

**SISTEM TRANSAKSI ANTAR PLAYER PADA
GAME MULTIPLAYER WISATA BROMO
MENGUNAKAN BLOCKCHAIN**

SKRIPSI

Oleh :
REZA PUTRA PRADANA
NIM. 16650069



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2020**

**SISTEM TRANSAKSI ANTAR PLAYER PADA
GAME MULTIPLAYER WISATA BROMO
MENGUNAKAN BLOCKCHAIN**

SKRIPSI

HALAMAN PE
NGAJU
AN

**Diajukan kepada:
Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan Dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)**

**Oleh:
REZA PUTRA PRADANA
NIM. 16650069**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2020**

LEMBAR PERSETUJUAN
SISTEM TRANSAKSI ANTAR PLAYER PADA
GAME MULTIPLAYER WISATA BROMO
MENGGUNAKAN BLOCKCHAIN

SKRIPSI

Oleh :
REZA PUTRA PRADANA
NIM. 16650069

Telah Diperiksa dan Disetujui untuk Diuji
Tanggal : 02 Juni 2020

Dosen Pembimbing I

Yunifa Miftachul Arif, M. T
NIP. 19830616 201101 1 004

Dosen Pembimbing II

Hani Nurhayati, M.T
NIP. 197806252008012006

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

Dr. Cahyo Crysdiان
NIP. 19740424 200901 1 008

LEMBAR PENGESAHAN
SISTEM TRANSAKSI ANTAR PLAYER PADA
GAME MULTIPLAYER WISATA BROMO
MENGGUNAKAN BLOCKCHAIN

SKRIPSI

Oleh:
Reza Putra Pradana
NIM. 16650069

Telah Dipertahankan di Depan Penguji Skripsi dan Dinyatakan Diterima Sebagai
Salah Satu Persyaratan Untuk Memperoleh Gelar Sarjana Komputer (S. Kom)
Tanggal 2020

Susunan Dewan Penguji

		Tanda Tangan
1. Penguji Utama	: <u>Dr. M. Amin Hariyadi, M.T</u> NIP. 196701182005011001	()
2. Ketua	: <u>Fajar Rohman Hariri, M. Kom</u> NIP. 198905152018011001	()
3. Sekretaris	: <u>Yunifa Miftachul Arif, M. T</u> NIP. 19830616 201101 1 004	()
4. Anggota	: <u>Hani Nurhayati, M.T</u> NIP. 197806252008012006	()

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

Dr. Cahyo Crysdiان
NIP. 19740424 200901 1 008

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Reza Putra Pradana
NIM : 16650069
Fakultas/ Jurusan : Sains dan Teknologi/ Teknik Infomatika
Judul Skripsi : Sistem Transaksi Antar Player Pada Game Multiplayer
Wisata Bromo Menggunakan Blockchain

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya sendiri, bukan merupakan pengambilalihan 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 hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 18 Juni 2020

Yang membuat pernyataan,


Keza Putra Pradana
NIM. 16650069

HALAMAN MOTTO

“Janganlah takut gagal dalam berproses, karena setiap kegagalan akan selalu ada hikmah yang secara langsung atau tidak langsung akan kita rasakan manfaatnya”



HALAMAN PERSEMBAHAN

الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ

Puji syukur kehadiran Allah, shalawat dan salam bagi Rasul-Nya

Penulis persembahkan sebuah karya ini kepada:

Kedua orang tua penulis tercinta, Bapak Saifullah dan Ibu Uun Rindiyati yang selalu memberikan suntikan motivasi yang tak terhingga.

Dosen pembimbing penulis Bapak Yunifa Miftachul Arif, M. T dan Ibu Hani Nurhayati, M.T yang telah dengan sabar membimbing jalannya penelitian skripsi ini dan selalu memberikan stimulus positif untuk tetap semangat menjalani setiap tahap ujian skripsi.

Seluruh dosen Teknik Informatika UIN Maulana Malik Ibrahim Malang, dan seluruh guru-guru penulis yang telah membimbing dan memberikan ilmunya yang sangat bermanfaat.

Teman-teman organisasi selama di UIN Maulana Malik Ibrahim Malang yang telah mengajarkanku berbagai hal untuk terus belajar semakin baik.

Keluarga Teknik Informatika 2016 “Andromeda” yang telah memberikan semangat dan doanya

Orang-orang yang penulis sayangi, yang tak bisa penulis sebutkan satu per satu yang selalu memberikan semangat dan motivasinya kepada penulis untuk menyelesaikan skripsi ini.

Penulis ucapkan terima kasih yang luar biasa. Semoga *ukhwah* kita tetap terjaga dan selalu mendapat *ridho* Allah SWT. *Allahumma Aamiin.*

KATA PENGANTAR

Assalamu'alaikum Warohmatullaahi Wabarakaatuh

Syukur nikmat terpanjatkan kepada *Ilahi Rabbi*, satu-satunya zat yang selalu memberikan kekuatan kepada penulis untuk dapat menyelesaikan skripsi dengan keadaan yang *insyaAllah* lancar dan baik. Sholawat sekaligus salam penghormatan kepada junjungan nabi *akhirruzzaman*, nabi pembawa kabar kemenangan untuk seluruh umat manusia tanpa terkecuali. Siapapun pengikutnya menantikan syafaatnya kelak di hari akhir. Semoga kita termasuk golongan yang dituntun Allah SWT dan mendapat pertolongan Nabi Muhamad SAW. *Aamiin*.

Selama proses pengerjaan skripsi, penulis mendapatkan banyak bantuan dan dukungan dari berbagai pihak. Maka dari itu, ucapan syukur dan terima kasih penulis sampaikan kepada:

1. Yunifa Miftachul Arif, M. T, selaku dosen pembimbing I yang telah dengan sabar membimbing penulis, memberikan masukan, saran dan juga arahan sehingga penulis tidak hanya mampu menyelesaikan pengerjaan skripsi tetapi juga dapat mengambil banyak hikmah dan pelajaran.
2. Hani Nurhayati, M.T, selaku dosen pembimbing II yang telah teliti membimbing penulis untuk dapat mencapai hasil skripsi yang lebih baik.
3. Dr. M. Amin Hariyadi dan Fajar Rohman Hariri, M. Kom selaku dosen penguji dengan sikap profesional telah menguji seluruh proses ujian skripsi penulis mulai dari seminar proposal hingga sidang skripsi dengan lancar.
4. Seluruh jajaran staf dan dosen jurusan Teknik Informatika yang secara langsung maupun tidak langsung terlibat dalam proses pengerjaan skripsi.

5. Seluruh teman-teman Andromeda TI 16, kakak tingkat, adik tingkat dan seluruh teman seperjuangan yang secara langsung maupun tidak langsung memberikan bantuan dalam proses pengerjaan skripsi.
6. Ibu, bapak, kakak dan adik menjadi manusia yang luar biasa memberikan dukungan dan semangat tanpa lelah kepada penulis.

Peneliti menyadari bahwa proses penelitian dari awal hingga akhir masih jauh dari kata sempurna. Maka dari itu penulis membuka kesempatan selebar-lebarnya untuk setiap saran dan kritik yang membangun. Terlepas dari itu semua, peneliti berharap ada manfaat yang dapat diambil dari skripsi penulis.

Wassalamu 'alaikum Warohmatullaahi Wabarakaatuh

Malang, 18 Mei 2020

Penulis



DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERSETUJUAN.....	ii
LEMBAR PENGESAHAN	iii
PERNYATAAN KEASLIAN TULISAN	iv
HALAMAN MOTTO	v
HALAMAN PERSEMBAHAN	vi
KATA PENGANTAR.....	vii
DAFTAR ISI.....	ixx
DAFTAR GAMBAR.....	xii
DAFTAR TABEL.....	xv
ABSTRAK	xvii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Identifikasi Masalah.....	5
1.3. Tujuan Penelitian.....	6
1.4. Batasan Penelitian	6
1.5. Manfaat Penelitian.....	6
BAB II LANDASAN TEORI	8
2.1. Penelitian Terkait	8
2.2. Landasan Teori.....	10
2.2.1. <i>Blockchain</i>	10
2.2.2. <i>Smart Contract</i>	11
2.2.3. <i>Blockchain Transaction</i>	11
2.2.4. <i>Ethereum Virtual Machine</i>	12
2.2.5. <i>Block</i>	13
2.2.6 <i>Ethereum Mining Process</i>	14
2.2.6 Ether dan Gas	15
2.2.7 Proof of Work.....	18
BAB III METODOLOGI PENELITIAN	20
3.1. Desain sistem.....	20

3.2.	Skenario Game	21
3.2.1	<i>Login Menu</i>	21
3.2.2.	<i>Main Menu</i>	22
3.2.3.	<i>Stage Select</i>	22
3.2.4.	<i>Character Select</i>	23
3.2.5.	<i>Gameplay</i>	24
3.3.	Pembuatan Wallet Ethereum	27
3.4.	Token Smart Contract	27
3.5.	<i>Create child Ethereum Account</i>	28
3.6.	<i>Multiplayer System</i>	29
3.7.	Transaction	29
3.7.1.	Create Block	31
3.7.2.	Hashing Process.....	33
3.7.3.	Proof-of-Work	34
3.7.4.	Blockchain Network.....	37
3.8.	Explore data Transaction.....	38
3.9.	Skenario Pengujian Hasil	40
3.9.1.	Pengujian estimasi development cost.....	40
3.9.2.	Pengujian Kecepatan proses transaksi.....	42
3.9.3.	Pengujian nilai hash transaksi	43
BAB IV PENGUJIAN DAN EVALUASI		45
4.1.	Implementasi	45
4.1.1	Hardware yang digunakan.....	45
4.1.2	Software yang digunakan	48
4.1.3	Implementasi Skenario Game.....	48
4.2.	Pengujian Multiplayer System	57
4.2.1	Pengujian Fungsionalitas.....	58
4.2.1.1	Pengujian Login.....	58
4.2.1.2	Pengujian Create & Join Room	60
4.2.1.3	Pengujian Character Selection	65
4.2.1.4	Pengujian Interaksi antar Player	70
4.2.1.5	Pengujian Fitur Chat	85

4.2.2	Pengujian Latency	88
4.3.	Pengujian Transaction System	89
4.3.1	Pengujian Kecepatan Transaksi.....	89
4.3.1.1	Pengujian Gas Price	90
4.3.1.1	Pengujian Berdasarkan SPM	92
4.3.2	Pengujian Estimasi Development Cost	93
4.3.2.1	Pengujian Gas Limit	95
4.3.2.2	Perhitungan Estimasi Development cost	96
4.3.2	Pengujian Fungsionalitas Sistem Transaksi	101
4.4.	Integrasi Islam	103
BAB V KESIMPULAN DAN SARAN		107
5.1.	Kesimpulan.....	107
5.2.	Saran.....	108
DAFTAR PUSTAKA		109

DAFTAR GAMBAR

Gambar 1.1 Grafik pendapatan Game Online di beberapa negara	2
Gambar 3.1 Desain Sistem	20
Gambar 3.2 Tampilan Login Menu	21
Gambar 3.3 Tampilan Main Menu Game	22
Gambar 3.4 tampilan pemilihan destinasi	23
Gambar 3.5 Tampilan pemilihan karakter	24
Gambar 3.6 tampilan spawn point wisatawan di lokasi penanjakan 1	25
Gambar 3.7 tampilan saat berada di sunrise	25
Gambar 3.8 tampilan Spawn point wisatawan dilokasi kawah bromo.....	25
Gambar 3.9 Pura Luhur Poten, Kawah Bromo.....	25
Gambar 3.10 Transaction UI	26
Gambar 3.11 Tampilan Proses transaksi yang telah di proses Ethereum.....	26
Gambar 3.12 FlowChart Token SmartContract.....	28
Gambar 3.13 Ilustrasi alur transaksi.....	30
Gambar 3.14 Flowchart transaksi.....	31
Gambar 3.15 Struktur Blok Ethereum.....	32
Gambar 3.16 Proses Hashing.....	33
Gambar 3.17 Flowchart Proof-of-Work	35
Gambar 3.18 Ilustrasi Blockchain Network	37
Gambar 3.19 Rantai Blok	38
Gambar 3.20 proses open data Transaction.....	39
Gambar 3.21 Detail transaksi yang tidak tervalidasi prose Proof-of-Work	39
Gambar 3.22 Detail Transaksi yang tervalidasi.....	40
Gambar 3.23 Grafis estimasi biaya transaksi dalam mata uang (GWEI).....	41
Gambar 3.24 Detail Hasil Transaksi.....	44
Gambar 4.1 Login Menu	48
Gambar 4.2 UI Find Match.....	49
Gambar 4.3 Find Match, waiting client, client found.....	49
Gambar 4.4 Log tanda Create room diserver berhasil.....	50
Gambar 4.5 Log Client sudah ditemukan	50

Gambar 4.6 Tampilan Main Menu	51
Gambar 4.7 Tampilan Character Selection.....	51
Gambar 4.8 Spawn Point Wisatawan di area Ticketing	52
Gambar 4.9 Proses pembayaran tiket masuk wisata bromo	53
Gambar 4.10 Proses pembayaran sewa jeep.....	53
Gambar 4.11 Simulasi supir jeep mengantar wisatawan ke tempat wisata.....	54
Gambar 4.12 Spawn point lokasi wisata bromo	54
Gambar 4.13 Pura luhur poten Kawah Bromo	55
Gambar 4.14 Proses transaksi dengan pedagang dan jasa penyewaan kuda.....	55
Gambar 4.15 Wisatawan Menunggangi kuda.....	56
Gambar 4.16 Spawn Point lokasi penanjakan 1	56
Gambar 4.17 Tampilan saat eksplorasi area penanjakan 1.....	57
Gambar 4.18 Lokasi Sunrise Penanjakan 1	57
Gambar 4.19 Login Menu	59
Gambar 4.20 Script Login	59
Gambar 4.21 Script Create Room.....	60
Gambar 4.22 Tampilan setelah Create Room berhasil.....	61
Gambar 4.23 Script Login	62
Gambar 4.24 Proses Finding Match	63
Gambar 4.25 Room Ditemukan.....	63
Gambar 4.26 Karakter Wisatawan.....	65
Gambar 4.27 Karakter Supir Jeep.....	66
Gambar 4.28 Karakter Jasa Penyewaan Kuda.....	67
Gambar 4.29 Karakter Penjaga Tiket	68
Gambar 4.30 Karakter Pedagang.....	69
Gambar 4.31 UI Transaksi Sewa Jeep	71
Gambar 4.32 UI Respon Inputan Transaksi sewa jeep.....	72
Gambar 4.33 Simulasi supir mengantar wisatawan ke lokasi wisata.....	73
Gambar 4.34 UI opsi perpindahan lokasi ke kawah bromo	74
Gambar 4.35 Spawn Point Kawah bromo	74
Gambar 4.36 UI transaksi dengan pedagang	75

Gambar 4.37 UI respon inputan transaksi pembelian.....	76
Gambar 4.38 Tampilan item masuk kedalam inventory.....	77
Gambar 4.39 UI transaksi penyewaan kuda	78
Gambar 4.40 UI respon inputan pembayaran sewa jasa kuda	79
Gambar 4.41 tampilan wisatawan menaiki kuda.....	80
Gambar 4.42 Wisatawan turun dari kuda	81
Gambar 4.43 UI transaksi pembelian tiket masuk wisata Bromo	82
Gambar 4.44 UI respon pembayaran tiket masuk wisata Bromo	83
Gambar 4.45 Tampilan Gerbang masuk wisata bromo terbuka	84
Gambar 4.46 Wisatawan dapat melewati palang yang terbuka.....	84
Gambar 4.47 Tampilan awal kolom chat setelah semua player login	85
Gambar 4.48 Percobaan chat wisatawan dan penjaga tiket.....	86
Gambar 4.49 Uji coba chat dari karakter supir jeep	86
Gambar 4.50 percobaan chat dari karakter jasa penyewaan kuda.....	86
Gambar 4.51 Uji coba chat dari karakter pedagang	87
Gambar 4.52 Grafis estimasi biaya transaksi dalam mata uang (GWEI).....	94

DAFTAR TABEL

Tabel 3.1 Data rata rata biaya transaksi selama 10 hari	41
Tabel 3.2 Estimasi waktu proses transaksi	43
Tabel 4.1 Perangkat keras yang digunakan untuk pengujian	58
Tabel 4.2 Pengujian Login	60
Tabel 4.3 Pengujian Create Room.....	61
Tabel 4.4 Pengujian login.....	64
Tabel 4.5 Pengujian Seleksi Karakter wisatawan.....	66
Tabel 4.6 Pengujian Seleksi Karakter Supir Jeep.....	67
Tabel 4.7 Pengujian Seleksi karakter jasa penyewaan kuda	68
Tabel 4.8 Pengujian Seleksi Karakter Penjaga Tiket	69
Tabel 4.9 Pengujian Seleksi Karakter Pedagang.....	70
Tabel 4.10 Pengujian UI transaksi sewa jeep.....	71
Tabel 4.11 Pengujian UI respon inputan transaksi sewa jeep	72
Tabel 4.12 Pengujian simulasi mobil jeep.....	73
Tabel 4.13 Pengujian Perpindahan lokasi ke kawah bromo.....	75
Tabel 4.14 pengujian UI transaksi dengan pedagang.....	76
Tabel 4.15 Pengujian UI respon transaksi pedangan.....	77
Tabel 4.16 Pengujian Item masuk kedalam inventory setelah pembelian.....	78
Tabel 4.17 Pengujian UI pembayaran sewa kuda	79
Tabel 4.18 Pengujian UI respon inputan pembayaran sewa jasa kuda.....	80
Tabel 4.19 Pengujian Simulasi wisatawan menaiki kuda	81
Tabel 4.20 Pengujian UI Transaksi Tiket masuk wisata bromo.....	82
Tabel 4.21 Pengujian UI respon Pembelian tiket masuk wisata bromo	83
Tabel 4.22 Pengujian skenario Pembayaran tiket masuk wisata bromo	85
Tabel 4.23 Pengujian fitur chat menggunakan 5 device berbeda.....	87
Tabel 4.24 Data rating latency untuk game online.....	88
Tabel 4.25 Uji coba latency.....	89
Tabel 4.26 estimasi waktu pembuatan block.....	90
Tabel 4.27 Pengujian waktu transaksi berdasarkan perubahan gas price.....	91
Tabel 4.28 data rata rata biaya transaksi selama 10 hari	94

Tabel 4.29 Pengujian Transaksi dengan gas limit berbeda	95
Tabel 4.30 Estimasi Pengeluaran	100
Tabel 4.31 Perbandingan Biaya dan Estimasi Waktu dengan Platform lain.....	100
Tabel 5.1 Hasil Uji Multiplayer System menggunakan Platform Photon 2.....	107
Tabel 5.2 Hasil Uji perbandingan harga dan kecepatan dengan platform lain.....	107



Pradana, Reza Putra. 2020. *Sistem Transaksi Antat Player Pada game Multiplayer Wisata Bromo Menggunakan Blockchain*. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.

Pembimbing: (I) Yunifa Miftachul Arif, M. T. (II) Hani Nurhayati, M.T

Kata kunci: *Ethereum*, Industri game, Optimasi, Sistem Transaksi, SPM.

Industri game saat ini memiliki prospek penghasilan yang sangat tinggi, tetapi permasalahan keamanan seperti peretasan masih menjadi momok bagi para *game developer*. *Cryptocurrency* berbasis *Blockchain* telah mendemonstrasikan bagaimana penerapan sistem transaksi yang aman dengan arsitektur terdesentralisasi dengan membuat banyak blok yang telah terenkripsi. Namun, penggunaan teknologi *blockchain* menimbulkan permasalahan lain, yaitu kecepatan pemrosesan data yang lama dan biaya transaksi yang tinggi. Penelitian ini bertujuan untuk mengetahui metode *blockchain* yang memiliki kecepatan paling efektif dengan biaya transaksi paling efisien. Metode yang digunakan pada penelitian ini adalah optimasi variabel *gas price*, *gas limit*, *gas used*, dan *data size* pada platform *Ethereum*. Variabel optimasi akan di implementasikan pada game simulasi multiplayer wisata bromo untuk pengujian kecepatan dan biaya transaksi. hasil pengujian menunjukkan bahwa dengan optimasi variabel pada platform *Ethereum* memiliki hasil yang lebih cepat dan tarif transaksi lebih murah dibandingkan Platform lainya seperti *Bitcoin*. Dengan tarif transaksi yang sama, *Ethereum* mendapatkan kecepatan rata rata 8,8–15 kali lebih cepat dari Platform lain, dan Untuk mendapatkan kecepatan transaksi sesuai dengan SPM (Standar Pelayanan Minimal) hanya membutuhkan tarif transaksi sebesar 165.000 Gwei atau setara Rp.491 harga ini lebih murah dibanding *Bitcoin* yang memiliki biaya transaksi 4950000 Gwei atau setara Rp.17.973 per transaksi.

Pradana, Reza Putra. 2020. *Transaction System between player In Bromo Tourism Multiplayer Game Using Blockchain*. Undergraduate thesis. Department of Informatics Engineering, Faculty of Science and Technology, Maulana Malik Ibrahim State Islamic University of Malang.

Supervisor: (I) Yunifa Miftachul Arif, M. T. (II) Hani Nurhayati, M.T

Keywords: Ethereum, Game industry, Optimization, Transaction Systems, SPM,

The game industry currently has a very high income prospect, but security issues such as hacking are still a scourge for game developers. Blockchain-based cryptocurrency has demonstrated how to implement a secure transaction system with a decentralized architecture by creating lots of encrypted blocks. However, the use of blockchain technology raises other problems, namely the long data processing speed and high transaction costs. This study aims to determine the blockchain method that has the most effective speed with the most efficient transaction costs. The method used in this study is the optimization of gas price, gas limit, gas used, and data size variables on the Ethereum platform. Optimization variables will be implemented in the Bromo tour multiplayer simulation game for testing the speed and transaction costs. test results show that the optimization of variables on the Ethereum platform has faster results and cheaper transaction rates compared to other platforms such as Bitcoin. With the same transaction rates, Ethereum gets an average speed of 8.8-15 times faster than other platforms, and to get transaction speeds according to SPM (Minimum Service Standards) only requires a transaction rate of 165,000 Gwei or this price equivalent to Rp.491 cheaper than Bitcoin which has a transaction fee of 49,50000 Gwei or the equivalent of Rp.17,973 per transaction.

للسفر متعددة Bromo في لعبة Inter Player برادانا ، رضا بوترا .2020 .معاملات نظام مقال .قسم هندسة المعلومات كلية العلوم .Blockchain اللاعبين باستخدام .والتكنولوجيا الجامعة الإسلامية الحكومية مولانا مالك إبراهيم مالانج .
ال مشرف : (I) جحيو كريسيديان الدكتور (II) عجب حنى الماجستير

الكلمات الرئيسية : صناعة الألعاب ، التحسين ، أنظمة المعاملات ، Ethereum

تتمتع صناعة الألعاب حاليًا بفرصة دخل عالية جدًا ، ولكن المشكلات الأمنية مثل القرصنة لا تزال آفة لمطوري الألعاب .أظهرت العملة المشفرة المستندة إلى كيفية تنفيذ نظام معاملات آمن بهندسة لا مركزية من خلال إنشاء العديد Blockchain يثير مشاكل أخرى ، blockchain من الكتل المشفرة .ومع ذلك ، فإن استخدام تقنية وهي سرعة معالجة البيانات الطويلة وتكاليف المعاملات المرتفعة .تهدف هذه الدراسة إلى التي لديها السرعة الأكثر فعالية مع تكاليف المعاملات الأكثر blockchain تحديد طريقة كفاءة .الطريقة المستخدمة في هذه الدراسة هي تحسين سعر الغاز ، حد الغاز ، الغاز سيتم تنفيذ متغيرات .Ethereum المستخدم ، ومتغيرات حجم البيانات على منصة التحسين في لعبة محاكاة برومو متعددة اللاعبين لاختبار السرعة وتكاليف المعاملات . له نتائج أسرع Ethereum تظهر نتائج الاختبار أن تحسين المتغيرات على منصة مع نفس Bitcoin .ومعدلات معاملات أرخص مقارنة بالأنظمة الأساسية الأخرى مثل على متوسط سرعة 8.8-15 مرة أسرع من Ethereum معدلات المعاملات ، يحصل الحد) SPM الأنظمة الأساسية الأخرى ، وللحصول على سرعات المعاملات وفقًا ل أو هذا السعر Gwei الأدنى من معايير الخدمة (يتطلب فقط معدل معاملات 165,000 أو ما يعادل 17,973 لكل Gwei التي لديها رسوم معاملة Bitcoin يعادل أرخص من معاملة .

BAB I

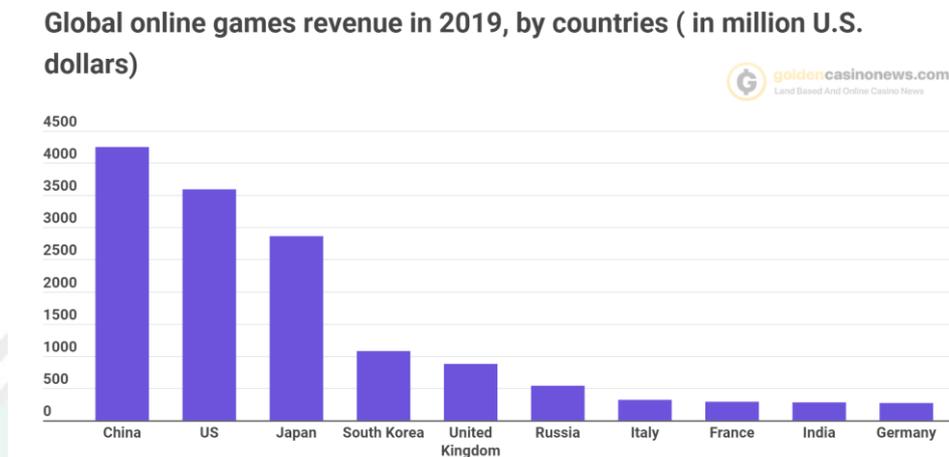
PENDAHULUAN

1.1 Latar Belakang

Industri game saat ini berkembang sangat cepat, hal ini dapat di lihat dari banyaknya game dengan menggunakan teknologi baru seperti *AI*, *VR*, dan *AR*. Hal ini juga menarik sebagian besar penduduk untuk bermain game. tidak hanya untuk mencoba, bahkan tidak sedikit dari mereka yang menjadikan game sebagai pekerjaannya. Mulai dari *Game Developer*, *Atlet Esport*, *Streamers*, dan masih banyak lagi. Bahkan Menurut sebuah survei online terhadap 5000 orang yang dirilis oleh Electronic Entertainment Design and Research pada tahun 2018, hampir 70% dari populasi Amerika, atau sekitar 211 juta orang, bermain video game di setidaknya satu jenis perangkat. Hasilnya menunjukkan bahwa 90% dari mereka yang disurvei bermain game di *Smartphone*, tablet, atau keduanya. Studi ini juga menemukan bahwa *Gamer A.S* rata-rata menghabiskan 12 jam per minggu untuk video game (Crecente 2018). Keterlibatan video game terus tumbuh dan itu bisa berarti prospek besar bagi industri game.

Dalam produksi Game, terdapat dua jenis Game, yaitu *Online* dan *Offline*, yang memiliki target pasar yang berbeda. Saat ini Game *online* mewakili penghasil pendapatan terbesar di seluruh pasar game. Pada 2019, pendapatan di segmen game online mencapai \$ 16,9 miliar. Bagian paling besar dari jumlah itu, atau \$ 4,2 miliar, dihasilkan di Cina. Amerika Serikat adalah penghasil pendapatan game online terbesar kedua pada tahun lalu, dengan laba \$ 3,5 miliar. Dengan pendapatan \$ 2,8 miliar, Jepang menempati posisi ketiga dalam daftar pendapatan game online global

ini, diikuti oleh Korea Selatan dan Inggris. Penelitian GoldenCasinoNews.com baru-baru ini menunjukkan peringkat yang sama di segmen game mobile.



Gambar 1.1 Grafik pendapatan Game Online di beberapa negara dari Penelitian GoldenCasinoNews.com

Meskipun prospek industri game menjanjikan, sejumlah besar uang hilang dari unduhan atau penggandaan asset secara tidak legal. Berdasarkan penelitian tahun 2015 yang dilakukan oleh Apsalar (perusahaan *Mobile Marketing Cloud*), untuk setiap asset *virtual* yang dijual dan diunduh, ada 7,5 asset *virtual* yang hilang karena penipuan dan peretasan. Jumlah ini jauh lebih tinggi di negara-negara seperti China di mana ada 273 asset *virtual* palsu yang diunduh untuk setiap asset yang legal. Selain itu, pembelian dalam aplikasi yang curang memungkinkan beberapa pemain mendapatkan keuntungan yang tidak adil. Ini merusak ekonomi permainan dan merusak pengalaman pemain lain. Teknik serangan peretas yang dihadapi oleh industry Game juga terus berubah untuk menghindari deteksi. hal ini menimbulkan kerugian biaya yang besar terkait dengan kehilangan dan duplikasi asset *virtual*. Hal ini juga menjadi tantangan bagi dan dampak negatif yang sangat besar pada pertumbuhan pengguna dan keuntungan industri game (Xie 2015).

Keamanan dan kejujuran merupakan sesuatu yang sulit ditemukan sekarang, penipuan, peretasan, *cracking* dan masih banyak lagi kejahatan *cyber* seiring dengan perkembangan teknologi. hal itu tidak hanya dirasakan oleh *developer*, bahkan *player* game juga tidak luput dari kejahatan *cyber* tersebut. Tidak sedikit dari para *player* yang kehilangan asset yang sudah mereka beli, dan juga tidak sedikit *player* yang sudah mengalokasikan sejumlah uang asli mereka kedalam game dan uang mereka hilang. Dalam kejahatan *cyber* ini juga tidak selalu dilakukan oleh *hacker* atau *cracker*, ada beberapa kasus penipuan yang melibatkan *developer* game, mereka dengan sengaja menghapus atau mengambil sebagian saldo uang yang dimiliki *player*. Hal ini juga menyebabkan kredibilitas *player* berkurang terhadap semua game, walaupun masih banyak *developer* game yang jujur. Di dalam al-quran surat Al-Ahzab ayat 23-24, Allah SWT telah memerintahkan kepada manusia untuk berlaku jujur.

مِنَ الْمُؤْمِنِينَ رَجَالٌ صَدَقُوا مَا عَاهَدُوا اللَّهَ عَلَيْهِ فَمِنْهُمْ مَّنْ قَضَىٰ نَحْبَهُ وَمِنْهُمْ مَّنْ يَنْتَظِرُ وَمَا بَدَّلُوا تَبْدِيلًا

لِيَجْزِيَ اللَّهُ الصَّادِقِينَ بِصِدْقِهِمْ وَيُعَذِّبَ الْمُنَافِقِينَ إِن شَاءَ أَوْ يَتُوبَ عَلَيْهِمْ إِنَّ اللَّهَ كَانَ غَفُورًا رَّحِيمًا *

Artinya : “Di antara orang-orang mukmin itu ada orang-orang yang menepati apa yang telah mereka janjikan kepada Allah. Dan di antara mereka ada yang gugur, dan di antara mereka ada (pula) yang menunggu-nunggu dan mereka sedikit pun tidak mengubah (janjinya),(23) agar Allah memberikan balasan kepada orang-orang yang benar itu karena kebenarannya, dan mengazab orang munafik jika Dia kehendaki, atau menerima tobat mereka. Sungguh, Allah Maha Pengampun, Maha Penyayang. (24)” – (Q.S Al-Ahzab: 23-24)

Dalam tafsir al-Jalalain ayat tersebut memiliki tafsir : Di antara orang-orang Mukmin itu ada orang-orang yang menepati apa-apa yang telah mereka janjikan kepada Allah) yaitu gigih bertahan bersama dengan Nabi saw. (maka di antara mereka ada yang gugur) mati atau terbunuh di jalan Allah (dan di antara mereka ada pula yang menunggu-nunggu) hal tersebut (dan mereka sedikit pun tidak mengubah) janjinya, berbeda halnya dengan orang-orang munafik. (Supaya Allah memberikan balasan kepada orang-orang yang benar itu karena kebenarannya, dan menyiksa orang-orang munafik jika dikehendaki-Nya) seumpamanya Dia mematikan mereka dalam kemunafikannya (atau menerima tobat mereka. Sesungguhnya Allah adalah Maha Pengampun)(lagi Maha Penyayang) kepada orang yang bertobat kepada-Nya.

Saat ini para *developer* sedang berlomba-lomba untuk terus mengembangkan keamanan game mereka. Salah satu dari strategi yang saat ini digunakan untuk meningkatkan keamanan game mereka adalah Teknik desentralisasi Blockchain menggunakan Platform Ethereum. Teknik ini sangat efektif untuk meningkatkan keamanan Game karena struktur desentralisasi Blockchain membuat banyak blok yang masing-masing blok telah terenkripsi. Hal ini membuat peretasan, duplikasi, dan mencuri kunci menjadi sangat sulit. Selain keamanan, blockchain juga memungkinkan pemain untuk bertransaksi tanpa batas, karena teknologi blockchain tidak memerlukan pengawasan, atau tidak memerlukan pihak ketiga untuk melakukan transaksi, mereka bisa langsung bertransaksi tanpa harus ada campur tangan pihak ketiga. Bagi developer Teknik blockchain juga dapat menjadi strategi pemasaran asset item game dengan

membatasi jumlahnya. Sebagai contoh, game CryptoKitties menggunakan teknologi ini untuk menyimpan kucing lucu yang dapat diperoleh dan ditukar pengguna. perusahaan itu disambut dengan kesuksesan tak terduga, Asset tersebut menjadi kucing terlaris dengan harga \$ 115.000. Bukti terbatasnya pasokan memicu tingginya permintaan (Katalyse 2018).

Diantara banyaknya subgenre yang ada sekarang ini, sub genre simulasi merupakan genre game yang paling beresiko terjadi suatu penipuan, pengrusakan data, atau hacking, hal ini dikarenakan sangat banyak transaksi yang dilakukan di dalam sub genre ini. Terlebih pada game simulasi seirng menggunakan konsep ekomoni yang hampir seperti kehidupan nyata. Namun, game simulasi yang beredar saat ini masing menggunakan konsep sentralisasi yang sangat riskan terjadi kerusakan data atau data hilang.

Pada penelitian ini penulis akan meneliti game simulasi online dengan penerapan Blockchain pada transaksi item. Dengan penelitian ini diharapkan dapat memudahkan pemain untuk bertransaksi dengan developer maupun player lain serta dapat terjamin keamanannya. Setiap proses transaksi nantinya bisa di akses oleh semua pihak yang terkait dan tingkat tingkat keamanan datanya terjaga. Sehingga resiko terjadinya data yang hilang atau rusak akan terminimalisir. Latar yang di ambil pada penelitian ini adalah wisata gunung bromo, sehingga penelitian ini di bangun semirip mungkin dengan keadaan gunung bromo. Hal ini bertujuan agar game wisata bromo ini juga dapat dijadikan simulasi bagi orang yang belum pernah mengunjungi wisata gunung bromo, dan informasi yang ada di game wisata bromo

diharapkan dapat menjadi landasan bagi mereka di saat mereka mengunjungi wisata gunung bromo.

1.2 Pernyataan Masalah

Berdasarkan latar belakang yang telah dipaparkan di atas, maka dapat diketahui bahwa permasalahan yang diangkat pada penelitian ini adalah sebagai berikut :

1. Seberapa efektifkah implementasi *multiplayer* dengan menggunakan platform Photon 2.
2. Apakah *Blockchain* dengan platform ethereum dapat menghemat pengeluaran produksi dan pemeliharaan game.
3. Seberapa efektifkah implementasi *Blockchain* pada kecepatan transaksi game simulasi multiplayer wisata bromo.

1.3 Tujuan Penelitian

Adapun Adapun tujuan dilakukannya penelitian ini adalah sebagai berikut :

1. Mengetahui keefektifitasan penggunaan platform photon 2 pada implementasi *multiplayer* game simulasi wisata bromo.
2. Mengetahui biaya pengeluaran *blockchain* dengan platform Ethereum pada produksi dan pemeliharaan game.
3. Mengetahui keefektifitasan implementasi *blockchain* platform Ethereum pada kecepatan transaksi game simulasi wisata bromo.

1.4 Batasan Penelitian

Agar pembahasan penelitian ini tidak menyimpang dari apa yang telah dirumuskan, maka diperlukan batasan-batasan dalam pengerjaannya. Berikut merupakan batasan-batasan yang ada dalam penelitian ini :

1. Objek wisata yang digunakan dalam penelitian ini yaitu taman wisata Bromo Tengger Semeru
2. Platform Blockchain yang digunakan adalah Ethereum.
3. Platform Multiplayer yang digunakan adalah Photon 2.
4. Fokus penelitian adalah transaksi pengiriman saldo mata uang kripto.

1.5 Manfaat Penelitian

Adapun manfaat dari penelitian ini nantinya yang dapat diambil adalah sebagai berikut :

1. Membantu para developer untuk meningkatkan sistem keamanan game mereka.
2. Membantu developer untuk membangun sistem transaksi yang dapat menghemat pengeluaran biaya development dan pemeliharaan.
3. Memberikan keamanan pada asset milik player yang telah mereka dapatkan dari game agar tidak mudah diretas atau takedown dari pusat server game.
4. Memberikan kredibilitas kepada setiap transaksi yang dilakukan oleh player, hal ini dikarenakan tidak adanya pihak ketiga yang melakukan proses transaksi.
5. Membangun sistem ekonomi game yang baru, yaitu sistem yang aman, jujur, dan transparan.

BAB II

LANDASAN TEORI

Bab ini menjelaskan beberapa studi pustaka yang digunakan sebagai dasar teori dalam penelitian. Selain itu, bab ini juga membahas tentang penelitian sebelumnya yang terkait dengan penelitian yang akan dilakukan.

2.1 Penelitian Terkait

Penelitian-penelitian yang telah dilakukan sebelumnya menjadi bahan referensi dalam melakukan penelitian ini.

1. ***Ethereum* Crypto-Games: Mechanics, Prevalence and Gambling Similarities**

Pada penelitian (James et al., 2019) melakukan penelitian implementasi *Blockchain Ethereum* untuk pembuatan game gambling. Pada penelitian ini, peneliti berfokus pada *Smart Contract* dan transaksi yang dilakukan oleh pemain dan juga developer. dari penelitian ini dihasilkan *Smart Contract* berupa mata uang yang dapat dikelola oleh developer tetapi tetap memiliki nilai jual dan dapat di tukarkan dengan mata uang *Ethereum*, penelitian ini juga menghasilkan sebuah metode transaksi yang aman dan transparan.

2. **Increasing Motivation for Playing *Blockchain* Games Using Proof-of-Achievement Algorithm**

Pada penelitian (kosuke, 2019) melakukan penelitian dengan mengimplementasikan *Blockchain* algoritma *Proof of achievement*. Pada penelitian ini, peneliti berfokus pada bagaimana cara agar pemain dapat bertahan untuk

memainkan game yang mereka buat. Faktor bosan selalu menghantui para developer game, oleh karena itu para developer selalu berlomba lomba untuk melakukan sebuah inovasi cara agar pemain dapat terus memainkan game mereka. Pada penelitian ini, menggunakan algoritma *Proof of Achievement* yang menggunakan *Smart Contract* bukan berbentuk mata uang, melainkan berbentuk penghargaan. Untuk mendapatkan penghargaan tersebut, pemain diharuskan menyelesaikan misi atau memecahkan suatu batasan. sama seperti token *Smart Contract* yang lain, pada game ini, pemain yang mendapatkan penghargaan dapat menjual penghargaan mereka menjadi mata uang *Ethereum*. Tujuan dari penelitian ini adalah membuat pemain dapat terus memainkan game, atau bahkan bertambah semangat karena setiap penghargaan yang mereka dapatkan memiliki nilai jual yang dapat mereka tukarkan menjadi uang asli.

3. Is a *Blockchain*-Based Game a Game for Fun, or Is It a Tool for Speculation? An Empirical Analysis of Player Behavior in Crypokitties.

Pada penelitian (jaehwaan et al., 2019) melakukan penelitian tentang seberapa efektifkah metode *Blockchain* di implementasikan kedalam game, apakah metode *Blockchain* bisa menambah unsur menyenangkan pada saat pemain memainkan game tersebut, atau ada kemungkinan yang lain. Dalam penelitian ini, peneliti melakukan analisis pada salah satu game pertama yang mengimplementasikan metode *Blockchain* dan tergolong game yang sukses, yaitu game Crypokitties. Metode yang digunakan peneliti adalah dengan cara membangun sebuah website yang dapat menganalisa segala transaksi dan beberapa perilaku pemain yang dilakukan pada game Cryptokitties. Hasil dari penelitian ini

adalah, didapatkan data mengenai transaksi yang dilakukan dan beberapa perilaku yang dilakukan pemain di dalam game, dari hasil data tersebut didapatkan kesimpulan berupa game yang mengimplementasikan metode *Blockchain* dapat menambah keseruan bermain, tetapi tidak kepada semua pemain, terdapat beberapa variabel yang membuat pengaruh *Blockchain* terhadap game Cryptokitties dapat meningkatkan jumlah pemain, dan juga tidak berpengaruh sama skali. Kesimpulan dari penelitian ini adalah masih belum jelas apakah pengguna memainkan game berbasis *Blockchain* hanya untuk bersenang-senang atau sebagai bentuk spekulasi.

2.2 Landasan Teori

Pada sub bab ini menjelaskan mengenai teori – teori yang berkaitan dengan penelitian ini.

2.2.1 *Blockchain*

Blockchain merupakan sebuah metode penyimpanan data secara terdesentralisasi terdiri dari *node* yang saling terhubung pada jaringan *Blockchain*, yang di dalamnya terdapat catatan – catatan transaksi data berbentuk blok yang saling membentuk rantai blok. *Blockchain* bertindak sebagai basis data terdistribusi atau buku besar global yang menyimpan catatan semua transaksi pada jaringan *Blockchain*. Transaksi dicap berdasarkan waktu prosesnya dan di konversi ke dalam blok di mana setiap blok diidentifikasi oleh *Hash* kriptografiknya. Blok membentuk urutan linier di mana setiap blok referensi *Hash* dari blok sebelumnya, membentuk rantai blok yang disebut *Blockchain*. *Blockchain* dikelola oleh jaringan *node* dan setiap *node* mengeksekusi dan mencatat transaksi yang sama. *Blockchain*

direplikasi di antara *node* di jaringan *Blockchain*. Setiap simpul dalam jaringan dapat membaca transaksi. (Bahga, et.,all, 2016)

2.2.2 *Smart Contract*

Smart contract merupakan salah satu kelebihan dari *Blockchain Ethereum*, dimana semua orang dapat memprogram sendiri transaksi yang mereka inginkan, misalnya, sistem akan mengirim sejumlah token yang di inputkan dengan syarat kondisi yang di inginkan pengirim terpenuhi. Selain itu Smart Contract juga dapat membuat suatu mata uang sendiri untuk beberapa keperluan.

Kasus penggunaan yang paling terbukti untuk kontrak pintar sejauh ini adalah token khusus dan crowdsales. Ratusan token telah diluncurkan dan dijual di *Ethereum* sejauh ini. Crowdsales biasanya disebut sebagai token sales, persembahan koin awal, atau ICO.

Escrow kontrak pintar telah menjadi populer untuk transfer token antara pihak yang tidak dipercaya. Penjual memberikan kendali token ke kontrak pintar, dan token dikirim ke pembeli hanya ketika pembeli mengirim eter ke kontrak.

Aset digital lainnya selain token dapat disimpan dengan menggunakan kontrak pintar juga. Perusahaan telah membangun kontrak untuk membuat saham, real estat, emas, dolar AS, dan banyak aset lainnya tersedia dan dapat diperdagangkan di *Blockchain Ethereum*. (Iyer, et.,all, 2018)

2.2.3 *Blockchain Transaction*

Transaksi mengirim eter, menyebarkan kontrak pintar, atau menjalankan fungsi pada kontrak pintar yang ada. Transaksi mengkonsumsi *Gas*, unit pengukuran *Ethereum* yang menentukan kompleksitas dan biaya jaringan

operasi kode. Biaya *Gas* dari suatu transaksi digunakan untuk menghitung biaya transaksi. Biaya transaksi dibayarkan oleh alamat yang mengirimkan transaksi ke penambang yang menambang blok tersebut.

Transaksi dapat berisi bidang data opsional. Untuk transaksi penyebaran kontrak, data adalah kode kunci kontrak. Untuk transaksi yang dikirim ke kontrak pintar, data berisi nama dan argumen untuk menjalankan fungsi. (Dannen, et.,all, 2018)

2.2.4 *Ethereum* Virtual Machine

Prosesor adalah sirkuit terintegrasi yang menjalankan serangkaian instruksi yang diberikan. Setiap prosesor memiliki serangkaian operasi yang dapat dilakukan. Instruksi terdiri dari kode operasi, atau opcode, diikuti oleh input data untuk operasi. Set instruksi x86 adalah set instruksi paling umum yang digunakan saat ini dan memiliki sekitar 1.000 opcode unik.

Suatu program adalah serangkaian instruksi yang dijalankan secara membabi buta. Semua kode baik itu kartu punch, rakitan, atau bahasa tingkat tinggi seperti Python dikompilasi atau ditafsirkan hingga serangkaian byte mentah. Bytes ini sesuai dengan serangkaian instruksi prosesor yang dapat dijalankan komputer secara berurutan, seperti mesin bisu. Listing 1-1 menunjukkan seperti apa program Hello World di x86 Linux Assembly. (Iyer, et.,all, 2018)

Ethereum adalah *Blockchain* yang dapat diprogram dengan jaringan peer to peer di mana setiap rekan menyimpan salinan yang sama dari database *Blockchain* dan menjalankan mesin virtual *Ethereum* untuk mempertahankan dan mengubah keadaannya. EVM adalah mesin virtual yang dirancang untuk dijalankan oleh

semua peserta dalam jaringan peer to peer, dapat membaca dan menulis ke *Blockchain* baik kode dan data yang dapat dieksekusi, Verifikasi tanda tangan digital, dan mampu menjalankan kode secara kuasi-Turing secara lengkap . Itu hanya akan mengeksekusi kode ketika menerima pesan yang diverifikasi oleh tanda tangan digital, dan informasi yang disimpan di *Blockchain* mengatakan bahwa itu pantas untuk dilakukan. EVM bekerja dengan cara yang sama seperti mesin virtual lainnya: EVM akan mengambil bahasa pemrograman tingkat tinggi yang dirancang untuk menulis kontrak pintar, dan dapat mengkompilasinya menjadi bytecode EVM yang dapat dipahami mesin. *Ethereum Virtual Machine* adalah komputer terdesentralisasi besar yang memiliki jutaan akun. Akun dapat dilihat sebagai objek yang memiliki kapasitas untuk menangani basis data internal, menjalankan kode kontrak dan dapat berkomunikasi dengan akun lain. Kontrak pintar itu sendiri dapat dianggap sebagai akun. Kunci pribadi digunakan untuk menangani EOA (Akun Milik Eksternal); hanya pemilik kunci pribadi ini yang memiliki hak istimewa lengkap untuk melakukan transaksi dan mengirim eter menggunakan EOA (*Ethereum Homestead Documentation*, 2016).

2.2.5 Block

Seperti yang dinyatakan sebelumnya, *Blockchain* terdiri dari serangkaian blok yang dipesan. Blok terdiri dari header dengan informasi meta dan serangkaian transaksi. Blok dibuat oleh penambang melalui proses Mining dan disiarkan ke sisa jaringan. Setiap node memverifikasi blok yang diterima terhadap serangkaian aturan konsensus. Blok yang tidak memenuhi aturan konsensus akan ditolak oleh jaringan.

Garpu terjadi ketika jaringan memiliki sekumpulan aturan konsensus yang bersaing. Ini biasanya terjadi melalui pembaruan di klien resmi, yang dalam kasus *Ethereum* adalah program yang disebut *geth*.

Garpu lunak terjadi ketika set aturan yang lebih baru adalah bagian dari aturan lama. Klien yang masih menggunakan aturan lama tidak akan menolak blok yang dibuat oleh klien menggunakan aturan baru, jadi hanya pembuat blok (penambang) yang harus memperbarui perangkat lunak mereka.

Garpu keras terjadi ketika set aturan baru tidak kompatibel dengan set lama. Dalam hal ini, semua klien harus memperbarui perangkat lunak mereka. Garpu keras cenderung menjadi pertengkar. Jika sekelompok pengguna menolak untuk memperbarui perangkat lunak mereka, terjadi split rantai, dan blok yang valid pada satu rantai tidak akan valid pada yang lain. Ada enam garpu keras di *Ethereum*, salah satunya menyebabkan perpecahan rantai dan penciptaan *Ethereum Classic* (ETC). (Iyer, et.,all, 2018)

2.2.6 *Ethereum* Mining Process

Mining node dalam jaringan *Ethereum* bersaing untuk membuat blok dengan menggunakan algoritma proof-of-work milik yang disebut *EtHash*. Input ke algoritma *EtHash* adalah header blok, yang mencakup nomor yang dihasilkan secara acak yang disebut *nonce*. Outputnya adalah nomor hex 32-byte. Memodifikasi *nonce* memodifikasi output, tetapi dengan cara yang tidak terduga.

Agar jaringan dapat menerima blok yang ditambang, output *EtHash* untuk header blok harus kurang dari kesulitan jaringan, nomor hex 32 byte lain yang bertindak sebagai target untuk dikalahkan. Setiap penambang yang menyiarkan

blok yang mengalahkan kesulitan target menerima hadiah blok. Hadiah blok diberikan dengan memasukkan transaksi coinbase di blok. Transaksi coinbase biasanya merupakan transaksi pertama di blok dan mengirimkan hadiah blok ke penambang. Balok blok saat ini karena garpu keras Bizantium adalah 3 eter.

Terkadang dua penambang menghasilkan blok sekitar waktu yang sama, dan hanya satu yang diterima di rantai utama. Blok yang tidak diterima disebut blok paman. Paman blok termasuk dalam rantai dan menerima hadiah blok yang lebih rendah, tetapi transaksi mereka tidak mengubah pohon negara.

Keamanan *Blockchain* sebanding dengan jumlah *Hashpower* dalam jaringan. Semakin banyak *Hashpower* dalam jaringan berarti setiap penambang individu memiliki persentase yang lebih kecil dari total *Hashpower* dan membuat serangan pengambilalihan jaringan menjadi lebih sulit (lihat “51% Serangan” di Bab 6). Termasuk blok paman dalam rantai meningkatkan keamanan rantai karena *Hashpower* yang digunakan untuk membuat blok yang tidak diterima tidak terbuang sia-sia. Kesulitan jaringan secara konstan disesuaikan sehingga blok diproduksi setiap 15-30 detik. (Iyer, et.,all, 2018)

2.2.7 Ether dan Gas

Seperti yang telah kita bahas sebelumnya, *Ethereum* menggunakan cryptocurrency ether-nya sendiri, untuk melaksanakan kontrak melalui *Blockchain*. Biaya perhitungan tergantung pada kerumitannya. Setiap kontrak dan transaksi memiliki harga awal tetap yang digunakan untuk membayar penambang sebagai imbalan penambangan sebagai imbalan atas daya yang dihabiskan oleh komputasi. Harga

Gas untuk setiap transaksi ditemukan dengan mengalikan jumlah perhitungan dengan harga *Gas* saat ini, dan dengan demikian memberikan persamaan

Proyek *Ethereum* berbeda dari Bitcoin dalam kenyataan bahwa para penambang akan memproses transaksi acak dari blok-blok sebelumnya dan menghasilkan *Hash* dari hasil untuk blok selanjutnya. Ini berarti bahwa, kontrak pintar memiliki kekuatan untuk melakukan segala bentuk urutan komputasi sementara transaksi berlangsung dan dapat membuat rantai sesuai dengan pernyataan kontrak. Selain itu, seorang penambang harus memiliki seluruh *Blockchain*, sehingga konsep kolam penambangan tidak berfungsi sama dengan Bitcoin.

Mirip dengan pengguna Bitcoin, pengguna *Ethereum* juga harus membayar biaya transaksi ke jaringan *Ethereum* untuk melakukan transaksi apa pun melalui *Ethereum Blockchain*. Gagasan biaya transaksi ini adalah untuk melindungi pemblokiran dari serangan pengguna yang tidak sah dan transaksi jahat. Para pengguna yang memiliki eter cukup untuk membayar biaya transaksi dapat melakukan transaksi di *Blockchain*. Mirip dengan bitcoin *Blockchain*, *Blockchain Ethereum* memungut biaya untuk setiap langkah perhitungan dan penyimpanan data. Eter yang dikumpulkan oleh para penambang di *Blockchain Ethereum* akan dikeluarkan sebagai hadiah untuk setiap blok valid yang ditambang. Penghargaan ini mempromosikan lebih banyak penambang untuk membangun blok *Blockchain* secara kompetitif dan untuk melindungi *Blockchain* dari penyerang (*Ethereum Homestead Documentation*, 2016).

Seperti yang kita bahas sebelumnya, *Gas* dalam *Etherchain Blockchain* adalah untuk memasukkan nilai tetap untuk setiap transaksi dan perhitungan di jaringan *Ethereum*. Ada beberapa konsep yang terkait dengan *Gas* seperti *Gas*

Harga, Biaya *Gas*, Batas *Gas*, dan Biaya *Gas*

- *Biaya Gas*: Ini mewakili nilai biaya statis untuk perhitungan lebih dari *Etherchain Blockchain* dalam hal *Gas*. *Biaya Gas* digunakan dalam kontrak pintar untuk memiliki kontrol atas perhitungan dari waktu ke waktu.
- *Harga Gas*: Ini mewakili biaya aktual yang harus dikeluarkan sebagai *biaya Gas* dalam bentuk mata uang seperti *Ether*. Biasanya harga *Gas* akan menjadi nilai mengambang untuk menstabilkan nilai *Gas*, yang berarti bahwa dengan perubahan nilai mata uang, harga *Gas* juga berubah untuk beradaptasi dengan perubahan nilai. Harga *Gas* ditentukan oleh harga ekuilibrium dari seberapa banyak pengguna bersedia untuk menghabiskan, dan berapa banyak node pemrosesan bersedia menerima.
- *Batas Gas*: Ini menentukan jumlah maksimum *Gas* yang dapat digunakan blok per. Ini digambarkan sebagai beban komputasi maksimum, volume transaksi, atau ukuran blok, dan para penambang dapat secara bertahap mengubah batas *Gas* selama periode waktu tertentu.
- *Biaya Gas*: Ini adalah jumlah *Gas* yang harus dibayar untuk melaksanakan transaksi atau kontrak tertentu. Para penambang akan dibayar dengan *biaya Gas* berdasarkan biaya komputasi yang dikeluarkan untuk membangun blok.

2.2.8 *Proof of Work*

Penciptaan metodologi *Proof of Work* adalah salah satu inovasi utama yang memungkinkan untuk Bitcoin. *Ethereum* telah berkomitmen untuk menggunakan *Proof of Work* selama tiga tahun pertama keberadaannya sejauh ini, setelah itu ia berencana untuk menggunakan bukti kepemilikan. Kami membahas Mining *Proof of Work* secara singkat di Bab 1.

Proof of Work pertama kali diusulkan dalam makalah 1992 sebagai tindakan antispam. Itu diusulkan lagi secara terpisah sebagai langkah untuk mencegah serangan DDoS. Tidak ada yang sangat sukses, tetapi membuka jalan untuk penggunaannya dalam cryptocurrency.

Bitcoin menggunakan *Hashing* SHA-256 dalam algoritma proof-of-work-nya. Penambang *Hash* header blok berulang kali dengan SHA-256, menggunakan bidang nonce dan timestamp untuk membuat input baru. Penambang yang menghasilkan *Hash* lebih rendah dari kesulitan jaringan 32-byte menyebarkan blok mereka ke jaringan. Sebagai hadiah untuk *Hashpower* mereka, penambang dapat memasukkan transaksi berbasis koin yang memberi hadiah bitcoin baru kepada diri mereka sendiri.

Banyak varian dari algoritma proof-of-work telah dirancang untuk memperbaiki kekurangan SHA-256. Kesederhanaan dari algoritma SHA-256 membuatnya mudah untuk ditambah dengan sirkuit terintegrasi khusus aplikasi (ASICs). Litecoin menggunakan Scrypt, yang dimaksudkan untuk intensif memori, tetapi ASIC tetap dibuat untuk itu. Monero menggunakan CryptoNight, yang telah

berhasil memanfaatkan panggilan sistem khusus untuk menjadikan Mining CPU tradisional sebagai metode yang paling efektif.

Ethereum menggunakan *EtHash*, algoritma khusus yang sesuai dengan desainnya, tetap merupakan metode yang paling berhasil untuk ditambang dengan unit pemrosesan grafis (GPU). *EtHash* membutuhkan pembacaan berulang dari grafik asiklik terarah (DAG) yang disimpan dalam memori. Ukuran DAG lebih besar dari 1GB, sehingga ASIC, yang umumnya mengandung sangat sedikit memori, tidak layak untuk ditambang.

Jika Anda ingin mengetahui detail pasti dari algoritma *EtHash*, lihat wiki *EtHash* (<https://github.com/Ethereum/wiki/wiki/EtHash>). Untuk kode uji Mining, Anda dapat membuat DAG saat ini dengan yang berikut: `geth makedag`

Mining *Proof of Work* mengamankan *Blockchain* dengan memaksa penyerang yang ingin mengambil alih jaringan untuk mengendalikan 51% dari *Hashpower* (lihat “51% Serangan”). Memblokir hadiah dan biaya transaksi memberi insentif kepada penambang untuk menambang blok. Kesulitan jaringan disesuaikan secara berkala untuk mempertahankan target waktu blok rata-rata. Target waktu blok rata-rata jaringan *Ethereum* saat ini sekitar 20 detik.

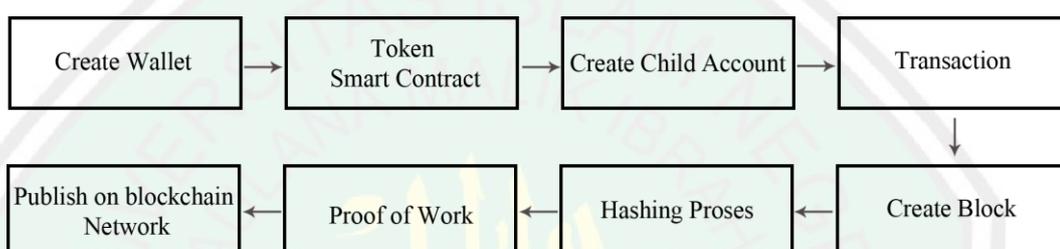
Angka ini perlahan-lahan merayap ke atas karena "bom kesulitan" dalam kode *Ethereum*, yang perlahan-lahan meningkatkan waktu blok rata-rata. Motif di balik ini adalah untuk memaksa *Ethereum* untuk beralih ke proof-of-stake ketika waktu blok menjadi terlalu besar dan menempatkan topi pada penciptaan eter baru dari hadiah blok. Setelah bukti kepemilikan diajukan, blok hadiah mungkin akan berkurang atau bahkan bisa menjadi nol.

BAB III

METODOLOGI PENELITIAN

3.1 Desain sistem

Desain system pada penelitian Penerapan *Blockchain* Untuk Transaksi Antar *Player* Pada *Game Multiplayer* Wisata Bromo Dapat dilihat pada gambar berikut :



Gambar 3.1 Desain Sistem

Pada gambar 3.1 desain sistem diatas, penelitian dimulai dengan pembuatan *Wallet Ethereum* melalui metamask, *Wallet* ini berfungsi sebagai parent dari segala transaksi yang akan dilakukan di dalam *Game*. Setelah pembuatan *Wallet*, proses selanjutnya adalah membuat contract token yang berfungsi sebagai mata uang kripto yang dapat diprogram sesuai dengan kebutuhan *Game*. Setelah contract selesai dibuat, proses selanjutnya adalah proses dimana pemain mulai memegang kendali, dimulai dengan pembuatan akun di dalam *Game* yang akan secara otomatis juga akan terdaftar pada *Wallet Ethereum*. Setelah proses pendaftaran, pemain dapat langsung masuk kedalam *Game*. Di dalam *Game*, pemain dapat bereksplorasi dan bertransaksi dengan pemain lain. pada proses transaksi, pemain dapat bertukar Token yang mereka miliki dan proses *Blockchain* berlanjut pada proses ini. Pada setiap proses transaksi, akan mengirim request pada *Ethereum* untuk membentuk struktur block sesuai dengan request, setelah struktur block benar, akan dilanjutkan

dengan proses *Hashing* menggunakan standart SHA3. Setelah didapatkan struktur block dan *Hash* number, akan dilakukan proses *Proof-of-Work* yang akan memvalidasi data yang ada di block apakah dapat di broadcast ke jaringan desentralisasi *Blockchain*. Setelah block tervalidasi, block akan membentuk sebuah rantai dengan block lainnya yang telah berhasil divalidasi. Setiap block yang terhubung dalam jaringan desentralisasi *Blockchain* tersebut akan membentuk sebuah record dimana data transaksi semua pemain tercatat didalamnya.

3.2 Skenario *Game*

Game simulasi wisata bromo merupakan *Game* yang mensimulasikan tentang aktifitas sehari-hari yang ada di tempat wisata gunung bromo. *Game* ini berawal dengan pemilihan karakter yang ingin dimainkan, setiap karakter memiliki peran yang berbeda-beda, diantaranya ada wisatawan, sopir jeep, pedagang toko, penjaga tiket, dan jasa penyewaan kuda. Setelah pemilihan karakter, pemain akan langsung masuk kedalam *Game* dan harus menjalankan tugas sesuai dengan peran masing masing. Untuk visualisasi storyboard dari *Game* wisata bromo seperti di bawah ini

3.2.1 Login menu.



Gambar 3.2 Tampilan Login Menu

Langkah pertama saat akan memainkan *Game* simulasi wisata gunung bromo adalah pemain diharuskan untuk mendaftarkan akun, hal ini difungsikan untuk inisialisasi informasi karakter di dalam *Game*, dan juga untuk mendapatkan akun *Ethereum* yang digunakan untuk transaksi di dalam *Game*

3.2.2 Main Menu



Gambar 3.3 Tampilan Main Menu Game

Selanjutnya jika pemain sudah mendaftar atau login, akan tampil UI Main menu seperti di atas, terdapat tiga button dalam UI ini, yaitu mulai untuk proses memulai *Game*, tentang untuk mengetahui lebih lanjut tentang *Game* ini, keluar untuk menutup aplikasi.

3.2.3 Stage Select

Saat menekan button mulai pada main menu UI, akan terlihat tampilan seperti ini. Terdapat dua lokasi yang dapat dimainkan oleh *Player*, yaitu lokasi penanjakan 1 dan kawah bromo.



Gambar 3.4 tampilan pemilihan destinasi

3.2.4 Character Select

Setelah memilih lokasi, pemain diharuskan memilih karakter mereka. setiap karakter memiliki peran yang berbeda-beda, diantaranya ada wisatawan, sopir jeep, pedagang toko, penjaga tiket, dan jasa penyewaan kuda. Setelah pemilihan karakter, pemain akan langsung masuk kedalam *Game* dan harus menjalankan tugas sesuai dengan peran masing masing, berikut adalah beberapa tugas yang harus diselesaikan oleh masing masing karakter.



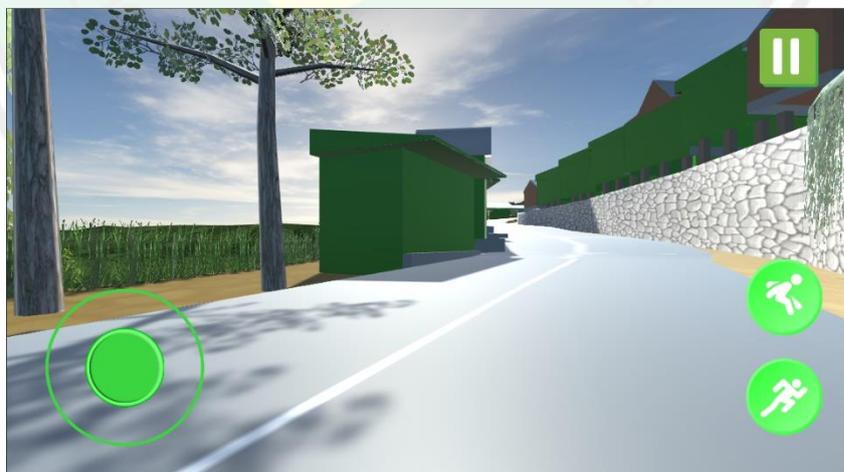
Gambar 3.5 Tampilan pemilihan karakter

Wisatawan : memiliki tugas untuk mengeksplorasi setiap tempat wisata.

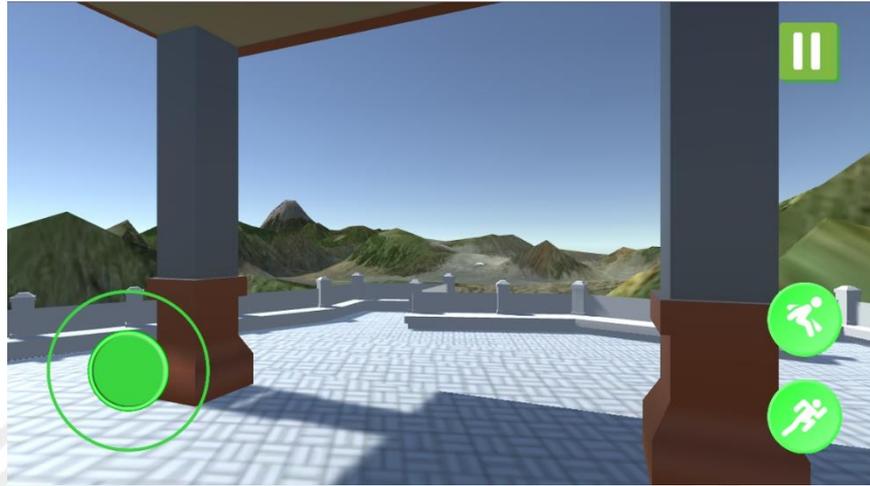
1. Sopir jeep : mengendarai mobil jeep yang di sewa oleh wisatawan.
2. Pedagang toko : menjual barang barang kepada wisatawan.
3. Penjaga gerbang tiket : menjual tiket kepada wisatawan yang akan masuk ke tempat wisata.
4. Jasa penyewaan kuda : memberikan jasa menunggangi kuda bagi para wisatawan.

3.2.5 Gameplay

Pada *Game* wisata bromo, pemain dapat memainkan dua peta lokasi, setelah pemain masuk kedalam lokasi, mereka dapat melakukan eksplorasi ke beberapa tempat. Perspektif kamera pada game ini adalah *FPS(First Perspon Shooter)* dan mengguakan device mobile. Untuk dapat berinteraksi dengan game, telah di sediakan beberapa tombol seperti *analog*, *Run button*, *Jump Button*, dan *Pause Button* yang dapat di tekan untuk menjalankan fungsinya. Untuk visualisasi dapat dilihat pada gambar berikut.



Gambar 3.6 tampilan spawn point wisatawan di lokasi



Gambar 3.7 tampilan saat berada di sunrise



Gambar 3.8 tampilan Spawn point dilokasi kawah bromo.



Gambar 3.9 Pura Luhur Poten, Kawah Bromo

Selain eksplorasi, pemain juga dapat melakukan transaksi yang merupakan fokus utama dalam penelitian ini. *Transaction* pada *Game* ini menggunakan kombinasi metode *Multiplayer* online *Photon Unity Networking* dan metode *Blockchain Ethereum* dimana saat permainan dimulai, segala gerakan dan interaksi yang dilakukan pemain lain akan di sinkronisasikan dengan *Photon network* dan setiap transaksi yang berlangsung akan di proses dengan *Blockchain Ethereum*. Berikut adalah visualisasi saat memproses transaksi.



Gambar 3.10 Transaction UI

Transaction Hash:	0xd68e252dee09dc7de14398948db2f9827b750b729511309c34eef4b7638a5772
Status:	Success
Block:	7407351 75499 Block Confirmations
Timestamp:	11 days 5 hrs ago (Feb-26-2020 04:09:07 PM +UTC)
From:	0xd204d1c8dbf6972b7ebe3561d24e93b824a67ffd
To:	Contract 0xfbbe3461788118f08109fbdbe7d0fc67968420
Tokens Transferred:	From 0xd204d1c8dbf6972... To 0x147e6329fd7d161... For 100 Happy Token (8D)
Value:	0 Ether (\$0.00)
Transaction Fee:	0.000000000000042 Ether (\$0.000000)

Gambar 3.11 Tampilan Proses transaksi yang telah di proses Ethereum

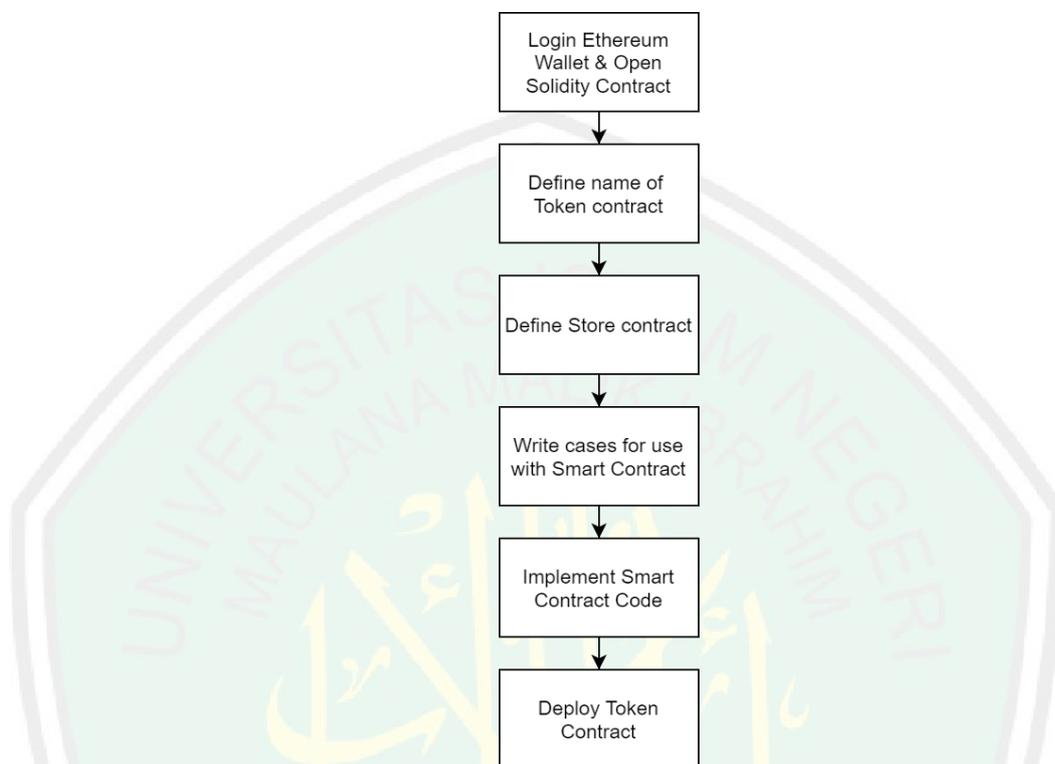
3.3 Pembuatan *Wallet Ethereum*

Proses pertama yang dilakukan pada penelitian ini adalah pembuatan *Wallet* yang akan menampung kontrak token yang akan menjadi mata uang utama pada *Game* penelitian ini. Pembuatan *Wallet* ini juga berfungsi sebagai *Wallet* utama atau parent yang dipegang oleh *Developer* untuk mengatur ekonomi awal dari *Game* tersebut. Pada *Game Blockchain* desentralisasi, *Developer* hanya dapat mengatur ekonomi awal dari *Player* yang baru mendaftar dengan memberikan token awal kepada *Player* baru, setelah itu *Player* dapat mengatur sendiri token yang mereka gunakan untuk bertransaksi kepada *Player* lain maupun langsung kepada *Developer*.

3.4 Token Smart Contract

Smart contract merupakan salah satu kelebihan dari *Blockchain Ethereum*, dimana semua orang dapat memprogram sendiri transaksi yang mereka inginkan, misalnya, sistem akan mengirim sejumlah token yang di inputkan dengan syarat kondisi yang di inginkan pengirim terpenuhi. Selain itu Smart Contract juga dapat membuat suatu mata uang sendiri untuk beberapa keperluan. Dalam penelitian ini, peneliti menggunakan Smart Contract untuk membuat mata uang dalam *Game* yang nantinya digunakan untuk transaksi antar pemain maupun pemain dengan *Developer*. pada proses pembuatan contract token, *Developer* bisa memprogram berapa nilai konversi token yang dibuat berdasarkan nilai jual *Ethereum*, dan untuk membuat token tersebut, *Developer* diharuskan untuk membayar sejumlah Ether yang dibutuhkan untuk membuat token tersebut. Setelah kontrak dibuat, saldo token akan dikirim ke *Wallet* parent yang dikelola *Developer*. sama seperti pembuatannya,

nantinya pemain juga dapat mengkonversi token yang mereka punya menjadi mata uang Ether. Berikut merupakan alur pembuatan token smart contract :



Gambar 3.12 Block Diagram Pembuatan *SmartContract*

3.5 Create child *Ethereum* Account

Proses ini merupakan proses dimana pemain memegang kendali, pemain diwajibkan mendaftar saat akan memainkan *Game* ini. Registrasi ini berfungsi sebagai inisialisasi identitas pemain saat memainkan *Game* ini. selain sebagai inisialisasi, registrasi ini juga langsung terhubung pada registrasi akun *Ethereum*, sehingga pada saat pemain registrasi *Game* wisata bromo, mereka juga sekaligus mendaftar akun *Ethereum* yang nantinya akan digunakan sebagai sistem transaksi desentralisasi *Blockchain*. Untuk proses pendaftaran, pemain dapat langsung mendaftar didalam aplikasi *Game* wisata bromo. Pada penelitian ini, pendaftaran

Game wisata bromo menggunakan gabungan dari *Ethereum* dan Photon Network yang juga berfungsi sebagai metode Real-Time *Multiplayer* Online.

3.6 Multiplayer System

Pada penelitian *Game* wisata Bromo, digunakan platform Photon 2 sebagai networking engine. Photon network memiliki banyak produk cabang untuk berbagai device dan platform, produk yang digunakan dalam penelitian ini adalah PUN (Photon Unity Network). Fungsi dari PUN dalam *Game* ini adalah untuk memungkinkan bermain *Game* wisata bromo dengan *Multiplayer* online dan mensinkronisasikan setiap transaksi yang ada.

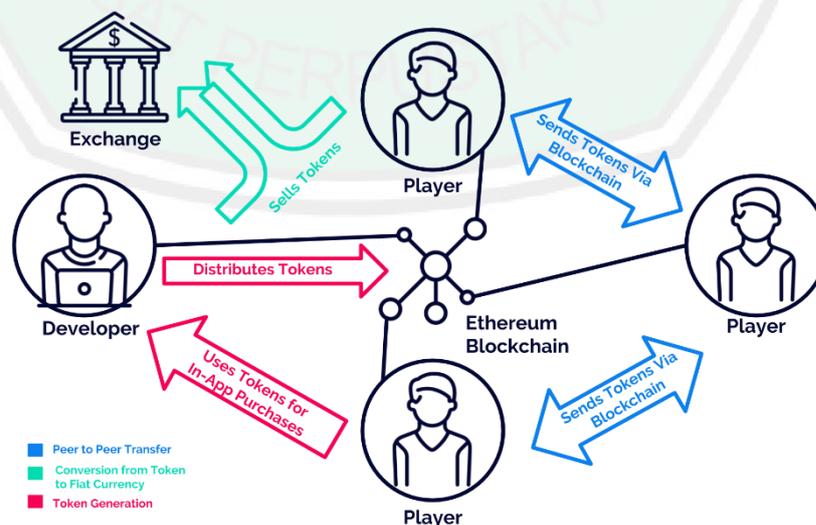
3.7 Transaction

Pada penelitian *Game* wisata Bromo, *Player* yang telah mendaftar dan masuk kedalam *Gameplay* dapat bereksplorasi ke semua peta yang telah disediakan didalam *Game*, selain dapat bereksplorasi, *Player* juga dapat berinteraksi dengan pemain lain dengan fitur transaksi. Dengan transaksi, *Player* dapat membeli suatu item dari *Player* lain, atau dapat membeli jasa *Player* lain. Transaksi pada penelitian ini diproses dengan menggunakan metode *Blockchain* dengan platform *Ethereum*.

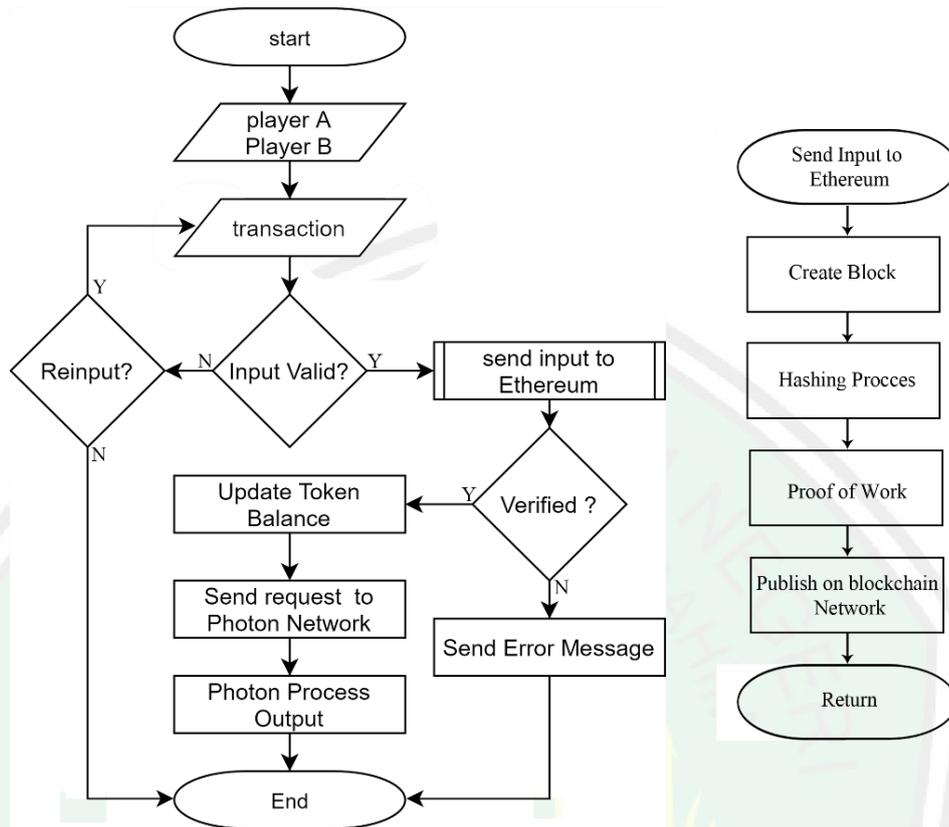
Dengan menggunakan metode desentralisasi *Blockchain*, memungkinkan *Player* dan *Developer* menggunakan sistem ekonomi yang sangat berbeda dengan sistem sentralisasi. Dimana pada ekonomi sistem sentralisasi sepenuhnya dipegang oleh *Developer* dan hanya sedikit fitur transaksi yang bisa diakses oleh *Player*, terlebih pada sistem sentralisasi, item yang dimiliki oleh *Player* tidak sepenuhnya dimiliki *Player*, *Developer* bisa dengan mudah menarik atau menghapus item yang dimiliki *Player* pada sistem ini. Hal ini berbeda dengan sistem desentralisasi yang

dimana setiap item yang dimiliki adalah sepenuhnya dimiliki pemilik, dan juga sistem ekonomi pada sistem ini sangat memungkinkan kontrol penuh atas aset yang mereka miliki. *Player* dapat menjual asset yang mereka miliki untuk mendapatkan mata uang Ether, dan mata uang tersebut juga dapat mereka gunakan untuk segala keperluan didalam maupun diluar *Game* wisata bromo. Untuk kepemilikan Item juga tidak dapat di rubah atau dihapus bahkan oleh *Developer*, hal itu karena sistem desentralisasi yang sangat sulit ditembus. Oleh karena itu dengan menggunakan sistem desentralisasi ini diharapkan juga akan meningkatkan kredibilitas pemain saat akan memainkan *Game* wisata bromo.

Pada proses transaksi antar *Player*, diperoleh beberapa data inputan yang dilakukan oleh pleyer. Dimana dari hasil data transaksi yang dilakukan *Player* didapatkan indentitas *Player*, jenis transaksi, jumlah token yang digunakan, data data tersebut akan digunakan sebagai input yang akan di proses oleh *Ethereum* untuk validasi input dan jika input benar akan dilanjutkan oleh Photon network untuk memproses output. Untuk visualisasi alurnya sebagai berikut.



Gambar 3.13 Ilustrasi alur transaksi



Gambar 3.14 Flowchart transaksi

Pada proses input data transaksi yang dilakukan di dalam *Game* wisata bromo, data tersebut dikirimkan ke *Ethereum* untuk di proses melalui beberapa proses, diantaranya pembuatan block, proses *Hashing*, validasi *Proof-of-Work* dan sampai ke publikasi hasil input ke *Blockchain* network. Untuk detail prosesnya akan dijelaskan di subbab dibawah.

3.7.1 Create Block

Proses pembuatan struktur blok ini dilakukan setelah *Ethereum* mendapatkan inputan dari *Game* wisata bromo. Selain beberapa inputan yang di inputkan oleh *Player*, terdapat beberapa inputan lain yang sudah secara otomatis di inputkan melalui koding program. Inputan tersebut digunakan untuk mengisi struktur block

yang akan di publikasi ke *Ethereum Blockchain* network. Untuk struktur block *Ethereum* dapat di visualisasikan seperti gambar dibawah.



Gambar 3.15 Struktur Blok Ethereum

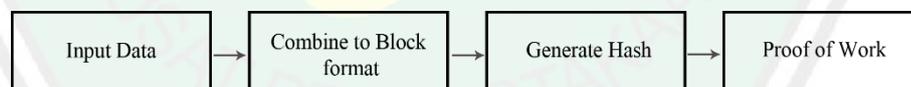
Pada Gambar 3.4 tentang isi block pada *Blockchain Ethereum* dijelaskan bahwa :

1. *Block Index Hash* merupakan nomor index dari block
2. *Block Hash* merupakan *Hash* dari block header
3. *Previous Block Hash* merupakan nilai *Hash* dari block sebelumnya
4. *Transaction Root Hash* merupakan nilai *Hash* yang dibuat setiap eksekusi transaksi
5. *Receipt Root Hash* merupakan nilai *Hash* dari detail transaksi yang dilakukan
6. *TimeStamp* Merupakan Waktu dari pembuatan blok

7. *Difficulty* merupakan tingkat kesulitan untuk menambangnya.
8. *Nonce* merupakan nilai integer yang ditentukan oleh penambang untuk menentukan nilai *Hash* yang valid.
9. *gas limit* adalah batas gas yang di tetapkan untuk block itu
10. *gas used* adalah jumlah gas yang digunakan oleh semua transaksi di blok itu setelah data yang dibutuhkan blok lengkap, selanjutnya data tersebut akan dilakukan proses *Hashing*. Dimana data dari inputan tersebut di rubah menjadi teks kriptografi unik sepanjang 66 kombinasi angka dan huruf.

3.7.2 Hashing Process

Proses *Hashing* merupakan proses yang dilakukan untuk menemukan *Hash* value pada setiap blok yang akan dipublikasikan ke *Blockchain* network. Pada *Ethereum*, proses *Hashing* menggunakan algoritma Keccak256. Keccak256 merupakan standard *Hash* yang digunakan *Ethereum* dimana data inputan yang di terima di rubah menjadi kode kriptografi sepanjang 256 bit. Untuk visualisasi proses *Hashing* dapat dilihat pada gambar dibawah ini.



Gambar 3.16 Proses *Hashing*

Dari proses *Hashing* tersebut akan didapatkan 66 karakter unik dari kombinasi angka dan huruf. Sebagai contoh terdapat kata “ Hello World” yang di proses *Hash* menjadi :

“0a35e4273bbff61e507e6d34c019fc11f1e5b2d3c4fdd85429ea94412a25d038”

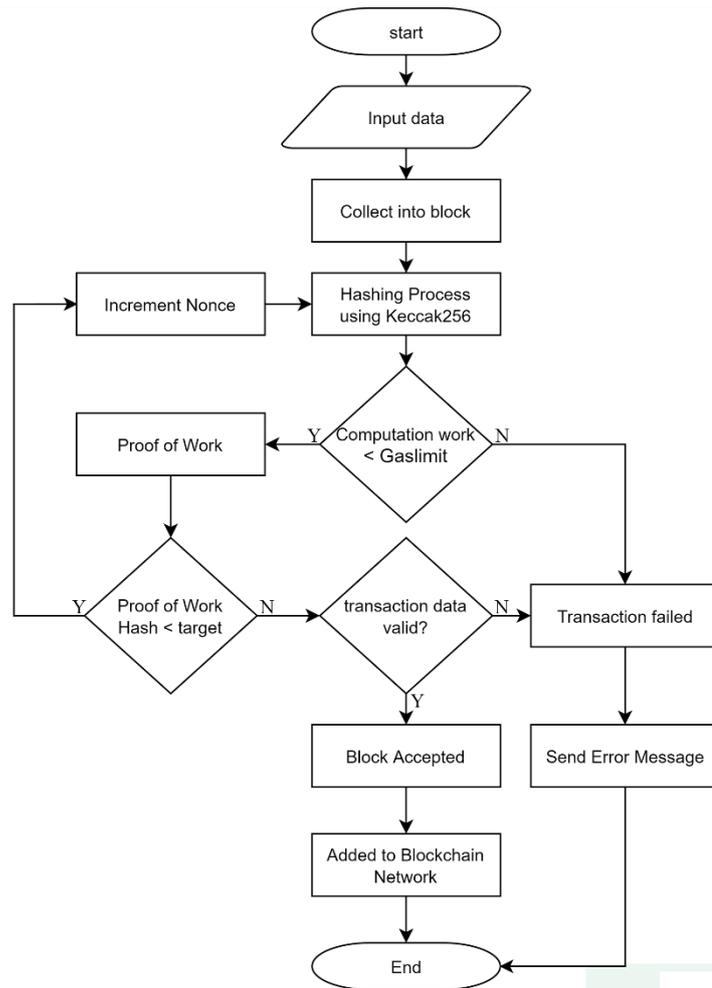
Tetapi dari hasil *Hash* tersebut belum memenuhi persyaratan *Hash* target, yang masih harus melakukan proses mining atau di sebut dengan *Proof-of-Work*. Untuk penjelasan dan proses *Proof-of-Work* akan dijelaskan di subab selanjutnya.

3.7.3 *Proof-of-Work*

Proses *Proof-of-Work* merupakan proses validasi value *Hash* dan data di dalam block yang telah di inputkan sebelumnya. Pada proses *Proof-of-Work* ini akan di lakukan komputasi sebuah bilangan integer yang di sebut dengan *Nonce*, proses ini akan melakukan algoritma Ethas untuk menemukan value *Hash* yang sesuai dengan standard value *Ethereum*.

Pada proses mining atau *Proof-of-Work* ini, para penambang membutuhkan spesifikasi komputer yang tinggi untuk melakukan komputasi *Nonce*. Komputasi *Nonce* setiap tahunya mengalami peningkatan spesifikasi minimum, hal ini disebabkan oleh semakin banyaknya block yang tersebar di jaringan *Ethereum* network yang mengharuskan setiap transaksi yang terjadi akan tercatat di semua bagian block yang tersebar, oleh karena itu spesifikasi yang tinggi diperlukan untuk proses mining.

untuk visualisasi proses *Proof-of-Work* dapat dilihat pada gambar dibawah ini :



Gambar 3.17 Flowchart Proof-of-Work

gambar di atas dijelaskan seperti berikut.

1. *Ethereum* menerima data yang di inputkan oleh *Game* wisata bromo, berupa jenis transaksi, value yang dikirim, tujuan transaksi, gas limit, private key akun.
2. Inputan tersebut selanjutnya akan dilakukan proses kombinasi menjadi bentuk blok.
3. Setelah data dikombinasi kedalam bentuk blok, selanjutnya akan di lakukan proses *Hashing* menggunakan algoritma keccak256.

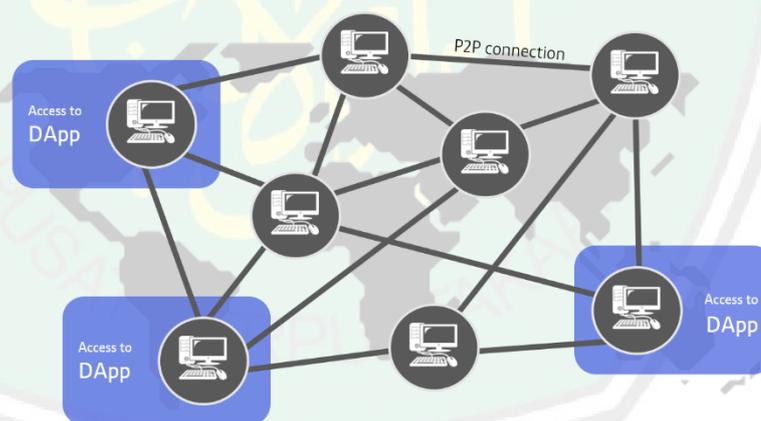
4. Pada proses *Hashing*, akan di cek apakah jumlah komputasi yang dilakukan sudah melebihi nilai gas limit yang sebelumnya di inputkan di dalam baris kode `unity3D`.
5. Jika nilai komputasi yang dilakukan sudah melebihi gas limit yang ditetapkan, maka proses akan di hentikan.
6. Jika komputasi yang dilakukan belum melebihi nilai gas limit, maka proses berlanjut pada *Proof-of-Work*.
7. Pada proses *Proof-of-Work* akan dilakukan proses validasi *Hash* value, apakah sudah mencapai target (*difficult value*).
8. Jika *Hash* value masih belum mencapai target, akan dilakukan proses *increment pseudorandom* dan mengulangi proses *Hashing* hingga *Hash* value mencapai target.
9. Jika value sudah mencapai target, akan dilanjutkan proses *Proof-of-Work* untuk melakukan validasi data inputan apakah sudah valid atau belum.
10. Jika data yang di inputkan tidak valid, maka proses akan dihentikan dan program akan mengirim sebuah pesan kepada sistem, berupa detail error.
11. Jika data yang di inputkan sudah valid, maka proses berlanjut pada proses publikasi blok ke jaringan *Blockchain* network.
12. Selanjutnya blok baru yang sudah di publikasi, akan ditambahkan sebuah rantai, dan data yang ditampung oleh blok baru akan di broadcast ke seluruh jaringan blok.

Selanjutnya blok yang telah tervalidasi akan dilakukan proses publikasi ke jaringan desentralisasi *Blockchain Ethereum* network, yang nantinya akan

ditambahkan sebuah rantai dan data yang di tampung blok baru akan di broadcast ke seluruh jaringan *Blockchain*.

3.7.4 *Blockchain* Network

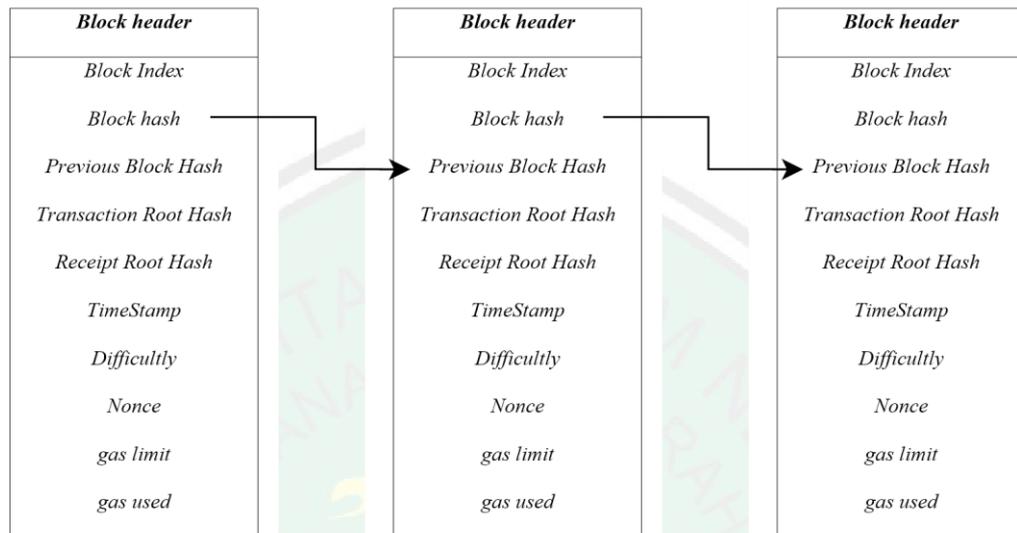
Pada tahapan ini, semua block baru yang telah melewati proses *Proof-of-Work* dan tervalidasi akan dilakukan publikasi ke setiap node yang ada di jaringan desentralisasi *Blockchain Ethereum* network. Setelah block baru di publikasikan ke jaringan network, block baru itu akan membentuk rantai yang saling menghubungkan masing masing block. Rantai ini berfungsi sebagai konsep dasar data sharing desentralisasi peer to peer. Dalam hal ini, data baru yang di tampung oleh block baru akan di broadcast ke seluruh jaringan sehingga data dari setiap block akan di simpan di setiap node yang terdapat pada jaringan desentralisasi *Ethereum*.



Gambar 3.18 *Ilustrasi Blockchain Network*

Pada jaringan *Blockchain* peer to peer *Ethereum*, setiap block yang sudah saling terhubung dalam jaringan akan menyimpan data catatan transaksi yang sudah dilakukan oleh masing-masing *Player*. Masing masing block tersebut saling

terhubung dengan membentuk rantai block yang saling terhubung antar node, seperti pada gambar di bawah ini.

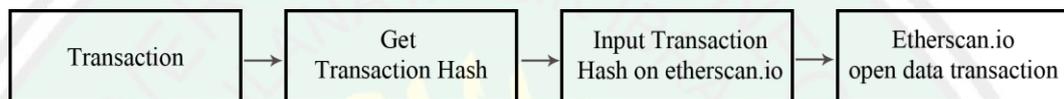


Gambar 3.19 Rantai Blok

3.8 Explore data *Transaction*

Proses ini merupakan proses pencarian data transaksi untuk setiap transaksi yang telah dikirimkan ke platform *Ethereum* menggunakan situs website <https://etherscan.io/>. Etherscan adalah Block Explorer dan Platform Analytics untuk *Ethereum* yang memungkinkan para pelaku transaksi *Blockchain Ethereum* untuk memonitor data transaksi mereka dan menganalisa data transaksi yang berada di jaringan desentralisasi *Ethereum*. Data transaksi yang dapat dilihat melalui situs tersebut bukan hanya data transaksi yang telah valid melewati proses *Proof-of-Work*, tetapi juga data yang sedang melakukan proses *Proof-of-Work*. Untuk melakukan proses data explorer dapat memasukan alamat transaksi yang di dapatkan pada saat transaksi berlangsung. Untuk visualisasi flow *Transaction* data explore dapat dilihat pada gambar dibawah ini.

Pada gambar di atas terlihat proses *Transaction* data explore, langkah pertama sebelum dilakukanya data explore adalah melakukan transaksi yang dilakukan di dalam *Game* wisata bromo. Setelah melakukan proses transaksi dan di konfirmasi oleh *Player*, sistem akan melakukan proses generate *Transaction Hash* dan mengirimkan nomor tersebut melalui notifikasi di dalam *Game*. Nomor *Transaction Hash* tersebut nantinya akan digunakan untuk memonitor status transaksi yang sedang melakukan proses *Proof-of-Work* ataupun yang sudah



Gambar 3.20 proses open data *Transaction*

melalui validasi *Proof-of-Work*. Terdapat tiga status transaksi yang di tampilkan etherscan, yaitu *Transaction* process, *Transaction* succes, dan *Transaction* failed. Yang masing masing memiliki keterangan detail lebih rinci setelah melalui proses *Proof-of-Work*. Untuk visualisasi *Transaction* data explore dapat dilihat pada gambar dibawah ini.

Transaction Hash:	0xd2dcd355696512d0cc7ebaf3c2692180ac8291c481ed9585358a3c1017e3621
Status:	Fail
Block:	7418848 63999 Block Confirmations
Timestamp:	9 days 11 hrs ago (Feb-28-2020 10:09:11 AM +UTC)
From:	0xd204d1c8dbf6972b7ebe3561d24e93b824a67fd
To:	Contract 0xfbbbe3461788118f08109ffdbbe7d0fc67968420 Warning! Error encountered during contract execution [Reverted] ERC-20 Token Transfer Error (Unable to locate corresponding Transfer Event Logs), Check with Sender
Value:	0 Ether (\$0.00)
Transaction Fee:	0.000000000000023 Ether (\$0.000000)

Gambar 3.21 Detail transaksi yang tidak tervalidasi proses *Proof-of-Work*

Transaction Hash:	0xd68e252dee09dc7de14398948db2f9827b750b729511309c34eef4b7638a5772
Status:	Success
Block:	7407351 75499 Block Confirmations
Timestamp:	11 days 5 hrs ago (Feb-26-2020 04:09:07 PM +UTC)
From:	0xd204d1c8dbf6972b7ebe3561d24e93b824a67ffd
To:	Contract 0xfbbbe3461788118f08109ffbdbe7d0fc67968420
Tokens Transferred:	From 0xd204d1c8dbf6972... To 0x147e6329fd7d161... For 100 Happy Token (8D)
Value:	0 Ether (\$0.00)
Transaction Fee:	0.000000000000042 Ether (\$0.000000)

Gambar 3.22 Detail Transaksi yang tervalidasi

3.9 Skenario Pengujian Hasil

Dalam proses pengujian terhadap sistem desentralisasi *Blockchain Ethereum*, akan dilakukan pengujian pemeriksaan pada nilai *Hash* setiap transaksi yang telah dilakukan, apakah transaksi tersebut tervalidasi pada saat proses *Proof-of-Work*. Penelitian ini juga akan melakukan pengujian estimasi *development cost* dan kecepatan transaksi.

3.9.1 Pengujian estimasi *development cost*

Proses pengujian ini dilakukan dengan cara menghitung biaya per transaksi kemudian di bandingkan dengan platform lain. Biaya transaksi pada platform Ethereum bergantung pada jumlah *Gas* yang digunakan pada saat mining, semakin banyak *Gas* yang di pakai maka semakin besar biaya yang di butuhkan, untuk pengukuran biaya yang di keluarkan setiap transaksi dapat di hitung dengan perhitungan sebagai berikut.

$$\text{Gas Price}_{\text{Transaction}} = \text{Gas Used} \times \text{Gas Price}$$

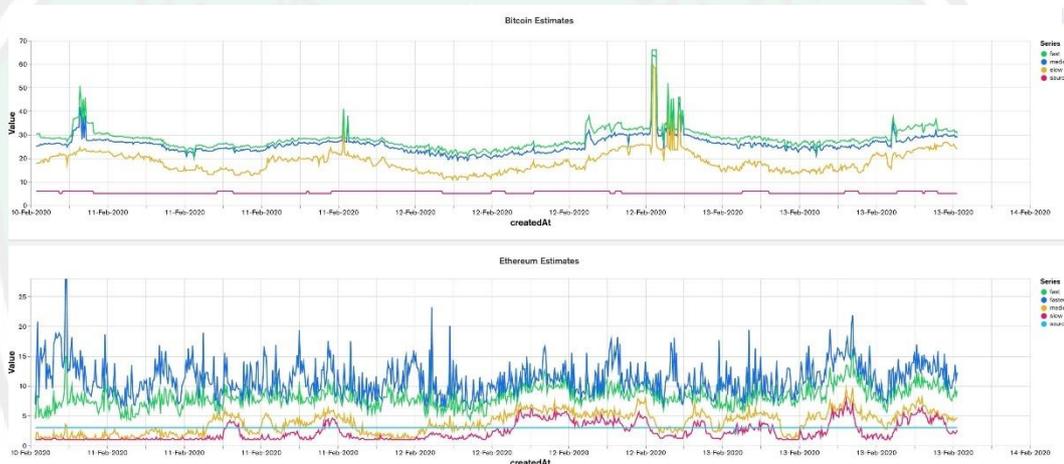
Definisi :

Gas used : jumlah gas yang di gunakan untuk komputasi.

Gas Price : biaya yang harus di keluarkan untuk setiap gas yang digunakan.

Gas Price_{Transaction} : Biaya yang harus dibayarkan untuk proses transaksi.

Rumus yang digunakan platform ethereum untuk menghitung biaya transaksi sama dengan platform Bitcoin, yang membedakan kedua platform tersebut adalah biaya yang harus di keluarkan untuk memproses transaksi. Untuk data grafisnya sebagai berikut.



Gambar 3.23 Grafis estimasi biaya transaksi dalam mata uang (*GWEI*)

Pada grafis di atas terlihat biaya yang dibutuhkan untuk memproses sebuah transaksi. Jika biaya tersebut di rata rata didapatkan biaya sebagai berikut.

Tabel 3.1 data rata rata biaya transaksi selama 10 hari.

	Bitcoin Fees	Ethereum Fees
Fastest	-	11.19
fast	32.876	8.38
Medium	27.649	5.32
Slow	19.426	3.2

Pada transaksi, bitcoin maupun Ethereum memiliki beberapa paket prediksi yang mempengaruhi kecepatan prediksi, diantaranya yaitu Fastest, Fast, Medium, dan Slow. Pada tabel 3.1 terlihat bahwa dalam beberapa paket yang sama, ethereum selalu memiliki biaya transaksi yang lebih murah.

Pada tabel 3.1 terlihat beberapa pilihan metode dikhususkan untuk trader yang berbeda dengan developer. pada developer, mereka dapat memprogram sendiri nilai Gas Price yang di inginkan, Gas limit, bobot data, dan smart contract. Pada penelitian ini, untuk menemukan harga yang paling murah akan dilakukan beberapa percobaan dengan melakukan pemrograman yang akan dibahas di bab 4.

3.9.2 Pengujian Kecepatan proses transaksi

Proses pengujian ini dilakukan dengan cara melihat berapa waktu yang dibutuhkan untuk memproses sebuah transaksi. Beberapa hal yang mempengaruhi kecepatan transaksi diantaranya adalah banyaknya block dalam jaringan, Gas Price, dan bobot data. Semakin banyak block yang ada dalam jaringan blockchain maka semakin lama juga proses *Proof-of-Work*, hal ini untuk memberikan nilai unik sesuai *difficulty* dan menghindari similaritas nilai hash. Gas Price juga mempengaruhi kecepatan prediksi nilai hash, semakin besar price yang ditawarkan, maka semakin cepat juga proses transaksi, dapat dilihat pada tabel yang memperlihatkan semakin besar Gas price yang di tawarkan, juga semakin cepat proses yang dilakukan. Bobot data juga mempengaruhi kecepatan proses transaksi, semakin besar bobot data yang diproses akan semakin lama waktu proses yang dibutuhkan. Pada platform ethereum dan bitcoin selain memiliki selain memiliki

biaya transaksi yang berbeda, kedua platform ini juga memiliki kecepatan proses yang berbeda juga.

Tabel 3.2 estimasi waktu proses transaksi

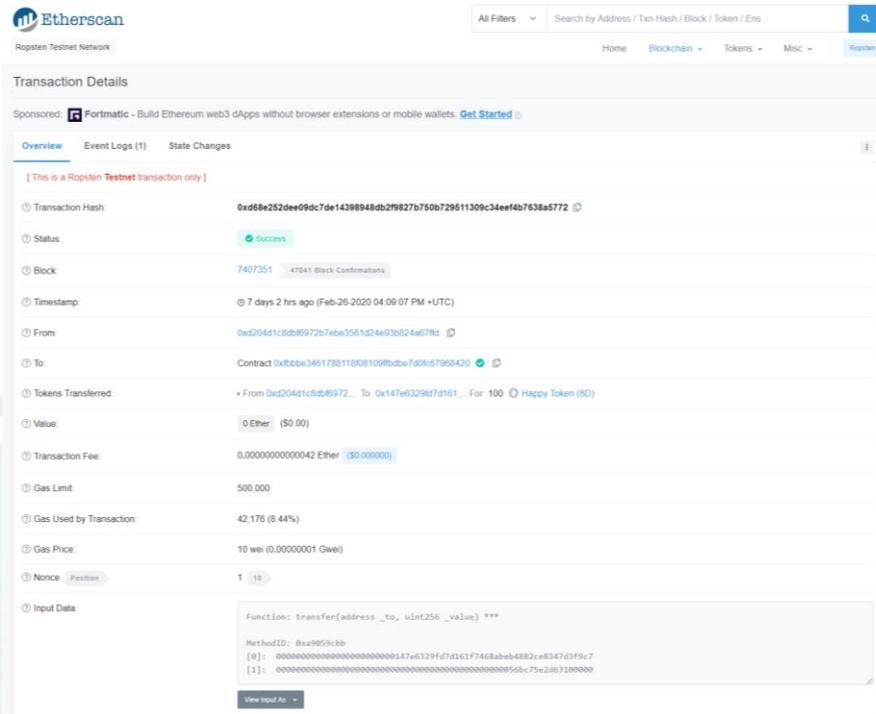
	Bitcoin		Ethereum	
	Fees	Estimated time	Fees	Estimated Time
Fastest	-	-	11.19	30 Second
fast	32.876	15 Minute	8.38	2 Minute
Medium	27.649	30 Minute	5.32	5 Minute
Slow	19.426	1 - 2 Hour	3.2	30 Minute

Tabel di atas terlihat beberapa paket kecepatan yang di tawarkan masing masing platform dengan detail harga dan estimasi waktu yang dibutuhkan untuk memroses data. Dari data tersebut, platform ethereum memiliki estimasi waktu lebih cepat dibandingkan Bitcoin.

Sama seperti pengujian estimasi development cost, akan di lakukan percobaan pemrograman untuk menemukan biaya transaksi yang murah namun dengan kecepatan yang paling efektif. Percobaan ini akan dibahas di bab 4.

3.9.3 Pengujian nilai hash transaksi

Proses pengujian ini dilakukan dengan cara memasukan kode *Hash* transaksi ke laman etherscan untuk mengetahui detail transaksi dan juga keterangan apakah data yang di inputkan berhasil di publikasi ke blockchai network atau tidak. Proses pengecekan saat memasukan kode *Hash* ke laman etherscan dapat dilihat pada gambar dibawah ini.



Gambar 3.24 Detail Hasil Transaksi

Pada gambar di atas, terlihat detail transaksi mulai dari *Hash*, token *Transfer*, timestamp, index block, *Nonce*, dan *gas limit*. Pada penelitian ini akan dilakukan percobaan menggunakan perhitungan input dengan biaya transaksi dan kecepatan paling efisien untuk pengembangan game. Percobaan transaksi akan dibahas pada bab 4.

BAB IV

PENGUJIAN DAN EVALUASI

Pada bab ini akan dijelaskan terkait pengujian dan evaluasi terhadap rancangan sistem yang telah diimplementasikan sebelumnya. Pengujian dan evaluasi ini dilakukan untuk membuktikan keefektifitasan sistem transaksi menggunakan metode *Blockchain* dengan *Platform Ethereum* pada game simulasi wisata Bromo.

4.1 Implementasi

Implementasi dilakukan untuk menerapkan langkah – langkah yang telah ditentukan sebelumnya. Pada implementasi kali ini akan di tampilkan setiap proses yang dijalankan pada program.

4.1.1 *Hardware* yang digunakan.

Perangkat *Hardware* yang digunakan dalam implementasi pengujian pada penelitian game wisata bromo ada lima *Hardware*, hal ini dikarenakan game *Multiplayer* yang membutuhkan pengujian di beberapa device yang berbeda dan juga *Platform* yang berbeda juga untuk mengetahui apakah program benar benar berjalan dengan baik atau tidak. Berikut adalah lima *Hardware* yang digunakan untuk pengujian penelitian game wisata bromo :

1. Device 1.

Tipe	Laptop
Merek	Asus A409FJ
OS	Windows 10 Home
CPU	Intel Core i5-8265U
GPU	Nvidia MX230
RAM	12GB DDR4
Jaringan, Provider	4G - Telkomsel

2. Device 2.

Tipe	Laptop
Merek	Lenovo Ideapad Z370
OS	Windows 10 Pro
CPU	Intel Core i3-2350U
GPU	Intel HD 3000
RAM	6GB DDR3
Jaringan, Provider	4G - Tri

3. Device 3.

Tipe	Laptop
Merek	Hp 14s-cf1047tu
OS	Windows 10 Pro
CPU	Intel celeron n4205

GPU	Intel UHD 610
RAM	4GB DDR4
Jaringan, Provider	4G - Smartfren

4. Device 4.

Tipe	Smartphone
Merek	Xiaomi Redmi Note 7
OS	Android 9.0 Pie
CPU	Snapdragon 660
GPU	Adreno 512
RAM	3GB
Jaringan, Provider	4G - Telkomsel

5. Device 5.

Tipe	Smartphone
Merek	Sony x compact Docomo SO-02J
OS	Android 8.0 Oreo
CPU	Snapdragon 650
GPU	Adreno 510
RAM	3GB
Jaringan, Provider	4G - Smartfren

4.1.2 Software yang digunakan.

Perangkat lunak yang digunakan dalam pembuatan maupun pengujian game wisata bromo adalah sebagai berikut :

Sistem Operasi	Windows 10 Home
Game Engine	Unity3D 2018 v3.14fl
3D Modeller	Blender 3D v2.82
Texture & UI Editor	Photoshop 2019
3D Texture	Substance Painter 2018
Script Editor	Visual Studio 2019
<i>Ethereum Gateway</i>	Metamask

4.1.3 Impelementasi Skenario Game

Impelementasi Skenario Game merupakan tampilan dari aplikasi yang telah di bangun menggunakan metode yang telah di bahas pada BAB 3. Berikut merupakan urutan skenario bagaimana game Wisata Bromo berjalan.

1. Login Menu



Gambar 4.1 Login Menu

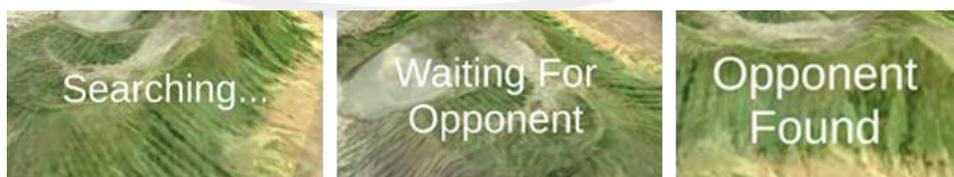
Gambar di atas merupakan tampilan saat sesi login, *Player* diharuskan mengisi nama mereka saat akan memainkan game ini, hal ini di fungsikan sebagai inisialisasi nama pada saat game sedang berjalan.

2. Create & Join *Room*



Gambar 4.2 IU Find Match

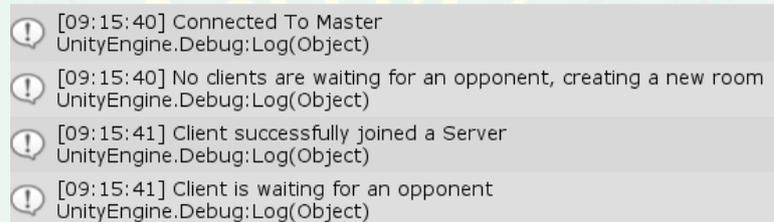
Gambar di atas merupakan tampilan setelah *Player* login kedalam game, UI akan berpindah dari UI login menuju UI Find match. Sistem *Multiplayer* pada game wisata bromo menggunakan auto match making yang berarti saat tombol cari pemain di tekan akan secara otomatis maencari random *Room* yang tersedia di dalam server Photon Network.



Gambar 4.3 dari kiri ke kanan a) finding match b) waiting *Client* c) *Client* found

Terdapat tiga kemungkinan yang akan terjadi saat proses find match berlangsung, diantaranya seperti yang terlihat pada gambar 4.3 diantaranya sebagai berikut :

- I. gambar a) adalah proses pencarian *Random Room* yang ada di server Photon, jika *Room* di temukan maka akan secara otomatis Join kedalam *Room Host*, namun jika *Room* tidak ditemukan maka akan secara otomatis *Player create Room*.
- II. gambar b) adalah proses saat tidak ada *Room* di server dan sistem akan secara otomatis mendaftarkan *Player* menjadi *Host Room* baru di server photon.



```

[09:15:40] Connected To Master
UnityEngine.Debug:Log(Object)
[09:15:40] No clients are waiting for an opponent, creating a new room
UnityEngine.Debug:Log(Object)
[09:15:41] Client successfully joined a Server
UnityEngine.Debug:Log(Object)
[09:15:41] Client is waiting for an opponent
UnityEngine.Debug:Log(Object)
  
```

Gambar 4.4 Log tanda Create *Room* di server berhasil

- III. gambar c) adalah proses saat menemukan *Room* di dalam server photon, pada sesi ini *Client* diharuskan menunggu *Host* untuk memprogram *Gameplay* yang akan di mainkan pada game wisata bromo.

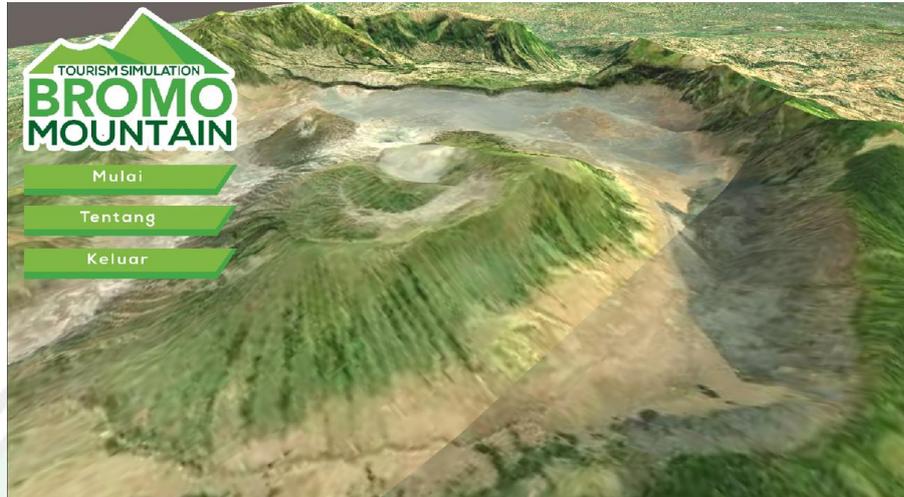


```

[09:17:07] Match is ready to begin
UnityEngine.Debug:Log(Object)
  
```

Gambar 4.5 Log *Client* sudah di temukan

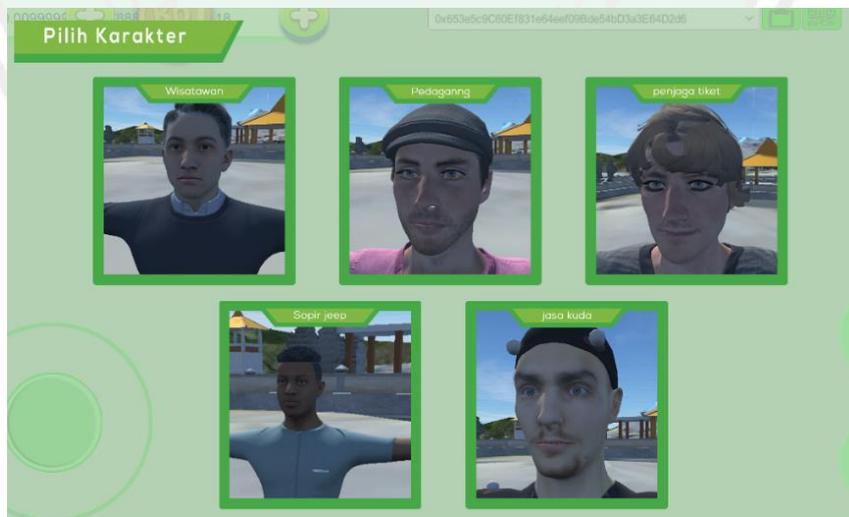
3. Main Menu



Gambar 4.6 Tampilan Main menu

Gambar diatas merupakan tampilan main menu yang dapat di akses oleh *Host* setelah mendapat *Client* pada *Room* mereka. Terdapat tiga menu yang dapat di pilih pada main menu yaitu mulai untuk masuk kedalam pemilihan lokasi *Gameplay*, tentang untuk mengetahui informasi mengenai taman wisata gunung bromo dan keluar untuk keluar dari game.

4. Character Selection.



Gambar 4.7 Tampilan Character Selection

Gambar di atas merupakan tampilan setelah *Host* selesai memilih lokasi *Gameplay*, *Player* yang sudah masuk kedalam *Room* akan diberikan pilihan karakter yang akan dimainkan. Terdapat 5 karakter yang dapat di mainkan dan masing masing memiliki *tuGas* yang berbeda beda diantaranya sebagai berikut :

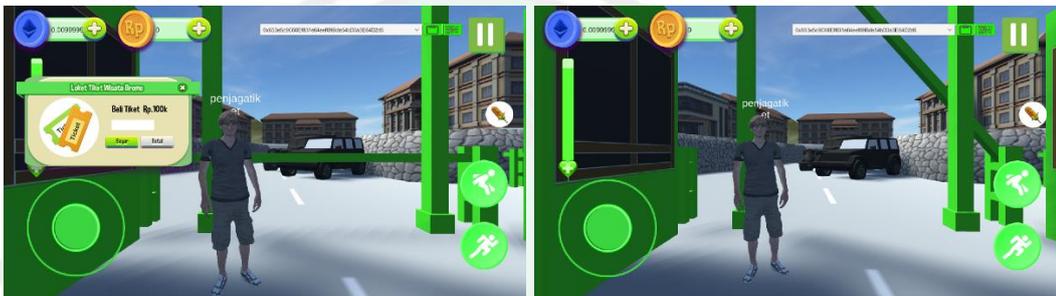
- i. Wisatawan : Memiliki *tuGas* eksplorasi seluruh lokasi pada taman wisata gunung bromo.
- ii. Sopir jeep : mengendarai mobil jeep yang di sewa oleh wisatawan.
- iii. Pedagang toko : menjual barang barang kepada wisatawan.
- iv. Penjaga gerbang tiket : menjual tiket kepada wisatawan yang akan masuk ke tempat wisata.
- v. Jasa penyewaan kuda : memberikan jasa menunggangi kuda bagi para wisatawan.

5. Skenario Area Ticketing



Gambar 4.8 Spawn point wisatawan di area Ticketing

Gambar di atas merupakan tampilan spawn point saat karakter wisatawan dipilih, karakter akan spawn di lokasi depan gapuro gerbang masuk wisata bromo pada sesi ini wisatawan diharuskan membeli tiket agar dapat masuk kedalam taman wisata.



Gambar 4.9 Proses Pembayaran tiket masuk wisata bromo

Gambar di atas adalah tampilan saat *Player* melakukan pembayaran tiket masuk wisata bromo, tanda *Player* sudah melakukan pembayaran dapat dilihat dari palang gerbang yang sudah terbuka.



Gambar 4.10 Proses Pembayaran Sewa Jeep

Setelah membayar tiket, wisatawan dapat melanjutkan perjalanan menuju lokasi selanjutnya, tetapi untuk perjalanan menuju lokasi kawah bromo dan penanjakan 1 tidak dapat di tempuh dengan jalan kaki, oleh karena itu wisatawan

diharuskan untuk menyewa jasa jeep untuk mengantar ke lokasi kawah bromo maupun penanjakan 1.



Gambar 4.11 Simulasi supir jeep mengantar wisatawan ke lokasi wisata

Gambar di atas merupakan tampilan saat wisatawan sudah melakukan pembayaran untuk jasa supir jeep, terlihat pada gambar 4.11 supir jeep mengantarkan wisatawan menuju lokasi kawah bromo.

6. Skenario Area Kawah Bromo



Gambar 4.12 Spawn point pada lokasi kawah bromo

Gambar di atas merupakan spawn point wisatawan yang baru sampai di lokasi kawah bromo.



Gambar 4.13 Pura Luhur Poten Kawah Bromo

Pada lokasi ini *Player* dapat eksplorasi ke lokasi sekitar kawah bromo, *Player* juga bisa berinteraksi dengan *Player* lain seperti pedagang dan jasa penyewaan kuda.



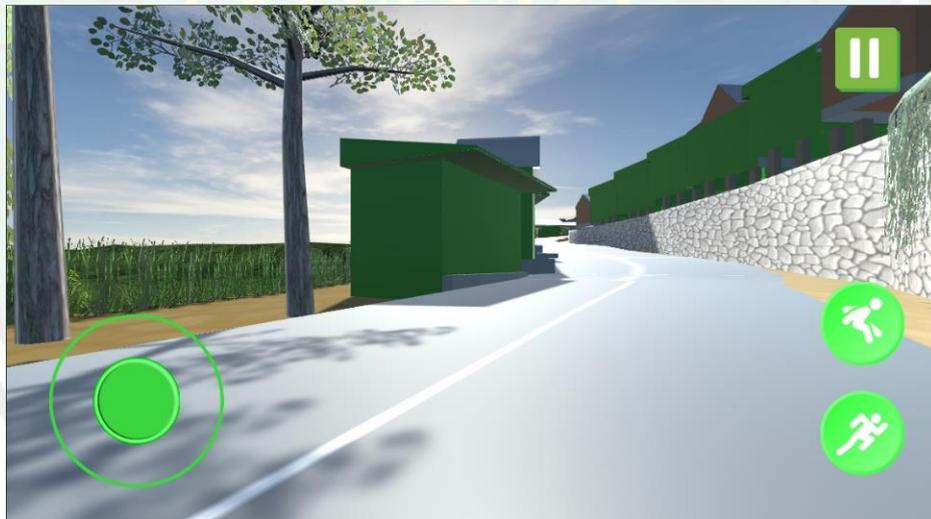
Gambar 4.14 dari kiri kekanan a) proses transaksi dengan pedagang b) proses transaksi dengan jasa penyewaan kuda

Gambar di atas merupakan tampilan *Player* yang memilih karakter pedagang dan jasa penyewaan kuda, wisatawan dapat berinteraksi dengan pedagang untuk membeli barang-barang yang dapat dikonsumsi untuk menambah stamina, wisatawan juga dapat berinteraksi dengan jasa penyewaan kuda untuk dapat bereksplorasi dengan menaiki kuda.



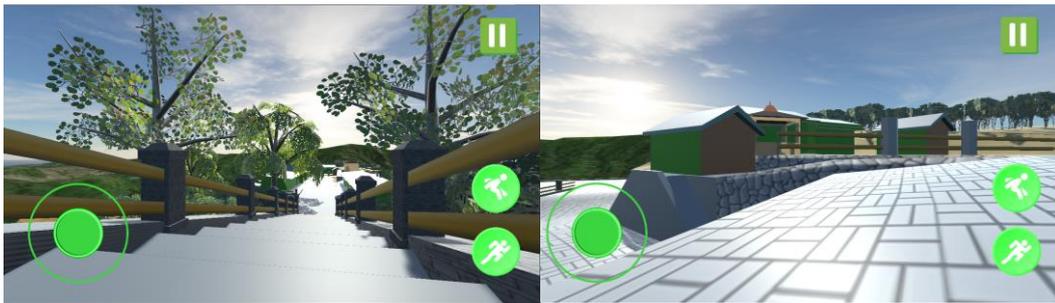
Gambar 4.15 Wisatawan Menunggangi kuda

7. Skenario Area Penanjakan 1



Gambar 4.16 Spawn Point Lokasi Penanjakan 1

Gambar di atas merupakan tampilan spawn point pada lokasi penanjakan 1, pada lokasi ini wisatawan juga dapat bereksplorasi dan juga berinteraksi dengan *Player* lainnya.



Gambar 4.17 Tampilan saat eksplorasi area penanjakan 1

Gambar di atas merupakan tampilan saat melakukan eksplorasi pada lokasi penanjakan 1. Selain pada lokasi penanjakan 1, *Player* juga dapat menuju lokasi sunrise.



Gambar 4.18 Lokasi Sunrise Penanjakan 1

4.2 Pengujian *Multiplayer System*

Pada penelitian *Game* wisata Bromo, digunakan *Platform Photon 2* sebagai networking engine. Photon network memiliki banyak produk cabang untuk berbagai device dan platform, produk yang digunakan dalam penelitian ini adalah PUN (Photon Unity Network). Fungsi dari PUN dalam *Game* ini adalah untuk memungkinkan bermain *Game* wisata bromo dengan *Multiplayer* online dan mensinkronisasikan setiap transaksi yang ada.

4.2.1 Pengujian Fungsionalitas

Proses pengujian ini dilakukan untuk mengetahui apakah Sistem *Multiplayer* pada game Wisata Bromo berjalan sesuai fungsi atau tidak. Pada pengujian ini akan digunakan 5 device yang berbeda dan menggunakan 3 jaringan dengan provider berbeda untuk mengetahui apakah sistem *Multiplayer* dapat berjalan dengan baik atau tidak. Berikut adalah spesifikasi device yang di gunakan dalam penelitian dalam penelitian ini :

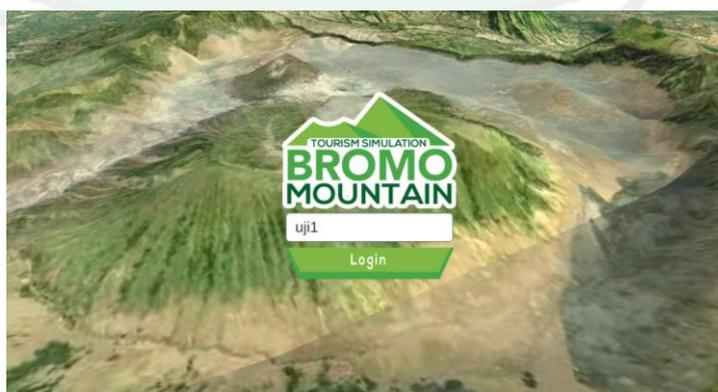
Tabel 4.1 Perangkat Keras yang digunakan untuk pengujian

no	Tipe	Merek	Jaringan
1	Laptop	Asus	Telkomsel
2	Laptop	Lenovo	Tri
3	Laptop	Hp	Smartfren
4	Smartphone	Xiaomi	Telkomsel
5	Smartphone	Sony	Smartfren

Kelima device tersebut akan mealakukan pengetesan fungsi *Multiplayer* berupa Login, Create *Room*, Join *Room*, Character selection, dan Interaction test.

4.2.1.1 Pengujian Login

Pengujian ini dilakukan dengan melakukan percobaan login menggunakan kelima device dengan tiga jaringan berbeda. Percobaan login dimulai dengan memasukan nickname yang akan digunakan didalam game.



Gambar 4.19 Login Menu

Baris program yang digunakan untuk project Game wisata bromo bagian

Login sebagai berikut :

```

21 private void SetUpInputField()
22 {
23     if (!PlayerPrefs.HasKey(PlayerPrefsNameKey)) { return; }
24     string defaultName = PlayerPrefs.GetString(PlayerPrefsNameKey);
25     nameInputField.text = defaultName;
26     SetPlayerName(defaultName);
27 }
28
29 public void SetPlayerName(string name)
30 {
31     continueButton.interactable = !string.IsNullOrEmpty(name);
32 }
33
34 public void SavePlayerName()
35 {
36     string playerName = nameInputField.text;
37     PhotonNetwork.NickName = playerName;
38     PlayerPrefs.SetString(PlayerPrefsNameKey, playerName);
39 }

```

Gambar 4.20 Script Login

Pada pengujian ini, device dinyatakan sudah login dengan masuknya data nickname kedalam server dan tampilan UI pada game berubah dari tampilan login menjadi tampilan search match. Data hasil login dapat dilihat pada tabel berikut. Percobaan dinyatakan sukses jika login berhasil.

Tabel 4.2 Pengujian Login

Percobaan Login	device 1	device 2	device 3	device 4	device 5
1	sukses	sukses	sukses	sukses	sukses
2	sukses	sukses	sukses	sukses	sukses
3	sukses	sukses	sukses	sukses	sukses
4	sukses	sukses	sukses	sukses	sukses
5	sukses	sukses	sukses	sukses	sukses
6	sukses	sukses	sukses	sukses	sukses
7	sukses	sukses	sukses	sukses	sukses
8	sukses	sukses	sukses	sukses	sukses
9	sukses	sukses	sukses	sukses	sukses
10	sukses	sukses	sukses	sukses	sukses

4.2.1.2 Pengujian Create & Join Room

Pengujian ini dilakukan dengan melakukan percobaan membuat *Room* lobby dan percobaan join *Room* yang telah di buat oleh *Host*. Proses ini dapat dilakukan setelah *Player* berhasil login,

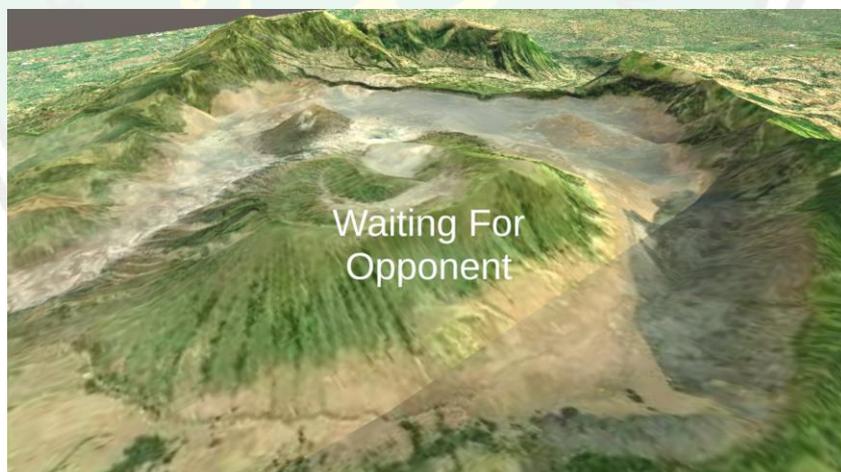
1. Pengujian Create.

Pada game wisata bromo *Room* akan secara otomatis di buat setelah proses pencarian *Room* tidak di temukan. Berikut merupakan baris koding pembuatan *Room*.

```
public override void OnJoinRandomFailed(short returnCode, string message)
{
    Debug.Log("No clients are waiting for an opponent, creating a new room");
    PhotonNetwork.CreateRoom(null, new RoomOptions { MaxPlayers = MaxPlayersPerRoom });
}
```

Gambar 4.21 Script Create Room

Pada pengujian ini dinyatakan sukses jika *Room* berhasil di buat dan terkonfirmasi oleh server Photon. Saat *Room* berhasil dibuat, tampilan di dalam game akan terlihat seperti berikut :



Gambar 4.22 Tampilan setelah create *Room* berhasil

Hal itu menandakan ruangan telah berhasil di buat dan *Host* harus menunggu *Client* untuk masuk kedalam ruangan yang telah di buat. Berikut merupakan data pengujian *Create Room* dengan menggunakan 5 device, pengujian dinyatakan sukses jika device berhasil create *Room*.

Tabel 4.3 Pengujian *Create Room*

Percobaan <i>Create Room</i>	device 1	device 2	device 3	device 4	device 5
1	sukses	sukses	sukses	sukses	sukses
2	sukses	sukses	sukses	sukses	sukses
3	sukses	sukses	sukses	sukses	sukses
4	sukses	sukses	sukses	sukses	sukses
5	sukses	sukses	sukses	sukses	sukses
6	sukses	sukses	sukses	sukses	sukses
7	sukses	sukses	sukses	sukses	sukses
8	sukses	sukses	sukses	sukses	sukses
9	sukses	sukses	sukses	sukses	sukses
10	sukses	sukses	sukses	sukses	sukses

2. Pengujian *Join Room*.

Join Room merupakan proses awal setelah proses login selesai, *Player* akan langsung mencari *Room* yang sudah terdaftar di server. Berikut adalah baris program *Join Room* :

```

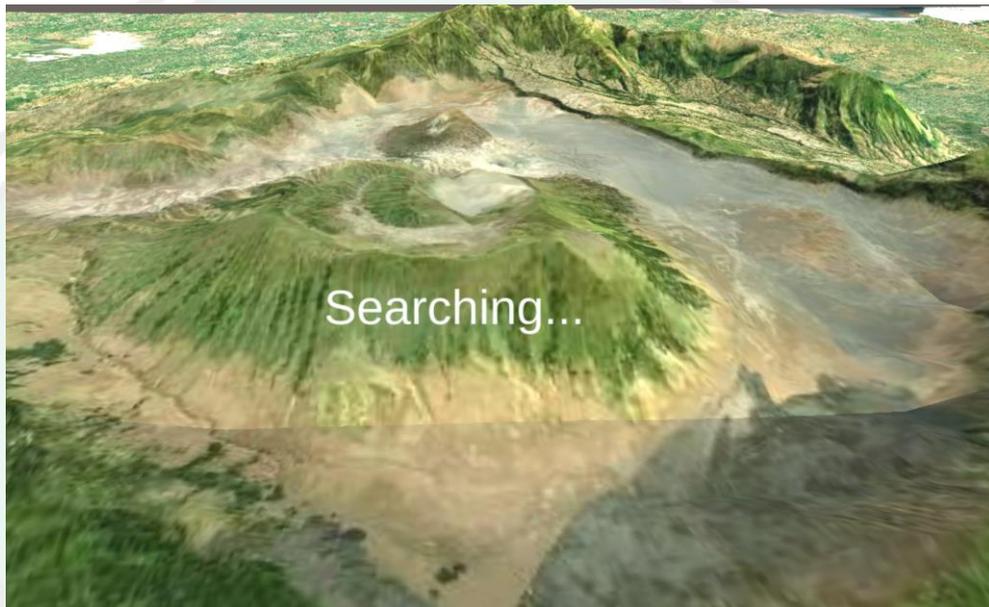
9 references
public override void OnConnectedToMaster()
{
    Debug.Log("Connected To Master");

    if (isConnecting)
    {
        PhotonNetwork.JoinRandomRoom();
    }
}

```

Gambar 4.23 Script Login

Pada game wisata bromo, pencarian *Room* dilakukan dengan Random, hal ini berarti *Room* apa saja yang sudah terdaftar pada server akan juga akan mendapat *Client* random, hal ini membuat waktu proses Lobby menjadi lebih singkat. Tampilan saat game wisata bromo melakukan pencarian *Room* dapat di lihat pada gambar di bawah ini.



Gambar 4.24 Proses Finding Match

Terdapat dua kemungkinan pada saat proses searching selesai di proses, pertama *Room Host* di temukan, kedua *Room Host* tidak ditemukan. Pada saat *Room Host* ditemukan, *Client* akan secara otomatis masuk kedalam *Room Host*, tetapi jika *Room* tidak ditemukan *Client* akan secara otomatis melakukan proses creating *Room*. Tampilan saat *Client* berhasil menemukan *Room* yang sudah terdaftar di server dapat terlihat di bawah ini :



Gambar 4.25 Room ditemukan

Berikut merupakan data pengujian Join *Room* menggunakan 5 Device berbeada, pengujian dinyatakan sukses jika device berhasil join *Room* :

Tabel 4.4 Pengujian Join *Room*

Percobaan Join <i>Room</i>	device 1	device 2	device 3	device 4	device 5
1	sukses	sukses	sukses	sukses	sukses
2	sukses	sukses	gagal	sukses	sukses
3	sukses	sukses	sukses	sukses	sukses
4	sukses	sukses	sukses	sukses	sukses
5	sukses	gagal	sukses	sukses	sukses
6	sukses	sukses	sukses	gagal	sukses
7	sukses	sukses	sukses	sukses	sukses
8	sukses	sukses	gagal	sukses	gagal
9	sukses	sukses	sukses	sukses	sukses
10	sukses	sukses	sukses	sukses	sukses

Terlihat bahwa pada setiap percobaan yang di lakukan hampir semua sukses terhubung dengan *Room* yang sudah di buat sebelumnya. beberapa kemungkinan gagalnya join *Room* yang sudah di buat antara lain seperti berikut :

1. Search time yang bersamaan menyebabkan 2 atau lebih device tidak menemukan *Room* di server dan akhirnya masing masing device create *Room* secara bersamaan.
2. Lokasi IP yang berbeda, photon network memiliki sistem regional untuk match making menyebabkan orang dengan regional berbeda tidak bisa terkoneksi dalam jaringan. Misalnya Ip yang berada di asia dengan eropa tidak dapat terhubung.
3. Koneksi internet yang kurang stabil, atau mungkin tidak ada jaringan internet menyebabkan game tidak dapat terhubung ke sistem.
4. Versi program yang di jalankan berbeda dengan versi aplikasi pada *Room Host*.

4.2.1.3 Pengujian Character Selection

Pengujian Character selection dilakukan dengan melakukan pengecekan terhadap character yang dipilih dan apakah sesuai dengan yang di lihat oleh karakter lain yang berada di dalam *Room* yang sama.

1. Pengujian karakter wisatawan.

Pengujian ini dilakukan dengan melakukan seleksi karakter wisatawan pada sesi Character Selection menggunakan lima device yang berbeda.



Gambar 4.26 Karakter Wisatawan

Berikut adalah data pengujian seleksi karakter wisatawan menggunakan lima device berbeda, kolom Spawn bernilai iya jika karakter muncul di dalam game saat setelah proses pemilihan karakter, kolom tampilan bernilai sesuai jika tampilan karakter sesuai dengan skenario dilihat dari perspektif pemain dan juga pemain lain :

Tabel 4.5 Pengujian Seleksi Karakter Wisatawan

device	karakter	Spawn	tampilan
1	Wisatawan	iya	sesuai
2		iya	sesuai
3		iya	sesuai
4		iya	sesuai
5		iya	sesuai

2. Pengujian karakter Supir Jeep

Pengujian ini dilakukan dengan melakukan seleksi karakter Supir Jeep pada sesi Character Selection menggunakan lima device yang berbeda.



Gambar 4.27 Karakter Supir Jeep

Berikut adalah data pengujian seleksi karakter Supir Jeep menggunakan lima device berbeda, kolom Spawn bernilai iya jika karakter muncul di dalam game saat setelah proses pemilihan karakter, kolom tampilan bernilai sesuai jika tampilan karakter sesuai dengan skenario dilihat dari perspektif pemain dan juga pemain lain :

Tabel 4.6 Pengujian Seleksi Karakter Supir Jeep

device	karakter	Spawn	tampilan
1	Supir Jeep	iya	sesuai
2		iya	sesuai
3		iya	sesuai
4		iya	sesuai
5		iya	sesuai

3. Pengujian Jasa Penyewaan Kuda

Pengujian ini dilakukan dengan melakukan seleksi karakter Jasa penyewaan kuda pada sesi Character Selection menggunakan lima device yang berbeda.



Gambar 4.28 Karakter jasa penyewaan Kuda

Berikut adalah data pengujian seleksi karakter Jasa penyewaan kuda menggunakan lima device berbeda, kolom Spawn bernilai iya jika karakter muncul di dalam game saat setelah proses pemilihan karakter, kolom tampilan bernilai sesuai jika tampilan karakter sesuai dengan skenario dilihat dari perspektif pemain dan juga pemain lain :

Tabel 4.7 Pengujian Seleksi Karakter Jasa penyewaan Kuda

device	karakter	Spawn	tampilan
1	Jasa Kuda	iya	sesuai
2		iya	sesuai
3		iya	sesuai
4		iya	sesuai
5		iya	sesuai

4. Pengujian Penjaga Tiket

Pengujian ini dilakukan dengan melakukan seleksi karakter Jasa penyewaan kuda pada sesi Character Selection menggunakan lima device yang berbeda.



Gambar 4.29 Karakter Penjaga Tiket

Berikut adalah data pengujian seleksi karakter Penjaga Tiket menggunakan lima device berbeda, kolom Spawn bernilai iya jika karakter muncul di dalam game saat setelah proses pemilihan karakter, kolom tampilan bernilai sesuai jika tampilan karakter sesuai dengan skenario dilihat dari perspektif pemain dan juga pemain lain :

Tabel 4.8 Pengujian Seleksi Karakter Penjaga Tiket

device	karakter	Spawn	tampilan
1	Penjaga Tiket	iya	sesuai
2		iya	sesuai
3		iya	sesuai
4		iya	sesuai
5		iya	sesuai

5. Pengujian Pedagang.

Pengujian ini dilakukan dengan melakukan seleksi karakter Jasa penyewaan kuda pada sesi Character Selection menggunakan lima device yang berbeda.



Gambar 4.30 Karakter Pedagang

Berikut adalah data pengujian seleksi karakter Jasa penyewaan kuda menggunakan lima device berbeda, kolom Spawn bernilai iya jika karakter muncul di dalam game saat setelah proses pemilihan karakter, kolom tampilan bernilai sesuai jika tampilan karakter sesuai dengan skenario dilihat dari perspektif pemain dan juga pemain lain :

Tabel 4.9 Pengujian Karakter Pedagang

device	karakter	Spawn	tampilan
1	Pedagang	iya	sesuai
2		iya	sesuai
3		iya	sesuai
4		iya	sesuai
5		iya	sesuai

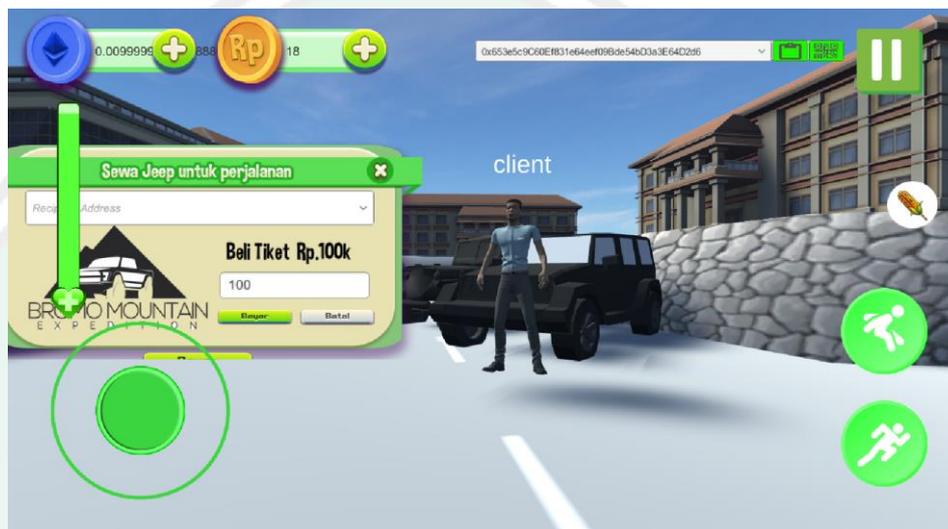
4.2.1.4 Pengujian Interaksi antar *Player*

proses pengujian ini dilakukan dengan cara melakukan percobaan interaksi antar *Player*. Pengujian ini dilakukan untuk mengetahui apakah interaksi yang berlangsung pada masing masing *Player* sesuai dengan skenario game atau tidak.

A. Pengujian interaksi terhadap Supir Jeep.

Pengujian interaksi terhadap Supir Jeep dinyatakan sesuai jika memenuhi skenario sebagai berikut :

1. Muncul Pop up UI opsi transaksi jika didekati oleh *Player* lainnya.



Gambar 4.31 UI Transaksi Sewa Jeep

Pada gambar di atas terlihat bahwa terdapat UI opsi transaksi penyewaan jasa jeep saat *Player* lain mendekati supir jeep. UI tersebut berguna untuk memberikan inputan penyewaan jeep dan akan langsung terkirim ke supir jeep. Berikut adalah data percobaan dengan menggunakan 5 device berbeda. Pengujian dinyatakan sesuai jika tampilan UI yang muncul sesuai dengan Skenario.

Tabel 4.10 Pengujian UI transaksi sewa jeep

device	Skenario	tampilan
1	Tampil UI Transaksi Sewa Jeep	sesuai
2		sesuai
3		sesuai
4		sesuai
5		sesuai

2. Dapat merespon inputan dari *Player* lain.



Gambar 4.32 UI respon inputan transaksi sewa jeep

Pada gambar di atas terlihat UI yang bertuliskan “ Wisatawan melakukan proses pembayaran, konfirmasi jika saldo anda bertambah ” UI tersebut merupakan respon dari inputan yang dilakukan *Player* lain saat melakukan transaksi penyewaan jasa supir jeep. Berikut adalah data percobaan menggunakan lima device berbeda, Pengujian dinyatakan sesuai jika tampilan UI yang muncul sesuai dengan Skenario.

Tabel 4.11 Pengujian UI respon transaksi sewa jeep

device	Skenario	tampilan
1	Tampil UI Respon Transaksi Sewa Jeep	sesuai
2		sesuai
3		sesuai
4		sesuai
5		sesuai

3. Berpindah ke *Gameplay* mengantar penumpang kelokasi kawah bromo ataupun ke lokasi penanjakan 1.



Gambar 4.33 Simulasi supir mengantar wisatawan ke lokasi wisata

Gambar diatas adalah proses simulasi supir jeep mengantar penumpang menuju lokasi kawah bromo. Proses ini berjalan setelah proses pembayaran biaya sewa jasa supir jeep berhasil dibayarkan. Mobil jeep secara otomatis akan spawn pada jalur lintasan menuju lokasi jika proses transaksi berhasil. Berikut merupakan data percobaan menggunakan lima device, kolom Spawn bernilai iya jika mobil jeep muncul di dalam game saat setelah konfirmasi pembayaran, kolom tampilan bernilai sesuai jika tampilan mobil sesuai skenario.

Tabel 4.12 Pengujian Simulasi Mobil Jeep

device	Skenario	Spawn	tampilan
1	Simulasi mengendarai mobil jeep	iya	sesuai
2		iya	sesuai
3		iya	sesuai
4		iya	sesuai
5		iya	sesuai

4. Muncul UI perpindahan lokasi saat sudah memasuki lokasi kawah bromo maupun lokasi penanjakan 1.



Gambar 4.34 UI opsi perpindahan lokasi ke kawah bromo

Gambar diatas merupakan tampilan setelah simulasi supir jeep mengantar penumpang selesai dilakukan. Di akhir jalur akan terdapat pilihan apakah *Player* ingin melanjutkan ke lokasi kawah bromo, jika iya maka *Player* akan langsung masuk ke lokasi kawah bromo.



Gambar 4.35 Spawn Point Kawah Bromo

Setelah *Player* memilih pilihan Iya, karakter akan spawn di lokasi parking jeep kawah bromo. Berikut adalah data percobaan menggunakan lima device

berbeda, kolom spawn bernilai iya jika karakter berpindah ke lokasi kawah bromo setelah menekan button iya pada UI perpindahan lokasi, kolom tampilan bernilai sesuai jika tampilan karakter sama seperti sebelum menaiki mobil jeep.

Tabel 4.13 Pengujian Perpindahan lokasi ke kawah bromo

device	Skenario	Spawn	tampilan
1	Perpindahan Lokasi dari lintasan jeep ke kawah bromo	iya	sesuai
2		iya	sesuai
3		iya	sesuai
4		iya	sesuai
5		iya	sesuai

B. Pengujian interaksi terhadap Pedagang

Pengujian interaksi terhadap Pedagang dinyatakan sesuai jika memenuhi skenario sebagai berikut :

1. Muncul Pop up UI opsi transaksi barang dagangan jika didekati *Player* lainnya.



Gambar 4.36 UI Transaksi dengan pedagang

Gambar di atas adalah tampilan saat pedagang di dekati oleh *Player* lain, akan muncul UI transaksi pembelian barang, pada UI tersebut terdapat 3 barang yang dapat di konsumsi diantaranya ada jagung bakar, sosis bakar, dan mie instan. Item tersebut adalah item yang dapat di konsumsi, dan akan berkurang jika dikonsumsi tetapi menambah stamina *Player*. Berikut adalah data percobaan menggunakan lima device berbeda, Pengujian dinyatakan sesuai jika tampilan UI yang muncul sesuai dengan Skenario.

Tabel 4.14 Pengujian UI transaksi dengan pedagang

device	Skenario	tampilan
1	Tampil UI Transaksi Pembelian Barang	sesuai
2		sesuai
3		sesuai
4		sesuai
5		sesuai

2. Dapat merespon inputan dari *Player* lain.



Gambar 4.37 UI respon inputan transaksi pembelian

Gambar di atas adalah tampilan saat *Player* lain melakukan proses pembelian barang. Pada UI ini pedagang harus memastikan uang yang masuk

muncul sesuai dengan Skenario, kolom jumlah item dinyatakan sesuai jika jumlah item yang ada di inventory sesuai dengan jumlah item yang di beli.

Tabel 4.16 Pengujian Item Masuk kedalam inventory setelah pembelian

device	Skenario	Tampilan	Jumlah Item
1	Item Masuk kedalam inventory	sesuai	sesuai
2		sesuai	sesuai
3		sesuai	sesuai
4		sesuai	sesuai
5		sesuai	sesuai

C. Pengujian interaksi terhadap jasa penyewaan kuda.

1. Muncul Pop up UI opsi transaksi pembayaran penyewaan kuda jika didekati *Player* lainnya.



Gambar 4.39 UI transaksi penyewaan jasa kuda

Gambar di atas merupakan tampilan saat jasa penyewaan kuda didekati oleh *Player* lain, akan muncul UI untuk pembayaran penyewaan kuda. Berikut adalah data percobaan menggunakan lima device berbeda, Pengujian dinyatakan sesuai jika tampilan UI yang muncul sesuai dengan Skenario.

Tabel 4.17 Pengujian UI Pembayaran jasa penyewaan kuda

device	Skenario	tampilan
1	Tampil UI Pembayaran jasa penyewaan kuda	sesuai
2		sesuai
3		sesuai
4		sesuai
5		sesuai

2. Dapat merespon inputan dari *Player* lainnya.

**Gambar 4.40** UI respon inputan pembayaran sewa jasa kuda

Gambar di atas merupakan tampilan saat *Player* sudah melakukan pembayaran, akan muncul UI untuk mengkonfirmasi pembayaran yang sudah dilakukan, apakah sesuai atau tidak, jika sesuai *Player* yang menggunakan karakter jasa penyewaan kuda dapat mengkonfirmasi pembayaran. Berikut merupakan data percobaan menggunakan lima device berbeda, Pengujian dinyatakan sesuai jika tampilan UI yang muncul sesuai dengan Skenario.

Tabel 4.18 Pengujian UI respon inputan pembayaran sewa jasa kuda

device	Skenario	tampilan
1	Tampil UI konfirmasi pembayaran jasa penyewaan kuda	sesuai
2		sesuai
3		sesuai
4		sesuai
5		sesuai

3. *Player* lain dapat menaiki kuda jika pembayaran berhasil.

**Gambar 4.41** tampilan wisatawan menaiki kuda

Gambar diatas merupakan tampilan saat *Player* sudah melakukan pembayaran dan sudah di konfirmasi oleh jasa penyewaan kuda, *Player* akan secara otomatis naik keatas kuda. Pada sesi ini wisatawan dapat berkeliling kawah bromo tanpa mengurangi stamina sama skali dan dapat merjalan lebih cepat. *Player* dapat mensudahi sesi ini dengan menekan tombol turun yang berfungsi untuk *Player* agar dapat turun dari kuda, berikut adalah tampilan saat *Player* menekan tombol turun.



Gambar 4.42 Wisatawan Turun Dari kuda

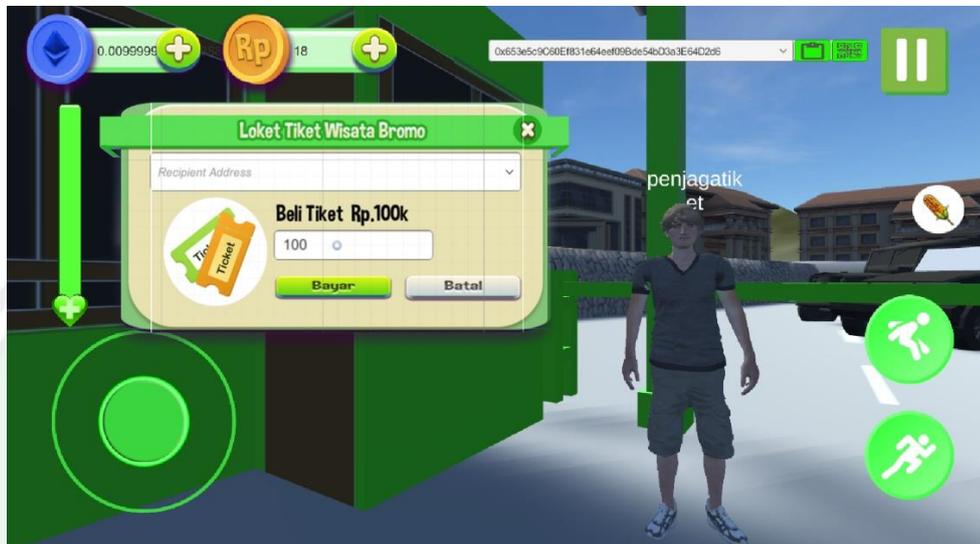
Setelah *Player* turun dari kuda, *Player* dapat berjalan dengan bebas tanpa harus di arahkan oleh pawang kuda lagi. Berikut merupakan data percobaan menaiki kuda dan turun dari kuda menggunakan 5 device berbeda, kolom naik kuda bernilai bisa jika wisatawan bisa menaiki kuda, kolom eksplorasi bernilai bisa jika pergerakan kuda di karakter wisatawan sama seperti pergerakan karakter jasa penyewaan kuda, turun kuda bernilai bisa jika *Player* dapat turun dari kuda.

Tabel 4.19 Pengujian Simulasi wisatawan menaiki kuda

device	Skenario	Naik kuda	Eksplorasi	Turun kuda
1	Berkuda	Bisa	Bisa	Bisa
2		Bisa	Bisa	Bisa
3		Bisa	Bisa	Bisa
4		Bisa	Bisa	Bisa
5		Bisa	Bisa	Bisa

D. Pengujian interaksi terhadap penjaga tiket.

1. Muncul Pop up UI opsi transaksi pembayaran tiket jika didekatii oleh *Player* lainnya.



Gambar 4.43 UI Transaksi Pembelian Tiket masuk wisata Bromo

Gambar di atas merupakan tampilan saat *Player* mendekati penjaga tiket, akan muncul Pop up UI transaksi pembelian tiket masuk wisata bromo. Berikut adalah data percobaan menggunakan lima device berbeda, Pengujian dinyatakan sesuai jika tampilan UI yang muncul sesuai dengan Skenario.

Tabel 4.20 Pengujian UI Transaksi Tiket masuk wisata bromo

device	Skenario	tampilan
1	Tampil UI transaksi tiket masuk wisata bromo	sesuai
2		sesuai
3		sesuai
4		sesuai
5		sesuai

2. Dapat merespon inputan dari *Player* lainya.



Gambar 4.44 UI respon Pembayaran tiket masuk wisata bromo

Gambar di atas merupakan tampilan saat *Player* lain melakukan pembayaran tiket masuk wisata bromo, di layar penjaga tiket akan muncul tampilan UI seperti ini. Berikut adalah data percobaan skenario ini menggunakan lima device berbeda, Pengujian dinyatakan sesuai jika tampilan UI yang muncul sesuai dengan Skenario .

Tabel 4.21 Pengujian UI respon pembelian tiket masuk wisata bromo

device	Skenario	tampilan
1	Tampil UI konfirmasi pembayaran tiket masuk wisata bromo	sesuai
2		sesuai
3		sesuai
4		sesuai
5		sesuai

3. Palang terbuka jika pembayaran berhasil dan dikonfirmasi oleh penjaga tiket.



Gambar 4.45 Tampilan palang gerbang masuk wisata terbuka

Gambar di atas merupakan tampilan saat *Player* sudah melakukan pembayaran tiket dan sudah dikonfirmasi oleh penjaga tiket, akan terlihat palang masuk kedalam tempat wisata bromo yang awalnya tertutup menjadi terbuka. Pada saat ini *Player* dapat melanjutkan perjalanan menuju destinasi selanjutnya dengan melewati palang tersebut.



Gambar 4.46 Wisatawan dapat melewati palang yang terbuka

Berikut merupakan data percobaan skenario pembayaran tiket dan melewati palang pembatas menggunakan lima device berbeda.

Tabel 4.22 Pengujian skenario Pembayaran Tiket masuk wisata bromo

device	Skenario	Status Pembayaran	Status pintu palang	Player dapat lewat
1	Pembayaran tiket masuk wisata bromo	Berhasil	Terbuka	Bisa
2		Berhasil	Terbuka	Bisa
3		Berhasil	Terbuka	Bisa
4		Berhasil	Terbuka	Bisa
5		Berhasil	Terbuka	Bisa

4.2.1.5 Pengujian Fitur Chat

Pengujian ini dilakukan untuk mengetahui apakah fitur *Chat* pada game wisata bromo dapat berjalan sesuai dengan fungsinya. fitur *Chat* pada game wisata bromo berfungsi sebagai sarana komunikasi antar *Player* yang berada didalam satu *Room*. pengujian fitur *Chat* dilakukan dengan melakukan uji coba mengirim pesan dengan menggunakan lima device berbeda dan jaringan dengan provider yang berbeda juga. Berikut adalah gambar pengujian fitur *Chat* pada game wisata bromo.



Gambar 4.47 Tampilan awal kolom *Chat* setelah semua *Player* login



Gambar 4.48 percobaan *Chat* wisatawan dan Penjaga Tiket



Gambar 4.49 Uji coba *Chat* dari karakter jasa supir jeep



Gambar 4.50 Percobaan *Chat* dari Karakter Jasa Penyewaan kuda



Gambar 4.51 Uji coba fitur *Chat* dari karakter pedagang

Pada beberapa gambar di atas terlihat pengujian fitur *Chat* dengan menggunakan lima device berbeda, provider jaringan berbeda, dan karakter yang berbeda, dan didapatkan hasil sebagai berikut, nick name bernilai benar jika nama sama dengan inputan saat login, *Chat* bernilai terkirim jika pesan yang di tulis dapat terkirim di server, *Chat* diterima bernilai diterima jika *Player* lain menerima pesan yang baru dikirim, isi *Chat* bernilai sesuai jika isi *Chat* yang dikirim sama dengan isi *Chat* yang di terima oleh *Player* lain.

Tabel 4.23 Pengujian Fitur *Chat* menggunakan 5 device berbeda

Uji	device	karakter	Nick Name benar	<i>Chat</i> Terkirim	<i>Chat</i> Diterima <i>Player</i> lain	Isi <i>Chat</i> Sesuai
1	1	wisatawan	benar	Terkirim	Diterima	Sesuai
2		penjaga tiket	benar	Terkirim	Diterima	Sesuai
3		supir jeep	benar	Terkirim	Diterima	Sesuai
4		pedagang	benar	Terkirim	Diterima	Sesuai
5		jasa sewa kuda	benar	Terkirim	Diterima	Sesuai
6	2	wisatawan	benar	Terkirim	Diterima	Sesuai
7		penjaga tiket	benar	Terkirim	Diterima	Sesuai
8		supir jeep	benar	Terkirim	Diterima	Sesuai
9		pedagang	benar	Terkirim	Diterima	Sesuai
10		jasa sewa kuda	benar	Terkirim	Diterima	Sesuai

11	3	wisatawan	benar	Terkirim	Diterima	Sesuai
12		penjaga tiket	benar	Terkirim	Diterima	Sesuai
13		supir jeep	benar	Terkirim	Diterima	Sesuai
14		pedagang	benar	Terkirim	Diterima	Sesuai
15		jasa sewa kuda	benar	Terkirim	Diterima	Sesuai
16	4	wisatawan	benar	Terkirim	Diterima	Sesuai
17		penjaga tiket	benar	Terkirim	Diterima	Sesuai
18		supir jeep	benar	Terkirim	Diterima	Sesuai
19		pedagang	benar	Terkirim	Diterima	Sesuai
20		jasa sewa kuda	benar	Terkirim	Diterima	Sesuai
21	5	wisatawan	benar	Terkirim	Diterima	Sesuai
22		penjaga tiket	benar	Terkirim	Diterima	Sesuai
23		supir jeep	benar	Terkirim	Diterima	Sesuai
24		pedagang	benar	Terkirim	Diterima	Sesuai
25		jasa sewa kuda	benar	Terkirim	Diterima	Sesuai

4.2.2 Pengujian *Latency*

Pengujian ini dilakukan untuk mengetahui apakah *Latency* atau jeda waktu yang dibutuhkan untuk pengiriman data sesuai dengan standar *Latency* game online real time. Standar yang penulis gunakan pada penelitian ini di ambil dari data website (<https://www.bandwidthplace.com/>) yang memiliki data standar sebagai berikut :

Tabel 4.24 Data Rating *Latency* untuk Game Online

No	<i>Latency</i> (ms)	rate
1	<20	Excelent
2	20 - 100	Very good
3	100 - 150	Good
4	150 - 200	Still Enjoyable
5	>200	Lag

Tabel di atas menunjukkan data rating *Latency* untuk standar game online, terlihat pada *Latency* di bawah 20ms mendapat rating excelent yang berarti hampir seperti tanpa jeda saat memainkan game, pada *Latency* 20 – 100ms memiliki rating

very good karena waktu jeda yang masih singkat, 100 – 150ms juga masih dapat menyesuaikan dengan *Player* yang memiliki rating *Latency* di atasnya, 150 – 200ms sedikit jeda namun tetap playable jika di sandingkan dengan *Player* yang memiliki rating *Latency* di atasnya, namun jika *Latency* sudah berada di atas 200, akan sangat tidak nyaman saat dimainkan, karena jeda pergerakan dengan pemain lain akan sangat terlihat. Berikut adalah data pengujian *Latency* game wisata bromo menggunakan tiga provider berbeda :

Tabel 4.25 Uji coba *Latency*

Provider	Signal	Avg Latency	Rate
Telkomsel	4G	61ms	Very Good
Tri	4G	144ms	Good
smartfren	4G	127ms	Good

Pada tabel di atas terlihat tiga provider berbeda dengan hasil uji coba *Latency* yang telah di rata rata setelah bermain selama satu jam, Dari hasil tersebut dapat di simpulkan jika game wisata bromo mendapatkan *Score Latency* yang playable dengan menggunakan tiga provider berbeda.

4.3 Pengujian *Transaction System*

Pengujian ini dilakukan untuk mengetahui apakah transaksi dengan custom *Ethereum* dapat mencapai target yang di inginkan. Uji coba ini dilakukan dengan melakukan pengujian kecepatan transaksi dan pengujian *Development Cost*.

4.3.1 Pengujian Kecepatan Transaksi

Proses pengujian ini dilakukan dengan cara melihat berapa waktu yang dibutuhkan untuk memroses sebuah transaksi. Beberapa hal yang mempengaruhi kecepatan transaksi diantaranya adalah banyaknya *Block* dalam jaringan, *Gas Price*, dan bobot data. Semakin banyak *Block* yang ada dalam jaringan *Blockchain* maka semakin lama juga proses *Proof-of-Work*, hal ini untuk memberikan nilai unik sesuai *diffuculty* dan menghindari similaritas nilai *Hash*. *Gas Price* juga mempengaruhi kecepatan prediksi nilai *Hash*, semakin besar *Price* yang ditawarkan, maka semakin cepat juga proses transaksi, dapat dilihat pada tabel 4.26 yang memperlihatkan semakin besar *Gas Price* yang di tawarkan, juga semakin cepat proses yang dilakukan. Bobot data juga mempengaruhi kecepatan proses transaksi, semakin besar bobot data yang diproses akan semakin lama waktu proses yang dibutuhkan. Pada *Platform Ethereum* dan *Bitcoin* selain memiliki selain memiliki biaya transaksi yang berbeda, kedua *Platform* ini juga memiliki kecepatan proses yang berbeda juga.

Tabel 4.26 estimasi waktu proses pembuatan *Block*

	<i>Bitcoin</i>		<i>Ethereum</i>	
	<i>Fees</i>	Estimated time	<i>Fees</i>	Estimated Time
Fastest	-	-	11.19	30 Second
fast	32.876	15 Minute	8.38	2 Minute
Medium	27.649	30 Minute	5.32	5 Minute
Slow	19.426	1 - 2 Hour	3.2	30 Minute

Tabel di atas terlihat beberapa paket kecepatan yang di tawarkan masing masing *Platform* dengan detail harga dan estimasi waktu yang dibutuhkan untuk memproses data. Dari data tersebut, *Platform Ethereum* memiliki estimasi waktu lebih cepat dibandingkan *Bitcoin*.

4.3.1.1 Pengujian *Gas Price*

Pengujian kinerja dilakukan dengan melakukan pengetesan percobaan perubahan variabel *Gas Price* dan melakukan percobaan transaksi beberapa kali dan dilakukan perhitungan rata-rata waktu yang di butuhkan untuk memproses transaksi. untuk perhitungan kecepatan rata rata dapat di hitung sebagai berikut :

$$Mean_t = \frac{T1 + T2 + T3 + \dots + TN}{N}$$

Mean_t = Rata-rata waktu yang di butuhkan untuk proses transaksi.

T = Waktu yang di butuhkan untuk memproses.

N = Banyaknya percobaan.

Sebagai percobaan pengecekan estimasi harga yang paling sesuai untuk *Development* game wisata bromo, dilakukan pengetesan dengan variabel biaya 0,001 Gwei, 0,01 Gwei, 0,1 Gwei, 1 Gwei, 3 Gwei, dan 30 Gwei. Nilai 0,0001 Gwei adalah nilai paling minim untuk percobaan ini karena nilai tersebut adalah batas bawah dari *Fee* para *Miner*, jika menggunakan harga di bawah itu akan secara otomatis di tolak oleh sistem *Ethereum*. Data percobaan dapat di lihat sebagai berikut :

Tabel 4.27 Pengujian waktu transaksi berdasarkan perubahan *Gas Price*

<i>Fee</i>	0,001 Gwei	<i>Fee</i>	0,01 Gwei	<i>Fee</i>	0,1 Gwei
Percobaan	waktu(detik)	Percobaan	waktu(detik)	Percobaan	waktu(detik)
1	48	1	40	1	26
2	58	2	39	2	27
3	66	3	45	3	30
4	56	4	47	4	33
5	55	5	39	5	31
6	65	6	41	6	32
7	64	7	44	7	28
8	69	8	50	8	27
9	61	9	40	9	35
10	64	10	46	10	23
Rata-rata	60,6	Rata-rata	43,1	Rata-rata	29,2

<i>Fee</i>	1 Gwei	<i>Fee</i>	3 Gwei	<i>Fee</i>	30 Gwei
Percobaan	waktu(detik)	Percobaan	waktu(detik)	Percobaan	waktu(detik)
1	11	1	10	1	2
2	13	2	2	2	8
3	9	3	4	3	4
4	6	4	7	4	2
5	9	5	8	5	5
6	12	6	6	6	3
7	10	7	5	7	4
8	7	8	4	8	2
9	8	9	9	9	5
10	12	10	8	10	3
Rata-rata	9,7	Rata-rata	6,3	Rata-rata	3,8

Dengan data tersebut terlihat perbandingan waktu berdasarkan tarif yang kita tawarkan pada *Miners*. Terlihat semakin banyak tarif yang kita tawarkan, maka akan semakin cepat juga proses mining yang dilakukan.

4.3.1.2 Pengujian Berdasarkan SPM

Pengujian berdasarkan SPM(Standar Pelayanan Minimal) dilakukan dengan melakukan pemrograman terhadap variabel *Gas Price*, *Gas Limit*, Smart Contract,

yang telah dilakukan pada subbab sebelumnya. data yang di dapatkan sebelumnya akan dibandingkan dengan Standar Pelayanan Minimal. Standar Pelayanan Minimal untuk kecepatan transaksi pada penelitian ini di ambil pada kecepatan transaksi pada jalan tol, sistem transaksi di develop tidak melebihi 7 detik tiap transaksi 1 mobil, standar tersebut akan menjadi acuan game ini untuk kecepatan transaksi menggunakan *Blockchain*. Dari pengujian pada sub bab sebelumnya telah di dapatkan data seperti pada tabel 4.27 , terlihat bahwa hanya ada 2 percobaan yang sesuai dengan standar pelayanan minimal, namun pada kecepatan seperti itu masih belum diketahui apakah dapat benar benar menghemat biaya *Development* dibandingkan dengan *Platform* lainnya. Pengujian *Development Cost* akan di bahas pada bab selanjutnya.

4.3.2 Pengujian Estimasi *Development Cost*

Proses pengujian ini dilakukan dengan cara menghitung biaya per transaksi kemudian di bandingkan dengan *Platform* lain. Biaya transaksi pada *Platform Ethereum* bergantung pada jumlah *Gas* yang digunakan pada saat mining, semakin banyak *Gas* yang di pakai maka semakin besar biaya yang di butuhkan, untuk pengukuran biaya yang di keluarkan setiap transaksi dapat di hitung dengan perhitungan sebagai berikut.

$$Gas\ Price_{Transaction} = Gas\ Used \times Gas\ Price$$

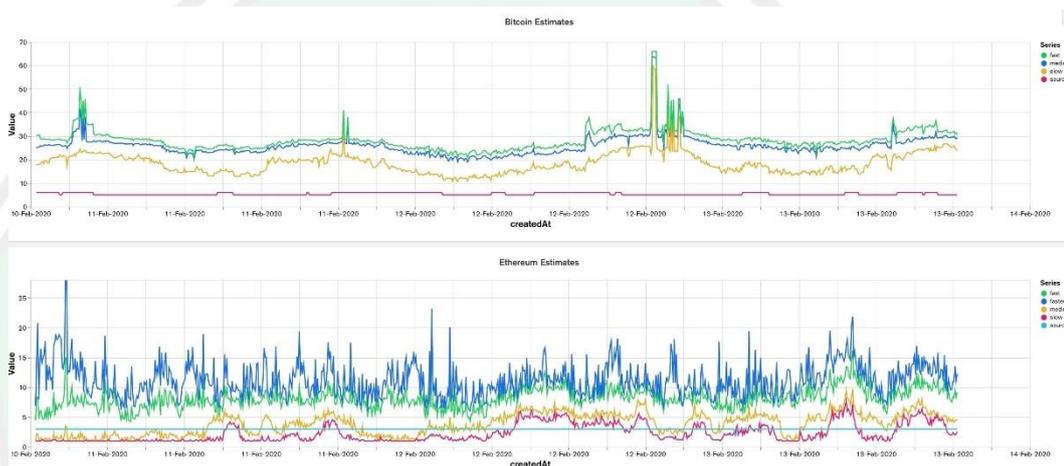
Definisi :

Gas Used : jumlah *Gas* yang di gunakan untuk komputasi.

Gas Price : biaya yang harus di keluarkan untuk setiap *Gas* yang digunakan.

Gas Price_{Transaction} : Biaya yang harus dibayarkan untuk proses transaksi.

Rumus yang digunakan *Platform Ethereum* untuk menghitung biaya transaksi sama dengan *Platform Bitcoin*, yang membedakan kedua *Platform* tersebut adalah biaya yang harus di keluarkan untuk memproses transaksi. Untuk data grafisnya sebagai berikut.



Gambar 4.52 Grafis estimasi biaya transaksi dalam mata uang (*GWEI*)

Pada grafis di atas terlihat biaya yang dibutuhkan untuk memproses sebuah transaksi. Jika biaya tersebut di rata rata didapatkan biaya sebagai berikut.

Tabel 4.28 data rata rata biaya transaksi selama 10 hari.

	<i>Bitcoin Fees</i>	<i>Ethereum Fees</i>
Fastest	-	11.19
fast	32.876	8.38
Medium	27.649	5.32
Slow	19.426	3.2

Pada transaksi, *Bitcoin* maupun *Ethereum* memiliki beberapa paket prediksi yang mempengaruhi kecepatan prediksi, diantaranya yaitu *Fastest*, *Fast*, *Medium*,

dan *Slow*. Pada tabel 4.28 terlihat bahwa dalam beberapa paket yang sama, *Ethereum* selalu memiliki biaya transaksi yang lebih murah.

Pada tabel 4.28 terlihat beberapa pilihan metode dikhususkan untuk trader yang berbeda dengan developer. pada developer, mereka dapat memrogram sendiri nilai *Gas Price* yang di inginkan, *Gas Limit*, bobot data, dan smart contract. Pada penelitian ini, untuk menemukan harga yang paling murah telah dilakukan beberapa percobaan dengan mengatur nilai *Gas Price*, *Gas Limit*, bobot data, dan Smart Contract.

4.3.2.1. Pengujian *Gas Limit*

Pada Tabel 4.27 terlihat waktu yang dibutuhkan untuk proses transaksi berdasarkan biaya yang kita tawarkan kepada *Miners*, Namun pada data di atas merupakan data *Cost* yang kita tawarkan untuk setiap *Gas* yang di gunakan untuk proses mining, sedangkan pada setiap transaksi memiliki *Gas* yang berbeda beda berdasarkan bobot, kesulitan, dan seberapa banyak blok yang sudah ada di jaringan. Hal ini mengharuskan setiap transaksi baru harus memiliki nilai *Hash* yang unik dan tidak boleh sama dengan *Hash* value sebelumnya. oleh karena itu semakin banyak blok yang tersebar dalam jaringan maka juga akan menyebabkan *Gas* yang di butuhkan untuk proses mining juga bertambah banyak, oleh karena itu para developer juga harus memikirkan strategi agar pengeluaran untuk sistem transaksi dapat seminim mungkin, namun memiliki kecepatan yang ideal. Pada penelitian juga dilakukan pengetesan pada *Gas Limit* agar *Fee* transaksi pada game wisata bromo dapat terkendali dan tidak melonjak diluar estimasi *Development Cost*. Data percobaan *Gas Limit* dapat di lihat pada tabel di bawah.

Tabel 4.29 Pengujian Transaksi Dengan *Gas Limit* berbeda

<i>Gas Limit</i>	40.000	<i>Gas Limit</i>	50.000	<i>Gas Limit</i>	60.000
Percobaan	Keterangan	Percobaan	Keterangan	Percobaan	Keterangan
1	sukses	1	sukses	1	sukses
2	sukses	2	sukses	2	sukses
3	sukses	3	sukses	3	sukses
4	gagal	4	sukses	4	sukses
5	gagal	5	sukses	5	sukses
6	gagal	6	sukses	6	sukses
7	gagal	7	gagal	7	sukses
8	gagal	8	gagal	8	sukses
9	sukses	9	sukses	9	sukses
10	gagal	10	sukses	10	sukses

Pada tabel diatas dapat terlihat bahwa *Gas Limit* dengan variabel 60.000 sukses dalam setiap percobaan transaksi. Pada hasil tersebut menjadi standard *Gas Limit* yang digunakan pada game wisata bromo.

4.3.2.2. Pengujian Estimasi *Development Cost*

Pengujian selanjutnya adalah menghitung berapa biaya yang di keluarkan developer dalam proses *Development* maupun pemeliharaan game, hal ini dapat di estimasikan dengan perkiraan transaksi yang di lakukan dalam satu hari, dilanjut dengan satu minggu, satu bulan, hingga satu tahun, lalu dilakukan penyesuaian biaya sesuai dengan data tabel sebelumnya, dan akan di dapatkan beberapa data estimasi *Development Cost*.

Dalam game wisata bromo, biaya setiap transaksi yang di lakukan di dalam game tidak selalu menjadi tanggung jawab developer, melainkan menjadi tanggung jawab sang pelaku transaksi, oleh karena itu developer hanya mengeluarkan biaya transaksi yang minim karena hanya transaksi yang dibutuhkan oleh developer saja

yang akan masuk kedalam perhitungan pengeluaran proses *Development*, sedangkan biaya transaksi antar *Player* akan dibebankan kepada *Player* sendiri, mereka juga diharuskan membayar sejumlah ether untuk transaksi yang mereka lakukan. Sebagai estimasi dalam sehari developer melakukan 10 transaksi, dan untuk proses estimasi selanjutnya dapat dilihat pada perhitungan dibawah ini.

1. Perhitungan *Development Cost* tabel 1

Gas Price : 0,001 Gwei. *Estimated Gas Used* : 55.000.

Gas Limit : 60.000 . *Transaction per year* : 10 x 365 = 3.650.

Transaction Cost = *Gas Price* x *Gas Used*

$$= 0,001 \times 55.000$$

$$= 55 \text{ Gwei}$$

55 Gwei = 0,000000055 Ether = Rp. 0,16

Estimated development cost = *Transaction Cost* x *Transaction per year*

$$= 0,16 \times 3.650.$$

$$= \text{Rp. } 584$$

2. Perhitungan *Development Cost* tabel 2

Gas Price : 0,01 Gwei. *Estimated Gas Used* : 55.000.

Gas Limit : 60.000 . *Transaction per year* : 20 x 365 = 3.650.

Transaction Cost = *Gas Price* x *Gas Used*

$$= 0,01 \times 55.000$$

$$= 550 \text{ Gwei}$$

550 Gwei = 0,0000055 Ether = Rp. 1,6

$$\begin{aligned} \text{Estimated development cost} &= \text{Transaction Cost} \times \text{Transaction per year} \\ &= 1,6 \times 3.650. \\ &= \text{Rp. 5.840} \end{aligned}$$

3. Perhitungan *Development Cost* tabel 3

Gas Price : 0,1 Gwei. *Estimated Gas Used* : 55.000.

Gas Limit : 60.000 . *Transaction per year* : 20 x 365 = 7.300.

$$\begin{aligned} \text{Transaction Cost} &= \text{Gas Price} \times \text{Gas Used} \\ &= 0,1 \times 55.000 \\ &= 5500 \text{ Gwei} \end{aligned}$$

5500 Gwei = 0,0000055 Ether = Rp. 16

$$\begin{aligned} \text{Estimated development cost} &= \text{Transaction Cost} \times \text{Transaction per year} \\ &= 16 \times 3.650. \\ &= \text{Rp. 58.400} \end{aligned}$$

4. Perhitungan *Development Cost* tabel 4

Gas Price : 1 Gwei. *Estimated Gas Used* : 55.000.

Gas Limit : 60.000 . *Transaction per year* : 20 x 365 = 3.650.

$$\begin{aligned} \text{Transaction Cost} &= \text{Gas Price} \times \text{Gas Used} \\ &= 1 \times 55.000 \\ &= 55.000 \text{ Gwei} \end{aligned}$$

55.000 Gwei = 0,000055 Ether = Rp. 160

$$\begin{aligned}
 \text{Estimated development cost} &= \text{Transaction Cost} \times \text{Transaction per year} \\
 &= 160 \times 3.650. \\
 &= \text{Rp. 584.000}
 \end{aligned}$$

5. Perhitungan *Development Cost* tabel 5

$$\begin{aligned}
 \text{Gas Price} &: 3 \text{ Gwei.} & \text{Estimated Gas Used} &: 55.000. \\
 \text{Gas Limit} &: 60.000 . & \text{Transaction per year} &: 20 \times 365 = 7.300.
 \end{aligned}$$

$$\begin{aligned}
 \text{Transaction Cost} &= \text{Gas Price} \times \text{Gas Used} \\
 &= 3 \times 55.000 \\
 &= 165.000 \text{ Gwei}
 \end{aligned}$$

$$165.000 \text{ Gwei} = 0,000165 \text{ Ether} = \text{Rp. 491}$$

$$\begin{aligned}
 \text{Estimated development cost} &= \text{Transaction Cost} \times \text{Transaction per year} \\
 &= 491 \times 3.650. \\
 &= \text{Rp. 1.752.000}
 \end{aligned}$$

6. Perhitungan *Development Cost* tabel 6

$$\begin{aligned}
 \text{Gas Price} &: 30 \text{ Gwei.} & \text{Estimated Gas Used} &: 55.000. \\
 \text{Gas Limit} &: 60.000 . & \text{Transaction per year} &: 20 \times 365 = 7.300.
 \end{aligned}$$

$$\begin{aligned}
 \text{Transaction Cost} &= \text{Gas Price} \times \text{Gas Used} \\
 &= 30 \times 55.000 \\
 &= 1.650.000 \text{ Gwei}
 \end{aligned}$$

$$1.650.000 \text{ Gwei} = 0,00165 \text{ Ether} = \text{Rp. 4.918}$$

$$\begin{aligned}
 \text{Estimated development cost} &= \text{Transaction Cost} \times \text{Transaction per year} \\
 &= 4.918 \times 3.650. \\
 &= \text{Rp. 17.520.000}
 \end{aligned}$$

Pada percobaan perhitungan *Development Cost* di atas di dapatkan estimasi total pengeluaran yang berdasarkan enam tabel data biaya transaksi yang di kalikan dengan jumlah *Gas* yang dibutuhkan. data ringkasan perhitungan estimasi di atas dapat di lihat pada tabel di bawah ini.

Tabel 4.30 Estimasi Pengeluaran

no	<i>Gas Price</i>	<i>Gas Used</i>	<i>Transaction Cost</i>	<i>Daily Cost</i>	<i>Monthly Cost</i>	<i>Annual Cost</i>
1	0,001	55.000	Rp 0,16	Rp 3,2	Rp 96	Rp 584
2	0,01	55.000	Rp 1,6	Rp 32	Rp 960	Rp 5.840
3	0,1	55.000	Rp 16	Rp 320	Rp 9.600	Rp 58.400
4	1	55.000	Rp 160	Rp 3.200	Rp 96.000	Rp 584.000
5	3	55.000	Rp 491	Rp 9.820	Rp 294.600	Rp 1.752.000
6	30	55.000	Rp 4.910	Rp 98.200	Rp 2.946.000	Rp 17.520.000

Dari tabel di atas terlihat estimasi pengeluaran biaya dalam hari, bulan, dan tahun. pada bab sebelumnya mengenai Standar Pelayanan Minimal, hanya ada dua percobaan yang memiliki rata rata kecepatan transaksi sesuai dengan Standar Pelayanan Minimal, yaitu pada percobaan nomor 5 dan 6. Namun dengan dua tarif tersebut masih ada di bawah tarif transaksi *Platform* lainnya, berikut adalah data tentang tarif percobaann penelitian dan tarif pada *Platform* lain.

Tabel 4.31 Perbandingan Biaya dan Estimasi waktu dengan *Platform* lain

<i>Bitcoin</i>		<i>Ethereum</i>		<i>Ethereum Custom</i>	
<i>Fees</i>	Estimated time	<i>Fees</i>	Estimated Time	<i>Fees</i>	Estimated Time
32.876	7 Minute	11.19	30 Second	30	3.8 Second
27.649	15 Minute	8.38	1 Minute	3	6.3 Second

Dari tabel di atas terlihat bahwa hasil penelitian dengan mengcustom *Gas Price*, *Gas Limit*, *Transaction size* dapat lebih menghemat biaya transaksi. data di atas diambil dari API *Bitcoin Fees* (https://Fees.upvest.co/estimate_btc_Fees) dan data *Ethereum* di ambil dari (https://Fees.upvest.co/estimate_eth_Fees).

4.3.3 Pengujian Fungsionalitas Sistem Transaksi

Pengujian ini dilakukan untuk mengetahui apakah implementasi yang dilakukan pada penelitian ini dapat berjalan dengan baik atau tidak. Pengujian dilakukan dengan melakukan transaksi pada karakter yang berbeda beda dengan menggunakan config custom *Ethereum*. Berikut adalah data percobaanya.

1. Transaksi dengan Penjaga Tiket.

Transaction Hash:	0x4378cb4f93a1d5489d6548b4c41600af197252c33f8424646c47a5d569e4c447
Status:	Success
Block:	7930533 2 Block Confirmations
Timestamp:	39 secs ago (May-19-2020 07:13:56 AM +UTC)
From:	0x653e5c9c60ef831e64eef09bde54bd3a3e64d2d6
To:	Contract 0x49c725c9e49bd3aa26e56eaaa0354c4f502bcf77
Tokens Transferred:	From 0x653e5c9c60ef831... To 0xb935a3fd31ac17c... For 100 Mata Uang Br... (Rp.)
Value:	0 Ether (\$0.00)
Transaction Fee:	0.000126528 Ether (\$0.000000)
Gas Limit:	500,000
Gas Used by Transaction:	42,176 (8.44%)
Gas Price:	0.000000003 Ether (3 Gwei)
Nonce	7 9

2. Transaksi dengan Supir Jeep.

Transaction Hash:	0x0edc9fc9b68928e4aacdc7e196f12a584440ded5f65a13a7ae882393fa0af3cd
Status:	Success
Block:	7930548 1 Block Confirmation
Timestamp:	22 secs ago (May-19-2020 07:18:18 AM +UTC)
From:	0x653e5c9c60ef831e64eef09bde54bd3a3e64d2d6
To:	Contract 0x49c725c9e49bd3aa26e56eaaa0354c4f502bcf77
Tokens Transferred:	From 0x653e5c9c60ef831... To 0x70a962e367c09f9... For 100 Mata Uang Br... (Rp.)
Value:	0 Ether (\$0.00)
Transaction Fee:	0.000126528 Ether (\$0.000000)
Gas Limit:	500,000
Gas Used by Transaction:	42,176 (8.44%)
Gas Price:	0.000000003 Ether (3 Gwei)
Nonce	8 12

3. Transaksi dengan Pedagang.

Transaction Hash:	0xb3ae8e654dc9a0613bd91c722e8f91decdec3c7a4d77075792393bf38e9604b3
Status:	Success
Block:	7930570 1 Block Confirmation
Timestamp:	24 secs ago (May-19-2020 07:22:55 AM +UTC)
From:	0x653e5c9c60ef831e64eef09bde54bd3a3e64d2d6
To:	Contract 0x49c725c9e49bd3aa26e56eaaa0354c4f502bcf77
Tokens Transferred:	From 0x653e5c9c60ef831... To 0x3eac11cfd6125a... For 67 Mata Uang Br... (Rp.)
Value:	0 Ether (\$0.00)
Transaction Fee:	0.000171528 Ether (\$0.000000)
Gas Limit:	500,000
Gas Used by Transaction:	57,176 (11.44%)
Gas Price:	0.000000003 Ether (3 Gwei)
Nonce	9 7

4. Transaksi dengan Jasa Penyewaan kuda.

Transaction Hash:	0x92596bf19f24e76f6fed8e01eda762f47620776db9251a6506c204727017e6f8
Status:	Success
Block:	7930591 2 Block Confirmations
Timestamp:	47 secs ago (May-19-2020 07:28:39 AM +UTC)
From:	0x653e5c9c60ef831e64eef09bde54bd3a3e64d2d6
To:	Contract 0x49c725c9e49bd3aa26e56eaaa0354c4f502bcf77
Tokens Transferred:	From 0x653e5c9c60ef831... To 0x3eac11cfd6125a... For 100 Mata Uang Br... (Rp.)
Value:	0 Ether (\$0.00)
Transaction Fee:	0.000126528 Ether (\$0.000000)
Gas Limit:	500,000
Gas Used by Transaction:	42,176 (8.44%)
Gas Price:	0.000000003 Ether (3 Gwei)
Nonce	10 Position 5

Terlihat dari pengujian di atas, 4 transaksi yang dilakukan dengan 4 *Player* yang berbeda memiliki hasil sukses, hal ini membuktikan jika config yang di implementasikan untuk penelitian ini berhasil.

4.4 Integrasi Islam

Keamanan dan kejujuran merupakan sesuatu yang sulit ditemukan sekarang, penipuan, peretasan, *cracking* dan masih banyak lagi kejahatan cyber seiring dengan perkembangan teknologi. hal itu tidak hanya dirasakan oleh developer, bahkan *Player* game juga tidak luput dari kejahatan cyber tersebut. Tidak sedikit dari para *Player* yang kehilangan asset yang sudah mereka beli, dan juga tidak sedikit *Player* yang sudah mengalokasikan sejumlah uang asli mereka kedalam game dan uang mereka hilang. Dalam kejahatan cyber ini juga tidak selalu

dilakukan oleh *hacker* atau *cracker*, ada beberapa kasus penipuan yang melibatkan developer game, mereka dengan sengaja menghapus atau mengambil sebagian saldo uang yang dimiliki *Player*. Hal ini juga menyebabkan kredibilitas *Player* berkurang terhadap semua game, walaupun masih banyak developer game yang jujur. Di dalam al-quran surat Al-Ahzab ayat 23-24, Allah SWT telah memerintahkan kepada manusia untuk berlaku jujur.

مِنَ الْمُؤْمِنِينَ رِجَالٌ صَدَقُوا مَا عَاهَدُوا اللَّهَ عَلَيْهِ فَمِنْهُمْ مَن قَضَىٰ نَحْبَهُ وَمِنْهُمْ مَن يَنْتَظِرُ وَمَا بَدَّلُوا تَبْدِيلًا
لِيَجْزِيَ اللَّهُ الصَّادِقِينَ بِصِدْقِهِمْ وَيُعَذِّبَ الْمُنَافِقِينَ إِن شَاءَ أَوْ يَتُوبَ عَلَيْهِمْ إِنَّ اللَّهَ كَانَ غَفُورًا رَحِيمًا *

Artinya : *“Di antara orang-orang mukmin itu ada orang-orang yang menepati apa yang telah mereka janjikan kepada Allah. Dan di antara mereka ada yang gugur, dan di antara mereka ada (pula) yang menunggu-nunggu dan mereka sedikit pun tidak mengubah (janjinya), (23) agar Allah memberikan balasan kepada orang-orang yang benar itu karena kebenarannya, dan mengazab orang munafik jika Dia kehendaki, atau menerima tobat mereka. Sungguh, Allah Maha Pengampun, Maha Penyayang. (24)”* – (Q.S Al-Ahzab: 23-24)

Dalam tafsir al-Jalalain ayat tersebut memiliki tafsir : Di antara orang-orang Mukmin itu ada orang-orang yang menepati apa-apa yang telah mereka janjikan kepada Allah) yaitu gigih bertahan bersama dengan Nabi saw. (maka di antara mereka ada yang gugur) mati atau terbunuh di jalan Allah (dan di antara mereka ada pula yang menunggu-nunggu) hal tersebut (dan mereka sedikit pun tidak mengubah) janjinya, berbeda halnya dengan orang-orang munafik. (Supaya Allah memberikan balasan kepada orang-orang yang benar itu karena kebenarannya, dan menyiksa orang-orang munafik jika dikehendaki-Nya) seumpamanya Dia

mematikan mereka dalam kemunafikannya (atau menerima tobat mereka. Sesungguhnya Allah adalah Maha Pengampun)(lagi Maha Penyayang) kepada orang yang bertobat kepada-Nya.

Nabi Muhammad shallallahu 'alaihi wa sallam merupakan teladan sempurna untuk kita. Beliau memiliki akhlak atau sifat yang begitu mulia. Beberapa sifat mulia yang dimiliki oleh Nabi Muhammad shallallahu 'alaihi wa sallam antara lain amanah dan jujur. Nabi Muhammad dikenal sebagai pribadi yang jujur, bahkan sejak beliau belum diangkat menjadi nabi. (Baca juga: Kisah Abu Bakar Ash Shiddiq). Jujur, dalam Bahasa Arab dikenal dengan istilah *ash shidqu* atau *shiddiq*, memiliki arti nyata atau berkata benar. Artinya, kejujuran merupakan bentuk kesesuaian antara ucapan dan perbuatan atau antara informasi dan kenyataan. Lebih jauh lagi, kejujuran berarti bebas dari kecurangan, mengikuti aturan yang berlaku dan kelurusan hati. Ada banyak sekali bentuk kejujuran dalam kehidupan kita sehari-hari. Sejak kecil kita pasti telah diajarkan oleh orang tua kita untuk selalu berbuat jujur dan tidak berbohong. Hal ini tentu sesuai dengan ajaran agama Islam yang telah dicontohkan oleh Nabi Muhammad shallallahu 'alaihi sallam sendiri.

Saat ini para developer sedang berlomba-lomba untuk terus mengembangkan keamanan game mereka. Salah satu dari strategi yang saat ini digunakan untuk meningkatkan keamanan game mereka adalah Teknik *Platform* desentralisasi menggunakan *Ethereum Blockchain*. Teknik ini sangat efektif untuk meningkatkan keamanan Game karena struktur desentralisasi *Blockchain* membuat banyak blok yang masing-masing blok telah terenkripsi. Hal ini membuat peretasan, duplikasi, dan mencuri kunci menjadi sangat sulit. Selain keamanan,

Blockchain juga memungkinkan pemain untuk bertransaksi tanpa batas, karena teknologi *Blockchain* tidak memerlukan pengawasan, atau tidak memerlukan pihak ketiga untuk melakukan transaksi, mereka bisa langsung bertransaksi tanpa harus ada campur tangan pihak ketiga. Bagi developer Teknik *Blockchain* juga dapat menjadi strategi pemasaran asset item game dengan membatasi jumlahnya. Sebagai contoh, game CryptoKitties menggunakan teknologi ini untuk menyimpan kucing lucu yang dapat diperoleh dan ditukar pengguna. perusahaan itu disambut dengan kesuksesan tak terduga, Asset tersebut menjadi kucing terlaris dengan harga \$ 115.000. Bukti terbatasnya pasokan memicu tingginya permintaan (Katalyse 2018).

Selain keamanan, para developer juga saling berinovasi dalam menemukan teknologi yang dapat menghemat biaya pengeluaran produksi maupun pemeliharaan game. Beberapa developer yang menggunakan teknologi *Blockchain* beberapa ada yang mengeluhkan biaya transaksi yang mahal jika digunakan berulang kali selama beberapa tahun. Oleh karena itu pada penelitian ini akan dilakukan riset tentang biaya produksi dan pemeliharaan agar biaya yang dikeluarkan bisa lebih hemat. Riset menemukan biaya yang lebih hemat ini sejalan dengan hadist berikut :

عن ابن عمر، قال رسول الله صلى الله عليه وسلم الإقتصاد في النفقه نصف العيش

Artinya: Dari Ibnu 'Umar Ra,

Rasulullah Saw bersabda: berlaku hemat (ekonomis) itu adalah separuh dari kehidupan. (HR. al-Syihab)

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian yang telah dilakukan merupakan penelitian implementasi blockchain pada sistem transaksi game multiplayer. Pengujian multiplayer dilakukan dengan uji fungsionalitas fitur interaksi antar player dan uji *latency* menggunakan lima device berbeda dan provider jaringan berbeda. Pengujian sistem transaksi dilakukan dengan optimalisasi variabel yang mempengaruhi efisiensi biaya dan keefektifitasan kecepatan transaksi. Pada penelitian ini kesimpulan yang didapat adalah sebagai berikut:

1. Uji coba *multiplayer system* dilakukan dengan uji *latency* menggunakan tiga provider jaringan berbeda yaitu Telkomsel dengan hasil *latency* 61ms, Smartfren 127ms, dan Tri 144ms. Pengujian fungsionalitas dilakukan dengan uji *request login*, *interaction* dan *chat* memiliki persentase keberhasilan sebesar 100%. Hal ini membuktikan bahwa penggunaan platform Photon 2 efektif untuk digunakan sebagai multiplayer system.
2. Uji coba transaksi menggunakan *Platform Ethereum* juga memiliki hasil yang di harapkan yaitu kecepatan transaksi yang lebih cepat dibanding platform *Bitcoin*. Dengan *Gas Price* sebesar 30 Gwei, custom *Ethereum* memiliki kecepatan rata-rata 3.8 detik per transaksi, sedangkan bitcoin memiliki kecepatan rata rata 7 menit per transaksi.

3. Pengujian estimasi development cost dilakukan dengan melakukan perhitungan estimasi penggunaan transaksi dalam waktu satu tahun berdasarkan *Gas Price* yang berbeda. Untuk mendapatkan kecepatan transaksi sesuai dengan SPM (Standar Pelayanan Minimal) Custom Ethereum hanya membutuhkan tarif transaksi sebesar 165.000 Gwei atau setara Rp.491 per transaksi, harga ini lebih murah dibanding *Bitcoin* yang memiliki biaya transaksi 4950000 Gwei atau setara Rp.17.973 per transaksi.

5.2 Saran

Pembuatan game simulasi pada penelitian ini tidak bisa dikatakan sempurna. Banyak kekurangan yang harus diperbaiki sebagai pengembangan untuk penelitian selanjutnya. Oleh karena itu peneliti menyarankan beberapa hal untuk pengembangan selanjutnya antara lain :

1. Mengembangkan game ini dengan lebih menarik seperti menambahkan environment yang kurang lengkap, memperbaiki texture *Player*, bangunan, dll.
2. Menambah destinasi lokasi wisata bromo yang belum ada di penelitian ini seperti bukit kingkong, bukit teletubies, bukit cinta, wisata agro kebun strawberry, dll.
3. Menambah karakter lain seperti pedagang pakaian, petani strawberry, jasa foto langsung jadi, dll.
4. Pengembangan dengan opsi custom match selain auto match making.

DAFTAR PUSTAKA

- Aaron Wright, Primavera De Filippi. (2015). *Decentralized Blockchain Technology and the Rise of Lex Cryptographica*
- Antonopoulos, A.M. & Wood, G. (2018). *Mastering Ethereum: Building Smart Contracts and DApps.*
- Claus Pahl, Maja Vulkovic, Jianwei Yin. (2018). *Service-Oriented Computing*
- Curran, B. (2018). *Blockchain Games: The Current State of Blockchain Gaming Technology*
- Dannen, Chris. (2018). *Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners.*
- Jiun-Ting Chen. (2020). *Blockchain and the Feature of Game Development*
- Kedar Iyer, Chris Dannen. (2019). *Building Games with Ethereum Smart Contracts.*
- Kosuke Komiya & Tatsuo Nakajima. (2019). *Increasing Motivation for Playing Blockchain Games Using Proof-of-Achievement Algorithm.*
- Melanie Swan, (2019). *Blockchain Economic Networks: Economic Network Theory—Systemic Risk and Blockchain Technology.*
- Mohsen Attaran, Angappa Gunasekaran. (2019). *Applications of Blockchain Technology in Business Challenges and Opportunities.*
- Mona Taghavi1, Jamal Bentahar, Hadi Otrok1. 2018. *Cloudchain: A Blockchain-Based Coopetition Differential Game Model for Cloud Computing*
- Oliver, Nathan, Sebastian. (2019). *Ethereum Crypto-Games: Mechanics, Prevalence and Gambling Similarities.*
- Pranay Valson. (10 maret 2020). diambil kembali dari Upvest : <https://upvest.co/blog/Transaction-Fee-estimations-how-to-save-on-Gas>
- Rajani Singh1, Ashutosh Dhar Dwivedi, Gautam Srivastava. (2019). *A game theoretic analysis of resource mining in Blockchain.*
- S. Asharaf. (2017). *Decentralized Computing Using Blockchain Technologies and SmartContracts*
- Shailak Jani. (2017). *An Overview of Ethereum & Its Comparison with Bitcoin.*