

BAB 5

SIMPULAN DAN SARAN

5.1. Simpulan

Berdasarkan implementasi dan analisis yang telah dilakukan, beberapa kesimpulan yang dapat ditarik pada penelitian ini adalah sebagai berikut.

1. Hyperledger Fabric berhasil diimplementasikan untuk aplikasi autentikasi produk dan diberi nama Oricon.
2. Evaluasi perfoma Hyperledger Fabric yang didapatkan adalah *latency* sebesar 1,028 detik dan *throughput* sebesar 18,2 TPS untuk setiap 2 transaksi paralel, dan *latency* sebesar 1,169 detik dan *throughput* sebesar 20,454 TPS untuk setiap 4 transaksi paralel. Evaluasi perfoma ini dilakukan pada *BatchTimeout* sebanyak 2s dan *MaxMessageCount* sebanyak 20.
3. *Latency* dipengaruhi oleh *BatchTimeout* dan *MaxMessageCount* secara signifikan. Semakin besar *BatchTimeout* dan *MaxMessageCount*, maka *latency* yang didapatkan semakin tinggi. Nilai *BatchTimeout* mempengaruhi *latency* karena waktu tunggu blok untuk di-*commit* dalam satu *batch* menjadi lebih lama. Sedangkan, *MaxMessageCount* mempengaruhi *latency* karena jumlah transaksi yang dapat ditampung dalam satu blok menjadi lebih banyak.
4. *Throughput* dipengaruhi dengan *latency* dan jumlah transaksi paralel yang dikirimkan. Semakin tinggi *latency* dapat menurunkan nilai *throughput*, namun pengaruhnya tidak begitu signifikan jika jumlah transaksi paralel

dan *MaxMessageCount* juga tinggi. *MaxMessageCount* yang tinggi memungkinkan banyak transaksi yang di-*commit* ke ledger. Agar dapat mempengaruhi *throughput* secara signifikan, maka transaksi-transaksi yang di-*commit* harus berstatus valid. Semakin banyak jumlah transaksi paralel membuat *MaxMessageCount* semakin cepat terpenuhi, sehingga jumlah transaksi yang di-*commit* ke *ledger* semakin banyak.

5. Jika jumlah transaksi paralel banyak dan tidak cenderung mengubah objek yang sama, maka nilai *BatchTimeout* dan *MaxMessageCount* dapat ditingkatkan agar dapat mempertahankan nilai *throughput* yang tinggi. Jika transaksi paralel banyak dan cenderung mengubah objek yang sama, maka nilai *BatchTimeout* dan *MaxMessageCount* dapat dikurangi agar transaksi lebih sering di-*commit* ke *ledger*, sehingga konflik pada tahap *validation* dapat dihindari.

5.2. Saran

Adapun saran yang dapat diberikan pada penelitian ini adalah sebagai berikut.

1. Autentikasi produk dapat ditambahkan dengan *NFC tag* agar Oricon dapat memindai kode produk yang ingin ditanamkan pada produk.
2. Standar *token* ERC-20 yang digunakan bersifat *account-based*, yang dapat menimbulkan banyak konflik pada saat tahap *validation* ketika jumlah transaksi paralel meningkat. Oleh karena itu, dapat dikembangkan model *token* lainnya yang cocok untuk transaksi paralel, seperti *unspent transaction input output* (UTXO).

3. Evaluasi performa berdasarkan *latency* dan *throughput* dapat dilakukan pada parameter terkait lainnya seperti jumlah *endorsing peers*, jumlah *ordering service nodes*, algoritma konsensus, alokasi *resources* pada *peers* (jumlah CPU dan RAM), dan jenis *database* yang digunakan sebagai *world state*, untuk melihat pengaruh parameter lainnya yang diberikan pada Hyperledger Fabric.
4. Jumlah *workers* yang digunakan pada Hyperledger Caliper dapat ditingkatkan menjadi puluhan atau ratusan, agar dapat melihat skalabilitas Hyperledger Fabric ketika memproses transaksi paralel dalam jumlah yang banyak. Pada penelitian ini, terdapat keterbatasan spesifikasi *hardware* yang digunakan sehingga jumlah *workers* yang digunakan pada Hyperledger Caliper masih sedikit.
5. Jika jaringan Hyperledger Fabric ingin diimplementasikan sebagai *public network*, maka dapat diterapkan algoritma konsensus lainnya yang bersifat *byzantine-fault tolerant* agar mampu melawan *malicious nodes* yang berpartisipasi ke jaringan.