SISTEM KEAMANAN MOBILITAS ORANG DI KOTA MALANG BERBASIS *BLOCKCHAIN*

SKRIPSI

Oleh:
DIAN PERMANA PUTRA
NIM. 16650108

JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2021

SISTEM KEAMANAN MOBILITAS ORANG DI KOTA MALANG BERBASIS *BLOCKCHAIN*

SKRIPSI

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

Oleh:
DIAN PERMANA PUTRA
NIM. 16650108

JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2021

LEMBAR PERSETUJUAN

SISTEM KEAMANAN MOBILITAS ORANG DI KOTA MALANG BERBASIS *BLOCKCHAIN*

SKRIPSI

Oleh :
DIAN PERMANA PUTRA
NIM. 16650108

Telah Diperiksa dan Disetujui untuk Diuji

Tanggal : 06 Desember 2021

Dosen Pembimbing I	Dosen Pembimbing II
Dr. Fachrul Kurniawan, ST., M.MT., IPM	Yunifa Miftachul Arif, M. T
NIP. 197710202009121001	NIP. 19830616 201101 1 004

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

Dr. Fachrul Kurniawan, ST., M.MT., IPM
NIP. 197710202009121001

LEMBAR PENGESAHAN

SISTEM KEAMANAN MOBILITAS ORANG DI KOTA MALANG BERBASIS *BLOCKCHAIN*

SKRIPSI

Oleh :
DIAN PERMANA PUTRA
NIM. 16650108

Telah Dipertahankan di Depan Dewan Penguji Skripsi dan
Dinyatakan Diterima sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 23 Desember 2021

Susunan Dewan Penguji :

Penguji Utama	:	Khadijah Fahmi Hayati Holle, M.Kom NIDT. 19900626 20160801 2 077	( )
Ketua Penguji	:	Hani Nurhayati, M.T NIP. 19780625 200801 2 006	( )
Sekretaris Penguji	:	Dr. Fachrul Kurniawan, ST., M.MT., IPM NIP. 19771020200912100l	( )
Anggota Penguji	:	Yunifa Miftachul Arif, M. T NIP. 19830616 201101 1 004	( )

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

Dr. Fachrul Kurniawan, ST., M.MT., IPM
NIP. 19771020200912100l

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan dibawah ini:

Nama	: Dian Permana Putra
NIM	: 16650108
Fakultas/Jurusan	: Sains dan Teknologi/Teknik Infomatika
Judul Skripsi	: Sistem Keamanan Mobilitas Orang Di Kota Malang Berbasis *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, 23 November 2021
Yang membuat pernyataan,

Dian Permana Putra
NIM. 16650108

HALAMAN MOTTO

"Kamu tidak perlu menjadi hebat untuk memulai,
Namun kamu perlu memulai untuk menjadi hebat"

HALAMAN PERSEMBAHAN

Alhamdulillah, puji dan syukur kehadirat Allah SWT yang telah memberikan nikmat dan hidayah sehingga penulis bisa menyelesaikan pendidikan S1 jurusan Teknik Informatika di Universitas Islam Negeri Maulana Malik Ibrahim Malang. Shalawat serta salam selalu tercurahkan kepada junjungan Nabi Agung Muhammad SAW yang telah membimbing umatnya dari jaman jahiliyah menuju jaman yang terang benderang dan dipenuh dengan ilmu pengetahuan seperti saat ini.

Terima kasih kepada Ibu dan Ayah yang selalu membuatku termotivasi dan selalu memberikan kasih sayang, mendoakanku, serta menasehatiku menjadi lebih baik. Terima kasih Ibu dan Ayah atas semua yang telah engkau.

Untuk Bapak Dr. Fachrul Kurniawan, ST., M.MT., IPM dan bapak Yunifa Miftachul Arif, M.T selaku pembimbing, terima kasih sudah membimbing serta memberikan arahan dengan tulus, sabar, dan ikhlas dalam penyelesaian skripsi ini.

Skripsi ini juga tidak akan selesai tanpa adanya dukungan dari sahabat saya Faizal Armas Fata dan juga teman lainnya yang tidak bisa saya sebutkan satu per satu. Terima kasih atas dukungan, kebaikan, serta perhatian kalian dalam pengerjaan skripsi ini dengan cara yang sangat menghibur. Tidak lupa ucapan terima kasih kepada teman seperjuangan ANDROMEDA TI'16 yang selalu meluangkan waktu serta memberikan semangat dalam penyelesaian skripsi. Semangat untuk teman-temanku yang masih berjuang untuk mengerjakan tugas akhir ini. Terima kasih teman-teman tercinta semoga kita terus terhubung melalui doa yang mengiringi kesuksesan kita. Aamiin.

KATA PENGANTAR

Assalamu'alaikum Warohmatullaahi Wabarakaatuh

Puji syukur kehadirat Allah SWT, atas limpahan Rahmat dan Karunia-Nya, sehingga penulis dapat menyelesaiakan penelitian dengan judul: "Sistem Keamanan Mobilitas Orang di Kota Malang Berbasis *Blockchain*". Sholawat dan salam kepada junjungan Nabi Muhammad SAW, Nabi yang syafaatnya dinantikan seluruh umat manusia baik di dunia maupun di akhirat. Semoga kita termasuk golongan yang mendapat petunjuk Allah SWT dan mendapat syafaat 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. Prof. Dr. M. Zainuddin, MA selaku Rektor Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
2. Dr. Sri Harini, M.Si selaku Dekan fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
3. Dr. Fachrul Kurniawan, ST., M.MT., IPM, Selaku Ketua Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang dan juga selaku dosen Pembimbing I yang telah 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.

4. Yunifa Miftachul Arif, M. T, selaku Dosen Pembimbing II yang telah teliti membimbing penulis untuk dapat mencapai hasil skripsi yang lebih baik.
5. Khadijah Fahmi Hayati Holle, M.Kom dan Hani Nurhayati, M.T, selaku Dosen Penguji dengan sikap profesional telah menguji seluruh proses ujian skripsi penulis mulai dari seminar proposal hingga sidang skripsi dengan lancar.
6. Seluruh jajaran staf dan dosen jurusan Teknik Informatika yang secara langsung maupun tidak langsung terlibat dalam proses pengerjaan skripsi.
7. Orang tua tercinta yang telah banyak memberikan doa dan dukungan kepada penulis secara moril maupun materil hingga skripsi ini dapat terselesaikan.
8. Sahabat-sahabat seperjuangan yang tiada henti memberi dukungan dan motivasi kepada penulis serta target bersama untuk lulus skripsi dan wisuda bersama.
9. Teman-teman andromeda yang selalu memberikan semangat dan doa kepada penulis.

Peneliti menyadari bahwa proses penelitian dari awal hingga akhir masih jauh dari kata sempurna. Oleh karena itu, penulis membuka kesempatan selebar-lebarnya untuk setiap saran dan kritik yang membangun. Peneliti berharap semoga skripsi yang telah peneliti tulis bermanfaat bagi generasi selanjutnya.

Wassalamu'alaikum Warohmatullaahi Wabarakaatuh

Malang, 23 November 2020

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAK

Putra, Dian Permana. 2021. **Sistem Keamanan Mobilitas Orang di Kota Malang Berbasis Blockchain**. Skripsi. Jurusan Teknik Informatika. Fakultas Sains dan Teknologi. Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Dr. Fachrul Kurniawan ST., M.MT., IPM, (II) Yunifa Miftachul Arif, M. T

Kata Kunci: Keamanan, Mobilitas, *Blockchain*, *PoW*, SHA256.

Penelitian ini bertujuan untuk mengamankan data dan transaksi informasi mobilitas orang di kota Malang yang dikumpulkan ke jaringan *blockchain* serta mengetahui hasil kinerja dari pembacaan koordinat, algoritma SHA256 dan kecepatan *block-mining* sebagai penerapan teknologi *blockchain* pada sistem yang dibuat. Hal ini diperlukan agar setiap *user* dan pengendara sebagai pengumpul data, bisa mengakses data mobilitas dengan tetap terjaga keamanan dan keaslian datanya. *Blockchain* merupakan teknik baru yang digunakan untuk menyimpan data agar tidak terpusat dan akan didesentralisasikan dalam jaringan terdistribusi. Sistem ini menyimpan informasi lokasi dari data *GPS* secara umum dengan penggunaan algoritma SHA256 dan *Proof of Work* (*PoW*). Subjek dalam penelitian ini adalah penduduk kota Malang dengan batasan area pelacakan wilayah terminal Arjosari, terminal Landungsari dan stasiun Kota Baru. Pengumpulan data menggunakan perangkat modul *GPS* dan *Raspberry Pi*. Validasi yang dilakukan menggunakan algoritma *PoW*. Hasil penelitian menunjukkan bahwa kinerja algoritma SHA256 dalam menghasilkan nilai *hash* sebagai penghubung blok-blok penyusun *blockchain* pada sistem keamanan mobilitas yang sudah dibuat, berhasil bekerja dengan baik. Antar blok saling terhubung satu sama lain melalui nilai *hash* dan *previous hash* masing-masing blok. Kecepatan *block-mining time Proof of Work* pada sistem yang dibuat terhitung cepat. Nilai *binary* setiap data memenuhi nilai target *difficulty* dengan kecepatan waktu *Mining* rata-rata adalah 66,6 milidetik.

ABSTRACT

Putra, Dian Permana. 2021. **Blockchain-Based Security System for People Mobility in Malang City**. Undergraduate Theses. Informatics Engineering Department. Faculty of Science and Technology. State Islamic University of Maulana Malik Ibrahim Malang. Advisors: (I) Dr. Fachrul Kurniawan ST., M.MT., IPM, (II) Yunifa Miftachul Arif, M. T

Keywords: *Security, Mobility, Blockchain, PoW,* SHA256.

This research aims to secure data and transaction information on the mobility of people in the city of Malang which is collected into the blockchain network and to find out the performance results of reading coordinates, SHA256 algorithm, and block-mining speed as the application of blockchain technology to the system created. This is required. So that every user and driver as a data collector can use data of mobility while maintaining the security and authenticity of the data. Blockchain is a new technique of decentralized data storage in a distributed network. This system stores information of location from GPS data, in general, using the SHA256 and Proof of Work (PoW) algorithms. The subjects in this research are residents of the city of Malang with a limited area tracking area Station Bus of Arjosari, Landungsari, and Station of Kota baru. Data collection used GPS module devices and Raspberry Pi. Validation is carried out using the PoW algorithm. The results showed that the performance of the SHA256 algorithm in generating hash values as a liaison for the blockchain building blocks in the mobility security system that had been created worked well. The blocks were connected through the hash value and previous hash of each block. The speed of the Proof of Work block-mining time on the system made is fast. The binary value of each data met the target difficulty value with the average Mining time speed of 66.6 milliseconds.

مستخلص البحث

فوتيرا، ديان فيرمانا. 2021. **نظام الأمان المستند إلى Blockchain لتنقل الأشخاص في مدينة مالانغ**. البحث العلمي. قسم هندسة المعلوماتية، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم مالانج. المشرف: (1) المشرف الأول: فاكر الكورنيوان، الماجستير، (2) المشرف الثاني: يونيفة مفتاح العاريف ، المجستير.

الكلمات المفتاحية : الأمن، التنقل، Blockchain، PoW، SHA256

تهدف هذه الدراسة إلى تأمين البيانات ومعلومات المعاملات حول تنقل الأشخاص في مدينة مالانج والتي يتم جمعها في شبكة blockchain ومعرفة نتائج أداء إحداثيات القراءة وخوارزمية SHA256 وسرعة تعدين الكتلة كتطبيق blockchain التكنولوجيا للنظام الذي تم إنشاؤه. يعد ذلك ضروريًا حتى يتمكن كل مستخدم وسائق كمجمع بيانات من الوصول إلى بيانات التنقل مع الحفاظ على أمان البيانات ومصداقيتها. Blockchain هي طريقة جديدة لتخزين البيانات اللامركزية في شبكة موزعة. . يخزن هذا النظام معلومات الموقع من بيانات GPS بشكل عام باستخدام خوارزميات SHA256 وإثبات العمل (PoW). كان المبحث في هذه الدراسة هي سكان مدينة مالانغ مع منطقة تتبع منطقة محدودة أخير أرجوساري وأخير لندونغياري ومحطة كوتا بارو. جمع البيانات باستخدام أجهزة وحدة GPS و Raspberry Pi. يتم إجراء التحقق باستخدام خوارزمية إثبات العمل. أظهرت النتائج أن أداء خوارزمية SHA256 في توليد قيم التجزئة كحلقة وصل لبنات بناء Blockchain على نظام أمان التنقل الذي تم إنشاؤه يعمل بشكل جيد. ترتبط الكتل ببعضها البعض من خلال قيمة التجزئة والتجزئة السابقة لكل كتلة. سرعة وقت استخراج كتلة إثبات العمل على النظام الذي تم تصنيعه سريعة. تتوافق القيمة الثنائية لكل بيانات مع قيمة الصعوبة المستهدفة بمتوسط سرعة وقت التعدين البالغ 66.6 مللي ثانية.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam menjalani kehidupan, tak dapat dipungkiri bahwa manusia adalah insan yang saling membutuhkan, sehingga manusia juga disebut sebagai makhluk sosial. Dalam kehidupan sosial, manusia tentunya saling berinteraksi dan bertukar informasi. Dalam kondisi tertentu, kelompok yang berbagi informasi hanya ingin diketahui oleh kelompok tertentu saja. Oleh sebab itu dibutuhkan suatu keamanan agar informasi yang akan dibagikan merupakan kesepakan bersama antar pemilik data yang saling menyetujui aturan transaksi.

Keamanan privasi individu merupakan masalah krusial yang perlu ditangani. Kasus-kasus peretasan sangat marak terjadi dalam beberapa tahun terakhir baik dalam negeri maupun di luar negeri. Beberapa kasus yang terjadi di luar negeri: pada tahun 2015, kelompok peretas sipil menyebarkan data lokasi bus yang tidak standar di Baltimore. Pada 2016, informasi milik 57 juta pelanggan dan pengemudi uber tersebar. Kemudian pada tahun 2018, agen transportasi regional Ontario, server Metrolinx mendapat serangan peretasan (Lopez dan Farooq, 2018). Sebagian besar data disimpan pada server milik perusahaan-perusahaan. Teknik enkripsi dengan kunci publik dan kunci privat telah dimanfaatkan penggunaannya untuk keamanan data dan komunikasi (Farooq dkk., 2015). Namun peneliti-peneliti menemukan bahwa blockchain bisa dimanfaatkan untuk menangani masalah privasi yakni dengan mendesentralisasikan informasi dan di sisi lain individu berperan sebagai pemilik tunggal dan pengendali atas informasi

mereka. Teknologi *blockchain* memiliki potensi untuk melindungi informasi mobilitas pribadi individu dengan keunggulannya yang sulit diretas, menghasilkan transaksi yang aman dan transparan bagi pihak-pihak yang berbagi data melalui jaringan *blockchain* (Lopez dan Farooq, 2018). Berkenaan dengan penyampaian informasi kepada pihak yang berhak dijelaskan dalam al-Quran surat an-Nisaa'/4:58, yakni:

اِنَّ اللّٰهَ يَأْ مُرُ كُمْ أَنْ تُؤَدُّوا الْأَمَا نَاتِ إِلَى أَهْلِهَا وَإِذَا حَكَمْتُمْ بَيْنَ النَّاسِ أَنْ تَحْكُمُوا بِالْعَدْلِ ۚ إِنَّ اللّٰهَ نِعِمَّا يَعِظُكُمْ بِهِ ۗ إِنَّ اللّٰهَ كَانَ سَمِيْعًا بَصِيْرًا

"Sesungguhnya Allah menyuruh kamu menyampaikan amanat kepada yang berhak menerimanya, dan (menyuruh kamu) apabila menetapkan hukum di antara manusia supaya kamu menetapkan dengan adil. Sesungguhnya Allah memberi pengajaran yang sebaik-baiknya kepadamu. Sesungguhnya Allah adalah Maha mendengar lagi Maha melihat" (QS. An-Nisaa'/4:58).

Sumber hukum islam memiliki sifat yang fleksibel dimana sumber hukum islam dapat diterapkan pada berbagai kondisi, situasi dan permasalahan. Hal ini menjadi sangat penting sebagai pedoman dan bagian dari latar belakang bagi penulis dalam mengambil judul skripsi ini serta berinovasi guna menciptakan sesuatu yang bermanfaat. Dalam menemukan sumber hukum islam dalam kasus peretasan, dalil-dalil yang ada di*istinbat*kan dan diwujudkan dalam model kaidah fikih. Tentunya ini menunjukkan bahwa hukum islam bisa memberikan sudut pandang yang berbeda dalam masalah-masalah terkait teknologi informasi termasuk kasus peretasan. Penyusupan merupakan jenis dari aksi peretasan yang memiliki makna yakni masuk tanpa izin dan tanpa sepengetahuan pemilik barang. Semua bentuk tindakan penyusupan yang menyangkut objek dan subjek memiliki hukum dasar dalam al-Quran surat an-Nur ayat 27, yaitu:

يَا أَيُّهَا الَّذِينَ آمَنُوا لَا تَدْخُلُوا بُيُوتًا غَيْرَ بُيُوتِكُمْ حَتَّىٰ تَسْتَأْنِسُوا وَتُسَلِّمُوا عَلَىٰ أَهْلِهَا ۚ ذَٰلِكُمْ خَيْرٌ لَكُمْ لَعَلَّكُمْ تَذَكَّرُونَ

"Hai orang-orang yang beriman, janganlah kamu memasuki rumah yang bukan rumahmu sebelum meminta izin dan memberi salam kepada penghuninya. Yang demikian itu lebih baik bagimu, agar kamu (selalu) ingat" (QS. An-Nur/24:27).

Kerangka kerja *blockchain* diterapkan untuk menangani masalah privasi dan keamanan. Masyarakat sipil merupakan salah satu pihak yang berperan sebagai pemilik dan pengumpul data mereka masing-masing. Pemilik data membuat aturan transaksi, sehingga pihak yang menyetujui aturan tersebut dapat melakukan transaksi informasi atau data yang telah dienkripsi oleh pemilik data melalui jaringan *blockchain*.

Konsep *blockchain* awal mulanya digunakan pada mata uang digital atau lebih dikenal dengan istilah bitcoin (Nakamoto, 2008). *Blockchain* merupakan buku besar milik bersama yang menyimpan rentetan catatan transaksi, dimana catatan tersebut tidak dapat diubah kecuali apabila semua pihak dalam jaringan *blockchain* menyetujui kesepakatan bersama (Swan, 2015). Pemanfaatan *blockchain* dirasa tepat karena sifatnya yang mampu menciptakan jaringan aman tanpa adanya satu organisasi pun yang mengendalikan transaksi dan data. Selanjutnya penulis menyajikan studi terkait dengan aplikasi *blockchain* dalam transportasi dan privasi.

Manajemen rantai pasokan merupakan bagian aplikasi transportasi utama *blockchain*. *Blockchain* yang disinergikan dengan teknik RFID dapat membantu perusahaan dalam melacak produk dari produser, distributor hingga konsumen (Lopez dan Farooq, 2018). *Blockchain* menjadi solusi permasalahan pasokan

transportasi. Dalam penelitian ini diusulkan jaringan *blockchain* agar dapat tercipta sistem keamanan mobilitas orang.

Di Korea, kecelakaan sepeda motor berada di kisaran 5% dari jumlah kecelakaan lalu lintas yang terjadi di sana, tetapi tingkat kematian hampir mencapai 12%, dimana hal ini menunjukkan angka yang lebih tinggi apabila dibandingkan dengan kematian yang disebabkan oleh kendaraan lain. Data di seluruh dunia menunjukkan bahwa pengendara sepeda motor dan penumpangnya sering terlibat dalam kecelakaan lalu lintas (Jo dkk., 2015). Administrasi Keselamatan Lalu Lintas Jalan Raya Nasional AS (NHTSA) menginformasikan bahwa pada 2012, sebanyak 4.957 pengendara sepeda motor meninggal dunia dalam kecelakaan lalu lintas, jumlah tersebut meningkat 7% dari jumlah 4.630 pengendara sepeda motor yang meninggal dunia pada tahun sebelumnya (Jo dkk., 2015). Departemen Transportasi Inggris menginformasikan bahwa pada 2013, terdapat 331 kasus pengendara sepeda motor meninggal dunia dan sebanyak 4.866 korban terluka parah dalam kecelakaan sepeda motor. Tercatat bahwa pengendara sepeda motor memiliki sekitar 38 kali lebih mungkin meninggal dunia dalam kecelakaan daripada penumpang mobil. Di Australia, sebuah studi nasional oleh Australian Safety Bureau (ATS) mengungkapkan bahwa pengendara sepeda motor 30 kali lebih mungkin meninggal dunia daripada pengemudi mobil (Johnston dkk., 2008).

Memahami perilaku pengendara sepeda motor sangat penting untuk memprediksi lokasi-lokasi hilangnya kontak terhadap pengendara sepeda motor akibat kecelakaan. Kemajuan dalam teknologi seperti pelacakan telah

dimanfaatkan untuk memantau perilaku pengendara dan juga prediksi kemacetan di jalanan kota. Anggota keluarga dan orang-orang terdekat seringkali merasa tidak aman karena meningkatnya tingkat kejahatan dan kecelakaan. Teknologi dapat dimanfaatkan pada skala yang lebih besar untuk membantu menangani masalah tersebut. Penelitian ini bertujuan mengamankan data mobilitas seseorang, menyimpulkan aktivitas di setiap lokasi, dan memprediksi jalur yang paling mungkin dilalui pengendara. Penelitian ini bermaksud untuk merancang model pendeteksian yang efisien yang relatif mudah dipahami dan dikembangkan lebih lanjut.

Peneliti mencoba mengaplikasikan *prototype* sistem pada area kota Malang. Peneliti memilih area kota Malang karena lokasinya dekat dengan domisili peneliti sehingga memudahkan dalam melakukan uji coba. Tidak hanya hal tersebut, alasan lain yang lebih penting adalah pemilihan kota Malang di rasa tepat karena berdasarkan data dari BPS kota Malang melalui publikasinya yang berjudul Kota Malang dalam Angka pada tahun 2019, mengungkapkan bahwa pada tahun 2018 jumlah penduduk kota Malang sebanyak 866.168 jiwa dengan laju pertumbuhan penduduk mencapai 0.68 persen per tahun. Kota Malang memiliki kepadatan penduduk sebesar 7.870 jiwa/km^2. Angka tersebut menunjukkan angka kepadatan yang tinggi di wilayah jawa timur, sehingga tingkat mobilitas orang di kota Malang juga relatif tinggi.

Berdasarkan fakta-fakta yang telah dijelaskan pada paragraf-paragraf sebelumnya, maka peneliti mengajukan penelitian dengan judul Sistem Keamanan Mobilitas Orang di Kota Malang Berbasis *Blockchain.* Pada sistem ini lokasi

pengguna dilacak menggunakan Global Positioning System (GPS). Fungsionalitas utama dari teknologi ini adalah untuk menyediakan informasi lokasi. Ketika kendaraan melintasi lokasi pelacakan ini, pengiriman informasi secara otomatis dari sistem pelokalan kendaraan dibuat. Sistem yang dibuat terdiri dari modul yang ditempatkan pada sepeda motor dengan perangkat *IoT* yang mudah tersedia untuk terus mengirim posisi pengendara yang melintasi lokasi pelacakan ke pusat pengendalian data dalam hal ini diasumsikan pemerintah, guna memantau perilaku pengendara sepeda motor, prediksi lokasi hilangnya pengendara sepeda motor akibat kecelakaan dan memprediksi area kemacetan yang terjadi di kota Malang. Teknologi *Blockchain* diintegrasikan untuk melindungi informasi mobilitas pribadi individu dan menjaga privasi pengendara. Teknologi ini sulit dirusak dan menciptakan transaksi yang aman serta transparan bagi semua pihak. Dalam penelitian ini kerangka kerja *blockchain* untuk transaksi data mobilitas. Tujuan utamanya adalah untuk mengamankan data yang dikumpulkan dan untuk menjaga privasi individu. Pengembangan terhadap sistem yang dibuat juga dapat dimanfaatkan untuk memantau pergerakan Orang Dalam Pemantauan atau yang lebih dikenal dengan sebutan ODP. Berdasarkan data dari Gugus Tugas Percepatan Penanganan covid-19 pada tanggal 1 Agustus 2020, jumlah kasus covid-19 di Indonesia mencapai angka 109.936 kasus positif dengan jumlah kematian mencapai angka 5.193 jiwa. Maka pengembangan sistem ini juga bermanfaat untuk memutus mata rantai penyebaran virus korona di Kota Malang.

1.2 Pernyataan Masalah

Berdasarkan latar belakang yang telah dijelaskan sebelumnya, penelitian ini difokuskan pada permasalahan berikut:

1. Seberapa baik kinerja algoritma *SHA256* dalam menghasilkan nilai *hash* sebagai penghubung blok-blok penyusun *blockchain*?
2. Seberapa tinggi kecepatan *block-mining time Proof of Work* dalam proses validasi nilai *hash* pada sistem yang dibuat?

1.3 Tujuan Penelitian

Berdasarkan pernyataan masalah di atas, maka tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut:

1. Mengamankan data dan transaksi informasi mobilitas yang dikumpulkan ke jaringan *blockchain*.
2. Mengetahui hasil kinerja dari pembacaan koordinat, algoritma SHA256 dan kecepatan *block-mining* sebagai penerapan teknologi *blockchain* pada sistem yang dibuat.

1.4 Manfaat Penelitian

Hasil dari penelitian yang dilakukan diharapkan dapat memberikan beberapa manfaat antara lain:

1. Memberikan solusi terhadap tantangan privasi dan keamanan yang terkait dengan data dalam mobilitas.
2. Menciptakan transaksi informasi yang aman dan transparan melalui adanya aturan transaksi yang dibuat oleh pemilik data.
3. Membantu pemerintah dan pihak-pihak terkait untuk lebih memahami mobilitas manusia guna merancang sistem transportasi yang lebih baik.
4. Mengurangi potensi terjadinya pelanggaran keamanan *cyber*.

1.5 Batasan Masalah

Penelitian ini memiliki batasan-batasan, dimana hal ini dimaksudkan untuk tidak memperluas ruang lingkup penelitian. Batasan masalah ini juga diperlukan untuk lebih mengarahkan atau memfokuskan penelitian ini. Adapun batasan-batasan pada penelitian ini adalah:

1. *Node* pengumpul data dalam jaringan *blockchain* pada penelitian ini adalah individu dan pemerintah yang diasumsikan mengambil peran sebagai pengendali data.
2. Data yang dapat dibagikan meliputi: data dari modul GPS, informasi pribadi dari penduduk kota malang dan informasi lokasi *real time*

penduduk kota Malang pada area yang sudah ditentukan.

3. Batasan area pelacakan adalah terminal Arjosari, terminal Landungsari dan stasiun Kota Baru. Pemilihan lokasi tersebut karena tempat-tempat tersebut merupakan tempat keramaian di kota Malang yang menunjukkan mobilitas orang di kota Malang.
4. Data mobilitas seseorang menjadi obyek yang diamankan terhadap potensi serangan

BAB II

STUDI PUSTAKA

2.1 *Blockchain*

Penjelasan awal dari teknik *blockchain* muncul pada tahun 2008, yakni pada makalah yang ditulis oleh Satoshi Nakamoto. Pada tulisannya tersebut dengan judul Bitcoin: *a Peer-to-peer Electronic Cash system*, dimana digambarkan sebagai rentetan rekaman transaksi Bitcoin (Nakamoto, 2008). Definisi lain, menurut buku putih yang dibuat oleh (Kementerian Perindustrian dan Teknologi Informasi, 2016), *blockchain* merupakan struktur data yang seolah memiliki rantai, dimana rantai tersebut menggabungkan blok data secara berurutan sesuai urutan waktu. Teknik kriptografi dimanfaatkan untuk meyakinkan bahwa buku akun yang didistribusikan tidak bisa dirusak dan tidak bisa dipalsukan. Secara umum, teknologi *blockchain* memakai struktur data *blockchain* untuk melakukan verifikasi dan penyimpanan data, menggunakan *node* terdistribusi dan algoritma konsensus untuk menciptakan dan memperbarui data, serta memanfaatkan kriptografi untuk melindungi keamanan transmisi dan akses data (Liang dkk., 2017a). Prasarana terdistribusi dan kerangka berpikir komputasi baru untuk pemrograman serta mengatur transaksi data menggunakan kontrak cerdas yang berisi kode skrip otomatis. *Blockchain* memiliki karakteristik berikut (Liang dkk., 2017b):

a. Kerangka kerja terdistribusi. *Blockchain* tercipta di atas jaringan terdistribusi *point to point*, dimana semua data transaksi terekam dalam buku besar yang disimpan pada setiap *node* dalam jaringan,

bukan pada *server* ataupun pusat data. Semua *node* secara selaras memperbarui buku besar, dimana hal tersebut menunjukkan karakteristik desentralisasi.

b. Asal data yang tepercaya. Seluruh sistem bersifat terbuka dan transparan, serta tidak memerlukan adanya pihak ketiga yang tepercaya guna membangun konsensus. Siapa saja bisa bergabung dengan *blockchain*, dan *blockchain* transparan serta terbuka untuk siapa saja melalui akses internet. Pada waktu bersamaan, semua pengguna atau *node* bisa melihat bahwa setiap transaksi yang ditulis dalam buku besar tidak dirusak.

c. Algoritma enkripsi. *Blockchain* menggunakan struktur data *blockchain* dengan nilai SHA256 tertentu dan stempel waktu pada setiap bloknya, sehingga *blockchain* memiliki sisi positif pada pelacakan dan verifikasi yang kuat. Pada waktu bersamaan, algoritma kriptografi dan mekanisme konsensus meyakinkan bahwa *blockchain* tidak terdefinisi.

Secara umum, *blockchain* merupakan jenis baru kerangka komputasi terdistribusi berdasarkan kriptografi, komunikasi jaringan *point-to-point*, algoritma konsensus dan kontrak cerdas. Dengan kemampuan yang tidak memerlukan adanya perantara atau pihak ketiga, karakteristik data tidak mudah dirusak, tidak mudah dipalsukan, bisa dilacak dan diaudit. Karakteristik ini bisa membuktikan bahwa teknologi *blockchain* dapat dimanfaatkan untuk menjamin keandalan dan keamanan pencatatan informasi maupun data.

2.1.1 *Blockchain* Publik

Bitcoin merupakan implementasi paling umum dari *blockchain* publik. Dalam Bitcoin, transaksi terjadi dengan memindahkan koin dari satu set alamat ke yang lain (Dinh dkk., 2018). Setiap *node* mengumumkan serangkaian transaksi yang ingin dilakukan. *Node* khusus yang merupakan penambang menghimpun transaksi ke dalam blok, memeriksa validitasnya, dan protokol konsensus untuk menambahkan blok ke dalam *blockchain* (Dinh dkk., 2018). Bitcoin menerapkan *Proof-of-Work* (PoW) untuk konsensus: hanya penambang yang berhasil menemukan angka yang tepat untuk *header* blok secara komputasi adalah blok yang dapat ditambahkan ke *blockchain*. *PoW* memiliki sifat probabilistik: artinya ada kemungkinan terjadi dua blok ditambahkan pada waktu bersamaan, menimbulkan *fork* pada *blockchain*. Bitcoin menyelesaikan masalah ini dengan mempertimbangkan blok yang dikonfirmasi sesudah diikuti oleh sejumlah blok, namun umumnya enam blok (Dinh dkk., 2018).

Pada umumnya sistem *blockchain* publik menggunakan varian *PoW* untuk konsensus. *PoW* berjalan dengan baik dalam pengaturan publik sebab melindungi terhadap serangan *cyber* (Vu dkk., 2009). Tetapi, karena bersifat *non-deterministik* dan mahal secara komputasional, sehingga tidak tepat apabila diterapkan pada aplikasi seperti perbankan dan keuangan dimana harus mengatasi volume transaksi yang besar secara deterministik (Dinh dkk., 2018).

2.1.2 *Blockchain* Privat

Hyperledger merupakan salah satu *blockchain*s privat paling populer. Sebab identitas simpul dikenal dalam pengaturan pribadi, pada umumnya *blockchain* mengadopsi salah satu protokol-protokol yang ada tentang konsensus terdistribusi (Dinh dkk., 2018). Zab, Raft, Paxos dan PBFT merupakan protokol populer yang aktif saat ini. PBFT merupakan protokol tiga tahap. Pada tahap pra-persiapan, seorang pemimpin mengumumkan nilai yang akan dipercaya oleh *node* lain. Kemudian, dalam tahap persiapan, *node* mengumumkan nilai-nilai yang akan mereka percaya. Akhirnya, tahap *commit* mengkonfirmasi nilai yang dipercaya saat lebih dari dua pertiga dari *node* setuju akan tahap sebelumnya. Selain konsensus deterministik, *blockchain* privat mendukung kontrak pintar yang dapat menyampaikan logika transaksi yang sangat rumit (Castro dan Liskov, 1999) dalam (Dinh dkk., 2018).

2.2 Sumber Data

Data mobilitas dihasilkan oleh banyak *node* yang berada dalam satu jaringan *blockchain*. Perusahaan atau penyedia layanan transportasi menghasilkan informasi transportasi yang penting untuk pemerintah dan individu. Contohnya, perusahaan telekomunikasi memproduksi data yang bisa digunakan untuk pemodelan transportasi, log perangkat seluler yang tersedia dan terhubung dengan hotspot atau Wi-Fi dapat digunakan untuk memantau lalu lintas (Farooq dkk., 2015) atau untuk merekam pola aktivitas harian dari individu (Phithakkitnukoon dkk., 2010). Data-data ini bisa kita sebut sebagai *Blockchain* for *Mobility Data*

dan dapat disingkat sebagai *BMD*. Penyedia layanan transportasi juga bisa memanfaatkan *blockchain* untuk menemukan pelanggan maupun menggunakan data yang dihasilkan oleh pemerintah, individu atau penyedia layanan transportasi lain untuk meningkatkan bisnis mereka (Lopez dan Farooq, 2018).

2.3 Privasi Individu

Privasi individu merupakan salah satu hal yang perlu diperhatikan, sehingga sebuah solusi untuk menciptakan kepercayaan sangat diperlukan agar pengguna, pemilik dan pengontrol informasi merasa aman saat berbagi informasi mereka. *Blockchain* memiliki potensi untuk menangani masalah privasi berikut (Lopez dan Farooq, 2018):

a. Kepemilikan data. Perusahaan pada umumnya memiliki data mobilitas masyarakat dan di sisi lain, individu tidak bisa mencabut akses kecuali apabila mereka memilih keluar. Bahkan dalam masalah tersebut, perusahaan tidak dapat menghapus informasi mereka secara menyeluruh. Pada *BMD* orang-orang mempunyai data mereka dan apabila mereka ingin keluar dari jaringan, informasi mereka dihapus dari *blockchain*.

b. Transparansi data. Ada banyak masalah dimana perusahaan membagikan data pribadi kepada pihak yang tidak dipercaya dan di sisi lain, individu tidak menyadarinya. Pemanfaatan *blockchain* dapat mengatasi permasalahan tersebut.

Kontrol akses. Dalam data milik perusahaan, individu tidak bisa memilih informasi mana yang ingin mereka bagikan dan kadang-kadang mereka tidak bisa mencabut akses ke bagian informasi tertentu. Dengan kontrak pintar, setiap orang bisa mengelola akses ke bagian tertentu dari informasi mereka.

2.4 Potensi Serangan

Tujuan pokok kerangka *BMD* adalah untuk melindungi informasi mobilitas pribadi dan mengamankan privasi orang-orang untuk memanfaatkan data yang aktif atau pasif.

Untuk mengukur tingkat perlindungan pada *BMD*, empat kelompok musuh dikenali, dimana serangannya dapat dicegah melalui penggunaan *blockchain*. Menurut Lopez dan Farooq. (2018), kelompok musuh bisa menyerang jaringan, *node* atau ketika informasi telah ditransfer. Kelompok musuh yang teridentifikasi adalah:

a. Intersepsi data: Semua transaksi informasi berjalan melalui koneksi *peer-to-peer* yang unik dan aman, sehingga menyerang satu *node* dirasa tidak sebanding dengan upaya yang dibutuhkan untuk mendekripsi data.

b. Kebocoran data: Semua informasi pribadi di-desentralisasi dan diamankan untuk setiap individu, sehingga kebocoran pada informasi akan memerlukan upaya yang sangat besar untuk meretas data pada sejumlah individu. Intersepsi beberapa koneksi pada waktu yang sama akan membutuhkan intersepsi semua koneksi sehingga daya

komputasi untuk melakukan tugas ini sulit berjalan.

c. Pembagian informasi yang tidak diminta: Setiap *node* mempunyai akses penuh ke bagian-bagian buku besar tempat informasi mereka terlibat sehingga mereka dapat dengan mudah memverifikasi informasinya pada tempat yang mereka inginkan.

Permintaan informasi yang tidak diminta: Kontrak pintar memberikan keleluasaan pada *node* untuk memutuskan informasi yang ingin mereka bagikan dengan *node* tertentu.

2.5 Kontrak Pintar

Kontrak pintar merupakan sebuah syarat, yang ditentukan dalam bentuk digital, termasuk peraturan dimana pihak-pihak membuat aturan-aturan (Lopez dan Farooq, 2018). Kontrak pintar bisa diartikan sebagai kode program yang menjelaskan serangkaian aset yang tersedia untuk dikirim dan jenis transaksi yang diizinkan. Semua kontrak pintar disimpan pada *blockchain* dan memiliki alamat yang berbeda-beda atau unik. Mereka berperan sebagai pelaku yang tujuannya adalah untuk melakukan transaksi aset dimana dalam transaksi tersebut terdapat adanya seperangkat aturan tertentu yang disetujui oleh pihak-pihak yang terlibat (Christidis dan Devetsik, 2016). Dalam *BMD* setiap pemilik data atau peserta dalam jaringan *blockchain* mendefinisikan kontrak pintar mereka sendiri dan kontrak pintar tersebut hanya bisa diubah oleh pemilik data. Semua peserta (*node*) dalam jaringan hanya bisa mendefinisikan kontrak pintar untuk data yang mereka punya serta orang-orang hanya bisa menentukan kontrak pintar untuk

informasi yang mereka himpun melalui *ponsel* mereka ataupun dengan cara yang lain. Adapun fungsi dari kontrak pintar adalah sebagai berikut (Lopez dan Farooq, 2018):

a. Suatu fungsi untuk menerima koneksi.
b. Fungsi untuk memberikan izin kepada pihak lain untuk memilih informasi.
c. Fungsi untuk memutus koneksi.

2.6 Konsensus

Konten buku besar menggambarkan keadaan historis saat ini yang dikelola oleh *blockchain*. Disimulasikan, pembaruan pada buku besar harus disetujui oleh semua pihak dalam satu jaringan *blockchain*. Sehingga, terdapat banyak pihak yang harus mencapai kesepakatan (Dinh dkk., 2018).

Literatur penelitian mengenai konsensus terdistribusi cukup luas, serta ada banyak versi protokol yang diusulkan sebelumnya sedang dikembangkan untuk *blockchain* (Vukolic, 2015). Protokol berbasis komputasi murni yang menggunakan bukti perhitungan untuk menentukan secara acak sebuah simpul guna menentukan operasi selanjutnya. Bukti kerja Bitcoin (*PoW*) adalah contohnya.

2.7 GPS

Penyebaran layanan berbasis lokasi tergantung pada keberadaan teknologi penentuan posisi yang bebas dan luas untuk mendapatkan lokasi pengguna atau

perangkat. *Global Positioning System* (GPS) adalah *Global Navigation Satellite System* (GNSS) yang dikembangkan oleh Departemen Pertahanan Amerika Serikat (Zhihong dkk., 2009). Ini merupakan satu-satunya GNSS yang berfungsi penuh di dunia yang menggunakan konstelasi antara 24 dan 32 satelit Orbit Bumi sedang yang mentransmisikan sinyal gelombang mikro yang tepat, dimana memungkinkan penerima GPS guna menentukan lokasi *real time*, waktu, dan kecepatannya (Zhihong dkk., 2009). Tetapi, GPS konvensional memiliki kesulitan dalam menyediakan posisi yang dapat diandalkan dalam situasi sinyal yang kurang baik (Zhihong dkk., 2009). Misalnya saat dikelilingi oleh gedung-gedung tinggi, atau saat sinyal satelit melemah ketika perangkat GPS berada di dalam ruangan atau di bawah pohon (Zhihong dkk., 2009).

2.8 Algoritma SHA256

SHA merupakan bagian keluarga MD fungsi hash, dipublikasikan oleh *American National Institute for Standards and Technology* dan dijadikan sebagai standar FIPS pada tahun 1993. SHA-0 merupakan model pertama. Pada tahun 1994, terdapat sedikit perubahan pada SHA-0, dan hadir sebagai SHA-1 (NIST, 1994)(NIST, 2002). Pada tahun 2000, fungsi SHA baru diperkenalkan dengan ukuran pesan digest yang lebih besar, yakni 256 bit yang disebut SHA256, 384 bit disebut SHA-384 dan 512 bit disebut SHA-512, kemudian dijadikan sebagai standar FIPS pada tahun 2002 (NIST, 2002).

Algoritma SHA256 dimanfaatkan untuk menghitung nilai *message digest* dari sebuah pesan yang mempunyai panjang maksimum 2^{64} bit. Algoritma ini memerlukan sebuah *message schedule* yang terdiri dari 64 element 32-bit *word*,

delapan buah variabel 32-bit, dan variabel penyimpan nilai *hash* 8 buah *word* 32-bit. Keluaran algoritma SHA256 adalah sebuah *message digest* dengan panjang 256-bit (Sebastian, 2007). SHA256 mengonversi pesan sebagai masukan ke bentuk *message digest* 256 bit. Berdasarkan *Secure Hash Signature Standard*, pesan masukan yang mempunyai panjang lebih pendek dari 2^{64} bit, harus diproses oleh 512 bit dalam satuan dan menjadi sebuah *message digest* 256-bit (Mankar dan Nipanikar, 2013). Sebagaimana diungkapkan oleh Mankar dan Nipanikar (2013), tahapan-tahapan algoritma SHA256 adalah sebagai berikut:

1. *Message Padding*: Pada level awal, pesan yang berbentuk *binary* disisipkan angka 1 dan ditambahkan bit-bit penunjang yakni angka 0 hingga panjang pesan tersebut selaras dengan 448 modulo 512. Panjang pesan *original* lalu ditambahkan sebagai angka biner 64 bit. Sehingga panjang pesan menjadi kelipatan 512 bit.
2. *Parsing*: Pesan yang telah di*padding* pada tahap pertama lalu dibagi menjadi N buah blok 512 bit: $M^{(1)}$, $M^{(2)}$, ..., $M^{(N)}$.
3. *Message Expansion*: Setiap blok 512-bit setelah proses *parsing* kemudian dipecah menjadi 16 buat *word* 32-bit: $M_0^{(i)}$, $M_1^{(i)}$, ..., $M_{15}^{(i)}$ dimana nantinya ditingkatkan menjadi 64 *word* yang diberi label W0, W1, ..., W63 dengan aturan tertentu yang telah ditetapkan sebelumnya oleh standar SHA-2.
4. *Message Compression*: Setiap 64 *word* yang berlabel W0, W1, ..., W63 kemudian dibuat 8 variabel yang diberikan nilai untuk nilai awal dari $H_0^{(0)}$ - $H_7^{(0)}$ setiap fungsi *hash*. Nilai-nilai awal ditunjukkan pada gambar 2.1.

$A=H_0^{(0)}$	6a09e667
$B=H_1^{(0)}$	bb67ae85
$C=H_2^{(0)}$	3c6ef372
$D=H_3^{(0)}$	a54ff53a
$E=H_4^{(0)}$	510e527f
$F=H_5^{(0)}$	9b05688c
$G=H_6^{(0)}$	1f83d9ab
$H=H_7^{(0)}$	5be0cd19

Gambar 2.1 *Initial Hash Value* dari SHA256

5. Algoritma ini menngerjakan perhitungan sebanyak 64 kali iterasi untuk masing-masing perhitungan blok. Delapan variabel berlabel A, B, C, ..., H nilainya terus berubah selama iterasi sebanyak 64 kali putaran.

$$T_1 = H + \sum_1(E) + Ch(E,F,G)[1] + K_t + W_t \quad (1)$$
$$T_2 = \sum_0(A) + Maj(A,B,C)[1] \quad (2)$$
$$H = G \quad (3)$$
$$G = F \quad (4)$$
$$F = E \quad (5)$$
$$E = D + T_1 \quad (6)$$
$$D = C \quad (7)$$
$$C = B \quad (8)$$
$$B = A \quad (9)$$
$$A = T_1 + T_2 \quad (10)$$

Gambar 2.2 Perhitungan blok

6. Setelah itersai sebanyak 64 kali, nilai *hash* $H^{(i)}$ lalu dihitung seperti yang ditunjukkan gambar 2.3.

$$H_0^{(i)} = a + H_0^{(i-1)}$$
$$H_1^{(i)} = b + H_1^{(i-1)}$$
$$H_2^{(i)} = c + H_2^{(i-1)}$$
$$H_3^{(i)} = d + H_3^{(i-1)}$$
$$H_4^{(i)} = e + H_4^{(i-1)}$$
$$H_5^{(i)} = f + H_5^{(i-1)}$$
$$H_6^{(i)} = g + H_6^{(i-1)}$$
$$H_7^{(i)} = h + H_7^{(i-1)}$$

Gambar 2.3 Perhitungan nilai *hash* $H^{(i)}$

7. Selanjutnya *output* SHA256 diperoleh dari penggabungan delapan variabel yang sebelumnya telah dikomputasi.

2.9 *Proof of Work*

Blockchain memiliki beberapa protokol konsensus yang ditawarkan seperti *Proof of Stake, Proof of Elapsed Time,* dan PBTS namun sebagian besar *blockchain* yang ada menerapkan mekanisme *Proof of Work (PoW)* yang dikomputasi. *Proof of Work (PoW)* yang diterapkan pada Bitcoin dipakai untuk melindungi *blockchain* buku besar dari perubahan yang tidak dikehendaki. Seperti yang dikemukakan oleh Gemeliarana dan Sari (2018), *PoW* pada Bitcoin beroperasi dengan cara berikut: Seluruh informasi yang terdapat dalam blok kandidat dihitung berdasarkan nilai *hash*-nya. Nilai *hash* yang dibuat harus memenuhi persyaratan tingkat kesulitan yang ditetapkan oleh sistem. Apabila nilai *hash* tidak memenuhi persyaratan, maka perhitungan akan diiterasi dengan

mengubah nilai *nonce*. *Nonce* merupakan nilai yang tidak mempunyai makna, namun dimaksudkan untuk ditambahkan ke blok guna menghasilkan nilai *hash* yang sesuai dengan kondisi. Apabila nilai *hash* belum memenuhi nilai ketentuan, nilai *nonce* akan diganti lagi hingga penambang mendapatkan nilai *hash* yang memenuhi persyaratan (Gervais, 2016). Selain memakai nilai *nonce*, para penambang juga dapat menggunakan nilai basis koin, transaksi awal dalam satu blok. Penambang bisa menambah atau mengganti informasi yang terkandung dalam *coinbase*, sehingga mengharuskan untuk variasi nilai *hash* yang lebih besar. Hal tersebut dijalankan karena *nonce* hanya memiliki panjang 32 bit (4 byte) yang bisa jadi tidak cukup untuk menciptakan nilai *hash* kurang dari target.

PoW yang diterapkan pada sistem Bitcoin umumnya disebut sebagai proses penambangan. Proses penambangan pada Bitcoin merupakan cara untuk menjalankan perhitungan menggunakan fungsi *hash* seperti *Hashcash* supaya blok baru bisa diterima ke dalam *blockchain*. Saat hasilnya dapat diterima, blok baru ditambahkan ke *blockchain*. *PoW* berfungsi untuk melindungi *blockchain*. Dengan tingkat kesulitan yang tinggi, seseorang yang bermaksud mengubah transaksi yang sudah dicatat dalam satu blok, wajib melakukan perhitungan ulang terhadap blok tersebut dan juga blok-blok selanjutnya. Hal tersebut harus dilakukan karena setiap blok terhubung ke blok lain dan membentuk rantai. Sehingga saat sebuah tautan akan berganti, tautan selanjutnya juga harus diganti. Oleh sebab itu, transaksi dalam Bitcoin sangat susah untuk dibatalkan apabila blok telah ditambahkan ke dalam blockchain (Gervais, 2016)(Gervais dkk., 2015).

Penambang Bitcoin merupakan mereka yang mempersiapkan peralatan komputasi guna membantu menciptakan perhitungan dalam pembuatan blok baru. Penambang Bitcoin akan memperoleh penghargaan dalam bentuk Bitcoin atas usaha yang dilakukan. Penambang berkomunikasi dengan simpul Bitcoin guna memperoleh informasi mengenai transaksi baru yang harus ditambahkan ke dalam blok baru. Bitcoin merupakan salah satu contoh penerapan *PoW* berbasis *hash* yang memerlukan penentuan nilai *nonce*, sehingga saat *hash* dengan parameter blok tambahan, nilai *hash* harus kurang dari nilai target yang ditentukan. Apabila tidak bisa ditemukan, penambang membuat blok dan meneruskannya pada lapisan jaringan pada rekan-rekan. *Peer* lain di dalam jaringan dapat memverifikasi *PoW* dengan cara menghitung nilai *hash* dari blok dan memeriksa apakah nilainya telah memenuhi ketentuan untuk lebih kecil dari nilai target (Eyal dan Sirer, 2014).

2.9.1 *Block Interval*

Interval blok menentukan latensi ketika konten ditulis ke *blockchain*. Semakin kecil interval blok, maka semakin cepat transaksi dikonfirmasi dan semakin tinggi kemungkinan *stale blocks*. Penyesuaian interval blok secara langsung terkait dengan perubahan dalam mekanisme *PoW* yang melandasinya. Kesulitan yang lebih rendah menciptakan lebih banyak blok dalam jaringan, tetapi hasil yang lebih sulit menciptakan lebih sedikit blok pada kondisi waktu yang sama. Oleh sebab itu, penting untuk mengkaji apakah kesulitan yang berubah berpengaruh pada kemampuan serangan terhadap rantai terpanjang, dimana hal tersebut adalah bagian utama keamanan pada *blockchain* berbasis *PoW* (Gemeliarana dan Sari, 2018).

2.9.2 Ukuran Blok

Ukuran blok maksimum secara langsung menentukan jumlah transaksi maksimum yang dikerjakan dalam satu blok. Ukuran ini mengendalikan *throughput* yang didapat oleh sistem. Blok besar menciptakan kecepatan jeda waktu yang lebih lambat, dimana pada gilirannya meningkatkan laju *stale blocks* dan melemahkan keamanan *blockchain* (Gemeliarana dan Sari, 2018).

2.9.3 Mekanisme Penyebaran Informasi

Sistem manajemen permintaan blok menentukan bagaimana informasi dikomunikasikan kepada rekan-rekan dalam jaringan. Semua rekan kerja diharapkan menerima semua blok, protokol *broadcast* dibutuhkan. Pilihan protokol *broadcast* yang melandasinya jelas memengaruhi ketahanan dan skalabilitas jaringan (Gemeliarana dan Sari, 2018).

2.9.4 *Stale Blocks*

Stale blocks ditunjukkan pada blok yang tidak termasuk dalam rantai terpanjang, misalnya, karena persaingan, dan pertentangan. *Stale blocks* merusak keamanan dan kinerja *blockchain* sehingga menyebabkan garpu rantai, kondisi tidak konsisten yang memperlambat pertumbuhan rantai utama dan menghasilkan kinerja serta keterlibatan keamanan yang signifikan. Di satu sisi, *stale blocks* meningkatkan manfaat musuh pada jaringan. Di sisi lain, *stale blocks* menghasilkan *overhead bandwidth* tambahan dan umumnya tidak diberikan untuk penambangan (Gervais, 2016).

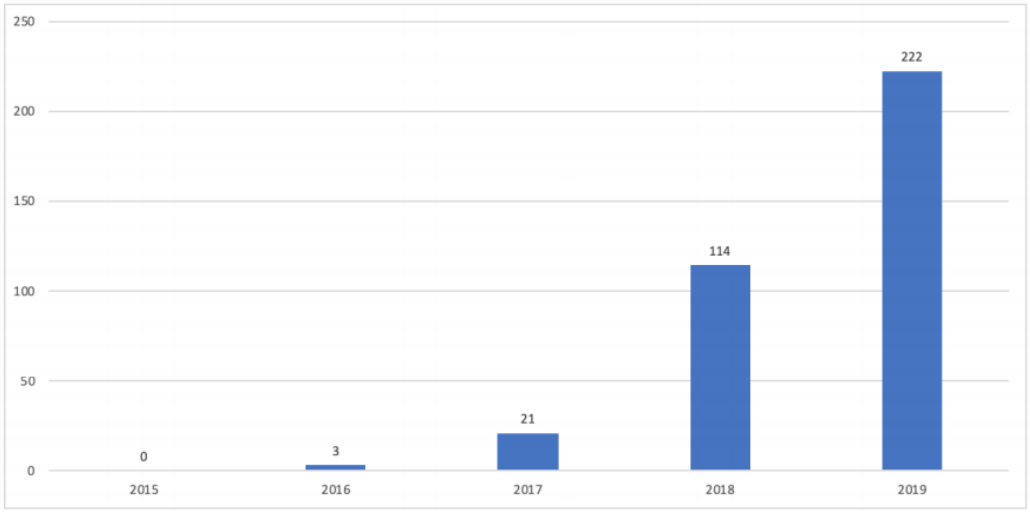
2.10 Penelitian Terkait

Peneliti mencoba mencari tahu arah penelitian mengenai implementasi *blockchain* pada dunia transportasi. Hal ini sangat penting untuk menemukan beberapa referensi mengenai penelitian-penelitian yang sudah dilakukan sebelumnya. Peneliti menemukan makalah dengan judul *'A Review of Blockchain-Based Systems in Transportation'*. Penelitian ini dilakukan oleh Astarita dkk. (2019), dimana pada makalah ini diklasifikasikan makalah-makalah penelitian menjadi 2 *cluster* utama terkait penelitian *blockchain* dalam transportasi. klasifikasi yang dilakukan pada makalah tersebut bertujuan untuk mengidentifikasi penerapan metodologi yang digunakan pada tren penelitian saat ini, mengetahui kesenjangan utama dalam literatur dan kemungkinan tantangan yang akan terjadi di masa yang akan datang. Adapun 2 *cluster* utama pada makalah tersebut, yang pertama adalah rantai pasokan dan logistik sedangkan yang ke dua adalah manajemen lalu lintas jalan dan kota pintar. Pada penelitian yang dilakukan Astarita dkk. (2019) tersebut dihimpun beberapa makalah dari *scopus* terpilih dengan penerbit *Springer*, *Elsevier, Taylor & Francis, IEEE* dan *Emerald*. Dalam penelitian tersebut dipilih beberapa kata kunci pencarian dengan tujuan membangun tinjauan literatur yang baik di sektor transportasi. Peneliti tersebut mengidentifikasi sejumlah besar kata kunci yang cukup untuk digabungkan dengan istilah *blockchain*. Kata kunci tersebut adalah: lalu lintas, logistik, rantai pasokan, bandara dan transportasi. Tabel hasil pencarian pertama penelitian pada makalah tersebut ditunjukkan pada gambar 2.4.

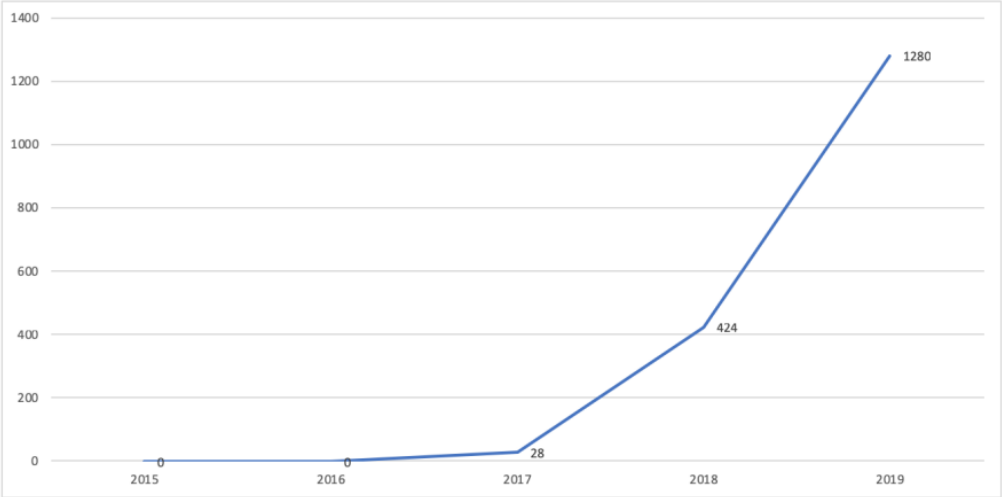
Search Keywords	Results (No. of Documents)
Blockchain AND Supply Chain	522
Blockchain AND Traffic	190
Blockchain AND Transportation	138
Blockchain AND Logistic	122
Blockchain AND Transport	72
Blockchain AND Airport	6
Total	1050

Gambar 2.4 Hasil pencarian pertama

Pada makalah tersebut disebutkan juga jumlah publikasi dan kutipan tentang penerapan teknologi *blockchain* di sektor transportasi dari beberapa tahun terakhir. Jumlah publikasi tentang *blockchain* dan sektor transportasi ditunjukkan pada gambar 2.5 dan jumlah kutipan pada *blockchain* dan sektor transportasi ditunjukkan pada gambar 2.6.



Gambar 2.5 Jumlah publikasi tentang *blockchain* dan sektor transportasi



Gambar 2.6 Jumlah kutipan pada *blockchain* dan sektor transportasi

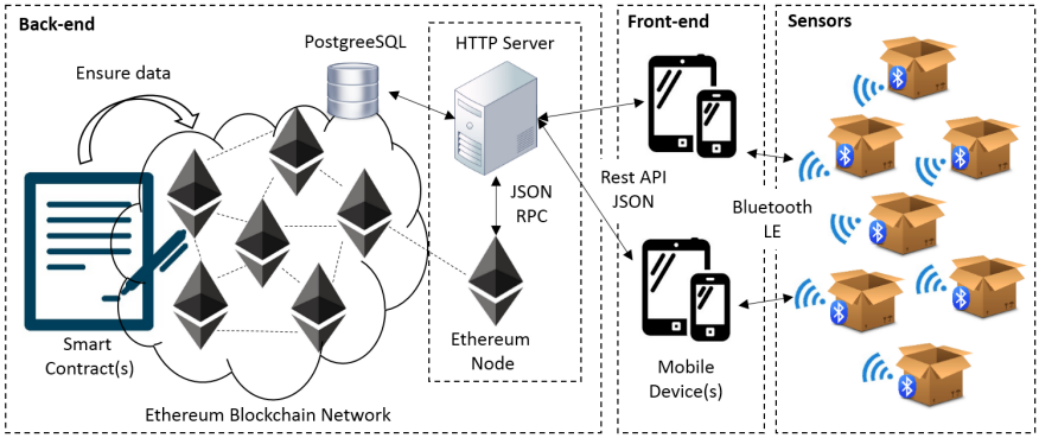
Dalam tiga tahun terakhir telah terjadi perkembangan yang cukup besar dalam *blockchain* di sektor transportasi baik secara teoritis maupun praktis. Antara tahun 2018 dan tahun 2019, jumlah makalah yang diterbitkan hampir dua kali lipat, sementara jumlah kutipan meningkat tiga kali lipat. Tren ini menunjukkan penelitian tentang *blockchain* dalam transportasi memiliki artikel terkini yang cukup banyak untuk menjadi referensi dalam penelitian tersebut (Pabla dkk., 2019).

Secara rinci, untuk klaster pertama, dimana sebagian besar terkait dengan masalah keterlacakan dalam pasokan dan logistik, satu-satunya makalah yang memberikan informasi implementasi nyata berdasarkan *blockchain* baru adalah penelitian Lu dkk. (2017) dengan judul makalah *"Adaptable Blockchain-Based Systems: A Case Study for Product Traceability"*, dimana dalam makalah tersebut diperkenalkan *blockchain* yang dikembangkan secara pribadi yang disebut OriginChain. Makalah penelitian Bocek dkk. (2017) dan makalah penelitian Caro

dkk. (2018) juga mengusulkan implementasi nyata berdasarkan *blockchain* yang ada. Bocek dkk. (2017) menggunakan *Ethereum* sedangkan Caro dkk. (2018) menggunakan *Ethereum dan Hyperledger*. Karya-karya lain di klaster pertama mengusulkan bukti konsep dan sisanya hanya menyajikan pada tingkat konsep. Dalam klaster ke dua penelitian *blockchain* dalam transportasi, belum ada implementasi nyata atau bukti konsep yang disajikankan secara komputasi. Hal ini menunjukan bagaimana implementasi teknologi *blockchain* sebagai solusi masalah lalu lintas jalan masih dalam tahap awal. Dari makalah-makalah yang ada, teknologi *blockchain* sangat mungkin diterapkan dalam berbagai bidang, seperti pelacakan dan penelusuran makanan, kepatuhan terhadap peraturan, sistem keamanan data, serta pencocokan antara permintaan dan penawaran.

Penelitian Bocek dkk. (2017) membahas implementasi penggunaan *blockchain* pada rantai pasokan di lingkup farmasi. Dalam makalah tersebut dikatakan bahwa banyak pemangku kepentingan yang terlibat dalam rantai pasokan, dimana penggunaan *blockchain* dapat digunakan untuk mengotomatisasi proses guna penghematan biaya. Tujuan dari makalah tersebut adalah menunjukkan beberapa area untuk pemanfaatan *blockchain* dan memberikan rincian mengenai aplikasi yang mereka buat dengan nama *modum.io AG* dalam menggunakan *blockchain* pada rantai pasokan farmasi. Pada aplikasi yang mereka buat, suhu dapat dinilai secara otomatis dan memberi tahu pengirim dan penerima menggunakan kontrak pintar. Disebutkan juga pada makalah tersebut bahwa distribusi produk obat untuk manusia sangat diatur sehingga mengelola logistik produk-produk dari pusat distribusi hingga pengiriman kepada pengguna

merupakan tugas paling penting dan melibatkan banyak perantara. Pelaporan penyimpangan apapun termasuk suhu ke distributor dan penerima produk obat yang terkena dampak serta pemantauan suhu setiap paket menjadi suatu hal yang wajib. Teknologi *Blockchain* memberikan konsensus yang terdesentralisasi dan tepercaya dimana data produk medis selama proses logistik dapat disimpan dan diakses oleh kedua pihak yang dipastikan dengan kontrak pintar. Gagasan mengenai kontrak pintar yakni memiliki aturan atau kode yang mewakili kontrak yang dijalankan sendiri dan pelibatan pihak ketiga yang tepercaya tidak diperlukan karena sudah digantikan dengan sistem konsensus yang disediakan oleh *blockchain*. Adapun arsitektur *modum.io AG* disusun menjadi perangkat sensor *back-end, front-end,* dan *IoT*, sebagaimana diuraikan dalam gambar 2.7.



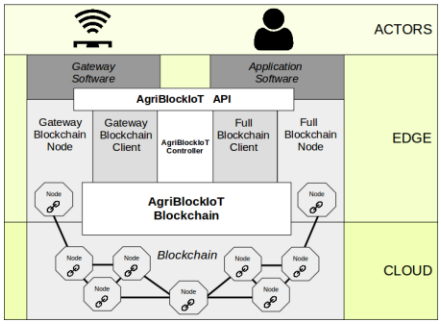
Gambar 2.7 Desain sistem *Modum.io*

Pemantauan suhu dimulai dengan klien Android. Untuk memulai proses, perangkat sensor diharuskan berada dalam jangkauan *Bluetooth*. Sebagai langkah pertama, *current track* dan *trace number* pada paket harus dikaitkan dengan alamat *MAC* perangkat sensor. Karena keduanya, *current track* dan *trace number*

serta alamat *MAC* merupakan *barcode*, masing-masing kode QR, klien Android menangkap keduanya menggunakan kamera. Setelah proses ini, klien Android mulai melalui *Bluetooth LE* pengukuran suhu pada perangkat sensor, dan mengirimkan nomor *track* and *trace* atau asosiasi alamat *MAC* ke *server*. Sensor juga menyimpan nomor lacak jika ada akses *server*. Dengan demikian, paket yang telah dikirim, selalu memiliki hubungan antara alamat *MAC*-nya serta *current track* dan *trace number*. *Server* menyimpan asosiasi dan membuat, menyiarkan kontrak pintar, dan menyimpan ID kontrak pintar pada perangkat sensor. Perangkat sensor dapat diletakkan di dalam paket produk medis. Perangkat sensor merekam setiap 10 menit suhu dan menyimpannya dalam memori internal pada perangkat sensor. Setelah menerima paket di tujuan, *track* dan *trace number* dipindai. Klien Android meminta Alamat *MAC* dari *server* untuk terhubung ke perangkat sensor. Kemudian klien Android secara otomatis mengunduh semua data suhu sekaligus melalui *Bluetooth LE,* dan mengirimkannya ke kontrak pintar. Setelah kontrak pintar memeriksa suhu, siapa pun yang tertarik dengan kontrak pintar itu dapat memverifikasi apakah suhu tersebut dalam spesifikasinya langsung di *blockchain Ethereum*. Dengan demikian, pengirim akan segera diberitahu tentang hasil tersebut.

Penelitian Caro dkk. (2018) membahas implementasi penggunaan *blockchain* pada rantai pasokan di lingkup pertanian dan makanan. Dalam makalah tersebut disajikan aplikasi *AgriBlockIoT*, dimana itu menjadi solusi keterlacakan berbasis *blockchain* yang seluruhnya terdesentralisasi untuk manajemen rantai pasokan *Agri-Food*. Solusi yang diusulkan menggunakan

Ethereum dan *Hyperledger Sawtooth* yang tersedia untuk implementasi *blockchain* untuk umum. Sistem yang dibuat dapat mengintegrasikan berbagai perangkat sensor *IoT. AgriBlockIoT* menjamin keterlacakan aset yang transparan dan dapat diaudit. Untuk menilai kelayakan solusi yang diusulkan, Caro dkk. (2018) merekayasa dan mengerahkan kasus penggunaan dari *farm-to-fork* yang disebut skenario keterlacakan makanan klasik yang mendorong keterlacakan makanan bersertifikat di sepanjang rantai pasokan keseluruhan, misalnya, dari produksi pertanian (*farm*) untuk konsumsi (*fork*). Kemudian, Caro dkk. (2018) membandingkan dua implementasi, dalam hal tiga metrik kinerja, yaitu latensi, beban CPU, dan penggunaan jaringan. Dalam penelitian tersebut diusulkan arsitektur berlapis yang dapat mengandalkan teknologi *Blockchain* dan *IoT* untuk mencapai transparansi, kemampuan audit, dan catatan tetap yang disimpan dalam lingkungan yang tidak dapat dipercaya. Arsitektur yang diusulkan mengambil keuntungan dari peningkatan kemampuan yang ditawarkan oleh perangkat tepi modern yang dapat langsung digunakan sebagai *node* penuh dari implementasi *blockchain* berlapis tersebut, sehingga memperluas resistensi, desentralisasi, keamanan dan kepercayaan seluruh jaringan. Modul utama *AgriBlockIoT* adalah *API, Controller,* dan *blockchain*.

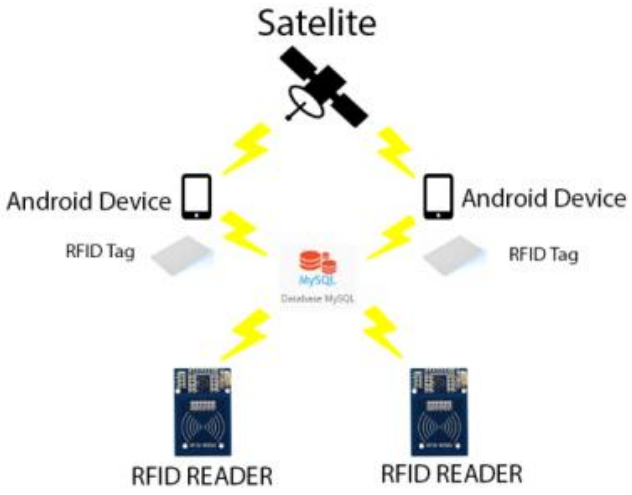


Gambar 2.8 Desain sistem *AgribotIoT*

AgriBlockIoT merupakan *blockchain*-agnostik, dimana itu mengimplementasikan modul *blockchain* yang mendasari lebih dari dua implementasi yang berbeda, privat, enam *node*, yaitu *Ethereum* dan *Hyperledger Sawtooth*. Alasan memilih implementasi ini adalah tingkat penyesuaian yang berbeda untuk catatan yang termasuk dalam buku besar (transaksi). Sementara kedua platform memungkinkan untuk menerapkan logika bisnis yang kompleks, *Ethereum* bekerja dengan struktur transaksi tunggal, sementara *Hyperledger Sawtooth* memungkinkan definisi struktur transaksi khusus. Serangkaian 100 tes yang dijalankan secara independen untuk setiap skenario. Selama setiap pengujian, *AgriBlockIoT* cukup menetapkan nilai sensor, seperti yang dilakukan oleh perangkat penginderaan *IoT* lingkungan melalui gateway, dan mengeluarkan transaksi di *blockchain*. Untuk setiap pengujian, Caro dkk. (2018) mengukur waktu yang diperlukan untuk menetapkan nilai dalam *blockchain* (latensi), kekuatan pemrosesan setiap *node* (beban CPU), dan penggunaan jaringan (dalam hal *byte* yang dikirim dan diterima).

Penelitian Hutabarat dkk. (2016) membahas implementasi penggunaan RFID dan GPS untuk pelacakan manusia di dalam dan di luar ruangan. Dalam penelitian tersebut disebutkan bahwa mekanisme pelacakan di dalam ruangan yakni, tag RFID dibawa oleh pengguna dan terus dibaca setiap kali mengakses ruangan. Sedangkan untuk di luar ruangan, GPS digunakan selagi pengguna tetap berada di luar ruangan. GPS akan secara otomatis diaktifkan setiap kali pengguna meninggalkan ruangan dengan jarak 3 meter. Untuk area *indoor*, pembaca RFID diletakkan di setiap ruangan dan pengguna harus mengetuk tag RFID nya untuk

diidentifikasi. Tetapi, GPS secara otomatis akan mengidentifikasi posisi pengguna setiap kali pengguna meninggalkan ruangan. Desain sistem dari penelitian tersebut dapat dilihat pada gambar 2.6 dimana digunakan RFID untuk pelacakan di dalam ruangan dan GPS untuk pelacakan di luar ruangan. Tag RFID pasif dipakai dan pengguna harus mengetuk tag RFID kepada pembaca setiap kali pengguna memasuki ruangan. *User ID* (UID) akan diverifikasi *online* melalui *server database* setelah pengguna mengetuk tag RFID ke pembacanya. Data dari input GPS disediakan oleh *Google Maps* sementara peralatan transmisi dan penerima menggunakan *smarthphone.*



Gambar 2.9 Desain sistem pelacakan

Jarak aktif antara tag RFID dan pembaca RFID adalah 50 mm, karena Hutabarat dkk. (2016) menggunakan tag pasif dalam bentuk kartu atau gantungan kunci. Pembaca RFID akan tetap dalam keadaan siaga hingga mendeteksi ketukan dari tag RFID. Saat terdapat proses penyadapan, pembaca RFID akan membaca UID dan dikirim ke *server* basis data untuk verifikasi. Setelah verifikasi tersebut

valid, pembaca RFID akan mengirimkan waktu akses dan posisi pengguna ke *server* basis data. Proses yang sama juga berjalan pada GPS. Koordinat lokasi pengguna akan dikirim ke *server* basis data saat pengguna menyetujui untuk melacak posisi tersebut. program akan terus aktif hingga salah satu atau keduanya baik RFID maupun GPS menerima sinyal untuk diidentifikasi atau diverifikasi. Saat status GPS adalah 1 dan status RFID adalah 0, pengguna akan menerima peta dari *Google*. Tetapi, saat status GPS adalah 0 dan status RFID adalah 1, program akan memvalidasi UID sebelum akhirnya mengirim waktu dan posisi pengguna. Jika kedua status adalah 1 maka sistem akan memilih kondisi RFID untuk dijalankan.

Penelitian Pabla dkk. (2019) membahas implementasi penggunaan *blockchain* untuk keamanan data pada sistem pelacakan dan pemantauan kesehatan serta posisi tentara pada medan perang. Dalam penelitian tersebut dijelaskan bahwa sistem yang dibuat terdiri dari modul sensor yang dipasang pada lengan tentara guna pemantauan kesehatan dan posisi waktu nyata, dimana perangkat tersebut mentransmisikan dan menyimpan data tentara dalam bentuk terenkripsi di *Blockchain* sehingga data dapat terdistribusi. Modul sensor yang dipasang terdiri dari modul suhu, detak jantung, dan GPS. *Blockchain* akan ada di komputer yang dikendalikan oleh tentara. Sistem yang dibuat terdiri dari modul tentara dengan perangkat *IoT* yang mudah tersedia untuk terus mengirim data kesehatan dan posisi tentara yang bertugas ke stasiun pangkalan guna memantau tingkat stres dan keadaan mereka secara keseluruhan serta memastikan tindakan cepat dalam bentuk apa pun. sistem yang dibuat terdiri dari modul sensor yang

dirangkai menjadi satu unit, ringan, berbiaya rendah untuk tentara yang akan memantau detak jantung, suhu masing-masing dari setiap pasukan di medan perang dengan andal, sambil menampilkan nilai suhu dan detak per menit pada layar LCD 20x4 *onboard*, dan secara bersamaan mentransmisikannya ke pusat kendali melalui GSM untuk disimpan dalam *blockchain*. Di sisi lain, sistem yang dibuat menjadi fasilitas untuk menangani kebutuhan medis instan tentara dan pertukaran perintah taktik. Perangkat keras yang digunakan adalah *Raspberry Pi*, Sensor Detak Jantung, Sensor Suhu, Catu daya, ADC atau Arduino, layar LCD, modul GSM, modul GPS, tombol, Resistor, dan kabel. Adapun tahapan dari sistem yang dibuat adalah akuisisi data, pelacakan lokasi dan pengiriman data serta keamanan data.

Berdasarkan penelitian yang dipublikasikan sebelumnya, diketahui bahwa penggunaan *blockchain* dan perangkat elektronik dapat membantu melakukan tugas pelacakan dan menjadi solusi keamanan data mobilitas orang di suatu wilayah. *Blockchain* merupakan struktur data terdistribusi, dimana setiap blok penyusunnya berisi *hash* dari representasi blok sebelumnya yang menghasilkan rantai blok. Transaksi historis dalam *blockchain* tidak dapat dihapus atau diubah tanpa membatalkan rantai *hash*. Digabungkan dengan kerumitan komputasi dan skema insentif untuk pembuatan blok, hal ini dapat mencegah perusakan dan revisi informasi dalam *blockchain* (Lu dan Xu, 2017). Oleh karena itu, pada penelitian ini akan dibuat sistem keamanan mobilitas orang di kota Malang berbasis *blockchain*.

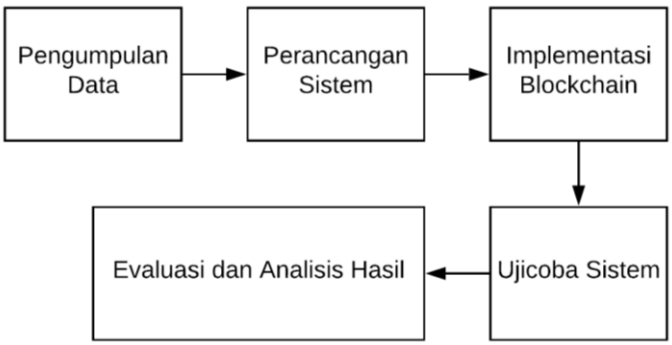
BAB III

DESAIN SISTEM

Pada bagian ini akan diuraikan mengenai analisa dan perencanaan sistem dari riset ini. Adapun yang akan diuraikan, yakni tahapan penelitian yang dikerjakan, detail cara kerja sistem yang meliputi bagian input, proses dan output serta kalkulasi kemungkinan peretasan pada sistem berbasis *blockchain*.

3.1 Alur Penelitian

Pada bagian ini, digambarkan tahap apa saja yang akan ditempuh dalam melakukan penelitian ini. Alur dari penelitian ini direpresentasikan pada gambar di bawah ini.



Gambar 3.1 Diagram Blok Alur Penelitian

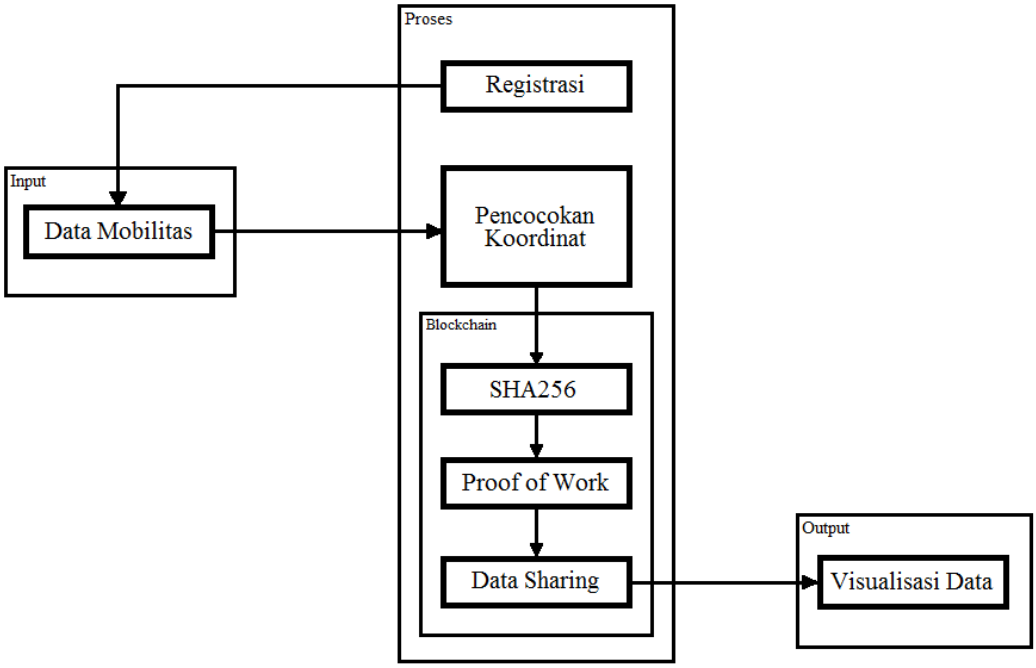
Jika dilihat pada gambar 3.1 di atas, dapat diketahui bahwa langkah-langkah apa saja yang harus dilakukan untuk memperoleh hasil dari penelitian. Langkah awal yang dikerjakan adalah pengumpulan data untuk penelitian, data ini berbentuk data *range* koordinat lokasi yang ditentukan. Selain sebagai informasi, bab tinjauan pusaka juga berfungsi sebagai dasar dalam melakukan penelitian, sehingga hal tersebut dapat menjadi pembelajaran guna menunjang keberhasilan

penelitian ini. Langkah selanjutnya yang harus dilakukan yakni proses perancangan sistem, pada proses ini akan diuraikan bekerjanya sistem yang akan dibuat. Tahap selanjutnya adalah bagaimana menerapkan metode dan algoritma yang akan dipakai pada sistem ini. Sebagaimana judul penelitian ini, yakni akan diciptakan sistem berbasis *blockchain*. Pada penelitian ini *blockchain* akan diimplementasikan untuk keamanan data mobilitas seseorang. *Blockchain* dipilih karena strukturnya yang terdapat blok berisi *hash* dari representasi blok sebelumnya, sehingga menciptakan rantai. Maka catatan transaksi dalam *blockchain* tidak dapat dihapus atau diubah tanpa membatalkan rantai *hash*. Dikombinasikan dengan kerumitan komputasi dan skema insentif untuk pembuatan blok, hal ini dapat mencegah perusakan dan revisi informasi dalam *blockchain* (Lu dan Xu, 2017). Oleh sebab itu penulis berharap dengan menggunakan teknik ini, dapat dibangun sistem yang mampu menjaga keamanan data mobilitas orang dari potensi serangan.

Apabila metode berhasil diimplementasikan pada sistem yang dibuat, maka akan dilakukan uji coba pada sistem. Uji coba pada sistem ini bermaksud untuk mengetahui apakah fungsi dari sistem ini berjalan dengan baik sehingga nantinya akan diperoleh data yang dapat dianalisis agar ditemukan kelemahan dan kelebihan sistem. Hal ini sangat membantu untuk upaya pengembangan sistem menjadi lebih baik. Ulasan, penilaian, catatan dan analisis dari hasil uji coba ini dibutuhkan untuk mengkaji dan mengetahui seberapa besar kinerja sistem dalam mengamankan data mobilitas dan pelacakan mobilitas orang.

3.2 Detail Cara Kerja Sistem

Pendekatan yang diusulkan untuk sistem keamanan data mobilitas manusia yakni dengan menyimpan informasi lokasi dari data GPS secara umum dengan penggunaan algoritma sha256 dan *PoW*. Desain sistem ini menjadi deskripsi umum seperti apa kerja sistem yang akan dibuat. Letak implementasi *blockchain* dapat dilihat dari diagram blok desain sistem. Secara sederhana desain dari sistem keamanan mobilitas orang berbasis *blockchain* ini ditunjukkan pada gambar 3.2.



Gambar 3.2 Diagram Blok Desain Sistem

Dalam penelitian ini, mobilitas manusia diasumsikan dalam bentuk transportasi karena adanya perangkat yang harus melekat dan terbawa kemanapun mereka pergi bersama kendaraannya. Pendekatan yang diusulkan terdiri dari tiga bagian utama yaitu:

a) Akuisisi data perpindahan lokasi

b) Transmisi data mobilitas

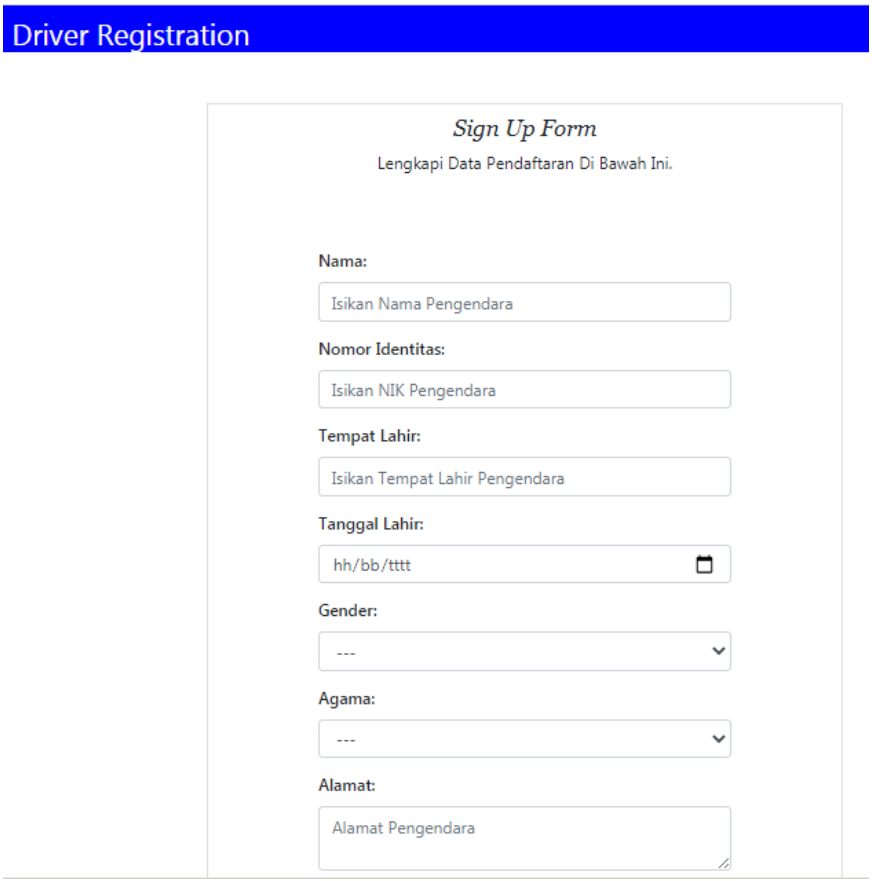
c) Keamanan data mobilitas

Dari bagian utama sistem ini, akan dijelaskan lebih rinci. Berikut akan dijelaskan detail proses yang terjadi pada setiap bagian.

3.2.1 Input

Data inputan berupa data mobilitas pengendara sebagai *node* dalam jaringan *blockchain*. Sebelum dapat melakukan pengumpulan data, pengendara terlebih dahulu melewati proses registrasi sebagi *node* baru dalam jaringan *blockchain*. Adapun data mobilitas pengendara meliputi: index, *previous hash*, *timestamp*, koordinat lintang dan garis bujur, *hash*, serta *nonce*.

Pada tahap awal, pemilik kendaraan yang akan dipasangkan *prototype* alat ini harus melakukan registrasi. Adapun data dalam proses registrasi tersebut meliputi: ID pengendara, nama, nomor identitas, tempat dan tanggal lahir, *gender*, alamat, agama, pekerjaan dan nomor telepon. Pada saat proses registrasi selesai secara otomatis pengendara tersebut juga telah memiliki *genesis block* yakni blok pertama dalam rantai *blockchain*. Identitas setiap pengendara disimpan dalam basis data kemudian perubahan koordinat lokasi pengendara digunakan untuk pelacakan mobilitas setiap pengendara yang telah terdaftar. Adapun tampilan halaman registrasi pengendara ditunjukkan pada gambar 3.3.



Gambar 3.3 Tampilan Halaman Registrasi Pengendara

3.2.2 Proses

3.2.2.1 Pencocokan Koordinat

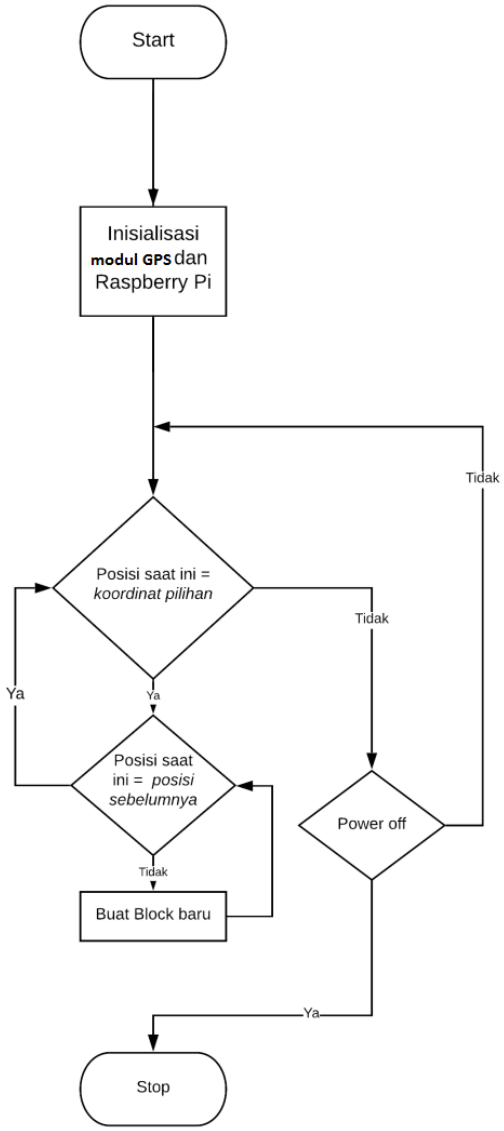
Proses diawali dengan tahap registrasi pengendara sebagai *node* baru dalam jaringan *blockchain*. Pada penelitian kali ini, untuk keperluan transaksi data dan pelacakan mobilitas orang di kota Malang, diperlukan data *range* koordinat dari lokasi-lokasi yang ditentukan dimana penentuan lokasi tersebut didasarkan atas tingkat keramaian dan aktivitas perpindahan orang di kawasan kota Malang. Data lokasi dan koordinat yang akan dipakai dalam penelitian ini diperoleh dari BPS kota Malang dan *Google Maps* Data Peta 2020. Koordinat lokasi pilihan akan tetap digunakan selama tidak ada perubahan pada penentuan lokasi pilihan.

Dari data koordinat pilihan tiap lokasi di kota Malang nantinya akan dilakukan pencocokan antara koordinat *real time node.* Sesudah dilakukan proses pencocokan koordinat, kemudian tahap selanjutnya ialah dilakukan proses *data sharing* atau *broadcast* sebagai bagian dari penerapan *blockchain*.

Tabel 3.1 Data Koordinat pada Tiap Lokasi Pilihan di kota Malang

No.	Lokasi Pelacakan	Koordinat pilihan
1.	Terminal Landungsari	{ -7.924741 > x > -7.926080 }, { 112.597417 < y < 112.598388 }
2.	Terminal Arjosari	{ -7.932930 > x > -7.933692 }, { 112.658108 < y < 112.659054 }
3.	Stasiun Kota Baru	{ -7.975806 > x > -7.976433 }, { 112.636307 < y < 112.636811 }

Perangkat yang dibuat menggunakan sumber daya 5 volt dan 2.1 ampere sebagai catu daya pada *Raspberry Pi* (Pi 3 model B),dimana sumber daya tersebut tersambung melalui kabel USB. Modul GPS digunakan untuk memperoleh data koordinat pengendara. Modul GPS terhubung ke Pi melalui pin GPIO. Modul program dibuat menggunakan Python. Pi menggunakan *Library* untuk menangkap output dari modul GPS. Demi keamanan guna mencegah pencurian data, data yang akan disimpan dienkripsi menggunakan algoritma SHA256. Proses akuisisi data ditunjukkan pada gambar 3.4.



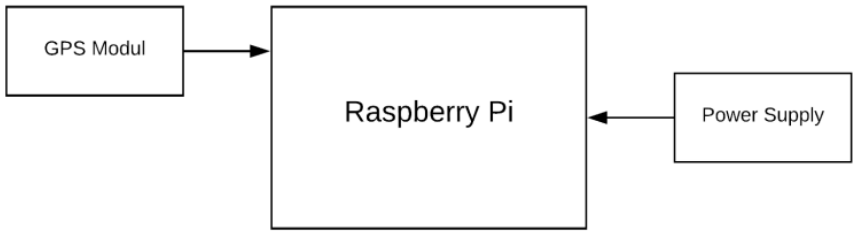
Gambar 3.4 *Flowchart* proses akuisisi data

3.2.2.2 Pelacakan Lokasi dan Transmisi Data Mobilitas

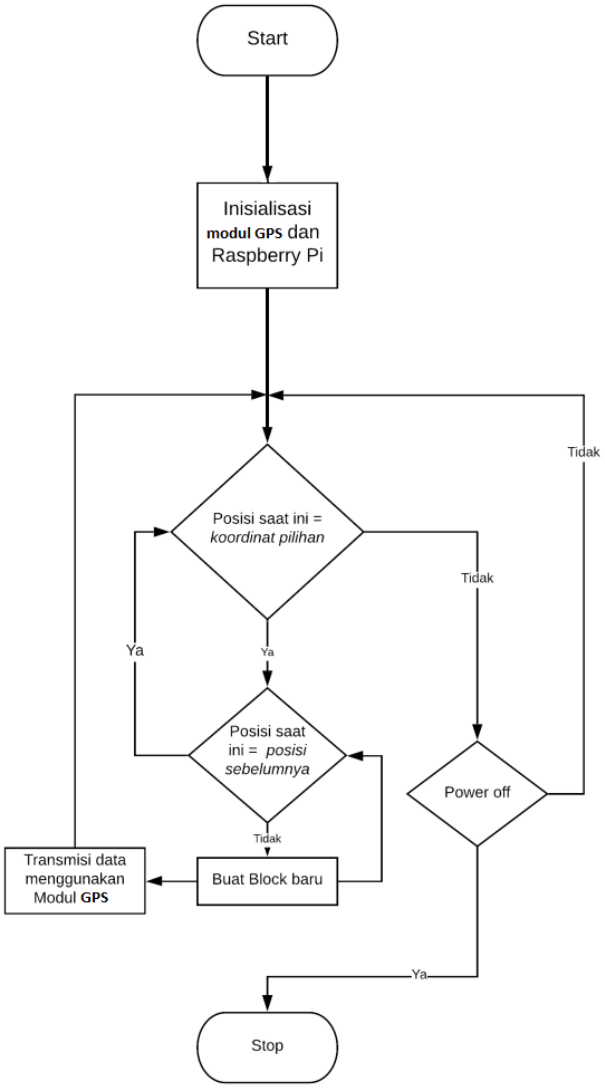
Modul GPS digunakan sebagai perangkat transmisi data dengan kecepatan yang baik menggunakan jangkauan efektif. Informasi mengenai perubahan posisi *node* dibutuhkan untuk dikirim ke *server*. Modem GPS akan menangkap nilai garis lintang dan bujur dari posisi pengendara sebagai *node* saat ini dalam waktu nyata di kota Malang. Perangkat keras lain yang digunakan adalah Kabel *jumper*

female to female sebagai penghubung antara modul GPS dan *Raspberry Pi* serta *Powerbank* sebagai catu daya 5 *volt*. Sistem ini mengirimkan informasi lokasi per menit pada program *python* di dalam *Raspberry Pi,* kemudian dilakukan pencocokan koordinat lokasi. Apabila koordinat sesuai, data akan melalui proses *hashing* sebelum disimpan dalam *blockchain*. Sistem yang diusulkan ini efektif dalam memonitor lokasi *node* di kota Malang. Secara keseluruhan, perangkat keras yang dipertimbangkan adalah *Raspberry Pi*, catu daya, modul GPS, dan kabel *jumper female to female*.

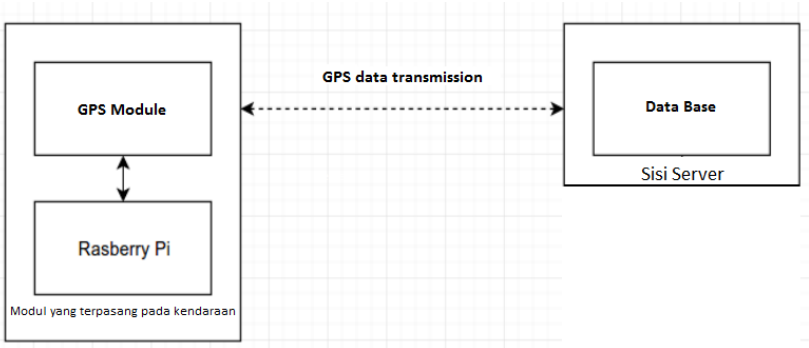
Modul GPS (dengan chip SIM28M) dapat dihubungkan langsung ke *Raspberry Pi*. Hal ini dilakukan karena sangat mempertimbangkan ramping atau tidaknya sirkuit yang dibuat. Perangkat lunak yang digunakan dalam modul ini adalah *Python* (*scripting* untuk modul GPS dan *Raspberry Pi*). Adapun diagram blok menggambarkan hubungan antara komponen perangkat keras yang diperlukan untuk memperoleh informasi terkait lokasi *node* ditunjukkan pada gambar 3.5.



Gambar 3.5 Diagram blok *Hardware*



Gambar 3.6 *Flowchart* mewakili transaksi dari proses akuisisi data lokasi



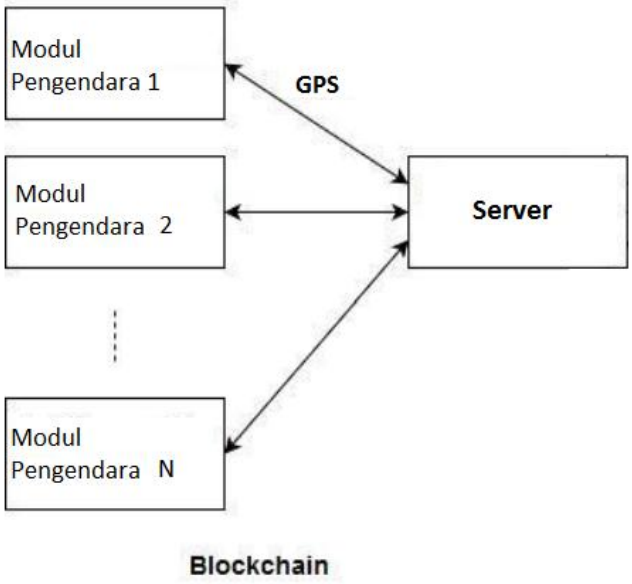
Gambar 3.7 Diagram skematik menggambarkan transmisi data dari modul pengendara ke *server* melalui GPS

3.2.2.3 Keamanan Data

Data yang diterima dari modul yang terpasang pada kendaraan yang digunakan *node* sangat sensitif dan memerlukan keamanan lebih agar data tidak dicuri maupun dirusak. Untuk mengatasi hal tersebut, teknik enkripsi data diintegrasikan. Penggunaan teknologi *blockchain* untuk menyimpan data, membuat data aman dari potensi gangguan dan pencurian. Mekanisme *Blockchain* terdiri dari blok yang saling berhubungan dan aman secara kriptografis. Setiap blok selanjutnya terdiri dari catatan yang terus bertambah dan digunakan ke dalam *Blockchain* untuk pemanfaatan. Setiap blok baru yang dibuat menggabungkan penghubung *hash* yang menghubungkan blok ke blok sebelumnya, bersama dengan *timestamp* dan muatan data. Sesuai mekanisme ini, rantai blok tidak dapat diubah dan juga kekuatan terhadap setiap perubahan informasi yang terkandung dalam *Blockchain*. Logika dasar mekanisme *Blockchain* terbuka untuk semua orang dan catatan di dalam blok tersebar.

Untuk implementasi *blockchain*, digunakan *Proof of Work (PoW)* sebagaimana yang digunakan dalam Bitcoin. *Nonce* digunakan sebagai penghitung. Pertama, dihitung nilai *hash* termasuk *Nonce* dan memeriksa beberapa digit pertama. Pada umumnya 5-6 digit. semua sama dengan 0, jika tidak maka nilai *nonce* bertambah, dan *hash* dihitung lagi sampai diperoleh yang diinginkan jumlah nol dalam *hash* (Nakamoto, 2008). Kode program ditulis dengan python. Penambangan blok secara lokal, menggunakan *server* untuk terhubung dengan *node* lain dan bersaing dengan mereka dalam proses penambangan, untuk secara bersamaan memeriksa *node* lain untuk blok baru atau

menyiarkan yang baru dihasilkan. Kode program berbeda untuk menambang blok pertama, yaitu, *Genesis*. Sistem ini memungkinkan *port* yang berbeda pada mesin yang sama bertindak seperti *node* yang berbeda serta mesin yang berbeda juga dapat bertindak sebagai *node* yang berbeda. Pada waktu tertentu, penambahan blok yang tidak valid tidak diakui oleh penambang. Demikian pula, pertukaran yang tidak valid tidak diakui oleh *node* untuk setiap bagian dari rantai yang diubah. Salah satu cara agar penyerang berhasil adalah dengan mengubah *hash* semua blok dari awal *Blockchain*, yang tidak mungkin dilakukan secara real time.



Gambar 3.8 Diagram skematik yang menggambarkan pengiriman data dari seluruh sistem dan penyimpanan dalam *Blockchain*

3.2.2.4 Implementasi *Blockchain*

Implementasi merupakan tahapan penerapan sistem yang dilakukan ketika sistem yang dibuat sudah sesuai dengan perancangan. Selain itu, proses penerapan sistem secara lengkap berada pada tahap ini, baik dari segi *software* maupun *hardware*. Implementasi sistem dimaksudkan untuk menerapkan perancangan

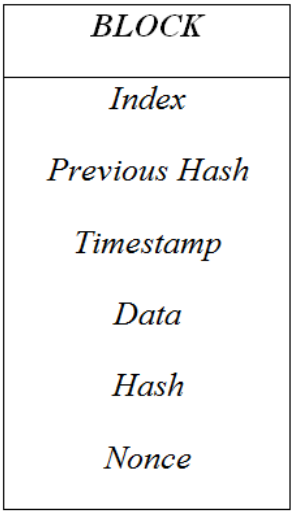
sistem yang telah dibuat, sehingga dapat menciptakan sistem yang sesuai dengan kebutuhan.

a. Kombinasi Data ke Bentuk *Block*

Setiap blok dalam rantai terdiri dari nilai *nonce* yang dihitung, indeks blok, *hash* blok, *timestamp* waktu dimana blok ditambang, *hash* blok sebelumnya, bidang data yang terdiri dari nama simpul yang ditambang, koordinat GPS, dan stempel waktu saat blok dibuat.

[{"nonce": "23052916", "index": "0", "hash":
"000000106cd1cc62c77e4701d061879c7474fa0bed22032825e2a4e590b806a7",
"timestamp": "20180425205734805868", "prev_hash": "", "data": "First
block."}, {"nonce": "4148028", "index": "1", "hash":
"000000837c699b9a075354bf0726fb65bbbc0222f47476163c2ad03424063ab8",
"timestamp": "20180425210118087917", "prev_hash":
"000000106cd1cc62c77e4701d061879c7474fa0bed22032825e2a4e590b806a7",
"data": "Block mined by node on port 5000mod_id , long: 28.63 , lat : 77.37 ,
0 , timestamp : 1524690078.0"}, {"nonce": "6985211", "index": "2", "hash":

Gambar 3.9 Data Mobilitas dalam Format JSON



Gambar 3.10 Struktur *Block*

Pada gambar 3.10 tentang isi blok pada *blockchain* dijelaskan bahwa :

1. *Index* = Letak blok di dalam rantai data *blockchain*
2. *Previous Hash* = nilai yang sama dengan nilai *hash* pada blok sebelumnya
3. *Timestamp* = atribut waktu saat blok tersebut dibuat
4. *Data* = data koordinat pada *node*, baik *latitude* dan *longtitude* di dalam blok tersebut.
5. *Hash* = data berupa nilai alphanumerik yang unik dari sebuah blok untuk enkripsi data.
6. *Nonce* = angka yang akan terus bertambah hingga mendapatkan nilai *hash* yang valid.

Data-data yang telah dijelaskan di atas, selanjutnya melalui proses kombinasi untuk membentuk suatu blok yang selanjutnya akan dilakukan proses *hashing*. Dimana fungsi *hash* menggunakan data sebagai inputan kemudian mengkonversinya berupa nilai *hash* yang unik seperti pada persamaan berikut ini yang di kemukakan oleh Han (2017).

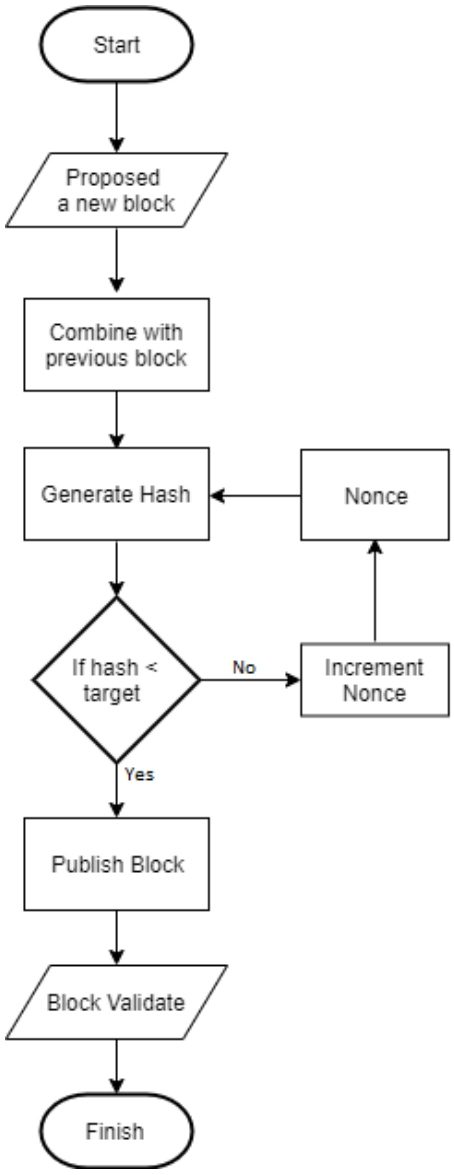
$$f\,(data) = hash$$

$$f\,(index + previous\ hash + timestamp + data + Nonce) = hash$$

Sehingga data yang ada di dalam blok berupa, *index, previous hash, timestamp, data, hash, nonce* akan digabung dan selanjutnya akan melalui proses *hashing*.

b. Proses *Hashing*

Proses *hashing* pada blok dilakukan untuk menemukan *hash value* setiap blok dengan menggunakan SHA256 *algorithm*. SHA256 *algorithm* mengubah data pesan masukan ke dalam *message digest* 256 bit dengan melakukan proses komputasi di dalamnya untuk menghasilkan nilai *hash* yang unik. Proses *hashing* pada blok dapat dilihat pada gambar 3.11 berikut ini.



Gambar 3.11 *Flowchart* proses *hashing*

Pada proses *hashing* dengan dilakukan pengubahan data pesan yang di inputkan menjadi bentuk kode *hash* alphanumerik. Dalam proses *hashing* tersebut dihasilkan *hash value* yang nantinya akan dilakukan proses validasi menggunakan *proof of work* untuk memperoleh *hash value* unik yang memenuhi target (difficult network) yang telah ditetapkan. Berikut adalah *pseudocode* algoritma SHA256. Semua variabel tidak ditandatangani 32 bit dan menggunakan modulo 2^{32} saat menghitung.

```
Tittle
Algoritma sha256

Initialize variables
(first 32 bits of the fractional parts of the square roots of
the first 8 primes 2..19):
h0 ← 0x6a09e667
h1 ← 0xbb67ae85
h2 ← 0x3c6ef372
h3 ← 0xa54ff53a
h4 ← 0x510e527f
h5 ← 0x9b05688c
h6 ← 0x1f83d9ab
h7 ← 0x5be0cd19

Initialize table of round constants
(first 32 bits of the fractional parts of the cube roots of
the first 64 primes 2..311):
k[0..63] ← 0x428a2f98, 0x71374491, 0xb5c0fbcf, 0xe9b5dba5,
0x3956c25b, 0x59f111f1, 0x923f82a4, 0xab1c5ed5, 0xd807aa98,
0x12835b01, 0x243185be, 0x550c7dc3, 0x72be5d74, 0x80deb1fe,
0x9bdc06a7, 0xc19bf174, 0xe49b69c1, 0xefbe4786, 0x0fc19dc6,
0x240ca1cc, 0x2de92c6f, 0x4a7484aa, 0x5cb0a9dc, 0x76f988da,
0x983e5152, 0xa831c66d, 0xb00327c8, 0xbf597fc7, 0xc6e00bf3,
0xd5a79147, 0x06ca6351, 0x14292967, 0x27b70a85, 0x2e1b2138,
0x4d2c6dfc, 0x53380d13, 0x650a7354, 0x766a0abb, 0x81c2c92e,
0x92722c85, 0xa2bfe8a1, 0xa81a664b, 0xc24b8b70, 0xc76c51a3,
0xd192e819, 0xd6990624, 0xf40e3585, 0x106aa070, 0x19a4c116,
0x1e376c08, 0x2748774c, 0x34b0bcb5, 0x391c0cb3, 0x4ed8aa4a,
0x5b9cca4f, 0x682e6ff3, 0x748f82ee, 0x78a5636f, 0x84c87814,
0x8cc70208, 0x90befffa, 0xa4506ceb, 0xbef9a3f7, 0xc67178f2

Algorithm
START
  Pre-processing:
  append the bit '1' to the message
  append k bits '0', where k is the minimum number >= 0 such
  that the resulting message
```

```
    length (in bits) is congruent to 448(mod 512)
append length of message (before pre-processing), in bits,
as 64-bit big-endian integer

Process the message in successive 512-bit chunks:
break message into 512-bit chunks

for each chunk
  break chunk into sixteen 32-bit big-endian words
  w[0..15]

  Extend the sixteen 32-bit words into sixty-four 32-bit
  words:
  for i from 16 to 63
       s0 ← (w[i-15] rightrotate 7) xor (w[i-15]
       rightrotate 18) xor(w[i-15] rightshift 3)
       s1 ← (w[i-2] rightrotate 17) xor (w[i-2]
       rightrotate 19) xor(w[i-2] rightshift 10)
       w[i] ← w[i-16] + s0 + w[i-7] + s1
  ENDFOR

  Initialize hash value for this chunk:
  a ← h0
  b ← h1
  c ← h2
  d ← h3
  e ← h4
  f ← h5
  g ← h6
  h ← h7

  Main loop:
  for i from 0 to 63
      s0 ← (a rightrotate 2) xor (a rightrotate 13) xor(a
      rightrotate 22)
      maj ← (a and b) xor (a and c) xor(b and c)
      t2 ← s0 + maj
      s1 ← (e rightrotate 6) xor (e rightrotate 11) xor(e
      rightrotate 25)
      ch ← (e and f) xor ((not e) and g)
      t1 ← h + s1 + ch + k[i] + w[i]
      h ← g
      g ← f
      f ← e
      e ← d + t1
      d ← c
      c ← b
      b ← a
      a ← t1 + t2
  ENDFOR

  Add this chunk's hash to result so far:
  h0 ← h0 + a
  h1 ← h1 + b
  h2 ← h2 + c
  h3 ← h3 + d
```

```
      h4 ← h4 + e
      h5 ← h5 + f
      h6 ← h6 + g
      h7 ← h7 + h
   ENDFOR

   Produce the final hash value (big-endian):
   digest ← hash ← h0 append h1 append h2 append h3 append h4
   append h5 append h6 append h7
END
```

Tahap awal disebut dengan *pre-processing*, yakni dengan mengubah inputan data menjadi bentuk biner, kemudian menambahkan angka 1, dilanjtukan dengan menambahkan 0 sampai data kelipatan 512, kurang 64 bit. Selanjutnya, ditambahkan 64 bit ke akhir, dimana 64 bit merupakan bilangan bulat *big-endian* yang mewakili panjang input asli dalam biner. Sehingga data inputan akan selalu habis dibagi 512.

Tahap kedua adalah inisialisasi nilai hash (h). Kemudian pada tahap ketiga dilakukan inisialisasi konstanta bulat (k). Setiap nilai (0-63) adalah 32 bit pertama dari bagian pecahan akar pangkat tiga dari 64 bilangan prima pertama (2 – 311).

Pada tahap keempat terdapat proses perulangan yang disebut dengan *chunk loop*. Pada tahap ini akan terjadi untuk setiap "*chunk*" atau potongan, yakni 512-bit data dari inputan. Pada setiap iterasi, akan dimutasikan nilai hash h0-h7, yang akan menjadi output akhir.

Tahap kelima adalah membuat *Message Schedule* (w). Menyalin data input dari langkah 1 ke *array* baru dimana setiap entri adalah kata 32-bit, menambahkan 48 kata lagi yang diinisialisasi ke nol, sehingga memiliki *array* w[0…63], mengubah indeks *zero-ed* di akhir *array* menggunakan aturan berikut:

```
For i from w[16…63]:
   s0 = (w[i-15] rightrotate 7) xor (w[i-15] rightrotate 18) xor
   (w[i-15] rightshift 3)
   s1 = (w[i- 2] rightrotate 17) xor (w[i- 2] rightrotate 19)
   xor (w[i- 2] rightshift 10)
   w[i] = w[i-16] + s0 + w[i-7] + s1
```

Tahap keenam pada proses *hashing* SHA256 adalah kompresi. Menginisialisasi variabel a, b, c, d, e, f, g, h dan mengatur masing-masing sehingga sama dengan nilai *current hash*. h0, h1, h2, h3, h4, h5, h6, h7. Menjalankan *Compression Loop*. *Compression Loop* akan mengubah nilai {a, … , h}. *Compression Loop* adalah sebagai berikut:

```
For i from 0 to 63:
    S1 = (e rightrotate 6) xor (e rightrotate 11) xor (e
    rightrotate 25)
    ch = (e and f) xor ((not e) and g)
    temp1 = h + S1 + ch + k[i] + w[i]
    S0 = (a rightrotate 2) xor (a rightrotate 13) xor (a
    rightrotate 22)
    maj = (a and b) xor (a and c) xor (b and c)
    temp2 := S0 + maj
    h = g
    g = f
    f = e
    e = d + temp1
    d = c
    c = b
    b = a
    a = temp1 + temp2
```

Pada iterasi pertama, semua penambahan dihitung modulo 2^{32}. Seluruh perhitungan itu dilakukan 63 kali lagi, memodifikasi variabel a-h secara keseluruhan.

Pada tahap ketujuh, yakni mengubah nilai akhir. Setelah *Compression Loop*, tetapi masih di dalam *chunk loop*, dilakukan modifikasi nilai *hash* dengan menambahkan variabel masing-masing ke dalamnya, a-h. Seperti sebelumnya, semua penambahan adalah modulo 2^{32}. Kemudian pada tahap akhir, yakni

menggabungkan hash terakhir.

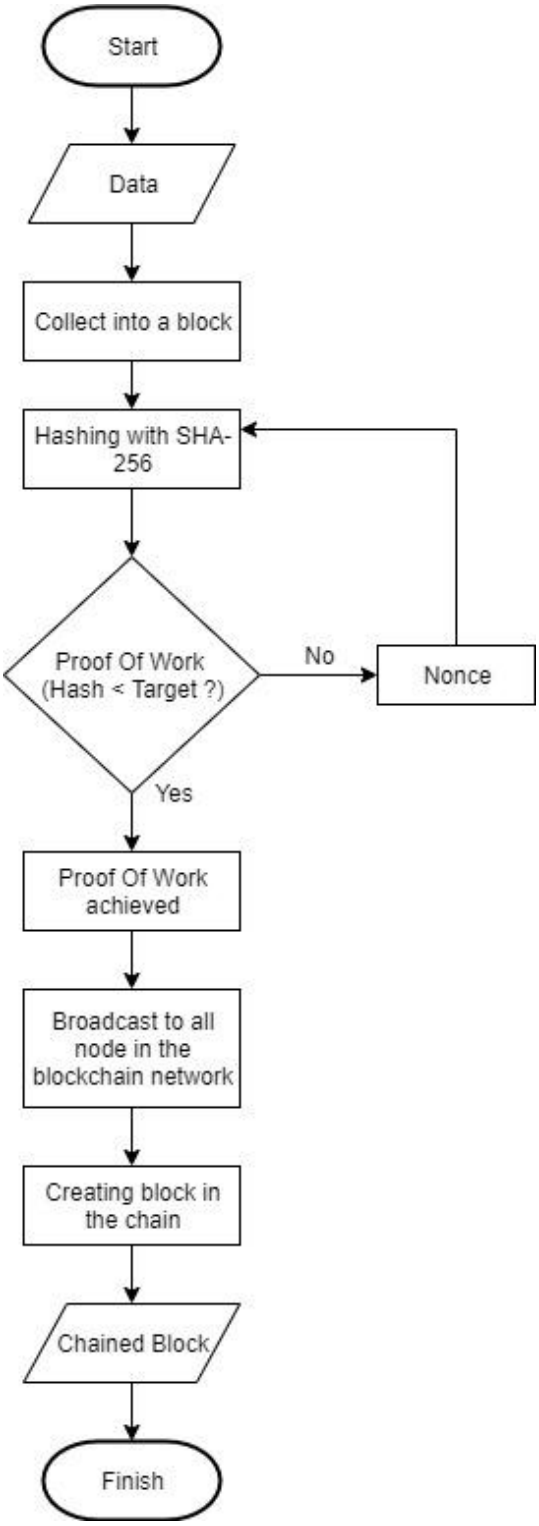
c. Pengukuran Block-Mining Time dan Validasi Blok dengan *Proof Of Work*

Dalam pengukuran block-mining time dan validasi blok terdapat konsep *Proof Of Work* (PoW) yang diperlukan untuk pengecekan nilai *hash* pada setiap blok yang dihasilkan melalui proses *hashing*. Proses validasi dengan *Proof Of Work* (PoW) akan melakukan validasi pada *hash value* yang terdapat pada setiap blok.

$$H\ (\ N\ ||\ Prev\_hash\ ||\ Tx\ ||\ Tx\ ||\ .\ .\ .\ Tx) < Target) \quad (3.1)$$

Dari persamaan (Han, 2017) di atas dapat dimaknai dimana (N) adalah *nonce* pada *block*, (*Prev_hash*) *hash value* dari *block* sebelumnya, (Tx) memiliki arti sebagai data mobilitas yang terdapat di dalam blok, dan (Target) merupakan nilai target *(difficult network)* yang sudah ditetapkan. Dalam konsep *Proof Of Work* di atas, *hash value* pada blok harus kurang dari target *value* yang sudah ditetapkan. Apabila *hash* value pada setiap blok telah mencapai target yang ditetapkan, maka *hash value* dari blok tersebut berhasil tervalidasi.

Tahapan proses validasi menggunakan *Proof Of Work* pada blok sebagaimana yang dikerjakan oleh Satoshi Nakamoto (Nakamoto, 2008) akan digambarkan seperti pada gambar 3.12 berikut.



Gambar 3.12 Proses *proof of work*

Langkah *Proof Of Work process* pada setiap blok di *blockchain* dijelaskan seperti berikut :

1. Data perpindahan lokasi yang akan di*publish* pada setiap *node* yang terdapat pada jaringan *blockchain* ditambahkan.
2. Dilakukan proses kombinasi data terlebih dahulu ke dalam bentuk blok.
3. Setelah data di kombinasi ke dalam bentuk blok, selanjutnya dilakukan proses *hashing* menggunakan algoritma SHA256.
4. Dilakukan proses validasi blok dengan *Proof Of Work* untuk mendapatkan nilai *hash* di setiap blok yang sesuai target *(difficult value)*.
5. Ketika blok sudah berhasil tervalidasi oleh *Proof Of Work*, selanjutnya blok siap dibagikan pada semua *node* yang terdapat pada *blockchain decentralized network*.
6. Setiap blok yang berhasil dibagikan ke setiap *node* pada *blockchain decentralized network* akan menghasilkan sebuah rantai blok *(chained block)* berisi data mobilitas seseorang.

Kemudian blok yang sudah tervalidasi akan dilakukan *broadcast* ke *blockchain decentralized network*, dimana nantinya akan menghasilkan sebuah rantai blok (*chained block*).

Proses validasi blok diperlukan untuk pengecekan *hash value* pada setiap blok yang dihasilkan melalui proses *hashing*. *Proof Of Work (PoW)* diimplementasikan dalam proses validasi blok. Berikut adalah *pseudocode* algoritma *Proof of Work (PoW)*.

```
Judul
Algoritma proof_of_work

Deklarasi
MINE_RATE ← 1000
difficulty ← nilai difficulty blok sebelumnya
nonce ← 0
timestampbefore = nilai timestamp blok sebelumnya
lat ← latitude dari sensor gps
lng ← lontitude dari sensor gps
p_hash ← nilai hash blok sebelumnya

Algoritma
START
    WHILE (loop ← True)
       nonce ← nonce + 1
       timestamp ← nilai timestamp saat ini
       tanggal ← tanggal saat ini
       waktu ← waktu saat ini

       IF difficulty < 1
             difficulty ← 1
       ENDIF

       IF (timestamp - timestampbefore) > MINE_RATE
             difficulty ← difficulty - 1
       ENDIF

       difficulty ← difficulty + 2
       IF difficulty >= 8
             difficulty ← 1
       ENDIF

       data = timestamp + tanggal + waktu + lat + lng +
       p_hash + nonce + difficulty
       hasil_hashing ← proses_hashing(data)
       binary_hashing ← convert_to_binary(hasil_hashing)
       get_binary ← get_char(binary_hashing, mulai 0 sampai
       difficulty)
       jumlah_nol ← 0 sebanyak difficulty

       IF get_binary == jumlah_nol
             loop ← False
       ENDIF
    ENDWHILE
END
```

Pada proses ini, mula-mula diperlukan sebuah *library* untuk memanggil *file javascript* agar dapat digunakan pada kode program *python*. Kemudian sebuah

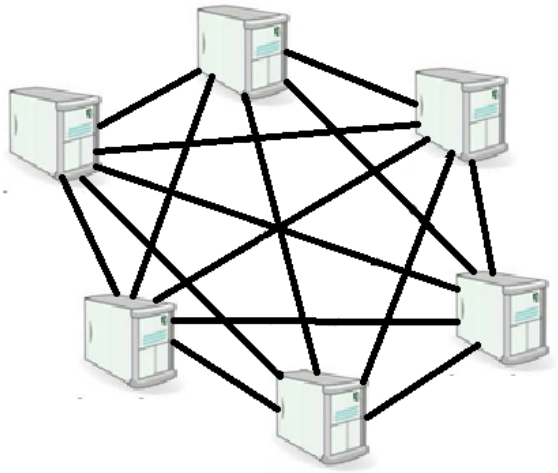
variabel juga diperlukan untuk memanggil fungsi SHA256 yang ditulis dalam kode program *javascript*.

Nilai *default difficulty* diambil dari nilai *difficulty* pada blok sebelumnya begitu juga *timestamp*. *Nonce* memiliki default nilai 0 dimana nilai *nonce* nantinya akan terus bertambah sebelum target *difficulty* tercapai. Penetapan nilai *difficulty* juga bergantung pada hasil pengurangan antara nilai *current timestamp* dan nilai *timestamp* pada blok sebelumnya.

Variabel **data** menyimpan gabungan data-data pembacaan sistem termasuk koordinat lintang dan bujur. Selanjutnya variabel **data** mengisi parameter fungsi *hashing SHA256*. Hasil proses *hashing* yang berupa *hexadecimal* selanjutnya diubah ke dalam bentuk *binary* untuk menentukan jumlah nol yang ditetapkan berdasarkan nilai *difficulty*. Proses ini akan terus berulang hingga jumlah nol tercapai.

d. *Publish Block* ke *Blockchain Network*

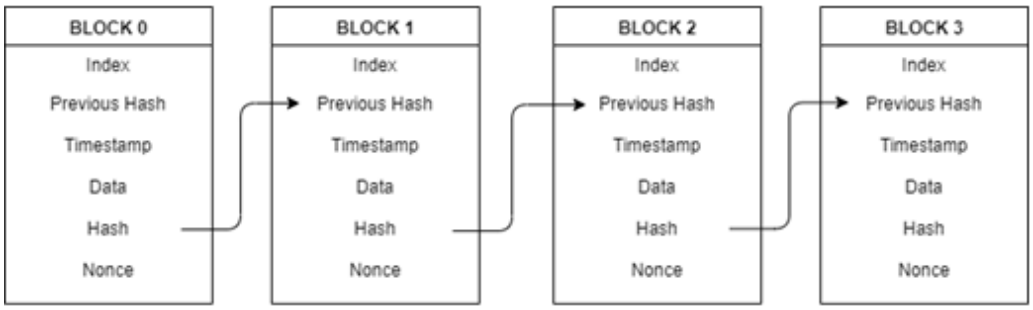
Pada langkah ini, setiap blok yang sudah berhasil melewati proses validasi, akan dibagikan ke setiap *node* yang terdapat pada jaringan desentralisasi *blockchain*, dimana setiap blok yang ditambahkan nantinya akan membentuk sebuah *node* yang saling berhubungan satu sama lain seperti pada gambar 3.13. Dalam proses *broadcast* ke jaringan desentralisasi *blockchain*, akan dilakukan konsep berbagi data model *peer to peer*. Sehingga setiap blok yang ditautkan, otomatis catatan data yang ada di setiap blok akan tersimpan di setiap *node* yang ada di jaringan desentralisasi *blockchain*.



Gambar 3.13 *Blockchain Network*

e. *Blockchain Ledger Record*

Pada jaringan desentralisasi *blockchain*, setiap blok yang tersimpan di setiap *node* dalam jaringan tersebut akan menyimpan catatan (*record*) mobilitas orang yang dilakukan oleh masing – masing user. Setiap blok yang ditambahkan akan menghasilkan rantai blok (*chained block*) yang saling berhubungan.rantai blok dapat direpresentasikan seperti pada gambar 3.14.



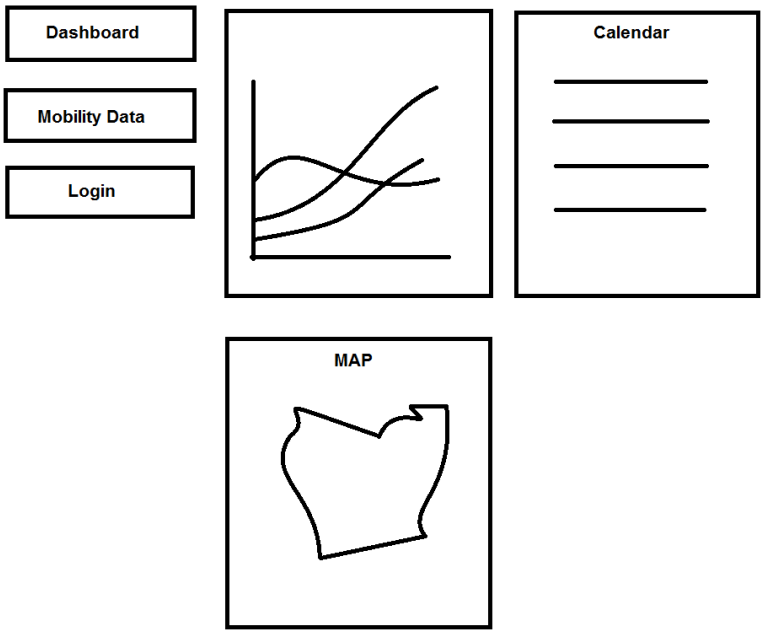
Gambar 3.14 Rantai Blok

Setiap blok yang terhubung dalam rantai blok pada jaringan desentralisasi *blockchain* akan membentuk sebuah *blockhain ledger record*, dimana semua *record* data mobilitas orang yang dihimpun oleh setiap *node* akan saling tertaut

dan tergabung menjadi sebuah *blockchain ledger record.*

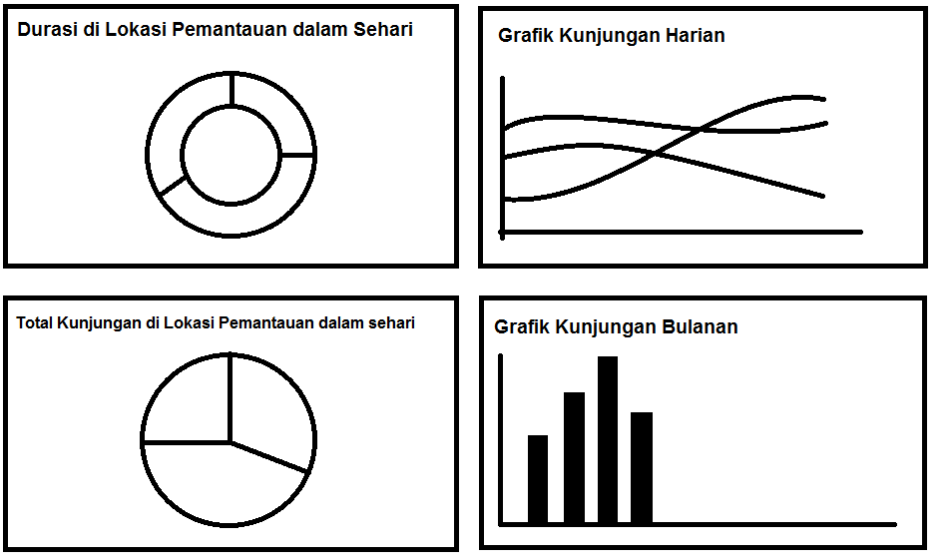
3.2.3 Output

Output data blockchain pada sistem ini divisualisasaikan berupa grafik, lokasi terakhir dan *tracking* pengendara



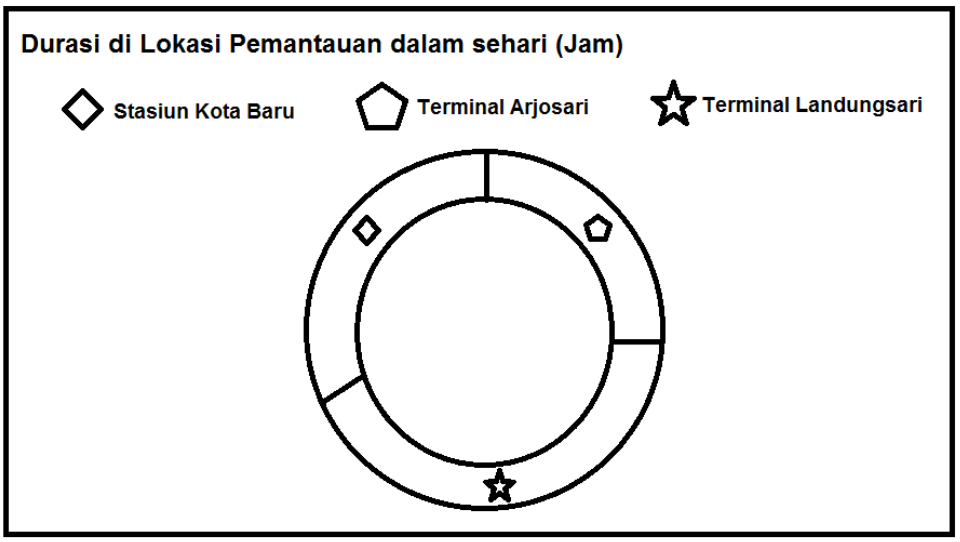
Gambar 3.15 Tampilan Desain *Dashboard*

Pada halaman *Dashboard,* disajikan beberapa informasi umum terkait hasil pelacakan sistem. Informasi yang dapat diperoleh dari halaman ini adalah data aktual pada satu hari sebelum hari tersebut yang meliputi: informasi jumlah pengendara yang terdaftar pada sistem, grafik total pengunjung pada lokasi pelacakan dalam sehari dan 1 bulan terakhir, informasi area pelacakan sistem serta sebuah kalender sebagai pengingat tanggal pada hari tersebut.

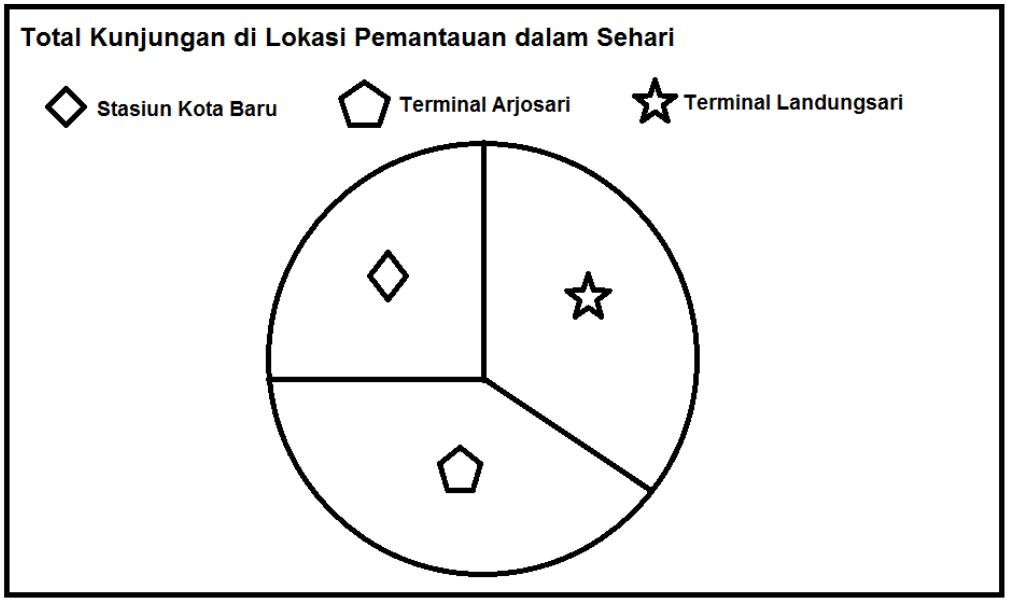


Gambar 3.16 Tampilan Desain Halaman Statistik Pemantauan Pengendara

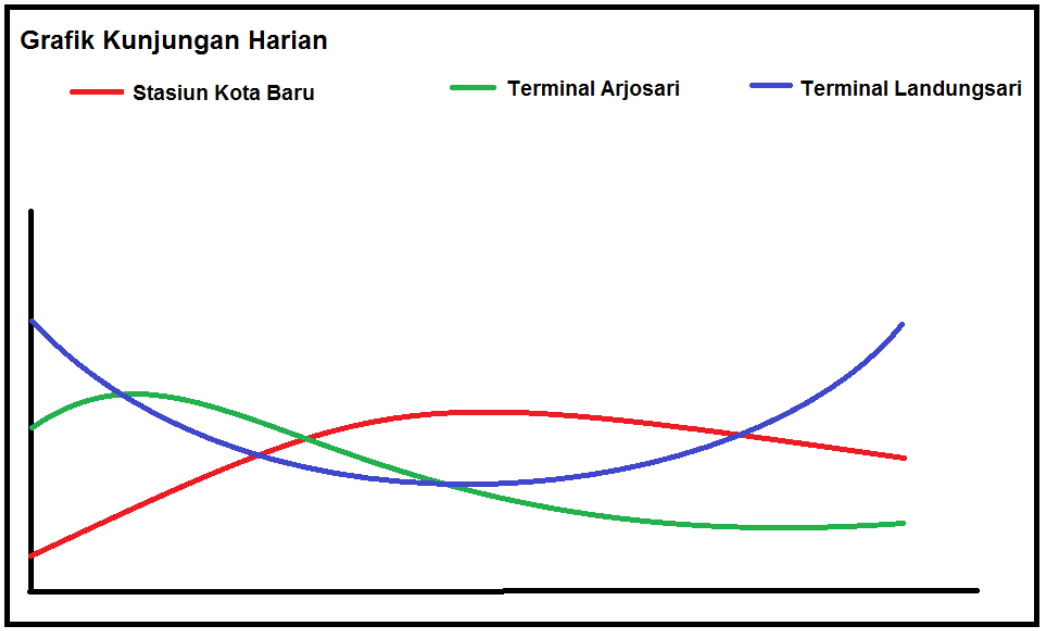
Pada halaman Statistik Pemantauan, disajikan beberapa *detail* informasi terkait hasil pemantauan sistem. Informasi yang dapat diperoleh dari halaman ini antara lain: informasi durasi atau lamanya seorang pengendara berada pada lokasi-lokasi pemantauan sistem pada hari tertentu, informasi total kunjungan seorang pengendara pada lokasi-lokasi pemantauan sistem pada hari tertentu, informasi grafik kunjungan harian seorang pengendara pada lokasi-lokasi pemantauan sistem pada 7 hari terakhir terhitung mundur dari masukan tanggal serta grafik kunjungan bulanan seorang pengendara pada lokasi-lokasi pemantauan sistem pada 7 bulan terakhir terhitung mundur dari masukan tanggal.



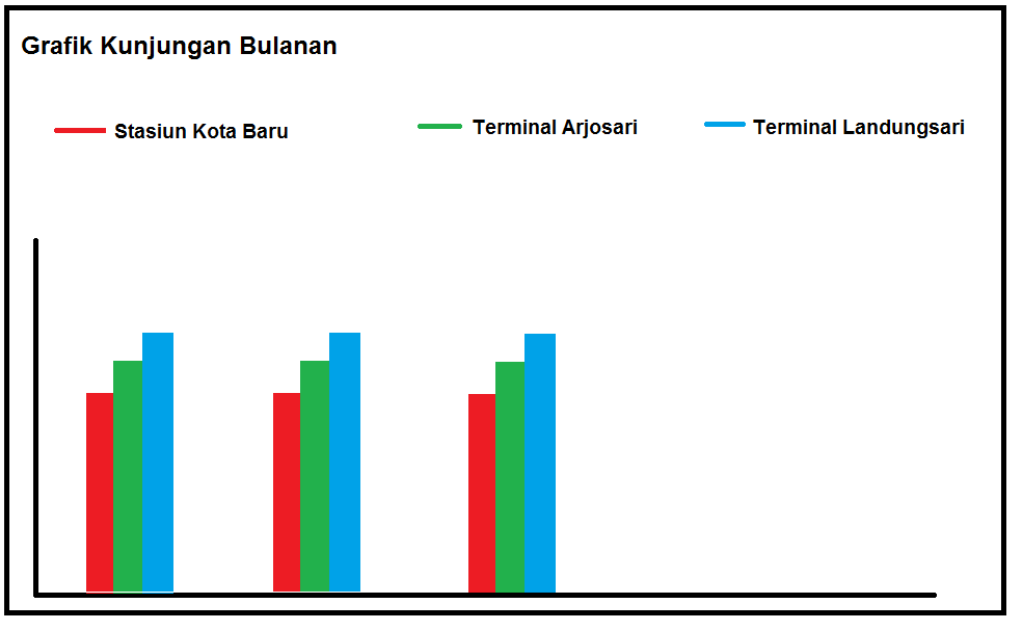
Gambar 3.17 Desain *Doughnut Chart* Durasi Seorang Pengendara di Lokasi Pemantauan dalam Sehari



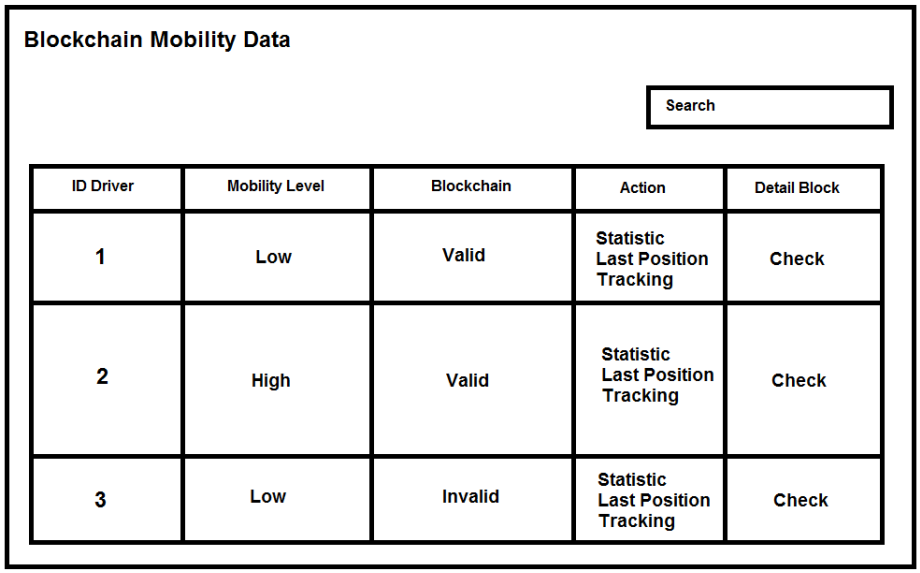
Gambar 3.18 Desain *Pie Chart* Total Kunjungan Seorang Pengendara di Lokasi Pemantauan dalam Sehari



Gambar 3.19 Desain *Line Chart* Grafik Kunjungan Harian Seorang Pengendara di Lokasi Pemantauan dalam 7 Hari Terakhir



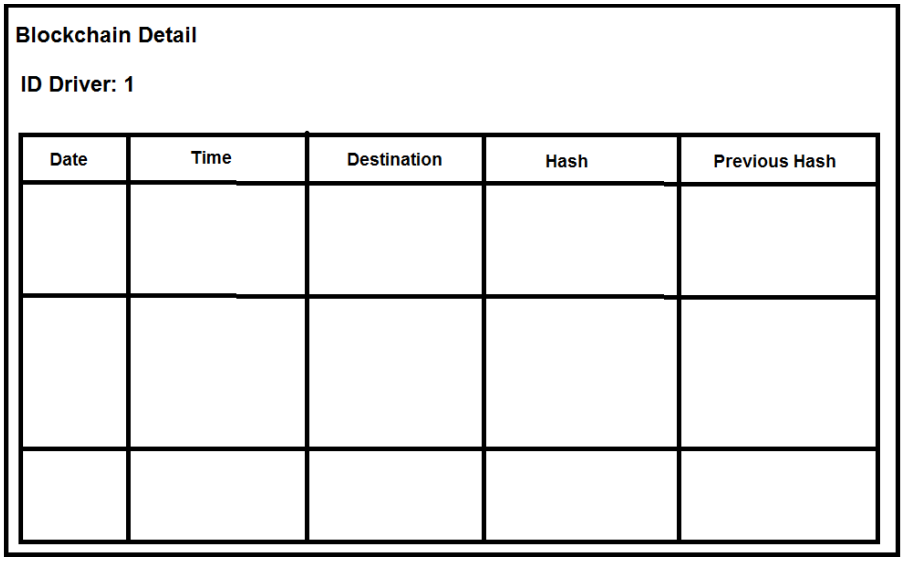
Gambar 3.20 Desain *Bar Chart* Grafik Kunjungan Bulanan Seorang Pengendara di Lokasi Pemantauan dalam 7 Bulan Terakhir



Gambar 3.21 Tampilan Desain Halaman *Blockchain* Data Mobilitas

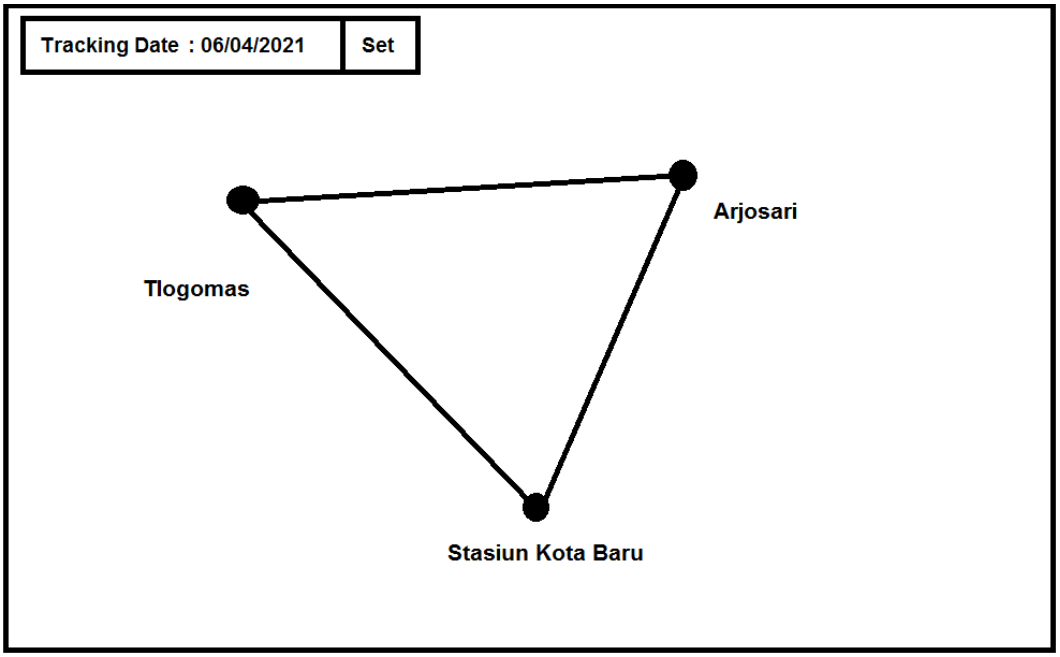
Pada halaman *Blockchain* Data Mobilitas, disajikan sebuah tabel terkait mobilitas pengendara-pengendara yang terdaftar pada sistem. Informasi yang dapat diperoleh dari halaman ini antara lain: ID pengendara, tingkat mobilitas pengendara, status *blockchain,* tindakan lanjutan serta *detail block.* Status *Invalid* pada blockchain terjadi akibat perubahan data pada bagian *block* yang tidak dikehendaki oleh node pada jaringan blockchain. Untuk mengetahui *detail block* dapat dilakukan dengan menekan *Check.* Pada kolom *Action,* terdapat tiga fitur dari *prototype* sistem yakni:

a. *Statistic,* digunakan untuk menampilkan halaman Statistik Pemantauan pengendara sebagaimana yang ditunjukkan pada gambar 3.16
b. *Last Position,* digunakan untuk menunjukkan lokasi terakhir pengendara yang terlacak oleh sistem
c. *Tracking,* digunakan untuk menampilkan urutan perjalanan seorang pengendara pada lokasi-lokasi pelacakan sistem



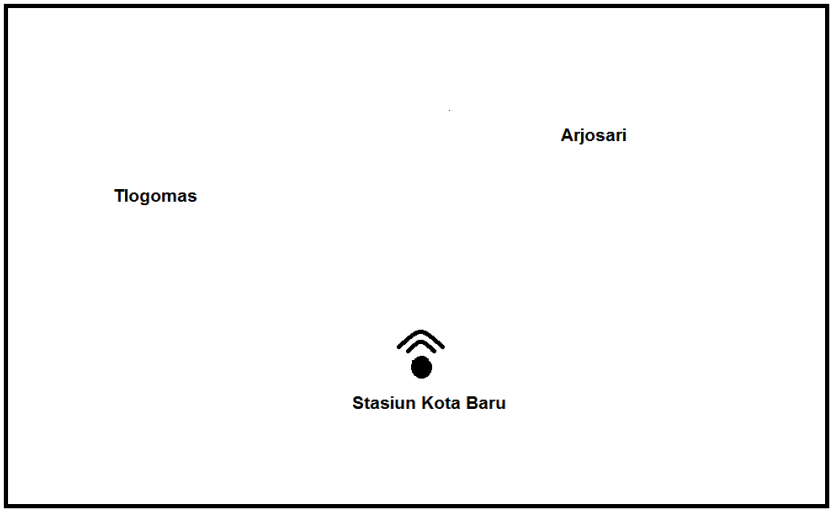
Gambar 3.22 Tampilan Desain Halaman *Blockchain detail*

Pada halaman *Blockchain Detail*, disajikan sebuah tabel terkait isi dari *block* penyusun *blockchain* sesuai *blockchain* yang dipilih. *Detail block* meliputi: waktu tiba di lokasi pelacakan, kunjungan pada lokasi pelacakan, nilai *hash* serta nilai *previous hash.* Tanda warna merah menunjukkan sebuah *block* yang tidak *valid*, hal ini disebabkan oleh perubahan isi data di dalam *block* yang tidak dikehendaki oleh *node* dalam jaringan *blockchain*. Perubahan pada isi *block* menyebabkan perubahan pada nilai *hash* sehingga nilai *previous hash* pada block setelahnya menjadi tidak sesuai. Peretas mungkin saja bisa merubah semua isi *block* dalam *blockchain* tersebut tetapi hal ini cukup sulit dilakukan karena membutuhkan waktu yang lama serta daya komputasi yang besar untuk bisa menyesuaikan semua isi *block* yang saling terhubung.



Gambar 3.23 Tampilan Desain Halaman *Tracking*

Pada halaman *Tracking*, disajikan sebuah peta beserta penanda dan juga garis untuk menampilkan urutan perjalanan seorang pengendara mulai dari awal perjalanan hingga akhir tujuan pengendara pada lokasi-lokasi pelacakan sistem pada hari tertentu.



Gambar 3.24 Tampilan Desain Halaman Lokasi Terakhir

Pada halaman *Last Position*, disajikan sebuah peta beserta penanda untuk menunjukkan lokasi terakhir pengendara yang terlacak oleh sistem pada hari tertentu.

3.3 Kemungkinan Peretasan

Pada bagian ini, penulis mencoba menggambarkan kinerja prototipe awal modul pengendara atau pengemudi, bersama dengan data yang disimpan di *blockchain*. Kemudian, penulis secara kuantitatif mengevaluasi dan mengukur serta menghitung waktu yang diperlukan dalam proses *Proof of Work* (*PoW*).

Fungsi utama penerapan *blockchain* dalam sistem ini adalah untuk mengamankan dan memberikan kerumitan pada data mobilitas seseorang dari potensi serangan yang dapat mengakibatkan perubahan data sehingga data mobilitas menjadi tidak valid. Sebagai perkiraan dasar mengenai kemungkinan serangan terhadap keamanan data berbasis *blockchain* yang diterapkan, digunakan rumus distribusi Poisson. Penyerang akan mencoba menciptakan rantai lain yang lebih cepat dibandingkan rantai yang sudah *valid* guna mengganti data mobilitas yang disimpan. Penyerang akan kesulitan karena memerlukan daya komputasi lebih dari 51% dari penambang *blockchain* (Pabla dkk., 2019). Cara yang harus dilakukan penyerang agar berhasil adalah dengan mengubah nilai hash dari semua *block* dari awal, dimana hal tersebut sulit dilakukan secara *real time*. Adapun rumus distribusi Poisson sebagaimana yang telah dibahas oleh Feller. (1957) adalah sebagai berikut:

$$\lambda = np \tag{3.3}$$

$$P(X=\chi) = \frac{e^{-\lambda} \quad \lambda^{\chi}}{\chi!} \tag{3.4}$$

Dimana:

1. λ adalah rata-rata kejadian sukses setelah sekian kali percobaan.
2. n adalah jumlah *block*.
3. *P* atau *p* adalah peluang.
4. X atau χ adalah jumlah *block* yang diserang atau jumlah *block* yang tidak *valid.*
5. e adalah logaritma natural yang nilainya 2,718281828459

Contoh kasus:

Seorang penyerang atau peretas diketahui bahwa rata-rata mampu merusak 1 *block* dari 8 *block* data mobilitas seorang pengendara yang dihasilkan. Dapat diperkirakan peluang bahwa:

a. Dari 10 *block* data mobilitas seorang pengendara yang dihasilkan terdapat 1 *block* rusak atau tidak valid
b. Dari 50 *block* data mobilitas seorang pengendara yang dihasilkan terdapat 1 *block* rusak atau tidak valid
c. Dari 100 *block* data mobilitas seorang pengendara yang dihasilkan terdapat 1 *block* rusak atau tidak valid

Perhitungan:

a. Diketahui: $p = \frac{1}{8} = 0{,}125$ dan $n = 10$ sehingga

$$\lambda = np$$

$$= (10) \,.\, (0{,}125)$$

$$= 1{,}25$$

Dengan demikian:

$$P(X = 1) = \frac{e^{-1{,}25} \; x \;\; 1{,}25^{1}}{1!}$$

$$= \frac{2{,}718281828459^{-1{,}25} \; x \;\; 1{,}25^{1}}{1} = 0{,}35813099607$$

b. Diketahui: $p = \frac{1}{8} = 0{,}125$ dan $n = 50$ sehingga

$$\lambda = np$$

$$= (50) \,.\, (0{,}125)$$

$$= 6{,}25$$

Dengan demikian:

$$P(X = 1) = \frac{e^{-6{,}25} \; x \;\; 6{,}25^{1}}{1!}$$

$$= \frac{2{,}718281828459^{-6{,}25} \; x \;\; 6{,}25^{1}}{1} = 0{,}01206533835$$

c. Diketahui: $p = \frac{1}{8} = 0{,}125$ dan $n = 100$ sehingga

$$\lambda = np$$

$$= (100) \,.\, (0{,}125)$$

$$= 12{,}5$$

Dengan demikian:

$$P(X = 1) = \frac{e^{-12{,}5} \; x \;\; 12{,}5^{1}}{1!}$$

$$= \frac{2{,}718281828459^{-12{,}5} \; x \;\; 12{,}5^{1}}{1} = 0{,}00004658316$$

Berdasarkan perkiraan dasar mengenai kemungkinan serangan terhadap keamanan data berbasis *blockchain* yang diterapkan, dapat diketahui bahwa peluang penyerang berhasil mengubah *block* data mobilitas pengendara akan semakin menurun seiring dengan bertambahnya jumlah *block*.

Pada pengujian terhadap kinerja *Proof of Work* (*PoW*) dimaksudkan guna mengetahui waktu yang diperlukan proses tersebut dalam memperoleh nilai *hash* yang memenuhi target yang ditentukan.

BAB IV

HASIL DAN PEMBAHASAN

Pada bab ini dijelaskan tentang hasil uji coba dan pembahasan tentang Sistem Keamanan Mobilitas Orang di Kota Malang Berbasis *Blockchain.* Dimana meliputi: implementasi, data pengujian dan hasil pengujian.

4.1 Implementasi

Implementasi merupakan tahapan penerapan sistem yang dilakukan ketika sistem yang dibuat sudah sesuai dengan perancangan. Selain itu, tahap ini juga sebagai proses implementasi sistem secara keseluruhan, baik dari segi *software* maupun *hardware*. Pengaplikasian sistem dimaksudkan untuk mewujudkan perancangan sistem yang telah dibuat, sehingga bisa menciptakan sistem sesuai dengan kebutuhan.

4.1.1 Implementasi Sistem

Pada bagian ini akan diberikan gambaran mengenai *output* atau visualisasi dari sistem keamanan mobilitas orang beserta dengan fitur-fitur yang terdapat pada sistem ini. Pembuatan sistem ini menggunakan bahasa pemrograman PHP, Javascript, CSS, dan Python, menggunakan *text editor Sublime Text 3* dan menggunakan *database MySQL versi 7.2.0*. *Database* digunakan untuk menyimpan data pada sistem keamanan mobilitas orang. Pada implementasi sistem akan dibahas tentang implementasi metode SHA256 dan Proof of Work (*PoW*) dalam sistem yang dibuat.

Tahap awal pada algoritma SHA256 disebut dengan *pre-processing*, yakni dengan mengubah inputan data menjadi bentuk biner, kemudian menambahkan angka 1, dilanjtukan dengan menambahkan 0 sampai data kelipatan 512, kurang 64 bit. Selanjutnya, ditambahkan 64 bit ke akhir, dimana 64 bit merupakan bilangan bulat *big-endian* yang mewakili panjang input asli dalam biner. Sehingga data inputan akan selalu habis dibagi 512.

Tahap kedua adalah inisialisasi nilai hash (h). Kemudian pada tahap ketiga dilakukan inisialisasi konstanta bulat (k). Setiap nilai (0-63) adalah 32 bit pertama dari bagian pecahan akar pangkat tiga dari 64 bilangan prima pertama (2 – 311).

Pada tahap keempat terdapat proses perulangan yang disebut dengan *chunk loop*. Pada tahap ini akan terjadi untuk setiap "*chunk*" atau potongan, yakni 512-bit data dari inputan. Pada setiap iterasi, akan dimutasikan nilai hash h0-h7, yang akan menjadi output akhir.

Tahap kelima adalah membuat *Message Schedule* (w). Menyalin data input dari langkah 1 ke *array* baru dimana setiap entri adalah kata 32-bit, menambahkan 48 kata lagi yang diinisialisasi ke nol, sehingga memiliki *array* w[0…63], mengubah indeks *zero-ed* di akhir *array* menggunakan aturan berikut:

```
For i from w[16…63]:
  s0 = (w[i-15] rightrotate 7) xor (w[i-15] rightrotate 18) xor
  (w[i-15] rightshift 3)
  s1 = (w[i- 2] rightrotate 17) xor (w[i- 2] rightrotate 19)
  xor (w[i- 2] rightshift 10)
  w[i] = w[i-16] + s0 + w[i-7] + s1
```

Tahap keenam pada proses *hashing* SHA256 adalah kompresi. Menginisialisasi variabel a, b, c, d, e, f, g, h dan mengatur masing-masing sehingga sama dengan nilai *current hash*. h0, h1, h2, h3, h4, h5, h6, h7.

Menjalankan *Compression Loop*. *Compression Loop* akan mengubah nilai {a, … , h}. *Compression Loop* adalah sebagai berikut:

```
For i from 0 to 63:
    S1 = (e rightrotate 6) xor (e rightrotate 11) xor (e
    rightrotate 25)
    ch = (e and f) xor ((not e) and g)
    temp1 = h + S1 + ch + k[i] + w[i]
    S0 = (a rightrotate 2) xor (a rightrotate 13) xor (a
    rightrotate 22)
    maj = (a and b) xor (a and c) xor (b and c)
    temp2 := S0 + maj
    h = g
    g = f
    f = e
    e = d + temp1
    d = c
    c = b
    b = a
    a = temp1 + temp2
```

Pada iterasi pertama, semua penambahan dihitung modulo 2^{32}. Seluruh perhitungan itu dilakukan 63 kali lagi, memodifikasi variabel a-h secara keseluruhan.

Pada tahap ketujuh, yakni mengubah nilai akhir. Setelah *Compression Loop*, tetapi masih di dalam *chunk loop*, dilakukan modifikasi nilai *hash* dengan menambahkan variabel masing-masing ke dalamnya, a-h. Seperti sebelumnya, semua penambahan adalah modulo 2^{32}. Kemudian pada tahap akhir, yakni menggabungkan hash terakhir.

Proof Of Work (PoW) diperlukan untuk mengukur kecepatan *hash value* pada setiap blok yang dihasilkan melalui proses *hashing* dalam mencapai target *difficulty*. *Proof Of Work (PoW)* diimplementasikan dalam proses validasi blok dan mengukur kecepatan *block-mining time* untuk mencapai target *difficulty*.

Implementasi *PoW* pada sistem yang dibuat, mula-mula diperlukan sebuah *library* untuk memanggil *file javascript* agar dapat digunakan pada kode program *python*. Kemudian sebuah variabel juga diperlukan untuk memanggil fungsi SHA256 yang ditulis dalam kode program *javascript*.

Nilai *default difficulty* diambil dari nilai *difficulty* pada blok sebelumnya begitu juga *timestamp*. *Nonce* memiliki default nilai 0 dimana nilai *nonce* nantinya akan terus bertambah sebelum target *difficulty* tercapai. Penetapan nilai *difficulty* juga bergantung pada hasil pengurangan antara nilai *current timestamp* dan nilai *timestamp* pada blok sebelumnya.

Variabel **data** menyimpan gabungan data-data pembacaan sistem termasuk koordinat lintang dan bujur. Selanjutnya variabel **data** mengisi parameter fungsi *hashing* SHA256. Hasil proses *hashing* yang berupa *hexadecimal* selanjutnya diubah ke dalam bentuk *binary* untuk menentukan jumlah nol yang ditetapkan berdasarkan nilai *difficulty*. Proses ini akan terus berulang hingga jumlah nol tercapai.

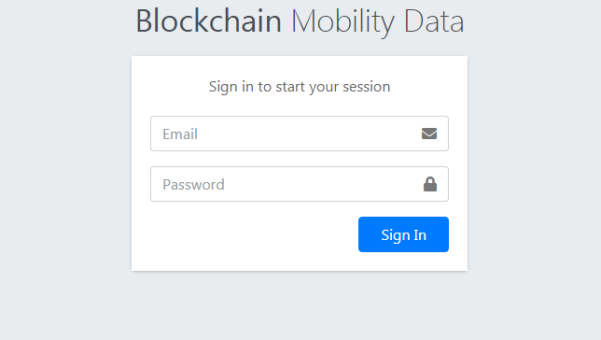
4.1.2 Implementasi *Interface*

Sistem yang dibuat dalam penelitian ini yaitu berbasis IoT dengan visualisasi melalui web menggunakan bahasa pemrograman PHP. Kemudian untuk tampilan antarmuka sistem yakni menggunakan *framework CSS Bootstrap*. Implementasi antarmuka dalam sistem ini adalah sebagai berikut:

1. Halaman Login

Sistem yang dibuat terdiri dari halaman admin dan halaman *user*, dalam hal ini *user* adalah pengendara sebagai *node* dalam jaringan *blockchain*. Untuk

dapat masuk ke halaman admin dan halaman *user*, terdapat proses pengecekkan akun yakni harus melewati halaman login dengan mengisi *email* dan *password* yang telah ada di *database*. Untuk akun *user* hanya dapat ditambahkan oleh admin.

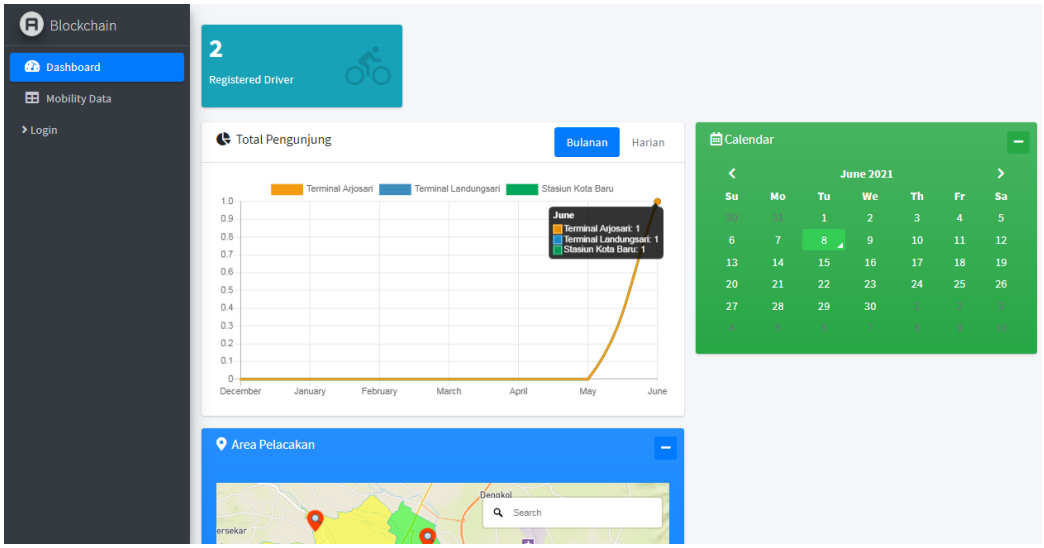


Gambar 4.1 Halaman Login Sistem Keamanan Mobilitas Orang di Kota Malang

2. Halaman *Home*

Pada sistem ini terdapat 2 halaman *home* yang berbeda, yakni halaman *home* untuk pengunjung yang berarti mereka yang bukan *user* ataupun admin dan halaman *home* yang lain adalah khusus untuk admin dan *user*. Tidak ada perbedaan secara visual antara halaman *home* pengunjung dan user atau admin, yang membedakan hanya pada menu *Mobility Data*, dimana pengunjung tidak dapat mengakses menu tersebut. Untuk mengakses halaman tersebut diperlukan login terlebih dahulu, yakni sebagai *user* atau admin. Pada halaman ini, disajikan beberapa informasi umum terkait hasil pelacakan sistem. Informasi yang dapat diperoleh dari halaman ini adalah data aktual pada satu hari sebelum hari tersebut yang meliputi: informasi jumlah pengendara yang terdaftar pada sistem, grafik total pengunjung pada lokasi pelacakan dalam sehari dan 1 bulan terakhir,

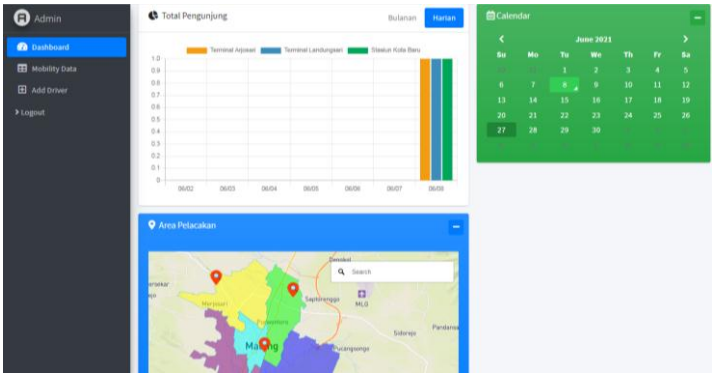
informasi area pelacakan sistem serta sebuah kalender sebagai pengingat tanggal pada hari tersebut.



Gambar 4.2 Halaman *Home* Sistem Keamanan Mobilitas Orang di Kota Malang

3. Halaman *Dashboard* admin

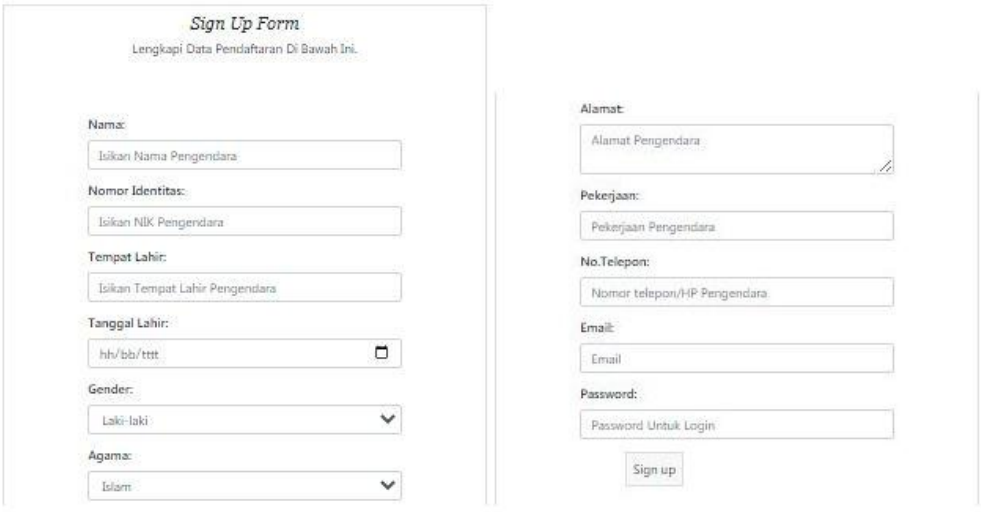
Pada halaman *dashboard* admin, terdapat satu menu khusus. *Add Driver* adalah menu yang hanya ada pada halaman *dashboard* admin. Menu ini berfungsi untuk menambahkan pengendara baru sekaligus sebagai *node* baru pada jaringan *blockchain*. Admin akan mengisi *form* registrasi pada halaman *Driver Registration.*



Gambar 4.3 Halaman *Dashboard* admin Sistem Keamanan Mobilitas Orang di Kota Malang

4. Halaman *Driver Registration*

Halaman ini merupakan halaman khusus yang hanya bisa diakses oleh admin. Pada halaman ini admin bisa menambahkan data pengendara baru atau *node* baru, yakni dengan mengisi *form* pendaftaran. Data yang harus diisi meliputi: nama, NIK, tempat lahir, *gender*, agama, alamat, pekerjaan, nomor telepon, email pengendara dan *password* untuk *login*. Setelah pendaftaran selesai, selanjutnya akan dipasang perangkat elektronik pada kendaraan milik pengendara tersebut, dimana perangkat tersebut telah berisi kode program untuk mengirimkan data lokasi pengendara. Berikut adalah gambar *form* registrasi pengendara:

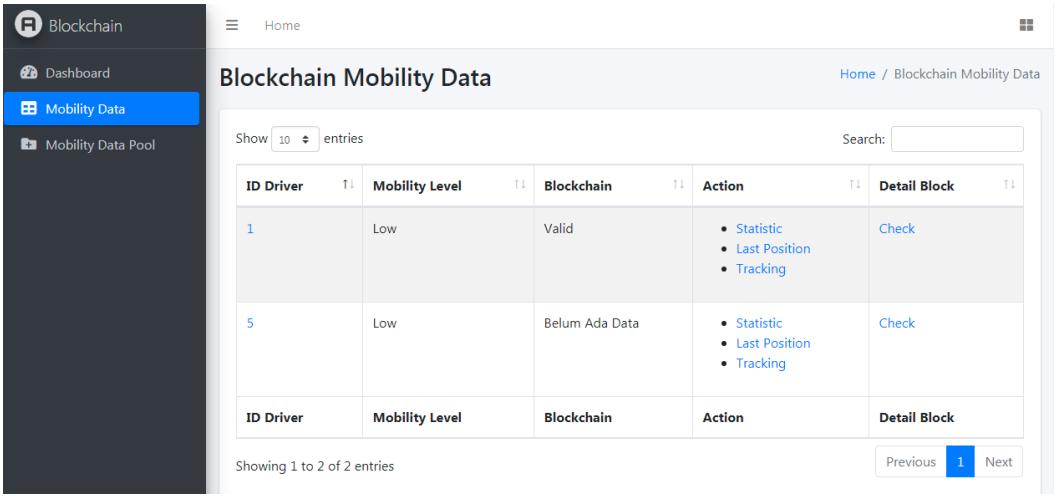


Gambar 4.4 Halaman *Driver Registration* Sistem Keamanan Mobilitas Orang di Kota Malang

5. Halaman *Blockchain Mobility Data*

Halaman ini hanya bisa diakses oleh admin dan pengendara yang terdaftar pada sistem atau *node* pada jaringan *blockchain*. Artinya pengunjung tidak dapat melihat halaman ini. Pengunjung hanya bisa melihat halaman utama atau *home*. Pada halaman ini disajikan tabel data mobilitas pengendara yang terdaftar pada

sistem. Adapun data tersebut meliputi: ID pengendara, level mobilitas pengendara, status *blockchain* serta 3 fitur untuk melihat statistik pemantauan pengendara, melihat posisi terakhir pengendara saat terekam sistem dan fitur *tracking*. Pada halaman ini *user* dan admin juga dapat melihat *detail block*, yakni blok-blok penyusun *blockchain* setiap pngendara.

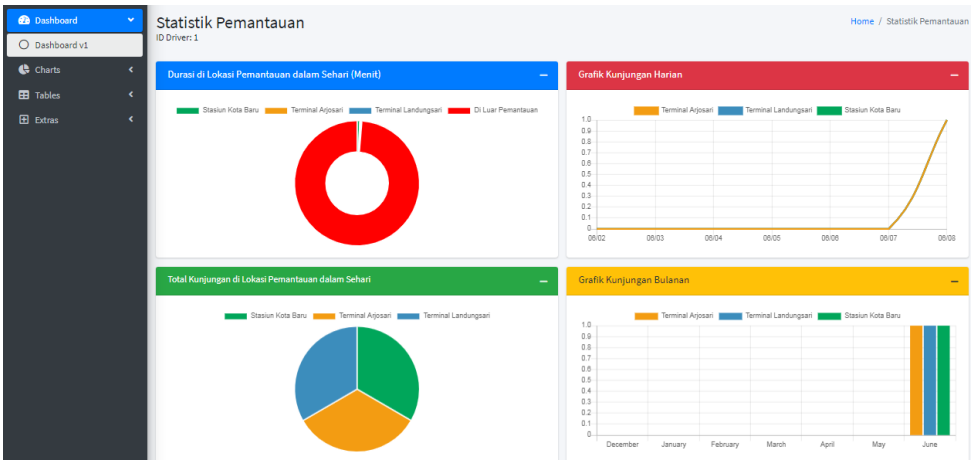


Gambar 4.5 Halaman *Blockchain Mobility Data* Sistem Keamanan Mobilitas Orang di Kota Malang

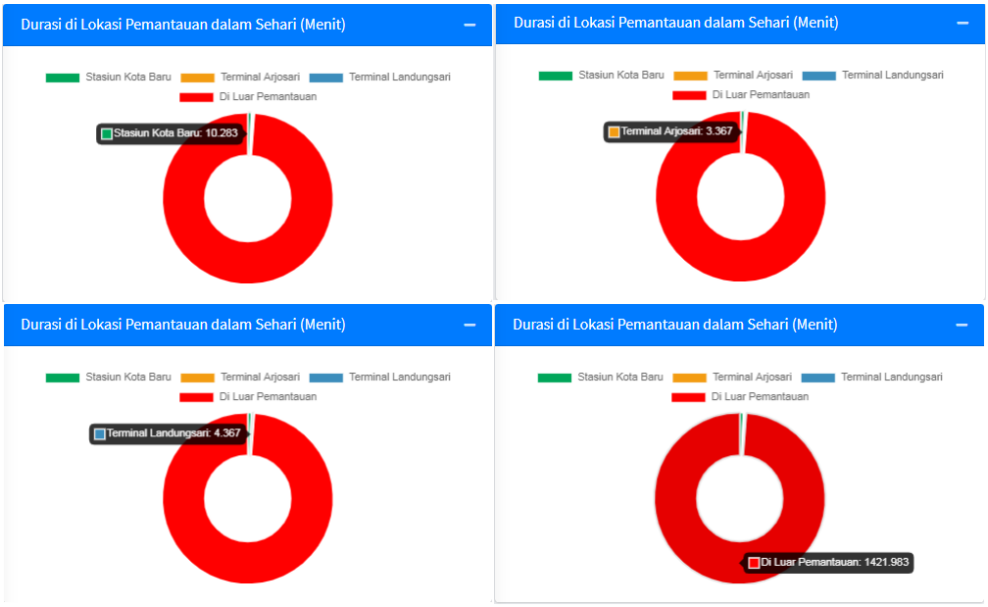
6. Halaman *Statistic*

Pada halaman ini disajikan beberapa data mengenai mobilitas seorang pengendara dalam bentuk diagram. Informasi yang dapat diperoleh dari halaman ini adalah durasi waktu pengendara saat berada di lokasi-lokasi pemantauan sistem dalam sehari, grafik kunjungan harian yakni menunjukkan berapa kali pengendara melewati lokasi-lokasi pemantauan sistem dalam satu minggu terakhir, informasi total kunjungan harian dan juga grafik kunjungan bulanan yakni rekap jumlah kunjungan pengendara pada lokasi-lokasi pemantauan sistem dalam satu bulan terakhir. Halaman ini hanya dapat diakses oleh admin dan juga

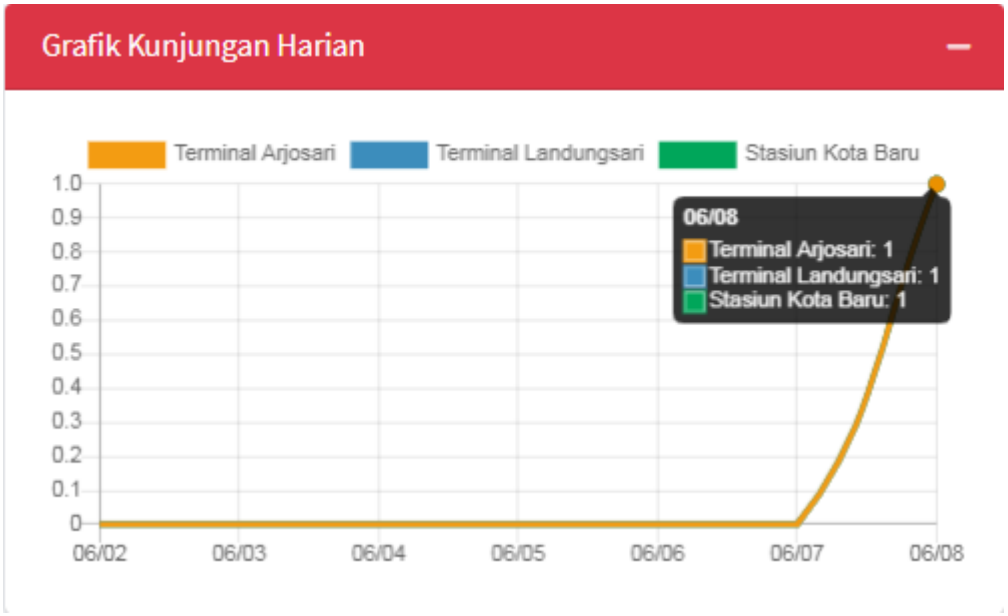
user atau pengendara yang terdaftar pada sistem sebagai bagian dari *node* pada jaringan *blockchain.*



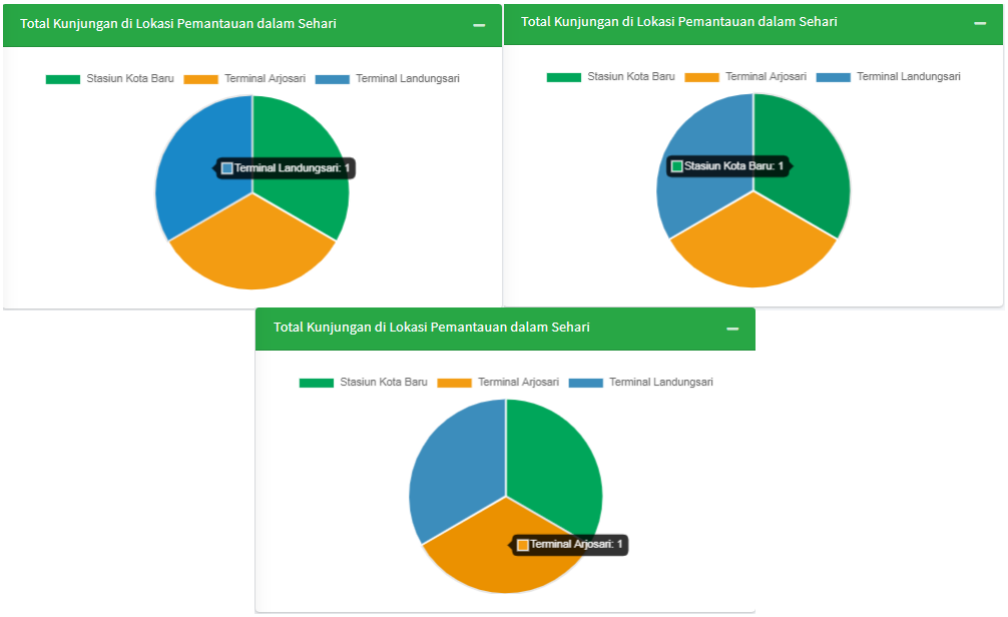
Gambar 4.6 Halaman *Statistic* Sistem Keamanan Mobilitas Orang di Kota Malang



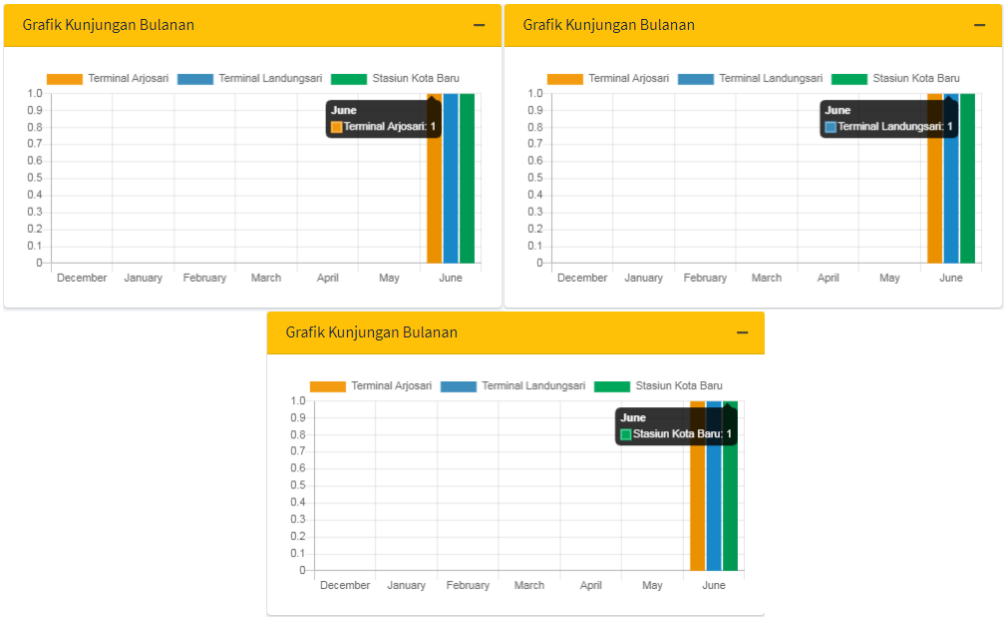
Gambar 4.7 Detail Grafik Durasi Pengendara pada Lokasi Pemantauan dalam Sehari



Gambar 4.8 Detail Grafik Kunjungan Harian Pengendara pada Lokasi Pemantauan



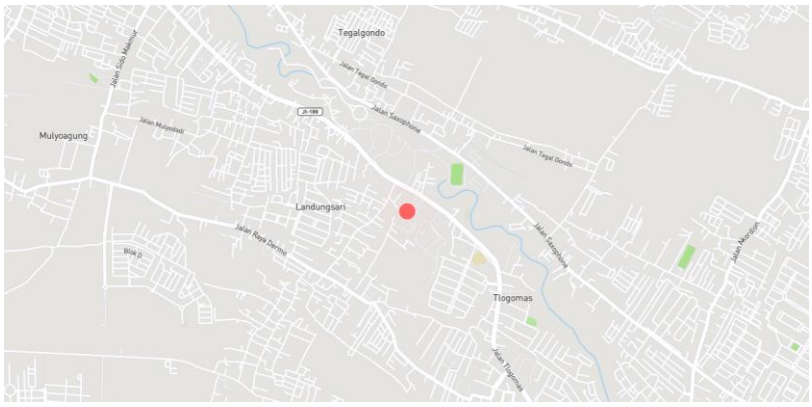
Gambar 4.9 Detail Grafik Total Kunjungan Pengendara ke Lokasi Pemantauan dalam Sehari



Gambar 4.10 Detail Grafik Kunjungan Bulanan Pengendara ke Lokasi Pemantauan

7. Halaman *Last Position*

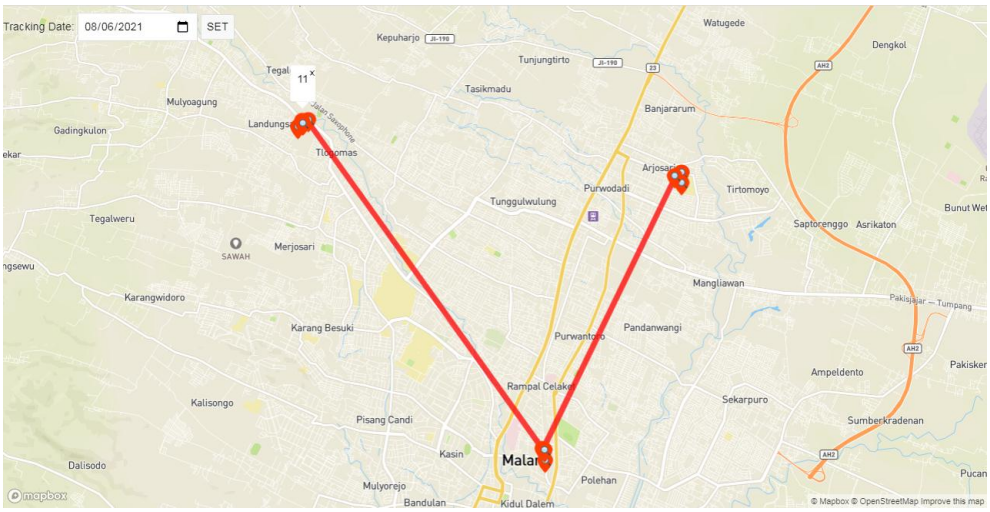
Pada halaman ini admin dan user atau pengendara dapat melihat lokasi terakhir pengendara yang terekam berdasarkan lokasi pemantauan yang telah ditentukan oleh sistem. Tanda lingkaran merah menunjukkan lokasi terakhir pengendara.



Gambar 4.11 Halaman *Last Position* Sistem Keamanan Mobilitas Orang di Kota Malang

8. Halaman *Tracking*

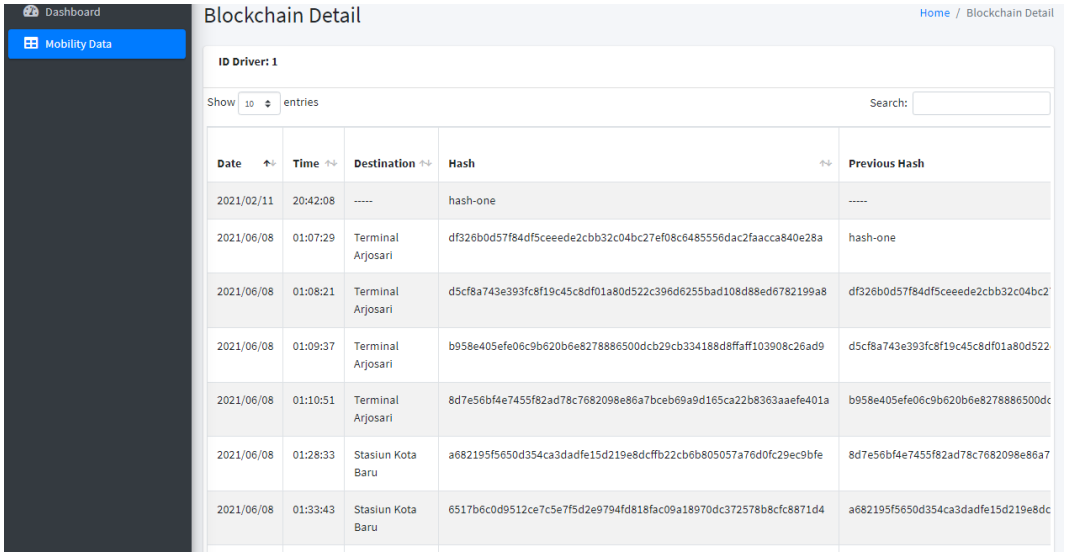
Pada halaman ini *user* dan admin dapat melihat perjalanan pengendara pada lokasi-lokasi pemantauan sistem di hari dan tanggal tertentu. *Default* perjalanan yang ditampilkan adalah perjalanan terakhir pengendara, namun user dan admin dapat melihat perjalanan pengendara di hari-hari sebelumnya dengan cara mengatur tanggal pada bagian pojok kiri atas halaman tersebut.



Gambar 4.12 Halaman *Tracking* Sistem Keamanan Mobilitas Orang di Kota Malang

9. Halaman *Detail Blockchain*

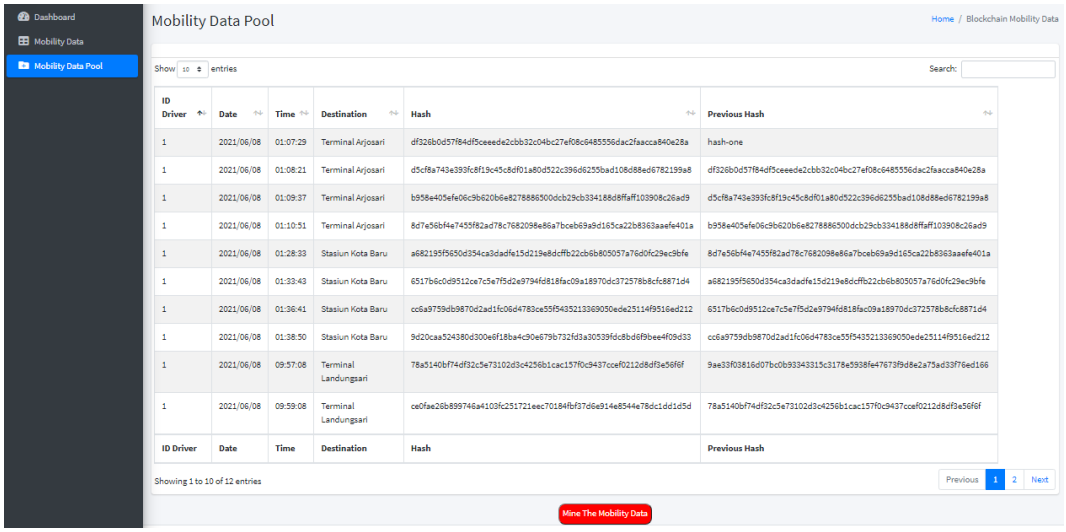
Pada halaman ini *user* dan admin dapat melihat rincian blok-blok penyusun blockchain pengendara. Informasi yang dapat diperoleh dari halaman ini adalah tanggal, waktu, lokasi yang dilintasi pengendara, nilai *hash*, nilai *previous hash*, dan penambang atau *miner*.



Gambar 4.13 Halaman *Detail Blockchain* Sistem Keamanan Mobilitas Orang di Kota Malang

10. Halaman *Mobility Data Pool*

Halaman ini merupakan halaman khusus yang hanya dapat diakses oleh *node* dalam jaringan *blockchain* yakni pengendara yang terdaftar pada sistem. Data mobilitas seluruh pengendara akan terkumpul disajikan pada halaman ini. *Proof of Work* digunakan pada proses ini, dimana *hash value* pada blok harus kurang dari target nilai yang sudah ditetapkan. Apabila nilai *hash* pada setiap blok sudah mencapai target yang ditetapkan, maka *hash value* dari blok tersebut berhasil tervalidasi. Nilai *hash* termasuk *nonce* diperiksa pada interval digit tertentu. Pada interval tersebut semua harus sama dengan 0, jika tidak maka nilai nonce bertambah, dan hash dihitung lagi sampai diperoleh yang diinginkan jumlah nol dalam *hash*.



Gambar 4.14 Halaman *Mobility Data Pool* Sistem Keamanan Mobilitas Orang di Kota Malang

4.2 Data Pengujian

Data yang digunakan pada sistem ini meliputi data identitas pengendara yang bertindak sebagai *node* pada jaringan *blockchain* dan data *range* koordinat dari lokasi-lokasi yang ditentukan dimana penentuan lokasi tersebut didasarkan atas tingkat keramaian dan aktivitas perpindahan orang di kawasan kota Malang. Data lokasi dan koordinat yang akan dipakai dalam riset ini diperoleh dari *Google Maps* Data Peta 2020. Koordinat lokasi pilihan akan tetap digunakan selama tidak ada perubahan pada penentuan lokasi pilihan. Dari data koordinat pilihan tiap lokasi di kota Malang tersebut nantinya akan dilakukan pencocokan antara koordinat *real time node*. Setelah tahap proses pencocokan koordinat, kemudian akan dilakukan proses berbagi data. Dalam uji coba ini digunakan 2 *node* dengan kemungkinan penambahan *node* yang masih bisa dilakukan. Berikut data identitas 2 node dalam uji coba sistem:

Tabel 4.1 Identitas pengendara yang bertindak sebagai *node* pada jaringan *blockchain*

ID	1	2
Nama	Dian Permana Putra	Karsono
NIK	3514110904950002	3514110812600003
Tempat Lahir	Pasuruan	Pasuruan
Tanggal Lahir	9/4/1995	8/12/1960
Gender	Laki-laki	Laki-laki
Agama	Islam	Islam
Alamat	Pandaan	Pandaan
Pekerjaan	Mahasiswa	Swasta
Nomor HP	087754321549	-
email	permana.pdian@gmail.com	permana.pdian2@gmail.com

4.3 Langkah-langkah Pengujian

Langkah-langkah uji coba untuk menerapkan algoritma SHA256 dan algoritma konsensus *Proof of Work* (*PoW*) untuk keamanan data jaringan desentralisasi adalah sebagai berikut:

a) Proses Inisialisasi awal, yakni memasang perangkat elektronik yang terdiri dari modul GPS dan *Raspberry Pi* pada kendaraan bermotor seseorang yang bertindak sebagai *node*. Perangkat ini memanfaatkan aki kendaraan bermotor sebagai catu daya. Perangkat ini mengirimkan

informasi lokasi per menit, kemudian melakukan pencocokan koordinat lintang dan bujur dengan *range* koordinat yang telah ditentukan untuk lokasi pemantauan dan menyimpan data dengan proses *hashing* mengunakan algoritma SHA256. Sebelum proses tersebut, admin menambahkan data pengendara sebagai *node* yang akan mengisi data yang disimpan dalam *blockchain*.

b) Proses validasi blok, terdapat konsep *Proof Of Work* (*PoW*) yang diperlukan untuk pengecekan nilai *hash* pada masing-masing blok yang dihasilkan melalui proses *hashing*.

c) Proses desentralisasi *blockchain*, semua catatan mobilitas orang yang dihimpun oleh masing-masing *node* akan saling tertaut dan tergabung menjadi sebuah *blockchain ledger record*.

Hasil dan evaluasi kinerja, secara kuantitatif mengevaluasi keakuratan modul dan mengukur serta menghitung waktu yang diperlukan dalam proses *Proof of Work* (*PoW*). Kesalahan relatif rata- rata dihitung untuk modul GPS dibandingkan dengan perangkat dan layanan yang tersedia serta pengujian terhadap proses *hashing* menggunakan metode SHA256.

4.4 Hasil Pengujian

Pada penelitian ini telah dilakukan pengujian untuk pengujian terhadap proses *hashing* SHA256 dan kecepatan *block-mining* time pada proses validasi menggunakan *Proof of Work* (*PoW*) dimana elemen-elemen pengujian tersebut juga disebutkan pada penelitian sebelumnya oleh Aziz dkk. (2020) yang

menyebutkan bahwa elemen-elemen pengujian *blockchain* adalah: *BlockHash, TestResult,* dan *TestTime.*

4.4.1 Hasil Uji Coba *Hashing* SHA256

Berikut hasil uji coba penerapan algoritma SHA256 untuk keamanan data jaringan desentralisasi. Dengan menggunakan dua *node* dilakukan pengambilan 24 kali, sesuai dengan tabel 4.2 dan tabel 4.3 di bawah ini.

Tabel 4.2 Tabel Hasil Pengujian *Hashing* SHA256 pengendara 1

Date	*Time*	*Destination*	*Hash*	*Previous Hash*
2021/ 02/11	20:42: 08	-----	hash-one	-----
2021/ 06/08	01:07: 29	Terminal Arjosari	df326b0d57f84df5ce eede2cbb32c04bc27 ef08c6485556dac2fa acca840e28a	hash-one
2021/ 06/08	01:08: 21	Terminal Arjosari	d5cf8a743e393fc8f1 9c45c8df01a80d522 c396d6255bad108d8 8ed6782199a8	df326b0d57f84df5ce eede2cbb32c04bc27 ef08c6485556dac2fa acca840e28a
2021/ 06/08	01:09: 37	Terminal Arjosari	b958e405efe06c9b6 20b6e8278886500dc b29cb334188d8ffaff 103908c26ad9	d5cf8a743e393fc8f1 9c45c8df01a80d522 c396d6255bad108d8 8ed6782199a8
2021/	01:10:	Terminal	8d7e56bf4e7455f82a	b958e405efe06c9b6

06/08	51	Arjosari	d78c7682098e86a7bceb69a9d165ca22b8363aaefe401a	20b6e8278886500dcb29cb334188d8ffaff103908c26ad9
2021/06/08	01:28:33	Stasiun Kota Baru	a682195f5650d354ca3dadfe15d219e8dcffb22cb6b805057a76d0fc29ec9bfe	8d7e56bf4e7455f82ad78c7682098e86a7bceb69a9d165ca22b8363aaefe401a
2021/06/08	01:33:43	Stasiun Kota Baru	6517b6c0d9512ce7c5e7f5d2e9794fd818fac09a18970dc372578b8cfc8871d4	a682195f5650d354ca3dadfe15d219e8dcffb22cb6b805057a76d0fc29ec9bfe
2021/06/08	01:36:41	Stasiun Kota Baru	cc6a9759db9870d2ad1fc06d4783ce55f5435213369050ede25114f9516ed212	6517b6c0d9512ce7c5e7f5d2e9794fd818fac09a18970dc372578b8cfc8871d4
2021/06/08	01:38:50	Stasiun Kota Baru	9d20caa524380d300e6f18ba4c90e679b732fd3a30539fdc8bd6f9bee4f09d33	cc6a9759db9870d2ad1fc06d4783ce55f5435213369050ede25114f9516ed212
2021/06/08	09:57:08	Terminal Landungsari	78a5140bf74df32c5e73102d3c4256b1cac157f0c9437ccef0212d8df3e56f6f	9d20caa524380d300e6f18ba4c90e679b732fd3a30539fdc8bd6f9bee4f09d33

2021/ 06/08	09:59: 08	Terminal Landungsari	ce0fae26b899746a4 103fc251721eec701 84fbf37d6e914e854 4e78dc1dd1d5d	78a5140bf74df32c5 e73102d3c4256b1ca c157f0c9437ccef021 2d8df3e56f6f
2021/ 06/08	10:01: 08	Terminal Landungsari	fc8c4b257a96c422c 876f5a2cac244e417 70eb1c34e03260fd6 d903b49a69be5	ce0fae26b899746a4 103fc251721eec701 84fbf37d6e914e854 4e78dc1dd1d5d
2021/ 06/08	10:01: 30	Terminal Landungsari	71260b2f591808a0b 22da8e44830be9b6b 6031301898dce27d7 e647915019b36	fc8c4b257a96c422c 876f5a2cac244e417 70eb1c34e03260fd6 d903b49a69be5

Tabel 4.3 Tabel Hasil Pengujian *Hashing* SHA256 pengendara 2

Date	*Time*	*Destination*	*Hash*	*Previous Hash*
2021/ 12/08	10:36: 45	-----	hash-one	-----
2021/ 12/17	08:45: 32	Terminal Arjosari	8d323745a79eb92b1 7dcc59a89c50d2845 4b75f42b780ee74bd 0f7b3b8f5aa07	hash-one
2021/ 12/17	08:48: 32	Terminal Arjosari	5ccd5223ac76a233d 79b7622849a5db640	8d323745a79eb92b1 7dcc59a89c50d2845

			99a115d1911ff7c3c617a3c02f85db	4b75f42b780ee74bd0f7b3b8f5aa07
2021/12/17	08:57:14	Stasiun Kota Baru	21104b898d6c3821fdce0208fdc18107cc8d5fddc7e9930116c73f6897d37074	5ccd5223ac76a233d79b7622849a5db64099a115d1911ff7c3c617a3c02f85db
2021/12/17	09:00:22	Stasiun Kota Baru	44f858bf07400700dc73b66bea7fcd065302e61ea4e46105d60d7318a44ea28c	21104b898d6c3821fdce0208fdc18107cc8d5fddc7e9930116c73f6897d37074
2021/12/17	09:11:19	Terminal Landungsari	a792dd9b8f49fde1d4217d576763f60000b1966cf8b0e21b56f31acab7b24965	44f858bf07400700dc73b66bea7fcd065302e61ea4e46105d60d7318a44ea28c
2021/12/17	09:23:37	Terminal Arjosari	911fdb9333a469fecd26d1985d6b9b23dc8d6658d39a0334a12d1b6f7c11a335	a792dd9b8f49fde1d4217d576763f60000b1966cf8b0e21b56f31acab7b24965
2021/12/17	09:26:37	Terminal Arjosari	10c376d4eb140b63899b36d700acd1378b8109c6b7d21d157884b49e87ed916d	911fdb9333a469fecd26d1985d6b9b23dc8d6658d39a0334a12d1b6f7c11a335
2021/	09:36:	Stasiun Kota	13bedd9357805e5eb	10c376d4eb140b638

12/17	08	Baru	faddbaac40bb2dee07c277c11ede83a3778569f6d961cb8	99b36d700acd1378b8109c6b7d21d157884b49e87ed916d
2021/12/17	09:39:08	Stasiun Kota Baru	2f34c49ed04561308cad90c72af56d7754c499ae87bcd9747499815d07a10008	13bedd9357805e5ebfaddbaac40bb2dee07c277c11ede83a3778569f6d961cb8
2021/12/17	09:50:04	Terminal Landungsari	227f199a1d47c7022a7e33e25ef2f570c22a03eb35ad4e557f0158861182310d	2f34c49ed04561308cad90c72af56d7754c499ae87bcd9747499815d07a10008
2021/12/17	09:53:04	Terminal Landungsari	999515185234a0eb380cc9ee4218f16da9797c7a879f2716c111dd9439f3a484	227f199a1d47c7022a7e33e25ef2f570c22a03eb35ad4e557f0158861182310d
2021/12/17	09:56:04	Terminal Landungsari	25ace8163804816d88cdd0e49427b347e1123538e65242bc9c7eecd7dc5299db	999515185234a0eb380cc9ee4218f16da9797c7a879f2716c111dd9439f3a484

Sesuai dengan tabel 4.2 dan tabel 4.3, nilai *hash* dan *previous hash* adalah sesuai, dimana metode hashing SHA256 dapat berjalan untuk proses enkripsi dan

antar blok tetap saling terhubung. Nilai kevalidan dapat dilihat dari hasil uji coba tiap blok pada data setiap melakukan mobilitas.

4.4.2 Hasil Uji Coba *Proof of Work (PoW)*

Dalam proses pengukuran *block-mining time* digunakan konsep *Proof Of Work (PoW,)* metode ini digunakan untuk pengecekan *hash value* pada setiap blok yang dihasilkan melalui proses *hashing*. Proses ini akan melakukan validasi dan mengukur *hash value* yang terdapat pada setiap blok. Ketika *hash value* pada setiap blok sudah mencapai target yang ditetapkan, maka *hash value* dari blok tersebut berhasil tervalidasi dan terhitung kecepatan *block-mining time*-nya. Nilai *timestamp* awal saat pengujian *PoW* adalah 1623194141.70. Berikut hasil uji coba penerapan algoritma *PoW* untuk keamanan data jaringan desentralisasi dapat dilihat sebagai berikut.

Tabel 4.4 Tabel Hasil Pengujian *Proof of Work (PoW)*

no	*Timestamp*	*Binary*	*Difficulty*	*Nonce*	*Mining Time*
1	1623194141.71	1**00**11110100000000010010110000110110000101100101100000110010010011111000101110110110011001001110101111011000110010000110000100001100010011000011110101010110101101100100101000101001000111011110100111000010110111000000000010110110111101000000000010100100110101	2	1	10 ms
2	1623194141.83	1**0**01001011100001001000110000011011000000001010010001100011100011100000101011010111111001111100111010100101110001100110111001100000100000000110000100011111011001101101100111001011000010000101111100110001011	1	8	12 ms

		010110010001100011011000101001 1010011111110010			
3	1623194141. 89	1**0**0100010001100010011010110001 000001010011100011110101100000 001101010101110011011100111110 100000011101101100111100000101 100011100100100100100010011000 000111111011101101001100001011 111100111010111110111101111010 110100011010100001011011000101 0010101110100101	1	1	60 ms
4	1623194141. 98	1**0**0110100000011111110110100000 000001000101011110100011111010 100000100001001110100100011000 011011000111001000000100111000 001111001110100011001101010000 100101101111001111111110111000 111110101110011011111100000100 000101010010111001110101010100 01011110100	1	12	90 ms
5	1623194142. 05	1**0**0101011101000111101011110111 010000000111111100000111111110 001001100101001010111001100111 001100001011011001011110010100 110000010101100110000110011101 110110100110011000010110101100 110100011001111101001100110001 000111001000100100111001001000 00101010110000	1	1	70 ms
6	1623194142. 12	1**0**0010000111010011000001011011 100101010000100011100110110110 111100010100100000110011110010 100011100100110100110000101000 101111001001010110011111011110 110000000010000001000000110111 0 100000011010010001001100101011 110110000111110011111110011000 0100001101001	1	16	70 ms
7	1623194142. 18	1**0**0010101110110111110111100010 010100001011101100111000010000 010011100101101000011110100110 101001101011010110001000010101 100011010001100011110110101100 101011001000110010111101000001 111001010001101000101110111011 001100010000101101010111110101	1	1	60 ms

		011111001111110			
8	1623194142.26	1**0**1000110110001101010111010001 1101001110010010001100001010011 1010011100011100000110110000100 0111010111010001010001010000110 0110101001110000111100010010011 1011011101010101101110010010011 1001001101101001111001000010011 1001110010101000100001000111000 01111000001111	1	24	80 ms
9	1623194142.33	1**0**1000110100111100111101011000 1111100011010110100010100000011 0100011010011110100110111010011 1101001001001010000111110010100 0011110100011001110011100111110 1111110101010011111011000110110 0100110011110110100000111011010 0011101001111101000101111011111 110101011000	1	1	70 ms
10	1623194142.39	1**0**1101000001000010100001100111 1111110010011100100100001001100 0000001010000110010101110100000 0100110110000100101111010100100 1001100001001101011000000101011 0101001100010101101100000000110 0001001011010011111101010000110 0010011110010001010110110101000 1111010010001	1	4	60 ms
11	1623194142.45	1**0**1101101101101100011011011010 0001000111111111001000010111100 0001000111001100100001000000010 0100011101000100001101011010100 0010000000110100011011101100101 1011110000100101101111011011111 1001100100111111101000100001110 0111011110010101010001001110011 11100001111110	1	6	60 ms
12	1623194142.50	1**0**1001111101111001000000011110 1101011101101101001000010110100 0111011101100000011010111111010 1110010101001110001011100101100 1101101111110011010100011001100 0100000011101000101100011011010 0000101011101101001000010011000 0010011110111000011011111001110 101000110010110	1	1	50 ms

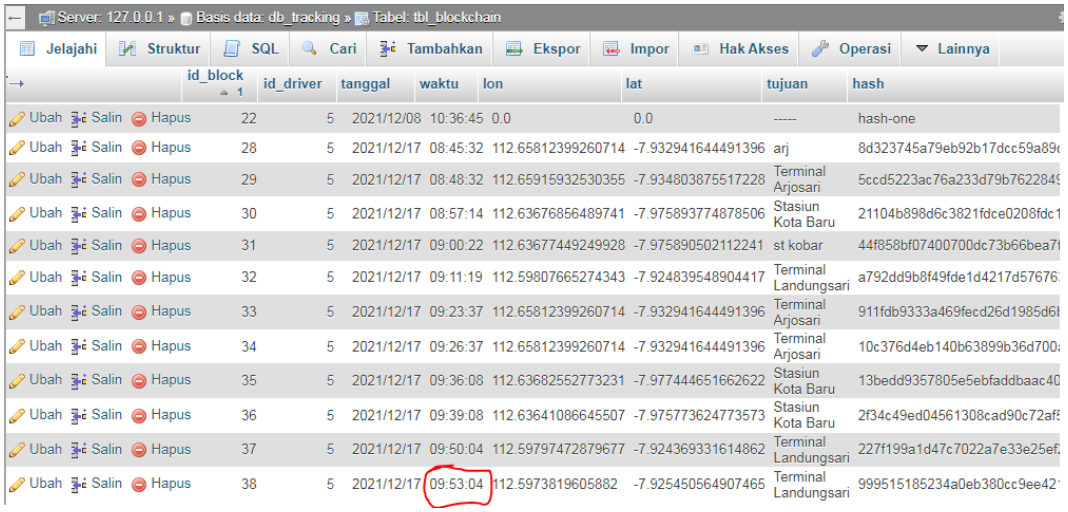
13	1623194142.61	1**0**1010100011000100010100101010101111010101101110011000010100011111110001111111000101101010000100111110111001000001011101000011011111111111101110111011001000000111000101000000000101011110010011101100100110111101011110101111100010001101011010101110010	1	8	110 ms
14	1623194142.67	1**0**0000010110000011110101010111111010100110100000010110101000010011001110100011100000000111001101010111010000101010000001011111100100000010110100001100111010110001001010001000011010010011000110101001101001100000111111111111111111110000101011101000111001100	1	1	60 ms
15	1623194142.76	1**0**1001010110001101110101110101110000000100001011100111001100010000011111101010111001111110010001000101101100010000000001011001011100010110010001001000111100101001110100001110001000100011111111001010001111111111100000111101010111011001001000011011	1	16	90 ms
16	1623194142.85	1**0**1000001011010110110111011010101000001110001100110110011101011111010010000010110001111101100001010000010010000100111011101010010010000111011100010011101100001111100001011110111001011010100101011110010000110100111001100	1	1	90 ms
17	1623194142.92	1**0**1100011011111010110011110101110001011111111111011001111001010011110111110010101000100011110010110001110010110111000100011111010001011011111111000000111111100110100011101010111001011011111011101010001001111111101111000011010110010001100001000100	1	12	70 ms
18	1623194143.	1**000**10011001001100101101010101	3	10	80 ms

	00	10101011111111000010101101000010001101011010011011100001110000100110001000001000000101111000010110111001111011101001100001010110011010001111000010011101000111001111000000110000000010001110011100111100001111110001011000001			
19	1623194143.07	1**0**01110001110101111110110111110101010000011101100111010110000011000110100111010011011111111101100010111111101111000101001000111100111011100011010011101111000111010000010100000011100000110010011100011010011001110100000011110110000111011011011000010100100110001	1	1	70 ms
20	1623194143.14	1**000**10110000001101010100001110111100011001010111001000010110101110111011011001000010110100110011100111000000110011100011101111010001010110010001010100000101100111001000100001000001101100100110101101000010100000010010111111110100000001001	3	5	70 ms

Dari hasil pengukuran *block-mining time* pada tabel 4.4, diketahui bahwa *PoW* berjalan dengan baik dimana hal itu ditunjukkan dengan jumlah nol setelah angka 1 pertama pada nilai binary yang memenuhi nilai target difficulty. Dari 20 data binary yang didapat dari proses *hashing*,dihasilkan satu data dengan block-mining time 10 milidetik, 12 milidetik, 50 milidetik dan 110 milidetik, dua data dengan block-mining time 80 milidetik, lima data dengan block-mining time 60 milidetik, enam data dengan block-mining time 70 milidetik, serta tiga data dengan block-mining time 90 milidetik Kecepatan waktu *Mining* rata-ratanya mencapai 66,6 milidetik.

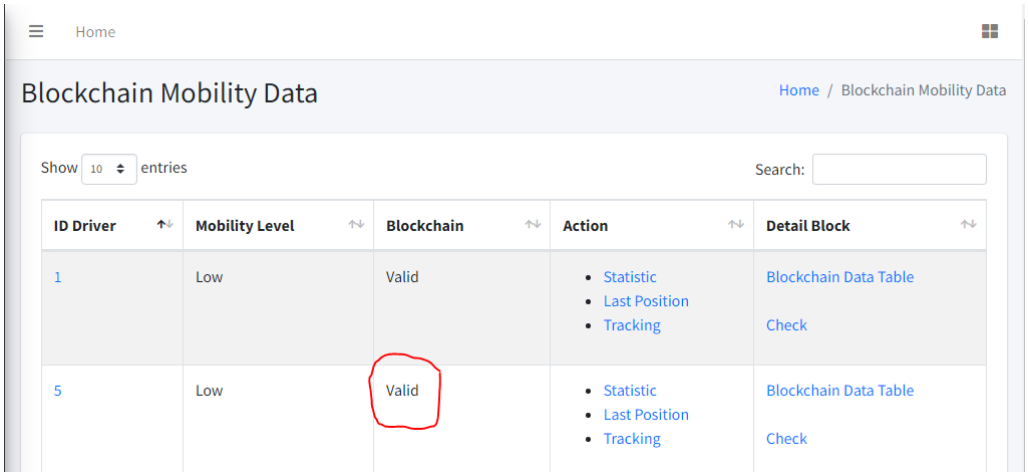
4.4.3 Hasil Uji Coba Keamanan

Uji coba ini dimaksudkan untuk mengetahui apakah data mobilitas aman terhadap potensi peretasan. Dilakukan percobaan perubahan bagian isi dari data mobilitas, dimana tindakan tersebut menyebabkan perubahan nilai *hash* pada blok yang diretas.



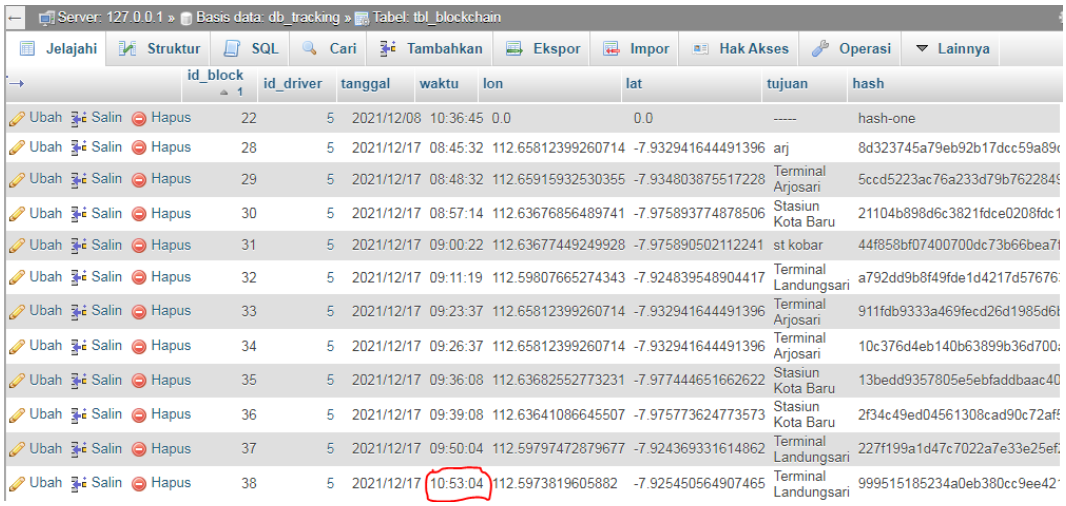
Gambar 4.15 Data Mobilitas Sebelum Diubah

Data waktu pada tanda merah adalah data asli sebelum dilakukan percobaan perubahan nilai. Detail waktu pada blok data id 38 menunjukkan nilai waktu 09:53:04.



Gambar 4.16 Tampilan Blockchain Data Mobilitas Sebelum Adanya Perubahan

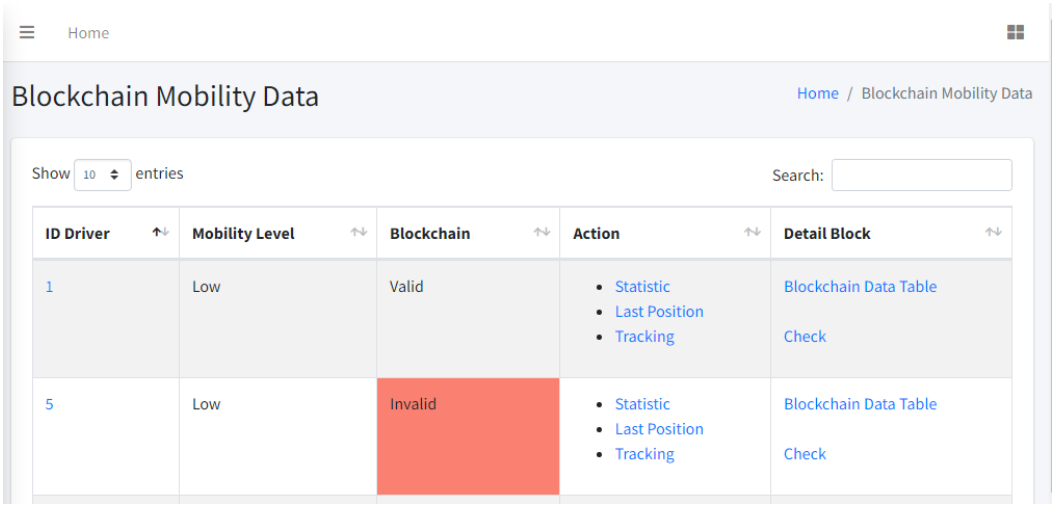
Pada halaman *blockchain* mobilitas data dapat dilihat pada tanda merah, terdapat keterangan “Valid”. Artinya data *blockchain* masih asli dan belum ada perubahan data.



	id_block	id_driver	tanggal	waktu	lon	lat	tujuan	hash
Ubah Salin Hapus	22	5	2021/12/08	10:36:45	0.0	0.0	-----	hash-one
Ubah Salin Hapus	28	5	2021/12/17	08:45:32	112.65812399260714	-7.932941644491396	arj	8d323745a79eb92b17dcc59a89
Ubah Salin Hapus	29	5	2021/12/17	08:48:32	112.65915932530355	-7.934803875517228	Terminal Arjosari	5ccd5223ac76a233d79b762284
Ubah Salin Hapus	30	5	2021/12/17	08:57:14	112.63676856489741	-7.975893774878506	Stasiun Kota Baru	21104b898d6c3821fdce0208fdc1
Ubah Salin Hapus	31	5	2021/12/17	09:00:22	112.63677449249928	-7.975890502112241	st kobar	44f858bf07400700dc73b66bea7
Ubah Salin Hapus	32	5	2021/12/17	09:11:19	112.59807665274343	-7.924839548904417	Terminal Landungsari	a792dd9b8f49fde1d4217d57676
Ubah Salin Hapus	33	5	2021/12/17	09:23:37	112.65812399260714	-7.932941644491396	Terminal Arjosari	911fdb9333a469fecd26d1985d6
Ubah Salin Hapus	34	5	2021/12/17	09:26:37	112.65812399260714	-7.932941644491396	Terminal Arjosari	10c376d4eb140b63899b36d700
Ubah Salin Hapus	35	5	2021/12/17	09:36:08	112.63682552773231	-7.977444651662622	Stasiun Kota Baru	13bedd9357805e5ebfaddbaac40
Ubah Salin Hapus	36	5	2021/12/17	09:39:08	112.63641086645507	-7.975773624773573	Stasiun Kota Baru	2f34c49ed04561308cad90c72af
Ubah Salin Hapus	37	5	2021/12/17	09:50:04	112.59797472879677	-7.924369331614862	Terminal Landungsari	227f199a1d47c7022a7e33e25ef
Ubah Salin Hapus	38	5	2021/12/17	10:53:04	112.5973819605882	-7.925450564907465	Terminal Landungsari	999515185234a0eb380cc9ee42

Gambar 4.17 Data Mobilitas Setelah Diubah

Data waktu pada tanda merah adalah data yang diubah sebelum dilakukan percobaan perubahan nilai. Detail waktu pada blok data id 38 menunjukkan nilai waktu 09:53:04 kemudian nilai waktunya diubah menjadi 10:53:04. Dalam aturan *blockchain* tidak diperbolehkan adanya perubahan dan penghapusan data.



Gambar 4.18 Tampilan Blockchain Data Mobilitas Setelah Adanya Perubahan

Pada halaman *blockchain* mobilitas data dapat dilihat keterangan yang semula menunjukkan informasi “Valid”, setelah adanya perubahan pada nilai waktu menyebabkan status *blockchain* menjadi “*Invalid*”. Perubahan nilai waktu atau perubahan data sekecil apapun tetap akan merubah nilai *hash.* Ketika nilai *hash* sudah tidak lagi sama, artinya data *blockchain* sudah tidak asli dan telah terjadi perubahan atau penghapusan data.

Keamanan privasi individu merupakan masalah penting yang perlu ditangani. Kasus-kasus peretasan sangat sering terjadi dalam beberapa tahun terakhir baik dalam negeri maupun di luar negeri. Beberapa kasus yang terjadi di luar negeri: pada tahun 2015, kelompok peretas sipil menyebarkan data lokasi bus yang tidak standar di Baltimore. Pada 2016, informasi milik 57 juta pelanggan dan pengemudi uber tersebar. Kemudian pada tahun 2018, agen transportasi regional Ontario, *server* Metrolinx mendapat serangan peretasan (Lopez dan Farooq, 2018). Sebagian besar data disimpan pada *server* milik perusahaan-perusahaan. Teknik enkripsi dengan kunci publik dan kunci privat telah dimanfaatkan penggunaannya untuk keamanan data dan komunikasi (Farooq dkk., 2015). Namun peneliti-peneliti menemukan bahwa *blockchain* dapat difungsikan untuk menangani masalah privasi yakni dengan mendesentralisasikan informasi dan di sisi lain individu berperan sebagai pemilik tunggal dan pengendali atas informasi mereka. Teknologi *blockchain* mempunyai potensi untuk melindungi informasi mobilitas pribadi individu dengan keunggulannya yang sulit diretas, menghasilkan transaksi yang aman dan transparan bagi pihak-pihak yang berbagi data melalui jaringan *blockchain* (Lopez dan Farooq, 2018). Berkenaan dengan penyampaian

informasi kepada pihak yang berhak dijelaskan dalam al-Quran surat an-Nisaa' ayat 58, yaitu:

إِنَّ اللهَ يَأْ مُرُ كُمْ أَنْ تُؤَدُّوا الْأَمَا نَاتِ إِلَى أَهْلِهَا وَإِذَا حَكَمْتُمْ بَيْنَ النَّاسِ أَنْ تَحْكُمُوا بِالْعَدْلِ ۚ إِنَّ اللهَ نِعِمَّا يَعِظُكُمْ بِهِ ۗ إِنَّ اللهَ كَانَ سَمِيْعًا بَصِيْرًا

"Sesungguhnya Allah menyuruh kamu menyampaikan amanat kepada yang berhak menerimanya, dan (menyuruh kamu) apabila menetapkan hukum di antara manusia supaya kamu menetapkan dengan adil. Sesungguhnya Allah memberi pengajaran yang sebaik-baiknya kepadamu. Sesungguhnya Allah adalah Maha mendengar lagi Maha melihat" (QS. An-Nisaa'/4:58).

Aksi peretasan merupakan bagian dari tindakan pencurian, dimana peretas memperoleh data tanpa mendapatkan izin dari pemilik data. Dalam Islam tentu hal tersebut dilarang dan bisa dikatakan sebagai tindakan pencurian serta melanggar hak dasar manusia. Tindakan peretasan memiliki jenis yang bervariasi, ada yang berupa penyusupan, pencurian data ataupun pelanggaran terhadap suatu perjanjian.

Sumber hukum islam memiliki sifat yang fleksibel dimana sumber hukum islam dapat diterapkan pada berbagai kondisi, situasi dan permasalahan. Dalam menemukan sumber hukum islam dalam kasus peretasan, dalil-dalil yang ada di*istinbat*kan dan diwujudkan dalam model kaidah fikih. Tentunya ini menunjukkan bahwa hukum islam bisa memberikan sudut pandang yang berbeda dalam masalah-masalah terkait teknologi informasi termasuk kasus peretasan. Penyusupan merupakan jenis dari aksi peretasan yang memiliki makna yakni masuk tanpa izin dan tanpa sepengetahuan pemilik barang. Semua bentuk tindakan penyusupan yang menyangkut objek dan subjek memiliki hukum dasar dalam al-Quran surat an-Nur ayat 27, yaitu:

يَا أَيُّهَا الَّذِينَ آمَنُوا لَا تَدْخُلُوا بُيُوتًا غَيْرَ بُيُوتِكُمْ حَتَّىٰ تَسْتَأْنِسُوا وَتُسَلِّمُوا عَلَىٰ أَهْلِهَا ۚ ذَٰلِكُمْ خَيْرٌ لَكُمْ لَعَلَّكُمْ تَذَكَّرُونَ

"Hai orang-orang yang beriman, janganlah kamu memasuki rumah yang bukan rumahmu sebelum meminta izin dan memberi salam kepada penghuninya. Yang demikian itu lebih baik bagimu, agar kamu (selalu) ingat" (QS. An-Nur/24:27).

Kerangka kerja blockchain diterapkan untuk menangani masalah privasi dan keamanan. Masyarakat sipil merupakan salah satu pihak yang berperan sebagai pemilik dan pengumpul data mereka masing-masing. Pemilik data membuat aturan transaksi, sehingga pihak yang menyetujui aturan tersebut dapat melakukan transaksi informasi atau data yang telah dienkripsi oleh pemilik data melalui jaringan blockchain.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil pengujian serta analisis terhadap sistem yg telah dibuat, dapat ditarik sebuah kesimpulan sebagai berikut.

1. Kinerja algoritma SHA256 dalam menghasilkan nilai hash sebagai penghubung blok-blok penyusun blockchain pada sistem keamanan mobilitas yang telah dibuat, berhasil berjalan dengan baik. Ditunjukkan dengan hasil pengujian pada tabel 4.2, dimana dari 12 perpindahan lokasi yang telah dilakukan pengendara, semuanya menunjukkan pasangan antara nilai hash dan previous hash dengan nilai yang sama. Hal ini juga menunjukkan bahwa blockchain yang diterapkan pada sistem yang dibuat bernilai valid, dimana *hashing* SHA256 berjalan dengan baik. Blok-blok saling terhubung satu sama lain melalui nilai *hash* dan *previous hash* masing-masing blok**.** Metode ini cocok digunakan untuk jenis data-data yang saling terhubung dan membutuhkan keamanan akan tingkat orisinalitas data.
2. Kecepatan block-mining time Proof of Work pada sistem yang dibuat terhitung cepat. Ditunjukkan dengan hasil pengujian pada tabel 4.3, diketahui bahwa *PoW* berjalan dengan baik dan cepat dimana hal itu terlihat dari jumlah nol setelah angka 1 pertama pada nilai binary setiap data yang memenuhi nilai target difficulty dengan kecepatan waktu *Mining* rata-rata adalah 66,6 milidetik. *Proof Of Work (PoW)* pada sistem yang dibuat diperlukan untuk pengecekan *hash value* pada setiap blok yang dihasilkan melalui proses

hashing. Proses validasi dengan *Proof Of Work (PoW)* akan melakukan validasi pada *hash value* yang terdapat pada setiap blok. Ketika *hash value* pada setiap blok sudah mencapai target yang ditetapkan, maka *hash value* dari blok tersebut berhasil tervalidasi.

3. Sistem memiliki tingkat keamanan, dimana setelah dilakukan percobaan perubahan detail waktu pada blok data id 38 yang semula menunjukkan nilai waktu 09:53:04 kemudian nilai waktunya diubah menjadi 10:53:04. Dalam aturan *blockchain* tidak diperbolehkan adanya perubahan atau penghapusan data. Pada halaman *blockchain* mobilitas data dapat dilihat keterangan yang semula menunjukkan informasi "Valid", setelah adanya perubahan pada nilai waktu menyebabkan status *blockchain* menjadi "*Invalid*". Perubahan nilai waktu atau perubahan data sekecil apapun tetap akan merubah nilai *hash*. Ketika nilai *hash* sudah tidak lagi sama, artinya data *blockchain* sudah tidak asli dan telah terjadi perubahan atau penghapusan data. Peretas harus mengganti semua nilai *hash* dan *previous hash* yang ada satu per satu untuk menjadikan data *blockchain* menjadi valid. Hal itu tentu sulit dilakukan apalagi ketika blok data dalam *blockchain* tersebut banyak.

5.2 Saran

Pemanfaatan algoritma SHA256 dan *PoW* untuk mengamankan dan menjaga orisinalitas data memang diperlukan pada semua aspek, termasuk data perjalanan. Namun pemanfaatannya akan lebih baik lagi apabila diterapkan pada aspek lain yang memiliki urgensi lebih pada tingkat keamanan dan keaslian data. Sistem berbasis blockchain pada riset ini masih jauh dari kata sempurna. Masih

ada banyak sekali kekurangan yang perlu dibenahi sebagai peningkatan untuk riset selanjutnya. Penggunaan modul GPS masih sedikit kurang konsisten dalam memberikan informasi data lokasi. Pengembangan terhadap sistem ini mungkin bisa diterapkan pada tracking kasus covid-19 atau tracking perjalanan paket pada jasa ekspedisi pengiriman barang dengan tetap menjaga keamanan, urutan serta keaslian data.

DAFTAR PUSTAKA

Astarita, V. Giofrè, V.P. Mirabelli, G. and Solina, V. 2020. *A Review of Blockchain-Based Systems in Transportation.* In Information 2020, 11, 21

Aziz, A. Riaz, M. T. Jahan, M. S and Ayub, K. 2020. *Meta-model for Stress Testing on Blockchain Nodes*. In 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Pakistan, pp. 1-4

BPS Kota Malang. 2019. *Kota Malang Dalam Angka 2019.* Malang: ASIA

Bocek, T. Rodrigues, B.B. Strasser, T and Stiller, B. 2017. *Blockchains everywhere-a use-case of blockchains in the pharma supply-chain*. In Proceedings of the 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), Lisbon, Portugal, 8–12 May 2017; pp. 772–777

Caro, M.P. Ali, M.S. Vecchio, M and Giaffreda, R. 2018. *Blockchain-based traceability in Agri-Food supply chain management: A practical implementation.* In Proceedings of the 2018 IoT Vertical and Topical Summit on Agriculture, Tuscany, Italy, 8–9 May 2018; pp. 1–4

Castro, M dan Liskov B. 1999. *Practical byzantine fault tolerance.* In Proceedings of the 3rd USENIX Symposium on Operating SystemsDesign and Implementation (OSDI), New Orleans, Louisiana, USA, pp. 173–186

Christidis, K dan Devetsik, M. 2016. *Blockchains and Smart Contracts for the Internet of Things.* In IEEE Access, vol. 4, pp. 2292–2303

Covid19.go.id. (2020, 1 Agustus). *Infografis Covid-19*. Diakses pada 3 Agustus 2020, dari https://covid19.go.id/p/berita/infografis-covid-19-1-agustus-2020

Departemen Agama RI. 2007. *Al-Qur'an dan Terjemahnya*. Semarang: CV. Toha Putra.

Department for Transport (UK), *Motorcycling*. Available: http://think.direct.gov.uk/motorcycles.html

Dinh, T.T.A. Liu, R. Zhang, M. Chen, G. Ooi, B.C dan Wang, J. 2018. *Untangling Blockchain: A Data Processing View of Blockchain Systems.* In IEEE Transactions on Knowledge and Data Engineering, Vol. 30, Issue: 7

Eyal, I and Sirer, E.G. 2014. *Majority is not enough: Bitcoin mining is vulnerable.* In Financial Cryptography and Data Security Springer, pages 436-454

Farooq, B. Beaulieu, A. Ragab, M dan Ba, V. D. 2015. *Ubiquitous monitoring of pedestrian dynamics: Exploring wireless ad hoc network of multi- sensor technologies.* In SENSORS, 2015 IEEE. IEEE, pp. 1–4

Feller, W. 1957. *An introduction to probability theory and its applications*, Vol.1, 3rd ed. New York: John Willey & Sons, Inc.

Gemeliarana, I. G. A. K and Sari, R. F. 2018. *Evaluation of Proof of Work (POW) Blockchains Security Network on Selfish Mining.* In 2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), Yogyakarta, Indonesia, pp. 126-130

Gervais, A. 2016. *On the Security and Performance of Proof of Work Blockchains.* In CCS'16 Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security

Gervais, A. Ritzdorf, H. Karame, G.O and Capkun, S. 2015. *Tampering with the Delivery of Blocks and Transactions in Bitcoin.* In Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur. - CCS '15, pp. 692–705

Han, S. 2017. *A Visual Demo of Blockchain Technology*, https://blockchaindemo.io, diakses pada 10 April 2020 pukul 10.27

Hutabarat, D. P. Hendry, H. Pranoto, J. A and Kurniawan, A. 2016. *Human tracking in certain indoor and outdoor area by combining the use of RFID and GPS.* In 2016 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob), Bandung, pp. 59-62

Jo, J. Kim, H. Park, H and Yoon, D. 2015. *A Monitoring System to Understand Postal Motorcyclist's Driving Behavior.* In 2015 IEEE 18th International Conference on Intelligent Transportation Systems, Las Palmas, pp. 2883-2888

Johnston, C. Brooks, and Savage, H. 2008. *Fatal and serious road crashes involving motorcyclists*. Monograph 20 Canberra: Department of Infrastructure, pp. 26

Liang, X. Shetty, S. Tosh, D. Kamhoua, C. Kwiat, K and Njilla, L. 2017. *Provchain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability.* In International Symposium on Cluster, Cloud and Grid Computing.IEEE/ACM

Liang, X. Shetty, S. Zhang, L. Kamhoua, C and Kwiat, K. 2017. *Man in the cloud (mitc) defender: Sgx-based user credential protection for synchronization applications in cloud computing platform.* In The 10thIEEE International Conference on Cloud Computing (CLOUD 2017)

Lopez, D and Farooq, B. 2018. *A blockchain framework for smart mobility.* In IEEE International Smart Cities Conference (ISC2)

Lu, Q and Xu, X. 2017. *Adaptable Blockchain-Based Systems: A Case Study for Product Traceability.* In IEEE Software, vol. 34, no. 6, pp. 21-27

Mankar, R.V and Nipanikar, S. I. 2013. *C Implementation of SHA256 Algorithm*. Pune: Pune University

Ministry of Industry and Information Technology. 2016. *China Block Chain Technology and Application Development White Paper*. China Block Chain Technology and Industry Development BBS

Nakamoto, S. 2008. *Bitcoin: A peer-to-peer electronic cash system*

National Institute of Standards and Technology (NIST). 1994. *Secure Hash Standard*. FIPS Publication 180-1. April. 176

National Institute of Standards and Technology (NIST). 2002. FIPS 180-2. http://csrc.nist.gov/encryption/tkhash.html. 176, 177

Pabla, J. Sharma, V and Krishnamurthi, R. 2019. *Developing a Secure Soldier Monitoring System using Internet of Things and Blockchain*. In 2019 International Conference on Signal Processing and Communication (ICSC), NOIDA, India, pp. 22-31

Phithakkitnukoon, S. Horanont, T. Di Lorenzo, G. Shibasaki, R and Ratti C. 2010. *Activity-aware map: Identifying human daily activity pattern using mobile phone data.* In Human Behavior Understanding, A. A. Salah, T. Gevers, N. Sebe, and A. Vinciarelli, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 14–25

Sebastian, A. 2007. *Implementasi dan Perbandingan Performa Algoritma Hash SHA-1, SHA256, dan SHA-512*. Bandung: Institut Teknologi Bandung

Swan, M. 2015. *Blockchain: Blueprint for a new economy*. O'Reilly Media, Inc

Vukolic, M. 2015. *The quest for scalable blockchain fabric: proof-of-work vs. bft replication*. In Open Problems in Network Security –iNetSec

Vu, Q. H. Lupu, M and Ooi, B. C. 2009. *Peer-to-Peer Computing Principlesand Applications.* Springer-Verlag

Zhihong, T. Yang, J and Zhang, J. 2009. *Location-based Services Applied to an Electric Wheelchair Based on the GPS and GSM Networks.* In 2009 International Workshop on Intelligent Systems and Applications, Wuhan,

pp.1-4

LAMPIRAN

Penggalan Kode Program SHA256

```
function SHA256(s){
    var chrsz   = 8;
    var hexcase = 0;
    function safe_add (x, y) {
          var lsw = (x & 0xFFFF) + (y & 0xFFFF);
          var msw = (x >> 16) + (y >> 16) + (lsw >> 16);
          return (msw << 16) | (lsw & 0xFFFF);
    }
    function S (X, n) { return ( X >>> n ) | (X << (32 - n)); }
    function R (X, n) { return ( X >>> n ); }
    function Ch(x, y, z) { return ((x & y) ^ ((~x) & z)); }
    function Maj(x,y,z) { return ((x&y) ^ (x&z) ^ (y&z)); }
    function Sigma0256(x){return (S(x,2) ^ S(x,13) ^ S(x,22));}
    function Sigma1256(x){return (S(x,6) ^ S(x,11) ^ S(x,25));}
    function Gamma0256(x){return (S(x,7) ^ S(x,18) ^ R(x,3));}
    function Gamma1256(x){return (S(x,17)^S(x,19)^R(x,10));}
    function core_sha256 (m, l) {
       var K = new Array(0x428A2F98, 0x71374491, 0xB5C0FBCF, 0xE
       9B5DBA5, 0x3956C25B, 0x59F111F1, 0x923F82A4, 0xAB1C5ED5,
       0xD807AA98, 0x12835B01, 0x243185BE, 0x550C7DC3, 0x72BE5D7
       4, 0x80DEB1FE, 0x9BDC06A7, 0xC19BF174, 0xE49B69C1, 0xEFBE
       4786, 0xFC19DC6, 0x240CA1CC, 0x2DE92C6F, 0x4A7484AA, 0x5C
       B0A9DC, 0x76F988DA, 0x983E5152, 0xA831C66D, 0xB00327C8, 0
       xBF597FC7, 0xC6E00BF3, 0xD5A79147, 0x6CA6351, 0x14292967,
       0x27B70A85, 0x2E1B2138, 0x4D2C6DFC, 0x53380D13, 0x650A735
       4, 0x766A0ABB, 0x81C2C92E, 0x92722C85, 0xA2BFE8A1, 0xA81A
       664B, 0xC24B8B70, 0xC76C51A3, 0xD192E819, 0xD6990624, 0xF
       40E3585, 0x106AA070, 0x19A4C116, 0x1E376C08, 0x2748774C,
       0x34B0BCB5, 0x391C0CB3, 0x4ED8AA4A, 0x5B9CCA4F, 0x682E6FF
       3, 0x748F82EE, 0x78A5636F, 0x84C87814, 0x8CC70208, 0x90BE
       FFFA, 0xA4506CEB, 0xBEF9A3F7, 0xC67178F2);
       var HASH = new Array(0x6A09E667, 0xBB67AE85, 0x3C6EF372,
       0xA54FF53A, 0x510E527F, 0x9B05688C, 0x1F83D9AB, 0x5BE0CD1
       9);
       var W = new Array(64);
       var a, b, c, d, e, f, g, h, i, j;
       var T1, T2;
       m[l >> 5] |= 0x80 << (24 - l % 32);
       m[((l + 64 >> 9) << 4) + 15] = l;
           for ( var i = 0; i<m.length; i+=16 ) {
                 a = HASH[0];
                 b = HASH[1];
                 c = HASH[2];
                 d = HASH[3];
                 e = HASH[4];
                 f = HASH[5];
                 g = HASH[6];
                 h = HASH[7];
                 for ( var j = 0; j<64; j++) {
                      if (j < 16) W[j] = m[j + i];
                                  ...
```

Kode Program *Proof of Work (PoW)*

```
eval_res, jsFile = js2py.run_file('sha256js.js')
MINE_RATE = 1000
difficulty = results[10]
nonce = 0
timestampbefore = p_timestamp
loop = True

while loop:
     nonce = nonce+1
     timestamp_ = time.time()

     if difficulty < 1:
         difficulty = 1
     if (timestamp_ - timestampbefore) > MINE_RATE:
         difficulty = difficulty - 1
     difficulty = difficulty + 2
     if difficulty >= 8:
         difficulty = 1

     timestampString=str(timestamp_)
     dtTimestamp=timestampString[0:10]
     data =
     dtTimestamp+str(tgl)+str(waktu)+str(lat)+str(lng)+str(p_hash)
     +str(nonce)+str(difficulty)
     vhashing = jsFile.SHA256(data)
     ini_string = str(vhashing)
     scale = 16
     res = bin(int(ini_string, scale)).zfill(8)
     batasCharBin = difficulty+3
     jmlNol = difficulty
     strbin = str(res[3:batasCharBin])
     ripit = '0'*jmlNol
     if strbin == ripit:
         loop = False
```