

Rancang Bangun Sistem Prediksi Dan Klasifikasi Radius *Shaping* Pada Proses *Curing* Ban Menggunakan Metode DNN

Bugi Nur Rohman¹

¹Teknik Elektro, Universitas Mercu Buana, Jakarta
buginurrohman7@gmail.com

Abstrak— Proses *shaping* adalah salah satu tahapan *curing* atau pemasakan *green tire* (ban setengah jadi). *Shaping* yaitu proses pembentukan *green tire* atau ban setengah jadi, dimana *green tire* diletakkan pada *bladder* dan diberi tekanan sesuai keinginan. Dalam proses *shaping* terdapat nilai tekanan dan dimensi (radius) *bladder* yang menjadi salah satu parameter penting yang berpengaruh terhadap hasil produk. Terdapat mesin *curing* yang belum menggunakan sistem radius *shaping* dan sistem IoT untuk melakukan pemantauan proses tersebut. Hal ini menyebabkan analisa cacat produk oleh mesin *curing* tidak optimal dan sulit menentukan nilai radius yang optimal, maka dibutuhkan alat untuk membuat sistem pemantauan nilai radius *shaping*, dan alat untuk mengklasifikasi dan memprediksi nilai parameter *shaping* (radius dan tekanan). Berdasarkan permasalahan tersebut penulis mencoba membuat purwarupa sistem pemantauan, klasifikasi dan prediksi nilai radius *shaping* menggunakan metode Deep Neural Network (DNN) dengan nilai masukan nilai radius dari sensor ultrasonik HC-SR04 yang terkoneksi dengan NodeMcu, kemudian menampilkan nilai radius *shaping* beserta hasil klasifikasi dan prediksi pada web server ExpressJs Heroku dan aplikasi Telegram. Penelitian ini bertujuan untuk menentukan nilai radius *shaping* dan mengklasifikasikannya, kemudian menampilkan hasilnya secara publik. Dari hasil pengujian diperoleh sistem pemantauan, prediksi dan klasifikasi nilai radius *shaping*. Hasil pengujian tersebut adalah Model DNN dengan metode prediksi yang optimal pada proses pelatihan Model ke-17 dimana memiliki akurasi 99.06% dan loss 0.16%. Kemudian untuk klasifikasi yang optimal pada proses pelatihan Model ke-9 dimana memiliki akurasi 99.86% dan loss 0.38%. Model tersebut kemudian digunakan pada web server ExpressJS untuk diintegrasikan dengan bot telegram.

Kata Kunci— Bot telegram, *Curing*, Deep Neural Network, ExpressJs, IoT, Klasifikasi, NodeMcu, Prediksi, *Shaping*, Ultrasonik

DOI: 10.22441/jte.2025.v16i2.009

I. PENDAHULUAN

Proses *curing* atau dikenal juga dengan proses vulkanisasi merupakan tahap akhir dari proses pembuatan ban. Selama proses vulkanisasi, sifat senyawa berubah dan karet plastik menjadi elastis dengan membentuk ikatan silang dalam struktur molekulnya (Yuniarto and Rangkuti 2019). Pada saat pemasakan ban, *green tire* atau ban setengah jadi akan diletakkan pada *mold* bawah dan bagian dalam *green tire* mengelilingi *bladder* (pembentuk ban bagian dalam) kemudian

mold atas akan menutup untuk melakukan proses *curing* lebih lanjut (Sumathy B. dkk., 2019). Pada proses *mold* menutup terdapat proses *Shaping*, proses *shaping* adalah proses pembentukan *green tire* atau ban setengah jadi, dimana *green tire* diletakkan pada *bladder* dan diberi tekanan sesuai keinginan (Lee 1968).

Berdasarkan data dan pengamatan dilapangan pada proses *curing* cacat produk rata-rata sebesar 70% disebabkan oleh proses *shaping* yang tidak sesuai. Dalam proses *shaping* terdapat nilai tekanan dan dimensi (radius) *bladder* yang menjadi salah satu parameter penting yang berpengaruh terhadap hasil produk. Parameter *shaping* sudah ditentukan dan tercantum dalam spesifikasi melalui analisa dilapangan oleh departemen terkait. Terdapat mesin *curing* yang sudah menggunakan sensor untuk mengendalikan radius *shaping* namun belum ada sistem IoT untuk melakukan pemantauan proses tersebut, selain itu penggunaan sistem kendali radius *shaping* hanya dimiliki beberapa mesin saja. Mesin yang sudah dilakukan pengaturan (setting parameter) nilai radius sesuai spesifikasi, aktual nilai tersebut tidak sesuai dengan pengaturan dan selalu berubah mengikuti kondisi mesin, sedangkan untuk mesin yang belum menggunakan sistem kendali radius *shaping* pengaturan hanya berdasarkan parameter tekanan pada *shaping*. Hal tersebut menyebabkan analisa cacat produk oleh mesin *curing* tidak optimal, maka dibutuhkan alat untuk membuat sistem pemantauan nilai radius *shaping*, dan alat untuk mengklasifikasi dan memprediksi nilai parameter *shaping* (radius dan tekanan) agar mesin yang belum terdapat sistem kendali radius *shaping* dapat menyerupai mesin yang lain, dan alat tersebut diharapkan dapat memudahkan analisa dan menjaga hasil produk tetap sesuai.

II. PENELITIAN TERKAIT

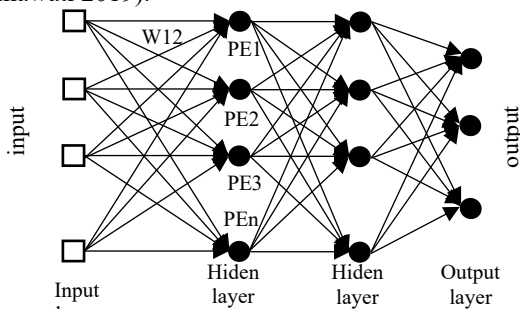
Salah satu penelitian sebelumnya yang dilakukan berkaitan dengan mesin *curing*, salah satunya penelitian yang dilakukan oleh Bayu Dwi Subiantoro yang berjudul “Peningkatan Kinerja Programmable Logic Control Mitsubishi Seri Q Menggunakan Optical Distance Sensor O1d100 Berbasis Sistem Kontrol *Shapping*”, pada tahun. Dalam penelitian tersebut dilakukan pengendalian proses *shaping* menggunakan Programmable Logic Control dan Optical Distance Sensor O1d100 sebagai sensor masukan (Subiantoro, Bayu D., 2019).

Pada bab ini akan dijelaskan hasil dari studi literatur dan teori-teori yang menjadi landasan pembahasan masalah dan

teori pendukung berdasarkan referensi yang berkaitan dengan penelitian ini.

A. Neural Network : Multi Layer Perceptron

Neural network adalah pemrosesan informasi sistem. Secara umum neural network dapat dianggap sebagai sistem kotak hitam yang menerima input dari lingkungan dan menghasilkan output. Neural network mengandung elemen pemrosesan dan pembobotan yang saling terhubung. Setiap lapisan dalam jaringan berisi oleh kelompok elemen pemrosesan seperti yang ditunjukkan pada gambar 2.1.(Hadianto, Novitasari, and Rahmawati 2019).



Sumber: (Fadly, Uddin, and Sutarto 2002)

Gambar 1. Multilayer perceptron neural networks

B. Artificial Neural Network

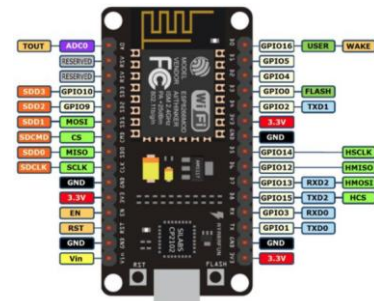
Artificial Neural Network (Jaringan Syaraf Tiruan) adalah model non-linear yang kompleks, dibangun dari komponen yang secara individu berperilaku mirip dengan model regresi. Jaringan syaraf tiruan dapat divisualisasikan dengan grafik, dan beberapa sub-grafik beberapa memiliki perilaku yang sama dengan grafik sebelumnya. Meskipun struktur dari jaringan saraf secara eksplisit dirancang terlebih dahulu, pengolahan jaringan saraf ini tidak untuk menghasilkan hipotesis (berbagai neuron dan pengolahan lainnya terstruktur dalam jaringan) berkembang selama proses pembelajaran. Hal ini memungkinkan neuron yang membentuk jaringan akan digunakan sebagai pemecahan masalah dari "program itu sendiri".(Hadianto et al. 2019).

C. Arsitektur Backpropagation

Algoritma back-propagation adalah algoritma yang paling populer dalam pembelajaran jaringan saraf tiruan. Algoritma ini didasarkan pada aturan pembelajaran dengan koreksi kesalahan (Fadly et al. 2002). Pembelajaran kesalahan-koreksi pada back-propagation dibagi dalam dua cara utama; pergerakan maju dan mundur. Di depan, vektor input diterapkan pada layer input di setiap jaringan kemudian mempengaruhi semua jaringan, lapis demi lapis. Output dari jaringan adalah respon dari jaringan saraf. Bobot di setiap lapisan di depan adalah tetap. Gambar dan persamaan pembelajaran maju ditunjukkan dalam gambar 2 dan persamaan 1 dan 2 (Fadly et al., 2002).

D. NodeMCU

NodeMCU adalah salah satu mikrokontroler papan tunggal dengan fungsi WiFi, yang dapat digunakan untuk membuat produk platform IoT. NodeMCU tersusun dari hardware berupa system-on-chip Esp8266.



Sumber : (Adinda et al., 2019)

Gambar 2. Modul Sheet NodeMCU ESP8266

ESP8266 adalah modul WiFi yang disukai oleh pengembang perangkat keras. Keunggulan utama memiliki harga yang sangat terjangkau dan keserbagunaan membuat ESP8266 menjadi produk yang paling diminati. ESP8266 memiliki fitur SoC (system on chip), memungkinkan pengguna untuk memprogram tanpa mikrokontroler tambahan. ESP8266 juga dapat digunakan sebagai titik akses atau klien Adhoc. Untuk pemrograman, sAnda dapat menambahkan ESPlorer ke firmware berbasis NodeMCU dan menggunakan Putty sebagai kontrol terminal dari perintah AT. Jangan lupa tambahkan library ESP8266 ke board manager di Arduino IDE nantinya (Adinda et. al, 2019).

E. Bot Telegram

Telegram adalah aplikasi dan alat enkripsi berbasis cloud. Sebagai aplikasi SMS real-time, Telegram memberikan kemudahan bagi pengguna untuk mengaksesnya karena tersedia di platform mobile dan desktop. Pada platform mobile, Telegram dapat digunakan pada platform iphone, android dan windows phone, sedangkan pada platform desktop, Telegram dapat digunakan pada browser Windows, Linux, Mac OS dan Web. Telegram juga menyediakan platform bagi pengembang yang ingin menggunakan API terbuka dan protokol yang disediakan oleh pengembangan Bot Telegram yang tercatat di situs resminya (Nufusula and Susanto 2018).

Bot adalah agen interaktif, program komputer yang dirancang untuk Simulasikan dialog cerdas dengan satu atau lebih pengguna manusia melalui pengenalan suara dan antarmuka obrolan. Di Telegram, Bot adalah akun Telegram khusus yang dirancang untuk membalas pesan secara otomatis tanpa memerlukan nomor telepon tambahan. Pengguna dapat berinteraksi dengan robot dengan mengirimkan pesan perintah melalui pesan pribadi atau grup. Bot biasanya diprogram untuk berinteraksi seperti orang normal. Bot tersebut dapat digunakan untuk mengingatkan (reminder), memutar, menyiarkan, bahkan

mengirim perintah ke perangkat lain (Nufusula and Susanto 2018).

F. Heroku

Heroku adalah platform cloud yang menjalankan bahasa pemrograman tertentu. Heroku mendukung bahasa pemrograman seperti Ruby, Node.js, Python, Java, dan PHP. Heroku termasuk dalam standar platform as a service (PaaS), jadi bagi yang ingin men-deploy aplikasi ke heroku, cukup mengonfigurasi aplikasi yang ingin di-deploy dan menyediakan platform yang memungkinkan pelanggan untuk mengembangkan dan menjalankannya tanpa harus membangun aplikasi yang kompleks, dan memelihara infrastruktur yang biasanya terkait dengan pengembangan dan penerapan aplikasi.

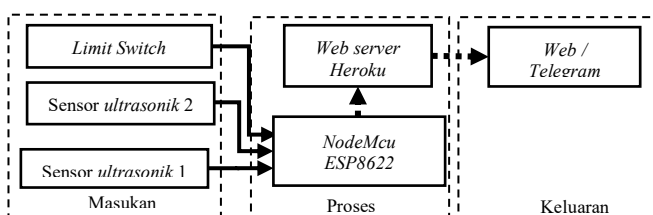
III. PERANCANGAN ALAT DAN SISTEM

Sistem pada rancangan ini bertujuan untuk pemantauan, memprediksi serta mengklasifikasi nilai radius shaping dari pembacaan sensor ultrasonik. Hasil pembacaan nilai radius shaping dari sensor akan diklasifikasi kemudian diprediksi nilai parameter shapingnya menggunakan metode DNN. Hasil klasifikasi yaitu menentukan nilai radius shaping tersebut dapat menghasilkan cacat produk atau tidak, sedangkan hasil prediksi menentukan pengaturan nilai tekanan dan radius shaping. Hasil klasifikasi, prediksi dan nilai radius shaping akan dipantau melalui telegram atau web server sehingga memudahkan dalam pemantauan.

Proses klasifikasi dan prediksi pada sistem ini menggunakan Jaringan Syaraf Tiruan dengan metode DNN yang sudah melalui tahap pelatihan menggunakan google colaboratory, kemudian model yang sudah dilatih di-Deploy pada cloud platform Heroku melalui GitHub sehingga dapat diakses oleh publik.

A. Blok Diagram

Pada sub bab ini akan membahas tentang blok diagram yang ada pada penelitian. Blok diagram tersebut meliputi blok diagram perancangan secara keseluruhan dan blok diagram sistem Jaringan Syaraf tiruan. Blok diagram memiliki fungsi yaitu untuk menjelaskan cara kerja sistem secara sederhana. Berikut adalah blok diagram dari sistem secara keseluruhan dapat dilihat pada gambar 3.



Gambar 3. Blok diagram sistem keseluruhan

1. Masukan

- bladder sebelah kanan kemudian mengirimkan hasil pembacaan ke NodeMcu untuk diproses.

- Sensor ultrasonik 1 berfungsi untuk membaca nilai radius shaping bladder sebelah kiri kemudian mengirimkan hasil pembacaan ke NodeMcu untuk diproses.

- Limit switch berfungsi sebagai trigger untuk memulai pembacaan sensor ultrasonik.

2. Proses

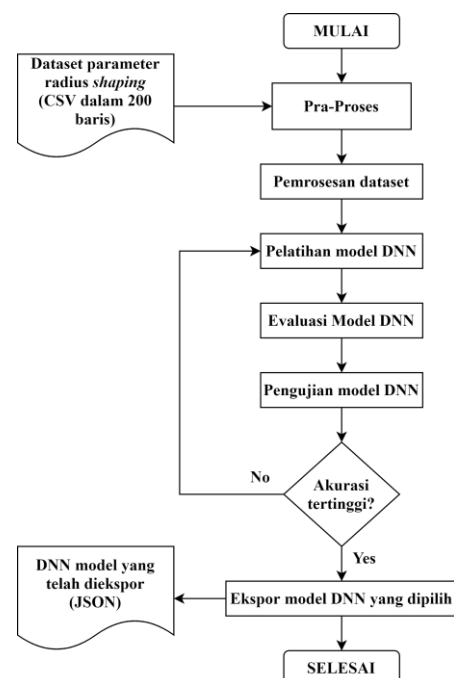
- NodeMcu berfungsi untuk mengontrol sistem elektrik, menerima nilai masukan sensor, dan mengirimkan nilai pembacaan sensor ke web server.
- Web Server Heroku berfungsi sebagai server dimana nilai yang diterima dari NodeMcu diklasifikasi dan diprediksi menggunakan metode DNN yang sebelumnya sudah dilatih kemudian di Deploy ke web server heroku.

3. Keluaran

- Relay merupakan saklar yang berfungsi untuk mengendalikan solenoid valve yang terdapat pada mesin curing.
- Telegram disini berfungsi sebagai media untuk pemantauan hasil klasifikasi, prediksi dan nilai aktual radius shaping melalui bot telegram.

B. Alur Kerja Sistem JST Model DNN

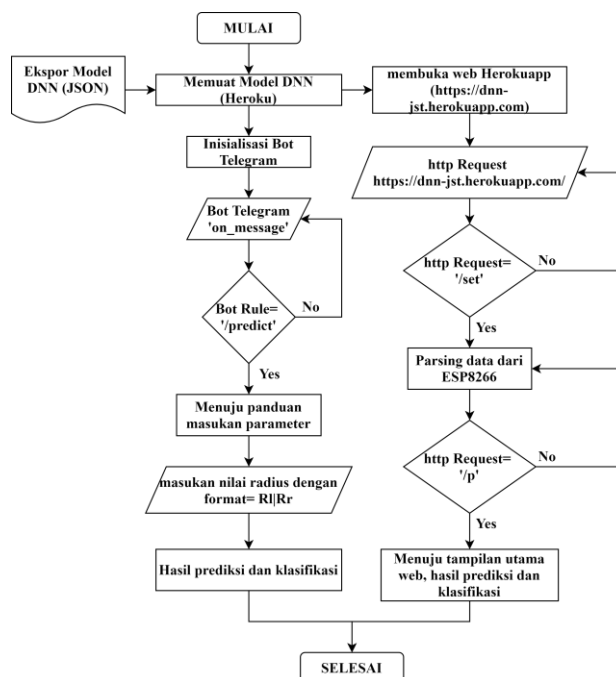
Pada sub bab ini dimana membahas blok diagram sistem Jaringan Syaraf Tiruan. Blok diagram ini memiliki fungsi yaitu untuk menjelaskan cara kerja sistem secara sederhana. Adapun diagram blok dari sistem yang dirancang adalah sebagai berikut.



Gambar 4. Blok Diagram Sistem Jaringan Syaraf Tiruan

Berdasarkan Gambar 4 secara sekuensial dapat dideskripsikan sebagai berikut:

- Pengambilan dataset: Data pengukuran radius shaping sebanyak 200 data dengan variabel radius (*bladder* kanan & kiri), radius optimal (pengaturan radius yang disarankan) dan klasifikasi.
- Pra - proses: Melakukan *indexing* data parameter radius *shaping* sehingga terdiri dari radius kanan, radius kiri, tekanan *shpaing* radius optimal dan klasifikasi
- Pemrosesan dataset: merupakan proses penyesuaian dataset dalam bentuk arsitektur DNN yang diinginkan berdasarkan jumlah *neuron* masukan dan keluaran.
- Pelatihan model DNN merupakan proses pelatihan data pada model arsitektur DNN yang dipilih.
- Evaluasi model DNN yang telah dilatih: Merupakan proses evaluasi tingkat akurasi yang telah didapatkan pada proses sebelumnya, dengan mengambil sampel data dalam pengujian tingkat akurasi yang dihasilkan tersebut.
- Pengujian model : Menguji hasil pelatihan model DNN untuk memperoleh tanggapan yang diinginkan terhadap masukan yang diberikan.
- Ekspor model : Mengekspor model arsitektur DNN yang sudah dilatih untuk dapat diaplikasikan atau digunakan kembali pada *platform* lain.



Gambar 5. Diagram alir integrasi Bot Telegram dan aplikasi web dengan Server

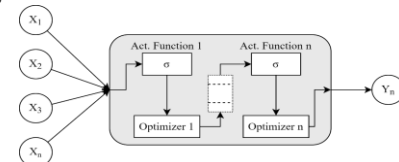
Bersadarkan Gambar 5. secara sekuensial dapat dideskripsikan sebagai berikut:

- Memuat model DNN: Proses ekstraksi dari arsitektur DNN yang telah dilatih sebelumnya sehingga tidak ada proses pelatihan pada web server ExpressJS.

- Inisialisasi Bot Telegram: Proses inisialisasi token bot telegram dari bot father telegram.
- Bot Rule = '/predict': Merupakan perintah kontrol bot (bot command) dengan nama 'Predict', dimana akan mengarahkan pengguna untuk mengirimkan nilai radius shaping sesuai dengan panduan yang ada yang ditampilkan setelah mengirimkan perintah bot '/predict'.
- Hasil prediksi dan klasifikasi merupakan hasil akhir dari proses pada bot telegram, dimana ditampilkan nilai dari hasil prediksi dan klasifikasi.
- Membuka web Herokuapp, pengguna akan diarahkan pada URL <https://dnn-jst.herokuapp.com/> yaitu halaman awal saat memulai aplikasi.
- http request = '/p' merupakan permintaan pengguna pada server untuk menampilkan hasil dari prediksi dan klasifikasi kemudian pengguna akan diarahkan pada halaman utama aplikasi web prediksi dan klasifikasi radius shaping.

C. Arsitektur DNN

Berikut merupakan model arsitektur DNN yang diusulkan seperti pada gambar 6.



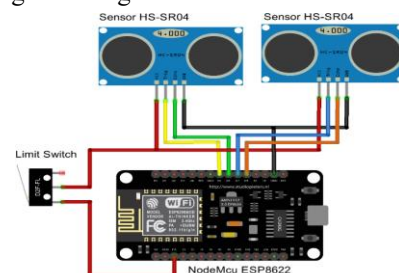
Gambar 6. Model Arsitektur DNN yang diusulkan.

Berdasarkan 6, dataset dapat dipetakan ke dalam model DNN yang terdiri dari input layer (X_1 s/d X_n) dengan variabel nilai radius kanan dan kiri kemudian output layer (Y_n) hasil prediksi dan klasifikasi. Tabel 1. menunjukan hyperparameter untuk model arsitektur DNN yang diusulkan.

Tabel 1. Hyperparameter model arsitektur DNN

DNN Hyperparameter	
<i>Num of hidden layers</i>	[2, 4, 6, 10, 16]
<i>Batch size & num of layers</i>	[32, 64, 96, 128]
<i>Activation function</i>	Sigmoid, ReLu, Leaky-ReLu
<i>Optimizer</i>	SGD, RMSProp, Adam
<i>Learning rate</i>	[0.001, 0.0001, 0.00001, 0.000001]

D. Perancangan Perangkat Keras



Gambar 7. Wiring diagram perangkat keras.

Berdasarkan gambar 7, perangkat keras yang digunakan dalam penelitian ini terdiri dari modul NodeMcu ESP8622 dan sensor *ultrasonic* HC-SR04. Sensor ultrasonic berfungsi sebagai pengirim, penerima, dan pengontrol gelombang *ultrasonic*. Sensor ini memiliki 4 pin, pin Vcc, Gnd, Trigger, dan Echo. Pin Vcc untuk listrik positif dan Gnd untuk ground-nya. Pin Trigger digunakan untuk trigger keluarnya sinyal dari sensor dan pin Echo digunakan untuk menangkap sinyal pantul dari benda. Selisih waktu pengiriman dan penerimaan sinyal digunakan untuk menentukan jarak benda tersebut. Adapun NodeMcu ESP8622 digunakan untuk mengirimkan data dari sensor *ultrasonic* HC-SR04 ke server berupa data radius. Kemudian untuk *limit switch* digunakan sebagai *trigger* sensor ultrasonik untuk memulai pengukuran. Koneksi perangkat keras ke modul NodeMcu ESP8622 dapat dilihat pada Tabel 2.

Tabel 2. Koneksi Pin NodeMcu ESP8622

Pin NodeMcu ESP8622	Limit Switch	Pin HC-SR04 (1)	Pin HC-SR04 (2)
VU (5v)	Com	-	-
-	NO	Vcc	Vcc
Gnd	-	Gnd	Gnd
D5	-	Trigger	-
D6	-	Echo	-
D7	-	-	Trigger
D8	-	-	Echo

IV. HASIL DAN ANALISA

A. Hasil Model DNN

Bagian ini menunjukkan hasil perancangan model DNN yang diusulkan meliputi hasil pemrosesan dataset, hasil perancangan model, pelatihan model, evaluasi model, pengujian model dan ekspor model.

1) Pemrosesan Dataset

Berikut hasil pemrosesan dataset dalam format sesuai model yang diusulkan, dapat dilihat pada Tabel 1.

Tabel 3. Dataset Radius Shaping

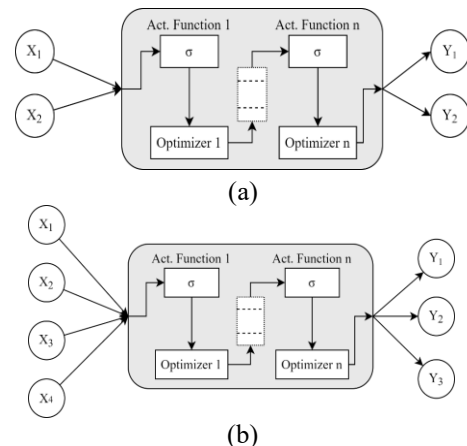
r	l	p	S	Nr	nl	np	ns	la
225	226	1.0	226	1.5223	1.5829	1.5742	1.5568	2
226	226	1.0	225	1.5223	1.4538	1.5742	1.4919	2
217	217	0.8	217	0.4843	0.4207	0.4195	0.4536	0
208	208	0.8	208	0.6833	0.7414	0.4195	0.7143	0
206	207	0.8	206	0.9428	0.8705	0.4195	0.909	1
206	205	0.8	205	0.9428	1.1288	0.4195	1.0387	1

Tabel 1. menunjukkan sample dataset radius shaping, dimana input radius kiri (r), radius kanan (l), output tekanan shaping (p), output setting radius (s), kemudian nilai tersebut dinormalisasi dengan urutan dari kanan menjadi nr, nl, np dan nSr. Normalisasi dilakukan dengan mencari rata – rata kemudian standar deviasi setiap nilai input dan output. Pada sisi output (label) terdapat 3

variable dengan label 1 : defect CIL, label 2 : defect CB dan label 0 : OK.

2) Hasil Perancangan Model Arsitektur DNN

Berikut merupakan hasil perancangan model arsitektur DNN meliputi model arsitektur prediksi dan klasifikasi.

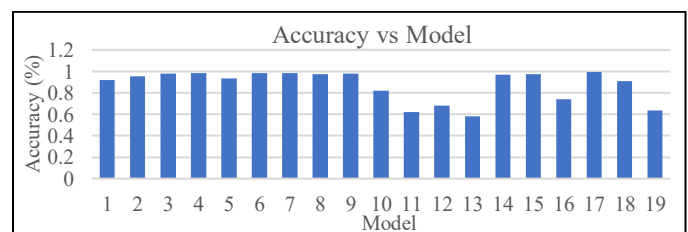


Gambar 8. Model arsitektur DNN (a) prediksi, (b) klasifikasi.

Berdasarkan gambar 8(a). dimana pada sisi input X1 dideklarasikan sebagai rating dari variabel radius kanan (r), X2 dideklarasikan sebagai rating dari variabel radius kiri (l), dan pada sisi output Y1 dideklarasikan sebagai variabel prediksi tekanan shaping (p), Y2 dideklarasikan sebagai rating dari prediksi variabel radius shaping (s). Sedangkan model arsitektur DNN klasifikasi pada gambar 8(b). input X1 dideklarasikan sebagai rating dari variabel radius kanan (r), X2 dideklarasikan sebagai rating dari variabel radius kiri (l), X3 dideklarasikan sebagai rating dari variabel tekanan shaping (p), X4 dideklarasikan sebagai rating dari variabel setting radius (s) dan pada sisi output Y1 dideklarasikan sebagai variabel label (la) hasil klasifikasi (label 1 : defect CIL, label 2 : defect CB dan label 0 : OK.).

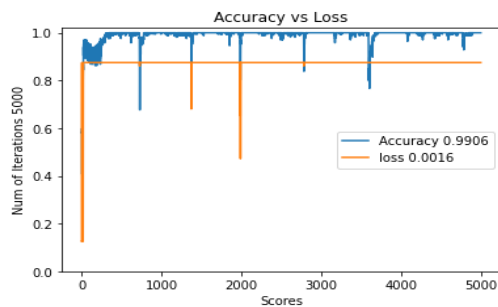
3) Evaluasi Model DNN

Bagian ini menunjukkan hasil evaluasi dimana data-data hasil pelatihan sebelumnya direkap.



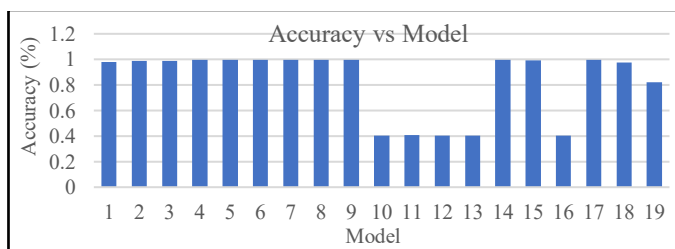
Gambar 9. Performa akurasi prediksi setiap model

Gambar 9 menunjukkan rekap akurasi prediksi tiap model DNN dimana menunjukkan grafik pada proses pelatihan 19 model.



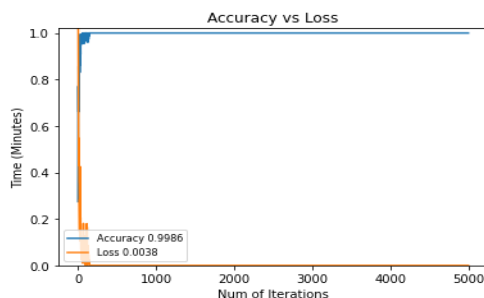
Gambar 10. Grafik Akurasi dan Loss pada 5000 iterasi

Gambar 10 menunjukkan model terpilih ke-17 dimana memiliki akurasi 99.06% dan loss 0.0016% dalam 5000 iterasi.



Gambar 11. Performa akurasi klasifikasi setiap model.

Gambar 11. Menunjukkan rekap akurasi klasifikasi setiap model DNN dimana menunjukkan grafik pada proses pelatihan 19 model.



Gambar 12. Grafik Akurasi dan Loss pada 5000 iterasi

Gambar 12 menunjukkan model terpilih ke-9 dimana memiliki akurasi 99.86% dan loss 0.0038% dalam 5000 iterasi.

4) Pengujian model DNN

Bagian ini menunjukkan pengujian hasil pelatihan model DNN dengan dataset yang ada guna mengetahui tanggapan yang diinginkan terhadap masukan yang diberikan.

Tabel 4. Hasil pengujian model DNN dengan dataset.

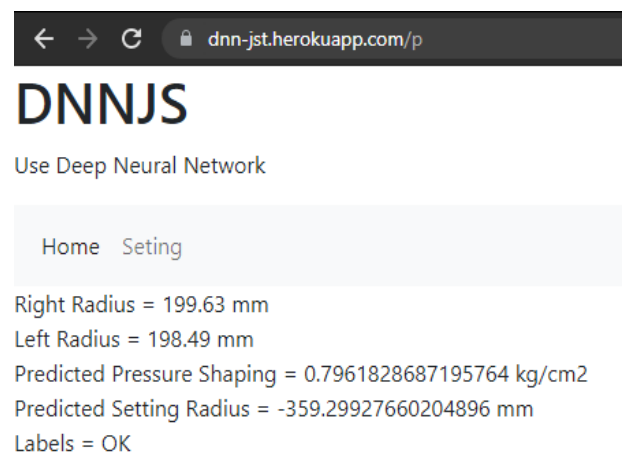
Input		Dataset			Prediksi & Klasifikasi			Error
r	l	p	s	la	p	s	la	
225	226	1.0	226	2	1.00	226	2	0.0%
223	225	1.0	224	2	1.00	224	2	0.0%
220	221	0.9	221	2	0.90	221	2	0.0%
218	217	0.9	218	0	0.90	218	0	0.0%

213	210	0.8	213	0	0.79	211	0	0.9%
208	210	0.8	208	0	0.79	209	0	1.0%
207	207	0.8	207	1	0.80	207	1	0.0%
202	205	0.8	203	1	0.78	204	1	1.5%
200	201	0.7	201	1	0.79	201	1	6.6%

Berdasarkan tabel 4 dapat dilihat hasil pengujian yang dilakukan pada sembilan sample dataset dengan hasil rata – rata eror adalah 1,1% dan eror tertinggi mencapai 6,6% pada nilai masukan ‘r’ adalah 200 dan ‘l’ adalah 201.

B. Hasil Perancangan IoT

Gambar 13 menunjukkan tampilan hasil aplikasi yang berhasil dibuat dengan mengakses ‘https://dnn-jst.herokuapp.com/p’, maka aplikasi tersebut sudah dapat diakses oleh publik dan dapat diintegrasikan dengan *bot telegram*.



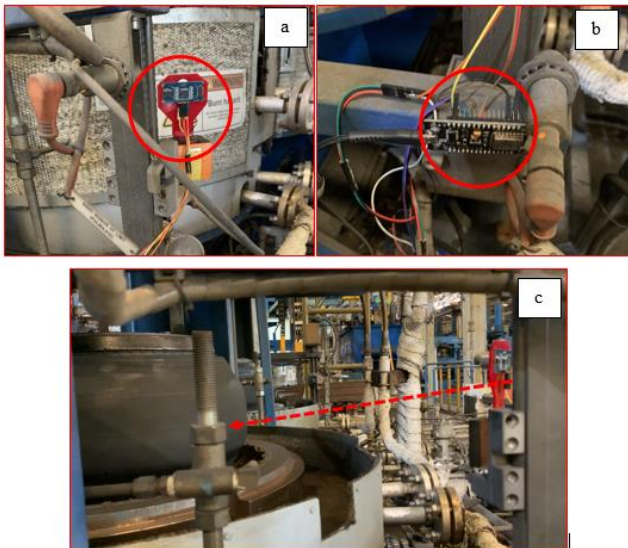
Gambar 13. Tampilan aplikasi web radius *shaping*.

C. Hasil Perancangan Perangkat Keras

Pada bagian ini menjelaskan hasil dari perancangan perangkat keras meliputi Nodemcu, sensor jarak dan *limit switch*, kemudian dilakukan pengujian akurasi pada sistem tersebut.

1. Hasil implementasi perangkat keras

Hasil perancangan perangkat keras berupa purwarupa alat pemantauan radius *shaping*. Hasil perancangan dapat dilihat pada gambar 14.

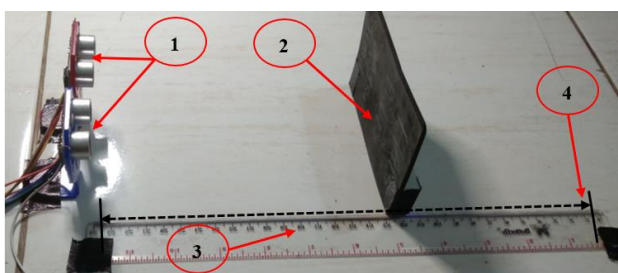


Gambar 14. Purwarupa alat pemantauan radius shaping (a) sensor ultrasonik (b) NodeMcu (c) arah dan penempatan sensor ultrasonik.

Gambar 14 memperlihatkan hasil perancangan perangkat keras yang sudah terpasang pada mesin *curing* dimana gambar 14.(a) menunjukkan implementasi sensor ultrasonik, dan gambar 14.(b) menunjukkan pemasangan nodemcu pada mesin *curing*. Pada gambar 14.(c) menunjukkan pemasangan sensor ultrasonik secara tegak lurus dan center dengan tengah *bladder*.

2. Pengujian perangkat keras

Dalam tahap ini dilakukan pengujian perangkat keras untuk mengetahui akurasi atau kesalahan pengukuran oleh sistem. Pengujian dilakukan dengan mengukur jarak suatu objek dengan variasi jarak tertentu oleh sensor ultrasonik. Nilai yang ingin didapat adalah radius *shaping* maka metode pengukuran yaitu menentukan titik nol radius (titik tengah *bladder*) kemudian dikurangi nilai jarak yang dibaca sensor, sehingga semakin dekat jarak objek (*bladder*) dengan sensor maka semakin besar nilai radius. Objek yang diukur berbahan dasar karet sehingga pengujian dilakukan menggunakan bahan dasar yang sama. Berdasarkan metode tersebut maka simulasi pengujian dilakukan seperti gambar 15.



Gambar 14. Metode pengujian perangkat keras

Keterangan gambar 14= 1. Sensor ultrasonik; 2. Objek (bahan dasar karet); 3. Penggaris; 4. Titik nol

Pengujian dilakukan dengan membandingkan jarak objek yang dibaca oleh sensor ultrasonik dengan pengukuran jarak manual menggunakan penggaris. Hasil dari pengujian tersebut dapat dilihat pada tabel 5.

Tabel 5. Hasil pengujian pembacaan radius *shaping*.

No.	Pengukuran Sensor (mm)		Pengukuran manual (mm)	Error (%)	
	Sensor 1	Sensor 2		Sensor 1	Sensor 2
1	201.08	199.80	200	0.54%	0.10%
2	204.71	205.23	205	0.14%	0.11%
3	209.89	210.17	210	0.05%	0.08%
4	215.89	215.14	215	0.41%	0.07%
5	222.16	220.78	220	0.98%	0.35%
6	227.12	225.50	225	0.94%	0.22%

Berdasarkan tabel 5 dapat diketahui rata – rata error pengukuran kedua sensor adalah 0,33% dan error pengukuran terbesar pada sensor 1 yaitu 0,98%, hal tersebut membuktikan bahwa sensor mampu mengukur radius *shaping*.

D. Pengujian Respon Pengiriman Data Ke Server Heroku.

Pada bagian ini dilakukan pengujian terhadap respon server terhadap penerimaan data oleh ESP8266 dengan mendata nilai, ukuran data dan waktu pengiriman data ke server.

Tabel 6. Data respon penerimaan data oleh server

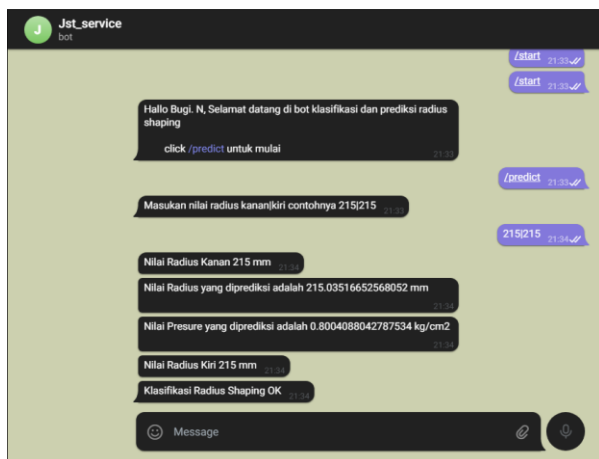
No.	Nilai pada serial monitor		Diterima server		Ukuran data (Byte)	Waktu (ms)
	l	r	l	r		
1	201.08	199.8	201.08	199.8	206	649
2	204.71	205.23	204.71	205.23	305	436
3	209.89	210.17	209.89	210.17	206	653
4	215.89	215.14	215.89	215.14	252	438
5	222.16	220.78	222.16	220.78	254	451
6	227.12	225.5	227.12	225.5	206	675

Pengujian dilakukan dengan mendata nilai yang ada pada serial monitor (arduino IDE), nilai yang diterima server, ukuran data yang diterima dan waktu penerimaan data. Berdasarkan tabel 4.10 dilakukan enam kali pengambilan sampel respon *server*, diketahui rata – rata ukuran data yang diterima oleh server yaitu 283,2 Byte dengan rata – rata waktu adalah 550,3 ms.

E. Hasil Perancangan Bot Telegram

Gambar 15 memperlihatkan hasil tampilan bot telegram dimana dalam bot tersebut dapat digunakan untuk prediksi dan klasifikasi nilai radius *shaping* dengan mengirimkan perintah bot `"/start"` lalu perintah `"/predict"`, kemudian akan muncul panduan format penulisan masukkan nilai radius *shaping* kanan dan kiri (`"r|l"`), contoh `"221|220"`. Setelah mengirimkan

masukan nilai radius *shaping*, maka bot akan menampilkan hasil prediksi dan klasifikasi radius *shaping*.



Gambar 15. Tampilan Bot Telegram

F. Hasil Perancangan Bot Telegram

Pengujian berdasarkan tabel 7 dilakukan dengan mengirimkan nilai radius *shaping* melalui bot telegram, nilai hasil prediksi dan klasifikasi di validasi dengan cara melakukan pengukuran radius dan nilai tekanan secara manual pada mesin, sehingga diketahui error antara prediksi bot telegram dengan aktual pengukuran manual.

Tabel 7. Data pengujian Bot Telegram

No.	Radius (mm)		Prediksi Bot Telegram			Aktual Pengukuran			Error (%)
	Kanan (r)	Kiri (l)	Tekanan (kg/cm ²)	Radius (mm)	Label (0, 1, 2)	Tekanan (kg/cm ²)	Radius (mm)	Label (0, 1, 2)	
1	225	226	1.0	226	CB	1.00	226	CB	0.0%
2	223	225	1.0	224	CB	1.00	224	CB	0.2%
3	220	221	0.90	221	CB	0.90	221	CB	0.9%
4	218	217	0.90	218	OK	0.90	218	OK	0.0%
5	213	210	0.79	211	OK	0.79	211	OK	0.2%
6	208	210	0.79	209	OK	0.79	209	OK	0.2%
7	207	207	0.80	207	CIL	0.80	207	CIL	0.7%
8	202	205	0.78	204	CIL	0.78	204	CIL	1.0%

Berdasarkan hasil pengujian pada tabel 7 didapat error terbesar yaitu 1,0%, hal tersebut masih dalam batas toleransi pengukuran yang ditentukan.

V. KESIMPULAN

Kesimpulan yang diperoleh berdasarkan hasil pengujian terbukti bahwa pembuatan purwarupa pemantauan radius *shaping* dengan menggunakan nodemcu dan sensor ultrasonik sebagai masukan dapat menghasilkan nilai pengukuran radius *shaping* dengan rentang pengukuran 150 mm sampai dengan 300 mm, dimana nilai tersebut dikirim ke web server oleh nodemcu dan nilai tersebut dapat diproses oleh server expressjs sehingga menghasilkan aplikasi web yang dapat diakses oleh publik. Model DNN dengan prediksi yang optimal dibandingkan dengan yang lain pada proses pelatihan Model ke-17 dimana

memiliki akurasi 99.06% dan loss 0.16%. Kemudian untuk klasifikasi yang optimal dibandingkan dengan yang lain pada proses pelatihan Model ke-9 dimana memiliki akurasi 99.86% dan loss 0.38%. Model tersebut kemudian digunakan pada web server ExpressJS untuk diintegrasikan dengan bot telegram.

DAFTAR PUSTAKA

- [1] Fadly, M., Nur Uddin, and H. Y. Sutarto. 2002. "Flutter Suppression Using Neural Networks : Design and Implementation." (January 2017).
- [2] Hadioanto, Nur, Hafifah Bella Novitasari, and Ami Rahmawati. 2019. "Klasifikasi Pemijinan Nasabah Bank Menggunakan Metode Neural Network." *Jurnal Pilar Nusa Mandiri* 15(2):163–70. doi: 10.33480/pilar.v15i2.658.
- [3] Herianto, and Kuku Pradityo. 2015. "Sistem Chatbot Untuk Membantu Diagnosa Kerusakan Sistem Komputer." *Jurnal Sains & Teknologi Fakultas Teknik Univ. Darma Persada* V(2).
- [4] Lee, R. S. 1968. "Earthmover Tires: After Design, Then What?" *SAE Technical Papers*. doi: 10.4271/680247.
- [5] Nufusula, Raga, and Ajib Susanto. 2018. "Rancang Bangun Chat Bot Pada Server Pulsa Menggunakan Telegram Bot API." *JOINS (Journal of Information System)* 3(1):80–88.
- [6] On, Notification, and T. H. E. Line. 2019. "Kata Kunci: Infus, NodeMCU, CloudMQTT, Heroku, Line." 6(2):4442–47.
- [7] Risanty, Rita Dewi, and Ade Sopiyan. 2017. "Pembuatan Aplikasi Kuesioner Evaluasi Belajar Mengajar Menggunakan Bot Telegram Pada Fakultas Teknik Universitas Muhammadiyah Jakarta (Ft-Umj) Dengan Metode Polling." *Seminar Nasional Sains Dan Teknologi* (November):1–9.
- [8] Rohmawan, Eko Prasetyo. 2018. "Menggunakan Metode Desicion Tree." *Jurnal Ilmiah MATRIK Vol.20 No.1, April 2018:21–30* 21–30.
- [9] Sumathy, B., D. Abinash, D. Aravindan, K. Prasaanth, and D. Thanga Muralidharan. 2019. "Automation of Tire Curing Machine Using IOT." *Indian Journal of Science and Technology* 12(29):1–7. doi: 10.17485/ijst/2019/v12i29/146978.
- [10] Windarto, Agus Perdana. 2017. "Implementasi JST Dalam Menentukan Kelayakan Nasabah Pinjaman KUR Pada Bank Mandiri Mikro Serbelawan Dengan Metode Backpropogation." *J-SAKTI (Jurnal Sains Komputer Dan Informatika)* 1(1):12. doi: 10.30645/j-sakti.v1i1.25.
- [11] Yuniarto, Tri, and Chalilullah Rangkuti. 2019. "Analisa Jenis Insulasi Terhadap Pipa Penghantar Uap Panas Untuk Mesin Pemasa Ban Motor Dengan Kapasitas 2000 Ban Per Hari Pada Pt. Xyz Di Tangerang." *Prosiding Seminar Nasional Cendekiawan* 1–23.