GEOSPATIAL RETRIEVAL HOTEL SYARI'AH BERBASIS PRACTICAL BYZANTINE FAULT TOLERANCE BLOCKCHAIN UNTUK MENINGKATKAN KEAMANAN DATA

SKRIPSI



Disusun Oleh:

MOCH. AINUR RAHMAN SURYANA H76216060

PROGRAM STUDI SISTEM INFORMASI FAKULTAS SAINS DAN TEKNOLOGI UNIVERSITAS ISLAM NEGERI SUNAN AMPEL SURABAYA 2020

PERNYATAAN KEASLIAN

Saya yang bertanda tangan dibawah ini,

NAMA

: Moch. Ainur Rahman Suryana

NIM

: H76216060

Program Studi

: Sistem Informasi

Angkatan

: 2016

Menyatakan bahwa saya tidak melakukan plagiat dalam penelitian skripsi saya yang berjudul: "GEOSPATIAL RETRIEVAL HOTEL SYARI'AH BERBASIS PRACTICAL BYZANTINE FAULT TOLERANCE BLOCKCHAIN UNTUK MENINGKATKAN KEAMANAN DATA". Apabila suatu saat nanti terbukti saya melakukan tindakan plagiat, maka saya bersedia menerima sanksi yang telah ditetapkan.

Demikian pernyataan keaslian ini saya buat dengan sebenar-benarnya.

Surabaya, 9 Januari 2021

Yang menyatakan,

(Moch. Ainur Rahman Suryana)

NIM H76216060

LEMBAR PERSETUJUAN PEMBIMBING

JUDUL : GEOSPATIAL RETRIEVAL HOTEL SYARI'AH BERBASIS

PRACTICAL BYZANTINE FAULT TOLERANCE

BLOCKCHAIN UNTUK MENINGKATKAN KEAMANAN

DATA

NAMA : MOCH. AINUR RAHMAN SURYANA

NIM : H76216060

Mahasiswa tersebut telah melakukan proses bimbingan dan dinyatakan layak untuk mengikuti Sidang (Proposal) Skripsi.

Surabaya, 22 Maret 2020

Menyetujui,

Dosen Pembimbing 1

Achmad Teguh Wibowo, MT

NIP. 198810262014031003

Down Pembimbing 2

Muhammad Andik Izuddin, M.T.

NIP. 198403072014031001

PENGESAHAN TIM PENGUJI SKRIPSI

Skripsi Moch. Ainur Rahman Suryana ini telah dipertahankan didepan tim penguji skripsi

di Surabaya, 16 Januari 2021

Mengesahkan,

Dewan Penguji

Penguji I

Aonmad Teguh Wibowo, MT

NIP. 198810262014031003

Penguji II

Mananmad Andik Izuddin, M.T.

NIP. 198403072014031001

Penguji III

Dwi Rolliawati, MT

NID. 197909272014032001

Dosen Penguji IV

Khalid M.Kom

NIP. 197906092014031002

Mengetahui,

Ran Pakottas Sains dan Teknologi

UN Sunan Ampel Surabaya

Evi Fatimatur Rusydiyah, M.Ag

NIP 197312272005012003



KEMENTERIAN AGAMA UNIVERSITAS ISLAM NEGERI SUNAN AMPEL SURABAYA PERPUSTAKAAN

Jl. Jend. A. Yani 117 Surabaya 60237 Telp. 031-8431972 Fax.031-8413300 E-Mail: perpus@uinsby.ac.id

LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademika UIN Sunan Ampel Surabaya, yang bertanda tangan di bawah ini, saya:

Nama	: Moch. Ainur Rahman Suryana					
NIM	: H76216060					
Fakultas/Jurusan	: FAKULTAS SAINS DAN TEKNOLOGI / SISTEM INFORMASI					
E-mail address	: ainurrahms@gmail.com / h76216060@uinsby.ac.id					
UIN Sunan Ampe ■ Sekripsi □ yang berjudul :	gan ilmu pengetahuan, menyetujui untuk memberikan kepada Perpustakaan I Surabaya, Hak Bebas Royalti Non-Eksklusif atas karya ilmiah : Tesis Desertasi Lain-lain () ETRIEVAL HOTEL SYARI'AH BERBASIS PRACTICAL BYZANTINE					
FAULT TOLERA	4NCE UNTUK MENINGKATKAN KEAMANAN DATA					
Perpustakaan UII mengelolanya di menampilkan/men akademis tanpa p	yang diperlukan (bila ada). Dengan Hak Bebas Royalti Non-Ekslusif ini N Sunan Ampel Surabaya berhak menyimpan, mengalih-media/format-kan, alam bentuk pangkalan data (database), mendistribusikannya, dan mpublikasikannya di Internet atau media lain secara fulltext untuk kepentingan erlu meminta ijin dari saya selama tetap mencantumkan nama saya sebagai lan atau penerbit yang bersangkutan.					
	tuk menanggung secara pribadi, tanpa melibatkan pihak Perpustakaan UIN abaya, segala bentuk tuntutan hukum yang timbul atas pelanggaran Hak Cipta saya ini.					
Demikian pernyat	aan ini yang saya buat dengan sebenarnya.					

Surabaya, 9 Januari 2021

(Moch. Ainur Rahman Suryana)

nama terang dan tanda tangan

ABSTRAK

GEOSPATIAL RETRIEVAL HOTEL SYARI'AH BERBASIS PRACTICAL BYZANTINE FAULT TOLERANCE BLOCKCHAIN UNTUK MENINGKATKAN KEAMANAN DATA

Oleh:

Moch. Ainur Rahman Suryana

Upaya penyerangan dunia digital (cybercrime) kini semakin canggih dan mematikan, teknik penetrasi, dan SQL Injection merupakan beberapa contoh bentuk penyerangan data pada dunia digital. Untuk mencegah terjadinya hal tersebut perlu memperhatikan aspek fundamental dari keamanan data serta keandalan dalam penyimpanan sebuah data. *Blockchain* merupakan teknologi modern dengan menerapkan integritas data serta dalam pengamanan sebuah data. Penelitian ini mengembangkan sebuah aplikasi berbasis android yang menggunakan 4 node practical byzantine fault tolerance (PBFT) blockchain sebagai database untuk pengamanan data dengan memanfaatkan nilai geospatial retrieval pada hotel syari'ah. Hasil penelitian menunjukkan bahwa pengujian test case scenario pada sistem operasi android 7, 8 dan 9 berjalan sesuai dengan test case yang diberikan, pengujian konsensus PBFT sesuai dengan skenario yang telah dibuat. Pengujian network latency menujukkan jaringan 4G memiliki 107% lebih cepat pada jaringan 3G pada saat melakukan enkripsi dan 32.9% lebih cepat saat melakukan dekripsi pada blockchain. Pengujian integrasi dilakukan tanpa PBFT menghasilkan 50% kesuksesan dan 50% kegagalan setelah menggunakan PBFT blockchain presentase kesuksesan meningkat 50% menjadi 100%.

Kata kunci: *Cybercrime*, Geospatial Retrieval, Practical Byzantine Fault Tolerance, *Blockchain*

ABSTRACT

GEOSPATIAL RETRIEVAL HOTEL SHARIA BASED PRACTICAL BYZANTINE FAULT TOLERANCE BLOCKCHAIN TO IMPROVE DATA SECURITY

By:

Moch. Ainur Rahman Suryana

Efforts to attack the digital world (cybercrime) are now increasingly sophisticated and deadly. Penetration techniques and SQL Injection are some examples of forms of data attack on the digital world. It is necessary to pay attention to data security and management's fundamental aspects of storing data. Blockchain is a modern technology by implementing data integrity and securing data. This study developed an android-based application that uses four practical Byzantine fault tolerance (PBFT) blockchain nodes as a database for data security by utilizing the value of geospatial retrieval in shari'ah hotels. The results showed that the test case scenarios on the android operating system 7, 8, and 9 run according to the given test cases; PBFT consensus testing according to the design made. Network latency testing showed the 4G network was 107% faster when encrypting 3G networks and 32.9% more quickly when decrypting the blockchain. Integration testing performed without PBFT resulted in 50% success and 50% failure after using PBFT blockchain, which increased 50% to 100%.

Keywords: Cybercrime, Geospatial Retrieval, Practical Byzantine Fault Tolerance, Blockchain

DAFTAR ISI

PERNY	YATAAN KEASLIAN	i
LEMB	AR PERSETUJUAN PEMBIMBING	ii
PENGI	ESAHAN TIM PENGUJI SKRIPSIError! Bookmark	not defined.
LEMB	AR PERNYATAAN PUBLIKASI KARYA ILMIAH	iv
MOTT	O	v
KATA	PENGANTAR	vi
ABSTF	RAK	viii
ABSTF	RACT	ix
DAFT	AR TABEL	xii
DAFT	AR GAMBAR	xiii
BAB 1	PENDAHULUAN	15
1.1	Latar Belakang	15
1.2	Perumusan Masalah	
1.3	Batasan Masalah	17
1.4	Tujuan Penelitian	18
1.5	Manfaat Penelitian	
BAB 2	TINJAUAN PUSTAKA	19
2.1	Tinjauan penelitian terdahulu	19
2.2	Keamanan Informasi	21
2.3	Blockchain	
2.4	Smart Contract	
2.5	Geospatial Retrieval	26
2.6	Geographic Information Retrieval (GIR)	26
2.7	Algoritma JavaScript Object Notation Diff (JSON-DIFF)	26
2.8	AES Encryption	27
2.9	Hashing SHA256	29
2.10	Algoritma Midpoint Circle	30
2.11	Hotel Syari'ah	31
2.12	Smartphone	31
2.13	Database	32
2.14	Docker	32
2.15	Javascript	33

2.16	Framework Javascript React Native	. 33
2.17	Node.Js	. 33
2.18	Unified Modelling Language (UML)	. 34
2.19	Network Latency	. 35
2.20	Test Case Scenario	. 35
2.21	Integrasi Keilmuan	. 36
BAB 3	METODOLOGI PENELITIAN	. 38
3.1	Tahapan Penelitian	. 38
BAB 4	HASIL DAN PEMBAHASAN	. 47
4.1	Hasil Analisis Kebutuhan Sistem	. 47
4.2	Desain Arsitektur Sistem	. 48
4.3	UML Use Case Diagram	. 49
4.4	UML Class Diagram	. 50
4.5	UML Sequence Diagram	. 52
4.6	UML Activity Diagram	. 57
4.7	Desain <i>User Interface</i> (UI) Aplikasi	
4.8	Implementasi	. 65
4.9	Evaluasi Aplikasi	. 69
BAB 5	PENUTUP	. 93
5.1	Kesimpulan	
5.2	Saran	. 94
DAFT	AR PUSTAKA	95

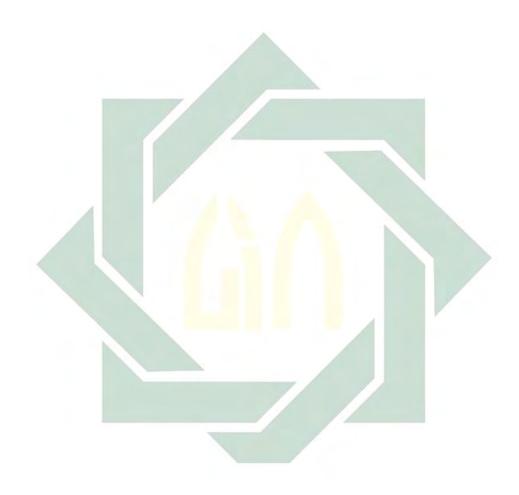
DAFTAR TABEL

Tabel 2.1 Perbandingan Konsensus Protokol (Zhang & Lee, 2019)	23
Tabel 3.1 Skenario Pengujian Konsensus PBFT	41
Tabel 3.2 Test Case Scenario Pengujian Blackbox Aplikasi	41
Tabel 3.3 Scenario Pengujian Integritas Data tanpa PBFT blockchain	44
Tabel 3.4 Scenario Pengujian Integritas Data dengan PBFT blockchain	45
Tabel 4.1 Pengujian Blackbox Aplikasi Android 7	70
Tabel 4.2 Pengujian Blackbox Aplikasi Android 8	73
Tabel 4.3 Pengujian Blackbox Aplikasi Android 9	77
Tabel 4.4 Pengujian Konsensus PBFT	81
Tabel 4.5 Rata-rata Network Latency (seconds)	
Tabel 4.6 Raw Data <i>Hasil Pengujian</i> Mid-Point Validation	
Tabel 4.7 Raw Data Hasil Pengujian Mid-Point Validation	
Tabel 4.8 Raw Data Hasil Pengujian Mid-Point Validation	
Tabel 4.9 Raw Data Hasil Pengujian Mid-Point Validation	85
Tabel 4.10 Hasil Akhir Pengujian Mid-Point Validation	86
Tabel 4.11 Contoh Data pen <mark>gu</mark> jia <mark>n integritas</mark> tanpa PBFT blockchain	86
Tabel 4.12 Contoh Data pengujian integritas dengan PBFT blockchain	88
Tabel 4.13 Hasil Pengujian Integritas data tanpa PBFT Blockchain	90
Tabel 4.14 Hasil Pengujian Integritas Data Menggunakan PBFT	90
Tabel 4.15 Tabel Perbandingan Presentase Pengujian Integritas Data	91

DAFTAR GAMBAR

Gambar 2.1 Ilustrasi Blockchain (Efanov & Roschin, 2018)	23
Gambar 2.2 Proses PBFT (Zhang & Lee, 2019)	25
Gambar 2.3 Substitusi Byte (Wadehra et al., 2018)	
Gambar 2.4 Shift Row (Wadehra et al., 2018)	28
Gambar 2.5 Mix Columns (Wadehra et al., 2018)	28
Gambar 2.6 Add Round Key (Wadehra et al., 2018)	29
Gambar 2.7 Dekripsi Algoritma AES (Wadehra et al., 2018)	29
Gambar 2.8 SHA-256 hash function (Harris et al., 2015)	30
Gambar 2.9 Docker Arsitektur	32
Gambar 2.10 Struktural UML Diagram (Fan, 2015)	35
Gambar 3.1 Tahapan Penelitian	38
Gambar 4.1 Desain Arsistektur Sistem	48
Gambar 4.4 UML Use Case Diagram	50
Gambar 4.5 UML Class Diag <mark>ram</mark>	50
Gambar 4.6 UML Sequence Diagram Show Maps	53
Gambar 4.7 UML Sequence Diagram Register User	54
Gambar 4.8 UML Sequence Diagram Login User	55
Gambar 4.9 UML Sequence Diagram Surveyor	56
Gambar 4.10 UML Activity Diagram Show Maps	57
Gambar 4.11 UML Activity Registration	58
Gambar 4.12 UML Activity Login	59
Gambar 4.13 UML Activity Surveyor	
Gambar 4.14 Service GPS	60
Gambar 4.15 Splash Screen	61
Gambar 4.16 Maps Screen	61
Gambar 4.17 Search Screen	62
Gambar 4.18 Login Screen	63
Gambar 4.19 Register Screen	63
Gambar 4.20 Surveyor Outside Area	64
Gambar 4.21 Form Surveyor	65
Gambar 4.22 Contoh Data Hashing-SHA256	68

Gambar 4.23 Enkripsi Data Network Latency 3G dan 4G	. 81
Gambar 4.24 Dekripsi Data Network Latency 3G dan 4G	. 82



BAB 1

PENDAHULUAN

1.1 Latar Belakang

Upaya penyerangan dunia digital pada sistem informasi baru-baru ini berkembang menjadi canggih dan mematikan (Stiawan, Idris, Abdullah, Aljaber, & Budiarto, 2017). Seiring dengan berkembangnya teknologi, berbagai macam jenis *cybercrime* terus berkembang bahkan lebih kompleks dan terkontrol dengan baik hingga tidak dapat dilacak (Bendovschi, 2015). Salah satunya yaitu teknik penetrasi, saat ini teknik tersebut mulai berkembang dengan cara menggunakan kerentananan yang ada pada *software* pihak ketiga untuk memperoleh data berharga dari sistem. (Gamundani & Nekare, 2018). Teknik penyerangan lainnya yaitu *virus ransomware*, kemampuannya dapat mengacak *file* sistem dan mereplikasi data dengan cepat pada komputer yang menggunakan jaringan yang (Gamundani & Nekare, 2018). Untuk menghindari hal tersebut, perlu diperhatikan mengenai aspek fundamental dari keamanan dan keandalan penyimpanan dalam Integritas Data (Sivathanu, Wright, & Zadok, 2005)

Salah satu teknologi modern yang menerapkan keamanan integritas data yaitu teknologi *Blockchain* (Efanov & Roschin, 2018). *Blockhain* merupakan teknologi yang menggabungkan keunggulan jaringan *peer-to-peer* dan kriptografi untuk memastikan validitas data dan keaslian data yang terhubung didalam jaringan *Blockchain* (Wibowo, Sulistyono, & Hariadi, 2020). Dengan cara seperti ini informasi yang disimpan didalam blok adalah permanen, tidak dapat diubah atau dimodifikasi. Untuk melakukan perubahan, semua *node* yang terhubung harus setuju untuk diperbaharui (Ali & Afzal, 2018).

Salah satu metode yang dimiliki *blockchain* yakni *Practical Byzantine Fault Tolerance* (PBFT). PBFT merupakan mekanisme konsensus representatif dan memiliki kompleksitas algoritma yang rendah sehingga mudah untuk diterapkan dan disebut sebagai protokol konsensus absoulut-finalitas (Zhang & Lee, 2019). PBFT digunakan dalam mekanisme konsensus *blockchain* karena memiliki keandalan serta keamanan yang tinggi (Chen, Zhang, & Shangguan, 2019).

Geospatial merupakan sebuah nilai ruang kebumian yang menunjukkan suatu nilai letak koordinat dan posisi suatu objek. Sedangkan information retrieval merupakan ilmu komputer yang berhubungan mengenai representasi, penyimpanan dan akses informasi (James & Kannan, 2017). Sebuah sistem yang memanfaatkan nilai koordinat sebagai basis informasi disebut Geographic Information Retrieval (GIR). GIR merupakan sistem informasi geografis yang memanfaatkan data ruang bumi yang disimpan melalui database yang terintegrasi dengan aplikasi pada sebuah perangkat salah satunya smartphone.

Smartphone merupakan perangkat populer yang dapat mengetahui informasi mengenai berita, makanan, kesehatan, wisata dan lain sebagainya (Beki Subaeki, 2016). Pariwisata sangat erat kaitannya dengan bisnis akomodasi terutama bisnis hotel (Mujahidin, 2018). Salah satu konsep hotel dengan memperhatikan nilai keislaman didalamnya yaitu hotel syari'ah (Battour & Ismail, 2016). Hotel syariah merupakan strategi pemasaran yang kuat untuk menarik wisatawan muslim. Meningkatnya wisatawan muslim merupakan peluang serta tantangan bagi bisnis hotel syariah bagi kota-kota besar yang ada di Indonesia salah satunya adalah kota Surabaya (Satriana & Faridah, 2018).

Kota Surabaya merupakan kota terbesar no 2 di indonesia yang memiliki total penduduk sebayak 3,095,026 jiwa dan sebagian besar penduduk kota Surabaya menganut agama islam (BPS-Statistics of Surabaya Municipality, 2018). Di sejumlah tempat di kota Surabaya telah tersebar beberapa hotel syariah dengan lokasi yang strategis serta didukung dengan banyaknya pengunjung wisatawan muslim. Informasi yang dapat dimanfaatkan bagi wisatawan muslim untuk mempermudah pencarian hotel syariah adalah alamat, nama hotel serta no. handphone.

Dengan terinsipirasi dari penelitian-penelitian yang ada diatas, penelitian ini menerapkan *smartphone* sebagai perangkat media penyimpanan informasi dan membangun aplikasi berbasis GIR dengan menggunakan peta digital berbasis *GoogleMaps* yang dibuat dengan tujuan untuk mengetahui informasi spesifik mengenai lokasi geografis hotel syariah. Kemudian informasi mengenai hotel syari'ah tersebut disimpan dalam sebuah *database* terdistribusi PBFT

blockchain yang terhubung dengan aplikasi. Kemudian pada PBFT blockchain menggunakan smart-contract midpoint circle untuk melakukan validasi lokasi user. Setelah itu jaringan PBFT blockchain menggunakan Algoritma JavaScript Object Notation (JSON) Diff untuk mencocokkan beberapa data object yang datanya diambil dari server yang terhubung dalam jaringan PBFT blockchain, maka perlu dilakukan penelitian dengan judul "GEOSPATIAL RETRIEVAL HOTEL SYARI'AH BERBASIS PRACTICAL BYZANTINE FAULT TOLERANCE BLOCKCHAIN UNTUK MENINGKATKAN KEAMANAN DATA". sehingga hasil dari penelitian ini diharapkan dapat meningkatkan keamanan data serta validitas data saat membangun sebuah aplikasi berbasis GIR yang memiliki informasi mengenai hotel syari'ah.

1.2 Perumusan Masalah

Berdasarkan pemaparan di latar belakang, maka rumusan masalah dari Pemanfaatan Nilai *Geospatial* Retrieval Hotel Syari'ah pada Teknologi *Blockchain* PBFT adalah:

- a. Bagaimana hasil evaluasi pengujian *blackbox testing* dengan *test case* scenario pada aplikasi hotel syari'ah berbasis android?
- b. Bagaimana hasil evaluasi konsensus metode PBFT *blockchain*, hasil pengujian *smart-contract midpoint circle* dan pengaruh *network latency* pada 3G dan 4G pada *blockchain*?
- c. Bagaimana hasil presentase keamanan data dengan pengujian integritas data menggunakan Algoritma JSON-DIFF didalam *blockchain*?

1.3 Batasan Masalah

Untuk menjaga penelitian di luar dari pada cakupan maka dilakukannya upaya pembatasan terhadap masalah yang akan diangkat, sebagai berikut.

- a. Data hotel yang digunakan pada aplikasi merupakan hotel syari'ah yang ada dikota Surabaya berdasarkan sertifikasi yang telah ditetapkan oleh perundang-undangan.
- b. Lokasi atau Hotel Syari'ah yang telah ditentukan berdasarkan *longitude* dan *latitude* yang diambil dari penyedia layanan data peta dari *GoogleMaps*.
- c. Node *blockchain* yang akan digunakan adalah *virtual machine computer* docker dengan 4 *container*.

d. Aplikasi yang dibuat berbasis *mobile* yang hanya akan dijalankan pada perangkat *Android* (versi 7, 8, 9).

1.4 Tujuan Penelitian

Penelitian ini memiliki tujuan sebagai berikut.

- a. Membuat aplikasi pemetaan Hotel Syari'ah berbasis perangkat mobile yang dapat diakses melalui sistem operasi android 7, 8 dan 9.
- b. Mengetahui hasil kesesuaian teori pengujian konsensus PBFT dengan menggunakan 4 node. Setelah itu mengetahui efektivitas *smart contract* menggunakan algoritma *midpoint circle* terhadap data yang diinputkan oleh *surveyor*. Dan menilai efektivitas *network latency* jaringan 3G dan 4G pada *blockchain*.
- c. Melakukan pengecekan terhadap struktur data *hash* pada *block* (kumpulan data) PBFT yang saling terhubung dalam satu jaringan blockchain

1.5 Manfaat Penelitian

Manfaat yang dapat diambil dari penelitian berdampak terhadap pihak dan dan lingkungan yang menjadi obyek penelitian, termasuk didalamnya pada sisi akademik dan aplikatif.

- a. Akademik
- 1. Mengimplementasikan ilmu yang telah didapat untuk menyelesaikan permasalahan yang dihadapi oleh masyarakat
- 2. Menambah pengetahuan baru dalam pengembangan metode yang diterapkan kedalam aplikasi yang dibangun.
- 3. Sebagai referensi ilmu pengetahuan dalam mengembangkan sistem pemetaan hotel syari'ah menggunakan PBFT *blockchain*.
- b. Aplikatif
- Meningkatkan keamanan data dengan mengkombinasikan PBFT blockchain, teknik hash SHA-256, AES enkripsi dan Algoritma JSON-DIFF string-match.
- 2. Menampilkan informasi meliputi alamat, nama hotel dan kontak yang dapat dihubungi oleh pengguna aplikasi.
- 3. Mendukung gerakan pariwisata halal di kota Surabaya.

BAB 2 TINJAUAN PUSTAKA

Bab dua membahas penelitian yang relevan serta dasar teori. Penelitian relevan berisi jurnal-jurnal penelitian pendukung sebelumnya. Berbagai penelitian tentang Hotel Syariah, GIR, PBFT *blockchain*, Algoritma SHA256, dan Algoritma JSON-DIFF.

2.1 Tinjauan penelitian terdahulu

"An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends", penelitian ini membahas mengenai teknologi blockchain sebagai teknologi pencatatan data yang terdesentralisasi serta membandingkan beberapa algoritma konsensus yang digunakan didalam blockchain. Penelitian ini menghasilkan bahwa teknologi blockchain menunjukkan potensi untuk mengubah industri tradisional menjadi industri yang terdesentralisasi (Zibin Zheng, 2017).

Penelitian selanjutnya dengan judul "Ensuring Data Integrity in Storage: Techniques and Applications", penelitian ini membahas kefektiftasan penggunaan teknik integritas data serta penyebab pelanggaran integritas dalam penyimpanan data dan menampilkan tingkat efektifitas mengenai beberapa teknik integritas data. Dengan memperhatikan integritas data sebuah sistem dapat berjalan secara efisien dan tanpa hambatan (Gopalan, 2005).

Penelitian dengan judul Securing Smart Cities Using Blockchain Technology yang berfokus pada masalah keamanan informasi dan privasi. Penelitian ini mengusulkan framework kemanan yang mengintegrasikan teknologi blockchain dengan perangkat pintar yang menyediakan platform komunikasi yang aman di Smart City (Biswas & Muthukkumarasamy, 2017).

Penelitian selanjutnya dengan judul *A Comprehensive Study in Smart City using Blockchain Techology* yang berfokus pada keamanan IoT dengan server terpusat rentan terhadap serangan perangkat lunak dan sistem dapat dengan mudah diretas. Maka penelitian ini mengusulkan membangun kota cerdas berbasis IoT menggunakan *blockchain* dengan *database* terdistribusi yang dapat dipercaya (Raja Anita, 2018).

Penelitian berikutnya yang berjudul "Performance Modelling of PBFT Consensus Process for Permissioned Blockchain Network (Hyperledger

Fabric)" yang ditulis oleh (Sukhwani, Mart, Chang, Trivedi, & Rindos, 2017). Penelitian ini bertujuan untuk menyelidiki proses konsensus menggunakan PBFT sebagai metode *blockchain* dalam penelitian tersebut. Kemudian menggunakan *Stochastic Reward Nets* (SRN) untuk menghitung waktu ratarata menyelesaikan konsensus hingga 100 *peers*.

Penelitian selanjutnya yang berjudul "Implementasi Algoritma SHA-256 Menggunakan Protokol MQTT pada Budidaya Ikan Hias" yang ditulis oleh (Anugrah, Hannats, Ichsan, & Kusyanti, 2019) Algoritma SHA-256 tahan terhadap brute force dan collision attack. Selain itu performa penggunaan memori pada saat mengimplementasikan HMAC-SHA256 sebesar 0.0078 MB pada publisher dan 0.0082MB pada subscriber. Ini membuktikan bahwa Algoritma SHA256 cukup efektif untuk digunakan selain tahan terhadap serangan digital juga memiliki konsumsi memori yang minim.

Penelitian yang membahas mengenai GIR yang berjudul "Geographic Information Retrieval Method for Geography Mark-Up Language Data" Penelitian ini merancang dan menyederhakan proses pengambilan data informasi geografis, dengan menggunakan metode Geographic Information Retrieval yang menunjukkan tingkat akurasi dan efisiensi dari penggunaan metode tersebut (Fang & Zhang, 2018).

Penelitian berikutnya mengenai "Mapping the Locational Pattern of Hotels in Akure, Ondo State." yang ditulis oleh (Adeyemi, 2013) dengan hasil data yang berhasil diproses menghasilkan data geospatial yang menggunakan ILWIS (3.3) dan ArcGIS (9.3). Dengan adanya penelitian tersebut membuktikan bahwa Geographic Information System yang dapat digunakan memunculkan informasi mengenai lokasi hotel serta memunculkan perencanaan dan pengelolaan yang tepat.

Penelitian selanjutnya dengan judul Wisata Halal: Perkembangan, Peluang dan Tantangan membahas tentang meningkatnya jumlah wisatawan muslim setiap tahunnya merupakan peluang serta tantangan bagi pariwisata halal, dari sudut pandang prinsip wisata halal banyak negara mayoritas muslim atau nonmuslim mencoba menciptakan suasanan yang ramah muslim (Satriana dan Faridah, 2018).

Penelitian selanjutnya mengenai hotel syariah dengan judul "Sharia Hotels in Indonesia: Concept and Potential Analysis". Penelitian ini membahas mengenai konsep dan analisa hotel syariah potensial di Indonesia. Penelitian ini menyimpulkan bahwa hotel syariah menjadi bagian dari industri pariwisata halal serta menjadi bagian industri pariwisata halal yang mampu memenuhi kebutuhan masyarakat muslim di Indonesia di masa depan (Muhammad, 2018).

Dari beberapa penelitian yang dilakukan, belum ada penelitian yang menginstegrasikan PBFT *blockchain* dengan menggunakan SHA-256 sebagai teknik *hashing* terhadap *prevhash* dalam *blockchain* serta proses enkripsi dengan menggunakan Algoritma AES terhadap data hotel syari'ah dan menggunakan Algoritma JSON-DIFF yang kemudian diimplementasikan kedalam sebuah aplikasi *smartphone* berbasis GIR.

2.2 Keamanan Informasi

Perkembangan internet yang kian hari meningkat begitu pesat membuat informasi semakin hari semakin mudah untuk didapatkan dan diakses. Dengan mudahnya akses informasi tersebut membuat maraknya terjadi *cybercrime*, untuk mengatasi hal tersebut perlu dilakukannya pengamanan data dengan cara memperhatikan *confidentiality*, *integrity*, dan *availability*.

Confidentiality merupakan cara agar informasi dapat diakses oleh orang yang berwenang, misalnya informasi pribadi karyawan dalam suatu organisasi atau sebuah aplikasi perusahaan yang hanya digunakan oleh karyawan dari perusahaan tersebut. Dengan adanya hal ini perusahaan dapat mempertahankan reputasi dan bisnisnya.

Integrity merupakan sebuah upaya penganan informasi dengan cara memberikan izin terhadap setiap data yang akan dilakukan modifikasi, apabila secara tiba-tiba informasi atau data berubah orang yang melakukannya adalah orang yang tidak memiliki wewenang atau hak untuk mengakses informasi tersebut.

Availability merupakan ketersediaan informasi apabila diakses oleh pengguna dapat selalu tersedia kapan pun dan dimanapun. Ini berarti bahwa sistem komputasi yang menyimpan dan memproses informasi harus menyimpan, memproses informasi dan mengamankan informasi agar informasi dapat berfungsi dengan baik (Mir, Dar, & Beig, 2011)

2.3 Blockchain

Secara sederhana teknologi *blockchain* dapat digambarkan sebagai sebuah basis data yang terdistribusi dan mencatat transaksi yang dibagikan kepada orang-orang yang tergabung didalam sebuah jaringan basis data terdistribusi (Efanov & Roschin, 2018) Setiap transaksi yang terjadi harus sesuai dengan konsensus yang telah disepakati dalam jaringan basis data terdistribusi, yang akhirnya membuat kemungkinan terjadi kecurangan sangat minim.

Dari awal munculnya *blockchain* sampai sekarang, *blockchain* mengalami evolusi yang cukup berarti meskipun secara harafiah, *blockchain* adalah sebuah kumpulan block yang saling berhubungan (ter-rantai) dan berisi informasi mengenai transaksi yang terjadi. Yang menjadi kunci di dalam teknologi *blockchain* adalah kemampuan untuk melacak kembali di dalam jaringan basis data terdistribusi. Secara sederhana, perkembangan teknologi *blockchain* sudah mencapai 3 fase, yaitu *blockchain* 1.0 yang awalnya muncul sebagai tonggak mata uang digital, kemudian berkembang menjadi *blockchain* 2.0 sebagai bentuk perkembangan selanjutnya pada bidang ekonomi digital, dan yang terakhir adalah *blockchain* 3.0 sebagai bentuk evolusi dari ekonomi digital ke dalam bentuk perhimpunan atau masyarakat digital (Efanov & Roschin, 2018).

Pada fase *blockchain* 1.0, teknologi *blockchain* muncul sebagai pelaku dibalik layar dan sebagai generasi perdana dari mata uang digital, meliputi platform teknologi seperti menambang (mining), hashing, dan buku besar umum. Pada fase ini, beberapa contoh keuntungan menggunakan *blockchain* seperti pengurangan biaya transaksi untuk pembelian berbasis on-line, menawarkan anonimitas yang lebih baik dari pada kartu kredit dan perlindungan dari inflasi karena hadirnya teknologi terdesentralisasi. Fase berikutnya, yaitu *blockchain* 2.0 yang lebih dikenal dengan fase eknomi digital membuat sebuah revolusi di dalam dunia finansial dengan hadirnya banyak aplikasi finansial yang menawarkan kemudahan untuk membayar, melakukan transfer, dan melakukan transaksi bisnis. Dalam fase inilah muncul yang namanya smart contract, dimana smart contract merupakan sebuah program yang dapat

memastikan bahwa transaksi yang terjadi sudah sesuai dengan perjanjian atau peraturan yang telah disepakati bersama di dalam jaringan basis data terdistribusi. Fase yang terakhir adalah *blockchain* 3.0 dan lebih dikenal dengan fase masyarakat digital. Pada fase ini, yang terlibat tidak lagi hanya dari dunia bisnis, tetapi dari bidang lain sudah mulai memanfaatkan teknologi *blockchain* seperti bidang kesehatan, pendidikan, pemerintahan, komunikasi, ilmu pengetahuan dan lainnya. Tahap ini, memunculkan konsep Smart City dan IoT sebagai platform bisnis yang baru (Efanov & Roschin, 2018).

Blockchain selain sebagai sebuah kumpulan block yang terhubung dengan mencatat tanda digital / hash dari block sebelumnya ditunjukkan pada gambar 1 dibawah ini.



Gambar 2.1 Ilustrasi *Blockchain* (Efanov & Roschin, 2018)

Dari gambar 2.1 dapat dikatakan bahwa *block* dalam *blockchain* memiliki 3 komponen penting, yaitu data, previous hash dari *block* sebelumnya, dan hash beserta waktu saat *block* dibuat untuk *block* yang bersangkutan. Didalam *blockchain* ada sebuah perjanjian digital yang harus di sepakati oleh berbagai *node* yang saling terdistribusi. Ada beberapa metode algoritma konsensus dalam *blockchain* yang telah ditunjukkan pada Tabel 2.1.

Tabel 2.1 Perbandingan Konsensus Protokol (Zhang & Lee, 2019)

Property	PoW	PoS	DPoS	PBFT
Type	Probabilistic-	Probabilistic-	Probabilistic-	Absolute-
	finality	finality	finality	Finaly
Fault	50%	50%	50%	33%
Tolerance	3070	3070	3070	3370
Power	Large	Less	Less	Negligible
Consumption	Large	LCSS	LCSS	regugiole

Scalability	Good	Good	Good	Bad
Application	Public	Public	Public	Permissioned

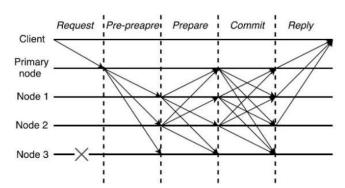
PBFT memiliki kelebihan dalam *fault tolerance*, konsumsi tenaga dan termasuk konsensus yang memiliki protokol konsensus absoulut-finalitas. Mekanisme yang digunakan di dalam *blockchain* adalah memastikan keamanan *blockchain* dalam beberapa hal, yaitu:

2.3.1 Mekanisme tanda digital / hash

Tanda digital/hash adalah sebuah hasil dari pemanfaatan teknik hash. Teknik hash merupakan salah satu teknik kriptografi. Hash di dalam kriptografi merupakan algoritma matematika yang memetakan data acak ke dalam bentuk string (hash) dan dirancang menjadi sebuah fungsi satu arah, yaitu fungsi yang hasilnya tidak dapat dikembalikan ke bentuk asalnya. Jadi dapat dikatakan bahwa bila sebuah informasi, katakanlah sebuah baris informasi yang dibuat nilai hashnya melalui fungsi hash kriptografi, maka tanda digital yang muncul adalah benar-benar unik milik baris informasi tersebut dan tidak dapat dibalikkan ke bentuk asalnya. Bila baris informasi yang sudah dibuat tanda digitalnya mengalami perubahan informasi, maka nilai hash kriptografinya juga akan berubah.

2.3.2 Practical Byzantine Fault Tolerance (PBFT)

PBFT merupakan mekanisme protokol toleransi bizantium yang menggunakan algoritma yang sederhana tetapi memiliki kepraktisan yang tinggi dengan sistem *database* terdistribusi (Castro & Liskov, 1999). PBFT merupakan mekanisme untuk mendistribusikan catatan transaksi kepada orangorang atau user yang tergabung dalam jaringan *blockchain* (Zhang & Lee, 2019). PBFT terdiri dari lima tahap yaitu:



Gambar 2.2 Proses PBFT (Zhang & Lee, 2019)

a. Request (Meminta)

Pada tahap ini data yang baru akan melakukan *request* terhadap *primary node*. Jika *primary node* aktif maka akan melakukan proses selanjutnya yaitu *Pre-Prepare*. Apabila *primary node* tidak aktif maka *request* yang dilakukan gagal.

b. Pre-Prepare (Pra-Persiapan)

Setelah melakukan *request*, *primary node* akan melakukan pengecekan terhadap *node* yang terhubung jaringan *blockchain* dengan melakukan ping pada semua *node*. Apabila hasilnya 3f proses selanjutnya bisa dilakukan.

c. *Prepare* (Persiapan)

Selanjutnya dilakukan pengiriman data nama, koordinat *longitude* dan *latitude*, alamat, dan nomor hotel. Proses ini akan berjalan apabila mendapat *reply* minimum dan maksimum masing-masing 3f(3f-1) dan $3f^2$.

d. *Commit* (Menjalankan)

Proses ini berjalan apabila *node* menerima *message minimum* (3f - f + 1)(3f - 1) dan *message maximum* 3f(3f + 1) dari tahap sebelumnya proses ini menambahkan blok baru ke *blockchain* setelah proses validasi melalui *smart-contract*.

e. Reply (Balasan)

User akan menerima konfirmasi bahwa blockchain telah berhasil di proses yang memiliki message minimum 3f dan message maximum 3f+1

2.4 Smart Contract

Blockchain memiliki perjanjian digital yang dibuat berdasarkan kumpulan data yang tergabung dalam jaringan disebut dengan smart-contract. Proses pertama yang dilakukan yaitu melakukan validasi data yang akan dimasukkan. PBFT akan melakukan pengecekan pada setiap node yang terhubung. Selanjutnya dilakukan validasi kecocokan data yang diterima oleh user dengan bentuk format data JSON menggunakan Algoritma JSON-DIFF. Data yang dibutuhkan ketika melakukan proses smart-contract adalah longitude, latitude, nama, alamat, dan no. hp hotel syari'ah yang dimasukkan kedalam aplikasi menggunakan algoritma midpoint-circle untuk mendeteksi user telah berada

pada jangkauan radius yang ditentukan oleh aplikasi. Kemudian batas maksimal radius dari lokasi yang ditentukan yakni 50m dari hotel syari'ah yang telah ditentukan.

2.5 Geospatial Retrieval

Ruang bumi memiliki sejumlah bidang dan beberapa data yang dapat dihitung dan juga dapat dianalisa. Dalam pemanfaatanya, banyak teknologi yang menerapkan data ruang tersebut kedalam sebuah aplikasi *modern* salah satu contoh adalah penerapan layanan *Google Maps*. Data ruang disebut dengan geospasial yang memiliki referensi keruangan serta koordinat suatu tempat atau geografi (Nanin Trianawati Sugito, ST., MT. dan DRS. Dede Sugandi, 2009) Selain itu data spasial dapat di integrasikan kedalam sebuah sistem geografis yang dapat dianalisa untuk mencari dan mengambil informasi (Han, Yang, Di, & Member, 2014)

2.6 Geographic Information Retrieval (GIR)

Bentuk penerapan data spasial kedalam sebuah sistem biasa disebut dengan Geographic Information Retrieval (GIR). GIR merupakan memprosesan data dari sistem informasi geografis yang kemudian data tersebut ditampilkan melalui aplikasi yang akan dibuat. Penggunaan GIR ini telah digunakan pada berbagai bidang mulai dari bidang kesehatan, sosial dan lain sebagainya. Pada penelitian berjudul "Geospatial blockchain: promises, challenges and scenarios in health and healthcare" memanfaatkan kombinasi teknologi GIR dan blockchain pada bidang kesehatan (Boulos, Wilson, & Clauson, 2018)

2.7 Algoritma JavaScript Object Notation Diff (JSON-DIFF)

Algoritma ini digunakan untuk membandingkan *value* yang didapatkan dari jaringan *blockchain* yang *value* nya berbentuk *object*. Setelah melakukan pembandingan maka akan mengeluarkan *output* berupa *true* atau *false*.

Pseudocode Algoritma JSON-DIFF

```
1.  n = {...}
2.  m = {...}
3.  compareJSON = (n,m) {
4.  obj = {}
5.  for( i in n ) {
6.   if(!n.hasOwnProperty(i) || m[i] !== obj1[i] ) {
7.   ret[i] = m[i]
8.  }
9.  }
10.  return obj
11. }
```

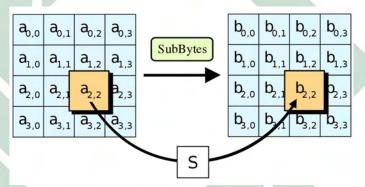
Value tersebut harus memenuhi syarat pada blockchain PBFT yaitu dengan nilai maksimum 3f + 1. Apabila pada satu server terjadi perubahan data pada jaringan blockchain tetap dapat diakses sedangkan perubahan data dilakukan pada lebih dari satu server maka jaringan blockchain tidak dapat diakses dan me return nilai false.

2.8 AES Encryption

Proses enkripsi dengan menggunakan Algoritma AES yang didasarkan pada *substitution-permutation network*. Algoritma AES memiliki dua proses utama yaitu proses enkripsi dan proses dekripsi (Wadehra, Goel, & Sengar, 2018). Pada proses enkripsi memiliki 4 tahapan yaitu:

2.8.1 Substitusi Byte

Pada Tahap ini input data diganti dengan *fixed table* (S-Box) yang kemudian disimpan kedalam matriks yang memiliki 4 baris dan 4 kolom.

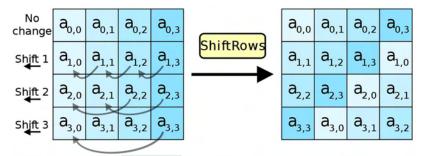


Gambar 2.3 Substitusi Byte (Wadehra et al., 2018)

2.8.2 Shift Row

Setelah melakukan penyimpanan kepada matriks 4x4 dengan dua bagian a dan b lalu dilakukan penggeseran matriks dengan ketentuan sebagai berikut.

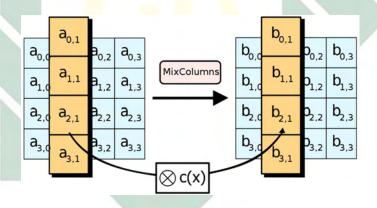
- a. Baris pertama tidak digeser
- b. Baris kedua digeser kekiri satu posisi
- c. Baris ketiga digeser kekiri dua posisi
- d. Baris keempat digeser kekiri tiga posisi



Gambar 2.4 Shift Row (Wadehra et al., 2018)

2.8.3 Mix Columns

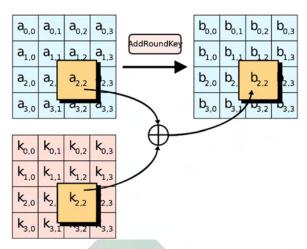
Kemudian keempat kolom matriks tersebut di kombinasikan dengan menggunakan fungsi khusus yang kemudian memiliki hasil *output* matriks yang berbeda.



Gambar 2.5 Mix Columns (Wadehra et al., 2018)

2.8.4 Add Round Key

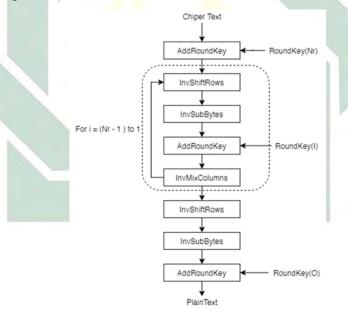
Setelah proses *mix columns* selesai selanjutnya dilakukan penggabungan *subkey* dengan proses yang berulang dengan menggunakan bit XOR.



Gambar 2.6 Add Round Key (Wadehra et al., 2018)

2.8.5 Proses Dekripsi

Proses dekripsi pada Algoritma AES memiliki proses yang sama dengan enkripsi tetapi dimulai pada tahapan *Add Round Key*, sehingga didapatkan ilustrasi sebagai berikut.

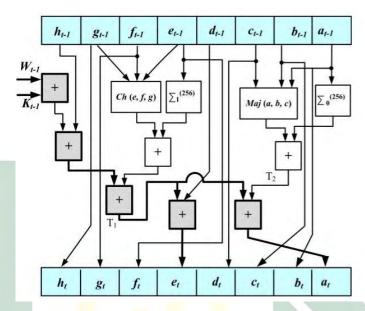


Gambar 2.7 Dekripsi Algoritma AES (Wadehra et al., 2018)

2.9 Hashing SHA256

Data merupakan sekumpulan informasi yang tersimpan secara *digital* dan data terebut dapat diakses secara terbuka oleh siapapun. Karena hal tersebut, perlu membuat sebuah mekanisme pengamanan data yang disebut enkripsi (D Rachmawati, 2018). Ada beberapa teknik untuk mengamankan sebuah data salah satunya fungsi *hash* SHA256. SHA256 menerima *input* pesan hingga 2⁶⁴

bit yang selanjutnya diproses melalui blok dengan ukuran 512 bit. Kemudian setelah *input* berubah menjadi blok, tahap selanjutnya yaitu menambahkan bit bernilai 1 dan 0 pada blok terakhir (Anugrah et al., 2019). Berikut bagaimana Algoritma SHA-256 bekerja.



Gambar 2.8 SHA-256 hash function (Harris et al., 2015)

2.10 Algoritma Midpoint Circle

Salah satu algoritma konvensional yang paling efisien untuk menggambarkan sebuah lingkaran adalah algoritma titik tengah (Lakshmi & Mohan, 2017). Pendefinisian lingkaran dilakukan dengan set titik pada jarak tertentu r dari pusat yang diposisikan di (X_c, Y_c). Berikut persamaan matematis dari pendenefisian lingkaran.

$$(x - x_c)^2 + (y - y_c)^2 = r^2$$
....(2.1)

Pada persamaan (1), dilakukan penghitungan terhadap nilai y dan nilai x kemudian mendapat persamaan sebagai berikut.

$$y = y_c \pm \sqrt{r^2 - (X_c - X)^2}$$
....(2.2)

Dengan demikian, melakukan penghitungan yang berbeda dengan menaikkan nilai x dan menghitung nilai y. dan didapatkan persamaan sebagai berikut.

$$f(x,y) = (x - x_c)^2 + (y - y_c)^2 - r^2$$
....(2.3)

Jadi f(x,y) = 0 mewakili persamaan lingkaran. Untuk setiap titik memiliki ketentuan sebagai berikut.

- 1. f(x,y) = 0 titik koordinat tepat terletak didalam lingkaran
- 2. f(x,y) < 0 titik koordinat terletak didalam lingkaran
- 3. f(x, y) > 0 titik koordinat terletak di luar lingkaran

Didalam Algoritma *mid-point* keputusan suatu titik didalam sebuah radius/lingkaran ditentukan berdasarkan hasil tersebut.

2.11 Hotel Syari'ah

Pariwisata halal adalah jenis pariwisata baru, islam menganjurkan untuk seorang muslim melakukan perjalanan dan menikmati keindahan ciptaan Allah SWT serta melakukan ziarah untuk mengenal warisan budaya islam (Vargas-Sánchez & Moral-Moral, 2019). Hubungan antara islam dan pariwisata dalam beberapa tahun terakhir telah banyak menarik minat wisatawan. Konsep-konsep seperti hotel syari'ah semakin mendapat perhatian masyarakat.

Hotel Syari'ah sendiri memiliki beberapa kriteria yang harus dipenuhi untuk mencapai keabsahan dalam mendirikan hotel halal tersebut. Sesuai peraturan kementrian pariwisata dan ekonomi kreatif Republik Indonesia nomor 2 tahun 2014 pasal 1 menjelaskan bahwa setiap hotel syariah harus memenuhi kriteria usaha hotel syariah. Pada pasal 1 ayat 5 dan ayat 6 menjelaskan bahwa setiap hotel syariah harus memenuhi syarat dari hotel syariah hilal-1 dan 2.

2.12 Smartphone

Dalam beberapa dekade perkembangan teknologi begitu pesat. Salah satu bukti nya adalah terciptanya perangkat *smartphone*. Kini *smartphone* menjadi perangkat yang sangat popular dan di inginkan. Pada tahun 2008, industri sistem operasi bernama Android, yang diperkenalkan oleh Google. Sejak itu, terdapat bermacam-macam aplikasi yang dijalankan pada sistem tersebut (Al Fawareh & Jusoh, 2017). Selain itu *smartphone* memiliki fitur-fitur canggih selain hanya untuk panggilan telepon dan mengirim pesan teks. *Smartphone* juga dilengkapi fitur seperti menampilkan foto, memainkan *game*, memutar video, menerima *e-mail*, *browsing* hanya dengan menghubungkannya dengan internet semua fitur-fitur tersebut dapat dijalankan (Sarwar, 2014) Dengan adanya fitur-fitur tersebut

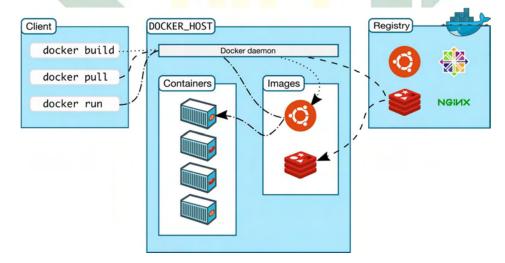
smartphone memungkinkan untuk menghubungkan teman dan keluarga sepanjang hari, berpotensi menuai manfaat dari interaksi sosial yang dilakukan dalam memanfaatkan fitur-fitur tersebut (Dwyer, Kushlev, & Dunn, 2017)

2.13 Database

Database adalah tempat penyimpanan data, efisiensi pemeliharaan data dan maintenance. Database memungkinkan untuk menyimpan dokumen, gambar, video dan berbagai jenis data lainnya (Sharma et al., 2010). Sejak kemunculannya, database menjadi salah satu domain pengetahuan yang paling banyak diteliti dalam ilmu komputer.

2.14 Docker

Docker merupakan sebuah *open platform* untuk melakukan *developing, shipping* dan menjalankan aplikasi. Docker berfungsi untuk menjadi wadah / *container* untuk memasukkan *software* secara utuh beserta dengan hal lain yang dibutuhkan oleh *software* tersebut. Pengaturan file pendukung akan diubah menjadi sebuah *Image*, dengan kata lain tujuan dari penggunaan docker adalah untuk aplikasi yang telah dibuat menjadi *portable*. Pada Gambar 2.9merupakan arstitektur docker.



Gambar 2.9 Docker Arsitektur

Docker menggunakan *client-server* arsitektur, docker *client* berkomunikasi dengan docker daemon yang berfungsi sebagai jembatan serta melakukan pekerjaan yang berat seperti *building, running* dan *distributing* ke docker *container*. Lalu didalam docker *container* akan menjalankan *image* yang kita

gunakan seperti *Operating System, Web Server, dll* yang digunakan sebagai pendukung aplikasi.

2.15 Javascript

Javascript mulai diperkenalkan pada tahun 1995 yang diciptakan oleh seseorang bernama Brendan Eich dalam kurun waktu 10 hari saat bekerja di Netscape. Sejak saat itu, bahasa ini dapat diadopsi oleh semua web browser, yang memungkinkan aplikasi web modern untuk berinteraksi secara langsung tanpa reload halaman untuk setiap perlakuan yang dilakukan (Haverbeke, 2018). Javascript dimanfaatkan untuk pengembang aplikasi terutama pada sisi client-side (Shah & Soomro, 2017).

2.16 Framework Javascript React Native

Pemrograman menggunakan basis bahasa javascript kini mulai berkembang pesat beberapa tahun terakhir. Bahasa ini merupakan bahasa pemrograman yang cukup fleksibel untuk diterapkan dan sangat cocok digunakan untuk segala jenis platform apapun. Dahulu bahasa pemrograman javascript digunakan hanya untuk membangun aplikasi berbasis website saja, namun seiring berjalannya waktu banyak sekali bermunculan framework berbasis javascript untuk memudahkan pengembang dalam melakukan inovasi untuk membuat suatu aplikasi. Salah satu framework javascript yang cukup terkenal yakni react native. Framework ini dikembangkan oleh salah satu perusahaan teknologi terbesar yaitu facebook untuk memudahkan dalam pengembangan aplikasi yang mengandalkan proses pengambilan data secara cepat. Framework ini adalah salah satu dari framework reactjs yang digunakan untuk membuat aplikasi website.

2.17 Node.Js

Dalam perkembangannya *javascript* yang hanya digunakan untuk *client-side*. Salah satu bentuk pengembangan bahasa pemrograman *javascript* yang merubah paradigma hanya untuk *client-side* kini berubah semenjak hadirnya Node.Js. Node.Js merupakan bahasa pemrograman berbasis *javascript* yang dapat berjalan pada sisi *server-side*. Tidak hanya itu Node.Js juga dapat berjalan pada sisi *client-side* seperti *javascript* pada umumnya. Node.Js

merupakan salah satu pesaing utama yang hadir dalam perkembangan teknologi kedepannya tertutama pada sisi *server-side* (Shah & Soomro, 2017).

2.18 Unified Modelling Language (UML)

Unified Modelling Language (UML) merupakan bahasa pemodelan visual yang digunakan untuk mengembangkan suatu perangkat lunak berorientasi objek. Bahasa ini diadopsi oleh Object Management Group (OMG) sebagai bahasa pemodelan standarnya pada tahun 1997 (Keng Siau, 2010). UML digunakan untuk menyatukan konsep dasar, terminologi, dan simbol grafik untuk menjelaskan langkah-langkah dasar yang harus diikuti dalam proses pembuatan perangkat lunak (Jianhu Zheng, 2014). UML terdiri atas 3 bagian utama antara lain sebagai berikut:

2.18.1 Structural Classification

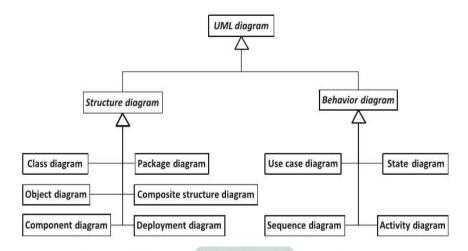
Pada Structural Classification ini menjelaskan hubungan sebuah diagram pada UML yang terdiri dari class diagram, use case diagram, component diagram, serta deployment diagram.

2.18.2 Dynamic Behavior

Dynamic Behavior menggambarkan perilaku sebuah sistem dari waktu ke waktu. Dynamic Behavior dapat digambarkan sebagai serangkaian perubahan pada snapshot dari sebuah sistem yang diambil dari tampilan statis. Dynamic Behavior terdiri dari beberapa komponen seperti state machine view, activity view, dan interaction view.

2.18.3 Model Management

Model Management mendeskripsikan sebuah organisasi kedalam unit hirarki. Tampilan model management juga berperan pada model lain serta melakukan pengembangan serta control konfigurasi dalam sebuah management. Dalam beberapa sumber menerangkan bahwa UML hanya memiliki 2 bagian, yaitu structure diagram dan behavior diagram. Pada gambar 2.9 dijelaskan pembagian struktural diagram UML.



Gambar 2.10 Struktural UML Diagram (Fan, 2015)

2.19 Network Latency

Network Latency merupakan kunci utama dalam sebuah jaringan untuk mengoptimalkan sebuah kinerja aplikasi dengan memanfaatkan informasi, proximity dan topologi (buford, 2009). Salah satu bagian yang terpenting dalam latency adalah time delays yang terjadi saat ada transfer data melalui jaringan komunikasi. Hal tersebut terjadi karena paket data harus melalui beberapa perangkat keras komunikasi lalu diproses oleh banyak komponen perangkat lunak (M. Sriram Iyengara, 2006). Salah satu cara untuk mengukur network latency dapat menggunakan Round-Trip-Time (RTT).

2.20 Test Case Scenario

Test Case Scenario merupakan salah satu tahap pada pengujian black box. Pengujian ini membutuhkan sebuah lampiran atau dokumen yang berisi tentang langkah-langkah pengujian fungsionalitas untuk perangkat lunak berdasarkan fitur-fitur yang disediakan oleh perangkat lunak tersebut. Didalam test case scenario ada beberapa tujuan, antara lain.

- Mempelajari sebuah produk atau perangkat lunak yang dibuat
- Menghubungkan dokumen pengujian dengan document requirement
- Mengatasi masalah atau kegagalan sistem lebih dini dengan menilai dokumen lama yang telah di identifikasi. (Kaner & Ph, 2013)

2.21 Integrasi Keilmuan

Implementasi teknologi *blockchain* merupakan sebagai bentuk upaya pengamanan data agar tidak disalah gunakan oleh pihak yang tidak bertanggung jawab serta mengintegrasikan apikasi pemetaan hotel syari'ah adalah untuk meningkatkan pariwisata halal untuk mendukung umat muslim dalam memecahkan permasalahan dalam hal pencarian hotel syari'ah agar tetap sesuai dengan ajaran agama islam yang berkonteks halal.

Berdasarkan hasil wawancara yang telah dilakukan dengan pakar agama islam bernama H. Abdul Rahman, M.Pd.I sebagai penyelenggara syari'ah kantor kementrian agama surabaya, mengatakan terdapat beberapa firman Allah SWT tentang penggunaan barang halal serta cara untuk menjaga barang yang bermanfaat. Ada 2 surah terkait hal itu, yang pertama yakni QS. At-Tahrim ayat 6 dan QS. Al-Baqarah ayat 168 dijelaskan sebagai berikut.

Artinya:

"Wahai orang-orang yang beriman! Peliharalah dirimu dan keluargamu dari api neraka yang bahan bakarnya adalah manusia dan batu; penjaganya malaikat-malaikat yang kasar, dan keras, yang tidak durhaka kepada Allah terhadap apa yang Dia perintahkan kepada mereka dan selalu mengerjakan apa yang diperintahkan. (QS. At-Tahrim [6]:66)"

Maksud dari ayat di atas adalah peliharalah berarti dalam hal ini menjaga diri sendiri dan keluarga dari api neraka. Diri sendiri dan keluarga merupakan aset penting dalam kehidupan, seperti hal nya data. Data merupakan salah satu aset yang harus dijaga dan disimpan. Dengan menggunakan teknologi blockchain diharapkan data tersebut dapat tersimpan dan dijaga dengan baik. Sehingga untuk menerapkan hal tersebut dibuatlah penelitian dengan judul geospatial retrieval hotel syari'ah berbasis PBFT untuk meningkatkan keamanan data.

Artinya:

"Wahai manusia! Makanlah dari (makanan) yang halal dan baik yang terdapat di bumi, dan janganlah kamu mengikuti langkah-langkah setan. Sungguh, setan itu musuh yang nyata bagimu. (QS. Al-Baqarah [168]:2)"

Ayat di atas menerangkan bahwa setiap manusia harus mengkonsumsi makan makanan yang halal. Selain halal, makanan tersebut juga harus baik, sehat, aman dan tidak berlebihan (Setiawan, 2014).



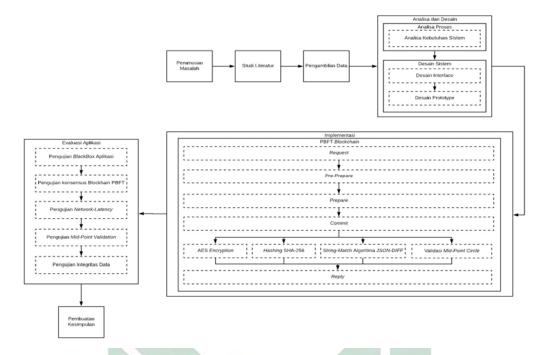
BAB 3

METODOLOGI PENELITIAN

Bab 3 menjelaskan tahapan-tahapan penelitian serta jadwal pelaksanaan yang dilakukan. Berikut adalah tahapan penelitian.

3.1 Tahapan Penelitian

Penelitian yang dilaksanakan tampak dalam Gambar 3.1 dibawah ini



Gambar 3.1 Tahapan Penelitian

3.1.1 Perumusan Masalah

Proses perumusan masalah yang menjadi latar belakang untuk dilakukannya penelitian *Practical Byzantine Fault Tolerance* (PBFT) pada aplikasi *geospatial retrieval* hotel syari'ah berbasis *mobile*, sesuai dengan perumusan yang tertera dalam bab pertama pada poin latar belakang. Permasalahan yang diangkat pada penelitian ini adalah pemanfaatan teknologi *blockchain* PBFT untuk penyimpanan data hotel syari'ah yang akan dipetakan melalui aplikasi berbasis *mobile* dengan menggunakan algoritma SHA-256 untuk melakukan teknik *hash*. Kemudian menggunakan algoritma AES sebagai teknik enkripsi dan Algoritma Javascript Object Notation – DIFF (JSON-DIFF) sebagai algoritma pencocokan data yang digunakan untuk mencocokkan data

hash yang dikirimkan oleh surveyor melalui aplikasi pada node blockchain yang terhubung dalam satu jaringan.

3.1.2 Studi Literatur

Dalam tahapan ini melakukan tinjauan pustaka dari beberapa studi terbaru mengenai *Geographic Infoamtion Retrieval* (GIR) berbasis teknologi *Blockchain*. Studi yang dilakukan meliputi *Practical Byzantine Fault Tolerance* (PBFT) *Blockchain, Geospatial Retrieval*, GIR, Algoritma JSON-DIFF, AES *encryption, Hashing* SHA-256, Algoritma *Midpoint Circle*, dan Pariwisata Halal. Metode PBFT *blockchain* dipilih karena memiliki *fault tolerance* yang lebih rendah dari metode yang lain.

3.1.3 Pengambilan Data

Pada tahap ini dilakukan pengambilan data koordinat *latitude* dan *longitude* hotel syari'ah yang ada di kota Surabaya. Hotel syari'ah yang akan diambil datanya adalah sebagai berikut:

- a. Penginapan Syari'ah Prenjak
- b. Hotel Syari'ah Walisongo Surabaya
- c. Grand Kalimas Hotel Syari'ah
- d. Namira Syari'ah Hotel Surabaya

Data hotel yang dibutuhkan pada masing-masing hotel yaitu 25 koordinat *latitude* dan *longitude* dengan total semua data 100 koordinat.

3.1.4 Analisa Kebutuhan Sistem dan Desain Sistem

Pada tahap ini terbagi menjadi 2 bagian yang pertama tahap analisa kebutuhan sistem dan tahap desain sistem. Tahap analisa kebutuhan sistem dilakukan untuk merancang kebutuhan hardware serta system requirements yang dibutuhkan untuk mengembangkan aplikasi. Pada tahap selanjutnya yaitu desain sistem. Desain sistem dilakukan untuk menganalisis desain interface dan desain prototype pada aplikasi yang akan dibangun. Selain itu desain sistem Dalam tahap ini dilakukan proses desain interface terhadap aplikasi mobile yang dibuat dan dilakukan proses desain prototype berdasarkan hasil analisa pada tahap sebelumnya.

3.1.5 Implementasi

Pada tahap ini dilakukan pembuatan program berdasarkan desain yang dibuat pada tahapan desain sistem. Aplikasi dikembangkan menggunakan framework React Native dari sisi front-end sedangkan dari sisi back-end menggunakan framework NodeJS sebagai integrasi aplikasi dengan blockchain PBFT.

3.1.5.1 PBFT Blockchain

Tahap implementasi dimulai dari tahapan PBFT *blockchain* yang memiliki 5 tahap yang harus dilalui. Tahapan tersebut adalah *request, pre-prepare, prepare, commit* dan *reply* apabila telah selesai melalui tahapan-tahapan tersebut proses PBFT *blokchain* akan selesai apabila memiliki nilai *reply* seperti pada poin 2.3.2.

3.1.5.2 Validasi Algoritma Mid-Point Circle

Tahap selanjutnya validasi *mid-point circle*. Validasi ini memerlukan sebuah nilai seperti pada poin 2.10 nomor 1. Nilai tersebut didapatkan dari koordinat *latitude* dan *longitude*. Nilai 0 pada persamaan tersebut berarti sebuah radius yang ditentukan oleh sistem nantinya, pada aplikasi yang dibuat menggunakan radius 50 meter. Apabila nilai x dan y kurang dari 50m maka titik koordinat berada di dalam lingkaran dengan kata lain bernilai *true*, dan apabila nilai x dan y lebih dari 50m maka titik koordinat berada di luar lingkaran dengan kata lain bernilai *false*.

3.1.5.3 *Hashing* SHA-256

Setelah tahapan pada PBFT *blockchain* selesai, tahapan selanjutnya yaitu melakukan teknik *hashing* menggunakan SHA-256 terhadap *prevhash* pada *block* yang yang saling terhubung dalam PBFT *blockchain*.

3.1.5.4 AES Encryption

Setelah *hashing* SHA-256 tahapan selanjutnya yaitu melakukan *enkripsi* terhadap data *geospatial retrieval* hotel syari'ah. Data tersebut meliputi nama hotel, *latitude* hotel, *longitude* hotel, alamat hotel dan no.hp hotel menggunakan Algoritma AES.

3.1.5.5 String-Match Algoritma JSON-DIFF

Data yang di *request* akan melalui proses pencocokkan antara *block* satu dengan *block* yang lain dalam PBFT *blockchain*. Setelah semua proses terlewati *user surveyor* akan menerima *reply* berupa *public key* yang telah di *decrypt* sehingga data dapat ditampilkan kedalam aplikasi.

3.1.6 Evaluasi Sistem

Pengujian sistem dilakukan dengan 5 tahap. Tahap yang pertama yaitu melakukan pengujian konsesus PBFT dengan nilai fault tolerance f=1. Dengan memasukkan nilai reply seperti pada poin 2.3.2. maka nilai maksimum fault tolerance adalah 4. Berdasarkan pada Tabel 2.1 PBFT memiliki nilai fault tolerance 33%, oleh karena itu nilai minimum yang bisa didapatkan dengan nilai f=1 adalah 3. Dengan skenario 1 primary node dan 3 node yang terhubung dalam blockchain. Berikut skenario pengujian konsensus blockchain PBFT.

Tabel 3.1 Skenario Pengujian Konsensus PBFT

No	Primary	Node 1	Node 2	Node 3	Expected Result
1	Connect	Connect	Connect	<u>Con</u> nect	Terbuat
2	Connect	Connect	Connect	<u>Not</u>	Gagal Terbuat
				Connected	
3	Connect	Connect	Not	Not	Gagal Terbuat
			Connected	Connected	
4	Connect	Not	Not	Not	Gagal Terbuat
		Connected	Connected	Connected	

Pengujian selanjutnya yaitu melakukan pengujian *blackbox* pada aplikasi yang dibuat pada sistem operasi android 7, 8 dan 9. Pengujian *blackbox* ini dibutuhkan sebuah *test case scenario* sebagaimana yang ditunjukkan pada Tabel 3.2.

Tabel 3.2 Test Case Scenario Pengujian Blackbox Aplikasi

No	Test Case Description	Untitled Question	Pre-Condition	Test Step	Expected Result
1	Check GPS Services	I V/ 9 I 1 CI	services GPS dalam keadaan	 Buka Aplikasi Tunggu Splash Screen berhasil dimuat 	Tampilan dialihkan ke halaman map

			I	T	
		Invalid	Kondisi services GPS dalam keadaan mati	 Buka Aplikasi Tunggu Splash Screen berhasil dimuat 	Menampilkan check services GPS
2	Register	Valid	1. Kondisi services GPS dalam keadaan menyala 2. Menampilkan halaman peta	1. Menekan tombol login pada sebelah kanan button search 2. Klik tombol SignUp 3. Mengisi email sebagai username pengguna (surveyor) pada form 4. Mengisi password pengguna pada form	Muncul popup akun berhasil dibuat
2	User	Invalid	1. Kondisi services GPS dalam keadaan menyala	sebagai	Muncul popup error akun gagal
			2. Menampilkan halaman peta	username pengguna (surveyor) pada form 4. Mengisi password pengguna pada form	dibuat
3	Login User	Valid	1. Kondisi services GPS dalam keadaan menyala 2. Menampilkan halaman peta	1. Menekan tombol login pada sebelah kanan button search 2. Mengisi email sebagai username pengguna	Tampilan dialihkan ke halaman map surveyor

		Invalid	1. Kondisi services GPS dalam keadaan menyala 2. Menampilkan halaman peta	(surveyor) pada form 3. Mengisi password pengguna pada form 1. Menekan tombol login pada sebelah kanan button search 2. Mengisi email sebagai username pengguna (surveyor) pada form 3. Mengisi password pengguna pada form	Muncul popup error username atau password yang dimasukkan salah
4	Check Mid Point Circle Hotel	Valid	1. Kondisi services GPS dalam keadaan menyala 2. Berada pada radius hotel syari'ah yang telah ditentukan	1. Buka Aplikasi 2. Ketika aplikasi tersebut dibuka lokasi dari user telah dideteksi oleh aplikasi 3. User Surveyor melakukan	User Surveyor dapat melakukan penginputan data
		Invalid	1. Kondisi services GPS dalam keadaan menyala 2. Berada pada radius hotel syari'ah yang telah ditentukan	1. Buka Aplikasi 2. Ketika aplikasi tersebut dibuka lokasi dari user telah dideteksi oleh aplikasi 3. User Surveyor melakukan penginputan data	Muncul popup error, user surveyor berada pada diluar radius hotel syari'ah yang telah ditentukan

				hotel syari'ah dengan cara menekan bottom tab pada halaman peta surveyor	
		Valid	1. Kondisi services GPS dalam keadaan menyala 2. Aplikasi melakukan ping ke blockchain primary	1. Buka Aplikasi 2. Tunggu aplikasi melakukan request	Aplikasi dapat digunakan seperti biasa
5	Check Request Blockchain	Invalid	1. Kondisi services GPS dalam keadaan menyala 2. Aplikasi melakukan ping ke blockchain primary	1. Buka Aplikasi 2. Tunggu aplikasi melakukan request	Muncul popup error bahwa aplikasi tidak dapat digunakan sampai blockchain primary kembali dalam kondisi aktif

Lalu pengujian tahap selanjutnya yaitu melakukan peghitungan *network-latency* enkripsi dan dekripsi *blockchain* dengan menggunakan jaringan 3G dan 4G. Pada saat enkripsi dan dekripsi data dilakukan sebanyak 100 kali dengan total 200 data pada masing-masing proses tersebut.

Pengujian tahap akhir yaitu melakukan pengujian integritas data pada penyimpanan tanpa PBFT *blockchain* dengan skenario pada Tabel 3.3 dan penyimpanan data menggunakan PBFT *blockchain* dengan menggunakan skenario seperti pada Tabel 3.4.

Pengujian data tanpa PBFT *blockchain* menggunakan 2 skenario sesuai dengan Tabel 3.3. Data yang digunakan adalah data berformat JSON dengan total 100 data. Total data tersebut digunakan pada masing-masing skenario dengan rincian 50 data benar dan 50 data salah.

Tabel 3.3 Scenario Pengujian Integritas Data tanpa PBFT blockchain

No.	SKENARIO	
110.	Primary	Expect Result
1	Data Benar	Bisa Diakses
2	Data Salah	Tidak Bisa Diakses

Kemudian pengujian dengan PBFT blockchain melakukan pengujian terhadap private key hash pada blockchain PBFT dengan teknik integritas data dengan memperhatikan struktur data yang tersimpan (Sivathanu et al., 2005). Skenario dibuat berdasarkan teori PBFT blockchain dengan memenuhi nilai fault tolerance sebesar 4 node dan menguji nilai minimum dari PBFT yaitu 3f atau 3 node. Pengujian dilakukan sesuai dengan skenario pada Tabel 3.4. Data yang diuji yaitu 104 data JSON dengan masing-masing 13 data pada setiap nomor skenario.

Tabel 3.4 Scenario Pengujian Integritas Data dengan PBFT blockchain

No.	4	S	KENARIO		
110.	Primary	Node 1	Node 2	Node 3	Expect Results
1	Data Benar	Data	Data	Data	Bisa Diakses
		Benar	Benar	Benar	
2	Data Benar	Data	Data	Data	Bisa Diakses
		Benar	Benar	Salah	
3	Data Benar	Data	Data	Data	Tidak Bisa
		Benar	Salah	Salah	Diakses
4	Data Benar	Data	Data	Data	Tidak Bisa
		Salah	Salah	Salah	Diakses
5	Data Salah	Data	Data	Data	Tidak Bisa
		Benar	Benar	Benar	Diakses
6	Data Salah	Data	Data	Data	Tidak Bisa
		Benar	Benar	Salah	Diakses
7	Data Salah	Data	Data	Data	Tidak Bisa
		Benar	Salah	Salah	Diakses
8	Data Salah	Data	Data	Data	Tidak Bisa
		Salah	Salah	Salah	Diakses

3.1.7 Pembuatan Kesimpulan

Pembuatan kesimpulan dapat dilakukan ketika telah melalui semua tahapan penelitian, berdasarkan hasil pengujian konsensus PBFT *blockchain*, hasil penghitungan *nework latency* enkripsi dan dekripsi PBFT *blockchain*, Pengujian *Mid-Point Validation*. serta hasil tingkat akurasi dari pengujian integritas data yang tersimpan didalam PBFT *blockchain*.



BAB 4

HASIL DAN PEMBAHASAN

4.1 Hasil Analisis Kebutuhan Sistem

Berdasarkan latar belakang serta rumusan masalah terkait dengan implementasi *Practical Byzantine Fault Tolerance* (PBFT) *Blockchain* untuk penyimpanan data hotel syari'ah menggunakan aplikasi berbasis *mobile* dengan sistem spesifikasi *minimum* perangkat menggunakan Android versi 7, 8 dan 9. Pada sisi *frontend* aplikasi ini dikembangkan menggunakan *framework javascript react native* serta layanan peta digital dari *GoogleMaps*. Pada sisi *backend* aplikasi ini dibangun dengan menggunakan 4 *container docker* yang memiliki *image* maria *database*, *framework javascript nodejs* dan adminer sebagai *database administrator*.

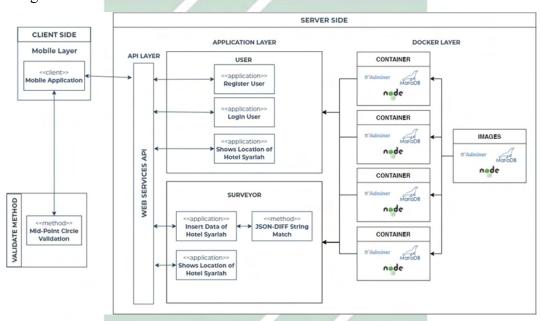
Fungsi utama pada aplikasi ini adalah menyajikan sebuah peta dengan informasi yang dimiliki oleh hotel syari'ah mulai dari nama, alamat, no handphone yang terletak di kota Surabaya. Aplikasi ini memiliki dua tipe jenis user, user general dan user surveyor. User general hanya dapat mengakses peta serta informasi dari hotel syari'ah yang disajikan oleh aplikasi sedangkan user surveyor dapat melakukan input data baru data hotel syari'ah kedalam aplikasi. Aplikasi ini membutuhkan Global Postitioning System (GPS) dari perangkat user surveyor untuk mengambil nilai koordinat longitude dan latitude agar aplikasi dapat memvalidasi posisi user surveyor yang telah berada pada jangkauan hotel syari'ah yang ditentukan oleh sistem. Apabila user surveyor diluar dari radius maka user surveyor tidak dapat mengisi form dan hanya dapat menikmati fitur sama seperti user general.

Aplikasi ini memiliki beberapa fitur seperti menu peta, menu search, menu login, menu registrasi, dan menu surveyor. Ketika sudah masuk kedalam aplikasi, nantinya akan ditampilkan sebuah peta digital yang bersumber dari library react native maps, yang merupakan sebuah pengembangan peta digital milik Google yang hanya dapat digunakan oleh framework react native. Pada menu ini, disajikan sebuah marker pada setiap hotel syari'ah yang datanya diambil dari blockchain yang berhasil decode.

Pada menu search, user dapat melakukan pencarian hotel syari'ah yang ada pada aplikasi. User dapat mengetikkan keyword hotel syari'ah yang ingin dicari. Selanjutnya menu registrasi, pada menu ini digunakan untuk mendaftar akun agar bisa login kedalam aplikasi. Menu login pada aplikasi digunakan untuk masuk kedalam aplikasi sebagai user surveyor. Kemudian pada menu surveyor dapat mengisi form untuk melengkapi data-data hotel syari'ah.

4.2 Desain Arsitektur Sistem

Berdasarkan hasil analisa kebutuhan bentuk desain arsitektur desain sistem pada aplikasi pemetaan hotel syari'ah ditunjukkan pada Gambar 4.1 berikut dari gambar tersebut.



Gambar 4.1 Desain Arsistektur Sistem

Pada Gambar 4.1 terdapat 3 bagian desain arsitektur yang terbuat, yaitu server side, client side serta validate method. Bagian pertama yaitu client side yang hanya berisi mobile layer, yang didalamnya terdapat client application atau sebuah aplikasi yang akan digunakan oleh pengguna. Bagian yang kedua yaitu validate method yang berguna untuk memvalidasi user surveyor apakah didalam radius dari hotel syari'ah yang nantinya akan disurvey.

Bagian terakhir yaitu *server side*. Pada bagian ini terdapat 3 bagian, yang pertama yaitu API Layer yang bertugas untuk membuat *Restful API* yang mengirimkan data dengan format JSON untuk mengkomunikasikan antara

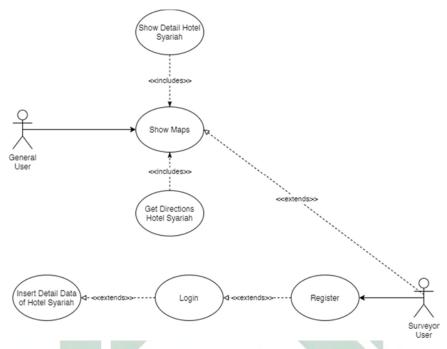
client side dengan server side. Yang kedua yaitu application layer, layer ini memiliki dua tipe user yaitu user dan surveyor.

User layer berfungsi untuk mengirimkan berkas data user sebagai back-end dari sistem untuk user layer. Layer selanjutnya adalah surveyor layer, pada bagian ini terdapat 2 bagian utama. Bagian yang pertama yaitu berfungsi untuk mengirimkan berkas data hotel syari'ah. Selanjutnya data tersebut akan divalidasi menggunakan JSON-DIFF string match.

Selanjutnya yaitu docker layer, pada layer ini memiliki 2 struktur inti yaitu container dan image. Container bertugas untuk menjalankan operasi yang dibutuhkan oleh sistem penyimpanan. Didalam container diperlukan yang namanya image, image berfungsi untuk menanamkan registry yang dibutuhkan oleh container. Dalam hal ini registry yang dibutuhkan adalah adminer sebagai database management, MySQL sebagai relation database management system (RDMBS) dan NodeJS sebagai bahasa pemrograman penghubung antara client side dan serverside yang berbasis javascript.

4.3 UML Use Case Diagram

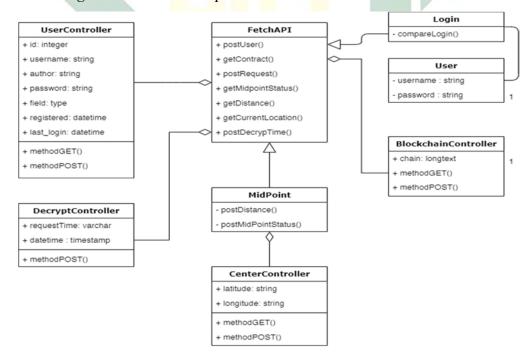
Pada diagram ini menampilkan skenario dari kedua jenis user, yakni user general dan user surveyor yang saling berhubungan. Akan tetapi setiap user hanya dapat melakukan hak akses nya seperti pada Gambar 4.4. Sesuai dengan analisis yang dilakukan user general hanya dapat menikmati beberapa fitur dari aplikasi. User general hanya dapat melihat peta, menekan marker hotel syari'ah, menampilkan informasi detail mengenai hotel syari'ah dan mendapatkan directions ke hotel syari'ah tanpa melakukan input data pada aplikasi. Sedangkan pada user surveyor diharuksan untuk melakukan login terelebih dahulu, akan tetapi sebelum melakukan login user diharapkan untuk melakukan register terlebih dahulu agar menjadi user surveyor. Setelah itu user surveyor dapat mengakses fitur penginputan data hotel syari'ah.



Gambar 4.2 UML Use Case Diagram

4.4 UML Class Diagram

Class Diagram merupakan diagram yang mempunyai sifat struktural berfungsi untuk menggambarkan rancangan class yang terdapat pada aplikasi. Class Diagram divisulisasikan pada Gambar 4.4 berikut ini.



Gambar 4.3 UML Class Diagram

Ada 8 class yang terdapat dalam rancangan class diagram pada aplikasi ini. Class tersebut terbagi kedalam 2 bagian, yaitu bagian front-end dan back-end. Pada bagian front-end terdapat classname Login, Login, dan FetchAPI, lalu pada bagian back-end terdapat class name CenterController, UserController, BlockchainController dan DecryptController. Setiap class memiliki fungsinya masing-masing berikut penjelasan fungsi pada setiap class.

1. User

Bagian classname user hanya terdapat private attribute berupa username dan password dengan tipe data string. Attribute username dan password didapatkan dari hasil masukkan dari user. Classname ini berelasi dengan class Login dengan tipe relasi association.

2. MidPoint

Classname midpoint terdiri dari 2 private operation yaitu postDistance() dan postMidPointStatus(). Operasi postDistance() melakukan penyimpanan data baru berupa data jarak antara koordinat user dengan hotel syari'ah. Kemudian operasi postMidPointStatus() melakukan penyimpanan status true false kedalam database yang didapatkan dari fitur GPS smartphone melalui operasi getMidPointStatus() dan getDistance() pada class FetchAPI.

3. Login

Didalam login tidak terdapat attribute melainkan sebuah private operation yaitu compareLogin(). Pada classname login terdapat 2 relasi, yang pertama yaitu tipe relasi association yang berelasi dengan classname user. Relasi berikutnya merupakan berelasi dengan classname FetchAPI dengan tipe relasi inheritance. Operasi ini menghasilkan parent dan child yang nantinya menjalan virtual function yang dimiliki oleh parent. Pada bagian operasi compareLogin() nantinya akan melakukan perbandingan email atau username dan password dari hasil masukkan user dengan email dan password yang berasal dari user.

4. FetchAPI

Didalam *class* ini hanya terdapat *public operation* didalamnya, karena *class* ini merupakan *class parent* dari 3 *class* lainnya yaitu *class login*, hotel dan

register. Dengan begitu class ini memiliki generalization terhadap ketiga class tersebut. Operasi yang terdapat pada class ini antara lain, postUser() yaitu melakukan proses operasi memasukkan data baru kedalam database user berdasarkan hasil masukkan username dan password dari user melalui aplikasi. Proses operasi getContract() melakukan proses dekripsi pada data yang telah tersimpan didalam blockchain, kemudian dilakukan dekripsi agar data yang tersimpan pada blockchain dapat terbaca dalam bentuk data asli.. Kemudian terdapat proses operasi postRequest() yaitu memasukkan data baru hotel syari'ah kedalam blockchain. Pada operasi getCurrentLocation() aplikasi menerima titik koordinat user dengan menggunakan fitur GPS pada smartphone user. Operasi terakhir pada FetchAPI merupakan proses pemasukkan data baru berupa requestTime dan timestamp ketika user membuka aplikasi, lalu data tersebut disimpan kedalam database.

5. CenterController

Dalam class CenterController terdapat sebuah public attribute berupa latitude dan longitude hotel syari'ah dengan tipe data string. Didalam class ini terdapat public operation methodGET() dan methodPOST().

6. DecrpytController

Pada class ini terdapat 2 public attribute dan public operation. Public attribute terdiri dari requestTIme yang memiliki tipe data varchar dan datettime dengan tipe data timestamp.

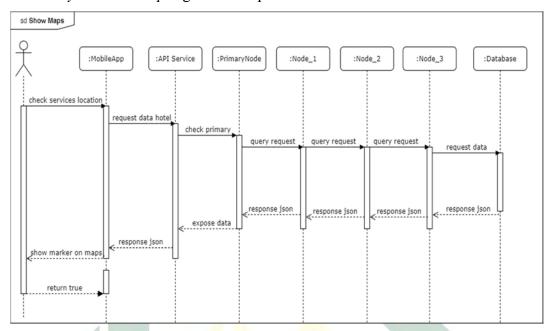
7. BlockchainController

Class terakhir pada kesuluruhan class diagram ini, merupakan class inti dari dibangunnya aplikasi ini. Didalam class ini terdapat 1 public attribute dan public operation. Public attribute tersebut adalah chain dengan tipedata longtext didalam attribute ini terdapat data terenkripsi yang terdiri dari hash, data, author, serta timestamp. Kemudian public operation terdiri dari methodGET() dan methodPOST().

4.5 UML Sequence Diagram

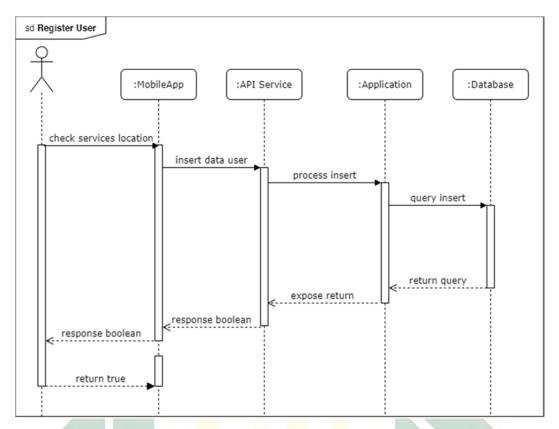
UML Sequence Diagram merupakan representasi sebuah diagram yang memiliki interaksi antar object yang terdapat dalam sebuah aplikasi. Ada 3 fungsi utama yang digambarkan kedalam bentuk sequence diagram, yakni

meliputi *show map, register user, login* serta *input* data hotel syari'ah oleh *user surveyor. Show Map* digambarkan pada Gambar 4.9 berikut ini.



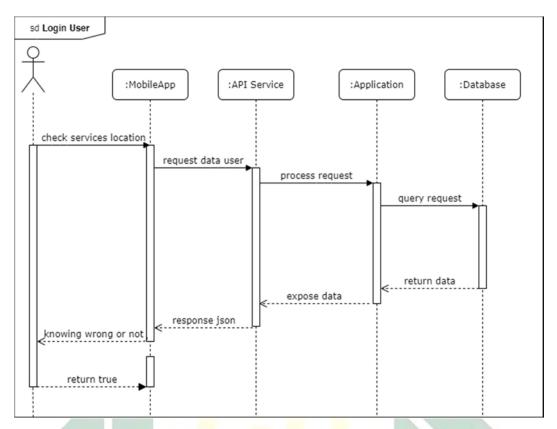
Gambar 4.4 *UML Sequence Diagram Show Maps*

Sequence Diagram Register User memiliki 1 actor dan 7 buah object yang berbeda dan memiliki fungsi masing-masing. Actor pada sequence berperan sebagai pengguna atau *user* dari aplikasi, sedangkan 7 *object* lainnya meliputi mobile app, api service, primary node, node 1, node 2, node 3 dan database. Alur interaksi dimulai dari actor yang mengirimkan pesan check service location kepada mobile app dengan nilai true. Setelah itu mobile app dapat melakukan request data hotel kepada api services. Setelah api services menerima request data hotel syari'ah maka api services akan meneruskan proses request ke primary node untuk melakukan pengecekan status primary node, apabila bernilai true maka akan diteruskan ke object node 1, node 2 dan node 3. Apabila nilai minimum dan maksimum telah terpenuhi maka proses selanjutnya yaitu melakukan query request yang nantinya dikirimkan ke database. Setelah itu database akan mengirimkan return data pada node 1, node 2, node 3 dan primary node yang lalu di expose ke api services untuk dilakukan pengonversian data berformat JSON. Lalu pada mobile app akan menampilkan marker hotel syari'ah. Apabila user ingin masuk kedalam aplikasi, user diharuskan untuk melakukan proses registrasi yang digambarkan pada Gambar 4.7.



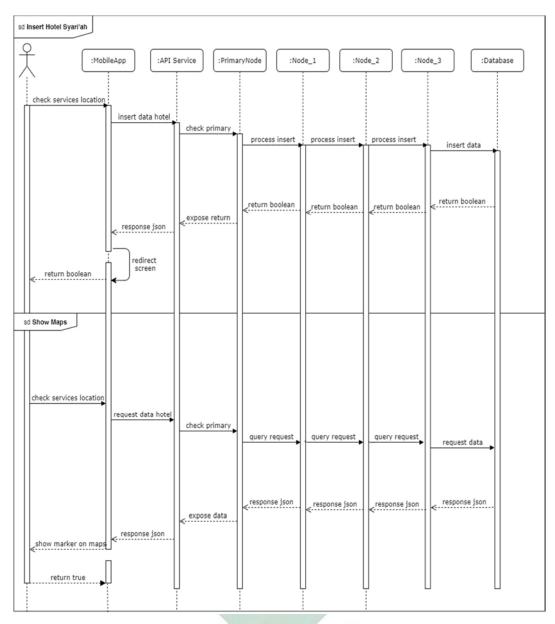
Gambar 4.5 UML Sequence Diagram Register User

Proses check services location tetap dilakukan karena user tidak bisa melakukan registrasi apabila tidak mengaktifkan service location. Proses selanjutnya yaitu proses insert data user yang berisi data username dan password yang diteruskan ke proses api service. Lalu api service akan mengirimkan data tersebut kepada proses application sehingga pada proses ini dapat pemrosesan insert yang kemudian dilakukan query untuk disimpan kedalam database. Setelah proses insert data berhasil lalu user akan mendapatkan return boolean bahwa akun telah berhasil dibuat. Proses selanjutnya yaitu user harus melakukan login agar dapat menjadi user surveyor, yang digambarkan pada Gambar 4.8.



Gambar 4.6 UML Sequence Diagram Login User

Sequence Diagram Login User memiliki alur interaksi yang sama dengan show maps, akan tetapi memiliki pada diagram ini data yang request berupa data user yang dikirimkan ke database dan mendapatkan nilai return apakah username dan password sesuai. Setelah login user surveyor dapat mengisi data hotel seperti yang digambarkan pada Gambar 4.9.



Gambar 4.7 UML Sequence Diagram Surveyor

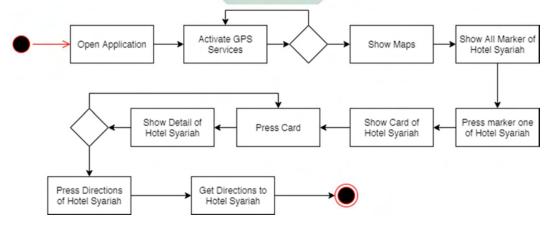
Dalam diagram sequence surveyor dapat melakukan proses insert data hotel syari'ah dengan syarat user surveyor telah didalam radius dari hotel syari'ah yang telah ditentukan oleh sistem, apabila return boolean bernilai false maka user surveyor tidak melakukan survey dan apabila bernilai true maka mobile app akan melakukan proses selanjutnya yaitu request data insert hotel yang telah di input oleh user. Api services akan menerima request dan akan diteruskan ke primary node untuk mendapatkan nilai return true pada primary node. Setelah itu primary node akan melakukan commit pada node 1, node 2, node 3 dengan memenuhi nilai minimum dan maksimum blockchain PBFT.

Setelah memenuhi nilai tersebut proses *query insert* akan dilakukan. Apabila proses *insert* berhasil maka *user surveyor* akan kembali ke *screen maps*. *User surveyor* juga memiliki *show maps* yang sama dengan *user general*.

4.6 UML Activity Diagram

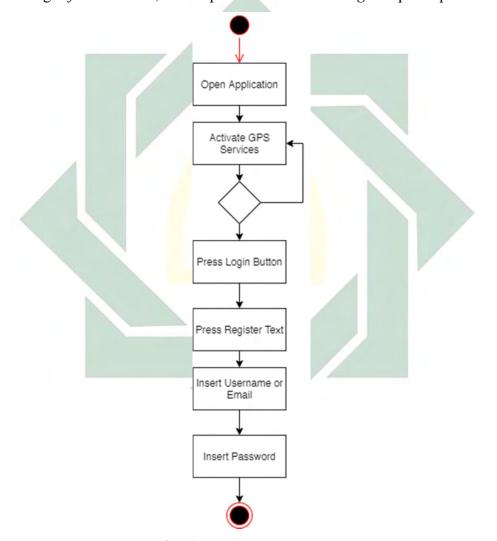
Activity Diagram menggambarkan sebuah aktivitas dari sebuah sistem dari awal hingga proses lainnya yang saling terkait untuk memudahkan dan menganalisa untuk sistem yang dibuat nantinya. Pada Gambar 4.4 menjelaskan ada 4 bagian utama yaitu show maps, register, login, serta insert data.

Diagram aktivitas show maps dimulai dari open application atau membuka aplikasi pemetaan hotel syari'ah. Setelah itu user diharuskan untuk mengaktifkan layanan GPS agar dapat membuka peta dari aplikasi, apabila user tidak mengaktifkan layanan GPS maka user tidak dapat menggunakan aplikasi. Setelah user mengaktifkan layanan GPS user dapat mengakses peta pada aplikasi, apabila user ingin mengetahui informasi detail mengenai hotel syari'ah user dapat menekan marker pada peta yang tersedia. Setelah menekan marker, maka card informasi akan muncul pada halaman yang sama yaitu peta, setelah itu user diharuskan untuk menekan card tersebut untuk mengetahui detail dari informasi hotel syari'ah. Sekarang posisi screen user adalah detail informasi dari hotel syari'ah, apabila user ingin mengetahui directions ke hotel syari'ah user dapat menekan tombol get directions pada aplikasi dan apabila user ingin kembali ke peta user dapat menekan tulisan get back pada aplikasi. Activity diagram untuk fitur show maps di ilustrasikan pada Gambar 4.10.



Gambar 4.8 UML Activity Diagram Show Maps

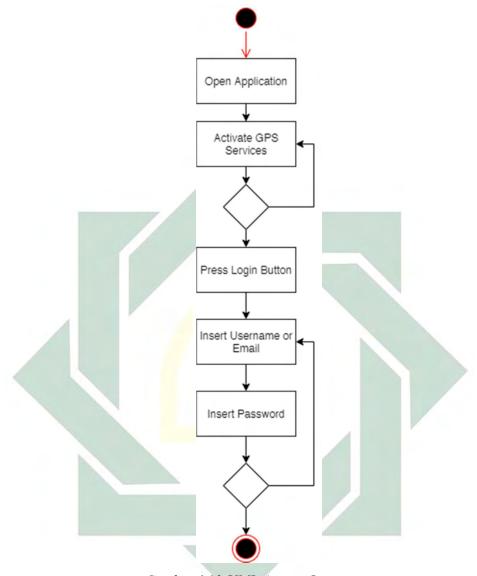
Diagram aktivitas yang kedua menjelaskan alur aktivitas pada proses register atau registrasi oleh user yang ingin mengakses aplikasi sebagai user surveyor. Aktivitas tersebut telah digambarkan pada Gambar 4.11, sebelum masuk pada menu registrasi user diharapkan untuk tetap mengaktifkan layanan GPS apabila tidak mengaktifkan maka user tidak dapat mengakses menu registrasi. Selanjutnya user harus memasukkan username atau email serta memasukkan password untuk melengkapi form register tersebut. Setelah mengisi form tersebut, user dapat menekan tombol register pada aplikasi.



Gambar 4.9 UML Activity Registration

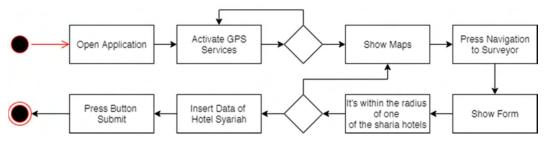
Setelah melakukan proses registrasi, apabila *user* ingin menjadi *user* surveyor, maka *user* tersebut harus melakukan *login* pada sistem dengan memasukkan *username* dan *password* yang telah didaftarkan pada proses

sebelumnya. Proses *login* ini telah digambarkan pada Gambar 4.12. Agar dapat mengakses aplikasi *user* tetap harus mengaktifkan layanan GPS.



Gambar 4.10 UML Activity Login

Setelah melakukan *login*, untuk melakukan penginputan terhadap detail data hotel syari'ah, *user surveyor* harus menekan tombol pada *bottom navigation* menuju *form* pengisian informasi hotel syari'ah. Akan tetapi *user surveyor* harus berada didalam radius dari hotel syari'ah yang akan disurvey apabila *user* melebihi radius 50m maka *user* tersebut tidak bisa melakukan penginputan data hotel syari'ah dan hanya dapat mengakses fitur yang sama seperti *user general* akan tetapi tetap sebagai *user surveyor*. Proses ini telah digambarkan pada Gambar 4.13 berikut ini.

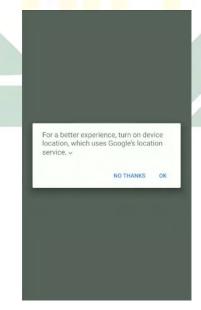


Gambar 4.11 UML Activity Surveyor

4.7 Desain User Interface (UI) Aplikasi

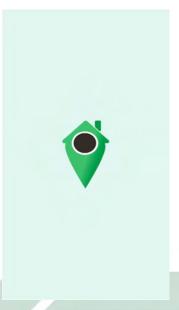
Berikut beberapa desain yang berhasil dibuat dari pengembangan aplikasi ini. Tampilan terbagi menjadi dua jenis tampilan untuk *user general* dan tampilan untuk *user surveyor*, untuk *user surveyor* memiliki navigasi *bottom tab* sedangkan *user general* tidak memiliki navigasi *bottom tab*. Tampilan tersebut meliputi *splash screen, maps screen, info detail screen, login screen, register screen, search screen,* dan *form surveyor*.

Pada tampilan *splash screen* terdapat logo aplikasi dengan background berwarna putih kehijauan. Proses pengecekan lokasi dilakukan ketika aplikasi ini telah dibuka apabila *service gps* tidak diaktifkan maka akan muncul peringatan seperti pada Gambar 4.14.



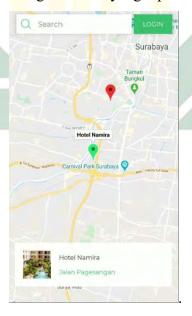
Gambar 4.12 Service GPS

Apabila *service gps* telah diaktifkan maka yang akan tampil hanya logoaplikasi saja seperti pada Gambar 4.15.



Gambar 4.13 Splash Screen

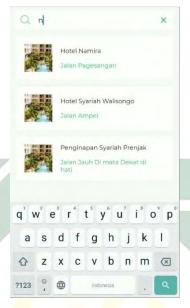
Setelah itu *screen* akan beralih ke *maps screen*, Terdapat beberapa fitur yang disediakan yaitu terdapat *marker* dari lokasi *user* beserta lokasi hotel syari'ah. Apabila *user* ingin menuju salah satu dari hotel tersebut, *user* dapat menekan *marker* berwarna hijau pada *maps* lalu akan muncul sebuah *card* yang berisi informasi singkat mengenai hotel yang dipilih. Seperti pada Gambar 4.16



Gambar 4.14 Maps Screen

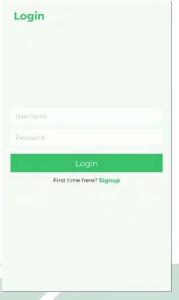
Untuk memudahkan *user* dalam pencarian hotel syari'ah, *user* dapat menggunakan fitur *search* yang disediakan oleh aplikasi. Fitur tersebut berada tepat di atas bagian *maps screen* seperti pada Gambar 4.17. Setelah *user*

menekan tombol tersebut maka tampilan akan dialihkan ke *search screen* seperti yang ditunjukkan pada Gambar 4.18. Pada tampilan ini *user* dapat mengetikkan kata kunci nama dari hotel syari'ah atau alamat dari hotel syari'ah tersebut.



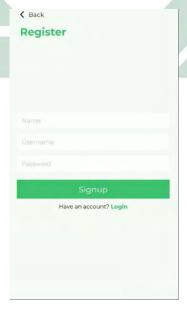
Gambar 4.15 Search Screen

Apabila seorang *user* ingin menjadi *user* surveyor, *user* tersebut diharuskan untuk melakukan *login* pada aplikasi. Tombol *login* terletak pada sebelah kanan tombol *search* seperti pada Gambar 4.19. Setelah itu *user* memasukkan *username* dan *password* pada *form login* seperti yang ditunjukkan pada Gambar 4.20. Apabila *username* dan *password* benar maka tampilan akan dialihkan ke *maps screen* surveyor yang memiliki navigasi *bottom tab*. Ketika *user* salah memasukkan *username* dan *password* maka akan muncul sebuah *popup* notifikasi peringatan errors.



Gambar 4.16 Login Screen

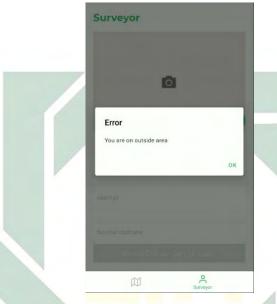
Untuk *user* yang tidak memiliki akun, *user* tersebut dapat melakukan registrasi dengan menekan tulisan *signup* seperti yang ditunjukkan pada Gambar 4.30. Setelah itu *user* diharuskan mengisi *form* yang meliputi *username*, *password* dan nama dari user tersebut seperti pada Gambar 4.31. Setelah *user* berhasil melakukan registrasi maka akan muncul *popup* notifikasi bahwa akun berhasil dibuat dan apabila *username* telah terdaftarkan maka akan muncul *popup* notifikasi error bahwa *username* tersebut telah terdaftarkan.



Gambar 4.17 Register Screen

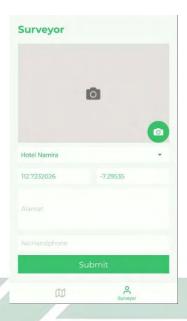
Apabila *user* ingin kembali ke halaman *login user* dapat menekan tulisan login yang berada di bawah *button signup* atau dengan menekan tombol *back*

yang berada di atas tulisan *register*. Setelah berhasil mendaftarkan akun dan melakukan *login*, *user* tersebut telah menjadi *user* surveyor. Apabila *user* ingin melakukan pengisian data hotel syari'ah, *user* dapat menekan *icon-user* pada navigasi *bottom tab*. Apabila *user* berada pada salah satu hotel yang jarak radius nya tidak lebih dari 50m, apabila *user* diluar dari radius tersebut, maka ketika *user* menekan *icon-user* aplikasi langsung akan memunculkan *popup* notifikasi "you are on outside area" seperti pada Gambar 4.32.



Gambar 4.18 Surveyor Outside Area

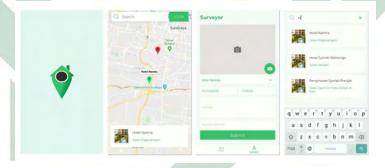
Apabila *user* telah berada didalam radius hotel syari'ah yang telah ditenukan *user* surveyor dapat melakukan pengisian data pada *form* yang meliputi foto hotel syari'ah, *dropdown* hotel syari'ah, *longitude*, *latitude*, alamat dan no handphone. Pada *text field longitude* dan *latitude user* tidak perlu mengisi *text field* tersebut secara manual karena sudah terisi secara otomatis oleh aplikasi tersebut, *user* hanya perlu melengkapi *text field* lain yang telah disebutkan di atas. Seperti yang ditunjukkan pada Gambar 4.33.



Gambar 4.19 Form Surveyor

4.8 Implementasi

Aplikasi pada sisi *front-end* dikembangkan dengan menggunakan *framework react native*. Berikut adalah beberapa gambar aplikasi.



Gambar 4.20 Front End Aplikasi

Setelah membuat aplikasi pada sisi *front-end* berikutnya yaitu membuat *back-end* untuk aplikasi dengan menggunakan *framework* NodeJS.

4.8.1 PBFT Blockchain

Pada sisi *backend* terdapat *function get request* untuk mendapatkan status *server primary* apabila memberikan nilai *true* maka *server* dapat diakses dan apabila bernilai *false* aplikasi tidak dapat digunakan. Berikut *function request* pada *backend*.

Pseudocode Request Blockchain

```
1. request = () {
2. ping('localhost', function(available)){
```

```
3.     if(available){
4.        return true
5.     }else{
6.        return false
7.     }
8.     }
9. }
```

Setelah proses pengecekan pada *function request*, selanjutnya membuat *function preprepare* untuk melakukan penyimpanan data ke *server* dari data yang dikirim oleh *user* melalui aplikasi dari sisi *front-end*. Function *request* berjalan akan melakukan eksekusi apabila telah bernilai 3 *f* .

Pseudocode Prepare Blockchain

```
1. prepare = () {
    data = {...}
2.
   hosts = {...}
3.
4.
    calcPar = hosts.length%3
5.
    calcLength = Math.floor(hosts.length/3)
     if(par == 1){
6.
7.
      return false
8.
    }else{
9.
      return true
10.
11. }
```

Setelah proses *function post prepare*, didalam *function* ini melakukan 2 proses sekaligus yaitu proses *commit* dan *reply*.

Pseudocode Commit dan Reply Blockchain

```
1. prepare = () {
   selectQuery("select author from db", function(result)) {
2.
3.
4.
      if(result == author){
5.
       selectQuery("select chain from blockchain", function(result)){
        if(result == 0){
6.
7.
        do insert
8.
        }else{
9.
         //commit
10.
         selectQuery(replace JSON), function (length)){
11.
           if(length > 7){
12.
            selectQuery(replaceJSON)
13.
            update blockchain
14.
            return "success"
15.
          }else{
16.
            selectQuery(selectChain)
17.
             update blockchain
             return "success"
18.
19.
           }
20.
          }
21.
        }
22.
       }
23.
      }else{
24.
       return false
25. }
26.}
```

Proses *commit* melakukan penyimpanan data pada *database blockchain*. Yang kemudian memberikkan *return reply* untuk *front-end* berupa nilai *true*.

4.8.2 Validasi Mid-Point Circle

Validasi *mid-point circle* berperan sebagai *smart-contract* pada *blockchain* yang digunakan untuk mengecek posisi *user* terletak pada radius 50m dari hotel syari'ah yang ditentukan. Berikut pembuatan *function validation mid-point blockchain*.

Pseudocode Mid-Point Circle

```
1. dataFromDbCenter (result) = {
2.
    currentLat = ...
3. currentLng = ...
4. midPointVal = []
5. resLength = result.length
6. doLooping(i = 0; i < resLength; i++) {
7. midpoint = isPointWithRadius(
8.
    latitude: currentLat, longitude: currentLng
9.
   ), 50
10.
11. if(midPointVal) {
12.
     return true
13. }else{
14.
     return false
15. ·
16.
```

Apabila posisi *user* berada pada radius yang ditentukan maka akan mengembalikan nilai *false*. Apabila posisi *user* berada pada didalam radius maka akan mengembalikkan nilai *true*, nilai ini berguna untuk aplikasi pada sisi *front-end* agar ditampilkan popup notifikasi menunjukkan bahwa *user* berada diluar radius.

4.8.3 Hashing SHA-256

Teknik ini digunakan untuk mengenkripsi *hash*, serta *nexthash* pada *blockchain. Hashing SHA-256* digunakan pada *author* yang didapat dari proses validasi *login* dan tanggal ketika data tersebut disimpan. Berikut penggunaan *hashing sha-256*.

Pseudocode *Hashing* SHA-256

```
    data = ...
    hashFirst = dohash(data)
    nextHash = doHash(data)
    insertHash = ("INSERT into database", hashFirst+nextHash)
    Setelah itu data akan tersimpan didalam blockchain dengan tipe data longtext.
    Berikut adalah contoh hash yang tersimpan didalam blockchain.
```

```
    hash: "0ad134471ed8fdb119654bceadea6b57583ea761167996ecff8ecc27f7ae82c6"
    prevhash: "0"
    author: "Ainur Rahman"
    date: "2020-10-12 10:49:07"
    hash: "98e443d0cf04e8a49a19e3720e63f29d3aa1559797aae03b7404ad80affbc7eb"
    prevhash: "0ad134471ed8fdb119654bceadea6b57583ea761167996ecff8ecc27f7ae82c6"
    author: "Ainur Rahman"
    date: "2020-10-12 10:49:07"
```

Gambar 4.21 Contoh Data Hashing-SHA256

4.8.4 AES Encryption

Teknik ini digunakan untuk mengenkripsi data hotel syari'ah. Data tersebut meliput nama hotel syari'ah, alamat hotel syari'ah, latitude hotel, longitude hotel, dan nomer hp. Berikut cara penggunaan AES Enkripsi pada blockchain.

Pseudocode AES Encryption

```
1. data = ...
2. doEncrypt = {CryptoJS.AES.Encrypt(data ...)}
3. doUpdateChain = "UPDATE blockchain set chain"
4. doQuery(doUpdateChain) {
5. return true
6. }
```

Teknik AES Enkripsi dapat didekripsi agar data asli dapat ditampilkan melalui aplikasi. Berikut cara melakukan dekripsi data pada sisi *front end* aplikasi.

Pseudocode AES Decryption

```
1. dataJSON = JSON.data(...)
2. doReplaceCharacter = dataJSON.replace(...)
3. arrLength = (...)
4. emptyArr = []
5. doLoop() {
6. trim = doReplaceCharacter.trim()
7. doDecrypt = CryptoJS.AES.decrypt()
8. }
9. finalRess = doJSOnParse{map(dataJSON)}
10. setData(finalRess)
```

4.8.5 String Match

Algoritma ini digunakan untuk mencokkan data pada 4 server yang saling terhubung dalam jaringan blockchain. Hash pada primary node menjadi pembanding utama terahdap hash node lain yang saling terhubung. Apabila 1 hash node diganti, maka data masih dapat diakses, akan tetapi jika 2 hash node

diganti data tidak dapat diakses. Nilai tersebut sesuai dengan *fault tolerance* yang dijelaskan pada poin 3.1.7.1. Berikut penggunaan dari Algoritma *String Match* JSON-DIFF.

Pseudocode String Match JSON-DIFF

```
1. reqNodeSatu = '...'
2. reqNodeDua = \...'
3. reqNodeTiga = \...'
4. emptyArr = []
5. doLooping(){
6. doDiff = diff(reqNodeSatu)
7. if(doDiff == undefined) {
8. emptyArr.push(true)
9. }else{
10. emptyArr.push(false)
11. \
12. ...
13. filteredArr = emptyArr()
14. foundTrue = isFoundMoreThanTwo () {
15. if(filteredArr \geq 2) {
16.
     return false
17. }else{
18.
     return true
19.
20. }
21. if(foundTrue){
22.
    return data
23. }else{
24.
      feturn false
25. }
26. }
```

4.9 Evaluasi Aplikasi

Evaluasi dilakukan untuk menguji aplikasi ini berdasarkan tindakan yang telah dilakukan. Pengujian ini terdiri dari 5 tahap yaitu pengujian *test case scenario blackbox* aplikasi, pengujian konsensus, pengujian *network-latency*, pengujian *mid-point validation*, dan pengujian integritas data.

Pengujian yang pertama yaitu melakukan pengujian *test case scenario blackbox* pada aplikasi yang telah dibuat. Pengujian ini dilakukan dengan tujuan agar aplikasi yang dibuat dapat berjalan dengan lancar sesuai dengan skenario pengujian *test case scenario blackbox* yang telah dibuat pada Tabel 3.2. Pengujian ini dilakukan pada 3 jenis *Operating System Android* yaitu Android 7, 8 dan 9. Berikut hasil yang dimuat pada Tabel 4.1, Tabel 4.2, dan Tabel 4.3.

Tabel 4.1 Pengujian *Blackbox* Aplikasi Android 7

			Android	! 7		
No	Test Case Description	Untitled Question	Pre- Condition	Test Step	Expected Result	Results
1	Check GPS	Valid	Kondisi services GPS dalam keadaan mati	1. Buka Aplikasi 2. Tunggu Splash Screen berhasil dimuat	Tampilan dialihkan ke halaman map	Pass
1	Services	Invalid	Kondisi services GPS dalam keadaan mati	1. Buka Aplikasi 2. Tunggu Splash Screen berhasil dimuat	Menampilkan check services GPS	Pass
2	Register User	Valid	1. Kondisi services GPS dalam keadaan menyala 2. Menampilkan halaman peta	1. Menekan tombol login pada sebelah kanan button search 2. Klik tombol SignUp 3. Mengisi email sebagai username pengguna (surveyor) pada form 4. Mengisi password pengguna pada form	Muncul popup akun berhasil dibuat	Pass
		Invalid	1. Kondisi services GPS dalam keadaan menyala 2.	1. Menekan tombol login pada sebelah kanan button search	Muncul popup error akun gagal dibuat	Pass

			Menampilkan halaman peta	2. Klik tombol SignUp 3. Mengisi email sebagai username pengguna (surveyor) pada form 4. Mengisi password pengguna		
3	Login User	Valid	1. Kondisi services GPS dalam keadaan menyala 2. Menampilkan halaman peta	pada form 1. Menekan tombol login pada sebelah kanan button search 2. Mengisi email sebagai username pengguna (surveyor) pada form 3. Mengisi password pengguna pada form	Tampilan dialihkan ke halaman map surveyor	Pass
		Invalid	1. Kondisi services GPS dalam keadaan menyala 2. Menampilkan halaman peta	1. Menekan tombol login pada sebelah kanan button search 2. Mengisi email sebagai username pengguna (surveyor) pada form 3. Mengisi	Muncul popup error username atau password yang dimasukkan salah	Pass

		1	1	1 .	I	
				password		
				pengguna		
				pada form		
				1. Buka		
				Aplikasi		
				2. Ketika		
				aplikasi		
				tersebut		
				dibuka		
				lokasi dari		
			1 17 1: . :			
			1. Kondisi	user telah		
			services GPS	dideteksi		
			dalam	oleh	User	
		- A	keadaan	aplikasi	Surveyor	
		Valid	menyala	3. User	dapat	Pass
		vand	2. Berada	Surveyor	melakukan	1 ass
			pada radius	melakukan	penginputan	
			hotel syari'ah	penginputan	data	
		100	yang telah	data hotel		
		1	ditentukan	syari'ah		
				dengan cara		
				menekan		
				bottom tab		
1	Check Mid			pa <mark>da</mark>		
	Point			halaman		
4	Circle			peta		
	Hotel			surveyor		
	110101			-		
				1. Buka		
				Aplikasi		
			7/	2. Ketika		
				aplikasi		
				tersebut		
			1. Kondisi	dibuka		
			services GPS	lokasi dari	Muncul	
			dalam	user telah	popup error,	
			keadaan	dideteksi	user surveyor	
		T 11. 1	menyala	oleh	berada pada	D
		Invalid	2. Berada	aplikasi	diluar radius	Pass
			pada radius	3. User	hotel syari'ah	
			hotel syari'ah	Surveyor	yang telah	
			yang telah	melakukan	ditentukan	
			ditentukan	penginputan		
				data hotel		
				syari'ah		
				dengan cara		
				menekan		
				bottom tab		
1				ponom tab	l	

				pada halaman peta surveyor		
	Check	Valid	1. Kondisi services GPS dalam keadaan menyala 2. Aplikasi melakukan ping ke blockchain primary	1. Buka Aplikasi 2. Tunggu aplikasi melakukan request	Aplikasi dapat digunakan seperti biasa	Pass
5	Request Blockchain	Invalid	1. Kondisi services GPS dalam keadaan menyala 2. Aplikasi melakukan ping ke blockchain primary	1. Buka Aplikasi 2. Tunggu aplikasi melakukan request	Muncul popup error bahwa aplikasi tidak dapat digunakan sampai blockchain primary kembali dalam kondisi aktif	Pass

Berikut adalah hasil pengujian *blackbox* pada sistem android 8 yang tertera pada Tabel 4.2.

Tabel 4.2 Pengujian Blackbox Aplikasi Android 8

			Android	18		
No	Test Case Description	Untitled Question		Test Step	Expected Result	Results
1	Check GPS Services	Valid	Kondisi services GPS dalam keadaan mati	1. Buka Aplikasi 2. Tunggu Splash Screen berhasil dimuat	Tampilan dialihkan ke halaman map	Pass
		Invalid	Kondisi services GPS dalam keadaan mati	1. Buka Aplikasi 2. Tunggu Splash Screen	Menampilkan check services GPS	Pass

				111		
				berhasil dimuat		
2	Register User	Valid	1. Kondisi services GPS dalam keadaan menyala 2. Menampilkan halaman peta	1. Menekan tombol login pada sebelah kanan button search 2. Klik tombol SignUp 3. Mengisi email sebagai username pengguna (surveyor) pada form 4. Mengisi password pengguna pada form	Muncul popup akun berhasil dibuat	Pass
		Invalid	1. Kondisi services GPS dalam keadaan menyala 2. Menampilkan halaman peta	1. Menekan tombol login pada sebelah kanan button search 2. Klik tombol SignUp 3. Mengisi email sebagai username pengguna (surveyor) pada form 4. Mengisi password pengguna pada form	Muncul popup error akun gagal dibuat	Pass
3	Login User	Valid	1. Kondisi services GPS dalam	1. Menekan tombol login pada	Tampilan dialihkan ke	Pass

1. Buka Aplikasi 1. Kondisi 2. Ketika	
keadaan menyala 2. Menampilkan halaman peta Nemampilkan halaman peta Nemampilkan search 2. Mengisi email sebagai username pengguna (surveyor) pada form 3. Mengisi password pengguna pada form 1. Menekan tombol login pada sebelah kanan button search 2. Mengisi services GPS dalam keadaan menyala 2. Menampilkan halaman peta Invalid I	Pass

				penginputan data hotel syari'ah dengan cara menekan bottom tab pada halaman peta surveyor		
		Invalid	1. Kondisi services GPS dalam keadaan menyala 2. Berada pada radius hotel syari'ah yang telah ditentukan	1. Buka Aplikasi 2. Ketika aplikasi tersebut dibuka lokasi dari user telah dideteksi oleh aplikasi 3. User Surveyor melakukan penginputan data hotel syari'ah dengan cara menekan bottom tab	Muncul popup error, user surveyor berada pada diluar radius hotel syari'ah yang telah ditentukan	Pass
				pada halaman peta surveyor		
5	Check Request Blockchain	Valid	1. Kondisi services GPS dalam keadaan menyala 2. Aplikasi melakukan ping ke blockchain primary	1. Buka Aplikasi 2. Tunggu aplikasi melakukan request	Aplikasi dapat digunakan seperti biasa	Pass
		Invalid	1. Kondisi services GPS dalam	1. Buka Aplikasi 2. Tunggu	Muncul popup error bahwa	Pass

	keadaan	aplikasi	aplikasi tidak	
	menyala	melakukan	dapat	
	2. Aplikasi	request	digunakan	
	melakukan		sampai	
	ping ke		blockchain	
	blockchain		primary	
	primary		kembali	
			dalam	
			kondisi aktif	

Berikut adalah hasil pengujian *blackbox* pada sistem android 9 yang tertera pada Tabel 4.3.

Tabel 4.3 Pengujian Blackbox Aplikasi Android 9

	Android 9								
No	Test Case Description	Untitled Question	Tost Ston		Expected Result	Results			
1	Check GPS	Valid	Kondisi services GPS dalam keadaan mati	1. Buka Aplikasi 2. Tunggu Splash Screen berhasil dimuat	Tampilan dialihkan ke halaman map	Pass			
1	Services	Invalid	Kondisi services GPS dalam keadaan mati	1. Buka Aplikasi 2. Tunggu Splash Screen berhasil dimuat	Menampilkan check services GPS	Pass			
2	Register User	Valid	1. Kondisi services GPS dalam keadaan menyala 2. Menampilkan halaman peta	1. Menekan tombol login pada sebelah kanan button search 2. Klik tombol SignUp 3. Mengisi email sebagai username pengguna	Muncul popup akun berhasil dibuat	Pass			

				(surveyor) pada form 4. Mengisi password pengguna pada form		
		Invalid	1. Kondisi services GPS dalam keadaan menyala 2. Menampilkan halaman peta	1. Menekan tombol login pada sebelah kanan button search 2. Klik tombol SignUp 3. Mengisi email sebagai username pengguna (surveyor) pada form 4. Mengisi	Muncul popup error akun gagal dibuat	Pass
				password pengguna pada form		
3	Login User	Valid	1. Kondisi services GPS dalam keadaan menyala 2. Menampilkan halaman peta	1. Menekan tombol login pada sebelah kanan button search 2. Mengisi email sebagai username pengguna (surveyor) pada form	Tampilan dialihkan ke halaman map surveyor	Pass

		<u> </u>				
				3. Mengisi		
				password		
				pengguna		
				pada form		
				1. Menekan		
				tombol		
				login pada		
				sebelah		
				kanan		
			1. Kondisi	button	Muncul	
			services GPS	search	popup error	
			dalam	2. Mengisi	username	
		Invalid	keadaan	email	atau	D
		invalid	menyala	sebagai	password	Pass
		7//	2.	username	yang	
		1 /	Menampilkan		dimasukkan	
		6	halaman peta	(surveyor)	salah	
		1	naraman pera	pada form	Salai	
		100		3. Mengisi		
				password		
	1			1		
				pengguna		
				p <mark>ada</mark> form		
				1. <mark>Bu</mark> ka		2
				Ap <mark>lik</mark> asi		
				2. Ketika	1	
				ap <mark>lika</mark> si		
				tersebut		
				dibuka		
				lokasi dari		
			1. Kondisi	user telah		
			services GPS	dideteksi		
			dalam	oleh	User	
	Check Mid		keadaan	aplikasi	Surveyor	
	Point			3. User	•	
4		Valid	menyala		dapat	Pass
	Circle		2. Berada	Surveyor	melakukan	
	Hotel		pada radius	melakukan	penginputan	
			hotel syari'ah	penginputan	data	
			yang telah	data hotel		
			ditentukan	syari'ah		
				dengan cara		
				menekan		
				bottom tab		
				pada		
				halaman		
				peta		
				surveyor		
				isui ve voi		

5	Check Request Blockchain	Invalid	2. Aplikasi melakukan ping ke blockchain primary 1. Kondisi services GPS dalam keadaan menyala 2. Aplikasi melakukan	aplikasi melakukan request 1. Buka Aplikasi 2. Tunggu aplikasi melakukan	digunakan seperti biasa Muncul popup error bahwa aplikasi tidak dapat digunakan sampai blockchain	Pass
		Valid	1. Kondisi services GPS dalam keadaan menyala	syari'ah dengan cara menekan bottom tab pada halaman peta surveyor 1. Buka Aplikasi 2. Tunggu	Aplikasi dapat	Pass
		Invalid	1. Kondisi services GPS dalam keadaan menyala 2. Berada pada radius hotel syari'ah	1. Buka Aplikasi 2. Ketika aplikasi tersebut dibuka lokasi dari user telah dideteksi oleh aplikasi 3. User Surveyor melakukan penginputan data hotel	Muncul popup error, user surveyor berada pada diluar radius hotel syari'ah yang telah ditentukan	Pass

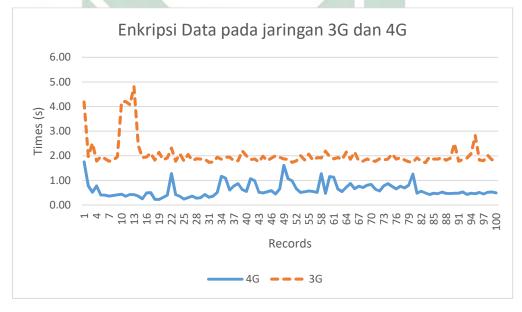
Hasil pada Tabel 4.1, Tabel 4.2, dan Tabel 4.3 menunjukkan bahwa dalam 5 *test case scenario* pada masing-masing sistem operasi ,aplikasi dapat berjalan sesuai dengan *test case* yang diberikan.

Pengujian yang selanjutnya yaitu pengujian konsensus PBFT dengan menggunakan 1 *primary node* dan 3 *node* yang terhubung dalam jaringan *distributed database*. Berikut data yang disajikan pada Tabel 4.5.

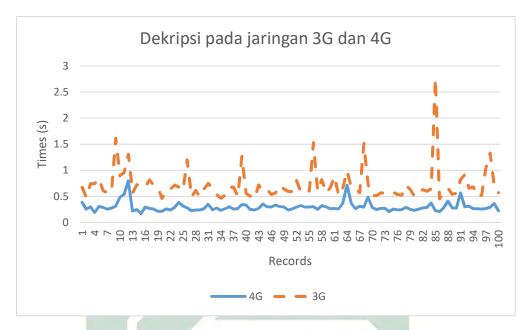
Tabel 4.4 Pengujian Konsensus PBFT

No	Primary	Node 1	Node 2	Node 3	Expected	Actual
110	1 Timary	Node 1	Node 2	Node 3	Result	Result
1	Connect	Connect	Connect	Connect	Terbuat	Terbuat
2	Connect	Connect	Connect	Not	Terbuat	Terbuat
				Connected		
3	Connect	Connect	Not	Not	Gagal	Gagal
			Connected	Connected	Terbuat	Terbuat
4	Connect	Not	Not	Not	Gagal	Gagal
		Conn <mark>e</mark> cte <mark>d</mark>	Connect <mark>ed</mark>	Connected	Terbuat	Terbuat

Kemudian pengujian selanjutnya yaitu melihat *latency* jaringan seluler ketika menginputkan (enkripsi) data hotel syari'ah dan mengambil data hotel syari'ah (dekripsi) pada *blockchain* PBFT saat menggunakan aplikasi, indikator jaringan dibagi menjadi beberapa bagian diantaranya 4G dan 3G yang disajikan pada Gambar 4.22 dan Gambar 4.23.



Gambar 4.22 Enkripsi Data Network Latency 3G dan 4G



Gambar 4.23 Dekripsi Data Network Latency 3G dan 4G

Hasil dari pengujian *network latency* enkripsi dan dekripsi pada setiap jaringan 3G dan 4G akan dilakukan penghitungan rata-rata untuk mendapatkan pilihan jaringan terbaik ketika menggunakan aplikasi ini.

Tabel 4.5 Rata-rata Network Latency (seconds)

	Enk <mark>ri</mark> psi	Dekripsi
3G	2.0 <mark>47</mark> s	0.414s
4G	0.617s	0.297s

Pada Tabel 4.5 menunjukkan bahwa jaringan 4G memiliki 231.77% lebih cepat daripada jaringan 3G pada saat melakukan enkripsi sedangkan pada saat dekripsi jaringan 4G memiliki 39.39% lebih cepat daripada jaringan 3G. Setelah melakukan penghitungan dan perbandingan pada *network latency*.

Pengujian berikutnya yaitu *mid-point validation*, pengujian ini dilakukan untuk memvalidasi apakah pengguna berada radius 50m hotel syari'ah yang telah ditentukan. Pengujian ini menghasilkan data mentah berupa titik koordinat pengguna, hasil serta jarak dari titik yang telah ditentukan, sesuai yang ditunjukkan pada Tabel 4.6, Tabel 4.7 dan Tabel 4.8

Tabel 4.6 Raw Data *Hasil Pengujian* Mid-Point Validation

No	Nome Hotel	Koor	Koordinat		Actual	Ionole/ma	
No Nama Hotel		Latitude	Longitude	Result	Result	Jarak/m	
1	Penginapan Syariah Prenjak	-7.23747	112.7365	TRUE	1	12.43257816	
2	Penginapan Syariah Prenjak	-7.2374686	112.7364498	TRUE	1	13.08556509	

3	Penginapan Syariah Prenjak	-7.2374693	112.7364441	TRUE	1	13.52911469
4	Penginapan Syariah Prenjak	-7.2374693	112.7364441	TRUE	1	13.52911469
5	Penginapan Syariah Prenjak	-7.2374693	112.7364441	TRUE	1	13.52911469
6	Penginapan Syariah Prenjak	-7.2374693	112.7364441	TRUE	1	13.52911469
7	Penginapan Syariah Prenjak	-7.2374693	112.7364441	TRUE	1	13.52911469
8	Penginapan Syariah Prenjak	-7.2374693	112.7364441	TRUE	1	13.52911469
9	Penginapan Syariah Prenjak	-7.2374693	112.7364441	TRUE	1	13.52911469
10	Penginapan Syariah Prenjak	-7.2374693	112.7364441	TRUE	1	13.52911469
11	Penginapan Syariah Prenjak	-7.2374693	112.7364441	TRUE	1	13.52911469
12	Penginapan Syariah Prenjak	-7.2374693	112.7364441	TRUE	1	13.52911469
13	Penginapan Syariah Prenjak	-7.2374693	112.7364441	TRUE	1	13.52911469
14	Penginapan Syariah Prenjak	-7.2374693	112.7364441	TRUE	1	13.52911469
15	Penginapan Syariah Prenjak	-7.2374693	112.7364441	TRUE	1	13.52911469
16	Penginapan Syariah Prenjak	-7.2303642	112.7408009	FALSE	0	912.3542305
17	Penginapan Syariah Prenjak	-7.2303681	112.7408273	FALSE	0	913.4976537
18	Penginapan Syariah Prenjak	-7.2303721	112.7408535	FALSE	0	914.6281033
19	Penginapan Syariah Prenjak	- <mark>7</mark> .2303718	112 <mark>.740</mark> 8382	FALSE	0	913.7736548
20	Penginapan Syariah Prenjak	-7 .23 <mark>0372</mark> 2	112.7408348	FALSE	0	913.5397702
21	Penginapan Syariah Prenjak	-7.2297834	112.7414626	FALSE	0	1005.9345041
22	Penginapan Syariah Prenjak	-7.2297734	112.7414554	FALSE	0	1006.4389393
23	Penginapan Syariah Prenjak	-7.3376265	112.7141247	FALSE	0	11430.7626357
24	Penginapan Syariah Prenjak	-7.3376352	112.714121	FALSE	0	1431.7965623
25	Penginapan Syariah Prenjak	-7.3376459	112.7141073	FALSE	0	11433.2868176
D 1	Tabal 4.7 mammalan		1	1:1		J. 1 4. 1 1 4. 1

Pada Tabel 4.7 merupakan *raw data* pengujian *mid-point validation* pada hotel hotel syari'ah walisongo

Tabel 4.7 Raw Data Hasil Pengujian Mid-Point Validation

	Hotel Syariah Walisongo					
26	Surabaya	-7.2374698	112.7364623	FALSE	0	1029.867667
	Hotel Syariah Walisongo					
27	Surabaya	-7.2374686	112.7364498	FALSE	0	1030.518608
	Hotel Syariah Walisongo					
28	Surabaya	-7.3376265	112.7141247	FALSE	0	12385.84634
	Hotel Syariah Walisongo					
29	Surabaya	-7.3376352	112.714121	FALSE	0	12386.8854
20	Hotel Syariah Walisongo	5.005 (450	110 51 11050	DATE		10000 41000
30	Surabaya	-7.3376459	112.7141073	FALSE	0	12388.41092
	Hotel Syariah Walisongo					
31	Surabaya	-7.2374693	112.7364441	FALSE	0	1030.931584
	Hotel Syariah Walisongo					
32	Surabaya	-7.2374693	112.7364441	FALSE	0	1030.931584
	Hotel Syariah Walisongo					
33	Surabaya	-7.2374693	112.7364441	FALSE	0	1030.931584
	Hotel Syariah Walisongo					
34	Surabaya	-7.2374693	112.7364441	FALSE	0	1030.931584
	Hotel Syariah Walisongo					
35	Surabaya	-7.2374693	112.7364441	FALSE	0	1030.931584
	Hotel Syariah Walisongo					
36	Surabaya	-7.2374693	112.7364441	FALSE	0	1030.931584
	Hotel Syariah Walisongo					
37	Surabaya	-7.2303718	112.7408382	FALSE	0	109.3200676

	Hotel Syariah Walisongo		I			
38	Surabaya	-7.2303642	112.7408009	FALSE	0	112.0364956
	Hotel Syariah Walisongo					
39	Surabaya	-7.2303681	112.7408273	FALSE	0	110.0017825
	Hotel Syariah Walisongo					
40	Surabaya	-7.2303721	112.7408535	FALSE	0	108.0333054
	Hotel Syariah Walisongo					
41	Surabaya	-7.2297734	112.7414554	TRUE	1	16.9477146
	Hotel Syariah Walisongo					
42	Surabaya	-7.2297734	112.7414554	TRUE	1	16.9477146
	Hotel Syariah Walisongo					
43	Surabaya	-7.2297734	112.7414554	TRUE	1	16.9477146
	Hotel Syariah Walisongo					
44	Surabaya	-7.2297734	112.7414554	TRUE	1	16.9477146
	Hotel Syariah Walisongo					
45	Surabaya	-7.2298043	112.7414995	TRUE	1	13.19062881
	Hotel Syariah Walisongo					
46	Surabaya	-7.2298000	112.7415018	TRUE	1	12.76129565
	Hotel Syariah Walisongo	<i>A</i>				
47	Surabaya	-7.2298063	112.7414482	TRUE	1	18.54453883
	Hotel Syariah Walisongo	6				
48	Surabaya	-7.2298074	112.7414446	TRUE	1	18.96044197
	Hotel Syariah Walisongo					
49	Surabaya	-7.2297834	112.7414626	TRUE	1	16.34625741
l	Hotel Syariah Walisongo	3/				
50	Surabaya	-7.2297734	112.7414554	TRUE	1	16.9477146

Pada Tabel 4.8 merupakan raw data hasil pengujian mid-point validation pada

Grand Kalimas Hotel Syari'ah

Tabel 4.8 Raw Data Hasil Pengujian Mid-Point Validation

	Grand Kalimas Hotel					
51	Syariah	-7.2374698	112.7364623	FALSE	0	929.1453251
	Grand Kalimas Hotel	71=071000				1
52	Syariah	-7 <mark>.237</mark> 4686	112.7364498	FALSE	0	929.7246854
	Grand Kalimas Hotel					7
53	Syariah	-7.2298043	112.7414995	FALSE	0	102.2891941
	Grand Kalimas Hotel					
54	Syariah	-7.2298000	112.7415018	FALSE	0	102.7455965
	Grand Kalimas Hotel					
55	Syariah	-7.2298063	112.7414482	FALSE	0	97.27598325
	Grand Kalimas Hotel					
56	Syariah	-7.2298074	112.7414446	FALSE	0	96.87195412
	Grand Kalimas Hotel					
57	Syariah	-7.2297834	112.7414626	FALSE	0	99.95824305
	Grand Kalimas Hotel					
58	Syariah	-7.2297734	112.7414554	FALSE	0	99.87487229
	Grand Kalimas Hotel				_	
59	Syariah	-7.2297734	112.7414554	FALSE	0	99.87487229
	Grand Kalimas Hotel	# 22# C2 C#		FATOR		10005 00554
60	Syariah	-7.3376265	112.7141247	FALSE	0	12307.32774
	Grand Kalimas Hotel	7.2276252	110 71 4101	EALGE		12200 26560
61	Syariah	-7.3376352	112.714121	FALSE	0	12308.36568
62	Grand Kalimas Hotel	7.2274602	110 7264441	EALCE		020 100222
62	Syariah Grand Kalimas Hotel	-7.2374693	112.7364441	FALSE	0	930.109322
63	Syariah	-7.2374693	112.7364441	FALSE	0	930.109322
0.5	Grand Kalimas Hotel	-7.2374093	112./304441	FALSE	U	930.109322
64	Syariah	-7.2374693	112.7364441	FALSE	0	930.109322
04	Grand Kalimas Hotel	-7.2374093	112./304441	TALSE	U	930.109322
65	Syariah	-7.2374693	112.7364441	FALSE	0	930.109322
- 05	Grand Kalimas Hotel	7.237 1073	112.7501111	TILDE		730.107522
66	Syariah	-7.2303642	112.7408009	TRUE	1	17.1237067
	Grand Kalimas Hotel					
67	Syariah	-7.2303681	112.7408273	TRUE	1	19.5592322
	Grand Kalimas Hotel	-				
68	Syariah	-7.2303721	112.7408535	TRUE	1	22.10850258
	Grand Kalimas Hotel					
69	Syariah	-7.2303718	112.7408382	TRUE	1	20.74934877

	Grand Kalimas Hotel					
70	Syariah	-7.2303722	112.7408348	TRUE	1	20.48716907
	Grand Kalimas Hotel					
71	Syariah	-7.2303722	112.7408348	TRUE	1	20.48716907
	Grand Kalimas Hotel					
72	Syariah	-7.2303722	112.7408348	TRUE	1	20.48716907
	Grand Kalimas Hotel					
73	Syariah	-7.2303722	112.7408348	TRUE	1	20.48716907
	Grand Kalimas Hotel					
74	Syariah	-7.2303722	112.7408348	TRUE	1	20.48716907
	Grand Kalimas Hotel					
75	Syariah	-7.2303722	112.7408348	TRUE	1	20.48716907

Pada Tabel 4.10 merupakan raw data hasil pengujian mid-point validation pada

Namira Syariah Hotel Surabaya

Tabel 4.9 Raw Data Hasil Pengujian Mid-Point Validation

7.0	N . G . 1 W . 1G . 1	7.2274622	110 72 (4622	EALCE		11451 20021
76	Namira Syariah Hotel Surabaya	-7.2374698	112.7364623	FALSE	0	11451.39021
77	Namira Syariah Hotel Surabaya	-7.2374686	112.7364498	FALSE	0	11451.21987
78	Namira Syariah Hotel Surabaya	-7.2374693	112.7364441	FALSE	0	11451.00675
79	Namira Syariah Hotel Surabaya	-7.2298063	112.7414482	FALSE	0	12408.94322
80	Namira Syariah Hotel Surabaya	-7.2298074	112.7414446	FALSE	0	12408.72694
81	Namira Syariah Hotel Surabaya	-7.2297834	112.7414626	FALSE	0	12411.80482
82	Namira Syariah Hotel Surabaya	- <mark>7.2</mark> 303722	112.7408348	FALSE	0	12331.35006
83	Namira Syariah Hotel Surabaya	-7.2298043	112.7414995	FALSE	0	12410.55077
84	Namira Syariah Hotel Surabaya	-7.2298000	112.7415018	FALSE	0	12411.0772
85	Namira Syariah Hotel Surabaya	-7.23746 93	112.7364441	FALSE	0	11451.00675
86	Namira Syariah Hotel Surabaya	<mark>-7.2</mark> 303642	112.7408009	FALSE	0	12331.31075
87	Namira Syariah Hotel Surabaya	-7.2 303681	112.7408273	FALSE	0	12331.59297
88	Namira Syariah Hotel Surabaya	-7.2303721	112.7408535	FALSE	0	12331.85975
89	Namira Syariah Hotel Surabaya	-7.2303718	112.7408382	FALSE	0	12331.48395
90	Namira Syariah Hotel Surabaya	-7.2303722	112.7408348	FALSE	0	12331.35006
91	Namira Syariah Hotel Surabaya	-7.3376459	112.7141073	TRUE	1	36.5961495
92	Namira Syariah Hotel Surabaya	-7.3376459	112.7141073	TRUE	1	36.5961495
93	Namira Syariah Hotel Surabaya	-7.3376459	112.7141073	TRUE	1	36.5961495
94	Namira Syariah Hotel Surabaya	-7.3376459	112.7141073	TRUE	1	36.5961495
95	Namira Syariah Hotel Surabaya	-7.3376459	112.7141073	TRUE	1	36.5961495
96	Namira Syariah Hotel Surabaya	-7.3376459	112.7141073	TRUE	1	36.5961495
97	Namira Syariah Hotel Surabaya	-7.3376459	112.7141073	TRUE	1	36.5961495
98	Namira Syariah Hotel Surabaya	-7.3376459	112.7141073	TRUE	1	36.5961495
99	Namira Syariah Hotel Surabaya	-7.3376265	112.7141247	TRUE	1	39.47148584
100	Namira Syariah Hotel Surabaya	-7.3376352	112.714121	TRUE	1	38.51686879
						•

Data mentah yang telah dipaparkan pada Tabel 4.9 di atas merupakan hasil pengecekan *mid-point validation* dengan radius masing-masing hotel yaitu 50m. Apabila *user* berada pada titik koordinat yang memiliki nilai jarak di atas 50m maka nilai yang diharapkan adalah *false* dan apabila *user* berada

pada titik koordinat yang memiliki jarak dibawah 50m maka nilai yang diharapkan adalah *true*. Hasil yang didapatkan dihimpun pada Tabel 4.10.

Tabel 4.10 Hasil Akhir Pengujian Mid-Point Validation

	Expected Result		Actual Result		
	False	True	False	True	
Jarak > 50m	55	0	55	0	
Jarak < 50m	0	45	0	45	
Total Data	1(00	10	0	
Presentase	100%	100%	100%	100%	
Kesuksesan					
Presentase	0%	0%	0%	0%	
Kegagalan					

Pengujian yang terakhir yaitu melakukan pengujian integritas data pada data tanpa menggunakan PBFT *blockchain* dan data yang menggunakan PBFT *blockchain*. Berikut contoh data pengujian yang dilakukan pada Tabel 4.11 tanpa menggunakan PBFT *blockchain* sebanyak 100 data dengan format data Javascript Object Notation (JSON).

Tabel 4.11 Contoh Data pengujian integritas tanpa PBFT blockchain

No.	Data	Expect Result	Actual Result
	3	Dapat Diakses	Dapat Diakses
	id:1, nama:Penginapan Syariah Prenjak,		
	alamat: Jl. Prenjak No.2		
	telp: (031) 3537103,		
	longitude:-7.2374686,		
	latitude:112.7364498,		
	pic: [object object]		
	}		
	{	Dapat Diakses	Dapat Diakses
	id:2, nama:Hotel Syariah Walisongo,		
	alamat: Jl. Petukangan		
	telp: (031) 3532197,		
	longitude:-7.2297734,		
	latitude:-12.7414554,		
	pic: [object object]		
	}		
	{	Dapat Diakses	Dapat Diakses
	id:3, nama: Grand Kalimas Hotel Syariah,		
	alamat: Jl. KH Mas Mansyur No.151		
	telp: (031) 3541616,		
	longitude:-7.2303718,		

	latitude:112.7408382,		
	pic: [object object]		
	}		
	{	Dapat Diakses	Dapat Diakses
	id:4, nama: Namira Syariah Hotel	1	1
	Surabaya,		
	alamat: Jl. Raya Wisma Pagesangan		
	No.203		
	telp: (031) 82518777,		
	longitude:-7.3376459,		
	latitude: 112.7141073,		
	pic: [object object]		
	}		
97	{	Tidak Dapat Diakses	Dapat Diakses
	id:97, nama: Bukan Penginapan Syariah	1	
	Prenjak,		
	alamat: Jl. Prenjak No.2211		
	telp: (031) 3537103,		
	longitude:-7.2374686,		
	latitude:112.7364498,		
	pic: [object object]		
	}		
98	1	Tidak Dapat Diakses	Dapat Diakses
	id:98, nama:Hotel Syariah Walisongo,	1	
	alamat: Jl. Petukangan		
	telp: (031) 23453532197,		
	longitude:-7.22977123JGS934,		
	latitude:-12.7414554,		
	pic: [object object]		
	}	3	
99	{	Tidak Dapat Diakses	Dapat Diakses
	id:99, nama: Grand <mark>ddd</mark> Kalimas Hotel	1	•
	Syariah,		
	alamat: Jl. KH Mas Mansyur No.2013		
	telp: (031) 3541616,		
	longitude:-7.2303718,		
	latitude:112.7408382,		
	pic: [object object]		
	}		
100	{	Tidak Dapat Diakses	Dapat Diakses
	id:100, nama: Namira#136@ Syariah	•	•
	Hotel Surabaya,		
	<i>y</i> ,		

```
alamat: Jl. Raya Wisma Pagesangan
No.203
telp: (031) 82518777@$-=,
longitude:-7.3376459,
latitude: 112.7141073,
pic: [object object]
}
```

Kemudian pengujian selanjutnya yaitu melakukan pengujian integritas data dengan menggunakan PBFT *blockchain* yang dilakukan sebanyak 104x seperti skenario yang terdapat pada Tabel 3.4. Hasil pengujian dihimpun pada Tabel 4.12.

Tabel 4.12 Contoh Data pengujian integritas dengan PBFT blockchain

		/ /			E X	A C
N O	NODE PRIMARY	NODE 1	NODE 2	NODE 3	P E C T	T U A L
1	"hash": "2345efb7725cc9 69551e20ed0e07 10d58628cf0d99b e404c6ef587145d fa0aef",	"hash": "2345efb7725cc96 9551e20ed0e0710 d58628cf0d99be4 04c6ef587145dfa0 aef",	"hash": "2345efb7725cc96 9551e20ed0e0710 d58628cf0d99be40 4c6ef587145dfa0a ef",	"hash": "2345efb7725cc9 69551e20ed0e07 10d58628cf0d99b e404c6ef587145d fa0aef",	T R U E	T R U E
2	"hash": "2345efb7725cc9 69551e20ed0e07 10d58628cf0d99b e404c6ef587145d fa0aef",	"hash": "2345efb7725cc96 9551e20ed0e0710 d58628cf0d99be4 04c6ef587145dfa0 aef", "hash":	"hash": "2345efb7725cc96 9551e20ed0e0710 d58628cf0d99be40 4c6ef587145dfa0a ef", "hash":	"hash": "abc2345efb7725 cc969551e20ed0 e0710d58628cf0d 99be404c6ef5871 45dfa0aef",	T R U E	TRUE
3	"hash": "2345efb7725cc9 69551e20ed0e07 10d58628cf0d99b e404c6ef587145d fa0aef".	"hash": "2345efb7725cc96 9551e20ed0e0710 d58628cf0d99be4 04c6ef587145dfa0 aef",	"hash": "abc2345efb7725c casdvzxc969551e2 0ed0e0710d58628 cf0d99be404c6ef5 87145dfa0aef",	"hash": "abc2345efb7725 cc969551e20ed0 e0710d58628cf0d 99be404c6ef5871 45dfa0aef",	F A L S E	H O L D H
4	"hash": "2345efb7725cc9 69551e20ed0e07 10d58628cf0d99b e404c6ef587145d fa0aef",	"hash": "11232345efb7725 cc969551e20ed0e 0710d58628cf0d9 9be404c6ef58714 5dfa0aef",	"hash": "abc2345efb7725c casdvzxc969551e2 0ed0e0710d58628 cf0d99be404c6ef5 87145dfa0aef",	"hash": "abc2345efb7725 cc969551e20ed0 e0710d58628cf0d 99be404c6ef5871 45dfa0aef",	FALSE	FALSE
5	"hash": "as12345efb7725 cc969551e20ed0 e0710d58628cf0d 99be404c6ef5871 45dfa0aef",	"hash": "2345efb7725cc96 9551e20ed0e0710 d58628cf0d99be4 04c6ef587145dfa0 aef",	"hash": "2345efb7725cc96 9551e20ed0e0710 d58628cf0d99be40 4c6ef587145dfa0a ef", "hash":	"hash": "2345efb7725cc9 69551e20ed0e07 10d58628cf0d99b e404c6ef587145d fa0aef",	FALSE	FALSE
6	"hash": "as12345efb7725 cc969551e20ed0 e0710d58628cf0d 99be404c6ef5871 45dfa0aef",	"hash": "2345efb7725cc96 9551e20ed0e0710 d58628cf0d99be4 04c6ef587145dfa0 aef",	"hash": "2345efb7725cc96 9551e20ed0e0710 d58628cf0d99be40 4c6ef587145dfa0a ef",	"hash": "2345efb7725cc9 69551e20ed0e07 10d58628cf0d99b e404c6ef587145d fa0aefXZA",	F A L S E	FALSE

	"hash":	"hash":	"hash":	"hash":	_	_
	"as12345efb7725	"2345efb7725cc96	"2345efb7725cc96	"2345efb7725cc9	F	F
_	cc969551e20ed0	9551e20ed0e0710	9551e20ed0e0710	69551e20ed0e07	Α	Α
7	e0710d58628cf0d	d58628cf0d99be4	d58628cf0d99be40	10d58628cf0d99b	L	L
	99be404c6ef5871	04c6ef587145dfa0	4c6ef587145dfa0a	e404c6ef587145d	S	S
	45dfa0aef",	aef",	ef <mark>]1[2"</mark> ,	fa0aefXZA",	E	E
	•	"hash":	"hash":	"hash":		
	"hash":				F	F
	"as12345efb7725	"F132345efb7725c	"2345efb7725cc96	"2345efb7725cc9	Α	Α
8	cc969551e20ed0	c969551e20ed0e0	9551e20ed0e0710	69551e20ed0e07	L	L
	e0710d58628cf0d	710d58628cf0d99	d58628cf0d99be40	10d58628cf0d99b	s	S
	99be404c6ef5871	be404c6ef587145	4c6ef587145dfa0a	e404c6ef587145d	F	Ē
	45dfa0aef",	dfa0aef <mark>XS</mark> ",	ef]1[2",	fa0aef <mark>XZA</mark> ",	_	-
						٠.
	"hash":	"hash":	"hash":	"hash":		
				"efca5b77d8edea	F	F
	"efca5b77d8edea	"efca5b77d8edeac	"efca5b77d8edeac	c296918fee2d21e	Α	Ι Δ
9	c296918fee2d21e	296918fee2d21ecf	296918fee2d21ecf	cf00fecf001136~-	L	L
6	cf00fecf00113653	00fecXNNNA1``f0	00fecf0011365390	A;H19""139T5390	s	S
	90c0abb97e18eS	011365390c0abb9	c0aSBN1bb97e18	c0abb97e18e3d3	Ē	E
	B1JP[3d399e",	7e18e3d399e",	e3d399e",	99e",	-	-
	"hash":	"hash":	"hash":	"hash":		
	"41dde147f05df19	"41dde147f05df19	"41dde147f05df19	"41dde147f05df19	Т	Т
					l .	1
9	1e563dc171918c	1e563dc171918c7	1e563dc171918c7	1e563dc171918c	R	F
7	796bc29e48cfef6f	96bc29e48cfef6f0c	96bc29e48cfef6f0c	796bc29e48cfef6f	Ū	L
	0c07bcb7be5168	07bcb7be <mark>516</mark> 88 <mark>4a</mark>	07bc <mark>b7be51</mark> 6884a	0c07bcb7be5168	E	E
	84adc",	dc",	dc",	84adc",		
	"hash":	"h <mark>ash</mark> ":	"hash":	"hash":		
	"41dde147f05df19	"41dde147f05df19	"4 <mark>1dd</mark> e147f <mark>05d</mark> f19	"41dde147f05df19	T	T
9	1e563dc171918c	1e563dc171918c7	1e563dc171918c7	1e563dc171918c	R	F
8	796bc29e48cfef6f	96bc29e48cfef6f0c	96bc29e48cfef6f0c	7S196bc29e48cfe	U	lι
Ŭ	0c07bcb7be5168	07bcb7be516884a	07bcb7be516884a	f6f0c07bcb7be51	Ē	E
	84adc",	dc",	dc",	6884adSKHc",	-	_
ŀ			uc,	,		ŀ
	"hash":	"hash":		"hash":	F	F
	"41dde147f05df19	"41dde147f05df19	"hash":	"41dde147f05df19	Α	1
9	1e563dc171918c	1e563dc171918c7	"41dde147f05cfef6	1e563dc171918c	L	L
9	796bc29e48cfef6f	96bc29e48cfef6f0c	f0c07bcb7be51688	7S196bc29e48cfe	s	5
	0c07bcb7be5168	07bcb7be516884a	4adc",	f6f0c07bcb7be51	Ē	E
	84adc",	dc",		6884adSKHc",	_	-
	"hash":			"hash":	F	F
4	"41dde147f05df19		"hash":	"41dde147f05df19	l .	Ι.
1	1e563dc171918c	W 1 W W	"41dde147f05cfef6	1e563dc171918c	A	^
0	796bc29e48cfef6f	"hash": "",	f0c07bcb7be51688	7S196bc29e48cfe	L	L
0	0c07bcb7be5168		4adc",	f6f0c07bcb7be51	S	S
	84adc",		, , ,	6884adSKHc",	E	E
	0.000,	"hash":	"hash":	"hash":		
		"41dde147f05df19	"41dde147f05df19	"41dde147f05df19	F	F
1		1e563dc171918c7	1e563dc171918c7	1e563dc171918c	Α	Α
0	"hash": "",	96bc29e48cfef6f0c	96bc29e48cfef6f0c		L	L
1				796bc29e48cfef6f	S	S
		07bcb7be516884a	07bcb7be516884a	0c07bcb7be5168	E	E
		dc",	dc",	84adc",		
		"hash":	"hash":	"hash":	F	F
1		"41dde147f05df19	"41dde147f05df19	"41dde147f05df19	Ä	Ι.
0	"hash": "",	1e563dc171918c7	1e563dc171918c7	1e563dc171918c	Ĺ	Ĺ
-	ilaəli . ,	96bc29e48cfef6f0c	96bc29e48cfef6f0c	7S196bc29e48cfe	S	S
2		07bcb7be516884a	07bcb7be516884a	f6f0c07bcb7be51		
		dc",	dc",	6884adSKHc",	E	E
		"hash":	,	"hash":		
		"41dde147f05df19	"hash":	"41dde147f05df19	F	F
1		1e563dc171918c7	"41dde147f05cfef6	1e563dc171918c	Α	Α
0	"hash": "",				L	L
	,	96bc29e48cfef6f0c	f0c07bcb7be51688	7S196bc29e48cfe	s	S
3						
3		07bcb7be516884a dc",	4adc",	f6f0c07bcb7be51 6884adSKHc",	ΙE	E

1 0 4	"hash": "",	"hash": "",	"hash": "41dde147f05cfef6 f0c07bcb7be51688	"hash": "41dde147f05df19 1e563dc171918c 7S196bc29e48cfe	F A L S	F A L S	
4	,	<u> </u>	4adc",	f6f0c07bcb7be51	S	S	
				6884adSKHc",	-	L .	

Untuk mempermudah klasifikasi informasi pada Tabel 4.12 dan Tabel 4.13 maka perlu dibuat tabel baru agar dapat dibaca dengan mudah. Berikut tabel penghitungan presentase pada pengujian integritas tanpa menggunakan PBFT blockchain,

Tabel 4.13 Hasil Pengujian Integritas data tanpa PBFT Blockchain

		SKENARIO					
No.	Primary	Exepct	Total Expect	Total Actual			
		Result	Result	Result			
1	Data Benar	Bisa	50	50			
		Diakses					
2	Data Salah	Tidak	50	0			
		Dapat					
		<mark>DIa</mark> ks <mark>es</mark>					
Presentase			100%	50%			
Kesuksesan			4				
Presentase			0%	50%			
Kegagalan							

Berikut adalah tabel baru yang dibuat untuk data pengujian integritas dengan menggunakan PBFT.

Tabel 4.14 Hasil Pengujian Integritas Data Menggunakan PBFT.

				Sk	ENARI	О		
	NO		Node	Node	Node	Expect	Total	Total
	110	Primary	1	2	3	Result	Expect	Actual
			1	<i>_</i>	3	Result	Result	Result
F	1	Data	Data	Data	Data	Bisa	13	13
		Benar	Benar	Benar	Benar	Diakses		

2	Data	Data	Data	Data	Bisa	13	13
	Benar	Benar	Benar	Salah	Diakses		
3	Data	Data	Data	Data	Tidak	13	13
	Benar	Benar	Salah	Salah	Bisa		
					Diakses		
4	Data	Data	Data	Data	Tidak	13	13
	Benar	Salah	Salah	Salah	Bisa		
					Diakses		
5	Data	Data	Data	Data	Tidak	13	13
	Salah	Benar	Benar	Benar	Bisa		
					Diakses		
6	Data	Data	Data	Data	Tidak	13	13
	Salah	Benar	Benar	Salah	Bisa		
					Diakses		
7	Data	Data	Data	Data	Tidak	13	13
	Salah	Benar	<mark>Sal</mark> ah	Sala <mark>h</mark>	Bisa		
					Diakses		
8	Data	Data	Data	Data	Tidak	13	13
	Salah	Salah	Salah	Salah	Bisa		
				, /	Diakses		
Presentase						100%	100%
Kesuksesan			-4				
Presentase				100		0%	0%
Kegagalan							

Untuk menilai peningkatan keamanan data, maka perlu dibuat tabel baru untuk melakukan perbandingan dari Tabel 4.13 dan Tabel 4.14. Berikut Tabel 4.15 yang menunjukkan hasil perbandingan tersebut.

Tabel 4.15 Tabel Perbandingan Presentase Pengujian Integritas Data

Skenario	

Hasil Pre	sentase H	[asil P	resentase	Hasil 1	Presentase	Hasil	Presentase
Gagal Tanpa	PBFT S	ukses	Tanpa	Gagal	dengan	Sukses	dengan
	P	BFT		PBFT		PBFT	
50%	50	0%		0%		100%	

Dari Tabel 4.15 menunjukkan bahwa hasil presentase gagal tanpa PBFT memiliki presentase 50% dan presentase sukes 50%. Kemudian pada hasil presentase gagal dengan menggunakan PBFT sebesar 100% dan presentase sukses sebesar 100%, ini menunjukkan bahwa peningkatan keamanan dengan menggunakan blockchain PBFT sebagai penyimpanan dan Algoritma JSON-DIFF sebagai algoritma pencocokkan data dapat meningkatkan keamanan data sebesar 50% dari presentase penggunaan tanpa PBFT *blockchain*.



BAB 5 PENUTUP

5.1 Kesimpulan

Dari hasil penelitian yang telah dilakukan untuk meningkatkan keamanan data dengan menggunakan *practical byzantine fault tolerance* (PBFT) blockchain pada geospatial retrieval hotel syari'ah, dapat ditarik kesimpulan sebagai berikut.

- 1. Hasil pengujian *test case scenario black box* aplikasi menunjukkan, pada sistem android 7,8 dan 9 telah melewati 5 *test case scenario* pada masing-masing sistem sesuai dengan *test scenario* yang diberikan.
- 2. Hasil pengujian konsensus PBFT menggunakan 4 *node* dengan nilai *fault tolerance* 33% sesuai dengan skenario yang telah dibuat. Kemudian pengujian *network latency* enkripsi data pada jaringan 3G memiliki nilai rata-rata nilai sebesar 2.047s dan enkripsi data pada jaringan 4G memiliki rata-rata nilai sebesar 0.617s, sedangkan pada dekripsi data jaringan 3G memiliki rata-rata nilai sebesar 0.414s dan jaringan 4G memiliki rata-rata nilai sebesar 0.297s. Jaringan 4G memiliki 107% lebih cepat daripada jaringan 3G pada saat melakukan enkripsi data pada *blockchain*. Kemudian saat dekripsi jaringan 4G memiliki 32.9% lebih cepat daripada jaringan 3G. Kemudian pada pengujian *smart-contract mid-point validation* memiliki nilai presentase kesuksesan sebesar 100% dan memiliki presentase kegagalan sebesar 0%.
- 3. Hasil evaluasi yang terakhir yaitu melakukan pengujian integritas data tanpa menggunakan PBFT *blockchain* dan menggunakan PBFT *blockchain*. Pada pengujian tanpa menggunakan PBFT *blockchain* menghasilkan nilai presentase kesuksesan sebesar 50% dan kegagalan 50%, kemudian pada pengujian integrasi dengan mengunakan PBFT *blockchain* memiliki nilai presentase kesuksesan sebesar 100% dan presentase kegagalan sebesar 0%. Presentase pada saat menggunakan PBFT *blockchain* meningkat sebanyak 50%.

5.2 Saran

Saran yang dapat penulis berikan terhadap penerapan *practical byzantine* fault tolerance (PBFT) blockchain sebagai teknologi peningkatan keamanan data atau pengembengan penelitian yang akan datang sebagai berikut.

- 1. Disarankan untuk menggunakan jaringan 4G saat menggunakan aplikasi berbasis *blockchain*. Karena pada hasil pengujian *network latency*, jaringan 4G memiliki nilai yang rendah.
- 2. Disarankan untuk menggunakan banyak jenis data serta menambah nilai *fault tolerance* pada metode PBFT untuk menghitung performa mesin serta penggunaan konsumsi *network latency*.
- 3. Menambah algoritma untuk melakukan pengujian integritas data seperti Algoritma Rabin Karp.
- 4. Mengembangkan aplikasi dengan berbagai jenis operasi sistem seperti Android 10 dan IOS

DAFTAR PUSTAKA

- Adeyemi, B. A. (2013). Mapping the Locational Pattern of Hotels in Akure, Ondo State. *Journal of Humanities and Social Science*, 14(3), 95–99.
- Ali, A., & Afzal, M. M. (2018). Confidentiality in Blockchain. *International Journal of Engineering Science I*, 7(1), 50–52.
- Anugrah, Y., Hannats, M., Ichsan, H., & Kusyanti, A. (2019). Implementasi Algoritme SHA-256 Menggunakan Protokol MQTT pada Budidaya Ikan Hias. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 3(4), 4066–4073.
- Battour, M., & Ismail, M. N. (2016). Halal tourism: Concepts, practises, challenges and future. *Tourism Management Perspectives*, 19, 150–154. https://doi.org/10.1016/j.tmp.2015.12.008
- Beki Subaeki, M. R. J. (2016). APLIKASI INFO HALAL MENGGUNAKAN BARCODE SCANNER UNTUK SMARTPHONE ANDROID. Jurnal Informatika, III(1).
- Bendovschi, A. (2015). Cyber-Attacks Trends , Patterns and Security Countermeasures. 7th International Conference on Financial Criminology 2015, 28(April), 24–31. https://doi.org/10.1016/S2212-5671(15)01077-1
- Biswas, K., & Muthukkumarasamy, V. (2017). Securing smart cities using blockchain technology. *Proceedings 18th IEEE International Conference on High Performance Computing and Communications, 14th IEEE International Conference on Smart City and 2nd IEEE International Conference on Data Science and Systems, HPCC/SmartCity/DSS 2016*, 1392–1393. https://doi.org/10.1109/HPCC-SmartCity-DSS.2016.0198
- Boulos, M. N. K., Wilson, J. T., & Clauson, K. A. (2018). Geospatial blockchain: promises, challenges, and scenarios in health and healthcare. *International Journal of Health Geographics*, 1–10. https://doi.org/10.1186/s12942-018-0144-x
- BPS-Statistics of Surabaya Municipality. (2018). Kota Surabaya Dalam Angka. In BPS-Statistics of Surabaya Municipality (Ed.), *BPS-Statistics of Surabaya Municipality* (1st ed., Vol. 1). https://doi.org/10.1192/bjp.111.479.1009-a
- Castro, M., & Liskov, B. (1999). Practical Byzantine Fault Tolerance. *Proceedings*

- of the Symposium on Operating System Design and Implementation, (February), 1–14. https://doi.org/10.1145/571637.571640
- Chen, J., Zhang, X., & Shangguan, P. (2019). Improved PBFT Algorithm Based on Reputation and Voting Mechanism Improved PBFT Algorithm Based on Reputation and Voting Mechanism. *Journal of Physics: Conference Series*. https://doi.org/10.1088/1742-6596/1486/3/032023
- D Rachmawati, J. T. T. and A. B. C. G. (2018). A comparative study of Message Digest 5 (MD5) and SHA256 algorithm. *2nd International Conference on Computing and Applied Informatics* 2017, 5.
- Dwyer, R., Kushlev, K., & Dunn, E. (2017). Journal of Experimental Social Psychology Smartphone use undermines enjoyment of face-to-face social interactions ★. *Journal of Experimental Social Psychology*, (March), 0–1. https://doi.org/10.1016/j.jesp.2017.10.007
- Efanov, D., & Roschin, P. (2018). The all-pervasiveness of the blockchain technology. *Procedia Computer Science*, 123, 116–121. https://doi.org/10.1016/j.procs.2018.01.019
- Fang, C., & Zhang, S. (2018). Geographic Information Retrieval Method for Geography Mark-Up Language Data. *International Journal of Geo-Information*, (Xml). https://doi.org/10.3390/ijgi7030089
- Gamundani, A. M., & Nekare, L. M. (2018). A Review of New Trends in Cyber Attacks: A Zoom into Distributed Database Systems. *IST-Africa 2018 Conference Proceedings Paul Cunningham and Miriam Cunningham*, (January).
- Han, W., Yang, Z., Di, L., & Member, S. (2014). Enhancing Agricultural Geospatial Data Dissemination and Applications Using Geospatial Web Services. *IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING*, 7(11), 4539–4547.
- Harris, E., George, S., Costas, E., Harris, E., George, S., & Costas, E. (2015). Areathroughput trade-offs for SHA-1 and SHA-256 hash functions 'pipelined designs. *Journal of Circuits, System and Computers*, 25.
- Haverbeke, M. (2018). Eloquent JavaScript.
- James, N. T., & Kannan, R. (2017). A Survey on Information Retrieval Models,

- Techniques and Applications. *International Journals of Advanced Research in Computer Science and Software Engineering*, (October 2019). https://doi.org/10.23956/ijarcsse.v7i7.90
- Kaner, C., & Ph, D. (2013). An Introduction to Scenario Testing. (January 2003).
- Lakshmi, B. V., & Mohan, V. (2017). Plant Leaf Image Detection Method Using a Midpoint Circle Algorithm for Shape-Based Feature Extraction Plant Leaf Image Detection Method Using a. *Journal of Modern Applied Statistical Methods*, 16(1), 461–480. https://doi.org/10.22237/jmasm/1493598420
- Mir, S. Q., Dar, M., & Beig, B. M. (2011). INFORMATION AVAILABILITY: COMPONENTS, THREATS AND PROTECTION MECHANISMS. *Journal of Global Research in Computer Science (JGRCS)*, 2(3).
- Mujahidin, M. (2018). Sharia hotels in Indonesia: Concept and potential analysis. Munich Personal RePEc Archive, (90819).
- Nanin Trianawati Sugito, ST., MT. dan DRS. Dede Sugandi, M. S. (2009).

 **URGENSI SISTEM INFORMASI GEOGRAFIS (SIG) UNTUK

 MENDUKUNG DATA GEOSPASIAL.
- Sarwar, M. (2014). Impact of Smartphone's on Society. European Journal of Scientific Research, 98(February 2013), 216–226.
- Satriana, E. D., & Faridah, H. D. (2018). Wisata halal: perkembangan, peluang, dan tantangan. *Journal of Halal Product and Research*, 01(02), 32–43.
- Setiawan, F. (2014). Konsep Maslahah (Utility) dalam al-Qur'an Surat al-Baqarah ayat 168 dan Surat al-A'raf ayat 31. *Journal Trunojoyo*.
- Shah, H., & Soomro, T. R. (2017). Node . js Challenges in Implementation. *Global Journal Of Computer Science and Technology: E Network, Web & Security*, 17(June), 2.
- Sharma, N., Perniu, L., Chong, R. F., Iyer, A., Nandan, C., Mitea, A., ... Danubianu, M. (2010). *Database Fundamentals*.
- Sivathanu, G., Wright, C. P., & Zadok, E. (2005). Ensuring Data Integrity in Storage: Techniques and Applications. *Proceedings of the 2005 ACM Workshop on Storage Security and Survivability StorageSS '05.*, 26–36.
- Stiawan, D., Idris, M. Y., Abdullah, A. H., Aljaber, F., & Budiarto, R. (2017). Cyber-Attack Penetration Test and Vulnerability Analysis. *International*

- *Journal of Online and Biomedical Engineering*, 13(1), 125–132.
- Sukhwani, H., Mart, M., Chang, X., Trivedi, K. S., & Rindos, A. (2017). Performance Modeling of PBFT Consensus Process for Permissioned Blockchain Network (Hyperledger Fabric). *IEEE 36th Symposium on Reliable Distributed Systems*, 7–9. https://doi.org/10.1109/SRDS.2017.36
- Vargas-Sánchez, A., & Moral-Moral, M. (2019). Halal tourism: literature review and experts' view. *Journal of Islamic Marketing*, 1. https://doi.org/10.1108/JIMA-04-2017-0039
- Wadehra, S., Goel, S., & Sengar, N. (2018). International Journal of Trend in Scientific Research and Development (IJTSRD) AES Algorithm: lgorithm: Encryption and Decryption. *International Journal of Trend in Scientific Research and Development (IJTSRD)*, 2(3), 1075–1077.
- Wibowo, A. T., Sulistyono, M. T., & Hariadi, M. (2020). CRYPTOSPATIAL COORDINATE USING THE RPCA BASED ON A POINT IN POLYGON TEST FOR CULTURAL. 4205, 3–9.
- Zhang, S., & Lee, J.-H. (2019). Analysis of the main consensus protocols of blockchain. *ICT Express*, (xxxx), 1–5. https://doi.org/10.1016/j.icte.2019.08.001