

Analisa Performansi dan Keamanan Penyimpanan Data Pada Distributed Hash Table

Ega Patra Nugraha¹, Vera Suryani², Aulia Arif Wardana³

^{1,2,3} Universitas Telkom, Bandung

eganugraha@students.telkomuniversity.ac.id¹, verasuryani@telkomuniversity.ac.id²,
auliawardan@telkomuniversity.ac.id³

Abstrak

Jaringan komputer pada saat ini sangat berkembang. Dengan adanya jaringan komputer semua orang dapat berkomunikasi, berbagi dan bertukar informasi. *Distributed Hash Table* merupakan salah satu metode untuk melakukan penyimpanan data dan berbagi data secara peer-to-peer. Namun terdapat resiko keamanan pada saat melakukan penyimpanan data seperti kehilangan ataupun pencurian data. Salah satu solusi yang dapat diusulkan untuk menjaga keamanan data *file sharing* yaitu dengan menggunakan sistem *Distributed Hash Table* dan mengukur analisa performansi pada *Distributed Hash Table*. Algoritma yang digunakan pada tugas akhir ini yaitu kademlia. Langkah awal pada penerapan ini yaitu melakukan pembuatan prototipe yang bertujuan untuk mempermudah dalam pembuatan sistem. Hasil akhir dari penelitian adalah sistem yang dapat melakukan *file sharing* dan keamanan pada data secara terdistribusi menggunakan *Distributed Hash Table*. File akan diunggah dan diunduh oleh *node* lain lalu disimpan di server yang sudah ditentukan. Kemudian dilakukan analisis performansi berupa *delay* dan *throughput* pada jaringan *Distributed Hash Table*. *Delay* yang didapatkan saat melakukan proses unggah *text* sebesar 2ms, gambar sebesar 1ms dan video 0.2ms. Pada proses unduh didapatkan *delay text* sebesar 0.19ms, gambar sebesar 0.16ms dan video 0.08ms. *Throughput* yang didapatkan saat melakukan proses unggah *text* sebesar 991 bit/detik, gambar sebesar 8005 bit/detik dan video 1135852 bit/detik. Pada proses unduh didapatkan *throughput text* sebesar 2230 bit/detik, gambar sebesar 19831 bit/detik dan video 2719202 bit/detik.

Kata kunci : *Distributed Hash Table(DHT), File Sharing, Kademlia, peer-to-peer.*

Abstract

Computer networks at this time are very developed. With a computer network everyone can communicate, share and exchange information. *Distributed Hash Table* is a method for storing data and sharing data peer-to-peer. However, there are security risks when storing data, such as data loss or theft. One solution that can be proposed to maintain the security of file sharing data is by using the *Distributed Hash Table* system and measuring the performance analysis of the *Distributed Hash Table*. The algorithm used in this final project is kademlia. The first step in this application is to make a prototype which aims to facilitate the manufacture of the system. The final result of this research is a system that can do file sharing and security on distributed data using *Distributed Hash Table*. The file will be uploaded and downloaded by other nodes and then stored on the specified server. Then perform a performance analysis in the form of delay and throughput on the *Distributed Hash Table* network. The delay obtained when uploading text is 2ms, images are 1ms and video is 0.2ms. In the download process, the text delay is 0.19ms, the image is 0.16ms and the video is 0.08ms. Throughput obtained when uploading text is 991 bits/second, images are 8005 bits/second and video is 1135852 bits/second. In the download process, the text throughput is 2230 bits/second, images are 19831 bits/second and video is 2719202 bits/second.

Keywords: *Distributed Hash Table(DHT), File Sharing, Kademlia, peer-to-peer.*

1. Pendahuluan

Latar Belakang

Semakin berkembangnya jaringan komputer, bermunculan beragam aplikasi *file sharing* yang dapat mempermudah pengguna internet dalam berbagi *file*. *Torrent* merupakan salah satu hasil aplikasi *file sharing* yang bekerja di jaringan *peer to peer*. Pengoptimalan kinerja dari *file sharing* ini membutuhkan

pengaturan dan algoritma dalam mekanisme penyimpanan pasangan *key-value* dan pencarian *value* dari *file* yang disimpan[1]. Terdapat kelebihan *file sharing* yang sudah ada seperti dapat menjaga *file* dari virus, dapat menghemat waktu dalam proses penyimpanan, dan dapat bertukar informasi data dengan mudah. Adapun kekurangan pada penerapan penyimpanan data pada *file sharing* yaitu kehilangan maupun pencurian data yang disebabkan oleh pihak tidak bertanggungjawab yang menginginkan data secara ilegal, sehingga banyak orang yang ragu untuk menggunakan *file sharing* sebagai tempat penyimpanan data.

Distributed Hash Table bisa menjadi suatu solusi untuk permasalahan tersebut. *Distributed Hash Table* (DHT) adalah bentuk data *hash* yang dimasukkan ke dalam jaringan *peer-to-peer* (P2P) dan memuat penyimpanan data dengan penyimpanan dari *key* dan *value*[1]. Penerapan *Distributed Hash Table* akan sangat berguna karena bisa sebagai tempat penyimpanan data. Penerapan *Distributed Hash Table* selain untuk penyimpanan data juga dapat untuk meningkatkan keamanan data pengguna sehingga pengguna tidak perlu khawatir data yang disimpan akan dicuri maupun hilang.

Distributed Hash Table merupakan perkembangan dari sebagian sistem *peer-to-peer*, seperti freenet dan gnutella yang mengambil keuntungan dari peningkatan *bandwith* dan kapasitas penyimpanan untuk berbagi data[2]. Berbeda halnya dengan DHT, sistem pada freenet sepenuhnya terdistribusi dengan menggunakan rute yang berbasis *key heuristic* kemudian *file* akan terkait. *Key* dan *file* yang sama akan mengelompokkan ke *node* yang sama sehingga freenet belum mampu menjamin keberhasilan dalam pencarian data[3]. *Distributed Hash Table* terdapat sistem keamanan yang sudah di rancang untuk menghindari serangan *eclipse*, *sybil*, dan *adaptive join-leave*[3].

Pada tugas akhir ini akan dibentuk sebuah sistem penyimpanan data menggunakan *Distributed Hash Table*. Melakukan analisa performansi pada *Distributed Hash Table*.

Perumusan Masalah

Rumusan masalah pada penelitian tugas akhir ini yaitu bagaimana melakukan unggah dan unduh file dengan metode *Distributed Hash Table* dan mengetahui performansi pada penyimpanan file menggunakan *Distributed Hash Table*.

Tujuan

Berdasarkan rumusan masalah terdapat tujuan pada penelitian tugas akhir ini yaitu sistem dapat melakukan unggah dan unduh file menggunakan *Distributed Hash Table* dan melakukan analisa performansi jaringan pada *Distributed Hash Table*.

Organisasi Tulisan

Organisasi penulisan dalam penulisan Tugas Akhir ini yaitu sudah dijelaskan pada bagian pertama mengenai latar belakang dari penelitian, perumusan masalah dari penelitian, tujuan dari penelitian, batasan masalah dari penelitian, dan organisasi penulisan. Pada bagian dua dijelaskan penelitian sebelumnya dan landasan teori yang dapat mendukung penelitian. Pada bagian tiga dijelaskan gambaran umum sistem dan skenario pengujian yang telah dibuat. Pada bagian empat dijelaskan hasil pengujian yang telah dilakukan. Pada bagian lima dijelaskan kesimpulan dan saran dari hasil penelitian yang telah dilakukan.

2. Studi Terkait

Pada penelitian dalam tugas akhir ini, peneliti menelusuri paper yang berkaitan dengan penelitian sebagai petunjuk acuan ataupun perbandingan, baik dari segi kelebihan maupun kekurangan.

Penelitian yang dilakukan sebelumnya [4], Penelitian menjelaskan tentang sistem informasi peer-to-peer berdasarkan metrik XOR. Penelitian sistem peer-to-peer memiliki konsistensi dan kinerja yang dibuktikan dalam tempat yang rawan kesalahan. Sistem merutekan pencarian dan menempatkan node menggunakan topologi metrik berbasis XOR yang menyederhanakan algoritma dan memfasilitasi bukti. Topologi memiliki properti di setiap pesan yang dipertukarkan untuk menyampaikan dan memperkuat informasi yang berguna. Sistem mengeksploitasi informasi untuk mengirim query paralel dan menyempurnakan kegagalan pada simpul tanpa memperlama waktu pengguna. [4].

Penelitian yang dilakukan sebelumnya [7], Penelitian ini menjelaskan tentang keamanan pada *peer-to-peer Distributed Hash Table*. Penelitian sebelumnya berfokus pada penyediaan sistem pencarian *hash* yang efisien dan dapat digunakan untuk membangun sistem yang kompleks. Sistem tersebut memiliki kemampuan yang baik saat algoritma dieksekusi dengan benar namun secara umum belum mampu memperhitungkan gangguan pada node. Berdasarkan kekurangan tersebut, makalah ini mengidentifikasi jenis permasalahan keamanan pada sistem *peer-to-peer* berdasarkan sistem pencarian *hash*. Selain itu, dilakukan pengambilan contoh dari sistem yang sudah ada serta mengusulkan beberapa prinsip desain untuk mendeteksi dan mencegah permasalahan yang mungkin dihadapi oleh sistem *peer-to-peer*. [7].

Penelitian yang dilakukan sebelumnya [1], Penelitian ini menjelaskan tentang layanan *peer-to-peer* yang terukur untuk aplikasi internet. Masalah yang dihadapi pada *peer-to-peer* yaitu menemukan node yang menyimpan data tertentu. Tahapan yang dilakukan yaitu chord diberikan kunci untuk ke node. Kemudian lokasi data diimplementasikan dengan menghubungkan kunci dengan setiap data dan menyimpan data pada node. Chord yang bergabung dan meninggalkan system akan menyesuaikan secara efisien. Hasil dari analisis teoritis, simulasi, dan eksperimen menunjukkan bahwa chord dapat terukur. [1].

Penelitian yang dilakukan sebelumnya [5], Penelitian ini membahas tentang kombinasi antara *Distributed Hash Table* dengan *Blockchains* yang difokuskan pada *decoupling validation* dari *State Storage*. Kendala yang sering terjadi pada penelitian ini yaitu membutuhkan waktu untuk mengunduh semua data yang diperlukan agar *node* berjalan dengan lancar pada saat pembuatan simpul untuk *public blockchains*. Solusi yang ditawarkan yaitu menunjukkan desain yang memungkinkan pengambilan dan *validasi blok* baru hanya dengan mempertahankan situasi yang sangat kecil. Solusi lainnya yaitu menunjukkan kemungkinan untuk menyimpan status *blockchains* dalam tabel *hash* yang terdistribusi untuk mendapatkan spektrum yang luas antara penyimpanan yang dilakukan *node* dan faktor replikasi. [5].

Penelitian yang dilakukan sebelumnya [6], Penelitian ini menjelaskan tentang replikasi *datastore* untuk IoT. Penelitian *datastore* dirancang untuk layanan data sensor IoT yang berdaya rendah. *EdiSense* berjalan pada perangkat keras yang berfungsi sebagai kerangka untuk mereplikasi data sensor di jaringan yang didistribusikan. Tujuan penelitian tersebut adalah memberikan toleransi kesalahan dan ketersediaan data dalam

jaringan sensor. Pengoptimalan Edisense untuk ketersediaan dan pencarian data tersebut dilakukan dengan DHT.[6]

Peer-to-Peer (P2P)

Peer-to-Peer (P2P) adalah semua jaringan *node* yang dapat bertindak sebagai *client* atau *server* karena semua *node* saling terhubung satu sama lain sehingga *Peer-to-Peer* tidak memiliki pengaturan yang terpusat. *Peer-to-Peer* memungkinkan pengguna untuk berbagi, mencari dan mengunduh berkas. Jaringan *Peer-to-Peer* ini menyeimbangkan diri secara terus menerus tanpa perlu menambah waktu untuk pencarian alamat dan tidak harus menggunakan sumber terpusat[15].

Distributed Hash Table

Distributed Hash Table adalah sistem distribusi yang menyediakan layanan pencarian yang mirip dengan tabel *hash*. Pasangan (*key,value*) disimpan di dalam *node* pada DHT. Setiap *node* yang berpartisipasi secara efisien dapat mengambil kembali nilai yang terkait pada *key*[1]. Cara kerja DHT yaitu menyimpan data *hash* yang terdapat pada tabel kemudian didistribusikan ke semua jaringan *peer-to-peer*[1]. Setiap *peer* pada DHT dapat menyimpan dan memiliki informasi jaringan yang ada di dalamnya. Informasi tersebut meliputi rute yang akan ditempuh, lokasi data yang tersedia, dan lokasi permintaan data. Keuntungan utama pada *Distributed Hash Table* yaitu bahwa *node* dapat ditambahkan atau dihapus dengan mengubah *key*. Kegunaan *key* pada *Distributed Hash Table* yaitu untuk identifikasi yang memetakan nilai-nilai tertentu seperti alamat dan data.

File Transfer Protocol (FTP)

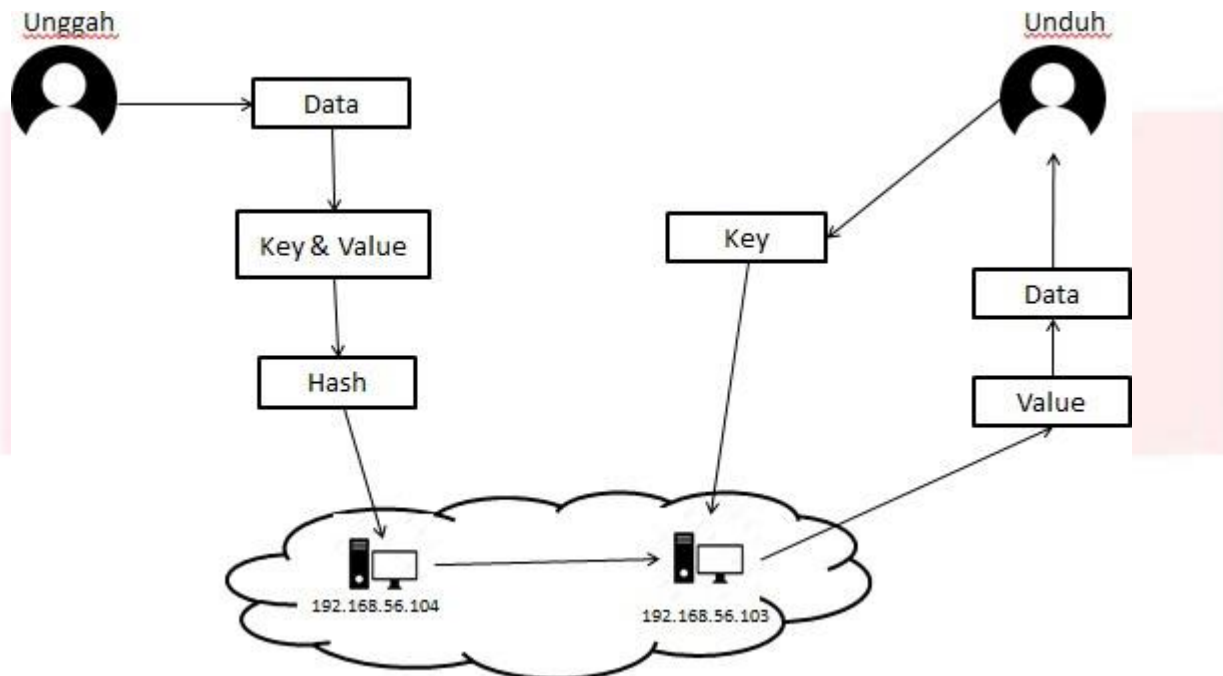
File transfer protocol adalah sebuah protocol yang memungkinkan untuk melakukan transfer file antar perangkat dalam satu jaringan yang menggunakan koneksi TCP. Pada file transfer protocol terdapat 2 jenis yaitu:

- a) FTP Server sebagai sebuah perangkat yang berfungsi untuk memberikan akses file dan menambahkan file kedalamnya.
- b) FTP Client berfungsi sebagai mengakses FTP server dengan menggunakan akun yang sudah diberikan akses oleh FTP server.

3. Sistem yang Dibangun

3.1 Gambaran Umum Sistem

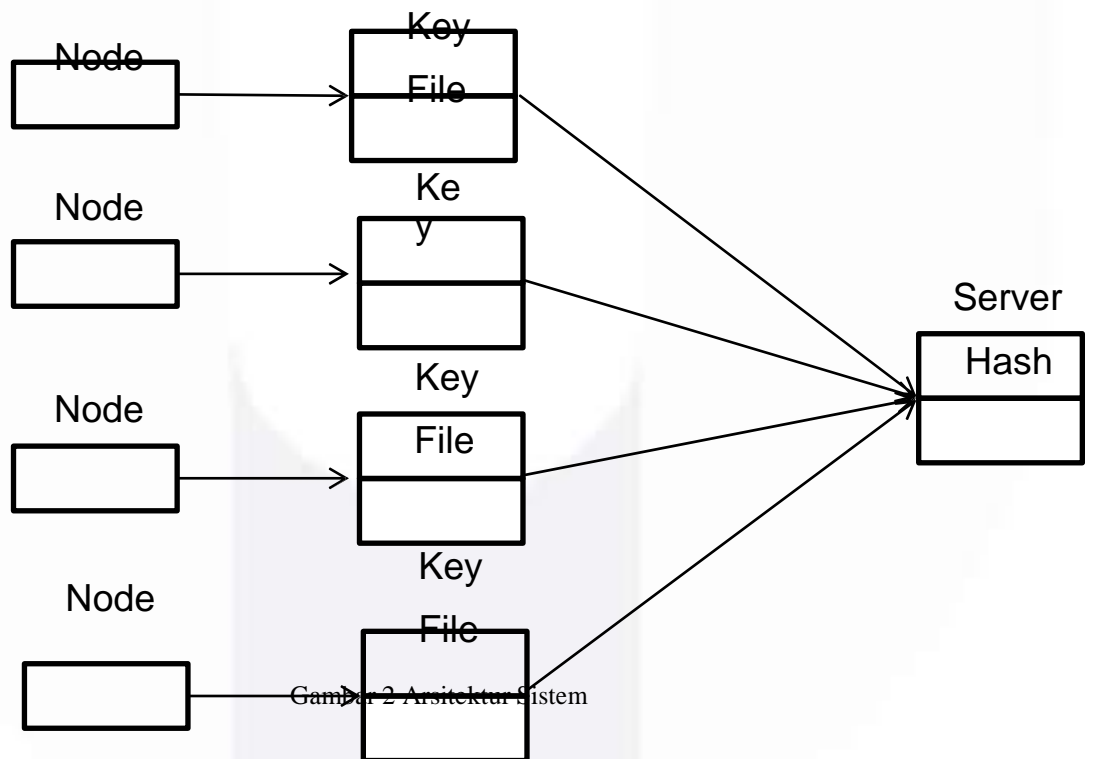
Sistem yang akan dirancang dan dibuat nantinya akan membantu mempermudah penyimpanan data menggunakan *Distributed Hash Table*. Pengguna mengunggah data yang akan disimpan pada server DHT, kemudian data diberi *key* yang berfungsi untuk memetakan data ke nilai-nilai tertentu. Key akan diterima oleh server DHT dalam bentuk hash. Selanjutnya, untuk mengambil data tersebut membutuhkan key supaya data tersebut dapat di unduh. Skema perancangan dapat dilihat pada Gambar 1.



Gambar 1 Rancangan Umum Sistem

3.2 Arsitektur Sistem

Arsitektur sistem terdiri dari *node*, *key*, *file*, *hash* dan *server DHT*. *Node* berfungsi sebagai *client* yang mengirimkan *key* yang berisi file kepada server DHT. *Key* diterima oleh server DHT dalam bentuk *hash*, data yang diunggah oleh client akan disimpan pada *server*. Setiap data yang sudah disimpan memiliki satu *hash node* tunggal yang dapat diakses di jaringan. Arsitektur sistem dapat dilihat pada Gambar 2.



3.3 Skenario Pengujian

Untuk mengetahui performansi sistem yang akan dibuat maka diperlukan pengujian sistem. Berikut merupakan skenario yang dibuat untuk menguji sistem.

1. Skenario Pengujian Performansi Unggah dan Unduh Data

Pengujian ini dilakukan untuk mengetahui delay dan throughput yang dibutuhkan saat melakukan unggah dan unduh pada berbagai macam data.

2. Skenario Pengujian Performansi *Distributed Hash Table*

Pengujian ini dilakukan untuk mengetahui performansi pada *Distributed Hash Table*.

4. Evaluasi

Tujuan pengujian pada penelitian tugas akhir ini adalah untuk menganalisis proses unggah dan unduh dengan berbagai tipe bentuk file menggunakan *Distributed Hash Table*. Adapun hasil dari skenario pengujian yang telah dilakukan sebagai berikut.

4.1 Hasil Pengujian

4.1.1 Hasil pengujian unggah dan unduh data

Pengujian unggah dan unduh data yang dilakukan adalah dengan cara mengirim file dari node ke node. Pada pengujian ini file yang dikirim berupa gambar, video dan .txt. Pada Gambar merupakan proses memasukkan *key* dan *value* lalu menginputkan file kedalam *key*.

```
D:\Kuliah\Skripsi\dht>python dht.py -c put -k testkey -val testvalue  
PUT OK testkey
```

Gambar 3 Proses Unduh Key dan Value

```
D:\Kuliah\Skripsi\dht>python dht.py -c put -k testkey -f 1.txt  
PUT OK testkey
```

Gambar 4 Proses Memasukkan File

```
D:\Kuliah\Skripsi\dht>python dht.py -c get -k testkey -o 1.txt
```

Gambar 5 Proses Unduh File

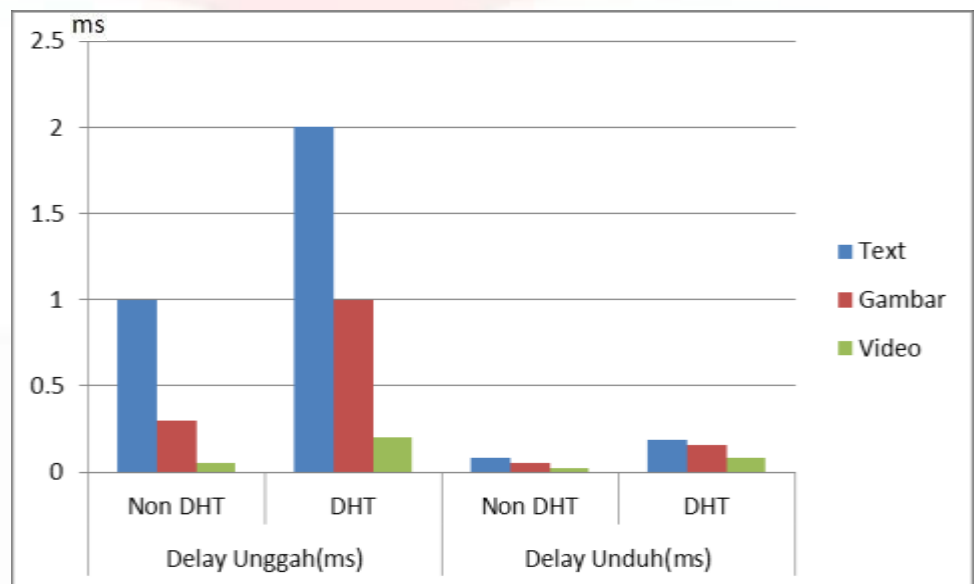
4.1.2 Analisis pengujian performansi unggah dan unduh data

Analisis pengujian terdapat 3 file yaitu:

- a) File text dengan ukuran file 6 bytes. Pada proses unggah terdapat delay 2ms dan throughput 991.65 bit/s. Pada proses unduh terdapat delay 0.19ms dan throughput 2230 bit/s.

- b) File gambar dengan ukuran file 10.8 KB. Pada proses unggah terdapat delay 1ms dan throughput 8005.3 bit/s. Pada proses unduh terdapat delay 0.16ms dan throughput 19831.34 bit/s.
- c) File video dengan ukuran file 1.7 MB. Pada proses unggah terdapat delay 0.2ms dan throughput 1135852.26 bit/s. Pada proses unduh terdapat delay 0.08ms dan throughput 2719202.37 bit/s.

Adapun perbandingan performansi menggunakan *Distributed Hash Table* dan tanpa *Distributed Hash Table*



Gambar 1 Performansi Delay

Bentuk File	Throughput Unggah(bps)		Throughput Unduh(bps)	
	Non DHT	DHT	Non DHT	DHT
Text	926	991	1845	2230
Gambar	6523	8005	14868	19831
Video	1086159	1135852	22684156	2719202

Tabel 1 Performansi Throughput

Perbandingan besar delay saat menggunakan DHT lebih besar dibandingkan menggunakan non DHT hal tersebut terjadi karena pada DHT terdapat media transmisi yang lebih banyak dibandingkan non DHT. Perbandingan pada throughput DHT lebih baik dibandingkan non DHT.

5. Kesimpulan

Kesimpulan dari penelitian tugas akhir ini adalah program mampu melakukan unggah dan unduh semua bentuk data dengan memasukkan *key* lalu disimpan pada server *Distributed Hash Table* yang disimpan dalam bentuk *hash*. Proses unggah dan unduh merupakan hasil dari berbagai macam tipe bentuk file. Hasil analisa performansi *delay* menggunakan *Distributed Hash Table* lebih besar sedangkan untuk *throughput* pada *Distributed Hash Table* lebih baik dibandingkan menggunakan non *Distributed Hash Table*.

Saran untuk penelitian selanjutnya diharapkan dari segi penyimpanan, sistem mampu melakukan *backup* data pada *Distributed Hash Table*.

Referensi

- [1] Stoica, I.; Morris, R.; Karger, D.; Kaashoek, M. F.; Balakrishnan, H.(2001). "Chord: A scalable peer-to-peer lookup service for internet applications" (PDF). *ACM SIGCOMM Computer Communication Review*. 31 (4): 149. doi: 10.1145/964723.383071. A value can be an address, a document, or an arbitrary data item.
- [2] Liz, Crowcroft; et al. (2005). "Sebuah survei dan perbandingan skema jaringan overlay peer-to-peer" (PDF). *Survei & Tutorial Komunikasi IEEE*. 7 (2): 72–93. doi: 10.1109 / COMST.2005.1610546.
- [3] Searching in a Small World Chapters 1 & 2 (PDF), retrieved 2012-01-10.
- [4] Maymounkov, Petar and Mazieres, David. *Kademlia: A Peer-to-Peer Information System Based on the XOR Metric*. New York University. 2002.
- [5] Bernardini, M. Diego, P. Maurizio, P. (2019). Blockchains Meet Distributed Hash Tables: Decoupling Validation from State Storage. *Sezione Informatica e Automazione*, Via della Vasca Navale 79, Roma, Italy.
- [6] J. Hong, J. Raymond, dan J. Shackelford, "EdiSense: A Replicated Datastore for IoT Data," p. 6, 2014.
- [7] Sit, E. & Robert, M. (2002). Security Consideration for Peer-toPeer Distributed Hash Tables. *International Workshop on Peer-to-Peer Systems*, 2429, 261-269.
- [8] Fersi G. 2015, A Distributed and Flexible Architecture for Internet of Things. *The Proceeding of The International Conference on Advanced Wireless, Information, and Communication Technologies*, Tunisia: 5-7 Oktober 2015. Hal. 130-137.
- [9] Bolonio J.J., Manuel U., Gonzalo C. 2013. A Distributed Control Plane for the Internet of Things Based on a Distributed Hash Table. *Mobile Networks and Management*. 125:9 108-121.
- [10] Shen Q., Yu L., Zhijun Z., Lagu C., Hui T. Distributed hash table based ID management optimization for internet of things. *Proceeding of the 6th International Wireless Communications and Mobile Computing Conference*, Caen: 28 Juni – 2 Juli 2010. Hal. 686-690.
- [11] Paganelli F. dan David P. 2012. A DHT-Based Discovery Service for the Internet of Things. *Journal of Computer Networks and Communications*. 2012:1 1-11.
- [12] Karnstedt M., Kai S., Martin R., Jessica M., Manfred H., Roman S., Renault J. 2007. UniStore: Querying a DHT-based Universal Storage. *Proceeding of IEEE 23rd International Conference on Data Engineering*, Istanbul: 15-20 April 2007.
- [13] Burange, A. W., & Misalkar, H. D. (2015). Review of Internet of Things in Development of Smart Cities with Data Management & Privacy.
- [14] Keoh, S. L., Kumar, S., & Tschofenig, H. (2014). Securing the Internet of Things: A Standardization Perspective. *IEEE Internet of Things Journal*, 1(3), 1–1. <http://doi.org/10.1109/JIOT.2014.2323395>.
- [15] B. Sosinsky. *Networking Bible*. Indiana: Wiley Publishing Inc., 2009.
- [16] Tiphon.1998. " Telecommunications and Internet Protocol Harmonization Over Network (TIPHON). General Aspects of Quality of Service (QoS)". DTR/TIPHON-05001