PENERAPAN *ETHEREUM BLOCKCHAIN* UNTUK SISTEM *DATA SHARING* PADA *GAME* SIMULASI PERAKITAN KOMPUTER BERBASIS *VIRTUAL REALITY*

SKRIPSI

Oleh :
MUHAMMAD KHAIRUL AMIN IMRON
NIM. 19650130

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2023

PENERAPAN *ETHEREUM BLOCKCHAIN* UNTUK SISTEM *DATA SHARING* PADA *GAME* SIMULASI PERAKITAN KOMPUTER BERBASIS *VIRTUAL REALITY*

SKRIPSI

Oleh :
MUHAMMAD KHAIRUL AMIN IMRON
NIM. 19650130

Diajukan kepada:
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2023

HALAMAN PERSETUJUAN

PENERAPAN *ETHEREUM BLOCKCHAIN* UNTUK SISTEM *DATA SHARING* PADA *GAME* SIMULASI PERAKITAN KOMPUTER BERBASIS *VIRTUAL REALITY*

SKRIPSI

Oleh :
MUHAMMAD KHAIRUL AMIN IMRON
NIM. 19650130

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 23 September 2023

Pembimbing I,	Pembimbing II,
Dr. Yunifa Miftachul Arif, M.T	Ajib Hanani, M.T
NIP. 19830616 201101 1 004	NIDT. 19840731 20160801 1 076

Mengetahui,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Dr. Fachrul Kurniawan, M.MT, IPM
NIP. 19771020 200912 1 001

HALAMAN PENGESAHAN

PENERAPAN *ETHEREUM BLOCKCHAIN* UNTUK SISTEM *DATA SHARING* PADA *GAME* SIMULASI PERAKITAN KOMPUTER BERBASIS *VIRTUAL REALITY*

SKRIPSI

Oleh :
MUHAMMAD KHAIRUL AMIN IMRON
NIM. 19650130

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 23 November 2023

Susunan Dewan Penguji

Ketua Penguji	: Dr. M. Amin Hariyadi, M.Kom NIP. 19670018 200501 1 001	( )
Anggota Penguji I	: Johan Ericka Wahyu P, M.Kom NIP. 19831213 201903 1 004	( )
Anggota Penguji II	: Dr. Yunifa Miftachul Arif, M.T NIP. 19830616 201101 1 004	( )
Anggota Penguji III	: Ajib Hanani, M.T NIDT. 19840731 20160801 1 076	( )

Mengetahui dan Mengesahkan,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Dr. Fachrul Kurniawan, M.MT, IPM
NIP. 19771020 200912 1 001

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama	: Muhammad Khairul Amin Imron
NIM	: 19650130
Fakultas / Jurusan	: Sains dan Teknologi / Teknik Informatika
Judul Skripsi	: Penerapan *Ethereum Blockchain* Untuk Sistem *Data Sharing* Pada *Game* Simulasi Komputer Berbasis *Virtual Reality*

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini merupakan hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 23 November 2023
Yang membuat pernyataan,

Muhammad Khairul Amin Imron
NIM.19650130

MOTTO

...“Semua orang memiliki eranya masing-masing. Bersabar dan tunggulah, kesempatan akan datang dengan sendirinya”...

(Gol D. Roger)

..."Jika keajaiban itu tidak berpihak pada kita, maka kita sendiri yang akan membuat keajaiban itu”...

(Roronoa Zoro)

HALAMAN PERSEMBAHAN

Saya bersyukur dan berterima kasih kepada Allah SWT karena telah memberikan rahmat dan petunjuk-Nya, sehingga saya berhasil menyelesaikan skripsi ini. Skripsi ini saya persembahkan untuk orang tua saya, yang selalu memberikan dukungan dan mendoakan kelancaran penulisan skripsi saya. Ibu Rini Marini yang selalu memberikan restu, semangat, kasih sayang, perhatian, dan dukungan yang tak terbatas. Bapak Imron yang selalu memberikan arahan dan bimbingan untuk terus melakukan segala sesuatu secara maksimal. Saya juga berterima kasih kepada Kedua orang tua penulis (ibu Rina Marini) yang senantiasa selalu memberikan do'a dan dukungannya untuk menyelesaikan tugas akhir ini dan (ayah Imron) yang senantiasa memotivasi untuk bertanggung jawab dalam menyelesaikan tugas akhir ini, kakak (Nisa), adik (Rachmie), dan seluruh keluarga yang selalu mendukung langkah-langkah saya dalam menyelesaikan skripsi ini. Terima kasih yang tak terhingga bagi semua dukungan yang diberikan.

KATA PENGANTAR

Assalamu'alaikum Wr.Wb.

Puji syukur penulis panjatkan kepada Allah SWT yang senantiasa memberikan rahmat serta kesehatan, sehingga penulis mampu menyelesaikan penelitian ini dengan baik. Penulis menyampaikan ucapan terima kasih kepada semua pihak yang pernah terlibat langsung maupun tidak langsung dalam menyelesaikan penelitian ini, bukan hanya karena usaha keras dari penulis sendiri, akan tetapi karena adanya dukungan dari berbagai pihak. Oleh karena itu penulis berterima kasih kepada:

1. Prof. Dr. M. Zainuddin, M.A., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Prof. Dr. Hj. Sri Harini, M.Si., selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Fachrul Kurniawan M.MT., IPM selaku Ketua Prodi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang
4. Dr. Yunifa Miftachul Arif, M.T selaku Dosen Pembimbing 1 dan Ajib Hanani, M.T selaku Dosen Pembimbing 2 yang telah memberikan bimbingan, arahan serta bantuan dalam terwujudnya karya tulis skripsi ini dari awal hingga akhir.
5. Dr. Ir. M. Amin Hariyadi. M.T selaku penguji I dan Johan Ericka Wahyu Prakasa, M.Kom selaku penguji II yang telah meluangkan waktunya untuk menguji dan memberi arahan dan saran dalam menyelesaikan skripsi ini.
6. Kedua orang tua penulis, Bapak Imron dan Ibu Rini Marini, serta keluarga besar yang telah memberikan banyak dukungan, doa serta selalu menjadi

semangat sehingga penulis mampu menyelesaikan masa studi hingga mencapai gelar sarjana.

7. Kepada seluruh anggota grup HTK (Healing Tai Kucing) dan RAFIKIR Boys yang memberikan bantuan serta menjadi teman nongkrong untuk menyusun skirpsi hingga akhir di penghujung jalan saat ini.
8. Kepada jodoh penulis kelak kamu adalah salah satu alasan penulis menyelesaikan skripsi ini, meskipun saat ini tidak tau keberadaanmu entah di bumi bagian mana dan menggenggam tangan siapa. Tetapi penulis yakin bahwa sesuatu yang ditakdirkan menjadi milik kita akan menuju kepada kita bagaimanapun caranya.
9. *Last but not least. I wanna thank me. I wanna thank me for believing in me, I wanna thank me for doing all this hard work, I wanna thank me for having no days off, I wanna thank me for never quitting, I wanna thank me for always being a giver and tryna give more than I receive, I wanna thank me for tryna do more right than wrong, I wanna thank me for just being me all times.*

Skripsi yang telah ditulis ini masih jauh dari kata sempurna, oleh karena itu penulis sangat menghargai dan senang jika terdapat kritik dan saran yang diberikan. Semoga skripsi ini dapat memberikan manfaat. Wassalamualaikum, Wr. Wb.

Malang, 23 November 2023

Penulis

DAFTAR ISI

HALAMAN PERSETUJUAN ..... **iii**
HALAMAN PENGESAHAN ..... **iv**
PERNYATAAN KEASLIAN TULISAN ..... **v**
MOTTO ..... **vi**
HALAMAN PERSEMBAHAN ..... **vii**
KATA PENGANTAR ..... **viii**
DAFTAR ISI ..... **x**
DAFTAR GAMBAR ..... **xii**
DAFTAR TABEL ..... **xiv**
ABSTRAK ..... **xv**
ABSTRACT ..... **xvi**
مستخلص البحث ..... **xvii**
BAB I PENDAHULUAN **1**
1.1 Latar Belakang 1
1.2 Pernyataan Masalah 3
1.3 Tujuan Penelitian 3
1.4 Batasan Masalah 3
1.5 Manfaat Penelitian 3
BAB II STUDI PUSTAKA **4**
2.1 Penelitian Terkait 4
2.2 *Game* 6
2.3 *Blockchain* 7
2.4 *Ethereum* 8
2.4.1 *Proof of Work* 8
2.4.2 *Smark Contract* 10
2.5 *Data Sharing* 10
2.6 *Virtual Reality* 12
BAB III DESAIN DAN IMPLEMENTASI **14**
3.1 Analisis dan Perancangan *Game* 14
3.1.1 Deskripsi *Game* 14
3.1.2 Perancangan Alur *Game* 15
3.2 Deskripsi Karakter dan Aset 16
3.3 Rancangan *User Interface* 18
3.4 Perancangan *Multiplayer* 21
3.5 Perancangan *Data Sharing* 22
3.6 Penggunaan *Blockchain* dalam *Game* 25
3.7 Uji Coba *Ethereum Blockchain* pada *Unity* 26
3.7.1 Uji Coba *Login* dengan *Wallet Metamask* 26
3.7.2 Menampilkan *Id Wallet* 27
3.7.3 Membuat *Smart Contract* 28
3.7.4 Melakukan Transaksi 28
3.8 Rancangan Pengujian 29
3.8.1 Analisis dan Perancangan *Game* 30

3.8.2 Rancangan Pengujian 30
BAB IV IMPLEMENTASI DAN PEMBAHASAN 31
4.1 Perangkat Lunak yang Digunakan 31
4.2 Data Penelitian 31
4.3 Implementasi Skenario *Game* 32
4.4 Uji Coba 37
4.4.1 Uji Koneksi *Blockchain* dengan *Wallet Metamask* 38
4.4.2 Pengujian *Create Room* 41
4.4.3 Pengujian *Join Room* 44
4.4.4 Pengujian *Claim Score dengan* ERC20 46
4.4.5 Pengujian *Multiplayer* 49
4.4.6 Pengujian *Data Sharing Nickname* 51
4.4.7 Pengujian *Data Sharing Saldo* dan *Score* 54
4.4.8 Pengujian Minting Smart Contract 57
4.4.9 Pengujian *Hash* 61
4.4.10 Pengujian *Gas Limit* 63
4.4.11 Uji Coba 65
4.5 Evaluasi 73
BAB V KESIMPULAN DAN SARAN 76
5.1 Kesimpulan 76
5.2 Saran 77
DAFTAR PUSTAKA 78

DAFTAR GAMBAR

Gambar 3.1 Perancangan Alur *Game* 15
Gambar 3.2 Desain Karakter Pemain 18
Gambar 3.3 *Splash Screen* 18
Gambar 3.4 *Login* 19
Gambar 3.5 *Main Menu* 20
Gambar 3.6 *Gameplay* 20
Gambar 3.7 Skor 21
Gambar 3.8 Alur Sinkronisasi Data *Multiplayer* 22
Gambar 3.9 Alur *Data sharing* 23
Gambar 3.10 Alur Data Pemain 24
Gambar 3.11 *Flowchart Blockchain* pada *Game* 25
Gambar 3.12 *Login ethereum* dengan ekstensi *Metamask* 27
Gambar 3.13 Menampilkan *address wallet* 27
Gambar 3.14 Membuat *smart contract* dengan *IDE Remix* 28
Gambar 3.15 Melakukan transaksi pengiriman *eth* 29
Gambar 3.16 Detail transaksi yang telah berhasil tercatat *blockchain* 29
Gambar 4.1 Koneksi *Wallet Metamask* 32
Gambar 4.2 *Main Menu* 33
Gambar 4.3 *Gameplay* Permainan 34
Gambar 4.4 *Multiplayer Menu* 34
Gambar 4.5 *Create Room* 35
Gambar 4.6 *Find Room* 36
Gambar 4.7 *Scoreboard* 36
Gambar 4.8 *Virtual Reality* 37
Gambar 4.9 Koneksi *Wallet Ethereum* 38
Gambar 4.10 Koneksi Sudah Terhubung 39
Gambar 4.11 Pengujian koneksi dengan 3 perangkat 41
Gambar 4.12 Tampilan *Create Room* 42
Gambar 4.13 *Room* telah dibuat 43
Gambar 4.14 *Create Room* dengan 3 perangkat 44
Gambar 4.15 Tampilan *Find Room* 45
Gambar 4.16 *Join Room* dengan 3 perangkat 46
Gambar 4.17 Tampilan Bermain Simulasi Merakit Komputer 47
Gambar 4.18 Tampilan *Claim Score* 48
Gambar 4.19 Pengujian *Claim Score* dengan 3 Perangkat 49
Gambar 4.20 Tampilan *Room Master* 49
Gambar 4.21 Tampilan *Multiplayer* 50
Gambar 4.22 Pengujian *Multiplayer* dengan 3 perangkat 51
Gambar 4.23 *Data Sharing Nickname* 52
Gambar 4.24 Objek karakter dalam *game* 53
Gambar 4.25 Pengujian *Data sharing Nickname* dengan 3 Perangkat 54
Gambar 4.26 Tampilan Antar Muka (UI) *Player* 55
Gambar 4.27 Informasi *Data sharing* Saldo dan *Score* 55

Gambar 4.28 Pengujian *Data sharing* Saldo dan *Score* dengan 3 Perangkat 57
Gambar 4.29 Pesan *Console Unity* ketika data terkirim 59
Gambar 4.30 Kode *Hash* 59
Gambar 4.31 Perangkat 1 62
Gambar 4.32 Perangkat 2 62
Gambar 4.33 Perangkat 3 63
Gambar 4.34 Pengaturan Gas *Limit* 68
Gambar 4.35 Grafik Kecepatan transaksi menggunakan Data Seluler 69
Gambar 4.36 Kecepatan Jaringan Data Seluler (XL) 69
Gambar 4.37 Grafik Kecepatan transaksi menggunakan Wifi UIN 70
Gambar 4.38 Kecepatan Jaringan Wifi UIN 71
Gambar 4.39 Grafik Kecepatan transaksi menggunakan Wifi Cafe Upojiwo 72
Gambar 4.40 Kecepatan Jaringan Wifi Cafe Upojiwo 72
Gambar 4.41 Estimasi Kecepatan *Metamask* 73

DAFTAR TABEL

Tabel 2.1 Perbandingan Penelitian 6
Tabel 3.1 Komponen *asset* 3D yang digunakan 16
Tabel 3.2 Pengujian Koneksi 30
Tabel 3.3 Pengujian *Data sharing* 30
Tabel 4.1 Perangkat Lunak Yang Digunakan 31
Tabel 4.2 Perangkat 1 31
Tabel 4.3 Perangkat 2 32
Tabel 4.4 Perangkat 3 32
Tabel 4.5 Pengujian koneksi *blockchain* 40
Tabel 4.6 Uji *Create Room* 43
Tabel 4.7 Uji *Join Room* 46
Tabel 4.8 Uji *Claim score* 48
Tabel 4.9 Uji *Multiplayer* 50
Tabel 4.10 Uji *Data sharing Nickname* 54
Tabel 4.11 Uji *Data sharing Nickname*, Sisa Saldo, dan *Score* 56
Tabel 4.12 *Smart contract Drop ERC20* 60
Tabel 4.13 Uji *Minting ERC20 Ethereum* 61
Tabel 4.14 Pengujian minimum gas *limit* 64
Tabel 4.15 Pengujian secara detail minimum gas *limit* 65
Tabel 4.16 Skenario Pengujian kecepatan Transaksi 67
Tabel 4.17 Uji Kecepatan transaksi menggunakan Data Seluler 68
Tabel 4.18 Uji Kecepatan transaksi menggunakan Wifi UIN 70
Tabel 4.19 Uji Kecepatan transaksi menggunakan Wifi Cafe Upojiwo 71
Tabel 4.20 Evaluasi pengujian 73

ABSTRAK

Imron, Muhammad Khairul Amin. 2023. **Penerapan *Ethereum Blockchain* Untuk Sistem *Data Sharing* Pada *Game* Simulasi Perakitan Komputer Berbasis *Virtual Reality*.** Skripsi. Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Dr. Yunifa Miftachul Arif, M.T (II) Ajib Hanani, M.T

Kata Kunci : *Ethereum Blockchain, Data Sharing, Game Simulasi, Perakitan Komputer.*

Sistem *data sharing* merupakan sebuah sistem yang memungkinkan untuk berbagi informasi atau data satu sama lain antara berbagai pihak. Pada penelitian *game* simulasi merakit komputer ini akan terdapat jaringan *blockchain* yang terintegrasi di dalam sistem. Penyimpanan data informasi satu sama lain pada game simulasi perakitan komputer menggunakan jaringan *blockchain* dengan memanfaatkan *smart contract*. Namun terdapat permasalahan lain dalam penerapan teknologi *blockchain*, yaitu kecepatan transaksi yang bervariatif dan sulit dipahami oleh orang awam. Penelitian ini bertujuan guna mengetahui waktu yang diperlukan ketika melakukan transaksi *data sharing score* dan *profil player* dengan pengaruh perbedaan gas *limit* yang digunakan. Pada hasil akhir penelitian didapatkan bahwa pemrosesan transaksi akan jauh lebih cepat dengan membayarkan gas yang lebih banyak seperti menggunakan gas limit dengan range 600.000 - 700.000, sebaliknya pengguna dapat membayarkan lebih sedikit gas seperti menggunakan gas limit dengan range 200.000 - 300.000 dengan konsekuensi besar kemungkinannya untuk menunggu proses validasi transaksi yang lebih lama.

ABSTRACT

Imron, Muhammad Khairul Amin. 2023. **The Implementation of *Ethereum* Blockchain for Data Sharing System in Computer Assembly Simulation Game Using Virtual Reality.** Thesis. Department of Informatics Engineering, Faculty of Science and Technology, State Islamic University of Maulana Malik Ibrahim Malang. Promoter: (I) Dr. Yunifa Miftachul Arif, M.T (II) Ajib Hanani, M.T

There is a system that makes it possible to share information or data between various parties which is called *data sharing*. In this computer building simulation game research there will be a blockchain network integrated into the system. Storing information data with each other in a computer assembly simulation game uses a blockchain network by implementing *smart contract*s. However, there is another issue in this research, namely transaction speeds that vary and are difficult for ordinary people to understand. This research aims to determine the time needed to carry out data transactions for sharing scores and player profiles with the influence of differences in the gas limit used. In the final results of the research, it was found that transaction processing would be much faster by paying more gas, such as using a gas limit in the range of 600,000 - 700,000, on the other hand, users could pay less gas, such as using a gas limit in the range of 200,000 - 300,000, with the consequence of being more likely to have to wait. longer transaction validation process.

Keywords *: Ethereum Blockchain, Data sharing, Simulation Games, Computer assembly.*

مستخلص البحث

عمران، مُحَمَّد خير الأمين. ٢٠٢٣. تطبيق منصة سلسلة الكتل (*Ethereum Blockchain*) لنظام مشاركة البيانات في ألعاب محاكاة تركيب الكمبيوتر على أساس الواقع الافتراضي. البحث الجامعي. قسم الهندسة المعلوماتية، كلية العلوم والتكنولوجيا بجامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف الأول: د. يونيفة مفتاح العارف، الماجستيرة. المشرف الثاني: عجيب حناني، الماجستير.

الكلمات الرئيسية: منصة سلسلة الكتل، مشاركة البيانات، ألعاب محاكاة، تركيب الكمبيوتر.

نظام مشاركة البيانات هو نظام يسمح بمشاركة المعلومات أو البيانات مع بعضها البعض بين مختلف الأطراف. في بحث ألعاب محاكاة تركيب الكمبيوتر هذا، ستكون هناك شبكة سلسلة الكتل المدمجة في النظام. تخزين بيانات المعلومات مع بعضها البعض في ألعاب محاكاة تركيب الكمبيوتر باستخدام شبكة سلسة الكتل من خلال استخدام العقود الذكية. ولكن هناك مشكلة أخرى في تطبيق تقنية سلسلة الكتل، وهي سرعة المعاملات التي تختلف ويصعب على الناس العاديين فهمها. يهدف هذا البحث إلى تحديد الوقت اللازم عند إجراء معاملة مشاركة البيانات في الدرجات وملفات تعريف اللاعبين مع تأثير الاختلافات في حدود الغاز المستخدمة. في النتائج النهائية للبحث، وجد أن معالجة المعاملات ستكون أسرع بكثير من خلال دفع المزيد من الغاز مثل استخدام حد الغاز بنطاق يتراوح بين ٦٠٠,٠٠٠ – ٧٠٠,٠٠٠، على العكس من ذلك، يمكن للمستخدمين دفع غاز أقل مثل استخدام حد الغاز بنطاق يتراوح بين ٢٠٠,٠٠٠ – ٣٠٠,٠٠٠ مع نتيجة من المحتمل أن ينتظروا عملية التحقق من صحة المعاملات لفترة أطول.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam kemajuan era digital seperti pada masa sekarang, *data sharing* atau berbagi data menjadi sangat penting dan diperlukan dalam berbagai sektor tidak terkecuali dalam *game*. Sistem *data sharing* dalam *game* akan sangat dibutuhkan karena mampu menyediakan informasi yang penting terhadap pemain, seperti informasi mengenai poin atau capaian level. Adanya sistem *data sharing* poin dapat memberikan dampak positif bagi tiap pemain, karena semangat berkompetisi akan terbentuk sehingga para pemain akan saling berlomba untuk memperoleh poin yang tinggi guna menyaingi pemain lainnya (Utomo et al., 2019).

Dalam proses *data sharing* yang konvensional, biasanya terdapat perantara atau pihak ketiga yang memfasilitasi proses tersebut. Namun, dengan adanya perantara terkadang transparansi dan akuntabilitas menjadi kurang terjaga. Pada penelitian yang dikerjakan di Universitas Cornell oleh beberapa peneliti, didapati bahwa keberadaan perantara dapat meningkatkan risiko keamanan data. Oleh Maka dari itu, dibutuhkan teknologi yang dapat meningkatkan keamanan dan transparansi dalam proses *data sharing*. Salah satu teknologi yang mampu dimanfaatkan guna meningkatkan keamanan dan transparansi dalam proses *data sharing* adalah *blockchain*.

Proses *data sharing* dengan menggunakan *blockchain* menjadi salah satu rekomendasi pilihan utama sebab teknologi ini menyediakan mekanisme

keamanan yang sangat baik. Setiap transaksi yang dilakukan di dalam *blockchain* direkam dan diverifikasi secara otomatis oleh jaringan. Dalam *blockchain*, setiap transaksi dilindungi oleh teknologi enkripsi yang sangat kuat sehingga data akan sangat sulit dimanipulasi dan dicuri oleh pihak yang tidak berwenang. Dalam proses *data sharing* terdapat hal fundamental yang begitu vital yaitu keamanan data, khususnya data pribadi. Dalam sebuah laporan yang diterbitkan oleh Verizon pada tahun 2021, ditemukan bahwa 61% dari semua pelanggaran keamanan data yang terjadi di seluruh dunia melibatkan data pribadi. Oleh karena itu, keamanan data menjadi prioritas utama dalam proses *data sharing*.

Proses *data sharing* dengan menggunakan *blockchain* juga dapat meningkatkan efisiensi dan kecepatan, sebab setiap transaksi dikirim secara langsung ke jaringan tanpa harus melalui perantara sehingga akan didapati proses dengan tingkat efisiensi dan kecepatan yang tinggi. Selain itu, *data sharing* dengan menggunakan *blockchain* juga dapat mengurangi biaya yang diperlukan. Dalam sebuah laporan yang diterbitkan oleh McKinsey pada tahun 2018, ditemukan bahwa penggunaan *blockchain* dalam proses *supply chain* dapat meningkatkan efisiensi dan mengurangi biaya hingga 25% (Carson et al., 2018). Hal ini *menu*njukkan bahwa penggunaan *blockchain* dalam proses *data sharing* dapat memberikan manfaat yang signifikan dalam hal efisiensi dan biaya.

Berdasarkan penelitan tersebut, dapat diketahui bahwa *data sharing* dengan menggunakan *blockchain* memberikan banyak manfaat, seperti keamanan data, transparansi, akuntabilitas, efisiensi dan kecepatan. Maka dari itu, dalam penelitian ini akan dilakukan pembuktian mengenai seberapa cepatkah transaksi

dengan menggunakan *blockchain* terhadap kecepatan transaksi *data sharing* dengan pengaruh perbedaan gas *limit* pada *Game* Simulasi Perakitan Komputer.

1.2 Pernyataan Masalah

Bagaimana pengaruh perbedaan gas *limit* terhadap waktu yang dibutuhkan untuk melakukan transaksi *data sharing* poin dan profil pemain pada *Game* Simulasi Perakitan Komputer menggunakan *Ethereum Blockchain*?

1.3 Tujuan Penelitian

Mengukur tingkat akurasi yang mampu dihasilkan oleh metode *naïve* Bayes dalam memberikan mengklasifikasikan gaya belajar berdasarkan perilaku dan kebiasaan.

1.4 Batasan Masalah

Adapun pada penelitian ini diberikan batasan masalah guna menjaga konteks dari fokus masalah penelitian yang diusung , yaitu sebagai berikut :

1. Jenis jaringan *Blockchain* terbatas pada jaringan *Ethereum*
2. Data yang akan ditampilkan adalah nilai poin dan profil data pemain

1.5 Manfaat Penelitian

Adapun terdapat manfaat yang didapatkan dari penelitian ini adalah mengetahui pengaruh perbedaan gas *limit* terhadap waktu yang dibutuhkan dalam mekanisme penerapan *Ethereum Blockchain* pada saat *data sharing* dilakukan dalam *game* Simulasi Perakitan Komputer.

BAB II

STUDI PUSTAKA

2.1 Penelitian Terkait

Dalam sebuah penelitian pada tahun 2020 oleh Nopendri telah dipaparkan tentang bagaimana simulasi penerapan *blockchain* pada bidang politik, tepatnya dalam proses *e-voting* untuk pemilihan presiden Indonesia. Adapun hasil dari penelitian tersebut mengungkapkan bahwa sangat memungkinkan untuk diterapkan teknologi *blockchain* pada pemilihan presiden. Kemudian berdasarkan simulasi dan ilustrasi dari proses transakasi data menggunakan *blockchain* yang telah didapatkan, transaksi data yang dilakukan aman sehingga mengurangi adanya resiko manipulasi data (Nopendri et al., 2020).

Selanjutnya dalam penelitian yang dilakukan oleh Untung Rahardja, dkk pada tahun 2020 melakukan penelitian dengan menerapkan teknologi *blockchain* dalam proses transaksi *e-commerce* untuk media pengamanan. Adapun terdapat dua keuntungan yang dijelaskan dalam penelitian ini terkait dengan penerapan teknologi *blockchain* dalam proses transaksi *e-commerce*, diantaranya yaitu proses transaksi melalui penerapan *blockchain* dapat terjamin sistem keamanannya, kemudian yaitu adanya kemudahan juga efisiensi dalam sistem pembayaran yang dilakukan dan hal tersebut terdokumentasikan dengan baik (Rahardja et al., 2020).

Masih pada konteks penerapan *blockchain* yaitu penelitian yang dilakukan oleh Reza Putra Pradana pada tahun 2020, mengimplementasikan teknologi *blockchain* pada *game multiplayer* wisata bromo dalam sistem transaksi antar user

atau pengguna. Adapun hasil dalam penelitian ini dengan melakukan optimasi pada variabel dalam *platform ethereum* memperlihatkan bahwa tarif transaksi dapat lebih murah dan memiliki hasil yang lebih cepat jika diperbandingkan dengan platform yang lain seperti Bitcoin (Pradana, 2020).

Kemudian pada penelitian yang dilakukan oleh Firdaus tahun 2020, penelitian dilakukan dengan menerapkan teknologi *Ethereum Blockchain* pada *game* yang sedang dikembangkannya yaitu *game* wisata widow waterfall. Dalam penelitian tersebut berfokus pada pengintegrasian jaringan *Ethereum Blockchain* terhadap *scoring game*. Pada *game* tersebut data akan tersimpan dalam jaringan *Ethereum Blockchain* dengan memanfaatkan *smart contract*. Adapun hasil dalam penelitian ini yaitu transaksi antar pengguna atau user berhasil dijalankan, dan telah terverifikasi dalam jaringan *Ethereum Blockchain* (Firdaus, 2020).

Kemudian pada penelitian yang dilakukan oleh Akbar tahun 2020, penelitian dilakukan dengan menerapkan teknologi *Ethereum Blockchain* pada *game* yang sedang dikembangkannya yaitu *game* tempat wisata di malang. Pada penelitian ini peneliti berfokus pada rating dan preference user yang akan diraih oleh pengguna terkait dengan rata-rata rating tertinggi dan kesesuaian rating dari keseluruhan tempat wisata. Pada penelitian tersebut data rating akan disimpan dengan menggunakan sistem yang terintegrasi dengan jaringan *blockchain* menggunakan *Ethereum Blockchain*. Hasil pada penelitian ini adalah *data sharing* list rekomendasi berhasil dilakukan, data list rekomendasi dapat dilihat oleh keseluruhan pemain secara sama. Kemudian berdasarkan hasil input *data sharing*

score setiap pemain didapatkan pula hasil rekomendasi destinasi wisata dari sistem (Darmawan, 2022).

Berikut ini pada tabel 2.1 merupakan ringkasan detail secara singkat mengenai penelitian yang terkait.

Tabel 0.1 Perbandingan Penelitian

No	Peneliti	Judul	Persamaaan	Perbedaan
1	Nopendri, Aurachman and Amrullah, 2020	Rancangan Simulasi Penerapan *Blockchain* dalam Pemilihan Presiden Indonesia	*Blockchain*	*E – Votting*
2	Rahardja et al., 2020	Penerapan Teknologi *Blockchain* Sebagai Media Pengamanan Proses Transaksi E-Commerce	*Blockchain*	*E - Commerce*, Sistem Keamanan
3	Pradana, 2020	Sistem Transaksi Antar *Player* Pada *Game Multiplayer* Wisata Bromo Menggunakan *Blockchain*.	*Blockchain*, *Game* 3D	Sistem Transaksi, Tanpa VR
4	Firdaus, 2020	Sistem *Data sharing* Untuk *Score* pada *Game* Widow Waterfall Menggunakan *Ethereum Blockchain*.	*Blockchain*, *Game 3D*, *Data sharing*	*Score*, Tanpa VR
5	Darmawan, 2022	Sistem *Data sharing* Untuk Rating Dan Preference User Menggunakan *Ethereum Blockchain* Pada *Game* Tempat Wisata Di Malang	*Blockchain*, *Game*, *Data sharing*	Rating Dan Preference User, Tanpa VR

2.2 *Game*

Penjelasan tentang teknologi blockchain ialah suatu metode untuk penyimpanan data, yang mana terdiri atas kumpulan node – node yang saling terkoneksi pada jaringan blockchain dan sifatnya terdesentralisasi. Kemudian

akan terdapat kumpulan record atau catatan transaksi digital data yang berbentuk blok dan saling berkaitan sehingga akan terlihat seperti bentuk rantai blok di dalamnya. Blockchain berperan sebagai sistem basis data yang terdistribusi atau dapat dianalogikan seperti suatu kolam berukuran besar tak terbatas yang berisikan kumpulan record atau catatan dari seluruh transaksi data yang terjadi dalam jaringan blockchain. Ketika adanya kegiatan transaksi maka akan ditandai menurut waktu prosesnya terjadi kemudian diubah ke dalam blok yang mana dengan hash kriptografiknya akan mengidentifikasi masing – masing dari blok tersebut (Bahga & Madisetti, 2016).

2.3 *Blockchain*

Penjelasan tentang teknologi *blockchain* ialah suatu metode untuk penyimpanan data, yang mana terdiri atas kumpulan node – node yang saling terkoneksi pada jaringan *blockchain* dan sifatnya terdesentralisasi. Kemudian akan terdapat kumpulan record atau catatan transaksi digital data yang berbentuk blok dan saling berkaitan sehingga akan terlihat seperti bentuk rantai blok di dalamnya. *Blockchain* berperan sebagai sistem basis data yang terdistribusi atau dapat dianalogikan seperti suatu kolam berukuran besar tak terbatas yang berisikan kumpulan *record* atau catatan dari seluruh transaksi data yang terjadi dalam jaringan *blockchain*. Ketika adanya kegiatan transaksi maka akan ditandai *menu*rut waktu prosesnya terjadi kemudian diubah ke dalam blok yang mana dengan *hash* kriptografiknya akan mengidentifikasi masing – masing dari blok tersebut (Bahga & Madisetti, 2016).

2.4 *Ethereum*

Ethereum dibangun oleh Vitalik Buterin pada tahun 2013 yang mana merupakan sebuah project berbasiskan *blockchain*. *Ethereum* disimbolkan dengan token uang kripto *ETH* (*Eth*er) dan dengan itu dapat menjalankan *smart contract* secara *peer-to-peer*. Terdapat banyak jenis jaringan *blockchain* yang mampu untuk diprogram dan dikembangkan kembali, salah satunya yaitu *ethereum*. *Ethereum Blockchain* adalah program yang bersifat *open source* sehingga semua orang dapat memanfaatkan *Ethereum Blockchain* untuk digunakan dalam membagun proyeknya, dengan begitu memungkinkan semua pengguna memiliki kebebasan untuk membuat *smart contract* serta aplikasinya sendiri secara terdesentralisasi. Pengguna memiliki kebebasan untuk menetapkan aturan sesuai preferensi mereka terkait kepemilikan token dan format transaksi.. (Buterin, 2014).

Dalam *Ethereum*, status memuat kepemilikan yang diidentifikasi sebagai "akun." Setiap akun akan terdapat alamat berupa sejumlah 20 byte, dan status tersebut memfasilitasi transfer nilai dan pertukaran informasi antara berbagai akun. (Jani, 2017).

2.4.1 *Proof of Work*

Proof of work (PoW) merupakan mekanisme konsensus yang berada dalam jaringan *blockchain* guna mengkonfirmasi transaksi dan menambang blok baru. PoW telah digunakan dalam jaringan *blockchain* Bitcoin sejak diluncurkan pada tahun 2009, dan banyak jaringan *blockchain* lainnya juga mengadopsi mekanisme konsensus ini.

Dalam jaringan *blockchain*, PoW memerlukan pemecahan masalah matematika yang kompleks dan memakan waktu untuk menyelesaikannya, sehingga membutuhkan daya komputasi yang besar dan banyak energi untuk menyelesaikannya. Tujuan dari PoW adalah untuk menghasilkan solusi unik untuk setiap blok baru yang harus diverifikasi sebelum ditambahkan ke *blockchain*. Solusi tersebut harus me*menu*hi kriteria tertentu, yang dikenal sebagai target, sehingga memastikan bahwa blok baru yang ditambahkan ke *blockchain* benar-benar sah (Arif, Pradana, et al., 2020).

Setiap node dalam jaringan *blockchain* dapat menambang blok baru, yang artinya setiap node harus menyelesaikan perhitungan matematika yang sama untuk memecahkan masalah. Proses ini membutuhkan daya komputasi yang besar dan memakan waktu untuk menyelesaikannya. Oleh karena itu, penambang biasanya menggunakan perangkat keras khusus, seperti ASIC (*Application-Specific Integrated Circuit*) dan GPU (*Graphics Processing Unit*) untuk meningkatkan kecepatan perhitungan dan mendapatkan peluang lebih besar untuk menambang blok baru.

Setiap blok baru yang ditambahkan ke *blockchain* memiliki *hash* unik, yang berfungsi sebagai sidik jari digital dari blok tersebut. *Hash* ini juga digunakan untuk membuat rantai blok, di mana *hash* blok sebelumnya menjadi input untuk menghasilkan *hash* blok berikutnya. Dengan cara ini, setiap blok dalam rantai blok bergantung pada blok sebelumnya, sehingga menciptakan keamanan dan keandalan jaringan *Blockchain*.

2.4.2 *Smark Contract*

Smart contract adalah program komputer yang dijalankan di dalam *Blockchain* untuk mengeksekusi transaksi secara otomatis berdasarkan kondisi yang telah ditentukan sebelumnya. *Smart contract* pada dasarnya adalah kontrak yang ditulis dalam kode komputer, dan dapat digunakan untuk mengotomatisasi berbagai jenis transaksi, seperti pembayaran, pemindahan aset, dan pengiriman pesanan (Arif et al., 2021). *Smart contract* pertama kali diperkenalkan oleh Nick Szabo pada tahun 1994, dan konsepnya kemudian diimplementasikan dalam *Blockchain Ethereum* pada tahun 2015 (Rochmatullah, 2022).

Smart contract pada *Ethereum* ditulis dalam bahasa pemrograman Solidity, dan dijalankan di dalam lingkungan virtual machine yang disebut *Ethereum* Virtual Machine (EVM). *Smart contract* pada *blockchain* dapat digunakan untuk berbagai jenis aplikasi, seperti perbankan, asuransi, dan logistik. Salah satu contoh penggunaan *smart contract* adalah dalam industri asuransi, di mana *smart contract* dapat digunakan untuk mengeksekusi klaim secara otomatis berdasarkan kondisi tertentu, seperti cuaca yang buruk atau kejadian bencana alam (Antonopoulos, 2014).

2.5 *Data Sharing*

Data sharing adalah salah satu aplikasi utama dari teknologi *blockchain*. *blockchain* memungkinkan pengguna untuk menyimpan data secara terdesentralisasi, transparan, aman, dan tanpa resiko manipulasi. Berikut ini, akan dibahas tentang konsep dasar *data sharing* dalam *blockchain*, manfaatnya, teknologi yang digunakan, dan beberapa contoh penggunaannya.

Data sharing dalam *blockchain* dapat diartikan sebagai proses berbagi data di antara beberapa partai atau pengguna melalui jaringan *blockchain* (Arif, Nurhayati, et al., 2020). Data yang dibagikan ini dapat berupa informasi keuangan, informasi medis, atau informasi lainnya yang memerlukan keamanan dan privasi. Untuk memulai proses *data sharing*, pengguna pertama-tama harus membuat dan memasukkan data ke dalam *blockchain*. Data ini kemudian disimpan dalam blok transaksi dan diverifikasi oleh jaringan. Setelah diverifikasi, data dapat diakses oleh pengguna yang telah diberikan hak akses oleh pengguna asli. Dalam hal ini, hak akses diberikan melalui penggunaan kunci pribadi. Proses *data sharing* dalam *blockchain* menghilangkan kebutuhan untuk menggunakan pihak ketiga atau mediator dalam pengaturan dan validasi data. Proses ini juga membuat data lebih terdesentralisasi, dan memungkinkan lebih banyak orang untuk mengaksesnya.

Data sharing dalam *blockchain* menawarkan beberapa manfaat. Beberapa di antaranya adalah:

1. Keamanan yang lebih baik. Data yang disimpan dalam *blockchain* dienkripsi dengan kunci pribadi dan publik yang memastikan keamanan dan privasi. Selain itu, *Blockchain* juga menggunakan algoritma konsensus yang dapat memastikan bahwa data tidak dapat dimanipulasi orang yang tidak punya wewenang dan tidak bertanggung jawab.
2. Kecepatan transaksi yang lebih baik. *blockchain* memungkinkan transfer data secara instan dan *real-time*. Ini memungkinkan *data sharing* yang lebih cepat, efisien dan efektif.

3. Biaya yang lebih rendah. *Data sharing* dalam *blockchain* dapat menghilangkan biaya tambahan yang biasanya dikeluarkan dalam pengaturan dan verifikasi data melalui pihak ketiga. Selain itu, *blockchain* juga dapat mengurangi biaya administrasi dan waktu yang diperlukan dalam proses transaksi.
4. Privasi yang lebih baik. *Data sharing* dalam *blockchain* menjamin bahwa hanya pengguna yang memiliki hak akses atas data kepemilikannya. Hal ini menjaga privasi pengguna dan mengurangi risiko data yang tidak diinginkan.

2.6 *Virtual Reality*

Virtual reality (VR) mulai diperkenalkan kepada publik pada awal tahun 1990 dalam sebuah pameran dan menurut beberapa pengamat industri mengungkapkan bahwa prospeknya akan cukup menjanjikan sehingga kemudian mulai banyak eksperimen-eksperimen yang dilakukan sendiri mengenai hal ini (Firu et al., 2020).

Virtual reality (VR) merupakan perangkat teknologi yang dapat dipakai penggunanya untuk bisa mengalami dunia virtual yang menyerupai dunia nyata dengan menggunakan perangkat seperti headset VR. Teknologi ini telah mengalami kemajuan pesat pada beberapa tahun belakangan dan dianggap sebagai salah satu tren teknologi terbesar di dunia. Fitur-fitur yang terdapat pada VR semakin baik dan inovatif mulai dari teknologi layar, sensor gerak dengan diikuti harganya yang semakin terjangkau juga. Terdapat beberapa produk VR yang terbilang masih cukup terjangkau seperti VR Box, Samsung Gear VR, *Google* Cardboard, dan masih banyak lagi. Namun ada kekurangan tersendiri dalam

produk-produk VR yang biasanya murah seperti memiliki fitur yang terbatas, tidak tahan lama, dan produk tergolong berat (Segura et al., 2020)

VR memungkinkan pengguna untuk memasuki dunia virtual yang memberikan pengalaman yang sangat realistis. Dalam dunia virtual, pengguna dapat merasakan suara, cahaya, dan gerakan seperti dalam dunia nyata, sehingga memberikan pengalaman yang lebih mendalam dan mengesankan. VR telah digunakan dalam berbagai bidang, termasuk hiburan, pendidikan, kesehatan, bisnis, dan lainnya (Zheng, 2019).

Salah satu aplikasi terbesar dari VR adalah dalam bidang hiburan, khususnya dalam *game*. Dalam *game* VR, pengguna dapat memasuki dunia yang menyerupai dunia nyata dan mengalami pengalaman yang benar-benar baru. *Game* VR dapat menjadi lebih menarik dan interaktif karena pengguna dapat memerintah karakter di dalam *game* dan mengikuti jalan cerita yang lebih kompleks dan mendalam (Faisal et al., 2016). VR juga dapat digunakan dalam film dan televisi untuk memberikan pengalaman yang lebih mendalam dan imersif. Di bidang pendidikan, VR dapat digunakan sebagai alat pembelajaran. Siswa dapat memasuki dunia virtual dan belajar tentang berbagai topik dengan cara yang lebih interaktif dan mendalam (Nurhayati & Arif, 2023). VR dapat digunakan dalam pembelajaran sains, sejarah, bahasa, dan lainnya. Dalam bidang kesehatan, VR juga digunakan untuk memberikan pengalaman yang lebih baik bagi pasien selama perawatan.

BAB III

DESAIN DAN IMPLEMENTASI

3.1 Analisis dan Perancangan *Game*

Untuk dapat mengetahui sebuah permasalahan yang dimiliki maka perlu dilakukan adanya analisa dan perancangan terhadap suatu *game* atau aplikasi, dengan begitu kita dapat menentukan kebutuhan yang diperlukan untuk menyelesaikan permasalahan juga memberikan solusi guna mempermudah pengerjaan proyek. Pada penelitian ini akan dibangun sebuah sistem *data sharing* pada *game* simulasi perakitan komputer berbasis *virtual reality* dengan menerapkan teknologi *ethereum blockchain*.

3.1.1 Deskripsi *Game*

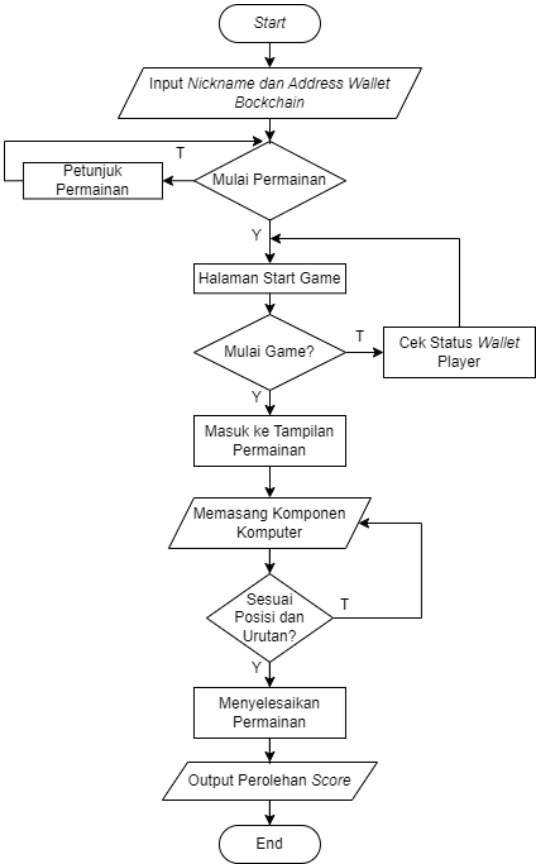
Game perakitan komputer atau PC (*Personal Computer*) biasanya dikategorikan sebagai *game* simulasi atau *game* manajemen. Dalam *penelitian* ini, pemain akan diminta untuk memilih dan memasang komponen - komponen yang perlukan untuk merakit sebuah komputer. Selain itu, pemain juga harus memperhatikan spesifikasi dan kompatibilitas komponen agar komputer yang dibuat dapat berfungsi dengan baik. *Game* ini sering kali mengandalkan aspek-aspek teknis yang detail dan akurat, sehingga dapat membantu pemain memahami cara kerja komputer dan proses perakitan.

Game ini akan dibangun berbasiskan *Virtual Reality* (VR) dengan begitu dapat memberikan pengalaman yang sangat mendalam dan realistis. Dalam *game* ini, pemain dapat merasakan sensasi langsung dari proses perakitan PC, seperti

memasang komponen, menghubungkan kabel, dan menyesuaikan konfigurasi. Penggunaan teknologi VR juga dapat memberikan nuansa ruang dan kedalaman yang lebih dalam, sehingga pemain dapat merasakan bahwa mereka benar-benar berada di dalam ruang perakitan PC yang sebenarnya. Selain itu, *game* merakit PC berbasis VR juga dapat membantu pemain memahami lebih baik tentang cara kerja PC dan mengembangkan keterampilan teknis yang dibutuhkan untuk merakit PC secara mandiri.

3.1.2 Perancangan Alur *Game*

Adapun *game* ini mempunyai rangkaian alur yang diilustrasikan melalui *flowchart*, seperti yang ditunjukkan dalam Gambar 3.1.



Gambar 0.1 Perancangan Alur *Game*

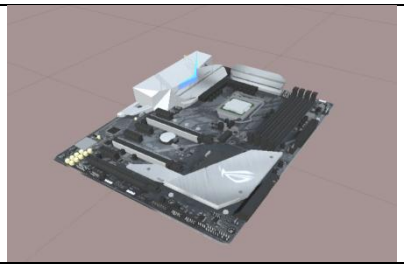
Flowchart dalam Gambar 3.1 menjelaskan secara keseluruhan langkah-langkah yang akan diambil dalam permainan ini. Saat permainan dimulai, *splash screen* akan ditampilkan dengan gambar latar belakang yang bertemakan rangkaian komponen komputer. Setelah tampilan *splash screen* muncul, pemain diharuskan *login* terlebih dahulu untuk bisa masuk ke halaman utama. Setelah berhasil *login*, pemain akan dialihkan menuju menu utama yang menyediakan opsi pilihan navigasi menuju menu permainan, menu petunjuk, dan keluar permainan. Menu petunjuk akan menampilkan informasi terkait panduan bermain, sementara menu *start game* berfungsi untuk memulai permainan atau melihat status dompet *ethereum* pemain. Pada *menu* status dompet pemain, informasi yang disediakan yaitu seperti alamat dan jumlah poin yang telah disimpan di *blockchain* akan ditampilkan.

3.2 Deskripsi Karakter dan Aset

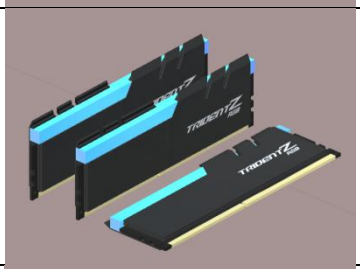
1. Aset

Untuk membuat *game* simulasi perakitan komputer, terdapat beberapa *asset* 3D yang diperlukan seperti yang tercantum dalam Tabel 3.1.

Tabel 0.1 Komponen *asset* 3D yang digunakan

Nama Komponen	Gambar
Motherboard	

Lanjutan Tabel 3.1

CPU			
RAM			
Memory			
Power Supply			

2. Karakter

Pada *game* ini akan terdapat tokoh yang dapat diperankan oleh pemain, yang mana tokoh tersebut merupakan representasi dari pemain itu sendiri. *Game* ini dikembangkan dengan berbasis *virtual reality*, sehingga dengan begitu visualisasi dari *player* akan dibuat dua jenis. Pertama yaitu ketika pemain belum memasuki *scene* permainan maka karakter *player* akan ditampilkan secara menyeluruh, namun jika sudah maka *visual* yang ditampilkan hanya tangannya saja.

Gambar 0.2 Desain Karakter Pemain

3.3 Rancangan *User Interface*

1. *Splash Screen*

Berikut ini pada Gambar 3.3 adalah rancangan tampilan ketika aplikasi atau *game* ini mulai dijalankan. Pada rancangan tampilannya akan terdapat nama dari yang akan dimainkan yaitu simulasi merakit komputer dan dibawahnya adalah keterangan bahwa *game* tersebut sedang melakukan proses *loading*.

Gambar 0.3 *Splash Screen*

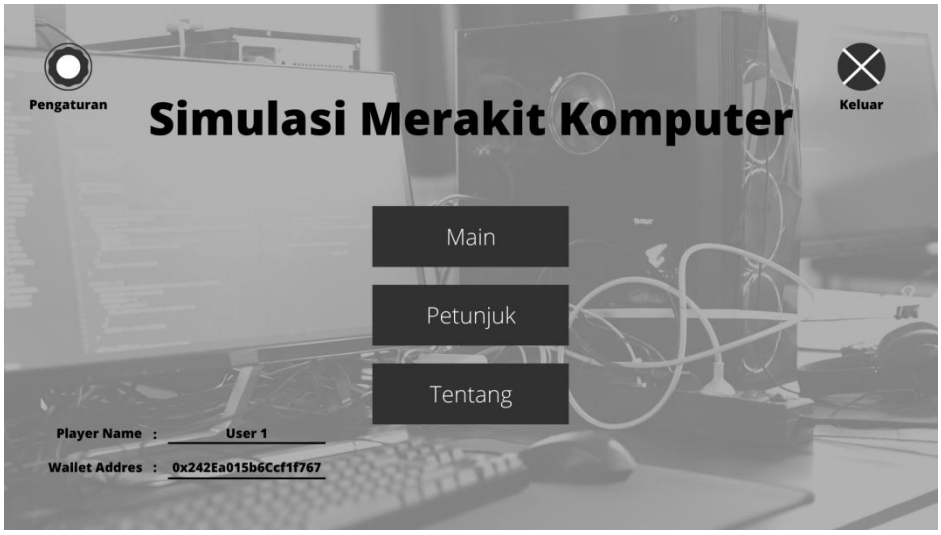
2. *Login*

Saat *login* ke dalam *game*, pemain diwajibkan untuk mengisi alamat akun yang diperoleh dari dompet digital yang telah dibuat seperti pada Gambar 3.4. Cara membuatnya dapat dilakukan dengan menggunakan *Metamask*, dan instruksinya dapat ditemukan di situs web *Metamask* (https://*Metamask*.io/).

Gambar 0.4 *Login*

3. *Main Menu*

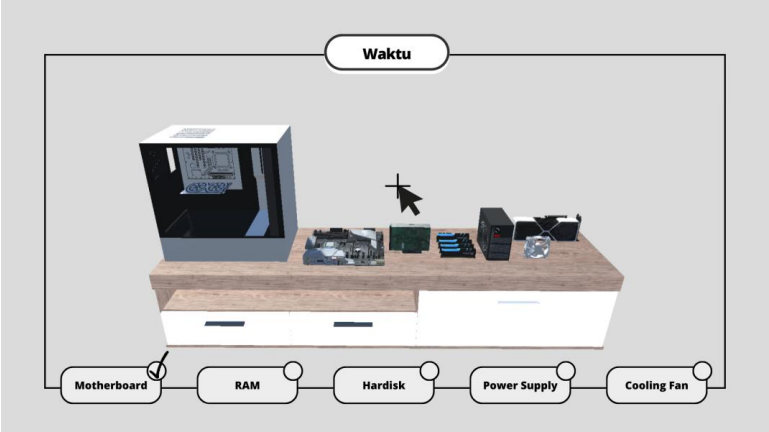
Selanjutnya adalah rancangan tampilan dari *scene Main Menu* pada *game* ini yang terlihat pada Gambar 3.5 pada tampilannya akan terdapat judul dari *game* yang terdapat di posisi tengah atas halaman. Kemudian terdapat tiga tombol pilihan yaitu "Main*"* untuk mulai permainan, opsi "Petunjuk*"* yang berisikan informasi tentang bagaimana cara dalam memainkan *game*, dan yang terakhir yaitu "Tentang*"* yang berisikan informasi *credits* dari pemain. Pada posisi kiri atas merupakan *button* pengaturan dari *game* dan pada posisi kanan atas merupakan *button* untuk dapat keluar atau menutup *game*.



Gambar 0.5 *Main Menu*

4. Gameplay

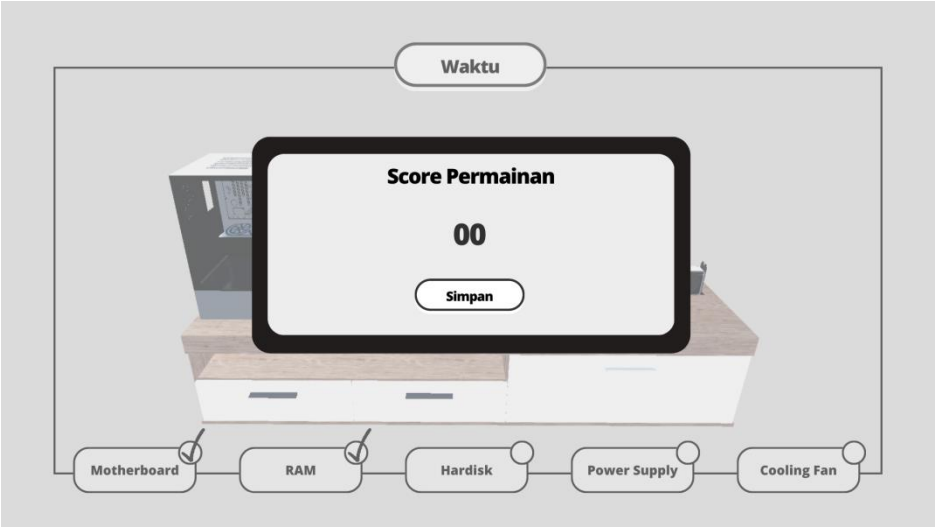
Ketika pengguna sudah memasuki *scene* permainan maka akan ditampilkan seperti pada Gambar 3.6 pada rancangan tampilan *gameplay*, *user* akan dihadapkan pada sekumpulan komponen - komponen dari sebuah komputer dan juga waktu yang tersedia untuk memainkannya. Kemudian pemain diharuskan mencocokan atau memasangkan komponen - komponen tersebut sesuai dengan posisi dan tempat yang sesuai. Apabila komponen yang dicocokan benar maka keterangan nama dari komponen tersebut akan diberikan tanda ceklis.



Gambar 0.6 *Gameplay*

5. Skor

Gambar 3.7 adalah rancangan tampilan ketika waktu permainan telah selesai, maka *game* akan otomatis melakukan *pause* atau berhenti. Kemudian akan ditampilkan sebuah panel papan *score* dari hasil permainan sesuai dengan waktu penyelesain yang didapat.



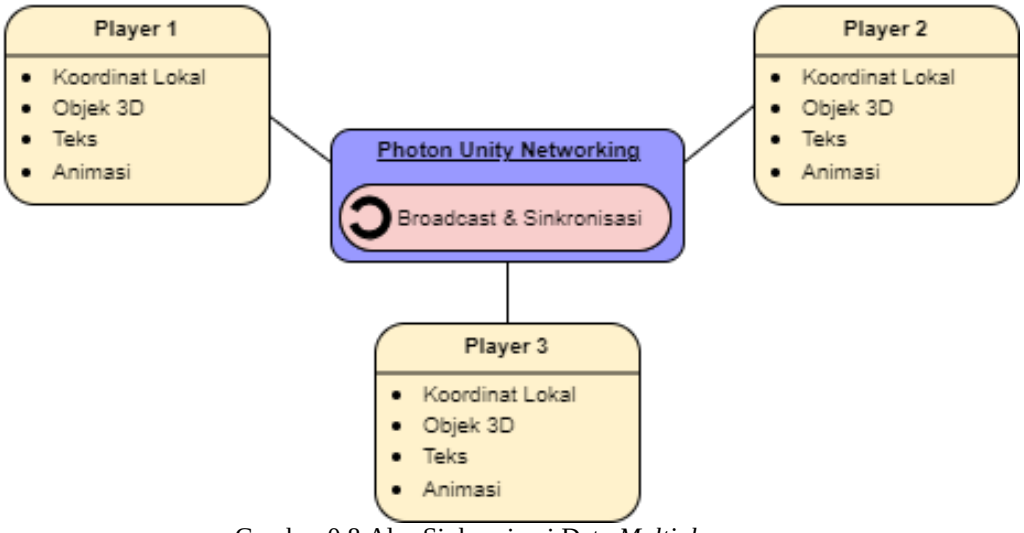
Gambar 0.7 Skor

3.4 Perancangan *Multiplayer*

Pada game *multiplayer*, *server* dan klien bertukar data dengan mengirimkan paket melalui jaringan. Untuk menciptakan ruang virtual bersama bagi pemain yang terhubung dari lokasi berbeda, peristiwa dalam proses permainan (seperti pergerakan karakter atau pembuatan objek) disinkronkan dengan klien lain dengan mengirimkan paket data kepada mereka. Pihak yang bertanggung jawab untuk mengirim dan menerima paket melalui jaringan disebut lapisan transport.

Dalam pengembangan game multiplayer setiap elemen pada game perlu tersinkronisasi, antara lain sistem koordinat lokal, objek 3D , teks, animasi. Untuk

menyinkronkan muatan elemen tersebut, peneliti menggunakan pengaturan dasar *Photon Unity Networking*. Adapun kerangka sinkronisasi elemen tersebut divisualisasikan pada Gambar 3.8 di bawah ini.

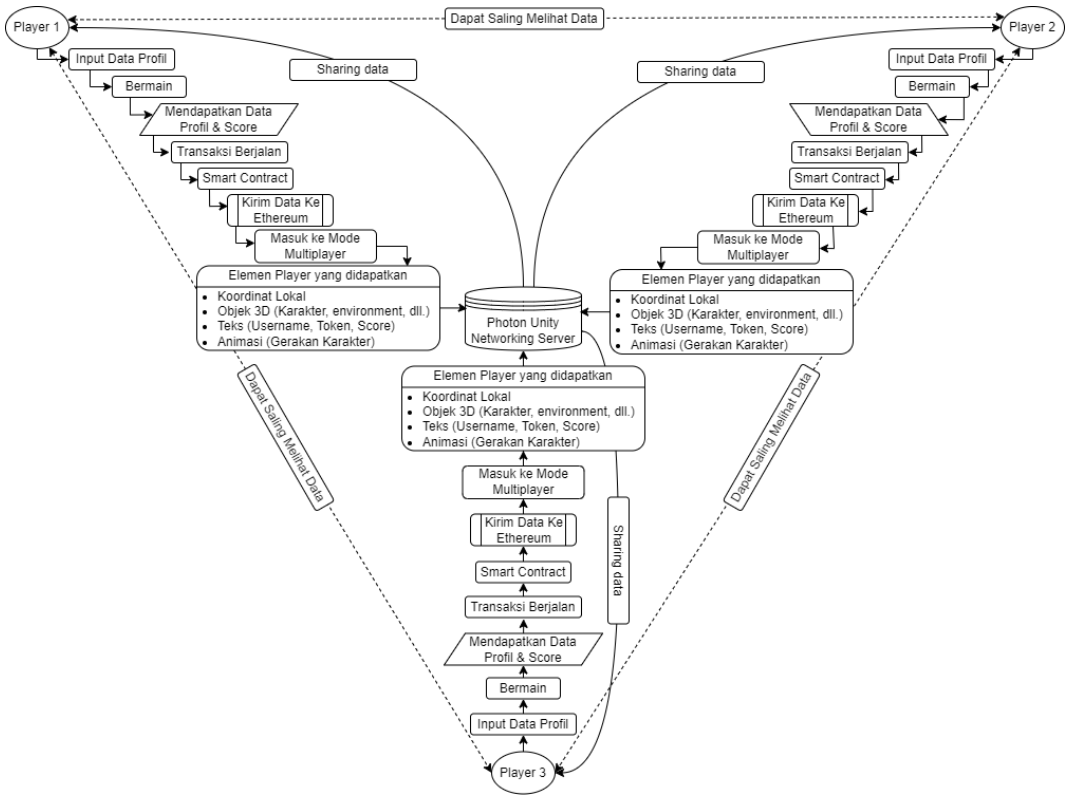


Gambar 0.8 Alur Sinkronisasi Data *Multiplayer*

Gambar 3.8 mengilustrasikan bagaimana berbagai komponen pengalaman bersama disinkronkan terhadap tiga pengguna, dengan menggunakan *Photon Unity Networking* dan RPC (*Remote Procedure Calls*) yang menghubungkan status konten pada tiga pengguna.

3.5 Perancangan *Data Sharing*

Gambar 3.8 adalah rancangan alur *data sharing* yang akan diterapkan pada *game* simulasi perakitan komputer. Proses *data sharing* antar *player* akan dapat dilakukan ketika *player* sedang berada dalam satu room yang sama.



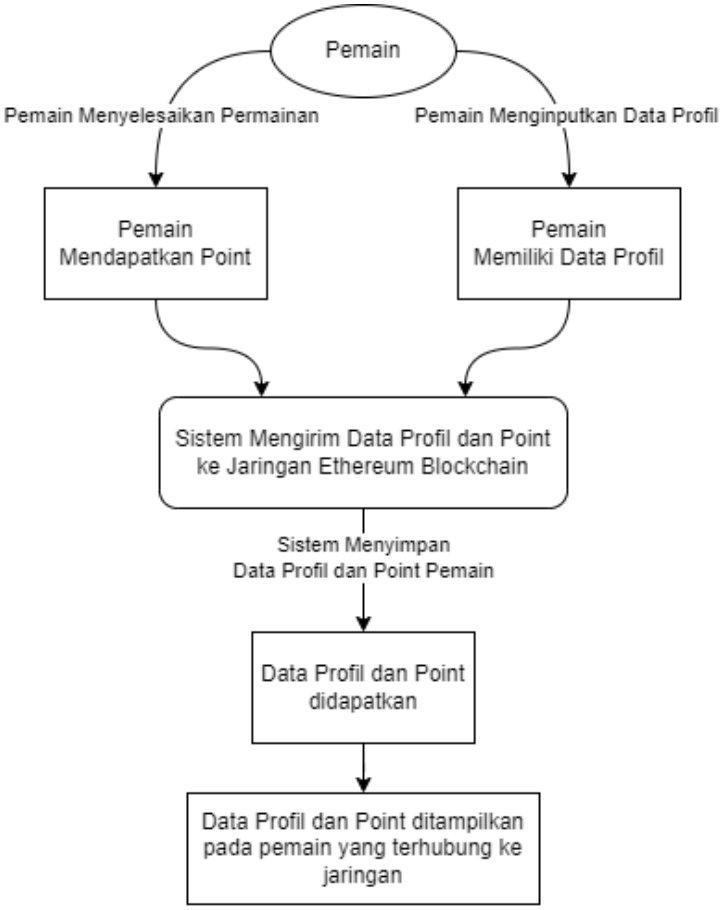
Gambar 0.9 Alur *Data sharing*

Pada awalnya suatu *player* akan melakukan input data profil terlebih dahulu ketika memasuki atau *login* ke dalam *game*, dengan begitu barulah *player* bisa memainkan *game*. Setelah menyelesaikan *game* maka data profil dan *score player* akan didapatkan kemudian akan dikirimkan ke jaringan *blockchain* dengan melewati *smart contract* yang telah ditentukan, sehingga data profil dan *score* antar pemain yang saling terhubung dalam jaringan dapat dikirim dan ditampilkan.

Kemudian ketika player memasuki mode multiplayer maka akan ada beberapa elemen pada game yang akan dipanggil antara lain sistem koordinat lokal, objek 3D , teks, animasi. Adapun untuk data profil dan *score* yang sebelumnya sudah termasuk sebagai elemen teks dalam game. Selanjutkan kumpulan elemen data tersebut akan dikirimkan menuju *Photon Server*, kemudian

dari *Photon Server* akan sinkronisasikan dan disiarkan kembali kepada tiap-tiap player yang terhubung.

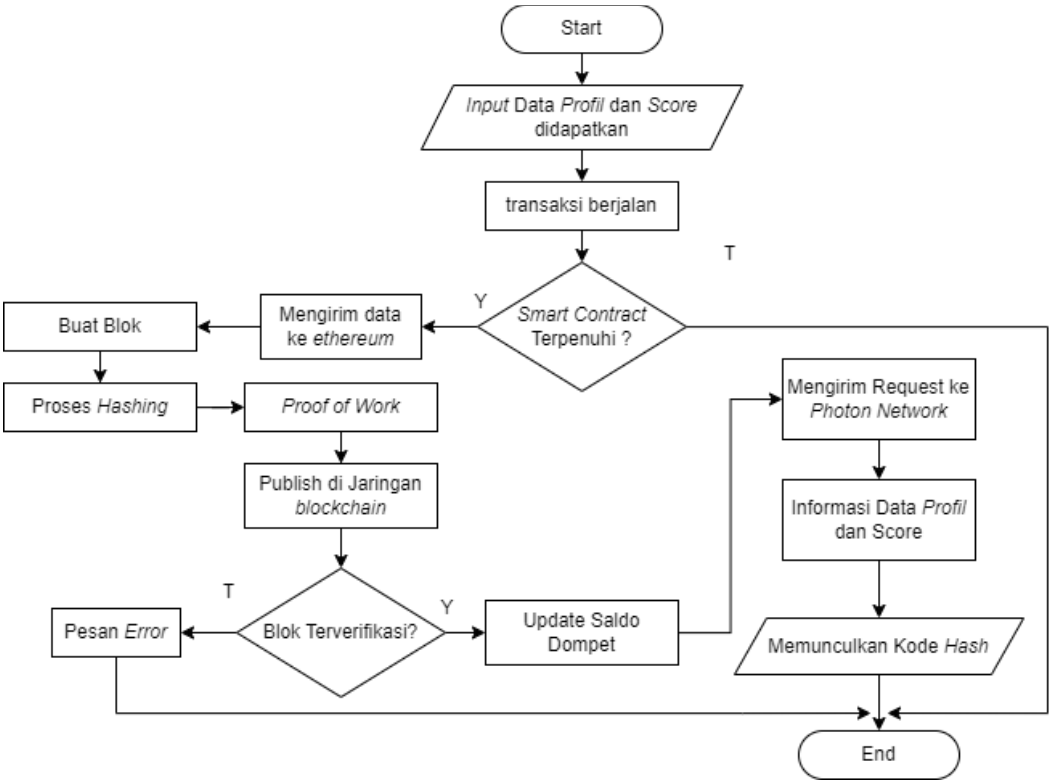
Setiap pemain dapat memiliki profil unik yang dapat disimpan di *Blockchain*. Ini memungkinkan pengguna untuk membagikan informasi profil mereka dengan mudah ke pemain lain dalam *game*, tanpa harus khawatir tentang kerahasiaan atau penggunaan data yang tidak sah. Kemudian maksud dari skor atau point yang dipunyai pemain adalah skor yang diperoleh ketika menyelesaikan permainan simulasi merakit komputer yang mana diiilustrasikan pada Gambar 3.10 sebagai alur data pemain.



Gambar 0.10 Alur Data Pemain

3.6 Penggunaan *Blockchain* dalam *Game*

Pada *game* simulasi merakit komputer ini akan terdapat jaringan *blockchain* yang terintegrasi di dalam sistem. Penyimpanan data informasi poin satu sama lain pada game simulasi perakitan komputer akan menggunakan jaringan *blockchain* dengan memanfaatkan *smart contract*. Poin bisa didapatkan oleh para pemain ketika mereka berhasil menyelesaikan permainan, kemudian data tersebut akan diambil dan disimpan dalam jaringan *Ethereum Blockchain* dalam bentuk transaksi. Penggunaan *Ethereum* Blockchain dipilih karena selain menjadi satu-satunya jaringan Blockchain yang mendukung integrasi dengan Game Engine Unity 3D melalui *library* N*ethereum*, jaringan *Ethereum* juga memiliki sifat yang dapat diprogram, memungkinkan penyesuaian sesuai dengan kebutuhan para pengembang.



Gambar 0.11 *Flowchart Blockchain* pada *Game*

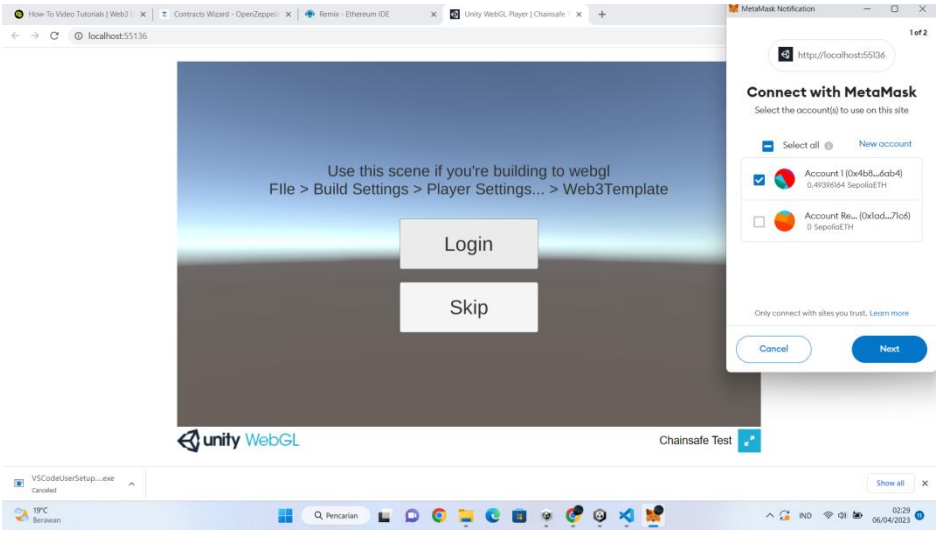
Flowchart pada Gambar 3.11 menjelaskan terkait dengan penggunaan *blockchain* dalam permainan ini. *Blockchain* diatur melalui pemahaman perdagangan antara pihak A dan pihak B, yang memicu pembentukan *smart contract* sebelum pengaturan dimulai. *Smart contract* berisi kode komputasi yang memastikan pemenuhan persyaratan perdagangan yang telah disepakati oleh kedua belah pihak hingga saat ini. Setelah semua kondisi dalam *smart contract* terpenuhi, pertukaran utama akan dilakukan.

3.7 Uji Coba *Ethereum Blockchain* pada *Unity*

Adapun uji coba yang dilakukan dalam penelitian ini merupakan uji coba implementasi sederhana *blockchain* di dalam *unity* engine.

3.7.1 Uji Coba *Login* dengan *Wallet Metamask*

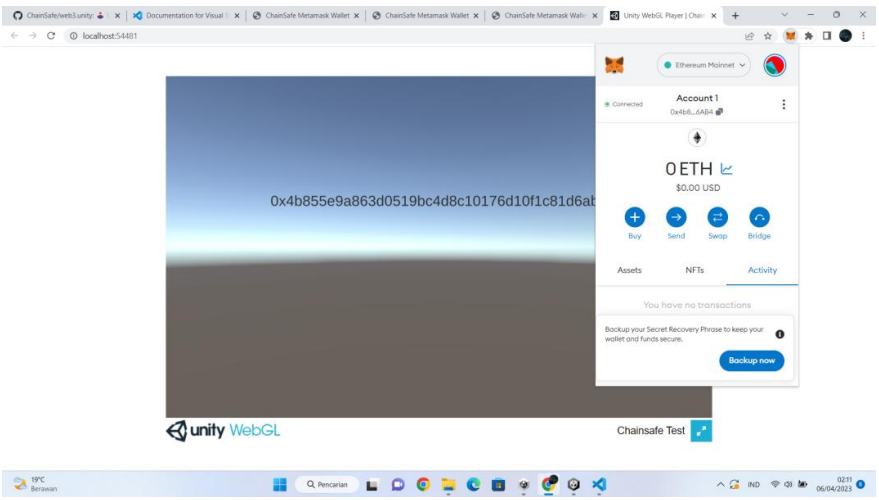
Uji coba pertama yang akan dilakukan ialah melakukan login pada akun Metamask kita melalui project unity yang telah di build dan run dalam bentuk Web3. Sebelumnya perlu disiapkan telebih dahulu wallet Metamask dalam browser kita, dalam hal ini saya menggunakan browser Google Chrome dan wallet Metamask dalam bentuk ekstension. Adapun kemudian player hanya perlu menekan tombol login, maka secara otomatis wallet Metamask akan muncul dan kita hanya perlu memasukan password dari wallet Metamask kita sendiri. Apabila kata sandi yang dimasukan benar maka login dinyatakan berhasil dan akan membawa kita menuju scene yang berikutnya. Berikut ini pada Gambar 3.12 adalah ketika koneksi wallet metamask terhubung dengan game.



Gambar 0.12 *Login ethereum* dengan ekstensi *Metamask*

3.7.2 Menampilkan *Id Wallet*

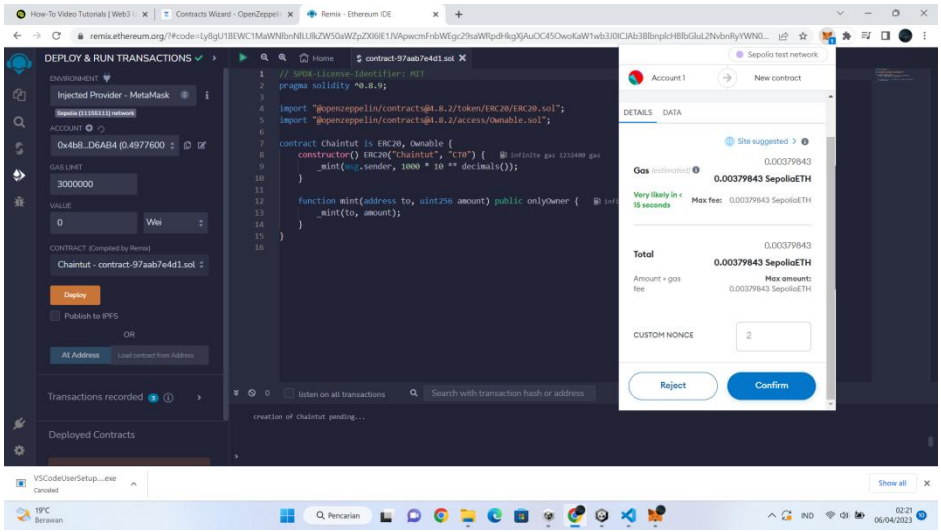
Uji coba selanjutnya ialah menampilkan *address wallet* kita setelah berhasil *login* ke dalam *unity*. Pengujian ini dan juga ketika *login* dilakukan dengan melakukan *build* terlebih dahulu dari *scene login* pada *unity* kedalam bentuk *Web3*. Setelah sebelumnya kita berhasil untuk *login*, maka kita akan diarahkan *menu*ju *scene* yang berikutnya yaitu menampilkan id dari akun *wallet* kita, seperti dalam Gambar 3.13



Gambar 0.13 Menampilkan *address wallet*

3.7.3 Membuat *Smart Contract*

Pada uji coba ini kita akan membuat implementasi sederhana *smart contract ERC20* agar dapat melakukan fungsi pengiriman atau aksi *transfer*, yang kemudian akan disematkan ke dalam *unity*. *Smart contract* akan dibuat dengan menggunakan *IDE Remix* yang kemudian akan coba di *deploy* kepada akun *Metamask* kita, berikut ini adalah tampilannya seperti yang terlihat pada Gambar 3.14 Adapun isi dari *smart contract* tersebut ialah melakukan pengiriman *eth* dari *account* 1 terhadap *account receiver*.

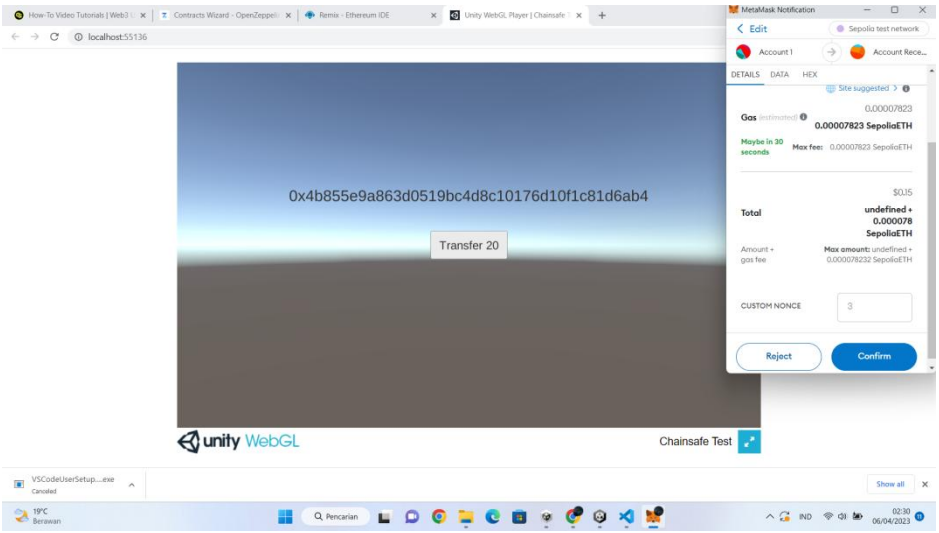


Gambar 0.14 Membuat *smart contract* dengan *IDE Remix*

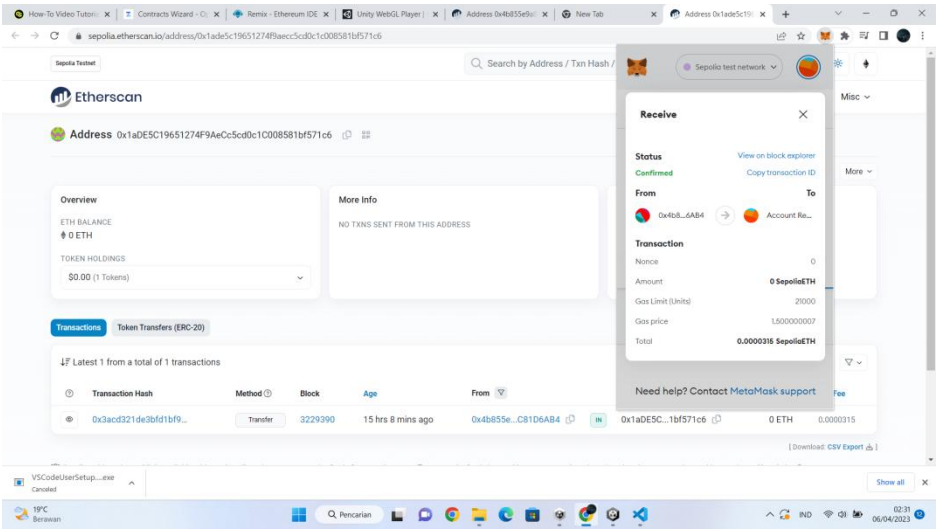
3.7.4 Melakukan Transaksi

Percobaan selanjutnya ialah melakukan transaksi pengiriman *eth* melalui *unity*. Setelah membuat *smart contract* tadi maka kita akan mendapatkan alamat dari *smart contract* tersebut. Melalui alamat tersebut kita akan sematkan ke dalam *unity* agar *smart contract* tersebut dapat digunakan, dalam hal ini diimplementasikan ke dalam bentuk transfer seperti dalam Gambar 3.15 dan 3.16

Ketika kondisi kita sudah *login* dengan menekan tombol *transfer*, maka *smart contract* akan dijalankan.



Gambar 0.15 Melakukan transaksi pengiriman *eth*



Gambar 0.16 Detail transaksi yang telah berhasil tercatat *blockchain*

3.8 Rancangan Pengujian

Adanya skenario pengujian dilaksanakan guna mengonfirmasi bahwa jaringan *Ethereum* Blockchain serta sistem yang sedang dikembangkan beroperasi dengan efisien sesuai dengan perencanaan awal. Pengujian ini juga diperlukan

untuk mengevaluasi kinerja jaringan *Ethereum* Blockchain saat diterapkan dalam konteks simulasi perakitan komputer.

3.8.1 Analisis dan Perancangan *Game*

Pengujian koneksi bertujuan untuk memastikan bahwa Jaringan *Ethereum* Blockchain telah berhasil terintegrasi dengan Game Engine Unity 3D dan bahwa Game Simulasi Perakitan Komputer berjalan dengan lancar. Pengujian ini akan dilakukan dengan memastikan koneksi internet tersedia. Hasil dari pengujian ini, yang tercatat dalam Tabel 3.2, akan dibahas lebih lanjut dalam bab berikutnya.

Tabel 0.2 Pengujian Koneksi

No	Skenario Pengujian	Hasil Pengujian
1	Dalam Console Unity akan memunculkan tulisan "Koneksi berhasil" dan juga menampilkan Token Hex Code sesuain *addres* pemain.	
2	Dalam *Game* akan memunculkan saldo token yang dimiliki pemain	

3.8.2 Rancangan Pengujian

Pengujian *data sharing* bertujuan untuk memastikan bahwa data yang telah dibagikan dalam sistem dapat digunakan dan ditampilkan dengan tepat. Hasil dari pengujian ini, yang dicatat dalam Tabel 3.3, akan dibahas lebih lanjut dalam bab berikutnya.

Tabel 0.3 Pengujian *Data sharing*

No	Skenario Pengujian	Hasil Pengujian
1	Pemain dapat melihat saldo token yang mereka miliki	
2	Terdapat catatan transaksi poin player yang terverivikasi dalam jaringan *blockchain*	
3	Seluruh pemain dalam *room* yang sama dapat melihat nama, *score* dan sisa saldo yang mereka dapat dalam *Game*	

BAB IV

IMPLEMENTASI DAN PEMBAHASAN

4.1 Perangkat Lunak yang Digunakan

Adapun untuk mengembangkan dan melakukan pengujian dalam *game* simulasi perakitan komputer digunakan perangkat lunak seperti dalam Tabel 4.1 sebagai berikut :

Tabel 0.1 Perangkat Lunak Yang Digunakan

Sistem Operasi	Windows 10
Ethereum Wallet	*Metamask*
***Game* Engine**	*Unity*3D 2021.3.21f1
Script Editor	Visual Studio Comm*unity* 2020 17.4.2
3D Modeller	Blender 3D 2.93.1
UI Editor	Adobe Illustrator CC 2019

4.2 Data Penelitian

Adapun terdapat 3 perangkat keras yang digunakan dalam rangka uji coba penelitian ini. Dalam sebuah *game multiplayer* akan dibutuhkan beberapa perangkat yang berbeda dalam pengujiannya guna mengetahui apakah aplikasi berjalan dengan sesuai. Berikut ini pada Tabel 4.2, 4.3 dan 4.4 adalah tiga perangkat keras yang digunakan dalam penelitian *game* simulasi merakit komputer:

Tabel 0.2 Perangkat 1

Perangkat	Laptop
Brand	Acer
Processor	Intel Core i5
Sistem Operasi	Windows 10
GPU	Intel(R) HD Graphics 520
RAM	8 GB

Tabel 0.3 Perangkat 2

Perangkat	Laptop
Brand	Dell
Processor	Intel Core i5
Sistem Operasi	Windows 11
GPU	NVIDIA T550
RAM	8 GB

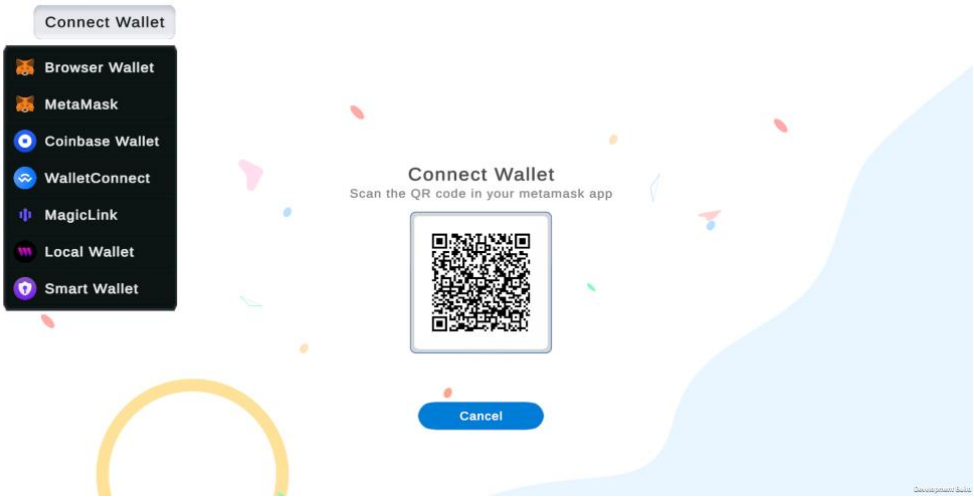
Tabel 0.4 Perangkat 3

Perangkat	Laptop
Brand	Acer
Processor	Intel Core i5
Sistem Operasi	Windows 11
GPU	NVIDIA T550
RAM	8 GB

4.3 Implementasi Skenario *Game*

Dalam sub-bab implementasi skenario game, ditampilkan implementasi visual dari game simulasi perakitan komputer yang telah dirancang dengan metode yang dijelaskan dalam bab sebelumnya. Berikut adalah tampilan ketika game dijalankan.

1. Koneksi *Wallet Metamask*



Gambar 0.1 Koneksi *Wallet Metamask*

Gambar 4.1 merupakan tampilan *menu* untuk *player* ketika ingin mengkoneksikan akun dompet *blockchain* miliknya sebelum memainkan permainan. *Player* dapat mengkoneksikan akun dompet *blockchain* dengan menggunakan *wallet Metamask* melalui *extension* yang ada di *google chrome* atau melalui *scanning wallet Metamask* pada *smartphone* yang dimiliki *player*. Ketika akun *wallet blockcahin* sudah terhubung barulah pemain dapat memainkan *game* simulasi merakit komputer.

2. *Main Menu*

Gambar 0.2 *Main Menu*

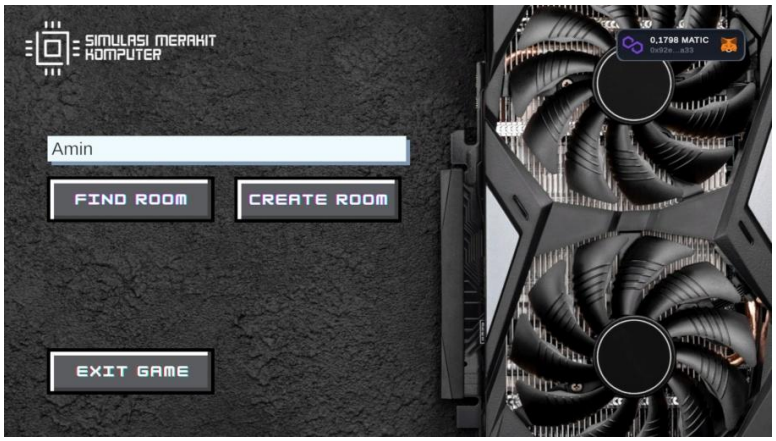
Gambar 4.2 adalah tampilan menu utama dimana *player* bisa memilih tiga tombol pilihan. Pertama yaitu *play game* dimana pemain akan langsung diarahkan *menu*ju *scene* permainan simulasi merakit komputer dan akan memperoleh *score* setelah menyelesaikan permainan. Tombol *multiplayer* digunakan untuk memasuki *room multiplayer*, sehingga *player* dapat berinteraksi satu sama lain dengan *player* lainnya dan yang terakhir adalah tombol untuk keluar dari *game*.

3. *Gameplay* Permainan

Gambar 0.3 *Gameplay* Permainan

Gambar 4.3 merupakan tampilan ketika *player* memilih *button play game* maka akan langsung diarahkan pada *scene* permainan. Pada saat *player* sudah memasuki permainan waktu hitung mundur akan ditampilkan yang menandakan sebagai indikator batas waktu permainan dan juga *player* dihadapkan dengan komponen – komponen komputer secara terpisah yang siap untuk dirakit oleh *player*. Setelah *player* berhasil menyelesaikan permainan baru akan ditampilkan skor yang telah diperoleh sesuai dengan waktu pengerjaan atau batas sisa waktu yang didapatkan.

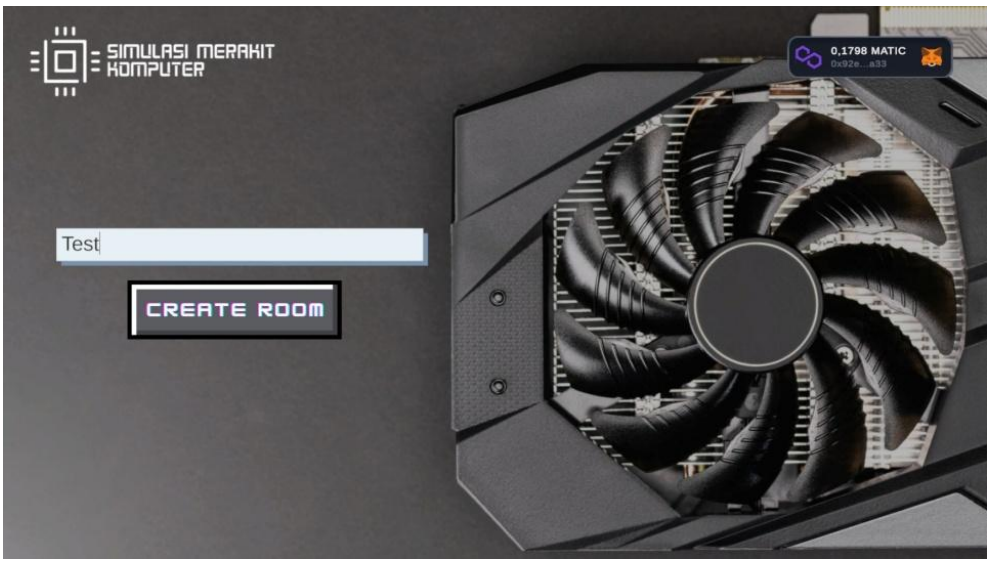
4. *Multiplayer Menu*



Gambar 0.4 *Multiplayer Menu*

Tampilan pada Gambar 4.4 merupakan bagian *multiplayer menu* yang mana dapat digunakan untuk *player* memilih membuat *room multiplayer* nya sendiri atau memilih untuk mencari *room* yang ada dan kemudian masuk kedalam *room* tersebut. Sebelum memilih kedua pilihan tersebut *player* diharuskan mengisi terlebih dahulu *nickcname* yang ingin dipakainya.

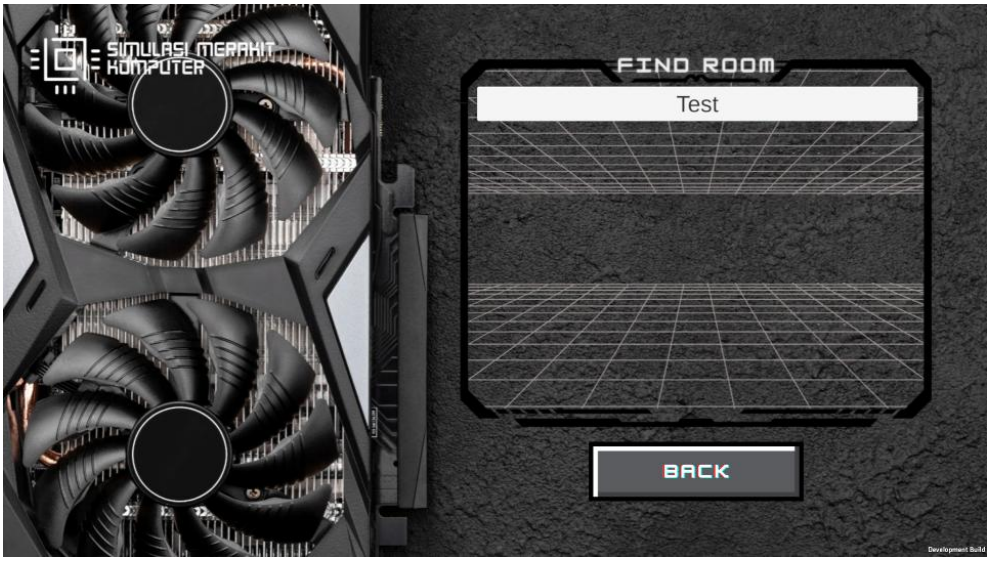
5. *Create Room*



Gambar 0.5 *Create Room*

Tampilan gambar 4.5 berikut ini merupakan ketika *player* memilih untuk menekan tombol *create room*. Untuk membuat *room player* diharuskan mengisi nama *room* nya terlebih dahulu, baru setelah itu menekan *button create*. Ketika *room* berhasil dibuat maka akan diarahkan menuju *waiting room menu* dan status *player* yang telah membuat *room* akan dijadikan sebagai master *room*.

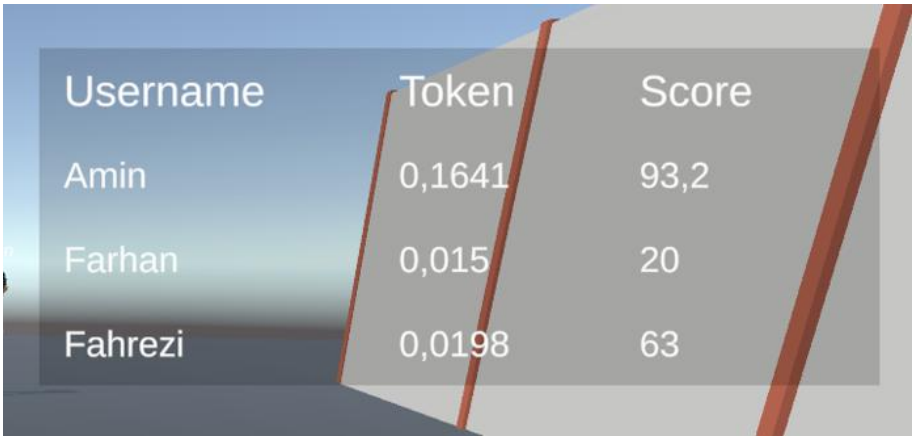
6. *Find Room*



Gambar 0.6 *Find Room*

Adapun jika *player* memilih untuk menekan tombol *find room* maka akan ditampilkan terlebih dahulu *room* apa saja yang tersedia dan kemudian baru *player* dapat memilih *room* mana yang ingin dimasuki seperti pada Gambar 4.6. Namun jika tidak tersedia *room* apapun maka *player* dapat kembali ke *menu create room* dan membuat *room*nya sendiri.

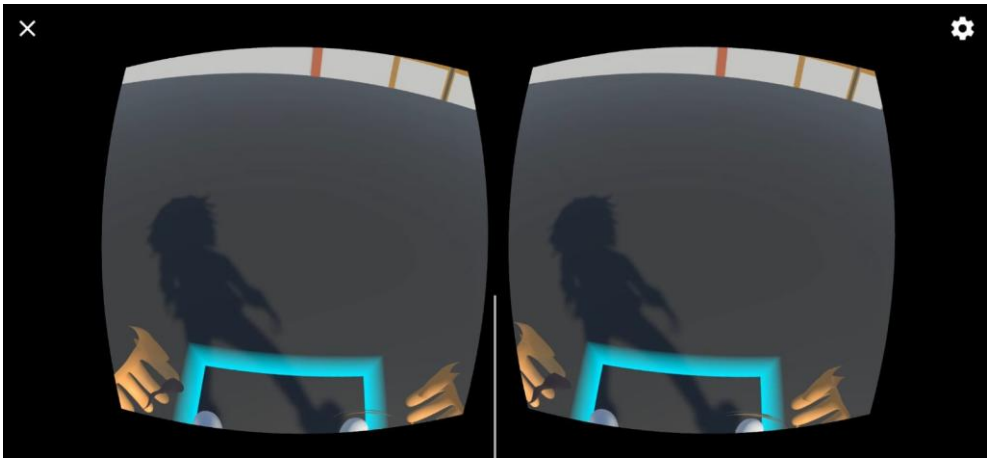
7. Tampilan *Scoreboard Player*



Gambar 0.7 *Scoreboard*

Gambar 4.7 adalah tampilan keseluruhan score yang dimiliki player yang ditampilkan dalam bentuk *scoreboard*. *Player* yang telah menyelesaikan permainannya maka dapat melakukan *claim score* yang didapatnya baru kemudian memasuki *room multiplayer*. Pada *room multiplayer* inilah setiap *player* dapat melihat informasi seperti *score*, saldo, dan juga nama yang mana tertera dalam *scoreboard*.

8. Tampilan *Virtual Reality*



Gambar 0.8 *Virtual Reality*

Gambar 4.8 merupakan tampilan ketika game menggunakan mode *virtual reality*. Adapun penerapan *virtual reality* pada game ini terbatas hanya pada tampilan *view* nya saja, sedangkan *controller* untuk pergerakan *player* dilakukan dengan inputan melalui *keyboard.*

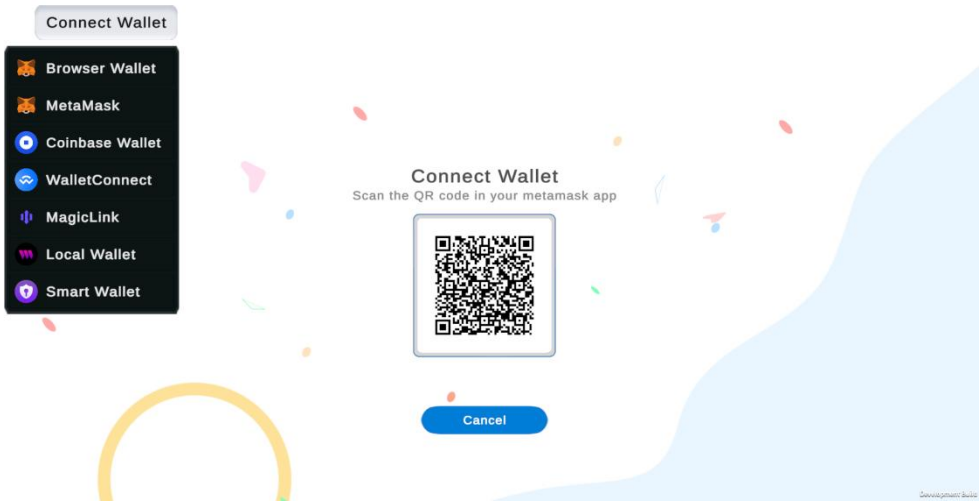
4.4 Uji Coba

Adanya tahap uji coba bertujuan untuk mengevaluasi apakah sistem yang telah dibuat mampu beroperasi sesuai dengan harapan. Selama tahap pengujian ini, kita akan mengamati bagaimana di dalam game jaringan blockchain dapat

terhubung dan berfungsi, serta bagaimana proses berbagi data skor, saldo, dan nama dapat berlangsung.

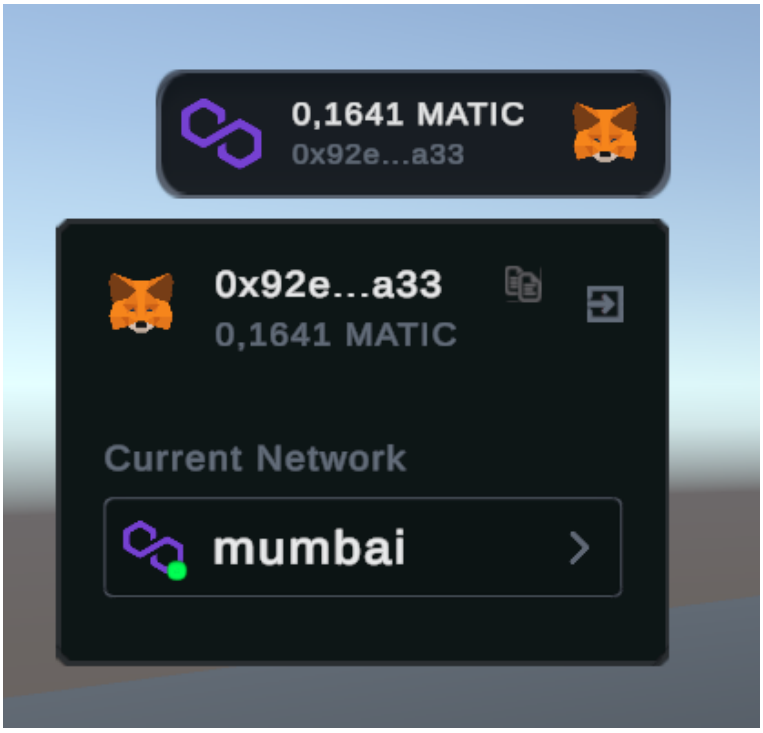
4.4.1 Uji Koneksi *Blockchain* dengan *Wallet Metamask*

Uji coba tersebut dilakukan guna mengetahui apakah jaringan *blockchain* dapat terhubung ke dalam *game*. Adanya koneksi antara *blockchain* dengan *game* sangat diperlukan oleh *player*, sebab nantinya segala macam aktivitas transaksi seperti penyimpanan skor akan membutuhkan koneksi tersebut. Pengiriman dan penyimpanan data *score* yang akan dilakukan nanti juga membutuhkan sedikit biaya menggunakan saldo token dari akun *wallet blockchain player*.



Gambar 0.9 Koneksi *Wallet Ethereum*

Seperti pada gambar 4.9 *player* dapat mengkoneksikan akun *blockchain* miliknya menggunakan *wallet Metamask* dengan melakukan *scanning* melalui *smartphone*. Apabila koneksi jaringan *ethereum blockchain* berhasil dilakukan maka akan ditampilkan saldo yang dimiliki *player* beserta alamat *wallet*nya seperti pada gambar 4.10.



Gambar 0.10 Koneksi Sudah Terhubung

Di bawah ini adalah baris script yang digunakan untuk menghubungkan jaringan blockchain dalam penelitian game simulasi perakitan komputer sebagai berikut:

```
//method ini menggunakan operasi asynchronous
async void OnConnected(){
        try{
//Menonaktifkan panel email dan password
            passwordPanel.SetActive(false);
            emailPanel.SetActive(false);

//Ambil data jumlah saldo token dan alamat wallet
            string _chain =
ThirdwebManager.Instance.chain;
            CurrencyValue nativeBalance = await
ThirdwebManager.Instance.SDK.wallet.GetBalance();
            balanceText.text =
$"{nativeBalance.value.ToEth()}
{nativeBalance.symbol}";
            balanceText2.text = balanceText.text;
            walletAddressText.text = await
Utils.GetENSName(address) ??
```

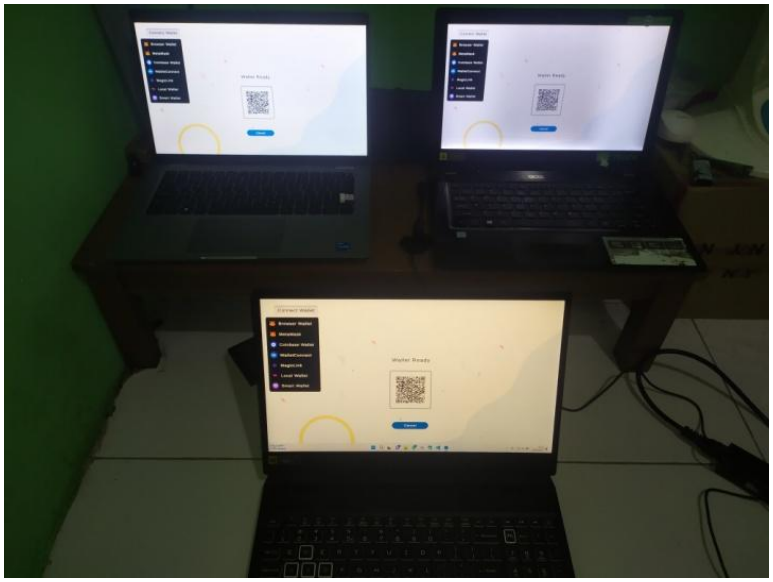
```
address.ShortenAddress();
            walletAddressText2.text =
walletAddressText.text;

// Menyembunyikan dan Menampilkan beberapa elemen UI
            connectButton.SetActive(false);
            connectedButton.SetActive(true);
            connectDropdown.SetActive(false);
            connectedDropdown.SetActive(false);
            networkDropdown.SetActive(false);
        }
        catch (Exception e)
        {
            Debug.LogWarning($"Error Fetching Native
Balance: {e.Message}");
        }
    }
```

Secara keseluruhan, script tersebut menangani berbagai tugas yang terkait dengan koneksi dan tampilan antarmuka pengguna (UI) dalam game dengan memanfaatkan library *Thirdweb*. Pelaksanaan uji coba dilaksanakan dengan menggunakan 3 perangkat berbeda dan juga 3 *account Metamask* yang berbeda. Uji coba dapat dikatakan berhasil apabila saldo token dan alamat *wallet* berhasil ditampilkan. Hasil percobaan koneksi dapat ditemukan dalam Tabel 4.5

Tabel 0.5 Pengujian koneksi *blockchain*

Account Metamask	**Perangkat 1**	**Perangkat 2**	**Perangkat 3**
Account 1	Berhasil	Berhasil	Berhasil
Account 2	Berhasil	Berhasil	Berhasil
Account 3	Berhasil	Berhasil	Berhasil

Terlihat pada Tabel 4.5 uji coba pengkoneksian jaringan *blockchain* dengan sistem melalui 3 perangkat yang berbeda dan juga 3 *account* yang berbeda menampilkan hasil yang tinggi pada tingkat keberhasilannya.



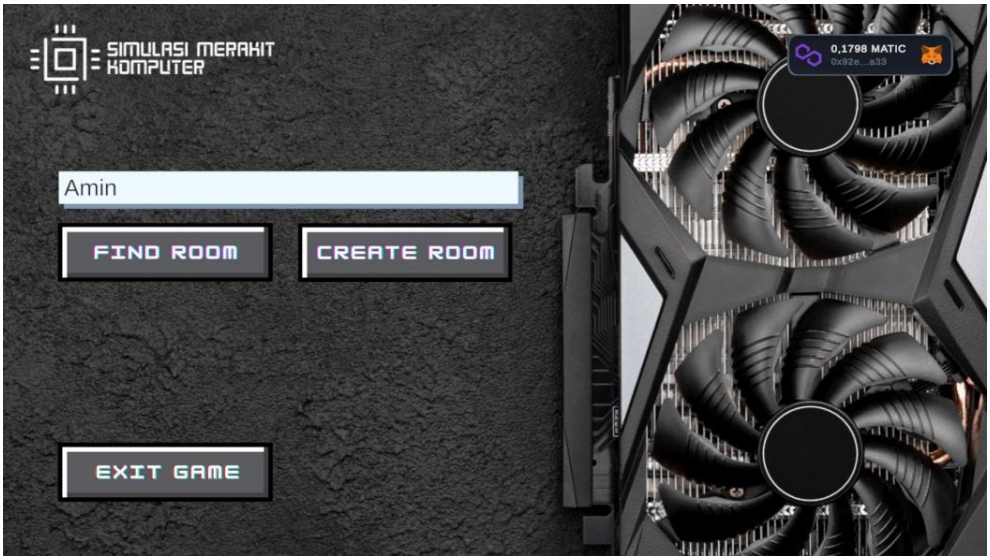
Gambar 0.11 Pengujian koneksi dengan 3 perangkat

4.4.2 Pengujian *Create Room*

Pengujian selanjutnya yang dilakukan ialah membuat *room* pada *game*. Dalam *game* simulasi merakit komputer ini *room* dapat dibuat sesuai dengan keinginan pengguna juga agar dapat memasuki mode *multiplayer* dan dapat berinteraksi satu sama lain dengan *player* lain. Uji coba ini dapat dilakukan pada *scene lobby game*. Berikut ini adalah baris *script* yang digunakan untuk *player* membuat *room*.

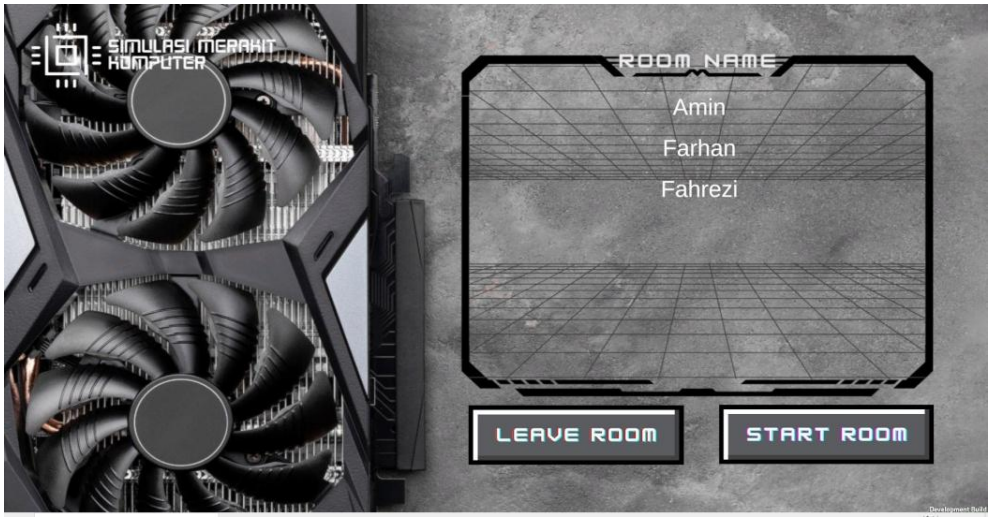
```
public void CreateRoom()
     {
//memeriksa apakah input field kosong atau null

     if(string.IsNullOrEmpty(roomNameInputField.text))
          {
               return;
          }

//Membuat Ruangan sesuai dengan nama room yang
diinputkan
     PhotonNetwork.CreateRoom(roomNameInputField.text);
          MenuManager.Instance.OpenMenu("loading");
     }
```

Script ini berfungsi untuk membuat ruangan baru dalam permainan menggunakan Photon Unity Networking dan memastikan bahwa nama ruangan valid sebelum membuatnya. Setelah itu, script membuka menu loading untuk memberi umpan balik visual kepada pengguna selama proses pembuatan ruangan. Pada *menu create room* ini *player* diharuskan terlebih dahulu menginputkan apa nama *player lobby game*, kemudian nama *room* yang ingin digunakan setelah itu barulah *room* dapat dibuat. Adapun berikut ini pada Gambar 4.12 merupakan tampilan menu *create room* :



Gambar 0.12 Tampilan *Create Room*

Pada tahap ini pengujian ini dianggap berhasil apabila room berhasil dibuat dan pemain yang membuat ruangan tersebut atau yang biasa disebut room master, dapat masuk ke dalamnya. Ketika room berhasil dibuat, tampilan antarmuka pengguna (UI) dalam permainan akan mengalami perubahan sesuai dengan Gambar yang diberikan 4.13



Gambar 0.13 *Room* telah dibuat

Gambar 4.13 merupakan tampilan apabila room berhasil dibuat oleh *player*. Gambar ini mencakup nama panggilan (nickname) dari semua pemain yang telah bergabung dalam ruangan tersebut. Hal ini menunjukkan bahwa ruangan telah berhasil dibuat dan *room master* atau pembuat ruangan sedang menunggu pemain lain (node) untuk masuk ke dalam ruangan yang telah dibuat. Data pengujian mencakup penciptaan ruangan dengan menggunakan 3 perangkat berbeda. Pengujian dinyatakan berhasil apabila room berhasil dibuat.

Tabel 0.6 Uji *Create Room*

Create Room	**Perangkat 1**	**Perangkat 2**	**Perangkat 3**
Create 1	Berhasil	Berhasil	Berhasil
Create 2	Berhasil	Berhasil	Berhasil
Create 3	Berhasil	Berhasil	Berhasil
Create 4	Berhasil	Berhasil	Berhasil
Create 5	Berhasil	Berhasil	Berhasil

Adapun hasil dari pengujian *create room* pada *game* simulasi merakit komputer berdasarkan Tabel 4.6, dapat dikatakan menampilkan hasil yang tinggi dengan melakukan 5 kali percobaan menggunakan 3 perangkat.

Gambar 0.14 *Create Room* dengan 3 perangkat

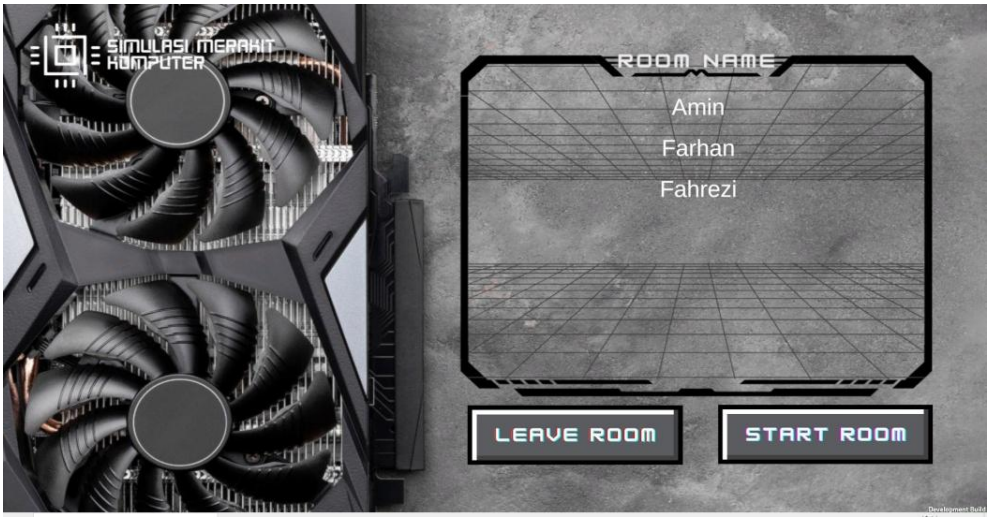
4.4.3 Pengujian *Join Room*

Selanjutnya pengujian yang dilakukan adalah memasuki *room* yang tersedia dimana *player* yang lain sudah terlebih dahulu membuat *room*. Tahap tersebut dapat dikerjakan oleh *player* pada *scene lobby game*. Melalui uji coba ini juga konektivitas antar *player* dapat diperiksa apakah berjalan dengan benar atau tidak. Adapun untuk dapat ikut dengan pembuat *room* adalah dengan cara melihat terlebih dahulu daftar *room* yang tersedia. Berikut ini adalah *script* yang dipakai agar dapat menampilkan daftar *room* yang tersedia dalam *game* simulasi merakit komputer.

```
// Override Method, dipanggil otomatis ketika ada
perubahan seperti ketika ruangan baru dibuat atau
dihapus
public override void OnRoomListUpdate(List<RoomInfo>
roomList)
     {
// Menghapus Elemen UI yang Lama:
          foreach(Transform trans in roomListContent)
          {
```

```
                Destroy(trans.gameObject);
            }

// Membuat Elemen UI untuk Setiap Ruangan Baru:
            for(int i = 0; i < roomList.Count; i++)
            {
                if(roomList[i].RemovedFromList)
                    continue;
                Instantiate(roomListItemPrefab,
roomListContent).GetComponent<RoomListItem>().SetUp(roo
mList[i]);
            }
        }
```

Script ini memastikan bahwa tampilan daftar ruangan selalu diperbarui dengan informasi terbaru setiap kali daftar ruangan berubah dalam game. Ketika pemain menekan tombol "Find Room" yang terletak di menu utama (scene main menu), tampilan antarmuka pengguna (UI) akan mengalami perubahan sesuai dengan yang ditunjukkan dalam Gambar 4.15.



Gambar 0.15 Tampilan *Find Room*

Berikut ini merupakan data hasil pengujian *Join Room* melalui 3 perangkat yang berbeda. Pengujian dinyatakan berhasil apabila perangkat sukses masuk ke dalam *Room Master* (*room host*). Pengujian akan dianggap berhasil apabila

perangkat dapat sukses masuk ke dalam *room* yang dimiliki tuan rumah (room host) atau Room Master.

Tabel 0.7 Uji *Join Room*

Join Room	**Perangkat 1**	**Perangkat 2**	**Perangkat 3**
Join 1	Berhasil	Berhasil	Berhasil
Join 2	Berhasil	Berhasil	Berhasil
Join 3	Berhasil	Berhasil	Berhasil
Join 4	Berhasil	Berhasil	Berhasil
Join 5	Berhasil	Berhasil	Berhasil

Adapun hasil pada pengujian *join room* dalam *game* simulasi merakit komputer berdasarkan Tabel 4.7, dapat dikatakan menampilkan hasil yang tinggi dengan melakukan 5 kali percobaan menggunakan 3 perangkat.

Gambar 0.16 *Join Room* dengan 3 perangkat

4.4.4 Pengujian *Claim Score dengan* ERC20

Pengujian ini dilakukan ketika *player* mulai memasuki *scene* permainan merakit komputer. *Player* diharuskan menempatkan komponen - komponen komputer yang diberikan sesuai dengan letak posisi dan urutannya dengan batas waktu yang diberikan.

Gambar 0.17 Tampilan Bermain Simulasi Merakit Komputer

Score yang akan didapatkan *player* dipengaruhi oleh sisa waktu yang ada setelah berhasil menyelesaikan permainan seperti dalam Gambar 4.17. Berikut adalah baris *script* yang digunakan guna memperoleh *score* dari hasil permainan.

```
void HitungSkor()
    {
// Pengecekan Waktu:
        if (currentTime <= 0)
        {
            score = 0;
            ScoreText.text = "skor = " + score;
        }
//Pengaturan Skor dari Waktu (Jika Waktu Masih
Tersisa):
        else
        {
            score = currentTime;
            ScoreText.text = "skor = " + score;
        }
    }
```

Jika currentTime masih lebih dari 0, maka nilai score diatur sama dengan currentTime. Ini bisa berarti bahwa skor dalam konteks ini adalah sisa waktu permainan. Kemudian, teks pada objek ScoreText diperbarui sesuai dengan nilai skor yang baru. Secara keseluruhan bertujuan untuk mengelola skor berdasarkan

kondisi waktu tertentu dalam permainan. Jika waktu sudah habis, skor diatur menjadi 0 dan jika waktu masih tersisa skor diatur sama dengan nilai waktu.

Gambar 4.18 menggambarkan tampilan yang muncul ketika seorang pemain berhasil menyelesaikan permainan, maka setelah itu secara otomatis akan ditampilkan berapa *score* yang didaptkan *player*. Kemudian dengan menekan tombol *Claim score* maka sistem akan melakukan proses *minting* menggunakan *smart contract* Drop *ERC20*. Apabila proses berhasil maka *score* yang didapat oleh pemain akan tersimpan pada jaring *blockchain* sesuai alamat yang dipunyai oleh pemain.

Gambar 0.18 Tampilan *Claim Score*

Berikut ini pada tabel 4.8 adalah hasil pengujian *claim score* pada *game* simulasi merakit komputer menggunakan 3 perangkat dan akun *blockchain* yang berbeda dengan hasil mendapatkan nilai *score* sesuai dengan yang ditampilkan.

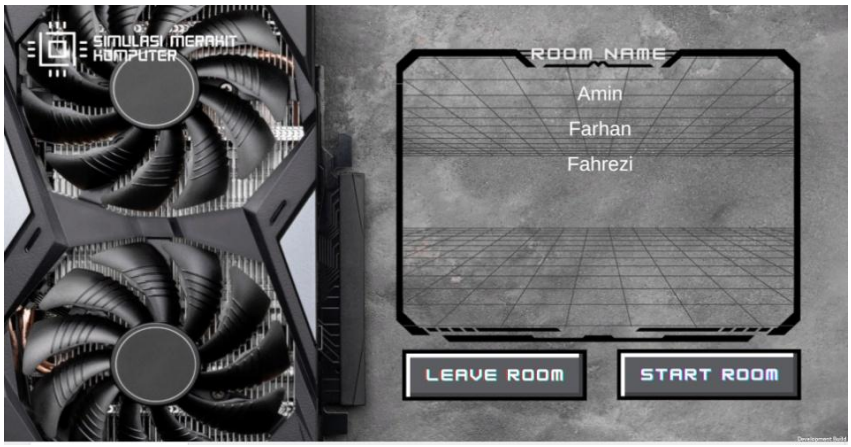
Tabel 0.8 Uji *Claim score*

Perangkat	***Account***	***Claim score***	***Score***
1	Amin	Berhasil	50
2	Farhan	Berhasil	70
3	Fahrezi	Berhasil	20

Gambar 0.19 Pengujian *Claim Score* dengan 3 Perangkat

4.4.5 Pengujian *Multiplayer*

Pada pada penelitian ini *game* simulasi merakit komputer ini terdapat *menu multiplayer* dimana pada *menu* ini pemain dapat memungkinkan untuk bisa saling berinteraksi satu sama lainnya. Adanya sistem *multiplayer* ini juga ditujukan guna dapat melakukan *data sharing* antar *player* yang saling terhubung. Untuk membangun sistem *multiplayer* pada penelitian ini maka digunakanlah Photon 2 sebagai mesin jaringan (*networking engine*).



Gambar 0.20 Tampilan *Room Master*

Gambar 4.20 menggambarkan tampilan dari pembuat *room* (*room master*). Untuk bisa memilih tombol *Start Room* hanya pembuat *room* yang mempunyai

akses tersebut. Akan terdapat daftar *list* setiap pemain yang berada di dalam *room* tersebut untuk dapat bermain bersama dengan ditampilkan berupa *nickname* setiap *Player*. Adapun berikut ini adalah tahap pengujian *multiplayer*, yang mana dijalankan dengan cara memilih tombol Start *Room* oleh pembuat *room*. Melalui pengujian tersebut maka dapat diamati apakah dari ke 3 perangkat yang digunakan untuk pengujian bisa memasuki *room* dan saling berinteraksi di dalamnya.

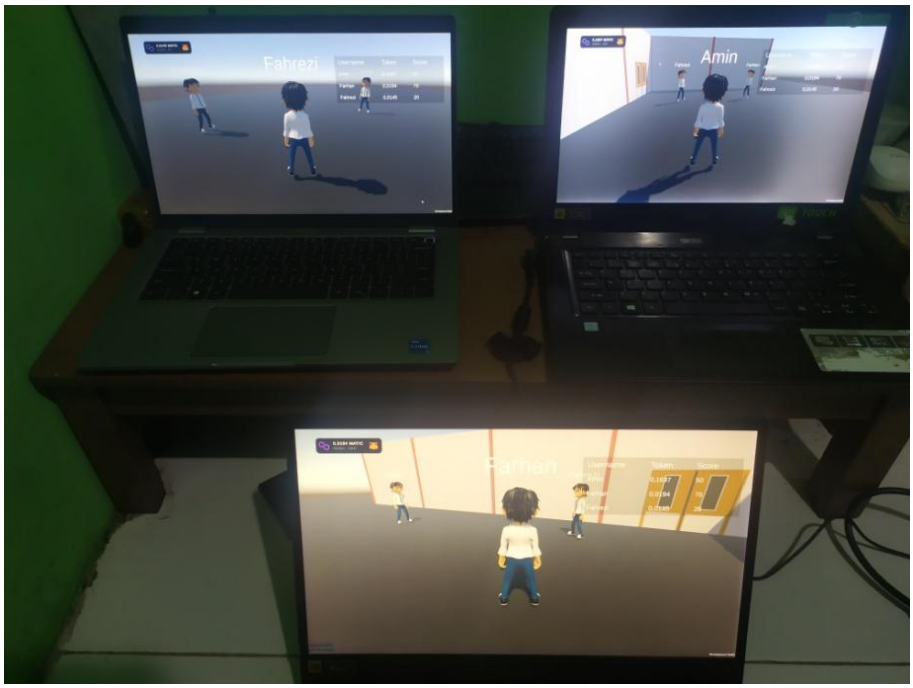
Gambar 0.21 Tampilan *Multiplayer*

Berdasarkan gambar 4.21 maka dapat diketahui bahwasanya seluruh *player* dapat saling terkoneksi dan masuk kedalam *room*. Pengujian ini dilakukan dengan menggunakan 3 perangkat yang tidak sama seperti yang sudah disebutkan sebelumnya. Berikut adalah hasil data pengujian *multiplayer* seperti yang tertera dalam Tabel 4.9

Tabel 0.9 Uji *Multiplayer*

Perangakat	***Nickname***	**Masuk ke dalam *Game***
1	Amin	Berhasil
2	Farhan	Berhasil
3	Fahrezi	Berhasil

Berdasarkan 4.9 maka dapat diketahui bahwasanya hasil pada pengujian sistem *multiplayer* dalam *game* simulasi merakit komputer melalui 3 perangkat yang saling berbeda seluruhnya berhasil dilakukan yaitu terpanggil dan masuk dalam *room*.



Gambar 0.22 Pengujian *Multiplayer* dengan 3 perangkat

4.4.6 Pengujian *Data Sharing Nickname*

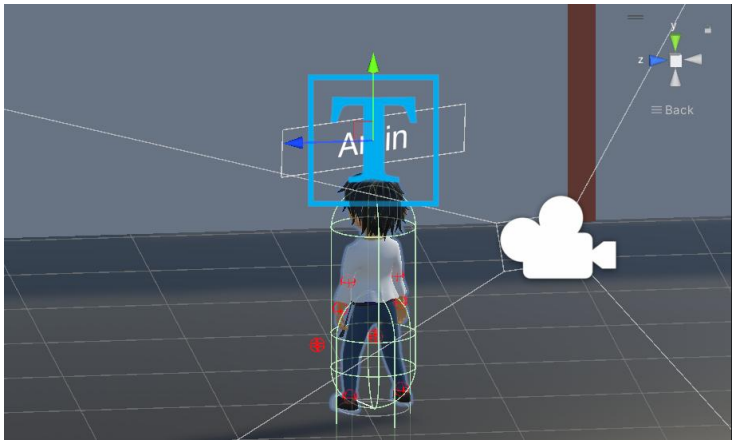
Langkah selanjutnya dalam pengujian adalah berfokus pada *data sharing* untuk nama pemain. Adanya pengujian tersebut guna mengamati apakah nama pemain dapat ditampilkan dengan sesuai dan seluruh player yang terhubung dapat melihatnya satu sama lain. Proses *data sharing* nama pemain dinyatakan sukses apabila antara *player* yang terhubung saling dapat melihat *nickname* yang dimilikinya. Pada tahap ini penerapan *Photon networking engin*e akan mempengaruhi bagaimana hasil uji coba yang akan didapat. Photon akan

mengkoordinasikan data nickname yang telah diinputkan sebelumnya dan telah terkoneksi dengan jaringan peer-to-peer.

Dari data nickname yang dimasukkan oleh setiap node, informasi tersebut akan didistribusikan melalui jaringan yang telah terhubung, dan kemudian akan ditampilkan kepada node-node lainnya. Hal ini memungkinkan semua node yang terhubung untuk melihat nickname satu sama lain. Penting untuk mencatat bahwa dalam penelitian game simulasi merakit komputer ini, yang merupakan tipe game multiplayer membedakan pemain satu dengan yang lain sangat krusial. Gambar 4.23 menggambarkan bahwa pemain atau node dengan nickname "AMIN" dapat melihat *nickname* dari node lainnya.

Gambar 0.23 *Data Sharing Nickname*

Adapun *script* untuk *data sharing nickname* akan disematkan ke dalam objek pemain yang telah buat sebelumnya dalam Unity, yang merupakan karakter utama dalam permainan seperti yang terlihat dalam Gambar 4.24.



Gambar 0.24 Objek karakter dalam *game*

Berikut ini merupakan *script* untuk melakukan *data sharing* pada *nickname player* dalam *game* simulasi merakit komputer.

```
public class UsernameDisplay : MonoBehaviour
{
//Deklarasi dan Inisialisasi Variabel:
     [SerializeField] PhotonView playerPV;
     [SerializeField] TMP_Text text;

     void Start()
     {
//memeriksa apakah objek ini (mungkin mewakili pemain)
          if(playerPV.IsMine)
          {
               gameObject.SetActive(true);//diaktifkan
          }
//menampilkan nama
          text.text = playerPV.Owner.Nickname;
     }
}
```

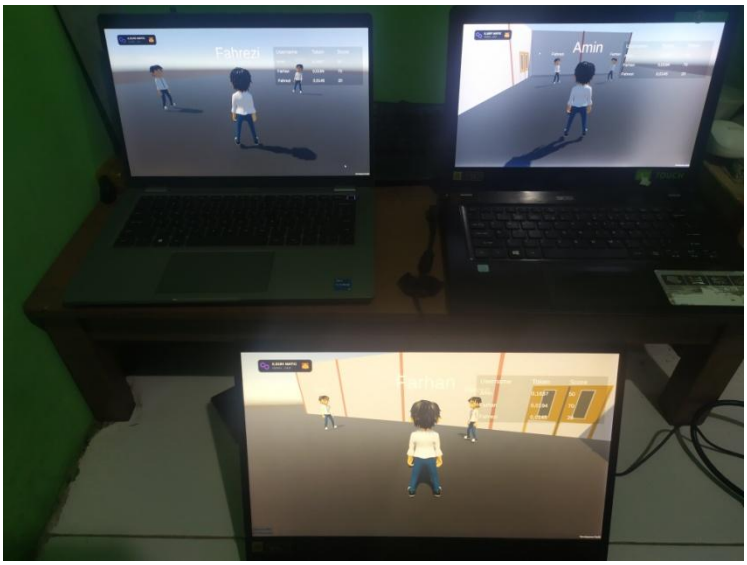
Seperti yang terlihat dalam baris script di atas, data nickname pemain diperoleh dari Photon Engine dengan mengambil data nickname dari pemilik node jaringan *Ethereum* Blockchain yang sesuai. Setelah data nickname tersebut didapat, kemudian akan ditampilkan pada objek game yang memiliki tipe teks

(string) sehingga dapat dilihat oleh semua pemain. Berikut ini adalah data hasil pengujian berbagi data nickname menggunakan 3 perangkat berbeda.

Tabel 0.10 Uji *Data sharing Nickname*

Perangkat	***Nickname***	***Spawn***	**Tampilan**
1	Amin	Berhasil	Sesuai
2	Farhan	Berhasil	Sesuai
3	Fahrezi	Berhasil	Sesuai

Dari Tabel 4.10, terlihat bahwa uji coba *data sharing* nickname pemain berhasil dilaksanakan. Hasilnya menunjukkan bahwa ketiga pemain dapat muncul dalam permainan, dan tampilan nickname pemain sesuai dengan nama yang mereka masukkan saat masuk ke dalam permainan.



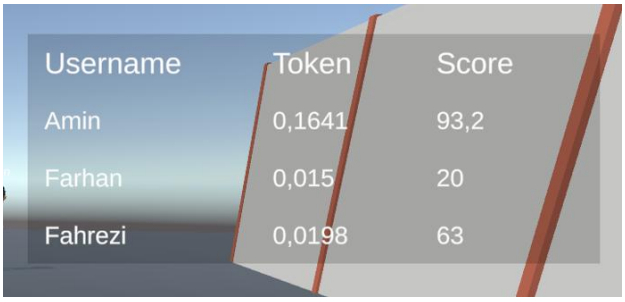
Gambar 0.25 Pengujian *Data sharing Nickname* dengan 3 Perangkat

4.4.7 Pengujian *Data Sharing Saldo* dan *Score*

Langkah pengujian selanjutnya ialah *data sharing* terhadap saldo token dan *score*. Pengujian tersebut dilakukan guna menentukan apakah setiap saldo dan *score* yang dimiliki masing-masing *player* dapat ditampilkan dengan sesuai juga bisa terlihat oleh *player* lain. Adapun informasi terkait dengan sisa saldo dan

score tiap pemain akan ditampilkan dalam bentuk *scoreboard* pada bagian pojok kanan atas UI. Setiap *player* dalam *room* dapat mengaktifkan *score*board tersebut dengan menekan tombol TAB pada *keyboard*. Berikut ini pada gambar 4.26 adalah tampilan antarmuka (UI) dari pemain yang menampilkan saldo dan *score* yang dimiliki oleh setiap *player* dalam satu *room*.

Gambar 0.26 Tampilan Antar Muka (UI) *Player*



Gambar 0.27 Informasi *Data sharing* Saldo dan *Score*

Dalam gambar 4.27 menampilkan kumpulan Informasi terkait nama, token dan score yang dimiliki oleh. Dapat dinyatakan bahwa dari *player* dengan *nickname* AMIN bisa mengetahui berapa sisa saldo dan *score* yang dimiliki oleh *player* dengan *nickname* FARHAN dan FAHREZI. Berikut dibawah ini baris *script* yang digunakan dalam *data sharing* untuk memanggil *score* dan sisa saldo setiap *player* pada *game* simulasi merakit komputer.

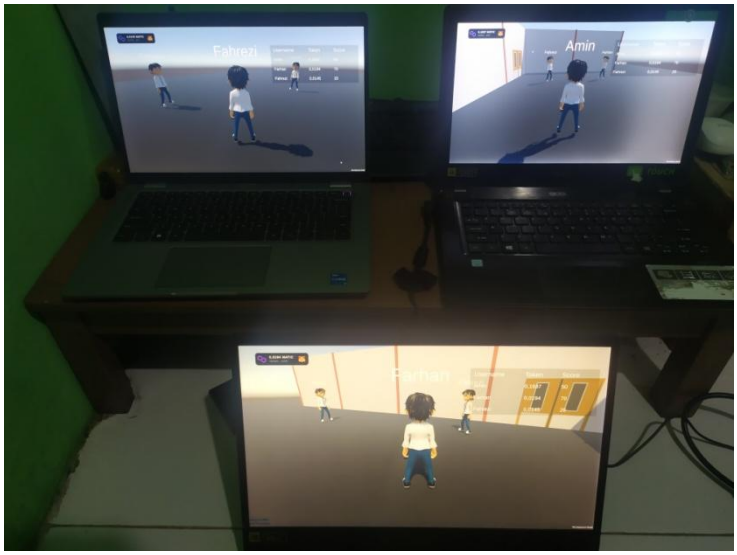
```
void UpdateStats()
     {

     if(player.CustomProperties.TryGetValue("token",
out object token))
          {
//menampilkan data saldo token
               tokenText.text = token.ToString();
          }

     if(player.CustomProperties.TryGetValue("score",
out object score))
          {
//menampilkan data score
               scoreText.text = score.ToString();
          }
     }
```

Berikut ini adalah data hasil pengujian dengan tiga perangkat yang telah disebutkan dalam sub-bab sebelumnya, serta pengujian yang telah dilakukan dengan pemain memasuki permainan. Kolom "Spawn" dianggap berhasil jika pemain berhasil masuk ke dalam arena permainan. Kolom "Tampilan" dianggap sesuai apabila *data sharing* saldo token, nama, dan skor berhasil dilakukan dalam permainan.

Tabel 0.11 Uji *Data sharing Nickname,* Sisa Saldo, dan *Score*

Perangkat	***Nickname***	***Spawn***	**Tampilan**
1	Amin	Berhasil	Sesuai
2	Farhan	Berhasil	Sesuai
3	Fahrezi	Berhasil	Sesuai

Dari Tabel 4.11, dapat disimpulkan bahwa hasil pengujian *data sharing untuk* saldo token dan skor pemain berhasil dilaksanakan. Semua pemain berhasil muncul dalam permainan, dan tampilan saldo dan skor antara pemain sesuai seperti yang mereka miliki masing-masing.



Gambar 0.28 Pengujian *Data sharing* Saldo dan *Score* dengan 3 Perangkat

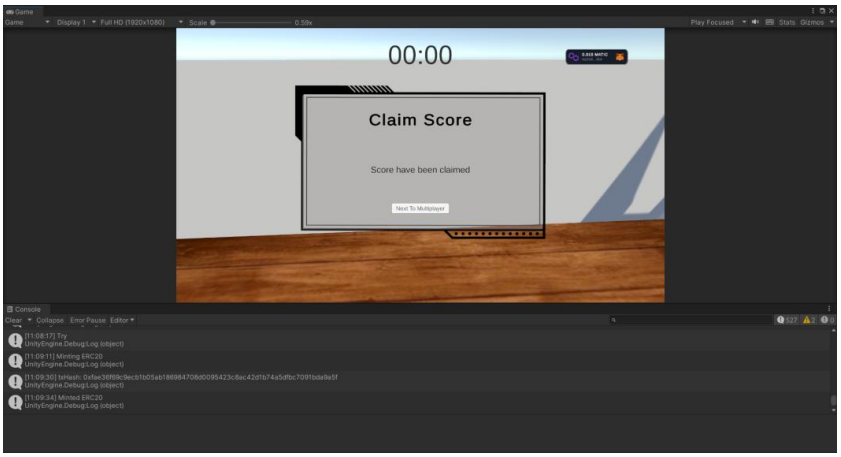
Adapun *data sharing* yang dilakukan kecil kemungkinannya terdapat manipulasi atau kecurangan data, sebab didalamnya terdapat penggunaan blockchain yang mana merupakan teknologi yang menerapkan sistem *peer-to-peer* dimana intergritas informasi digital dilindungi.

4.4.8 Pengujian Minting Smart Contract

Proses *minting smart contract* terjadi ketika *player* melakukan *claim score* pada *game* simulasi merakit komputer. Angka *score* yang didapat oleh *player* akan dikonversikan kedalam bentuk token *custom blockchain* menggunakan *smart contract Drop ERC-20*. Adapun keunggulan dari *smart contract* tersebut ialah tidak perlu melakukan *buying* atau pembelian untuk mendapatkan token. Token dapat diperoleh dengan mengatur *claim conditions* secara independen oleh pemilik *contract*. Sehingga apabila kondisi dari *smart contract* tersebut terpenuhi maka token akan didapatkan secara gratis tanpa harus melakukan pembelian. Penerapannya dalam penelitian ini ialah token custom tersebut akan dikategorikan

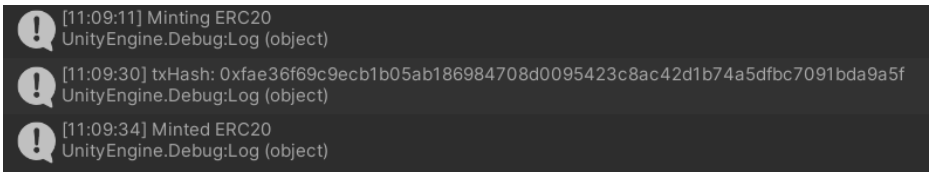
sebagai perolehan *score* yang dimiliki *player* ketika menyelesaikan permainan dan dapat dilakukan *data sharing* pada *menu multiplayer*. Berikut ini adalah baris *script* yang digunakan untuk melakukan *minting smart contract ERC-20*.

```
public async void MintERC20()
    {
        try
        {
            Debug.Log("Minting ERC20");

// Mendapatkan alamat dompet saat ini
            var address = await
ThirdwebManager.Instance.SDK.wallet.GetAddress();

// Mendapatkan kontrak smart contract menggunakan
alamat kontrak ERC20
            Contract contract =
ThirdwebManager.Instance.SDK.GetContract(DROP_ERC20_CON
TRACT);

// Menyembunyikan dan menampilkan elemen UI
            HasNotClaimedState.SetActive(false);
            ClaimingState.SetActive(true);

            var tx = await
contract.ERC20.ClaimTo(address,
scoreToClaim.ToString());
            var receipt = tx.receipt;
            Debug.Log("Minted ERC20");

// Memperbarui saldo token setelah klaim
            GetTokenBalance();

// Menyembunyikan elemen UI klaim dan menampilkan
elemen UI telah melakukan klaim
            ClaimingState.SetActive(false);
            HasClaimedState.SetActive(true);
        }
        catch(System.Exception)
        {
// Menangani kesalahan jika terjadi error
            Debug.Log("Error Minting ERC20");

        }
    }
```



Gambar 0.29 Pesan *Console Unity* ketika data terkirim

Apabila proses *minting* berhasil, maka di dalam Console Unity akan menampilkan kode hash dari proses *minting* tersebut. Seperti yang terlihat pada Gambar 4.29, dalam *Console Unity* terdapat pesan log seperti yang ditunjukkan pada Gambar 4.30.



Gambar 0.30 Kode *Hash*

Untuk melakukan implementasi ke dalam jaringan blockchain, diperlukan ABI (Application Binary Interface) dan Byte Code. ABI dan byte code digunakan untuk membuat dan berinteraksi dengan *smart contract* yang telah dibuat ketika transaksi dijalankan. ABI dan byte code diperoleh dari hasil kompilasi *smart contract* yang telah ditulis sebelumnya. Dalam konteks penelitian game simulasi merakit komputer, data yang dikirim ke jaringan blockchain adalah data skor, yang kemudian akan dikonversi menjadi token kustom menggunakan *smart contract* Drop ERC20. Berikut ini pada tabel 4.12 adalah *pseudocode* dari *smart*

contract Drop ERC20 yang digunakan. Kode tersebut kemudian di *compile* dan kemudian hasil dari *compile* code tersebut akan didapatkan *ABI code*.

Tabel 0.12 *Smart contract Drop ERC20*

Drop ERC20

```
contract DropContract {
    // Deklarasi variabel untuk menyimpan alamat pengirim dan token ERC-20 yang diterima
    address public owner;
    IERC20 public token;

    // Event yang akan dipancarkan ketika token di-drop
    event TokensDropped(address indexed sender, uint256 amount);

    // Konstruktor kontrak, dijalankan saat kontrak dibuat
    constructor(address _tokenAddress) {
        owner = msg.sender;
        token = IERC20(_tokenAddress);
    }

    // Fungsi untuk mengambil token yang di-drop ke kontrak
    function retrieveTokens() public {
        require(msg.sender == owner, "Hanya pemilik kontrak yang dapat mengambil token");
        uint256 balance = token.balanceOf(address(this));
        require(balance > 0, "Tidak ada token yang tersimpan di kontrak");
        require(token.transfer(owner, balance), "Gagal mengambil token");
    }

    // Fungsi ini memungkinkan pengguna untuk melemparkan token ERC-20 ke dalam kontrak
    function dropTokens(uint256 _amount) public {
        require(_amount > 0, "Jumlah token harus lebih dari 0");
        require(token.transferFrom(msg.sender, address(this), _amount), "Gagal menerima token");
        emit TokensDropped(msg.sender, _amount);
    }
}
```

Berikut adalah data pengujian untuk proses minting melalui tiga perangkat yang telah disebutkan dalam sub-bab sebelumnya, bersama dengan pengujian-pengujian sebelumnya. Tabel 4.13 mencatat hasil dari pengujian minting. Adapun hasil yang didapatkan adalah ketiga perangkat tersebut berhasil melakukan *minting smart contract*.

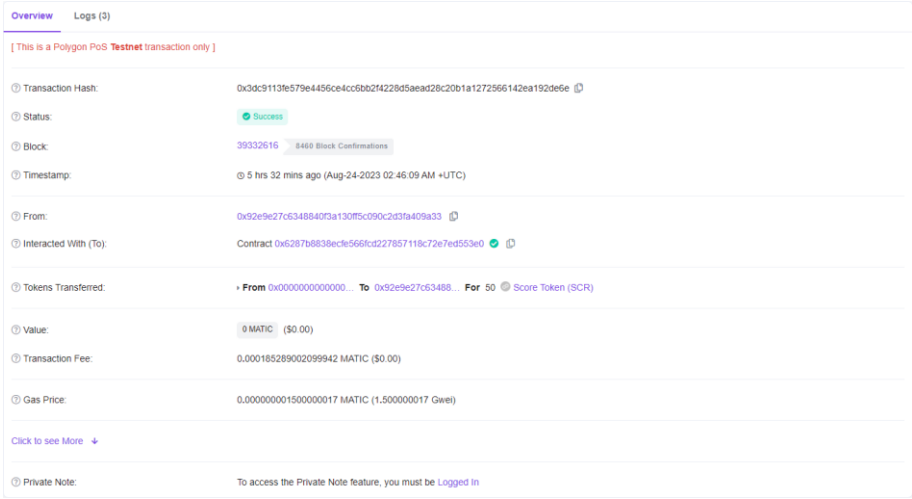
Tabel 0.13 Uji *Minting ERC20 Ethereum*

Perangakat	***Nickname***	***Minting***	***Hash***
1	Amin	Berhasil	0x9b96c138e5568d52c100568ffc4068e5ef179a97f288c2dcb663e2641e2eb5c7
2	Farhan	Berhasil	0xe8cfd9cbf15f35bbce688c6178cdb3bbad4c332a96a8bdca455e7fbb30ff5d98
3	Fahrezi	Berhasil	0xaa61ea7240ce15fc0e881b91f46cd2793e5c0ec3b6469b22fdce19530add7913

4.4.9 Pengujian *Hash*

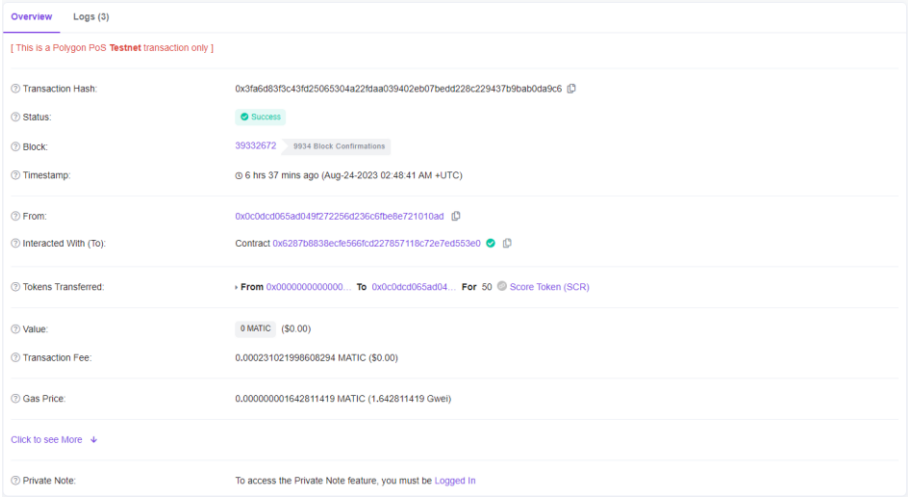
Proses pengujian melibatkan pengaksesan ke situs web https://mumbai.polygonscan.com/ dengan tujuan memvalidasi kelancaran implementasi *Ethereum* Blockchain dalam penelitian ini. Saat kode hash berhasil ditemukan dan terverifikasi di situs tersebut, maka transaksi pengiriman data pada jaringan *Ethereum* Blockchain dianggap berhasil. Hasil dari pengujian dapat ditemukan dalam ilustrasi yang terdokumentasikan pada Gambar 4.31, 4.32, serta 4.33

1. Perangkat 1 *Nickname* Amin



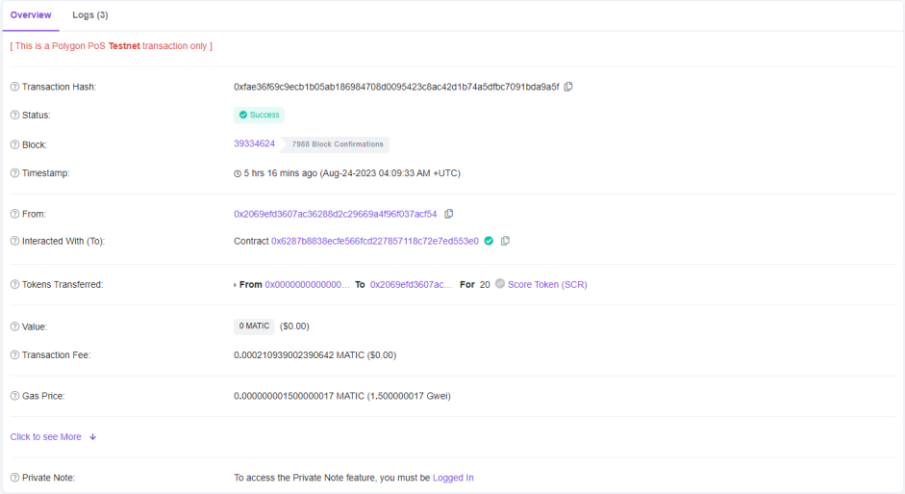
Gambar 0.31 Perangkat 1

2. Perangkat 2 *Nickname* Farhan



Gambar 0.32 Perangkat 2

3. Perangkat 3 *Nickname* Fahrezi



Gambar 0.33 Perangkat 3

Dalam semua pengujian yang telah dilakukan, dapat diamati bahwa semua kode hash yang dihasilkan dari proses minting oleh tiga akun yang berbeda berhasil teridentifikasi pada situs pengecekan transaksi blockchain https://mumbai.polygonscan.com/. Gambar tersebut menunjukkan bahwa nilai transaksi adalah 0 Matic (0$). Proses pengiriman data skor pada jaringan *Ethereum* Blockchain dilakukan oleh tiga pemain dengan akun yang berbeda, yang mengkonfirmasi keberhasilan implementasi dalam penelitian ini.

4.4.10 Pengujian *Gas Limit*

Pengujian ini dilakukan guna mengetahui berapa minimum jumlah gas yang dibutuhkan untuk melakukan suatu transaksi pada jaringan *blockchain* dalam *game* simulasi merakit komputer. Ketersediaan jumlah gas yang cukup akan sangat mempengaruhi keberhasilan dari proses transaksi pada jaringan *blockchain* (Pradana et al., 2022). Dalam Tabel 4.14, terlihat bahwa batasan gas (gas limit)

yang berhasil adalah sebesar 400.000, yang memungkinkan transaksi yang efisien dengan gas limit sesedikit mungkin. Kolom yang menunjukkan "berhasil" mengindikasikan bahwa transaksi telah sukses, gas limit sesuai, dan telah diverifikasi di situs https://mumbai.polygonscan.com/. Sementara kolom yang menunjukkan "gagal" mengindikasikan bahwa gas limit tidak sesuai, dan situs https://mumbai.polygonscan.com/ menunjukkan bahwa transaksi tersebut gagal diverifikasi.

Tabel 0.14 Pengujian minimum gas *limit*

Pengujian	**Gas *Limit***			
	100.000	**200.000**	**300.000**	**400.000**
1	Gagal	Berhasil	Berhasil	Berhasil
2	Gagal	Berhasil	Berhasil	Berhasil
3	Gagal	Berhasil	Berhasil	Berhasil
4	Gagal	Berhasil	Berhasil	Berhasil
5	Gagal	Berhasil	Berhasil	Berhasil
6	Gagal	Berhasil	Berhasil	Berhasil
7	Gagal	Berhasil	Berhasil	Berhasil
8	Gagal	Berhasil	Berhasil	Berhasil
9	Gagal	Berhasil	Berhasil	Berhasil
10	Gagal	Berhasil	Berhasil	Berhasil

Seperti dalam tabel 4.14 dapat diketahui bahwasanya gas *limit* yang digunakan dalam penelitian *game* simulasi merakit komputer untuk *minimum cost* dan tingkat keberhasilan pengiriman transaksi ke dalam jaringan *blockchain* berada di atas gas *limit* 100.000. Berdasarkan percobaan tersebut terdapat *range* antara 100.000 - 200.000 gas *limit* yang belum diketahui secara detail berapa minimum gas *limit* yang dapat diterapkan. Maka dari itu akan dilakukan percobaan kembali dengan selisih 5.000 gas *limit* antara 100.000 - 200.000 seperti pada tabel 4.16.

Tabel 0.15 Pengujian secara detail minimum gas *limit*

Pengujian Gas *Limit*	**Hasil**
105.000	Gagal
110.000	Gagal
115.000	Gagal
120.000	Gagal
125.000	Berhasil
130.000	Berhasil
135.000	Berhasil
140.000	Berhasil
145.000	Berhasil
150.000	Berhasil
155.000	Berhasil
160.000	Berhasil
165.000	Berhasil
170.000	Berhasil
175.000	Berhasil
180.000	Berhasil
185.000	Berhasil
190.000	Berhasil
195.000	Berhasil

Dengan hasil pengujian seperti pada tabel 4.15 maka standar minimum gas *limit* yang dapat diterapkan adalah 125.000.

4.4.11 Uji Coba

Pengujian ini bertujuan untuk mengevaluasi kecepatan konfirmasi transaksi oleh situs pengecekan blockchain, yaitu https://mumbai.polygonscan.com/. Pengujian melibatkan perubahan nilai variabel gas limit yang telah diuji sebelumnya. Selanjutnya, sejumlah percobaan transaksi dilakukan, dan waktu yang dibutuhkan untuk memproses setiap transaksi diukur. Rata-rata waktu yang diperlukan untuk memproses sebuah transaksi dapat dihitung menggunakan rumus yang disebutkan pada nomor 4.1.

$$\bar{T} = \frac{1}{n} \sum_{i=1}^{n} Ti \tag{4.1}$$

Keterangan :

$\bar{T}$ = rata – rata waktu transaksi
n = Banyak percobaan
T_i = Waktu yang diperlukan dalam sebuah proses transaksi
i = Transaksi

Gas *limit* (satuan gas yang digunakan) adalah jumlah maksimum unit gas yang bersedia dibayarkan untuk melakukan transaksi atau pengoperasian EVM. Operasi yang berbeda memerlukan jumlah unit gas yang berbeda pula.

Setiap blok di jaringan *Ethereum* memiliki *base fee* (biaya dasar) yang ditentukan oleh permintaan jaringan: *base fee* didasarkan pada ukuran blok dari blok sebelumnya, dibandingkan dengan ukuran blok target (di mana ukuran mengacu pada jumlah total gas yang digunakan untuk seluruh blok). Total *gas fee* harus memenuhi harga minimum ini agar dapat dipertimbangkan untuk dimasukkan dalam blok tersebut.

Kemudian terdapat pula *priority fee* atau disebut juga "*miner tip*", yaitu memberi insentif kepada penambang untuk memprioritaskan transaksi kita. Sehingga *max fee* adalah jumlah total global yang dibayarkan untuk transaksi Anda. Untuk rumus yang digunakan dalam menghitung *max fee* adalah seperti pada nomor 4.2 :

$$(base\ fee + \ priority\ fee)\ x\ gas\ limit \tag{4.2}$$

Keterangan :

base fee = Dihasilkan oleh protokol. Merupakan pengganda '*gasUsed*'

minimum yang diperlukan agar agar suatu transaksi dapat diselesaikan.

priority fee = Jumlah intensif yang diberikan kepada penambang untuk memprioritaskan transaksi kita.

gas limit = jumlah maksimum unit gas yang bersedia dibayarkan untuk melakukan transaksi atau pengoperasian EVM.

Untuk mengetahui bagaimana mekanisme gas *limit* mempengaruhi kecepatan transaksi, maka akan dilakukan pengujian melalui beberapa skenario yang di telah disiapkan dengan acuan parameter tertentu seperti pada tabel 4.16 sebagai berikut:

Tabel 0.16 Skenario Pengujian kecepatan Transaksi

Skenario	**Banyak Pengujian**	***Range* Gas *Limit***	**Data Pengujian**	**Jaringan**
1.	20 kali	200.000 – 700.000	Score 100	Data Seluler
2.	20 kali	200.000 – 700.000	Score 100	Wifi UIN
3.	20 kali	200.000 – 700.000	Score 100	Wifi Upojiwo

Sebelum melakukan pengujian pada tiap skenario diperlukan penyesuaian terlebih dahulu pada pengaturan gas ketika akan melakukan transaksi. Pada notifikasi *metamask* dengan menekan tombol seperti pada gambar 4.34 ini akan memungkinkan kita untuk menyesuaikan parameter gas secara langsung.

Gambar 0.34 Pengaturan Gas *Limit*

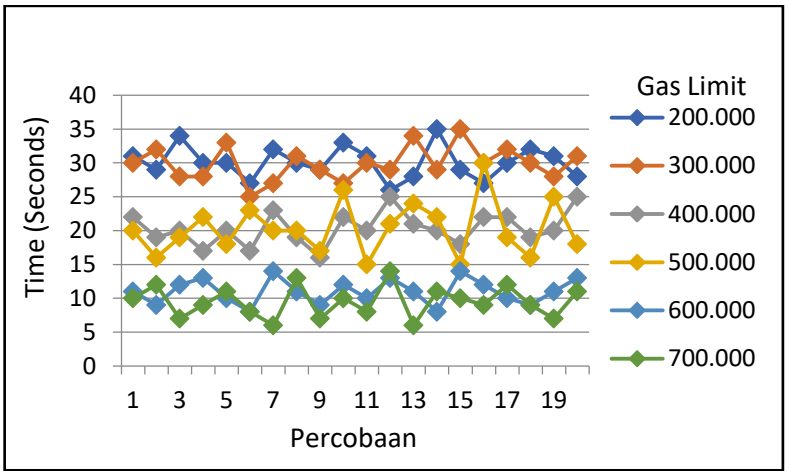
Pada pengujian skenario pertama dilakukan uji coba sebanyak 20 kali dengan *range* gas *limit* dari 200.000 – 700.000, kemudian data pengujian diseragamkan dengan *score* 100 dan menggunakan jaringan data seluler (xl *provider*). Berikut ini adalah hasil dari pengujian yang tertera pada tabel 4.17 gambar 4.35 dengan kecepatan jaringan seperti pada gambar 4.36.

Tabel 0.17 Uji Kecepatan transaksi menggunakan Data Seluler

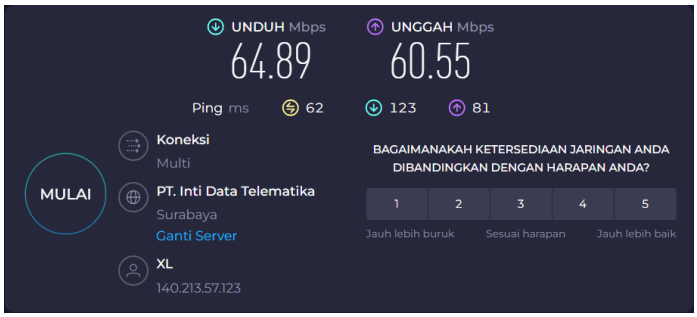
Skenario 1						
Pengujian	**Gas *Limit***					
	200.000	**300.000**	**400.000**	**500.000**	**600.000**	**700.000**
1	31	30	22	20	11	10
2	29	32	19	16	9	12
3	34	28	20	19	12	7
4	30	28	17	22	13	9
5	30	33	20	18	10	11
6	27	25	17	23	8	8
7	32	27	23	20	14	6
8	30	31	19	20	11	13
9	29	29	16	17	9	7
10	33	27	22	26	12	10

Lanjutan Tabel 4.17

11	31	30	20	15	10	8
12	26	29	25	21	13	14
13	28	34	21	24	11	6
14	35	29	20	22	8	11
15	29	35	18	15	14	10
16	27	30	22	30	12	9
17	30	32	22	19	10	12
18	32	30	19	16	9	9
19	31	28	20	25	11	7
20	28	31	25	18	13	11
Rata-rata	30,1	29,9	20,35	20,3	11	9,5



Gambar 0.35 Grafik Kecepatan transaksi menggunakan Data Seluler



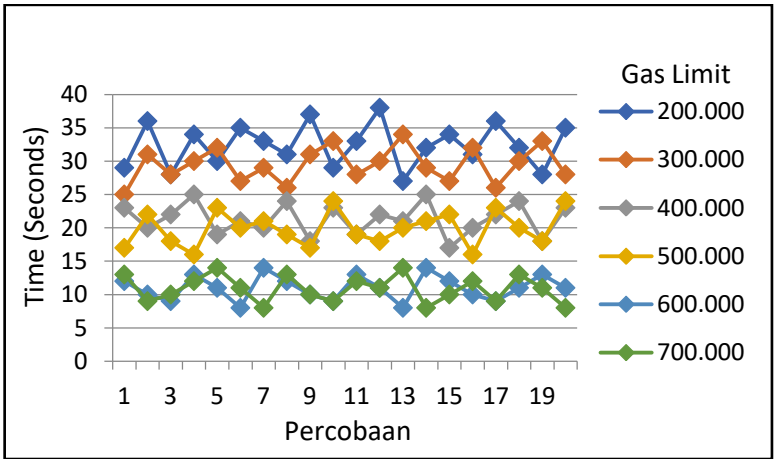
Gambar 0.36 Kecepatan Jaringan Data Seluler (XL)

Kemudian pada pengujian skenario kedua dilakukan uji coba sebanyak 20 kali dengan *range* gas *limit (*GWEI) dari 200.000 – 700.000, kemudian data pengujian diseragamkan dengan *score* 100 dan menggunakan jaringan wifi UIN.

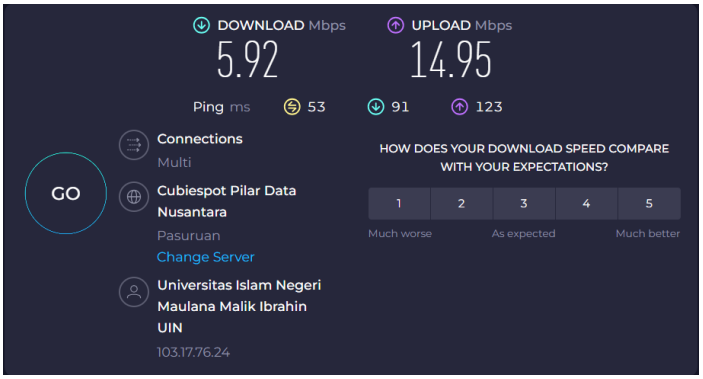
Berikut ini adalah hasil dari pengujian yang tertera pada tabel 4.18 dan gambar 4.37 dengan kecepatan jaringan seperti pada gambar 4.38.

Tabel 0.18 Uji Kecepatan transaksi menggunakan Wifi UIN

Skenario 2						
	Gas *Limit* (GWEI)					
Pengujian	**200.000**	**300.000**	**400.000**	**500.000**	**600.000**	**700.000**
1	29	25	23	17	12	13
2	36	31	20	22	10	9
3	28	28	22	18	9	10
4	34	30	25	16	13	12
5	30	32	19	23	11	14
6	35	27	21	20	8	11
7	33	29	20	21	14	8
8	31	26	24	19	12	13
9	37	31	18	17	10	10
10	29	33	23	24	9	9
11	33	28	19	19	13	12
12	38	30	22	18	11	11
13	27	34	21	20	8	14
14	32	29	25	21	14	8
15	34	27	17	22	12	10
16	31	32	20	16	10	12
17	36	26	22	23	9	9
18	32	30	24	20	11	13
19	28	33	18	18	13	11
20	35	28	23	24	11	8
Rata-rata	32,4	29,45	21,3	19,9	11	10,85



Gambar 0.37 Grafik Kecepatan transaksi menggunakan Wifi UIN



Gambar 0.38 Kecepatan Jaringan Wifi UIN

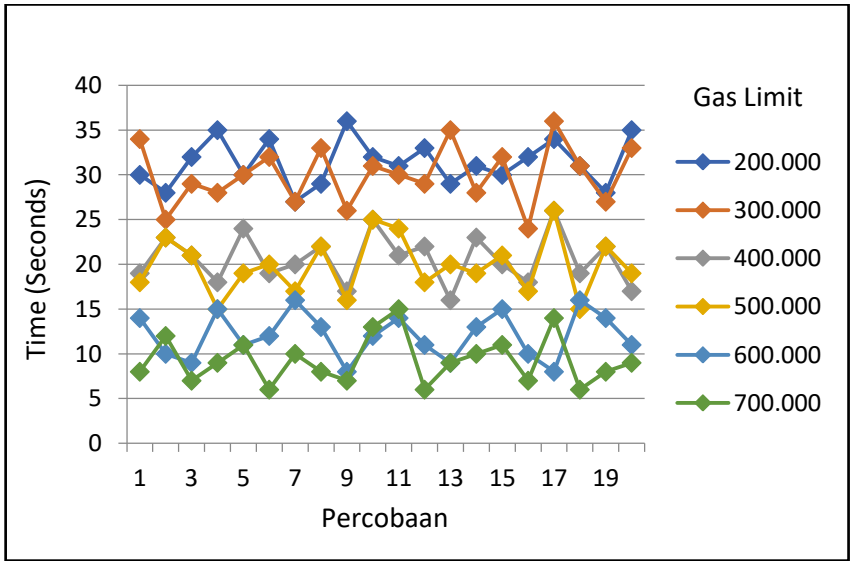
Selanjutnya pada pengujian skenario terakhir dilakukan uji coba sebanyak 20 kali dengan *range* gas *limit (*GWEI) dari 200.000 – 700.000, kemudian data pengujian diseragamkan dengan *score* 100 dan menggunakan jaringan wifi cafe upojiwo. Berikut ini adalah hasil dari pengujian yang tertera pada tabel 4.19 gambar 4.39 dengan kecepatan jaringan seperti pada gambar 4.40.

Tabel 0.19 Uji Kecepatan transaksi menggunakan Wifi Cafe Upojiwo

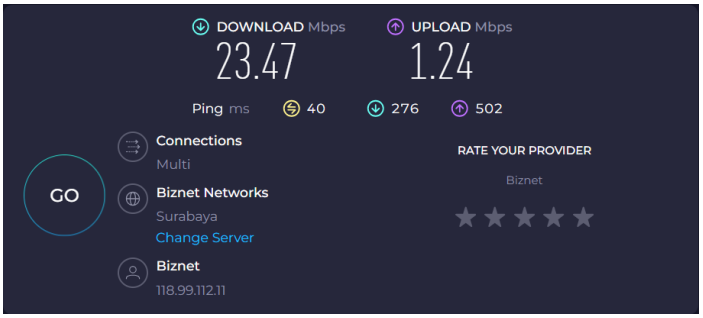
Skenario 3						
	Gas *Limit* (GWEI)					
Pengujian	**200.000**	**300.000**	**400.000**	**500.000**	**600.000**	**700.000**
1	30	34	19	18	14	8
2	28	25	23	23	10	12
3	32	29	21	21	9	7
4	35	28	18	15	15	9
5	30	30	24	19	11	11
6	34	32	19	20	12	6
7	27	27	20	17	16	10
8	29	33	22	22	13	8
9	36	26	17	16	8	7
10	32	31	25	25	12	13
11	31	30	21	24	14	15
12	33	29	22	18	11	6
13	29	35	16	20	9	9
14	31	28	23	19	13	10
15	30	32	20	21	15	11
16	32	24	18	17	10	7
17	34	36	26	26	8	14
18	31	31	19	15	16	6
19	28	27	22	22	14	8

Lanjutan Tabel 4.19

20	35	33	17	19	11	9
Rata-rata	31,35	30	20,6	19,85	12,05	9,3



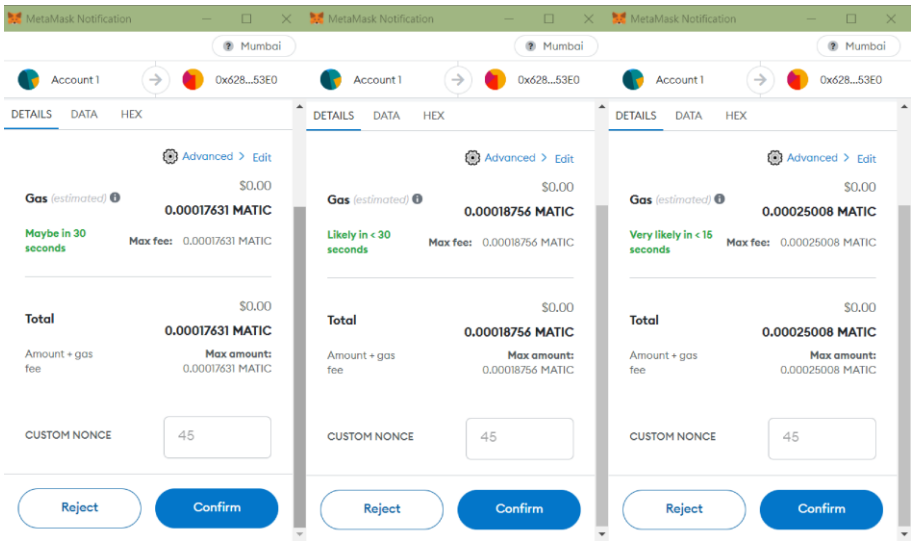
Gambar 0.39 Grafik Kecepatan transaksi menggunakan Wifi Cafe Upojiwo



Gambar 0.40 Kecepatan Jaringan Wifi Cafe Upojiwo

Setelah melakukan pengujian kecepatan transaksi berdasarkan 3 skenario yang telah dirancangkan menunjukkan perbandingan waktu berdasarkan batas gas *limit* yang terapkan. Terlihat bahwa dengan menggunakan gas *limit* 200.000 – 300.000 kecepatan transaksi yang dihasilkan berada di kisaran 30 detik, kemudian dengan menggunakan gas *limit* 400.000 – 500.000 kecepatan transaksi yang dihasilkan berada di kisaran 30 - 15 detik, sedangkan dengan menggunakan gas

limit 600.000 – 700.000 waktu yang didapatkan berada dibawah 15 detik. Adapun hal tersebut sesuai dengan estimasi perkiraan kecepatan transaksi yang diberikan *metamask* seperti pada Gambar 4.41.



Gambar 0.41 Estimasi Kecepatan *Metamask*

4.5 Evaluasi

Dari seluruh pengujian dan percobaan yang telah dilakukan, termasuk uji koneksi, pembuatan ruangan (create room), bergabung ke ruangan (join room), mode multiplayer, *data sharing* nama dan saldo, klaim skor, deploy kontrak *Ethereum* untuk skor, berbagi data skor, Pengujian kecepatan transaksi, serta uji gas limit, hasilnya dapat dilihat dalam Tabel 4.20.

Tabel 0.20 Evaluasi pengujian

No	Skenario Pengujian	Hasil Pengujian
1	Dalam Console Unity akan memunculkan tulisan "Koneksi berhasil" dan juga menampilkan Token Hex Code sesuai *addres* pemain.	*Unity* terhubung dengan *ethereum* secara baik dan berhasil dilakukan.
2	Ketika Pemain berhasil mengkoneksikan akun *blockchain*nnya, maka saldo token pemain akan ditampilkan	Seluruh pemain berhasil mengkoneksikan akun *blockchain* nnya

Lanjutan Tabel 4.20

3	Pemain bisa mengetahui saldo token dompet mereka dalam *game*	Seluruh *player* berhasil mengetahui saldo token yang mereka miliki
4	*Score* tercatat sebagai transaksi valid dalam jaringan *Ethereum Blockchain*	Pada situs pengecekan *hash* polygonscan kode *hash* berhasil dikonfirmasi.
5	Setiap pemain yang terhubung dapat melihat hasil *score*, saldo dan nama masing-masing	Seluruh pemain dapat mengetahui kepemilikan *score*, saldo dan nama setiap pemain

Tabel 4.20 menunjukkan bahwa kelima pengujian utama telah berhasil dilaksanakan sesuai dengan harapan peneliti dalam konteks penelitian *data sharing* skor dalam game simulasi merakit komputer menggunakan *Ethereum* Blockchain.

Adapun pembahasan terkait konteks kecepatan dalam islam dapat dikatakan sedikit yang menjelaskan secara spesifik tentang apa yang disebut konsep kecepatan. Meskipun begitu, terdapat hadits yang dapat dihubungkan dengan konsep kecepatan dalam pengertian secara umum yaitu perjalanan. Seperti yang dijelaskan Hadits Ibnu Majah nomor 3008:

حَدَّثَنَا عَلِيُّ بْنُ مُحَمَّدٍ وَعَمْرُو بْنُ عَبْدِ اللَّهِ قَالَا حَدَّثَنَا وَكِيعٌ حَدَّثَنَا هِشَامُ بْنُ عُرْوَةَ عَنْ أَبِيهِ عَنْ أُسَامَةَ بْنِ زَيْدٍ أَنَّهُ سُئِلَ كَيْفَ كَانَ رَسُولُ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ يَسِيرُ حِينَ دَفَعَ مِنْ عَرَفَةَ قَالَ كَانَ يَسِيرُ الْعَنَقَ فَإِذَا وَجَدَ فَجْوَةً نَصَّ قَالَ وَكِيعٌ وَالنَّصُّ يَعْنِي فَوْقَ الْعَنَقِ

"Telah menceritakan kepada kami [Ali bin Muhammad] dan [Amru bin Abdullah], keduanya berkata; telah menceritakan kepada kami [Waki']; telah menceritakan kepada kami [Hisyam bin Urwah] dari [Bapaknya] dari [Usamah bin Zaid], bahwa ia pernah ditanya: 'Bagaimana Rasulullah shallallahu 'alaihi wasallam berjalan ketika berangkat dari Arafah? ' Usamah menjawab; 'Beliau berjalan dengan kecepatan sedang. Dan jika beliau menemukan kekosongan (di jalan), maka beliau akan mempercepat jalannya.' Waki' berkata; 'Yang dimaksud

adalah kecepatan yang melebihi kecepatan biasa (rata-rata).'"(HR. Ibnu Majah, No. 887).

Hadits tersebut menjelaskan bagaimana Rasulullah shallallahu 'alaihi wasallam menyesuaikan kecepatan berjalannya sesuai dengan kondisi dan juga prioritasnya. Rasulullah shallallahu 'alaihi wasallam berjalan dengan kecepatan sedang dan jika beliau menemukan kelenggangan di jalan, maka beliau akan mempercepat jalannya. Hal tersebut dapat kita implementasikan pada saat memilih gas *limit* yang ingin digunakan guna mengatur kecepatan transaksi dalam jaringan *blockchain*.

Apabila pengguna mempunyai keleluasaan dana yang berlebih maupun cukup disertai kebutuhan yang mendesak sehingga diharuskan melakukan transaksi secara cepat, maka pengguna dapat memilih untuk membayarkan gas dengan harga yang lebih tinggi agar mendapatkan transaksi yang jauh lebih cepat. Namun jika pengguna merasa tidak terlalu membutuhkan kecepatan transaksi yang tinggi, maka pengguna dapat memilih untuk membayarkan gas dengan harga yang sedang atau normal.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Adapun penelitian yang dilakukan merupakan penelitian penerapan *ethereum blockchain* untuk sistem *data sharing* pada game simulasi perakitan komputer berbasis *virtual reality*. Pada penelitian ini berdasarkan hasil pengujian yang telah dilakukan sebelumnya yaitu dengan pengujian kecepatan transaksi melalui 3 skenario yang berbeda didapatkan kesimpulan bahwa perbedaan Gas *limit* yang digunakan dalam sebuah transaksi pada jaringan blockchain akan sangat mempengaruhi kecepatan proses validasi transaksi.

Hasil dari pengujian menggunakan 3 jaringan yang berbeda dengan dilakukan uji coba sebanyak 20 kali dengan range gas *limit* 200.000 – 700.000 dan data score 100 menghasilkan kecepatan transaksi yang berbeda-beda. Adapun dengan menggunakan gas *limit* 200.000 - 300.000 kecepatan transaksi yang dihasilkan berada di kisaran 30 detik, kemudian dengan menggunakan gas *limit* 400.000 - 500.000 kecepatan transaksi yang dihasilkan berada di kisaran 30 - 15 detik, sedangkan dengan menggunakan gas *limit* 600.000 – 700.000 waktu yang didapatkan berada dibawah 15 detik.

Maka dari itu, pengguna dapat menyesuaikan gas yang ingin dibayarkan sesuai dengan kebutuhan dan juga prioritasnya. Pemrosesan transaksi akan jauh lebih cepat dengan membayarkan gas yang lebih banyak seperti menggunakan gas *limit* dengan *range* 600.000 - 700.000, sebaliknya pengguna dapat membayarkan lebih sedikit gas seperti menggunakan gas *limit* dengan *range* 200.000 - 300.000

dengan konsekuensi besar kemungkinannya untuk menunggu proses validasi transaksi yang lebih lama.

5.2 Saran

Pembangunan sistem *data sharing* dalam game simulasi perakitan komputer berbasis virtual reality dengan memanfaatkan teknologi blockchain *Ethereum* masih memiliki ruang untuk peningkatan yang signifikan. Terdapat beberapa aspek yang memerlukan perbaikan guna mendukung penelitian masa depan dan untuk meningkatkan kualitas sistem dan permainan tersebut. Dalam hal ini, beberapa saran yang dapat diberikan oleh penulis adalah sebagai berikut:

1. Melakukan pengembangan pada tampilan UI UX dan juga *environment* pada mode multiplayernya sehingga *game* akan terlihat menjadi lebih menarik sehingga alur permainan menjadi lebih mudah untuk dipahami.
2. Menambahkan model karakter sehingga semua *player* dapat menggunakan karakter yang lebih bervariatif dan berbeda.
3. Menambahkan fitur untuk saling dapat bertransaksi koin *eth* antar sesama player.
4. Menambahkan model game pembelajaran yang lainnya seperti pembelajaran robotik atau pun jaringan, dll.

DAFTAR PUSTAKA

Antonopoulos, A. M. (2014). Mastering Bitcoin: unlocking digital cryptocurrencies. In *Journal of World Trade* (Vol. 50, Issue 4).

Arif, Y. M., Firdaus, M. N., & Nurhayati, H. (2021). A Scoring System for Multiplayer Game Base on Blockchain Technology. *Proceedings - 2021 IEEE Asia Pacific Conference on Wireless and Mobile, APWiMob 2021*, 200–205. https://doi.org/10.1109/APWiMob51111.2021.9435249

Arif, Y. M., Nurhayati, H., Kurniawan, F., Nugroho, S. M. S., & Hariadi, M. (2020). Blockchain-Based Data Sharing for Decentralized Tourism Destinations Recommendation System. *International Journal of Intelligent Engineering and Systems*, *13*(6), 472–486. https://doi.org/10.22266/ijies2020.1231.42

Arif, Y. M., Pradana, R. P., Nurhayati, H., Nugroho, S. M. S., & Hariadi, M. (2020). A Blockchain-Based Multiplayer Transaction for Tourism Serious Game. *CENIM 2020 - Proceeding: International Conference on Computer Engineering, Network, and Intelligent Multimedia 2020*, 138–143. https://doi.org/10.1109/CENIM51130.2020.9297837

Bahga, A., & Madisetti, V. K. (2016). Blockchain Platform for Industrial Internet of Things. *Journal of Software Engineering and Applications*, *09*(10), 533–546. https://doi.org/10.4236/jsea.2016.910036

Buterin, V. (2014). A next-generation smart contract and decentralized application platform. *Etherum*, *January*, 1–36. http://buyxpr.com/build/pdfs/EthereumWhitePaper.pdf

Carson, B., Romanelli, G., Walsh, P., & Zhumaev, A. (2018). Blockchain beyond the hype: What is the strategic business value? | McKinsey&Company. *Digital McKinsey*, *June*, p.1-13. https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/blockchain-beyond-the-hype-what-is-the-strategic-business-value

Darmawan, A. M. (2022). *Sistem data sharing untuk rating dan preference user menggunakan ethereum blockchain pada game tempat wisata di Malang*. Diss. Universitas Islam Negeri Maulana Malik Ibrahim.

Faisal, M., Nurhayati, H., Arif, Y. M., Kurniawan, F., & Nugroho, F. (2016). Immersive bicycle game for health virtual tour of uin maulana malik ibrahim Malang. *Jurnal Teknologi*, *78*(5), 325–328. https://doi.org/10.11113/jt.v78.8330

Firdaus, M. N. (2020). *Sistem Data Sharing Untuk Score Pada Game Widow Waterfall Menggunakan Ethereum Blockchain*. Diss. Universitas Islam Negeri Maulana Malik Ibrahim.

Firu, A. C., Tapîrdea, A. I., Feier, A. I., & Draghici, G. (2020). Virtual reality in the automotive field in industry 4.0. *Materials Today: Proceedings*, *45*(xxxx), 4177–4182. https://doi.org/10.1016/j.matpr.2020.12.037

Jani, S. (2017). An Overview of Ethereum & Its Comparison with Bitcoin. *International Journal of Scientific & Engineering Research*, *10*(8).

Majah, I. (887). *Sunan Ibnu Majah*.

Nopendri, N., Aurachman, R., & Amrullah, M. R. (2020). Rancangan Simulasi Penerapan Blockchain dalam Pemilihan Presiden Indonesia. *Jurnal Rekayasa Sistem & Industri (JRSI)*, *7*(1), 10. https://jrsi.sie.telkomuniversity.ac.id/JRSI/article/view/373

Nurhayati, H., & Arif, Y. M. (2023). Math-VR: Mathematics Serious Game for Madrasah Students using Combination of Virtual Reality and Ambient Intelligence. *International Journal of Advanced Computer Science and Applications*, *14*(5), 233–239. https://doi.org/10.14569/IJACSA.2023.0140524

Pradana, R. P. (2020). *Sistem transaksi antar player pada game multiplayer wisata Bromo menggunakan blockchain*. Universitas Islam Negeri Maulana Malik Ibrahim.

Pradana, R. P., Hariadi, M., Rachmadi, R. F., & Arif, Y. M. (2022). A Multi-Criteria Recommender System For NFT Based IAP In RPG Game. *2022 International Seminar on Intelligent Technology and Its Applications: Advanced Innovations of Electrical Systems for Humanity, ISITIA 2022 - Proceeding*, 214–219. https://doi.org/10.1109/ISITIA56226.2022.9855272

Rahardja, U., Aini, Q., Yusup, M., & Edliyanti, A. (2020). Penerapan Teknologi Blockchain Sebagai Media Pengamanan Proses Transaksi E-Commerce. *CESS (Journal of Computer Engineering, System and Science)*, *5*(1), 28. https://doi.org/10.24114/cess.v5i1.14893

Rochmatullah, M. F. (2022). *Smart Contract Dengan Ethereum Sebagai Dasar*. Politeknik Keuangan Negara STAN.

Segura, R. J., del Pino, F. J., Ogáyar, C. J., & Rueda, A. J. (2020). VR-OCKS: A virtual reality game for learning the basic concepts of programming. *Computer Applications in Engineering Education*, *28*(1), 31–41. https://doi.org/10.1002/cae.22172

Utomo, A. W. B., Bhawiyuga, A., & Amron. (2019). Pengembangan Sistem Penyimpanan Data Sensor Menggunakan Platform Hyperledger. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *3*(5), 5154–5164.

Zheng, Y. (2019). Utility-Based Multi-Criteria Recommender Systems. *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, 2529–2531. https://doi.org/10.1145/3297280.3297641