

Kendali Aliran dan Tekanan Adaptif dengan Metode *Artificial Neural Network* pada Alat Terapi Oksigen

ABYANUDDIN SALAM, NUR WISMA NUGRAHA, WILDAN ALFARIDHANI

Jurusan Teknik Otomasi Manufaktur dan Mekatronika, Politeknik Manufaktur Bandung, Indonesia
Email: aby@ae.polman-bandung.ac.id

Received 22 Agustus 2023 | *Revised* 5 Oktober 2023 | *Accepted* 22 Oktober 2023

ABSTRAK

Penelitian ini bertujuan untuk merancang prototype pengendalian aliran dan tekanan adaptif pada alat terapi oksigen. Sensor yang digunakan yaitu sensor MAX30100 untuk membaca saturasi oksigen dan sensor MLX90614 sebagai sensor yang dapat menghitung Respiration Rate atau laju napas. Metode yang digunakan yaitu Artificial Neural Network yang diimplementasikan pada Raspberry Pi. Sistem akan bekerja dengan memprediksi nilai laju aliran dan tekanan oksigen yang diperlukan pasien berdasarkan nilai Respiration Rate (RR). Artificial Neural Network (ANN) dapat diimplmentasikan pada rancangan alat terapi oksigen, dengan persentase akurasi Output ANN terhadap perhitungan yaitu 99,39%, sedangkan persentase akurasi ANN terhadap pembacaan aliran oksigen yang terbaca pada sensor flow sebesar yaitu 94,73% dan persentase akurasi ANN terhadap pembacaan tekanan oksigen pada sensor pressure sebesar 89,03%.

Kata kunci: Terapi Oksigen, Respiration Rate, Artificial Neural Network

ABSTRACT

This research aims to design a prototype of flow and pressure control in an adaptive oxygen therapy device. The sensors used are MAX30100 sensors to read oxygen saturation and MLX90614 sensors as sensors that can calculate Respiration Rate or breath rate. The method used is Artificial Neural Network which is implemented on Raspberry Pi. The system will work by predicting the value of the flow rate and oxygen pressure required by the patient based on the Respiration Rate (RR) value. Artificial Neural Network (ANN) can be implemented in the design of oxygen therapy devices, with the percentage of ANN Output accuracy to the calculation of 99.39%, while the percentage of ANN accuracy on oxygen flow readings on the flow sensor is 94.73% and the percentage of ANN accuracy on oxygen pressure readings on the pressure sensor is 89.03%.

Keywords: Oxygen Therapy, Respiration Rate, Artificial Neural Network

1. PENDAHULUAN

Oksigen (O₂) adalah salah satu unsur gas yang sangat penting dalam proses metabolisme **(Purba & Harefa, 2020)**. Kekurangan oksigen berdampak pada respon tubuh mudah lelah, letih dan mudah mengantuk **(Nugroho, dkk, 2020)**. Secara medis, kekurangan oksigen atau hipoksia ditangani dengan meningkatkan konsumsi oksigen (FiO₂), dan salah satu cara penanganannya adalah dengan terapi oksigen **(Sahid, dkk, 2020) (Irawati, dkk, 2022)**. Dalam pemberian terapi oksigen, perlu dilakukan pengawasan baik pasien maupun alat terapi oksigen yang digunakan **(Kane, dkk, 2013)**. Sebab kelebihan atau overdosis oksigen dapat menyebabkan keracunan oksigen dan dapat menimbulkan kerusakan sel hingga kematian **(Anekawati, dkk, 2021)**. Salah satu upaya dalam pengawasan terapi oksigen yaitu melakukan pengontrolan terhadap aliran dan tekanan oksigen yang diberikan kepada pasien **(Khairunnisa, dkk, 2018)**. Oleh karena itu maka diperlukan sebuah sistem pengontrolan aliran dan tekanan oksigen yang secara adaptif mampu mengontrol aliran dan tekanan oksigen sesuai kondisi *real time* pasien.

Beberapa penelitian sebelumnya telah merancang sistem ventilator IoT pada pengendalian tekanan oksigen dengan hasil penelitian tekanan rata-rata sebesar 0,98 Pa dan rata-rata *delay time* komunikasi *phone to device* (Android ke ventilator) sebesar 1,69 detik **(Salam, dkk, 2021)**. Kemudian metode PID diterapkan dalam pengendalian tekanan sistem *Continuous Positive Airway Pressure (CPAP) ventilation*, dan diperoleh respons sistem yang cukup stabil serta memiliki *error* keluaran yang cukup kecil terhadap sistem *CPAP* komersil **(Dai, dkk, 2013)**. Metode *fuzzy logic* diterapkan pada kendali aliran oksigen dengan hasil yang lebih baik, nilai rata-rata *error* kurang dari 2% **(Ardhianto, n.d.)**. Kemudian dengan metode yang sama *fuzzy logic* pada kendali tekanan oksigen menghasilkan akurasi rata-rata sebesar 98,77% **(Nugraha, n.d.)**. Sistem lain dari ventilator yang dirancang menggunakan Arduino, Raspberry Pi 4, dan *Human Machine Interface (HMI)* hal ini membuat sistem lebih cerdas dan dapat digunakan di rumah tanpa perlu bantuan profesional **(Malaekah, dkk, 2022)**.

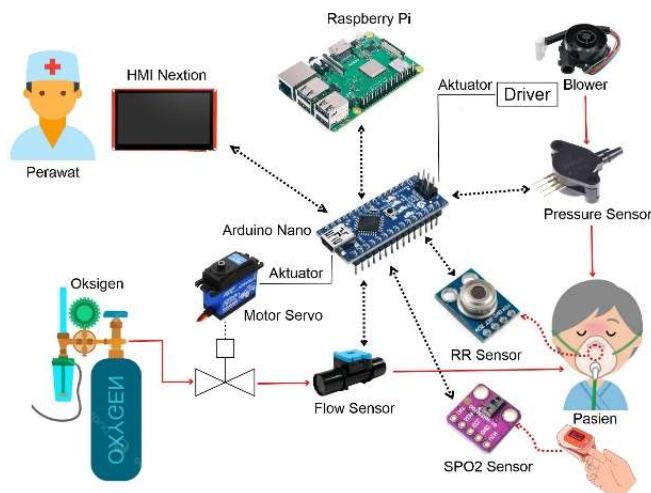
Berdasarkan hasil uraian penelitian sebelumnya, rancangan sistem ventilator kebanyakan diketahui menggunakan metode PID dan *fuzzy logic* **(Kesuma, dkk, 2020)**. Sedikit penelitian yang berfokus pada penerapan *machine learning* **(Aryanto & Augustman, 2021)**. Padahal pengendalian yang bersifat adaptif hanya dapat dimungkinkan melalui permodelan sistem berbasis kecerdasan buatan **(Sita, dkk, 2022)**. Oleh karena itu pada penelitian ini memiliki perbedaan metode dari yang sebelumnya yaitu menggunakan metode *Artificial Neural Network* **(Timorremboko & Karya, 2020)**. Dengan metode tersebut sistem diharapkan dapat melakukan prediksi keluaran aliran dan tekanan oksigen berdasarkan hasil *training dataset*, serta sistem mampu secara adaptif membaca konsisi *realtime* pasien sebagai *input* dalam proses pengendalian aliran dan tekanan oksigen yang dibutuhkan pasien

2. METODOLOGI PENELITIAN

2.1 Gambaran Umum Sistem

Gambaran umum sistem pada rancangan alat terapi oksigen ini yang terbagi menjadi empat bagian utama, yaitu *input*, pengendalian aliran dan tekanan oksigen dengan *Artificial Neural Network*, aktuator dan antarmuka. *Input* terdiri dari dua buah sensor yaitu sensor MAX30100 untuk mendeteksi saturasi oksigen pasien dan sensor suhu MLX90614 untuk mendeteksi perubahan suhu di dalam masker oksigen ketika bernapas lalu perubahan suhu tersebut dikonversi menjadi nilai *Respiration Rate (RR)* atau laju napas. Sensor *Flow* dan Sensor *Pressure* digunakan untuk pengukuran nilai aliran dan tekanan oksigen yang diberikan kepada pasien.

Gambaran umum sistem ditunjukkan pada Gambar 1.

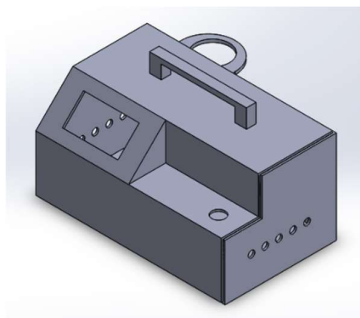


Gambar 1. Gambaran Umum Sistem

Akuisisi data pada sensor diakses oleh Arduino Nano, sedangkan pengolahan data dengan metode *Artificial Neural Network* diproses oleh Raspberry Pi setelah menerima data *Respiration Rate* (RR) dari Arduino melalui komunikasi serial. *Output* yang diperoleh dari metode *Artificial Neural Network* (ANN) yaitu prediksi nilai *Minute Ventilation* (MV) atau nilai aliran oksigen yang diperlukan pasien dan nilai *Pressure Ventilation* (PV) sebagai indikator perlu tidaknya pasien mendapatkan *pressure* udara tambahan dari blower. Hasil *output* tersebut kemudian akan dikirim ke Arduino Nano dan dikonversi menjadi derajat sudut servo untuk mengatur bukaan katup aliran oksigen dan blower sebagai pemberi tekanan udara tambahan. Antarmuka yang digunakan yaitu *Nextion HMI* untuk mengatur mode operasi sistem yang dibutuhkan serta *monitoring* sistem.

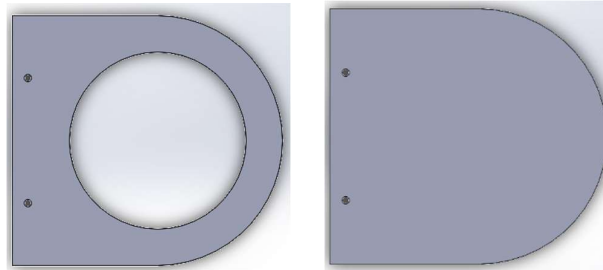
2.2 Perancangan Mekanik

Perancangan mekanik mencakup *box cover* dan elemen mekanik aktuator yang bergerak sebagai perlakuan terhadap *input* yang diberikan. Perancangan *box cover* terdiri dari tiga bagian, bagian utama yaitu *box* atau *cover* yang di dalamnya berisi komponen elektrik. Bagian berikutnya yaitu dudukan dan pencekam untuk komponen *heater*, dudukan ini diposisikan di belakang *cover*. Bagian terakhir yaitu dua buah silinder yang digunakan sebagai pencekam dari regulator oksigen. Masing-masing bagian tersebut ditunjukkan secara berurutan pada Gambar 2., Gambar 3. dan Gambar 4.

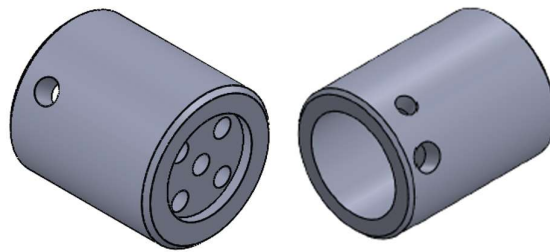


Gambar 2. Box Cover

Salam, dkk



Gambar 3. Dudukan & Pencekam Heater

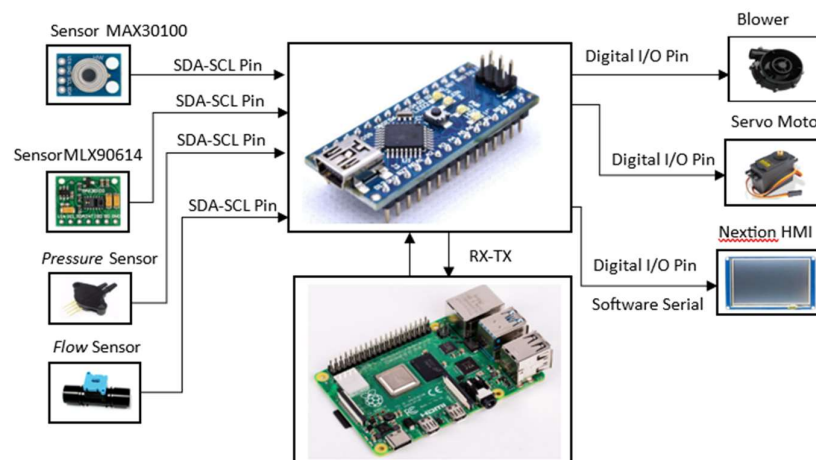


Gambar 4. Pencekam Regulator Tabung

Salain *box cover* pada rancangan mekanik, terdapat aktuator yang digunakan sebagai *output* dari proses sistem. Aktuator yang digunakan yaitu motor servo yang digunakan sebagai bukaan katup aliran pada tabung oksigen dan Blower yang digunakan sebagai pemberi *pressure* udara tambahan.

2.3 Perancangan Elektrik

Perancangan elektrik meliputi seluruh komponen elektrik yang digunakan pada sistem. Rancangan ini meliputi rangkaian akuisisi data, pengolahan data oleh *Artificial Neural Network* (ANN), aktuator dan antarmuka. Pada rangkaian akuisisi data terdiri dari papan PCB yang di dalamnya terdapat Arduino untuk mengakses sensor dan menggerakkan aktuator, kemudian pin SDA-SCL Arduino disambungkan dengan sensor utama yang terdiri dari sensor saturasi oksigen, sensor suhu, sensor *flow* dan sensor *pressure*, sedangkan untuk aktuator menggunakan pin PWM digital. Perancangan elektrik ditunjukkan pada Gambar 5.

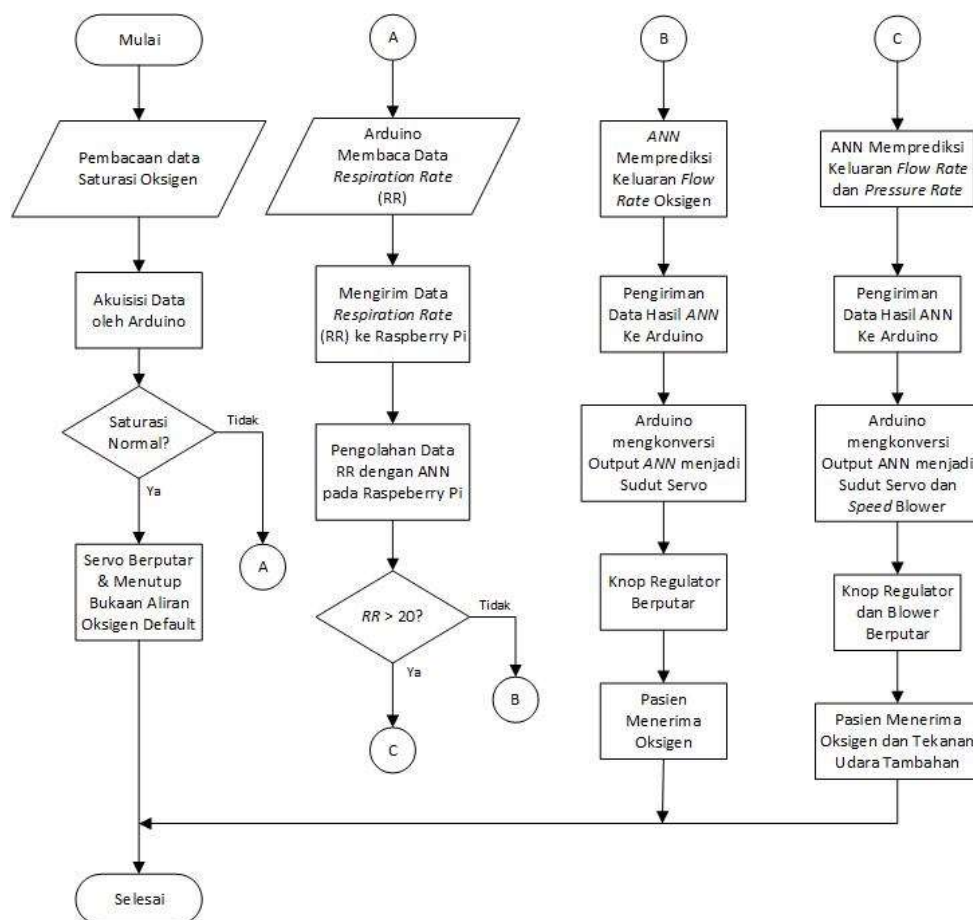


Gambar 5. Perancangan Elektrik

Raspberry Pi dikomunikasikan dengan Arduino secara serial menggunakan kabel microUSB. Kemudian antarmuka menggunakan Nextion HMI yang dikomunikasikan dengan Arduino secara *software* serial menggunakan pin digital Arduino.

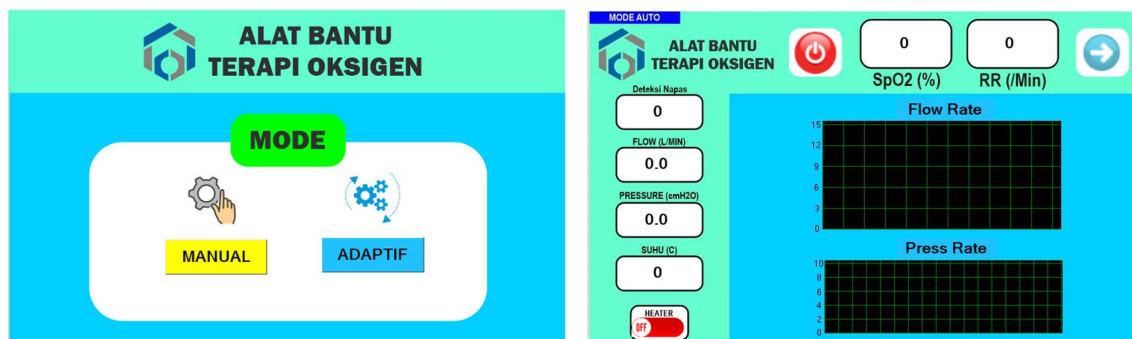
2.4 Perancangan Perangkat Lunak

Domain perancangan perangkat lunak meliputi perancangan program akuisisi data, program kendali adaptif dengan *Artificial Neural Network* (ANN) dan antarmuka. Untuk perancangan perangkat lunak diawali dengan membuat diagram alir sistem yang ditunjukkan pada Gambar 6. Pada diagram alir tersebut sistem diawali dengan proses pembacaan saturasi oksigen, jika saturasi normal maka Arduino akan men-*trigger* servo dan menutup katup bukaan aliran oksigen *default* dan jika saturasi oksigen tidak normal, maka Arduino akan membaca nilai laju napas atau *Respiration Rate* (RR) setiap 1 menit.



Gambar 6. Diagram Alir Sistem

Program kendali adaptif dengan *Artificial Neural Network* (ANN) diawali ketika Arduino mengirimkan data *Respiration Rate* (RR) ke Raspberry Pi sebagai *input*-an dalam proses pengendalian. *Output* yang diperoleh dari pengendalian dengan ANN ini yaitu hasil prediksi nilai *Minute Ventilation* (MV) dan *Pressure Ventilation* (PV) yang dibutuhkan pasien. Kemudian nilai tersebut akan dikirim ke Arduino dan dikonversi sebagai derajat servo untuk membuka katup aliran oksigen dan nilai kecepatan blower untuk memberikan tekanan udara tambahan jika nilai RR pada pasien lebih dari 20 kali napas/menit.



Gambar 7. Rancangan Antarmuka HMI

Perancangan antarmuka HMI dibuat menggunakan *software* Nextion HMI, sedangkan pemrograman komunikasi Arduino dengan HMI dilakukan pada program Arduino yang digabungkan dengan program akuisisi data sensor dan aktuator. Antarmuka HMI ini menyediakan dua buah opsi kontrol, yaitu manual dan adaptif. Di dalam masing-masing opsi terdapat tombol *power* dan beberapa tombol pengaturan *input* kontrol sistem. Setiap perubahan data yang terbaca akan ditampilkan secara *realtime* baik berupa angka ataupun grafik. Perancangan antarmuka HMI ditunjukkan pada Gambar 7.

2.5 Perancangan Sistem *Artificial Neural Network* (ANN)

Perancangan sistem *Artificial Neural Network* (ANN) pada kendali aliran dan tekanan adaptif terbagi menjadi tiga tahap, yaitu pengumpulan data, *training model* dan *testing model*. Pengumpulan data diperlukan sebagai data latih dalam proses *training model*, kemudian dilanjutkan dengan tahap *training model* dan *testing model* dengan mengubah beberapa variabel, seperti jumlah *hidden layer* dan fungsi aktivasi, sampai diperoleh sebuah sistem *Artificial Neural Network* (ANN) dengan *output* sesuai yang diharapkan dan nilai *error* yang kecil.

Tahap pertama yaitu pengumpulan *dataset*. Data yang diperlukan merupakan data nilai *Respiration Rate* (RR) pasien. Data tersebut diperoleh dari platform *Kaggle.com*. Dari data RR tersebut dilakukan perhitungan untuk mengetahui nilai aliran oksigen atau *Minute Ventilation* (MV) dan tekanan udara tambahan atau *Pressure Ventilation* (PV) yang diperlukan pasien. Hubungan antara nilai RR terhadap nilai *Minute Ventilation* (MV) diperoleh berdasarkan Persamaan (1) berikut:

$$MV = \frac{Tidal\ Volume \times Respiration\ Rate\ (RR)}{1000\ ml/L} \quad (1)$$

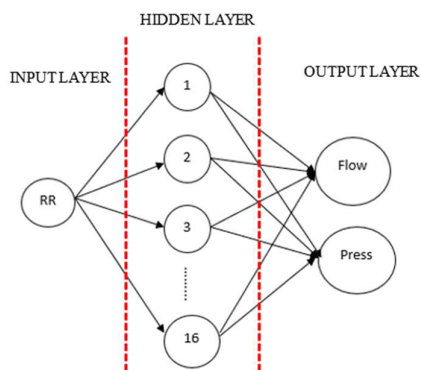
Tidal volume merupakan volume udara yang diinspirasi atau diekspirasi setiap kali melakukan pernapasan normal, besarnya 500 mililiter pada orang dewasa. Sedangkan hubungan nilai RR terhadap nilai *Pressure Ventilation* (PV) diperoleh dengan Persamaan (2) berikut:

$$P_{mus}/PV = R_{RS} \cdot \frac{MV}{60} + E_{RS} \cdot Vt \quad (2)$$

R_{RS} atau resistansi normal paru-paru yaitu 4 cmH₂O/L/sec, E_{RS} atau elastansi normal paru-paru yaitu 10 cmH₂O/L, dan Vt atau volume tidal normal dewasa yaitu 500 ml atau 0,5 L.

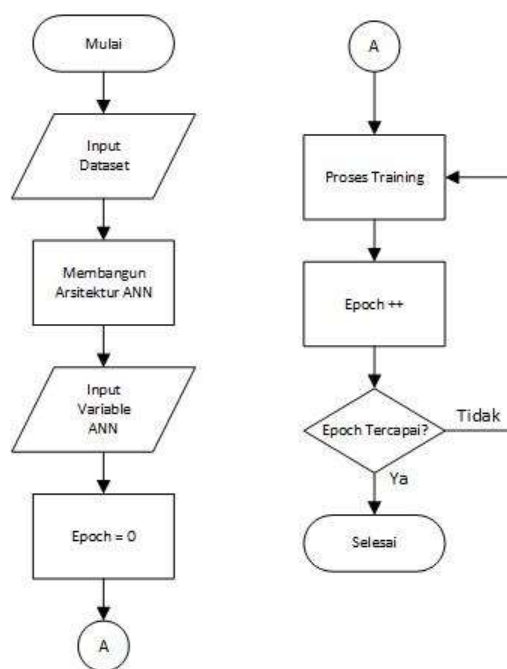
Tahap berikutnya yaitu *Training Model*, data yang diperoleh dibagi menjadi dua, yaitu *dataset training* dan *dataset testing*. Proses *training dataset* dilakukan dengan program *python* pada Raspberry Pi. Pada proses pemrograman *training model* digunakan beberapa *library* untuk

mempermudah proses *training* seperti *tensorflow*, *keras*, *numpy* dll. Sistem yang dilatih menggunakan algoritma *Optimaizer ADAM*, dan dilakukan *trial & error* pada beberapa variabel, seperti jumlah *Hidden Layer*, *learning rate* dan fungsi aktivasi, sampai diperoleh arsitektur *Artificial Neural Networ* (ANN) dengan akurasi yang baik dan nilai *error* yang kecil. Arsitektur ANN ditunjukkan pada Gambar 8.



Gambar 8. Arsitektur *Artificial Neural Network* (ANN)

Arsitektur model ANN yang dibangun terdiri dari satu *input layer*, satu *hidden layer* dan satu *output layer*. *Input layer* merupakan nilai *Respiration Rate* (RR), kemudian *output layer* terdiri dari dua *node* yaitu nilai *Flow Rate* atau *Minute Ventilation* (MV) dan *Press Rate* atau *Pressure Ventilation* (PV), sedangkan *hidden layer* terdapat enam belas *node*. Setelah arsitektur dan variabel ANN ditentukan maka *training model* dilakukan, proses ini akan terus berulang-ulang sampai batas maksimum *epoch* terpenuhi. *Epoch* merupakan kondisi ketika seluruh *dataset* sudah melalui proses *training* pada *Neural Network* sampai dikembalikan ke awal untuk sekali putaran sampai *epoch* tersebut terpenuhi. Diagram alir *training model* ditunjukkan pada Gambar 9.

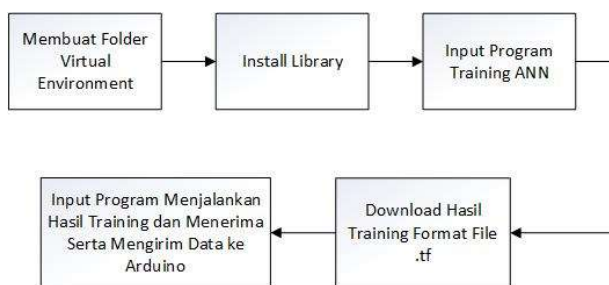


Gambar 9. Diagram Alir *Training Model ANN*

Tahap terakhir yaitu *testing model*, program testing model merupakan program yang berfungsi untuk memprediksi *Output* berdasarkan *dataset* yang telah di masukan. Pada *dataset* yang digunakan dibagi menjadi 80% sebagai data *training* dan 20% sebagai data *testing*. Dengan data testing tersebut dapat diketahui nilai error atau selisih antara nilai hasil prediksi dengan nilai real data *testing*. Kemudian *error* setiap data akan dikalkulasikan rata-ratanya dengan metode *Mean Absolute Error (MAE)*.

2.6 Implementasi *Artificial Neural Network (ANN)* pada *Raspberry Pi*

Implementasi ANN pada *Raspberry Pi* dibuat menggunakan bahasa pemrograman *python* dengan beberapa tahapan yang perlu dilakukan, kemudian penggunaan *library* yang dibutuhkan sangat berpengaruh terhadap keberhasilan implementasi ANN pada *Raspberry Pi*. Diagram alir untuk proses implementasi ANN pada *Raspberry Pi* ditunjukkan pada Gambar 10.



Gambar 10. Diagram Alir Penerapan ANN pada *Raspberry Pi*

Folder Virtual Environment merupakan folder yang digunakan untuk menyimpan semua *library* yang digunakan untuk menjalankan program *training* ANN ataupun merekayasa tipe versi *python* yang digunakan. Beberapa *library* yang dibutuhkan dan harus ter-*install* di dalam virtual *environment* tersebut yaitu *Numpy*, *Pandas*, *Tensorflow*, *Keras* dan *Sklearn*.

Berikutnya buat program model *training* ANN menggunakan bahasa pemrograman *python*, program *training* ANN ini dibuat dengan memasukan beberapa parameter seperti arsitektur ANN, fungsi aktivasi dan jumlah *epoch training* seperti yang sudah dijelaskan pada sub bab sebelumnya. Program model *training* ANN yang diterapkan pada sistem *Raspberry Pi* ditunjukkan pada Gambar 11.

```

model = tf.keras.models.Sequential()
model.add(tf.keras.layers.Dense(units=64, activation='sigmoid', input_shape=(1,)))
model.add(tf.keras.layers.Dense(units=2))
model.compile(loss='mae', optimizer=tf.keras.optimizers.Adam(0.01))
model.fit(X_train, Y_train, epochs=3000, verbose=0, validation_split=0.2 )
  
```

Gambar 11. Program Model *Training* ANN

Kemudian model *training* ANN di-*download* ke dalam sebuah *file* dengan fomart *.tf* supaya dapat dijalankan dan dapat melakukan prediksi terhadap nilai aliran dan tekanan oksigen jika diberikan nilai *input Respiration Rate (RR)* dari *Arduino*. Program tersebut dapat ditambahkan pada bagian akhir program *training* ANN. Program untuk men-*download* model *training* ANN ditunjukkan pada Gambar 12.

```

model.save('model_ann.tf')
  
```

Gambar 12. Program Menyimpan Model *Training* ANN

Model *Training* ANN dengan format file *.tf* dapat dijalankan dengan perintah *load* model atau memanggil model *training* pada program *python* yang berbeda, selain itu program *python* ini dibuat dengan menambahkan perintah untuk menerima data *RR* dari *Arduino* dan mengirim

data nilai aliran dan tekanan oksigen hasil prediksi ke Arduino secara serial. Program *python* perintah untuk memanggil file hasil *training* ANN dengan format *tf*. ditunjukkan pada Gambar 13.

```
# Load the model
model = tf.keras.models.load_model('model_ann.tf')
```

Gambar 13. Program Memanggil Model *Training* ANN

3. HASIL DAN PEMBAHASAN

3.1 Hasil Aktualisasi Perancangan

Hasil aktualisasi dari rancangan alat terapi oksigen ditunjukkan pada Gambar 14 dan Gambar 15. Komponen yang berada di dalam *box cover* terdiri dari semua komponen elektrik. Sedangkan pada luar *cover box*, terdapat sensor MAX30100 untuk mendeteksi saturasi oksigen, sensor MLX90614 sebagai pendeteksi napas, *heater* untuk menyesuaikan suhu oksigen yang diberikan kepada pasien serta aktuator motor servo sebagai bukaan katup aliran pada tabung oksigen.



Gambar 14. Aktualisasi Alat Tampak Depan & Tampak Belakang



Gambar 15. Aktualisasi Elektrik & Aktuator Servo

Sistem yang telah diimplementasikan sudah dintegrasikan antar komponen, sehingga dapat dilakukan pengujian dan analisis, sehingga presentasi ketercapaian sistem dapat dibandingkan terhadap target sistem yang diinginkan.

3.4 Pengujian Nilai *Respiration Rate* (RR)

Pengujian nilai RR dilakukan dengan membandingkan nilai perhitungan RR secara manual terhadap perhitungan RR yang terbaca pada alat. Data hasil pengujian nilai *Respiration Rate* (RR) ditunjukkan pada Tabel 1.

Tabel 1. Pengujian Nilai *Respiration Rate* (RR)

Uji ke-	Perhitungan Manual [/Menit]	Perhitungan Pada Alat [/Menit]	Selisih	Error [%]
1	24	24	0	0,00
2	27	25	2	7,41
3	28	26	2	7,14
4	25	27	-2	8,00
5	25	25	0	0,00
6	25	25	0	0,00
7	27	27	0	0,00
8	25	25	0	0,00
9	25	23	2	8,00
10	28	27	1	3,57
Persentase Error				3,41%
Akurasi				96,59%

Hasil perhitungan nilai RR pada alat memiliki rata-rata *error* sebesar 3.41% yang dibuktikan dengan menghitung rata-rata persentase *error* dengan persamaan *Mean Absolute Percentage Error* (MAPE). Kemudian berdasarkan nilai presentasi *error* dilakukan perhitungan untuk mengetahui nilai akurasi, sehingga diperoleh nilai akurasi sensor sebesar 96,59%.

3.3 Pengujian *Training Artificial Neural Network* (ANN)

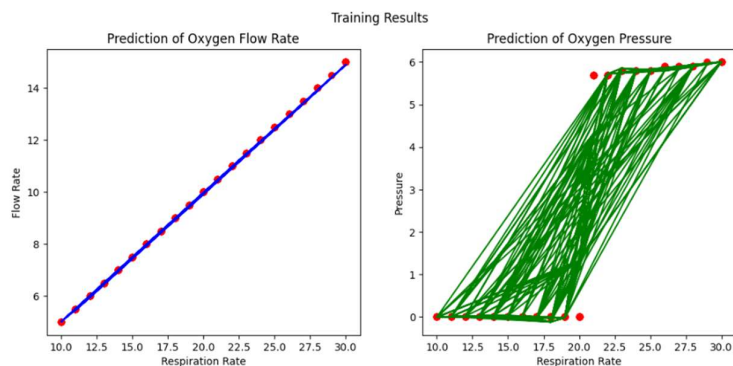
Pengujian ini dilakukan pada program *training* di Raspberry Pi dengan mengubah parameter yang dapat berpengaruh terhadap akurasi hasil *training* seperti fungsi aktivasi, *hidden layer* dan banyaknya *epoch*. Hasil pengujian *training* ANN pada Raspberry Pi ditunjukkan pada Tabel 2.

Tabel 2. Pengujian Training ANN

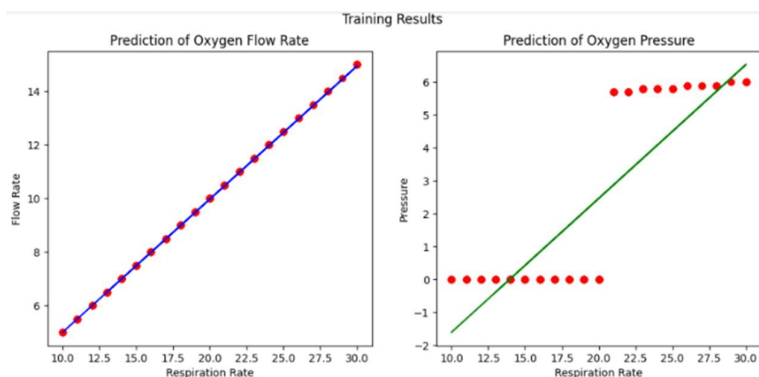
Fungsi Aktivasi	Epoch	Hidden Layer	Loss Error [%]
Sigmoid	2000	8	0,177
		16	0,18
	3000	8	0,205
		16	0,142
Linear	2000	8	0,713
		16	0,701
	3000	8	0,728
		16	0,711
Relu	2000	8	0,286
		16	0,471
	3000	8	0,468
		16	0,464

Berdasarkan hasil *training* diketahui jika fungsi aktivasi, jumlah *hidden layer* dan banyaknya *epoch* berpengaruh terhadap hasil *error training*. Nilai *error* terkecil diperoleh dengan fungsi aktivasi *sigmoid* dengan *hidden layer* 16 dan jumlah *epoch* 3000, sehingga arsitektur ANN sistem yang dibangun menggunakan fungsi aktivasi *sigmoid* dan *hidden layer* 16.

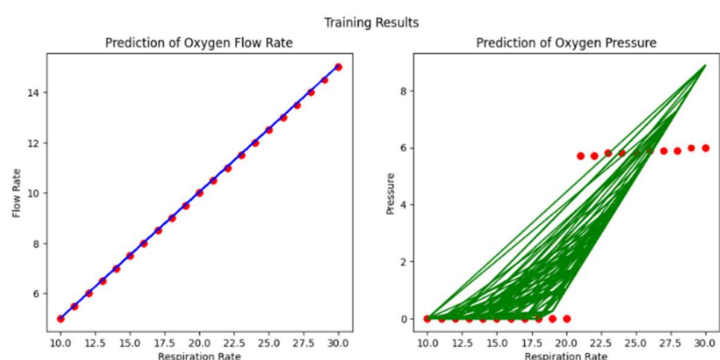
Karakteristik grafik dari masing-masing hasil pengujian *training* ANN ditunjukkan berturut-turut pada Gambar 16, Gambar 17 dan Gambar 18.



Gambar 16. Grafik Fungsi Aktivasi *Sigmoid*



Gambar 17. Grafik Fungsi Aktivasi Linear



Gambar 18. Grafik Fungsi Aktivasi Relu

Berdasarkan gambar di atas, terlihat jika pada gambar fungsi aktivasi *sigmoid* memiliki akurasi yang cukup baik, dimana *output* hasil prediksi mendekati target data *training* yang diberikan.

3.4 Pengujian *Output Artificial Neural Network (ANN)*

Pengujian *output* ANN dilakukan dengan membandingkan *output* ANN dengan nilai hasil menurut perhitungan. Diketahui jika *input* dari arsitektur ANN yang dibangun yaitu nilai RR dengan rentang nilai RR 10-30 kali napas/menit, sedangkan *output* prediksi ANN yaitu nilai *flow rate* dan *pressure rate*. Untuk mengetahui nilai *flow rate* dengan perhitungan diperoleh melalui Persamaan (1) sedangkan untuk mengetahui nilai *pressure rate* secara perhitungan diperoleh melalui Persamaan (2).

Sehingga dari hasil perhitungan tersebut dapat dilakukan perbandingan persentase *error* dan nilai akurasi *output* ANN terhadap hasil perhitungan manual. Dari data di atas, diperoleh rata-rata nilai persentase *error flow rate* sebesar 0,61% dengan akurasi sebesar 99,39%, dan rata-rata nilai persentase *error pressure rate* sebesar 6,11% dengan akurasi sebesar 93,89%. Dapat disimpulkan jika *Artificial Neural Network (ANN)* yang dibangun memiliki hasil *output* yang baik. Hasil pengujian *output* Artificial Neural Network (ANN) ditunjukkan pada Tabel 3.

Tabel 3. Perbandingan *Output* ANN Terhadap Perhitungan

RR [/Min]	<i>Output</i> Perhitungan		<i>Output</i> Prediksi ANN		Error Flow [%]	Error Pressure [%]
	Flow Rate [L/Min]	Pressure Rate [cmH2O]	Prediksi Flow [L/Min]	Prediksi Pressure [cmH2O]		
10	5	0,00	4,92	0	1,60	0,00
11	5,5	0,00	5,46	0	0,73	0,00
12	6	0,00	5,96	0	0,67	0,00
13	6,5	0,00	6,46	0	0,62	0,00
14	7	0,00	6,97	0	0,43	0,00
15	7,5	0,00	7,47	0	0,40	0,00
16	8	0,00	7,96	0	0,50	0,00
17	8,5	0,00	8,45	0	0,59	0,00
18	9	0,00	8,95	0	0,56	0,00
19	9,5	0,00	9,45	0	0,53	0,00
20	10	0,00	9,94	0	0,60	0,00
21	10,5	5,70	10,43	3,64	0,67	36,14
22	11	5,73	10,93	5,16	0,64	10,00
23	11,5	5,77	11,43	5,79	0,61	0,40
24	12	5,80	11,93	5,91	0,58	1,90
25	12,5	5,83	12,43	5,85	0,56	0,29
26	13	5,87	12,93	5,77	0,54	1,65
27	13,5	5,90	13,43	5,72	0,52	3,05
28	14	5,93	13,93	5,73	0,50	3,43
29	14,5	5,97	14,42	5,80	0,55	2,79
30	15	6,00	14,92	5,91	0,53	1,50
Error					0,61%	6,11%
Akurasi					99,39%	93,89%

3.5 Pengujian *Output ANN* terhadap Pembacaan Sensor

Pengujian yang dilakukan yaitu membandingkan *output* prediksi ANN terhadap nilai yang terbaca pada sensor. Seperti yang sudah disebutkan sebelumnya jika *output* prediksi ANN merupakan nilai *flow rate* dan *pressure rate* maka nilai tersebut akan dibandingkan dengan nilai *flow rate* dan *pressure rate* yang terbaca pada sensor. Pengujian *output ANN* terhadap nilai yang terbaca pada sensor ditunjukkan pada Tabel 4.

Tabel 4. Perbandingan *Output ANN* Terhadap pembacaan Sensor

RR [L/Min]	<i>Output</i> Perhitungan		<i>Output</i> Prediksi ANN		<i>Output</i> Purwarupa		Error Flow [%]	Error Pressure [%]
	Flow Rate [L/Min]	Pressure Rate [cmH2O]	Prediksi Flow [L/Min]	Prediksi Pressure [cmH2O]	Flow Rate [L/Min]	Pressure Rate [cmH2O]		
10	5	0,00	4,92	0,00	3,5	0,00	28,86	0,00
11	5,5	0,00	5,46	0,00	4,6	0,00	15,75	0,00
12	6	0,00	5,96	0,00	4,6	0,00	22,82	0,00
13	6,5	0,00	6,46	0,00	6,6	0,00	2,17	0,00
14	7	0,00	6,97	0,00	6,7	0,00	3,87	0,00
15	7,5	0,00	7,47	0,00	7,7	0,00	3,08	0,00
16	8	0,00	7,96	0,00	7,6	0,00	4,52	0,00
17	8,5	0,00	8,45	0,00	9,0	0,00	6,51	0,00
18	9	0,00	8,95	0,00	9,0	0,00	0,56	0,00
19	9,5	0,00	9,45	0,00	9,9	0,00	4,76	0,00
20	10	0,00	9,94	0,00	9,8	0,00	1,41	0,00
21	10,5	5,70	10,43	3,64	10,1	4,0	3,16	9,89
22	11	5,73	10,93	5,16	10,6	4,7	3,02	8,91
23	11,5	5,77	11,43	5,79	11,4	5,3	0,26	8,46
24	12	5,80	11,93	5,91	11,7	5,3	1,93	10,32
25	12,5	5,83	12,43	5,85	12,1	5	2,65	14,53
26	13	5,87	12,93	5,77	12,7	5	1,78	13,34
27	13,5	5,90	13,43	5,72	13,6	5	1,27	12,59
28	14	5,93	13,93	5,73	13,9	5,3	0,22	7,50
29	14,5	5,97	14,42	5,80	14,5	5,0	0,55	13,79
30	15	6,00	14,92	5,91	14,7	5,3	1,47	10,32
Rata-Rata Error							5,27%	10,97%
Akurasi							94,73%	89,03%

Berdasarkan hasil pada Tabel 4, diketahui jika masih terdapat *error* yang cukup jauh, baik itu nilai *flow rate* ataupun nilai *pressure rate*. Nilai persentase rata-rata *error* untuk pembacaan *flow rate* yaitu sebesar 5,27% dengan akurasi 94,73% sedangkan persentase *error* untuk *pressure rate* yaitu sebesar 10,97% dengan tingkat akurasi 89,03%. Volume tabung oksigen yang digunakan dapat mempengaruhi keakuratan pembacaan sensor terhadap hasil prediksi atau *output ANN*.

4. KESIMPULAN

Artificial Neural Network (ANN) dapat diimplementasikan pada rancangan alat terapi oksigen dengan baik melalui Raspberry Pi, *library* yang digunakan untuk membangun ANN sangat berpengaruh terhadap hasil performa sistem. Dalam membangun model ANN, dilakukan beberapa kali percobaan *training* dan hasil model *training* terbaik diperoleh dengan parameter fungsi aktivasi *sigmoid* dan enam belas *hidden layer* serta jumlah *epoch* sebanyak 3000 kali dengan persentase akurasi *output* ANN terhadap hasil perhitungan manual untuk *flow rate* sebesar 99,39% dan nilai akurasi *output* ANN untuk *pressure rate* sebesar 93,89%. Sedangkan persentase akurasi *output* ANN terhadap nilai pembacaan sensor *flow* sebesar 94,73% dan nilai akurasi *output* ANN terhadap nilai pembacaan sensor *pressure* sebesar 89,03%.

DAFTAR RUJUKAN

- Anekawati, A., Wibisono, A., & Annisa, S. (2021). Kadar Oksigen Terbaik Sebagai Daya Tarik Wisata Kesehatan Pada Pulau Gili Iyang. *PERFORMANCE: Jurnal Bisnis & Akuntansi*, 11(1), 63–74. <https://doi.org/10.24929/feb.v11i1.1317>
- Ardhianto, F. W. (n.d.). *Design Of Adaptive Oxygen Flow Control For Oxygen Therapy Devices*. *Mv*.
- Aryanto, D., & Augusman, V. (2021). Penerapan Machine Learning Untuk Mengategorikan Sampah Plastik Rumah Tangga. *Jurnal TIMES*, 10(1), 39–43. <https://ejournal.stmik-time.ac.id/index.php/jurnalTIMES/article/view/649>
- Dai, M., Zhang, Z. S., Liu, Z. G., & Yin, D. F. (2013). Control module design for a continuous positive airway pressure ventilator. *Applied Mechanics and Materials*, 321–324, 1657–1661. <https://doi.org/10.4028/www.scientific.net/AMM.321-324.1657>
- Irawati, I. D., Hadiyoso, S., Alfaruq, A., Novianti, A., & Rizal, A. (2022). Self-Oxygen Regulator System for COVID-19 Patients Based on Body Weight, Respiration Rate, and Blood Saturation. *Electronics (Switzerland)*, 11(9). <https://doi.org/10.3390/electronics11091380>
- Kane, B., Decalmer, S., & O'Driscoll, B. R. (2013). Emergency oxygen therapy: From guideline to implementation. *Breathe*, 9(4), 247–254. <https://doi.org/10.1183/20734735.025212>
- Kesuma, M. F., & T, M. Z. S. (2020). *Perbandingan Sistem Kendali PID dan Fuzzy Logic Pada Quadcopter Jurusan Teknik Elektro, Fakultas Teknologi Industri Universitas Bung Hatta Padang – Indonesia fajarkesuma07@gmail.com*. 2020.
- Khairunnisa, S., Gede, I. D., Wisana, H., Priyambada, I., Nugraha, C., & Elektromedik, J. T. (2018). Rancang Bangun Pulse Oximeter Berbasis Iot (Internet of Things). *E-Journal Poltekes Kemenkes Surabaya*, 1–9.
- Malaekah, E., Al Awam, K., Farouk, H., Abuabid, E., Mukhanov, V. V., Alahmari, A., & Alshagag, H. (2022). Designing Hybrid Mechanical Ventilator System Based on Arduino and

- Raspberry Pi 4. *Journal of Medical Devices, Transactions of the ASME*, 16(1), 0–7.
<https://doi.org/10.1115/1.4054036>
- Nugraha, N. W. (n.d.). *Design of Adaptive Oxygen Flow And Pressure Control For Oxygen Therapy Devices Using The Fuzzy Logic Method*.
- Nugroho, C. R., Yuniarti, E., & Hartono, A. (2020). Alat Pengukur Saturasi Oksigen Dalam Darah Menggunakan Metode Photoplethysmograph Reflectance. *Al-Fiziya: Journal of Materials Science, Geophysics, Instrumentation and Theoretical Physics*, 3(2), 84–93.
<https://doi.org/10.15408/fiziya.v3i2.17721>
- Purba, L. L. S., & Harefa, N. (2020). Pengaruh Kandungan Oksigen Udara Sekolah Terhadap Konsentrasi Belajar Siswa SMAN 9 Jakarta Timur. *Jurnal EduMatSains*, 4(2), 169–182.
<https://media.neliti.com/media/publications/14469-ID-analisa-kadar-co-dan-no2-di-udara-dan-keluhan-gangguan-saluran-pernapasan-pada-p.pdf>
- Sahid Dwi Prasetyo, Rizka Hayyu Nafi'ah, G. S. (2020). Asuhan Keperawatan Gawat Darurat Pada Pasien Dengan Stroke Hemoragik di Ruang Instalasi Gawat Darurat. *Universitas Kusuma Husada Surakarta*.
- Salam, A., Rokhim, I., Supriyanto, H., Suryatini, F., & Wiyono, A. (2021). Rancang Bangun Ventilator Controller Berbasis Tekanan dengan Teknologi Internet of Things. *JTERA (Jurnal Teknologi Rekayasa)*, 6(1), 53. <https://doi.org/10.31544/jtera.v6.i1.2021.53-60>
- Sita Radhakrishnan, Suresh G. Nair, J. I. (2022). No Title. *Biomedical Signal Processing and Control, Volume 71*, (Multilayer perceptron neural network model development for mechanical ventilator parameters prediction by real time system learning).
<https://doi.org/https://doi.org/10.1016/j.bspc.2021.103170>
- Timorremboko, F., & Karya, O. T. (2020). Implementasi Jaringan Syaraf Tiruan Pada Kendali Lampu Sorot Mobil Adaptif Berbasis Python. *Jurnal Teknologi Elektro*, 11(3), 142.
<https://doi.org/10.22441/jte.2020.v11i3.006>