



**SISTEM PENDETEKSI KONDISI PARU-PARU BERDASARKAN
SUHU TUBUH, WARNA KUKU, TINGKAT RESPIRASI DAN
KADAR OKSIGEN DALAM DARAH DENGAN METODE K-
NEAREST NEIGHBOR**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun oleh:

Alfatehan Arsyah Baharin

NIM: 165150307111032



**PROGRAM STUDI TEKNIK KOMPUTER
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2020**

PENGESAHAN

SISTEM PENDETEKSI KONDISI PARU-PARU BERDASARKAN SUHU TUBUH, WARNA KUKU, TINGKAT RESPIRASI DAN KADAR OKSIGEN DALAM DARAH DENGAN METODE K-NEAREST NEIGHBOR

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Teknik

Disusun Oleh :
Alfatehan Arsyah Baharin
NIM: 165150307111032

Skripsi ini telah diuji dan dinyatakan lulus pada
3 Juni 2020

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Rizal Maulana S.T., M.T., M.Sc.
NIK: 2016078910091001

Dosen Pembimbing 2



Hurriyatul Fitriyah, S.T., M.Sc.
NIP: 198510012015042003

Mengetahui

Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T., M.T., Ph.D
NIP: 197105182003121001



PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 3 Juni 2020

Alfatehan Arsyah Baharin

NIM: 165150307111032



ABSTRAK

Alfatehan Arsyah Baharin, Sistem Pendeteksi Kondisi Paru-Paru Berdasarkan Suhu Tubuh, Warna Kuku, Tingkat Respirasi, Dan Kadar Oksigen Dalam Darah Dengan Metode K-Nearest Neighbor

Pembimbing : Rizal Maulana S.T., M.T., M.Sc. dan Hurriyatul Fitriyah, S.T., M.Sc

Ilmu pengobatan saat ini agar dapat mendeteksi kondisi paru-paru yang mempunyai penyakit tuberkulosis dan *Pneumonia* ialah dengan *CT-Scan* dan melakukan pengujian tes dahak. Untuk mengkonfirmasi kondisi paru-paru dari tes tersebut membutuhkan waktu yang lama. Dari permasalahan yang ada, diperlukan adanya penelitian untuk pendeteksi kondisi paru-paru, sehingga hasil dari pemeriksaan kondisi paru-paru dapat segera diketahui setelah dilakukannya pemeriksaan. Penyakit tuberkulosis dan *Pneumonia* mempunyai beberapa gejala yang sama, yang nantinya pada sistem pendeteksi kondisi paru-paru akan digunakan sebagai paramater untuk menentukan *output* sistem. Parameter yang digunakan yaitu kadar oksigen dalam darah, warna kuku, suhu tubuh, dan tingkat respirasi. Komponen yang berperan sebagai pengolah data ialah mikrokontroler Arduino Mega dengan menggunakan empat inputan sensor yaitu MAX 30100, TCS 3200, MLX 90614, dan *Flex* sensor. Yang nantinya akan digunakan metode K-NN untuk menentukan keluaran kelas yang terdapat pada sistem, yaitu kelas Normal, TB, dan *Pneumonia*. Terdapat 14 data yang diujikan dengan 30 data latih dalam menentukan keakuratan klasifikasi menggunakan metode K-NN. 1 dari 14 data yang diujikan mengalami ketidaksesuaian kelas yang mengakibatkan tingkat keakuratan bernilai 92,8% dengan K bernilai 3 sebagai nilai K terbaik. Dari 10 kali pengujian sistem, didapatkan waktu rata-rata kecepatan komputasi sebesar 1030,5 ms.

Kata kunci: paru-paru, kuku, kadar oksigen dalam darah, suhu tubuh, tingkat respirasi, K-NN

**ABSTRACT**

Alfatehan Arsyah Baharin, Lung Condition Detection System Based On Body Temperature, Nail Color, Respiration Rate, And Oxygen Levels In The Blood With The K-Nearest Neighbor

Supervisors : Rizal Maulana S.T., M.T., M.Sc. dan Hurriyatul Fitriyah, S.T., M.Sc.

Current medical science in order to detect the condition of the lungs that have tuberculosis and pneumonia is by CT-Scan and sputum testing. To confirm the condition of the lungs from the test requires a long time. From the existing problems, research is needed to detect the condition of the lungs, so that the results of the examination of the condition of the lungs can be immediately known after the examination. Tuberculosis and Pneumonia have some of the same symptoms, which later in the lung condition detection system will be used as a parameter to determine the system output. The parameters used are oxygen levels in the blood, nail color, body temperature, and respiration rate. Components that act as data processors are the Arduino Mega microcontroller by using four sensor inputs, namely MAX 30100, TCS 3200, MLX 90614, and Flex sensors. The K-NN method will be used to determine the class output contained in the system, namely the Normal, TB, and Pneumonia classes. There are 14 data tested with 30 training data in determining the accuracy of classification using the K-NN method. 1 of the 14 data tested experienced a class mismatch which resulted in an accuracy rate of 92.8% with K valued at 3 as the best K value. From 10 times of system testing, an average computational speed of 1030.5 ms is obtained.

Keywords : lungs, nail, oxygen levels in the blood, body temperature, respiration rate, K-NN



DAFTAR ISI

PENGESAHAN.....	ii
PERNYATAAN ORISINALITAS.....	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT.....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xiii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan.....	2
1.4 Manfaat.....	3
1.5 Batasan Masalah.....	3
1.6 Sistematika Pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN.....	5
2.1 Tinjauan Pustaka.....	5
2.2 Dasar Teori.....	7
2.2.1 Paru-Paru.....	7
2.2.2 Penyakit Paru-Paru.....	8
2.2.3 Flex Sensor.....	8
2.2.4 Max 30100.....	9
2.2.5 TCS 3200.....	10
2.2.6 MLX 90614.....	10
2.2.7 LCD 16X2 dengan I2C.....	10
2.2.8 Arduino Mega.....	11
2.2.9 K-NN (K-NEAREST NEIGHBOR).....	12
BAB 3 METODOLOGI PENELITIAN.....	14
3.1 Tipe Penelitian.....	14
3.2 Subjek Penelitian.....	14



DAFTAR TABEL

Tabel 2.1 Tinjauan Pustaka	5
Tabel 4.1 Spesifikasi Arduino Mega	21
Tabel 4.2 Spesifikasi TCS 3200	22
Tabel 4.3 Output Frequency Scaling	22
Tabel 4.4 Tipe Fotodioda	22
Tabel 4.5 Spesifikasi MLX 90614	23
Tabel 4.6 Spesifikasi Flex Sensor	23
Tabel 4.7 Spesifikasi MAX 30100	24
Tabel 4.8 Deskripsi Pin LCD 16X2	25
Tabel 5.1 Koneksi Pin Arduino Mega dengan Pin Sensor TCS 3200	29
Tabel 5.2 Koneksi Pin Arduino Mega dengan Pin Sensor MLX 90614	30
Tabel 5.3 Koneksi Pin Arduino Mega dengan Pin Sensor MAX 30100	30
Tabel 5.4 Koneksi Pin Arduino Mega dengan Pin Flex Sensor dengan OP-AMP ..	30
Tabel 5.5 Koneksi Pin Arduino Mega dengan Pin Push Button	31
Tabel 5.6 Koneksi Pin Arduino Mega dengan Pin I2C LCD 16X2	32
Tabel 5.7 Data Latih Klasifikasi Metode K-NN	36
Tabel 5.8 Hasil Jarak Euclidean	37
Tabel 5.9 Hasil Sorting Klasifikasi K-NN	38
Tabel 5.10 Nilai Jarak Euclidean Lima Teratas	39
Tabel 5.11 Kode Library Program Utama	42
Tabel 5.12 Deklarasi dan Inisialisasi Variabel	42
Tabel 5.13 Kode Trigger Push Button	43
Tabel 5.14 Kode Perhitungan K-NN	44
Tabel 5.15 Kode Hasil Klasifikasi dan Waktu Komputasi	46
Tabel 5.16 Kode Pengambilan Data Sensor TCS 3200	47
Tabel 5.17 Kode Pengambilan R,G,dan B	47
Tabel 5.18 Kode Inisialisasi Library dan Variabel	50
Tabel 5.19 Kode Mendeteksi Detak Jantung	50
Tabel 5.20 Kode Menghitung dan Menampilkan Spo2	51
Tabel 5.21 Kode Pengambilan Data Sensor MLX 90614	51



DAFTAR GAMBAR

Gambar 2.1 (a) Bentuk Paru-Paru dan (b) Paru-Paru Dalam Xray	7
Gambar 2.2 (a) Pneumonia Dan (b) TB	8
Gambar 2.3 Sudut bengkokan Flex sensor	9
Gambar 2.4 Pemasangan Max 30100	9
Gambar 2.5 TCS 3200	10
Gambar 2.6 MLX 90614	10
Gambar 2.7 LCD 16X2	11
Gambar 2.8 I2C	11
Gambar 2.9 Arduino Mega	11
Gambar 3.1 Flow Chart Metodologi Penelitian	16
Gambar 4.1 Bentuk dan Pemakaian Flex Sensor	24
Gambar 5.1 Tampak Atas Purwarupa Sistem	27
Gambar 5.2 Purwarupa Sistem	27
Gambar 5.3 Pemakaian Flex Sensor	28
Gambar 5.4 Skematik Perangkat Keras Sistem	29
Gambar 5.5 Rangkaian Flex Sensor dengan OP- AMP	31
Gambar 5.6 Rangkaian Penguat Diferensial	31
Gambar 5.7 Diagram Alir Perancangan Perangkat Lunak Program Utama	33
Gambar 5.8 Diagram Alir Pengambilan Nilai Sensor	34
Gambar 5.9 Diagram Alir Klasifikasi K-NN	35
Gambar 5.10 Implementasi Purwarupa Sistem Pendeteksi Kondisi Paru-Paru ...	40
Gambar 5.11 Implementasi Perangkat Keras Sistem Pendeteksi Kondisi Paru-Paru	41
Gambar 6.1 Pengambilan Nilai RGB dengan Sensor TCS 3200	59
Gambar 6.2 Pengambilan nilai Suhu Tubuh dengan Sensor MLX90614	60
Gambar 6.3 Pengambilan nilai Spo2 dengan Sensor MAX 30100	62
Gambar 6.4 Pengambilan Tingkat Respirasi dengan Flex Sensor	64
Gambar 6.5 Pengujian Akurasi Klasifikasi K-NN	67



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Dari beberapa organ vital pada manusia, terdapat organ yang dibutuhkan manusia untuk bernapas yaitu paru-paru. Karena paru-paru merupakan organ pada sistem pernapasan manusia dan berhubungan dengan sistem peredaran darah manusia. Tetapi kesadaran masyarakat terutama di Indonesia sangatlah minim terhadap gejala penyakit paru-paru. Dari informasi yang dipaparkan oleh *World Health Organization* (WHO) pada tahun 2006, Indonesia menempati peringkat tiga dengan 539 ribu kasus dan 101 ribu kematian yang terjadi setiap tahun dikarenakan tuberkulosis. Indonesia juga penyumbang kematian terbanyak nomer 6 di dunia yang disebabkan oleh pneumonia berdasarkan informasi yang didapatkan dari *South East Asia Medical Center* (SEAMIC) pada tahun 2001. pneumonia dapat menyebabkan kematian apabila sudah parah dan tidak diobati karena penyakit ini sulit ditemukan dan apabila dites akan memerlukan waktu yang lama untuk mengetahui hasilnya (Yunus & Setyowibowo, 2010).

Pengetahuan medis yang dimiliki tenaga kesehatan untuk mendapati penyakit Tuberkulosis dan *Pneumonia* pada pasien saat ini dengan cara mendengarkan keluhan pasien yang sebagian besar batuk terus menerus dan tak kunjung sembuh. Masalahnya terletak pada sebagian besar masyarakat menganggapnya sebagai batuk biasa dan bukan gejala adanya penyakit pada paru-parunya, yang akan mengakibatkan penyakitnya akan bertambah parah seiring bertambahnya waktu. Kedua dengan *CT-Scan* dan dilanjutkan dengan tes dahak, hasil dari pengujian *CT-Scan* dan tes dahak memerlukan waktu beberapa hari untuk mengkonfirmasi penyakit. Hal itu kurang efisien jika ada pasien yang harus mengkonfirmasi penyakitnya secara cepat, sehingga diperlukan sebuah upaya agar pasien tidak perlu menanti hasil dari tes laboratorium dan segera mendapati hasil kondisi paru-parunya. Terdapat beberapa gejala yang sama antara penyakit tuberkulosis dan *Pneumonia*, beberapa gejala yang sama ialah tingkat respirasi, karena orang yang mempunyai masalah pada paru-paru memiliki tingkat respirasi yang rendah. Begitupun dengan tingkat oksigen dalam darah, karena tidak tersalurkan dengan baik karena paru-paru yang bermasalah. Kemudian dengan suhu tubuh karena beberapa penyakit paru-paru memiliki gejala yang sama yaitu demam. Dari suhu tubuh bisa dideteksi bahwa apakah seseorang memiliki penyakit pada paru-paru. Kemudian dengan warna kuku, karena kuku sering disebut tolak ukur kesehatan. Apabila ada sesuatu yang salah di dalam tubuh, akan segera terlihat dari luar tubuh (Mayer, 2014).

Dari penelitian sebelumnya yang dilakukan (Kurniawan, et al., 2019), telah dikerjakan sebuah sistem yang mengimplementasikan deteksi penyakit pada paru-paru dengan menggunakan dua parameter yaitu nilai RGB dari warna kuku dan suhu tubuh. Dari 12 data yang diujikan menggunakan metode *Naive bayes*, terdapat 1 data yang mengalami ketidaksesuaian kelas dan menjadikan akurasi sistem sebesar 91,6% dengan rata-rata waktu komputasi yang dihasilkan selama



0,69 detik. Dari hasil pengujian tersebut maka peneliti ingin mengembangkan dari penelitian sebelumnya dengan menambahkan parameter baru yaitu tingkat respirasi dan kadar oksigen dalam darah dengan metode K-NN untuk meningkatkan akurasi.

K-Nearest Neighbor atau metode K-NN termasuk dari algoritma *instance based learning*, banyak aplikasi data *mining*, *statistical pattern recognition*, dan *image processing* biasanya menggunakan metode K-NN. Proses klasifikasi dari K-Nearest Neighbor berdasarkan kelas terbanyak dari tetangga terdekat sebanyak nilai K dan akan menghasilkan *Instance* yang baru atau biasanya disebut algoritma *supervise learning*. Algoritma ini bertujuan untuk mengklasifikasikan kelas dari objek yang diuji menggunakan nilai fitur yang terdapat pada setiap data latih yang ada. Algoritma ini cocok digunakan untuk pengklasifikasi kondisi paru paru dikarenakan K-NN mempunyai keunggulan apabila terdapat data training yang *noisy* dan besar, algoritma K-NN dapat mengatasi permasalahan tersebut secara efektif. Nilai dari *instance* yang baru dihasilkan dari nilai prediksi menggunakan klasifikasi *neighborhood* pada algoritma *K-Nearest Neighbor* (Enda, 2015).

Berdasarkan dari latar belakang, maka penulis mengusulkan untuk membuat Sistem Pendeteksi Kondisi Paru-Paru Berdasarkan Kadar Oksigen Dalam Darah, Warna Kuku, Suhu Tubuh, dan Tingkat Respirasi Dengan Metode *K-Nearest Neighbor*. Dengan hasil klasifikasi Paru-Paru normal, Tuberkulosis (TB), *Pneumonia* (radang paru-paru).

1.2 Rumusan Masalah

Didapatkan rumusan masalah pada penelitian yang akan dilakukan berdasarkan latar belakang yang ada sebelumnya sebagai berikut :

1. Bagaimana tingkat akurasi sensor TCS3200, MLX 90614, *Flex Sensor* dan Max 30100?
2. Bagaimana cara mengenali paru-paru normal, Tuberkulosis (TB), *Pneumonia* (radang paru-paru) dari kadar oksigen dalam darah, warna kuku, suhu tubuh, dan tingkat respirasi?
3. Bagaimana tingkat akurasi dan waktu komputasi sistem pendeteksi kondisi paru-paru melalui deteksi kadar oksigen dalam darah, warna kuku, suhu tubuh, dan tingkat respirasi?

1.3 Tujuan

Penelitian ini bertujuan untuk :

1. Mendapati keakuratan dari sensor TCS3200, MLX 90614, *Flex Sensor* dan Max 30100.
2. Mengenali Tuberkulosis (TB), *Pneumonia* (radang paru-paru), dan paru-paru normal dari kadar oksigen dalam darah, warna kuku, suhu tubuh, dan tingkat respirasi.



3. Mendapatkan nilai akurasi dan lama waktu komputasi dari sistem pendeteksi kondisi paru-paru menggunakan kadar oksigen dalam darah, warna kuku, suhu tubuh, dan tingkat respirasi.

1.4 Manfaat

Penelitian ini berfungsi untuk memberi tahu kondisi paru-paru pengguna sehingga dapat melakukan diagnosa awal apakah kondisi paru-paru Tuberkulosis (TB), *Pneumonia* (radang paru-paru), atau normal. Apabila kondisi paru-paru bermasalah akan segera diketahui dan dapat langsung merujuk ke dokter spesialis paru-paru.

1.5 Batasan Masalah

Supaya sistem ini terfokus, ada beberapa batasan pada penelitian, yaitu:

1. Hanya dapat mengklasifikasikan kondisi paru-paru normal, Tuberkulosis (TB), *Pneumonia* (radang paru-paru).
2. Hanya dapat mendeteksi dari suhu tubuh, warna kuku, tingkat respirasi dan kadar oksigen dalam darah.
3. Hanya dapat mendeteksi subjek saat dalam keadaan diam atau tidak ada gerakan.
4. Pengujian dilakukan pada rentang usia remaja sampai lansia.

1.6 Sistematika Pembahasan

Pada sub bab ini bertujuan menjelaskan penelitian secara rinci. Tujuan penggunaan sistematika pembahasan agar dapat dengan mudah dipahami secara alur penelitian.

1. BAB I PENDAHULUAN

Pada bab pertama dari penelitian ini bertujuan pengurai penjelasan mengenai latar belakang atau alasan dilakukannya penelitian ini, masalah yang dirumuskan, manfaat dari dilakukannya penelitian, batasan-batasan agar menjadikan penelitian terfokus, dan sistematika pembahasan

2. BAB II LANDASAN PUSTAKAAN

Bab landasan pustaka bertujuan untuk menjelaskan kajian teori yang dilakukan penelitian serupa sebelumnya sebagai referensi untuk penelitian yang akan dilakukan. Dasar teori yang mendukung penelitian ini antara lain *Flex Sensor*, *Max 30100*, *TCS 3200*, *LM35*, *K-Nearest Neighbor*, *Arduino*

3. BAB III METODOLOGI

Pada bab ketiga penelitian ini bertujuan untuk menjelaskan alur pengerjaan yang akan dilakukan, antara lain mencari studi literatur, menganalisis apa saja yang dibutuhkan sistem, pengolahan data, melakukan perancangan, kemudian akan diimplementasikan berdasarkan rancangan, kemudian menguji dan menganalisis



hasil pengujian yang dilakukan, dan terakhir akan dihasilkan kesimpulan dari dilakukannya penelitian ini.

4. BAB IV ANALISIS KEBUTUHAN

Pada bab keempat penelitian ini akan menganalisis kebutuhan apa saja yang dibutuhkan oleh sistem agar dapat beroperasi. Rincian isi dari bab ini ialah tentang gambaran umum sistem, kebutuhan perangkat keras, kebutuhan fungsional agar sistem dapat beroperasi dan kebutuhan non fungsional.

5. BAB V PERANCANGAN DAN IMPLEMENTASI

Pada bab kelima dari penelitian ini akan membahas secara detail dan jelas rancangan sekaligus penerapan dari "Sistem Pendeteksi Kondisi Paru-Paru Berdasarkan Suhu Tubuh, Warna Kuku, Tingkat Respirasi Dan Kadar Oksigen Dalam Darah Dengan Metode K-Nearest Neighbor". Isi dari bab ini antara lain perancangan setiap *software* dan *hardware*, implementasi setiap *software* dan *hardware*, serta penjelasan dari setiap tahapan coding.

6. BAB VI PENGUJIAN DAN ANALISIS

Pada bab keenam dari penelitian ini akan dilakukan pengujian terhadap setiap sensor sekaligus sistem yang telah diimplementasikan dengan tahapan pengujian fungsional dan pengujian non fungsional. Kemudian dari hasil pengujian dilakukan analisis.

7. BAB VII PENUTUP

Pada bab ketujuh dari penelitian ini akan dihasilkan kesimpulan dari penerapan dan pengujian yang sudah dilakukan serta akan diberikan saran untuk penelitian serupa selanjutnya agar melakukan penelitian lebih baik.

8. DAFTAR PUSTAKA



BAB 2 LANDASAN KEPUSTAKAAN

Pada bab kedua penelitian ini, akan dilakukan pencarian dan analisis dari teori dasar yang berkaitan dengan topik penelitian ini yang bersumber dari penelitian terdahulu yang memiliki topik serupa. Untuk tabel Tinjauan Pustaka dapat dilihat pada Tabel 2.1.

2.1 Tinjauan Pustaka

Tabel 2.1 Tinjauan Pustaka

No	Nama Penulis [Tahun], Judu	Persamaan Penelitian	Perbedaan	
			Penelitian Terdahulu	Rencana Penelitian
1.	(Kurniawan, et al., 2019) <i>Implementasi Pendeteksi Penyakit Paru-Paru Berdasarkan Warna Kuku dan Suhu Tubuh Berbasis Sensor TCS3200 Dan Sensor LM35 dengan Metode Naive Bayes</i>	Sistem melakukan pendeteksian penyakit pada paru-paru.	Sistem menggunakan warna kuku dan suhu tubuh untuk menentukan kondisi paru-paru normal, paru-paru ringan, paru-paru berat dan menggunakan metode naive bayes	Sistem pendeteksi menggunakan tingkat respirasi dan kadar oksigen dalam darah untuk menentukan paru-paru tuberkulosis (TB), <i>Pneumonia</i> (radang paru-paru), paru-paru normal dan menggunakan metode K-NN
2.	(WARDANA, et al., 2017) <i>Spirometer Non-Invasive dengan Sensor Piezoelektrik untuk Deteksi Kesehatan Paru-Paru</i>	Mengukur tingkat respirasi menggunakan sensor piezoelektrik untuk mendeteksi kondisi paru-paru	Hanya menggunakan sensor piezoelektrik untuk mengukur tingkat respirasi dan kondisi paru-paru	Menggunakan suhu tubuh, warna kuku, tingkat respirasi, kadar oksigen dalam darah untuk mendapatkan kondisi paru-paru



3.	(Setyobudi, et al., 2019) <i>Sistem Deteksi Gejala Hipoksia Berdasarkan Saturasi Oksigen dan Detak Jantung Menggunakan Metode Fuzzy Berbasis Arduino</i>	Mengukur tingkat kadar oksigen dalam darah	Menggunakan tingkat saturasi oksigen dalam darah untuk mendeteksi gejala <i>hipoksia</i>	Menggunakan kadar oksigen dalam darah untuk menentukan kondisi paru-paru
----	--	--	--	--

Berdasarkan sumber pertama yaitu penelitian yang dilakukan oleh (Kurniawan, et al., 2019), dibuat pengimplementasian suatu sistem untuk mendeteksi penyakit paru-paru berdasarkan parameter warna kuku dan suhu tubuh serta menggunakan algoritma *Naive Bayes*. Untuk mengambil nilai warna kuku menggunakan sensor TCS 3200, sedangkan untuk pengambilan nilai suhu tubuh menggunakan sensor LM35. Setelah didapatkan nilai RGB dari warna kuku dan suhu tubuh, nilai tersebut akan diolah Arduino Uno sehingga sistem mengeluarkan *output* hasil penyakit paru-paru menggunakan algoritma *Naive Bayes*. Apakah hasil klasifikasi paru-paru normal, paru-paru ringan, atau paru-paru berat. Hasil pengujian sensor warna TSC3200 memiliki tingkat *error* sebesar 1,478% dan tingkat *error* pembacaan sensor suhu LM35 terhadap pembacaan termometer sebesar 1,13%. Hasil akurasi sistem sebesar 91,6% atau 1 dari 14 data yang mengalami ketidaksesuaian kelas dan waktu komputasi rata-rata selama 0.69 detik selama 10 kali pengujian. Kekurangan dari penelitian ini sedikitnya parameter yang digunakan.

Berdasarkan sumber kedua yaitu penelitian yang dilakukan oleh (WARDANA, et al., 2017), penelitian ini mengukur tingkat respirasi dan kapasitas vital paru-paru menggunakan sensor piezoelektrik yang diletakkan di dada. Pada sensor yang digunakan terdapat perubahan tekanan berkisar 10-80 mV, jadi dibutuhkan rangkaian filter, *calmper*, *amplifier*, dan mikrokontroler AVR AT Mega 32 sebagai pengolah data *input*, *SD card* untuk menyimpan data, dan *LCD* untuk menampilkan grafik hasil ukur. Hasil dari pengujian menunjukkan bahwa persentase nilai keberhasilan alat adalah 95,70 %, kekurangan dari penelitian ini hanya mendeteksi apakah kondisi paru-paru sehat atau tidak. Apabila hasil pengukuran kapasitas paru-paru kurang dari 80%, maka akan dinyatakan paru-paru tidak sehat. Kekurangan dari penelitian ini tidak adanya metode untuk menentukan hasil kondisi paru paru. Kelebihan dari penelitian ini sistem berukuran kecil dan dapat digunakan dengan mudah, pada *LCD* akan ditampilkan hasil dari pengukuran serta pendeteksian kondisi paru-paru.

Berdasarkan sumber ketiga yaitu penelitian yang dilakukan oleh (Setyobudi, et al., 2019), dibuat suatu alat deteksi gejala awal *hipoksia* menggunakan metode



noninvasive yaitu tanpa melukai pasien dengan menjepitkan sensor MAX 30100 diujung jari. Metode *fuzzy* digunakan untuk mendeteksi gejala awal *hipoksia* dan *Arduino UNO* sebagai pengolah data inputan yang nantinya akan dikirim ke *smartphone* menggunakan koneksi *bluetooth* untuk melihat data. Hasil dari pengujian ini didapatkan hasil *error* pada alat saturasi oksigen sebesar 2,96% dan 2,86% untuk detak jantung, sedangkan hasil akurasi dari metode *fuzzy* dengan 12 percobaan data sebesar 100%. Kelebihan dari penelitian ini alat dapat digunakan dengan mudah dan hasil dari pengukuran tingkat respirasi dan tingkat saturasi dapat diketahui dari tampilan *smartphone*. Kekurangan dari penelitian ini variasi inputan himpunan *fuzzynya* kurang serta dimensi alat yang terlalu besar.

Berdasarkan ketiga penelitian diatas, maka dijadikan referensi untuk mengembangkan penelitian sebelumnya dengan kelemahan yang dapat diatasi. Kelemahan tersebut ialah kurangnya parameter dalam mendeteksi kondisi paru-paru, tidak dapat menentukan penyakit secara spesifik. Oleh karena itu, pada penelitian akan digunakan data pasien penyakit Tuberkulosis dan Pneumonia yang didapatkan langsung dari ahli atau pakar dari bidang penyakit paru. Dan akan dibuat pendeteksi kondisi paru-paru dan mengklasifikasikannya menjadi normal, tuberkulosis (TB), Pneumonia (radang paru-paru) dengan menggunakan Suhu Tubuh, Warna Kuku, Tingkat Respirasi Dan Kadar Oksigen Dalam Darah.

2.2 Dasar Teori

Berdasarkan sub bab ini akan membahas mengenai teori sebagai pendukung penelitian antara lain sebagai berikut :Paru-paru, Penyakit paru-paru, *K-Nearest Neighbor*, *Flex Sensor*, *Max 30100*, *TCS3200*, *MLX 90614*, *Arduino Mega*.

2.2.1 Paru-Paru



(a)

(b)

Gambar 2.1 (a) Bentuk Paru-Paru dan (b) Paru-Paru Dalam Xray

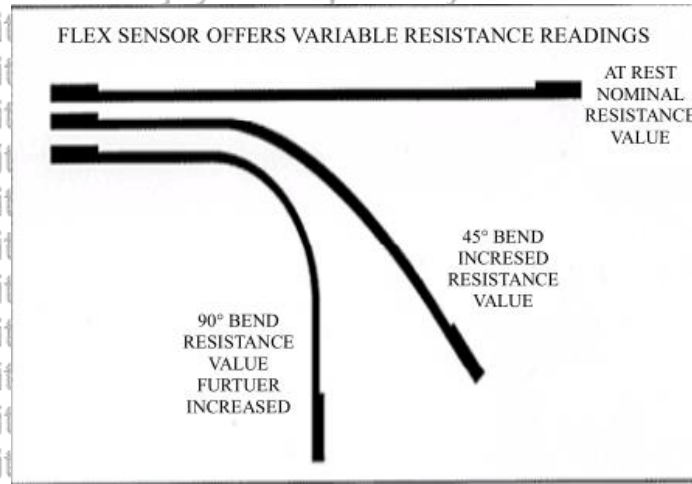
Sumber: i2-prod.cambridge-news.co.uk

Sumber: chestx-ray.com

Ada beberapa organ vital pada manusia, salah satunya adalah paru-paru. Fungsi utama paru-paru adalah satu satunya alat pernapasan bagi manusia. Ketika menarik napas, kita akan memasukkan oksigen kedalam paru-paru, dan ketika mengeluarkan napas akan mengeluarkan karbondioksida dari paru-paru kita. Tingkat populasi yang semakin memburuk apalagi di daerah kota-kota besar akan



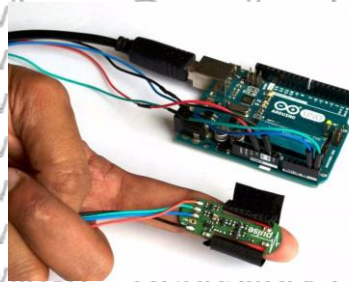
tersebut melengkung, sensor mengeluarkan keluaran resistansi berkorelasi dengan jarak lengkungan yang terdapat pada Gambar 2.3 (Foxlin & Naimark, 2003). Semakin melengkung, semakin besar juga nilai resistansinya.



Gambar 2.3 Sudut bengkokan Flex sensor
Sumber: (Syed, et al., 2012)

2.2.4 Max 30100

Max 30100 adalah modul sensor untuk memonitor denyut jantung dan Spo₂ dengan menggabungkan dua LED, photodetektor, optik yang dioptimalkan. Untuk mendeteksi denyut jantung dan kadar oksigen dalam darah, digunakan pemrosesan sinyal rendah dengan cara *non-invasive* atau tanpa melukai pasien. Data sensor akan diolah apabila sensor disambungkan dengan mikrokontroler. Digunakannya algoritma K-NN untuk megolah data sensor yang didapat sebelumnya (Setyobudi, et al., 2019). Cara pemasangan alat MAX 30100 bisa dilihat pada Gambar 2.4.



Gambar 2.4 Pemasangan Max 30100

Sumber: <https://hackster.imgix.net>



2.2.5 TCS 3200

TCS 3200 merupakan sensor warna yang dapat dipogram dari cahaya ke frekuensi yang menggabungkan konfigurasi fotodiode silikon dan arus ke frekuensi converter pada sirkuit CMOS monolitik. Pada TCS 3200, terdapat 16 fotodiode dari filter merah, hijau, biru dan tanpa filter, yang nantinya cahaya dari fotodiode akan dikonversi ke frekuensi menggunakan 8×8 array (TAOS, n.d.). Gambar 2.5 memperlihatkan TCS 3200.



Gambar 2.5 TCS 3200

Sumber: <https://cdn3.volusion.com>

2.2.6 MLX 90614

Sensor ini terdiri dari *chip* detektor yang peka terhadap suhu berbasis inframerah dan pengkondisi sinyal yang terintegrasi dengan TO-39. Sensor ini didukung ADC 17-bit, unit DSP dan termometer yang memiliki akurasi dan resolusi tinggi. Output digital dari PWM dan SMBus terkalibrasi dengan termometernya. Sensor ini dapat mengukur perubahan suhu secara terus menerus dengan rentang -40°C hingga 120°C dan jangkauan suhu objek berkisar -70°C hingga 380°C dengan resolusi output $0,14^{\circ}\text{C}$ (Sibuea, 2018). Gambar sensor suhu MLX 90614 dapat dilihat pada Gambar 2.6.



Gambar 2.6 MLX 90614

Sumber: <https://www.melexis.com>

2.2.7 LCD 16X2 dengan I2C

LCD (Liquid Crystal Display) merupakan sebuah modul komponen tampilan elektronik yang lebih bagus daripada *seven segments* dan *LED multi segments*

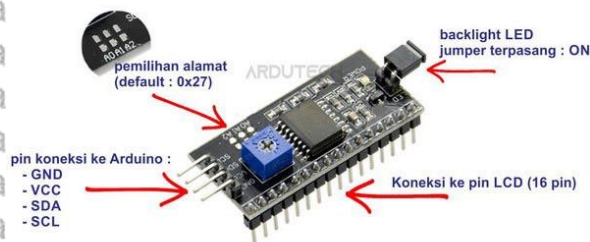


karena lebih murah, pemrogramannya tidak sulit, dan dapat menampilkan 32 karakter yang terdiri dari 16 karakter pada tiap barisnya (Kurniawan, et al., 2019). Untuk mengoperasikan LCD 16X2 dibutuhkan mikrokontroler untuk pengoperasiannya. Pada mikrokontroler akan diberikan program untuk mengoperasikan LCD 16x2 agar dapat menampilkan suatu karakter. Kemudian adanya modul I2C yang digunakan pada LCD 16X2. Digunakannya modul I2C bertujuan untuk menghemat pin Arduino. Untuk gambar LCD 16x2 dapat dilihat pada Gambar 2.7 dan gambar I2C dapat dilihat pada Gambar 2.8



Gambar 2.7 LCD 16X2

Sumber: (Kurniawan, et al., 2019)



Gambar 2.8 I2C

Sumber: <https://www.ardutech.com/lcd-i2c-dengan-arduino/>

2.2.8 Arduino Mega



Gambar 2.9 Arduino Mega

Sumber: www.theengineeringprojects.com

Arduino Mega merupakan mikrokontroler berbasis ATmega2560, Arduino Mega mempunyai *memory space* dan pin *input output* lebih banyak dari pada mikrokontroler yang ada di pasaran. Terdapat 54 pin digital *input* maupun *output* dan 15 diantaranya digunakan untuk pin *PWM*, 16 pin analog, osilator kristal



berfrekuensi 16Mhz, kabel *USB* untuk menyambungkan dan mentransfer program dari komputer ke mikrokontroler, *jack power*, *ICSP header*, 4 buah *UART*. Arduino Mega dirancang khusus untuk proyek-proyek yang membutuhkan sirkuit yang kompleks dan lebih banyak ruang memori (Aqeel, 2018). Gambar Arduino Mega dapat dilihat pada Gambar 2.9.

2.2.9 K-NN (K-NEAREST NEIGHBOR)

Algoritma K-NN adalah metode untuk menentukan objek baru berdasarkan tetangga terdekat dari objek tersebut berdasarkan data pembelajaran. Dalam mencari K data *training*, akan dicari dengan jarak yang paling dekat dengan data uji. Jarak antara data uji dan data *training* akan dihitung dengan antara titik pada data uji ke semua titik pada data *training* dengan rumus *Euclidean Distance* (Musa &, 2017).

Metode K-NN memiliki keunggulan yaitu apabila jumlah data yang digunakan banyak dan *noisy*, metode K-NN tangguh terhadap hal tersebut. Karena jumlah data pada penelitian ini sebanyak 30 data, maka metode K-NN cocok digunakan pada penelitian ini. Sehingga digunakannya metode K-NN agar dapat menghasilkan nilai akurasi yang tinggi.

Pada metode K-NN, digunakan jarak *Euclidean* untuk pengklasifikasiannya. Jarak *Euclidean* atau *Euclidean Distance* adalah algoritma yang digunakan untuk mencari jarak terdekat antara data uji dengan data latih yang tersedia. Untuk persamaan *Euclidean Distance* dapat dilihat di **Persamaan 2.1**.

$$d = \sqrt{|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + |x_{i3} - x_{j3}|^2 + |x_{i4} - x_{j4}|^2 + |x_{i5} - x_{j5}|^2 + |x_{i6} - x_{j6}|^2 \dots} \quad (2.1)$$

Keterangan :

d = nilai jarak *Euclidean*

x_{i1} = nilai pada fitur R data uji

x_{j1} = nilai pada fitur R data latih

x_{i2} = nilai pada fitur G data uji

x_{j2} = nilai pada fitur G data latih

x_{i3} = nilai pada fitur B data uji

x_{j3} = nilai pada fitur B data latih

x_{i4} = nilai pada fitur Spo2 data uji

x_{j4} = nilai pada fitur Spo2 data latih

x_{i5} = nilai pada fitur Suhu data uji

x_{j5} = nilai pada fitur Suhu data latih

x_{i6} = nilai pada fitur Tingkat Respirasi data uji

x_{j6} = nilai pada fitur Tingkat Respirasi data latih



Berikut adalah cara menggunakan metode K- Nearest Neighbors :

1. Menentukan nilai K yang akan diambil sebagai hasil dari pengklasifikasian
2. Menghitung *Euclidean Distance* setiap data uji ke data latih yang tersedia.
3. Mengurutkan perhitungan *Euclidean Distance* dari yang terkecil
4. Mengumpulkan label *class* dengan mengambil data tetangga yang terdekat sebanyak K.



BAB 3 METODOLOGI PENELITIAN

Terdapat runtunan kerja yang dilakukan pada penelitian ini yang terdiri dari, tipe penelitian, subjek penelitian, lokasi penelitian, teknik pengumpulan data, peralatan pendukung, metode, perancangan dan implementasi, dan kesimpulan.

3.1 Tipe Penelitian

Tipe penelitian yang digunakan pada penelitian ini adalah impelementatif pengembangan. Implementatif menerapkan pemakaian langsung alat yang diterapkan pada subjek, meliputi *hardware* dan *software*. Pengembangan merupakan hasil dari pengembangan dari penelitian sebelumnya.

3.2 Subjek Penelitian

Subjek yang digunakan pada penelitian ini yaitu orang yang mempunyai penyakit paru-paru tuberkulosis, *pneumonia*, dan orang yang mempunyai paru-paru normal. Karena sistem ini akan digunakan untuk mendeteksi kondisi paru-paru seseorang dengan memanfaatkan warna kuku, kadar oksigen dalam darah, suhu tubuh, dan tingkat respirasi.

3.3 Lokasi Penelitian

Lokasi penelitian dilakukan di Puskesmas Janti, Kota Malang dan Fakultas Ilmu Komputer, Universitas Brawijaya. Penelitian dilakukan dengan alat yang sudah disediakan.

3.4 Teknik Pengumpulan Data

Pengumpulan data dilakukan dengan cara mengambil langsung dengan alat yang sudah jadi. Proses pengumpulan data dilakukan dengan meletakkan jari jempol subjek pada lubang di alat yang sudah disediakan untuk diukur kadar oksigen dalam darah dan warna kuku, kemudian dahi subjek akan ditembakkan sinar inframerah untuk diukur suhunya, yang terakhir subjek akan diikatkan sabuk berbahan karet yang telah terpasang flex sensor untuk diukur tingkat pernapasannya. Karakter dari subjek yang diukur memiliki usia dengan rentang 18-60 tahun dengan mempunyai penyakit tuberkulosis dan pneumonia tanpa adanya batasan gender. Lokasi pengambilan data bertempat di Puskesmas Janti, Kota Malang dengan bantuan Ibu Jukhriyah, Amd.Kep untuk memastikan apakah subjek tersebut mempunyai penyakit tuberkulosis atau pneumonia.

3.5 Peralatan Pendukung

Peralatan pendukung merupakan peralatan yang kaan digunakan pada penelitian ini. Tidak adanya peralatan pendukung mengakibatkan sistem tidak dapat berjalan seperti yang diharapkan. Peralatan pendukung yang digunakan meliputi *hardware* dan *software* ialah:



Hardware/perangkat keras:

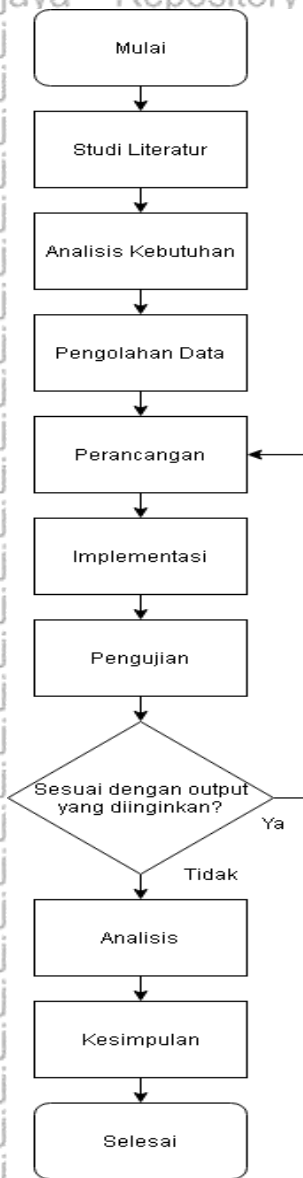
1. Mikrokontroler Arduino Mega
2. MAX 30100
3. TCS 3200
4. MLX 90614
5. Flex Sensor
6. Push Button
7. Kabel Jumper
8. LCD 16x2

Software/perangkat lunak:

1. Arduino IDE
2. Library

3.6 Metodologi Penelitian

Pada sub bab ini akan dijelaskan alur jalannya pengerjaan penelitian ini. Alur jalannya pengerjaan penelitian ini dapat dilihat pada Gambar 3.1.



Gambar 3.1 Flow Chart Metodologi Penelitian

3.6.1 Studi dan Pengkajian Literatur

Diperlukan dilakukannya studi literatur pada setiap penelitian untuk menjadi referensi pada tahap perancangan dan implementasi sistem serta digunakan untuk menguatkan teori. Studi literatur dapat ditemukan pada paper, website resmi dan jurnal. Ruang lingkup studi literatur penelitian ini meliputi :

1. Pendeteksi Kondisi paru-paru
2. Penggunaan MAX 30100, TCS 3200, MMLX 90614, dan Flex sensor
3. Penggunaan metode K-NN

3.6.2 Rekayasa Persyaratan

Pada sub bab ini akan dijelaskan rekayasa persyaratan yang bertujuan agar sistem berfungsi sesuai yang diharapkan. Dibutuhkan beberapa rekayasa



persyaratan pada penelitian ini meliputi manfaat dan tujuan dari pembuatan sistem, lingkungan kerja sistem yang digunakan, karakteristik user, analisis kebutuhan, batasan perancangan sistem, kebutuhan komunikasi, kebutuhan fungsional, kebutuhan non-fungsional, kebutuhan *hardware* dan *software*, dan *output* dari sistem.

3.7 Perancangan Sistem

Tipe penelitian ini adalah implementatif pengembangan. Penulis mengklasifikasikan teknik perancangan sistem pada perangkat keras meliputi :

3.7.1 Perancangan Sistem Perangkat Keras

Perancangan sistem perangkat keras ialah tahapan yang dikerjakan setelah analisis kebutuhan sistem. Perancangan dan implementasi dapat dilakukan apabila seluruh kebutuhan sistem sudah terpenuhi agar dapat menjalankan tujuan sistem seperti yang diharapkan pada deteksi warna kuku, kadar oksigen dalam darah, suhu tubuh, dan tingkat respirasi. perangkat keras yang digunakan adalah mikrokontroler Arduino Mega, MAX 30100, TCS 3200, MLX 90614, dan Flex Sensor.

3.7.2 Perancangan Sistem Perangkat Lunak

Perancangan sistem perangkat lunak ialah tahapan yang dikerjakan untuk melakukan perancangan algoritma atau metode yang akan digunakan dalam runtutan kerja sistem. Perancangan dapat dilakukan apabila seluruh kebutuhan sistem sudah terpenuhi. Perancangan perangkat lunak sendiri berisi tentang langkah-langkah yang akan dijalankan oleh sistem, yang menjadikan perangkat lunak merupakan bagian penting dalam sistem.

3.8 Implementasi Sistem

Implementasi akan dilakukan apabila seluruh perancangan software dan hardware terlaksana. Implementasi berupa susunan hardware dan menggunakan metode K-NN dalam penentuan hasil kondisi paru-paru yang diderita pasien. Tahapan dari implementasi ialah :

1. Implementasi sistem pada orang yang mempunyai penyakit paru-paru tuberkulosis, *pneumonia*, maupun yang tidak punya penyakit pada paru-paru.
2. Implementasi perangkat keras terdiri dari mikrokontroler Arduino Mega, MAX 30100, TCS 3200, MLX 90614, dan Flex Sensor.
3. Implementasi perangkat lunak menggunakan Arduino IDE untuk memprogram dan metode K-NN berfungsi untuk pengklasifikasian kelas dari *output* sistem.

Ketika tahapan dari implementasi sudah dilaksanakan, diharapkan sistem dapat berfungsi sebagai berikut :

1. Dapat membedakan warna kuku, kadar oksigen dalam darah, suhu tubuh, dan tingkat respirasi pada orang normal, orang berpenyakit tuberkulosis dan *pneumonia*.



2. Dapat mengeluarkan *output* yang sesuai.

3.9 Pengujian dan Analisis

Pengujian dan analisis sistem dilakukan untuk membuktikan kesesuaian kerja sistem dengan rancangan. Berikut merupakan skenario pengujian sistem :

1. Pengujian MAX 30100, TCS 3200, MLX 90614, dan Flex Sensor sebagai input sistem
 - a) Mendapatkan nilai kadar oksigen dalam darah, warna kuku, suhu tubuh, dan tingkat respirasi.
2. Pengujian metode K-NN sebagai pengklasifikasi kelas dari *output* sistem.
 - a) sistem mengeluarkan *output* sistem pada LCS 16x2 berupa "Normal", "TB", dan "Pneumonia".

Setelah selesai melakukan pengujian, akan memaparkan analisis yang didapatkan saat melakukan pengujian. Analisis berfungsi untuk menganalisa hasil klasifikasi dan *output* sistem tersebut dibandingkan dengan referensi kelas yang diberikan oleh verifikator pada Puskesmas Janti, Kota Malang.

3.10 Kesimpulan

Kesimpulan dapat ditarik setelah semua tahapan rekayasa persyaratan, perancangan, implementasi, pengujian dan analisis terpenuhi. Untuk itu, kesimpulan adalah tahapan akhir yang dilakukan peneliti pada penelitian. Dalam penarikan kesimpulan, didasarkan pada tercapai tidaknya tujuan yang telah ditentukan dan poin penting yang bersangkutan.



BAB 4 ANALISIS KEBUTUHAN

Pada bab ini akan menjelaskan terkait gambaran umum sistem, lingkungan operasional sistem, analisis kebutuhan yang terdiri dari kebutuhan fungsional, non-fungsional, perangkat keras, dan perangkat lunak, dan kemudian akan menjelaskan tentang batasan *design* sistem.

4.1 Gambaran Umum Sistem

Sistem pendeteksi kondisi paru-paru berdasarkan kadar oksigen dalam darah, warna kuku, suhu tubuh, dan tingkat respirasi dengan metode *K-Nearest Neighbor* yaitu suatu sistem yang dapat mengklasifikasikan kondisi paru-paru. Adapun nilai dari warna kuku didapatkan oleh sensor TCS 3200, nilai suhu tubuh didapatkan oleh sensor MLX 90614, nilai tingkat respirasi didapatkan oleh *Flex Sensor*, sedangkan nilai kadar oksigen dalam darah didapatkan oleh sensor MAX 30100.

Metode *K-Nearest Neighbor* bertujuan untuk mengklasifikasikan hasilnya dari keempat fitur yang sudah didapatkan dan dapat dibaca pada LCD 16x2. Apabila tidak ada alat untuk memeriksa kondisi paru-paru, seorang dokter dapat menggunakan sistem ini sebagai alat bantu untuk mendiagnosis awal penyakit yang diderita pasien. Selain itu metode *K-Nearest Neighbor* ini cocok digunakan untuk pengklasifikasi kondisi paru paru karena K-NN mempunyai beberapa kelebihan, yaitu mampu mengatasi apabila terdapat data latih yang *noisy* dan efektif apabila mempunyai data latih yang banyak. Klasifikasi *neighborhood* digunakan sebagai nilai estimasi dan nilai objek yang baru pada Algoritma *K-Nearest Neighbor*.

4.2 Lingkungan Operasional Sistem

Lingkungan Operasional Sistem dari Sistem Pendeteksi Kondisi Paru-Paru berada di Fakultas Ilmu Komputer Universitas Brawijaya dan Puskesmas Janti, Kota Malang.

4.3 Analisis Kebutuhan

Sub bab ini bertujuan untuk menentukan kebutuhan sistem agar sistem dapat berjalan sesuai ekspektasi. Analisis sistem yang akan diuraikan pada bagian ini adalah kebutuhan fungsional dan non fungsional.

4.3.1 Kebutuhan Fungsional

Berikut ini adalah kebutuhan fungsional yang harus mampu dilakukan oleh sistem.

1. Sensor warna berkemampuan membaca nilai RGB dari warna kuku
Sensor warna TCS 3200 berfungsi mendeteksi nilai pada warna kuku. Nilai warna kuku yang dideteksi oleh sensor berupa nilai frekuensi dari fotodiode merah, hijau, dan biru yang terdapat pada sensor TCS 3200.



2. Sensor suhu berkemampuan membaca suhu tubuh
Sensor suhu MLX 90614 berfungsi untuk mendeteksi suhu tubuh seseorang. Nilai yang didapat saat mengukur suhu tubuh berupa nilai derajat celsius.
3. Sensor *Flex* berkemampuan membaca nilai digital tingkat respirasi
Sensor *Flex Sensor* berfungsi untuk mendeteksi adanya tingkat respirasi dan memperoleh nilai digital saat pasien mengambil napas dalam-dalam dan menghembuskan nafas dalam dalam.
4. Sensor MAX 30100 dapat membaca kadar oksigen dalam darah
Sensor MAX 30100 berfungsi untuk mendeteksi kadar oksigen dalam darah dan memperoleh nilai dari kadar oksigen dalam darah pasien. Nilai yang dihasilkan berupa nilai persen.
5. K-NN berkemampuan untuk mengolah data dari sensor untuk menentukan kondisi paru-paru
Bagian ini bertujuan untuk melakukan pengklasifikasian menggunakan *K-Nearest Neighbor* dengan nilai dari keempat sensor dan data latih yang sebelumnya sudah didapatkan.
6. LCD 16x2 berkemampuan menampilkan hasil dari klasifikasi kondisi paru-paru
Bagian ini bertugas untuk menampilkan nilai yang didapat dari masing-masing sensor serta dapat menampilkan kondisi paru-paru apakah normal, tb, atau pneumonia.

4.3.2 Kebutuhan Non-Fungsional

Kebutuhan Non-Fungsional ialah kebutuhan sistem yang berkaitan dengan perangkat keras maupun perangkat lunak. Sistem ini mempunyai kebutuhan perangkat keras sebagai berikut:

4.3.2.1 Kebutuhan Perangkat Keras

Keperluan dari perangkat keras agar sistem dapat beroperasi sebagai berikut:

1. Mikrokontroler Arduino Mega
Mikrokontroler berfungsi mengolah data yang didapat dari setiap sensor dan pengatur dari berjalannya sistem melalui *push button*. Pada sistem pendeteksi kondisi paru-paru, Mikrokontroler Arduino Mega akan menerima inputan dari beberapa output sensor yaitu sensor MLX 90614, TCS 3200, MAX 30100, dan *Flex Sensor*. Kemudian



dengan menggunakan metode K-NN, Mikrokontroler Arduino Mega akan mengolah data yang didapat dari sensor. Untuk mengolah data menggunakan metode K-NN dan data latih yang banyak, maka diperlukan sebuah mikrokontroler yang mempunyai spesifikasi tinggi dengan pin input output yang banyak. Oleh karena itu, pada Sistem Pendeteksi Kondisi Paru-Paru menggunakan Arduino Mega untuk mengolah data dan pengklasifikasian dengan metode K-NN. Untuk spesifikasi lengkap Arduino Mega dapat dilihat pada Tabel 4.1.

Tabel 4.1 Spesifikasi Arduino Mega

Mikrokontroler	ATMega2560
Operating Voltage	5V
Input Voltage (Recommend)	7-12V
Input Voltage (Limits)	6-20V
Digital I/O Pin	54 (14 diantaranya bisa dipakai sebagai PWM output)
Analog Input Pin	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

2. Sensor Warna TCS 3200

Sensor TCS 3200 merupakan sensor yang berfungsi pendeteksi warna kuku pasien. Dalam pengoperasiannya, dibutuhkan sebuah mikrokontroler untuk memprogram sensor dan agar sensor dapat berfungsi. Pada sistem ini digunakan Mikrokontroler Arduino Mega dalam pengoperasian sensor TCS 3200. Pada sensor TCS 3200, terdapat 4 filter yang digunakan untuk mendeteksi warna yaitu filter merah, filter biru, filter hijau, dan tanpa filter yang dapat dilihat pada



Tabel 4.4. Pada Sitem Pendeteksi Kondisi Paru-Paru digunakan TCS 3200 dibandingkan kamera karena yang dibutuhkan hanya untuk pendeteksi warna. Apabila warna berhasil dideteksi oleh sensor TCS 3200, maka hasil deteksi tersebut akan ditampilkan pada serial monitor dengan menunjukkan nilai warna RGB dengan range nilai 0-255. Keluaran dari TCS 3200 akan *discaling* dengan mengubah nilai *High* atau *Low* pada pin S0 dan S1 yang dapat dilihat pada Tabel 4.3. Adapun spesifikasi pada TCS 3200 dapa dilihat pada Tabel 4.2.

Tabel 4.2 Spesifikasi TCS 3200

Tegangan Suplai	2,7 V – 5,5 V
Tegangan Input High-level	2 V – 5,5 V
Tegangan Input Low-level	0 V – 0,8 V
Kisaran suhu beroperasi	-40°C – 70°C

Tabel 4.3 Output Frequency Scaling

SO	S1	Output Frequency Scaling
L	L	<i>Power Down</i>
L	H	2%
H	L	20%
H	H	100%

Tabel 4.4 Tipe Fotodioda

S2	S3	Tipe Fotodioda
L	L	Merah
L	H	Biru
H	L	Tanpa Filter
H	H	Hijau

3. Sensor suhu MLX 90614

Sensor MLX 90614 merupakan sensor yang berfungsi pendeteksi suhu badan dari pasien. Dalam pengoperasiannya, dibutuhkan sebuah mikrokontroler untuk memprogram sensor dan agar sensor dapat berfungsi. Pada sistem ini digunakan Mikrokontroler Arduino Mega dalam pengoperasian sensor MLX 90614. Pada sensor MLX 90614, cara pendeteksian suhu badan hanya menembakkan sinyal infrared ke pasien dan tidak perlu kontak langsung dengan pasien. Pada Sistem Pendeteksi Kondisi Paru-Paru digunakan MLX 90614 dibandingkan



menggunakan LM35 karena pada MLX 90614 hanya menembakkan sinyal infrared ke pasien dan tidak perlu kontak langsung dengan pasien dan juga memudahkan penggunaan ke pasien. Oleh karena itu, di penelitian Sistem Pendeteksi Kondisi Paru-Paru menggunakan sensor MLX 90614 untuk mendeteksi suhu badan dari pasien. Adapun spesifikasi dari sensor MLX 90614 dapat dilihat pada Tabel 4.5.

Tabel 4.5 Spesifikasi MLX 90614

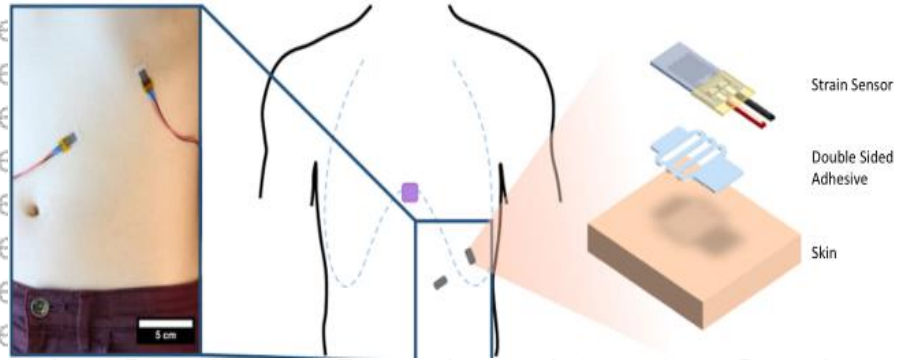
Tegangan Suplai (Tegangan Lebih)	7 V
Tegangan Suplai (Operasi)	5,5 V
Tegangan Balik	0,4 V
Kisaran Suhu Beroperasi	-40°C...+85°C
Kisaran Suhu Penyimpanan	-40°C...+125°C

4. Flex Sensor

Flex sensor merupakan sensor yang berfungsi pendeteksi tingkat respirasi pada pasien. Dalam pengoperasiannya, dibutuhkan sebuah mikrokontroler untuk memprogram sensor dan agar sensor dapat berfungsi. Pada sistem ini digunakan Mikrokontroler Arduino Mega dalam pengoperasian *Flex sensor*. Pada *Flex sensor*, cara pendeteksian tingkat respirasi dengan menempatkan *Flex sensor* di dada pasien untuk mengukur tingkat respirasi saat menghirup napas dan menghembuskan napas. Pada Sistem Pendeteksi Kondisi Paru-Paru digunakan *Flex sensor* dibandingkan *Flex Sensor* karena pada *Flex sensor* perubahan resistansi yang dihasilkan terlihat jelas saat pemakaian. Adapun bentuk dan pemakaian dari *Flex sensor* dapat dilihat pada Gambar 4.1 dan spesifikasinya dapat dilihat pada Tabel 4.6.

Tabel 4.6 Spesifikasi Flex Sensor

Resistansi Datar	25K Ohms
Toleransi Resistansi	±30%
Rentang Resistansi Lengkungan	45K – 125K Ohms
Kisaran Suhu Beroperasi	-35°C - +80°C
Daya	0.50 Watt – 1 Watt



Gambar 4.1 Bentuk dan Pemakaian Flex Sensor

Sumber: (Chu, et al., 2019)

5. Sensor MAX30100

Sensor MAX30100 berfungsi sebagai pendeteksi kadar oksigen dalam darah pada pasien penderita penyakit paru-paru. Dalam pengoperasiannya, dibutuhkan sebuah mikrokontroler untuk memprogram sensor dan agar sensor dapat berfungsi. Pada sistem ini digunakan Mikrokontroler Arduino Mega dalam pengoperasian sensor MAX30100. Pada sensor MAX30100, cara pendeteksian kadar oksigen dalam darah dengan menempatkan sensor MAX30100 di ujung jari tangan dan sensor akan menembakan infra merah untuk mengetahui kadar oksigen. Hasil dari deteksi kadar oksigen bisa ditampilkan pada serial monitor dalam satuan persen pada arduino IDE.

Tabel 4.7 Spesifikasi MAX 30100

Tegangan Suplai-Power	1,7V – 2V
Tegangan Suplai LED	3,1V – 5V
Tegangan Suplai SpO2 dan Detak Jantung	600 μ A – 1200 μ A
Kisaran Suhu Beroperasi	-40°C - 85°C

6. Kabel Jumper

Kabel jumper berperan sebagai perantara pengiriman data dari sensor ke Mikrokontroler Arduino Mega.

7. Push Button

Push button berperan untuk mentrigger sistem kapan harus mengoperasikan pengambilan data dari setiap sensor dan mengoperasikan metode K-NN menggunakan data yang diterima.



8. LCD 16X2

LCD 16X2 berperan menampilkan hasil klasifikasi kondisi paru-paru berdasarkan perhitungan yang dilakukan sistem. Apakah kondisi paru-paru normal, TB, maupun Pneumonia. Pada Sistem Pendeteksi Kondisi Paru-Paru digunakan LCD 16x2 karena akan menampilkan karakter dari sistem sebanyak lebih dari 16 karakter. Untuk deskripsi pin LCD 16X2 dapat dilihat pada Tabel 4.2.

Tabel 4.8 Deskripsi Pin LCD 16X2

No	PIN	FUNGSI
1	GND	<i>Ground</i>
2	Vcc	<i>Power Supply (5V)</i>
3	Vee	Menengatur kontras dari lampu latar
4	RS	Memilih data register dan <i>command</i> register
5	R/W	Melaksanakan perintah <i>write</i> dan <i>read</i> ke register
6	EN	Saat diberi sinyal "1" ke "0", ditugaskan untuk membawa data ke pin yang digunakan
7	DB0	Bertugas untuk mentransfer data
8	DB1	
9	DB2	
10	DB3	
11	DB4	
12	DB5	
13	DB6	
14	DB7	Mengatur lampu latar dari LCD
15	LED +	
16	LED -	

Sumber: (Kurniawan, et al., 2019)



4.3.2.2 Kebutuhan Perangkat Lunak

Sistem memerlukan beberapa perangkat lunak antara lain:

1. Arduino IDE

Agar Arduino Mega dapat beroperasi dibutuhkan perangkat lunak untuk dapat mengoperasikannya, perangkat lunak tersebut yaitu Arduino IDE.

2. Library

Arduino IDE mempunyai beberapa library untuk memudahkan pemrograman pada Arduino Mega. Library yang digunakan dalam penelitian ini ialah :

1. LiquidCrystal_12C.h untuk menggunakan LCD 16x2.

2. CircularBuffer.h, MAX30100.h, MAX30100_BeatDetector, MAX30100_filters.h, MAX30100_Registers.h, MAX30100_Spo2Calculator.h, MAX30100_PulseOximeter.h untuk menggunakan sensor Max 30100.

3. Adafruit_MLX90614.h untuk menggunakan sensor MLX 90614.

4.4 Batasan Desain Sistem

Pada penelitian ini akan menerapkan batasan sistem yang dibutuhkan, yaitu :

1. Sistem melakukan klasifikasi terhadap 3 kondisi yaitu Normal, TB, Pneumonia.
2. Sistem akan mengerjakan pendeteksian dan pengklasifikasian kondisi paru-paru setelah menekan *push button*.
3. Untuk mengukur tingkat respirasi, sensor akan diikatkan dengan sabuk berbahan karet di dada.

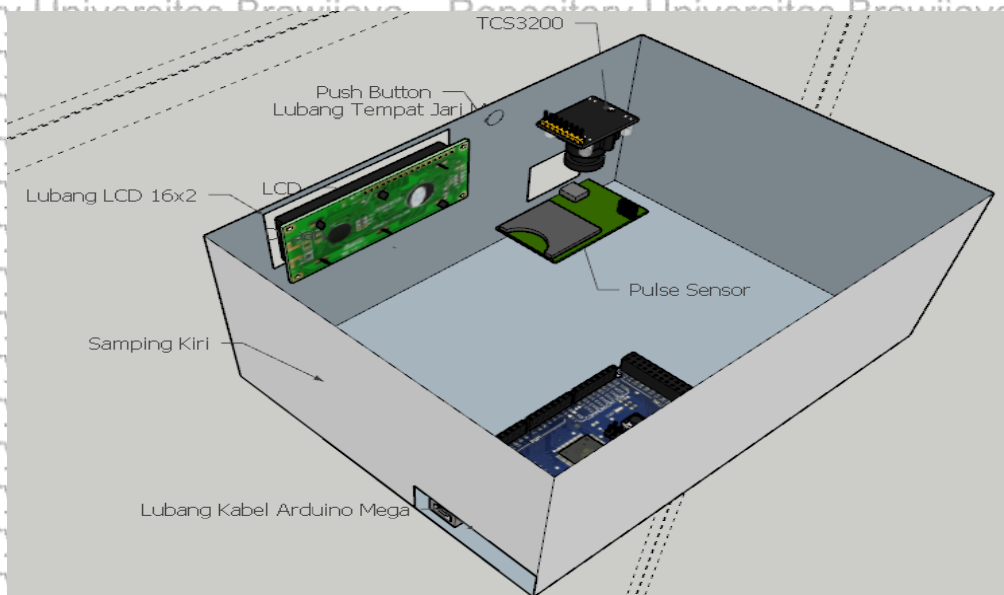
BAB 5 PERANCANGAN

5.1 Perancangan Sistem

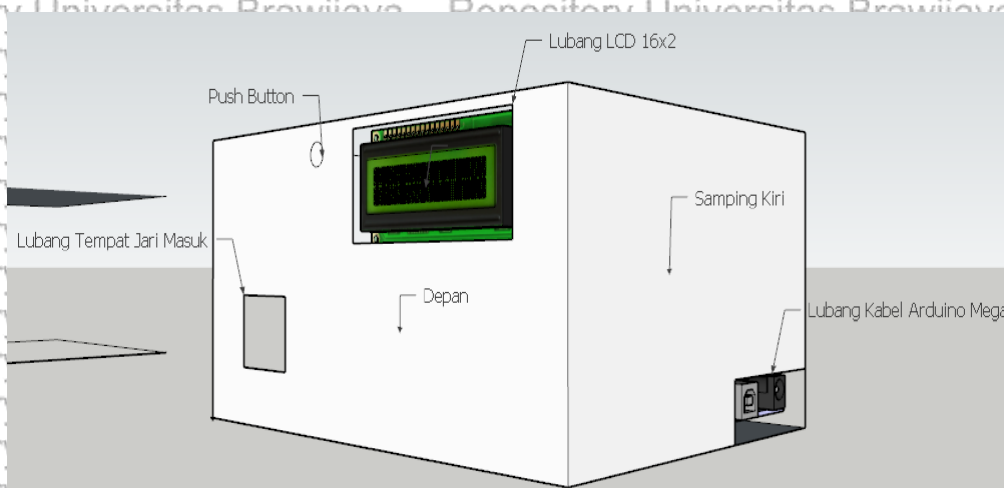
Perancangan sistem menjelaskan perihal perancangan dari sistem pendeteksi kondisi paru-paru secara lengkap.

5.1.1 Perancangan Purwarupa Sistem

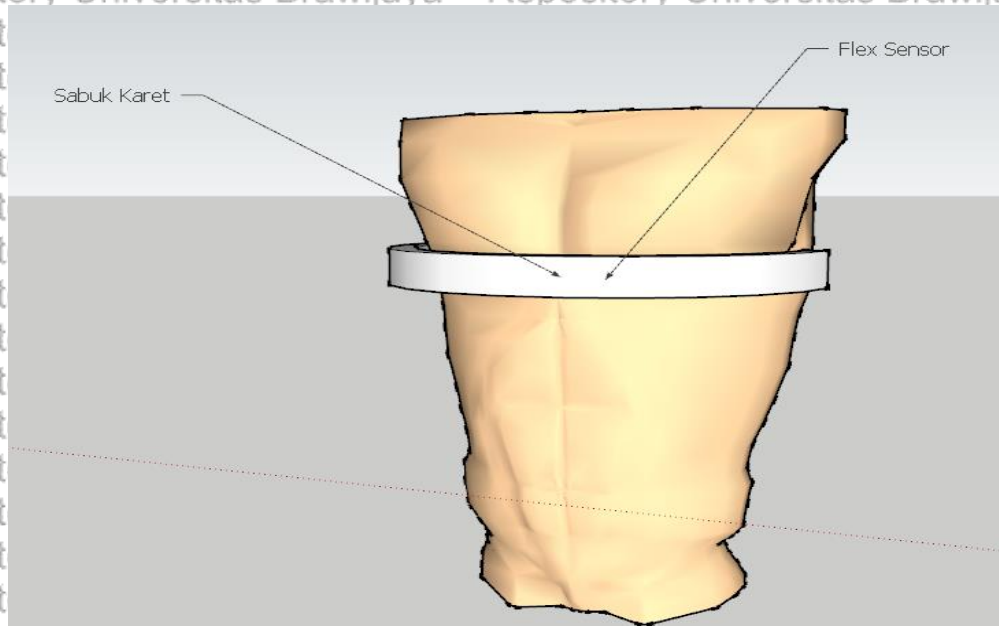
Pada Gambar 5.1 dan Gambar 5.2 diperlihatkan sketsa dari prototipe sistem yang dibuat.



Gambar 5.1 Tampak Atas Purwarupa Sistem



Gambar 5.2 Purwarupa Sistem

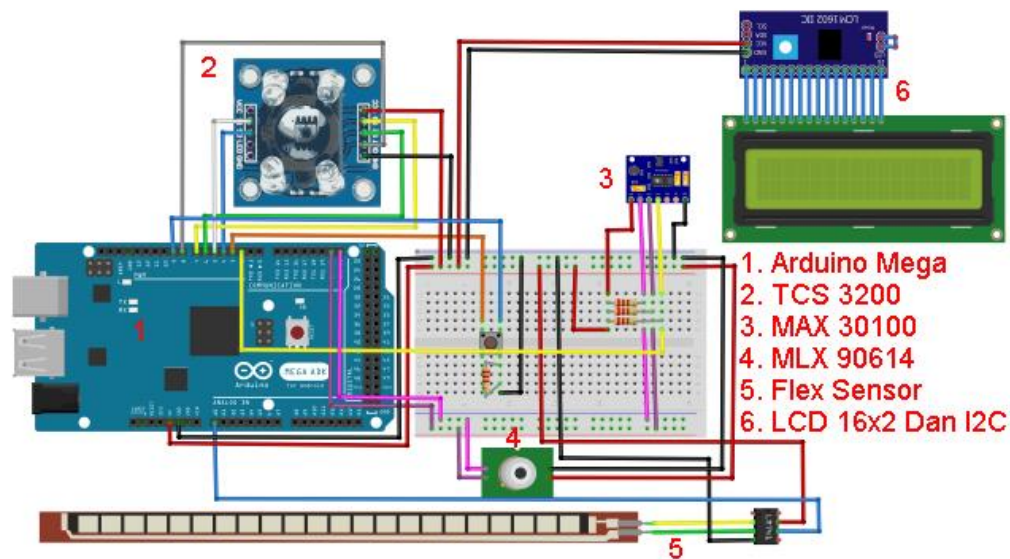


Gambar 5.3 Pemakaian Flex Sensor

Dapat dilihat pada Gambar 5.1 dan Gambar 5.2 diatas, sistem pendeteksi kondisi paru-paru akan dikemas berbentuk kotak dengan ukuran 20x15x10 cm dengan bahan akrilik. Akan dilakukan 3 pelubangan pada sistem yaitu, lubang tempat jari masuk untuk pengukuran warna kuku dan kadar oksigen dalam darah kemudian lubang LCD 16x2 dan yang terakhir lubang kabel arduino, *flex sensor*, dan sensor suhu. Posisi dari sensor warna TCS 3200 diletakkan didalam kotak, hal ini bertujuan untuk mengurangi gangguan cahaya dari luar, selain itu posisi dari sensor warna TCS 3200 menghadap kebawah ke lubang jari agar dapat mengukur warna kuku hanya dengan memasukkan kuku pasien ke dalam lubang tersebut dan sensor MAX 30100 menghadap keatas ke lubang jari, letaknya dibuat sedemikian rupa agar proses pengambilan data warna kuku dan kadar oksigen dalam darah dapat dilakukan secara efisien. Sensor suhu MLX 90614 dan *flex sensor* diletakkan diluar bungkus, hal ini dilakukan agar pengambilan data suhu dan tingkat respirasi dari pasien dapat dilakukan dengan mudah. Untuk tempat pengukuran, sensor suhu MLX 90614 diukur di bagian dahi sedangkan *flex sensor* diletakkan di dada pasien yang diikatkan sabuk karet seperti yang terlihat pada Gambar 5.3.

5.1.2 Perancangan Perangkat Keras Sistem

Dari hasil analisis kebutuhan dan perangkat keras digunakan, akan dilakukan penyusunan dari perangkat keras. Sehingga sistem dapat berjalan sesuai dengan ekspektasi. Penyusunan perangkat keras sistem sistem pendeteksi kondisi paru-paru dapat dilihat pada Gambar 5.4.



Gambar 5.4 Skematik Perangkat Keras Sistem

Dapat dilihat pada Gambar 5.2, pada skematik sistem pendeteksi kondisi paru-paru yang terdiri dari Arduino Mega, MLX 90614, MAX 30100, TCS 3200, Flex sensor, Push button, dan LCD 16X2 dengan modul I2C. agar sistem beroperasi sesuai keinginan, maka setiap komponen yang ada harus tersambung dengan Arduino Mega. Uraian pin yang menghubungkan antara TCS 3200 dengan Arduino Mega dapat dilihat pada Tabel 5.1.

Tabel 5.1 Koneksi Pin Arduino Mega dengan Pin Sensor TCS 3200

Pin Arduino Mega	Pin sensor TCS 3200
5V	VCC
GND	GND
D4	S0
D5	S1
D6	S2
D7	S3
D8	OUT
_	OE

Tabel 5.1 berisikan uraian pin Arduino Mega dengan TCS 3200. Sensor TCS 3200 memiliki pin Vcc dan GND yang terhubung pada pin 5V dan Ground pada Arduino Mega. Kemudian untuk pin S0, S1, S2, dan S3 terhubung dengan pin D4, D5, D6, dan D7 pada Arduino Mega. Terakhir untuk pin output terhubung dengan pin D8 pada Arduino Mega.



Uraian tentang keterangan pin yang menghubungkan antara MLX 90614 dengan Arduino Mega dapat dilihat pada Tabel 5.2.

Tabel 5.2 Koneksi Pin Arduino Mega dengan Pin Sensor MLX 90614

Pin Arduino Mega	Pin Sensor MLX 90614
5V	VCC
GND	GND
SDA	SDA
SCL	SCL

Tabel 5.2 berisikan uraian pin Arduino Mega dengan MLX 90614. Sensor MLX 90614 memiliki pin Vcc dan GND yang terhubung ke pin 5v dan ground pada Arduino Mega. Kemudian untuk pin SDA dan SCL terhubung ke pin SDA dan pin SCL pada Arduino Mega.

Uraian tentang pin yang menghubungkan antara Max 30100 dengan Arduino Mega dapat dilihat pada Tabel 5.3.

Tabel 5.3 Koneksi Pin Arduino Mega dengan Pin Sensor MAX 30100

Pin Arduino Mega	Pin Sensor MAX 30100
5V	VIN
SDA	SDA
SCL	SCL
D2	INT
Ground	GND

Tabel 5.3 berisikan merupakan koneksi pin antara pin Arduino Mega dengan MAX 30100. Sensor MAX 30100 memiliki pin Vcc dan GND yang terhubung ke pin 5v dan ground pada Arduino Mega. Kemudian untuk pin SDA, SCL, dan INT terhubung ke pin SDA, SCL, dan D2 pada Arduino Mega.

Uraian tentang pin yang menghubungkan antara Flex sensor dengan Arduino Mega dapat dilihat pada Tabel 5.4.

Tabel 5.4 Koneksi Pin Arduino Mega dengan Pin Flex Sensor dengan OP-AMP

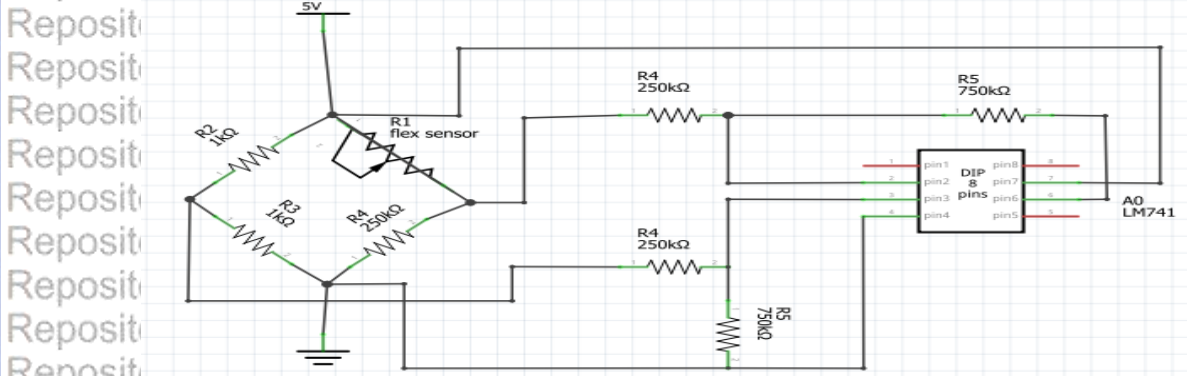
Pin Arduino Mega	Pin Flex Sensor dengan OP-AMP
5V	PIN 7
GND	PIN 4
A7	PIN 6



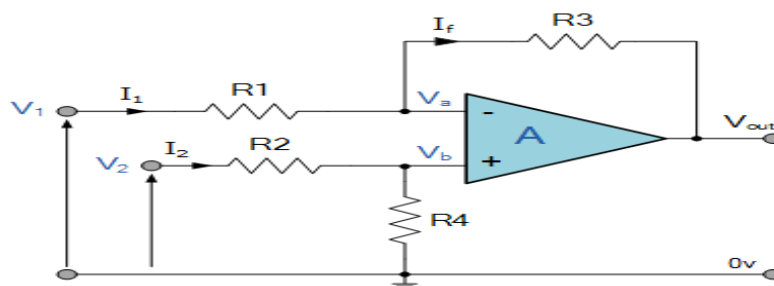
Repository Universitas Brawijaya
 Repository Universitas Brawijaya
 Repository Universitas Brawijaya
 Repository Universitas Brawijaya
 Repository Universitas Brawijaya
 Repository Universitas Brawijaya

Repository Universitas Brawijaya
 Repository Universitas Brawijaya
 Repository Universitas Brawijaya
 Repository Universitas Brawijaya
 Repository Universitas Brawijaya
 Repository Universitas Brawijaya

Repository
 Repository
 Repository
 Repository
 Repository
 Repository



Gambar 5.5 Rangkaian Flex Sensor dengan OP- AMP



Gambar 5.6 Rangkaian Penguat Diferensial

Terlihat pada Gambar 5.5, *Flex sensor* dirangkai menggunakan jembatan *wheatstone* yang outputnya disambungkan ke *op-amp* LM 741, digunakannya *op-amp* pada penelitian ini untuk memperkuat tegangan menggunakan rumus persamaan 2 apabila nilai resistor $R1=R2$ dan $R3=R4$ seperti pada rangkaian Gambar 5.6 (Tutorials, 2014). Jadi Penguatan yang diberikan dari *op-amp* LM 741 sebesar 3x penguatan. Kemudian dari pin 4 dan pin 7 LM 741 disambungkan ke pin ground dan 5v dari Arduino mega. Untuk output dari LM741 terdapat pada pin 6 yang disambungkan pada pin A7 pada Arduino Mega. Maka nilai tegangan dari *flex sensor* sesuai dengan persamaan 2. Dimana nilai tersebut akan diubah menjadi nilai digital menggunakan ADC yang ada di Arduino Mega.

$$V_{out} = \frac{R3}{R1} \times V_{in} \dots \dots \dots (2)$$

Uraian tentang pin yang menghubungkan antara *push button* dengan pin Arduino Mega dapat dilihat pada Tabel 5.5

Tabel 5.5 Koneksi Pin Arduino Mega dengan Pin Push Button

Pin Arduino Mega	Pin Push Button
D3	OUT
GND	GND
D9	PIN

Repository Universitas Brawijaya
 Repository Universitas Brawijaya
 Repository Universitas Brawijaya
 Repository Universitas Brawijaya
 Repository Universitas Brawijaya
 Repository Universitas Brawijaya

Repository Universitas Brawijaya
 Repository Universitas Brawijaya
 Repository Universitas Brawijaya
 Repository Universitas Brawijaya
 Repository Universitas Brawijaya
 Repository Universitas Brawijaya

Repository
 Repository
 Repository
 Repository
 Repository
 Repository



Tabel 5.5 berisikan koneksi pin antara pin Arduino Mega dengan pin *push button* yang mempunyai pin output dan gnd yang terhubung ke pin D3 dan ground pada Arduino Mega.

Uraian tentang pin yang menghubungkan antara LCD 16x2 dengan Arduino Mega dapat dilihat pada Tabel 5.6.

Tabel 5.6 Koneksi Pin Arduino Mega dengan Pin I2C LCD 16X2

Pin Arduino Mega	Pin I2C LCD 16X2
5V	VCC
GND	GND
SDA	SDA
SCL	SCL

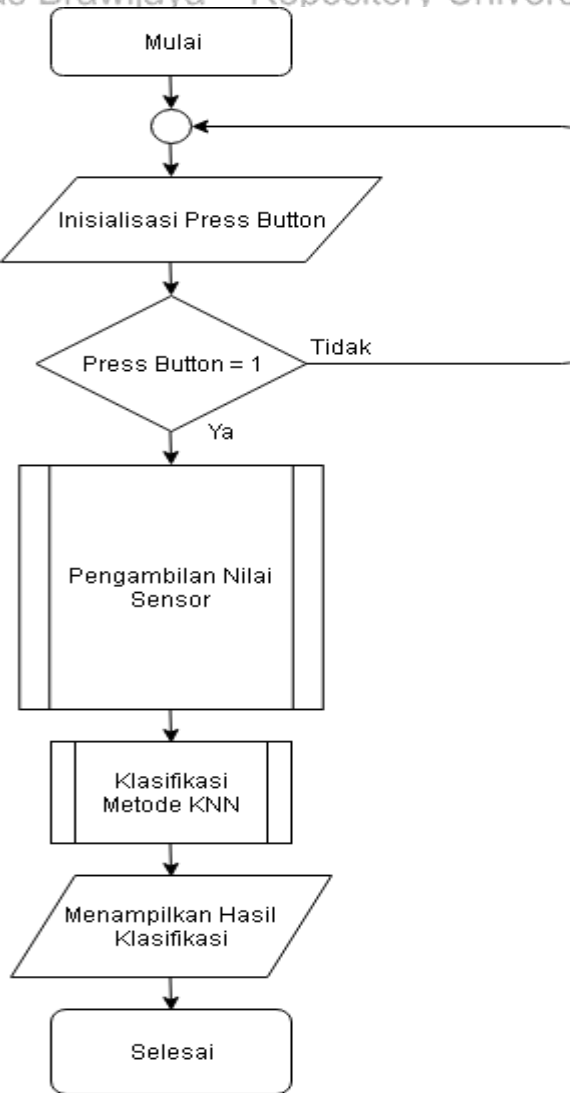
Tabel 5.6 berisikan tentang koneksi pin untuk dapat menampilkan keluaran pada LCD 16X2 dengan tambahan modul I2C. Modul I2C digunakan agar meminimalisir penggunaan pin yang berlebihan pada Arduino Mega. Sehingga pin yang terpakai untuk mengoperasikan LCD hanya 4 pin, yaitu pin GND, SCL, SDA, dan VCC.

5.1.3 Perancangan Perangkat Lunak Sistem

Terdapat tiga bagian yang akan dibahas pada sub bab ini. Pertama, pemrograman perangkat lunak pada mikrokontroler dalam program utama. Kedua, pemrograman perangkat lunak untuk pengukuran nilai suhu, nilai RGB, kadar oksigen dalam darah, dan tingkat respirasi. Ketiga, pemrograman perangkat lunak untuk menentukan kondisi paru-paru menggunakan metode K-NN.

5.1.3.1 Perancangan Program Utama

Pada sub bab perancangan perangkat lunak, penyusunan program utama dari Sistem Pendeteksi Kondisi Paru-Paru menggunakan metode K-NN akan dirancang menggunakan skema. Adapun skema dari penyusunan program utama dapat dilihat pada Gambar 5.7.



Gambar 5.7 Diagram Alir Perancangan Perangkat Lunak Program Utama

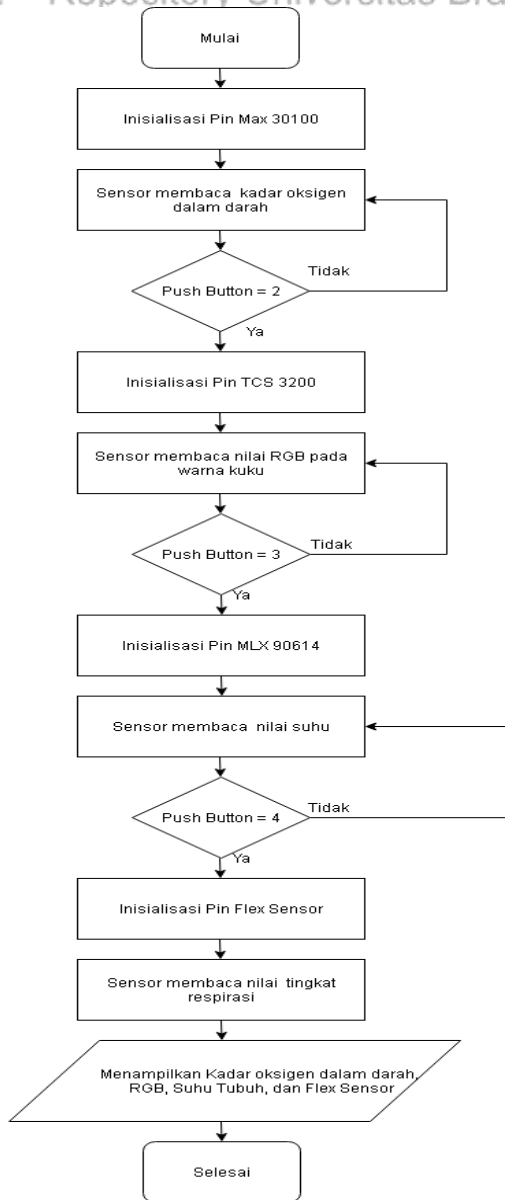
Langkah awal skema program utama, dilakukan inisialisasi *push button* yang digunakan iterasi mikrokontroler, disaat push button ditekan satu kali maka sistem akan membaca kadar oksigen dalam darah, saat ditekan kedua kali maka akan membaca nilai R, G, B, dari warna kuku, penekanan ketiga akan mengukur suhu tubuh, penekanan keempat akan membaca tingkat respirasi, dan penekanan kelima akan melakukan klasifikasi dengan metode KNN dari nilai kadar oksigen dalam darah, R, G, B, suhu, dan tingkat respirasi yang nantinya akan ditampilkan di LCD 16x2.

5.1.3.2 Perancangan Pengambilan Nilai Sensor

Pengambilan data sensor akan dirancang menggunakan skema. Adapun skema dari perancangan pengambilan data sensor dapat dilihat pada Gambar 5.8.



Pengambilan Nilai Sensor



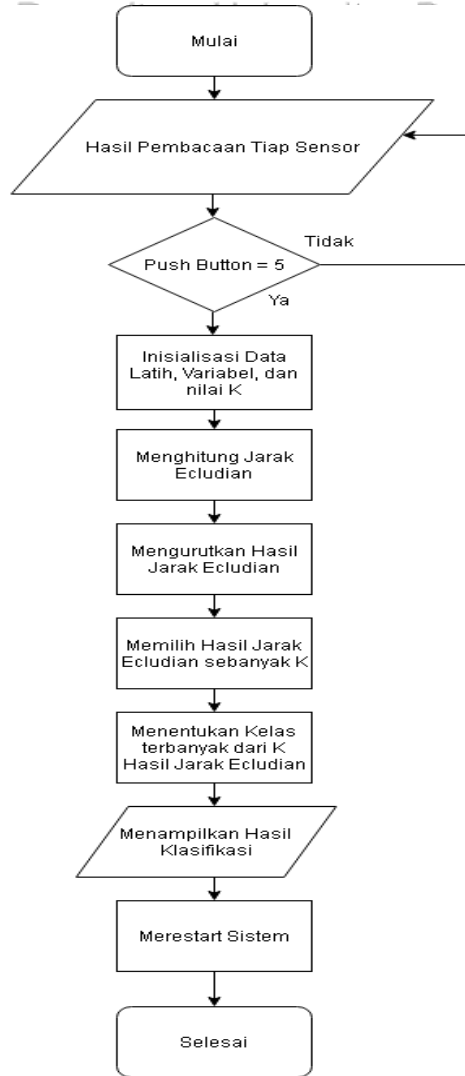
Gambar 5.8 Diagram Alir Pengambilan Nilai Sensor

Pada awal tahapan skema pengambilan nilai sensor, dilakukan inisialisasi pin dari sensor MAX 30100 dan akan membaca nilai kadar oksigen dalam darah. Apabila *push button* ditekan kedua kali akan menginisialisasi pin TCS 3200 dan membaca nilai RGB dari warna kuku. Saat penekanan ketiga kali oleh *push button* akan menginisialisasi pin dari MLX 90614 dan mengukur suhu tubuh. Dan saat penekanan keempat *push button*, akan menginisialisasi pin sensor *Flex* dan mengukur tingkat respirasi. Hasil dari pengukuran tiap sensor yang sudah didapatkan akan ditampilkan pada serial monitor dan diolah untuk menghasilkan *output* sistem.



5.1.3.3 Perancangan Klasifikasi K-NN

Perancangan pemograman klasifikasi KNN akan dibuat menggunakan diagram alir. Adapun diagram alir dari perancangan perangkat lunak klasifikasi KNN dapat dilihat pada Gambar 5.9.



Gambar 5.9 Diagram Alir Klasifikasi K-NN

Langkah awal skema pemograman KNN untuk mengetahui jarak *Euclidean* terhadap data uji harus terdapat data latih, variabel, dan nilai K yang sebelumnya harus diinisialisasi terlebih dahulu. Setelah mendapatkan semua jarak *Euclidean* terhadap data latih, akan di urutkan secara runtun dari jarak terdekat ke jarak terjauh. Setelah hasil dari setiap jarak didapatkan, kemudian akan dipilih jarak terdekat sebanyak K yang sudah diinisialisasi pada awal skema. Kemudian akan dilakukan voting kelas terbanyak dalam data tersebut menggunakan *weighted voting*, kelas terbanyak yang terdapat pada data tersebut akan ditampilkan pada layar LCD.

Pada setiap data dari 14 data uji akan dihitung jarak *Euclidean* satu per satu terhadap 30 data latih pada penelitian ini yang diambil di Puskesmas Janti, Kota



Malang dan referensi kelas diberikan oleh kepala klinik TB dari Puskesmas Janti. Adapun data latih yang digunakan dapat dilihat pada Tabel 5.7.

Tabel 5.7 Data Latih Klasifikasi Metode K-NN

No	R	G	B	SpO2(%)	Suhu(°C)	Flex	Kelas
1	183	132	123	94	36,3	40	Normal
2	183	141	143	95	36,3	40	Normal
3	190	180	172	96	36,5	43	Normal
4	190	139	134	96	36,7	38	Normal
5	192	171	181	96	36,7	40	Normal
6	194	154	154	96	36,4	37	Normal
7	195	183	179	95	36,3	41	Normal
8	205	175	173	94	36,3	40	Normal
9	206	217	212	97	36,3	50	Normal
10	207	183	183	95	36,3	38	Normal
11	207	167	162	95	36,7	56	Normal
12	210	180	185	96	36,4	63	Normal
13	210	191	172	96	36,2	40	Normal
14	216	233	220	94	36,8	35	TB
15	216	181	179	95	36,6	25	TB
16	217	204	199	94	36,2	26	TB
17	217	153	210	94	36,9	25	TB
18	217	179	178	93	36,9	29	TB
19	225	237	250	93	37,6	27	TB
20	226	195	186	94	36,9	30	TB
21	226	172	155	94	36,7	31	TB
22	227	220	236	93	37,1	33	TB
23	227	201	200	94	37,0	26	Pneu
24	228	177	173	94	37,2	23	Pneu
25	228	187	184	94	36,9	25	Pneu
26	229	196	187	92	37,3	22	Pneu
27	231	241	272	94	37,9	24	Pneu
28	240	234	230	94	37,6	24	Pneu
29	240	236	254	94	37,7	30	Pneu
30	242	188	149	92	37,0	22	Pneu

Data uji bernilai $R=195$, $G=160$, $B=165$, $SpO_2=94\%$, $Suhu=36^\circ C$, dan $Flex=60$ akan dihitung jarak *Euclidean* terhadap 30 data latih yang tersedia menggunakan persamaan 1. Hasil dari perhitungan jarak *Euclidean* dapat dilihat pada Tabel 5.8.

**Tabel 5.8 Hasil Jarak Euclidean**

No	Jarak Euclidean	Kelas
1	55,6685663	Normal
2	37,453674	Normal
3	27,5717148	Normal
4	43,7890098	Normal
5	28,1986237	Normal
6	26,2728355	Normal
7	32,8273694	Normal
8	27,8070642	Normal
9	74,7256947	Normal
10	38,3170455	Normal
11	14,1839376	Normal
12	31,9617933	Normal
13	39,9549083	Normal
14	96,6416895	TB
15	47,7226054	TB
16	68,5439514	TB
17	61,6451681	TB
18	44,1588148	TB
19	122,420573	TB
20	59,029766	TB
21	44,8194906	TB
22	101,533207	TB
23	71,0443842	Pneu
24	52,6682462	Pneu
25	58,0109765	Pneu
26	66,3256211	Pneu
27	143,249993	Pneu
28	113,622942	Pneu



29	129,034024	Pneu
30	67,6250232	Pneu

Data jarak terhadap semua data latih akan diurutkan dari jarak paling pendek ke jarak paling jauh apabila jarak ke semua data latih sudah dihitung. Hasil dari pengurutan jarak *Euclidean* dapat dilihat pada Tabel 5.9.

Tabel 5.9 Hasil Sorting Klasifikasi K-NN

No	Jarak Euclidean	Kelas
1	14,1839376	Normal
2	26,2728355	Normal
3	27,5717148	Normal
4	27,8070642	Normal
5	28,1986237	Normal
6	31,9617933	Normal
7	32,8273694	Normal
8	37,453674	Normal
9	38,3170455	Normal
10	39,9549083	Normal
11	43,7890098	Normal
12	44,1588148	TB
13	44,8194906	TB
14	47,7226054	TB
15	52,6682462	Pneu
16	55,6685663	Normal
17	58,0109765	Pneu
18	59,029766	TB
19	61,6451681	TB
20	66,3256211	Pneu
21	67,6250232	Pneu
22	68,5439514	TB
23	71,0443842	Pneu



24	74,7256947	Normal
25	96,6416895	TB
26	101,533207	TB
27	113,622942	Pneu
28	122,420573	TB
29	129,034024	Pneu
30	143,249993	Pneu

Setelah mengurutkan jarak *Euclidean* secara *ascending*, nilai K yang sudah ditetapkan diawal akan di simpan untuk menentukan jarak terpendek. K bernilai 3, nilai hasil jarak *Euclidean* teratas dapat dilihat pada Tabel 5.10.

Tabel 5.10 Nilai Jarak Euclidean Lima Teratas

No	Jarak Euclidean	Kelas
1	14,1839376	Normal
2	26,2728355	Normal
3	27,5717148	Normal

Dari hasil jarak *Euclidean* tiga teratas yang sudah didapatkan, voting akan dilakukan untuk menentukan mayoritas kelas yang ada, pada Tabel 5.10 kelas terbanyak adalah kelas normal. Sehingga kelas dari data uji nomor 1 adalah Kelas Normal.

5.2 Implementasi Sistem

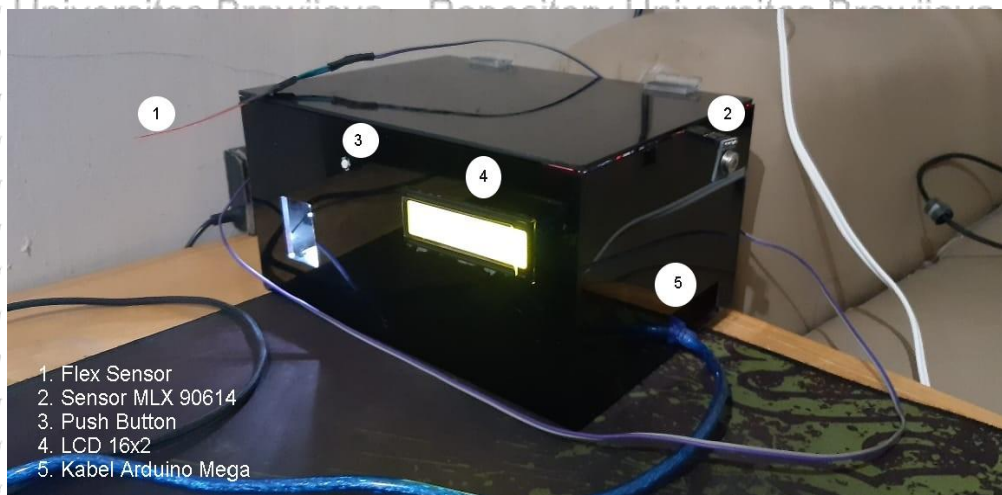
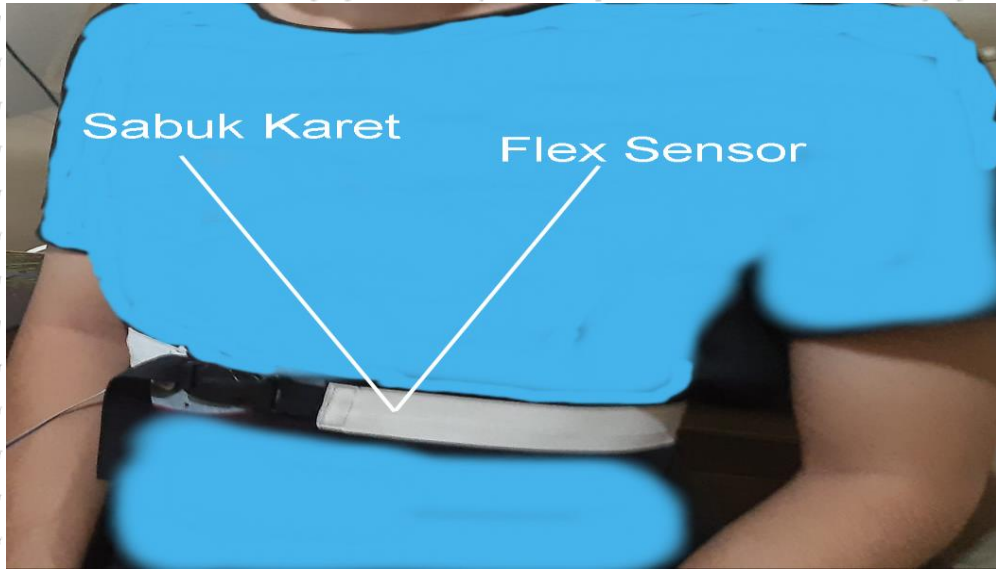
Berdasarkan perancangan sebelumnya, implementasi sistem dilakukan untuk merealisasikan perancangan sebelumnya. Implementasi purwarupa, implementasi perangkat keras, dan implementasi perangkat lunak akan dijadikan pembahasan kali ini.

5.2.1 Implementasi Purwarupa Sistem

Perancangan prototipe dari sistem yang dirancang pada bab sebelumnya, pada penjelasan kali ini akan diterapkan pembuatan dari prototipe sistem berdasarkan rancangan sebelumnya. Pada implementasi purwarupa sistem pendeteksi kondisi paru-paru prototipe yang dibangun memiliki panjang 20cm, lebar 15cm dan tinggi 10cm. prototipe berbentuk kotak dan berwarna hitam agar meredam cahaya dari luar saat pengukuran warna kuku dan flex sensor diikatkan didada subjek menggunakan sabuk berbahan karet. Pada Gambar 5.10 menampakkan penerapan dari rancangan bentuk prototipe yang dirancang bab



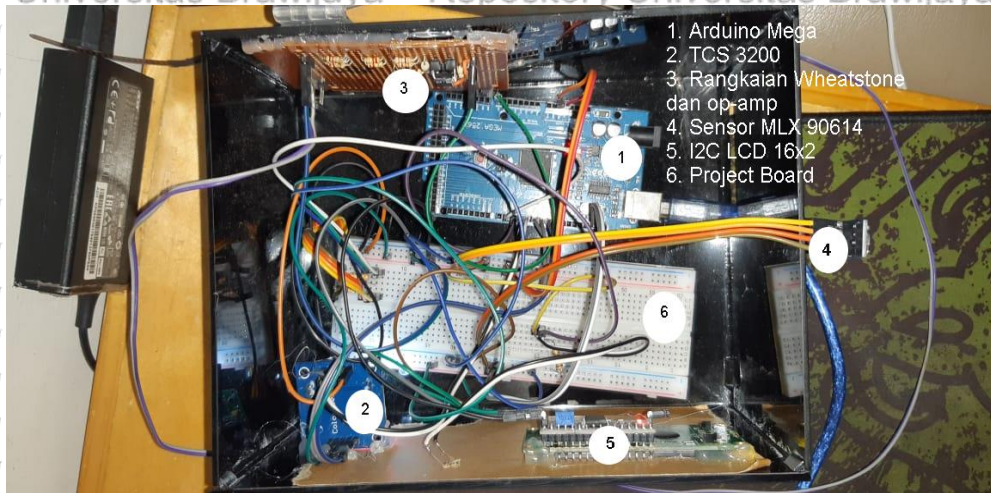
sebelumnya pada yang didalamnya terdapat Arduino Mega, Sensor TCS 3200, Sensor Max 30100, Sensor MLX 90614, Flex Sensor dan LCD 16X2.



Gambar 5.10 Implementasi Purwarupa Sistem Pendeteksi Kondisi Paru-Paru

5.2.2 Implementasi Perangkat Keras

Berdasarkan rancangan penerapan setiap komponen pada sistem pendeteksi kondisi paru paru, Gambar 5.11 A dan B merupakan penerapan dari rancangan setiap komponen perangkat keras yang akan digunakan pada penelitian ini.



1. Arduino Mega
2. TCS 3200
3. Rangkaian Wheatstone dan op-amp
4. Sensor MLX 90614
5. I2C LCD 16x2
6. Project Board

(A) Tampak Atas



1. Sensor MAX 30100
2. Push Button
3. LCD 16x2

(B) Tampak Depan

Gambar 5.11 Implementasi Perangkat Keras Sistem Pendeteksi Kondisi Paru-Paru

Hasil dari penerapan rancangan perangkat keras yang dilakukan sistem ini secara lengkap dapat diamati pada Gambar 5.11 B. Terlihat dari gambar nomer 1, posisi dari sensor TCS 3200 diletakkan diatas lubang dengan menghadap kebawah dan MAX 30100 diletakkan dibawah lubang dengan menghadap keatas lubang, sedangkan kan sensor suhu MLX 90614 di letakkan diluar bersamaan dengan Flex Sensor yang disambungkan dengan kain elastis berbahan karet. Pada Gambar 5.11 B menunjukkan LCD 16x2 terdapat pada lubang nomer 3 dan ditempat lubang nomer 2 terpasang Push Button dengan penempatan yang sesuai dengan sketsa rancangan sebelumnya

5.2.3 Implementasi Perangkat Lunak

Pada implementasi perangkat lunak menjelaskan tentang penerapan rancangan dari perangkat lunak sistem untuk menentukan hasil kondisi paru-paru. Pada penerapan rancangan perangkat lunak sistem akan dibuat program untuk



menerapkan kalsifikasi K-NN yang dimana program tersebut di program menggunakan pemogrman Arduino.

5.2.3.1 Implementasi Kode Program Utama

Tabel 5.11 akan memeprilihatkan penggalan kode dari dari program pusat yang mengatur perintah setiap jalannya pembacaan setiap komponen.

Tabel 5.11 Kode Library Program Utama

No	Kode Program
1	#include<LiquidCrystal_I2C.h>
2	#include<CircularBuffer.h>
3	#include<MAX30100.h>
4	#include<MAX30100_BeatDetector.h>
5	#include<MAX30100_Filters.h>
6	#include<MAX30100_Registers.h>
7	#include<MAX30100_SpO2Calculator.h>
8	#include<Wire.h>
9	#include<MAX30100_PulseOximeter.h>

Tabel 5.11 dari merupakan penggalan kode program yang fungsinya untuk menginisialisasi beberapa *library* untuk dapat mengoperasikan komponen yang membutuhkan *library*.

Tabel 5.12 Deklarasi dan Inisialisasi Variabel

No	Kode Program
1	#define REPORTING_PERIOD_MS 1000
2	LiquidCrystal_I2C lcd = LiquidCrystal_I2C(0x27, 16, 2);
3	PulseOximeter pox;
4	uint32_t tsLastReport = 0;
5	float datasensor[6];
6	float range[30][2];
7	unsigned long timeku, start, finish, waktu;
8	int pbState = 0;
9	int pbPin = 3;
10	int state = 0;
11	int getppg = 0;

Tabel 5.12 merupakan penggalan kode program yang berfungsi untuk menginisialisasi setiap variabel dari tiap komponen yang akan digunakan pada penerapan pemograman Arduino. Baris kedua digunakan untuk inisialisasi akan



memakai LCD dengan I2C dengan rincian LCD yang digunakan memiliki 16 karakter tiap barisnya. Pada baris 3, 4, dan 11 digunakan oleh sensor MAX 30100 untuk beroperasi. Pada baris 5 dan 6 digunakan untuk menyimpan nilai tiap sensor dan menentukan banyak data latih. Pada baris 7 digunakan untuk menyimpan nilai dari fungsi *millis()* untuk mengetahui waktu komputasi dari sistem. Pada baris 8-10 digunakan *push button* agar dapat menyimpan nilai dari tiap penekanan tombol.

Tabel 5.13 Kode Trigger Push Button

No	Kode Program
1	lcd.print("Tekan Tombol");
2	if(digitalRead(pbPin)==HIGH){
3	Serial.println("Tombol ditekan");
4	pbState++;
5	delay(1000);
6	if(pbState == 4) timeku = millis();
7	}
8	if(pbState == 1){
9	lcd.setCursor(0,1);
10	lcd.print(String(getppg));
11	datasensor[3] = getppg;
12	Serial.println(getppg);
13	Serial.println();
14	}
15	if(pbState == 2){
16	Serial.println("Penekanan Kedua");
17	Serial.println("Pengukuran kuku");
18	loopkuku();
19	delay(1000);
20	}
21	if(pbState == 3){
22	Serial.println("Penekanan Ketiga");
23	Serial.println("Pengukuran suhu");
24	loopsuhu();
25	delay(1000);
26	}
27	if(pbState == 4){
28	int tes = 1;
29	Serial.println("Penekanan Keempat");
30	if((tes == 1){



```

31 Serial.println(millis()-timeku);
32 Serial.println("Pengukuran Flex");
33 loopFlex();
34 delay(100);
35 tes++;
36 } else{
37 Serial.print("R = ");
38 Serial.println(datasensor[0]);
39 Serial.print("G = ");
40 Serial.println(datasensor[1]);
41 Serial.print("B = ");
42 Serial.println(datasensor[2]);
43 Serial.print("Spo2 = ");
44 Serial.println(datasensor[3]);
45 Serial.print("Suhu = ");
46 Serial.println(datasensor[4]);
47 Serial.print("Flex = ");
48 Serial.println(datasensor[5]);}
49 }
    
```

Pada Tabel 5.13 memiliki fungsi untuk mentrigger *push button* untuk melakukan pengukuran nilai dari tiap sensor. Pada penekanan pertama akan mengukur tingkat kadar oksigen dalam darah, pada penekanan kedua akan membaca nilai RGB pada warna kuku, penekanan ketiga akan mengukur suhu tubuh, penekanan keempat akan mengukur tingkat respirasi dan menampilkan nilai dari hasil tiap sensor.

Tabel 5.14 Kode Perhitungan K-NN

No	Kode Program
1	//Perhitungan KNN
2	int DataJarak = 0;
3	int label = 1;
4	//Perhitungan Eclidean
5	for(int a=0;a<30;a++){
6	range[a][0] = 0; //reset data
7	for(int b=0;b<6;b++){
8	range[a][0] += pow(DataLatih (a, b) -
9	datasensor[b], 2);
10	}



```

11 range[a][0] = sqrt(range[a][0]); //akar
12 switch(a){
13     case 0 ... 12:
14         range[a][1] = 0; //angka 0 untuk label normal
15         break;
16     case 13 ... 21:
17         range[a][1] = 1; //angka 1 untuk label TB
18         break;
19     case 22 ... 29:
20         range[a][1] = 2; //angka 2 untuk label Pneumonia
21         break;
22     }
23     //Serial.print("Hasil Ecludian = ");
24     //Serial.println(range[a][0]);
25     }
26     //SORTING DATA
27     for(int i=0;i<29;i++){
28         for(int j=i+1;j<30;j++){
29             if(range[i][DataJarak]>range[j][DataJarak]){
30                 float temp = range[i][DataJarak];
31                 range[i][DataJarak] = range[j][DataJarak];
32                 range[j][DataJarak] = temp;
33             }
34             temp = range[i][label];
35             range[i][label] = range[j][label];
36             range[j][label] = temp;
37         }
38     }
39     }
40     float nol = 0, satu = 0, dua = 0;
41     String hasil;
42     for(int a=0;a<3;a++){
43         if(range[a][label] == 0.00){
44             nol += 1/ pow(range[a][DataJarak],2);
45         }
46         else if(range[a][label] == 1.00){
47             satu += 1/ pow(range[a][DataJarak],2);
48         }

```




```

49 else if(range[a][label] == 2,00){
50     dua += 1/ pow(range[a][DataJarak],2);
51 }
52 }
53 if(nol > satu && nol > dua){
54     hasil = "NORMAL";
55 }
56 else if(satu > nol && satu > dua){
57     hasil = "TB";
58 }
59 else if(dua > satu && dua > nol){
60     hasil = "PNEUMONIA";
61 }
    
```

Pada Tabel 5.14 memiliki fungsi sebagai perhitungan dan pengklasifikasian kondisi paru-paru menggunakan metode K-NN dari sistem pendeteksi kondisi paru-paru. Pada baris 1-21 berfungsi untuk memprhitungkan jarak *Euclidean* dari tiap data yang diujikan terhadap data latih yang dipakai kemudian akan diberi label. Pada baris 26-52 akan mensorting data jarak *euclidean* dari terkecil ke terbesar, kemudian akan diambil 3 data teratas untuk dilakukan voting banyaknya kelas pada 3 data tersebut. Pada baris 53-61 akan menentukan hasil klasifikasi dari hasil voting sebelumnya apakah yang paling banyak kelas Normal, TB, atau Pneumonia.

Tabel 5.15 Kode Hasil Klasifikasi dan Waktu Komputasi

No	Kode Program
1	finish = millis();
2	waktu = (finish - start);
3	lcd.clear();
4	lcd.setCursor(0,0);
5	lcd.print(hasil);
6	Serial.print("Hasil Klasifikasi :");
7	Serial.println(hasil);
8	Serial.print("Waktu Komputasi = ");
9	Serial.println(waktu);
10	delay(15000);
11	resetFunc();

Pada Tabel 5.15 memiliki fungsi untuk menampilkan hasil dari waktu komputasi dari sistem pendeteksi kondisi paru-paru. Pada baris 2 digunakan untuk



menyimpan nilai waktu komputasi di variabel bernama waktu. Pada baris 3-5 akan menghapus segala karakter pada LCD dan kemudian akan diisi hasil klasifikasi sistem di pojok kanan atas LCD. Pada baris 6-9 akan menampilkan hasil klasifikasi sistem dan waktu komputasi pada serial monitor.

5.2.3.2 Implementasi Kode Pengambilan Data Sensor

Penerapan dari program untuk mendapatkan nilai dari tiap sensor akan dilakukan setelah setiap susunan pin komponen terhubung dengan Arduino Mega. Akan ditampilkan kode program dari tiap sensor yang digunakan yaitu sensor TCS 3200, MAX 30100, MLX 90614, dan Flex sensor. Tabel 5.16 memperlihatkan penggalan kode program dari pengambilan nilai dari TCS 3200.

Tabel 5.16 Kode Pengambilan Data Sensor TCS 3200

No	Kode Program
1	#define S0 4
2	#define S1 5
3	#define S2 6
4	#define S3 7
5	#define sensorOut 8
6	int redfrequency = 0;
7	int greenfrequency = 0;
8	int bluefrequency = 0;
9	int redcolor = 0;
10	int greencolor = 0;
11	int bluecolor = 0;

Penggalan kode program yang terdapat pada Tabel 5.16 memiliki fungsi untuk mendefinisikan susunan pin dan variabel dari TCS 3200. Pada baris 1-5 digunakan untuk inialisasi pin pada sensor TCS 3200. Baris 6-11 digunakan untuk deklarasi variabel untuk menyimpan nilai warna RGB yang diukur.

Tabel 5.17 Kode Pengambilan R,G,dan B

No	Kode Program
1	void setupkuku() {
2	pinMode(S0, OUTPUT);
3	pinMode(S1, OUTPUT);
4	pinMode(S2, OUTPUT);
5	pinMode(S3, OUTPUT);
6	pinMode(sensorOut, INPUT);
7	digitalWrite(S0, HIGH);



```
8 digitalWrite(S1,LOW);
9 }
10 void loopkuku() {
11 // Setting red filtered photodiodes to be read
12 digitalWrite(S2,LOW);
13 digitalWrite(S3,LOW);
14 // Reading the output frequency
15 redfrequency = pulseIn(sensorOut, LOW);
16 redcolor = map(redfrequency, 52,98,255,0); //34,167//49,63
17 if (redcolor > 255) {
18 redcolor = 255;
19 } else if (redcolor <= 0) {
20 redcolor = 0;
21 }
22 // Printing the value on the serial monitor
23 Serial.print("R= "); //printing name
24 Serial.print(redcolor); //printing RED color frequency
25 Serial.print(" ");
26 datasensor[0] = redcolor;
27 delay(100);
28 // Setting Green filtered photodiodes to be read
29 digitalWrite(S2,HIGH);
30 digitalWrite(S3,HIGH);
31 // Reading the output frequency
32 greenfrequency = pulseIn(sensorOut, LOW);
33 greencolor=map(greenfrequency,71,131,255,0);
34 if (greencolor > 255) {
35 greencolor = 255;
36 } else if (greencolor <= 0) {
37 greencolor = 0;
38 }
39 // Printing the value on the serial monitor
40 Serial.print("G= "); //printing name
41 Serial.print(greencolor); //printing RED color frequency
42 Serial.print(" ");
43 datasensor[1] = greencolor;
44 delay(100);
45 // Setting Blue filtered photodiodes to be read
```




```

46 digitalWrite(S2, LOW);
47 digitalWrite(S3, HIGH);
48 // Reading the output frequency
49 bluefrequency = pulseIn(sensorOut, LOW);
50 bluecolor=map(bluefrequency, 60, 110, 255, 0); //45, 172 //59, 91
51 if(bluecolor > 255) {
52 bluecolor = 255;
53 } else if (bluecolor <= 0) {
54 bluecolor = 0;
55 }
56 // Printing the value on the serial monitor
57 Serial.print("B= "); //printing name
58 Serial.print(bluecolor); //printing RED color frequency
59 Serial.println(" ");
60 datasensor[2] = bluecolor;
61 delay(3000);
62 if(redcolor > greencolor && redcolor > bluecolor){
63 Serial.println(" - RED detected!");
64 }
65 if(greencolor > redcolor && greencolor > bluecolor){
66 Serial.println(" - GREEN detected!");
67 }
68 if(bluecolor > redcolor && bluecolor > greencolor){
69 Serial.println(" - BLUE detected!");
70 }
71 }

```

Penggalan kode program pada Tabel 5.17 bertujuan membaca nilai tiap fotodiode merah, hijau, dan biru serta menyimpan nilainya pada array. Pada baris 1-8 digunakan untuk inialisasi pin input output pada TCS 3200 serta mengatur *Output Frequency Scaling* di nilai 20%. Pada baris 11-26 digunakan untuk membaca nilai fotodiode merah dan menyimpannya pada array `datasensor[0]`. Pada baris 29-43 digunakan untuk membaca nilai fotodiode hijau dan menyimpannya pada array `datasensor[1]`. Pada baris 46-60 digunakan untuk membaca nilai fotodiode biru dan menyimpannya pada array `datasensor[2]`.

Penggalan kode program yang terdapat pada Tabel 5.18 merupakan penerapan program dari sensor MAX 30100.

**Tabel 5.18 Kode Inisialisasi Library dan Variabel**

No	Kode Program
1	#include <CircularBuffer.h>
2	#include <MAX30100.h>
3	#include <MAX30100_BeatDetector.h>
4	#include <MAX30100_Filters.h>
5	#include <MAX30100_PulseOximeter.h>
6	#include <MAX30100_Registers.h>
7	#include <MAX30100_SpO2Calculator.h>
8	#include <Wire.h>
9	#include "MAX30100_PulseOximeter.h"
10	#define REPORTING_PERIOD_MS 10000
11	PulseOximeter pox;
12	uint32_t tsLastReport = 0;
13	void onBeatDetected()
14	{
15	Serial.println("Beat!");
16	}

Penggalan kode program pada Tabel 5.18 berisikan tentang library yang akan digunakan, sebelumnya akan didefinisikan pada baris 1-12. Baris 13-16 bertujuan mendeteksi apakah ada detak jantung pada sensor.

Tabel 5.19 Kode Mendeteksi Detak Jantung

No	Kode Program
1	void setupppg()
2	{
3	Serial.print("Initializing pulse oximeter..");
4	if (!pox.begin()) {
5	Serial.println("FAILED");
6	for(;;);
7	} else {
8	Serial.println("SUCCESS");
9	}
10	pox.setOnBeatDetectedCallback(onBeatDetected);
11	}



Kemudian pada Tabel 5.19 di baris 3-9 digunakan untuk inialisasi MAX30100, dan apabila berhasil akan menampilkan tulisan “SUCCESS”, apabila gagal akan menampilkan tulisan “FAILED” pada serial monitor. Baris ke 10 memiliki fungsi untuk mendeteksi apakah ada detak jantung yang terdeteksi, serial monitor akan menampilkan kata “Beat!” apabila terdeteksi detak jantung.

Tabel 5.20 Kode Menghitung dan Menampilkan Spo2

No	Kode Program
1	<code>void loopppg ()</code>
2	<code>{</code>
3	<code> pox.update();</code>
4	<code> if (millis() - tsLastReport > REPORTING_PERIOD_MS) {</code>
5	<code> Serial.print("SpO2:");</code>
6	<code> Serial.print(pox.getSpO2());</code>
7	<code> Serial.println("%");</code>
8	<code> float getppg = pox.getSpO2();</code>
9	<code> datasensor[3] = getppg;</code>
10	<code> tsLastReport = millis();</code>
11	<code> }</code>
12	<code>}</code>

Dan pada Tabel 5.20 memiliki fungsi untuk membaca nilai kadar oksigen dalam darah dan menyimpannya di array `datasensor[3]`.

Tabel 5.21 menampilkan penerapan dari pembacaan sensor MLX 90614.

Tabel 5.21 Kode Pengambilan Data Sensor MLX 90614

No	Kode Program
1	<code>#include <Wire.h></code>
2	<code>#include <Adafruit_MLX90614.h></code>
3	<code>Adafruit_MLX90614 mlx = Adafruit_MLX90614();</code>

Penggalan kode program pada Tabel 5.21 bertujuan untuk mendefinisikan setiap *library* dalam kode bertuliskan `#include` untuk megoperasikan MLX 90614.

Tabel 5.22 Kode Melakukan Pengukuran Suhu Tubuh

No	Kode Program
----	--------------



```

1 void setupsuhu() {
2     mlx.begin();
3 }
    
```

Kemudian pada Tabel 5.22 akan memanggil fungsi *begin* dari sensor untuk melakukan pengukuran.

Tabel 5.23 Kode Mendapatkan nilai Suhu

No	Kode Program
1	<code>void loopsuhu() {</code>
2	<code>float getsuhu = mlx.readObjectTempC();</code>
3	<code>datasensor[4] = getsuhu;</code>
4	<code>delay(500);</code>
5	<code>}</code>

Penggalan kode program pada Tabel 5.23 bertugas untuk menembakkan sinar infra merah pada MLX 90164 agar dapat melakukan pengukuran suhu tubuh yang akan disimpan pada *array* `datasensor[4]`.

Penggalan kode program pada Tabel 5.24 merupakan penerapan program dari sensor *Flex*.

Tabel 5.24 Kode Pengambilan Data Flex Sensor

No	Kode Program
1	<code>int flexPin = A7;</code>
2	<code>int i, value, low, high, hasil, temp, flex[10];</code>

Penggalan kode program pada Tabel 5.24 bertujuan untuk mendefinisikan susunan pin sesuai dengan rancangan sebelumnya serta terdapat variabel array untuk menentukan nilai terbesar dan terkecil dari *Flex* sensor.

Tabel 5.25 Kode Perhitungan Flex Sensor

No	Kode Program
1	<code>void loopflex() {</code>
2	<code>for (i = 0; i < 10; i++) {</code>
3	<code>value = analogRead(flexPin);</code>
4	<code>flex[i] = value;</code>
5	<code>delay(500);</code>



```

6     }
7     high = flex[0];
8     low = flex[0];
9     for (i = 0; i < 10; i++){
10    if (high < flex[i]){
11        high = flex[i];
12    }
13    else if (low > flex[i]){
14        low = flex[i];
15    }
16    }
17    hasil = high - low;
18    datasensor[5] = hasil;
19 }
    
```

Pada Tabel 5.25 akan dilakukan perhitungan dan pengambilan nilai data sensor dari nilai yang tertinggi dan terendah dari setiap pengambilan nilai sensor, kemudian nilai tersebut akan cari selisihnya kemudian akan disimpan nilai selisihnya pada array datasensor[5].

5.2.3.3 Implementasi Kode Program Klasifikasi K-NN

Penerapan kode program pada klasifikasi K-NN akan dilakukan apabila dari tiap sensor sudah mempunyai nilai dari masing-masing pembacaan sesuai pada rancangan sebelumnya yang akan digunakan pada klasifikasi K-NN.

Penggalan kode program pada Table 5.26 merupakan penerapan dari program klasifikasi K-NN.

Tabel 5.26 Implementasi Kode Program Klasifikasi K-NN

No	Kode Program
1	float datasensor[6];
2	float range[30][2];
3	int DataJarak = 0;
4	int label = 1;

Penggalan kode program pada Tabel 5.26 bertujuan untuk mendefinisikan tiap variabel yang dibutuhkan pada penerapan program klasifikasi K-NN. Pada baris 1 digunakan untuk menyimpan nilai hasil pengukuran tiap sensor. Pada baris 2 digunakan untuk menentukan banyaknya data latih yang akan diberi label.

**Tabel 5.27 Kode Perhitungan Euclidean Distance**

No	Kode Program
1	for(int a=0;a<30;a++){
2	range[a][0] = 0; //reset data
3	for(int b=0;b<6;b++){
4	range[a][0] += pow(DataLatih (a, b) -
5	datasensor[b], 2);
6	}
7	range[a][0] = sqrt(range[a][0]); //akar
8	switch(a){
9	case 0 ... 12:
10	range[a][1] = 0; //angka 0 untuk label normal
11	break;
12	case 13 ... 21:
13	range[a][1] = 1; //angka 1 untuk label TB
14	break;
15	case 22 ... 29:
16	range[a][1] = 2; //angka 2 untuk label Pneumonia
17	break;
18	}

Penggalan kode program pada Tabel 5.27 akan memperhitungkan tiap jarak data latih ke data yang sedang diujikan menggunakan Persamaan (1) di sub bab 2.2.3. Pada baris 1-7 penerapan program lunak perhitungan data latih ke satu uji yang sedang diujikan oleh K-NN. Pada baris 8-18 digunakan untuk memberi label dengan rincian angka 0 untuk label normal, angka 1 untuk label TB, dan angka 2 untuk label pneumonia.

Tabel 5.28 Kode Sorting Data

No	Kode Program
1	//SORTING DATA
2	for(int i=0;i<29;i++){
3	for(int j=i+1;j<30;j++){
4	if(range[i][DataJarak]>range[j][DataJarak]){
5	float temp = range[i][DataJarak];
6	range[i][DataJarak] = range[j][DataJarak];
7	range[j][DataJarak] = temp;
8	}



```

9      temp = range[i][label];
10     range[i][label] = range[j][label];
11     range[j][label] = temp;
12     }
13     }
14     }
    
```

Penggalan kode program pada Tabel 5.28 bertugas untuk mengurutkan hasil dari perhitungan jarak *Euclidean* dari jarak terdekat ke data uji, sampai jarak terjauh ke data uji.

Tabel 5.29 Kode Melakukan Weighted Voting

No	Kode Program
1	float nol = 0, satu = 0, dua = 0;
2	String hasil;
3	for(int a=0; a<3; a++){
4	if(range[a][label] == 0.00){
5	nol += 1/ pow(range[a][DataJarak], 2);
6	}
7	else if(range[a][label] == 1.00){
8	satu += 1/ pow(range[a][DataJarak], 2);
9	}
10	else if(range[a][label] == 2.00){
11	dua += 1/ pow(range[a][DataJarak], 2);
12	}
13	}

Pada Tabel 5.29 memiliki fungsi untuk menentukan nilai k pada baris 3 dan melakukan voting secara *weighted* voting. Variabel nol diperuntukkan untuk data normal, variabel satu diperuntukkan data TB, dan variabel dua diperuntukkan data Pneumonia.

Tabel 5.30 Kode Hasil Klasifikasi K-NN

No	Kode Program
1	if(nol > satu && nol > dua){
2	hasil = "NORMAL";
3	}
4	else if(satu > nol && satu > dua){
5	hasil = "TB";



```

6   }
7   else if (dua > satu && dua > nol) {
8       hasil = "PNEUMONIA";
9   }
    
```

Kemudian Tabel 5.30 akan menentukan hasil dari klasifikasi yang apabila variabel nol lebih besar daripada variabel satu dan dua akan menghasilkan klasifikasi "NORMAL", apabila variabel satu lebih besar dari pada nol dan dua akan menghasilkan klasifikasi "TB", dan apabila variabel dua lebih besar daripada variabel satu dan nol akan menghasilkan klasifikasi "PNEUMONIA".



BAB 6 PENGUJIAN DAN ANALISIS

Bab pengujian dan analisis merupakan bab yang akan membahas proses dari beberapa pengujian yang kemudian akan dianalisis berdasarkan rancangan dan penerapan sebenarnya dari sistem ini. Bab ini bertujuan untuk melakukan pengecekan ulang terhadap sistem apakah keluaran dari sistem sudah sesuai dengan ekspektasi atau belum.

6.1 Pengujian sensor warna TCS 3200

Pada pengujian sensor TCS 3200 akan dilakukan pengujian terhadap TCS 3200 untuk membaca nilai RGB dari warna kuku, disaat yang sama akan dilakukan juga pembacaan dari perangkat lunak *instant eyedropper* terhadap warna kuku yang sama dengan pembacaan yang dilakukan TCS 3200. Kemudian akan dibandingkan selisihnya untuk mendapatkan nilai *error* yang terjadi.

6.1.1 Tujuan Penelitian

Tujuan dari pengujian ini adalah untuk mengetahui tingkat *error* yang didapat apabila membandingkan nilai dengan pembacaan RGB dari *instant eyedropper*. Sistem akan mengalami pengaruh negatif apabila tingkat *error* dari TCS 3200 memiliki nilai yang besar.

6.1.2 Prosedur Pengujian

Untuk melakukan pengujian, dibutuhkan prosedur agar dapat berhasil melakukan pengujian sesuai yang direncanakan :

1. Pastikan sensor TCS 3200 terhubung dengan Arduino Mega.
2. Unggah kode program TCS 3200 yang sudah diterapkan di Arduino IDE.
3. Masukkan jari jempol ke lubang yang terdapat pada sistem sesuai dengan penerapan prototipe sistem.
4. Ambil gambar dari warna kuku setelah pembacaan sensor berakhir untuk dideteksi ulang oleh *instant eyedropper*.
5. Bandingkan nilai RGB yang dideteksi oleh TCS 3200 dengan *instant eyedropper*. Agar dapat membandingkan nilai RGB dari masing masing pembacaan, nilai RGB yang didapat sebelumnya akan diubah menjadi nilai hex agar dapat mudah dicari selisihnya.
6. Kemudian akan dihitung perhitunagn persentase *error* berdasarkan pembacaan dari TCS 3200 dan *instant eyedropper* dengan mencari selisih dari pembacaan kedua sensor, kemudian dibagi oleh nilai yang terdeteksi *instant eyedropper* seperti yang tertulis pada persamaan (6.1).

$$Error = \frac{\text{Selisih Error}}{\text{Nilai instant eyedropper}} \times 100\% \quad (6.1)$$

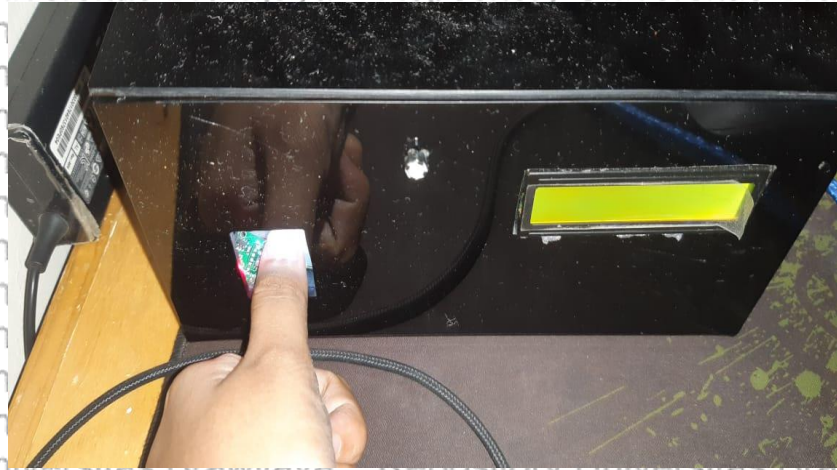


6.1.3 Hasil dan Analisis Pengujian

Setelah nilai HEX hasil pembacaan TCS 3200 dan *instant eyedropper* sudah didapat nilai selisihnya, maka akan mendapatkan hasil rata-rata *error* sebesar 0.36% pada Tabel 6.1. Akan tetapi, pembacaan nilai sensor akan mengalami perubahan nilai yang besar apabila subjek yang diukur tangannya melakukan pergerakan atau adanya interferensi cahaya dari luar. Tampak dari Gambar 6.1 menunjukkan posisi dari kuku saat pengukuran dengan cara menghadapkan keatas lubang agar sensor TCS 3200 yang menghadap ke bawah lubang dapat membaca nilai RGB dari warna kuku.

Tabel 6.1 Hasil Pengujian Sensor Warna TCS 3200

No	Warna Kuku	Pembacaan Sensor					<i>Instant eyedropper</i>					Selisih error		Error
		R	G	B	Hex	Warna	R	G	B	Hex	Warna	Hex	Dec	
1		217	204	199	d9cc7		217	179	178	d9b3b2		1915	6421	0.04%
2		195	183	179	c3b7b3		190	180	172	beb4ac		50307	328455	2.62%
3		228	187	184	e4bbb8		228	177	173	e4b1ad		A0B	2571	0.01%
4		161	158	159	a19e9f		159	109	104	9f6d68		23137	143671	1.37%
5		200	153	148	c89994		194	154	154	c29a9a		5FEFA	392954	3.07%
6		222	213	174	ded5ae		221	190	188	ddbec		116F2	71410	0.49%
7		183	132	123	b7847b		183	141	143	b78d8f		-914	-2324	0.01%
8		217	153	210	d999d2		216	181	179	d8b5b3		E41F	58399	0.41%
9		167	119	117	a77775		168	120	118	a87876		-10101	-65793	0.59%
10		111	87	82	6f5752		125	76	72	7d4c48		-DF4F6	914678	11.1%
Rata-rata													0.36%	



Gambar 6.1 Pengambilan Nilai RGB dengan Sensor TCS 3200

6.2 Pengujian Sensor Suhu MLX 90614

Pada pengujian sensor MLX 90614 akan dilakukan pengujian pada sensor MLX 90614 untuk membaca nilai suhu tubuh. Kemudian akan dilakukan juga pembacaan nilai suhu tubuh menggunakan termometer dengan subjek pengukuran yang sama dengan subjek yang diukur saat pengukuran MLX 90614.

6.2.1 Tujuan Pengujian

Tujuan pengujian yang dilakukan terhadap MLX 90614 untuk mengetahui rasio *error* yang didapat dengan membandingkan hasil pengukuran terhadap alat pengukuran suhu konvensional yaitu termometer.

6.2.2 Prosedur Pengujian

Untuk melakukan pengujian, dibutuhkan prosedur agar dapat berhasil melakukan pengujian sesuai yang direncanakan :

1. Pastikan sensor MLX 90614 terhubung dengan Arduino Mega.
2. Unggah kode program MLX 90614 yang sudah diterapkan di Arduino IDE.
3. Dekatkan sensor MLX 90614 dengan dahi subjek yang akan diukur dan tunggu hasil dari suhu tubuh pada serial monitor terlihat stabil.
4. Mengukur nilai suhu dari subjek yang sama saat pengukuran suhu yang dilakukan sensor MLX 90614 menggunakan termometer yang diletakkan pada ketiak.
5. Membandingkan pembacaan nilai suhu sensor MLX 90614 dengan nilai suhu termometer.
6. Hitung nilai persentase *error* berdasarkan nilai selisih dari pembacaan termometer dikurangi nilai dari pembacaan MLX 90614 kemudian dibagi nilai dari pembacaan termometer seperti yang tertulis pada persamaan 6.2

$$Error = \frac{Selisih\ Error}{Nilai\ Termometer} \times 100\% \dots\dots\dots (6.2)$$



6.2.3 Hasil dan Analisis Pengujian

Setelah dilakukan prosedur pengujian untuk mengukur suhu tubuh menggunakan sensor MLX 90614 dan termometer, maka selanjutnya akan dihitung selisih *error* dan persentase *error*. Gambar 6.2 menunjukkan penerapan dari pengukuran suhu tubuh melalui sensor MLX 90164.

Tabel 6.2 Hasil Pengujian Sensor Suhu MLX 90614

Pengujian ke - n	Termometer (°C)	Sensor Suhu MLX 90614 (°C)	Selisih <i>error</i> (°C)	Persentase <i>error</i> %
1	36.9	36	0.9	2.43%
2	36.3	36.3	0	0%
3	36.3	36.2	0.1	0.27%
4	37	36.6	0.4	1.08%
5	36.3	36.2	0.1	0.27%
6	36.2	36.3	0.1	0.27%
7	36.7	36.3	0.4	1.08%
8	36.7	36.5	0.2	0.54%
9	36.6	36.3	0.3	0.81%
10	36.7	36.4	0.3	0.81%
Rata-rata				0.75%



Gambar 6.2 Pengambilan nilai Suhu Tubuh dengan Sensor MLX 90614

Sehabis didapatkan selisih pembacaan nilai suhu sensor dengan nilai suhu termometer, maka akan mendapatkan rata-rata *error* sebesar 0.75% pada Tabel



6.2. Dapat dikatakan tingkat *error* saat pengujian cukup baik karena terbilang kecil dan *error* yang didapatkan disebabkan karena perbedaan peletakan sensor dan peletakan termometer, untuk termometer langsung ditempelkan di ketiak dari subjek yang diukur, sedangkan untuk sensor diletakkan didepan dahi karena sensor melakukan pengukuran suhu dengan cara menembakan sinar Infra merah.

6.3 Pengujian Sensor MAX 30100

Akan dilakukan pengujian fungsionalitas terhadap MAX 30100 dengan melakukan pengukuran tingkat kadar oksigen dalam darah.

6.3.1 Tujuan Pengujian

Pengujian fungsionalitas dilakukan sensor MAX 30100 dalam melakukan pengukuran kadar oksigen dalam darah, agar mendapati apakah MAX 30100 beroperasi sesuai fungsionalitasnya atau tidak.

6.3.2 Prosedur Pengujian

Dibutuhkan serangkaian prosedur yang harus dilakukan apabila ingin mengukur tingkat kadar oksigen dalam darah. Berikut adalah prosedur dalam melakukan pengujian mendeteksi kadar oksigen dalam darah :

1. Pastikan sensor MAX 30100 terhubung dengan Arduino Mega.
2. Unggah kode program sensor MAX 30100 yang sudah diterapkan pada Arduino IDE.
3. Masukkan jempol kelubang yang terdapat pada sistem sesuai dengan penerapan prototipe sistem.
4. Membuka serial monitor pada Arduino IDE untuk melihat hasil kadar oksigen dalam darah.
5. Simpan hasil dari pengukuran kadar oksigen dalam darah pada subjek yang diperiksa.

6.3.3 Hasil dan Analisis Pengujian

Selepas dilakukannya serangkaian prosedur untuk mendeteksi kadar oksigen dalam darah, akan mendapatkan hasil dari sensor MAX 30100. Tabel 6.3 menunjukkan hasil dari pengujian fungsionalitas yang dilakukan terhadap sensor MAX 30100. Gambar 6.3 merupakan prosedur pengambilan nilai kadar oksigen dalam darah menggunakan sensor MAX 30100.

Tabel 6.3 Hasil Pengujian Mendeteksi Kadar Oksigen Dalam Darah

No	Kadar Oksigen Dalam Darah (SPO2)	Kelas
1	97%	Normal
2	96%	Normal
3	95%	Normal

4	94%	Normal
5	94%	TB
6	94%	TB
7	93%	TB
8	93%	Pneu
9	93%	Pneu
10	92%	Pneu



Gambar 6.3 Pengambilan nilai Spo2 dengan Sensor MAX 30100

Pada Tabel 6.3, sensor MAX 30100 dapat mendeteksi perbedaan kadar oksigen dalam darah pada orang dengan kondisi paru-paru normal dengan orang berpenyakit TB dan pneumonia. Hasil kadar oksigen dalam darah yang dideteksi Max 30100 pada orang normal lebih tinggi dari pada orang yang mempunyai penyakit TB dan Pneumonia.

6.4 Pengujian Flex Sensor

Akan dilakukan pengujian fungsionalitas terhadap *flex sensor* dengan mengukur tingkat respirasi.

6.4.1 Tujuan Pengujian

Pengujian fungsionalitas dilakukan *flex sensor* dalam mendeteksi tingkat respirasi, agar mendapati apakah *flex sensor* beroperasi sesuai fungsionalitasnya atau tidak.

6.4.2 Prosedur Pengujian

Dibutuhkan serangkaian prosedur yang harus dilakukan apabila ingin mengukur tingkat respirasi. Berikut adalah prosedur dalam melakukan pengujian mengukur tingkat respirasi :



1. Pastikan *flex sensor* terhubung dengan rangkaian OP-AMP.
2. Menghubungkan rangkain OP-AMP dengan Arduino Mega.
3. Unggah kode program *Flex sensor* yang sudah diterapkan di Arduino IDE.
4. Mengikatkan *Flex sensor* ke dada subjek yang diteliti dengan sabuk karet.
5. Pasien akan diukur saat bernafas dalam-dalam dan saat menghembuskan napas dalam-dalam.
6. Mebuka serial monitor untuk melihat tingkat respirasi pada permagkat lunak Arduino IDE.
7. Mencatat hasil dari tingkat respirasi subjek yang diteliti.

6.4.3 Hasil dan Analisis Pengujian


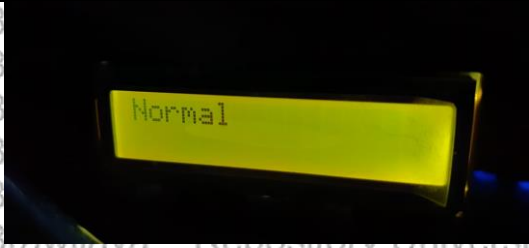
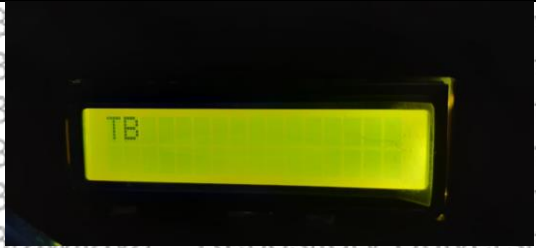
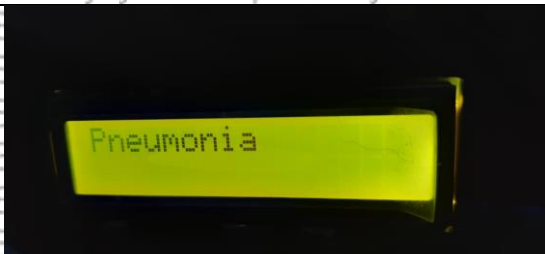
Selepas dilakukannya serangkaian prosedur pengujian untuk mengukur tingkat respirasi, serial monitor Arduino IDE akan menampilkan hasilnya. Tabel 6.4 menunjukkan hasil dari pengujian fungsionalitas yang dilakukan terhadap *flex sensor*. Gambar 6.4 menampilkan proses pengambilan nilai tingkat respirasi menggunakan *flex sensor*.

Tabel 6.4 Hasil Pengujian Mengukur Tingkat Respirasi

No	Tingkat Respirasi	Kelas
1	60	Normal
2	42	Normal
3	41	Normal
4	40	Normal
5	47	TB
6	20	TB
7	19	TB
8	15	Pneu
9	14	Pneu
10	12	Pneu

Pada Tabel 6.4, nilai dari *Flex sensor* berupa resistansi yang sebelumnya diubah menjadi tegangan melalui jembatan *wheatstone* dan diberi penguatan 3 kali dengan rumus persamaan 2 pada sub bab 5.1.2, kemudian nilai tegangan diubah menjadi nilai digital. Nilai dari tingkat respirasi pada orang normal pada pengujian ini berkisar 40-60, sedangkan pada penyakit TB dan Pneumonia lebih rendah dibandingkan pada orang normal yaitu berkisar 12-47. Pada orang normal nilai yang didapat lebih besar karena pada paru-paru sehat dapat menghirup

Tabel 6.5 Hasil Pengujian Tampilan LCD 16x2

No	Kondisi	Tampilan	Keterangan
1	Sleep		Kondisi <i>Sleep</i> ditandai dengan muncul tulisan "Tekan Tombol" pada LCD
2	Hasil Klasifikasi		Hasil Klasifikasi Paru-paru Normal
			Hasil Klasifikasi Pru-paru penyakit TB
			Hasil Klasifikasi Paru-paru penyakit Pneumonia

6.6 Pengujian Akurasi Hasil Klasifikasi K-NN

6.6.1 Tujuan Pengujian

Tujuan pengujian dari sistem pendeteksi kondisi paru-paru ialah untuk mengetahui keakuratan klasifikasi K-NN terhadap penentuan kelas kondisi paru-paru yang dipakai pada sistem.

6.6.2 Prosedur Pengujian

Serangkaian prosedur harus dilakukan apabila ingin mengetahui waktu yang dibutuhkan sistem untuk menentukan kelas kondisi paru-paru. Urutan dari prosedur pengujian waktu ialah :



1. Pastikan Sistem terhubung dengan laptop yang terdapat Arduino IDE dan kode program.
2. Unggah kode program yang digunakan untuk menentukan hasil kondisi paru-paru.
3. Lakukan pembacaan nilai pada kadar oksigen dalam darah, membaca nilai R,G,B yang didapatkan pada warna kuku, lakukan pendeteksian pada suhu tubuh, dan mendeteksi tingkat respirasi. Akan diperoleh 6 nilai yang telah dibaca oleh sistem untuk diolah dengan memakai metode K-NN.
4. Membandingkan hasil kelas yang ditampilkan LCD terhadap kelas yang telah ditentukan sebelumnya oleh diagnosis pakar di Puskesmas Janti, Kota Malang. Keakuratan yang didapat pada sistem ini, dihitung berdasarkan persamaan 6.6

$$Akurasi = \frac{Jumlah\ Data - Jumlah\ data\ yang\ tidak\ sesuai}{Jumlah\ Data} \times 100\% \dots\dots\dots (6.6)$$

6.6.3 Hasil dan Analisis Pengujian

Rincian keakuratan dari pengujian penentuan kelas menggunakan klasifikasi K-NN akan dimuat pada Tabel 6.6. Terdapat tiga nilai K yang digunakan pada pengujian ini, yaitu K dengan nilai 3,5, dan 7. Digunakan bervariasinya nilai K bertujuan agar dapat menentukan banyaknya kelas terdekat yang akan dipakai sebagai acuan untuk menentukan kelas yang baru. Gambar 6.5 menampilkan saat melakukan prosedur pengujian.

Tabel 6.6 Hasil Pengujian Akurasi Klasifikasi K-NN

No	R	G	B	SpO2	Suhu	Flex	Kelas	K=3	K=5	K=7
1	195	160	165	94	36	60	Normal	Normal	Normal	Normal
2	199	159	159	96	36,3	37	Normal	Normal	Normal	Normal
3	200	153	148	95	36,6	42	Normal	Normal	Normal	Normal
4	200	161	164	94	36,3	40	Normal	Normal	Normal	Normal
5	201	151	150	96	36,3	33	Normal	Normal	Normal	Normal
6	202	166	168	95	36,3	56	Normal	Normal	Normal	Normal
7	204	171	157	97	36,3	33	Normal	Normal	Normal	Normal
8	221	190	188	93	36,9	31	TB	TB	TB	TB
9	222	197	174	94	36,7	35	TB	TB	TB	TB
10	222	192	194	95	36,6	29	TB	TB	TB	TB
11	225	244	277	94	37,4	30	TB	Pneu	Pneu	Pneu
12	233	211	194	94	36,9	27	Pneu	Pneu	Pneu	Pneu
13	237	254	267	93	38,3	26	Pneu	Pneu	Pneu	Pneu
14	238	186	181	93	36,8	23	Pneu	Pneu	Pneu	Pneu



Gambar 6.5 Pengujian Akurasi Klasifikasi K-NN

Dari 14 data yang digunakan untuk pengujian pada Tabel 6.6, 1 data hasil diagnosis kelas yang dilakukan sistem tidak sesuai dengan hasil diagnosis yang ditentukan oleh pakar. Terjadinya ketidaksesuaian kelas dikarenakan terdapat kesamaan nilai yang dideteksi antara data TB dan Pneumonia. Ketidaksesuaian yang terjadi menjadikan keakuratan sistem bernilai 92,8%.

$$\begin{aligned} \text{Akurasi} &= \frac{\text{Jumlah Data} - \text{Jumlah data yang tidak sesuai}}{\text{Jumlah Data}} \times 100\% \\ &= \frac{14-1}{14} \times 100\% \\ &= 92,8\% \end{aligned}$$

6.7 Pengujian Waktu Komputasi Program

6.7.1 Tujuan Pengujian

Pengujian ini bertujuan untuk menguji waktu yang dibutuhkan sistem mulai dari melakukan pembacaan nilai masing-masing sensor sampai sistem menentukan kelas dari kondisi paru-paru. Pengujian ini juga bertujuan menjawab masalah yang telah dirumuskan pada penelitian ini.

6.7.2 Prosedur Pengujian

Adapun prosedur pengujian apabila ingin mengetahui lama waktu komputasi dari sistem pendeteksi paru-paru. Berikut adalah prosedur pengujian yang harus dilakukan :

1. Pastikan sistem terhubung dengan laptop yang terdapat Arduino IDE dan kode program.



2. Unggah kode program untuk sistem pendeteksi kondisi paru-paru.
3. Membuka serial monitor untuk melihat waktu komputasi sistem yang terdapat pada perangkat lunak Arduino IDE.

6.7.3 Hasil dan Analisis Pengujian

Tabel 6.7 Hasil Waktu Komputasi Sistem Pendeteksi Kondisi Paru-Paru

No	Pengujian Ke-n	Waktu Komputasi (ms)
1	Pengujian Ke-1	1029
2	Pengujian Ke-2	1030
3	Pengujian Ke-3	1030
4	Pengujian Ke-4	1035
5	Pengujian Ke-5	1030
6	Pengujian Ke-6	1030
7	Pengujian Ke-7	1030
8	Pengujian Ke-8	1030
9	Pengujian Ke-9	1030
10	Pengujian Ke-10	1031
	Rata-rata	1030,5

Dari serangkaian prosedur yang dilakukan, akan menghasilkan waktu yang dibutuhkan sistem dalam menentukan kelas kondisi paru-paru. Rata-rata waktu yang dibutuhkan sistem untuk menentukan kelas kondisi paru-paru dari 10 pengujian yang dilakukan bernilai 1030,5 ms atau 1,305 detik.



BAB 7 KESIMPULAN

Dari pengujian yang telah dilakukan, akan dihasilkan beberapa kesimpulan hasil dari serangkaian pengujian yang telah dilakukan. Terdapat saran yang dapat digunakan oleh peneliti selanjutnya untuk melakukan penelitian serupa kelak.

7.1 Kesimpulan

Untuk menjawab masalah yang telah dirumuskan sebelumnya, akan dilakukan pengujian untuk menghasilkan kesimpulan dari penelitian ini. Maka terdapat beberapa hasil kesimpulan yang didapatkan yaitu :

1. Keakuratan pembacaan dari sensor TCS 3200 dan MLX 90614 pada penelitian ini memiliki rata-rata rasio *error* sebesar 0,36% dan 0,75%. Adapun untuk sensor MAX 30100 mampu beroperasi sesuai fungsionalitasnya dalam mendeteksi kadar oksigen dalam darah dengan nilai 92%-94% untuk kondisi paru-paru Pneumonia dan TB, sedangkan 94%-97% untuk kondisi paru-paru normal, *Flex* sensor juga dapat membaca nilai tingkat respirasi sesuai yang diharapkan dengan nilai 12-47 untuk kondisi paru-paru Pneumonia dan TB, sedangkan 40-60 untuk kondisi paru-paru normal.
2. Untuk mengenali kondisi paru-paru TB, Pneumonia dan normal dari suhu tubuh, pada orang dengan kondisi paru-paru mempunyai penyakit TB dan Pneumonia sering mengalami demam sehingga suhu tubuh lebih tinggi dari suhu tubuh orang yang mempunyai paru-paru normal. Sedangkan untuk warna kuku, orang dengan kondisi paru-paru TB dan Pneumonia mempunyai warna kuku pucat atau kusam sedangkan pada orang normal bewarna merah cerah atau keunguan. Untuk kadar oksigen dalam darah pada kondisi paru-paru TB dan Pneumonia bernilai 92%-94%, sedangkan untuk kondisi paru-paru normal bernilai 94%-97%. Dan untuk tingkat respirasi pada kondisi paru-paru TB dan Pneumonia mempunyai nilai lebih kecil dari kondisi normal yaitu sebesar 12-47, sedangkan kondisi paru-paru normal sebesar 40-60.
3. Sebanyak 14 data yang diujikan, hanya 1 data yang memiliki ketidaksesuaian kelas terhadap kelas diagnosis yang dilakukan oleh pakar. Terdapat tiga nilai K yang berbeda dalam pengujian ini. K yang dipakai bernilai 3,5, dan 7. Hasil dari pengujian, mendapati keakuratan sistem bernilai 92,8% sebagai banyaknya referensi kelas yang diambil untuk menentukan kelas selanjutnya. Kemudian dari 10 kali pengujian yang dilakukan untuk mengukur waktu komputasi, didapatkan nilai rata-rata waktu yang dibutuhkan untuk menentukan kelas kondisi paru-paru dari pembacaan nilai masing-masing sensor sebesar 1030,5 ms dengan rincian waktu komputasi tercepat dengan waktu 1029 ms dan terlama 1035 ms.



7.2 Saran

Setelah dihasilkan kesimpulan, dihasilkan juga saran dari penelitian ini. Anjuran yang diberikan bertujuan untuk memudahkan peneliti selanjutnya yang akan melakukan penelitian serupa kelak. Beberapa saran yang dihasilkan dari dilakukannya penelitian ini ialah:

1. Desain sistem tanpa adanya cahaya yang masuk dari luar saat melakukan pengukuran nilai R,G,B.
2. Dapat digunakan secara *wearable*.



DAFTAR PUSTAKA

- Aqeel, A., 2018. *Introduction to Arduino Mega 2560*. [Online] Available at: <https://www.theengineeringprojects.com/2018/06/introduction-to-arduino-mega-2560.html> [Accessed 15 April 2020].
- Chu, M. et al., 2019. Respiration rate and volume measurements using wearable strain sensors. *Nature Partner Journal*, pp. 2-9.
- Enda, D., 2015. *Portfolio Devandi Enda*. [Online] Available at: http://depandienda.it.student.pens.ac.id/file/knn_references.pdf [Accessed 26 August 2019].
- Foxlin, E. & Naimark, L., 2003. *VIS-Tracker: a wearable vision-inertial self-tracker*. Los Angeles, IEEE.
- Kurniawan, D., Maulana, R. & Hanafi, M. H., 2019. Implementasi Pendeteksi Penyakit Paru-Paru Berdasarkan Warna Kuku dan Suhu Tubuh Berbasis Sensor TCS3200 Dan Sensor LM35 dengan Metode Naïve Bayes. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, III(4), p. 3383.
- K., Wardana, P. S. & Adil, R., 2017. Spirometer Non-Invasive dengan Sensor Piezoelektrik untuk Deteksi Kesehatan Paru-Paru. *Jurnal ELKOMIKA*, V(2), pp. 188-206.
- Mayer, L., 2014. Nails as Indicator of Health Status.
- Musa, O. & A., 2017. Analisis Penyakit Paru-Paru Menggunakan Algoritma K-Nearest Neighbors Pada Rumah Sakit Aloe Saboe Kota Gorontalo. *ILKOM Jurnal Ilmiah*, IX(3), pp. 348-352.
- Sejati, A. & Sofiana, L., 2015. Faktor-Faktor Terjadinya Tuberkulosis. *Jurnal Kesehatan Masyarakat*, X(2), pp. 122-128.
- Setyobudi, D. B., Maulana, R. & Fitriyah, H., 2019. Sistem Deteksi Gejala Hipoksia Berdasarkan Saturasi Oksigen dan Detak Jantung Menggunakan Metode Fuzzy Berbasis Arduino. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, III(2), pp. 1925-1933.
- Sibuea, M. O., 2018. *Pengukuran Suhu Dengan Sensor Suhu Inframerah MLX 90614 Berbasis Arduino*. [Online] Available at: https://repository.usd.ac.id/34082/2/135114047_full.pdf [Accessed 15 April 2020].
- Syed, A., Agasbal, Z. T. H., Melligeri, T. & Gudur, B., 2012. Flex Sensor Based Robotic Arm Controller Using Microcontroller. *Journal of Software Engineering and Applications*, Issue 5, p. 365.



TAOS, n.d. [Online]
Available at: <https://pdf1.alldatasheet.com/datasheet-pdf/view/454462/TAOS/TCS3200.html>
[Accessed 29 August 2019].

Tutorials, E., 2014. *The Differential Amplifier*. [Online]
Available at: https://www.electronics-tutorials.ws/opamp/opamp_5.html
[Accessed 13 April 2020].

Wardlaw, T. Johansson, E. W. & Hodge, M., 2006. *Pneumonia The Forgotten Killer of Children*, New York: WHO(World Health Organization).

Yunus, M. & Setyowibowo, S., 2010. Aplikasi Sistem Pendukung Keputusan Diagnosa Penyakit Paru-Paru Dengan Metode Forward Chaining. *Jurnal Teknologi Informasi*, 11(2), p. 95.



LAMPIRAN A HASIL WAWANCARA DENGAN PAKAR PENYAKIT PARU

A. Latar Belakang Wawancara

Kegiatan ini dilatarbelakangi oleh tugas skripsi yang mengharuskan saya untuk mewawancarai seseorang yang ahli atau pakar di bidang penyakit paru-paru, untuk nantinya akan dijadikan referensi dari laporan skripsi.

B. Tujuan Wawancara

Tujuan dari wawancara ini adalah untuk mencari tahu adakah hubungan antara seorang pasien penderita TB paru-paru dengan warna kuku, suhu tubuh, kadar oksigen dalam darah dan tingkat respirasinya.

C. Pokok bahasan

1. Gejala yang dialami oleh penderita TB
2. Hubungan pasien TB paru-paru dengan parameter penelitian

D. Waktu dan Tempat Wawancara

Wawancara dilaksanakan pada hari Senin (02/03/2020), yang bertempat di Puskesmas Janti, Kota Malang.

E. Hasil Wawancara

1. Narasumber : Jukhriyah, Amd. Kep (Kepala Klinik TB)
2. Pewawancara : Alfatehan Arsyah Baharin
3. Hasil wawancara :

Pertanyaan : Dengan ibu siapa saya bicara?

Jawab : Julia.

Pertanyaan : Ibu bekerja disini sebagai apa?



Jawab : Sebagai Kepala Klinik TB.

Pertanyaan : Apa sajakah gejala TB paru?

Jawab : Kalau gejalanya batuk berdahak atau tidak berdahak, berat badan menurun tanpa sebab, sering demam, ada batuk darah, dan dada sakit. Apabila ada salah satu gejala akan diperiksa dahaknya dengan metode TCM, apabila nanti hasilnya positif akan langsung diobati. Dan apabila negatif, akan kita lihat kondisinya jika mengarah ke TB akan kita minta fotokan (Foto Rontgen).

Pertanyaan : Mengenai penelitian saya, pada penelitian saya terdapat parameter yaitu kadar oksigen dalam darah dan warna kuku. Biasanya pada pasien TB itu warna kukunya terlihat kusam, sedangkan untuk orang normal biasanya bewarna merah.

Apakah bisa kadar oksigen dalam darah dan warna kuku mengindikasikan seseorang terkena penyakit paru-paru?

Jawab : Bisa, itukan menunjukkan kadar oksigen yang ada di dalam darah, biasanya kan Hb kurang berarti oksigen yang masuk juga kurang. Jadi kukunya tidak mungkin merah, apabila Hbnya mencukupi otomatis oksigennya juga mencukupi karena oksigen dibawa oleh Hb.

Pertanyaan : Kemudian ada juga parameter suhu tubuh, apakah bisa suhu tubuh juga mempengaruhi penyakit paru-paru?

Jawab : Bisa, karena kan salah satu gejalanya sering demam.

Pertanyaan : Kemudian ada sensor tingkat respirasi, apakah tingkat respirasi juga mempengaruhi penyakit paru-paru?

Jawab : Bisa, misal diawal sakit berapa tingkat pernapasannya, kemudian setelah pengobatan berapa nilai tingkat pernapasannya, kemudian setelah sembuh juga diukur. Karena banyak setelah sembuh masih mengalami sesak napas.

**LAMPIRAN B DATA LATIH**

No	R	G	B	Sp02	Suhu	Flex	Kelas
1	183	132	123	94	36,3	40	Normal
2	183	141	143	95	36,3	40	Normal
3	190	180	172	96	36,5	43	Normal
4	190	139	134	96	36,7	38	Normal
5	192	171	181	96	36,7	40	Normal
6	194	154	154	96	36,4	37	Normal
7	195	183	179	95	36,3	41	Normal
8	205	175	173	94	36,3	40	Normal
9	206	217	212	97	36,3	50	Normal
10	207	183	183	95	36,3	38	Normal
11	207	167	162	95	36,7	56	Normal
12	210	180	185	96	36,4	63	Normal
13	210	191	172	96	36,2	40	Normal
14	216	233	220	94	36,8	35	TB
15	216	181	179	95	36,6	25	TB
16	217	204	199	94	36,2	26	TB
17	217	153	210	94	36,9	25	TB
18	217	179	178	93	36,9	29	TB
19	225	237	250	93	37,6	27	TB
20	226	195	186	94	36,9	30	TB
21	226	172	155	94	36,7	31	TB
22	227	220	236	93	37,1	33	TB
23	227	201	200	94	37,0	26	Pneu
24	228	177	173	94	37,2	23	Pneu
25	228	187	184	94	36,9	25	Pneu
26	229	196	187	92	37,3	22	Pneu
27	231	241	272	94	37,9	24	Pneu
28	240	234	230	94	37,6	24	Pneu
29	240	236	254	94	37,7	30	Pneu
30	242	188	149	92	37,0	22	Pneu



LAMPIRAN C DATA UJI

No	R	G	B	SpO2	Suhu	Flex	Kelas
1	195	160	165	94	36	60	Normal
2	199	159	159	96	36,3	37	Normal
3	200	153	148	95	36,6	42	Normal
4	200	161	164	94	36,3	40	Normal
5	201	151	150	96	36,3	33	Normal
6	202	166	168	95	36,3	56	Normal
7	204	171	157	97	36,3	33	Normal
8	221	190	188	93	36,9	31	TB
9	222	197	174	94	36,7	35	TB
10	222	192	194	95	36,6	29	TB
11	225	244	277	94	37,4	30	TB
12	233	211	194	94	36,9	27	Pneu
13	237	254	267	93	38,3	26	Pneu
14	238	186	181	93	36,8	23	Pneu



LAMPIRAN D KODE PROGRAM UTAMA

D.1 Kode Program Utama

No	Kode Program
1	#include <LiquidCrystal_I2C.h>
2	#include <CircularBuffer.h>
3	#include <MAX30100.h>
4	#include <MAX30100_BeatDetector.h>
5	#include <MAX30100_Filters.h>
6	#include <MAX30100_Registers.h>
7	#include <MAX30100_SpO2Calculator.h>
8	
9	#include <Wire.h>
10	#include <MAX30100_PulseOximeter.h>
11	
12	#define REPORTING_PERIOD_MS 1000
13	LiquidCrystal_I2C lcd = LiquidCrystal_I2C(0x27, 16, 2);
14	PulseOximeter pox;
15	uint32_t tsLastReport = 0;
16	float datasensor[6];
17	float range[30][2];
18	unsigned long timeku, start, finish, waktu;
19	int pbState = 0;
20	int pbPin = 3;
21	int state = 0;
22	int getppg = 0;
23	void onBeatDetected()
24	{
25	Serial.println("Beat!");
26	}
27	void(* resetFunc) (void) = 0;
28	void setup() {
29	Serial.begin(9600);
30	pinMode(pbPin, INPUT);
31	lcd.init();
32	lcd.backlight();
33	setupkuku();



```
34 Serial.print("Initializing pulse oximeter..");
35 if (!pox.begin()) {
36     Serial.println("FAILED");
37     for(;;);
38 } else {
39     Serial.println("SUCCESS");
40 }
41 pox.setOnBeatDetectedCallback(onBeatDetected);
42 setupsuhu();
43 }
44 void loop() {
45     start = millis();
46     pox.update();
47     if (millis() - tsLastReport > REPORTING_PERIOD_MS) {
48         Serial.print(pox.getSpO2());
49         Serial.println("%");
50         getppg = pox.getSpO2();
51         tsLastReport = millis();
52     }
53     lcd.setCursor(0,0);
54     lcd.print("Tekan Tombol");
55     if(digitalRead(pbPin)==HIGH){
56         Serial.println("Tombol ditekan");
57         pbState++;
58         delay(1000);
59         if(pbState == 4) timeku = millis();
60     }
61     if(pbState == 1){
62         lcd.setCursor(0,1);
63         lcd.print(String(getppg));
64         datasensor[3] = getppg;
65         Serial.println(getppg);
66         Serial.println();
67     }
68     if(pbState == 2){
69         Serial.println("Penekanan Kedua");
70         Serial.println("Pengukuran kuku");
71         loopkuku();
```




```
72     delay(1000);
73     }
74     if(pbState == 3){
75         Serial.println("Penekanan Ketiga");
76         Serial.println("Pengukuran suhu");
77         loopsuhu();
78         delay(1000);
79     }
80     if(pbState == 4){
81         //lcd.setCursor(0,0);
82         int tes = 1;
83         Serial.println("Penekanan Keempat");
84         if(tes == 1){
85             Serial.println(millis()-timeku);
86             Serial.println("Pengukuran Flex");
87             loopflex();
88             delay(100);
89             tes++;
90         } else {
91             Serial.print("R = ");
92             Serial.println(datasensor[0]);
93             Serial.print("G = ");
94             Serial.println(datasensor[1]);
95             Serial.print("B = ");
96             Serial.println(datasensor[2]);
97             Serial.print("Spo2 = ");
98             Serial.println(datasensor[3]);
99             Serial.print("Suhu = ");
100            Serial.println(datasensor[4]);
101            Serial.print("Flex = ");
102            Serial.println(datasensor[5]);}
103     }
104     if(pbState == 5){
105         //Perhitungan KNN
106         int DataJarak = 0;
107         int label = 1;
108         //Perhitungan Eclidean
109         for(int a=0;a<30;a++){
```




```

110     range[a][0] = 0; //reset data
111     for(int b=0;b<6;b++){
112         range[a][0] += pow( DataLatih (a, b) -
113         datasensor[b], 2);
114     }
115     range[a][0] = sqrt(range[a][0]); //akar
116     switch(a){
117         case 0... 12:
118             range[a][1] = 0; //angka 0 untuk label normal
119             break;
120         case 13... 21:
121             range[a][1] = 1; //angka 1 untuk label TB
122             break;
123         case 22... 29:
124             range[a][1] = 2; //angka 2 untuk label Pneumonia
125             break;
126     }
127     //Serial.print("Hasil Ecludian = ");
128     //Serial.println(range[a][0]);
129 }
130 //SORTING DATA
131 for(int i=0;i<29;i++){
132     for(int j=i+1;j<30;j++){
133         if(range[i][DataJarak]>range[j][DataJarak]){
134             float temp = range[i][DataJarak];
135             range[i][DataJarak] = range[j][DataJarak];
136             range[j][DataJarak] = temp;
137         }
138         temp = range[i][label];
139         range[i][label] = range[j][label];
140         range[j][label] = temp;
141     }
142 }
143 }
144 float nol = 0, satu = 0, dua = 0;
145 String hasil;
146 for(int a=0;a<3;a++){
147     if(range[a][label] == 0.00){
148         nol += 1/ pow(range[a][DataJarak], 2);

```




```

148     }
149     else if(range[a][label] == 1.00){
150         satu += 1/ pow(range[a][DataJarak],2);
151     }
152     else if(range[a][label] == 2.00){
153         dua += 1/ pow(range[a][DataJarak],2);
154     }
155 }
156 if(nol > satu && nol > dua){
157     hasil = "NORMAL";
158 }
159 else if(satu > nol && satu > dua){
160     hasil = "TB";
161 }
162 else if(dua > satu && dua > nol){
163     hasil = "PNEUMONIA";
164 }
165 finish = millis();
166 waktu = (finish - start);
167 lcd.clear();
168 lcd.setCursor(0,0);
169 lcd.print(hasil);
170 Serial.print("Hasil Klasifikasi : ");
171 Serial.println(hasil);
172 Serial.print("Waktu Komputasi = ");
173 Serial.println(waktu);
174 delay(15000);
175 resetFunc();
176 }
}

```

D.2 Kode Program Sensor Suhu MLX90614

No	Kode Program
1	#include <Wire.h>
2	#include <Adafruit_MLX90614.h>
3	Adafruit_MLX90614 mlx = Adafruit_MLX90614();
4	void setupsuhu() {
5	mlx.begin();



```

6     }
7     void loopsuhu() {
8         float getsuhu = mlx.readObjectTempC();
9         datasensor[4] = getsuhu;
10        delay(500);
11    }

```

D.3 Kode Program Sensor Warna TCS 3200

No	Kode Program
1	#define S0 4
2	#define S1 5
3	#define S2 6
4	#define S3 7
5	#define sensorOut 8
6	int redfrequency = 0;
7	int greenfrequency = 0;
8	int bluefrequency = 0;
9	int redcolor = 0;
10	int greencolor = 0;
11	int bluecolor = 0;
12	void setupkuku() {
13	pinMode(S0, OUTPUT);
14	pinMode(S1, OUTPUT);
15	pinMode(S2, OUTPUT);
16	pinMode(S3, OUTPUT);
17	pinMode(sensorOut, INPUT);
18	digitalWrite(S0, HIGH);
19	digitalWrite(S1, LOW);
20	}
21	void loopkuku() {
22	// Setting red filtered photodiodes to be read
23	digitalWrite(S2, LOW);
24	digitalWrite(S3, LOW);
25	// Reading the output frequency
26	redfrequency = pulseIn(sensorOut, LOW);
27	redcolor = map(redfrequency, 52, 98, 255, 0); //34, 167//49, 63
28	if (redcolor > 255) {
29	redcolor = 255;



```
30 } else if (redcolor <= 0) {
31   redcolor = 0;
32 }
33 // Printing the value on the serial monitor
34 Serial.print("R= "); //printing name
35 Serial.print(redcolor); //printing RED color frequency
36 Serial.print(" ");
37 datasensor[0] = redcolor;
38 delay(100);
39 // Setting Green filtered photodiodes to be read
40 digitalWrite(S2,HIGH);
41 digitalWrite(S3,HIGH);
42 // Reading the output frequency
43 greenfrequency = pulseIn(sensorOut, LOW);
44 greencolor=map(greenfrequency, 71, 131, 255, 0);
45 if (greencolor > 255) {
46   greencolor = 255;
47 } else if (greencolor <= 0) {
48   greencolor = 0;
49 }
50 // Printing the value on the serial monitor
51 Serial.print("G= "); //printing name
52 Serial.print(greencolor); //printing RED color frequency
53 Serial.print(" ");
54 datasensor[1] = greencolor;
55 delay(100);
56 // Setting Blue filtered photodiodes to be read
57 digitalWrite(S2,LOW);
58 digitalWrite(S3,HIGH);
59 // Reading the output frequency
60 bluefrequency = pulseIn(sensorOut, LOW);
61 bluecolor=map(bluefrequency, 60, 110, 255, 0); //45, 172//59, 91
62 if (bluecolor > 255) {
63   bluecolor = 255;
64 } else if (bluecolor <= 0) {
65   bluecolor = 0;
66 }
67 // Printing the value on the serial monitor
```




```

68 Serial.print("B= "); //printing name
69 Serial.print(bluecolor); //printing RED color frequency
70 Serial.println(" ");
71 datasensor[2] = bluecolor;
72 delay(3000);
73 if(redcolor > greencolor && redcolor > bluecolor){
74 Serial.println(" - RED detected!");
75 }
76 if(greencolor > redcolor && greencolor > bluecolor){
77 Serial.println(" - GREEN detected!");
78 }
79 if(bluecolor > redcolor && bluecolor > greencolor){
80 Serial.println(" - BLUE detected!");
81 }
82 }

```

D.4 Kode Program Sensor Flex

No	Kode Program
1	int flexPin = A7;
2	int i, value, low, high, hasil, temp, flex[10];
3	void loopflex(){
4	for (i = 0; i < 10; i++){
5	value = analogRead(flexPin);
6	flex[i] = value;
7	delay(500);
8	}
9	high = flex[0];
10	low = flex[0];
11	for (i = 0; i < 10; i++){
12	if (high < flex[i]){
13	high = flex[i];
14	}
15	else if (low > flex[i]){
16	low = flex[i];
17	}
18	}
19	hasil = high - low;

