Daftar Pustaka

Ahmad, K. A. (2021). *Evaluasi Kinerja Penerapan Multi-Channel Transfer Learning CNN pada Klasifikasi Citra Penyakit Daun Padi*. Makassar.

Aulia, S., & Hadiyoso, S. (2022). Tuberculosis Detection in X-Ray Image Using Deep Learning Approach with VGG-16 Architecture. *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, *8*(2), 290. Universitas Ahmad Dahlan.

Azmi, K., Defit, S., & Sumijan. (2023). Implementasi Convolutional Neural Network (CNN) Untuk Klasifikasi Batik Tanah Liat Sumatera Barat. *Jurnal Unitek*, *16*(1).

Briceno, R. K., Sergent, S. R., Benites, S. M., & Alocilja, E. C. (2019). Nanoparticle-based biosensing assay for universally accessible low-cost tb detection with comparable sensitivity as culture. *Diagnostics*, *9*(4). Multidisciplinary Digital Publishing Institute (MDPI).

Chakrabarty, N. (2019). A novel strategy for gender identification from hand dorsal images using computer vision. *Proceedings of the 3rd International Conference on Computing Methodologies and Communication, ICCMC 2019* (pp. 108–113). Institute of Electrical and Electronics Engineers Inc.

Clara, S., Laksmi Prianto, D., Al Habsi, R., Friscila Lumbantobing, E., Chamidah, N., Kom, S., Kom, M., et al. (2021). *Implementasi Seleksi Fitur Pada Algoritma Klasifikasi Machine Learning Untuk Prediksi Penghasilan Pada Adult Income Dataset*. *Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA) Jakarta-Indonesia*.

Do, S., Song, K. D., & Chung, J. W. (2020, January 1). Basics of deep learning: A radiologist's guide to understanding published radiology articles on deep learning. *Korean Journal of Radiology*. Korean Radiological Society.

Ezzat, D., Hassanien, A., & Ella, H. A. (2020). GSA-DenseNet121-COVID-19: a Hybrid Deep Learning Architecture for the Diagnosis of COVID-19 Disease based on Gravitational Search Optimization Algorithm. *ArXiv*, *abs/2004.05084*.

Ghosh, A., Sufian, A., Sultana, F., Chakrabarti, A., & De, D. (2019). Fundamental concepts of convolutional neural network. *Intelligent Systems Reference Library* (Vol. 172, pp. 519–567). Springer.

Global Tuberculosis Report 2023. (2023). Retrieved from https://iris.who.int/.

Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., & Lew, M. S. (2016). Deep learning for visual understanding: A review. *Neurocomputing*, *187*, 27–48. Elsevier B.V.

Gupta, J., Pathak, S., & Kumar, G. (2022). Deep Learning (CNN) and Transfer Learning: A Review. *Journal of Physics: Conference Series* (Vol. 2273). Institute of Physics.

Han, J., Kamber, M., & Pei, J. (2012). *Data Mining. Concepts and Techniques; The Morgan Kaufmann Series in Data Management Systems* (3rd ed.). Elsevier Inc.

Han, X., Zhang, Z., Ding, N., Gu, Y., Liu, X., Huo, Y., Qiu, J., et al. (2021). Pre-trained models: Past, present and future. *AI Open*, *2*, 225–250. Elsevier B.V.

Hartato, B. P. (2021). Penerapan Convolutional Neural Network pada Citra Rontgen Paru-Paru untuk Deteksi SARS-CoV-2. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, *5*(4), 747–759. Ikatan Ahli Informatika Indonesia (IAII).

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., et al. (2019). Searching for MobileNetV3. Retrieved from http://arxiv.org/abs/1905.02244

Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K. Q. (2016). Densely Connected Convolutional Networks. Retrieved from http://arxiv.org/abs/1608.06993

Iqbal, S., Qureshi, A. N., Ullah, A., Li, J., & Mahmood, T. (2022). Improving the Robustness and Quality of Biomedical CNN Models through Adaptive *Hyperparameter Tuning*. *Applied Sciences (Switzerland)*, *12*(22). MDPI.

Kamal, M. R. M., Shahbudin, S., & Rahman, F. Y. A. (2023). Photovoltaic (PV) Module Defect Image Classification Analysis Using EfficientNetV2 Architectures. *2023 IEEE 14th Control and System Graduate Research Colloquium (ICSGRC)*, 236–241.

Kesse, M. A., Buah, E., Handroos, H., & Ayetor, G. K. (2020). Development of an artificial intelligence powered tig welding algorithm for the prediction of bead geometry for tig welding processes using hybrid deep learning. *Metals*, *10*(4). MDPI AG.

Khan, S., Rahmani, H., Shah, S., & Bennamoun, M. (2018). A Guide to Convolutional Neural Networks for Computer Vision. *Synthesis Lectures on Computer Vision*, *8*, 1–207.

Kotei, E., & Thirunavukarasu, R. (2022). Ensemble Technique Coupled with Deep Transfer Learning Framework for Automatic Detection of

Tuberculosis from Chest X-ray Radiographs. *Healthcare (Switzerland)*, *10*(11). MDPI.

Kuzlu, M., Catak, F. O., Sarp, S., Cali, U., & Gueler, O. (2022). *A Streamlit-based Artificial Intelligence Trust Platform for Next-Generation Wireless Networks*. Retrieved from https://github.com/muratkuzlu/NextG

Lecun, Y., Bengio, Y., & Hinton, G. (2015, May 27). Deep learning. *Nature*. Nature Publishing Group.

Lu, D., Chen, C., Yu, S., & Chen, S. (2016). Diagnosis of tuberculous meningitis using a combination of peripheral blood T-SPOT.TB and cerebrospinal fluid interferon-c detection methods. *Lab Medicine*, *47*(1), 6–12. Oxford University Press.

Luger, G. F., & Stubblefield, W. A. (1993). *Artificial Intelligence: Structures and Strategies for Complex Problem Solving* (2th Edition.). Michigan: Benjamin/Cummings Publishing Company. Retrieved July 15, 2024, from https://books.google.co.id/books/about/Artificial_Intelligence.html?id=UHVQAAAAMAAJ&redir_esc=y

Manaswi, N. (2018). Basics of TensorFlow, 1–30.

Mien, T. L., Van An, V., & Huong, N. T. T. (2022). Research and Development of the Pupil Identification and Warning System using AI-IoT. *Journal of Robotics and Control (JRC)*, *3*(4), 528–534. Department of Agribusiness, Universitas Muhammadiyah Yogyakarta.

Mubin, R. A. (2021). *Implementasi Deep Learning (Convolutional Neural Network) Untuk Mendeteksi Pemakaian Masker Secara Real-Time Video Stream*. Makassar.

Nápoles-Duarte, J. M., Biswas, A., Parker, M. I., Palomares-Baez, J. P., Chávez-Rojo, M. A., & Rodríguez-Valdez, L. M. (2022). Stmol: A component for building interactive molecular visualizations within streamlit web-applications. *Frontiers in Molecular Biosciences*, *9*. Frontiers Media S.A.

Pahlevi, S. M. (2023). *Kecerdasan Buatan dengan Deep Learning*. Elex Media Komputindo.

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*.

Pardede, J., & Purohita, A. S. (2023). *The Advantage of Transfer Learning with Pre-Trained Model in CNN Towards Ct-Scan Classification* (Vol. 9).

Rahman, T., Khandakar, A., Kadir, M. A., Islam, K. R., Islam, K. F., Mazhar, R., Hamid, T., et al. (2020). Reliable tuberculosis detection using chest

X-ray with deep learning, segmentation and visualization. *IEEE Access*, *8*, 191586–191601. Institute of Electrical and Electronics Engineers Inc.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. Retrieved from http://arxiv.org/abs/1801.04381

Santoso, A. (2018). *Implementasi Deep Learning Berbasis Keras untuk Pengenalan Wajah*.

Satheeshkumar, K. G., & Arunachalam, V. (2020). Automatic Detection of Tuberculosis from Chest X-Rays using Convolutional Neural Network. *International Journal of Engineering and Advanced Technology*, *9*(5), 72–77. Retrieved from https://www.ijeat.org/wp-content/uploads/papers/v9i5/E9292069520.pdf

Sejati, A., & Sofiana, L. (2015). *Faktor-faktor Terjadinya Tuberkulosis*. *KEMAS* (Vol. 10). Retrieved from http://journal.unnes.ac.id/nju/index.php/kemas

Tan, M., Chen, B., Pang, R., Vasudevan, V., & Le, Q. V. (2018). MnasNet: Platform-Aware Neural Architecture Search for Mobile. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2815–2823.

Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. Retrieved from http://arxiv.org/abs/1905.11946

Tan, M., & Le, Q. V. (2021). EfficientNetV2: Smaller Models and Faster Training. Retrieved from http://arxiv.org/abs/2104.00298

Teo, Y., Shin, S., Jeong, H., Kim, Y., Kim, Y.-H., Struchalin, G., Kovlakov, E., et al. (2021). Benchmarking quantum tomography completeness and fidelity with machine learning. *New Journal of Physics*, *23*.

Uplekar, M., Weil, D., Lonnroth, K., Jaramillo, E., Lienhardt, C., Dias, H. M., Falzon, D., et al. (2015, May 2). WHO's new end TB strategy. *The Lancet*. Lancet Publishing Group.

Xin, M., & Wang, Y. (2019). Research on image classification model based on deep convolution neural network. *Eurasip Journal on Image and Video Processing*, *2019*(1). Springer International Publishing.

Zhao, J., He, X., & Kemao, Q. (2019). Automatic Body Measurement by Neural Networks. *Proceedings of the 2019 2nd International Conference on Robot Systems and Applications*. Retrieved from https://api.semanticscholar.org/CorpusID:181852822

Zoph, B., Vasudevan, V., Shlens, J., & Le, Q. V. (2017). Learning Transferable Architectures for Scalable Image Recognition. Retrieved from http://arxiv.org/abs/1707.07012

Zufar, M., Setiyono, B., & Matematika, J. (2016). *Convolutional Neural Networks untuk Pengenalan Wajah Secara Real-Time* (Vol. 5).

Lampiran

Lampiran 1. *Log Training Model*

EfficientNetV2B0

```
Epoch 1/10
46/46 ━━━━━━━━━━━━━━━━━━━━ 96s 2s/step - accuracy: 0.8682 - loss: 7.6445 -
val_accuracy: 0.9794 - val_loss: 2.3710
Epoch 2/10
46/46 ━━━━━━━━━━━━━━━━━━━━ 76s 2s/step - accuracy: 0.9910 - loss: 1.8386 -
val_accuracy: 0.9937 - val_loss: 0.9087
Epoch 3/10
46/46 ━━━━━━━━━━━━━━━━━━━━ 76s 2s/step - accuracy: 0.9862 - loss: 0.8107 -
val_accuracy: 0.9825 - val_loss: 0.6142
Epoch 4/10
46/46 ━━━━━━━━━━━━━━━━━━━━ 77s 2s/step - accuracy: 0.9882 - loss: 0.5970 -
val_accuracy: 0.9889 - val_loss: 0.4540
Epoch 5/10
46/46 ━━━━━━━━━━━━━━━━━━━━ 76s 2s/step - accuracy: 0.9777 - loss: 0.4875 -
val_accuracy: 0.9825 - val_loss: 0.4101
Epoch 6/10
46/46 ━━━━━━━━━━━━━━━━━━━━ 76s 2s/step - accuracy: 0.9919 - loss: 0.3702 -
val_accuracy: 0.9937 - val_loss: 0.3012
Epoch 7/10
46/46 ━━━━━━━━━━━━━━━━━━━━ 80s 2s/step - accuracy: 0.9973 - loss: 0.2778 -
val_accuracy: 0.9968 - val_loss: 0.2498
Epoch 8/10
46/46 ━━━━━━━━━━━━━━━━━━━━ 82s 2s/step - accuracy: 0.9844 - loss: 0.3053 -
val_accuracy: 0.9921 - val_loss: 0.3242
Epoch 9/10
46/46 ━━━━━━━━━━━━━━━━━━━━ 74s 2s/step - accuracy: 0.9894 - loss: 0.3127 -
val_accuracy: 0.9841 - val_loss: 0.2906
Epoch 10/10
46/46 ━━━━━━━━━━━━━━━━━━━━ 73s 2s/step - accuracy: 0.9932 - loss: 0.2638 -
val_accuracy: 0.9905 - val_loss: 0.2167
```

MobileNetV3

```
Epoch 1/10
46/46 ━━━━━━━━━━━━━━━━━━━━ 69s 1s/step - accuracy: 0.8082 - loss: 7.3196 -
val_accuracy: 0.9905 - val_loss: 3.2173
Epoch 2/10
46/46 ━━━━━━━━━━━━━━━━━━━━ 62s 1s/step - accuracy: 0.9875 - loss: 2.8177 -
val_accuracy: 0.9889 - val_loss: 1.9202
Epoch 3/10
46/46 ━━━━━━━━━━━━━━━━━━━━ 61s 1s/step - accuracy: 0.9929 - loss: 1.7307 -
val_accuracy: 0.9889 - val_loss: 1.3210
Epoch 4/10
46/46 ━━━━━━━━━━━━━━━━━━━━ 61s 1s/step - accuracy: 0.9885 - loss: 1.2366 -
val_accuracy: 0.9857 - val_loss: 1.0070
Epoch 5/10
46/46 ━━━━━━━━━━━━━━━━━━━━ 60s 1s/step - accuracy: 0.9904 - loss: 0.9431 -
```

```
val_accuracy: 0.9937 - val_loss: 0.7909
Epoch 6/10
46/46 ━━━━━━━━━━━━━━━━━━━━ 60s 1s/step - accuracy: 0.9941 - loss: 0.7459 -
val_accuracy: 0.9921 - val_loss: 0.6939
Epoch 7/10
46/46 ━━━━━━━━━━━━━━━━━━━━ 60s 1s/step - accuracy: 0.9950 - loss: 0.6565 -
val_accuracy: 0.9984 - val_loss: 0.5279
Epoch 8/10
46/46 ━━━━━━━━━━━━━━━━━━━━ 60s 1s/step - accuracy: 0.9958 - loss: 0.5108 -
val_accuracy: 0.9905 - val_loss: 0.4772
Epoch 9/10
46/46 ━━━━━━━━━━━━━━━━━━━━ 61s 1s/step - accuracy: 0.9918 - loss: 0.4655 -
val_accuracy: 0.9889 - val_loss: 0.4265
Epoch 10/10
46/46 ━━━━━━━━━━━━━━━━━━━━ 60s 1s/step - accuracy: 0.9945 - loss: 0.3940 -
val_accuracy: 0.9810 - val_loss: 0.3969
```

NASNetMobile

```
Epoch 1/10
46/46 ━━━━━━━━━━━━━━━━━━━━ 116s 2s/step - accuracy: 0.7891 - loss: 7.5881 -
val_accuracy: 0.9159 - val_loss: 1.5261
Epoch 2/10
46/46 ━━━━━━━━━━━━━━━━━━━━ 87s 2s/step - accuracy: 0.9070 - loss: 1.6069 -
val_accuracy: 0.9254 - val_loss: 1.2891
Epoch 3/10
46/46 ━━━━━━━━━━━━━━━━━━━━ 88s 2s/step - accuracy: 0.8926 - loss: 1.4356 -
val_accuracy: 0.9429 - val_loss: 0.9096
Epoch 4/10
46/46 ━━━━━━━━━━━━━━━━━━━━ 86s 2s/step - accuracy: 0.8874 - loss: 1.5890 -
val_accuracy: 0.9143 - val_loss: 1.1134
Epoch 5/10
46/46 ━━━━━━━━━━━━━━━━━━━━ 84s 2s/step - accuracy: 0.8847 - loss: 1.9920 -
val_accuracy: 0.8889 - val_loss: 1.5435
Epoch 6/10
46/46 ━━━━━━━━━━━━━━━━━━━━ 83s 2s/step - accuracy: 0.9158 - loss: 1.0672 -
val_accuracy: 0.9476 - val_loss: 0.7714
Epoch 7/10
46/46 ━━━━━━━━━━━━━━━━━━━━ 83s 2s/step - accuracy: 0.9381 - loss: 0.8437 -
val_accuracy: 0.9429 - val_loss: 0.7372
Epoch 8/10
46/46 ━━━━━━━━━━━━━━━━━━━━ 84s 2s/step - accuracy: 0.9279 - loss: 0.8153 -
val_accuracy: 0.9635 - val_loss: 0.5899
Epoch 9/10
46/46 ━━━━━━━━━━━━━━━━━━━━ 90s 2s/step - accuracy: 0.9495 - loss: 0.6060 -
val_accuracy: 0.9619 - val_loss: 0.5588
Epoch 10/10
46/46 ━━━━━━━━━━━━━━━━━━━━ 133s 2s/step - accuracy: 0.9491 - loss: 0.5718 -
val_accuracy: 0.9492 - val_loss: 0.5876
```

DenseNet121

```
Epoch 1/10
46/46 ━━━━━━━━━━━━━━━━━━━━ 241s 5s/step - accuracy: 0.8102 - loss: 27.3806 -
val_accuracy: 0.9746 - val_loss: 4.4586
Epoch 2/10
46/46 ━━━━━━━━━━━━━━━━━━━━ 219s 5s/step - accuracy: 0.9627 - loss: 4.4634 -
val_accuracy: 0.9778 - val_loss: 3.2087
Epoch 3/10
46/46 ━━━━━━━━━━━━━━━━━━━━ 219s 5s/step - accuracy: 0.9590 - loss: 3.7641 -
val_accuracy: 0.9810 - val_loss: 2.6573
Epoch 4/10
46/46 ━━━━━━━━━━━━━━━━━━━━ 196s 4s/step - accuracy: 0.9618 - loss: 3.6503 -
val_accuracy: 0.9810 - val_loss: 2.6091
Epoch 5/10
46/46 ━━━━━━━━━━━━━━━━━━━━ 196s 4s/step - accuracy: 0.9834 - loss: 2.5750 -
val_accuracy: 0.9746 - val_loss: 2.5057
Epoch 6/10
46/46 ━━━━━━━━━━━━━━━━━━━━ 197s 4s/step - accuracy: 0.9817 - loss: 1.9752 -
val_accuracy: 0.9921 - val_loss: 1.6391
Epoch 7/10
46/46 ━━━━━━━━━━━━━━━━━━━━ 198s 4s/step - accuracy: 0.9793 - loss: 1.8167 -
val_accuracy: 0.9810 - val_loss: 1.8012
Epoch 8/10
46/46 ━━━━━━━━━━━━━━━━━━━━ 198s 4s/step - accuracy: 0.9858 - loss: 1.6830 -
val_accuracy: 0.9698 - val_loss: 1.9331
Epoch 9/10
46/46 ━━━━━━━━━━━━━━━━━━━━ 203s 4s/step - accuracy: 0.9768 - loss: 1.7394 -
val_accuracy: 0.9889 - val_loss: 1.4039
Epoch 10/10
46/46 ━━━━━━━━━━━━━━━━━━━━ 195s 4s/step - accuracy: 0.9845 - loss: 1.4578 -
val_accuracy: 0.9825 - val_loss: 1.3594
```

Lampiran 2. *Source code* implementasi model

```
import streamlit as st
from PIL import Image, ImageChops
import numpy as np
from tensorflow.keras.applications import mobilenet_v3
from tensorflow.keras.models import load_model
from huggingface_hub import hf_hub_download

preprocess_input = mobilenet_v3.preprocess_input

# Set page configuration
st.set_page_config(
    page_title="Klasifikasi Tuberkulosis",
    page_icon="🩺",
    layout="centered",
)

# Load your trained model
@st.cache_resource
def load_my_model():
    # Download the MobileNetV3 model from the Hugging Face Hub
    model_path = hf_hub_download(repo_id="reaim70/MobileNetV3.keras",
filename="MobileNetV3.keras")
    model = load_model(model_path)
    return model

model = load_my_model()

# Fungsi prediksi menggunakan model yang dimuat
def predict(image):
    # Preprocess the image to match the input shape the model expects
    img = image.resize((224, 224))  # Ganti ukuran sesuai dengan input model Anda
    img_array = np.array(img)

    # Ensure the image has 3 channels (RGB)
    if img_array.shape[-1] == 3:  # Already RGB
        img_array = preprocess_input(img_array)
    else:  # Grayscale or other format
        img_array = np.stack((img_array,) * 3, axis=-1)  # Convert to RGB
        img_array = preprocess_input(img_array)

    img_array = np.expand_dims(img_array, axis=0)  # Add batch dimension
    prediction = model.predict(img_array)

    # Assuming model output is probability, choose the class with highest
probability
    if prediction[0][0] > prediction[0][1]:
        return "Normal"
    else:
        return "Tuberkulosis"

def is_chest_xray(image):
```

```
    # Check image dimensions and aspect ratio
    width, height = image.size
    aspect_ratio = width / height

    # Typical chest X-ray aspect ratio is approximately 0.7 - 1.3
    if not (0.7 < aspect_ratio < 1.3):
        return False

    # Convert the image to grayscale and check if it's nearly grayscale
    grayscale_image = image.convert("L")
    diff = ImageChops.difference(image.convert("RGB"),
grayscale_image.convert("RGB"))

    # Check the difference (whether the image is close to grayscale)
    diff_stat = diff.getbbox()

    # If there is no bounding box (None), the image is already grayscale
    if diff_stat is None:
        return True

    return False

# Title of the web app
st.title("Klasifikasi Tuberkulosis")

# Center the icon representation
st.markdown(
    """
    <div style='text-align: center; margin-bottom: 20px;'>
        <i class="fas fa-lungs" style='font-size: 100px; color: grey;'></i>
    </div>
    """, unsafe_allow_html=True
)

# Container for upload and results
with st.container():
    # File uploader
    uploaded_file = st.file_uploader("Upload Gambar Rontgen Dada", type=["jpg",
"jpeg", "png"])

    if uploaded_file is not None:
        image = Image.open(uploaded_file)

        # Columns for image preview and detection button
        col1, col2 = st.columns([1, 1])
        with col1:
            st.image(image, caption='Gambar yang diupload',
use_column_width=True)

        with col2:
            if st.button("Klasifikasi"):
                # Display loading message
                with st.spinner('Mengklasifikasi...'):
```

```
# Check if the uploaded image is likely a chest X-ray
if not is_chest_xray(image):
    st.warning("Bukan Citra Rontgen Dada")
else:
    # Prediction using the loaded model
    result = predict(image)

    # Display result below the button
    st.subheader("Hasil Klasifikasi")
    if result == "Normal":
        st.success("Terklasifikasi Normal")
    else:
        st.error("Terklasifikasi Tuberkulosis")
```

Lampiran 1. Riwayat Hidup

Nama	: Muhammad Alim Ma'arij
Tempat / Tanggal Lahir	: Makassar / 02 April 2002
Jenis Kelaminn	: Laki-Laki
Agama	: Islam
Suku	: Bugis-Makassar
Alamat	: Jl. Tamangapa Raya 3, Komplek Pesona Prima Griya A1/2
No.Hp	: 087886131448
E-mail	: alimmaarij@gmail.com
Riwayat Pendidikan	: 1. SD Inpres BTN Pemda 2. SD Negeri Cipayung 2 3. SMP Muhammadiyah 22 Setiabudi Pamulang 4. MAN 1 Makassar 5. SMA Negeri 3 Makassar 6. Universitas Hasanuddin – Sistem Informasi (S1)