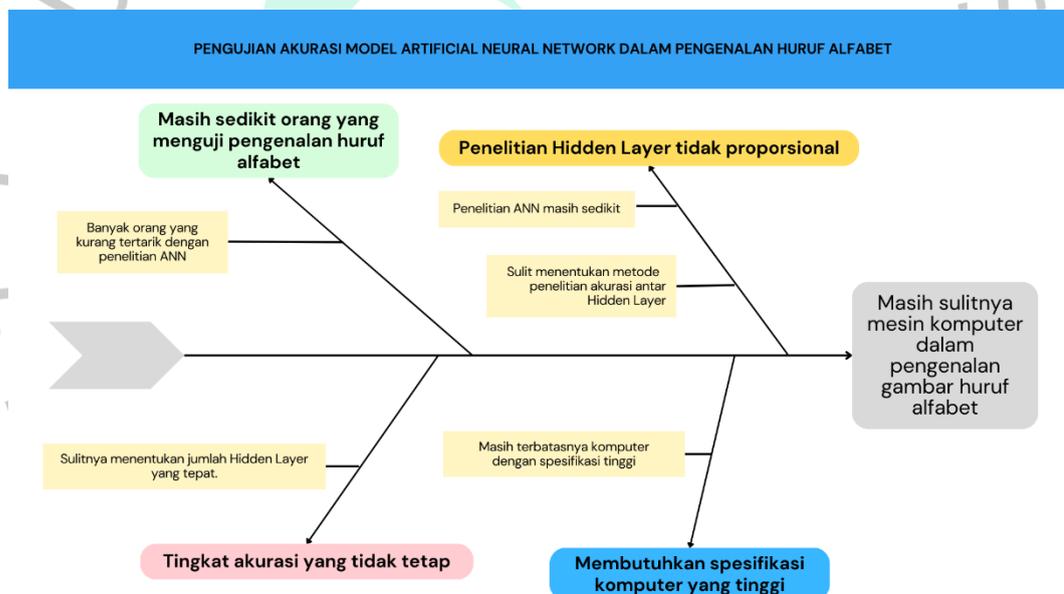


BAB III

Metode Penelitian

Pada bab III ini, akan dijelaskan mengenai metode penelitian yang digunakan dalam pengujian akurasi model *Artificial Neural Networks* (ANN) dengan satu dan dua lapisan tersembunyi dalam pengenalan huruf alfabet. Penjelasan mengenai tahapan-tahapan yang dilakukan dalam penelitian, termasuk pemilihan dataset, *preprocessing* data, pemilihan parameter ANN, serta pengukuran akurasi model akan diuraikan secara rinci.

3.1 Paradigma Penelitian



Gambar 3. 1 Fishbone Diagram

Fishbone diagram di atas menggambarkan paradigma penelitian untuk penelitian yang bertujuan untuk menguji akurasi model *Artificial Neural Networks* (ANN) dengan satu dan dua lapisan tersembunyi dalam pengenalan huruf alfabet. Kepala *fishbone* adalah tujuan utama dari penelitian, yaitu pengujian akurasi model ANN dengan menggunakan satu atau dua lapisan tersembunyi dalam pengenalan huruf alfabet.

Cabang pada diagram menjabarkan berbagai masalah dan faktor yang mempengaruhi akurasi model ANN dalam pengenalan huruf alfabet, seperti

kesulitan dalam mengenali huruf alfabet menggunakan ANN dan tingginya tingkat *error* dalam pengenalan huruf alfabet menggunakan ANN. Variabel penelitian yang menjadi fokus dalam penelitian ini adalah jumlah lapisan tersembunyi pada model ANN dan akurasi model ANN dalam pengenalan huruf alfabet.

Metode penelitian yang akan digunakan adalah studi literatur mengenai ANN dan pengenalan huruf alfabet, pengumpulan dataset huruf alfabet, pelatihan model ANN dengan satu dan dua lapisan tersembunyi menggunakan dataset yang sama, pengujian akurasi model ANN dengan menggunakan data *testing* yang berbeda, dan perbandingan akurasi model ANN dengan satu dan dua lapisan tersembunyi.

Pengumpulan data dilakukan dengan mengumpulkan dataset huruf alfabet dari beberapa sumber terpercaya dan membaginya menjadi data training dan data testing. Hasil penelitian diharapkan dapat memberikan analisis mengenai akurasi model ANN dengan satu dan dua lapisan tersembunyi dalam pengenalan huruf alfabet, serta perbandingan akurasi antara model ANN dengan satu dan dua lapisan tersembunyi. Kesimpulan penelitian dan saran untuk penelitian selanjutnya akan menjadi penutup dari penelitian ini.

3.2 Metode Penelitian

Metode penelitian adalah bagian penting dalam sebuah penelitian karena metode penelitian yang dipilih akan mempengaruhi cara pengumpulan dan analisis data serta hasil penelitian yang dihasilkan. Menurut Creswell (2014), ada tiga jenis metode penelitian yang umum digunakan dalam penelitian, yaitu metode kuantitatif, metode kualitatif, dan metode campuran (*mixed methods*).

Metode kuantitatif merupakan metode penelitian yang menggunakan pendekatan positivistik dengan tujuan mengukur dan menganalisis data yang berupa angka atau variabel. Metode kualitatif, di sisi lain, fokus pada pemahaman mendalam dan deskriptif terhadap fenomena yang diteliti melalui observasi, wawancara, atau analisis dokumen. Sementara itu, metode campuran adalah kombinasi dari metode kuantitatif dan kualitatif yang digunakan secara terpadu dalam sebuah penelitian.

Penelitian ini menggunakan metode campuran (*mix methods*), pendekatan ini menggabungkan elemen-elemen dari penelitian kualitatif dan kuantitatif untuk menghasilkan pemahaman yang lebih komprehensif mengenai topik yang diteliti. Penggunaan *mix method* dalam penelitian ini memungkinkan peneliti untuk mengeksplorasi aspek yang berbeda dari fenomena yang diteliti dan memperoleh gambaran yang lebih holistik tentang isu yang dihadapi.

Dalam konteks penelitian ini, pendekatan kuantitatif digunakan untuk mengumpulkan data numerik dan statistik dari luaran program. Data kuantitatif yang dikumpulkan ini memungkinkan peneliti untuk menilai dampak program secara objektif dan mengidentifikasi tren atau pola yang mungkin muncul.

Di sisi lain, pendekatan kualitatif diterapkan untuk memahami perspektif, pengalaman, dan opini peneliti. Data kualitatif yang dihasilkan dapat membantu peneliti memahami segala yang mempengaruhi keberhasilan program, serta mengungkapkan faktor-faktor yang mungkin tidak terlihat melalui analisis kuantitatif saja.

Dengan menggabungkan kedua metode ini, penelitian ini menciptakan gambaran yang lebih lengkap dan kaya tentang program yang sedang diteliti. Pendekatan *mix method* memungkinkan peneliti untuk memanfaatkan kelebihan kedua metode penelitian kualitatif dan kuantitatif, sambil mengatasi keterbatasan yang mungkin ada dalam masing-masing metode. Oleh karena itu, hasil dari penelitian ini diharapkan lebih akurat, valid, dan relevan untuk pengambilan keputusan dan perbaikan program di masa depan.

3.3 Metode Pengumpulan Data

Dalam melakukan penelitian ini, peneliti mengikuti langkah-langkah yang telah disebutkan di atas dan menyesuaikan dengan masalah serta metode penyelesaian yang diambil pada skripsi. Hal ini akan memastikan bahwa penelitian berjalan dengan baik dan memberikan hasil yang valid dan dapat diandalkan.

Berikut ini adalah tentang bagaimana langkah penelitian ini dilakukan dalam beberapa tahapan:

A. Pengumpulan Data:

Pada tahap ini, peneliti mengumpulkan data berupa gambar karakter dari 6 jenis *font* yang mencakup huruf besar dan huruf kecil (A-Z, a-z). Data ini akan digunakan sebagai input untuk melatih dan menguji model ANN. Pengumpulan data yang baik dan representatif akan membantu model ANN untuk menghasilkan kinerja yang lebih baik dalam mengenali karakter dari berbagai jenis *font*.

Penggunaan 6 jenis *font* berbeda dalam tahap pengumpulan data memiliki beberapa alasan:

1. Variabilitas: Dalam kehidupan nyata, karakter yang ditemui bisa berasal dari berbagai jenis *font*. Dengan menggunakan 6 jenis *font* berbeda, model ANN dapat belajar mengenali variasi yang ada di antara berbagai *font*, sehingga model menjadi lebih *robust* dan mampu mengenali karakter dari font yang berbeda dengan akurasi yang lebih baik.
2. Generalisasi: Melatih model ANN dengan variasi *font* yang lebih luas akan membantu model dalam menggeneralisasi pola yang ada di antara karakter dari *font* yang berbeda. Hal ini akan membantu model dalam mengenali karakter dari *font* yang tidak termasuk dalam data pelatihan.

Berdasarkan kode program yang telah dibuat, jumlah gambar yang dihasilkan akan bervariasi untuk setiap kombinasi karakter, *font*, ukuran, dan rotasi.

Dalam kode ini, ada:

- a. 52 karakter (A-Z, a-z)
- b. 6 jenis *font*
- c. ukuran (60)
- d. rotasi ((-15 sampai 15), 0)

Dengan demikian, jumlah total gambar yang dihasilkan adalah $52 * 6 * 3 = 936$ gambar. Jumlah gambar yang bervariasi memiliki beberapa keuntungan:

1. Melatih model dengan lebih banyak data: Semakin banyak data yang digunakan dalam pelatihan, semakin baik model dapat belajar pola yang ada di antara berbagai karakter dan *font*.

2. Menguji kinerja model dengan data yang belum pernah dilihat sebelumnya: Jika jumlah gambar yang dihasilkan bervariasi, kita dapat mengambil sebagian gambar sebagai data pengujian. Dengan demikian, kita dapat menguji kinerja model dalam mengenali karakter yang belum pernah dilihat sebelumnya dan menilai seberapa baik model dapat menggeneralisasi hasilnya.

Secara keseluruhan, penggunaan 6 jenis font berbeda dan variasi jumlah gambar dalam pengumpulan data akan membantu meningkatkan kinerja dan generalisasi model ANN dalam mengenali karakter dari berbagai font.

B. *Pra-olah Data:*

Sebelum data digunakan dalam proses pelatihan dan pengujian, data perlu melalui tahap *preprocessing*. Pada tahap ini, gambar karakter akan dikonversi menjadi skala abu-abu dan diubah ukurannya menjadi 128x128 piksel. Selanjutnya, gambar akan diubah menjadi vektor fitur dengan fungsi `image_to_feature_vector()` dan dinormalisasi menggunakan fungsi `Standardize()`. Berikut adalah penjelasan dari beberapa langkah yang dilakukan pada tahapan ini:

- a. Fungsi `image_to_feature_vector`: Fungsi ini digunakan untuk mengubah citra menjadi vektor fitur. Citra masukan diubah menjadi array NumPy dan di-flatten menjadi vektor 1D. Fungsi ini akan dipanggil nantinya untuk mengubah citra yang sudah diubah menjadi skala abu-abu dan diubah ukurannya.
- b. Inisialisasi variabel: Beberapa variabel yang digunakan dalam pengolahan data diinisialisasi pada awal kode, seperti alfabet, jenis *font*, ukuran citra, dan variasi ukuran dan rotasi.
- c. Pengolahan citra: Untuk setiap kombinasi karakter, *font*, ukuran, dan rotasi yang ada, program melakukan hal berikut:
 - a) Membuka citra dengan menggunakan `Image.open` dari *library* PIL (*Python Imaging Library*).
 - b) Mengonversi citra ke skala abu-abu dengan metode `convert('L')`.
 - c) Mengubah ukuran citra menjadi ukuran yang telah ditentukan (128 x 128)

menggunakan metode *resize* dengan interpolasi LANCZOS. *Interpolasi Lanczos* digunakan karena memiliki beberapa kelebihan dibandingkan dengan metode interpolasi lain. *Interpolasi Lanczos* digunakan karena:

1. Kualitas hasil: Menghasilkan interpolasi yang lebih halus dan akurat dibandingkan metode lain.
 2. Anti-aliasing: Mengurangi efek *aliasing* saat melakukan *resampling data*.
 3. Efisiensi komputasi: Lebih efisien daripada metode *interpolasi polinomial* derajat tinggi.
 4. Lokalitas: Fokus pada titik-titik data terdekat, mengurangi pengaruh *noise* atau *outlier*.
 5. Adaptabilitas: Parameter N dapat disesuaikan untuk menemukan keseimbangan antara akurasi dan kecepatan.
- d) Memanggil fungsi `image_to_feature_vector` untuk mengubah citra skala abu-abu menjadi vektor fitur.
- e) Menambahkan vektor fitur tersebut ke dalam *list* data dengan menambahkan label yang sesuai (indeks alfabet) pada akhir vektor fitur.
- d. Membuat *DataFrame* pandas: Setelah semua citra telah diolah, program akan membuat *DataFrame* pandas dengan menggunakan list data. Kolom-kolom dari *DataFrame* ini akan diberi nama 'pixel_0', 'pixel_1', ..., 'pixel_{n-1}', dan 'label', di mana n adalah jumlah total piksel dalam citra (128 x 128).
- e. Menyimpan *DataFrame* sebagai file CSV: *DataFrame* yang telah dibersihkan akan disimpan sebagai file CSV dengan nama 'dataset_New.csv'.

Dalam tahapan *preprocessing* data ini, citra-citra diubah menjadi vektor fitur dan disimpan dalam format yang mudah diolah oleh model ANN.

C. Pembagian Data:

Data yang telah melalui tahap *preprocessing* kemudian dibagi menjadi dua set: data pelatihan dan data pengujian. Data pelatihan digunakan untuk melatih

model ANN, sedangkan data pengujian digunakan untuk menguji kinerja model dalam mengenali karakter yang belum pernah dilihat sebelumnya.

Pada tahap pembagian data, dataset yang digunakan dalam penelitian ini akan dibagi menjadi dua bagian, yaitu data latihan dan data uji. Pembagian ini dilakukan untuk melatih model pada data latihan dan kemudian mengevaluasi kinerja model pada data yang belum pernah dilihat sebelumnya (data uji). Berikut adalah langkah-langkah yang dilakukan dalam tahap pembagian data:

- a. Membaca dataset dari file CSV dan mengonversinya menjadi *NumPy array*.
- b. Mengacak data agar data terdistribusi secara acak.
- c. Membagi data menjadi data latihan dan data uji dengan proporsi 70% untuk data latihan dan 30% untuk data uji. Pembagian ini dilakukan dengan menghitung 70% dari jumlah total sampel dalam dataset dan menggunakan nilai ini sebagai titik pemisah antara data latihan dan data uji.
- d. Mengeluarkan fitur dan label dari data latihan dan data uji. Fitur disimpan dalam kolom pertama hingga kolom kedua terakhir, dan label disimpan dalam kolom terakhir. Data fitur dan label kemudian ditransposisikan untuk memenuhi format yang diinginkan.
- e. Melakukan normalisasi pada fitur. Normalisasi ini dilakukan untuk mengurangi pengaruh skala fitur yang berbeda pada proses pelatihan model. Normalisasi menggunakan rumus berikut:

$$X_{norm} = \frac{X - \mu}{\sigma + \epsilon} \quad 3.1$$

di mana X_{norm} adalah fitur yang dinormalisasi, X adalah fitur asli, μ adalah mean dari fitur, sigma (σ) adalah standar deviasi dari fitur, dan epsilon (ϵ) adalah konstanta kecil untuk menghindari pembagian dengan nol.

- f. Mengubah label menjadi *one-hot encoding*. *One-hot encoding* digunakan untuk mengubah label kelas menjadi vektor *biner*, di mana hanya satu elemen yang bernilai 1 dan sisanya bernilai 0. Misalnya, jika ada 3 kelas dan label adalah 2, vektor one-hot encoding-nya adalah [0, 1, 0].

D. Pelatihan Model ANN:

Pada tahap ini, dua model ANN dengan arsitektur yang berbeda (1-layer dan 2-layer) akan dilatih menggunakan data pelatihan. Model ANN akan mempelajari hubungan antara vektor fitur input dan label karakter yang sesuai. Proses pelatihan melibatkan metode *forward propagation*, perhitungan gradien, dan pembaruan parameter menggunakan gradien yang dihitung.

E. Evaluasi Model ANN:

Setelah model ANN dilatih, kinerja model akan dievaluasi menggunakan data pengujian. Pada tahap ini, model akan mencoba mengenali karakter dari data pengujian, dan hasilnya akan dibandingkan dengan label sebenarnya untuk menghitung akurasi model.

F. Implementasi GUI dan Prediksi Karakter:

Terakhir, model ANN yang telah dilatih akan diintegrasikan ke dalam aplikasi GUI berbasis tkinter. Aplikasi ini memungkinkan pengguna untuk memuat gambar karakter dan melihat prediksi dari kedua model ANN (1-layer dan 2-layer). Pengguna dapat membandingkan kinerja model dalam mengenali karakter dari 6 jenis *font* yang berbeda.

Dengan menerapkan proses di atas, penelitian ini bertujuan untuk membandingkan kinerja model ANN dengan arsitektur yang berbeda dalam mengenali karakter dari berbagai jenis *font*.

3.3.1 Instrumen Penelitian

Instrumen penelitian ini meliputi teknik pembuatan dataset, pengolahan data, dan algoritma pembelajaran mesin. Berikut adalah instrumen penelitian yang digunakan dalam ketiga program tersebut secara detail:

1. *Alfabet Generator* : Pembuatan Dataset

Instrumen dalam program ini adalah teknik pembuatan dataset berupa gambar huruf dengan variasi *font*, ukuran, dan rotasi. Gambar-gambar ini akan digunakan sebagai input untuk model jaringan saraf tiruan (ANN).

- a. Fungsi `create_image`: Fungsi ini digunakan untuk membuat gambar huruf

dengan spesifikasi font, ukuran, dan rotasi yang diberikan. Fungsi ini menggunakan library *Python Image Library* (PIL) untuk menghasilkan gambar dan menyimpannya dalam folder yang ditentukan.

- b. Pengulangan: Program menggunakan *nested loop* untuk menghasilkan kombinasi semua huruf, *font*, ukuran, dan rotasi yang didefinisikan dalam variabel penelitian. Setiap kombinasi tersebut akan diproses oleh fungsi `create_image`.

2. Konversi Gambar : Pengolahan Data

Instrumen dalam program ini adalah teknik pengolahan data untuk mengkonversi gambar huruf menjadi bentuk yang dapat digunakan oleh model ANN.

- a. Fungsi `load_images`: Fungsi ini digunakan untuk membaca gambar huruf dari folder yang ditentukan dan mengubahnya menjadi *array* dengan ukuran 128x128 piksel. *Array* ini kemudian diubah menjadi bentuk vektor 1D sepanjang 16.384 elemen.
- b. Fungsi `save_to_csv`: Fungsi ini digunakan untuk menyimpan *array* yang dihasilkan oleh fungsi `load_images` ke dalam file CSV bersama dengan label kelas (misalnya, A, B, C, dll.) yang sesuai.

3. Pelatihan : Algoritma Pembelajaran Mesin

Instrumen dalam program ini adalah algoritma jaringan saraf tiruan (ANN) yang digunakan untuk mengidentifikasi huruf dari gambar yang dihasilkan.

- a. Fungsi `load_data`: Fungsi ini digunakan untuk membaca file CSV yang dihasilkan oleh `csv_dataFrame.py` dan membagi data menjadi data latih dan data uji dengan proporsi yang ditentukan (contoh: 70% data latih dan 30% data uji).
- b. Fungsi `train_model`: Fungsi ini digunakan untuk melatih model ANN dengan data latih yang dihasilkan oleh fungsi `load_data`. Model ANN menggunakan algoritma *backpropagation* dengan metode *Gradient Descent* untuk optimasi.

Parameter seperti *learning rate*, *epoch*, dan *batch ratio* ditentukan oleh variabel penelitian.

- c. Fungsi *evaluate_model*: Fungsi ini digunakan untuk menguji kinerja model ANN yang telah dilatih dengan menggunakan data uji yang dihasilkan oleh fungsi *load_data*. Kinerja model diukur dengan menghitung akurasi, presisi, *recall*, dan *F1-score*.

4. Pengujian : Pengujian Model

Instrumen dalam program ini adalah teknik pengujian model ANN yang telah dilatih pada data baru yang tidak pernah dilihat sebelumnya oleh model. Ini membantu dalam mengevaluasi sejauh mana model berhasil menggeneralisasi pengetahuan yang diperoleh selama pelatihan.

- a. Fungsi *load_model*: Fungsi ini digunakan untuk memuat model ANN yang telah dilatih dari file yang disimpan sebelumnya. Model ini akan digunakan untuk menguji kinerjanya pada data baru.
- b. Fungsi *load_test_data*: Fungsi ini digunakan untuk membaca gambar huruf baru yang akan digunakan sebagai data pengujian. Gambar ini diolah dengan cara yang sama seperti yang dilakukan oleh *csv_dataframe.py*, yaitu dengan mengubahnya menjadi *array* berukuran 128x128 piksel, kemudian diubah menjadi bentuk vektor 1D sepanjang 16.384 elemen.
- c. Fungsi *test_model*: Fungsi ini digunakan untuk menguji kinerja model ANN yang telah dilatih pada data pengujian yang dihasilkan oleh fungsi *load_test_data*. Model akan menghasilkan prediksi untuk setiap gambar dalam data pengujian, dan prediksi.

Instrumen penelitian ini secara keseluruhan melibatkan pembuatan dataset, pengolahan data, dan penerapan algoritma ANN untuk mengidentifikasi huruf dari gambar yang dihasilkan. Instrumen ini penting untuk memahami bagaimana kombinasi *font*, ukuran, dan rotasi mempengaruhi kinerja model ANN dalam mengenali huruf alfabet berdasarkan beberapa variasi *font*. Instrumen ini juga mencakup validasi silang dan metrik evaluasi yang digunakan untuk mengukur efektivitas model dalam mengklasifikasikan data uji.

3.4 Metode Pengujian Data

Pada bab ini, akan dibahas mengenai metode pengujian *black box* dan *white box* yang digunakan dalam penelitian ini. *Black box testing* digunakan untuk menguji antarmuka grafis pada program yang digunakan untuk menguji model satu dan dua lapisan tersembunyi dalam memprediksi gambar huruf. Sementara itu, *white box* testing dilakukan dengan cara yang lebih mendalam. Metode ini melibatkan pemeriksaan struktur dan kode dari aplikasi secara detail. Tujuannya adalah untuk memastikan bahwa aplikasi telah dibangun dengan baik dan sesuai dengan standar yang ditetapkan. Dalam *white box* testing, pengujian dilakukan dengan memeriksa logika program, aliran data, dan mengevaluasi penggunaan variabel dan fungsi.

3.4.1 Black Box

Dalam bab pengujian *black box*, metode pengujian antarmuka grafis digunakan untuk menguji model satu dan dua lapisan tersembunyi dalam memprediksi gambar huruf. Pengujian ini bertujuan untuk mengevaluasi fungsionalitas dan responsivitas antarmuka serta memastikan keakuratan prediksi dari model tersebut. Metode *black box* testing memandang antarmuka sebagai sebuah entitas terpisah yang diuji tanpa memperhatikan rincian implementasi internal dari model yang digunakan.

Tabel 3. 1 Uji Fungsionalitas dengan Pengujian *Black Box*

No.	Fungsi Pengujian
1	Load Image
2	Predict Image
3	Save Image
4	Update Timestamp in Frame

3.4.1 White Box

Pada pengujian *white box*, dilakukan pemeriksaan mendalam terhadap struktur dan kode aplikasi untuk memastikan kualitas dan kesesuaian dengan

standar yang ditetapkan. Metode ini melibatkan analisis logika program, aliran data, serta evaluasi penggunaan variabel dan fungsi. Pengujian white box bertujuan untuk mengidentifikasi dan memperbaiki potensi bug, kesalahan logika, dan kelemahan dalam implementasi aplikasi. Dengan pendekatan yang lebih detail, pengujian white box memberikan kepercayaan bahwa aplikasi telah dibangun dengan baik dan sesuai dengan persyaratan yang ditentukan.

Tabel 3. 2 Uji Fungsionalitas dengan Pengujian *White Box*

NO	FUNGSI	HASIL YANG DIHARAPKAN
1	image_to_feature_vector	Mengembalikan vektor fitur gambar
2	Standardize	Mengembalikan matriks vektor fitur yang telah distandarisasi
3	SoftmaxFunction	Mengembalikan vektor output softmax
4	Sigmoid	Mengembalikan output sigmoid
5	forwardProp	Mengembalikan array yang berisi nilai setiap node dalam jaringan