

## Penerapan Antarmuka Transmisi Data EKG Berbasis Komunikasi BLE (Bluetooth Low Energy) Pada Mobile Middleware

Yugi Trilia Septiana<sup>1</sup>, Eko Sakti Pramukantoro<sup>2</sup>, Rakhmadhany Primananda<sup>3</sup>

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya  
Email: <sup>1</sup>yugitrilias24@gmail.com, <sup>2</sup>ekosakti@ub.ac.id, <sup>3</sup>rakhmadhany@ub.ac.id

### Abstrak

Pada tahun 2016, penyebab kematian terbesar pertama di dunia yaitu penyakit jantung. Hal tersebut dapat terjadi disebabkan oleh sering terlambatnya proses penanganan kesehatan. *Internet of Health Things* (IoHT) merupakan salah satu perkembangan teknologi dari *IoT* yang dapat dijadikan sebagai solusi untuk masalah pemantauan elektrokardiogram (EKG). Sistem tersebut terdiri dari *sensor EKG AD8232* yang terhubung dengan *node sensor ESP32* yang memiliki module *Bluetooth Low Energy (BLE)* untuk mengirimkan data yang diperoleh dari sensor EKG ke perangkat *mobile middleware*. BLE merupakan salah satu dari media komunikasi data yang memiliki kelebihan dalam penggunaan daya yang rendah, radiasi kecil, sehingga aman penggunaannya apabila alat kesehatan saling terhubung dengan bagian tubuh manusia. BLE tergolong protokol *Low-Power Wide-Area-Network (LP-WAN)* sehingga dinilai tepat untuk dapat diterapkan dalam lingkungan *IoT*. Penelitian ini menghasilkan sebuah pengukuran kinerja BLE *node sensor ESP32* yang mampu mengolah dan mengirimkan data EKG ke *mobile middleware* dengan delay sebesar 4,02 detik. Jadi, dengan hasil yang didapatkan tersebut tidak memenuhi persyaratan dalam transmisi data EKG, yaitu delay maksimum pemantauan sebesar 4 detik.

**Keywords:** BLE, EKG, *Internet of Health Things (IoHT)*

### Abstract

*In 2016, the first cause of death in the world was heart disease. It happened because heart disease was often detected late. Internet of Health Things (IoHT) is one of the technological developments of the IoT that can be used as a solution to the problem of electrocardiogram monitoring (ECG). The system consists of AD8232 ECG sensor that connected to an ESP32 sensor that has a BLE module to send the data that was obtained from sensor ECG to mobile middleware. BLE is one of the data transmission media that has advantages in the use of low power, low radiation that make the BLE is suitable to be used in human body. BLE is one of Low-Power Wide-Area-Network (LP-WAN) protocol that makes BLE suitable to be used in IoT environment. This research resulted in a measurement of the performance of the ESP32 BLE sensor node that is capable of processing and sending ECG data to mobile middleware with a delay of 4.02 seconds. So, the results didn't fulfill the requirements in ECG data transmission which is 4 seconds maximum for monitoring uses delay.*

**Keywords:** BLE, EKG, *Internet of Health Things (IoHT)*

### 1. PENDAHULUAN

Berdasarkan data dari (WHO, 2018) pada tahun 2016, sekitar lebih dari 54% penyebab kematian terbesar didunia yaitu penyakit jantung. Sedangkan menurut penelitian sebelumnya (Shaad Mahmud, 2017), lebih dari 50% populasi di Amerika Utara penderita penyakit jantung terbesar dialami oleh masyarakat yang berusia lebih dari 55 tahun.

Menurut data Fakultas Kedokteran Universitas Islam Indonesia pada tahun 2016, sekitar 9,2% warga di Indonesia menjadi populasi penyakit jantung tertinggi, di Sulawesi Tengah sebesar 16,9% dan populasi terendah 3,5% di Lampung (Gilang Gusti Arif, 2017). Dengan adanya hal tersebut menunjukkan bahwa tidak adanya pemerataan ahli medis yang menangani perihal penyakit jantung pada daerah-daerah sehingga penyakit jantung menjadi salah satu faktor

penyebab kematian di Indonesia. Salah satu teknologi yang dapat mengatasi masalah tersebut yaitu *Internet of Things* (Carnaz, 2019). Sehingga pada saat ini *homecare*, klinik, dan rumah sakit memanfaatkan teknologi pemantauan kesehatan jarak jauh berbasis mobile untuk melakukan pemantauan kondisi vital kesehatan pasien secara *real-time* (Rodrigues, 2018).

Salah satu bagian terpenting dari “*Things*” yang terdapat pada IoHT yakni *Wireless Body Area Network (WBAN)* yang berguna untuk menyediakan layanan pemantauan kesehatan manusia dengan jarak jauh tanpa membatasi aktivitas pasien (Ullah Sana, 2012). Pemantauan detak jantung dengan menggunakan arsitektur IoT pernah dilakukan oleh (Ullah Sana, 2012). Penelitian tersebut menghasilkan sistem yang memanfaatkan *Bluetooth* sebagai media komunikasi pertukaran data. Penggunaan BLE memiliki keterkaitan dengan perangkat *mobile* sebagai salah satu media komunikasi pertukaran data dalam kehidupan sehari-hari. Pada umumnya aplikasi pemantauan kesehatan jarak jauh membutuhkan sensor untuk mengumpulkan data, *mobile* sebagai tampilan, dan koneksi internet (Rodrigues, 2018). Pada penelitian (Huang, 2015) yaitu rancang bangun alat pemantauan jumlah denyut jantung berbasis Android menggunakan protokol *Bluetooth* pada Modul HC-05 untuk proses perekaman dan pengukuran data.

Fokus dalam penelitian ini adalah penerapan sistem pemantauan dan analisis data EKG secara realtime yang didapatkan dari BLE node sensor dan dihubungkan dengan perangkat *mobile*. Protokol komunikasi yang digunakan dalam penelitian ini yaitu BLE pada ESP32 yang memiliki kelebihan tingkat radiasi yang rendah, hemat daya penggunaan dibandingkan dengan *Bluetooth* klasik pada modul HC-05, dan sudah banyak perangkat *mobile android* yang memiliki protokol BLE. Dalam penelitian ini, perangkat yang dipakai terdiri atas Sensor EKG AD8232, node sensor ESP32 yang memiliki protokol BLE, dan perangkat *mobile android* sebagai *mobile middleware*. Selanjutnya, sistem akan diuji dengan mengacu pada kebutuhan sistem yang digunakan dalam pengujian. Hasil yang diharapkan dari penelitian ini adalah keberhasilan dalam penggunaan protokol BLE pada node sensor ESP32 dalam mentransmisikan data EKG serta terhubung dan dapat ditampilkan pada perangkat *mobile*

android.

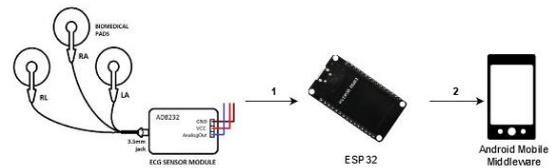
## 2. LANDASAN KEPUSTAKAAN

Farid Touati, dalam penelitiannya pada (Farid Touati, 2013) menerapkan pemantauan EKG secara real-time dengan menggunakan media komunikasi *Bluetooth Low Energy (BLE)*. Perangkat yang digunakan untuk menampilkan hasil dari penelitian yang ia lakukan yaitu pada perangkat PC/Komputer.

Huang, dalam penelitiannya pada (Huang, 2015) menerapkan sistem pengukuran detak jantung dengan melakukan klasifikasi dengan berbagai kategori *class*. Perangkat sensor yang digunakan dalam penelitian Huang, yaitu sensor *accelerometer* sebagai perekam data EKG yang akan digunakan.

Muhajirin dalam penelitiannya pada (Muhajirin, 2018) menerapkan sistem pada perangkat PC/Komputer sebagai media tampilan untuk melakukan pemantauan detak jantung.

## 3. PERANCANGAN



Gambar 1. Gambaran Umum Sistem

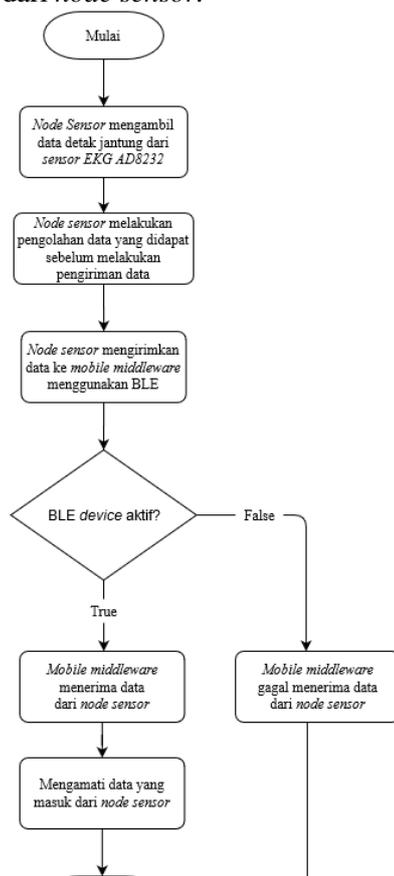
Gambar 1 memberikan gambaran umum terkait perancangan sistem pemantauan detak jantung yang saling terhubung antara *node sensor ESP32* dengan perangkat *mobile*. Protokol BLE digunakan sebagai media pengiriman diantara *node sensor* dengan *mobile* dalam hal menerima data EKG. Pada awalnya, data sensor yang digunakan berupa data real-time yang didapatkan dari penempelan elektroda pada tubuh manusia dan tersambung dengan sensor EKG AD8232. Setelah itu, data yang telah didapat dari sensor EKG akan diteruskan pada *node sensor ESP32* untuk dilakukan proses pengolahan data dari data analog menjadi digital.

Kemudian, *node sensor* akan mengirimkan data tersebut ke *mobile*. Sebelum proses pengiriman data berlangsung, langkah yang akan dilakukan yaitu dengan menghubungkan antar BLE terlebih dahulu diantara kedua perangkat tersebut. Apabila kedua BLE pada kedua perangkat sudah dalam keadaan

“standby” maka akan *mobile* akan melakukan proses *scanning device BLE* dan mencoba menemukan BLE *node sensor ESP32* yang digunakan. Jika *device BLE node sensor* telah ditemukan oleh perangkat *mobile*, maka *mobile android* akan menghubungkan dengan BLE *node sensor*. Apabila *device BLE* antar keduanya telah saling terhubung, maka proses transmisi data ke dalam *mobile android*, dan pengiriman data tersebut berhasil, maka *mobile android* akan dapat menerima dan menampilkan data yang telah diolah oleh *node sensor ESP32*.

### 3.1 Perancangan Pengambilan Data dari Node Sensor

Rancangan pengambilan data dari *node sensor* terdapat pada Gambar 2. diagram alur perancangan dalam pengambilan data yang berasal dari *node sensor*.



Gambar 2. Diagram Alur Perancangan Pengambilan Data dari Node Sensor

Pada Gambar 2 menjelaskan bagaimana alur pengambilan data dari *node sensor* ke *mobile middleware*. Proses pengambilan data ini dapat dilakukan ketika *mobile middleware* telah berhasil melakukan proses *scanning* protokol *device Bluetooth Low Energy (BLE)* yang dimiliki oleh *node sensor*. Setelah BLE

*node sensor* ditemukan, kedua perangkat *node sensor ESP32* dan *mobile middleware* disambungkan. Kemudian apabila kedua perangkat sudah saling terhubung, maka data yang telah dilakukan pengolahan pada *node sensor*, dapat dikirimkan dan diterima oleh perangkat *mobile middleware*.

### 3.2 Perancangan Pengukuran Kinerja Connecting Bluetooth Low Energy Node Sensor dan Mobile Middleware

Rancangan pengukuran kinerja BLE. Dalam melakukan pengukuran kinerja ini memiliki beberapa parameter yang digunakan dalam pengukuran, yaitu *throughput*, *delay transmission data*, *delay scanning BLE*, *packet lost*, dan *validasi data*.

#### 3.2.1 Perancangan Pengukuran Throughput

Rancangan pengukuran kinerja *throughput* ini dilakukan setelah perangkat *mobile middleware* berhasil menerima data dari *node sensor*. Kemudian peneliti melakukan pengamatan dari data-data yang masuk dalam proses transmisi data. Dalam proses transmisi data tersebut, peneliti memperhatikan berapa banyak jumlah data yang masuk dan jumlah waktu selama proses transmisi data dilakukan. Setelah itu, peneliti melakukan perhitungan antara jumlah paket data yang dikirimkan dari *node sensor* ke *mobile middleware* dan membagi dengan waktu pengiriman yang telah ditempuh. Perhitungan *throughput* terdapat pada persamaan (1) di bawah ini.

$$Throughput = \frac{\sum Packet\ Sent}{Sent\ Time} \tag{1}$$

#### 3.2.2 Perancangan Pengukuran Delay Transmission Data

Rancangan pengukuran kinerja *delay transmission data* dilakukan peneliti untuk dapat menghitung dan memperoleh *delay* pada setiap proses transmisi data dari *node sensor* ke perangkat *mobile*. Rancangan *delay transmisi data* dapat dilihat pada Gambar 3.

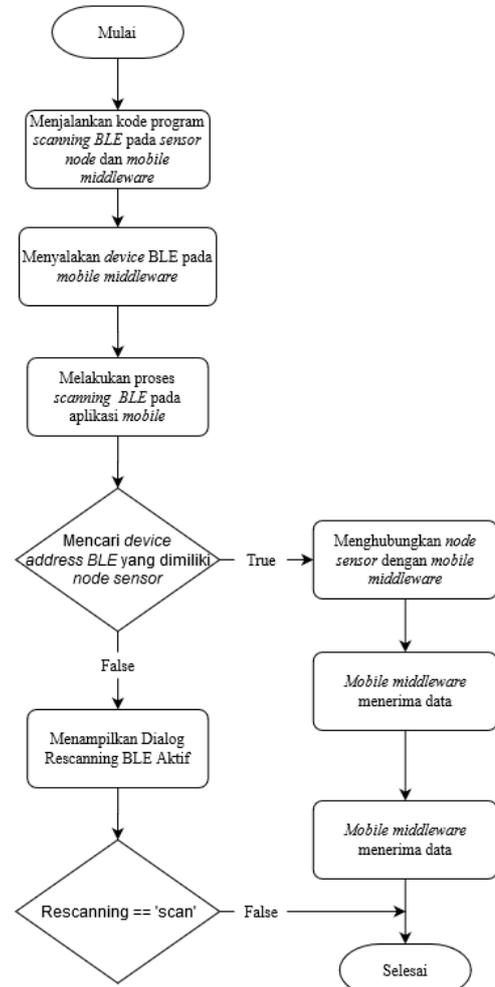


Gambar 3. Diagram Alur Pengukuran Delay Transmission Data

Gambar 3 menjelaskan tentang bagaimana alur pengukuran delay dalam proses transmisi data. Tahap pertama yang dilakukan yaitu menjalankan kode program pengiriman data yang sudah berhasil diolah oleh *node sensor*. Kedua, mengamati dan mencatat *timestamp* dalam setiap proses data yang masuk pada *mobile middleware*. Selanjutnya proses terakhir, yaitu melakukan perhitungan selisih dari *timestamp* dalam setiap proses *mobile middleware* menerima data. Kemudian, peneliti menghitung rata-rata dari perhitungan selisih pengiriman data.

### 3.2.3 Perancangan Pengukuran Delay Scanning Bluetooth Low Energy (BLE)

Rancangan *delay scanning Bluetooth Low Energy (BLE)* terdapat pada Gambar 4.



Gambar 4. Diagram Alur Perancangan Delay Scanning BLE

Gambar 4 menjelaskan tentang alur dari pengukuran *delay scanning BLE*. Tahap pertama yang dilakukan yaitu menjalankan kode program pengiriman data yang sudah berhasil diolah oleh *node sensor*. Selanjutnya, menyalakan *device BLE* pada *mobile middleware* dan melakukan proses *scanning BLE* yang dimiliki oleh *node sensor*. Proses *scanning* tersebut dilakukan dengan mencari *address BLE* yang terdapat pada tampilan dialog *scanning mobile middleware*. Setelah *device BLE node sensor* dapat terdeteksi, maka antar kedua perangkat tersebut dihubungkan agar *mobile middleware* dapat menerima data. Kemudian, peneliti mengamati dan mencatat *timestamp* dalam setiap proses data yang masuk pada *mobile middleware*. Selanjutnya proses terakhir, yaitu melakukan perhitungan selisih

dari *timestamp* dalam setiap proses *mobile middleware* menerima data. Kemudian, peneliti menghitung rata-rata dari perhitungan selisih pengiriman data. Apabila dalam proses *scanning* yang dilakukan *mobile middleware* tidak ditemukan *address* dari *BLE node sensor*, maka proses *scanning BLE* terus dilanjutkan hingga mendapatkannya.

### 3.2.4 Perancangan Pengukuran *Packet Lost*

Pengukuran kinerja paket data *lost* ini perlu dilakukan agar dapat mengetahui banyak data terkirim dan masuk dari *node sensor ESP32* ke *mobile middleware*. Peneliti melakukan pengamatan data-data dalam proses transmisi data. Dalam proses transmisi data tersebut, peneliti memperhatikan dan mencatat data pada waktu berapa saja yang tidak berhasil terkirim. Setelah itu, peneliti melakukan perhitungan antara jumlah paket data yang dikirimkan dari *node sensor* ke *mobile middleware* dan membagi dengan waktu pengiriman yang telah ditempuh.

### 3.2.5 Perancangan Pengukuran Validasi Data

Rancangan pengukuran validasi data pada penelitian ini dilakukan untuk mengetahui data-data yang diterima oleh sistem *mobile middleware* apakah sama dengan data-data yang dikirimkan. Integritas data yang didapatkan harus memiliki kesamaan data dari awal hingga akhir pengiriman. Data yang akan dilakukan pengamatan untuk dianalisis memiliki integritas data, yaitu data yang berasal dari perekaman data *sensor* detak jantung yang diteruskan ke perangkat *mobile middleware*.

### 3.3 Perancangan Pengujian

Agar dapat memberikan gambaran mengenai pengujian sistem yang akan dilakukan, maka perlu adanya proses perancangan dalam sebuah proses pengujian. Perancangan ini berisi tentang rancangan skenario pengujian, tahapan prosedur dalam melakukan pengujian, pengukuran sebuah kesesuaian, dan keberhasilan sistem dalam memenuhi kebutuhan-kebutuhan sistem.

## 4. IMPLEMENTASI

### 4.1 Implementasi Kode Program Pengambilan Data dari *Node Sensor*

Dalam implementasi ini, kode program yang digunakan oleh *node sensor* untuk mengolah data dan mengirim data yaitu menggunakan Bahasa pemrograman Arduino. Sedangkan untuk kode program yang digunakan dalam menyambungkan perangkat *mobile middleware* dengan *node sensor* yakni dengan menggunakan Bahasa pemrograman Android Studio. Kode program yang terdapat dalam *node sensor* berisi tentang bagaimana cara menyambungkan antara *device BLE node sensor* dengan *mobile middleware*. Sebelumnya, penulis melakukan proses pengolahan data EKG yang telah didapatkan secara real-time dari sensor EKG AD8232 yang berupa data analog dan diubah menjadi data digital. Selain itu, terdapat baris kode program yang berisi tentang cara menyambungkan perangkat BLE dan menghentikan sambungan BLE antara *mobile middleware* dengan *node sensor* yang digunakan. Sedangkan kode program pada tampilan *mobile* berisi tentang kode untuk menerima data dari *node sensor* yang telah berhasil diolah menjadi data digital. Selain itu juga, kode program untuk melakukan proses *scanning device BLE* serta pengiriman data menuju *cloud* juga disertakan dalam kode program android studio.

### 4.2 Implementasi Pengukuran *Throughput*

Implementasi *throughput* dalam penelitian ini dilakukan berdasarkan oleh jumlah packet data dan waktu dalam setiap pengiriman data. Pengukuran *throughput* ini digunakan sebagai salah satu parameter untuk dapat mengetahui kinerja dari protokol BLE yang diterapkan dalam penelitian ini.

### 4.3 Implementasi Pengukuran *Delay Scanning BLE*

Implementasi ini dilakukan untuk mencari perangkat *device BLE* yang digunakan oleh *node sensor ESP32*. *Device BLE node sensor ESP32* yang digunakan memiliki nama *device 24L0A:C4:30:F3:0E*. Apabila *device BLE* dari *node sensor* berhasil ditemukan, maka aplikasi pada sistem *mobile* menyimpan *device* pada *scannCallback*. Kemudian, perangkat *mobile*

akan menampilkan log BLE dari *node sensor* yang digunakan. Implementasi *delay scanning* yang dimaksud dapat dilihat dalam Gambar 5.

```

MainActivity.java
1 private BluetoothAdapter.LeScanCallback scanCallback =
2     new BluetoothAdapter.LeScanCallback() {
3         @Override
4         public void onLeScan(BluetoothDevice device, int rssi,
5             byte[] scanRecord) {
6             if (device.getAddress().equals("24:0A:C4:30:F3:0E "))
7             {
8                 myDevice = device;
9                 Log.d("packetReceived", "packetReceived+"
10                    "+device.getAddress());
11                 packetReceived++;
12                 Log.d("bleAddress", "device1 address : " +
13                    myDevice.getAddress());
14             }
15         };
    
```

Gambar 5. Kode Program Scanning BLE

#### 4.4 Implementasi Pengukuran Delay Transmisi Data

Implementasi ini, dilakukan pada saat perangkat *mobile middleware* sudah berhasil mendapatkan data yang diperoleh dari sensor EKG AD8232 dan *node sensor ESP32*. Perangkat *mobile* menampilkan aplikasi yang berisi *device address*, *status device* terkoneksi atau tidak, data yang diterima, status response yang didapatkan setelah data tersebut dikirimkan ke *cloud IoTApps*. Pengukuran *delay* transmisi data dapat dilihat dari *timestamp* yang muncul pada aplikasi *mobile* pada setiap pengiriman data dan dilakukan proses perhitungan *delay* sesuai dengan rumus yang ada. Perhitungan *delay* transmisi data dilakukan berdasarkan waktu dimulainya pengiriman *packet* data sampai *packet* data diterima. Data yang digunakan berasal dari sensor EKG AD8232 yang dilakukan secara *realtime*.

#### 4.5 Implementasi Pengukuran Packet Lost

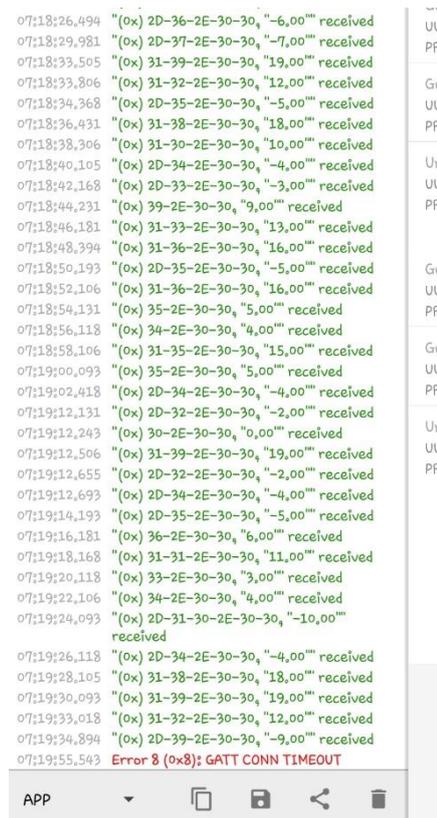
Implementasi *packet lost* dilakukan agar dapat mengetahui jumlah banyak data yang diterima oleh *mobile middleware* dari *node sensor ESP32*. Dengan persamaan rumus perhitungan *packet lost* dapat dilihat dari persamaan (2)

$$Lost = \frac{\sum Packet Loss}{\sum Packet Total} \times 100 \quad (2)$$

#### 4.6 Implementasi Pengukuran Validasi Data

Implementasi validasi data ini dilakukan untuk mengetahui *packet* data yang diterima

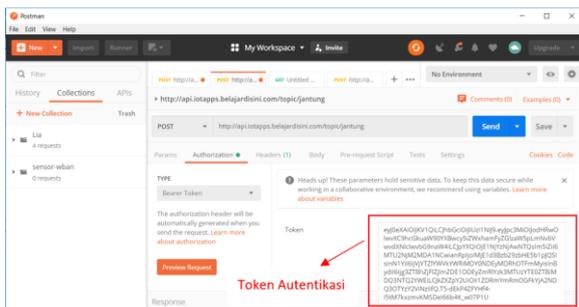
oleh perangkat *mobile middleware* sama dengan *packet* data yang berhasil dikirim perangkat yang digunakan oleh sistem. Untuk dapat mengetahui data-data yang berhasil diterima sama dengan data yang dikirim, dapat diketahui pada tampilan data pada perangkat *mobile middleware*. Tampilan pada perangkat *mobile* yang dimaksud dapat dilihat pada Gambar 6.



Gambar 6. Tampilan Data EKG pada *Mobile Middleware*

#### 4.7 Implementasi Konfigurasi Postman

Implementasi konfigurasi postman digunakan untuk menyimpan data yang didapatkan dari *node sensor*. Tahapan untuk dapat menyimpan data-data yang ingin disimpan pada *cloud* yang pertama yakni memasukkan autentikasi yang terdapat pada aplikasi postman. Autentikasi dapat dilakukan dengan cara menambahkan kode token yang valid pada bagian *request* alamat autentikasi postman. Tampilan tahapan autentikasi aplikasi postman dapat dilihat pada Gambar 7. Untuk selanjutnya menjalankan tahap *request GET* untuk dapat mengakses data dari perangkat *mobile middleware*. Tampilan kode program *request GET* dalam postman dapat dilihat pada Gambar 8 di bawah ini.



Gambar 7. Auentikasi dalam Postman

```

1 var request = require('request');
2 var options = {
3   'method': 'GET',
4   'url': 'http://api.iotapps.belajardisini.com/user/login',
5   'headers': {
6     'Content-Type': 'application/x-www-form-urlencoded'
7   },
8   formData: {
9     'token': '10ccdeb64c61f8e84b00e05630992e64b782751',
10    'secret': '4c3c6774db2998ae4d902930462dc9e8'
11  }
12 };
13 request(options, function (error, response) {
14   if (error) throw new Error(error);
15   console.log(response.body);
16 });
17
    
```

Gambar 8. Tampilan Request GET Postman

#### 4.8 Implementasi Pengiriman Data ke Cloud

Implementasi pengiriman data ke Cloud, dilakukan secara langsung pada saat data pertama kali masuk pada perangkat *mobile*. Sehingga data-data yang telah berhasil masuk dalam *mobile* akan langsung tersimpan pada *cloud*. Untuk dapat melakukan penyimpanan data pada *cloud* perlu dibuat kode program pada *mobile* sehingga dapat terhubung dengan media penyimpanan *cloud*. Potongan kode program yang dapat terhubung dengan media penyimpanan *cloud* dapat dilihat pada Gambar 9 di bawah ini.

```

DeviceControlActivity.java
1 private void postRequest(String postdata) throws
  IOException {
2   String url =
3     "http://api.iotapps.belajardisini.com/topic/jantung";
4
5   String token = "Bearer
6   eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOlwvX
7   C9hcGtuaW90YXBwcy5iZWxhamFyZGlzaW5pLnVudXN1c1wvYz9naW4
8   iLCJpYXQiOiJlbnYyZnJhbnR0SIm5iZiI6MTUzNmM2MDA1NCwianRpIjo
9   iMjE1d3Bzb29zbHE5b1pJQSIsInN1YiI6IjYvYjY7LWVxYWRlMDY0ND
10  EYMDRlOTFmYyIsInR5cCI6IjE6Ijg3ZTBhZjFlZjlmZDZlODRlODRlOD
11  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
12  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
13  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
14  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
15  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
16  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
17  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
18  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
19  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
20  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
21  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
22  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
23  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
24  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
25  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
26  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
27  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
28  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
29  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
30  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
31  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
32  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
33  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
34  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
35  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
36  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
37  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
38  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
39  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
40  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
41  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
42  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
43  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
44  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
45  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
46  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
47  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
48  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
49  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
50  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
51  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
52  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
53  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
54  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
55  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
56  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
57  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
58  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
59  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
60  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
61  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
62  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
63  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
64  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
65  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
66  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
67  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
68  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
69  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
70  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
71  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
72  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
73  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
74  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
75  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
76  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
77  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
78  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
79  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
80  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
81  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
82  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
83  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
84  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
85  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
86  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
87  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
88  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
89  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
90  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
91  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
92  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
93  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
94  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
95  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
96  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
97  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
98  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
99  RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
100 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
101 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
102 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
103 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
104 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
105 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
106 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
107 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
108 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
109 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
110 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
111 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
112 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
113 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
114 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
115 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
116 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
117 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
118 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
119 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
120 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
121 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
122 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
123 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
124 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
125 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
126 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
127 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
128 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
129 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
130 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
131 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
132 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
133 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
134 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
135 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
136 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
137 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
138 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
139 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
140 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
141 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
142 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
143 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
144 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
145 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
146 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
147 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
148 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
149 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
150 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
151 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
152 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
153 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
154 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
155 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
156 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
157 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
158 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
159 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
160 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
161 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
162 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
163 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
164 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
165 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
166 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
167 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
168 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
169 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
170 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
171 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
172 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
173 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
174 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
175 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
176 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
177 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
178 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
179 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
180 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
181 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
182 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
183 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
184 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
185 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
186 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
187 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
188 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
189 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
190 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
191 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
192 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
193 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
194 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
195 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
196 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
197 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
198 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
199 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
200 RlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlODRlOD
    
```

Gambar 9. Program Pengiriman Data ke Cloud

## 5. PENGUJIAN DAN ANALISIS

Pelaksanaan pengujian dilakukan dengan tujuan untuk mengukur tingkat keberhasilan, kemampuan, dan kesesuaian sistem. Cara untuk dapat mengukur tingkat keberhasilan sistem yakni dengan menerapkan analisis kebutuhan dan perancangan yang telah dilakukan pada tahap sebelumnya. Kemudian, setelah pengujian berhasil dilakukan maka hasil dari pengujian tersebut akan diproses lebih lanjut untuk di analisis data agar mendapatkan satu atau beberapa kesimpulan yang akan dibahas pada bab kesimpulan dan saran. Dalam penelitian ini, pengujian terbagi menjadi enam bagian, yaitu sebagai berikut:

### 5.1 Pengujian Pengambilan Data dari Node Sensor

Tujuan dari pelaksanaan pengujian ini yakni untuk melakukan validasi kemampuan perangkat *mobile middleware* dalam menerima data dari *node sensor* melalui protokol *Bluetooth Low Energy (BLE)*. Penerapan pengujian dilakukan sesuai dengan skenario pengujian yang telah dibuat terdapat pada Tabel 1.

Tabel 1. Skenario Pengujian Pengambilan Data

Kode	M_01
Tujuan Pengujian	Menguji <i>mobile middleware</i> untuk melakukan penerimaan data dari <i>node sensor</i> menggunakan protokol <i>Bluetooth Low Energy (BLE)</i> .
Skenario Pengujian	<ol style="list-style-type: none"> <li><i>Mobile middleware</i> telah menyalakan koneksi <i>Bluetooth Low Energy</i> dan tersambung dengan BLE pada <i>node sensor</i> untuk menerima data.</li> <li>Program <i>node sensor</i> dijalankan untuk melakukan pengiriman data.</li> <li><i>Mobile middleware</i> dapat menerima data yang dikirimkan oleh <i>node sensor</i>.</li> </ol>
Hasil yang diharapkan	Data dari <i>node sensor</i> berhasil diterima oleh <i>mobile middleware</i> melalui protokol <i>Bluetooth Low Energy (BLE)</i> .
Hasil Pengujian	<i>Mobile middleware</i> dapat menerima data detak jantung dari <i>node sensor</i> menggunakan protokol <i>Bluetooth Low Energy (BLE)</i> .
Keterangan	Success

### 5.2 Pengujian Throughput

Pengujian ini dilakukan untuk dapat mengetahui ukuran *bandwidth* yang terpakai sebenarnya dalam proses transmisi data detak jantung yang dikirimkan oleh sensor. Pelaksanaan pengujian ini dilakukan dengan mengacu pada skenario pengujian yang telah dibuat dalam tahap perancangan. Skenario pengujian ini terdapat pada Tabel 2 dan hasil dari pengujian terdapat pada Tabel 3.

Tabel 2. Skenario Pengujian Throughput

Kode	M_02
Tujuan Pengujian	Menguji kinerja sistem dengan mengetahui ukuran <i>bandwidth</i> dihasilkan selama proses transmisi data.
Skenario Pengujian	<ol style="list-style-type: none"> <li>1. <i>Mobile middleware</i> telah menyalakan koneksi <i>Bluetooth Low Energy</i> dan tersambung dengan BLE pada <i>node sensor</i> untuk menerima data.</li> <li>2. Program <i>node sensor</i> dijalankan untuk melakukan pengiriman data.</li> <li>3. <i>Mobile middleware</i> dapat menerima data dari <i>node sensor</i>.</li> <li>4. <i>Mobile middleware</i> dapat menampilkan ukuran data yang diterima.</li> </ol>
Hasil yang diharapkan	<i>Mobile middleware</i> dapat menampilkan waktu transmisi data, jumlah data, dan ukuran data yang didapatkan.
Hasil Pengujian	Mampu mengetahui hasil perhitungan <i>throughput</i> dari proses transmisi data detak jantung.
Keterangan	Success

Tabel 3. Hasil Pengujian Throughput

Percobaan ke-	$\Sigma$ Packet Sent (mb)	Sent Time (s)	Throughput
1	0.195	0.00213162	91.47970376
2	0.937	0.012211771	76.72924859
3	0.25	0.002442743	102.343961
4	0.293	0.003043426	96.27308406
5	0.398	0.004548183	87.50747526
6	0.46	0.005427512	84.75338905
7	0.128	0.001290174	99.2114541
8	5.25	0.022007431	238.5557908
9	5.36	0.022067639	242.8896008
10	1.16	0.020018576	57.94617846
Rata-rata		0.009518907	117.7689886

5.1 Pengujian Delay Transmisi Data

Untuk dapat mengetahui *delay* transmisi data selama proses pengiriman data menuju perangkat *mobile middleware* maka perlu dilakukan pengujian *delay* transmisi data. Dalam pelaksanaan pengujian ini akan dilakukan sesuai skenario pengujian pada Tabel 4 serta hasil pengujiannya terdapat pada Tabel 5

Tabel 4. Skenario Pengujian Transmisi Data

Kode	M_03
Tujuan Pengujian	Menguji <i>mobile middleware</i> untuk dapat menerima dan menampilkan data-data yang masuk dan waktu pengiriman data.
Skenario Pengujian	<ol style="list-style-type: none"> <li>1. <i>Mobile middleware</i> telah menyalakan koneksi <i>Bluetooth Low Energy</i> dan tersambung dengan BLE pada <i>node sensor</i> untuk menerima data.</li> <li>2. Program <i>sensor node</i> dijalankan untuk melakukan pengiriman data sensor.</li> <li>3. <i>Mobile middleware</i> dapat menerima data dari <i>node sensor</i>.</li> <li>4. <i>Mobile middleware</i> dapat menampilkan banyaknya data yang masuk dan waktu pengiriman data.</li> </ol>
Hasil yang diharapkan	Dapat melakukan perhitungan <i>delay</i> transmisi data dari data-data yang masuk dan waktu pengiriman.
Hasil Pengujian	<i>Mobile middleware</i> dapat menerima dan menampilkan jumlah data dan waktu proses pengiriman data.
Keterangan	Success

Tabel 5. Hasil Pengujian Transmisi Data

Data ke	Timestamp Mulai	Timestamp Selesai	Selisih Timestamp (detik)
1	11:32:53	11:36:04	0:03.11
2	11:36:06	11:41:00	0:04.54
3	11:41:03	11:46:12	0:05.09
4	11:46:14	11:51:22	0:05.08
5	11:51:24	11:55:04	0:03.40
6	11:55:06	12:00:01	0:04.55
7	12:00:02	12:04:04	0:04.02
8	12:04:06	12:08:10	0:04.04
9	12:08:12	12:11:02	0:02.50
10	12:11:04	12:15:06	0:04.02
11	12:15:08	12:19:14	0:04.06
12	12:19:16	12:23:02	0:03.46
13	12:23:04	12:27:06	0:04.02
14	12:27:08	12:31:00	0:03.52
15	12:31:02	12:34:14	0:03.12
16	12:34:16	12:38:01	0:03.45
17	12:40:02	12:44:08	0:04.06
18	12:44:10	12:48:02	0:03.52
19	12:48:04	12:52:00	0:03.56
20	12:52:02	12:56:02	0:04.00
Rata-rata			0:04.02

5.2 Pengujian Delay Scanning BLE

Pada pengujian ini dilakukan proses *scanning device Bluetooth Low Energy* yang dimiliki oleh *node sensor* pada perangkat *mobile*. Jika pengujian dapat berjalan langsung dengan sukses, maka pelaksanaan pengujian tersebut telah memenuhi kebutuhan sistem. Tahap pengujian ini terdapat pada skenario pengujian Tabel 6.

Tabel 6. Skenario Pengujian Delay Scanning BLE

Kode	M_04
Tujuan Pengujian	Menguji <i>mobile middleware</i> untuk dapat melakukan proses <i>scanning device Bluetooth Low Energy (BLE)</i> yang dimiliki oleh <i>node sensor</i> yang dipakai untuk proses pengiriman data.
Skenario Pengujian	<ol style="list-style-type: none"> <li>1. <i>Mobile middleware</i> dan <i>node sensor</i> telah menyalakan koneksi <i>Bluetooth Low Energy (BLE)</i>.</li> <li>2. <i>Mobile middleware</i> dapat melakukan proses <i>scanning</i> pada aplikasi <i>Scanning Device BLE</i> dan menemukan <i>device BLE</i> dari <i>node sensor</i> yang dipakai.</li> <li>3. <i>Mobile middleware</i> menyambungkan <i>device BLE</i> yang dimilikinya dengan <i>BLE</i> milik <i>node sensor</i>.</li> </ol>
Hasil yang diharapkan	<i>Mobile middleware</i> dapat melakukan proses <i>scanning device BLE</i> dan menemukan <i>BLE</i> yang dimiliki oleh <i>node sensor ESP32</i> yang dipakai.
Hasil Pengujian	<i>Bluetooth Low Energy (BLE)</i> milik <i>node sensor ESP32</i> yang dipakai, dapat ditemukan oleh perangkat <i>mobile</i> dengan waktu sesuai dengan proses <i>scanning</i> berlangsung.
Keterangan	Success

Tabel 7. Hasil Pengujian Delay Scanning BLE

Perangkat	Time Scanning Mulai	Time Scanning Selesai	Selisih delay Scanning (s)
A	10:39:13	10:39:14	00:00:01
B	10:56:32	10:56:33	00:00:01
C	10:43:28	10:43:29	00:00:01

### 5.3 Pengujian Packet Lost

Pengujian ini dilakukan untuk dapat mengetahui paket data yang tidak berhasil diterima oleh perangkat *mobile middleware*. Pelaksanaan pengujian ini mengacu dalam skenario pengujian pada Tabel 8 di bawah ini.

Tabel 8. Skenario Pengujian *Delay Scanning BLE*

Kode	M_05
Tujuan Pengujian	Mengetahui data yang mengalami kegagalan untuk dapat diterima oleh <i>mobile middleware</i> atau <i>packet lost</i> .
Skenario Pengujian	<ol style="list-style-type: none"> <li>1. <i>Mobile middleware</i> dan <i>node sensor</i> telah menyalakan koneksi <i>Bluetooth Low Energy (BLE)</i>.</li> <li>2. <i>Mobile middleware</i> dapat melakukan proses <i>scanning</i> pada aplikasi <i>Scanning Device BLE</i> dan menemukan <i>device BLE</i> dari <i>node sensor</i> yang dipakai.</li> <li>3. <i>Mobile middleware</i> menyambungkan <i>device BLE</i> yang dimilikinya dengan <i>BLE</i> milik <i>node sensor</i>.</li> <li>4. <i>Mobile middleware</i> mampu menerima data-data yang masuk selama proses transmisi data berlangsung.</li> <li>5. <i>Mobile middleware</i> menampilkan data-data yang berhasil masuk dan data yang tidak berhasil masuk pada aplikasi <i>mobile</i> yang telah dirancang dan dibuat peneliti.</li> <li>6. Mengamati data-data yang masuk dan diterima oleh <i>mobile middleware</i>.</li> </ol>
Hasil yang diharapkan	<i>Mobile middleware</i> mampu menampilkan data-data yang berhasil masuk dan tidak sesuai dengan waktu yang berjalan pada saat proses berlangsung.
Hasil Pengujian	Mengetahui data-data yang terjadi gagal dalam pengiriman atau <i>packet lost</i> dan melakukan perhitungan.
Keterangan	Success

### 5.4 Pengujian Validasi Data

Pengujian validasi data, dilakukan untuk mengetahui data yang dikirim oleh perangkat *mobile middleware* sama atau berbeda dengan data yang berhasil diterima. Pelaksanaan pengujian ini mengacu pada skenario pengujian pada Tabel 9.

Tabel 9. Skenario Pengujian *Validasi Data*

Kode	M_06
Tujuan Pengujian	Mengetahui validitas data pada penelitian ini.
Skenario Pengujian	<ol style="list-style-type: none"> <li>1. <i>Mobile middleware</i> mampu menerima data-data yang masuk selama proses transmisi data berlangsung.</li> <li>2. <i>Mobile middleware</i> mampu menyimpan data-data yang berhasil masuk ke dalam <i>cloud IoTApps</i>.</li> <li>3. Mengamati data-data yang masuk dan diterima oleh <i>mobile middleware</i> sama dengan data yang berhasil dikirimkan semua oleh <i>node sensor</i>.</li> </ol>
Hasil yang diharapkan	<i>Mobile middleware</i> mampu menerima dan menampilkan data-data yang berhasil masuk.
Hasil Pengujian	<i>Mobile middleware</i> menerima data-data yang sesuai dengan data yang berhasil dikirim oleh perangkat <i>node sensor</i> .
Keterangan	Success

## 6. KESIMPULAN

*Mobile middleware* berhasil menerima data dari *node sensor* menggunakan protokol *Bluetooth Low Energy (BLE)*. Kinerja protokol BLE dalam menghubungkan jaringan *Internet of Health Things (IoHT)* dan perangkat *mobile middleware* diukur menggunakan parameter *throughput*, *delay*, *packet lost*, dan validasi data. Dari hasil pengujian, *throughput* data pada proses pengiriman data yaitu sebesar 117.76 MBps. Selanjutnya, dari pengukuran *delay scanning* dan *delay transmission data* menghasilkan rata-rata *processing 1000ms* dan 4 detik. Sedangkan untuk hasil rata-rata dari pengukuran *packet lost* menghasilkan 3% data yang mengalami gagal untuk diterima sistem dari beberapa data yang berhasil terkirim.

## 7. DAFTAR PUSTAKA

- Carnaz, G. J. a. N., 2019. An Overview of IoT and Healthcare. *ResearchGate*, Volume 2, pp. 1-3.
- Farid Touati, e., 2013. A Real-Time BLE enabled ECG System for Remote Monitoring. *Science Direct*, Volume 7, pp. 124-131.
- Huang, C., 2015. *ECG Monitoring And Data Analysis On Mobile Platform*. Singapore: National University of Singapore.
- Muhajjirin, e., 2018. Perancangan Sistem Pengukuran Detak Jantung Menggunakan Arduino Dengan Tampilan Personal Computer. *ResearchGate*, Volume 8.
- RodrIgues, e. a., 2018. Enabling Technologies for the Internet of Health Things. *IEEE*, Volume XX, p. 9.
- Shaad Mahmud, M. W. A. H. d. F. H., 2017. A Wireless Health Monitoring System Using Mobile Phone Accecories. *IEEE*, Volume 4, p. 6.
- Ullah Sana, H. H. B. B., 2012. A Comprehensive Survey of Wireless Body Area Networks. *Springer Science*, Volume 36, p. 1065–1094.
- WHO, 2018. *The Toop 10 causes of death*. [Online] Available at: <https://www.who.int/en/news-room/fact-sheets/detail/the-top-10-causes-of-death> [Accessed 20 12 2019].