

Analisa Perbandingan Arsitektur Data Center Switch Centric dan Server Centric Pada Sistem E-Voting

(Comparison Analysis of Data Center Architectures: Switch-Centric and Server-Centric in E-Voting Systems)

Raphael Bianco Huwae, Andy Hidayat Jatmika*, Noor Alamsyah
Dept. Informatics Engineering, University of Mataram
Jl. Majapahit 62, Mataram, Lombok NTB, INDONESIA
Email: [raphael.bianco.huwae, andy, nooralamsyah]@unram.ac.id

Abstract

The development of e-voting was first introduced in the United States in 1974. Various countries around the world have also implemented e-voting using Electronic Voting Machines (EVMs), the internet, mobile phones, fax, and other technologies to facilitate the electoral process. Meanwhile, in Indonesia, the voting system is currently conducted solely through legislative and voting systems such as paper ballots or checkmarks. Therefore, in building an e-voting system, a Data Center is needed to store, process, disseminate data, accelerate the electoral process, and prevent the fraud that has been observed in traditional elections. This often leads to the idea of an e-voting system that is expected to effectively and efficiently accommodate all electoral principles while ensuring that democratic principles are upheld. Based on the simulation results, the use of the Fat Tree architecture appears to be more suitable for environments with numerous nodes, as this architecture operates more stably. On the other hand, the Bcube architecture excels in terms of speed and has lower packet delay, making it suitable for scenarios with fewer nodes compared to Fat Tree.

Keywords: Data Center, E-voting, Architecture, Networking

*Penulis Korespondensi

1. PENDAHULUAN

Perkembangan dari berbagai macam aplikasi yang diterapkan dengan internet saat ini telah mempengaruhi perangkat data center sebagai fasilitas yang digunakan untuk mengatur, menyimpan, memproses, dan menyebarkan data. Jaringan pusat data memerlukan suatu desain struktur jaringan dan protokol yang terkait untuk menghubungkan ribuan atau bahkan ratusan ribu server, perangkat penyimpanan dan peralatan jaringan kedalam sebuah data center yang terpusat. Dalam pembangunan data center biasanya memperhitungkan biaya peralatan yang rendah, kapasitas jaringan yang tinggi dan seimbang, penambahan perangkat yang mudah, kinerja komunikasi yang mampu diukur dan ketahanan terhadap toleransi kesalahan.

Melihat dari jaringan arsitektur yang umum digunakan saat ini, terdapat beberapa jenis arsitektur data center yang cocok digunakan dalam e-voting. Berdasarkan jaringan nya, arsitektur dapat dibedakan dalam 2 kategori yaitu server centric dan switch centric. Server centric merupakan data center yang dimana server menjadi penyimpanan dan pengiriman packet data. Switch centric terdiri dari beberapa bagan dimana fungsi penyimpanan dan pengiriman data tergantung pada arsitektur nya itu sendiri.

Untuk penerapan dalam e-voting, Arsitektur Data Center yang dibangun harus mengandung unsur availability, flexibility, dan scalability. Arsitektur data center harus mampu beradaptasi dengan pertumbuhan kebutuhan yang cepat atau ketika adanya kebutuhan perangkat baru yang harus disediakan oleh data center dampak yang diberikan tidak berpengaruh secara keseluruhan. Unsur

tersebut harus dipenuhi agar data center yang diciptakan mampu memberikan operasi yang berkelanjutan untuk melancarkan system e-voting

Arsitektur yang dikembangkan diharapkan mampu untuk menampung dari semua data yang diterima dari seluruh TPS. Dalam tahapan perhitungan pemilu manual setelah semua pemilih di TPS menggunakan hak politiknya, Ketua Kelompok Penyelenggara Pemungutan Suara (KPPS) akan mengumumkan bahwa tahap pencoblosan telah selesai dan dilanjutkan dengan rapat penghitungan suara. Hal itu tertuang dalam Pasal 46 ayat (2) Peraturan KPU (PKPU) Nomor 9 Tahun 2019 tentang Pemungutan dan Penghitungan Suara dalam Pemilihan Umum [1]. Berikutnya, kotak suara dan dokumen administrasi lainnya diberikan kepada Panitia Pemilihan Kecamatan (PPK) untuk dilanjutkan pada tahap rekapitulasi di tingkat kecamatan. Berlanjut rekapitulasi dilakukan di tingkat kabupaten/kota, kemudian rekapitulasi di tingkat provinsi oleh KPU provinsi. Terakhir, rekapitulasi dilakukan di tingkat nasional oleh KPU RI.

Seperti halnya pada arsitektur konvensional, arsitektur untuk e-voting ini dibuat secara ber-level. Pada level paling bawah atau TPS penyimpanan media suara dilakukan menggunakan mesin Electronic Voting Machine (EVM). Hasil dari EVM nanti akan diteruskan pihak KPPS setelah diumumkan hasilnya kepada server PPK untuk dilakukan klarifikasi kembali hasil rekapitulasinya. Siklus yang sama juga terulang pada server KPU provinsi dan server KPU nasional. Klarifikasi berulang ini adalah untuk menjaga aspek Trust dalam pemilihan umum. Oleh karena itu, Indonesia dinyatakan siap untuk menggunakan sistem e-voting [2].

Maka dari itu arsitektur yang sesuai untuk E-Voting perlu dilakukan uji coba sebelum ditetapkan dalam lingkungan. Simulasi pada paper ini menggunakan NS-3, dimana aplikasi tersebut sudah digunakan berbagai macam komunitas pendidikan.

Dengan adanya simulasi ini bertujuan untuk menghemat biaya dan mampu mengeluarkan hasil yang mendekati dengan kenyataan.

1.1 ARSITEKTUR DATA CENTER

Arsitektur data center yang biasa digunakan saat ini ada 3 jenis berdasarkan interkoneksi nya, yaitu, server centric, switch centric dan perkembangan yang terbaru adalah jaringan hybrid dimana menggabungkan jaringan server dan switch [3].

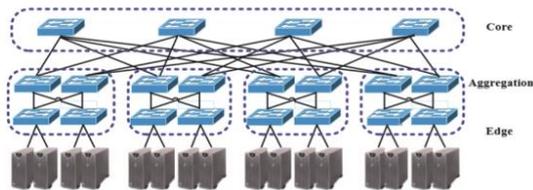
a. Jaringan switch centric

Jaringan berbasis switch, yang disebut "jaringan tidak langsung", biasanya terdiri dari multi-level switch untuk menghubungkan ke server akhir (biasanya dua atau tiga level). Jaringan berbasis switch diadopsi dan diimplementasikan secara luas berbagai data center saat ini, karena mendukung komunikasi antara puluhan ribu server.

Fat-Tree adalah sejenis arsitektur berbasis level. Jenis arsitektur ini salah satu model arsitektur yang sudah lama ada, dan merupakan kelas yang mendasar perutean universal. Arsitektur Fat-Tree terdiri dari empat bagian, Core, edge, server, dan host. Berdasarkan lapisan arsitektur ini digunakan untuk membuat bandwidth tinggi pada perangkat akhir menggunakan komoditas switch yang lebih kecil untuk saling terhubung.

Seperti yang ditunjukkan pada Gambar. 2.1, switch di Fat-Tree dibagi menjadi tiga level yaitu edge level, agregation level, dan core level. Arsitektur Fat-Tree yang dibangun dengan n-port switch memiliki n pod. Setiap pod berisi dua level $n/2$ switch. Pada edge level, masing-masing switch menggunakan $n/2$ port untuk menghubungkan $n/2$ server, dan sisanya untuk menghubungkan $n/2$ port untuk menghubungkan $n/2$ server ke agregation level. Ada $n^2/4$ switch pada level inti, dan setiap switch menggunakan satu port untuk menghubungkan satu pod. Oleh karena itu, arsitektur Fat-Tree terdiri dari $5n^2/4$ switch dan mensupports $n^3/4$ server secara total. Sehingga

skalabilitas Fat-Tree dibatasi oleh port switch, dimana jika memerlukan penambahan server maka port yang ditambahkan juga perlu menambahkan switch [4].

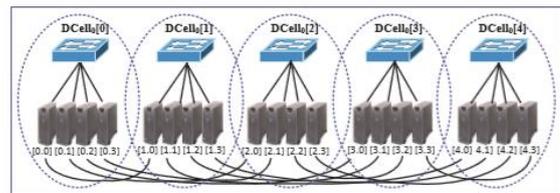


Gambar 2.1 Ilustrasi Arsitektur Fat-tree

b. Jaringan server centric

Pilihan lain untuk membuat koneksi antara server adalah dengan jaringan langsung. Jaringan langsung menghubungkan server secara langsung ke server lain dalam jaringan tanpa sakelar, router, atau perangkat jaringan apa pun. Server akan berfungsi sebagai penyimpanan dan penerus jaringan. Jaringan langsung sering digunakan untuk memberikan skalabilitas yang lebih baik, toleransi kesalahan, dan kapasitas jaringan yang tinggi. Beberapa implementasi praktis dari jaringan langsung akan disajikan

DCell merupakan salah satu jaringan langsung pertama dalam arsitektur data center. Di DCell, server terhubung ke beberapa server lain melalui switch mini dengan komunikasi dua arah. Tingkatan pada DCell dinyatakan dalam k di mana $k \geq 0$. Pada gambar 2.2, Sejumlah n server dan switch mini membentuk DCell0 di mana semua server terhubung ke switch mini. Karena itu DCell1 terbuat dari $n + 1$ DCell0. Semua pasangan n DCell0 ini akan dihubungkan ke satu link. Oleh karena itu di DCell1, masing-masing server memiliki dua link, satu untuk terhubung ke miniswitch dan yang lainnya terhubung ke server di DCell0. Sama halnya pada DCell2 dengan $n + 1$ DCell1 dan juga seterusnya dengan DCellk yaitu $n + 1$ DCellk - 1 [5].



Gambar 2.2 Ilustrasi Arsitektur D-Cell

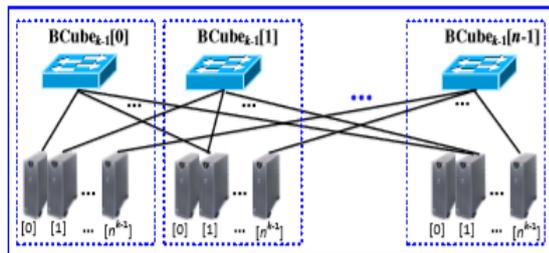
Kapasitas jaringan yang tinggi dan toleransi kesalahan sedikit menjadi tujuan dari DCell. Akan tetapi untuk skalabilitas penambahan sangat buruk. Terutama jika arsitektur DCell terpenuhi, sangat sulit untuk menambahkan sedikit server ke dalam arsitektur tanpa merusak struktur arsitektur asli. Selain itu, beban lalu lintas yang tidak seimbang juga membuat DCell berkinerja buruk.

Arsitektur BCube dirancang khusus untuk pusat data modular berbasis container-pengiriman. Inti dari arsitektur BCube adalah struktur jaringan server-centric, di mana server dengan beberapa port jaringan terhubung ke beberapa lapisan switch mini COTS (komoditas di luar rak). Server bertindak sebagai tidak hanya mengakhiri host, tetapi juga menyampaikan node untuk satu sama lain. BCube mendukung berbagai hal aplikasi intensif bandwidth dengan mempercepat lalu lintas satu-ke-satu, satu-ke-beberapa, dan satu-ke-semua pola, dan dengan menyediakan kapasitas jaringan tinggi untuk lalu lintas semua-untuk-semua.

Arsitektur Bcube memiliki waktu pengerjaan yang sangat cepat dari pemesanan hingga pemasangan karena sifatnya modular. Dengan rancangan desain yang dapat dirubah, mudah disesuaikan dengan permintaan dan pembangunan infrastruktur dengan cepat. Sehingga dapat di alokasikan di mana pun diinginkan oleh pengguna.

Sama seperti dengan DCell, BCube dibangun secara rekursif dari n switch BCube0 dan n -port. BCube $k \geq 1$ dibangun dari n BCube $k - 1$ di mana setiap server memiliki $k + 1$. Sangat mudah untuk melihat bahwa BCube k terdiri dari $n k + 1$ server dan $k + 1$ tingkat switch. Gambar 2.3 merupakan prosedur dasar arsitektur BCube k .

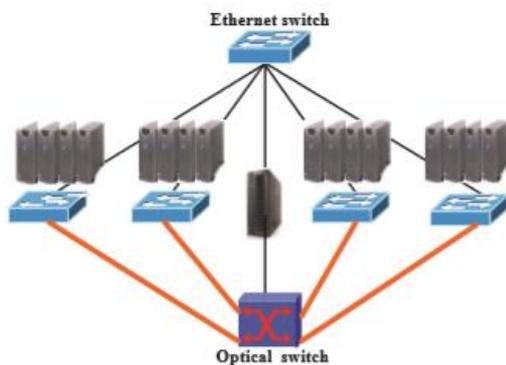
perbandingannya dengan DCell, pada BCube switch hanya terhubung ke server daripada switch lainnya.



Gambar 2.3 Ilustrasi Arsitektur B-Cube

c. Jaringan Hybrid

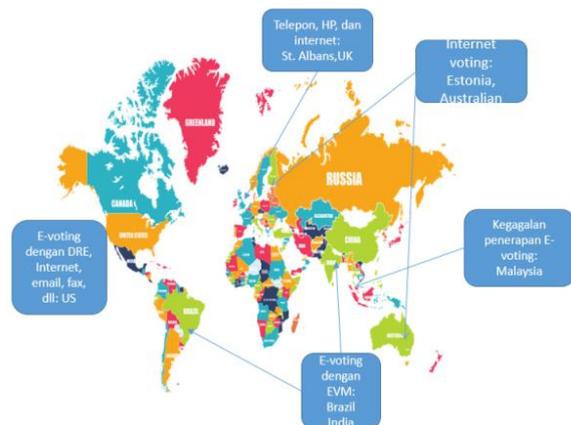
Arsitektur Data Center terbaru saat ini adalah jaringan Hybrid dimana arsitektur tersebut menggunakan optical switch. Dibandingkan dengan packet switch, optical switch lebih unggul dari bandwidth yang besar, transmisi yang rendah dan konsumsi daya yang rendah. Sesuai dengan namanya, hybrid adalah sebuah teknologi yang menggabungkan packet switching dengan optical switch seperti yang dapat terlihat pada gambar 2.4. Sehingga tanpa perlu merubah packet switching yang ada dan hanya perlu menambahkan optical switch membuat arsitektur hybrid mudah untuk diimplementasi [6].



Gambar 2.4 Ilustrasi Jaringan Hybrid

1.2 E-Voting

e-voting merupakan proses pemungutan suara yang memungkinkan pemilih untuk memberikan suara melalui Internet atau Intranet. Perhitungan intranet biasa dilakukan dengan suatu mesin yang disebut Electronic Voting Machine (EVM) pada tempat pemilihan, yang kemudian hasilnya akan dikirimkan secara terpusat. Sistem e-voting dibuat dalam rangka penghitungan suara dan tabulasi data dapat menghitung hasil lebih cepat dan lebih akurat daripada sistem penghitungan konvensional yang manual. Sistem e-voting juga mampu untuk mengurangi biaya yang harus dikeluarkan pada saat pemilu. Sebab, pada pemilu konvensional, Komisi Pemilihan Umum (KPU) harus mencetak surat suara dalam jumlah banyak, kotak suara serta kartu tanda pemilih. Namun, dengan e-voting KPU hanya perlu menyediakan mesin elektronik dengan menggunakan e-KTP pemilih, yang mana mesin ini dapat dipergunakan berulang-ulang.



Gambar 2.3 Perkembangan E-voting di dunia

Dengan perkembangan teknologi informasi, negara-negara di seluruh dunia mengganti sistem pemungutan suara mekanis dengan sistem pemungutan suara elektronik (e-voting) yang bertujuan untuk meningkatkan partisipasi pemilih dan mempercepat pelepasan hasil pemilu. Brasil dan India adalah contoh dari negara-negara yang menggunakan e-voting untuk pemilihan umum dan pemilihan umum. Komisi Pemilihan India menyatakan bahwa sistem e-voting memerangi masalah kecurangan pemilu India, seperti

menambah polling di tempat pemungutan suara atau mencuri kotak suara.

St. Albans, Inggris, pada Mei 2007, melaksanakan pemilihan elektronik sepenuhnya tanpa pemungutan suara berbasis kertas diizinkan. Orang-orang menggunakan sejumlah saluran untuk memilih, Internet, kios, pengenalan suara interaktif, melalui telepon atau ponsel, dan juga melalui pos. Dalam waktu enam menit, sistem telah menghitung semua surat suara dan menjadi penghitungan suara tercepat yang pernah ada.

Walaupun sistem e-voting menjadi alternatif terbaik untuk sistem berbasis kertas dan lainnya, penerapannya juga harus dilakukan dengan hati-hati karena sistem ini memiliki tantangan mulai dari rekayasa perangkat lunak, audit perangkat, ancaman orang dalam, sehingga dapat merusak hasil pemilihan. Sebagai contoh, Pada tahun 2018 Malaysia mencoba menerapkan e-voting, akan tetapi gagal karena jaringan yang buruk dan alasan teknis lainnya. Oleh karena itu perlu diperhatikan serangan menggunakan worm, virus dan Denial-of-Service yang dapat menumbangkan sistem e-voting [7].

2. TINJAUAN PUSTAKA

Pada penelitian “Skema Penyeimbangan Beban untuk Sistem Pemungutan Suara Elektronik Dinamis dengan Menggunakan Agen Bergerak” [8] menyatakan salah satu tantangan pada sistem pemungutan suara elektronik dinamis harus mampu mengatasi lalu lintas, atau kinerja sistem akan menurun. Sistem pemungutan suara elektronik dinamis perlu memiliki waktu respons yang cepat. Dalam lingkungan multi-server, jika server terlalu terbebani, akan sulit untuk mencapai waktu respons yang cepat. Penelitian tersebut mengusulkan skema penyeimbangan beban untuk sistem pemungutan suara elektronik dinamis menggunakan agen bergerak. Pendekatan tersebut menjanjikan peningkatan efisiensi dan efektivitas dalam hal ketersediaan sistem untuk pemilihan berskala besar. Pada penelitian tersebut disimpulkan bahwa

penyeimbangan beban untuk sistem pemungutan suara elektronik dinamis dengan Teknologi Agen meningkatkan kinerja sistem pemungutan suara elektronik dinamis. Teknik ini meningkatkan waktu respons sistem dan mengurangi waktu tunggu pemilihan. Teknik penyeimbangan beban menggunakan agen bergerak menjamin ketersediaan sistem untuk pemilihan berskala besar.

Pada penelitian tentang “Desain dan Implementasi Aplikasi E-Voting Berbasis Blockchain dalam Kota Cerdas Berorientasi IoT” [9] menyatakan terdapat masalah besar di mana penyusup dapat melakukan sejumlah tindakan curang untuk mengatur hasil pemilihan. Oleh karena itu, tantangan potensialnya adalah membedakan perangkat IoT yang sah dari yang jahat dengan menghitung nilai kepercayaan mereka melalui optimisasi sosial untuk menjaga lingkungan komunikasi yang sah. Selanjutnya, untuk mencegah modifikasi data yang mungkin terjadi pada perangkat pintar di masa depan, sebuah Blockchain server digunakan di mana blok dari semua perangkat IoT yang sah dicatat. Artikel tersebut memperkenalkan mekanisme e-voting yang aman dan transparan melalui perangkat IoT menggunakan teknologi Blockchain dengan tujuan mendeteksi dan mengatasi berbagai ancaman yang disebabkan oleh penyusup pada berbagai tingkatan. Penelitian ini menghasilkan kerangka kerja yang diusulkan secara signifikan mengungguli mekanisme dasar dengan melacak aktivitas setiap tingkat proses pemilihan. Selanjutnya, kerangka kerja yang diusulkan menunjukkan tingkat keberhasilan yang lebih baik dalam semua hasil simulasi dibandingkan dengan mekanisme dasar dalam mengatasi perubahan pesan, ancaman DoS, DDoS, dan mekanisme otentikasi.

Pada penelitian mengenai “Model Konseptual E-Voting di Indonesia” [2] bertujuan untuk menemukan kesiapan implementasi sistem e-voting di Indonesia dengan melakukan survei kuantitatif dan mengusulkan arsitektur sistem pemungutan

suara elektronik terkini yang cocok untuk diterapkan di Indonesia. Penelitian ini menyatakan bahwa secara umum masyarakat Indonesia bersedia menggunakan sistem e-voting. Meskipun masalah terkait keamanan dan kerahasiaan data juga merupakan topik utama yang sering dipertanyakan dalam implementasi sistem e-voting, Kesiapan masyarakat yang menggunakan sistem e-voting adalah salah satu faktor penentu keberhasilan implementasi sistem tersebut. Selain itu, faktor kesiapan untuk infrastruktur pendukung dan penyelenggara pemilihan adalah faktor lain yang dapat diteliti lebih lanjut.

Pada penelitian “Tinjauan Sistematis tentang Tantangan dan Peluang Blockchain untuk E-Voting” [10], Artikel ini menguraikan sistem e-voting berbasis blockchain dengan tujuan utama untuk menyelidiki perkembangan terkini dalam penelitian e-voting berbasis blockchain, serta mengidentifikasi potensi tantangan yang terkait dan meramalkan arah masa depan. Metode tinjauan sistematis diterapkan dalam penelitian ini untuk dapat mengadopsi kerangka kerja blockchain yang digunakan untuk sistem pemungutan suara dapat menghasilkan solusi yang berbeda dibandingkan e-voting tradisional. Masalah utama diklasifikasikan ke dalam lima kategori: umum, integritas, berbasis koin, privasi, dan konsensus. Penelitian menyimpulkan bahwa sistem blockchain dapat memberikan solusi untuk beberapa masalah yang umumnya muncul dalam sistem pemilihan saat ini. Namun, perhatian utama seringkali tertuju pada perlindungan privasi dan kecepatan transaksi dalam aplikasi blockchain. Keamanan partisipasi jarak jauh dan skalabilitas perlu diperkuat untuk menjadikan e-voting berbasis blockchain berkelanjutan. Kesimpulan dari penelitian tersebut, diperlukan peningkatan pada kerangka kerja agar dapat diimplementasikan dengan efektif dalam sistem pemilihan, mengatasi berbagai kekhawatiran yang muncul.

Berdasarkan kajian pustaka yang telah dilakukan, peneliti melakukan analisis perbandingan

arsitektur Data Center D-Cell dan B-Cube dalam pengaplikasian sistem E-Voting. Studi-studi sebelumnya menunjukkan bahwa sistem E-Voting memiliki potensi untuk meningkatkan efisiensi dalam proses pemilihan, mengurangi kesalahan, dan mempercepat penghitungan suara. Namun, keberhasilan implementasi E-Voting juga sangat tergantung pada infrastruktur jaringan dan komputasi yang mendukungnya. Oleh karena itu, penelitian ini bertujuan untuk membandingkan dua arsitektur Data Center, yaitu D-Cell dan B-Cube, untuk memahami bagaimana kinerja dan skalabilitas keduanya dapat memengaruhi pengaplikasian E-Voting. Hasil dari analisis perbandingan ini diharapkan dapat memberikan wawasan berharga dalam pemilihan arsitektur yang paling cocok untuk mendukung keberhasilan sistem E-Voting di masa depan.

3. METODOLOGI PENELITIAN

Metodologi penelitian yang digunakan dalam penulisan paper ini adalah dengan melakukan studi literatur berupa paper, laporan maupun tulisan-tulisan terkait arsitektur data center dan implementasinya pada e-voting. Setelah mengetahui jenis arsitektur yang ada, perbandingan terhadap berbagai jenis arsitektur akan diketahui sehingga dapat ditentukan arsitektur yang cocok untuk diimplementasikan pada e-voting.

Dari informasi tersebut lalu dilakukan suatu pengujian terhadap arsitektur yang dipilih dengan menggunakan simulasi (dengan tools NS-3) dan dibuat suatu kajian yang sistematis sehingga implementasinya dapat lebih dikembangkan dimasa mendatang

4. PEMBAHASAN

Sebelum melakukan simulasi terhadap arsitektur yang sudah ada, dilakukan evaluasi terlebih dahulu berdasarkan dari literature yang sudah ada. Dengan mengetahui kemampuan dari masing-masing jenis arsitektur akan mempersempit cakupan arsitektur yang akan disimulasikan. Kemampuan yang akan dibandingkan seperti

skalabilitas, penarikan jalur kabel, toleransi kesalahan, dan keseimbangan traffic untuk memenuhi aspek availability, flexibility dan scalability dalam perancangan data center

Hasil arsitektur yang terpilih dari pengelompokan tersebut kemudian akan dilakukan simulasi menggunakan tools NS-3. Simulasi tersebut akan dijalankan menggunakan parameter yang sama sehingga perbandingan antara arsitektur tersebut dapat dilihat secara objektif. Besaran throughput dan delay menjadi fokus dari analisa ini karena pada e-voting membutuhkan kecepatan yang stabil.

4.1 Perbedaan Jenis Arsitektur Data Center

Seperti yang sudah dijelaskan pada bagian sebelumnya, arsitektur data center terbagi menjadi server centric, switch centric dan jaringan hybrid. Pada server centric akan diwakili dengan tree dan fat-tree, sedangkan pada switch centric dengan D-Cell dan B-Cube. Masing-masing arsitektur memiliki kelebihan dan kekurangannya, akan tetapi yang perlu diperhatikan adalah pada paper ini tujuannya adalah untuk mencari arsitektur yang dapat digunakan untuk e-voting. Berdasarkan penjelasan yang dibahas sebelumnya pada bagian 2, di klasifikasikan seperti tabel dibawah ini.

Tabel 4.1 Perbandingan Arsitektur Data Center

	Tree	Fat tree	DCell	BCube
Scalability	Poor (scale up)	Good (scale out)	Excellent (scale out)	Good (scale out)
Incremental scalability	Good	Good	Poor	Not necessary
Wiring	Easy	Easy	Very difficult	Difficult
Multipath routing	No	Switch and router protocol upgrade	End-host protocol upgrade	
Fault tolerance	Poor	Against switch and router failures	Against switch, router and end-host port failures	
Cost	High-end switches and routers	Low-end customized switches and routers (cheap but many)		
Traffic balance	No	Yes	No	Yes
Graceful degradation	Poor	Good	Excellent	Excellent

Dalam segi skalabilitas semua arsitektur selain Tree tidak memiliki masalah dalam jumlah server

yang besar. Akan tetapi, dalam segi penambahan kapasitas D-Cell memiliki keseimbangan yang buruk, karena setiap penambahannya membutuhkan switch baru yang terhubung dengan jumlah server yang sama dengan sebelumnya.

Untuk jalur pengkabelan server centric dan hybrid memiliki kemudahan dalam instalasinya jika dibandingkan dengan switch centric. Akan tetapi masing-masing memiliki masalah pada keseimbangan traffic nya sehingga hanya Fat-tree dan B-Cube yang dinilai layak jika memperhatikan dari traffic nya. Hal tersebut tentu akan berpengaruh terhadap packet loss dan toleransi terhadap kesalahan paket.

Melihat dari perbandingan yang dilakukan diatas maka pada tahapan simulasi yang selanjutnya akan dilakukan penulis memilih arsitektur Fat-Tree dan B-Cube untuk di uji coba dalam tools NS-3. Karena jika diperhatikan pada masing-masing arsitektur memiliki traffic balance yang seimbang sehingga meminimalisir kesalahan terhadap hasil pemilihan nantinya. Selain itu kemampuan untuk menambahkan server tidak terlihat bermasalah pada kedua arsitektur tersebut. Sedangkan untuk arsitektur hybrid walaupun telah memenuhi aspek yang berlaku akan tetapi memiliki kekurangan karena kebutuhan instalasi yang cukup besar dan adaptasi dengan arsitektur baru ini belum sepenuhnya bias dikuasai oleh pengguna,

4.2 Simulasi Arsitektur Fat-Tree dan B-Cube

Simulasi dilakukan dengan spesifikasi computer intel core i5 5200U 2.20Ghz dan 8 Gb RAM dijalankan pada linux Ubuntu 18.04. Kedua arsitektur ini di simulasikan dengan NS-3 untuk mengukur performansi. Tools tersebut telah digunakan pada berbagai komunitas untuk melakukan simulasi sehingga hasilnya tidak terlalu berbeda dengan realisasi nya.

Untuk perbandingan yang objektif, ukuran kedua arsitektur, konfigurasi perangkat jaringan dan protokol, dan generasi arus lalu lintas harus

konsisten. Satu-satunya variabel yang harus diubah adalah topologi jaringan. Tabel 4.2 menunjukkan pengaturan simulasi yang digunakan untuk model kinerja Fat tree dan arsitektur BCube

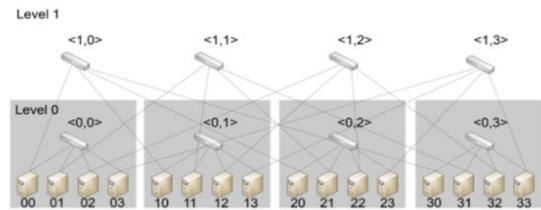
Tabel 4.2 Parameter Simulasi NS-3

	Fat tree	BCube
Number of pods (k)	4-24	-
Number of BCube levels	-	3 ($k = 2$)
Number of nodes in BCube ₀ (n)	-	4-15
Number of nodes	16-3,456	64-3,375
Simulation running time	100s	100s
Packet size	1024 bytes	1024 bytes
Data rate for packet sending	1 Mbps	1 Mbps
Data rate for device channel	1000 Mbps	1000 Mbps
Communication pairs selection	Random selection with uniform probability	Random selection with uniform probability
Traffic flow pattern	Exponential random traffic	Exponential random traffic
Routing protocol	Nix-Vector	Nix-Vector

Ukuran untuk kedua arsitektur berkisar dari 16 hingga 3.456 node. Switch ethernet 1.000 Mbps dengan alamat IP realistis dan protokol routing Nix-Vector. Untuk menghasilkan arus lalu lintas di dalamnya, pasangan komunikasi di semua node dipilih secara acak, dengan masing-masing pasangan terdiri dari pengirim dan penerima. Setiap pasangan secara bersamaan akan mengirim aliran data 1 Mbps dari pengirimnya ke penerima dan simulasi untuk kedua arsitektur dijalankan selama 100 detik. Pola aliran lalu lintas untuk kedua arsitektur mengikuti perilaku onoff dengan distribusi acak eksponensial, yang telah terbukti secara wajar memodelkan arus lalu lintas di pusat data dunia nyata Dengan pengaturan simulasi yang konsisten seperti table 1, hasil dilanjutkan dengan jumlah kapasitas yang berbeda dan data tersebut dikumpulkan. Statistik kinerja kemudian dapat dilihat dan dianalisis menggunakan modul Flow Monitor di ns-3.

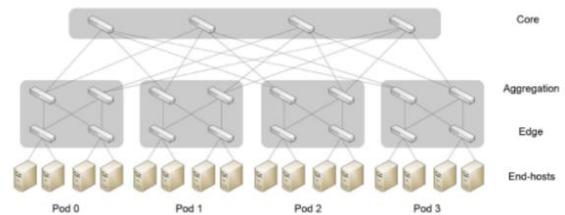
Arsitektur Fat-Tree dan B-Cube yang dijalankan juga mengikut dengan dasar arsitektur mereka. Bcube jumlah lebel dinyatakan dalam k di mana $k \geq 0$. Pada gambar 4.1 Sejumlah n server dan switch mini membentuk Bcube0 di mana semua server terhubung ke switch mini. dan BCube1 terbuat dari $n + 1$ BCube0. Semua pasangan n BCube0 ini akan

dihubungkan ke satu link dan BCube switch hanya terhubung ke server.



Gambar 4.1 Arsitektur Simulasi B-Cube

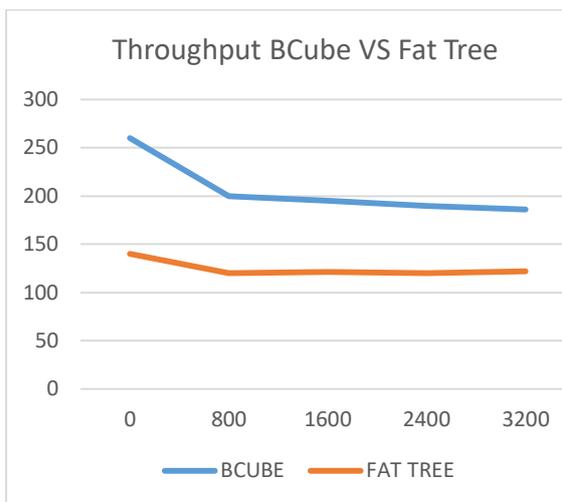
Sedangkan arsitektur Fat-Tree jga mengikuti arsitektur umum biasanya. Dimana pada gambar 4.2 Fat-Tree dibagi menjadi tiga level yaitu edge level, agregation level, dan core level. Jumlah pods yang terpasang akan mempengaruhi jumlah host yang akan terinstall [11]



Gambar 4.2 Arsitektur Simulasi Fat-Tree

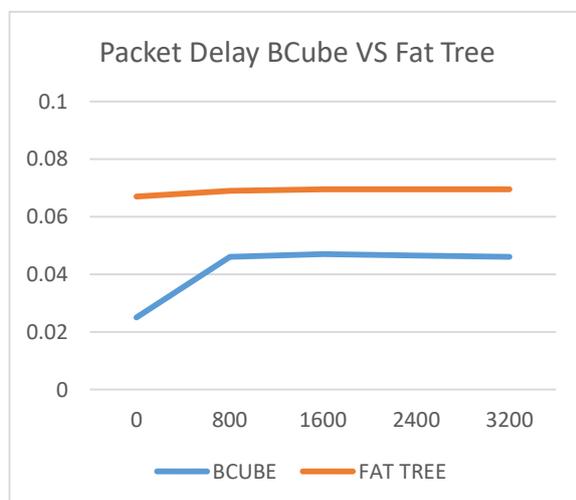
4.3 Analisa Hasil Simulasi

Berdasarkan hasil simulasi seperti gambar 4.3 yang menggambarkan hasil throughput antara BCube dan Fat-Tree dapat diperhatikan jika performa BCube lebih baik jika dibandingkan dengan fat tree. Dengan jumlah nodes dinaikkan dari 16 ke 3,456. Terlihat perbedaan hasil throughput pada BCube yang turun dari 237 Mbps ke 174 Mbps. Sedangkan arsitektur Fat tree bergerak secara stabil dari 126 Mbps ke 117 Mbps.



Gambar 4.3 Hasil Simulasi Throughput

Untuk hasil packet delay yang terlihat pada gambar 4.4 terlihat arsitektur Bcube lebih sedikit delay dibandingkan dengan Fat-Tree. BCube memiliki 0.036 ms hingga 0.048 ms dan arsitektur Fat tree memiliki kisaran dari 0.066 ms hingga 0.072 ms.



BCube bekerja lebih baik dari perhitungan tersebut adalah karena arsitektur tersebut memiliki berbagai macam jalur jaringan untuk mengirimkan paket dari poin A ke poin B, sehingga trafik menjadi lebih sedikit dan bandwidth terlihat lebih lebar. Terlebih lagi salah satu server pada BCube digunakan sebagai pengatur trafik dan penerus ke

server lainnya sehingga menjadikan lebih cepat dibandingkan Fat Tree. Sedangkan Fat Tree sendiri memiliki performa yang konsisten adalah karena topologinya yang fundamental yang telah digunakan diberbagai data center sehingga interkoneksinya menjadi stabil.

5. KESIMPULAN

Arsitektur dalam simulasi yang dilakukan untuk membandingkan Switch Centric dan Server Centric dan kemudian berfokus pada arsitektur Fat tree dan BCube yang bertujuan agar dapat diterapkan dalam E-Voting. Arsitektur Data Center yang digunakan pada E-Voting harus memenuhi asas availability, flexibility dan Scalability. Dilihat dari hasil simulasi penggunaan arsitektur Fat tree akan lebih cocok jika digunakan pada arsitektur yang memiliki banyak node, karena arsitektur ini berjalan lebih stabil. Di sisi lain, arsitektur Bcube menang dalam hal kecepatan dan delay paket lebih sedikit, sehingga arsitektur ini cocok untuk node yang tidak terlalu banyak jika dibandingkan dengan Fat tree. Oleh karena itu jika melihat jumlah kebutuhan kapasitasnya, maka arsitektur Fat-Tree memiliki kapasitas yang diperlukan dalam mewujudkan E-voting. Sehingga pada paper ini penulis menyarankan untuk menggunakan arsitektur Fat-Tree dalam perancangan data center E-Voting

UCAPAN TERIMA KASIH

Terima kasih kepada semua pihak yang telah berkontribusi dalam proses penerbitan artikel ini, termasuk penulis pendamping dan semua orang yang terlibat atas dukungan, bimbingan dan bantuan yang diberikan sebagai masukan pada penelitian ini dan juga kepada Jurnal Teknologi Informasi Komputer dan Aplikasinya yang telah memfasilitasi publikasi artikel penelitian kami.

DAFTAR PUSTAKA

- [1] K. P. Umum, "Peraturan Komisi Pemilihan Umum Nomor 9 Tahun 2019," Komisi Pemilihan Umum, Jakarta, 2019.
- [2] D. I. & P. P. B. Sensuse, "Conceptual model of e-voting in Indonesia," *International Conference on Information Management and Technology (ICIMTech). IEEE*, pp. 387-392, 2020.
- [3] Z. e. a. Zhang, "HSDC: A Highly Scalable Data Center Network Architecture for Greater Incremental Scalability," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30(5), pp. 1105-1119., 2018.
- [4] A. Przygienda, A. Sharma, P. Thubert, B. Rijsman and D. Afanasiev, "RIFT: Routing in Fat Trees," RIFT Working Group, 2021.
- [5] X. & M. J. Liu, "The k-restricted edge-connectivity of the data center network DCell," *Applied Mathematics and Computation*, vol. 396 , no. 125941, 2021.
- [6] Rahman, M. Naimur and A. Esmailpour, "A hybrid data center architecture for big data," *Big Data Research*, vol. 3, pp. 29-40, 2016.
- [7] Wikipedia, "Elecronic Voting," [Online]. Available: https://en.wikipedia.org/wiki/Electronic_voting. [Accessed 11 September 2023].
- [8] F. T. ibharalu, "A LOAD BALANCING SCHEME FOR DYNAMIC E-VOTING SYSTEM USING," *The Nigerian Association of Mathematical Physics*, pp. 1-13, 2020.
- [9] Rathee, "On the Design and Implementation of a Blockchain Enabled E-Voting Application Within IoT-Oriented Smart Cities," *IEEE Access*, vol. 9, pp. 34165-34176, 2021.
- [10] T. R and T. ÖÖ, "A Systematic Review of Challenges and Opportunities of Blockchain for E-Voting," *Symmetry*, vol. 12, no. 1328, p. 24, 2020.
- [11] "Towards reproducible performance studies of datacenter network architectures using an open-source simulation approach," *IEEE Global Communications Conference (GLOBECOM)*, pp. 1373-1378, 2013.
- [12] G. O. a. E. P. Ofori-Dwumfuo, "the design of an electronic voting system," *Research Journal of Information Technology*, vol. 3.2, pp. 91-98, 2011.
- [13] K. M. e. a. AboSamra, "A practical, secure, and auditable e-voting system," *Journal of information security and applications*, vol. 36, pp. 69-89, 2017.
- [14] Salman, S. Abdulbaqi, S. Al-Janabi and A. M. Saghee, "Security Attacks on E-Voting System Using Blockchain," *IJCSM*, vol. 4, pp. 179-188, 2023.
- [15] M. A. Cheema, N. Ashraf, A. Aftab, H. K. Qureshi, M. Kazim and A. T. Azar, "Machine learning with blockchain for secure E-voting system," *SMARTTECH*, vol. 1, pp. 177-182, 2020.
- [16] H. Yi, "Securing e-voting based on blockchain in P2P network," *EURASIP*, vol. 1, p. 1-9, 2019.
- [17] Z.-Y. L. a. J.-H. H. R. Tso, "Distributed E-voting and E-bidding systems based on smart contract," *Electronics*, vol. 8, p. 422, 2019.

- [18 S. Latif and T. Anees, "Blockchain based
] Decentralized Electronic Voting System: A
Step towards Transparent," *IJCSNS*, vol. 19,
no. 12, p. 165, 2019.
- [19 A. Hassan and X. Zhang, "Design and build a
] secure e-voting infrastructure," *LISAT*, pp.
1-7, 2013.