PROSIDING SEMINAR NASIONAL SANATA DHARMA BERBAGI (USDB)
"70 Tahun Sanata Dharma Menggali Makna dan Membangun Harapan:
Paradigma Keberlanjutan dan Tantangan Perubahan Iklim"
https://e-conf.usd.ac.id/index.php/usdb
ISSN: 3063-556X | Vol 3, 2025

PENGENALAN NADA PIANIKA MENGGUNAKAN *CONVOLUTIONAL NEURAL NETWORK*

Yahya Hanadi Arimatea[1*], Linggo Sumarno[2]
Universitas Sanata Dharma, Indonesia
Universitas Sanata Dharma, Indonesia
yahyaarimatea03@gmail.com, dan lingsum@usd.ac.id
*korespondensi: yahyaarimatea03@gmail.com

Abstrak
Pianika merupakan alat musik tiup kecil menggunakan bilah-bilah *keyboard* yang mencakup tiga oktaf. Dalam perkembangan teknologi, pengenalan nada pianika semakin banyak menggunakan jaringan saraf tiruan salah satunya *Convolutional Neural Network*. Penelitian ini bertujuan untuk merancang sistem pengenalan nada alat musik pianika dengan minimal *input* namun tetap optimal menggunakan metode *Convolutional Neural Network* satu dimensi (CNN 1D), serta dilengkapi antarmuka pengguna (GUI) untuk memudahkan pengguna seperti siswa dan guru dalam pembelajaran musik. Metode yang digunakan mencakup tahap *pre-processing* sinyal suara yang kemudian diolah oleh CNN satu dimensi. Sistem dilatih menggunakan data audio hasil rekaman dalam *file wav* dengan delapan nada dasar: C, D, E, F, G, A, B, dan C'. sebanyak 30 sampel per nada, dengan 20 data pelatihan dan 10 data untuk pengujian menggunakan model yang dibangun di lingkungan *Python* dengan *library TensorFlow*. Hasil penelitian menunjukkan sistem mampu bekerja secara efisien dan akurat pada kondisi optimal dengan parameter minimum, yaitu *input flatten* sebanyak 16, *epoch* sebanyak 30, jumlah *neuron* pada *fully connected layer* sebanyak 16, dan *batch size* sebesar 2. Dengan konfigurasi ini, sistem berhasil mencapai tingkat akurasi pengenalan nada *real-time* sebesar 91,25%. Hal ini memperlihatkan bahwa CNN 1D dapat diterapkan secara minimum tetapi optimal untuk mendeteksi nada pianika.
Kata Kunci: *Convolutional Neural Network*, Pengenalan Nada, Pianika

RECOGNITION OF PIANICA NOTES USING CONVOLUTIONAL NEURAL NETWORK

Yahya Hanadi Arimatea[1*], Linggo Sumarno[2]
Sanata Dharma University, Indonesia
Sanata Dharma University, Indonesia
yahyaarimatea03@gmail.com, dan lingsum@usd.ac.id
*correspondence: yahyaarimatea03@gmail.com

Abstract
Pianica is a small wind instrument with keyboard keys that cover three octaves. With the technological development, notes recognition for the pianica can now use artificial neural networks, such as the Convolutional Neural Network (CNN). This research aimed to design a notes recognition system using a one-dimensional CNN (1D CNN) with minimal input while still maintaining good performance. A graphical user interface (GUI) was also added to assist users such as students and teachers in music learning. The method included preprocessing the sound signal, which was then processed by the 1D CNN. The system was trained using recorded audio data in WAV format for eight basic notes: C, D, E, F, G, A, B, and C'. Each note

had 30 samples, with 20 used for training and 10 for testing. The model was built using Python and the TensorFlow library. The results showed that the system performed efficiently and accurately with a simple configuration, consisting 16 flatten inputs, 30 epochs, 16 neurons in the fully connected layer, and a batch size of 2. With this setup, the system reached 91.25% accuracy in real-time notes recognition. This shows that 1D CNN can be used in a simple but effective way to recognize pianica notes.
Keywords: Convolutional Neural Network, Note Recognition, Pianica

Pendahuluan

Latar Belakang

Nada merupakan suatu bunyi yang beraturan dengan frekuensi tunggal tertentu dan memliki tinggi nada tertentu menurut frekuensinya (Khairally, 2023). Nada memiliki tangga nada yang terbagi menjadi tiga, yaitu nada diatonis, nada pentatonis, kromatis. Tangga nada mayor merupakan nada diatonik dengan jarak atau interval nada 1-1-½-1-1-1-½. Tangga nada mayor ini biasanya menghasilkan suara yang terdengar riang, ceria, dan lebih semangat. Tangga nada minor memiliki jarak atau interval nada 1-½-1-1-½-1-1. Tangga nada minor ini memiliki suasana yang terkesan lebih sedih dan sendu. Tangga nada pentatonis merupakan tangga nada yang terdiri dari lima nada pokok. Tangga nada kromatik atau yang biasa disebut kres merupakan kenaikan satu atau setengah nada dari nada dasarnya, sedangkan nada mol merupakan penurunan satu atau setengah nada dari nada dasarnya (Azizah, 2024).

Seiring dengan peningkatan kebutuhan akan sistem otomatis yang dapat mendeteksi dan menganalisis nada secara akurat, teknologi pengenalan suara dan musik telah berkembang pesat. Pianika, sebuah alat musik tiup dengan tuts seperti piano, menghasilkan nada berdasarkan tiupan dan penekanan tuts, membuatnya menjadi tantangan tersendiri bagi sebagian besar pemula dalam pembelajaran musik di sekolah khususnya. Oleh karena itu, dibutuhkan sistem otomatis yang mampu mengenali nada-nada pianika secara tepat untuk membantu proses pembelajaran musik menjadi lebih interaktif, efisien, dan mudah dipahami.

Pada era digital ini, pengolahan sinyal audio menjadi sangat penting, terutama berkat perkembangan teknik kecerdasan buatan seperti Convolutional Neural Network (CNN). CNN adalah salah satu algoritma pembelajaran mesin yang paling akurat dalam mengenali pola terutama dalam bentuk visual atau citra dengan menggunakan CNN dua dimensi (Nurlatifa, 2024). Namun dalam perkembangannya, CNN juga telah dikembangkan dalam bentuk satu dimensi yang dapat digunakan untuk menganalisis data audio, seperti frekuensi nada. Untuk tiap nada pianika, CNN satu dimensi dapat belajar mengenali pola-pola spektrum suara yang khas, yang memungkinkan sistem untuk secara otomatis mengenali dan mengklasifikasikan nada tersebut (Kiranyaz dkk., 2021). CNN satu dimensi memiliki potensi besar untuk mendukung proses pembelajaran musik dengan menggunakan pengenalan nada untuk instrumen seperti pianika.

Sistem otomatis pengenalan nada membantu siswa dan guru musik melacak ketepatan permainan, sehingga proses pembelajaran menjadi lebih efisien. Selain itu, sistem ini berpotensi dikembangkan menjadi perangkat lunak atau aplikasi untuk mendukung pembelajaran mandiri. Berdasarkan hal tersebut, penelitian ini bertujuan untuk merancang dan menerapkan model CNN yang mampu mengenali nada pianika secara otomatis melalui pengolahan sinyal audio, sehingga nada dapat diklasifikasikan dengan tingkat akurasi yang tinggi menggunakan teknik jaringan saraf tiruan.

Penelitian sebelumnya telah dilakukan oleh Ferdiawan dkk. (2022) dimana penelitian terdahulu telah dapat mengenali akor dengan akurat, tetapi menggunakan banyak *input* pada *flatten layer* hingga ribuan data yang masuk. Penelitian ini menghasilkan tingkat akurasi yang optimal tetapi dengan minimum *input* pada *flatten layer* khususnya CNN 1 dimensi. Penelitian yang dilakukan menggunakan data masukan yaitu wav yang menjadi acuan agar jaringan saraf

tiruan pada model dapat terlatih dengan lancar dan dapat mengenali nada secara efisien dan optimal. Selain itu, pada penelitian ini menggunakan tampilan antarmuka pengguna GUI yang tidak menggunakan tombol rekam agar dapat mempermudah pengguna dalam menggunakan sistem yang ada.

Tujuan dan Manfaat

Tujuan dari penelitian ini adalah menghasilkan sistem pengenalan nada alat musik pianika yang efisien dan optimal dengan menggunakan *input* pada *flatten layer* CNN 1D yang minimum. Manfaat dari penelitian ini bagi masyarakat khususnya untuk siswa dan guru musik adalah agar dapat mempermudah proses penganalan nada pada setiap not alat musik pianika dengan nada pengenalan yang dimainkan adalah C, D, E, F, G, A, B, dan C'.

Spesifikasi dan Batasan Masalah

Pengenalan nada pianika terdiri dari 2 perangkat yaitu perangkat keras (hardware) dan perangkat lunak *(software)*. Perangkat keras *(hardware)* yang digunakan berupa alat musik pianika, sedangkan perangkat lunak *(software)* yang digunakan adalah *python*. Berikut ini merupakan spesifikasi yang telah ditentukan:

1. GUI sebagai antarmuka pengguna.
2. Menggunakan bahasa pemrograman *python*.
3. Pengklasifikasian menggunakan masukan *frame* audio dan keluaran label nada yang sesuai (menggunakan CNN).
4. Data yang akan ditampilkan di keluaran berupa nada dalam bentuk teks.

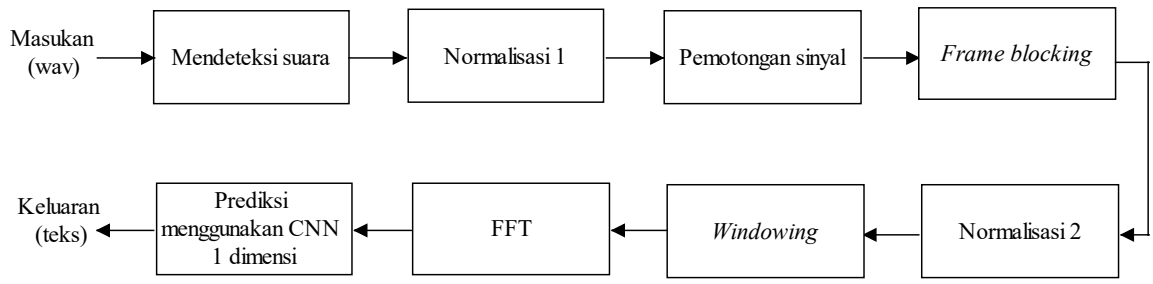
Agar penelitian ini berfokus pada tujuan utama dan menghindari adanya permasalahan yang lebih kompleks, maka perlu ada batasan-batasan masalah. Berikut merupakan batasan masalah yang telah ditentukan:

1. Penelitian ini terbatas hanya pada pengenalan nada dari instrumen musik pianika.
2. GUI akan menampilkan nada-nada yang dikenali yaitu 1 oktaf penuh.
3. Menggunakan mikrofon audio eksternal.
4. Pengenalan nada bersifat *real-time*.
5. *Frame blocking* yang digunakan 256.
6. Menggunakan FFT, *Windowing Hamming*.
7. Jarak antara pianika dan mikrofon adalah ±15 cm.
8. *Neural network* menggunakan 1 *hidden layer* pada *fully connected layer*.
9. Dalam GUI, nada langsung dikenali begitu terdengar (tanpa tombol rekam).
10. Ada GUI untuk *training* dan *testing*.

Metode

A. Perancangan Sistem keseluruhan

Sistem bekerja dengan menggunakan *Convolutional Neural Network* (CNN) secara optimal dengan jumlah masukan yang tidak terlalu banyak pada *flatten layer*. Diagram alir secara keseluruhan dapat dilihat dalam gambar 1. Masukan yang ada di dalam sistem adalah format *file* wav. Keluaran dari sistem adalah teks nada alat musik pianika yang ditampilkan menggunakan *user interface* yaitu GUI.



Gambar 4. Diagram blok keseluruhan sistem pengenalan nada alat musik pianika

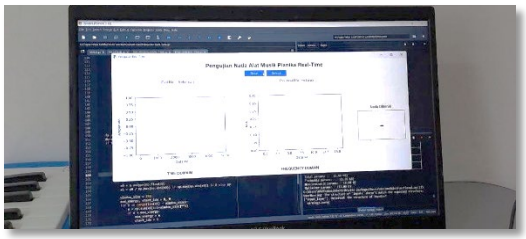
a) Masukan

Masukan dari keseluruhan sistem pengenalan nada alat musik pianika yang dibuat adalah *file* dengan format wav. Masukan yang dapat dideteksi oleh sistem ini berasal dari suara pianika. Dalam penelitian ini, nada pianika yang dikenali terdapat delapan nada yaitu C, D, E, F, G, A, B, C' dengan frekuensi *sampling* untuk perekaman sinyal yaitu 4800 Hz (Huizen dkk, 2015). Proses *sampling rate* harus memenuhi kriteria *Nyquist* yang menyatakan bahwa *sampling rate* harus lebih besar sama dengan dua kali dari frekuensi tertinggi sinyal analog. Persamaan di bawah merupakan rumus pada kriteria *Nyquist* yang secara sistematis ditulis sebagai berikut (Sumarno, 2018):

$$F_s \geq 2F_m \quad (1)$$

Dalam penelitian ini, alat musik pianika yang digunakan adalah pianika Yamada dan mikrofon L20. Gambar 2 merupakan setup penelitian yang dilakukan. Gambar 3 merupakan setup tampilan penelitian pada laptop

Gambar 5. Setup penelitian yang dilakukan



Gambar 6. Setup tampilan penelitian pada laptop

b) Mendeteksi suara

Suara pada alat musik pianika akan terdeteksi jika suara tersebut cukup keras dari yang telah ditetapkan pada sistem yaitu 0.02 untuk *threshold* yang didapat dari hasil empiris atau hasil *trial and error* yang sudah dilakukan. Jika sistem tidak mendeteksi sinyal suara yang cukup kuat, maka sinyal tersebut dianggap sebagai keheningan. Sinyal akan diperoleh selama satu detik selama suara dibunyikan maka sinyal akan langsung diproses untuk mendeteksi nada. Proses tersebut akan terus berjalan selama pengguna melakukan pengenalan nada alat musik pianika.

c) Normalisasi 1

Normalisasi adalah proses untuk memperbaiki hasil rekaman karena terpengaruh dari jarak sumber suara alat musik dengan mikrofon (Suryadi, 2019). Tujuan proses normalisasi

untuk mencari nilai maksimum skala amplitudo, mempermudah ekstraksi fitur, dan meningkatkan akurasi model (Riyani dkk., 2019). Nilai normalisasi dapat dirumuskan pada persamaan sebagai berikut:

$$X_{norm} = \frac{X_{masukan}}{\max(abs(X_{masukan}))} \quad (2)$$

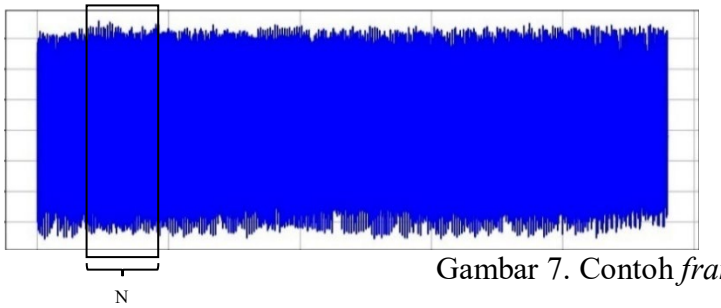
Keterangan: X_{norm} = Sinyal ternormalisasi
$X_{masukan}$ = Sinyal masukan
abs = nilai *absolute*
max=nilai tertinggi

d) Pemotongan sinyal

Pemotongan sinyal adalah proses pemotongan daerah hening *(silence)* dan daerah transisi awal pada suatu sinyal nada (Riyani dkk., 2019). Tujuan dari pemotongan sinyal ini adalah untuk membuat hasil suara yang diperoleh merupakan suara murni dari alat musik pianika tanpa gangguan *(noise)* dalam hasil pengenalan nada. Pemotongan sinyal ini menggunakan ambang batas atau *threshold* yaitu 0.5 yang didapat dari hasil *trial and error* yang dilakukan.

e) *Frame blocking*

Frame blocking merupakan suatu proses memecah sinyal suara menjadi potongan kecil yang disebut *frame*. Tujuan dari *frame blocking* adalah untuk memproses suatu sinyal ke dalam waktu yang lebih pendek agar dapat lebih mudah dalam perhitungan dan analisa suara. Proses ini merupakan hasil dari tahap pemotongan sinyal, di mana data diambil sepanjang nilai frame dengan ukuran 256. Pemotongan ini dilakukan satu kali pada bagian awal sinyal suara yang masuk (Sugianta dkk., 2020). Gambar di bawah merupakan contoh dari *frame blocking*.



N=sampel *frame blocking*

Gambar 7. Contoh *frame blocking*

f) Normalisasi 2

Selanjutnya proses data masukan yang sudah melewati proses sebelumnya akan dibagi dengan nilai maksimal absolut dari *frame blocking*, proses ini akan mendapatkan nilai normalisasi akhir. Proses normalisasi 2 memiliki pengertian yang sama dengan normalisasi 1 yang ada sebelumnya. Tujuan dari normalisasi kedua ini adalah untuk menstandarkan kembali amplitudo setiap frame agar data memiliki skala yang seragam dan stabil sebelum masuk ke tahap pemrosesan berikutnya.

g) *Windowing*

Proses *windowing* bertujuan untuk mengurangi efek dikontinuitas yang terdapat pada tepi-tepi sinyal (Riyani dkk., 2019). Efek diskontinuitas muncul akibat dari proses *frame blocking*. Tahap *windowing* terdapat beberapa jenis *windowing* untuk membuat efek diskontinuitas ini lebih sedikit. Penelitian yang dilakukan menggunakan *hamming window*. Rumus yang digunakan dapat dilihat pada persamaan sebgai berikut (Podder dkk., 2014):

$$W(n) = 0{,}54 + 0{,}46\cos\left(\frac{2\pi n}{N-1}\right) \quad (3)$$

Keterangan: W(n) = *Hamming window*
n = Waktu diskrit
N = jumlah data dari sinyal

h) FFT

Fast Fourier Transform (FFT) merupakan metode yang lebih efisien dalam menghitung koefisien dari *DFT* ke *finite sequence* dari data yang kompleks (Sipasulta dkk., 2014). *DFT* atau *Discrete Fourier Transform* adalah suatu teknik dalam matematika yang digunakan untuk mengubah nilai sekuen atau urutan nilai diskrit tertentu. *FFT* berfungsi untuk mengubah sinyal dari *domain waktu* menjadi *domain frekuensi*, agar informasi spektrum frekuensi yang terkandung dalam sinyal dapat dianalisis secara lebih mendalam.

Hubungan antara *DFT* dan *Transformasi Fourier* adalah bahwa *DFT* merupakan bentuk diskrit dari *Transformasi Fourier* yang diterapkan pada sinyal digital. *Transformasi Fourier* sendiri digunakan untuk menganalisis komponen frekuensi dari sinyal kontinu, sedangkan *DFT* memungkinkan analisis tersebut dilakukan secara komputasional.

Dalam penelitian ini, penggunaan *FFT* menjadi langkah penting untuk mengonversi sinyal suara pianika ke dalam *domain frekuensi*, sehingga pola spektrum tiap nada dapat diekstraksi dan dimanfaatkan sebagai fitur utama dalam proses pengenalan nada menggunakan *Convolutional Neural Network (CNN)* satu dimensi. Rumus DFT dapat dilihat pada persamaan sebagai berikut:

$$X(k) = \sum_{n=0}^{N-1} X(n) e^{-j\left(\frac{2\pi}{N}\right)kn} \quad (4)$$

Keterangan: X = Indeks domain
N = Jumlah sampel data
e = *Natural number* (2,7182818284…)
k = Indeks *frekuency domain* (0, 1, 2, 3, …, N-1)
n = Indeks *time domain* (0, 1, 2, 3, …, N-1)

i) Prediksi menggunakan CNN 1 dimensi

Subsistem *CNN (Convolutional Neural Network)* terdiri atas *convolutional layer*, *pooling layer*, dan *fully connected layer*. *CNN* yang telah dilatih hingga memperoleh hasil akurasi yang memadai kemudian diuji untuk mengenali nada alat musik pianika agar data hasil pengenalan nada menjadi lebih akurat. Proses prediksi nada pada sistem pengenalan menggunakan *CNN* satu dimensi ditentukan oleh lapisan terakhir, yaitu *fully connected layer* dengan fungsi aktivasi *softmax*.

Fungsi aktivasi *softmax* berperan untuk mengubah output dari jaringan saraf menjadi probabilitas bagi setiap kelas atau label yang ada, sehingga total seluruh nilai output bernilai 1. Dengan demikian, *softmax* memungkinkan sistem untuk menentukan nada mana yang memiliki kemungkinan tertinggi untuk dikenali. Batas yang digunakan dalam *softmax* adalah nilai probabilitas tertinggi di antara seluruh kelas yang tersedia, di mana kelas dengan nilai terbesar akan dipilih sebagai hasil prediksi akhir.

Dari beberapa nada yang tersedia *(C, D, E, F, G, A, B, C')*, sistem akan memberikan nilai tertinggi sebagai hasil prediksi akhir untuk salah satu nada yang dikenali menggunakan fungsi aktivasi *softmax* tersebut. Setiap indeks pada output tersebut telah dipetakan (*mapping*) sebelumnya ke label nada, seperti *0 = C, 1 = D, 2 = E, 3 = F, 4 = G, 5 = A, 6 = B, 7 = C'*. Dengan demikian, angka hasil prediksi tersebut secara otomatis akan dikonversi menjadi huruf alfabet yang merepresentasikan nada-nada pianika.

j) Keluaran

Dalam proses menentukan nada pianika, pianika akan dimainkan dengan menekan sebuah nada yang akhirnya akan menampilkan sebuah keluaran dalam tampilan *user interface*

pada GUI yaitu berupa teks nada pianika. Nada akan dikenali oleh program dengan cepat dan tepat.

B. Evaluasi parameter

Dalam perancangan sistem pengenalan nada pianika, kualitas klasifikasi dipengaruhi oleh beberapa parameter, yaitu jumlah *epoch*, *batch size*, jumlah *neuron* pada *fully connected layer*, dan jumlah *input flatten*. Untuk menemukan konfigurasi terbaik, dilakukan pengujian dengan variasi nilai pada tiap parameter. Perbandingan ini bertujuan menilai pengaruh perubahan parameter terhadap akurasi sekaligus mengevaluasi kinerja sistem dari segi ketepatan dan efisiensi.

a) Pengaruh jumlah *epoch*

Epoch merupakan satuan yang menyatakan berapa kali seluruh data pelatihan digunakan untuk memperbarui sistem pengenalan nada (Browniee, 2018). Semakin banyak *epoch*, semakin sering sistem menyesuaikan data latih sehingga tingkat akurasi meningkat. Namun, jumlah *epoch* yang terlalu tinggi berisiko menimbulkan overfitting, yaitu sistem hanya mengenali data latih dan gagal pada data baru. Karena itu, dilakukan pengujian dengan variasi epoch 5, 10, 15, 20, 25, 30, 35, 40, 50, 60, dan 100 untuk menilai pengaruhnya terhadap kinerja sistem.

b) Pengaruh jumlah *batch size*

Batch size adalah jumlah data yang digunakan dalam satu kali proses penyesuaian parameter (Rochmawati dkk., 2021). *Batch size* yang kecil cenderung menghasilkan proses pelatihan yang lebih detail dan akurat, namun membutuhkan waktu yang lebih lama. Sebaliknya, *batch size* yang besar memang mempercepat proses pelatihan kerena lebih banyak data diproses secaara bersamaan, tetapi berisiko menurunkan kemampuan sistem dalam mengenali suara nada. *Batch size* yang divariasi adalah dengan ukuran 2, 4, 8, 16, dan 32. Dengan melakukan variasi terhadap *batch size*, dapat diketahui ukuran yang paling seimbang antara ketepatan dan efisiensi pelatihan.

c) Pengaruh jumlah *neuron* pada *fully connected layer*

Jumlah *neuron* pada *fully connected layer* menentukan tingkat kompleksitas sistem dalam mengolah informasi yang telah diproses sebelumnya (Christianto dkk., 2021). Semakin banyak *neuron* yang digunakan, semakin tinggi pula kemampuan sistem untuk membedakan karakteristik dari setiap nada. Namun demikian, penggunaan *neuron* yang terlalu banyak dapat menyebabkan sistem menjadi kompleks dan kurang efisien. Oleh karena itu, dilakukan pengujian dengan jumlah neuron yang berbeda untuk memperoleh konfigurasi yang paling sesuai yaitu dengan jumlah neuron 2, 4, 8, 16, 32, dan 64.

d) Pengaruh jumlah *input* pada *flatten layer*

Tahap *flatten* merupakan proses perubahan bentuk data menjadi satu dimensi sebelum dilakukan proses pengenalan akhir (Ardiansyah dkk., 2023). Data yang ada pada ekstraksi ciri pada CNN menghasilkan data yang sangat banyak untuk pengenalan nada. Jumlah data pada tahap ini mencerminkan seberapa besar informasi yang dikumpulkan dari proses ekstraksi ciri sebelumnya. Jumlah data yang terlalu sedikit dapat menyebabkan hilangnya informasi penting, sedangkan jumlah yang terlalu banyak dapat memperbesar beban pemrosesan. Oleh sebab itu, dilakukan pengujian terhadap jumlah data masukan pada tahap ini yang berguna mengetahui seberapa besar pengaruhnya terhadap hasil akhir sistem pengenalan nada. Sistem menggunakan *input* yaitu 8, 16, 32, 64, 128, 256, 512, dan 1024.

C. Pelatihan dan pengujian tidak *real-time*

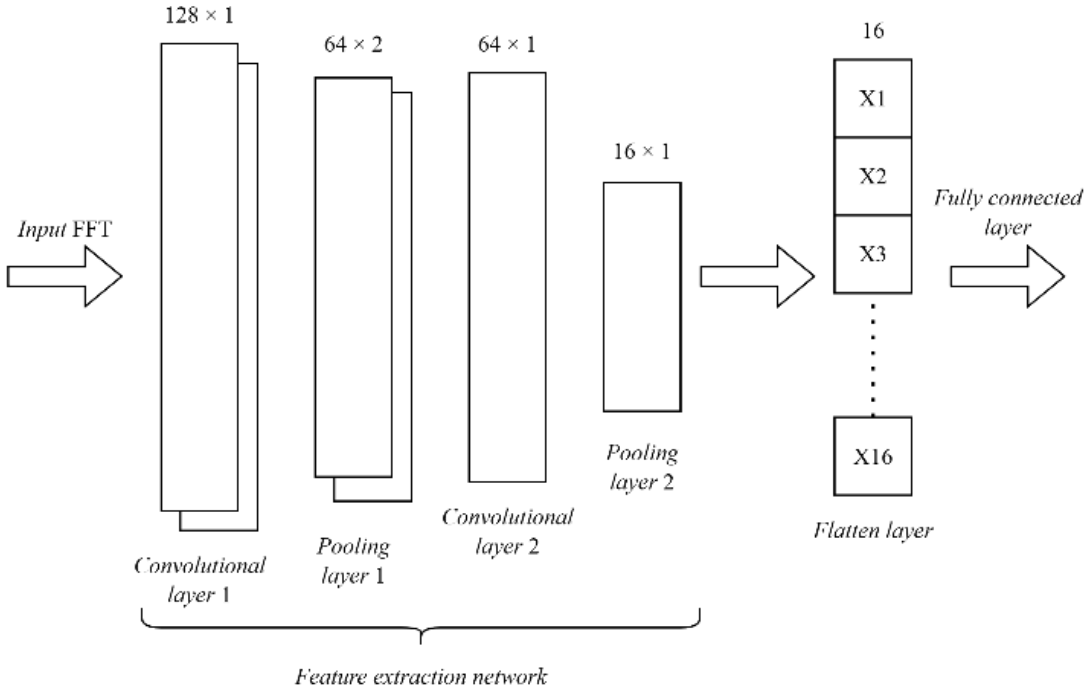
Proses pelatihan dilakukan secara tidak *real time* dengan menggunakan program *python*. Dalam proses pelatihan dilakukan untuk melatih program agar lebih mengenal nada pianika secara tepat dan akurat. Proses awal dilakukan dengan merekam delapan nada pianika, yaitu C, D, E, F, G, A, B, C'. Setiap nada diambil datanya sebanyak 30 kali dan diberi nama yang berbeda untuk setiap rekamannya, dengan *file* penyimpanan dalam bentuk *wav*. Kemudian pada data yang ada dipisahkan menjadi data

untuk pelatihan sebanyak 20 kali dan untuk pengujian tidak *real-time* sebanyak 10 kali. Proses pelatihan dan pengujian tidak *real-time* sebelum menuju ke *convolutional neural network* satu dimensi sebagai ekstraksi ciri sekaligus klasifikasi akan diolah terlebih dahulu ke *pre-processing*. Pelatihan akan dilakukan pembaruan bobot menggunakan *epoch* yang ditentukan. Proses pertama setelah mengambil adalah melakukan *pre-processing* yang sama dengan proses yang telah dijelaskan sebelumnya dari tahap c) normalisasi sampai tahap h) FFT.

Setelah melakukan *pre-processing* maka data yang ada akan masuk ke *convolutional neural network* satu dimensi. Tahap pertama yang dilakukan adalah *convolutional layer* yang berfungsi untuk mengekstraksi fitur dari sinyal masukan (Mahendra dkk., 2020). *Layer* ini akan menerapkan filter (jumlah kernel) ke sinyal satu dimensi dari hasil ekstraksi FFT sebelumnya, guna menghasilkan fitur penting yang mewakili karakteristik dari masing-masing nada pianika (Alwanda dkk., 2020).

Convolutional layer ini mempunyai ukuran *kernel* yaitu tiga. *Convolutional layer* membuat *input* yang telah masuk didalamnya akan semakin besar karena pengaruh jumlah filter yang ditentukan. Tetapi pada arsitektur model ini, jumlah filter justru dikurangi, untuk *layer* pertama menggunakan 2 filter dan layer kedua menggunakan 1 filter. Jumlah filter ini yang menentukan jumlah *channel* (dimensi fitur) pada *output convolutional layer*, maka semakin sedikit filter yang digunakan, semakin sedikit *output* yang dihasilkan. Kemudian setelah *convolutional layer*, dilakukan proses pada *pooling layer* digunakan untuk mengurangi dimensi dari *output convolutional layer* tanpa menghilangkan informasi penting (Rodiah dkk., 2024). *Pooling* yang digunakan dalam sistem ini adalah *average pooling*, yang akan menghitung nilai rata-rata dari sejumlah fitur dalam jendela tertentu (Anhar dkk., 2024).

Pooling layer yang digunakan untuk pertama dengan *pool_size*=2, *strides*=2 sehingga didapat 64×2, kemudian proses *pooling* yang kedua menggunakan *pool*_size=4, *strides*=4 sehingga didapat 16×1. Gambar dibawah ini merupakan proses *convolutional layer* dan *pooling layer*.

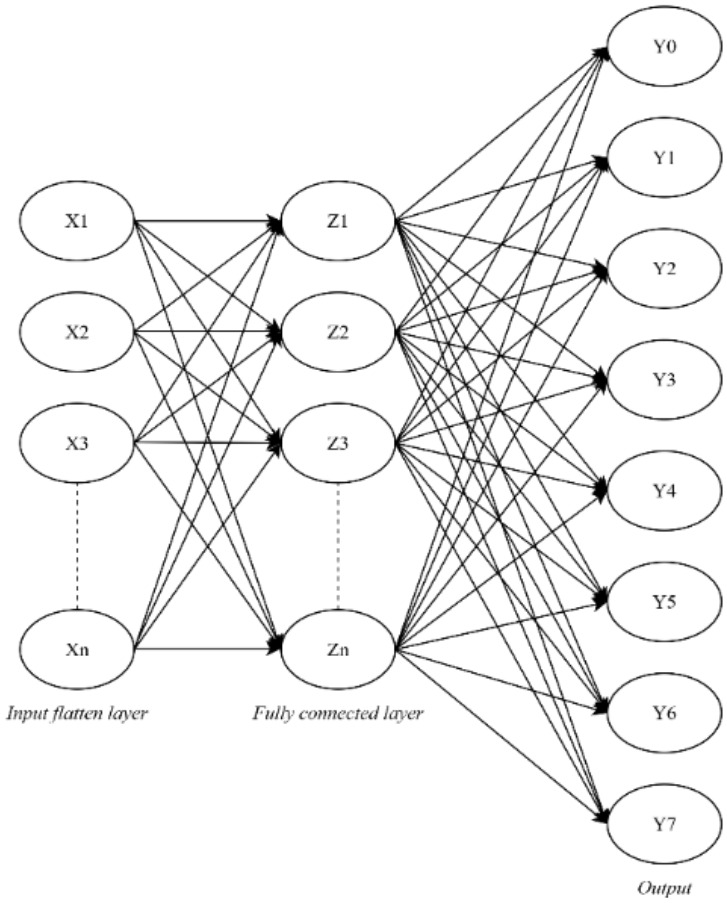


Gambar 8. Proses *convolutional layer* dan *pooling layer*

Setelah melewati beberapa lapisan konvolusi dan *pooling*, data akan diratakan *(flatten)* dan dimasukkan ke dalam *fully connected layer (dense layer)*. Tahap ini, setiap *neuron* terhubung penuh ke *neuron* di lapisan sebelumnya. *Neuron* yang digunakan pada *fully connected layer* ini adalah 8 dengan fungsi aktifasi yaitu ReLU yang berfungsi mengaktivasi memungkinkan model untuk menyelesaikan masalah *non-linear* (Lu dkk., 2020).

Lapisan ini bertugas untuk melakukan klasifikasi akhir berdasarkan fitur-fitur yang telah

diekstraksi sebelumnya dengan menggunakan fungsi aktivasi *softmax* (Pangestu dkk., 2020). *Output* dari layer ini adalah prediksi terhadap kelas nada pianika yang dikenali, misalnya C, D, E, F, G, A, B, dan C'. Gambar berikut merupakan proses *fully connected layer* menuju hasil *output* yang diperoleh.



Gambar 9. Proses *fully connected layer*

Selama pelatihan menggunakan optimasi adam yang lebih stabil dan akurat (Irfan dkk., 2022). Penetapan *learning rate* yaitu 0,001 dan *random number* yang dipakai adalah *random number* 42 karena sering digunakan pada beberapa penelitian sekaligus yang memberikan hasil optimal pada sistem pengenalan nada alat musik pianika (Panjaitan, 2022).

D. Pengujian *real-time*

Pengujian model tidak untuk memperbarui bobot, tetapi dalam *testing* atau dalam pengujian ini hanya *forward propagation* atau tidak belajar (Budiarti dkk., 2024). Pengujian ini tidak menggunakan tombol mulai dan *reset* karena pada pengujian program *python* dibuat agar dapat mengulang kembali perintah untuk dapat mengenali nada alat musik pianika. Proses awal dalam pengujian adalah mikrofon akan merekam suara dari nada yang ditekan pada alat musik pianika tanpa menggunakan tombol mulai, lalu hasil rekaman tersebut langsung diproses oleh program yang telah dibuat pada *python.* Proses pengujian dilakukan oleh satu orang yang memainkan pianika. Untuk pengujian nada pada alat musik pianika akan menyembunyikan 8 nada yaitu: C, D, E, F, G, A, B, C' pada setiap nada akan dilakukan sebanyak 10 kali, sehingga data yang akan dikenali dapat dilakukan secara *real time.*

Setiap nada diuji sebanyak sepuluh kali, kemudian dihitung jumlah nada pianika yang berhasil dikenali dengan benar menggunakan rumus yaitu:

$$\text{Tingkat Pengenalan} = \frac{\sum v}{\text{Banyak Percobaan}} \times 100\% \quad (5)$$

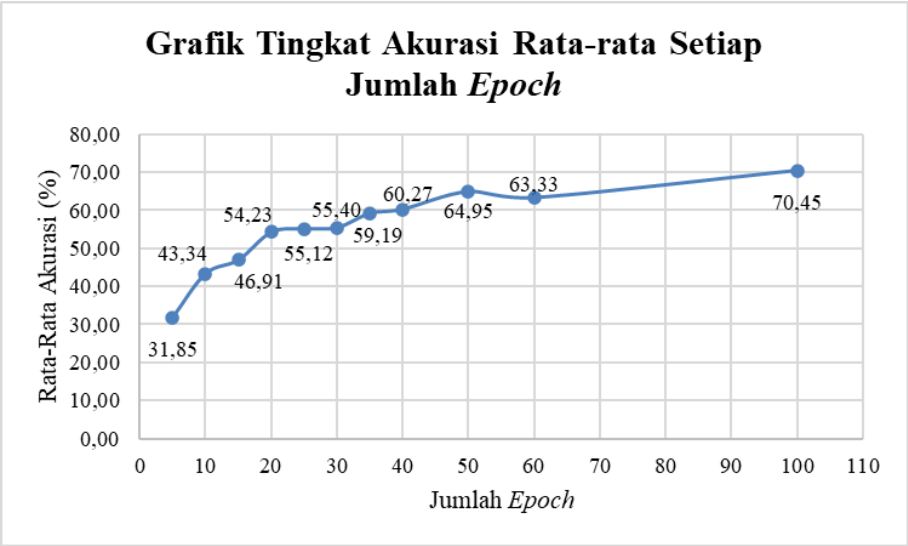
Dengan $\sum v$ merupakan jumlah benar tiap nada pianika.

Hasil dan Pembahasan

A. Hasil evaluasi parameter

a) Pengaruh jumlah *epoch*

Sama seperti yang ada pada perancangan evaluasi dilakukan pada nilai *epoch* seperti 5, 10, 15, 20, 25, 30, 35, 40, 50, 60, dan 100, hasilnya dibandingkan untuk mengetahui titik optimal pelatihan. Semua percobaan pada setiap jumlah *epoch* diberikan rata-rata akurasi dengan berbagai variasi parameter-parameter yang lain seperti *batch size*, *neuron* yang ada pada *fully connected layer*, dan *input* pada *flatten layer*. Gambar berikut merupakan grafik rata-rata akurasi *epoch*. Data yang diperoleh setiap rata-rata akurasi *epoch* adalah 240 data.

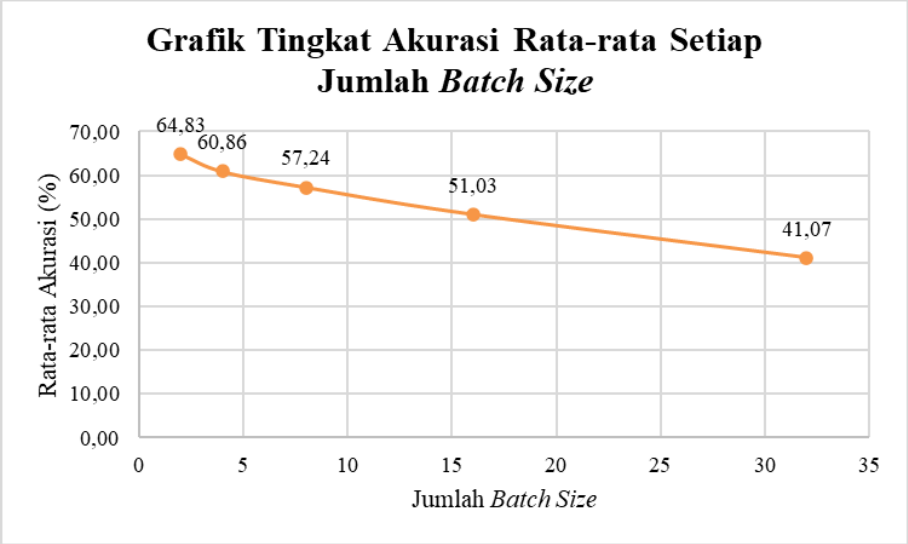


Gambar 10. Grafik tingkat rata-rata akurasi *epoch*

Berdasarkan grafik, semakin besar *epoch* maka sistem semakin banyak belajar sehingga akurasi meningkat. Saat *epoch* 5 hingga 50 akurasi terus naik, tetapi menurun pada epoch 60. Sementara itu, epoch 100 menunjukkan kenaikan kembali, namun tidak signifikan sehingga pengaruhnya kecil.

b) Pengaruh jumlah *batch size*

Dalam percobaan ini sama seperti pada perancangan digunakan beberapa nilai *batch size* seperti 2, 4, 8, 16, dan 32. Semua percobaan pada setiap jumlah *batch size* diberikan rata-rata akurasi dengan berbagai variasi parameter-parameter yang lain seperti *epoch*, *neuron* yang ada pada *fully connected layer*, dan *input* pada *flatten layer*. Dalam gambar di bawah merupakan visualisasi dari tingkat rata-rata akurasi dari *batch size*. Data yang diperoleh untuk setiap *batch size* yaitu 528 data.

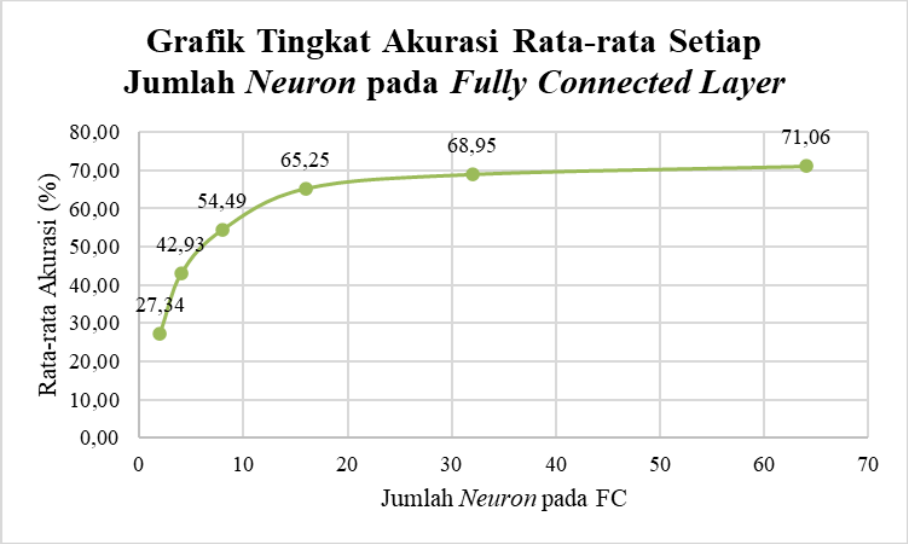


Gambar 11. Grafik tingkat rata-rata akurasi *batch size*

Berdasarkan tabel, semakin besar *batch size* maka akurasi menurun. *Batch size* besar memang mempercepat pelatihan karena lebih banyak data diproses sekaligus, tetapi berpotensi menurunkan akurasi sistem pengenalan nada.

c) Pengaruh jumlah *neuron* pada *fully connected layer*

Percobaan dibawah ini merupakan implementasi dari metode evaluasi parameter tentang pengaruh *neuron* pada *fully connected layer* dengan memvariasikan jumlah *neuron* pada *hidden layer fully connected* yaitu *neuron* 2, 4, 8, 16, 32, dan 64 untuk melihat apakah perubahan kompleksitas jaringan pada tahap akhir berpengaruh terhadap hasil klasifikasi nada. Semua percobaan pada setiap jumlah *neuron* pada *fully connected layer* diberikan rata-rata akurasi dengan berbagai variasi parameter-parameter yang lain seperti *epoch*, *batch size*, dan *input* pada *flatten layer*. Gambar berikut merupakan grafik untuk mempermudah melihat tingkat akurasi rata-rata pada *neuron* yang ada pada *fully connected layer*. Data yang diperoleh untuk setiap rata-rata akurasi setiap *neuron* pada *fully connected layer* adalah 440 data.

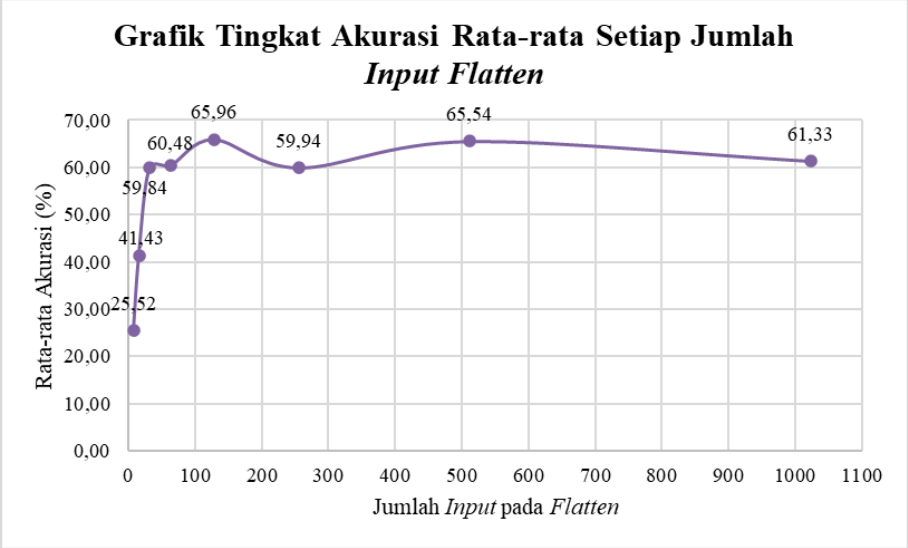


Gambar 12. Grafik tingkat rata-rata akurasi *neuron fully connected layer*

Berdasarkan tabel, penambahan *neuron* pada *fully connected layer* meningkatkan akurasi karena sistem lebih mampu menangkap pola. Namun, penggunaan neuron terlalu banyak tidak memberi peningkatan signifikan sehingga tidak selalu optimal.

d) Pengaruh jumlah *input* pada *flatten layer*

Hasil implementasi pada metode evaluasi parameter bagian *input* yang masuk pada *flatten layer* yaitu dengan memvariasikan jumlah *input* ini, sistem dapat menguji seberapa banyak fitur yang ideal untuk diteruskan ke proses klasifikasi. Seperti yang telah dirancang sebelumnya percobaan dilakukan dengan nilai *input* seperti 8, 16, 32, 64, 128, 256, 512, dan 1024 untuk menilai pengaruh kompleksitas *input* terhadap performa model. Semua percobaan pada setiap jumlah *input* pada *flatten layer* diberikan rata-rata akurasi dengan berbagai variasi parameter-parameter yang lain seperti *epoch*, *batch size*, dan *neuron* pada *fully connected layer*. Gambar 14. merupakan grafik tingkat akurasi rata-rata pada *input* yang masuk pada *flatten layer.* Data yang diperoleh untuk setiap rata-rata *input* pada *flatten layer* adalah 330 data.

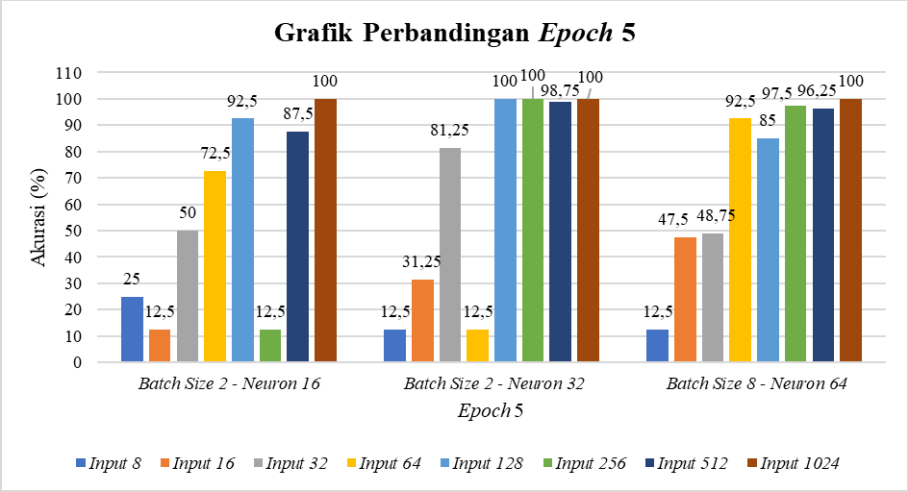


Gambar 13. Grafik tingkat rata-rata akurasi *input flatten layer*

Data menunjukkan bahwa semakin banyak *input* pada *flatten layer*, kemampuan deteksi nada semakin akurat. Namun, *input* besar perlu diimbangi dengan penyesuaian parameter, misalnya menurunkan *epoch*. Karena itu, pemilihan dimensi *input* harus seimbang agar model tetap akurat dan efisien.

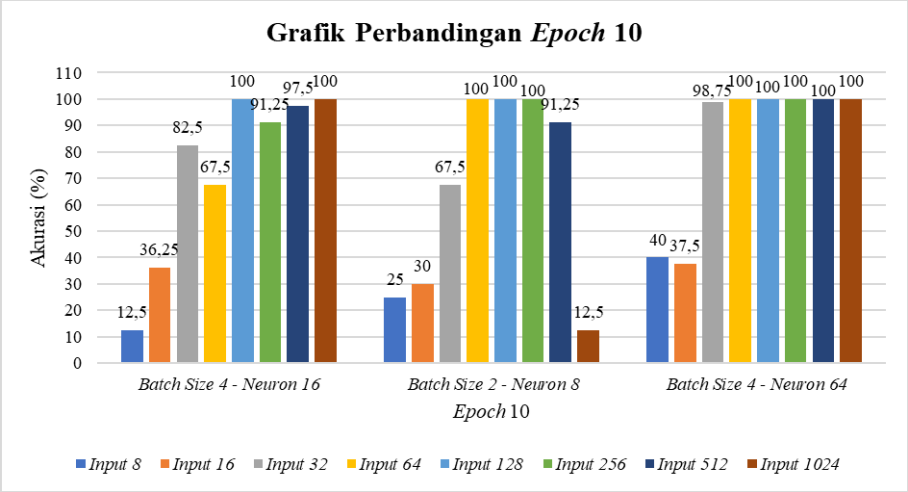
B. Hasil pelatihan dan pengujian tidak *real-time*

Tahap ini merupakan implementasi metode pelatihan dan pengujian tidak *real-time* untuk mengetahui pengaruh yang telah diberikan beberapa variasi parameter. Dalam grafik yang digambarkan menunjukkan tingkat akurasi yang telah dicapai oleh *input* pada *flatten layer* untuk mencapai hasil yang maksimal. Data diperoleh dari hasil terbaik dalam setiap *epoch* untuk menunjukkan bagaimana tingkat sistem dapat mengenali nada dengan akurat. Gambar berikut merupakan grafik perbandingan untuk *epoch* 5.



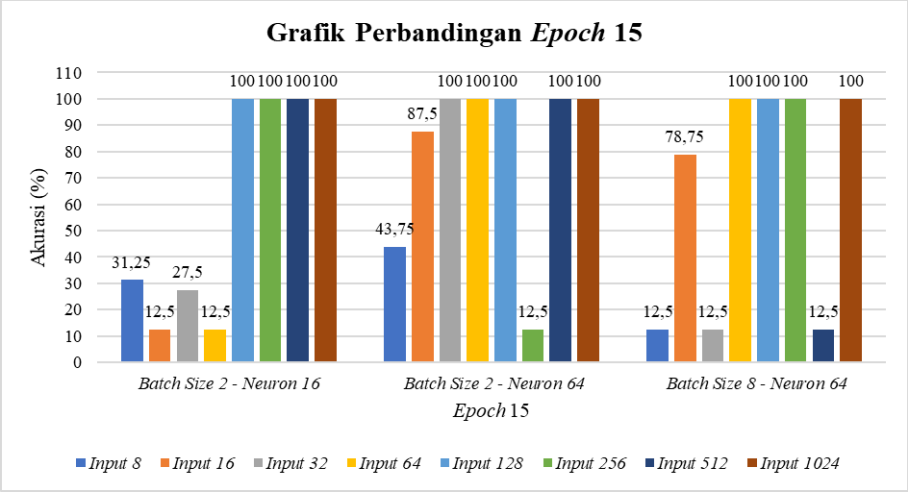
Gambar 14. Grafik hasil perbandingan *epoch* 5

Gambar 15 dapat dilihat bahwa dengan *input* yang banyak yaitu 128, 256, 512, dan 1024 dengan *batch size* 2, 2, 8 dan *neuron* 16, 32, 64 sudah dapat untuk memperoleh hasil yang akurat karena *convolutional neural network* dapat mendeteksi nada banyak *input*. Saat *input* yang kecil belum ada pengaruh yang terlalu banyak. Dapat dilihat bahwa beberapa *input* kecil belum dapat terlalu stabil untuk dapat mendapatkan hasil yang optimal untuk mendeteksi nada.



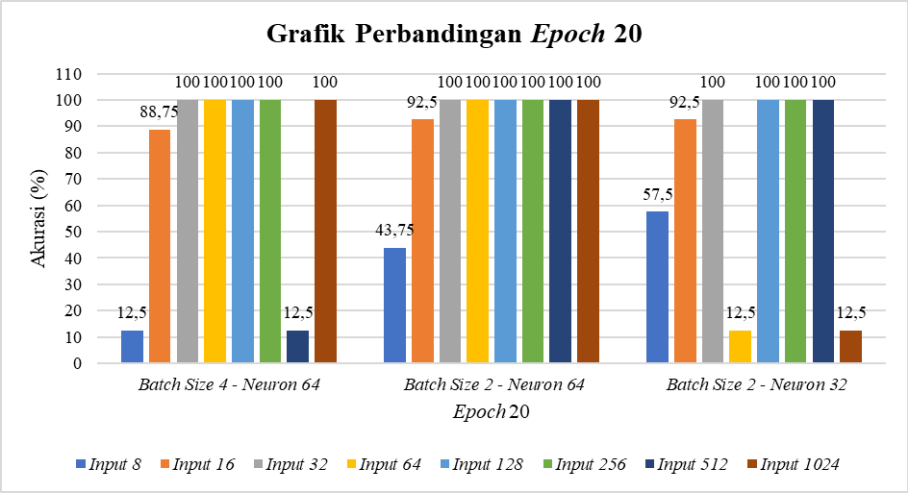
Gambar 15. Grafik hasil perbandingan *epoch* 10

Gambar 16 merupakan grafik hasil perbandingan *epoch* 10. Saat *input yang besar* masih stabil dengan tingkat akurasi yang tinggi. Saat *input* 32 *neuron* sudah menunjukkan beberapa peningkatan yang dapat dikatakan sebagai sistem yang optimal, tetapi untuk penelitian ini tetap mencari dengan seluruh variasi jumlah *epoch* yang ada agar mendapatkan sistem dengan minimal *input* yang dapat diterima oleh *convolutional neural network*.



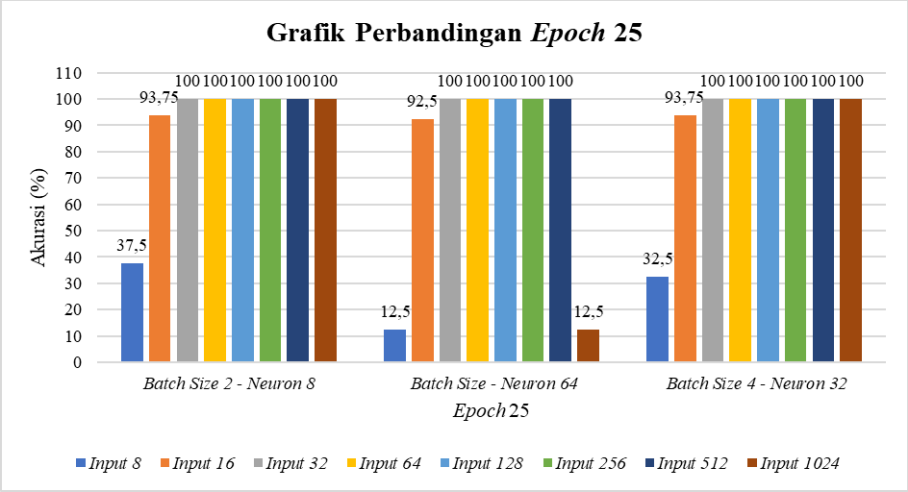
Gambar 16. Grafik hasil perbandingan *epoch* 15

Gambar 17 merupakan grafik hasil perbandingan *epoch* 15. Saat *input* besar (≥32 *neuron*) tetap menunjukkan tingkat akurasi yang konsisten jika dilihat dan dibandingkan. Saat *input* kecil seperti 16 *neuron* sudah mulai menunjukkan tingkat akurasi yang cukup, sedangkan untuk *input* pada 32 *neuron* sama seperti *epoch* 10 masih menunjukkan hasil yang cukup akurat untuk sistem. Saat *input* 8 *neuron* menunjukkan akurasi sedang atau belum optimal.



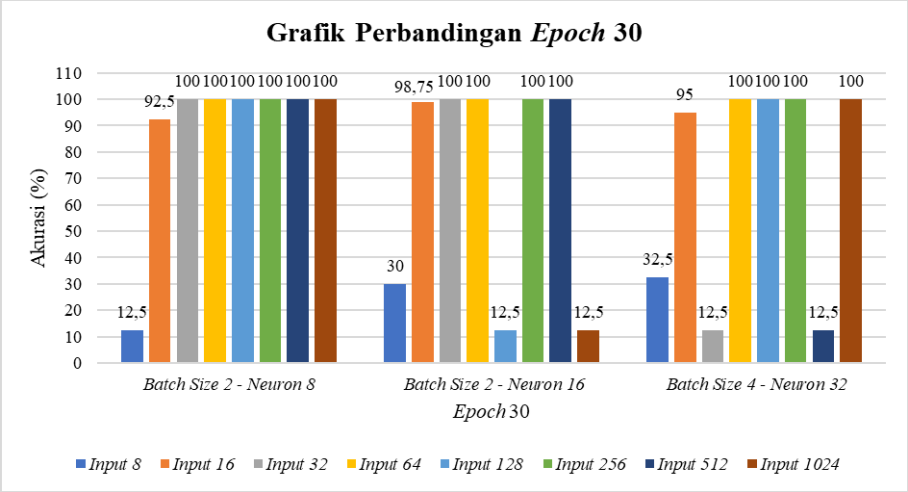
Gambar 17. Grafik hasil perbandingan *epoch* 20

Grafik perbandingan *epoch* 20 pada gambar 18 terlihat bahwa *input* besar masih menghasilkan tingkat akurasi yang tinggi. *Input* sebesar 32 menunjukkan hasil yang mendekati nilai maksimal pada beberapa kondisi. Sementara itu, *input* 16 memberikan performa yang cukup baik dengan akurasi di atas 80%. Adapun *input* 8 hanya mencapai akurasi sedang dan belum mampu melampaui 80%.



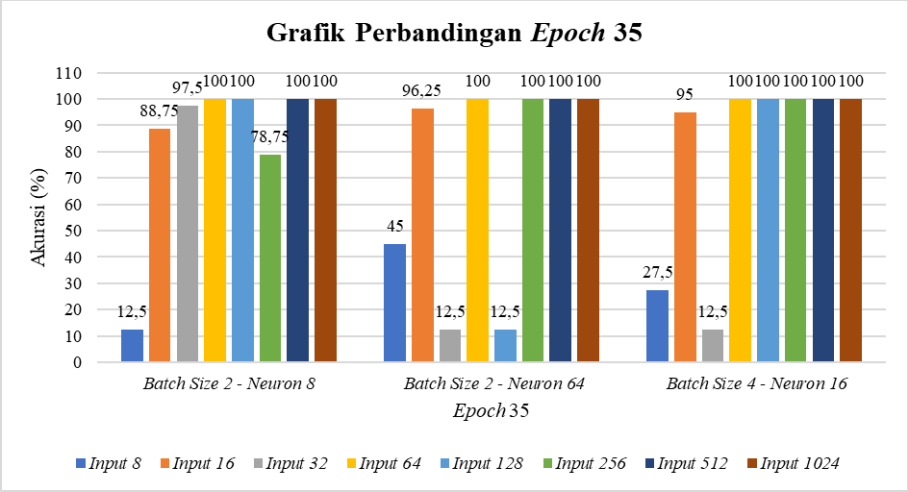
Gambar 18. Grafik hasil perbandingan *epoch* 25

Gambar 19 menunjukkan hasil perbandingan pada *epoch* 25. Grafik tersebut memperlihatkan adanya peningkatan akurasi pada beberapa *input*, terutama pada *input* kecil seperti 16 dan 32 neuron yang mencapai tingkat akurasi optimal. *Input* besar tetap menunjukkan performa yang akurat, sedangkan *input* 8 belum mampu mencapai akurasi yang optimal.



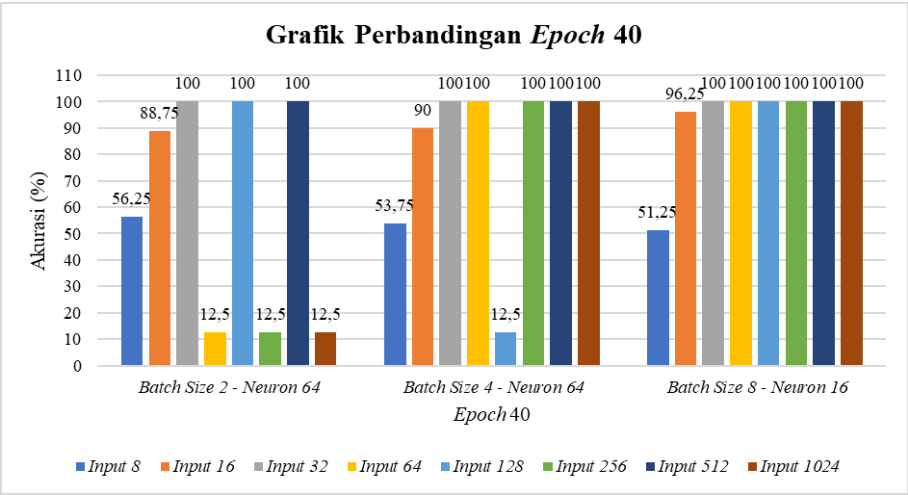
Gambar 19. Grafik hasil perbandingan *epoch* 30

Gambar 20 menunjukkan hasil perbandingan pada *epoch* 30, di mana sistem mulai menunjukkan performa optimal dalam mengenali nada untuk *input* di atas 8. Terlihat bahwa *input* 16 merupakan ukuran terkecil yang mampu menghasilkan kinerja sistem yang optimal.



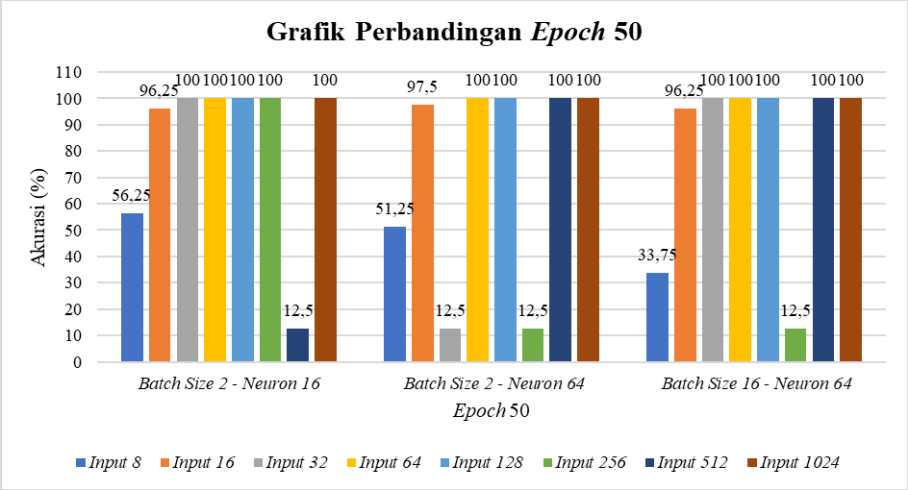
Gambar 20. Grafik hasil perbandingan *epoch* 35

Gambar 21 menunjukkan hasil perbandingan pada *epoch* 35. Terlihat bahwa *input* 8 masih belum mampu memberikan performa yang cukup akurat sebagai sistem pengenalan nada. Sementara itu, *input* di atas 8 tetap menunjukkan performa yang akurat dan optimal dalam mengenali nada.



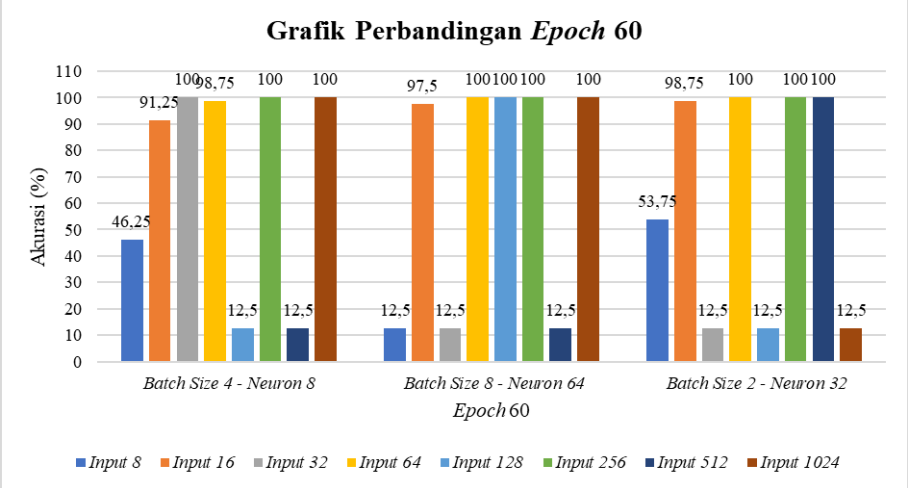
Gambar 21. Grafik hasil perbandingan *epoch* 40

Gambar 22 merupakan grafik perbandingan *epoch* 40 yang dapat dilihat *input* 8 sudah menunjukkan beberapa peningkatan dengan memperoleh tingkat akurasi diatas 50 dan beberapa *input* yang diatas 8 menunjukkan performa yang sama yaitu optimal karena dengan *input* yang banyak mendapat akurasi yang tinggi.



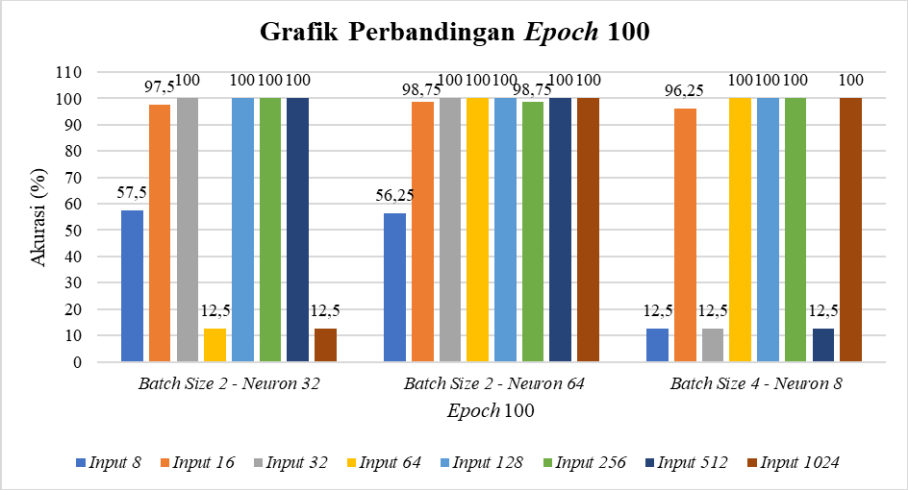
Gambar 22. Grafik hasil perbandingan *epoch* 50

Gambar 23 merupakan grafik perbandingan *epoch* 50 yang dapat dilihat *input* 8 belum dapat dikatakan sebagai sistem pengenalan yang akurat karena masih belum dapat menunjukkan tingkat akurasi diatas 80%, tetapi sudah cukup karena tingkat akurasi diatas 50. Untuk *input* diatas 8 masih tetap memiliki tingkat akurasi yang mirip dengan sebelumnya.



Gambar 23. Grafik hasil perbandingan *epoch* 60

Grafik perbandingan *epoch* 60 yang ditunjukkan oleh gambar 24 dapat dilihat pada *input* 8 masih belum dapat terlihat dapat mendeteksi suatu nada dengan akurat dan justru ada beberapa yang dibawah tingkat akurasi 50%. Jika dilihat beberapa *input* yang ada sudah menunjukkan sistem yang optimal untuk pengenalan nada.



Gambar 24. Grafik hasil perbandingan *epoch* 100

Gambar 25 menunjukkan grafik perbandingan pada *epoch* 100. Terlihat bahwa *input* 8 masih belum memenuhi kriteria sebagai *input* minimal hingga *epoch* 100. Ukuran *input flatten layer* yang paling kecil namun tetap dapat diterima adalah 16 neuron.

Hasil evaluasi sistem pengenalan nada pianika menunjukkan bahwa konfigurasi parameter pelatihan yang optimal adalah penggunaan *input flatten layer* sebesar 16, jumlah *epoch* 30, *batch size* 2, dan jumlah *neuron* pada *fully connected layer* sebanyak 16. *Input flatten layer* 16 terbukti cukup efisien secara komputasi namun tetap mampu menghasilkan akurasi di atas 90%, meskipun dimensi lebih besar memang memberikan akurasi lebih tinggi. Jumlah *epoch* optimal adalah 30, karena setelah titik ini peningkatan akurasi tidak signifikan dan justru berpotensi menimbulkan *overfitting* serta memperpanjang waktu pelatihan. Penggunaan *batch size* kecil yaitu 2 memberikan hasil terbaik, karena membuat model belajar lebih stabil pada dataset kecil sekaligus meningkatkan generalisasi, meskipun memerlukan waktu pelatihan lebih lama. Sementara itu, penggunaan 16 *neuron* pada *fully connected layer* dinilai cukup untuk menangkap pola penting tanpa menambah kompleksitas berlebih. Dengan kombinasi parameter tersebut, sistem mampu mengenali nada pianika secara konsisten dengan akurasi tinggi, efisien, dan tidak membutuhkan sumber daya komputasi yang besar.

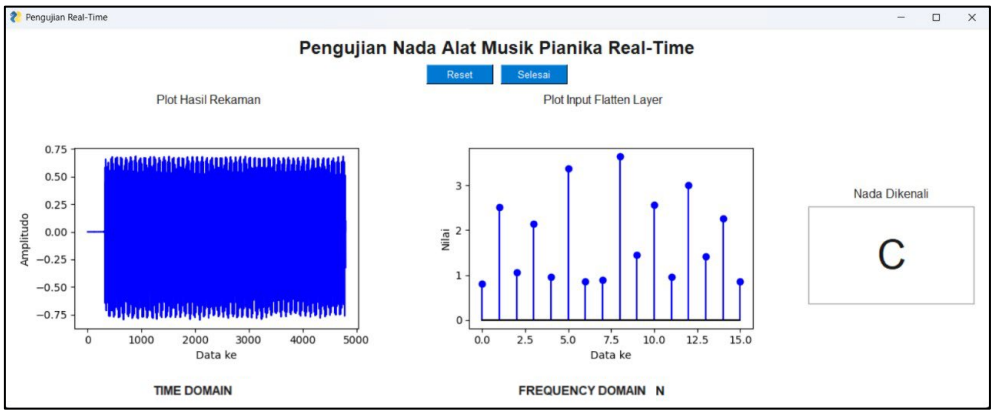
C. Hasil pengujian secara *real-time*

Pengujian *real-time* dilakukan dengan *input* langsung dari suara pianika sebanyak sepuluh kali percobaan pada tiap nada. Dalam tahap ini tidak ada lagi *update* bobot dan parameter, melainkan menggunakan konfigurasi terbaik hasil pelatihan, yaitu *epoch* 30, *batch size* 2, *neuron* 16, dan *input flatten* 16 dengan akurasi pelatihan 98,75%. Proses pengujian sama seperti tidak *real-time*, melalui tahap *pre-processing* dan CNN satu dimensi. Tujuannya adalah mengukur kinerja sistem dalam kondisi nyata. Hasil menunjukkan adanya sedikit kesalahan pengenalan, namun secara keseluruhan sistem mampu mendeteksi nada pianika secara *real-time* dengan akurat dan mencapai akurasi 91,25%.

Tabel 2. Tabel pengujian secara *real-time*

Input	***Output***								**Jumlah Nada Dikenali**
	C	**D**	**E**	**F**	**G**	**A**	**B**	**C'**	
C	8	0	0	0	0	0	0	2	8
D	0	8	0	0	0	0	2	0	8
E	0	0	10	0	0	0	0	0	10
F	0	0	0	10	0	0	0	0	10
G	1	0	0	0	9	0	0	0	9
A	0	0	0	0	0	10	0	0	10
B	0	1	0	0	0	0	9	0	9
C'	0	0	0	0	0	0	1	9	9

Hasil pengujian *real-time* yang dilihat pada tabel 1, untuk melihat perbedaan pada setiap nada dan dianalisis, maka dapat diberikan beberapa contoh gambar.



Gambar 25. Pengujian *real-time* nada C (pertama)

Gambar 26 merupakan hasil dari pengujian *real-time* nada C yang pertama. Gambar tersebut menunjukkan beberapa nilai yang tinggi pada nada C yang terdapat pada data ke-5, 8, dan 12.

Contoh gambar yang ada pada gambar 26 menunjukkan nilai-nilai atau data ke-5, 8, dan 12 tersebut akan menjadi dominan untuk setiap nada karena pada *convolutional neural network* dan memiliki pola yang sama untuk setiap ada *input* yang masuk pada mikrofon diproses oleh CNN satu dimensi dengan mencari nilai terdekat dengan data yang telah dilatih sebelumnya.

Kesimpulan

Berdasarkan hasil pengujian terhadap pengenalan nada alat musik pianika menggunakan *convolutional neural network* satu dimensi, dapat ditarik kesimpulan bahwa:

1. Sistem pengenalan nada alat musik pianika yang dibuat telah berfungsi sesuai dengan perancangan awal.
2. Pelatihan parameter-parameter diubah untuk memperoleh hasil yang optimal dan memungkinkan program bekerja secara efisien dengan mengatur *epoch*, *batch size*, *neuron fully connected layer*, dan *input flatten layer* sehingga pada saat pengujian baik

secara *real-time* sistem dapat memperoleh hasil akurasi pengenalan nada dengan tepat dan efisien dengan tingkat akurasi yang diperoleh sebesar 98,75%.

3. Pengujian *real-time* tetap menggunakan *epoch* 30, *batch size* 2, *neuron fully connected layer* 16, dan *input flatten layer* 16 tingkat pengenalan nada yang dihasilkan yaitu sebesar 91,25%.

Saran

Setiap penelitian memiliki kekurangan masing-masing, maka dari itu saran yang dapat diberikan untuk pengembangan sistem pengenalan nada alat musik pianika agar menjadi lebih baik dan berkembang adalah:

1. Disarankan untuk menggunakan *dataset* yang lebih besar guna menguji sejauh mana sistem mampu mempertahankan akurasi pengenalan nada alat musik pianika ketika jumlah masukan ke *Convolutional Neural Network* lebih sedikit.
2. Pengaruh menggunakan frekuensi *sampling* yang berbeda untuk hasil ekstraksi ciri dan akurasi pengenalan nada alat musik pianika, karena pada penelitian ini fokus pada jaringan saraf tiruan sehingga hanya memiliki satu frekuensi yaitu 4800 Hz.

Daftar Pustaka

Alwanda, M. R., Ramadhan, R. P. K., & Alamsyah, D. (2020). Implementasi Metode Convolutional Neural Network Menggunakan Arsitektur LeNet-5 untuk Pengenalan Doodle. *Jurnal Algoritme*, 1(1), 45-56.

Anhar & Putra, R. A. (2023). Perancangan dan Implementasi Self-Checkout System pada Toko Ritel menggunakan Convolutional Neural Network (CNN). *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 11(2), 466-478. Doi: https://doi.org/10.26760/elkomika.v11i2.466

Ardiansyah, R., & Sela, E. I. (2023). Implementasi Convolutional Neural Network Untuk Klasifikasi Jenis Beras Berdasarkan Citra Digital. *Indonesian Journal of Computer Science Attribution*, 12(6), 2023–4172.

Azizah, U. N. (2024, 03 Aug). Apa Itu Tangga Nada? Ini Pengertian, Ciri, Jenis, dan Contohnya. Diakses dari: https://www.detik.com/jateng/berita/d-7471743/apa-itu-tangga-nada-ini-pengertian-ciri-jenis-dan-contohnya

Brownlee, J. (2018). *What is the Difference Between a Batch and an Epoch in a Neural Network?* Dikutip dari: https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/

Budiarti, L., Nurcahyo, G. W., & Sumijan. (2024). Penerapan Metode Jaringan Syaraf Tiruan Backpropagation Untuk Memprediksi Kualitas Makanan Kucing. *Jurnal KomtekInfo*, 390–397. Doi: https://doi.org/10.35134/komtekinfo.v12i1.561

Christianto, E., Sambul, A. M., Kambey, F. D., Implementation of Convolutional Neural Network on Images for Starlings Classification. *Jurnal Teknik Informatika*. Dikutip dari: https://ejournal.unsrat.ac.id/index.php/informatika

Ferdiawan, F., & Budi Hartono. (2022). Deteksi Suara Chord Piano Menggunakan Metode Convolutional Neural Network. *Jurnal Informatika dan Rekayasa Elektronik*, 5(1), 62–68. Doi: https://doi.org/10.36595/jire.v5i1.552

Huizen, R. R., Jayanti, N. K. D. A., & Hostiadi, D. P. (2015). Analisis Pengaruh Sampling Rate Dalam Melakukan Identifikasi Pembicara Pada Rekaman Audio. *Konferensi Nasional Sistem & Informatika*. Bali: STMIK STIKOM.

Irfan, D., Rosnelly, R., Wahyuni, M., Samudra, J. T., & Rangga, A. (2022). Perbandingan Optimasi SGD, Adadelta, dan Adam Dalam Klasifikasi Hydrangea Menggunakan CNN. *Journal of Science and Social Research*, V (2), 244-253. Dikutip dari: http://jurnal.goretanpena.com/index.php/JSSR

Khairally, E. T. (2023, 05 Jan). Nada Adalah Bunyi Beraturan, Apa Jenis, Sifat, dan Contohnya?. Diambil dari https://www.detik.com/bali/berita/d-6500326/nada-adalah-bunyi-beraturan-apa-jenis-sifat-dan-contohnya

Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., & Inman, D. J. (2021). 1D convolutional neural networks and applications: A survey. *Mechanical Systems and Signal Processing*, 151. Doi: https://doi.org/10.1016/j.ymssp.2020.107398

Lu, L., Shin, Y., Su, Y., & Karniadakis, G. E. (2020). Dying ReLU and Initialization: Theory and Numerical Examples. *Global Science Preprint*. Doi: https://doi.org/10.4208/cicp.OA-2020-0165

Mahendra, Y. E., Ilyas, R., & Kasyidi, F. (2020). Klasifikasi Kalimat Ilmiah Menggunakan 1D Convolutional Neural Network. *Prosiding The 11th Industrial Research Workshop and National Seminar*. Bandung: Universitas Jenderal Achmad Yani & Institut Teknologi.

Nurlatifa, Nurhaeni, Hidayat, A., & Prasetya, M. R. A. (2024). Metode Convolutional Neural Network (CNN) Untuk Klasifikasi Tingkat Kesehatan Tanaman Lidah Buaya Berbasis Web. *Jurnal Teknik Informatika dan Sistem Informasi*, 11(4), 392-406.

Pangestu, R. A., Rahmat, B., & Anggraeny, F. T. (2020). Implementasi Algoritma CNN Untuk Klasifikasi Citra Lahan dan Perhitungan Luas. *Jurnal Informatika dan Sistem Informasi (JIFoSI)*, 1(1), 166-174.

Panjaitan, A. I. (2022). Perancangan Aplikasi Memory Card Games Dengan Menerapkan Metode Multiplicative Random Number Generation. *Journal Global Tecnology Computer*, *2*(1), 24–30.

Podder, P., Khan, T. Z., Khan, M. H., & Rahman, M. M. (2014). Comparative Performance Analysis of Hamming, Hanning and Blackman Window. Dalam *International Journal of Computer Applications*, 96(18).

Riyani, A., Nurrochman, A., Sanjaya, E., Rizqiyah, P., & Junaidi, A. (2019). Mengidentifikasi Sinyal Suara Manusia Menggunakan Metode Fast Fourier Transform (Fft) Berbasis Matlab. *Journal of Informatics, Information System, Software Engineering and Applications*, 1(2), 42–050. Doi: https://doi.org/10.20895/INISTA.V1I2

Rochmawawti, N., Hidayati, H. B., Yamasari, Y. (2021). Analisa Learning rate dan Batch size Pada Klasifikasi Covid Menggunakan Deep learning dengan Optimizer Adam. *Journal Information Engineering and Educational Technology*, 05(02). Dikutip dari: https://journal.unesa.ac.id/index.php/jieet/article/view/15213/7330

Rodiah, Susetianingtias, D. T., & Patriya, E. (2024). Identifikasi Fitur Suara Menggunakan Model Convolutional Neural Network (CNN) pada Speech-to-Text (STT). *Jurnal Pendidikan Teknologi Informasi*, *4*(3), 809–820. Doi: https://doi.org/10.51454/decode.v4i3.631

Sugianta, K. A., Gunadi, G. A., & Indrawan, G. (2020). Analisis Pola Bunyi Sunari Berdasarkan Metode Fast Fourier Transform. *Jurnal Ilmu Komputer Indonesia (JIK)*, 5(2), 2615-2703.

Sumarno, L. (2018). Pengenalan Nada Alat Musik Menggunakan Ekstraksi Ciri Perataan Segmen Berbasis DST, Dan Pengklasifikasi SVM. *Jurnal Teknologi*, 10(2), 101-109.

Suryadi, S. (2019). Implementasi Normalisasi Dalam Perancangan Database Relational. *Jurnal Teknik Informatika*, 3(2).

Sipasulta, R. Y., Lumenta, A. S. M., & Sompie, S. R. U. A. (2014). Simulasi Sistem Pengacak Sinyal Dengan Metode FFT (Fast Fourier Transform). *E-journal Teknik Elektro dan Komputer*.