

Terbit online pada laman web jurnal: <https://jurnal.plb.ac.id/index.php/tematik/index>

TEMATIK

Jurnal Teknologi Informasi Komunikasi (e-Journal)

Vol. 10 No. 1 (2023) 61 - 66

ISSN Media Elektronik: 2443-3640

Implementasi Purwarupa Aplikasi Komunikasi Menggunakan Jaringan Delay Tolerant Network Disertai Fleksibilitas Pemilihan Teknik Routing *Implementation of Communication Application Prototypes Using Delay Tolerant Networks Accompanied by Flexibility in Routing Technique Selection*

Slameta¹, Ashari², Griffani Megiyanto Rahmatullah³

^{1,2,3}Prodi Teknik Telekomunikasi, Jurusan Teknik Elektro, Politeknik Negeri Bandung

¹slameta@polban.ac.id, ²ashari@polban.ac.id, ³griffani.megiyanto@polban.ac.id

Abstract

In the context of information exchange through the internet, communication technology plays a crucial role. However, its implementation still has shortcomings, such as disconnections, which frequently occur in areas with weak or intermittent signal coverage. As an alternative solution, wireless network technology can leverage the Delay Tolerant Network (DTN) architecture to address these situations. The conducted research involves creating a prototype data communication application that utilizes the DTN architecture with the implementation of routing feature selection, such as Epidemic, Spray and Wait, MaxProp, and ProphetV2. The application is developed using a customized GUI to facilitate data transmission and reception. Evaluation indicators measured include the success rate of data delivery for each routing type and data integrity during the transmission and reception of information at defined distances, with wireless media being the chosen communication medium. The system is tested using test scenarios based on full signal conditions and intermittent conditions. The test results show that the application is capable of functioning according to the initial design using the configured routing types, with an average variation in time ranging from 800-1000ms. Further development can be carried out by implementing the system on mobile devices, making it more flexible and convenient to use.

Keywords: delay tolerant network; epidemic; spray and wait; maxprop; prophet V2

Abstrak

Dalam konteks pertukaran informasi melalui jaringan internet, teknologi komunikasi memainkan peran yang sangat penting. Namun, implementasinya masih memiliki kekurangan, seperti putus koneksi, yang sering terjadi di daerah cakupan sinyal lemah atau intermitten. Sebagai solusi alternatif, teknologi jaringan wireless dapat memanfaatkan arsitektur Delay Tolerant Network (DTN) untuk mengatasi keadaan tersebut. Penelitian yang dilakukan merupakan pembuatan prototipe aplikasi komunikasi data dengan memanfaatkan penggunaan arsitektur DTN dengan implementasi pemilihan fitur routing seperti Epidemic, Spray and Wait, MaxProp, dan ProphetV2. Pembuatan aplikasi dibuat dengan menggunakan kustomisasi GUI sehingga memudahkan dalam hal pengiriman dan penerimaan data. Indikator penilaian yang diukur adalah jumlah berhasilnya pengiriman data pada setiap pilihan jenis routing serta keutuhan data pada pengiriman dan penerimaan informasi dengan jarak yang didefinisikan sedangkan media komunikasi yang digunakan yaitu media wireless. Sistem diuji menggunakan skenario uji berdasarkan kondisi sinyal penuh dan kondisi intermitten. Hasil pengujian menunjukkan bahwa aplikasi mampu berfungsi berdasarkan desain awal menggunakan jenis routing yang diatur dengan rata-rata variasi waktu berkisar 800-1000ms. Pengembangan lebih lanjut dapat dilakukan dengan melakukan implementasi pada perangkat mobile sehingga menjadi lebih fleksibel dan mudah untuk digunakan.

Kata kunci: delay tolerant network; epidemic; spray and wait; maxprop; prophet V2.

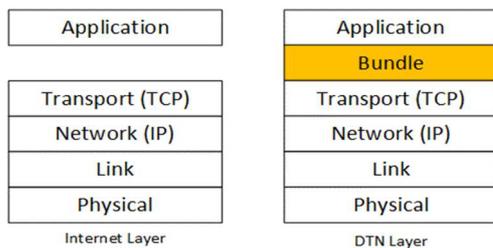
1. Pendahuluan

Kondisi jaringan merupakan aspek penting yang perlu diperhitungkan dalam memastikan keberhasilan sebuah komunikasi. Salah satu faktor penentu yaitu sifat dari

kondisi area yang dimaksud, seperti halnya intermittent area yang memiliki sifat putus koneksi secara berkala dikarenakan waktu delay yang cukup lama [1]. Waktu delay yang cukup lama dapat menyebabkan komunikasi yang dibangun melewati batas waktu timeout untuk menunggu response dari penerima. Pada studi [2] juga dijelaskan bahwa perlu dirumuskan dan dibutuhkan solusi alternatif untuk mengatasi permasalahan yang terjadi seiring dengan perkembangan teknologi. Salah satu solusinya yaitu penerapan arsitektur jaringan alternatif yang mampu bertahan pada kondisi intermittent.

Solusi yang ditawarkan adalah menerapkan arsitektur Delay Tolerant Network (DTN) sebagai arsitektur komunikasi antara source dan destination. Arsitektur tersebut dapat meminimalisir terjadinya kegagalan dalam komunikasi ketika terjadi putus koneksi sehingga data dapat dikirimkan pada saat koneksi sudah terhubung kembali. Pemilihan arsitektur DTN dapat dikatakan cocok untuk mengatasi permasalahan di intermittent area karena memiliki kelebihan berupa sifat non-connection-oriented. Dengan menggunakan sifat jaringan tersebut, DTN dapat melakukan koneksi ulang dengan fleksibilitas waktu yang ditetapkan oleh user apabila terjadi putus koneksi. Pada penelitian sebelumnya [3], telah dilakukan pembuatan aplikasi prototipe komunikasi menggunakan arsitektur DTN berfokus pada fungsionalitas penerapan arsitektur. Maka, diperlukan pengembangan lebih lanjut mengenai pembangunan jaringan dengan arsitektur tersebut sehingga mendapatkan hasil yang optimal.

Studi yang dilakukan oleh Kevin Fall menunjukkan bahwa arsitektur DTN dapat diterapkan dalam jaringan dengan penundaan waktu yang panjang dan koneksi yang tidak terjamin [2]. DTN adalah arsitektur jaringan yang dirancang untuk mengatasi masalah konektivitas yang tidak stabil, penundaan yang lama, kecepatan data yang bervariasi, dan tingkat kesalahan yang tinggi.



Gambar 1. Perbandingan arsitektur TCP/IP dan DTN

Secara garis besar, arsitektur DTN mencakup semua lapisan dalam arsitektur internet ditambah dengan lapisan tambahan yaitu bundle layer. Bundle layer ditempatkan di antara Application dan Transport layer seperti yang ditunjukkan pada Gambar 1. Bundle adalah unit data dasar yang terdiri dari ukuran variabel dan sinyal yang digunakan dalam jaringan DTN. Bundle layer memegang peran penting dalam DTN karena

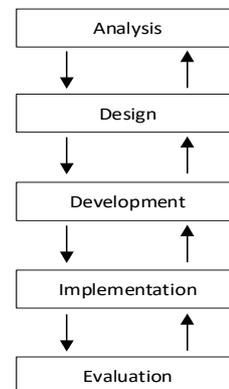
bertugas menyimpan dan mengirimkan seluruh atau sebagian bundle antar node [3].

Pada setiap node DTN dalam jaringan, pesan dikirimkan ke node lain melalui rute di mana pesan tersebut diteruskan dari sumber ke tujuan. Berbeda dengan arsitektur TCP/IP atau arsitektur internet yang menggunakan sifat connection-oriented, jalur komunikasi harus dibangun terlebih dahulu dan pengirim harus mengetahui jalur tersebut untuk mencapai tujuan. Jika terjadi putus koneksi, kemungkinan packet loss akan tinggi karena tidak ada penyimpanan sementara untuk paket yang sedang dikirim. Di sisi lain, arsitektur DTN menawarkan konsep berbeda, yaitu tidak selalu mengharuskan ketersediaan jaringan dan mengatur agar node menyimpan bundle dalam jangka waktu tertentu. Proses penyimpanan sementara dan pengiriman ke node berikutnya disebut dengan store-carry-forward [4]. Beberapa penelitian [5]-[9] secara garis besar memberikan informasi mengenai performa dan implementasi nyata penggunaan jaringan arsitektur DTN.

Pembuatan aplikasi bertujuan untuk dapat membangun komunikasi antara dua pengguna dengan prinsip komunikasi chat. Selain itu, aplikasi juga dibangun berfokus pada fitur selektif mengenai jenis routing yang dapat digunakan. Hal ini akan dapat bermanfaat seperti halnya untuk digunakan di daerah yang sulit dan sering putus sinyal untuk melakukan komunikasi dengan jalur routing yang optimal. Spesifikasi khusus yang digunakan adalah pengiriman dan penerimaan data menggunakan komputer portable, menggunakan jaringan wifi, serta melakukan manajemen pemilihan routing oleh user untuk mendukung fungsionalitas komunikasi. Pemilihan perangkat serta jaringan komunikasi dianggap sebagai ilustrasi komunikasi antara pengirim dan penerima di situasi nyata.

2. Metode Penelitian

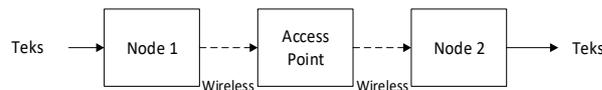
Metode penelitian secara umum menggunakan konsep Analyze, Design, Develop, Implement, dan Evaluate (ADDIE). Metode tersebut ditunjukkan pada Gambar 2.



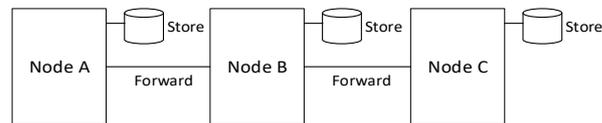
Gambar 2. Metode ADDIE

Dalam model penelitian pengembangan ADDIE, tahap pertama adalah menganalisis keperluan (model, metode, media) dan menganalisis kelayakan untuk pengembangan aplikasi. Tahapan kedua, Design, melibatkan proses perancangan yang sistematis, yaitu merancang konsep, konten, dan struktur aplikasi yang dikembangkan. Pada tahap ini, aspek-aspek seperti desain antarmuka pengguna, interaksi, dan alur kerja, serta memastikan bahwa aplikasi tersebut dapat digunakan dengan mudah oleh pengguna. Tahap ketiga, Development, merupakan realisasi dari rancangan aplikasi yang telah dibuat sebelumnya. Tahap keempat, Implementation, melibatkan pengujian dan umpan balik terhadap kinerja aplikasi yang dikembangkan. Proses identifikasi dilakukan untuk mengatasi masalah dan area yang memerlukan perbaikan. Umpan balik ini kemudian digunakan untuk meningkatkan kinerja aplikasi. Tahap terakhir, Evaluation, yaitu melakukan evaluasi terhadap efektivitas aplikasi dan mengidentifikasi area yang dapat ditingkatkan lebih lanjut. Setiap tahap dalam metode penelitian ADDIE saling berkomunikasi, memungkinkan penyesuaian dan peningkatan fleksibilitas dalam pengembangan aplikasi.

Dengan menggunakan konsep metode ADDIE, perancangan sistem akan lebih mudah dilakukan secara bertahap dan diimplementasikan. Teknis perancangan sistem ditunjukkan oleh Gambar 3.



Gambar 3. Ilustrasi Blok Sistem

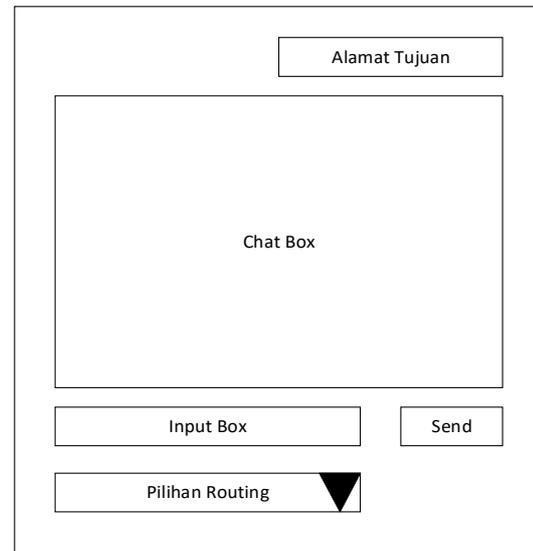


Gambar 4. Ilustrasi Store-Carry-Forward

Rancangan aplikasi ditunjukkan pada Gambar 5. Aplikasi dibuat menggunakan tkinter, yaitu salah satu Python library untuk pembuatan antarmuka grafis. Desain antarmuka utama aplikasi ini merupakan sebuah kotak chat, yang memfasilitasi interaksi antara satu pengguna dengan pengguna lainnya. Kotak chat ini berisi percakapan pengguna secara real-time, memungkinkan dialog interaktif dan komunikasi dua arah. Lalu, ditambahkan juga kotak pilihan dengan tipe menu dropdown untuk mencakup pilihan jenis routing yang akan digunakan.

Selain itu, aplikasi ini menggunakan kotak input untuk pengguna memasukkan pesan yang akan dikirimkan. Kotak ini berfungsi sebagai penampung data sehingga pengguna dapat menulis dan mengedit pesan yang akan dikirimkan. Berikutnya adalah peletakan tombol 'send'

disamping kotak input yang berfungsi untuk tombol eksekusi pengiriman pesan.



Gambar 5. Rancangan desain aplikasi

Aplikasi yang dirancang meliputi input Teks yang dimasukkan ke Node 1 dan dikirimkan kepada Node 2 sebagai penerima menggunakan media wireless. Prototype aplikasi komunikasi data tersebut menggunakan basis perangkat lunak IBR-DTN. IBR-DTN adalah suatu protokol routing yang dirancang untuk mengatasi tantangan dalam jaringan dengan koneksi yang sering terputus atau memiliki waktu tunda yang lama. IBR-DTN menjadi salah satu pendekatan yang efisien dalam menghadapi lingkungan jaringan yang tidak stabil, dengan menyimpan dan meneruskan paket secara adaptif sesuai dengan kondisi jaringan. Protokol ini juga memanfaatkan prediksi kontak antarnode untuk meningkatkan kinerja pengiriman pesan [10]. IBR-DTN menggunakan prinsip store-carry-forward (ditunjukkan pada Gambar 4), node menyimpan paket saat koneksi tidak tersedia dan meneruskannya ketika kontak dengan node lain terjalin. Hal ini memungkinkan protokol untuk menyesuaikan strategi routingnya berdasarkan karakteristik jaringan yang berubah-ubah. Berikutnya dilakukan pemilihan jenis routing yang kompatibel dengan arsitektur DTN dilengkapi kustom GUI untuk memudahkan pengiriman dan penerimaan data. Berikut pemilihan jenis routing yaitu Epidemic, Spray and Wait, MaxProp, dan Prophet V2.

Epidemic routing merupakan protocol yang menyebar data ke semua node di jaringan [11]. Ketika sebuah node menerima pesan, pesan tersebut disimpan dan disebarkan ke semua node lainnya yang ditemuinya. Protokol ini mengandalkan distribusi transitive pesan melalui jaringan ad hoc, dengan pesan akhirnya mencapai tujuannya. Setiap host menyimpan buffer yang terdiri dari pesan yang dihasilkannya dan pesan yang disimpan untuk host lain. Ketika dua host

berada dalam jangkauan komunikasi satu sama lain, host dengan status identifier yang lebih kecil akan memulai sesi anti-entropy [12] dengan host yang memiliki status identifier yang lebih besar. Untuk menghindari koneksi berlebihan, setiap host menyimpan cache host yang terbaru. Meskipun proses ini dapat memastikan pengiriman data, tetapi dapat menghabiskan banyak sumber daya jaringan.

Spray and Wait, adalah proses dimana Routing protocol mengirimkan beberapa salinan pesan ke beberapa node secara acak dan menunggu hingga ada kontak langsung dengan node tujuan [13]. Ketika ada kontak langsung, salah satu dari salinan pesan akan dikirimkan ke node tujuan. Ketika data sudah terkirim sampai dengan tujuan, maka pengiriman pesan ack dilakukan Kembali dari penerima ke pengirim. Hal tersebut untuk memberitahukan terkait status pesan yang telah dikirim. Protocol ini efektif pada jaringan yang memiliki banyak node, tetapi membutuhkan waktu yang lama untuk pengiriman data.

MaxProp routing merupakan salah satu algoritma routing yang dirancang khusus untuk Delay Tolerant Networks [14]. Algoritma ini mengoptimalkan penyebaran pesan dengan memprioritaskan pengiriman pesan berdasarkan estimasi peluang pengiriman yang berhasil ke tujuan. Dalam MaxProp, setiap node mengestimasi peluang pengiriman yang berhasil ke setiap node lain berdasarkan riwayat pertemuan antara node tersebut. Pesan dengan peluang pengiriman yang lebih tinggi akan diteruskan terlebih dahulu. Selain itu, MaxProp juga mengelola penyimpanan buffer dengan cara menghapus pesan yang sudah mencapai tujuan atau memiliki peluang pengiriman yang rendah untuk memberikan ruang bagi pesan baru yang masuk.

Probabilistic Routing Protocol using History of Encounters and Transitivity version 2 (ProphetV2) [15] merupakan algoritma routing yang dirancang khusus untuk Delay Tolerant Networks (DTN). Algoritma ini menggunakan pendekatan berbasis probabilitas untuk menentukan jalur pengiriman pesan yang optimal. Dalam ProphetV2, setiap node mengestimasi peluang pengiriman pesan yang berhasil ke setiap node lain berdasarkan riwayat pertemuan antara node tersebut dan sifat transitif dari pertemuan tersebut. Ketika dua node bertemu, mereka bertukar informasi tentang peluang pengiriman yang berhasil ke node lain dan memperbarui estimasi peluang pengiriman mereka masing-masing. Algoritma ini mengoptimalkan pengiriman pesan dengan memilih jalur yang memiliki peluang pengiriman yang lebih tinggi, sehingga meningkatkan efisiensi pengiriman pesan.

Jenis komunikasi tersebut akan dimasukkan kedalam opsi pemilihan jenis routing pada aplikasi yang dirancang. Proses yang dijalankan dimulai dengan menyediakan sumber dan tujuan sebagai parameter yang akan dimasukkan ke dalam skrip shell DTN.

Sumber dan tujuan dianggap sebagai ID, sehingga alamat pengiriman dan penerimaan lebih fleksibel dan tidak terbatas pada alamat IP. Kemudian, pengguna diminta untuk memilih salah satu tipe routing yang tersedia. Pilihan tipe routing akan disinkronkan antara klien untuk menghindari perbedaan dalam jenis routing. Selanjutnya, proses pendeteksian file temporer dilakukan, yang berfungsi sebagai penyimpanan sementara data yang diterima dari klien lain. Jika terdapat perubahan dalam penyimpanan basis data, maka dianggap sebagai data yang diterima dari klien lain dan ditampilkan di layar aplikasi. Susunan prosedur teknis aplikasi secara detail ditunjukkan pada Tabel 1.

Tabel 1. Prosedur prototipe aplikasi komunikasi menggunakan jaringan dengan arsitektur DTN

Deskripsi: Prototipe Aplikasi Komunikasi	
1:	Mulai
2:	Masukkan sumber dan tujuan sebagai ID (bukan alamat IP)
3:	Tampilkan pilihan jenis routing:
4:	- Epidemic Routing
5:	- Spray and Wait Routing
6:	- MaxProp Routing
7:	- ProphetV2 Routing
8:	Pengguna memilih jenis routing
9:	Sinkronkan pilihan jenis routing antara klien
10:	Lakukan proses berikut secara berulang sampai sesi diakhiri:
11:	a. Deteksi file temporer sebagai penyimpanan sementara data yang diterima dari klien lain
12:	b. Jika ada perubahan dalam penyimpanan basis data:
13:	i. Anggap sebagai data yang diterima dari klien lain
14:	ii. Tampilkan data di layar aplikasi
15:	c. Pantau kolom chat box untuk data yang akan dikirim
16:	d. Jika data siap dikirim (tombol "Enter" ditekan):
17:	i. Gunakan skrip shell DTN untuk mengirim data ke klien tujuan
18:	Akhiri sesi jika salah satu klien mengakhiri sesi
19:	Selesai

Selanjutnya, sistem akan terus memantau apakah ada data yang akan dikirim ke klien lain melalui kolom chat box dengan pemicu tombol "Enter". Jika data yang siap dikirim terdeteksi, aplikasi akan menggunakan skrip shell DTN untuk mengirimkannya ke klien tujuan. Kedua proses ini (penerimaan dan pengiriman) akan terus dilakukan secara berulang hingga sesi diakhiri oleh salah satu klien. Informasi lainnya mengenai perangkat keras yang digunakan sebagai node pengirim dan penerima ditunjukkan pada Tabel 2.

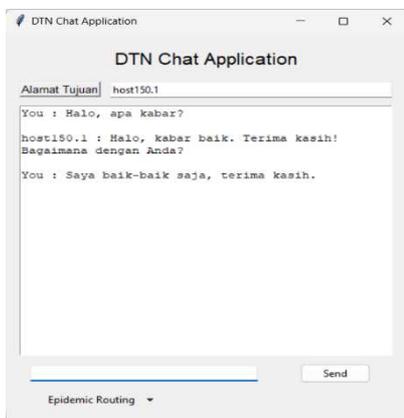
Tabel 2. Spesifikasi perangkat keras sebagai node pengirim dan penerima

Spesifikasi	Keterangan
SoC	BCM2837
Processor	1.2GHZ 64-bit quad-core ARMv8
Memory/RAM	CPU
GPU	1 GB SDRAM 400 MHz
RTC	VideoCore IV 3D graphics core
Wireless	DS3231
Adapter/LAN	802.11n Wireless LAN
Storage	Micro SD card 16GB
Sistem Operasi	Debian GNU/Linux
Catu daya	DC Adaptor 5V 3A

3. Hasil dan Pembahasan

Aplikasi yang direalisasikan berdasarkan analisis yang telah dilakukan dalam tahapan rancangan. Aplikasi didesain untuk mengirimkan pesan antara dua orang pengguna sehingga dapat dilakukan interaksi chat secara real time. Aplikasi yang direalisasikan juga memenuhi kebutuhan tampilan isi chat yang sedang dilakukan dan menampilkan pilihan jenis routing yang dapat dipilih pada saat inisiasi obrolan dengan penerima. Terdapat evaluasi terkait antarmuka aplikasi yaitu agar ditampilkan alamat tujuan dari penerima. Dengan menerapkan evaluasi tersebut, maka aplikasi menjadi lebih fleksibel dalam hal pengiriman teks ke beberapa penerima.

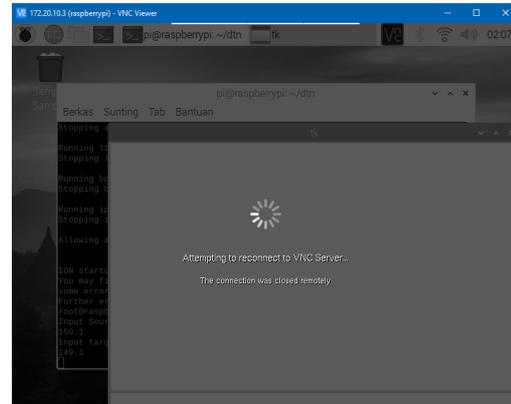
Hasil lainnya yang didapatkan telah diujikan secara fungsionalitas dan juga kehandalan. Pengujian fungsionalitas dilakukan dengan cara melakukan pengiriman dan penerimaan data berupa teks. Berikutnya, pengujian kehandalan sistem dilakukan dengan cara melakukan komunikasi saat kondisi sinyal penuh dan kondisi sinyal putus. Kondisi sinyal penuh maksudnya adalah kedua perangkat terhubung satu sama lain dalam jaringan. Sedangkan sinyal putus adalah kondisi saat salah satu *node* atau perangkat terputus dari jaringan. Secara lebih detail, pengujian tersebut dilakukan dengan memutuskan koneksi Node 1 ke jaringan. Seluruh perangkat terhubung dalam jaringan yang sama dengan memanfaatkan jaringan wireless yang dipancarkan oleh mobile hotspot 2.4 GHz. Hasil pengujian ditunjukkan pada Gambar 6 dan Gambar 7.



Gambar 6. Hasil pengujian antara dua klien

Aplikasi yang dirancang berhasil diimplementasikan dan berfungsi dengan melakukan pengujian sederhana pengiriman teks dari node 1 ke node 2. Tampilan teks setiap kiriman data akan ditampilkan pada chat box masing-masing node pengirim dan penerima. Pada prototipe aplikasi juga dibuatkan pilihan jenis routing sesuai dengan perancangan sebelumnya. Pengujian juga dilakukan saat melakukan koneksi putus dan disambungkan kembali dengan variasi waktu yang ditentukan. Hasil yang didapatkan yaitu pengiriman

teks masih memungkinkan dilakukan karena dilakukan penyimpanan buffer dengan durasi waktu yang lama sehingga pesan dari pengirim masih tersimpan dan siap dikirimkan ketika terkoneksi kembali.

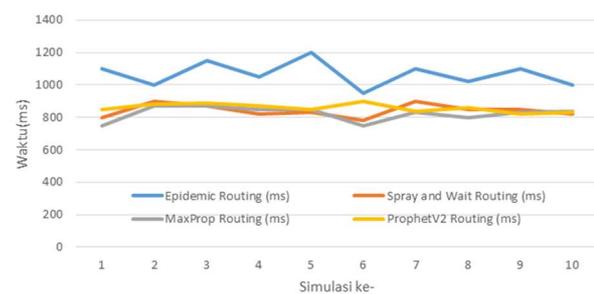


Gambar 7. Hasil pengujian pemutusan koneksi

Berikutnya dilakukan pengujian kehandalan sistem yaitu melakukan komunikasi dengan 10 kali simulasi untuk melihat waktu latency semua proses dari pengiriman dan penerimaan data teks. Hasil pengujian tersebut ditunjukkan pada Tabel 3 dan Gambar 8.

Tabel 3. Tabel hasil pengukuran nilai latency pengujian pengiriman teks

Simulasi	Epidemic Routing (ms)	Spray and Wait Routing (ms)	MaxProp Routing (ms)	ProphetV2 Routing (ms)
1	1100	800	750	850
2	1000	900	870	880
3	1150	870	870	890
4	1050	820	850	870
5	1200	830	850	850
6	950	780	750	900
7	1100	900	830	840
8	1020	850	800	860
9	1100	850	830	820
10	1000	820	840	830



Gambar 8. Grafik hasil pengukuran latency

Epidemic routing mendapatkan nilai latency rata-rata 1067ms, sedangkan routing protocol lain mendapatkan latency rata-rata dibawah 1000ms. Secara lebih detail, nilai rata-rata latency Spray and Wait adalah 842ms, MaxProp 824ms, dan ProphetV2 859ms. Berdasarkan nilai tersebut maka protokol Spray and Wait memiliki kinerja yang lebih baik dibandingkan dengan protokol lainnya. Namun, nilai ini didapatkan hanya berdasarkan skenario uji yang telah dirancang dan dapat berbeda

secara dunia nyata. Perbedaan nilai tersebut dapat dikarenakan beberapa faktor seperti halnya topologi jaringan, mobilitas simpul, ukuran buffer, dan tingkat generasi pesan. Dalam beberapa kasus, protokol routing ProphetV2 mungkin lebih unggul daripada MaxProp, sementara dalam skenario lain, MaxProp memungkinkan untuk lebih efektif.

4. Kesimpulan

Pembuatan aplikasi chat disertai penerapan arsitektur jaringan DTN telah berfungsi sesuai perancangan yang telah dilakukan. Aplikasi chat dapat memanfaatkan penyimpanan file sementara untuk menampung data yang dikomunikasikan antara pengirim dan penerima. Kehandalan prototipe aplikasi chatting menggunakan arsitektur DTN telah berhasil teruji untuk masing-masing jenis routing protocol dengan hasil variasi nilai latency yang berbeda. Dengan situasi dan kondisi yang telah didefinisikan sebelumnya, kemampuan routing protocol MaxProp memiliki performa lebih baik dibandingkan dengan protocol yang lain. Namun, hal tersebut tidak dapat menjadi patokan umum karena performa routing protocol juga dipengaruhi beberapa factor seperti halnya topologi jaringan, ukuran buffer, juga algoritma komunikasi data. Walaupun secara fungsionalitas sudah memenuhi tujuan penelitian yang dirancang, penelitian juga dapat dikembangkan lebih lanjut dengan fokus pada fleksibilitas aplikasi. Salah satunya yaitu dengan membuat prototipe aplikasi yang dapat diimplementasikan secara mobile. Hal ini akan menjadi lebih praktis digunakan oleh pengguna mengingat perangkat komunikasi yang umum digunakan saat ini adalah perangkat mobile.

Ucapan Terimakasih

Ucapan terima kasih kepada Unit Penelitian dan Pengabdian Masyarakat Politeknik Negeri Bandung dengan bantuan pendanaan penelitian yang tertuang pada surat nomor: B/98.24/PL1.R7/PG.00.03/2023.

Daftar Rujukan

- [1] L. Gao et al., "Routing protocols in delay tolerant networks," *Delay Tolerant Networks*, pp. 19–34, 2015.

- [2] K. Fal, "A delay—tolerant network architecture for challenged internets [C]," in *Proc of ACM SIGCOMM*, 2003, vol. 3, pp. 27–34.
- [3] M. H. P. Situmorang, K. F. Firdaus, and others, "Prototype Of A Desktop-Based Communication Application Using A Delay Tolerant Network Architecture," *Jurnal Teknologi Informasi Universitas Lambung Mangkurat (JTIULM)*, vol. 7, no. 2, pp. 99–104, 2022.
- [4] S. D. Siswanti, "Pengembangan sistem aplikasi pengiriman data daerah terpencil berbasis Delay Tolerant Network," *Generic*, vol. 8, no. 2, pp. 238–253, 2013.
- [5] I. Fadillah and others, "Sistem Komunikasi Jaringan Wireless Menggunakan Raspberry Pi Dengan Arsitektur Delay Tolerant Network," in *Prosiding Industrial Research Workshop and National Seminar*, 2020, vol. 11, no. 1, pp. 429–434.
- [6] H. E. Wahanani, I. Suartana, and D. Adityawati, "Analisa kinerja protokol routing Delay Tolerant Network (DTN) untuk transportasi publik," *Scan: Jurnal Teknologi Informasi dan Komunikasi*, vol. 10, no. 3, pp. 39–46, 2015.
- [7] D. D. Mursyitah and others, "Perbandingan Kinerja ION-DTN Dan IBR-DTN Menggunakan Raspberry Pi Sebagai Router Delay Tolerant Network," in *Seminar Nasional Teknologi Informasi Komunikasi dan Industri*, pp. 286–294.
- [8] H. Ntareme, M. Zennaro, and B. Pehrson, "Delay tolerant network on smartphones: Applications for communication challenged areas," in *Proceedings of the 3rd Extreme Conference on Communication: The Amazon Expedition*, 2011, pp. 1–6.
- [9] M. N. Jamal, R. Asif, F. N. Khan, and M. Z. ur Rehman, "Disaster Management through Delay Tolerant Networks," *International Journal of Future Generation Communication and Networking*, vol. 10, no. 3, pp. 57–69, 2017.
- [10] S. Schildt, J. Morgenroth, W.-B. Pöttner, and L. Wolf, "IBR-DTN: A lightweight, modular and highly portable Bundle Protocol implementation," *Electronic Communications of the EASST*, vol. 37, 2011.
- [11] A. Vahdat, D. Becker, and others, *Epidemic routing for partially connected ad hoc networks*. Technical Report CS-200006, Duke University, 2000.
- [12] D. B. Terry, M. M. Theimer, K. Petersen, A. J. Demers, M. J. Spreitzer, and C. H. Hauser, "Managing update conflicts in Bayou, a weakly connected replicated storage system," *ACM SIGOPS Operating Systems Review*, vol. 29, no. 5, pp. 172–182, 1995.
- [13] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, 2005, pp. 252–259.
- [14] Y. Harrati and A. Abdali, "MaxHopCount: DTN congestion control algorithm under MaxProp routing," *IJCSNS*, vol. 17, no. 5, p. 206, 2017.
- [15] S. Grasic, E. Davies, A. Lindgren, and A. Doria, "The evolution of a DTN routing protocol-PROPHETv2," in *Proceedings of the 6th ACM Workshop on Challenged Networks*, 2011, pp. 27–30.