# BAB II
# TINJAUAN PUSTAKA

## 2.1 Pencapaian Terdahulu

Penelitian ini mendasarkan pengembangannya pada berbagai penelitian terdahulu yang dijadikan sebagai referensi dan rujukan utama.

Tabel 2. 1 Pencapaian Terdahulu

| No. | Nama Peneliti | Publikasi | Hasil |
|---|---|---|---|
| 1. | Rizal Muhamd Aldi (2022) | ImpelmentasiI OCR Dengan Metode Autoencoder Untuk Verifikasi Data KTP. Thesis (S1), FIK/INFO. 22 082. 26. Sep 2022 . | Penelitian ini memiliki tujuan agar gambar menjadi lebih bersih dan hasil OCR menjadi lebih akurat. Hal ini di dasari keperluan mendaftar akun dalam webside atau aplikasi |
| 2. | Kristian Adi Nugraha. (2024) | Penerapan Optical Character Recognition Untuk Pengenalan Variasi Teks Pada Media Persentasi Pembelajaran Jurnal Buana informatika , Volume 15, Nomor 01 ,April 2024 | Penelitian ini memiliki latar belakang permasalahan media pembelajaran digital umumnya tersimpan dalam bentuk citra karena memiliki unsur visual bentuk citra dianggap sebagai gambar |
| 3. | Firhan Maulana R., Kevin Akbar A, Hendy I. (2021) | Indonesia Id Card Extractor Using Optical Character Recognition and Natural Language Post Processing | Penelitian ini memiliki latar belakang untuk meningkatkan akurasi dalam mengekstrak yang menghasilkan tesk dari gambar |

| | | | |
|---|---|---|---|
| 4. | A Syarif R., Tubagus M A. (2020) | Kombinasi Metode NER – OCR untuk Meningkatkan efesiensi Pengambilan informasi Diposter berbahasa indonesia Vol 8 issue 4. Oktober 202o. | Penelitian ini memiliki latar Belakang dimana penyelengara Acara di Indonesia sering Menggunakan media poster digital Namun proses manula untuk Menstranfer informasi dari poster Digital ke situs web sering terkenda. |
| 5. | Nehru, Yosi Raduas H., Amelinda Callsta D I. (2024) | Implementasi cannyfilter Optical Character Recognition (OCR) untuk Identifikasi Tanda Nomor Kendaraan Bermotor Volume 6 No.1 2024 | Penelitian ini memiliki latar Belakang memonitoring kendaraan Yang melanggar aturan lalu lintas Tetapi pencatatan masih dilakukan Secara manual oleh operator |

## 2.2 Tinjauan Teoritis

Tinjauan teoritis adalah bagian dari penelitian yang menyajikan dan menganalisis kerangka konseptual serta teori-teori yang mendukung atau relevan dengan topik penelitian yang sedang dijalankan. Tujuan dari tinjauan teoritis adalah untuk memberikan dasar pemahaman yang kuat tentang landasan konseptual dan teoritis yang melatar belakangi penelitian.

### 2.2.1 Optical Character Recognition (OCR)

OCR memungkinkan komputer menerjemahkan teks dari gambar atau dokumen cetak menjadi teks digital. Prosesnya melibatkan pra-pemrosesan gambar (seperti binarisasi, perataan, dan penghapusan noise), segmentasi karakter, pengenalan pola, dan pemrosesan ulang hasil ekstraksi Berdasarkan studi yang dilakukan oleh Meharuniza Nazeem dan rekan-rekannya (2024), sejumlah pustaka OCR seperti Tesseract, EasyOCR, MMOCR, dan PaddleOCR terbukti mampu memberikan akurasi tinggi, terutama pada bahasa-bahasa dengan sumber daya terbatas (low-resource languages)

Tesseract, misalnya, berhasil mencapai tingkat akurasi sebesar 92% untuk teks berbahasa Inggris dan hingga 93% pada beberapa bahasa lokal di India. Di sisi lain, pendekatan terbaru yang menggabungkan arsitektur Transformer secara hibrida, yaitu model DOTA, mampu meningkatkan ketepatan dalam mengenali urutan karakter (sequence OCR) dengan rata-rata akurasi sebesar 77,8% pada dataset uji standar. Peningkatan ini diperoleh dengan menyisipkan algoritma *Conditional Random Field* (CRF) sebagai mekanisme pelabelan berbasis urutan (DOTA Consortium, 2025). Kendati demikian, sistem OCR masih menghadapi kendala saat berhadapan dengan dokumen yang buram atau menggunakan jenis huruf yang tidak umum, sehingga dibutuhkan proses koreksi lanjutan untuk memperbaiki hasil ekstraksi teks..

### 2.2.2 Natural Language Processing (NLP)

NLP adalah cabang ilmu yang memungkinkan komputer memahami bahasa manusia. Tujuan utama dalam penelitian ini adalah menggunakan NLP untuk memproses teks dari OCR agar bisa dikenali struktur serta konteksnya.

Proses NLP yang digunakan meliputi:

a. Tokenisasi: memisahkan teks menjadi kata atau frasa,
b. POS tagging: memberi label bagian kata berdasarkan fungsi sintaksis,
c. Stemming/Lematisasi: menyederhanakan kata ke bentuk dasarnya,
d. Named Entity Recognition (NER): mengenali dan menandai entitas seperti nama, tanggal, atau nomor.

Studi terkini Indigo-BiGRU-CRF (2023) untuk NER teks bahasa Indonesia menunjukkan peningkatan performa, mempertegas bahwa metode sequence modeling seperti CRF sangat efektif untuk dokumen lokal

:

### 2.2.3 Conditional Random Field (CRF)

CRF adalah algoritma pembelajaran terawasi yang digunakan untuk memberi label pada urutan data. Pada OCR+NLP, CRF efektif untuk memprediksi jenis entitas berdasarkan konteks sekeliling token.

Menurut Lafferty dkk. (2001), CRF mempertimbangkan ketergantungan label antar kata, sehingga mampu menghasilkan pemetaan entitas yang lebih akurat dibanding model kla-sik. Studi kombinasi BiGRU-CRF pada data teks Bahasa Indonesia (2023) menunjukkan efektivitasnya dalam meningkatkan skor F1 pada tugas ekstraksi informasi dan penelitian klinis CRF juga menegaskan bahwa CRF unggul dalam struktur teks berbasiskan entitas

1. Kelebihan CRF:

A. Memperhatikan Konteks secara Menyeluruh CRF mempertimbangkan ketergantungan antar label dalam satu rangkaian data. Hal ini memungkinkan sistem memahami konteks kalimat atau susunan elemen data lebih akurat dibanding model klasifikasi independen seperti Naive Bayes.

B. Menghindari Masalah Label yang Tidak Konsisten dengan menggunakan pendekatan global (global normalization), CRF dapat menghindari kombinasi label yang tidak masuk akal, karena model belajar dari keseluruhan struktur sekuens, bukan dari prediksi token per token saja.

C. Fleksibel dalam Penambahan Fitur CRF memungkinkan integrasi berbagai jenis fitur seperti kata sebelumnya, panjang kata, kapitalisasi, dan lainnya tanpa harus membuat asumsi independensi antar fitur.

D. Performa Unggul untuk NER dan Validasi Teks Terstruktur berdasarkan studi oleh Lample et al. (2016) dan penelitian lokal seperti Rizka & Harjoko (2021), CRF menunjukkan hasil yang lebih stabil dan akurat dalam menangani teks formal seperti formulir data, artikel, maupun dokumen administratif.

2 Kekurangan CRF:

a. Waktu Pelatihan Lebih Lama karena sifatnya yang menghitung dependensi antar label secara menyeluruh, pelatihan model CRF bisa memakan waktu lama, terutama pada dataset besar atau sangat kompleks.

b. Memerlukan Ekstraksi Fitur Manual tidak seperti deep learning modern (misalnya LSTM atau BERT), CRF tidak melakukan learning terhadap representasi data. Oleh karena itu, performanya sangat bergantung pada kualitas fitur yang dirancang secara manual oleh peneliti.
c. Tidak Cocok untuk Data Tidak Terstruktur atau Ambigu untuk teks yang sangat bebas (seperti komentar media sosial), CRF kurang andal karena struktur datanya sulit dipelajari hanya dari fitur statistik permukaan.
d. Kesulitan dalam Generalisasi jika Overfitting bila jumlah fitur terlalu banyak atau tidak relevan, CRF bisa mengalami overfitting terhadap data latih, sehingga kurang efektif pada data uji yang belum dikenali sebelumnya

.

### 2.2.4 Validasi Entri Data

Validasi data memastikan bahwa output digital telah sesuai format, struktur, dan isi yang benar sebelum disimpan dan digunakan. Jenis validasi umum mencakup:

a) Validasi format (misalnya pola tanggal DD-MM-YYYY),
b) Validasi nilai (rentang atau tipe data),
c) Validasi logika (kontrol kesesuaian antar entitas, seperti tanggal lahir lebih kecil dari tanggal entri)

Dengan integrasi OCR + NLP + CRF, validasi dapat dilakukan secara otomatis dengan mempertimbangkan pola dan konteks, bukan hanya aturan statis. Misalnya, CRF membantu membedakan antara tanggal dan angka seri berdasarkan urutan token—suatu pendekatan yang lebih adaptif dan kontekstual dibanding validasi manual.

### 2.2.5 Metodologi Pengembangan Sistem

Metodologi pengembangan sistem digunakan sebagai panduan dalam merancang, membangun, dan menguji sistem yang diusulkan. Dalam penelitian ini, pendekatan yang digunakan adalah **metode prototyping**, yaitu metode yang menekankan pada pembuatan versi awal (prototipe) dari sistem yang kemudian diperbaiki secara bertahap berdasarkan masukan pengguna.

1. Pemilihan Metode Prototyping
   Metode prototyping dipilih karena:

1 Membantu peneliti dan pengguna memperoleh pemahaman awal tentang sistem yang sedang dikembangkan.
2 Memungkinkan evaluasi fungsionalitas sejak tahap awal.
3 Cocok untuk sistem dengan kebutuhan yang dapat berubah atau disesuaikan melalui uji coba langsung.
4 Mendorong partisipasi aktif pengguna dalam siklus pengembangan sistem.

Metode ini sangat relevan dalam konteks pengembangan sistem validasi entri data berbasis OCR dan NLP, karena memungkinkan pengujian langsung terhadap fungsionalitas seperti ekstraksi teks, pengenalan entitas, serta mekanisme validasi otomatis sebelum sistem dikembangkan secara penuh.

2. Tahapan Metode Prototyping

1. Pengumpulan Kebutuhan (Requirement Gathering)
   Tahap awal melibatkan identifikasi kebutuhan sistem berdasarkan studi literatur, observasi proses entry data secara manual, dan konsultasi dengan calon pengguna.

2. Pembuatan Prototipe Awal (Build Prototype)
   Prototipe sistem awal dikembangkan menggunakan HTML, CSS, dan JavaScript sebagai antarmuka pengguna, serta Python atau Node.js pada sisi server untuk menangani proses OCR, NLP, dan CRF.

3. Evaluasi dan Umpan Balik Pengguna (User Evaluation & Feedback)
   Prototipe diuji oleh pengguna untuk mengamati kinerja sistem, kejelasan tampilan, serta akurasi validasi data. Masukan dari pengguna dicatat untuk perbaikan selanjutnya.
4. Pengumpulan Kebutuhan (Requirement Gathering)
   Tahap awal melibatkan identifikasi kebutuhan sistem berdasarkan studi literatur, observasi proses entry data secara manual, dan konsultasi dengan calon pengguna.
5. Pembuatan Prototipe Awal (Build Prototype)
   Prototipe sistem awal dikembangkan menggunakan HTML, CSS, dan JavaScript sebagai antarmuka pengguna, serta Python atau Node.js pada sisi server untuk menangani proses OCR, NLP, dan CRF.
6. Evaluasi dan Umpan Balik Pengguna (User Evaluation & Feedback)
   Prototipe diuji oleh pengguna untuk mengamati kinerja sistem, kejelasan tampilan, serta akurasi validasi data. Masukan dari pengguna dicatat untuk perbaikan selanjutnya.
7. Penyempurnaan Sistem (Refinement)
   Berdasarkan umpan balik, sistem disesuaikan dan diperbaiki hingga mencapai performa dan fungsionalitas yang diharapkan. Proses iteratif ini dapat dilakukan beberapa kali
8. Implementasi Akhir (Final Implementation)
   Setelah prototipe dianggap stabil dan sesuai dengan kebutuhan, sistem diimplementasikan secara penuh dan diuji secara menyeluruh