

SISTEM MONITORING PASIEN ISOLASI MANDIRI COVID-19 BERBASIS *INTERNET OF THINGS*

Zain Bahaul Anwar

S1 Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya
Email: zain.17050874014@mhs.unesa.ac.id

Arif Widodo, Nur Kholis, Nurhayati

S1 Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya
Email: arifwidodo@unesa.ac.id, nurkholis@unesa.ac.id, nurhayati@unesa.ac.id

Abstrak

Munculnya penyakit Coronavirus Disease 2019 (COVID-19) membuat banyak yang terkena penyakit ini membuat terbatasnya fasilitas rumah sakit sebagai tempat isolasi, mulai dari yang bergejala ringan hingga tidak bergejala seperti Happy Hipoxia (pneumonia berat) membuat kadar saturasi oksigen, detak jantung, serta suhu badan akan terjadi perubahan yang abnormal, hal ini membuat pemerintah menyarankan agar yang bergejala ringan untuk isolasi mandiri. Sehingga penelitian ini membuat sistem yang dapat memonitoring pasien yang sedang melakukan isolasi mandiri agar beban rumah sakit tidak semakin bertambah. Dengan menggunakan mikrokontroler ESP32 sebagai perangkat IoT (*Internet of Things*) data seperti SpO₂ (*Oxygen Saturation*), *heartrate*, serta suhu badan sebagai paramater monitoring dapat di kirim ke server *ThingSpeak*. Pada penelitian ini didapatkan hasil pengukuran dari 10 subyek yang diukur. Didapatkan *error* pembacaan nilai SpO₂ dari yang terendah 0,11% dan tertinggi 1,20%. Kemudian nilai *heartrate error* pembacaan terendah 0,89% dan tertinggi 1,54%. Kemudian nilai suhu badan didapatkan *error* pembacaan terendah 0,19% dan tertinggi 2,78%. Serta interval pengiriman data dari *smartphone* ke server *ThingSpeak* didapat sebesar <20 detik. Sehingga Sistem ini dapat digunakan dengan baik oleh pasien yang sedang melakukan isolasi mandiri karena memiliki *error* SpO₂ yang rendah, tidak melebihi standar akurasi SpO₂ sebesar ≤4%.

Kata Kunci : ESP32, IoT *ThingSpeak*, MAX30102, DS18B20, *Bluetooth*.

Abstract

The emergence of Coronavirus Disease 2019 (COVID-19) has made many people affected by this disease, making hospital facilities limited as a place of isolation, ranging from mild symptoms to asymptomatic ones such as Happy Hypoxia (severe pneumonia) causing oxygen saturation levels, heart rate, and temperature to increase abnormal changes will occur, this makes the government recommend that those with mild symptoms to self-isolate. So that this study creates a system that can monitor patients who are self-isolating so that the hospital burden does not increase. By using the ESP32 microcontroller as an IoT (*Internet of Things*) device, data such as SpO₂ (*Oxygen Saturation*), heart rate, and body temperature as monitoring parameters can be sent to the *ThingSpeak* server. In this study, the measurement results obtained from 10 subjects were measured. The SpO₂ value reading error is obtained from the lowest 0.11% and the highest 1.20%. Then the lowest reading heart rate error is 0.89% and the highest is 1.54%. Then the body temperature value obtained the lowest reading error of 0.19% and the highest 2.78%. And the interval of sending data from the smartphone to the *ThingSpeak* server is <20 seconds. So that this system can be used properly by patients who are self-isolating because they have a low SpO₂ error, not exceeding the standard SpO₂ accuracy of 4%.

Keyword : ESP32, IoT *Thingspeak*, MAX30102, DS18B20, *Bluetooth*.

PENDAHULUAN

Pada tahun 2020 muncul penyakit baru yang belum pernah diidentifikasi oleh manusia, yang diberi nama *Coronavirus Disease 2019* (COVID-19). Untuk gejala klinis penderita COVID-19 sebagian besar kasus mengalami kesulitan bernafas dan hasil rontgen *infiltrat* pneumonia luas di kedua paru. Pada kasus pneumonia berat pada pasien remaja atau dewasa membuat kadar saturasi oksigen dalam darah dapat menurun hingga <90% saat berada pada udara kamar serta diikuti perubahan heart rate dengan gejala *takikardia* atau *bradikardia*

dengan *HeartRate*<90/menit atau >160/menit pada bayi dan *HeartRate*<70/menit atau >150/menit pada anak dan suhu tubuh yang abnormal (Kemenkes RI, 2020). Dengan tren kasus positif COVID-19 di Indonesia yang terus meningkat seiring waktu dan terbatasnya fasilitas rumah sakit untuk tempat isolasi, pasien dengan hasil tes SWAB menunjukkan hasil positif dengan gejala ringan bahkan tidak bergejala seperti *Happy Hipoxia*, maka dianjurkan untuk isolasi secara mandiri dirumah masing – masing. Hal ini dapat mengurangi beban rumah sakit

yang tiap harinya bertambah untuk pasien COVID-19 yang diisolasi.

Dengan ini rumah sakit perlu memiliki sistem yang dapat terhubung dengan pasien yang melakukan isolasi mandiri. Beberapa jurnal telah melakukan penelitian untuk melakukan monitoring jarak jauh. Penelitian menggunakan ESP8266 sebagai *wireless*. Niswar dkk, (2019) melakukan penelitian dengan menggunakan mikrokontroler Atmega328 sebagai pengontrol *fingertip* sensor SpO_2 serta sensor DS18B20, kemudian menggunakan ESP8266 sebagai komunikasi *wireless* sehingga dapat ditampilkan pada web dengan menggunakan metode *web browser*.

Penelitian yang dilakukan Marathe, dkk (2019) sama menggunakan ESP8266 sebagai modul IoT-nya, sehingga data yang dia dapat akan diposting pada server *ThingSpeak*. Pada penelitian dengan menggunakan MAX30100 untuk pembacaan SpO_2 serta detak jantung untuk suhu badan menggunakan LM35. Tetapi dalam jurnalnya belum menampilkan hasil dari penggunaan server *ThingSpeak*.

Pada penelitian yang lain didapat menggunakan Zigbee sebagai pengiriman datanya. Penelitian yang dilakukan oleh Cao (2017), menggunakan sensor SpO_2 dengan Tipe DA-100A dengan mikrokontrolernya menggunakan arduino micro, Zigbee akan mengirim data hasil pembacaan secara *wireless* ke komputer yang nantinya akan dikirim ke internet. Penelitian lainnya dengan menggunakan Zigbee oleh Hamdala, dkk (2015) dimana untuk pembacaan SpO_2 mereka membuat sendiri sensornya dengan mikrokontroler Atmega8535 kemudian data akan dikirim ke komputer untuk diolah dan ditampilkan oleh GUI (*Graphical User Interface*) LabView melalui Zigbee. Oleh Omar, dkk (2015) menggunakan Wi-Fi Module Roving RN-Xvee sebagai pengiriman data yang dapat terkoneksi dengan router yang ada, kemudian Aithal (2015) menggunakan modul BlueSMiRF sebagai pengiriman data ke *smartphone*, menggunakan Android Studio sebagai pembuatan aplikasi *smartphone* untuk menerima data.

Penelitian dari Ambadkar dan Nikam (2020) menggunakan nRF24L01 sebagai pemisah antara sensor (RX) dan bagian penerima (TX) yang berisi module SIM900L sebagai pengirim SMS sebagai mengirim informasi, menggunakan Atmega328 sebagai mikrokontroler kemudian LM35 sebagai pembaca suhu badan dan MAX30100 sebagai pembaca SpO_2 . Beberapa penelitian juga masih menggunakan sistem jaringan lokal untuk mengirim informasinya, seperti yang dilakukan Hadiyoso, dkk (2011) menggunakan jaringan LAN (802.11b) sebagai pengirim data dari sensor ke komputer *user* dan didapat jangkauan sejauh 70meter dengan *delay* pengiriman dibawah 200 ms.

Tetapi untuk memaksimalkan dalam melihat kondisi pasien selain mengirim ke pihak rumah sakit dari sisi keluarga juga perlu untuk mengetahui juga tanpa kontak langsung dengan pasien. Dengan memanfaatkan *smartphone* android dapat digunakan sebagai monitoring oleh keluarga. Untuk penelitian yang memanfaatkan *smartphone* android untuk *interface*-nya, Sameh, dkk (2020) untuk membuat *interface*-nya menggunakan Android Studio dengan cara mentransmisikan data dari alat menggunakan modul *bluetooth* HC-06 ke *smartphone*, sensor yang digunakan menggunakan MAX30102 serta XME231 sebagai sensor suhu badan. Kemudian penelitian lainnya oleh Ali, dkk (2020) menggunakan modul HC-05 dengan *interface* pada *smartphone* android untuk menampilkan hasil pembacaan sensor berupa SpO_2 serta Suhu badan. Kemudian sistemnya dipararel dengan ESP-01 untuk pengiriman data ke server *ThingSpeak* sebagai metode *Internet of Things*.

Untuk itu dalam kegiatan penelitian ini akan melakukan bagaimana cara deteksi *heartrate*, kadar saturasi oksigen dalam darah (SpO_2), serta suhu badan pasien, dengan menggunakan ESP32 sebagai mikrokontroler serta *bluetooth* yang nantinya data yang terbaca akan di tampilkan melalui *smartphone android* yang kemudian akan diteruskan ke server *ThingSpeak* sebagai monitoring jarak jauhnya.

Agar nantinya alat ini dapat digunakan oleh keluarga sehingga dapat mengetahui kondisi pasien tanpa adanya kontak langsung serta dapat dimonitoring oleh petugas medis dari sisi server *ThingSpeak* sebagai monitoring jarak jauhnya, hal ini dapat mengurangi beban rumah sakit yang meningkat seiring naiknya kasus positif, sehingga pasien dengan gejala ringan dapat isolasi mandiri dan tentunya tetap di monitor oleh petugas medis.

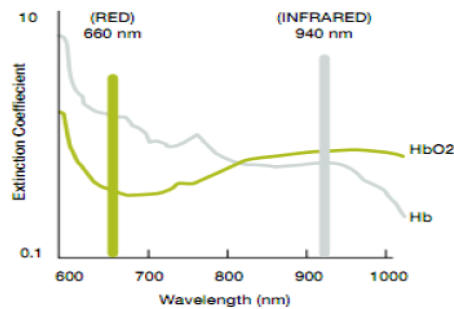
TINJAUAN PUSTAKA

Pulse Oximetry

Perjalanan oksigen melalui sistem peredaran darah yang ada pada tubuh. Darah terdeoksigenasi memasuki jantung kemudian darah dipompa ke paru-paru untuk dioksigenasi. Proses oksigenasi, darah melewati bagian alveolus paru dimana terjadi pertukaran gas (difusi). Sehingga karbondioksida (CO_2) dilepaskan sehingga darah dioksigenasi, setelah itu darah akan dipompa kembali ke aorta.

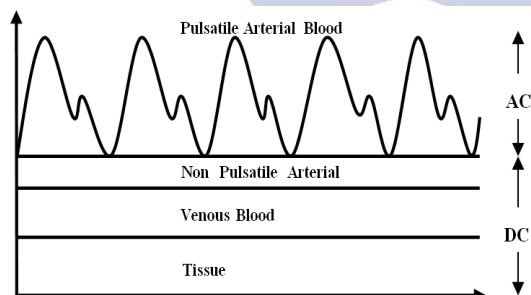
Pulse oximeter sendiri adalah cara pengukuran cara *non-invasif* untuk mengetahui kadar saturasi oksigen dalam tubuh (SpO_2). Pengukuran ini berdasarkan deteksi Hemoglobin dan *Deoxyhemoglobin* yang terlarut dalam darah, perlu menggunakan dua panjang gelombang cahaya dalam pengukuran ini, yaitu 660nm dan 940nm. Gambar

1 memberikan grafik penyerapan cahaya oleh Hb dan HbO₂.



Gambar 1. Grafik Penyerapan Cahaya Oleh Hb dan HbO₂
(Sumber : Lopez, 2012)

Hemoglobin terdeoksigenasi (Hb) memiliki penyerapan yang lebih tinggi pada 660nm dan hemoglobin teroksigenasi (HbO₂) memiliki penyerapan yang lebih tinggi pada 940nm. Kemudian pada sisi penerima pembacaan ada sebuah fotodetektor yang nantinya sinyal hasil pembacaan akan diperkuat dengan *operasional amplifier* (OpAmp) hingga dapat dibaca, kemudian sinyal tersebut akan dibagi menjadi dua yaitu, komponen sinyal AC dan komponen sinyal DC, dapat dilihat pada Gambar 2. Komponen DC hasil dari penyerapan cahaya dari jaringan kulit (kuku, otot, tulang), darah vena, dan darah arteri tidak berdenyut. Komponen AC hasil darah arteri yang berdenyut.



Gambar 2. Penyerapan Cahaya Oleh Jaringan Tubuh
(Sumber : Lopez, 2012)

Pulse Oximeter menganalisis penyerapan cahaya dari dua panjang gelombang LED (*Ligth Emitting Diode*) dari volume tambahan darah arteri berdenyut teroksigenasi (AC/DC) dan menghitung rasio penyerapan menggunakan persamaan 1.

$$R = \frac{(AC_{red}/DC_{red})}{(AC_{ir}/DC_{ir})} \quad (1)$$

Keterangan :

R = Rasio antara dua penyerapan panjang gelombang yang berbeda.

AC_{red} = Komponen sinyal (LED merah) yang nilainya berubah – ubah.

AC_{ir} = Komponen sinyal (LED *infrared*) yang nilainya berubah – ubah.

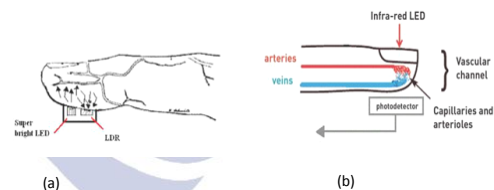
DC_{red} = Komponen sinyal (LED merah) yang nilainya tetap.

DC_{ir} = Komponen sinyal (LED *infrared*) yang nilainya tetap.

Setelah mendapatkan nilai rasio dari persamaan diatas, dapat dihitung untuk nilai SpO₂ dengan persamaan 2 berikut :

$$SpO_2 = 110 - 25 \cdot R \quad (2)$$

Dalam prakteknya, terdapat dua cara untuk penempatannya salah satunya pada jari tangan diilustrasikan pada Gambar 3. Dengan cara menstransmisikan pancaran cahaya kemudian akan didapat pantulan (*reflection*), dimana beberapa cahaya telah diserap kemudian akan ditangkap oleh penerima, cara ini dapat dilihat pada Gambar (a). Kemudian untuk cara kedua dengan cara cahaya dipancarkan (*transmisi*) pada bagian atas jari, hal ini membuat cahaya menembus bagian jari dan beberapa akan diserap oleh bagian tubuh sehingga sisa cahaya akan diterima oleh bagian penerima dibagian sisi lainnya, cara ini dapat dilihat pada Gambar (b).



Gambar 3. Metode Pengiriman Cahaya
(Sumber : Sameh dkk, 2020)

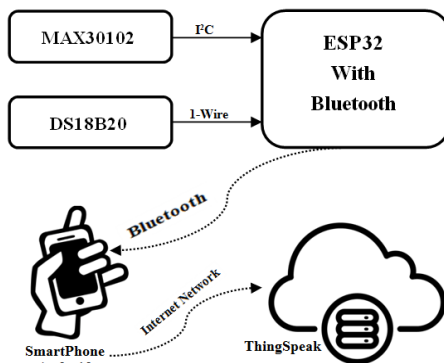
METODE PENELITIAN

Desain Sistem

Sistem monitoring yang dibuat akan memiliki bagian sensor dimana terdapat sensor MAX30102 yang akan bertugas membaca parameter berupa saturasi oksigen dalam darah (SpO₂) serta detak jantung tiap menit (BPM). Penggunaannya dengan cara menjepit jari subyek ke bagian sensor agar terbaca nilai SpO₂ serta detak jantungnya. Dimana sensor MAX30102 menggunakan teknik pantulan cahaya untuk penempatan pembacaan sensornya. Kemudian untuk pembacaan suhu badan menggunakan sensor DS18B20 yang nantinya akan membaca suhu badan subyek pada bagian lengan. Bagian mikrokontroler akan menggunakan ESP32 dimana telah memiliki *bluetooth* yang telah tertanam didalam ESP32

sehingga tidak perlu menggunakan modul *bluetooth* seperti HC-05 ataupun HC-06.

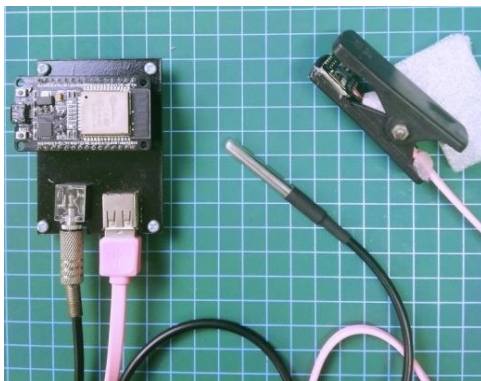
Berikut diagram blok rangkaian sistem monitoring yang dibuat pada Gambar 4, kedua sensor yaitu MAX30102 dan DS18B20 akan membaca nilai yang dikontrol oleh mikrokontroler ESP32 jika ada jari yang diletakkan pada sensor MAX30102. Setelah itu proses selanjutnya data akan dikirim ke aplikasi *smartphone* melalui jaringan *bluetooth* agar dapat ditampilkan pada layar *smartphone*. Kemudian data yang diterima akan diposting ke *cloud server ThingSpeak* sebagai monitor jarak jauhnya sehingga dapat dilihat oleh tenaga kesehatan. Pada bagian pengiriman ke server, jaringan internet pada *smartphone* akan memiliki peranan penting untuk proses pengiriman data ke server. Pada penelitian ini menggunakan *smartphone* yang memiliki jaringan 4G.



Gambar 4. Diagram Blok Kerja Sistem

Desain Hardware

Agar mempermudah dalam penggunaannya dan mengurangi banyaknya pengkabelan pada rangkain, maka rangkaian dibuat pada papan PCB yang telah didesain sedemikian rupa, dapat dilihat pada Gambar 5. Dengan mendesain PCB menggunakan aplikasi EAGLE CAD dalam pembuatannya.



Gambar 5. Desain Prototipe

Pada bagian sensor MAX30102 pengkabelan menggunakan kabel *Micro USB* yang ujungnya disambungkan secara langsung hanya memanfaatkan bagian USB tipe A sebagai koneksi antara sensor dengan kontrolernya. Bagian pembacaan sensor menggunakan jepitan hanger pakaian agar dalam uji coba tinggal menjampit jari subyek yang ingin diukur. Untuk sensor DS18B20 menggunakan konektor *jack audio* 3.5mm, susunan dari ujung konektor yaitu tegangan, data, *ground*. Hal ini bertujuan agar mudah penggunaannya dan agar terlihat rapi.

Sensor MAX30102 menggunakan komunikasi I²C, dimana pada ESP32 pin untuk komunikasi I²C pada pin GPIO22 sebagai SDA dan GPIO21 sebagai SCL. Untuk penggunaannya harus ditambahkan resistor *pull-up* sebesar 1k5Ω agar dapat terbaca oleh ESP32. Kemudian untuk sensor DS18B20 menggunakan pin digital, disini menggunakan pin GPIO4 yang di *pull-up* dengan resistor 4k7Ω karena sensor ini menggunakan komunikasi *OneWire*.

Desain Software

Pada bagian ini akan dibagi menjadi dua tahapan, pertama bagian pemrograman ESP32 serta pemrograman APK (*Android Package Kit*) pada bagian *smartphone* serta setting GUI *ThingSpeak*. Untuk pemrograman ESP32 menggunakan *Arduino IDE* kemudian untuk pembuatan APK menggunakan web *Kodular* serta bagian server menggunakan *ThingSpeak*.

Disisi ESP32 sendiri untuk pemrograman dilakukan beberapa penyesuaian. Dalam mendapatkan nilai SpO₂, dengan menggunakan modul MAX30102. Dengan memanfaatkan *library* yang ada untuk menjalankan modul. Nilai LED merah dan IR yang didapat akan dihitung Rasio dengan menggunakan rumus pada persamaan 3 yang diterapkan pada kode pemrograman.

$$R = \frac{(\sqrt{\text{sum}_{\text{red}}}) / \text{ave}_{\text{red}}}{(\sqrt{\text{sum}_{\text{ir}}}) / \text{ave}_{\text{ir}}} \quad (3)$$

Keterangan :

- R = Rasio antara dua penyerapan panjang gelombang yang berbeda.
- sum_{red} = Hasil jumlah kuadrat nilai yang dibaca terus menerus dari pancaran LED dengan warna merah.
- sum_{ir} = Hasil jumlah kuadrat nilai yang dibaca terus menerus dari pancaran LED dengan warna *infrared*.
- ave_{red} = Hasil rata - rata dari pancaran LED dengan warna merah.
- ave_{ir} = Hasil rata - rata dari pancaran LED dengan warna *infrared*.

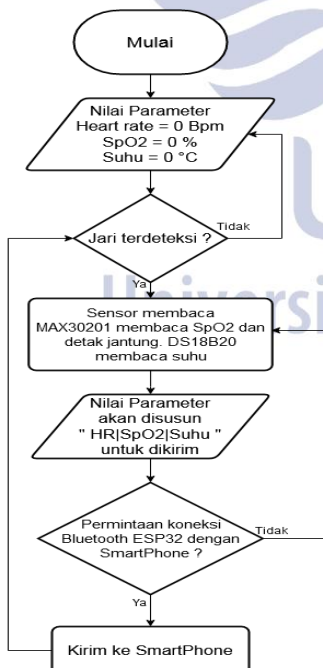
Dalam prakteknya untuk pengukuran detak jantung ini hanya membutuhkan salah satu nilai dari LED, yang warna merah ataupun *infrared*. Dengan menggunakan salah satu nilai tersebut didapatkan nilai waktu antara tiap puncak gelombang setiap kali jantung berdetak ke puncak selanjutnya (*beat*) seperti Gambar 2. Untuk mendapatkan waktunya pada kode program menggunakan fungsi waktu *microsecond* dalam menghitung selisih waktu antara beat yang terjadi sehingga lebih baik nilai yang didapat. Kemudian waktu yang didapatkan dikonversi ke nilai BPM (*beats per minute*), dengan menggunakan persamaan 4.

$$BPM = 59 / \left(\frac{time\ interval}{1000000.0} \right) \quad (4)$$

Keterangan :

BPM = Jumlah detak jantung tiap menitnya.
time interval = Nilai yang didapat dari puncak gelombang setiap kali jantung berdetak ke puncak selanjutnya.
 1000000.0 = Nilai pembagi *microsecond*.

Setelah didapatkan nilai BPM dengan perhitungan diatas nilai akan disimpan terlebih dahulu didalam sebuah variabel. Nilai BPM yang didapat akan dijumlah dengan nilai yang didapat selanjutnya, sebanyak 20 data BPM. Setelah itu dilakukan perhitungan rata-rata nilai dari 20 data tersebut sehingga akan didapat nilai BPM yang dapat stabil. Secara umum kerja bagian ESP32 dapat dilihat pada diagram alir Gambar 6.



Gambar 6. Diagram Alir Kerja ESP32

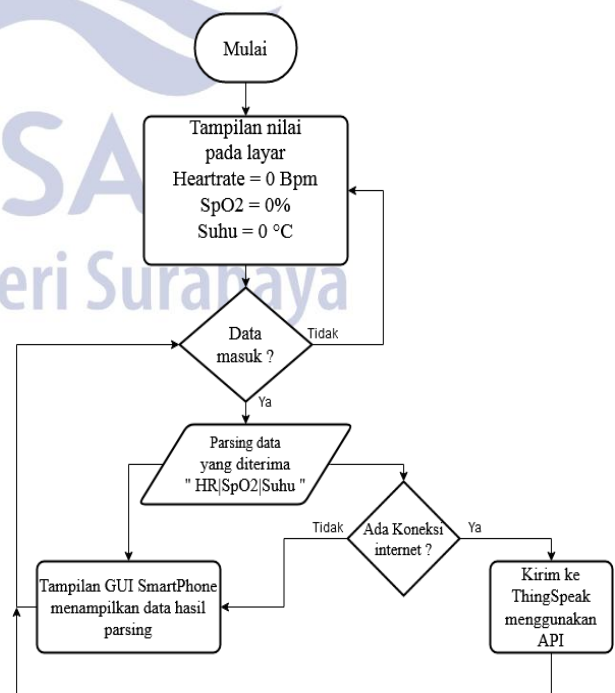
Saat alat dihidupkan, nilai dari beberapa parameter seperti *heartrate*, *SpO₂*, serta suhu akan bernilai 0 semuanya sebagai nilai awal. Kemudian sistem akan membaca, apakah sensor MAX30102 mendeteksi adanya jari yang menempel, jika tidak nilai parameter diatas akan tetap bernilai 0. Tetapi jika sensor mendeteksi adanya jari yang menempel pada sensor, maka seluruh sensor akan membaca dan nilai parameter *heartrate*, *SpO₂*, serta suhu akan berubah mengikuti hasil pembacaan sensor.

Kemudian dari nilai parameter yang didapat akan disusun menjadi satu kesatuan data. Data tersebut akan diberi pemisah berupa karakter “|” agar nantinya dapat dibedakan datanya pada bagian android. Gambar 7 menggambarkan data yang dikirim.

Heartrate		SpO2		Suhu
-----------	--	------	--	------

Gambar 7. Susunan Data Dengan Pemisah

Pada bagian *smartphone*, APK dibangun menggunakan web Kodular. Dengan jalan menyusun kode blok serta menyusun GUI untuk tampilan yang nantinya akan menjadi APK *smartphone*. Untuk mengirim data dari APK ke server *ThingSpeak* dibutuhkan API (*Application Programming Interface*) untuk menghubungkan ke server. Dengan meng-copy API yang ada pada bagian server dan dimasukkan pada bagian penyusun aplikasi yang dibuat. Gambar 8 akan menjelaskan bagaimana kerja dari bagian aplikasi yang telah dibuat.



Gambar 8. Diagram Alir Aplikasi Smartphone

Penjelasan dari diagram alir diatas sebagai berikut. Ketika membuka aplikasi yang telah dibuat maka akan tampilan di layar akan menampilkan HeartRate = 0BPM, SpO2 = 0% kemudian Suhu = 0°C, karena belum adanya koneksi *bluetooth* antara *smartphone* dengan alat sehingga tidak ada data yang dikirim ke *smartphone* serta server *ThingSpeak*.

Jika telah disambungkan antara kedua *bluetooth*, maka data akan dikirim ke aplikasi *smartphone* agar dapat dilakukan proses selanjutnya. Kemudian data yang diterima akan dilakukan proses *parsing data*. Dimana data yang diterima masih menjadi satu data, sehingga perlu dipisah antara *heartrate*, SpO2, serta suhu yang telah diberi pembatas karakter “[”], sehingga akan diambil satu persatu data. Jika proses *parsing data* selesai maka nilai yang didapat akan ditampilkan ke layar, disertai data yang didapat dikirim ke server *ThingSpeak*. Tetapi jika tidak ada koneksi internet untuk mengirim ke server maka tidak terjadi pengiriman data ke server tetapi hanya akan ditampilkan pada layar *smartphone* data yang telah diolah tadi.

Rancang Uji Coba

Pada bagian tampilan *smartphone* sendiri, dibuat sebuah aplikasi yang dapat menerima data dari alat melalui *bluetooth* yang ada pada MCU ESP32. Saat *bluetooth smartphone* dan *bluetooth* alat sudah tersambung maka data akan ditampilkan pada layar *smartphone* dan langsung akan diposting pada *ThingSpeak* dengan menggunakan jaringan internet *smartphone*. Dalam hal ini aplikasi dibuat dengan mengakses web Kodular.io, kemudian dapat diinstall pada *smartphone* android dengan file berupa .apk .

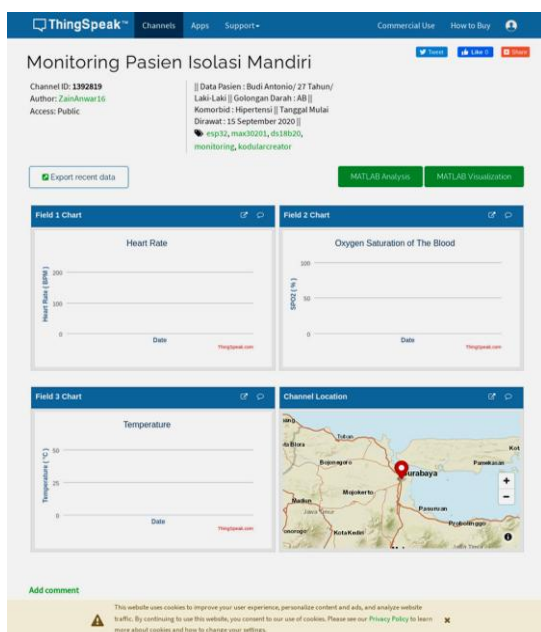
Bagian *ThingSpeak* sendiri data yang diterima dari *smartphone* akan dijadikan menjadi sebuah grafik setiap waktunya. Dengan *ThingSpeak* mempermudah pembacaan data yang diterima, dan dapat diunduh juga untuk rekap datanya dalam bentuk CSV (*Comma Separated Values*). Pada Gambar 9 menampilkan isi dari *ThingSpeak* yang memuat beberapa informasi subyek. Dimana ada tampilan peta, tampilan ini hanya sebagai tambahan informasi subyek saja serta cara merubah posisi petanya hanya melalui pengaturan yang ada pada webnya bukan menggunakan sensor tertentu.

Dalam pengukuran nantinya akan dilakukan beberapa pengaturan untuk pengukuran. Secara umum, pengukuran akan mengambil 10 orang dengan rentang usia berbeda untuk didapatkan data SpO₂, *heartrate* serta suhu badan.

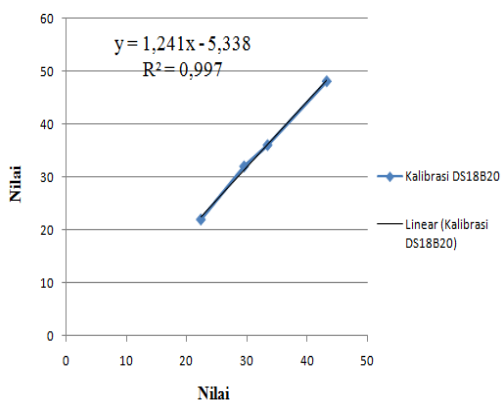
Kemudian untuk metode pembacaan parameter SpO₂ dan *heartrate* dengan cara meletakkan sensor MAX30102 dan alat pembanding pada jari subyek. Kemudian pembacaan dilakukan dengan menunggu beberapa menit agar didapatkan nilai stabil. Saat pengukuran subyek harus diam jangan banyak pergerakan agar didapat nilai yang baik.

Untuk pengambilan data, sensor akan diletakkan pada bagian tubuh yaitu pada lipatan lengan tangan, dimana bagian yang biasanya untuk mengukur suhu badan. Jadi sensor DS18B20 dan sensor suhu multimeter akan diletakkan bersamaan untuk mendapatkan pembacaan dari keduanya. Kemudian untuk bagian pembacaan suhu menggunakan DS18B20. Perlu dilakukan kalibrasi agar mendapatkan nilai yang mendekati dengan alat pembanding yang digunakan. Dengan cara mengumpulkan perbedaan beberapa nilai suhu yang didapat dari sensor yang dibuat serta dengan alat ukur suhu yang ada pada multimeter Zotex ZT C-4.

Kemudian diambil yang bervariasi dimulai dari suhu yang rendah hingga suhu yang tinggi, jika pada data suhu yang diambil dari kisaran 22 hingga 46 derajat. Setelah itu dengan menggunakan Microsoft Excel sebagai pengolah data agar mendapatkan nilai kalibrasi, dengan menggunakan fungsi *Format Trendline* maka akan ditemukan nilainya berupa persamaan y, dimana nilai y akan digunakan didalam perhitungan kode programing ESP32. Gambar 10 memberikan gambaran hasil perhitungan kalibrasi.



Gambar 9. Tampilan Bagian Server Thingspeak



Gambar 10. Hasil Kalibrasi DS18B20

Untuk menentukan apakah alat yang dibuat sudah sama dengan alat ukur yang terstandart maka dilakukan perhitungan *error* dengan alat ukur yang telah ada dipasaran. Menggunakan Serenity Pulse Oximetry SP090 sebagai pembandingnya dan mencari *error* pembacaan. Serenity SP090 sendiri sudah terstandarisasi oleh Kementerian Kesehatan Indonesia dengan kode KEMENKES RI AKL: 20502914772.

Untuk memberikan tingkat akurasi yang ingin didapatkan maka perlu standarisasi akurasi dari suatu alat. Oleh lembaga P2 SMTP LIPI (2016), dimana sesuai ISO 80601-2-61 akurasi SpO₂ yang direkomendasikan sebesar ≤4% pada range 70 % - 100 %. Untuk mencari nilai persentase *error* antara nilai yang dibuat dengan Serenity SP090, menggunakan rumus *error* relatif ditampilkan pada persamaan 5.

$$\text{Error (\%)} = \left| \frac{(\text{Nilai Serenity/Zotek} - \text{Nilai alat})}{\text{nilai Serenity/Zotek}} \right| * 100\% \quad (5)$$

Keterangan :

Error (%) = *Error* relatif.

Nilai Serenity/Zotek = Nilai yang didapat dari alat Serenity (SpO₂ dan *heartrate*), serta dari Zotek berupa suhu.

Nilai alat = Nilai yang didapat dari alat penelitian berupa SpO₂, *heartrate*, dan suhu.

HASIL DAN PEMBAHASAN

Hasil Pembacaan SpO₂, *HeartRate* serta Suhu Badan

Berikut hasil pengukuran yang didapat dari 10 subyek dengan beberapa umur yang berbeda agar didapat nilai parameter yang bervariasi. Tabel 1 akan menyajikan data SpO₂ dari 10 subyek yang diukur. Didapatkan perbandingan data antara SpO₂ dari alat serenity dan SpO₂ dari alat penelitian. Kemudian dicari *error* relatifnya, dimana didapatkan nilai dari yang terkecil 0,11% dan tertinggi 1,20%.

Tabel 1. Hasil Pengukuran SpO₂

Subyek	Umur	SpO ₂ Serenity	SpO ₂ Alat Penelitian	Error
1	5	97 %	98,16 %	1,20 %
2	11	98 %	97,64 %	0,37 %
3	23	98 %	97,83 %	0,17 %
4	22	99 %	99,77 %	0,71 %
5	22	98 %	99 %	1,02 %
6	22	100 %	99,16 %	0,84 %
7	22	99 %	98,83 %	0,17 %
8	42	99 %	98,87 %	0,13 %
9	49	99 %	98,38 %	0,63 %
10	50	99 %	99,11 %	0,11 %

Tabel 2 menyajikan hasil dari pembacaan *heartrate* dari 10 subyek yang telah diukur. Didapatkan perbandingan data antara *heartrate* dari alat serenity dan *heartrate* dari alat penelitian. Kemudian dicari *error* relatifnya, dimana didapatkan nilai dari yang terkecil 0,89% dan tertinggi 1,54%.

Tabel 2. Hasil Pengukuran *HeartRate*

Subyek	Umur	<i>HeartRate</i> Serenity	<i>HeartRate</i> Alat Penelitian	Error
1	5	107 Bpm	106 Bpm	0,93 %
2	11	112 Bpm	111 Bpm	0,89 %
3	23	77 Bpm	76 Bpm	1,30 %
4	22	78 Bpm	77 Bpm	1,28 %
5	22	100 Bpm	101 Bpm	1,00 %
6	22	88 Bpm	87 Bpm	1,14 %
7	22	75 Bpm	76 Bpm	1,33 %
8	42	70 Bpm	69 Bpm	1,43 %
9	49	65 Bpm	64 Bpm	1,54 %
10	50	77 Bpm	76 Bpm	1,30 %

Tabel 3 menyajikan hasil dari pembacaan suhu badan dari 10 subyek yang telah diukur. Didapatkan

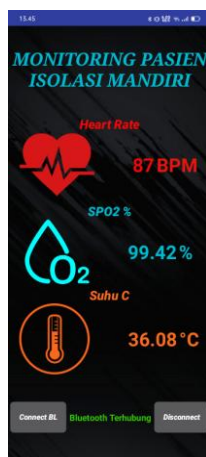
perbandingan data antara suhu badan dari alat Zotek ZT-C4 dan suhu badan dari alat penelitian. Kemudian dicari *error* relatifnya, dimana didapatkan nilai dari yang terkecil 0,19% dan tertinggi 2,78%. Sensor DS18B20 kurang sensitif karena daerah yang harusnya dikenai suhu terlalu luas, sedangkan pembanding yang digunakan menggunakan *thermocouple* yang memiliki laju perubahan data yang cepat.

Tabel 3. Hasil Pengukuran Suhu Badan

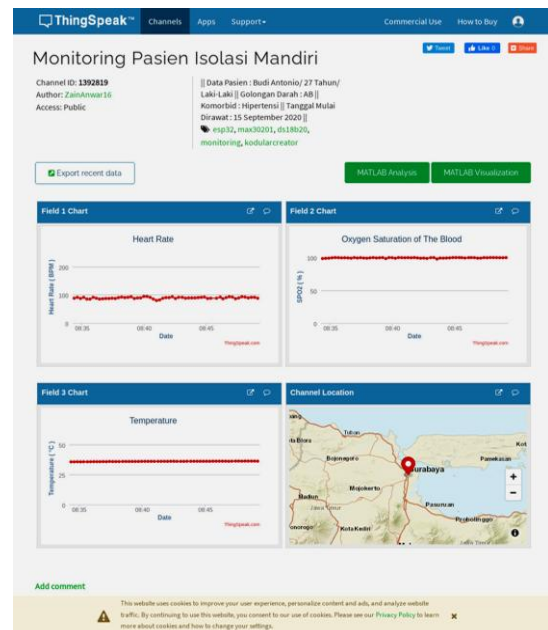
Subyek	Umur	Suhu Zotek ZT-C4	Suhu Alat Penelitian	Error
1	5	35°C	34,37°C	1,80 %
2	11	36°C	35°C	2,78 %
3	23	36°C	35,75°C	0,69 %
4	22	36°C	35,69°C	0,86 %
5	22	37°C	36,31°C	1,86 %
6	22	36°C	34,96°C	2,89 %
7	22	36°C	35,93°C	0,19 %
8	42	36°C	35,15°C	2,36 %
9	49	37°C	36,04°C	1,89 %
10	50	36°C	35,30°C	1,94 %

Hasil Bluetooth dan ThingSpeak

Untuk tampilan hasil dari pengiriman data ke *smartphone* dengan menggunakan *bluetooth* dapat dilihat pada Gambar 11, Pada tampilan data *smartphone* akan berubah datanya sesuai kode pemrograman yang telah dibuat yaitu setiap 100*milisecond*. Untuk tampilan berupa gambar serta nilai yang dikirim dari mikrokontroler.

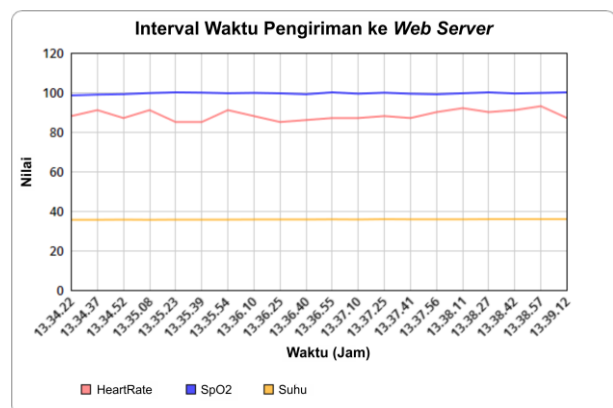


Gambar 11. Tampilan Smartphone Saat Digunakan



Gambar 12. Tampilan Web Server

Hasil pengiriman data dari *smartphone* menuju ke web server disajikan pada Gambar 12. Dimana data akan ditampilkan dengan berupa grafik garis warna merah. Untuk tampilan peta dapat dirubah sesuai posisi pasien yang sedang isolasi mandiri.



Gambar 13. Interval Pengiriman Menuju Server

Sedangkan Gambar 13 memberikan gambaran untuk interval waktu pengiriman data ke server. Dengan cara mengunduh bagian file CSV, didapat interval waktu sebesar <20 detik sekali. Hal ini dipengaruhi oleh cepat atau lambatnya jaringan internet yang ada pada *smartphone* yang digunakan.

PENUTUP Simpulan

Dari penelitian yang dilakukan didapatkan beberapa hasil, dapat diambil beberapa kesimpulan. Hasil *error* yang didapat untuk pengukuran SpO₂ mulai dari nilai terendah 0,11% dan tertinggi 1,20%. Hasil *error* untuk

pengukuran *heartrate* nilai terendah 0,89% dan tertinggi 1,54%. Hasil untuk *error* yang didapat pengukuran suhu badan yaitu terendah 0,19% dan tertinggi 2,78%. Didapatkan hasil untuk pengiriman data dari *smartphone* ke server *ThingSpeak* didapat interval sebesar < 20 detik hal ini bergantung dari jaringan internet pada *smartphone*, serta dapat diunduh datanya dalam bentuk CSV. Sistem ini dapat digunakan dengan baik oleh pasien yang sedang melakukan isolasi mandiri karena memiliki *error* SpO_2 yang rendah, tidak melebihi standar akurasi SpO_2 sebesar $\leq 4\%$.

Saran

Untuk kedepannya dan penelitian selanjutnya dapat dikembangkan lagi. Terutama pada bagian sensor suhu, bisa menggunakan sensor yang memiliki respon cepat, agar tidak menunggu lama dalam pembacaannya serta mengurangi *error* pembacaan. Kemudian untuk server kedepannya dapat membaca juga jika subyek atau pasien sudah dalam keadaan yang darurat, sehingga dapat memberikan *early warning* kepada petugas medis yang memantaunya agar dapat diberikan tindakan medis selanjutnya.

DAFTAR PUSTAKA

- Aithal, Anusha. 2015. *Wireless Sensor Platform for Pulse Oximetry*. Rochester Institute of Technology, 75.
- Ali, Mian Mujtaba; Haxha, Shyqyri; Alam, Munna M; Nwibor, Chike; dan Sakel, Mohammed. 2020. *Design of Internet of Things (IoT) and Android Based Low Cost Health Monitoring Embedded System Wearable Sensor for Measuring SpO_2 , Heart Rate and Body Temperature Simultaneously*. *Wireless Personal Communications*, 111(4), 2449–2463. <https://doi.org/10.1007/s11277-019-06995-7>
- Ambadkar, Sakshi D. dan Nikam, Sobha S. 2020. *NRF Transceiver Based Saline Level, Health Monitoring & Control System*. *International Research Journal of Engineering and Technology (IRJET)*, 07(10), 1789–1795.
- Cao, Hongfei. 2017. *Design of Wireless Physiological Measurement Systems for Patients Affected by COPD*. (Université du Québec à Chicoutimi, 2017). 76.
- Hadiyoso, Sugondo; Rizal, Achmad; dan Magdalena, Rita. 2011. *Monitoring Photoplethysmograph Digital dengan Wireless LAN (802.11b)*. Konferensi Nasional Sistem dan Informatika 2011; Bali, November 12, 2011, 5.
- Hamdala; Rahayu, Yusnita; dan Yasri, Indra. 2015. *Sistem Monitoring Photoplethysmograph Berbasis Zigbee dan Labview*. *jom FTEKNIK*, vol. 2, no. 2, 1–10.
- Kementerian Kesehatan RI. 2020. *Pedoman Kesiapsiagaan Menghadapi Infeksi Novel Coronavirus (2019-Ncov)*. Diakses pada 25 juni 2021, dari https://infeksiemerging.kemkes.go.id/download/DOK-UMEN_RESMI_Pedoman_Kesiapsiagaan_nCoV_Indonesia_28_Jan_2020.pdf
- Lopez, Santiago. 2012. *Pulse Oximeter—Fundamentals and Design*. 39.
- Marathe, Sachi; Zeeshan, Dilleas; Thomas, Tanya; dan Vidhya, S. 2019. *A Wireless Patient Monitoring System Using Integrated ECG Module, Pulse Oximeter, Blood Pressure and Temperature Sensor*. 2019 international conference on Vision Towards Emerging trends in Communication and Networking (ViTECoN), 1–4. <https://doi.org/10.1109/ViTECoN.2019.8899541>
- Maxim Integrated. 2018. *High-Sensitivity Pulse Oximeter and Heart-Rate Sensor for Wearable Health*. 32.
- Maxim Integrated. 2019. *Programmable Resolution 1-Wire Digital Thermometer*. 20.
- Niswar, Muhammad; Nur, Muhammad; dan Mappangara, Idar. 2019. *A Low Cost Wearable Medical Device for Vital Signs Monitoring in Low-Resource Settings*. *International journal of electrical and computer engineering (IJECE)*, 9(4), 2321–2327. <https://doi.org/10.11591/ijece.v9i4.pp2321-2327>
- Omar, Monira; Omar, Mahmoud; dan Abdallah, Farouk. 2015. *Merging and Designing Biomedical Devices with Telecommunication, for Creating Tele Pulse Oxi-Meter*. *International Journal of Scientific & Engineering Research*, 6(4), 1006–1011.
- P2 SMTP LIPI. 2016. Diakses pada 15 Juni 2021, *Jaminan Akurasi Pulse Oximeter Melalui Standar dan Pengujian*. dari <http://www.smtp.lipi.go.id/berita441-Jaminan-Akurasi-Pulse-Oximeter-Melalui-Standar-dan-Pengujian.html>
- Sameh, Radwa; Genedy, M; Abdeldayem, A; dan Abdelazeem, M. H. 2020. *Design and Implementation of An SpO_2 Based Sensor for Heart Monitoring Using An Android Application*. *Journal of physics: conference series*, 1447, 012004. <https://doi.org/10.1088/1742-6596/1447/1/012004>
- Umar, Lazuardi; Firmansyah, Irfan; and Setiadi, Rahmondia Nanda. 2018. *Design of Pulse Oximetry Based on Photoplethysmography and Beat Rate Signal Using DS-100 Probe Sensor for SpO_2 Measurement*. 2018 3rd International Seminar on Sensors, Instrumentation, Measurement and Metrology (ISSIMM), 25–29. <https://doi.org/10.1109/ISSIMM.2018.8727725>