BAB III
METODE PENELITIAN

3.1 Objek Penelitian

Objek utama penelitian ini adalah arsitektur model *Hybrid Quantum-Klasik* (QCNN-ResNet-18) yang dioptimalkan, dirancang, dan diimplementasikan sebagai studi kelayakan untuk tugas Pengenalan Tulisan Tangan (*Handwritten Text Recognition* - HTR) karakter dalam format spesifik. Fokus spesifik terletak pada analisis kinerja model ini dalam mengenali karakter Hangeul dan angka yang ditulis pada format kertas ujian TOPIK II, yaitu kertas *wongoji*.

Lingkup data yang menjadi subjek penelitian bersumber dari penggabungan dataset primer dan dataset sekunder. Dataset primer didapatkan melalui simulasi secara langsung berupa 100 lembar citra pindaian *wongoji* tulisan tangan yang dikumpulkan dari 50 responden. Sementara itu, dataset sekunder diakuisisi dari sumber publik yang terdiri dari 128.006 citra *Handwritten Hangeul* dari repositori Kaggle (mewakili 64 suku kata Hangeul) serta 70.000 citra tulisan digit 0-9 dari dataset MNIST.

Pangkalan data gabungan ini akan mencakup variasi gaya tulisan tangan yang komprehensif dari kedua sumber untuk menguji ketahanan (*robustness*) dan kemampuan generalisasi model *Hybrid* QCNN-ResNet-18 dalam membedakan total 74 kelas target klasifikasi yang digunakan untuk menyusun teks pada lembar *wongoji*.

3.1.1 Alat dan Bahan Penelitian

Pelaksanaan penelitian ini didukung oleh perangkat keras dan perangkat lunak dengan spesifikasi sebagai berikut:

1) Perangkat Keras
    a) Laptop: ASUS TUF Gaming A15 (FA507NV)
    b) CPU: AMD Ryzen™ 7 7735HS
    c) RAM: 32 GB DDR5
    d) GPU: NVIDIA® GeForce RTX™ 4050 Laptop

2) Perangkat Lunak
   a) Sistem Operasi: Windows 11
   b) *Code Editor:* Visual Studio Code & Antigravity
   c) Lingkungan Pengembangan: Anaconda Distribution
   d) Bahasa Pemrorgraman: Python 3.10
   e) *Framework Backend*: Django
   f) *Library* Utama:
      i. Quantum Machine Learning: PennyLane (inti QML); PennyLane-Qiskit (integrasi Qiskit)
      ii. Deep Learning Klasik: PyTorch (untuk *layer* CNN dan *training loop*)
      iii. Computer Vision: OpenCV (untuk segmenteasi *wongoji* dan *preprocessing* citra)
      iv. Manipulasi Data: Pandas, NumPy
      v. Evaluasi Model: Scikit-learn (untuk metrik CER dan *Confusion Matrix*)
      vi. Visuaslisasi: Matplotlib, Seaborn
   g) Dependensi Lain: CUDA Toolkit 12.1

3.1.2 Jadwal Penelitian

Pelaksanaan penelitian ini akan mengikuti jadwal yang telah disusun secara sistematis. Rincian tahapan beserta alokasi waktunya disajikan pada Tabel 3.1.

Tabel 3.1 Jadwal Penelitian

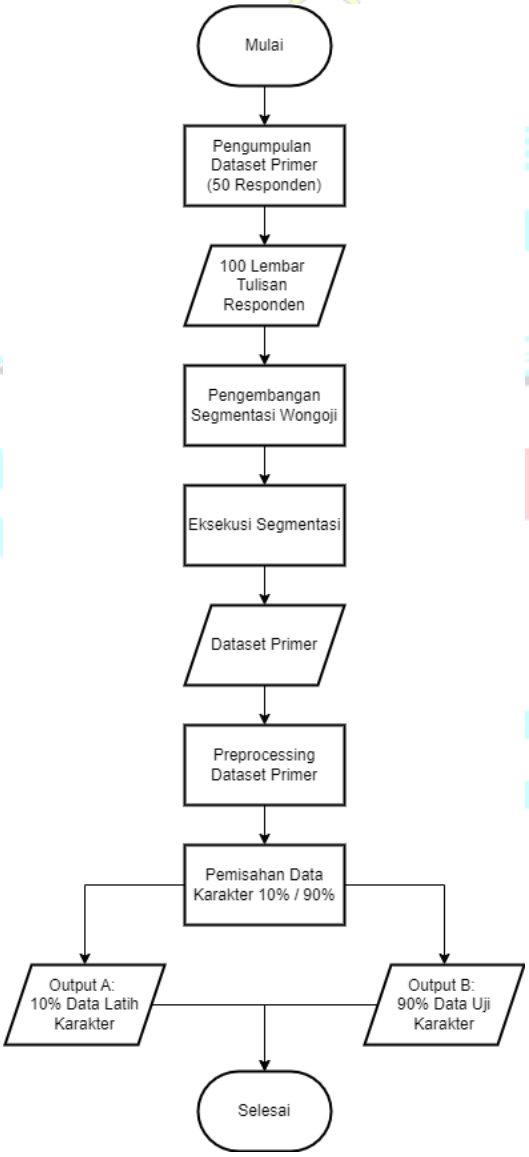
No.	Kegiatan	Bulan ke-1	Bulan ke-2	Bulan ke-3	Bulan ke-4
1.	Studi Literatur & Finalisasi Metode				
2.	Pengambilan *Dataset Wongoji* (Skala Kecil/Terfokus)				
3.	Pengembangan Algoritma Segmentasi *Wongoji*				
4.	Akuisisi Data *Sekunder* & *Setup* QML				
5.	Implementasi Arsitektur Hybrid QCNN-ResNet-18				
6.	*Training* Model				
7.	Pengujian dan Evaluasi Model				
8.	Pengujian, Evaluasi (CER), & Analisis Hasil				

No.	Kegiatan	Bulan ke-1	Bulan ke-2	Bulan ke-3	Bulan ke-4
9.	Pengembangan Dasbor *Web* (Django)				
10.	Penyusunan Laporan Akhir & Bimbingan				

3.2 Prosedur Penelitian

Penelitian ini dirancang menggunakan alur kerja sistematis dan modular untuk memastikan kejelasan metodologi pada setiap tahapannya. Secara keseluruhan, prosedur penelitian dibagi menjadi lima alur utama yang saling berkesinambungan. Rincian dari kelima alur tersebut adalah sebagai berikut:

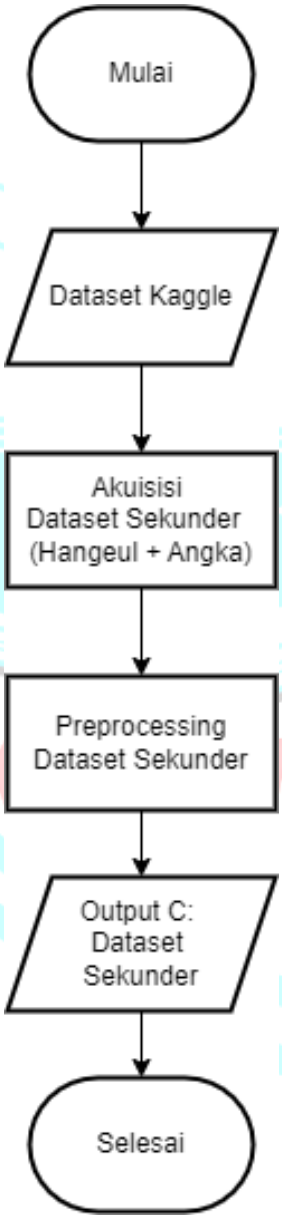
1) Alur 1 - Persiapan Data Primer



Gambar 3.1 Alur 1 – Persiapan Data Primer

Tahap ini berfokus pada akuisisi, pra-pemrosesan, dan segmentasi data tulisan tangan yang diperoleh dari 50 responden. Proses ini menghasilkan dua luaran utama, yaitu Output A (10% Data Latih Karakter) dan Output B (90% Data Uji Karakter).

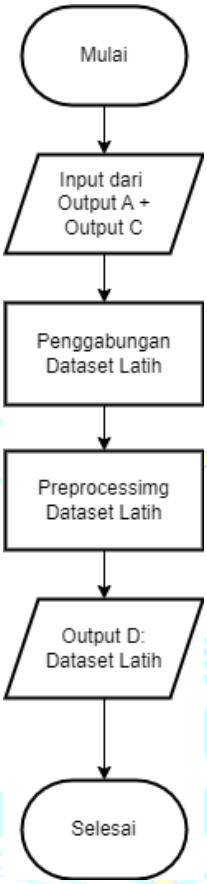
2) Alur 2 - Persiapan Data Sekunder



Gambar 3.2 Alur 2 – Persiapan Data Sekunder

Tahap ini berfokus pada akuisisi dan pra-pemrosesan data publik eksternal, yaitu *dataset* Kaggle Hangeul dan *dataset* MNIST. Tahap ini menghasilkan luaran berupa Output C (*Dataset* Sekunder) yang susunan labelnya telah distrukturisasi dan siap untuk digabungkan dengan data primer.

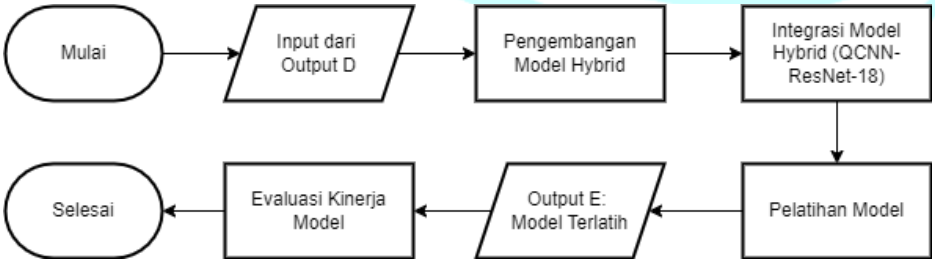
3) Alur 3 - Konstruksi *Dataset* Latih



Gambar 3.3 Alur 3 – Konstruksi Data Latih

Tahap ini berfokus pada proses penggabungan dan pra-pemrosesan pangkalan data secara keseluruhan. Masukan dari Output A (10% Data Latih Karakter) dan Output C (*Dataset* Sekunder) digabungkan untuk menghasilkan pangkalan data akhir berupa Output D (*Dataset* Latih).

4) Alur 4 - Pelatihan dan Evaluasi Model Hybrid (Gambar 3.4)

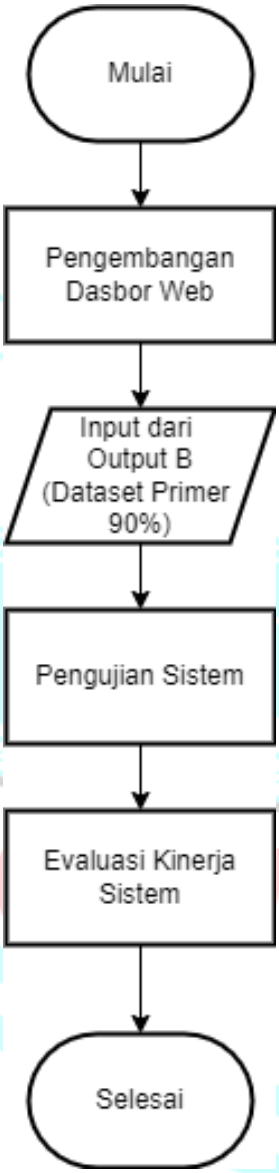


Gambar 3.4 Alur 4 – Pelatihan dan Evaluasi Model Hybrid

Tahap ini berfokus pada proses integrasi arsitektur, pelatihan, dan pengujian kinerja klasifikasi model secara objektif. Data masukan dari Output D digunakan

untuk melatih model, yang kemudian dievaluasi menggunakan Output B (90% Data Uji Karakter) untuk menghasilkan luaran utama berupa Output E (Model Terlatih).

5) Alur 5 - Implementasi dan Pengujian Sistem (Gambar 3.5)



Gambar 3.5 Alur 5 – Implementasi & Pengujian Sistem

Tahap ini berfokus pada pengembangan dasbor *web* dan pengujian fungsional sistem secara *end-to-end*. Output E diintegrasikan ke dalam sistem *backend web*, kemudian dokumen pindaian lembar jawaban utuh dari Output B diunggah ke dalam sistem sebagai masukan untuk mengevaluasi kinerja akhir (inferensi) berdasarkan metrik *Character Error Rate* (CER).

Penjelasan rinci untuk setiap tahapan dalam kelima alur tersebut akan dijabarkan dalam sub-bab berikut.

3.2.1 Persiapan Data Primer (Alur 1)

Alur ini (ditunjukkan pada Gambar 3.1) berfokus pada tahapan akuisisi, pra-pemrosesan, dan segmentasi data citra tulisan tangan yang disusun secara spesifik mengikuti struktur format *wongoji*. Rangkaian tahapan ini dirancang untuk memproduksi dua himpunan luaran data utama.

1) Pengumpulan *Dataset* Primer (50 Responden)

Mengingat *dataset* publik tidak menyediakan sampel tulisan tangan dalam format kertas ujian TOPIK II, pengumpulan data primer dilakukan melalui simulasi ujian langsung. Proses ini melibatkan 50 responden yang ditugaskan untuk menyalin teks pendek pada kertas *wongoji*. Hasil tulisan tangan yang berjumlah 100 lembar ini kemudian dipindai (*scan*) dengan resolusi optikal minimum 300 *dots per inch* (dpi) untuk menghasilkan citra digital dengan kualitas yang memadai. Format kertas *wongoji* yang digunakan dapat dilihat pada Gambar 3.6.



Gambar 3.6 Format Lembar *Wongoji* Standar

Karakteristik visual utama dari kertas ini adalah susunan *grid* kotak berwarna merah yang membatasi setiap karakter. Struktur *grid* inilah yang menjadi tantangan teknis utama dalam proses segmentasi komputasi citra.

2) Pengembangan Segmentasi *Wongoji*

Untuk dapat mengekstraksi dan mengisolasi setiap karakter dari dalam *grid* kertas *wongoji*, penelitian ini merancang sebuah algoritma *Computer Vision* berbasis pustaka OpenCV. Algoritma ini mengintegrasikan beberapa teknik

pemrosesan citra, meliputi: lokalisasi filter garis kontur horisontal dan vertikal untuk mendeteksi susunan kotak matriks, operasi *Connected Components* untuk membentuk *bounding box* karakter secara presisi di dalam sel, serta modul klasifikasi ruang kosong (*blank-space classification*) untuk mendeteksi spasi antar kata.

Secara komputasional, proses ekstraksi sel dari format kertas ujian ini dirancang melalui serangkaian operasi perbaikan citra dan morfologi. Algoritma harus mampu menangani anomali fisis pada dokumen pindaian, seperti kemiringan kertas (*skew*), keberadaan teks *header* yang tidak relevan, hingga anomali garis sel yang terputus atau menyatu akibat degradasi resolusi optikal.

Untuk mengatasi tantangan tersebut, penelitian ini merancang Algoritma Segmentasi Wongoji Berbasis OpenCV. Alur kerja algoritma ini mencakup deteksi garis kontur secara adaptif, pengurutan letak sel secara sekuensial (baris dan kolom), hingga tahap pembersihan bentuk sel (merestorasi sel yang terpecah dan membagi sel yang tumpang-tindih) untuk mencapai target rasio 220 kotak pada *grid* standar. Proses segmentasi ini diakhiri dengan klasifikasi awal untuk membedakan antara sel berisi karakter dan sel kosong yang merepresentasikan spasi.

Rancangan logika dari metode ekstraksi *Computer Vision* ini direpresentasikan melalui *Pseudocode* 1 berikut:

Pseudocode 1: Segmentasi Grid Wongoji Berbasis OpenCV

```
FUNCTION SegmentasiWongoji(raw_image):
    corrected_image = CorrectSkew(raw_image)
    gray_image = ConvertToGrayscale(corrected_image)

    binary_image = AdaptiveThreshold(gray_image)
    horizontal_lines = MorphologicalOperation(binary_image, "HORIZONTAL")
    vertical_lines = MorphologicalOperation(binary_image, "VERTICAL")
    grid_structure = Combine(horizontal_lines, vertical_lines)

    raw_boxes = FindContours(grid_structure)

    filtered_boxes = FilterHeaderNoise(raw_boxes)
    sorted_boxes = SortBoxesGrid(filtered_boxes)

    merged_boxes = MergeSplitCells(sorted_boxes)
    split_horiz = SplitOversizedCells(merged_boxes)
    split_vert = SplitOversizedCellsVertical(split_horiz)
    final_boxes = FillMissingCells(split_vert, target=220)

    extracted_cells = []
    FOR EACH box IN final_boxes DO:
        cell_image = CropImage(corrected_image, box)
```

Pseudocode 1: Segmentasi Grid Wongoji Berbasis OpenCV

```
        IF IsCellEmpty(cell_image) == TRUE THEN:
            cell_type = "EMPTY_SPACE"
        ELSE:
            cell_type = ClassifyCellContent(cell_image)
        END IF

        extracted_cells.Append({box, cell_image, cell_type})
    END FOR

    RETURN extracted_cells
END FUNCTION
```

3) Eksekusi Segmentasi dan Preprocessing *Dataset* Primer

Algoritma segmentasi yang telah dikembangkan kemudian diterapkan pada 100 lembar citra pindaian responden. Proses ini diawali dengan mengaplikasikan koreksi kemiringan citra (*skew correction*) dan penghapusan grid menggunakan operasi morfologi. Tujuannya adalah untuk memisahkan setiap coretan karakter dari garis bingkai *wongoji*. Kumpulan citra karakter tunggal hasil ekstraksi inilah yang dikategorikan sebagai *Dataset* Primer.

Selanjutnya, *Dataset* Primer ini harus melewati tahap pra-pemrosesan (*preprocessing*). Tahap filtrasi ini mencakup penghalusan *noise* menggunakan *Gaussian Blur* dengan *filter* 5x5, serta penerapan binarisasi adaptif menggunakan *Otsu Thresholding* untuk mempertegas batas tulisan. Selain itu, diaplikasikan pula operasi morfologi *Opening* dan *Closing* dengan kernel persegi panjang untuk menghubungkan piksel garis yang terputus akibat cacat pemindaian sehingga bentuk karakter menjadi lebih utuh. Proses ini kemudian diikuti dengan pemerataan kontras, penghapusan citra kotak yang kosong, serta normalisasi ukuran citra (*resize*) secara seragam menjadi dimensi 64x64 piksel. Normalisasi ini krusial agar dimensi karakter selaras dengan spesifikasi dataset sekunder publik yang akan digabungkan pada tahapan Alur 3.

4) Pemisahan Data Karakter 10% / 90%

Dataset Primer yang telah melalui pra-pemrosesan kemudian dipartisi (*split*) ke dalam dua himpunan dengan rasio yang asimetris, yaitu 10% dan 90%. Pemisahan ini ditujukan untuk kebutuhan fungsional yang berbeda:

a) Output A: 10% Data Latih Karakter

Sebagian kecil data dialokasikan untuk diteruskan ke Alur 3. Data primer ini berfungsi sebagai tambahan variasi karakteristik tulisan tangan nyata (*noise*) untuk memperkaya bobot model saat pelatihan.

b) Output B: 90% Data Uji Karakter

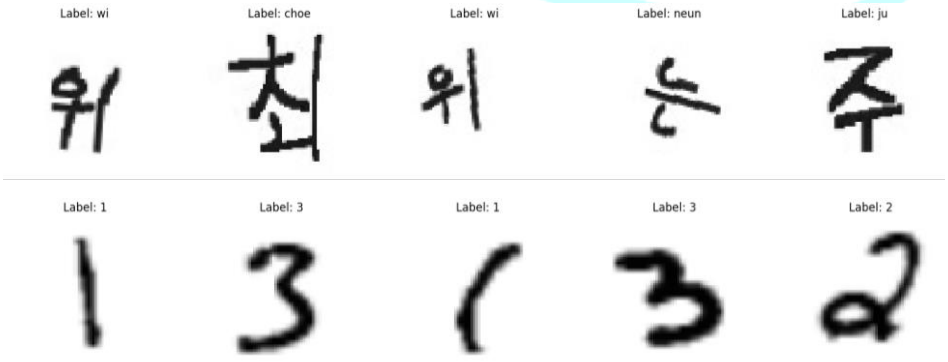
Sebagian besar proporsi data (90%) diisolasi secara ketat sebagai data uji murni (*unseen data*). Himpunan data ini sama sekali tidak dilibatkan dalam proses pelatihan model. Fungsinya murni sebagai instrumen pengujian untuk memvalidasi kinerja inferensi sistem secara *End-to-End* pada tahapan Alur 4 dan Alur 5.

3.2.2 Persiapan Data Sekunder (Alur 2)

Tahap ini (ditunjukkan pada Gambar 3.2) berfokus pada proses akuisisi dan pra-pemrosesan data eksternal (publik) yang digunakan sebagai fondasi utama untuk memperkaya basis data pelatihan model.

1) Akusisi *Dataset* Sekunder (Hangeul + Angka)

Penelitian ini menggunakan *dataset* publik dari repositori Kaggle yang berjudul *"Handwritten Hangul Korean Syllables"* untuk mewakili 64 kelas suku kata Hangeul target. Untuk mengakomodasi penulisan numerik pada format *wongoji*, *dataset* ini kemudian digabungkan dengan *dataset* angka tulisan tangan (digit 0-9) dari basis data MNIST. Kedua *dataset* tersebut secara bawaan memiliki karakteristik visual yang seragam, yaitu tulisan berwarna gelap di atas latar belakang terang (putih). Penggabungan ini bertujuan membentuk ruang klasifikasi target yang terdiri dari 74 kelas probabilitas. Contoh citra dari *dataset* sekunder ini disajikan pada Gambar 3.7.



Gambar 3.7 Sampel *Dataset* Sekunder

2) Penstrukturan Label *Dataset* Sekunder

Sebelum kedua himpunan data tersebut digabungkan, dilakukan tahap pra-pemrosesan khusus untuk menangani tumpang-tindih (*overlap*) indeks label. Indeks kelas pada *dataset* MNIST secara bawaan bernilai 0 hingga 9, yang bertabrakan dengan indeks awal pada kelas Hangeul (0 hingga 63). Untuk mencegah benturan identitas (*ID collision*) saat proses komputasi, algoritma menerapkan penambahan nilai indeks (*Label Offset*) sebesar +64 khusus untuk label *dataset* MNIST. Dengan pergeseran ini, kelas angka menduduki urutan ke-64 hingga 73 secara berurutan. Gabungan data yang labelnya telah terstruktur ini ditetapkan sebagai Output C (*Dataset* Sekunder), yang siap untuk dilanjutkan ke tahapan Alur 3.

3.2.3 Penggabungan dan *Preprocessing Dataset* Latih (Alur 3)

Alur ini berfungsi sebagai tahap integrasi untuk membentuk satu pangkalan data latih yang padu dan terstandardisasi secara dimensi.

1) Penggabungan *Dataset* Latih

Tahap ini menggabungkan Output A (10% Data Latih Karakter *wongoji* internal) dan Output C (*Dataset* Sekunder). Penggabungan ini sangat penting agar model tidak hanya mengenali karakteristik bentuk dari data publik, tetapi juga mampu beradaptasi dengan variasi ketebalan goresan pena tulisan tangan riil dari lingkungan pindaian *wongoji*.

2) *Preprocessing* Pengerucutan *Dataset* Latih Terpadu

Setelah seluruh sumber data digabungkan, tahapan standardisasi dimensi diterapkan secara serentak. Seluruh citra diubah ukurannya secara seragam (*uniform resize*) menjadi resolusi 32x32 piksel. Setelah itu, ruang matriks citra dikonversi menjadi format 3-*channel grayscale* agar dapat disesuaikan dengan arsitektur masukan standar *ImageNet* yang dibutuhkan oleh *Convolutional Neural Network* (ResNet-18). Pangkalan data yang telah terstandardisasi ini ditetapkan sebagai Output D (*Dataset* Latih), yang siap menjadi masukan utama untuk proses ekstraksi fitur dan pelatihan sirkuit kuantum.

3.2.4 Pengembangan, Pelatihan, dan Evaluasi Model Hibrida (Alur 4)

Alur ini (ditunjukkan pada Gambar 3.4) merupakan tahapan inti dalam penelitian ini, yang berfokus pada pengembangan, integrasi, pelatihan, dan evaluasi kinerja arsitektur *Handwritten Text Recognition* (HTR). Proses komputasi pada

tahap ini berjalan dengan menggunakan masukan pangkalan data yang telah dipersiapkan sebelumnya, yaitu Output D (*Dataset* Latih).

1) Pengembangan Model Hybrid

Pembangunan arsitektur model klasifikasi ini dibagi menjadi dua komponen komputasi yang bekerja secara paralel:

a) Ekstraktor Fitur Klasik (ResNet-18)

Proses ekstraksi fitur dilakukan menggunakan arsitektur ResNet-18 yang dimodifikasi tanpa memuat bobot pra-pelatihan (*without pre-trained weights*). Modifikasi utama dilakukan dengan menghapus lapisan klasifikasi bawaan (*fully connected layer*) dan menggantinya dengan lapisan nn.Linear kustom untuk mereduksi dimensi fitur keluaran dari 512 menjadi 256 parameter. Reduksi ini diwajibkan untuk menyesuaikan dimensi vektor dengan batas kapasitas *encoding* sirkuit kuantum (8-*qubit*).

b) *Classifier* Kuantum (QCNN)

Lapisan klasifikasi dibangun menggunakan ekosistem komputasi kuantum PennyLane dengan alokasi 8-*qubit*. Vektor representasi berdimensi 256 dari fitur klasik dipetakan ke dalam keadaan kuantum (*quantum state*) menggunakan metode pengkodean amplitudo (qml.AmplitudeEmbedding). Status *qubit* tersebut kemudian diproses secara terbelit (*entangled*) melalui *Variational Quantum Circuit* (VQC) dengan kedalaman 6 lapisan *BasicEntanglerLayers*. Proses kuantum ini diakhiri dengan pengukuran nilai ekspektasi observasi operator Pauli-Z pada masing-masing dari ke-8 *qubit*.

2) Integrasi Model Hybrid (QCNN-ResNet-18)

Arsitektur ekstraktor fitur klasik dan sirkuit *classifier* kuantum tersebut diintegrasikan secara utuh (*end-to-end*) menggunakan antarmuka qml.qnn.TorchLayer. Matriks nilai ekspektasi dari 8-*qubit* (yang memuat pola fitur ekstraksi berdimensi 256) diteruskan ke lapisan klasifikasi klasik nn.Linear terakhir untuk memetakan probabilitas prediksi ke dalam 74 kelas target (64 kelas karakter Hangeul dan 10 kelas angka).

3) Pelatihan Model

Model hibrida PyTorch-PennyLane yang telah terintegrasi tersebut kemudian dilatih menggunakan pangkalan data latih (*Dataset Latih*). Proses optimasi jaringan

dilakukan menggunakan algoritma *Adam Optimizer* yang dipadukan dengan fungsi kerugian *Cross-Entropy Loss*. Fungsi ini bertujuan untuk memperbarui bobot parameter komputasi—baik pada lapisan saraf klasik maupun parameter sirkuit kuantum—secara iteratif melalui mekanisme propagasi balik (*backpropagation*). Setelah proses pelatihan (*training loop*) selesai dan konvergen, keadaan jaringan ini dibekukan dan disimpan sebagai luaran Output E (Model Terlatih).

4) Evaluasi Kinerja Model

Kinerja dari Output E (Model Terlatih) divalidasi dan dievaluasi secara objektif menggunakan himpunan data uji murni yang belum pernah dilibatkan dalam tahap pelatihan, yaitu Output B (90% Data Uji Karakter). Kapabilitas lapisan prediksi QCNN dalam mengenali 74 kelas target tersebut diukur melalui matriks kesalahan (*Confusion Matrix*), yang memuat empat metrik kinerja, antara lain:

a) *Accuracy* (Akurasi Keseluruhan)

Mengukur persentase tebakan kelas karakter (Hangeul dan angka) yang diprediksi secara benar oleh sistem dibandingkan total populasi pengujian.

b) *Precision*

Mengukur rasio ketepatan prediksi positif untuk setiap sub-kelas.

c) *Recall*

Mengukur proporsi sampel positif aktual (*ground truth*) yang berhasil diidentifikasi dengan benar oleh model.

d) *F1-Score*

Rata-rata harmonik yang menyeimbangkan metrik *Precision* dan *Recall*, berfungsi sebagai parameter komparasi stabilitas yang krusial pada sebaran jumlah data kelas yang tidak seimbang (*imbalanced class*).

Analisis menyeluruh dari komputasi metrik evaluasi ini menentukan keandalan struktur model klasifikasi sebelum masuk pada pengujian fungsi *end-to-end* (Alur 5).

3.2.5 Implementasi dan Pengujian Sistem (Alur 5)

Tahap terakhir ini (ditunjukkan pada Gambar 3.5) berfokus pada implementasi praktis dan pengujian fungsionalitas sistem secara menyeluruh (*End-to-End*).

1) Pengembangan Dasbor *Web*

Berdasarkan model yang telah divalidasi pada tahap sebelumnya (Output E: Model Terlatih) beserta algoritma segmentasi *wongoji*, sistem diintegrasikan ke dalam arsitektur aplikasi berbasis *web* menggunakan *framework* Django. Antarmuka *web* ini dirancang sebagai prototipe (*proof-of-concept*) yang berfungsi untuk memfasilitasi pengguna akhir (*end-user*) dalam mengunggah citra pindaian dan menjalankan proses ekstraksi teks secara otomatis.

2) Pemanggilan Data Uji (Output B)

Sebagai masukan untuk pengujian fungsionalitas sistem, digunakan Output B (90% Data Uji Karakter). Himpunan data ini terdiri dari 90 lembar pindaian *wongoji* (memuat karakter dari 45 responden) yang telah diisolasi sejak Alur 1. Karena tidak pernah dilibatkan dalam proses pelatihan model (*unseen data*), Output B menjadi instrumen yang valid untuk menguji kemampuan inferensi arsitektur secara objektif..

3) Pengujian Sistem (Uji Operasionalisasi Inferensi) dan Penanganan Eksepsi

Data Output B diuji melalui serangkaian tahapan komputasi secara *end-to-end*: mulai dari pengunggahan gambar mentah, pemotongan (*cropping*) petak karakter, inferensi model hibrida 8-qubit, hingga penerjemahan menjadi teks yang dapat dibaca manusia (*Human-Readable*) pada antarmuka *frontend*.

Untuk mencegah eksepsi dan kegagalan fungsi akibat dokumen masukan yang tidak sesuai standar, sistem pada dasbor web menerapkan *Strict Wongoji Validation* pada tahap awal pemrosesan. Sistem diatur agar hanya menerima dan memproses dokumen yang terdeteksi memiliki minimal 100 sel karakter, dengan sekurang-kurangnya 5 baris horisontal dan 10 kolom vertikal. Apabila struktur *grid* tersebut tidak terdeteksi secara valid, algoritma akan secara otomatis menolak memproses citra tersebut. Mekanisme pembatasan ketat ini dirancang untuk mencegah sistem mengekstraksi dan menebak gambar acak di luar standar *wongoji*.

Selanjutnya, saat proses inferensi dilakukan lapis demi lapis pada sel karakter, mekanisme penanganan eksepsi sistem diaktifkan. Sistem menerapkan *Softmax Thresholding* berjenjang untuk menyaring probabilitas kelas yang rendah (*Out-of-Vocabulary / OOV Rejection*). Pada pembacaan sel secara utuh (*Full-Cell*), karakter dengan tingkat probabilitas di bawah 50% akan ditolak dan ditampilkan sebagai

karakter '[?]'. Selain itu, karena format wongoji memungkinkan penulisan dua angka di dalam satu kotak, sistem menerapkan *Conditional Splitter* berbasis *Connected Component Analysis* dan *Vertical Projection Profile*. Modul ini dirancang dengan syarat mutlak untuk mendeteksi renggangan fisis antar karakter (*Vertical Gap*) dan memutusnya menjadi dua citra mandiri hanya jika model QCNN memprediksi kelas awal sebagai numerik. Pada tahap inferensi lanjutan untuk citra hasil potongan tersebut (*Sub-Inference*), batas toleransi OOV disesuaikan menjadi 30% untuk mengakomodasi penurunan detail piksel pada karakter yang lebih sempit.

Adapun alur logika sistem dalam memvalidasi dokumen dan menangani eksepsi multi-karakter (*Conditional Splitter*) saat inferensi secara *end-to-end* direpresentasikan pada *Pseudocode* 2 berikut:

***Pseudocode* 2: Inferensi *End-to-End* dengan Validasi Wongoji & *Splitter* Bersyarat**

```
FUNCTION ProcessWongoji(scanned_image):
    detected_grid = DetectGridStructure(scanned_image)

    IF (TotalCells(detected_grid) < 100) OR (TotalRows < 5) OR (TotalColumns < 10) THEN:
        RETURN "Error: Dokumen tidak valid atau bukan format Wongoji."
    END IF

    cleaned_grid = ApplyMorphologicalFilters(detected_grid)
    result_text = ""

    FOR EACH cell_box IN cleaned_grid DO:
        probability, predicted_class = QCNN_ResNet18_Model(cell_box)

        IF DetectVerticalGap(cell_box) == TRUE AND predicted_class == "Numeric" THEN:
            image_1, image_2 = VerticalSplit(cell_box)
            prob_1, class_1 = QCNN_ResNet18_Model(image_1)
            prob_2, class_2 = QCNN_ResNet18_Model(image_2)

            char_1 = IF prob_1 > 30% THEN class_1 ELSE "[?]"
            char_2 = IF prob_2 > 30% THEN class_2 ELSE "[?]"

            result_text = result_text + char_1 + char_2

        ELSE:
            IF probability < 50% THEN:
                result_text = result_text + "[?]"
            ELSE:
                result_text = result_text + predicted_class
            END IF
        END IF
    END FOR

    RETURN result_text
END FUNCTION
```

4) Evaluasi Kinerja Sistem Berbasis Karakter

Kinerja akhir dari sistem inferensi *end-to-end* ini dievaluasi secara kuantitatif menggunakan metrik *Character Error Rate* (CER). Perhitungan CER dilakukan dengan membandingkan teks prediksi yang dihasilkan oleh *web* Django terhadap teks rujukan asli (*Ground Truth*) menggunakan metode jarak Levenshtein (*Levenshtein Distance*). Metrik CER ini akan mengukur rasio kesalahan sistem secara agregat, yang bersumber dari keterbatasan resolusi segmentasi citra maupun keterbatasan retensi model akibat *Catastrophic Forgetting*. Nilai evaluasi CER tersebut kemudian digunakan sebagai landasan kesimpulan operasional dari keseluruhan sistem.