

CUSTOM CRC DAN MANAJEMEN KOMUNIKASI PENGIRIMAN DAN PENERIMAAN DATA PADA ARSITEKTUR *DELAY TOLERANT NETWORK*

Griffani Megiyanto Rahmatullah¹⁾, Slameta²⁾, Muhamad Rizki³⁾

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

Jl. Gegerkalong Hilir, Ds.Ciwaruga Kecamatan Parongpong Kabupaten Bandung Barat 40559

e-mail: griffani.megiyanto@polban.ac.id¹⁾, slameta@polban.ac.id²⁾, muhamad.rizki.tkom18@polban.ac.id³⁾

ABSTRAK

Proses pertukaran pada saat ini banyak yang menggunakan media jaringan internet beriringan dengan berkembangnya teknologi IoT. Untuk mendukung teknologi tersebut dibutuhkan jaringan internet yang memadai di setiap kondisi tempat, terutama pada tempat yang masih rentan terjadi putus koneksi seperti pada area terbuka perkebunan. Masalah tersebut dapat diatasi dengan mengaplikasikan arsitektur Delay Tolerant Network (DTN) dengan konfigurasi protokol User Datagram Protocol (UDP) yang memiliki keunggulan dalam kecepatan transmisi pengiriman data, namun memiliki kelemahan belum dilengkapi dengan sistem cek error dan memiliki batas pengiriman hanya untuk file berukuran dibawah 64KB. Tujuan dari penelitian ini adalah untuk membuat sistem komunikasi pengiriman UDP pada arsitektur DTN dengan fitur cek error CRC dan dapat mengirimkan file lebih dari 64 KB dengan merealisasikannya pada perangkat keras dua buah raspberry pi 3 dan perangkat lunak yang digunakan adalah VNC, aplikasi IBR-DTN, dan bahasa pemrograman python. Pengujian dilakukan dengan mengirimkan gambar format PNG dengan ukuran 62,5 KB dan 350 KB baik pada saat kondisi jaringan stabil maupun putus koneksi dengan hasil gambar yang diterima utuh yang menunjukkan sistem secara fungsionalitas berhasil dibuat.

Kata Kunci: UDP, IBR-DTN, Custom CRC, Raspberry Pi, Komunikasi Data

ABSTRACT

The exchange process at this time uses internet network media along with the development of IoT technology. To support this technology, an adequate internet network is needed in every condition of the place, especially in places that are still prone to connection drops, such as in open plantation areas. This problem can be overcome by applying the Delay Tolerant Network (DTN) architecture with the User Data Salt Protocol (UDP) configuration which has advantages in the speed of data transmission. However, it has the disadvantage that it is not equipped with an error checking system and has a sending limit for files under 64KB in size. The purpose of this research is to create a UDP delivery communication system on DTN architecture with a CRC error check feature and can send files more than 64KB by realizing it on two raspberry pi 3 hardware and the software used is VNC, IBR-DTN application, and Python programming language. The test is carried out by sending PNG format images with sizes of 62,5 KB and 350KB both when the network conditions are stable or when the connection is lost with the results of the received images intact which shows the system functionality has been successfully created.

Keywords: UDP, IBR-DTN, Custom CRC, Raspberry Pi, Komunikasi Data

I. PENDAHULUAN

Pada saat ini pertukaran data banyak yang memanfaatkan jaringan internet. Sektor perkebunan pun saat ini menjadi objek penelitian dalam pengembangan penggunaan teknologi berbasis *Internet of Things* (IoT) dengan harapan hasil pertanian menjadi maksimal dengan adanya bantuan teknologi informasi sehingga pemantauan dan pengelolaan tanaman pun menjadi mudah [1]. Lahan pertanian yang hendak menggunakan teknologi IoT termasuk daerah yang memiliki kondisi konektivitas *intermittent*, terlebih jaringan internet pada umumnya menggunakan protokol internet TCP/IP yang belum dapat menangani permasalahan tersebut [2]. Pada penelitian sebelumnya, terdapat solusi untuk mengatasi permasalahan tersebut yaitu mekanisme komunikasi dengan menerapkan arsitektur *Delay Tolerant Network* (DTN) yang diterapkan pada perangkat android dengan hasil pengujian sistem dapat mengirimkan pesan walaupun pada kondisi jaringan *intermittent* [3][4-5].

Arsitektur *Delay Tolerant Network* merupakan arsitektur jaringan internet dengan konsep *store-carry-forward* yang dapat melakukan komunikasi data pada daerah dengan kondisi jaringan *intermittent*, *long delay*, dan terjadi putus koneksi [6]. Penggunaan DTN ini dapat diatur konfigurasi protokol internet yang akan digunakan yaitu TCP/IP atau UDP. Pada penelitian sebelumnya telah dibuat sistem komunikasi jaringan wireless menggunakan raspberry pi dengan arsitektur *delay tolerant network*, konfigurasi protokol yang digunakannya adalah TCP/IP, dari penelitian ini diperoleh hasil sistem komunikasi berhasil difungsikan saat kondisi jaringan sinyal penuh dan sinyal berselang. Pengujian dibandingkan dengan pengiriman menggunakan protokol UDP, diperoleh hasil DTN lebih

unggul digunakan untuk kondisi jaringan putus koneksi dan lebih handal dalam menjaga keutuhan data yang ditranmisikan sedangkan UDP unggul dalam kecepatan transmisi data[7].

Berdasarkan permasalahan tersebut maka akan dibuatkan realisasi sistem pengiriman data pada daerah rentan putus koneksi dengan menerapkan arsitektur DTN. Konfigurasi protokol yang digunakan adalah UDP dilengkapi dengan program cek kesalahan data selama transmisi yaitu *cylic redundancy check* (CRC) yang akan diaplikasikan pada dua buah raspberry pi sebagai perangkat keras *node* pengirim dan penerima. Dengan adanya sistem ini maka pada area dengan potensi *delay* tinggi diharapkan masih dapat melakukan pengiriman data, dan sifat UDP yang rentan kehilangan data dapat teratasi dengan adanya tambahan bit CRC untuk memastikan keutuhan data yang dikirimkan selama komunikasi berlangsung[8][9].

II. METODE PENELITIAN

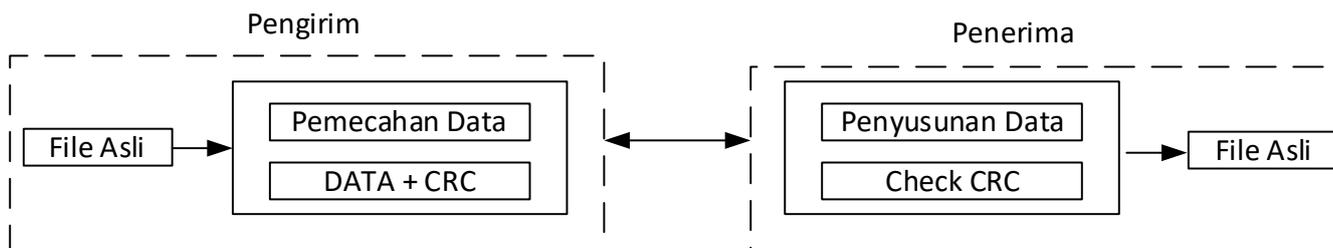
Berikut ini merupakan metode-metode yang dilakukan peneliti dalam pelaksanaan penelitian ini.

A. Persiapan Realisasi

Dalam proses persiapan realisasi sistem yang akan dibangun, dilakukan terlebih dahulu studi pustaka terkait untuk mendukung dasar teori dalam proses penelitian ini. Dasar teori utama yang dikaji dalam penelitian ini adalah arsitektur DTN, IBR-DTN, raspberry pi, protokol UDP, dan CRC. Berikut ini merupakan persiapan realisasi yang dilakukan.

1. Diagram Blok

Tahap persiapan realisasi dimulai dengan perencanaan blok diagram sistem yang akan dibuat dalam penelitian ini sebagai berikut.



Gambar. 1. Diagram Blok Sistem

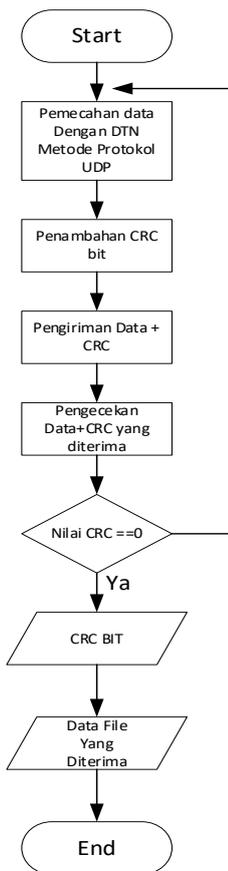
Berdasarkan Gambar 1., sistem terdiri dari bagian pengirim dan penerima yang merupakan perangkat keras raspberry pi 3 yang sudah mendukung layanan DTN dengan menggunakan aplikasi IBR-DTN. Pada blok pengirim, file asli yang merupakan sebuah data gambar akan dikonversi terlebih dahulu menjadi potongan gambar yang lebih kecil dan dibaca dalam bentuk deretan bit kemudian proses penambahan bit CRC dengan menggunakan logika XOR. Maka file yang dikirim merupakan potongan data gambar yang telah disertai dengan bit CRC. Kemudian pada sisi penerima, potongan data gambar yang telah diterima akan dilakukan proses pengecekan bit CRC dan dilakukan proses perbandingan bit yang diterima dengan bit pembandingan menggunakan program yang dibuat dengan bahasa pemrograman python. Apabila data yang diterima menghasilkan bit CRC 0, maka data yang telah diterima sesuai dan akan disimpan. Namun, apabila bit CRC yang diterima tidak sama dengan 0, maka data yang diterima tidak sesuai dengan yang dikirim. Maka, penerima akan melakukan respon kepada pengirim untuk melakukan pengiriman ulang data pada data yang terdeteksi *error* tersebut.

2. Flowchart Sistem

Berikut ini merupakan gambaran dari alur sistem yang dibuat.

1. File berupa gambar dengan format PNG akan dikonversi menjadi data biner.
2. Data biner gambar akan dipecah kembali menjadi beberapa potongan data dan pada penelitian ini data dibagi menjadi per 64000 bit data.
3. Setiap data 64000 bit akan diproses CRC untuk mendapatkan bit tambahan CRC sebagai fitur cek kesalahan yang selanjutnya akan dikirimkan dengan menggunakan arsitektur DTN yang telah dipasang pada perangkat raspberry pi.
4. Data yang diterima akan dicek terlebih dahulu CRC nya, apabila terdapat kesalahan data maka akan dilakukan pengiriman ulang data hingga data yang diterima sesuai. Jika data sesuai, maka data akan disimpan dan disusun menjadi bentuk gambar utuh kembali.

Adapun flowchart dari sistem yang akan dibuat ditunjukkan oleh Gambar 2.



Gambar. 2. Flowchart Sistem

B. Realisasi

Setelah melakukan perancangan blok diagram dan *flowchart* dari sistem yang akan dibuat. Selanjutnya, dilakukan realisasi perangkat keras dan perangkat lunak berdasarkan perancangan yang telah dibuat.

1. Realisasi Perangkat Keras

Pada realisasi perangkat keras sistem yang digunakan yaitu perangkat raspberry pi 3 yang digunakan sebagai pengolah data bagian *node* pengirim dan *node* penerima. Untuk menggunakan raspberry 3 ini membutuhkan catu daya sebesar 5v 2.5A. Berikut gambaran pada Gambar 3. bagaimana realisasi perangkat keras yang digunakan dalam penelitian dengan menggunakan raspberry pi 3 dan sumber catu daya berupa *power bank*



Gambar. 3. Raspberry pi 3

Raspberry yang digunakan dalam penelitian ini adalah raspberry pi 3. Pemilihan raspberry pi 3 ini dikarenakan pada raspberry model ini sudah mendukung teknologi *wireless*.

2. Realisasi Perangkat Lunak

2.1 Implementasi IBR-DTN

Untuk menggunakan arsitektur DTN perlu dilakukan instalasi aplikasi IBR-DTN pada kedua perangkat raspberry pi. Proses instalasi IBR-DTN pada kedua raspberry pi dilakukan pada terminal perangkat dengan mengetikkan *command* sesuai dengan panduan IBR-DTN pada website github [10]. Setelah proses install IBR-DTN berhasil dilakukan, langkah selanjutnya adalah melakukan beberapa konfigurasi pada IBR-DTN yang digunakan. Konfigurasi yang paling penting dalam penelitian adalah konfigurasi pada *convergence layer* dengan *setting* protokol yang digunakan adalah protokol UDP.

TABEL I
SETTING UDP PADA KONFIGURASI IBRDTN

Konfigurasi	Keterangan
<code>net_lan1_type = udp</code>	#menggunakan protocol UDP
<code>Net_lan1_interface = wlan0</code>	#memasang pada interface eth0
<code>Net_lan1_port = 4556</code>	#menggunakan port 4556 (<i>default</i>)

Tabel I menunjukkan bagaimana konfigurasi IBR-DTN dengan menggunakan protokol UDP dilakukan pada folder `ibrdtm-repo/ibrdtm/daemon/etc/ibrdtnd.conf`. Setelah melakukan konfigurasi IBR-DTN, maka diperlukan aktivasi untuk mengaktifkan layanan DTN pada kedua raspberry pi dengan mengetikkan *command* `dtnd -c ibrdtm-repo/ibrdtm/daemon/etc/ibrdtnd.conf` pada terminal raspberry pi. Apabila instalasi aplikasi IBR-DTN telah selesai dan dapat digunakan maka tampilan pada terminal raspberry menunjukkan nama *node* DTN dan protokol yang digunakan adalah UDP *Convergence layer* seperti yang ditunjukkan pada Gambar 4

```

pi@node1:~
File Edit Tabs Help
ing 4 processes.
Mon Jun 21 07:24:42 2021 INFO BundleCore: Local node name: dtn://node1.dtn
Mon Jun 21 07:24:42 2021 INFO BundleCore: Block size limited to 1300000000 bytes
Mon Jun 21 07:24:42 2021 INFO BundleCore: Block size of foreign bundles limited
to 500000000 bytes
Mon Jun 21 07:24:42 2021 INFO BundleCore: Lifetime limited to 604800 seconds
Mon Jun 21 07:24:42 2021 INFO BundleCore: Limit the number of bundles in transit
to 5
Mon Jun 21 07:24:42 2021 INFO BundleCore: Forwarding of bundles enabled.
Mon Jun 21 07:24:42 2021 INFO BundleCore: Non-singleton bundles are accepted.
Mon Jun 21 07:24:42 2021 INFO NativeDaemon: using persistent bundle-sets
Mon Jun 21 07:24:42 2021 INFO NativeDaemon: using simple bundle storage in /home
/pi/bundles
Mon Jun 21 07:24:42 2021 INFO NativeDaemon: using BLOB path: /tmp
Mon Jun 21 07:24:42 2021 INFO SimpleBundleStorage: 0 Bundles restored.
Mon Jun 21 07:24:42 2021 INFO NativeDaemon: API initialized using tcp socket: <l
oopback>:4550
Mon Jun 21 07:24:42 2021 INFO NativeDaemon: UDP ConvergenceLayer added on wlan0:
4556
Mon Jun 21 07:24:42 2021 INFO IPNDAgent: listen to [224.0.0.142]:4551
Mon Jun 21 07:24:42 2021 INFO IPNDAgent: listen to [ff02::142]:4551
Mon Jun 21 07:24:42 2021 INFO IPNDAgent: advertise on interface wlan0
Mon Jun 21 07:24:42 2021 INFO NativeDaemon: Using prophet routing extensions wit
h GRTR as forwarding strategy.
    
```

Gambar. 4. Aplikasi IBR-DTN yang telah aktif pada raspberry pi

2.2 Implementasi Program

Pada tahap realisasi perangkat lunak, alur sistem yang telah dijelaskan sebelumnya pada Gambar 1. akan dibuat dalam bentuk *script* program dengan menggunakan bahasa pemrograman python. Pada bagian pengirim dibuat program untuk mengkonversi gambar menjadi deretan biner, yang selanjutnya data biner tersebut akan dibagi menjadi potongan data lebih kecil per 64000 bits. Kemudian, data akan diproses penambahan bit CRC yang nantinya sebagai bit untuk pengecekan *error* pengiriman data selama transmisi berlangsung. Pada bagian penerima, terdapat program untuk menjalankan perintah menerima data dengan aplikasi IBR-DTN. Selanjutnya data tersebut akan diproses pengecekan bit CRC untuk mengecek apakah terdapat data yang *error* atau tidak selama transmisi data berlangsung. Apabila terdapat pengiriman data yang *error*, maka *node* penerima akan meminta pengiriman ulang data tersebut. Begitupun pada *node* pengirim akan melakukan pengiriman ulang pada

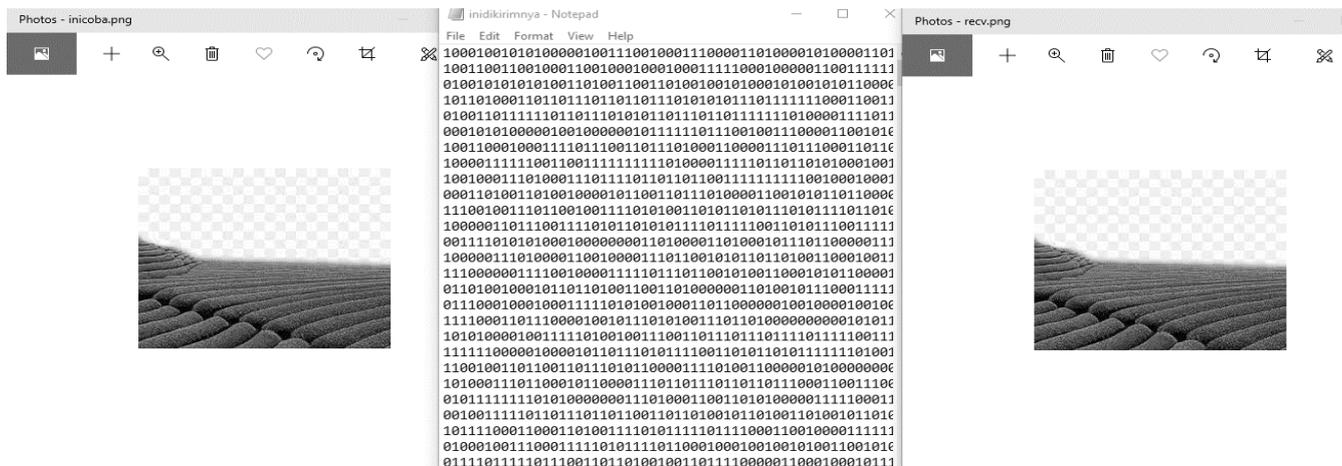
data yang terdeteksi error. Apabila seluruh potongan data yang dikirim telah diterima, maka data yang diterima dalam bentuk deretan biner tersebut akan dikonversi menjadi gambar utuh seperti pada sisi penerima.

III. HASIL DAN PEMBAHASAN

Untuk mengetahui fungsionalitas dari sistem yang telah dibuat dilakukan pengujian untuk mengetahui kinerja sistem yang telah direalisasikan apakah sesuai dengan perancangan yang telah dilakukan. Berikut pengujian dan hasil yang diperoleh.

1. Pengujian program konversi gambar

Pada sistem terdapat program untuk mengkonversi atau membaca gambar menjadi dalam bentuk deretan biner pada sisi pengirim. Sedangkan pada sisi penerima, terdapat program untuk mengkonversi deretan biner menjadi gambar kembali agar data yang diterima sesuai yaitu data berupa gambar dengan ekstensi PNG. Dan berikut hasil pengujian program ditunjukkan pada Gambar 5.



Gambar 5. Program konversi gambar

Berdasarkan Gambar 5. menunjukkan bahwa program pembacaan gambar menjadi deretan biner sudah berhasil dibuat dan hasilnya sesuai dengan yang dirancang. Pada awalnya gambar dengan nama file inicoba.png dikonversi menjadi deretan biner, kemudian deretan biner tersebut berhasil dikonversi kembali menjadi data gambar dengan nama file recv.png.

2. Pengujian program CRC

Setelah melakukan pengujian program konversi gambar selanjutnya adalah melakukan pengujian pada program proses penambahan bit CRC. Pada program yang dibuat ini Panjang bit CRC yang digunakan dapat di custom sesuai keinginan dan pada penelitian ini bit CRC yang digunakan adalah penambahan 8 bit CRC diakhir setiap potongan data gambar.

```
E:\SEM.6\BISMILLAH TA\Bismillah TA RIZKI\CRC> kirim.py
Data yang akan dikirim = 11011001
11011001
pembagi = 1101
data yang ditransmisikan = 11011001011
```

Gambar 6. Hasil penambahan bit CRC pada pengirim

Pada Gambar 6. menunjukkan hasil dari program CRC yang telah dibuat, data yang akan dikirim adalah 11011001 diproses CRC terlebih dahulu dengan bit pembanding 1101 diperoleh hasil total data yang dikirim adalah data yang dikirim + bit CRC. Maka, data yang akan ditransmisikan adalah 11011001011. Selanjutnya dilakukan pengujian pada bagian penerima, dengan hasil uji sebagai berikut.

```
E:\SEM.6\BISMILLAH TA\Bismillah TA RIZKI\CRC> terima.py
Data yang diterima = 10110011001
10110011001
pembagi = 1101
sisa bagi = 110
Terdapat kesalahan pengiriman data
```

Gambar 7. Hasil pengecekan bit CRC pada penerima

Gambar 7 menunjukkan program pada penerima sudah dapat dijalankan dan sesuai dengan perancangan yang dilakukan yaitu dapat mendeteksi kesalahan data selama transmisi berlangsung. Terlihat pada tampilannya, program CRC berhasil

mendeteksi adanya kesalahan pengiriman data dengan sisa bagi crc adalah 110, yang menunjukkan bahwa terdapat *error* pada pengiriman data. Apabila tidak ada *error*, maka seharusnya hasil sisa bagi adalah 0. Dari pengujian ini menunjukkan program CRC yang telah dibuat pada bagian pengirim dan penerima sudah sesuai fungsionalitasnya dalam mendeteksi kesalahan yang terjadi.

3. Pengujian Fungsionalitas Sistem

Setelah melakukan pengujian pada program konversi gambar dan program CRC, selanjutnya dilakukan pengujian pada keseluruhan sistem yang telah dibuat. Pengujian dilakukan dengan mengirimkan gambar yang berukuran 62,5 KB dan gambar berukuran 350 KB pada kondisi sinyal penuh dan sinyal berselang dengan menggunakan arsitektur DTN.

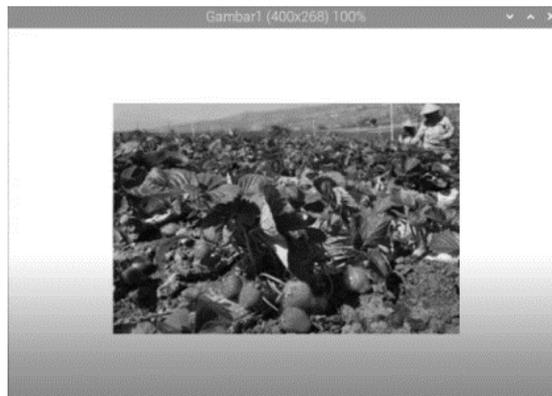
1. Pengiriman gambar 62,5 KB

```

pi@node1: ~/TugasAkhir/FIX
File Edit Tabs Help
Wait for incoming bundle...
Bundle received (1).
done.
DATA KE- 7 BERHASIL DIKIRIM
Total Data Kirim 64000
8981
DATA +CRC = 64008
Transfer file "kiritemp.bin" to dtn://node2.dtn/penerima
Wait for incoming bundle...
Bundle received (1).
done.
DATA KE- 8 BERHASIL DIKIRIM
9816
1228
DATA +CRC :
9824
Transfer file "kiritemp.bin" to dtn://node2.dtn/penerima
Wait for incoming bundle...
Bundle received (1).
done.
DATA KE- 9 BERHASIL DIKIRIM
Seluruh Data Gambar Berhasil Dikirim
Waktu Kirim 29.33746933370728
pi@node1:~/TugasAkhir/FIX $
    
```

Gambar. 8. Tampilan proses pengiriman data gambar

Gambar 8. menunjukkan bagaimana proses yang terjadi selama pengiriman data gambar berukuran 62,5 KB. Gambar dibagi menjadi potongan data yang lebih kecil per 64000 bits data sehingga potongan data pada gambar dengan ukuran 62,5 KB sebanyak 9 potongan data gambar. Pada Gambar 8 tersebut terlihat pada proses pengiriman data yang dikirim dilengkapi dengan CRC 8 bit. Pengiriman dilakukan secara berulang sebanyak 9 data, diperoleh hasil pengiriman berhasil dilakukan. Seluruh data gambar berhasil dikirim, dengan hasil gambar yang diterima dan ditunjukkan oleh Gambar 9.



Gambar. 9. Gambar yang diterima

2. Pengiriman gambar 350 KB

Pengujian fungsionalitas sistem selanjutnya adalah melakukan pengiriman gambar dengan ukuran 350 KB. Dari pengujian yang dilakukan yaitu menggunakan gambar dengan ukuran 350KB diperoleh hasil gambar berhasil terkirim dan diterima dengan utuh walaupun batasan pengiriman data pada protokol UDP adalah 64KB. Hal tersebut terjadi karena pada sistem gambar telah dibagi menjadi ukuran yang lebih kecil dan disertai pengecekan *error* data dan dilakukan pengiriman ulang untuk data yang terdeteksi *error*.

```

pi@node1: ~/TugasAkhir/FIX
File Edit Tabs Help
done.
DATA KE- 46 BERHASIL DIKIRIM
14992
1875
DATA +CRC :
15000
Transfer file "kiritemp.bin" to dtn://node2.dtn/penerima
Wait for incoming bundle...
Bundle received (1).
done.
HARUS KIRIM ULANG
Dilakukan Pengiriman Ulang Data Ke- 47
14992
1875
DATA +CRC :
15000
Transfer file "kiritemp.bin" to dtn://node2.dtn/penerima
Wait for incoming bundle...
Bundle received (1).
done.
DATA KE- 47 BERHASIL DIKIRIM
Seluruh Data Gambar Berhasil Dikirim
Waktu Kirim 169.22747611999512
pi@node1:~/TugasAkhir/FIX $
    
```

Gambar. 10. Tampilan proses pengiriman gambar 350KB

Dari gambar 10 terlihat terdapat data gambar yang *error* selama transmisi berlangsung yaitu pada data ke-47 sehingga terjadi proses pengiriman ulang pada potongan data tersebut. Dari pengujian tersebut menunjukkan gambar dengan ukuran 350 KB tetap berhasil dikirim dengan protokol UDP pada arsitektur DTN. Pada program CRC yang dibuat dapat mendeteksi *error* yang terjadi selama transmisi berlangsung dan adanya proses pengiriman ulang pada data yang terdeteksi *error*. Adapun hasil gambar yang diterima ditunjukkan pada gambar 11.



Gambar. 11. Hasil gambar yang diterima *node 2*

3. Kondisi Putus Koneksi

Pengujian selanjutnya adalah pengujian pengiriman gambar pada saat koneksi putus untuk mengecek fungsionalitas dari arsitektur DTN yang seharusnya tetap dapat mengirimkan gambar walaupun kondisi koneksi terputus. Pengujian ini dilakukan dengan cara memutuskan koneksi wifi pada *node* pengirim selama 1 menit. Hasil menunjukkan dengan menggunakan arsitektur DTN walaupun terjadi putus koneksi selama transmisi data berlangsung maka data akan tetap terkirim karena pada DTN menggunakan prinsip *store-carry-forward* sehingga apabila salah satu *node* terputus dari jaringan maka data tersebut akan disimpan pada *storage bundle* raspberry pi sampai kembali terhubung dengan jaringan. Berikut ini gambar yang diterima dengan kondisi putus koneksi yang terjadi pada saat proses transmisi berlangsung selama 1 menit pada Gambar 12.



Gambar. 12. Gambar diterima kondisi delay

IV. KESIMPULAN

Berdasarkan program yang telah dibuat serta pengujian fungsionalitas sistem dapat disimpulkan bahwa sistem yang telah dibuat sudah sesuai dengan perancangan yang telah dibuat. Program yang dibuat dapat mengatasi kekurangan UDP yang rentan kehilangan data selama transmisi berlangsung, program dapat mengirimkan gambar dengan ukuran lebih dari 64KB dan pengiriman data di daerah rentan putus koneksi dapat teratasi dengan adanya arsitektur DTN. Sistem yang diuji adalah pengiriman *bundle* dengan ukuran 64000 bit data. Kedepannya pengiriman dapat dilakukan dengan ukuran *bundle* 512000 bit pada *sistem processor* yang lebih cepat agar pengiriman data lebih cepat dibanding per 64000 bit data. Diharapkan sistem ini dapat digunakan pada teknologi IoT khususnya pada daerah-daerah yang rentan terjadi putus koneksi.

DAFTAR PUSTAKA

- [1] S. A, W. P. Putra, E. Ismantohadi, Supardi, and M. Qomarrudin, "SISTEM MONITORING TANAMAN HORTIKULTURA PERTANIAN DI KABUPATEN INDRAMAYU BERBASIS INTERNET OF THINGS," *J. Teknol. dan Inf. UNIKOM*, vol. 9, no. 1, pp. 45–54, 2019.
- [2] S. Siswanti, "Pengembangan Sistem Aplikasi Pengiriman Data Daerah Terpencil Berbasis Delay Tolerant Network," *J. Generic*, vol. 8, no. 2, pp. 221–271, 2013.
- [3] H. D. Laksana, A. Bhawiyuga, and K. Amron, "Rancang Bangun Perangkat Mobile Berbasis Delay Tolerant Network Sebagai Perantara Pengiriman Data Sensor Dari Lapangan Ke Pusat Data," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 9, pp. 3058–3064, 2018.
- [4] G. R Megiyanto, "Sistem Pengumpulan Data Menggunakan Arsitektur Delay Tolerant Network Pada Sistem Pengawasan Konservasi Badak," Institut Teknologi Bandung, 2017.
- [5] B. Aulia Rahmataufany, R. Primananda, and A. Suharsono, "Analisis Pengiriman Data Di Daerah Terpencil Menggunakan Dua Node Bergerak Berbasis Delay Tolerant Network (DTN) Dengan Algoritme Flooding," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 9, pp. 2882–2891, 2018.
- [6] K. Fall, "A delay-tolerant network architecture for challenged internets," p. 27, 2003, doi: 10.1145/863955.863960.
- [7] G. R Megiyanto and I. Fadillah, "Sistem Komunikasi Jaringan Wireless Menggunakan Raspberry Pi Dengan Arsitektur Delay Tolerant Network," in *Industrial Research Workshop and National Seminar*, 2020, pp. 429–434.
- [8] M. Julius and F. Sirait, "Pendeteksian Bit Error Dalam Transmisi Data Dengan Menerapkan Cyclic Redudancy Check," *MEANS (Media Inf. Anal. dan Sist.*, vol. 3, no. 2, pp. 190–193, 2018.
- [9] S. Wahyuni, A. Lubis, S. Batubara, and I. K. Siregar, "Implementasi Algoritma CRC 32 Dalam Mengidentifikasi Keaslian File," in *Seminar Nasional Royal (SENAR)*, 2018, pp. 1–6.
- [10] S. Schildt, J. Morgenroth, W.-B. Pottner, and L. Wolf, "Electronic Communications of the EASST Workshops der wissenschaftlichen Konferenz Kommunikation in verteilten Systemen 2011 (WowKiVS 2011) Towards NFC-Aware Process Execution for Dynamic Environments," *Communications*, vol. 37, 2011.