## BAB III. METODOLOGI PENELITIAN

###  Metode Penelitian

Metode yang dipakai dalam penelitian ini adalah menggunakan pendekatan Eksperimental.

#### Desain

 Tujuan bagian ini adalah untuk menetapkan kerangka kerja yang menjalankan percobaan pada DLT. Ini akan mendukung konfigurasi teknologi dan pengukuran aktual serta evaluasi hasil. Desain didefinisikan sebagai berikut:

Untuk menentukan berbagai tujuan (Througput, Latency), dibahas, dibagi menjadi tiga fase, persiapan, pengukuran, dan evaluasi, yang akan dibahas di bawah ini.

 

Gambar 3.1. Desain Kerangka Kerja

Secara garis besar, penelitian ini melakukan dua jenis percobaan yaitu: Eksperimen pertama adalah untuk menganalisis kinerja platform Hyperledger Fabric v1.0. Evaluasi kinerja platform dinilai dalam hal waktu pelaksanaan latensi dan throughput dengan memvariasikan beban kerja jumlah transaksi dan permintaan (permintaan atau permintaan) yang diminta secara bersamaan dengan delapan node yang memproses hingga 20.000 transaksi, transaksi diukur dengan pengajuan transaksi untuk konsensus oleh node, untuk menambahkan transaksi ke blok. Waktu eksekusi adalah waktu yang dibutuhkan platform untuk menambah dan menjalankan transaksi dengan sukses. Throughput dapat didefinisikan sebagai "jumlah transaksi yang berhasil per detik." Akhirnya, latensi dalam blockchain dapat diukur karena waktu yang dibutuhkan untuk platform tertentu untuk merespons setiap transaksi.

Eksperimen kedua membandingkan Hyperledger dengan Relational Database, dalam hal ini, MySQL, dengan beban data yang sama.

**3.1.2. Fase Persiapan**

 Sebelum melakukan pengukuran, testbed harus disiapkan. Ini dimulai dengan pemasangan atau konfigurasi perangkat lunak khusus yang diuji untuk memfasilitasi semua langkah berikut. Ini termasuk instalasi dan konfigurasi perangkat lunak (OS, Docker, Network) yang diperlukan untuk memfasilitasi langkah-langkah berikut. Ledger Hosts dilengkapi dengan apa pun (JSON, bahasa GO, Docker CE) yang diperlukan untuk benchmark DLT. Ini termasuk tool (Chaincode-Payload-Size, Chaincode-Scalability, Channel-Scalability), yang digunakan selama fase pengukuran untuk merekam host eksperimental. Setelah itu, host eksperimen murni dievaluasi. Dengan langkah-langkah pengaturan umum dilakukan, instalasi teknologi Ledger pada host Ledger berikut. Semua host Ledger disediakan dengan perangkat lunak yang harus mereka jalankan dan file konfigurasi, yang diperlukan untuk berfungsi di dalam testbed. Hal yang sama berlaku untuk host tolok ukur, yang digunakan untuk aplikasi tolok ukur. Akhirnya pada tahap penerapan DLT dimulai dan sistem lengkap disiapkan untuk menjalani pengukuran.

Gambar 3.2 Alur Kerja Fase Deployment

Peralatan yang dipergunakan:

* 1 Laptop i7 - 3687u, 2.1GHz, 8 GB memory, 500 GB SSD
* 8 PCs, dengan spesifikasi sebagai berikut:: CPU (Intel(R) Core 2 Duo) 2,8G + Mb.G31 Ddr2 4GB memory • 100 Mbp/s network • 120 GB HDD • Ubuntu v 18.04 LTS, VM

#### **Fase Pengukuran**



Gambar 3.3. Alur Kerja Metode Pengukuran

Gambar 3.2. menjelaskan fase pengukuran, dimulai dengan langkah konfigurasi awal. Untuk menjalankan percobaan, setiap perubahan konfigurasi akhir diterapkan ke host percobaan, seperti misalnya gangguan jaringan. Setelah inisialisasi semua monitor pada percobaan host berlangsung. Ini mungkin termasuk, pencatatan lalu lintas jaringan.

Eksekusi beban kerja kemudian diprakarsai oleh host orkestrasi, yang dilakukan tanpa mengganggu eksperimen dan menunggu hingga selesai. Alih-alih host benchmark mengambil alih dan mengoperasikan percobaan berdasarkan definisi beban kerja. Semua langkah konfigurasi dan inisialisasi sebelumnya memungkinkan sistem menjalankan percobaan tanpa tindakan eksternal apa pun.

Perubahan selama percobaan didasarkan pada waktu, langsung oleh eksekusi beban kerja pada host benchmark. Ini diperlukan, misalnya, untuk mencegah distorsi percobaan melalui lalu lintas manajemenAkhirnya setelah eksekusi beban kerja selesai, percobaan dimatikan. Monitor dihentikan pada host eksperimen dan gangguan apa pun yang diterapkan selama konfigurasi pengukuran langkah dibatalkan. Misalnya menghapus kehilangan jaringan buatan yang telah diletakkan di jaringan, untuk memungkinkan pekerjaan yang terus-menerus tidak terganggu pada testbed.

Terakhir, setiap informasi yang dikumpulkan diambil dari host percobaan dan dikumpulkan di host orkestrasi. Ini memungkinkan untuk segera mereset semua host di jaringan selain orkestrasi tuan rumah, untuk pengukuran selanjutnya.

Dalam percobaan yang dilakukan, platform testbed yang dioperasikan menggunakan parameter berikut:

* Total jumlah rantai (default - 10)
* Jumlah transaksi yang disimulasikan secara paralel pada setiap rantai (default - 10)
* Total Block di seluruh rantai (kunci didistribusikan secara merata di seluruh rantai) (default - 1.000)
* Jumlah kunci yang dibaca dan dimodifikasi secara acak oleh setiap transaksi (default - 4)
* Besar Block: Nilai ukuran untuk setiap Block (default - 200 byte)
* Jumlah transaksi di setiap blok (default – 50)

#### 3.1.4. Fase Evaluasi

Fase terakhir adalah fase evaluasi didapat dari hasil benchmarking, di mana data yang dikumpulkan harus dievaluasi, dimulai dengan fase preprocessing, yaitu:

* Tingkat kesuksesan : Ukur semua transaksi yang berhasil dan gagal untuk siklus tes.
* Transaksi & Baca Latensi: Ukur waktu untuk transaksi yang dikeluarkan untuk diselesaikan dan respons yang tersedia untuk aplikasi yang mengeluarkan transaksi. Latensi maksimum, minimum, dan rata-rata untuk siklus tes disediakan.
* Transaksi & Throughput Baca: Ukur laju aliran semua transaksi melalui sistem, dalam transaksi per detik, selama siklus.

Data yang diproses sebelumnya ini kemudian dapat dievaluasi berdasarkan metrik yang relevan. Beberapa cara evaluasi terdefinisi dan format data yang diproses sebelumnya memungkinkan untuk dengan mudah menambahkan metode evaluasi lebih lanjut.

###  Distributed Ledger

Arsitektur DLT yang kompleks dibagi menjadi empat lapisan, yaitu: Jaringan, Chain, Ledger, dan Lapisan Aplikasi untuk memudahkan analisis lebih lanjut. (Dinh TT, 2017), (Pongnumkul, 2017) Pada setiap lapisan, beberapa metrik dan faktor pengaruh ditentukan. Metrik yang berbeda dan dipengaruhi oleh berbagai faktor diukur menjadi tolok ukur konsep beban kerja DLT primer dan simulasi diperkenalkan (Pongnumkul, 2017). Berdasarkan analisis ini, sebuah kerangka kerja dirancang untuk membangun fondasi terdistribusi dari lingkungan pengujian dan melakukan pengukuran yang dapat direproduksi. Kerangka kerja ini dirancang agar teknologi dievaluasi dan lingkungan pengujian mudah dipertukarkan. Akibatnya, kerangka kerja desain diimplementasikan dengan alat benchmarking. Misalnya, pengukuran kinerja dan evaluasi, Hyperledger Fabric, percobaan dilakukan di lingkungan laboratorium yang terkontrol.

Evaluasi hasil pengukuran akan memberikan informasi tentang efek kinerja dari empat faktor, perubahan secara eksplisit dalam tingkat transaksi, beban kerja, ukuran blok dan juga dampak dari kehilangan paket. Pengukuran menunjukkan bahwa faktor-faktor dari setiap lapisan dapat secara langsung mempengaruhi kinerja seluruh jaringan, meningkatkan tingkat transaksi, menuntut beban kerja, konfigurasi memori yang tidak menguntungkan, atau kondisi jaringan yang tidak menguntungkan.

**Throughput**

Throughput didefinisikan sebagai jumlah data yang berhasil ditransfer antar node per unit waktu, biasanya diukur dalam detik seperti yang ditunjukkan pada Gambar 2.1. Throughput sering didefinisikan untuk koneksi atau sesi tertentu, tetapi dalam beberapa kasus, throughput jaringan total telah ditentukan. Idealnya, throughput harus sama dengan kapasitas. Kapasitas tergantung pada teknologi lapisan fisik yang digunakan. Kapasitas jaringan harus memadai untuk menangani beban yang terjadi, bahkan ketika maksimum sibuk dalam lalu lintas jaringan. Secara teoritis, throughput akan meningkat ketika beban yang ditawarkan meningkat, hingga maksimum kapasitas jaringan. Namun, throughput jaringan tergantung pada metode akses (misalnya, menyampaikan token atau penginderaan operator), beban jaringan, dan tingkat kesalahan. Pada gambar di bawah, ini menunjukkan situasi yang ideal, di mana throughput meningkat secara linear dengan beban yang ditawarkan, dan pada kenyataannya, di mana throughput aktual berkurang karena beban yang ditawarkan mencapai titik maksimum tertentu (Moussavi, 2011).



Gambar 3.4.. Alur Kerja Throughput

**Latency**

Latensi terkait dengan waktu yang diperlukan untuk mengirim pesan dari satu ujung jaringan ke ujung lainnya. Latensi mungkin juga merupakan jeda waktu yang diperlukan dalam mengirimkan paket data dari pengirim ke penerima. Semakin tinggi jeda waktu atau latensi, semakin tinggi risiko kegagalan akses. Latensi jaringan juga sering diartikan sebagai tingkat keterlambatan pengiriman ke jaringan komunikasi data dan suara. Latensi diukur secara ketat dalam bentuk waktu. Misalnya, jaringan untuk mengirim pesan membutuhkan 24 milidetik (milidetik) dari satu ujung ke ujung lainnya. Secara umum, ada tiga komponen latensi, yaitu, penundaan propagasi, transit, dan antrian (Peterson, 2007).

Script yang menetapkan eksperimen default yang ditulis menggunakan JavaScript Object Notation (JSON), adalah format ringkas untuk bertukar data komputer. Formatnya berbasis teks dan dapat dibaca manusia dan digunakan untuk merepresentasikan struktur data sederhana dan array asosiatif (disebut objek) (Bourhis, 2017). Format JSON digunakan untuk mengirim data terstruktur melalui koneksi jaringan dalam proses yang disebut serialisasi (Maeda, 2012).