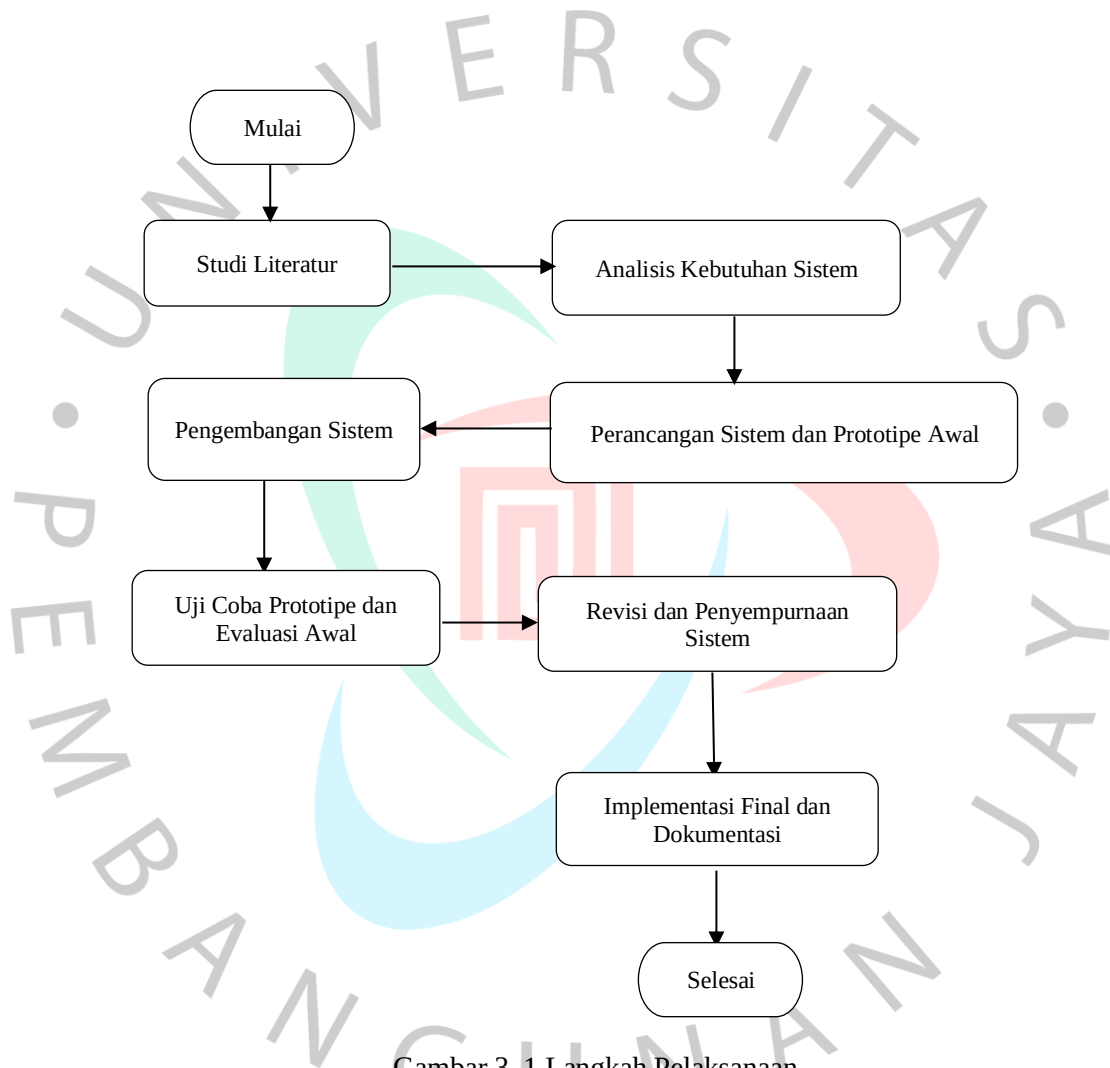


BAB III TAHAPAN PELAKSANAAN

3.1 Langkah – langkah pelaksanaan

Dalam tahapan pelaksanaan tugas akhir, peneliti menggunakan tabel yang dapat memberikan gambaran yang jelas. Berikut adalah langkah-langkah yang di terapkan:



Gambar 3. 1 Langkah Pelaksanaan

Pelaksanaan pada penelitian ini dirancang secara terstruktur dengan mengacu pada metode prototyping, yang memungkinkan proses pengembangan sistem dilakukan secara bertahap dan melibatkan interaksi langsung dengan pengguna. Setiap fase dalam proses ini disusun untuk memastikan bahwa solusi yang dibangun mampu mengatasi permasalahan dalam validasi data secara otomatis, dengan memanfaatkan integrasi antara teknologi OCR, NLP, dan metode CRF, Berikut uraian detail setiap tahap:

1. Studi Literatur dan Observasi Lapangan

Tahap awal dilakukan untuk memperoleh pemahaman konseptual dan teknis mengenai teknologi yang akan digunakan. Aktivitas yang dilakukan meliputi:

- a) Menelaah jurnal-jurnal ilmiah, laporan tugas akhir, dan buku teks yang relevan mengenai OCR, NLP, dan CRF;
- b) Mengidentifikasi kendala umum dalam proses entri data manual;
- c) Melakukan observasi di lingkungan nyata (instansi/organisasi) untuk memahami proses entri data yang saat ini berjalan secara konvensional.

2. Analisis Kebutuhan Sistem

Tahap ini bertujuan untuk mengidentifikasi kebutuhan pengguna dan merumuskan spesifikasi sistem. Aktivitasnya meliputi:

- a) Menentukan jenis dokumen yang akan diproses (misalnya formulir isian atau dokumen cetak);
- b) Menentukan data yang perlu diekstrak dan divalidasi (misalnya: nama, NIK, tanggal, kode tertentu);
- c) Menyusun use-case diagram atau skenario penggunaan sistem;
- d) Merancang kebutuhan fungsional (apa yang harus dilakukan sistem) dan non-fungsional (kecepatan, akurasi, kemudahan penggunaan).

3. Perancangan Sistem dan Prototipe Awal

Pada tahap ini dilakukan perancangan awal sistem, baik dari sisi antarmuka maupun alur data. Rancangan mencakup:

- a) Arsitektur sistem berbasis client-server;
- b) Desain tampilan antarmuka pengguna menggunakan HTML/CSS/JS;
- c) Alur kerja mulai dari input gambar, proses OCR, pengolahan teks NLP, hingga validasi data dengan CRF;
- d) Diagram alir proses (flowchart), dan diagram konteks sistem.

4. Pengembangan Sistem

Sistem dibangun berdasarkan desain yang telah disepakati. Pengembangan dilakukan secara modular agar tiap komponen bisa diuji secara terpisah. Beberapa tahapan penting meliputi:

- a) Implementasi proses OCR menggunakan Tesseract untuk ekstraksi teks dari gambar;
- b) Penerapan NLP (tokenisasi, normalisasi, POS tagging) menggunakan pustaka Python seperti spaCy atau NLTK;
- c) Integrasi algoritma CRF untuk mengenali pola entitas dan memvalidasi data;
- d) Pembuatan antarmuka web interaktif menggunakan HTML, CSS, dan JavaScript agar pengguna dapat mengunggah gambar dan melihat hasil validasi secara langsung.

5. Uji Coba Prototipe dan Evaluasi Awal

Prototipe diuji dengan dataset awal berupa dokumen gambar yang telah dipilih sebelumnya. Tujuannya adalah:

- a) Memastikan alur kerja sistem sudah berjalan sesuai rencana;
- b) Menilai keakuratan hasil ekstraksi dan validasi;
- c) Mencatat bug atau kelemahan sistem yang perlu diperbaiki;
- d) Mengumpulkan masukan dari pengguna tentang antarmuka dan kemudahan penggunaan.

6. Revisi dan Penyempurnaan Sistem

Setelah mendapat umpan balik dari pengguna dan hasil pengujian awal, dilakukan perbaikan pada bagian-bagian sistem yang kurang optimal. Langkah ini bisa berlangsung dalam beberapa iterasi tergantung kompleksitas masalah, dan mencakup:

- a) Perbaikan pada proses ekstraksi teks yang kurang akurat;
- b) Penyesuaian model CRF agar lebih tepat dalam mengenali pola data;
- c) Penyempurnaan tampilan atau responsivitas sistem.

7. Implementasi Final dan Dokumentasi

Setelah sistem dinyatakan stabil dan memenuhi kebutuhan pengguna, dilakukan:

- a) Pengujian akhir pada sistem secara keseluruhan;
- b) Penyusunan dokumentasi teknis dan laporan penelitian;
- c) Penyimpulan terhadap performa sistem berdasarkan hasil uji dan pengamatan.

3.2 Metode Pengujian

Dalam menguji kualitas perangkat lunak peneliti menggunakan metode pengujian:

3.2.1 Black box testing

Metode pengujian *black box* berfokus pada pengujian fungsi aplikasi tanpa memerlukan pemahaman tentang struktur kode internal. Tujuannya adalah untuk menjamin bahwa aplikasi beroperasi sesuai dengan spesifikasi yang telah ditentukan. Metode ini menguji semua input dan menentukan apakah output sesuai dengan harapan.

Berikut adalah proses pengujian *black box testing*:

Tabel 3. 1 Pengujian *Black box testing*

NO	Fitur	Keterangan	Hasil Sistem
1	Upload dokumen gambar	Pengguna memilih file gambar dari perangkat lokal dan menekan tombol "Upload" berformat JPG/PNG	Gambar berhasil diunggah dan ditampilkan
2	Ekstraksi teks (OCR)	Pengguna klik tombol "Proses OCR" setelah gambar tampil	Sistem menampilkan teks hasil ekstraksi dari gambar
3	Proses NLP dan normalisasi data	Pengguna klik tombol "Proses NLP" untuk membersihkan dan menstrukturkan teks	Sistem menampilkan teks yang telah dibersihkan dan dipisahkan per entitas
4	Validasi data menggunakan CRF	Pengguna klik tombol "Validasi Data"	Sistem menandai dan menampilkan data yang valid/tidak valid dengan label CRF
5	Export hasil validasi ke Excel/PDF	Tombol "Export" diklik	File hasil export berhasil diunduh

3.2.2 White box testing

White box testing melakukan pengujian struktur internal beserta kode program. Penguji melakukan pengujian dengan pemahaman mendalam tentang kode sumber dan logika program. Tujuan utama dari *white box testing* adalah untuk memeriksa alur kontrol, jalur data, kondisi, dan logika program secara keseluruhan.

Berikut adalah tabel untuk proses pengujian *white box testing* pada implementasi algoritma CRF

Tabel 3. 2 White Box Testing

No	Algoritma
1	Pengujian Modul OCR (Tesseract API)
2	Pengujian Modul NLP (Tokenisasi, Stopword)
3	Pengujian Modul CRF Validation

3.2.3 Pemecahan Masalah dan Strategi Implementasi

Dalam penelitian ini, pendekatan pemecahan masalah dilakukan dengan merancang sistem terintegrasi yang menggabungkan Optical Character Recognition (OCR), Natural Language Processing (NLP), dan algoritma Conditional Random Field (CRF) secara komprehensif dalam satu kerangka kerja. Kebaruan dari pendekatan ini terletak pada integrasi langsung antara ketiga komponen tersebut, yang belum banyak diterapkan secara lengkap dalam konteks validasi data invoice di Indonesia. Sebagian besar studi sebelumnya hanya mengadopsi dua komponen secara terpisah, seperti Rusli et al. (2020) yang menggunakan OCR dan NLP secara sederhana tanpa penguatan model berbasis CRF, serta penelitian oleh Permana (2019) dan Husein (2022) yang mengimplementasikan CRF namun tanpa mengintegrasikan proses akuisisi data melalui OCR dan pemrosesan semantik dari NLP

1. Kategori Entitas yang Dievaluasi:

Untuk setiap invoice, sistem diminta mengenali entitas berikut:

- a. Tanggal Invoice
 - b. Nomor Invoice
 - c. Tanggal Terima
 - d. Deskripsi Barang
 - e. Jumlah/QTY
 - f. Harga
 - g. Total Invoice
- 2 Label Manual

No	Nama Entitas	Nilai yang Benar
1	Invoice Date	02.07.2025
2	Invoice Number	BA10007527
3	Received Date	(tidak tersedia)
4	Description Item 1	HF10TQ-P IPP FILM - PRIME
5	Description Item 2	HY3.8FY-P YARN 3.8 - PRIME
6	Description Item 3	BI5.0GA-P ICP INJECTION 5 - PRIME
7	Qty Item 1	15,000
8	Qty Item 2	55,000
9	Qty Item 3	20,000
10	Price Item 1	17,051
11	Price Item 2	15,829
12	Price Item 3	18,198
13	Total Amount	1,654,255,200

a. Hasil Pengujian Sistem NER:

Entitas yang seharusnya dikenali (Ground Truth): 13

Entitas yang berhasil dikenali dengan benar (True Positive): 12

Entitas yang dikenali tapi salah (False Positive): 1

Entitas yang terlewat / tidak dikenali (False Negative): 0

b. Rumus Evaluasi:

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$$

$$= 12 / (12 + 0) = 100\%$$

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$$

$$= 12 / (12 + 1) = 92.3\%$$

$$\text{F1-Score} = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$$

$$= 2 \times (1.0 \times 0.923) / (0.923 + 0.923) \approx 96\%$$

$$\text{Accuracy} = (\text{Jumlah entitas benar} / \text{Total entitas yang diuji})$$

$$= 12 / 13 = 92.3\%$$

Dari hasil di atas, error sebesar 8% terjadi bukan karena salah ekstraksi nilai, melainkan karena entitas “Tanggal Terima” tidak tersedia dalam teks (kosong pada dokumen PDF atau tidak terbaca oleh OCR).

Penyebab utama ketidakakuratan:

Entitas tidak dicetak dalam dokumen (field kosong) → sistem tidak bisa menebak tanpa informasi eksplisit.

Variasi posisi dan label field di tiap invoice → sistem perlu pendekatan lebih fleksibel terhadap layout.

OCR gagal membaca teks di area dengan resolusi rendah atau watermark/stempel manual.