TESIS

Rancangan Sistem Manajemen Pertukaran Energi Menggunakan Blockchain Hyperledger Fabric

Disusun dan Diajukan oleh :

NAMA : FARA TRIADI

NIM : D032211002

Dosen Pembimbing :

Prof. Dr. Eng. Ir. Syafaruddin, S.T, M.Eng.IPU

Prof. Dr. Ir. Andani, M.T.

FAKULTAS TEKNIK

UNIVERSITAS HASANUDDIN

MAKASSAR

2023

TESIS

Rancangan Sistem Manajemen Pertukaran Energi Menggunakan Blockchain Hyperledger Fabric

FARA TRIADI
D032 211 002

Telah dipertahankan di hadapan Panitia Ujian Tesis yang dibentuk dalam rangka penyelesaian studi pada Program Magister Teknik Elektro, Fakultas Teknik Universitas Hasanuddin
pada tanggal 5 Juli 2023
dan dinyatakan telah memenuhi syarat kelulusan

Menyetujui,

Pembimbing Utama

Prof. Dr. Eng.Ir. Syafaruddin, ST., M.Eng.IPU
NIP. 19740530 1999 031 003

Pembimbing Pendamping

Prof.Dr.Ir. Andani Achmad, MT
NIP. 19601231 1987 031 022

Dekan Fakultas Teknik
Universitas Hasanuddin

Prof. Dr. Eng. Ir. Muhammad Isran Ramli, S.T., M.T.,IPM
NIP. 19730926 2000 121 002

Ketua Program Studi
S2 Teknik Elektro

Dr. Eng. Ir. Wardi, S.T., M.Eng
NIP. 19720828 199903 1 003

PERNYATAAN KEASLIAN
TESIS DAN PELIMPAHAN HAK CIPTA

Dengan ini saya menyatakan bahwa, disertasi berjudul "**Rancangan Sistem Manajemen Pertukaran Energi Menggunakan Blockchain Hyperledger Fabric**" adalah benar karya saya dengan arahan dari komisi pembimbing. Karya ilmiah ini belum diajukan dan tidak sedang diajukan dalam bentuk apa pun kepada perguruan tinggi mana pun. Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan maupun tidak diterbitkan dari penulis lain telah disebutkan dalam teks dan dicantumkan dalam Daftar Pustaka disertasi ini. Sebagian dari isi disertasi ini telah dipublikasikan di **International Conference On Cyber Management And Engineering (CyMaEn) 2023** sebagai artikel dengan judul "*Design of Trading Energy System Management Using Blockchain Hyperledger Fabric*".

Dengan ini saya melimpahkan hak cipta dari karya tulis saya berupa disertasi ini kepada Universitas Hasanuddin.

Makassar, 5-07-2023

FARA TRIADI

NIM D032211002

Kata Pengantar

Saya bersyukur bahwa tesis ini akhirnya dapat terselesaikan dengan baik. Penelitian yang saya lakukan dapat terlaksana dengan sukses dan tesis ini dapat terampungkan atas bimbingan, diskusi dan arahan Prof. Dr. Eng. Syafaruddin, ST., M.Eng.IPU sebagai pembimbing-1, dan Prof. Dr. Ir. Andani Achmad, MT sebagai pembimbing-2. Saya mengucapkan berlimpah terima kasih kepada mereka.

Ucapan terima kasih juga saya ucapkan kepada pimpinan Universitas Hasanuddin dan Sekolah Pascasarjana Universitas Hasanuddin yang telah memfasilitasi saya menempuh program magister serta para dosen dan rekan-rekan dalam tim penelitian.

Akhirnya, kepada kedua orang tua tercinta saya mengucapkan limpah terima kasih dan sembah sujud atas doa, pengorbanan dan memotivasi mereka selama saya menempuh pendidikan. Penghargaan yang besar juga saya sampaikan kepada seluruh keluarga atas motivasi dan dukungan yang tak ternilai.

Penulis,

FARA TRIADI

ABSTRAK

Fara Triadi, *Rancangan Sistem Manajemen Pertukaran Energi Menggunakan Blockchain Hyperledger FabricI* (dibimbing oleh **Syafaruddin** dan **Andani Achmad**)

Ada beberapa penelitian yang telah dilakukan untuk meningkatkan keamanan, transparansi, dan kejujuran transaksi listrik di komunitas perdagangan smartgrid atau peer-to-peer tanpa mengekspos privasi konsumen dan prosumer mereka, salah satunya adalah blockchain. Saat ini, blockchain terkenal dan kebanyakan digunakan dalam kasus keuangan. Blockchain digunakan untuk membuat sistem manajemen perdagangan menggunakan Hyperledger Fabric dimana sistem blockchain yang diizinkan bertujuan untuk mengamankan privasi konsumen. Dalam tesis ini, Hyperledger Caliper oleh IBM digunakan untuk membuat skenario pengujian 3 nodes konfigurasi. Hasil menunjukkan bahwa untuk konfigurasi 3 nodes dengan 1000 transaksi, latensi rata-rata 50 tps adalah 0,4 detik, 100 tps adalah 6,57 detik dan 200 tps adalah 9,18 detik. Hasil ini menunjukkan validasi membutuhkan waktu lebih lama di 200 tps. Semakin besar transaksi per detik, semakin banyak pula validasi yang terjadi. Daya komputasi juga meningkat ketika sistem mulai berjalan. Jadi, dengan konfigurasi 3 node organisasi, masih dapat menangani 1000 transaksi per detik dengan baik tanpa kegagalan dan konfigurasi ini dapat diintegrasikan dengan cloud server atau blockchain publik melalui API sehingga dapat diperoleh model yang lebih dinamis dari usulan ini.

Kata kunci: *Hypeledger Fabric, Blockchain, microgrid, Hyperledger Caliper, Blockexplorer.*

ABSTRACT

Fara Triadi, Design of Energy Exchange Management System Using Blockchain Hyperledger FabricI (supervised by **Syafaruddin** and **Andani Achmad**)

There are several studies that have been conducted to improve the security, transparency and honesty of electricity transactions in the smartgrid or peer-to-peer trading community without exposing the privacy of their consumers and prosumers, one of which is the blockchain. Currently, blockchain is well-known and mostly used in financial cases. Blockchain is used to create a trade management system using Hyperledger Fabric where the allowed blockchain system aims to secure consumer privacy. In this thesis, IBM's Hyperledger Caliper is used to create a test scenario of 3 configuration nodes. The results show that for a configuration of 3 nodes with 1000 transactions, the average latency of 50 tps is 0.4 seconds, 100 tps is 6.57 seconds and 200 tps is 9.18 seconds. These results indicate validation takes longer at 200 tps. The greater the transaction per second, the more validation that occurs. Computing power also increases when the system starts up. So, with the configuration of 3 organizational nodes, it can still handle 1000 transactions per second well without failure and this configuration can be integrated with a cloud server or public blockchain via API so that a more dynamic model can be obtained from this proposal.

Keywords: Hypeledger Fabric, Blockchain, microgrid, Hyperledger Caliper, Blockexplorer.

DAFTAR ISI

HALAMAN SAMPUL DEPAN ..... **i**

LEMBAR PENGESAHAN ..... **ii**

PERNYATAAN KEASLIAN TESIS DAN PELIMPAHAN HAK CIPTA .... **iii**

KATA PENGANTAR ..... **iv**

ABSTRAK……..**v**

DAFTAR ISI ..... **vii**

DAFTAR GAMBAR ..... **ix**

DAFTAR TABEL ..... **ix**

CURRICULUM VITAE ..... **x**

BAB I PENDAHULUAN **1**

1.1 Latar Belakang 1

1.2 Rumusan Masalah 3

1.3 Tujuan Penelitian 4

1.4 Manfaat Penelitian 4

1.5 Batasan Masalah 4

1.6 State of the art 4

1.7 Kerangka Pikir 7

1.8 *Novelty* dan Kontribusi 8

1.9 Sistematika Penulisan 9

BAB II TINJAUAN PUSTAKA **10**

2.1 Tinjauan Teori 10

2.2 Blockchain 10

2.2.1 Blockchain Explorer 12

2.2.2 Hyperledger Fabric 13

2.2.3 Hyperledger Caliper 14

2.2.4 Internet of Things (IOT) 15

2.2.5 Arduino 16

2.2.6 Arduino IDE 17

2.2.7 Sensor ACS712 19
2.2.8 Modul Sensor Tegangan 19
2.2.9 Docker 20
2.2.10 CouchDB 22
2.2.11 NodeJS 23
2.2.12 Firebase 23
2.2.13 Socket.IO 24
2.2.14 Golang 25
2.2.15 Javascript 26
2.2.16 Python 27
2.2.17 C++ 28
BAB III METODE PENELITIAN 30
3.1 Tahap Penelitian 30
3.2 Jenis Penelitian 30
3.3 Perancangan Sistem 31
3.3.1 Flowchart IoT 40
3.4. Pengambilan Data 40
BAB IV HASIL DAN PEMBAHASAN 42
4.1. Hasil Simulasi Menggunakan Caliper Benchmark 42
4.2. Hasil Menggunakan Skenario Konsumsi Klien 65
4.3. Pembahasan 68
BAB V PENUTUP 70
5.1. Kesimpulan 70
5.2. Saran 71
DAFTAR PUSTAKA 72
LAMPIRAN SPESIFIKASI CPU 74
LAMPIRAN CODE 75

DAFTAR GAMBAR

Gambar 1.7. Kerangka Pikir ……7

Gambar 1.8. Alur Penelitian……..…….……8

Gambar 2. 1. Ilustrasi Blockchain……..…….....11

Gambar 2.2. Prinsip Kerja Blockchain ……..……12

Gambar 2.3. Blockexplorer…….....13

Gambar 2.4. Hyperledger caliper arsitektur…….....15

Gambar 2.5. Blockchain Based IoT Model…….....16

Gambar 2.6. Arduino Esp32 Pins…….…..17

Gambar 2.7. Arduino IDE……..……17

Gambar 2.8. Typical Modul Sensor ACS712……...…….....19

Gambar 2.9. Bentuk rangkaian sensor tegangan untuk mengukur tegangan beban pada Panel Surya……...…..20

Gambar 2.10. Docker Architecture…….……...22

Gambar 3.1. Aristektur Hyperledger Fabric yang diusulkan…….31

Gambar 3. 2. Arsitektur IoT yang diusulkan ……..……..34

Gambar 3. 3. Flowchart IoT yang diusulkan……...……..……...…..40

Gambar 4 1. Hasil konfigurasi 50 TPS…….....….43

Gambar 4.2. Hasil konfigurasi 100 TPS…….....……..49

Gambar 4.3. Hasil konfigurasi 200 TPS…….....…...…53

Gambar 4.4. Hash kode block 107……..……...…57

Gambar 4.5. Hash kode block 108……..…...…59

Gambar 4.6. Hash kode block 109……..…...…61

Gambar 4.7. Hasil Resources Monitor…….....……63

Gambar 4.8. Hasil Blockexplorer dari transaksi klien……...66

DAFTAR TABEL

TABEL1.6. STATE OF ART……..……5

TABEL 4.2.1 TIMESTAMP TRANSAKSI …….63

TABEL 4.2.2 TABEL MINIMUM LATENCY……...…….……..…..65

TABEL 4.2.3 TABEL PRESENTASI ERROR…….……..……65

CURRICULUM VITAE

A. Data Pribadi

1. Nama : FARA TRIADI
2. Tempat, tgl. lahir : Palopo, 12 Mei 1991
3. Alamat : Btn Antang Jaya Blok I No.2
4. Kewarganegaraan : Warga Negara Indonesia

B. Riwayat Pendidikan

1. Tamat SMAN tahun 2009 di SMAN 12 Makassar
2. Sarjana (S1) tahun 2015 di Universitas Hasanuddin

C. Pekerjaan dan Riwayat Pekerjaan

Jenis pekerjaan : Mahasiswa

NIP atau identitas lain (NIK) : 7371121205910004

Pangkat/Jabatan : -

D. Karya ilmiah yang telah dipublikasikan

International Conference On Cyber Management And Engineering (CyMaEn) 2023

E. Makalah pada Seminar/Konferensi Ilmiah Nasional dan Internasional

International Conference On Cyber Management And Engineering (CyMaEn) 2023

BAB I PENDAHULUAN

1.1 Latar Belakang

Kebutuhan energi terus meningkat setiap waktu seiring ditemukannya pasar energi pada revolusi industrial kedua. Kebutuhan energi pada tahun 2015 mencapai 21,153 TWH, pada tahun 2015 berdasarkan sumber energi dan permintaan konsumen, harga pada tahun itu mencapai range US$0.01 per KWH di Argentina hingga US$0,33 per kWh di Jerman (Statista: Portal Statistik). Untuk menghasilkan energi listrik dibutuhkan peralatan yang sangat mahal dan tidak ekonomis untuk skala kecil maupun besar dan konsumen mendapatkan energi listrik mereka dari power plant, energi listrik disalurkan oleh jalur distribusi listrik (power grid/ smart grid) dan konsumen membayar apa yang mereka konsumsi. Pendsitribusisian dan proses mengahasilkan energi tersebut mengalami perubahan diiringi dengan kemajuan teknologi, Automasi mengarahkan produksi ke produksi massal yang mengarah ke pembangkit listrik yang lebih murah, keutungungan signifikan untuk energi listrik dan ketersediaan biaya produksi yang murah dari teknologi energi terbarukan juga mengalami pengingkatan termasuk di dalamnya adalah teknologi photovoltaic (PV) Alhasil, minat jual atau sharing cukup besar pada energi listrik telah disaksikan di pasar listrik (Baig et al., 2020). FiTs atau Feed in Tariffs di beberapa negara telah berlaku dimana para prosumer (producer and consumer) memungkinkan mereka menjual kembali energi mereka ke jaringan utilitas dengan harga yang tetap, FiT memberikan solusi finansial yang lebih efesien dan memaksimalkan konsumsi sendiri dengan mengoptimalkan penyimpanan energi local dari prosumer. Dengan berkembangnya teknologi di bidang energi dan internet, pangsa pasar energi terdistribusi telah meningkat secara signifikan [1,2], dan banyak konsumen telah berubah menjadi prosumer. Akses produsen dan konsumen transaksi bilateral yang bebas dari daya, membuat transaksi

daya lebih fleksibel. Namun, Transaksi daya peer-to-peer (P2P) memerlukan keamanan dan kejujuran atau transparansi dari komunitas tersebut tanpa harus mengorbankan privasi dari setiap prosumer. Sebagai tambahan, komunikasi dan layanan penjualan atau trading pengguna listrik sering disediakan oleh penyedia layanan pihak ketiga. Agen mungkin mencuri informasi yang relevan untuk memaksimalkan milik mereka sendiri kepentingan, yang mengarah ke persaingan berbahaya. Oleh karena itu, perlu untuk menemukan cara yang efektif untuk meningkatkan kredibilitas dan transparansi sistem transaksi tenaga listrik. Blockchain adalah sebuah teknologi yang digunakan untuk membangun database terdistribusi yang aman dan transparan. Dalam sistem perdagangan tradisional, perusahaan besar sering mendominasi pasar dan mengendalikan harga. Namun, sistem perdagangan peer to peer dapat membantu mengatasi masalah ini dengan memungkinkan para pengguna untuk berdagang secara langsung tanpa melalui pihak ketiga.

Namun, meskipun perdagangan peer to peer memiliki potensi untuk membantu mendorong pertumbuhan ekonomi, masih ada beberapa tantangan yang perlu diatasi, terutama dalam hal keamanan dan kepercayaan. Selain itu, penyedia layanan pihak ketiga sering menyediakan penjualan atau komunikasi perdagangan, dan layanan pengguna yang kuat. Beberapa agen mungkin mengambil beberapa informasi yang relevan untuk memaksimalkan kepentingan mereka yang mengarah ke persaingan jahat atau menggunakan identitas untuk tindakan ilegal. Oleh karena itu, dalam tesis ini, akan diusulkan rancangan sistem manajemen perdagangan peer to peer yang menggunakan teknologi blockchain private untuk meningkatkan keamanan dan transparansi perdagangan untuk mengatasi sistem keamanan.

Blockchain private adalah blockchain yang hanya dapat diakses oleh orang-orang tertentu. Dalam hal ini, blockchain private akan digunakan untuk mengelola transaksi perdagangan peer to peer, sehingga hanya pengguna yang terverifikasi yang dapat melakukan transaksi di platform ini.

Dengan demikian, transaksi perdagangan menjadi lebih aman dan terpercaya.

Selain itu, sistem manajemen perdagangan peer to peer yang diusulkan juga akan menggunakan smart contract untuk mengatur dan menjalankan transaksi secara otomatis, sehingga proses perdagangan menjadi lebih efisien dan transparan. Dengan adanya sistem manajemen perdagangan yang lebih aman dan efisien ini, diharapkan dapat mendorong pertumbuhan ekonomi dengan membantu para pelaku bisnis kecil dan menengah untuk menjual produk mereka secara langsung dan mengatasi ketergantungan pada perusahaan besar.

Dalam rangka meningkatkan keamanan transaksi ketenaga listrikan dalam smartgrid komunitas dan proses transaksi energi peer to peer dalam hubungannya antara saluran data produsen konsumen dalam jaringan grid yang terhubung dengan kaitannya dengan energi terbarukan yaitu solar panel stand alone, dalam penilitian ini akan dibuat Rancangan Management Pertukaran Energi Menggunakan Sistem Hyperledger fabric, yang dimana Hyperledger fabric merupakan framework blockchain yang bertipe permissioned blockchain, yang menyediakan solusi baru untuk penyimpanan energi terdistribusi, perlindungan data dan penelusuran sejarah transaksi.

1.2 Rumusan Masalah

1. Bagaimana merancang Sistem Manajemen Pertukaran Energi Menggunakan Sistem Blockchain Hyperledger Fabric?
2. Bagaimana performansi dari Sistem Manajemen Pertukaran Energi Menggunakan Sistem Blockchain Hyperledger Fabric yang diusulkan?

1.3 Tujuan Penelitian

1. Merancang Sistem Manajemen Pertukaran Energi Menggunakan Blockchain Hyperledger Fabric Hyperledger Fabric 2.0 dengan 3 nodes konfigurasi.
2. Menganalisia Kinerja Sistem Manajemen Pertukaran Energi Menggunakan Blockchain Hyperledger Fabric 2.0 dengan 3 nodes konfigurasi menggunakan Hyperledger caliper.

1.4 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah :

1. Merancang sistem yang dapat memanfaatkan cloud sistem
2. Dapat digunakan sebagai referensi untuk membangun sistem blockchain dengan 3 nodes.
3. Bagi institusi akan memperoleh referensi alternatif dalam membangun sistem memanfaatkan teknologi blockchain.

1.5 Batasan Masalah

Batasan masalah pada proposal ini yakni :

1. Penelitian bersifat simulasi.
2. Penelitian difokuskan pada performansi sistem yang dirancang
3. Menggunakan Hyperledger Caliper untuk pengujian performansi sistem

1.6 State of the art

Berikut merupakan beberapa penelitian terkait studi tentang perancangan sistem blockchain pada dan perancangangan sistem pertukaran energi menggunakan blockchain beserta penggunaannya seperti yang ditunjukkan padal tabel 1.6.

TABEL1.6. STATE OF ART

Nama Peneliti	Judul	Identifikasi Masalah	Tujuan	Metode	Hasil yang dicapai
Anak Agung Gde Agung, Rini handayani	Blockhain for smartgrid [6]	Keamanan transaksi pada smartgrid	Menguji sistem blockchain pada smartgrid	Simulasi Blockchain ethereum dan node red server	Sistem smartgrid menggunakan user interface yang terhubung ke public blockchain
Mizra Jabbar Asis Baig, M. Tariq Iqbal, Moshin Jamil, Jahangir Khan,	Design and implementation of an open-Source IoT and blockchain-based peer-to-peer energy trading platform using ESP32-S2, Node-Red and, MQTT protocol	Keamanan transaksi pada smartgrid dan penerapan IoT menggunakan MQTT protocol dikombinasikan dengan public blockchain	Tujuam membuat desain peer to peer pertukaran energi platform menggunakan Ethereum, Esp32 dan MQTT protocol	Merancang dan mensimulasikan sistem yang dibuat	Sebuah platform pertukaran energi yang berkerja dengan baik dengan mengimplementasikan public blockchain

Xiaoqiong Xu, Athanasios V. Vasilakos, Gang Sun, Long Luo, Huilong Cao, Hongfang Yu	*Latency performance modeling and analysis for hyperledger fabric blockchain network*	Latensi yang terjadi pada sistem blockchain	Menganalisa performansi private blockchain dalam hal ini Hyperledger fabric versi 1.4. menggunkan 2 atau 4 node organisasi	Simulasi menggunakan Hyperledger caliper dengan konfigurasi 2 atau 4 nodes	Model performansi latensi konfigurasi 2 atau 4 nodes organisasi

Dalam state of art ini kami menemukan beberapa kelemahan misalkan perancangan hanya berupa simulasi yang bersifat pemodelan di localhost, penelitian 1 dan 2 pada table menggunakan sistem blockchain public yang transaksinya bisa di akses oleh pubic dan tidak dianjurkan jika rancangan tersebut di tujukan untuk perusahaan yang bersifat privat, pada reference ketiga perancangan sistem minim fitur karena masih menggunakan Hyperledger fabric versi 1.4 yang oleh IBM dan linux foundation sendiri mengklaim masih belum aman dan menggunakan 2 atau 4 node konfigurasi. Pada penelitan ini akan dirancang sistem manajemen trading enegi menggunakan hyperledger fabric 2.0 dengan menggunakan 3 nodes organisasi .

1.7 Kerangka Pikir

Pada Gambar 1.7 Kerangka Pikir Penelitian menjelaskan mengenai alur penelitian yang akan dilakukan. Pada tahap pertama menjelaskan permasalahan yang ada yaitu peenelitian sebelumnya tanpa menggunakan framework sehingga masih menggunkan jaringan public atau jaringan Ethereum, blockchain yang terbentuk masih bersifat public.

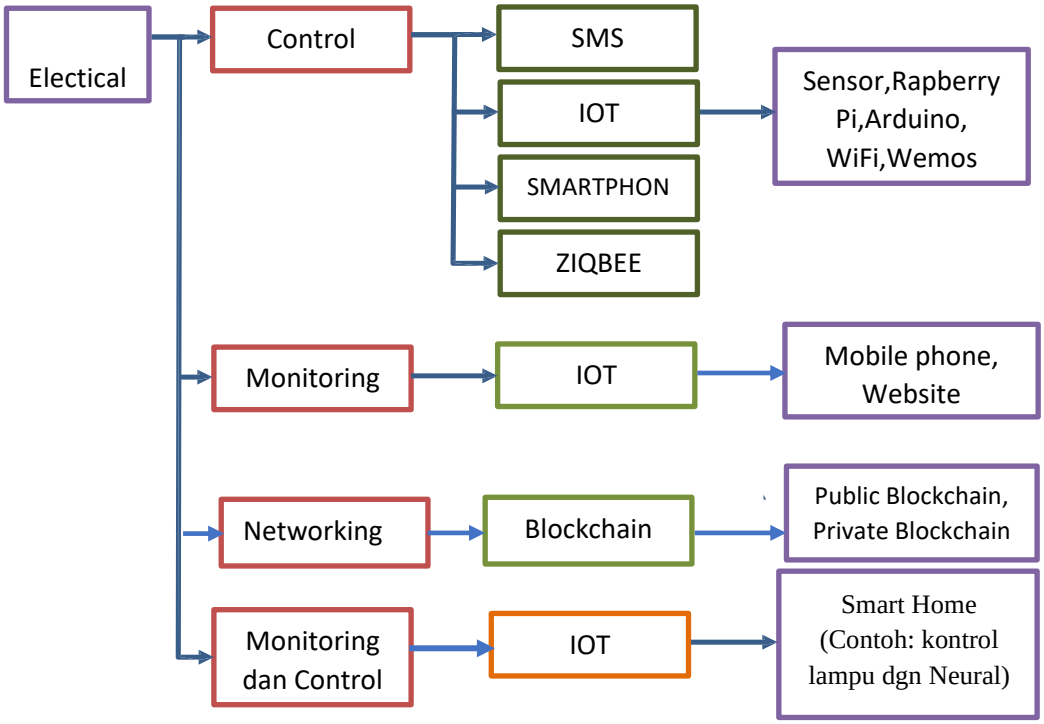
Gambar 1.7. Kerangka Pikir

1.8 *Novelty* dan Kontribusi

Berdasarkan literatur yang tertera pada *state of the art* maka dapat di simpulkan bahwa telah banyak penelitian yang dilakukan untuk melakukan konfigurasi system dalam pemanfaatan Iot, monitoring dan blockhain seperti yang ditunjukkan pada gambar 1.8.

ini memTesusiliki beberapa kontribusi yaitu:

1. Memberikan performansi dari konfigurasi 3 nodes organisasi pada Hyperledger fabric versi 2.0 atau validator yang merupakan kebaruan yang dalam beberapa jurnal menggunakan konfigurasi genap yaitu 2 atau 4 yang belum pernah dilakukan oleh peneliti sebelumnya dalam merancang permissioned blockchain terkhususnya Hyperledger fabric.
2. Menggunakan Cloud sistem untuk menyimpan data sensorik sehingga dapat dihasilkan model yang lebih dinamis dan tidak membebani state database.
3. Pengunaan blockchain private pada trading energi management



Gambar 1.8. Alur Penelitian

1.9 Sistematika Penulisan

Penulisan draft proposal penelitian ini terdiri dari 3 bab yang masing-masing terdiri dari sub-sub bab dengan sistematika sebagai berikut :

a. Bab 1 pendahuluan menguraikan tentang latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, *state of the art* dan batasan masalah pada penelitian ini
b. Bab 2 tinjauan pustaka menguraikan dan membahas teori dan konsep yang relevan dengan permasalahan yang diteliti.
c. Bab 3 metode penelitian menguraikan tentang metode yang digunakan baik yang berhubungan dengan teknik pengumpulan data sampai dengan teknik analisis data dan informasi yang digunakan
d. Bab 4 hasil dan pembahasan menunjukkan hasil yang dicapai dalam penelitian ini dan membahas tentang hasil yang diperoleh dan memberikan analisa terhadap hasil tersebut
e. Bab 5 kesimpulan dan saran menyimpulkan hasil yang diperoleh dari penelitian ini dan menjawab pertanyaan dari rumusan masalah dan saran berisi hal yang perlu ditambahkan pada penelitian ini

BAB II TINJAUAN PUSTAKA

2.1 Tinjauan Teori

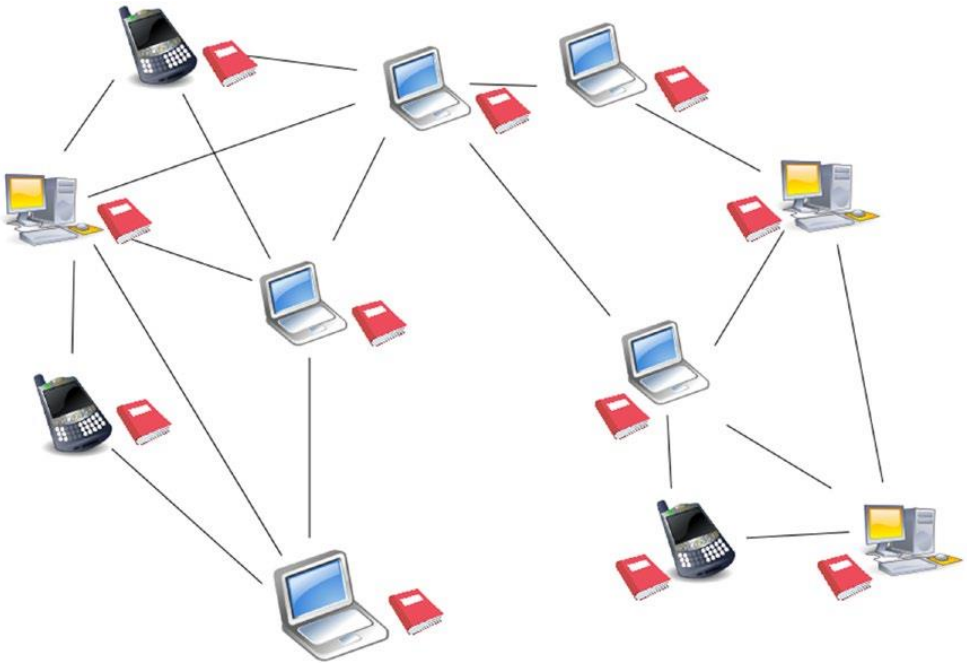
Dalam bab ini akan dibahas literatur studi yang diambil dari beberapa referensi yang menjelaskan tentang blockchain, Hyperledger fabric, Hyperledger caliper, Internet of Things, Arduino, CouchDb, NodeJs dan materi pendukung lainnya yang berkaitan dengan penelitian ini.

2.2 Blockchain

Blockchain dapat dianggap sebagai database yang didistribusikan, atau diduplikasi, di seluruh node. Inovasi dari blockchain yang spesifik adalah kemampuan database jaringan ini untuk mensinkronkan urutan transaksi, bahkan ketika beberapa node di jaringan menerima transaksi dalam berbagai urutan [5]. Blockchain merupakan kombinasi dari tiga teknologi yaitu :

1. Jaringan peer-to-peer: Sekelompok node yang dapat berkomunikasi di antara mereka sendiri tanpa bergantung pada otoritas pusat tunggal dan karena itu tidak menyajikan satu titik kegagalan.
2. Kriptografi asimetris: Cara node ini mengirim pesan dienkripsi untuk penerima tertentu sehingga siapa pun dapat memverifikasi keaslian pengirim, tetapi hanya penerima yang dituju yang dapat membaca isi pesan. Di Bitcoin dan Ethereum, asimetris kriptografi digunakan untuk membuat satu set kredensial untuk akun Anda, untuk memastikan bahwa hanya Anda yang dapat mentransfer token yang Anda miliki.
3. Hashing kriptografis: Cara untuk menghasilkan "sidik jari" yang unik untuk data apa pun, memungkinkan perbandingan cepat dari kumpulan data yang besar dan cara aman untuk memverifikasi bahwa data belum diubah; baik di Bitcoin dan Ethereum, data pohon Merkle struktur digunakan untuk mencatat urutan kanonik transaksi, yang kemudian di-hash menjadi "sidik

jari" yang berfungsi sebagai dasar perbandingan untuk node di dalam jaringan, dan sekitarnya yang dapat disinkronkan dengan cepat.



Gambar 2. 1. Ilustrasi Blockchain

2.2.1.1 Tipe Blockhain

Secara umum ada 2 tipe blockchain yaitu public blockchain dan permissioned blockchain dan dalam penelitian ini akan digunakan Hyperledger Fabric yang merupakan persmissioned blockchain

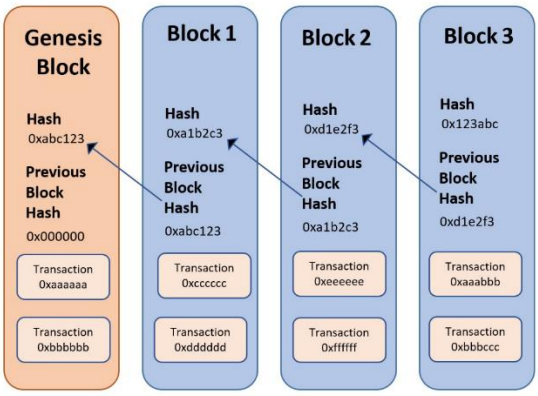
A. Public Blockchain

   Public blockchain adalah blockchain yang bersifat umum atau public yang dimana semua orang dapat mengaksesnya darimanapun dan kapanpun.

B. Permissioned blockchain atau private blockchain adalah blockchain di mana izin menulis dan membaca data diatur terpusat oleh sebuah organisasi. Izin baca dapat bersifat public jika organizasi itu mengizinkan. Seperti aplikasi manajemen basis data dan audit internal ke satu perusahaan, jadi akses publik mungkin tidak diperlukan dalam banyak kasus sama sekali, meskipun dalam kasus lain kemampuan audit publik diinginkan.

2.2.1.2 Prinsip Kerja Blockhain

Di dalam blockchain, setiap blok terkait dengan blok lain. Ada hubungan Induk-Anak antara dua blok. Hanya boleh ada satu anak untuk satu induk dan seorang anak hanya dapat memiliki satu induk tunggal. Ini membantu dalam membentuk rantai di blockchain. Dalam diagram berikut, tiga blok ditampilkan Blok 1, Blok 2, dan Blok 3. Blok 1 adalah induk dari Blok 2 dan Blok 2 adalah induk dari Blok 3. hubungan dibuat dengan menyimpan hash blok induk di header blok anak. Blok 2 menyimpan hash Blok 1 di dalamnya header dan Blok 3 menyimpan hash Blok 2 di headernya. Jadi, muncul pertanyaan siapa induk dari blok pertama? Ethereum memiliki konsep blok genesis juga dikenal sebagai blok pertama [14]. Blok ini dibuat secara otomatis ketika rantai pertama kali dimulai. Anda dapat mengatakan bahwa rantai dimulai dengan blok pertama yang dikenal sebagai Blok Genesis dan pembentukan blok ini didorong melalui file genesis.json. untuk lebih jelas perhatikan pada gambar 2.2.

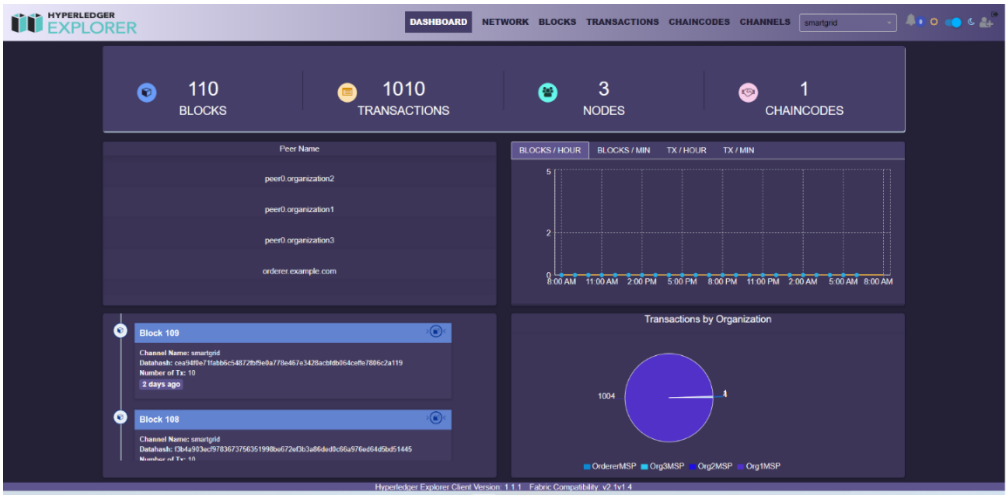


Gambar 2.2. Prinsip Kerja Blockchain

2.2.1 Blockchain Explorer

Hyperledger Explorer, dikembangkan oleh IBM, Intel, dan DTCC, kegunaanya dapat melihat, meminta, menyebarkan, atau meminta blok, transaksi dengan data terkait, informasi jaringan (misalnya, nama, status), kode rantai dan transaksi dalam

suatu block, dan informasi relevan lainnya yang terkandung dalam blockchain seperti pada gambar 2.3.



Gambar 2.3. Blockexplorer

2.2.2 Hyperledger Fabric

Hyperledger fabric saat ini adalah framework permissioned blockchain yang paling popular dan proyek yang diadopsi secara luas di bawah Hyperledger Umberella. Fabric dilengkapi dengan komponen modular seperti peers (rekan), smart contract, dan channel yang membuatnya cocok untuk membangun dan mengelola aplikasi blockchain tingkat perusahaan multiledger. Smart contract-nya dapat ditulis dalam berbagai bahasa seperti Go, JavaScript, dan Java.

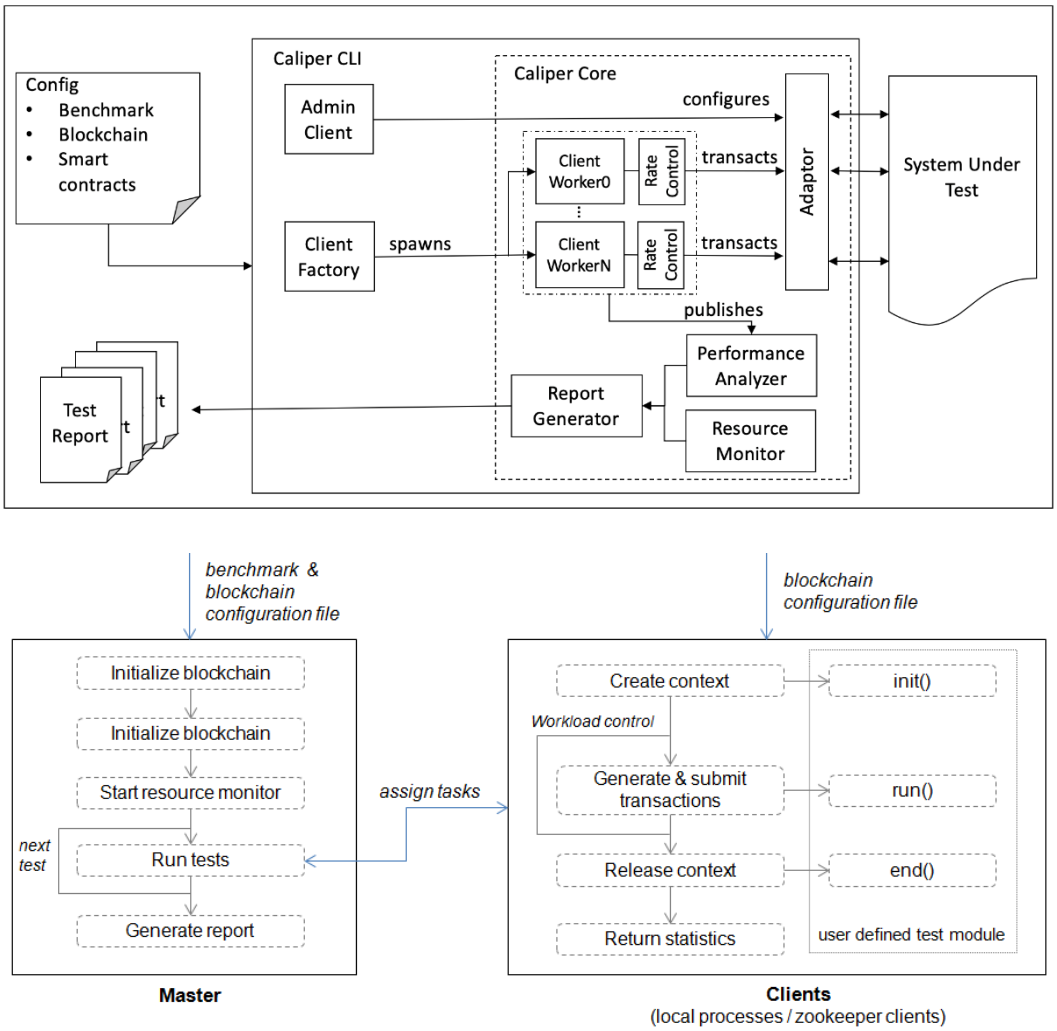
Arsitektur Hyperledger memungkinkan beberapa perusahaan untuk bergabung dan melakukan transaksi sebagai satu konsorsium. Demikian juga, ketika kebutuhan bisnis tumbuh, perusahaan dapat menjadi anggota beberapa konsorsium sekaligus. Memang, kebijakan(proof of authority) adalah satu dari komponen Fabric yang paling kuat, kebijakan(auth) datang dengan konfigurasi yang memungkinkan baik sederhana (terdiri dari dua anggota) dan kompleks (terdiri dari konsorsium tiple, masing-masing dengan banyak anggota) jaringan blockchain untuk beroperasi.

2.2.3 Hyperledger Caliper

Hyperledger Caliper adalah alat untuk mengukur efisiensi permissioned blockchain. Ini dapat membantu pengguna untuk membandingkan berbagai blockchain di lingkungan yang sama [14]. Caliper adalah alat benchmarking blockchain yang memungkinkan pengguna untuk menilai keberhasilan implementasi blockchain terhadap kumpulan kasus penggunaan yang telah ditentukan, dapat menghasilkan laporan dengan berbagai metrik kinerja [15][16]. Yang ini dirancang untuk diperpanjang, memungkinkannya bekerja dengan solusi pemantauan dan infrastruktur yang paling populer [17]. Throughput, latensi, tingkat kinerja, dan pemanfaatan sumber daya Memori CPU adalah semua metrik yang dapat dipantau oleh Caliper. Ini dicapai dengan mendengarkan stempel waktu (Timestamp) transaksi dan mengukur metrik [18]. Dalam Penelitian ini platform benchmarking ini digunakan untuk mengukur kinerja blockchain.

Hyperledger Caliper dapat diabstraksikan menjadi dua komponen: Caliper Core: Paket Core mengimplementasikan fungsi inti untuk menjalankan benchmark, termasuk: Caliper CLI: Paket CLI juga disediakan untuk kenyamanan menjalankan benchmark Pembuatan beban klien: Klien berinteraksi melalui adaptor untuk mendorong beban tolok ukur, ditentukan oleh mekanisme kontrol kecepatan. Pemantauan Sumber Daya: berisi operasi untuk memulai/menghentikan monitor dan mengambil status konsumsi sumber daya dari sistem blockchain backend, termasuk CPU, memori, IO jaringan, dll. Analisis Kinerja: berisi operasi untuk membaca statistik kinerja yang telah ditentukan sebelumnya (termasuk TPS, penundaan, rasio keberhasilan, dll) dan mencetak hasil benchmark. Metrik kunci dicatat saat menjalankan NBI blockchain dan digunakan nanti untuk menghasilkan statistik. Pembuatan Laporan: berisi operasi untuk menghasilkan laporan pengujian format HTML Adaptor Caliper: Adaptor digunakan untuk mengintegrasikan sistem blockchain yang ada ke dalam kerangka Caliper. Setiap adaptor mengimplementasikan 'Antarmuka Blockchain Caliper' dengan menggunakan SDK asli blockchain yang sesuai atau API RESTful untuk memetakan operasi seperti menyebarkan kontrak pintar pada blockchain backend, menjalankan kontrak,

menanyakan status dari buku besar, dll. Aristektur caliper benchmark ini dapat dilihat pada gambar 4.

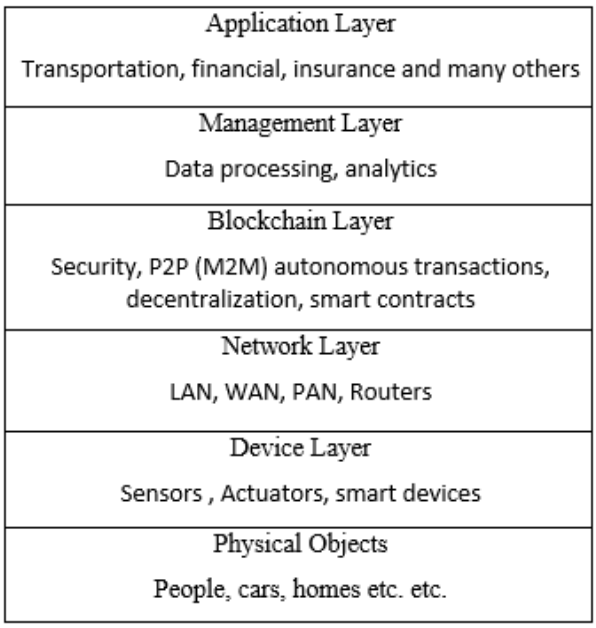


Gambar 2.4. Hyperledger caliper arsitektur

2.2.4 Internet of Things (IOT)

Dalam arti luas, Internet of Things (IoT) mengacu pada jaringan global dari perangkat yang saling berhubungan atau "hal"[9]. Ini akan menjadi masa depan Internet, yang saat ini merupakan jaringan global komputer yang saling berhubungan, termasuk Ponsel Pintar dan personal komputer. Hal-hal di IoT mengacu pada perangkat fisik sehari-hari yang bukan komputer utama tetapi memiliki perangkat keras komputasi yang tertanam (mikrokontroller). Model IoT

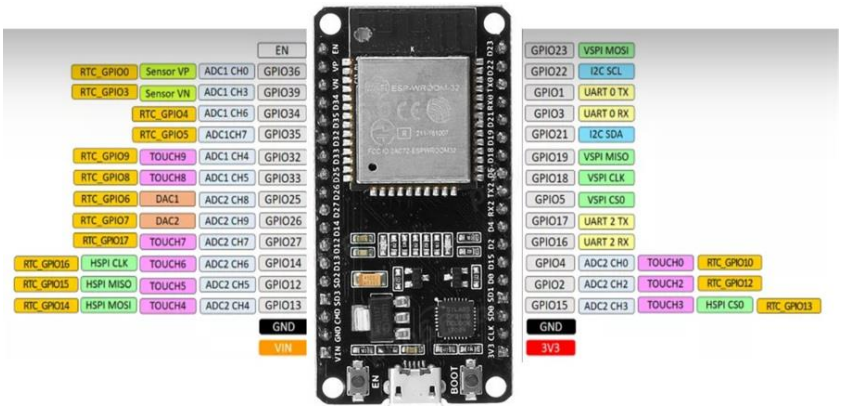
memilki lima lapis model yang dapat disesuaikan dengan model berbasis blockchain dengan: menambahkan lapisan blockchain di atas lapisan network [2]. Lapisan ini akan menjalankan kontrak pintar, dan menyediakan layanan keamanan, privasi, integritas, otonomi, skalabilitas, dan desentralisasi ke ekosistem IoT. Lapisan manajemen dalam hal ini hanya dapat terdiri dari perangkat lunak yang terkait ke analitik dan pemrosesan, dan keamanan dan kontrol dapat dipindahkan ke lapisan blockchain. Hal ini dapat divisualisasikan pada gambar 2.3.



Gambar 2.5. Blockchain Based IoT Model

2.2.5 Arduino

Arduino adalah mikrokotroller singelboard, dirancang untuk mengontrol rangkaian secara logis. Arduino memiliki komponen utama chip sirkuit terpadu yang dapat diprogram menggunakan bahasa C++ [8]. Mikrokontroler ini adalah jenis AVR yang diproduksi oleh perusahaan Atmel. Mikrokontroller ini dapat membaca input, memproses program, dan menghasilkan banyak output berdasarkan kebutuhan kita. Dalam penelitian ini menggunakan Arduino ESP32.Gambar 2.6 menunjukkan konfigurasi pin apada ESP32.



Gambar 2.6. Arduino Esp32 Pins

2.2.6 Arduino IDE

IDE adalah singkatan dari "Integrated Development Environment": ini adalah perangkat lunak resmi yang diperkenalkan oleh Arduino.cc, yang terutama digunakan untuk mengedit, menyusun, dan mengunggah kode di Perangkat Arduino. Hampir semua modul Arduino kompatibel dengan perangkat lunak ini yang merupakan sumber terbuka dan siap pakai tersedia untuk diinstal dan mulai mengompilasi kode dan logika developer [25]. Beberapa keuntungan mengunakan Arduino IDE:

1. Arduino IDE adalah perangkat lunak sumber terbuka yang terutama digunakan untuk menulis dan menyusun kode Modul Arduino.

2. Ini adalah perangkat lunak Arduino resmi, membuat kompilasi kode terlalu mudah bahkan untuk orang biasa sekalipun tidak ada pengetahuan teknis sebelumnya atau dapat digunakan oleh orang pada umumnya.

3. Mudah tersedia untuk sistem operasi seperti MAC, Windows, Linux dan berjalan di Platform Java yang dilengkapi dengan fungsi dan perintah bawaan yang memainkan peran penting untuk debugging, pengeditan, dan mengkompilasi kode di lingkungan.

4. Berbagai modul Arduino tersedia termasuk Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino Mikro dan banyak lagi.

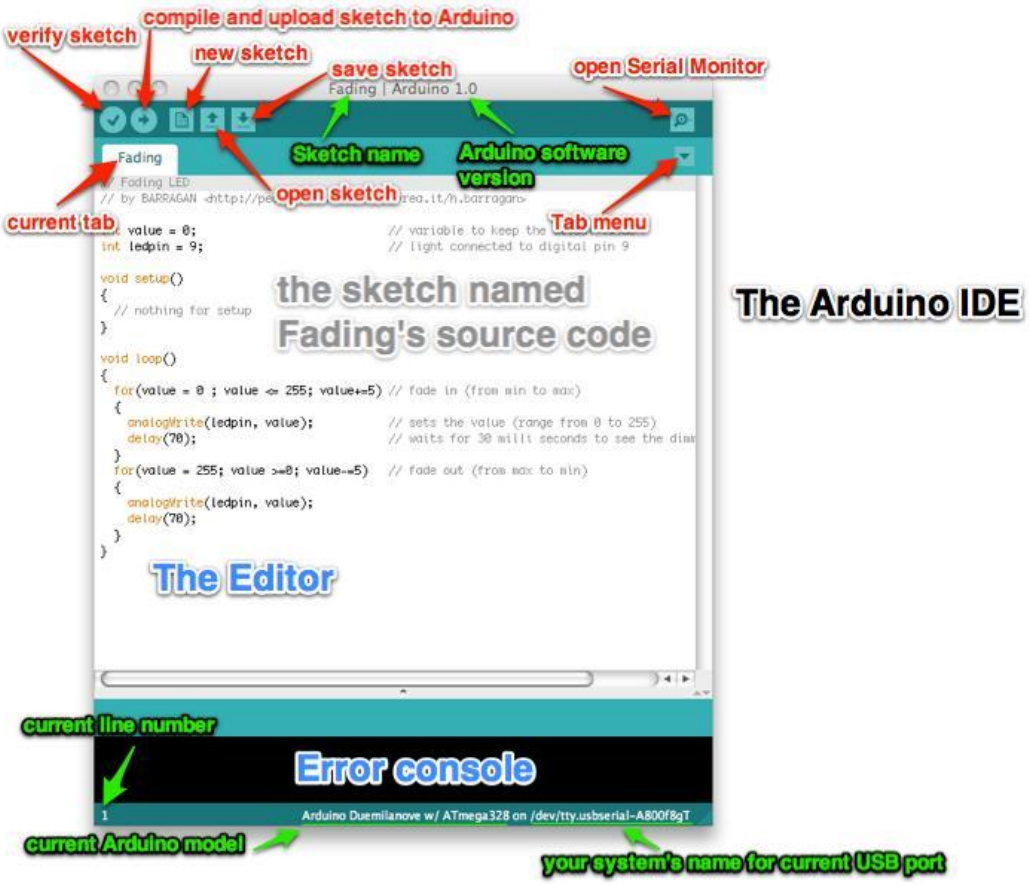
5. Masing-masing berisi mikrokontroler di papan yang benar-benar diprogram dan menerima informasi dalam bentuk kode.

6. Kode utama, juga dikenal sebagai sketsa, yang dibuat di platform IDE pada akhirnya akan menghasilkan Hex File yang kemudian ditransfer dan diunggah di pengontrol di papan tulis.

7. Lingkungan IDE terutama berisi dua bagian dasar: Editor dan Compiler di mana mantan digunakan untuk menulis kode yang diperlukan dan kemudian digunakan untuk mengkompilasi dan mengunggah kode ke kode yang diberikan Modul Arduino.

8. Lingkungan ini mendukung bahasa C dan C++.

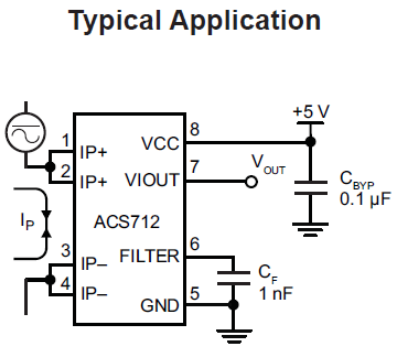
Untuk tampilan ditunjukkan pada gambar di bawah ini:



Gambar 2.7. Arduino IDE

2.2.7 Sensor ACS712

ACS712 merupakan sebuah modul sensor untuk mengukur arus baik arus AC maupun arus DC yang menggunakan teknologi Hall Effect. Yang dimaksud dengan Hall Effect yaitu mengalirkan jalur beban yang diukur melalui suatu media konduksi tembaga untuk menghasilkan medan magnet.Medan magnet tersebut kemudian diubah menjadi tegangan yang proporsional terhadap arus yang mengalir oleh sebuah IC Hall seperti yang ditunjukkan gambar gambar 2.8.



Gambar 2.8. Typical Modul Sensor ACS712

2.2.8 Modul Sensor Tegangan

Prinsip kerja modul sensor tegangan yaitu didasarkan pada prinsip penekanan resistansi, dan dapat membuat tegangan input berkurang hingga 5 kali dari tegangan asli. Bentuk modul sensor tegangan seperti ditunjukkan pada gambar 2.9 berikut:

Fitur-fitur dan kelebihannya:

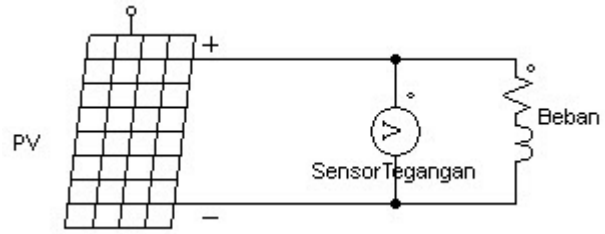
1. Variasi Tegangan masukan: DC 0 - 25 V
2. Deteksi tegangan dengan jangkauan: DC 0.02445 V - 25 V
3. Tegangan resolusi analog: 0,00489 V
4. Tegangan DC masukan antarmuka: terminal positif dengan VCC, negatif dengan GND
5. Output Interface: "+" Koneksi 5 / 3.3V, "-" terhubung GND, "s" terhubung Arduino pin A0

6. DC antarmuka masukan: red terminal positif dengan VCC, negatif dengan GND

Prinsip kerja modul sensor tegangan ini dapat membuat tegangan input mengurangi 5 kali dari tegangan asli. Sehingga, sensor hanya mampu membaca tegangan maksimal 25 V bila diinginkan Arduino analog input dengan tegangan 5 V, dan jika untuk tegangan 3,3 V, tegangan input harus tidak lebih dari 16.5 V. Pada dasarnya pembacaan sensor hanya dirubah dalam bentuk bilngan dari 0 sampai 1023, karena chip Arduino AVR memiliki 10 bit, jadi resolusi simulasi modul 0,00489 V yaitu dari (5 V / 1023), dan tegangan input dari modul ini harus lebih dari 0,00489 V x 5 = 0,02445 V. Sehingga dapat dirumuskan seperti persamaan (1.1) berikut :

$$Volt = (Vout \times 0.00489) \times 5 \qquad (1.1)$$

Modul tegangan ini disusun secara parallel terhadap beban. Pada gambar 2 menunjukan contoh cara mengukur tegangan beban pada panel surya (PV) dengan sensor tegangan yang dihubungkan secara paralel, seperti gambar 2.9 berikut :



Gambar 2.9. Bentuk rangkaian sensor tegangan untuk mengukur tegangan beban pada Panel Surya

2.2.9 Docker

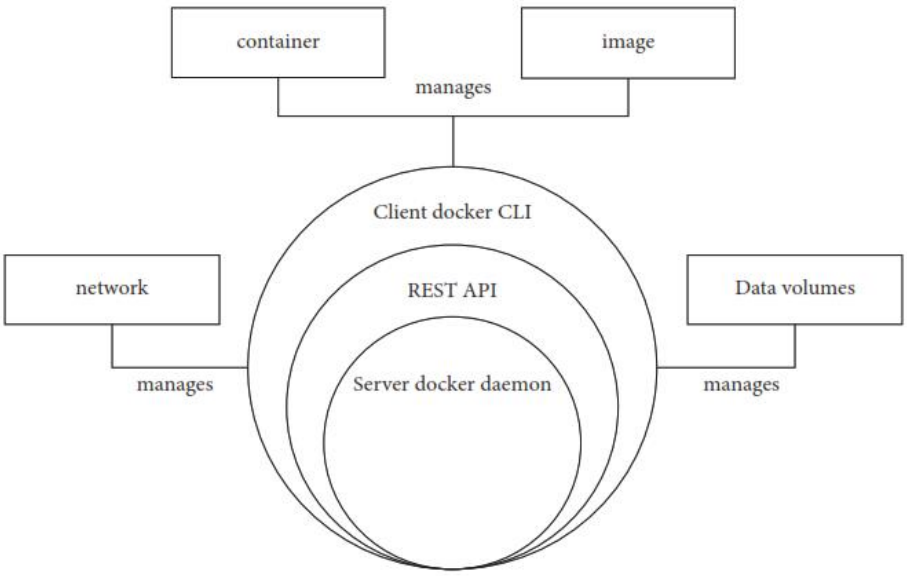
Dalam komputasi awan, Platform as a Service (PaaS) merupakan mode layanan yang menyediakan pengguna dengan perangkat lunak platform dan lingkungan pengembangan (sumber [22-25]). Docker, sebuah proyek sumber terbuka yang

dikembangkan menggunakan bahasa pemrograman Golang, dapat mempercepat implementasi mode PaaS (sumber [26]). Dalam pengembangan perangkat lunak, sering kali menghadapi masalah ketidakmampuan untuk menggunakan kembali perangkat lunak karena perbedaan lingkungan pengembangan. Docker, bagaimanapun, dapat efektif memecahkan masalah tersebut. Baik dalam tahap pengembangan maupun pengujian perangkat lunak, Docker mampu menciptakan lingkungan yang konsisten. Dengan kata lain, Docker dapat mengatur konfigurasi kontainer untuk memastikan ketergantungan yang konsisten dari semua konfigurasi dalam kontainer, sehingga memungkinkan pengembangan, pengujian, dan rilis perangkat lunak dilakukan dalam lingkungan yang sama.

Docker memiliki tiga keunggulan. Pertama, ia dapat menggunakan repositori gambar (image repositories). Kedua, Docker mampu melakukan penyebaran (deployment) secara berkelanjutan. Ketiga, Docker memiliki tingkat pemanfaatan sumber daya yang tinggi. Dalam penelitian ini, penulis menggunakan manajemen berbasis peran (role-based) pada sistem manajemen Docker. Setelah server Docker dijalankan dan gambar (image) didapatkan, proses instalasi sistem operasi, repositori, dan layanan aplikasi dapat diselesaikan dengan cepat, sehingga mempercepat proses pengembangan sistem. Karena Docker tidak memerlukan sistem operasi tambahan, maka ruang disk yang cukup terhindari, sumber daya sistem yang melimpah dapat dihemat, dan tingkat pemanfaatan sumber daya sistem meningkat. Docker dapat dianggap sebagai virtualisasi tingkat sistem operasi, sedangkan virtualisasi mesin virtual (virtual machine) diimplementasikan pada tingkat perangkat keras.

Arsitektur Docker ditunjukkan dalam Gambar 1. Dalam Gambar 1 terlihat bahwa Server Docker adalah daemon kernel yang dapat digunakan tidak hanya secara lokal, tetapi juga di server jarak jauh. Jembatan komunikasi antara Server dan Klien menggunakan REST API . Sebagai Klien, Docker Klien CLI bisa, selain menyediakan antarmuka yang sesuai untuk pengguna, mengelola wadah dan gambar. Setelah gambar dikemas, pengguna dapat membuat wadah berdasarkan gambar dan kemudian mengoperasikannya, di mana Server dapat dipanggil untuk

mengontrol sumber daya disk. Compose di Docker digunakan untuk melakukan penataan cepat cluster kontainer sebelum pengoperasian beberapa kontainer. Praktis, sulit untuk memberikan kondisi pengoperasian yang baik untuk sistem dengan mengandalkan hanya pada satu wadah, dan database dan server harus dimulai secara bersamaan untuk menjamin pengoperasian sistem suara. Selain itu, layanan dan proyek adalah dua konsep penting dalam Composee first artinya bisa mendefinisikan layanan yang dibutuhkan oleh aplikasi dalam hal nama, jaringan, lingkungan konfigurasi, ketergantungan kondisi, dll. Layanan dilakukan dalam perincian kontainer dan setiap kontainer membawa satu layanan perdetik mengacu pada keseluruhan proyek yang perlu dilaksanakan oleh pengguna konten (penyebaran kontainer, manajemen kontainer,dll.) diungkapkan dalam file YMAL dapat dianggap sebagai project dan Compose biasanya akan mengelola proyektanpa gangguan tambahan [27].



Gambar 2.10. Docker Architecture

2.2.10 CouchDB

CouchDB adalah database bersifat Non Sql (NOSQL) yang sepenuhnya mencakup web. Menyimpan data dengan dokumen JSON. Akses dokumen dapat menggunakan terminal atau browser web, melalui HTTP. Kueri, gabungkan, dan

ubah dokumen dengan JavaScript dapat dilakukan. CouchDB bekerja dengan baik dengan aplikasi web dan seluler modern. Dapat mendistribusikan data, secara efisien menggunakan replikasi inkremental CouchDB. CouchDB mendukung pengaturan master-master dengan deteksi konflik otomatis. CouchDB hadir dengan serangkaian fitur, seperti transformasi dokumen on-the-fly dan pemberitahuan perubahan waktu nyata, yang membuat pengembangan web menjadi mudah. Itu bahkan dilengkapi dengan konsol administrasi web yang mudah digunakan, disajikan langsung dari CouchDB dengan penskalaan terdistribusi. CouchDB sangat tersedia dan toleran terhadap partisi, tetapi juga pada akhirnya konsisten. CouchDB memiliki mesin penyimpanan yang toleran terhadap kesalahan yang mengutamakan keamanan data.

2.2.11 NodeJS

CouchDB Sebagai runtime JavaScript berbasis peristiwa asinkron, Node.js dirancang untuk membangun aplikasi jaringan yang skalabel. Banyak koneksi dapat ditangani secara bersamaan. Pada setiap koneksi, panggilan balik diaktifkan, tetapi jika tidak ada pekerjaan yang harus dilakukan, Node.js akan dalam keadaan tidur. Ini berbeda dengan model konkurensi yang lebih umum saat ini, di mana utas OS digunakan. Jaringan berbasis thread relatif tidak efisien dan sangat sulit digunakan. Selain itu, pengguna Node.js bebas dari kekhawatiran proses dead-lock, karena tidak ada kunci. Hampir tidak ada fungsi di Node.js yang secara langsung melakukan I/O, sehingga proses tidak pernah memblokir kecuali ketika I/O dilakukan menggunakan metode sinkron dari pustaka standar Node.js. Karena tidak ada yang menghalangi, sistem scalable yang cocok untuk dikembangkan di Node.js. (Sumber: https://nodejs.org/en/about/)

2.2.12 Firebase

Firebase Realtime Database adalah database yang dihosting di cloud. Data disimpan sebagai JSON dan disinkronkan secara realtime ke setiap klien yang terhubung. Saat Anda membuat aplikasi lintas platform dengan platform Apple, Android, dan

SDK JavaScript kami, semua klien Anda berbagi satu instans Realtime Database dan secara otomatis menerima pembaruan dengan data terbaru

Firebase Realtime Database memungkinkan Anda membangun aplikasi yang kaya dan kolaboratif dengan mengizinkan akses aman ke database langsung dari kode sisi klien. Data disimpan secara lokal, dan bahkan saat offline, peristiwa waktu nyata terus menyala, memberikan pengalaman responsif kepada pengguna akhir. Saat perangkat mendapatkan kembali koneksi, Database Realtime menyinkronkan perubahan data lokal dengan pembaruan jarak jauh yang terjadi saat klien offline, menggabungkan konflik apa pun secara otomatis.

Realtime Database menyediakan bahasa aturan berbasis ekspresi yang fleksibel, yang disebut Aturan Keamanan Firebase Realtime Database, untuk menentukan bagaimana data Anda harus terstruktur dan kapan data dapat dibaca atau ditulis. Saat terintegrasi dengan Firebase Authentication, developer dapat menentukan siapa yang memiliki akses ke data apa, dan bagaimana mereka dapat mengaksesnya.

Database Realtime adalah database NoSQL dan dengan demikian memiliki optimasi dan fungsionalitas yang berbeda dibandingkan dengan database relasional. Realtime Database API dirancang untuk hanya mengizinkan operasi yang dapat dijalankan dengan cepat. Hal ini memungkinkan Anda membangun pengalaman realtime yang hebat yang dapat melayani jutaan pengguna tanpa mengorbankan daya tanggap.

2.2.13 Socket.IO

Socket.IO adalah pustaka yang memungkinkan komunikasi latensi rendah, dua arah, dan berbasis peristiwa antara klien dan server. Itu dibangun di atas protokol WebSocket dan memberikan jaminan tambahan seperti fallback ke HTTP long-polling atau koneksi ulang otomatis. WebSocket adalah protokol komunikasi yang menyediakan saluran dupleks penuh dan latensi rendah antara server dan browser[22].

Gambar 2.11. Ilustrasi cara kerja socket.IO

2.2.14 Golang

Go telah digambarkan sebagai "C untuk abad ke-21" Ini adalah bahasa yang dikompilasi dan diketik secara statis dengan sintaks C-like tetapi lebih sederhana dan pengumpulan sampah. Itu dirancang dan dikembangkan untuk memenuhi masalah tertentu yang dialami oleh insinyur perangkat lunak di Google. Perusahaan ini biasanya menggunakan Java, C++ dan Python untuk proyek besar mereka dan beberapa proyek mereka diklaim sangat besar yang membuat beberapa masalah dengan siklus pengembangan menjadi lebih jelas [4]. Beberapa poin utamanya adalah:

1.Build time tidak diskalakan dengan baik, yang memperlambat seluruh siklus pengembangan, Sebagian disebabkan oleh kode terbatas dan pemahaman paket dalam tim yang ditambakan

3.penggunaan himpunan bagian dan pola bahasa yang berbeda di antara pengembang (misalnya dalam C++)

4. impor paket aman dan memperlambat down dibangun dengan bahasa seperti C di mana file biasanya harus dibuka sebelumnya mencapai penjaga sundulan.

Go dirancang khusus untuk mengatasi poin-poin ini melalui:

1.Mengurangi waktu pembuatan dengan model untuk manajemen ketergantungan.

2.Mengurangi bug yang timbul karena tidak ada aritmatika penunjuk dan dengan menggunakan sampah run-time pengumpul.

3.Tidak ada konversi tipe implisit (yang sering tidak terduga/dilupakan),

4.Ketik inferensi yaitu tidak perlu mendeklarasikan tipe variabel karena tipenya bisa disimpulkan pada waktu kompilasi.

5. Tidak ada sistem tipe yang rumit.

6.Spesifikasi bahasa ringkas yang memiliki sedikit kata kunci dan tidak memiliki konstruksi yang rumit.

7.Bahasa memiliki mekanisme konkurensi bawaan dan mudah digunakan.

Satu hal penting yang perlu diperhatikan di sini, yang juga ditekankan oleh pembuat Go, adalah itu konkurensi bukanlah paralelisme. Concurrency adalah tentang mengatur kode sedemikian rupa sehingga berbeda bagian dapat berjalan pada waktu yang sama bersama dengan sinkronisasi dan komunikasi antara bagian. Paralelisme adalah tentang benar-benar menjalankan berbagai hal pada saat yang bersamaan. Ini berarti bahwa konkurensi memungkinkan paralelisme.

Mekanisme bawaan untuk konkurensi berkisar pada apa yang disebut goroutine. Mereka berfungsi sebagai thread ringan yang ditangani oleh runtime Go dan harganya murah untuk memulai banyak dari mereka (dalam skala ribuan). Ada juga tipe bawaan yang disebut saluran yang memungkinkan komunikasi dan sinkronisasi yang aman antar goroutine. Tujuan Go adalah menyesuaikan antara kemudahan penggunaan bahasa dinamis seperti Python dan keamanan serta kecepatan bahasa kompilasi yang diketik secara statis seperti C++ atau Java. Itu bahasa dirancang dengan prinsip sederhana dan bersih yaitu konsep seharusnya mudah dimengerti, dan kesalahan keselamatan harus dideteksi.

2.2.15 Javascript

JavaScript (atau "JS") adalah bahasa pemrograman yang paling sering digunakan untuk skrip sisi klien dinamis di halaman web, tetapi juga sering digunakan di sisi server, menggunakan runtime seperti Node.js.

JavaScript terutama digunakan di browser, memungkinkan developer untuk memanipulasi konten halaman web melalui DOM, memanipulasi data dengan AJAX dan IndexedDB, menggambar grafik dengan kanvas, berinteraksi dengan

perangkat yang menjalankan browser melalui berbagai API, dan banyak lagi. JavaScript adalah salah satu bahasa yang paling umum digunakan di dunia, karena perkembangan terbaru dan peningkatan kinerja API yang tersedia di browser.

Diciptakan sebagai bahasa server-side oleh Brendan Eich (kemudian dipekerjakan oleh Netscape Corporation), JavaScript segera datang ke Netscape Navigator 2.0 pada bulan September 1995.

Pada November 1996, Netscape mulai bekerja dengan Ecma International untuk menjadikan JavaScript sebagai standar industri. Sejak itu, JavaScript standar disebut ECMAScript dan ditentukan di bawah ECMA-262, yang edisi terbarunya (kedua belas, ES2021) tersedia mulai Juni 2021.

Popularitas JavaScript telah berkembang lebih jauh melalui platform Node.js yang sukses—lingkungan runtime JavaScript lintas platform paling populer di luar browser. Node.js - dibuat menggunakan Mesin JavaScript V8 Chrome - memungkinkan pengembang menggunakan JavaScript sebagai bahasa scripting untuk mengotomatiskan berbagai hal di komputer dan membangun server HTTP dan WebSockets yang berfungsi penuh [23].

2.2.16 Python

Pada tahun 1991, bahasa Python dikembangkan oleh Guido van Rossum. Ada cerita menarik di baliknya memberikan nama "Python" untuk pemrograman bahasa. Pada saat pengembangan python, the pengembang sedang membaca skrip “Monty's Python Terbang yang merupakan serial BBC. Saat membaca ini buku dia mendapat ide untuk memberi nama pemrograman bahasa sebagai "Python" memiliki singkat dan unik. Python berorientasi objek, ditafsirkan, dan bahasa pemrograman interaktif. Ini menyediakan perfomansi tingkat tinggi struktur data seperti list, tuple, set, array asosiatif (disebut kamus), dinamis mengetik dan mengikat, modul, kelas, pengecualian, manajemen memori otomatis, dll. Ini juga digunakan untuk sistem komputasi paralel dan memiliki sintaks yang relatif sederhana dan mudah untuk pengkodean dan tetap saja itu adalah bahasa pemrograman yang kuat. Python memiliki juru bahasa untuk java yang dikenal sebagai JPython, yang mirip dengan

penerjemah untuk C bahasa. Python memiliki banyak keunggulan dibanding apapun bahasa lain, seperti memiliki berbagai perpustakaan yang mengurangi kode menjadi sepertiga untuk programmer dan karena Python ini telah mencapai + puncak tertinggi dalam hal Pembelajaran Mesin.

2.2.17 C++

Bahasa pemrograman C++ diciptakan oleh Bjarne Stroustrup dan timnya di Bell Laboratories (AT&T, USA) untuk membantu mengimplementasikan proyek simulasi dalam berorientasi objek dan cara yang efisien. Versi paling awal, yang awalnya disebut sebagai "C dengan kelas", tanggal kembali ke 1980. Seperti namanya C ++, C ++ berasal dari C pemrograman bahasa: ++ adalah operator kenaikan di C. Sebagai awal tahun 1989 Komite ANSI (American Institut Standar Nasional) adalah didirikan untuk membakukan bahasa pemrograman C++. Tujuannya adalah untuk memiliki sebanyak mungkin vendor kompiler dan pengembang perangkat lunak mungkin menyepakati deskripsi terpadu bahasa untuk menghindari kebingungan yang disebabkan oleh berbagai dialek. Pada tahun 1998 ISO (Organisasi Internasional untuk Standardisasi) menyetujui standar untuk C++ (ISO/IEC 14882).

C ++ bukan bahasa berorientasi objek murni tetapi hibrida yang berisi fungsionalitas dari bahasa pemrograman C. Ini berarti Anda memiliki semua fitur yang tersedia di C:

1. program modular yang dapat digunakan secara universal
2. efisien, dekat dengan pemrograman mesin
3. program portabel untuk berbagai platform.

Sejumlah besar kode sumber C yang ada juga dapat digunakan dalam program C++. C ++ mendukung konsep pemrograman berorientasi objek (atau disingkat OOP), yang mana:

1. abstraksi data, yaitu pembuatan kelas untuk mendeskripsikan objek
2. enkapsulasi data untuk akses terkontrol ke data objek

3. pewarisan dengan membuat kelas turunan (termasuk beberapa kelas turunan)
4. polimorfisme (Yunani untuk multiform), yaitu pelaksanaan instruksi yang dapat memiliki berbagai efek selama eksekusi program.

Berbagai elemen bahasa ditambahkan ke C++, seperti referensi, templat, dan pengecualian penanganan. Meskipun elemen bahasa ini tidak sepenuhnya berorientasi objek fitur pemrograman, tapi penting untuk implementasi program yang efisien [22].