com*. July 22, 2019. https://hukamnas.com/4-kasus-pelanggaran-hak-cipta-buku-yang-pernah-ada-di-indonesia

Ma, Z., Jiang, M., Gao, H., & Wang, Z. (2018). Blockchain for digital rights management. *Future Generation Computer Systems*, *89*, 746–764. https://doi.org/10.1016/j.future.2018.07.029

MacPherson, L. (2021). *5 Steps to a Proof of Concept for Successful Software Development - Designli Blog*. October 13, 2021. https://designli.co/blog/5-steps-proof-concept-successful-software-development/

Mike, E. (2019). Perlindungan Hukum Hak Kekayaan Intelektual Terhadap Tindakan Pelanggaran Pembajakan Buku Elektronik Melalui Media Online. *Al Imarah : Jurnal Pemerintahan Dan Politik Islam*, *2*(2), 135–144. https://doi.org/10.29300/imr.v2i2.1449

Nareswari Manuaba, I. A. L., & Sukihana, I. A. (2020). PERLINDUNGAN HAK CIPTA PADA BUKU ELEKTRONIK (E-BOOK) DI INDONESIA. *Kertha Semaya : Journal Ilmu Hukum*, *8*(10), 1589. https://doi.org/10.24843/ks.2020.v08.i10.p09

Nizamuddin, N., Hasan, H., Salah, K., & Iqbal, R. (2019). Blockchain-Based Framework for Protecting Author Royalty of Digital Assets. *Arabian Journal for Science and Engineering*, *44*(4), 3849–3866. https://doi.org/10.1007/s13369-018-03715-4

Pangestu Pratama, W. (2021). *Ada 1.184 Kasus Pelanggaran Haki Ditindak di RI Sejak 2015*. 08 Oktober 2021. https://ekonomi.bisnis.com/read/20211006/9/1451327/ada-1184-kasus-pelanggaran-haki-ditindak-di-ri-sejak-2015

Partnerships for Innovation: Accelerating Innovation Research- Technology Translation (PFI: AIR-TT) (nsf14569). (n.d.). Retrieved October 14, 2022, from https://www.nsf.gov/pubs/2014/nsf14569/nsf14569.htm

Penjualan, A., Bajakan, B., & Islamiah, A. P. C. (n.d.). *Latar Belakang Metode Pengumpulan Data*.

Pratt, M. K. (2020). *What is proof of concept (POC)? - Definition from WhatIs.com*. April 2020. https://www.techtarget.com/searchcio/definition/proof-of-concept-POC

Proof of concept Definition & Meaning - Merriam-Webster. (n.d.). Retrieved October 14, 2022, from https://www.merriam-webster.com/dictionary/proof of concept

Puput. (2017). *Dirjen HKI Tangani 60 Kasus Pelanggaran Hak Cipta - StartupHKI*. 11 Juli 2017. https://startuphki.com/dirjen-hki-tangani-60-kasus-pelanggaran-hak-cipta/

Rachmawati, A. R. (2019). *Kerugian Pembajakan Capai Puluhan Triliun Rupiah Per Tahun - Pikiran-Rakyat.com*. 10 September 2019. https://www.pikiran-rakyat.com/ekonomi/pr-01318977/kerugian-pembajakan-capai-puluhan-triliun-rupiah-per-tahun

Risky, N. F., & Bintang, S. (2019). Perlindungan Karya Derivatif Fanfiksi di Internet Berdasarkan Undang-Undang Nomor 28 Tahun 2014 Tentang Hak Cipta. *JIM Bidang Hukum Keperdataan*, *3*(1), 165–174.

Rosenblatt, B. (2018). *Can Blockchain Disrupt The E-Book Market? Two Startups Will Find Out*. Aug 18, 2018,08:27am. https://www.forbes.com/sites/billrosenblatt/2018/08/18/can-blockchains-disrupt-the-e-book-market-two-startups-will-find-out/?sh=25593b65a0b6

Saelan, A., Bandung, I. T., & Bandung, J. G. (2011). *Analisis Beberapa Teknik Watermarking dengan Domain Spasial pada Citra Digital*. *13508029*.

Setiawan, A., Kusumaningtyas, R. F., & Yudistira, I. B. (n.d.). *Diseminasi Hukum Hak Cipta pada Produk Digital di Kota Semarang*. https://doi.org/10.15294/jphi.v1i01.27279

Silfia, I. (2021). *Marak Pembajakan Buku: Penulis dan Penerbit Rugi, Pemerintah Tidak Bisa Melindungi*. Kamis, 01 Juli 2021. https://wartaekonomi.co.id/read348463/marak-pembajakan-buku-penulis-dan-penerbit-rugi-pemerintah-tidak-bisa-melindungi

Simangunsong, H. L., Santoso, B., & Lumbanraja, A. D. (2020). Perlindungan Hak Cipta Terhadap Pembajakan Karya Sastra Novel Versi E-Book Di Tokopedia. *Notarius*, *13*(2), 442–454. https://doi.org/10.14710/nts.v13i2.30504

Simatupang, K. M. (2021). Tinjauan Yuridis Perlindungan Hak Cipta dalam Ranah Digital. *Jurnal Ilmiah Kebijakan Hukum*, *15*(1), 67. https://doi.org/10.30641/kebijakan.2021.v15.67-80

What are smart contracts on blockchain? | IBM. (n.d.). Retrieved October 13, 2022, from https://www.ibm.com/topics/smart-contracts

Wicaksono, A. P., & Urumsah, D. (2017). Perilaku pembajakan produk digital: Cerita dari mahasiswa di Yogyakarta. *Jurnal Aplikasi Bisnis*, *17*(1), 22–42. https://doi.org/10.20885/jabis.vol17.iss1.art2

Zhang, Z., & Zhao, L. (2018). A design of digital rights management mechanism based on blockchain technology. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *10974 LNCS*, 32–46. https://doi.org/10.1007/978-3-319-94478-4_3

Zheng, Z., Xie, S., Dai, H. N., Chen, X., & Wang, H. (2018). Blockchain challenges and opportunities: a survey. *International Journal of Web and Grid Services*, *14*(4), 352. https://doi.org/10.1504/IJWGS.2018.095647