

Sistem Deteksi Anomali pada Proses Radius Shaping Menggunakan Deep Neural Network Berbasis Bidirectional Protocol

Zendi Iklima^{1*}, Trie Maya Kadarina¹

¹Teknik Elektro, Universitas Mercu Buana, Jakarta

*zendi.iklima@mercubuana.ac.id

Abstrak—Proses shaping merupakan langkah dalam proses curing atau pemasakan ban setengah jadi (green tire), di mana ban tersebut ditempatkan pada bladder dan diberi tekanan yang sesuai. Dalam proses shaping, nilai tekanan dan dimensi (radius) bladder menjadi parameter utama yang memengaruhi hasil produk. Mesin curing yang digunakan saat ini masih mengadopsi sistem konvensional, memerlukan pemantauan konstan oleh operator untuk mencapai hasil shaping yang optimal dan mengurangi kegagalan dalam proses curing ban. Pantauan mesin curing yang dilakukan secara manual menyebabkan analisis cacat produk tidak optimal, dan penentuan nilai radius yang optimal sulit dilakukan. Oleh karena itu, diperlukan suatu metode agar sistem pemantauan nilai radius shaping dapat mendeteksi dan memberikan rekomendasi untuk parameter shaping yang tepat, seperti nilai radius dan tekanan. Dengan perkembangan teknologi kecerdasan buatan yang pesat di berbagai sektor, termasuk medis, militer, pendidikan, ekonomi, politik, industri, dan lainnya, teknologi ini dapat mengatasi tugas-tugas berbahaya dan berulang. Sejumlah metode AI telah berhasil diimplementasikan dalam industri untuk memastikan produksi yang memenuhi standar kualitas. Dalam konteks ini, penulis mengusulkan penerapan metode AI untuk melakukan analisis terhadap proses radius shaping pada pemasakan ban, dengan tujuan mengurangi kegagalan produksi yang sering disebabkan oleh berbagai faktor, salah satunya adalah kesalahan manusia. Penelitian ini mencakup pengidentifikasian pola atau perubahan yang tidak sesuai dengan kondisi normal yang diharapkan. Merujuk pada metode komunikasi yang memungkinkan aliran informasi dua arah, memungkinkan respons yang lebih cepat dan interaksi yang dinamis antara sistem deteksi anomali dengan sistem pembelajaran mandiri.

Kata Kunci— *Bidirectional Protocol, Deep Neural Network, Deteksi Anomali, Radius Shaping, Sistem Pembelajaran Mandiri*

DOI: 10.22441/jte.2025.v16i2.006

I. PENDAHULUAN

Curing atau vulkanisasi adalah tahap krusial dalam pembuatan ban, di mana sifat senyawa ban berubah dari bahan karet plastik menjadi elastis melalui pembentukan ikatan silang dalam struktur molekulnya. Proses dimulai dengan green tire / setengah jadi, ditempatkan pada mold bawah, sementara bagian dalamnya mengelilingi bladder (pembentuk ban bagian dalam). Selanjutnya, mold atas menutup untuk melakukan proses curing dan shaping yang melibatkan pemberian tekanan proporsional pada green tire [1]–[3]. Tahapan ini menghasilkan ban akhir yang memiliki kualitas elastis yang diinginkan. Data dan

observasi lapangan menunjukkan bahwa sebanyak 70% cacat produk dalam proses curing disebabkan oleh ketidaksesuaian dalam proses shaping. Konfigurasi nilai tekanan dan dimensi bladder dimaksudkan untuk menghasilkan produk berkualitas, namun masalah muncul karena parameter yang telah dikonfigurasi sering berubah sesuai dengan kondisi mesin. Kondisi yang fluktuatif ini dapat merugikan, mengakibatkan proses shaping yang tidak optimal dan mengganggu analisis cacat produk pada mesin curing. Oleh karena itu, diperlukan sistem pendukung untuk menjaga stabilitas mesin curing dan memastikan konfigurasi parameter radius shaping tetap sesuai. Konfigurasi dapat dikelola melalui berbagai cara, salah satunya dengan menggunakan *Optical Distance Sensor* O1D100. Sensor ini memungkinkan pengendalian parameter yang telah dikonfigurasi oleh kontroler sesuai pembacaan sensor tersebut [4]. Sistem ini dapat ditingkatkan menjadi otomatis dan dapat dimonitor melalui *Internet of Things* [5]. dan Jaringan Syaraf Tiruan [6]. Metode Jaringan Syaraf Tiruan menggunakan proses kalkulasi *backpropagation* untuk menyederhanakan masalah kompleks terkait dengan input, prediksi, pengenalan pola, dan lainnya. Penelitian ini mengembangkan hasil penelitian sebelumnya dimana telah diimplementasikan model Jaringan Syaraf Tiruan untuk prediksi dan klasifikasi dalam proses vulkanisasi green tire. Data dicapture pada sistem deteksi dan klasifikasi akan digunakan untuk model Jaringan Syaraf Tiruan dilatih pada data baru dan pada jenis ban yang lain. Sehingga model Jaringan Syaraf Tiruan semakin efektif dan efisien dalam menentukan konfigurasi parameter radius shaping pada jenis ban yang berbeda. Model Jaringan Syaraf Tiruan akan melakukan pembelajaran mandiri berdasarkan hasil konfigurasi parameter radius shaping pada jenis ban yang merupakan luaran dari model Jaringan Syaraf Tiruan tersebut.

II. PENELITIAN TERKAIT

Penelitian terhadap proses curing pada cerobong tembakau curah memberikan kontribusi penting dalam mengembangkan sistem curing terintegrasi *deep learning*. TobaccoNet, berbasis *Convolutional Neural Network* (CNN), berhasil mengklasifikasikan *dry-bulb temperature* dan *wet-bulb temperature* dengan akurasi luar biasa, mencapai 99.65% dan 96.83% masing-masing. Keberhasilan ini menegaskan bahwa TobaccoNet efektif dan dapat diandalkan dalam memodelkan proses curing tembakau massal secara cerdas. Analisis struktur CNN menunjukkan bahwa struktur sekuensial adalah pilihan

optimal, mempertimbangkan kompleksitas komputasi dan kinerja prediksi dalam analisis curing tembakau massal [3].

Dalam upaya meningkatkan teknologi kontrol curing tembakau, terdapat tantangan terkait intensitas tenaga kerja dan kesulitan dalam menyesuaikan diri dengan kondisi batch daun tembakau yang bervariasi, yang dapat merugikan kualitas tembakau. Salah satu pendekatan yang menjanjikan adalah menggunakan model prediksi keadaan proses pengawetan tembakau. Model ini memanfaatkan fitur input seperti area, warna, berat, dan perubahan zat kimia daun tembakau selama proses pengawetan. Contoh konkrit dari solusi ini adalah *State Prediction Fusion Model* (SPFM) yang dikembangkan oleh Wang Y, yang mencapai akurasi 97.4% selama proses pelatihan. Keberhasilan ini menunjukkan bahwa SPFM mampu efektif mengendalikan proses curing tembakau dengan memadukan informasi dari berbagai fitur dan memastikan penyesuaian tepat waktu [2].

Physics-Informed Neural Network (PINN) untuk mensimulasikan evolusi termokimia dari material komposit pada alat yang menjalani proses curing dalam autoklaf. kerangka kerja PINN untuk pemodelan perpindahan panas eksotermik dalam sistem alat komposit yang menjalani siklus penyembuhan penuh. PINN yang diusulkan adalah jaringan saraf umpan-maju yang menangani solusi dari dua persamaan diferensial yang digabungkan, yaitu, perpindahan panas eksotermik dan reaksi resin yang mengatur distribusi dan evolusi suhu dan tingkat penyembuhan dalam material komposit dan alat. Arsitektur jaringan yang terputus-putus digunakan untuk memanfaatkan parameter jaringan independen untuk variabel bidang keluaran, yaitu suhu dan derajat penyembuhan. Pendekatan berurutan untuk pelatihan PINN terputus-putus yang diusulkan disajikan yang mengatasi ketidakstabilan dalam pelatihan PINN. Dalam proses pelatihan, parameter jaringan dibatasi melalui pengenalan fungsi kerugian berbasis fisika. Pembelajaran transfer menunjukkan peningkatan pelatihan PINN, dan menunjukkan perluasannya ke pengaturan pemodelan pengganti dengan memasukkan koefisien perpindahan panas sebagai parameter masukan [7].

Sistem kontrol suhu cerdas berbasis pembelajaran online diusulkan untuk meningkatkan homogenitas proses pengeringan gelombang mikro pada bahan komposit canggi. *Microwave Control Strategy* (MCS) berbasis *Convolutional Neural Network* (CNN) dirancang dan diimplementasikan sebagai sistem kontrol. Dalam eksperimen, MCS ditunjukkan sebagai pilihan yang layak melalui perbandingan perbedaan suhu dan kesalahan pelacakan dengan dua metode kontrol lainnya. Hasilnya menunjukkan penurunan 42,3% dalam perbedaan suhu rata-rata pada pelat komposit serat karbon/BMI ($500 \times 400 \times 2 \text{ mm}^3$) dibandingkan dengan metode kompensasi acak yang umum digunakan, serta penurunan 31,6% dibandingkan dengan metode kontrol neural semi-langsung yang baru-baru ini diusulkan. Metode ini juga menghasilkan kesalahan pelacakan rata-rata 87,4% lebih rendah daripada metode kontrol jaringan saraf semi-langsung. Peningkatan keseragaman suhu terutama disebabkan oleh penguatan kompensasi suhu aktif dengan memisahkan kontrol keseragaman dari kontrol daya gelombang mikro [8].

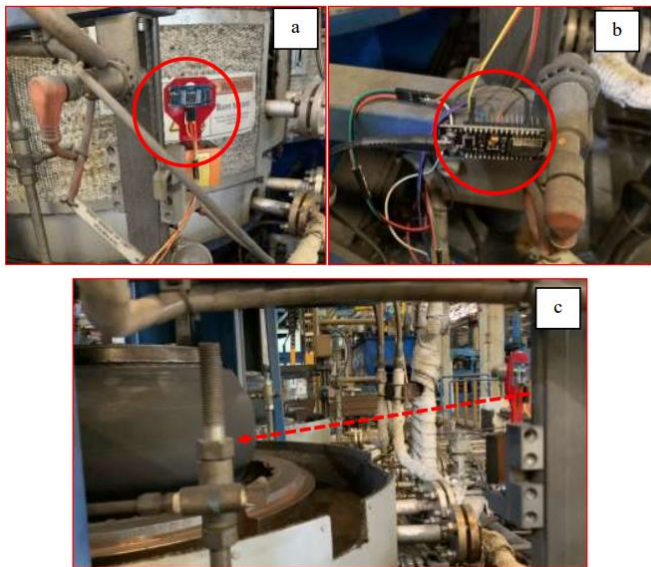
Peneliti mengimplementasikan model klasifikasi cacat pada proses pembentukan radius dalam proses *curing* ban menggunakan *Deep Neural Network* (DNN) yang disesuaikan dengan baik. Proses curing ban melibatkan tahap akhir pembuatan ban, di mana ban setengah jadi (*green tires*) divulkanisasi untuk membentuk produk akhir. Pembentukan radius yang tidak tepat selama proses curing menyebabkan 70% cacat produk. Makalah tersebut mengusulkan deteksi ketidaknormalan pembentukan radius menggunakan DNN yang disesuaikan dengan baik dan menjelajahi berbagai model DNN untuk mengoptimalkan proses deteksi. Arsitektur DNN melibatkan model prediktor untuk menghasilkan nilai tekanan dan radius, serta model klasifikasi untuk mengklasifikasikan cacat dalam proses curing. Dataset dikumpulkan menggunakan sensor ultrasonik dan mikrokontroler ESP8266 yang terpasang pada mesin curing. Model DNN dilatih dan divalidasi, dan performanya dievaluasi. Model yang diusulkan mencapai akurasi pelatihan sebesar 97,88%, akurasi validasi sebesar 95%, dan akurasi pengujian sebesar 100%, dengan kerugian sebesar 4,93%. Matriks kebingungan menunjukkan presisi, recall, dan nilai F1 yang tinggi untuk label cacat, menandakan kemampuan model untuk memprediksi dan mengklasifikasikan cacat dengan akurasi yang luar biasa.

Berdasarkan beberapa literatur diatas, penulis mengembangkan system deteksi kegagalan radius shaping pada proses *curing* ban. Dimana terdapat dua parameter yang berhasil diambil dari mesin *curing* ban yaitu radius dan tekanan. Analisa model tersebut merupakan model DNN yang paling optimal dalam mendeteksi kesalahan konfigurasi parameter radius *shaping* pada proses *curing* ban. Adapun kontribusi pada penelitian ini adalah mengembangkan system pembelajaran mandiri dalam model deteksi dan klasifikasi yang dapat menjadi rekomendasi konfigurasi parameter radius shaping pada proses vulkanisasi ban setengah jadi (*green tires*).

III. METODOLOGI PENELITIAN

Fokus utama dari rancangan ini adalah menciptakan sistem yang mampu meramalkan parameter pembentukan radius dengan memanfaatkan data dari sensor ultrasonik. Dalam konteks ini, model DNN digunakan untuk menetapkan konfigurasi nilai tekanan dan pembentukan radius. Pemilihan model ini didasarkan pada pilihan arsitektur DNN yang menunjukkan performa optimal, sebagaimana dibuktikan melalui hasil pelatihan, validasi, dan pengujian.

Proses merancang model arsitektur DNN melibatkan tahapan penting, yaitu pengolahan data sebelum melatih dataset agar sesuai dengan format yang diusulkan. Tabel 2 memberikan contoh dataset pembentukan radius yang melibatkan input radius kiri (r) dan radius kanan (l), serta output tekanan pembentukan (p) dan pengaturan radius (s). Nilai-nilai ini kemudian mengalami proses normalisasi berurutan menjadi nr , nl , np , dan ns . Penting untuk dicatat bahwa normalisasi dilakukan dengan menghitung rata-rata dan standar deviasi dari setiap nilai input dan output.



Gambar 1. Peletakan Sensor pada Mesin Curing

Table 1. Format Dataset *Radius Shaping*

<i>r</i>	<i>l</i>	<i>p</i>	<i>s</i>
225	226	1.0	226
208	208	0.8	208
207	208	0.8	208
218	219	0.9	219
225	225	1.0	225
213	214	0.8	214
204	205	0.8	205
225	226	1.0	226
200	200	0.7	200

Tabel 1 menerangkan format dataset radius shaping yang diusulkan dimana dataset tadi diolah menggunakan menormalisasi setiap variable guna meminimalisir error pada tahap training, mengingat beberapa fungsi aktivasi dalam rentang 0~1. Sesuai Tabel 1 dan Gambar 1, 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.

Berdasarkan Gambar 2, dataset dapat dinyatakan sebagai pasangan masukan-keluaran pada rentang N sebagai $X = \{(\bar{x}_i, \bar{y}_i), \dots, (\bar{x}_N, \bar{y}_N)\}$, dimana nilai error pada proses backpropagation dapat dinyatakan sebagai berikut [9]–[14]:

$$E(X, \theta) = \frac{1}{2N} \sum_{i=1}^N (\hat{y}_i - y_i)^2 \quad (1)$$

Dimana $\hat{y}_i$ menyatakan nilai keluaran yang dikomputasi berdasarkan network pada masukan $\bar{x}_i \cdot y_i$ menyatakan nilai target berdasarkan pasangan masukan-keluaran $\bar{x}_i, \bar{y}_i$.

Fungsi partial derivative dinyatakan sebagai fungsi error E berdasarkan bobot arsitektur w_{ij}^k pada $1 \leq k < m$ dinyatakan sebagai [9]–[14]:

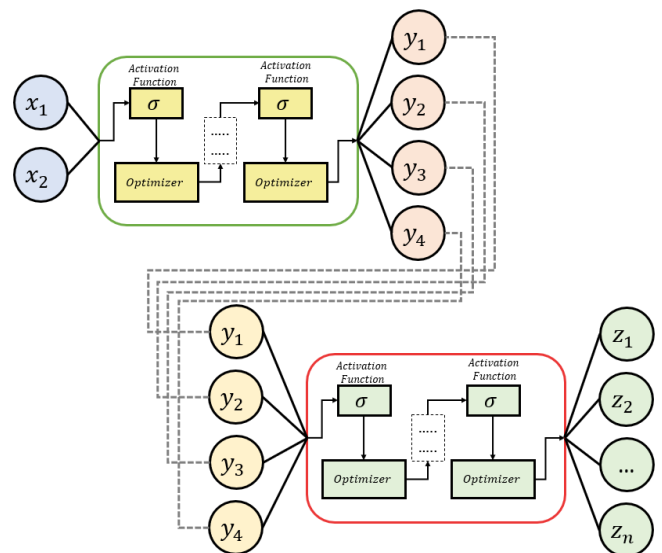
$$\frac{\partial E}{\partial w_{ij}^k} = g'(a_j^k) o_i^{k-1} \sum_{l=1}^{k+1} w_{jl}^{k+1} \delta_l^{k+1} = \delta_j^k o_i^{k-1} \quad (2)$$

Nilai error pada lapisan k dinyatakan sebagai δ_j^k

tergantung pada nilai error pada lapisan selanjutnya δ_l^{k+1} .

Sementara itu, partial derivative pada proses backpropagation dipengaruhi oleh nilai *learning rate* dan total *gradient* $\frac{\partial E(X, \theta)}{\partial w_{ij}^k}$, sehingga dinyatakan sebagai [9]–[14]:

$$\Delta w_{ij}^k = -\alpha \frac{\partial E(X, \theta)}{\partial w_{ij}^k} \quad (3)$$

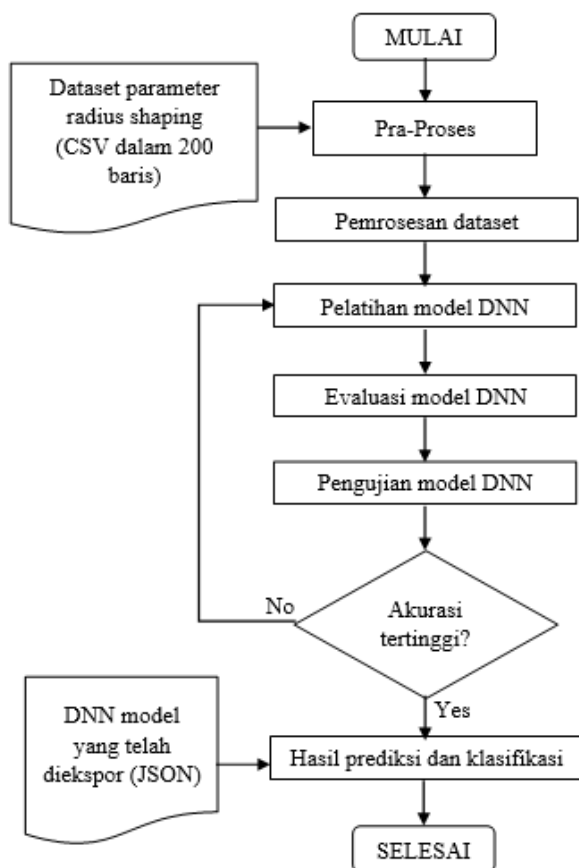


Gambar 2. Arsitektur DNN Detektor dan Klasifikator

Maka dari itu diperlukan beberapa parameter yg digunakan untuk analisa contoh DNN yang paling optimal terkait pemilihan fungsi aktivasi (*activation function*), *optimizer*, jumlah lapisan di hidden layer, jumlah neuron pada masing-masing lapisan di hidden layer serta *learning rate*. Tabel 3 membuktikan Hyperparameter untuk Arsitektur DNN yg diusulkan. Alur Kerja Pelatihan Model DNN membahas blok diagram sistem Jaringan Syaraf Tiruan. Blok diagram ini memiliki fungsi yaitu buat menjelaskan cara kerja sistem secara sederhana.

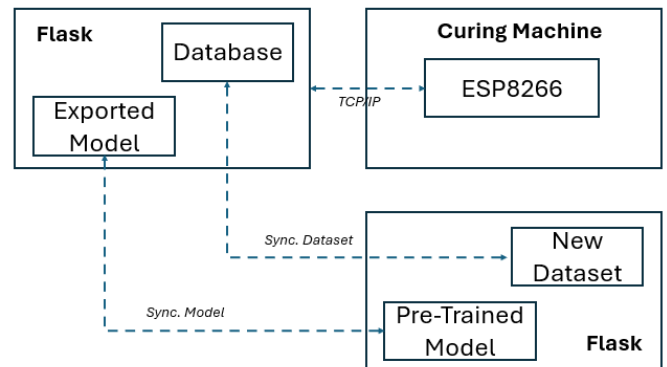
Gambar 3 secara sekuensial merepresentasikan diagram alir pelatihan model DNN dideskripsikan sebagai berikut:

- 1) Pengambilan dataset: Data pengukuran radius shaping sebesar 200 data dengan variabel radius (*bladder* kanan & kiri), radius optimal (pengaturan radius yang disarankan) dan klasifikasi.
- 2) Pra - proses: Melakukan indexing data parameter radius shaping sehingga terdiri berasal radius kanan, radius kiri, tekanan *radius shaping* optimal dan klasifikasi. Adapun label pada model klasifikasi ini yaitu *Defect Crown Bare* (CB), *Defect Crack Inner Linner* (CIL), dan Normal.
- 3) Pemrosesan dataset: merupakan proses penyesuaian dataset pada bentuk arsitektur DNN yang diinginkan berdasarkan jumlah neuron masukan serta keluaran.
- 4) Pelatihan model DNN: artinya proses pelatihan data pada model arsitektur DNN yang dipilih.
- 5) Evaluasi model DNN yang sudah dilatih: adalah proses evaluasi tingkat akurasi yang telah dihasilkan pada proses sebelumnya, dengan mengambil sampel data pada pengujian tingkat akurasi yang didapatkan tadi.
- 6) Pengujian model: Menguji hasil training model DNN buat memperoleh tanggapan yang diinginkan terhadap masukan yang diberikan.
- 7) Ekspor model: Mengekspor model arsitektur DNN yang sudah dilatih buat dapat diaplikasikan atau dipergunakan kembali di platform lain.



Gambar 3. Diagram Alir Pelatihan Model DNN

Penelitian ini berfokus pada perancangan arsitektur framework pembelajaran mandiri Model DNN sehingga dapat di representasikan pada Gambar 4.



Gambar 4. Arsitektur Framework Pembelajaran Mandiri Model DNN

Berdasarkan Gambar 4 secara sekuensial merepresentasikan Arsitektur Framework Pembelajaran Mandiri Model DNN dideskripsikan sebagai berikut:

- 1) Data dari mesin *curing* dibaca oleh sensor dan dikirimkan melalui protocol TCP/IP bidirectional.
- 2) Data disimpan kedalam database sebagai informasi baru terhadap hasil implementasi model DNN terhadap system deteksi dan klasifikasi parameter proses *radius shaping*.
- 3) DNN Model akan mengirimkan kembali informasi dari hasil deteksi dan klasifikasi ke ESP8266 sebagai feedback berupa parameter proses *radius shaping*.
- 4) System pembelajaran mandiri terjadwal pada setiap periode tertentu atau setting jumlah data baru yang tersimpan kedalam database.
- 5) *Flask Framework* menjalankan task untuk melakukan pelatihan model secara mandiri dengan melakukan sinkronisasi dataset ke database yang ada di web server *Flask Framework*.

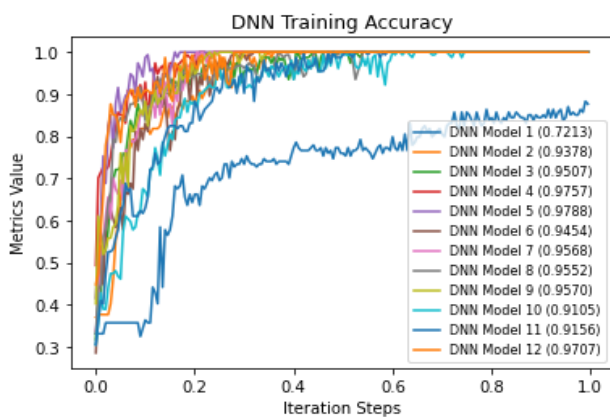
IV. HASIL DAN ANALISA

Model DNN dilakukan menggunakan Intel(R) Core i5-6300HQ CPU@2.30GHz, RAM 16GB, dan Nvidia GeForce GTX 960M 4GB VRAM. Model yang diusulkan dilatih menggunakan algoritma pengoptimal Adam selama 200 iterasi. Algoritma Adam merupakan salah satu metode pengoptimalan yang paling populer dalam pelatihan jaringan saraf, karena kemampuannya untuk menggabungkan keuntungan dari dua metode lain, yaitu AdaGrad dan RMSProp, untuk mencapai konvergensi yang cepat dan stabil. Selama proses pelatihan ini, model diujicobakan dengan berbagai konfigurasi untuk menentukan kombinasi parameter yang memberikan hasil terbaik.

Penelitian ini mengusulkan 12 kombinasi Model DNN yang berbeda arsitekturnya. Model 4, 5, dan 6 terbukti memiliki performa yang sangat baik, dengan akurasi pelatihan yang

mencapai lebih dari 97%. Tingkat akurasi ini menunjukkan bahwa model mampu mengenali pola dalam data pelatihan dengan sangat efektif. Namun, penting untuk diingat bahwa performa tinggi pada data pelatihan saja tidak cukup. Oleh karena itu, langkah selanjutnya adalah melakukan validasi model untuk memastikan bahwa model tidak hanya menghafal data pelatihan, tetapi juga memiliki kemampuan generalisasi yang baik. Validasi dilakukan dengan menggunakan dataset yang berbeda dari dataset pelatihan, untuk mengevaluasi apakah model dapat mengklasifikasikan data yang tidak pernah dilihat sebelumnya dengan baik.

Gambar 5 memberikan ringkasan dari keseluruhan proses pelatihan model yang diusulkan. Dalam gambar ini, berbagai metrik pelatihan, seperti akurasi dan kehilangan (loss), diilustrasikan untuk menunjukkan bagaimana model belajar seiring waktu. Grafik ini sangat penting untuk memahami dinamika pelatihan dan untuk mendeteksi potensi masalah seperti *overfitting* atau *underfitting*.

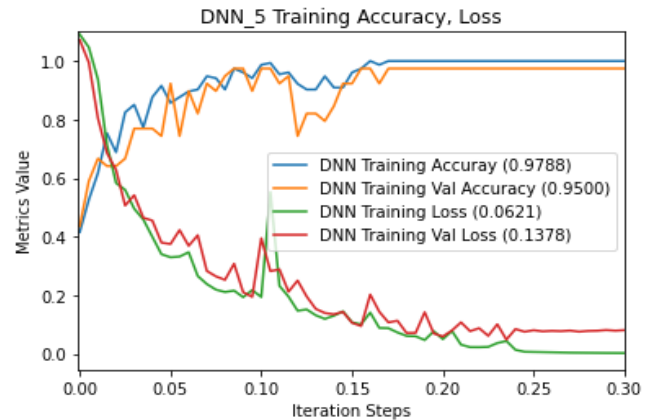


Gambar 6. Performa Pelatihan Model DNN

Gambar 5 fokus pada Model DNN (5), yang telah dikonfigurasi dengan delapan lapisan ($nL=8$), masing-masing dengan 32 neuron ($nN=32$), ukuran batch sebesar 4 ($bS=4$), dan menggunakan fungsi aktivasi "LeakyReLU". Konfigurasi ini dipilih berdasarkan eksperimen yang menunjukkan bahwa kombinasi tersebut memberikan performa terbaik dalam hal akurasi dan kecepatan konvergensi. Fungsi aktivasi "LeakyReLU" dipilih karena mampu mengatasi masalah "vanishing gradient" yang sering terjadi pada fungsi aktivasi tradisional seperti sigmoid atau tanh, sehingga memungkinkan model untuk belajar lebih efisien.

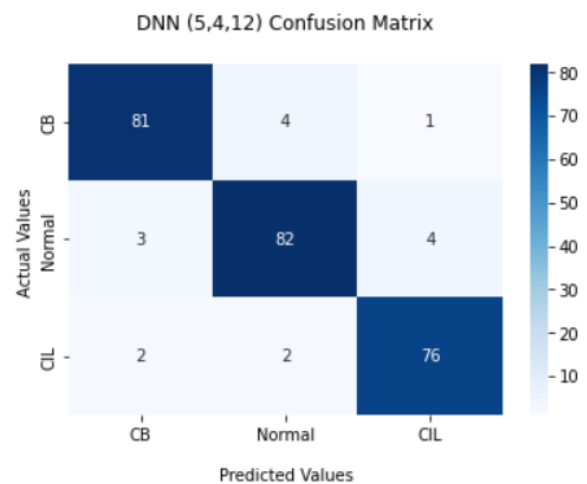
Model DNN ini menunjukkan kecocokan yang baik antara nilai pelatihan dan validasi, yang berarti bahwa model tidak hanya berkinerja baik pada data yang dilatih tetapi juga pada data yang divalidasi. Ini menunjukkan bahwa model memiliki distribusi data yang baik untuk pelatihan dan validasi, serta kemampuan generalisasi yang kuat. Hal ini sangat penting dalam konteks aplikasi nyata, dimana model harus mampu bekerja dengan baik pada data baru yang mungkin memiliki distribusi yang berbeda dari data pelatihan. Model DNN ini juga telah terbukti mampu memprediksi dan mengklasifikasikan cacat dalam proses pembentukan radius dengan baik ketika diuji pada distribusi data uji. Ini menunjukkan bahwa model yang dilatih tidak hanya efektif dalam fase pelatihan dan validasi, tetapi juga mampu memberikan performa yang konsisten dalam situasi praktis, di

mana data baru dan tidak terduga dapat muncul. Keberhasilan ini memberikan keyakinan bahwa model yang diusulkan siap untuk diterapkan dalam lingkungan nyata untuk mengidentifikasi dan mengklasifikasikan cacat proses pembentukan radius. Gambar 6 merepresentasikan nilai akurasi, loss, validasi akurasi, dan validasi loss dari model DNN (5).



Gambar 6. Model DNN (5) Kinerja Pelatihan

Gambar 7, menunjukan *Confusion Matrix* Model DNN (5). Matriks konfusi untuk Model DNN (5) memberikan metrik evaluasi yang mendalam seperti presisi, recall, dan F1-score. Khusus untuk Label CB, model ini menunjukkan performa yang sangat baik dengan presisi 94%, recall 94%, dan F1-score 94%. Ini menunjukkan bahwa model mampu mengidentifikasi Label CB dengan tingkat akurasi yang tinggi, mengurangi jumlah kesalahan positif dan negatif.



Gambar 7. *Confusion Matrix* Model DNN (5).

Label CIL, model menunjukkan presisi 94%, recall 95%, dan F1-score 94%. Angka recall yang lebih tinggi (95%) menunjukkan bahwa model ini sangat efektif dalam mendeteksi sebagian besar contoh CIL dalam dataset, dengan sedikit kesalahan negatif. Sementara itu, Label Normal memiliki presisi 93%, recall 92%, dan F1-score 93%. Meskipun sedikit lebih rendah dibandingkan dengan Label CB dan CIL, performa untuk Label Normal masih sangat baik, menunjukkan bahwa model mampu membedakan data normal dari label lainnya

dengan cukup akurat. Secara keseluruhan, metrik ini menunjukkan bahwa Model DNN (5) memiliki kemampuan yang kuat dalam mengklasifikasikan berbagai label dengan akurasi tinggi, baik dalam hal presisi, recall, maupun F1-score. Hal ini mencerminkan kemampuan model untuk menangani ketidakseimbangan data dan memastikan bahwa setiap kategori diidentifikasi dengan benar.

Tabel 2 menunjukkan response system terintegrasi antara ESP8266 dengan *Flask Framework* untuk dapat mengklasifikasikan hasil pembacaan alat yang terdapat pada mesin curing.

Tabel 2. Respon Sistem Integrasi

<i>Req. Num</i>	<i>Data Size (Byte)</i>	<i>Response (ms)</i>	<i>Pkg. Err. / Loss (%)</i>
1	123.56	22.56	0
10	301.12	166.78	0
50	489.01	445.10	0
100	698.75	1101.98	2.19
200	888.14	2489.19	6.98

Sistem terintegrasi ESP8266 dengan *Flask Framework* untuk mengirimkan data menunjukkan performa yang bervariasi berdasarkan banyaknya permintaan (*Req. Num*) dan

ukuran data (*Data Size*). Berdasarkan tabel yang disediakan, kita dapat melihat bahwa untuk jumlah permintaan sebesar 1 permintaan, data yang dikirimkan sebesar 123.56 Byte dengan waktu respons 22.56 ms dan tanpa kehilangan paket (0%). Ini menunjukkan bahwa sistem sangat efisien dan cepat untuk permintaan kecil. Ketika jumlah permintaan meningkat menjadi 10 permintaan, ukuran data yang dikirim bertambah menjadi 301.12 Byte, dan waktu respons juga meningkat menjadi 166.78 ms, tetapi tetap tanpa kehilangan paket. Ini menunjukkan bahwa sistem masih efisien untuk ukuran data yang lebih besar.

Pada jumlah permintaan 50 permintaan, data yang dikirimkan adalah 489.01 Byte dengan waktu respons 445.10 ms, dan tidak ada kehilangan paket yang terdeteksi. Namun, saat jumlah permintaan meningkat menjadi 100 permintaan, data yang dikirim menjadi 698.75 Byte dengan waktu respons 1101.98 ms, dan mulai muncul kehilangan paket sebesar 2.19%. Ini menunjukkan bahwa sistem mulai mengalami degradasi performa dengan peningkatan ukuran data. Akhirnya, pada jumlah permintaan 200 permintaan, data yang dikirim mencapai 888.14 Byte, dengan waktu respons meningkat tajam menjadi 2489.19 ms, dan tingkat kehilangan paket sebesar 6.98%. Ini menandakan bahwa dengan ukuran permintaan dan

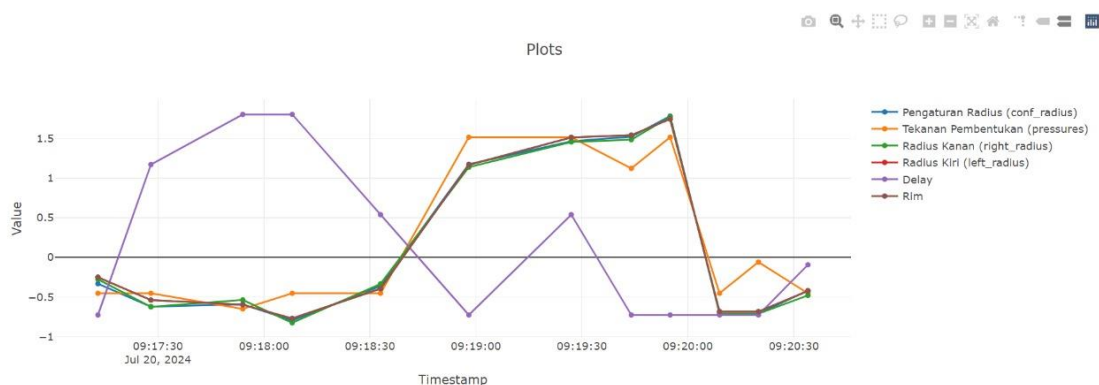
Sensor datas

Pengaturan Radius (conf_radius)	Tekanan Pembentukan (pressures)	Radius Kanan (right_radius)	Radius Kiri (left_radius)	Delay	Rim	Status	Action
-0.3324	-0.4534	-0.2775	-0.2504	-0.7268	-1.2685	Defect Crown Bare	Delete
-0.6221	-0.4534	-0.6241	-0.5395	1.1711	0.0200	Normal	Delete
-0.5931	-0.6503	-0.5375	-0.5973	1.8037	-0.4095	Normal	Delete
-0.7959	-0.4534	-0.8263	-0.7707	1.8037	-0.4095	Defect Crack Inner Linner	Delete
-0.3614	-0.4534	-0.3353	-0.3950	0.5385	-1.2685	Defect Crown Bare	Delete
1.1740	1.5157	1.1377	1.1658	-0.7268	1.7379	Defect Crack Inner Linner	Delete
1.4637	1.5157	1.4554	1.5127	0.5385	1.3085	Normal	Delete
1.5216	1.1219	1.4842	1.5416	-0.7268	1.3085	Normal	Delete
1.7824	1.5157	1.7730	1.7439	-0.7268	1.3085	Defect Crown Bare	Delete
-0.7090	-0.4534	-0.7107	-0.6840	-0.7268	-0.4095	Defect Crack Inner Linner	Delete
-0.7090	-0.0595	-0.7107	-0.6840	-0.7268	-0.4095	Defect Crack Inner Linner	Delete
-0.4193	-0.4534	-0.4797	-0.4239	-0.0942	0.0200	Normal	Delete

VLCover Data

Gambar 8. Tampilan Histori Penyimpanan Pembacaan

Plot



Gambar 9. Tampilan Grafik Histori Penyimpanan Pembacaan Parameter Mesin Curing

data yang besar, performa sistem menurun secara signifikan, dengan lebih banyak paket yang hilang.

Secara keseluruhan, sistem terintegrasi ini menunjukkan bahwa ESP8266 dengan *Flask Framework* mampu menangani permintaan dan pengiriman data kecil hingga sedang dengan sangat baik, menjaga waktu respons yang cepat dan tingkat kehilangan paket yang rendah. Namun, dengan peningkatan ukuran permintaan dan data, waktu respons meningkat secara signifikan dan tingkat kehilangan paket juga naik. Hal ini mengindikasikan bahwa untuk aplikasi yang memerlukan pengiriman data besar, perlu dilakukan optimalisasi lebih lanjut untuk menjaga kinerja yang efisien dan meminimalkan kehilangan paket. Optimalisasi ini bisa berupa peningkatan kapasitas jaringan, penggunaan algoritma kompresi data, atau pembagian data menjadi paket yang lebih kecil untuk dikirim secara bertahap. Gambar 8 dan Gambar 9 merepresentasikan hasil perancangan system monitoring pembacaan parameter Tekanan Pembentukan (pressures), Radius Kanan (right_radius), Radius Kiri (left_radius), Delay, Rim dari mesin curing melalui ESP8266 dikirimkan ke *Web Server Flask Framework*. Kemudian bagian status menunjukan hasil prediksi dari model (5) yang di export / pre-trained untuk mengklasifikasikan adanya kecacatan pada produksi ban setengah jadi dalam 3 klasifikasi yaitu CB, CIL, dan Normal.

V. KESIMPULAN

Evaluasi Model DNN (5) dan sistem terintegrasi ESP8266 dengan *Flask Framework* menunjukkan hasil yang signifikan dalam berbagai aspek. Matriks konfusi untuk Model DNN (5) memberikan metrik evaluasi yang mendalam, seperti presisi, recall, dan F1-score. Model ini menunjukkan performa yang sangat baik untuk Label CB dengan presisi, recall, dan F1-score masing-masing sebesar 94%. Hal ini menandakan bahwa model mampu mengidentifikasi Label CB dengan akurasi tinggi, mengurangi jumlah kesalahan positif dan negatif. Untuk Label CIL, model menunjukkan presisi 94%, recall 95%, dan F1-score 94%, yang mengindikasikan efektivitas model dalam mendeteksi contoh CIL dengan sedikit kesalahan negatif. Label Normal memiliki presisi 93%, recall 92%, dan F1-score 93%, menunjukkan bahwa model juga dapat membedakan data normal dengan cukup akurat. Secara keseluruhan, Model DNN (5) memiliki kemampuan kuat dalam mengklasifikasikan berbagai label dengan akurasi tinggi, baik dalam hal presisi, recall, maupun F1-score, mencerminkan kemampuannya untuk menangani ketidakseimbangan data dan mengidentifikasi setiap kategori dengan benar.

Pada sisi lain, sistem terintegrasi ESP8266 dengan *Flask Framework* menunjukkan performa yang bervariasi berdasarkan jumlah permintaan dan ukuran data. Untuk permintaan kecil (1 permintaan), sistem mampu mengirimkan data sebesar 123.56 Byte dengan waktu respons 22.56 ms dan tanpa kehilangan paket, menunjukkan efisiensi yang tinggi. Namun, dengan meningkatnya jumlah permintaan menjadi 100, data yang dikirim mencapai 698.75 Byte dengan waktu respons 1101.98 ms dan tingkat kehilangan paket 2.19%,

mengindikasikan degradasi performa. Pada 200 permintaan, data yang dikirim mencapai 888.14 Byte dengan waktu respons 2489.19 ms dan tingkat kehilangan paket sebesar 6.98%. Ini menunjukkan bahwa dengan ukuran permintaan dan data yang besar, performa sistem menurun signifikan, dengan lebih banyak paket yang hilang. Secara keseluruhan, sistem terintegrasi ini efisien untuk permintaan kecil hingga sedang, tetapi memerlukan optimalisasi lebih lanjut untuk menangani permintaan dan data yang lebih besar secara efisien.

DAFTAR PUSTAKA

- [1] J. Wu and S. X. Yang, "Intelligent Control of Bulk Tobacco Curing Schedule Using LS-SVM- and ANFIS-Based Multi-Sensor Data Fusion Approaches," *Sensors*, vol. 19, p. 1778, 2019, doi: <https://doi.org/10.3390/s19081778>.
- [2] Y. Wang and L. Qin, "Research on state prediction method of tobacco curing process based on model fusion," *J. Ambient Intell. Humaniz. Comput.*, no. April, 2021, doi: 10.1007/s12652-021-03129-5.
- [3] J. Wu and S. X. Yang, "Modeling of the Bulk Tobacco Flue-Curing Process Using a Deep Learning-Based Method," *IEEE Access*, vol. 9, pp. 140424–140436, 2021, doi: 10.1109/ACCESS.2021.3119544.
- [4] B. D. SUBIANTORO, *PENINGKATAN KINERJA PROGRAMMABLE LOGIC CONTROL MITSUBISHI SERI Q MENGGUNAKAN OPTICAL DISTANCE SENSOR O1D100 BERBASIS SISTEM KONTROL SHAPPING*. 2019.
- [5] B. Sumathy, D. Abinash, D. Aravindan, K. Prasaanth, and D. T. Muralidharan, "Automation of Tire Curing Machine Using IOT," *Indian J. Sci. Technol.*, vol. 12, no. 29, pp. 1–7, 2019, doi: 10.17485/ijst/2019/v12i29/146978.
- [6] Z. Iklima, B. N. Rohman, R. Muwardi, A. Khan, and Z. Arifiansyah, "DEFECT CLASSIFICATION OF RADIUS SHAPING IN THE TIRE CURING PROCESS USING FINE-TUNED DEEP NEURAL NETWORK," vol. xx, no. x, pp. 1–7.
- [7] S. Amini Niaki, E. Haghighat, T. Campbell, A. Poursartip, and R. Vaziri, "Physics-informed neural network for modelling the thermochemical curing process of composite-tool systems during manufacture," *Comput. Methods Appl. Mech. Eng.*, vol. 384, p. 113959, 2021, doi: 10.1016/j.cma.2021.113959.
- [8] J. Zhou, Y. Li, D. Li, and Y. Wen, "Online learning based intelligent temperature control during polymer composites microwave curing process," *Chem. Eng. J.*, vol. 370, pp. 455–465, 2019, doi: 10.1016/j.cej.2019.03.204.
- [9] M. A. Nielsen, "Neural Networks and Deep Learning," Determination Press, 2015. Accessed: Jan. 18, 2022. [Online]. Available: <http://neuralnetworksanddeeplearning.com>
- [10] John McGonagle, George Shaikouski, Christopher Williams, Andrew Hsu, Jimin Khim, and Aaron Miller, "Backpropagation | Brilliant Math & Science," Jan. 20, 2020. <https://brilliant.org/wiki/backpropagation/> (accessed Jan. 20, 2022).
- [11] I. Zendi, "A Microservices-based for Distributed Deep Neural Network of Delta Robot Control System," *2020 IEEE Int. Conf. Commun. Networks Satell. Commun. 2020 - Proc.*, pp. 218–221, 2020, doi: 10.1109/Commnetsat50391.2020.9328936.
- [12] I. Zendi, A. Zody, M. Imam, A. Zody, and A. Khan, "SELF-LEARNING OF DELTA ROBOT USING INVERSE KINEMATICS AND ARTIFICIAL NEURAL NETWORKS," *SINERGI*, vol. xx, no. x, pp. 1–7, 2021.
- [13] S. Teerapittayanon, B. McDanel, and H. T. Kung, "Distributed Deep Neural Networks over the Cloud, the Edge and End Devices," *Proc. - Int. Conf. Distrib. Comput. Syst.*, pp. 328–339, 2017, doi: 10.1109/ICDCS.2017.226.
- [14] Z. Zhang, "Improved Adam Optimizer for Deep Neural Networks," *2018 IEEE/ACM 26th Int. Symp. Qual. Serv. IWQoS 2018*, pp. 1–2, 2019, doi: 10.1109/IWQoS.2018.8624183.