

# Evaluasi Keandalan Model Rekognisi Suara Burung Hama Menggunakan Platform *Edge Impulse* Pada Mikrokontroler *Low Power*

Abdul Wahib Hasbullah<sup>1</sup>, Eko Setiawan<sup>2</sup>, Aeri Rachmad<sup>3</sup>

Program Studi Magister Pengelolaan Sumber Daya Alam Fakultas Pertanian<sup>1,2</sup>

Program Studi Teknik Informatika Fakultas Teknik<sup>3</sup>

Universitas Trunojoyo Madura

E-mail: wahib.hasbullah@trunojoyo.ac.id

**Abstrak** - Penelitian ini mengeksplorasi kemungkinan pemanfaatan teknologi *edge machine learning* dalam hal rekognisi suara-suara burung hama yang bisa diaplikasikan pada mikrokontroler *ultra low power*. Dalam paper ini dilakukan uji kehandalan dari tiga algoritma mesin pembelajaran (*Machine Learning*), kemudian menyematkannya ke mikrokontroler Seeed Xiao NRF52840 Sense. Model pembelajaran mesin yang pertama adalah *Fast Convolutional Neural Network (CNNs) 1D* dengan 2 layer, model ke-2 adalah menggunakan arsitektur berbasis *transfer learning MobileNet*. Dalam melakukan training dan testing digunakan mesin pembelajaran *embedded platform Edge Impulse*. Model pembelajaran yang dihasilkan kemudian diimplementasi sebagai *Arduino Library* sebagai representasi *32-bit floating point* dan *8-bit fixed integer*. Nilai dugaan yang dihasilkan oleh mikrokontroler dievaluasi dalam 4 kasus, yaitu menggunakan kompiler *Edge Impulse EON* dan *Tensor Flow Lite (TFLite)*. Hasil penelitian juga melaporkan *memory footprint* (RAM dan Flash), nilai akurasi, dan waktu dugaan (*time inference*).

**Kata Kunci**—deteksi suara, hama padi, burung pipit sawah, *edge impulse*, rekognisi audio, arduino, Seeed Studio NRF52840 sense.

## I. PENDAHULUAN

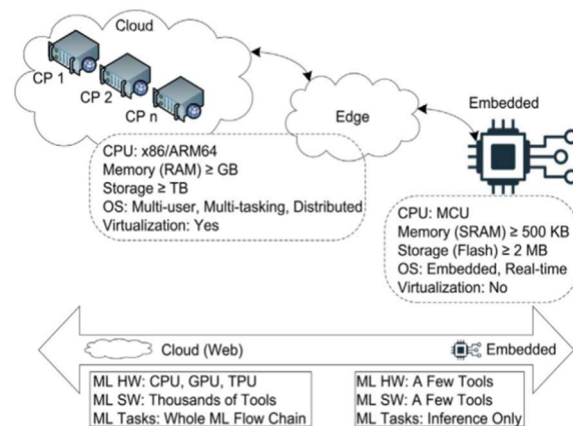
Hama burung merupakan ancaman bagi petani, karena ketika menyerang jumlahnya ratusan hingga ribuan ekor, hama burung jenis burung pipit dan burung gereja berpindah-pindah dan membentuk koloni. Dalam upaya para peneliti membuat alat yang dapat mengusir hama burung, diperlukan metode alat agar alat-alat tersebut bekerja dengan optimal seperti menggunakan sensor[1], menggunakan timer, atau teknik deteksi. Hal ini penting agar alat-alat yang dioperasikan di sawah atau ladang tersebut bekerja dengan efektif yang implikasinya pada konsumsi daya battery maupun efektifitas sumberdaya lainnya. Pada penelitian sebelumnya dikemukakan teknik deteksi dengan gerakan dan monitoring kamera [2] yang belum teruji efektif, pada penelitian lain juga pernah dibuat menggunakan teknik deteksi objek gambar yang menggunakan dua (2) buah kamera beresolusi tinggi [3] yang tentu adalah opsi yang memerlukan komputasi yang kompleks yang tentunya berimplikasi pada komsumsi battery.

Burung merupakan penanda ekologi yang sangat berguna. Keberadaannya mencerminkan tingkat keseimbangan lingkungan, namun keberadaan burung hama merupakan suatu ancaman bagi petani. Ruhnya suara burung di areal persawahan bisa dijadikan indikator

keberadaan hama burung pipit dan hal ini bisa dijadikan parameter untuk rekognisi audio.

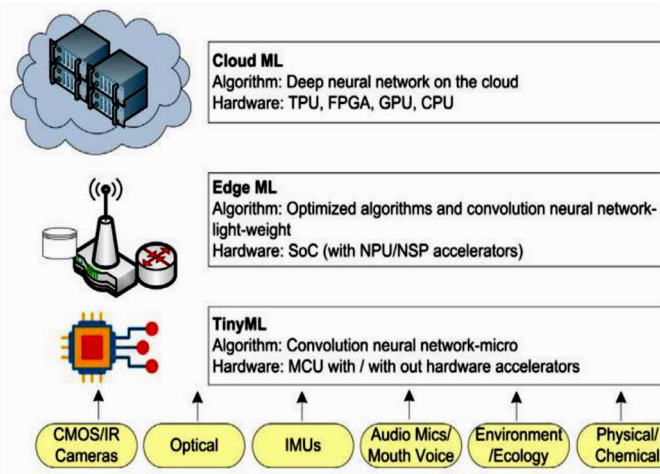
Dalam kehidupan sehari-hari, telah dikenal berbagai aplikasi kecerdasan buatan (*AI*), namun, pekerjaan seperti mengambil foto, memonitor cuaca dan sebagainya tersebut adalah model *AI* yang mahal dan memerlukan koneksi internet. Secara umum, perhitungan *Little ML* bekerja dengan cara yang sama sebagaimana kecerdasan *AI* biasa, model dipersiapkan pada *directory cloud* atau komputer klien. Gambar 1 menunjukkan kebutuhan teknologi pada *cloud ML*, *edge ML* dan *Tiny ML*[4]. Teknologinya dalam hal algoritma, perangkat keras, dan sebagainya.

Teknologi *TinyML* mengambil data dari sensor dan mengantarkan data ke tingkat mikro atau nano *Convolutional Neural Network* dimana mikrokontroler digunakan untuk menjalankan jaringan saraf. Mikrokontroler tersebut mungkin juga memiliki akselerator perangkat keras. Dalam kasus yang sangat kompleks, proses tersebut dibawa ke *deep neural network* dengan bantuan GPU, multi-core CPU dan TPU.



Gambar 1. Tiny ML berskala sangat kecil [4]

Gambar 2 menunjukkan pendekatan layer dalam pembelajaran mesin.[5]–[9]



Gambar 2. Tiny ML Dengan Pendekatan Layer [4]

Di dalam dunia AI juga telah dikembangkan model-model yang dibentuk langsung dari raw data dengan pemrosesan rendah [10], termasuk dalam klasifikasi suara burung [11]. Dari tahun 2016 dan seterusnya, keandalan model jaringan saraf konvolusional (CNN) secara konsisten mengungguli pengklasifikasi lain dalam mengklasifikasikan burung [12].

Pada penelitian ini dilakukan uji kelayakan penerapan ML pada rekognisi audio suara burung pipit dan burung hama yang kemudian akan menjadi langkah awal untuk sistem deteksi suara yang berdaya rendah dan pastinya cocok untuk ditanamkan ke mikrokontroler dengan sumber daya rendah juga.

Metodologi yang digunakan untuk melatih dan menyebarkan model audio klasifikasi suara disajikan pada bagian II. Arsitektur dan kinerja kedua model klasifikasi dikembangkan, berdasarkan konvolusi CNN 1D dan transfer learning, dibahas di bagian III. Pada bagian IV, hasil yang sesuai dengan penerapan kedua model dan analisis trade-off kinerja yang berbeda.

## II. BAHAN DAN METODE

Saat ini sangat mungkin menyematkan algoritma pembelajaran mesin pada sumber daya mikrokontroler terbatas dengan alat software dan framework. Maka pada penelitian ini digunakan platform edge impulse[13].

### 1. Edge Impulse Studio

Edge Impulse menyediakan sebuah platform tunggal bagi pengembang AI untuk mengumpulkan atau import data, memvisualisasikannya, memprosesnya sebagai persiapan pada training dan testing model pembelajaran mesin dan pada penerapan deployment dapat memilih embedded system yang akan digunakan. Edge Impulse menggunakan TensorFlow dan Keras untuk training model NN, menggunakan sklearn pada model pembelajaran klasik. Pada studio untuk developer disediakan dua pilihan untuk merancang arsitektur model : (1) interface sederhana bagi non-expert yang disajikan menggunakan lapis arsitektur visual; dan (2) expert view bagi pengembang yang membutuhkan optimasi arsitektur tingkat lanjut pada mode Keras.

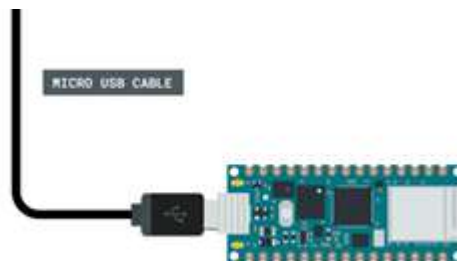
Selain itu Edge Impulse juga menyediakan Kompiler Edge Optimized Neural-nya sendiri (EON) yang bisa

menghemat 25-55% penggunaan RAM dan hingga 35% pada Flash. Sebagai perbandingan TFLite untuk mikrokontroler dilaporkan tidak berpengaruh pada akurasi model dan waktu per dugaan (time per inference) [14]. Hal ini sangat memungkinkan dikarenakan kompiler membangun model ke dalam bentuk source code, sebagai kebalikan TFLite untuk mikrokontroler yang membutuhkan interpreter untuk menjalankan model. Namun kekurangannya, dikarenakan model pada edge impulse ini telah disematkan pada firmware, sulit diubah saat diterapkan di lapangan.

Pada paper ini, dibuat 5 kelas model klasifikasi audio, kemudian ditrain pada model CNN 1D dan dua buah model transfer learning lalu disematkan pada sebuah mikrokontroler Seeed Xiao NRF52840 Sense yang memiliki mikrofon PDM internal. Keandalan dari kedua model dievaluasi dan dibandingkan pada opsi Kompiler EON dan TFLite deployment, menggunakan opsi quantized dan floating point.

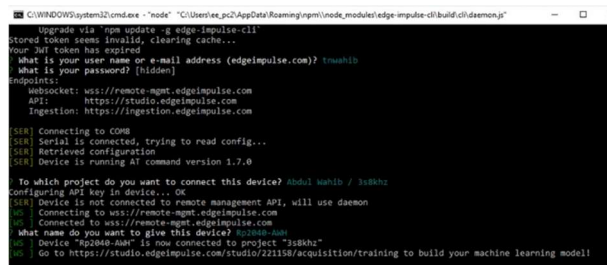
### 2. Dataset

Hardware yang digunakan saat pengambilan data sampling adalah Arduino Nano RP2040 Connect yang dikembangkan khusus pengembangan model Artificial Intelligence (AI). Didalamnya ditanamkan 32 bit dual-core ARM Cortex M0+ yang bekerja pada 133MHZ. Modul ini memiliki memori SRAM 264KB, dan memori flash 16MB chip untuk penyimpanan ekstra. RP2040 Connect juga dibekali dengan Mikrofon internal untuk keperluan pengembangan audio recognition ataupun sound activation.



Gambar 3. Arduino RP2040 Connect

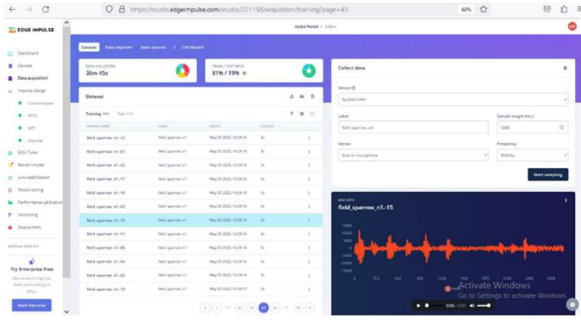
Pada saat pengambilan sampel, perangkat ini dihubungkan dengan komputer menggunakan kabel Micro USB dengan komputer yang telah terhubung dengan software edge-impulse CLI dan edge-impulse studio .



Gambar 4. Proses menghubungkan Edge Impulse CLI dengan perangkat dengan kabel USB

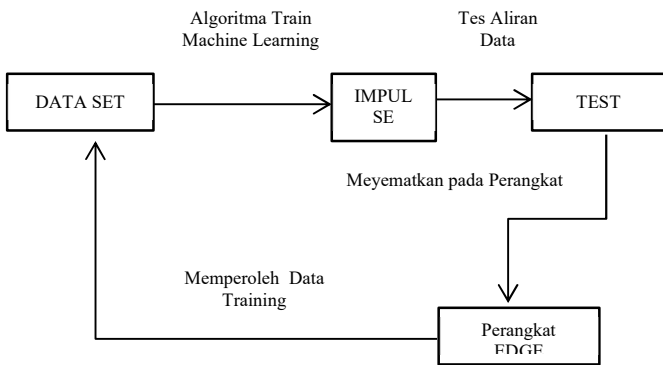
Setelah terhubung, maka sampel sudah bisa diambil, dengan memilih sensor yang digunakan, dalam penelitian ini sampel diambil dengan menggunakan sensor mikrofon internal Arduino RP2040 Connect, dengan rate 8Ksa/detik

untuk Model 1D CNN dan dengan rate 16Ksa/detik untuk pengolahann data *MobileNet*.



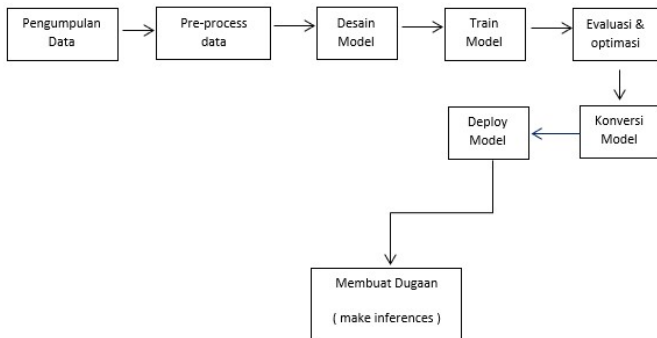
Gambar 5. Tampilan *Graphical User Interface* edge-impulse saat pengambilan sampel suara

Framework pembelajaran mesin digambarkan pada gambar 5 berikut yang mewakili model Edge Impulse.



Gambar 5. Diagram Blok Edge Impulse

Gambar 6. Blok Diagram Metode



### 3. Menyiapkan Model

Setelah informasi yang berupa rekaman suara-suara burung dan suara-suara noise didapat, kemudian akan dipersiapkan suatu model yang cocok. Pada edge- impulse studio, terdapat kemudahan dalam menangani blok secara grafis dan juga pengelompokan *learning Block Brain Keras*. Hasil dari desain disimpan pada *edge drive*. Desain model dengan memulai set up skala, channel, dan pengaturan FFT.

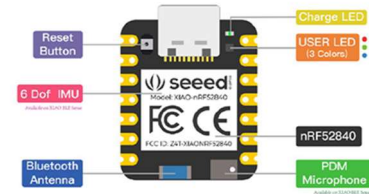
Pada tool *Edge-Impulse* proses-proses itu dimulai dari membuat set data training, data training set dihasilkan dengan proses split. Data taining dan data test dibuat pada rasio 80:20. Pre-processing data dilakukan menggunakan analisis spektral. Tahap berikutnya adalah melakukan

ekstraksi fitur pada training dataset yang mengikuti klasifikasi. Kemudian melakuksn klasifikasi menggunakan NN classifier menggunakan metode Keras. Tahap terakhir adalah melakukan pengujian model (jika hasil kurang memuaskan train ulang model) yang dilanjutkan dengan pembuatan model untuk ditanam ke Hardware (mikrokontroler).

### 4. Menjalankan Inference

Salah satu cara untuk menjalankan impuls adalah menggunakan Arduino IDE. Impulse dapat dijalankan pada library Arduino yakni berupa paket file yang berisikan semua blok *signal processing*, konfigurasi dan *learning block* ke dalam 1 paket. Berbeda halnya saat pengambilan data digunakan Mikrokontroler Raspberry *RP2040 Connect*, pada saat deployment digunakan Seeed Studio Xiao nRF52840 Sense yang juga merupakan Development Board Arduino berbasis Arm dengan RAM 256 kB dan konsumsi daya yang sangat rendah hingga 5  $\mu$ A pada mode *deep sleep*, perangkat ini juga dilengkapi 6-axis *Inertial Measurement Unit (IMU)* , PDM dan Bluetooth 5.0.

Sebelumnya pengambilan sampel menggunakan Arduino Rp2040 Connect dikarenakan mikrokontroler *Seeed Xiao NRF 52840 Sense* belum didukung oleh edge-impulse studio.



Gambar 7. Mikrokontroler *Seeed Xiao NRF52840 Sense*

## III. HASIL DAN PEMBAHASAN

### 1. Hasil Pra-Proses – Akuisisi Data

Hasil ini adalah hasil tahap segmentasi sinyal audio yaitu 735 suara sampel 8Ksa/s dan 16Ksa/s yang terdiri dari rekaman audio suara burung yang dipotong per 3 detik. Hasil dari proses segmentasi ini kemudian dibagi dalam kedua kelompok yaitu kelompok data latih (training set) dan kelompok data uji (*testing set*). Pembagian kelompok-kelompok data tersebut adalah untuk mengetahui data mana saja yang akan dijadikan data latih pada pembentukan model jaringan syaraf *Tiny ML* dan data mana saja yang akan dijadikan data uji.

Dari 735 data suara hasil segmentasi suara akan diambil 595 data suara sebagai data latih dan 118 data sisanya sebagai data uji. Komposisi pembagian data latih dan data uji dapat dilihat pada Tabel 1.

TABEL I. SUBGRUB JUMLAH DATA TRAINING DAN TESTING SAMPEL AUDIO

Kategori Kelas Audio	Training	Testing
Burung Pipit	189	36
Burung Gereja n1	65	13
Burung Gereja n2	130	33
Burung Lain	62	12
Sepi / senyap	149	24

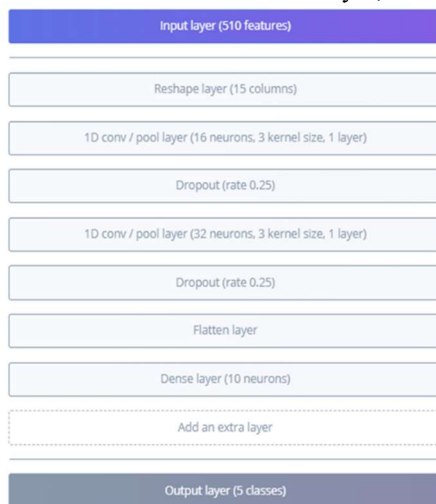
## 2. Hasil Ekstraksi Fitur

Proses ekstraksi fitur bertujuan untuk mendapatkan vektor penciri suatu objek. Prinsip dasarnya adalah mengurangi ukuran data, tanpa menghilangkan informasi penting yang menjadi ciri suatu objek. Pada penelitian ini akan digunakan koefisien MFCC 17, sehingga akan dihasilkan vektor penciri yang berukuran 17x238, 17x80, 17x166, 17x77 dan 17x174.

## 3. Pengujian Model

### A. CNN 1D

Model klasifikasi suara burung dengan menggunakan jaringan saraf konvolusi telah digunakan pada penelitian-penelitian sebelumnya dan terbukti handal [12], [15], maka pada desain model pada penelitian ini digunakan desain *custom CNN 1D 2 Layer, 10 Neuron*



Gambar 8. Pra Set Model CNN 1D

Model tersebut berisikan 2 lapis layer konvolusi 1D, Koefisien MFCC 17, jumlah Filter 32 yang diikuti oleh dropout (0.25), sebuah flatten layer, dense layer (10 neuron) sebelum output akhir 5 Neuron. Model diiterasi 100 epoch dengan learning rate 0.007.

TABEL II. CONFUSION MATRIX PADA KLASIFIKASI MENGGUNAKAN MODEL CNN 1D

Kelas Audio	B. Pipit_n1	B. Gereja_n1	B. Gereja_n2	Sepi	B. Perkutut	uncertain
B. Pipit_n1	84.3%	3.7%	0	3.7%	0	8.3%
B. Gereja_n1	0	97.6%	0%	0	0%	2.4%
B. Gereja_n2	0	5.1%	80.8%	4.0%	0	10.1%
Sepi	1.4%	5.6%	0	66.7%	2.8%	23.6%
B. Perkutut	0	0%	0%	8.3%	86.1%	5.6%
<b>f1 score</b>	0.91	0.85	0.89	0.73	0.90	

Dari confusion matrix pada tabel II dapat disimpulkan bahwa nilai akurasi dari 2 layer *CNN 1D* sudah cukup bagus, pada kelas suara Burung gereja\_n1 didapat nilai akurasi diatas 90%, namun nilai kelas kategori suara sepi atau suara noise background masih dibawah 70%.

TABEL III. CONFUSION MATRIX PADA KLASIFIKASI MENGGUNAKAN MODEL CNN 1D 10 NEURON

Kelas Audio	B. Pipit_n1	B. Gereja_n1	B. Gereja_n2	Sepi	B. Perkutut	uncertain
B. Pipit_n1	77.8%	3.7%	0%	5.6%	0.9%	12.0%
B. Gereja_n1	0%	100%	0%	0%	0%	0%
B. Gereja_n2	0%	0%	84.8%	8.1%	0%	7.1%
Sepi	0%	5.6%	0%	73.6%	1.4%	19.4%
B. Perkutut	0%	0%	0%	8.3%	88.9%	2.8%
<b>f1 score</b>	0.88	0.91	0.92	0.75	0.91	

Model *CNN 1D* yang ke-2 hampir sama dengan arsitektur yang pertama dengan penambahan 1 Layer Dense sehingga didapat Fully Connected layer. Dengan moden *CNN 1D fully connected* ini didapat hasil yang lebih baik sehingga tidak ada kelas kategori dengan nilai akurasi dibawah 70%.

### B. MobileNet

MobileNet juga telah banyak digunakan pada aplikasi suara burung di berbagai penelitian seperti A' gnes Incze dkk [16], juga Aymen Saad dkk [15]. Model ini dipilih karena ukurannya yang relatif kecil sehingga memungkinkan untuk menyematkannya pada perangkat edge. Pada penelitian ini digunakan model MobileNet V1 0.1 dan MobileNet V2 0.35 dengan input 16Ksa/detik agar optimum. Diberikan lapis dense 128 Neuron pada percobaan MobileNet V1 dan lapi dense 10 Neuron pada percobaan MobileNet V2 sebelum output akhir 5 Neuron. Model juga diiterasi 100 epoch dengan learning rate

0.007. Nilai akurasi saat validasi 68.5% dan saat testing 62.71% pada MobileNet V1, sedangkan nilai akurasi dari model MobileNet V2 81.4% saat testing 74.29%. Hasil klasifikasi diberikan pada tabel V dan tabel VI.

TABEL IV. CONFUSION MATRIX PADA KLASIFIKASI MENGGUNAKAN MODEL TRANSFER LEARNING MOBILENET V1 128 NEURON

Kelas Audio	B. Pipit_n1	B. Gereja_n1	B. Gereja_n2	Sepi	B. Perkutut	uncertain
B. Pipit_n1	46.3%	7.4%	2.8%	4.6%	0.9%	38%
B. Gereja_n1	0	51.3%	0	0	0	48.7%
B. Gereja_n2	1.0%	0	78.8%	1.0%	0	19.2%
Sepi	0	4.2%	0	65.3%	1.4%	29.2%
B. Perkutut	0	0	0	0	75%	25%
fl score	0.63	0.57	0.87	0.75	0.83	

TABEL V. CONFUSION MATRIX PADA KLASIFIKASI MENGGUNAKAN MODEL TRANSFER LEARNING MOBILENET V2 10 NEURON

Kelas Audio	B. Pipit_n1	B. Gereja_n1	B. Gereja_n2	Sepi	B. Perkutut	uncertain
B. Pipit_n1	78.7%	5.6%	0.9%	0.9%	0.9%	13.0%
B. Gereja_n1	0	82.1%	0	0	0	17.9%
B. Gereja_n2	0	0	84.8%	2.0%	0	13.1%
Sepi	1.4%	6.9%	1.4%	55.6%	1.4%	33.3%
B. Perkutut	0	0	0	2.8%	86.1%	11.1%
fl score	0.88	0.78	0.91	0.69	0.90	

Dapat dilihat dari keempat model, nilai kalisifikasi *uncertain* cenderung tinggi ( CNN 1D 19,4% dan 23,6%, MobileNet V1 48,7% , MobileNet V2 33,3% ). Masalah ini dapat diatasi dengan menyediakan lebih banyak lagi data train. Namun karena keterbatasan lisensi gratis dari Edge Impulse, nilai akurasi pada penelitian ini bukanlah tujuan utama, namun pada paper ini lebih difokuskan pada evaluasi penggunaan opsi deploymen yang disedian oleh edge-impulse. Dan model yang belum dioptimakan tersebut akan dijadikan pertimbangan terhadap ketersediaan resource yang memadai. Isu sangat penting dimana akhir-akhir ini bebarapa perusahaan semikonduktor sukses menanamkan model AI pada CPU 8-bit

#### 4. Hasil Inference of Deployed Model

Edge Impulse menyediakan dua cara penyematan ( 2 opsi deployment) : membuat library *source code* dengan pilihan bahasa pemrograman, atau dengan menanamkan

file binary yang bisa langsung dimasukkan ke dalam board sebagai firmware. Pada penelitian ini dipilih Library Arduino supaya bisa disematkan ke board Seeed Studio NRF52840 Sense. Board Seeed Studio NRF52840 Sense, dengan prosesor *ARM Cortex-M4 32-bit based processor* dengan FPU hingga 64MHz, RAM 256 kB, *1MB Flash on-chip memory*, 2MB QSPI Flash On-Board Memory dan konsumsi daya yang sangat rendah hingga 5  $\mu$ A pada mode *deep sleep*, perangkat ini juga dilengkapi 6-axis *Inertial Measurement Unit (IMU)* , PDM dan Bluetooth 5.0. Library Arduino dibuat pada setiap model (1D CNN dan MobileNet) pada kedua representasi 8-bit quantized dan 32-bit floating point. Dengan menggunakan kompiler EON atau TFLite interpreter. Hasil penelitiain diplot pada 4 nilai yang berbeda per model

Deployment pada Edge Impulse menyediakan nilai perkiraan penggunaan memory pada setiap model dan nilai perkiraan waktu per inference berdasarkan pilihan *deployment board*. Pada software ini disediakan baik model floating-point (F) dan model optimasinya (quantized Q) menggunakan EON atau TFLite interpreter. Library arduino pada setiap 4 model yang berbeda tersebut diunduh dan bisa disematkan terpisah untuk mengukur keandalan board. Hasil nilai inference untuk mikrokontroler *Seeed Xiao NRF5280 Sense* bisa dilihat pada tabel VII.

TABEL VI. HASIL INFERENCE MODEL

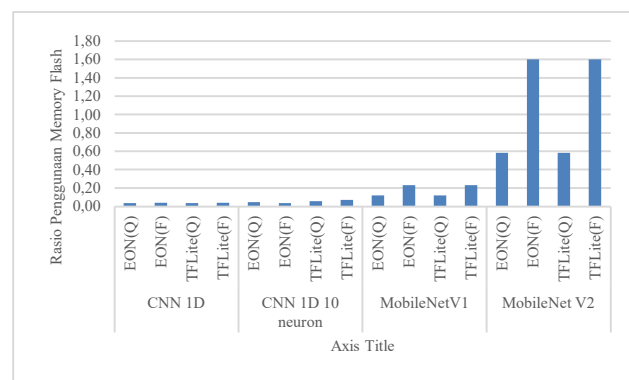
	TFLite			
	EON(Q)	EON(F)	(Q)	TFLite(F)
<b>CNN 1D</b>				
RAM (KB)	11.8 K	11.8 K	11,8 K	11,8 K
Flash (KB)	34,0 K	37,6 K	34.0 K	37.6 K
Akurasi (%)	81,51%	81,51%	84.9%	84.9%
Time (ms)	806 ms	904 ms	806 ms	904 ms
<b>CNN 1D 10 Neuron</b>				
RAM (KB)	11.8 K	11.8 K	11,8 K	11,8 K
Flash (KB)	44,1 K	35,9 K	56.6 K	71.2 K
Akurasi (%)	82.07%	82.63%	82.07%	82.63%
Time (ms)	807 ms	904 ms	807 ms	904 ms
<b>MobileNet V1 0.1 128 Neuron</b>				
RAM (KB)	37,0 K	54,9K	37,8K	70,8K
Flash (KB)	117,8K	230,9K	117,8K	230,9K
Akurasi (%)	59.04%	62.71%	68.5%	71.6%
Time (ms)	428 ms	2221 ms	428 ms	2221 ms
<b>MobileNet V2 0.35 10 Neuron</b>				
RAM (KB)	216,3K	399,9K	205,3K	488,5K
Flash (KB)	581,7K	1,6M	581,7K	1,6 M

Akurasi (%)	78.25%	77.12%	81.0%	81.4%
Time (ms)	1510 ms	8249 ms	1510 ms	8249 ms

Kinerja Sistem dengan menggunakan Pembelajaran mesin embedded sebagaimana percobaan-percobaan yang telah dilakukan dipengaruhi oleh faktor-faktor yang berbeda. Model *CNN* 1D Kustom memiliki nilai akurasi 3,5% lebih besar dibandingkan dengan MobileNet V2 untuk representasi *floating point* dan 3,9% lebih baik untuk representasi *quantized*. Beda akurasi ini biasanya mengorbankan jumlah memory footprint dan waktu dugaan. Namun pada penelitian ini pada kedua model menggunakan frekuensi sampling yang berbeda (dimana 1D CNN memungkinkan menggunakan 8Ksa/detik dan MobileNet dengan ekstraksi fitur MFE-nya mempersyaratkan minimal 16Ksa/detk) sehingga didapatkan nilai akurasi yang lebih baik dengan model 1D CNN tanpa mengorbankan memory footprint dan waktu dugaan.

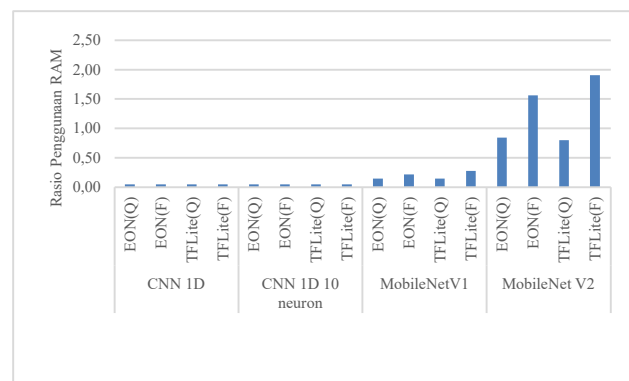
Dalam semua kasus representasi *floating point* maupun *quantized*, tidak mungkin untuk menerapkan model MobileNet V2 karena memerlukan x1,91 dan x1,56 dari RAM yang tersedia untuk model TFLite dan EON. Selain itu, ukuran memori model MobileNet V2 melebihi batas Flash memory yang tersedia pada TFLite yaitu membutuhkan 160% Memory pada *floating point* dan melebihi 50% yakni 58,17% pada *quantized* pada kedua TFLite maupun EON. Sedangkan model 1D *CNN* meskipun, dengan representasi *floating point*nya-pun masih menempati kurang dari 50% dari Flash dan RAM yang tersedia untuk kedua model EON dan model TFLite. Namun demikian, untuk penyematan yang optimal algoritma pembelajaran mesin pada mikrokontroler difokuskan pada kinerja yang model terkuantisasi. Setelah kuantisasi, ukuran model MobileNet V2 adalah dikompresi dengan faktor 2,75 pada MobileNet V1 dengan faktor 1,97 untuk TFLite dan EON. Sedangkan untuk penggunaan RAM lebih setengah resource ditempati pada MobileNet V2, dengan penurunan setelah kuantisasi sebesar faktor 1,85 untuk EON dan 2,38 untuk model TFLite, yang mana tidak memungkinkan untuk inferensi on-board, namun jika menggunakan model MobileNet V1 penggunaan RAM 49080 bytes atau sekitar 20% dari dynamic memory sehingga masi memungkinkan melakukan inference on-board. Adapun model *CNN* 1D setelah kuantisasi, kompresi tidak menurunkan nilai memori footprint dan menurunkan 0,36% penggunaan Flash memory untuk EON. Adapun model *CNN* 1D 10 neuron setelah kuantisasi, kompresi tidak menurunkan memori footprint dan justru terdapat peningkatan penggunaan Flash Memory sebesar 3,2% untuk EON, hasil ini berbanding terbalik dengan TFLite yang justru menurunkan penggunaan Flash sebanyak 1,46%. Kuantisasi model *CNN* pada penelitian tidak mengurangi memory footprint tapi mengurangi penggunaan Flash, hal ini sedikit memengaruhi keakuratan model (-0,56%) pada 1D *CNN*, MobileNet menurunkan 3,1% keakuratan setelah kuantisasi akibat penurunan footprint memory yang diperlukan. Kebutuhan *Memory footprint* dalam rasio penggunaan Flash dan RAM yang tersedia

digambarkan pada enam belas kasus berbeda divisualisasikan dalam Gambar 9 dan Gambar 10.



Gambar 9. Grafik Rasio Penggunaan Flash Memory Seed Studio NRf528410 Sense model CNN 1D dan MobileNet pada Interpreter EON dan TFLite dalam representasi *floating point* (F) dan *quantized* (Q).

Dari grafik 10 penggunaan interpreter EON atau TFLite untuk penggunaan RAM terdapat perbedaan yang nyata untuk model MobileNetV1 dan MobileNetV2, dimana pada MobileNetV1 menurunkan 6,25% penggunaan RAM pada representasi *floating point* dan pada model MobileNetV2 saat representasi *floating point* menurunkan 34,61% penggunaan RAM, namun saat kuantisasi tidak ada penurunan penggunaan RAM.



Gambar 10. Grafik Rasio Penggunaan RAM Seed Studio NRf528410 Sense model CNN 1D dan MobileNet pada Interpreter EON dan TFLite dalam representasi *floating point* (F) dan *quantized* (Q).

Untuk penggunaan interpreter EON atau TFLite, penerapan modelnya dan inferensi on-board, terlihat jelas EON membutuhkan lebih sedikit memori footprint dibandingkan TFLite. Namun, kekurangan menggunakan EON adalah sulitnya mengupdate model, berbeda dengan TFLite. Penggunaan EON tidak mempengaruhi akurasi atau waktu inferensi karena masih menjalankan TFLite di latar belakang kompilernya [14]. Untuk kedua model, akurasinya berbeda dimana nilai akurasi model CNN 1D cenderung lebih baik, sedangkan waktu tidak menunjukkan perbedaan antara model terkuantisasi EON dan TFLite.

Pada model MobileNetV1, terdapat perbedaan sebesar 378 ms, dengan MobileNetV1 memiliki waktu inferensi yang lebih cepat dibandingkan CNN.

#### IV. KESIMPULAN

Dari hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa telah dibuat suatu model kustom CNN 1D menggunakan edge impulse dengan frekuensi audio sampel 8Ksa/detik dengan ekstraksi ciri MFCC. Klasifikasi suara-suara burung hama padi yang dilakukan dengan metode kustom menggunakan embedded machine learning ini menghasilkan suatu rekognisi suara yang memungkinkan ditanam pada mikrokontroler yang sangat kecil (*Tiny ML*) yaitu *Seed Xiao NRF52840 Sense* yang memiliki konsumsi daya yang sangat kecil pula. Metode CNN 1D 2 layer memberikan keakuratan rekognisi suara 82,63% dengan penggunaan memori Flash 35,9 Kbyte dengan *Peak* RAM 11.8 Kbyte. Jika menggunakan metode MobileNet V2 untuk mendapatkan nilai keakuratan yang hampir sama dibutuhkan ukuran memori Flash yang 16x lebih besar dan ukuran RAM yang sekitar 17x lebih besar akibat ekstraksi ciri MFE-nya mempersyaratkan frekuensi sampel yang besar yaitu minimal 16Ksa/detik sehingga tidak mungkin ditanamkan pada board *Seed Xiao* ini.

#### DAFTAR PUSTAKA

- [1] M. S. Bana, D. Rahmawati, K. Joni, and M. Ulum, "Rancang Bangun Alat Pengusir Tikus dan Burung pada Tanaman Padi," *J-Eltrik*, vol. 2, no. 1, p. 53, Nov. 2021, doi: <https://doi.org/10.30649/j-eltrik.v2i1.53>.
- [2] S. F. Nabilah, R. Agustin, and F. N. Fauziah, "Sistem Monitoring Pengusir Hama Burung Pada Tanaman Padi Menggunakan Sensor PIR dan ESPCamera Berbasis Internet Of Thing," *Politeknik Harapan Tegal*, no. Tugas Akhir Teknik Komputer, pp. 1–9, 2020.
- [3] A. Muminov, Y. C. Jeon, D. Na, C. Lee, and H. S. Jeon, "Development of a solar powered bird repeller system with effective bird scarer sounds," *International Conference on Information Science and Communications Technologies, ICISCT 2017*, vol. 2017-Decem, pp. 1–4, 2017, doi: [10.1109/ICISCT.2017.8188587](https://doi.org/10.1109/ICISCT.2017.8188587).
- [4] R. R. Prasanna, P. Chowdary Kakarla, V. P. Simha, and N. Mohan, "IMPLEMENTATION OF TINY MACHINE LEARNING MODELS ON ARDUINO 33-BLE FOR GESTURE AND SPEECH RECOGNITION," *Electrical Engineering and Systems Science*, no. 2207, 2022, doi: <https://doi.org/10.48550/arXiv.2207.12866>.
- [5] W. Yu and C. Zhao, "Broad Convolutional Neural Network Based Industrial Process Fault Diagnosis with Incremental Learning Capability," *IEEE*, vol. 67, no. 6, pp. 5081–5091, 2020, doi: <https://doi.org/10.1109/TIE.2019.2931255>.
- [6] J. Song, "7.1 An 11.5TOPS/W 1024-MAC butterfly structure dualcore sparsity-aware neural processing unit in 8nm flagship mobile SoC," in *2019 IEEE International Solid-State Circuits Conference - (ISSCC)*, IEEE, 2019, doi: <https://doi.org/10.1109/ISSCC.2019.8662476>.
- [7] B. Li, M. H. Najafi, B. Yuan, and D. J. Lilja, "Quantized neural networks with new stochastic multipliers," in *2018 19th International Symposium on Quality Electronic Design (ISQED)*, IEEE, 2018, doi: <https://doi.org/10.1109/ISQED.2018.8357316>.
- [8] M. H. Najafi, D. J. Lilja, and M. Riedel, "Deterministic methods for stochastic computing using low-discrepancy sequences," in *IEEE, 2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2018, doi: <https://doi.org/10.1145/3240765.3240797>.
- [9] L. Yang, W. Jiang, W. Liu, and H. Edwin, "Co-exploring neural architecture and network-on-chip design for real-time artificial intelligence," in *IEEE, 2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2020, doi: <https://doi.org/10.1109/ASP-DAC47756.2020.9045595>.
- [10] W. Dai, C. Dai, S. Qu, J. Li, and S. Das, "07952190," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pp. 421–425, 2017.
- [11] A. Saad, J. Ahmed, and A. Elaraby, "Classification of Bird Sound Using High-and Low-Complexity Convolutional Neural Networks," *Traitement du Signal*, vol. 39, no. 1, pp. 187–193, 2022, doi: [10.18280/ts.390119](https://doi.org/10.18280/ts.390119).
- [12] H. Goëau, H. Glotin, W. P. Vellinga, Planqué, and A. R., Joly, "Lifeclef bird identification task 2016: The arrival of deep learning," in *CLEF: Conference and Labs of the Evaluation Forum, 1609*, <https://hal.archives-ouvertes.fr/hal-01373779>, 2016, pp. 440–449.
- [13] "Edge Impulse." Accessed: Sep. 18, 2023. [Online]. Available: <https://www.edgeimpulse.com/>
- [14] J. Jongboom, "Introducing EON: Neural networks in up to 55% less RAM and 35% less ROM," Sep. 2020. [Online]. Available: <https://www.edgeimpulse.com/blog/introducing-eon>
- [15] A. Saad, J. Ahmed, and A. Elaraby, "Classification of Bird Sound Using High-and Low-Complexity Convolutional Neural Networks," *Traitement du Signal*, vol. 39, no. 1, pp. 187–193, 2022, doi: <https://doi.org/10.18280/ts.390119>.
- [16] Institute of Electrical and Electronics Engineers, "Bird Sound Recognition Using a Convolutional Neural Network," in *SISY 2018 • IEEE 16th International Symposium on Intelligent Systems and Informatics*, 2018 IEEE 16th International Symposium on Intelligent Systems and Informatics (SISY), Sep. 2018, doi: <https://doi.org/10.1109/SISY.2018.8524677>.