

Implementasi Metode Kompresi Gambar dari *Node* Sensor ke *Node* Server untuk Meningkatkan Efisiensi Waktu Pengiriman pada Jaringan 6LoWPAN

Mohammad Harits¹, Adhitya Bhawiyuga², Reza Andria Sinegar³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹mohammadharits29@gmail.com, ²bhawiyuga@ub.ac.id, ³reza.jalin@ub.ac.id

Abstrak

Wireless Sensor Network telah mampu diterapkan pada bidang pertanian untuk memantau kondisi lahan pertanian yaitu mengenai kelembapan, pH dan suhu dari tanah. Namun proses monitoring ini tidak dapat mengambil gambar pada lokasi dikarenakan WSN tidak mampu untuk mengirim data yang berukuran besar. Untuk mengatasi masalah tersebut, dapat digunakan modul komunikasi yang berkapasitas besar berfungsi supaya data yang dikirimkan lebih banyak dan metode kompresi data berfungsi mengurangi ukuran data sebelum dikirimkan. Modul komunikasi yang digunakan dalam penelitian adalah 6LoWPAN. 6LoWPAN dipilih karena memiliki keunggulan pada kapasitas transmisi, jangkauan dan memiliki konsumsi daya yang rendah. Kemudian ada dua metode kompresi yang digunakan adalah algoritma DCT yang digunakan untuk mengurangi redundansi interpixel pada gambar dan RLE untuk memadatkan hasil dari algoritma DCT sehingga ukurannya lebih kecil lagi. Hasil yang diperoleh setelah melakukan pengujian adalah penerapan metode kompresi dapat menghemat ukuran gambar sebesar $\pm 24\%$ dari ukuran gambar semula dan mengurangi total paket yang dikirim sebesar $\pm 77\%$ dari paket semula. Performansi pengiriman menggunakan metode kompresi memiliki keunggulan dalam waktu pengiriman tetapi menurunkan tingkat kesuksesan dan meningkatkan rerata delay saat pengiriman data gambar. Namun penurunan yang terjadi masih dianggap wajar karena perbedaan hasil yang diperoleh hanya sedikit dibandingkan dengan pengiriman yang tidak menggunakan kompresi.

Kata kunci: WSN, 6LoWPAN, DCT, RLE, performa pengiriman.

Abstract

Wireless Sensor Network has been able to be applied in agriculture to monitor the condition of agricultural land, namely regarding the humidity, pH, and temperature of the soil. However, this monitoring process cannot take pictures on the location because Wireless Sensor Network has the disadvantage of not being able to send large datas. To solve this problem, you can use a large capacity communication module so that more data is sent and also use the compression method for data to reduce the size of data before being sent. The communication module used in this study is 6LoWPAN. 6LoWPAN was chosen because it has advantages in terms of transmission capacity, range and has a low power consumption. There are 2 compression methods used, namely the DCT algorithm which is used to reduce interpixel redundancy and RLE to condense the results of the DCT algorithm so it can be smaller. The results obtained after testing are that the application of the compression method can save image size by around 24% from the size of the original image and reduce the total package sent by about 77% of the original packet. Delivery performance using the compression method has the advantage of sending time but decreases the success rate and increases average delay when sending image data.

Keywords: Monitoring system, Internet of Things (IoT), WebSocket, AMQP

1. PENDAHULUAN

Wireless Sensor Network merupakan kumpulan *node* berupa sensor yang digunakan untuk mengambil parameter ukur yang diletakkan pada titik-titik tertentu,

kemudian dikirimkan ke *node* pusat atau sebuah server untuk diolah (Munir, 2018). Wireless Sensor Network dapat diterapkan pada bidang pertanian untuk memantau kondisi lahan pertanian. Kondisi yang

dipantau yaitu mengenai kelembapan, pH dan suhu dari tanah. Sensor yang digunakan adalah sensor kelembapan, suhu dan pH tanah. Namun, proses monitoring tersebut hanya memantau keadaan tanah tanpa mengambil gambar kondisi lahan pertanian tersebut. Apabila mengambil gambar kondisi pertanian dengan menggunakan Wireless Sensor Network maka terdapat dua hal yang perlu dipertimbangkan yaitu memilih modul komunikasi sesuai kebutuhan dan mengurangi ukuran data gambar yang akan dikirim. Hal tersebut disebabkan karena Wireless Sensor Network tidak dapat melakukan pengiriman data berukuran besar.

Menurut Aji Binariyanto (2018), 6LoWPAN memiliki keunggulan pada konsumsi daya yang rendah dan berkapasitas transmisi yaitu 250 kbps, 40 kbps dan 20 kbps berturut turut sesuai dengan layer fisiknya (2,4GHz, 915 MHz, dan 868Mhz). 6LoWPAN juga dapat mengirim data berukuran besar dengan cepat dan mampu bekerja pada jarak < 100m. Berdasarkan keunggulan yang terdapat pada 6LoWPAN, maka 6LoWPAN digunakan dalam penelitian karena telah memenuhi kebutuhan sistem pada sisi kapasitas transmisi dan jarak pengiriman data.

Solusi berikutnya adalah menggunakan metode kompresi untuk mengurangi ukuran data gambar sebelum dikirim. Menurut Suryaningrat (2018), algoritma DCT merupakan suatu transformasi one-to-one mapping dari suatu array yang terdiri dari nilai pixel menjadi komponen-komponen yang terbagi berdasarkan frekuensinya. Algoritma DCT digunakan untuk mengurangi redudansi nilai setiap pixel dari gambar masukan, sehingga gambar hasil kompresi akan berukuran lebih kecil daripada sebelumnya namun dari sudut pandang mata manusia masih toleran. Setelah menerapkan algoritma DCT nilai setiap pixel pada gambar pasti banyak memiliki nilai yang sama, sehingga hal ini memungkinkan untuk menggabungkan algoritma lainnya.

Menurut Fahmi (2016), Algoritma RLE menggunakan pendekatan ruang, sehingga sangat cocok digunakan untuk memampatkan citra (gambar) yang memiliki kelompok-kelompok piksel berderajat keabuan atau nilai pixel yang sama. Apabila

semakin banyak derajat keabuan yang sama, maka hasil kompresi akan semakin baik. Berdasarkan penjelasan dari algoritma RLE, algoritma RLE dapat digunakan setelah algoritma DCT untuk lebih mengurangi ukuran data gambar yang akan dikirimkan melalui jaringan 6LoWPAN. Jadi semakin kecil data gambar yang akan dikirim, maka dapat meningkatkan efisiensi pada waktu yang pengiriman.

2. DASAR TEORI

2.1 Kompresi Gambar

Kompresi gambar adalah proses yang menyandikan gambar ke dalam kode sederhana tanpa ada kehilangan informasi (Varsha Bansal,dkk. 2014). Dengan adanya proses pengkodean ini, gambar yang awalnya memiliki jumlah data yang besar dapat diperkecil lagi sebelum data tersebut dikirim. Didalam proses image compression ada 2 tahap yang wajib dilakukan yaitu *encode* dan *decode*. *Encode* merupakan proses pengkodean gambar sebelum dikirim dan dilakukan pada sisi pengiriman data, yang bertujuan untuk mengurangi ukuran data sebelum dikirim. Sehingga data yang dikirim memiliki ukuran lebih kecil dan cepat terkirim ke tujuan. Sedangkan *decode* merupakan proses untuk merekonstruksi kembali data yang diterima dalam bentuk kode-kode menjadi gambar aslinya dan dilakukan pada sisi penerima data yang kemudian gambar tersebut ditampilkan pada layar monitor. Dengan adanya image compression data, kita dapat mengurangi konsumsi resource (sumber daya) dalam pengiriman data dan mengurangi waktu tunggu saat pengiriman data.

2.1.1 DCT (Discreate Cossinus Tranform)

DCT merupakan suatu transformasi one-to-one mapping dari suatu array yang terdiri dari nilai pixel menjadi komponen-komponen yang terbagi berdasarkan frekuensinya (Suryaningrat I Gusti Ngurah Jelantik, dkk. 2018). Algoritma DCT digunakan untuk mengurangi redudansi nilai setiap pixel dari gambar masukan, sehingga gambar hasil kompresi akan berukuran lebih kecil daripada sebelumnya namun dari sudut pandang mata manusia masih toleran. Rumus DCT terdiri dari 2 bagian yaitu

a. Rumus kompresi

$$C(u, v) = \frac{1}{4} C(u) C(v) \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \cos \left| \frac{\pi(2x+1)u}{2N} \right| \cos \left| \frac{\pi(2y+1)v}{2M} \right|$$

b. Rumus dekomposisi

$$f(x, y) = \frac{2}{\sqrt{MN}} \sum_{v=0}^{M-1} a(u) a(v) C(u, v) \cos \left| \frac{\pi(2x+1)u}{2N} \right| \cos \left| \frac{\pi(2y+1)v}{2M} \right|$$

$$\text{Dimana } a(k) = \begin{cases} 1/\sqrt{2} & \text{untuk } k = 0 \\ 1 & \text{untuk } k \neq 0 \end{cases}$$

Keterangan:

- $C(u, v)$ adalah titik koordinat dari matriks yang telah mengalami transformasi DCT 2 dimensi.
- M dan N adalah banyak kolom dan baris. Apabila ukuran matriks adalah 8×8 , maka nilai M dan N adalah 8.
- $a(u)$ dan $a(v)$ adalah himpunan hasil yang nilainya ditentukan dari nilai koefisien u dan v .
- $f(x, y)$ adalah nilai pixel dari matriks pada titik (x, y) .

2.1.2 RLE (Run-Length Encoding)

RLE merupakan salah satu dari algoritma kompresi yang menggunakan pendekatan ruang. Algoritma ini sangat cocok digunakan untuk memampatkan citra (gambar) yang memiliki kelompok-kelompok piksel berderajat keabuan yang sama (Fahmi Khairul, 2016). Cara kerja dari algoritma ini yaitu gambar akan dirubah terlebih dahulu menjadi bentuk *pixels*, kemudian pixel-pixel tersebut dirubah menjadi 1 *row* (baris). Perubahan ini bertujuan untuk mempermudah Algoritma RLE mencari nilai yang sama. Setelah itu dilakukan proses encode dengan aturan adalah saat menemukan data yang sama maka akan dilakukan kompresi menjadi 2 variabel yaitu nilai data tersebut dan berapa banyak data tersebut diulangi.

2.2 6LoWPAN

Penelitian yang dilakukan oleh (Aji Binariyanto, 2018) yang berjudul “Penerapan Komunikasi Berbasis 6LoWPAN(802.15.4) Antara Node Sensor dengan IoT Middleware”. Penelitian ini menggunakan modul komunikasi 6LoWPAN untuk berkomunikasi antar *node*. Dalam penelitian ini membuktikan bahwa 6LoWPAN memiliki keunggulan pada jangkauan, keamanan, skalabilitas, dan 6LoWPAN juga didesain untuk perangkat berdaya rendah. Berdasarkan keunggulan yang terdapat pada 6LoWPAN, maka 6LoWPAN digunakan dalam penelitian karena telah memenuhi kebutuhan sistem pada sisi kapasitas transmisi dan jarak pengiriman data.

3. PERANCANGAN DAN IMPLEMENTASI

3.1. Gambaran Umum Sistem

Pada penelitian ini terdapat 3 jenis perangkat keras yang saling terhubung yaitu Kamera, Raspberry pi 2 model B dan modul MRF24J40MA. Kamera akan dihubungkan dengan *node* sensor dan digunakan sebagai sensor untuk menangkap gambar. Modul MRF24J40MA dihubungkan pada *node* sensor dan *node* server dan digunakan sebagai media komunikasi antar *node*. Raspberry pi pada *node* sensor digunakan sebagai tempat mengolah data yang diambil oleh sensor dan melakukan proses *encode* sebelum data akan dikirim. Sedangkan raspberry pi pada *node* server melakukan proses *decode* setelah menerima data kiriman dari *node* sensor. Berikut rancangan sistem dapat dilihat pada gambar 1.



Gambar 1 Gambaran Umum Sistem

Berikut penjelasan pada gambar 3.2 :

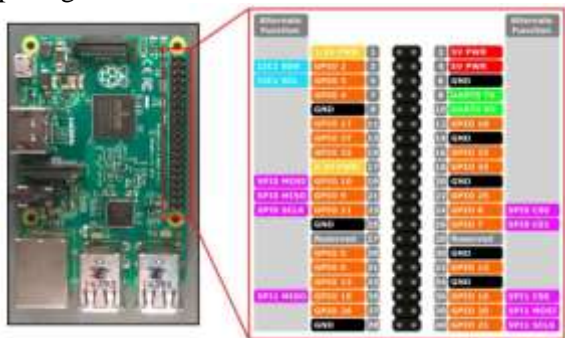
1. *Node* sensor mentrigger kamera untuk menangkap gambar.
2. *Node* sensor akan melakukan proses kompresi dan proses *encode* sebelum gambar akan dikirim. Data akan dikirim melalui protocol TCP dan dilewatkan melalui jaringan 6LoWPAN.
3. *Node* Server bertindak sebagai *node* yang menerima data kiriman dari *node* sensor. Data yang diterima akan melalui proses *decode* dan dekomposisi hingga data kiriman dapat diubah kembali menjadi gambar.

3.2 Perancangan Perangkat

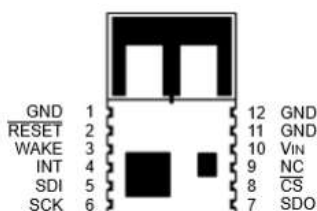
3.2.1 Desain Integrasi Raspberry Pi dengan Modul MRF24J40MA

Menjelaskan tentang cara menghubungkan 2 perangkat yaitu modul MRF24J40MA dan Raspberry pi 2 modul B. Pada raspberry pi 2 model B terdapat dua SPI (*Serial Peripheral Interface*) yaitu SPI0 dan SPI1. SPI0 akan digunakan sebagai cara untuk berkomunikasi antara modul MRF24J40MA dengan raspberry. Raspberry akan berfungsi sebagai *microcontroller*, yang akan mengkompresi gambar yang diambil oleh sensor dan melakukan proses *encode* sebelum data

dikirimkan. Sedangkan modul MRF24J40MA bertugas untuk menghubungkan setiap *node* mampu berkomunikasi dengan memanfaatkan teknologi 6LoWPAN melalui jaringan WPAN. Penggambaran pin layout Raspberry dilihatkan pada Gambar 2 dan pin layout MRF24J40MA pada gambar 3.



Gambar 2. Pin layout raspberry



Gambar 3. Pin layout MRF24J40MA

Untuk membuat raspberry pi dan modul MRF24J40MA saling terhubung, dimana perlu adanya jalur komunikasi. Jalur komunikasi ini dengan menghubungkan antar pin layout dan dapat dibuat melalui kabel jumper atau membuat jalur pada PCB. Rancangan jalurnya dapat dilihat pada table 3.1

Table 1. Rancangan Jalur pin Raspberry Pi dan Modul MRF24J40MA

Type	Raspberry Pi	Modul MRF24j40MA
GND	20	1/11/12
RESET	-	2
WAKE	-	3
INT	22	4
SDI	19	5
SCK	23	6
SDO	21	7
CS	24	8
NC	-	9
VIN	17	10

3.2.2 Desain Alur Kompresi pada *Node* Sensor

Sub bab ini akan membahas tentang proses kompresi yang ada di *node* sensor. Proses kompresi pada *node* sensor ada 6 tahap yaitu :

1. Mentriger kamera untuk menangkap gambar kemudian menyimpannya di *node* sensor.
2. Mengubah warna gambar hasil tangkapan kamera menjadi format YCbCr.
3. Proses Subsampling warna (Downsampling).

Proses downsampling dalam penelitian ini menggunakan skala 4:2:2. Maksud dari skala tersebut adalah ukuran panjang dan lebar dari nilai Y tidak berubah, sedangkan ukuran panjang dari Cb dan Cr menjadi setengahnya dari ukuran aslinya dan nilai lebarnya tidak berubah.

4. Proses DCT dan Quantization pada blok-blok gambar.

Setelah proses downsampling, gambar akan diubah menjadi blok-blok dan setiap blok terdiri dari 8x8 pixel. Kemudian setiap blok array akan dimasukkan kedalam algoitma DCT. Kemudian hasilnya akan dilakukan proses quantization. Proses quantization dilakukan dengan cara membagi hasil dari proses DCT dengan nilai quantization dari masing-masing YCbCr. Kemudian hasil dari quantization akan disimpan dalam array baru

5. Proses Zig-zag

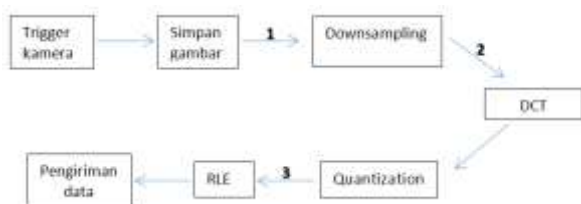
Proses zig-zag ini bertujuan untuk mengubah array 2D menjadi array 1D.

6. Proses RLE

Setelah mejadi array 1D, maka proses selanjutnya adalah memasukkan setiap nilai array dari YCbCr untuk diproses kedalam algoritma RLE.

7. Pengiriman data

Dalam proses pengiriman data, data-data penting seperti ukuran gambar asli, ukuran data sewaktu downsampilng dan hasil RLE akan di pack menggunakan struct. Kemudian hasil struct kan dikirimkan ke *node* server.



Gambar 4. Alur proses kompresi

Keterangan Gambar 4 :

1. Proses merubah gambar yang awalnya bertipe RGB menjadi YCbCr
2. Membuat blok-blok pixel yang terdiri dari 8x8 pixel.
3. Proses zig-zag untuk mengubah array 2D menjadi array 1D

3.2.3 Desain Alur Dekompresi pada Node Server

Sub bab ini akan membahas tentang alur dekompresi pada *node* server. Proses dekompresi pada *node* server akan dilakukan dalam 5 tahap yaitu :

1. Proses unpack paket yang kiriman dari *node* sensor
2. Proses invers RLE

Proses dilakukan untuk mengembalikan nilai sebelum diproses kedalam algoritma RLE.

3. Proses invers zigzag

Proses ini dilakukan untuk mengubah array 1D menjadi 2D sesuai dengan ukuran gambar yang dikirimkan oleh *node* sensor.

4. Proses invers DCT dan Quantization

Setelah proses invers zigzag, gambar akan diubah menjadi blok-blok dan setiap blok terdiri dari 8x8 pixel. Kemudian setiap blok array akan dimasukkan kedalam algoritma invers DCT. Kemudian hasilnya akan dilakukan proses quantization. Proses quantization dilakukan dengan cara mengalikan hasil dari proses DCT dengan nilai quantization dari masing-masing YCbCr. Kemudian hasil dari quantization akan disimpan dalam array baru.

5. Penyimpanan gambar di server

Sebelum disimpan di server, array yang menyimpan gambar bertipe YCbCr harus dikembalikan menjadi RGB dengan cara

ditambahkan dengan angka 128. Setelah selesai

dirubah, array tersebut dapat disimpan menjadi gambar di server.



Gambar 5. Alur proses dekompresi

Keterangan gambar 5 :

1. Proses Invers zigzag untuk membuat array 1D menjadi 2D
2. Proses merubah gambar bertipe YCbCr menjadi RGB

4. HASIL PENGUJIAN DAN ANALISA

4.1. Pengujian Kompresi Gambar

Pengujian ini bertujuan untuk mengetahui rasio kompresi yang dihasilkan. Cara pengujiannya adalah membandingkan ukuran gambar yang ada di *node* sensor dengan gambar yang diterima pada *node* server dan membandingkan total paket yang dikirimkan dengan kompresi dan tanpa kompresi. Hasilnya dapat dilihat pada tabel 5.1.

Table 2. Tabel Pengujian Kompresi Gambar

Ukuran frame	Total Paket dikirim		Ukuran Gambar (KB)		Presentase kompresi	
	Kompresi	Tanpa Kompresi	Kompresi	Tanpa Kompresi	Total Paket Dikirim	Ukuran Gambar
240 x 160	150	478	15.5	20.4	68.62	24.02
320 x 240	243	941	25.4	34.0	74.18	25.29
480 x 400	491	2332	52.1	68.9	78.95	24.38
640 x 480	707	3780	77.2	103	81.30	25.05
720 x 640	896	5750	95.6	125	84.42	23.52

Hasil yang diperoleh setelah melakukan pengujian adalah penerapan metode kompresi dapat menghemat ukuran gambar sebesar $\pm 24\%$ dari ukuran gambar semula dan mengurangi total paket yang dikirim sebesar $\pm 77\%$ dari paket semula. Jadi penerapan kompresi dapat menghemat total paket pengiriman dan memperkecil ukuran gambar.

4.2 Pengujian Performa Pengiriman

Pengujian performa pengiriman bertujuan untuk membandingkan pengiriman

gambar menggunakan kompresi dengan yang tidak menggunakan kompresi. Data yang digunakan untuk proses pengujian ini adalah 5 buah gambar dengan ukuran frame yang berbeda. Ukuran tersebut adalah 240x160, 320x240, 480x400, 640x480 dan 720x640. Kemudian akan lakukan pengiriman sebanyak 20 kali ada setiap ukuran tersebut pada jarak ± 50 meter, sehingga hasil pengujian yang didapatkan menjadi lebih baik. Cara pengujiannya akan dibagi menjadi 3 bagian yaitu waktu pengiriman, tingkat keberhasilan paket yang diterima dan rerata delay paket yang dihasilkan saat pengiriman.

4.2.1 Waktu Pengiriman

Pengujian ini bertujuan untuk membandingkan waktu yang diperlukan untuk paket telah terkirim antara paket yang telah dikompresi dengan paket tanpa kompresi. Hasil dari pengujian tersebut dapat dilihat pada tabel 3.

Table 3. Hasil Pengujian Waktu Pengiriman

Ukuran frame	Waktu (detik)		Waktu yang dihemat
	Kompresi	Tanpa Kompresi	
240 x 160	18.38247	57.4067357	67.98%
320 x 240	29.7006	111.785026	73.43%
480 x 400	60.99466	276.380274	77.93%
640 x 480	97.55424	530.053195	81.60%
720 x 640	142.787	745.184671	80.84%

Hasil yang diperoleh dari pengujian menjelaskan bahwa waktu pengiriman menggunakan kompresi lebih cepat daripada tidak menggunakan kompresi. Penghematan waktu yang diperoleh lebih dari 67% pada setiap ukuran frame yang berbeda.

4.2.2 Tingkat keberhasilan paket yang diterima

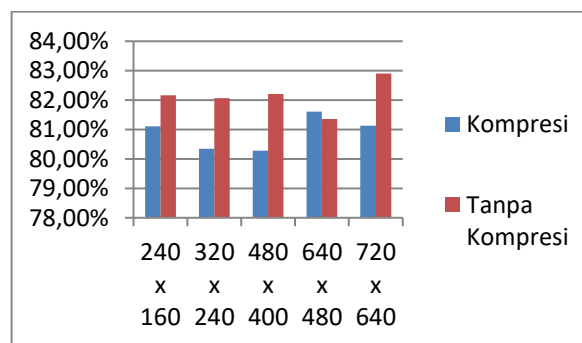
Pengujian ini bertujuan untuk melihat tingkat kesuksesan pengiriman paket dari *node* sensor ke *node* server menggunakan jaringan 6LoWPAN. Rumus yang digunakan untuk mengukur tingkat kesuksesan pengiriman paket adalah total paket diterima ada *node* server dibagi dengan total paket dikirim dari *node*

sensor dikali dengan 100%. Total paket diterima diperoleh dari total paket yang dikirim dikurangi dengan total paket retransmisi.

$$\text{Persentasi} = \frac{\text{Total paket diterima}}{\text{Total paket dikirim}} \times 100\%$$

Table 5. Pengukuran Tingkat Keberhasilan Paket yang Diterima

Ukuran frame	Tingkat keberhasilan	
	Kompresi	Tanpa Kompresi
240 x 160	81.11%	82.16%
320 x 240	80.35%	82.06%
480 x 400	80.28%	82.21%
640 x 480	81.61%	81.36%
720 x 640	81.13%	82.90%



Gambar 6. Grafik Pengukuran Tingkat Keberhasilan Paket yang Diterima

Berdasarkan hasil yang diperoleh bahwa rata-rata tingkat kesuksesan pengiriman paket yang menggunakan dan tidak menggunakan kompresi adalah $\pm 81\%$. Terjadi penurunan $\pm 1\%$ setelah menggunakan kompresi, karena perbedaan total pengiriman paket. Jadi semakin banyak paket yang dikirimkan maka peluang persentase tingkat kesuksesan pengiriman akan naik.

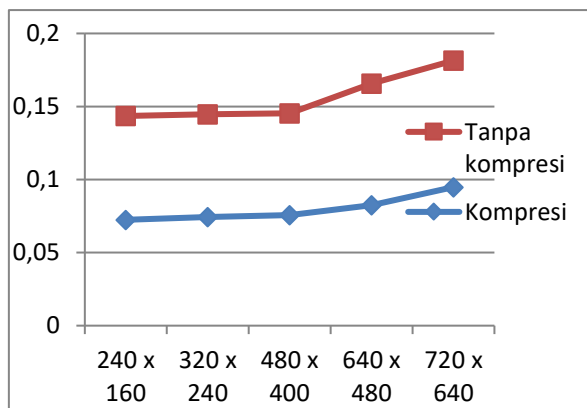
4.2.3 Rerata Delay

Pengujian ini bertujuan untuk mengukur delay yang terjadi pada setiap pengiriman setiap paket yang dikirim dari *node* sensor ke *node* server. Rumus yang digunakan adalah total delay yang dihasilkan dibagi dengan total paket. Hasilnya data dilihat ada tabel 6.

Table 6. Pengukuran Rerata Delay

No	Ukuran frame	Rerata Delay (detik)	
		Kompresi	Tanpa Kompresi
1	240 x 160	0.072433	0.071078
2	320 x 240	0.074343	0.070295
3	480 x 400	0.075589	0.069722

4	640 x 480	0.082365	0.083394
5	720 x 640	0.094813	0.086613



Gambar 7. Grafik Pengukuran Rerata Delay

Berdasarkan hasil yang diperoleh bahwa terdapat perbedaan delay antara pengiriman menggunakan kompresi dan tidak menggunakan kompresi. Perbedaan ini disebabkan karena total paket pengiriman yang tidak menggunakan kompresi lebih besar daripada yang menggunakan kompresi. Namun saat dilihat secara keseluruhan delay akan naik seiring dengan total paket yang dikirimkan dan jumlah retransmisi yang terjadi sewaktu pengiriman.

Setelah melakukan ketiga pengujian performa pengiriman dapat disimpulkan bahwa performa pengiriman yang menggunakan kompresi memiliki keunggulan dalam waktu pengiriman tetapi menurunkan tingkat kesuksesan dan meningkatkan rerata delay saat pengiriman data gambar. Namun penurunan yang terjadi masih dianggap wajar karena perbedaan hasil yang diperoleh hanya sedikit dibandingkan dengan pengiriman yang tidak menggunakan kompresi.

5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan penelitian yang telah dilakukan dapat disimpulkan bahwa :

1. Hasil yang diperoleh setelah melakukan pengujian adalah penerapan metode kompresi dapat menghemat ukuran gambar sebesar $\pm 24\%$ dari ukuran gambar semula dan mengurangi total paket yang dikirim sebesar $\pm 77\%$ dari paket semula.
2. Performansi pengiriman menggunakan metode kompresi memiliki keunggulan dalam waktu pengiriman tetapi menurunkan

tingkat kesuksesan dan meningkatkan rerata delay saat pengiriman data gambar. Namun penurunan yang terjadi masih dianggap wajar karena perbedaan hasil yang diperoleh hanya sedikit dibandingkan dengan pengiriman yang tidak menggunakan kompresi. Hasil yang diperoleh dari pengujian performansi pengiriman sebagai berikut,

- a. Dari sisi waktu pengiriman adalah penghematan waktu yang diperoleh lebih dari 67% pada setiap ukuran frame yang berbeda.
- b. Dari sisi tingkat keberhasilan pengiriman paket terjadi penurunan $\pm 1\%$ pada pengiriman paket menggunakan metode kompresi.
- c. Dari sisi rerata delay terjadi peningkatan rerata delay saat menggunakan metode kompresi. Peningkatan ini disebabkan total paket pengiriman yang tidak menggunakan kompresi lebih besar daripada yang menggunakan kompresi dan jumlah retransmisi yang terjadi.

5.2. Saran

Berikut saran peneliti yang dapat dikembangkan pada penelitian berikutnya yang didasari dari penelitian ini

1. Diharapkan penelitian berikutnya dapat melakukan perbandingan atau memadukan algoritma lainnya supaya dapat membandingkan tingkat efisiensi pengiriman.
2. Diharapkan penelitian berikutnya dapat membandingkan jarak antar *node* supaya dapat mengetahui jarak efisien untuk pengiriman gambar.
3. Penelitian berikutnya dapat melakukan pengembangan pada jumlah *node* yang digunakan.

6. DAFTAR PUSTAKA

- Munir, Muhammad Misbahul. November 2018. Implementasi Wireless Sensor Node untuk Pemantauan Lahan Pertanian Berbasis Protokol 802.15.4. Skripsi. Fakultas Ilmu Komputer Universitas Brawijaya, Malang.

- Bansal Varsha, dkk. Desember 2014. The Implementation of Run Length Encoding for RGB Image Compression. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET). Volume 3 Issue 12
- Suryaningrat I Gusti Ngurah Jelantik, dkk. 2018. KOMPRESI CITRA DIGITAL MENGGUNAKAN METODE DISCRETE COSINE TRANSFORM. Prosiding SINTAK 2018. ISBN: 978-602-8557-20-7
- Aji Binariyanto, dkk. Desember 2018. Penerapan Komunikasi Berbasis 6LoWPAN(802.15.4) Antara *Node* Sensor dengan IoT Middleware. Skripsi. Fakultas Ilmu Komputer Universitas Brawijaya, Malang
- Wang, Zhongeng dan Chen, Shoufa. 2017. Performance Comparison of Image Block Compressive Sensing Based on Chaotic Sensing Matrix Using Different Basis Matrices. 2nd International Conference on Image, Vision and Computing
- Anasta, Bayu Angga. Agustus 2018. Analisis Perbandingan Algoritma Run-Length Encoding dan Algoritma Fibonacci Code dalam Kompresi Citra. Skripsi. Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara, Medan.
- IETF, 2007. IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals. *IETF Datatracker*.
- Olsson, J., 2014. *6LoWPAN demystified*. s.l.:Texas Instrument.