

# ***Intelligent Document Processing Berbasis OCR + Transformers dan CNN untuk Verifikasi Dokumen Bantuan Pangan***

DOI: <http://dx.doi.org/10.35889/jutisi.v14i3.3351>

Creative Commons License 4.0 (CC BY – NC)

**David Bagas Santoso<sup>1\*</sup>, Muhammad Fachrie<sup>2</sup>**<sup>1</sup>Informatika, Universitas Teknologi Yogyakarta, Yogyakarta, Indonesia<sup>2</sup>Sains Data, Universitas Teknologi Yogyakarta, Yogyakarta, Indonesia

\*e-mail Corresponding Author: Bagasdavit14@gmail.com

## **Abstract**

*The rice assistance program is a government initiative to alleviate poverty by distributing rice to low-income households. This study aims to design and implement an Intelligent Document Processing system based on Convolutional Neural Networks (CNN) and hybrid Optical Character Recognition (OCR) to facilitate the verification of social assistance documents. The system integrates the LayoutLMv3 model for text entity extraction and a ResNet-18 CNN for signature-box classification, supported by conventional OCR and TrOCR for challenging text regions. The research methodology covers system design and model development. Experimental results show that the system can successfully convert PDF documents into images and accurately recognize key entities such as names and national ID numbers using LayoutLMv3 (validation accuracy 96.78%). Although ResNet-18 achieves a validation accuracy of 99.04%, it remains biased toward majority classes and is not yet reliable for verification decisions. The proposed system has the potential to accelerate and standardize document verification, with future work focusing on dataset rebalancing and improving the signature classification model.*

**Keywords:** LayoutLMv3; ResNet-18; Optical Character Recognition; TrOCR

## **Abstrak**

Program bantuan pangan beras merupakan upaya pemerintah dalam menanggulangi kemiskinan melalui distribusi beras kepada masyarakat kurang mampu. Penelitian ini bertujuan merancang dan mengimplementasikan *Intelligent Document Processing* berbasis *Convolutional Neural Network (CNN)* dan *OCR hibrida* untuk mempermudah proses verifikasi dokumen bantuan pangan. Sistem ini mengintegrasikan model *machine learning LayoutLMv3* untuk ekstraksi entitas teks serta *CNN ResNet-18* untuk klasifikasi tanda tangan, dengan dukungan *Optical Character Recognition (OCR)* dan *TrOCR*. Metode penelitian mencakup perancangan sistem dan pengembangan perangkat. Hasil pengujian menunjukkan bahwa sistem mampu mengekstraksi dokumen PDF menjadi citra, mengenali entitas seperti nama dan NIK dengan akurasi tinggi menggunakan LayoutLMv3 (*val\_accuracy* 96,78%), serta mengklasifikasikan kotak tanda tangan dengan ResNet-18 (*val\_accuracy* 99,04%). Namun, ResNet-18 masih bias terhadap kelas mayoritas sehingga belum dapat dianggap andal untuk keputusan verifikasi. Sistem yang diusulkan berpotensi mempercepat pemeriksaan dokumen bantuan pangan, sementara pengembangan lanjut difokuskan pada penyeimbangan data dan penyempurnaan model klasifikasi tanda tangan.

**Kata kunci:** LayoutLMv3; ResNet-18; Optical Character Recognition; TrOCR

## **1. Pendahuluan**

Menurut Badan Pangan Nasional (2024) Bantuan Pangan Beras (Banpang) adalah program pemerintah berupa penyaluran beras yang bersumber dari stok Cadangan Beras Pemerintah (CBP) sebagai intervensi pemerintah dalam menanggulangi kemiskinan dan meningkatkan kesejahteraan masyarakat. Program ini merupakan salah satu pemanfaatan CBP

sesuai amanat Peraturan Presiden Nomor 125 Tahun 2022 tentang Pengolahan Cadangan Pangan Pemerintah [1]. Menurut data Badan Pusat Statistik (BPS), hingga Maret 2024 sekitar 25,22 juta penduduk Indonesia atau 9.03% dari total penduduk masih hidup dibawah garis kemiskinan [2].

Salah satu tantangan utama dalam pelaksanaan program Banpang adalah proses verifikasi data Penerima Bantuan Pangan (PBP). Dengan jutaan PBP yang tersebar di berbagai wilayah, keabsahan data penerima menjadi sangat penting untuk memastikan bantuan diterima oleh penerima yang bersangkutan. Perum BULOG Kanwil Yogyakarta, sebagai penanggung jawab distribusi bantuan pangan untuk wilayah Yogyakarta, memiliki tugas untuk memastikan data PBP di lapangan dengan yang tertulis di dokumen telah sesuai. Dalam proses konvensional yang telah diterapkan, banyak masalah yang muncul terkait dengan efisiensi operasional dan akurasi data.

Dokumen seperti, Daftar Nomor (Danom), Berita Acara Serah Terima (BAST) Perwakilan, Berita Acara Serah Terima (BAST) Pengganti, dan Surat Pernyataan Tanggung Jawab Mutlak (SPTJM) digunakan untuk mencatat data penerima asli, perwakilan, dan pengganti. Namun, sering kali terjadi ketidaksesuaian data seperti perbedaan Nama, NIK, dan Alamat. Selain itu, tanda tangan penerima asli, perwakilan, maupun pengganti sering kali tidak ada dan tidak tercatat dengan benar. Contoh proses verifikasi yang dilakukan adalah, jika dokumen Danom pada kolom terdapat tanda tangan, maka dianggap bantuan diambil langsung oleh penerima asli, sedangkan jika pada kolom terdapat keterangan W (wakil) atau G (Ganti), maka Danom nomor tersebut harus disesuaikan dengan dokumen BAST Perwakilan atau BAST Pengganti. Dalam praktiknya, banyak ditemukan masalah seperti NIK, Nama, Alamat PBP yang tidak tercatat sesuai keterangan yang ada selain itu proses verifikasi sangat memakan waktu sehingga operasional menjadi lama.

Untuk mengatasi permasalahan tersebut, penelitian ini mengusulkan pendekatan 2 model *machine learning* berbasis *LayoutLMv3* [3] untuk memahami struktur dari dokumen, yang diintegrasikan menggunakan OCR untuk ekstraksi teks [4], jika gagal akan di *fallback* oleh *TrOCR* [5] dengan *single* penyimpanan. Selanjutnya *Convolutional Neural Network (CNN)* digunakan untuk mengklasifikasikan jenis tanda tangan dengan integrasi *ResNet* [6]. Solusi ini di nilai tepat karena dapat meningkatkan akurasi ekstraksi teks dan mengklasifikasikan jenis tanda tangan, seperti beberapa studi sebelumnya [7][8][9][10][11] yang juga menunjukkan keberhasilan.

Penelitian ini bertujuan merancang sistem verifikasi dokumen penerima Banpang berbasis web yang memudahkan pegawai Perum BULOG melakukan verifikasi secara otomatis menggunakan teknologi *hybrid OCR*. Manfaat yang dihasilkan adalah percepatan dan peningkatan akurasi proses verifikasi, fitur deteksi otomatis terhadap ketidaksesuaian NIK, nama, alamat, no urut, pekerjaan, sebab penggantian, dan tanda tangan mengurangi ketergantungan pada pengecekan konvensional satu per satu dokumen, sehingga alur kerja menjadi lebih efisien, tepat, dan terukur. Dengan demikian, terbentuk *Intelligent Document Processing (IDP)* secara *end-to-end* yang mampu menurunkan waktu verifikasi dan meminimalkan kesalahan pencatatan di lapangan. Kontribusi utama penelitian ini meliputi perancangan pipeline *LayoutLMv3 + hybrid OCR* untuk ekstraksi entitas, rancangan *CNN-ResNet* untuk klasifikasi tanda tangan (TTD), serta aturan verifikasi / *rule-based* lintas dokumen (Danom, BAST Perwakilan / Pengganti, SPTJM) yang terstandarisasi. Kinerja dievaluasi menggunakan *macro-F1 (NER)* dan akurasi klasifikasi TTD.

## 2. Tinjauan Pustaka

Penelitian oleh T. Heriyanto et al. [12] menggunakan OCR untuk verifikasi berkas tugas akhir mahasiswa sebagai syarat akhir kelulusan studinya. OCR dipakai untuk verifikasi berkas di *Digital Library* untuk mengubah berkas menjadi teks agar pemeriksaan lebih cepat. Kemudian hasil ekstraksi di bandingkan dengan teks dari aplikasi, dimana menunjukkan sistem mampu menyaring tujuh dari delapan dokumen. Penelitian lain [4][9] mengungkapkan bahwa OCR menunjukkan hasil yang positif dalam mengekstrak data pada KTP dan KK, sehingga mengurangi kesalahan input dan mempercepat proses verifikasi. Menurut penelitian [13] tantangan utama OCR adalah variasi *template* dokumen. Pada dokumen tagihan pembelian, OCR difokuskan pada penemuan nomor *invoice* dan total tagihan, dengan hal ini adaptasi terhadap struktur dokumen sangat penting. Selain itu kualitas hasil OCR sangat dipengaruhi mutu citra, *fonts*, dan praproses. Evaluasi lintas jenis *font* menunjukkan gap besar, beberapa *font* seperti *Georgia*, *LucidaSans Unicode*, *Times New Roman* terbaca hampir 100%, lalu untuk jenis *font Roman* memiliki akurasi

sebesar 98.48% sementara *Bahnschrift Condensed* turun hingga 71.21% yang menandakan kebutuhan adaptasi model terhadap variasi tipografi sangat penting [14].

Penelitian oleh M. Li et al. [5] mengusulkan pemrosesan teks dengan pendekatan *Transformer* yaitu *TrOCR*. Dari hasil percobaan, *TrOCR* menggunakan satu sintetis set pelatihan menunjukkan hasil positif pada teks cetak dan tulisan tangan hanya dengan model *encoder-decoder* tanpa langkah pasca-pemrosesan. Penelitian oleh L. Huttner et al. [15] yang mencoba melakukan penyempurnaan pengenalan tulisan tangan dengan eksplorasi *trade-off TrOCR* yang dilatih secara sintetis untuk pengenalan teks tulisan tangan dengan *LoRA* dan *DoRA* menunjukkan bahwa *LoRA/DoRA* tidak meningkatkan kinerja pengenalan tulisan tangan. Penelitian lain [16][17] memperkenalkan *TrOCR* untuk penyempurnaan model berbasis *Transformer* yang terdiri dari encoder yang dilatih untuk satu halaman dokumen penuh. Hasil eksperimen menunjukkan strategi ini lebih efektif dalam meningkatkan kemampuan generalisasi, model *TrOCR* mengungguli *baseline* dibandingkan yang hanya dilatih dalam potongan-potongan dokumen kecil.

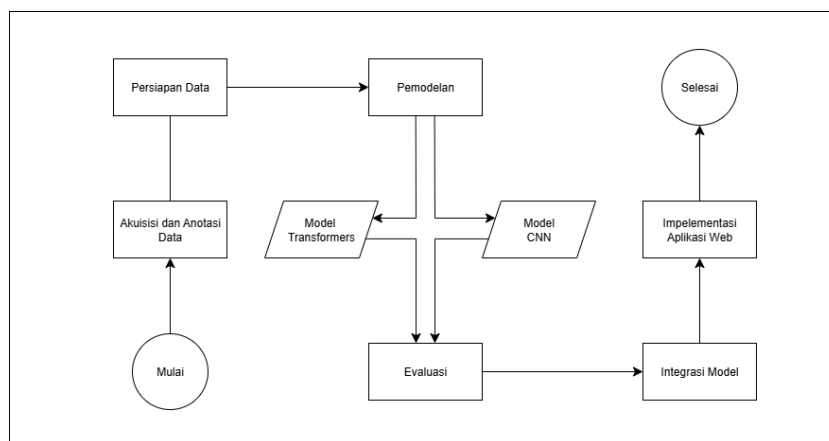
Penelitian oleh P. Hu et al. [18] menampilkan keterbatasan *LayoutLMv3* dimana token dibatasi hingga 512. Akibatnya proses dokumen yang padat teks perlu menggabungkan strategi jendela geser, yang dapat mengakibatkan hilangnya informasi global dokumen. Kemudian L. Wang et al. [8] mengusulkan untuk melakukan regresi pada model terbaru *LayoutLMv3* dengan merekonstruksi token *patch* dari modalitas visual. Meskipun penyelarasan antara gambar dan teks telah mengalami peningkatan, teknik ini membutuhkan biaya dan waktu komputasi yang sangat besar. Selain itu *LayoutLMv3* menunjukkan bahwa model tersebut terlalu banyak informasi tata letak yang membuat model melakukan prediksi bukan memahaminya.

Penelitian oleh B. M. Sujatmiko et al. [10] yang mengembangkan *Multilayer Perceptron (MLP)* dalam melakukan identifikasi terhadap penyakit tumor kulit menggunakan *CNN* dengan desain jaringan *ResNet* untuk membantu diagnosis. Hasil klasifikasi tumor kulit dengan menggunakan *CNN* dengan variasi desain arsitektur *ResNet* dapat mengklasifikasi tumor kulit dengan akurasi lebih dari 90%. Penelitian oleh [6] terkait klasifikasi citra tanda tangan untuk mengklasifikasi keaslian tanda tangan pemilik menggunakan *CNN* dengan arsitektur *ResNet* dan *VGG*. Pada tahap awal dilakukan *preprocessing* dengan mengubah ukuran citra menjadi 224x224 *pixel*. Hasil penelitian menunjukkan bahwa kedua arsitektur mampu memprediksi tanda tangan sesuai antara label prediksi dengan label asli, hal ini dipengaruhi oleh nilai akurasi yang tinggi dan nilai *loss* yang rendah.

Berbeda dengan penelitian sebelum-sebelumnya, penelitian ini mengusulkan sistem berbasis *OCR hybrid + Transformer* dan *CNN* untuk efisiensi operasional verifikasi dokumen Banpang. Sistem yang dihasilkan diharapkan mempercepat dan meningkatkan akurasi proses verifikasi, fitur deteksi otomatis terhadap ketidaksesuaian NIK, nama, alamat, no urut, pekerjaan, sebab penggantian, dan tanda tangan sehingga mengurangi kesalahan pada pengecekan konvensional satu per satu dokumen, sehingga alur kerja menjadi lebih efisien, tepat, dan terukur. Dengan pendekatan ini, terbentuk *Intelligent Document Processing (IDP)* secara *end-to-end* yang mampu menurunkan waktu verifikasi dan meminimalkan kesalahan pencatatan. Oleh karena itu, kebaruan penelitian ini terletak pada integrasi model *hybrid* dengan 2 tugas yang berbeda *OCR + LayoutLMv3* untuk ekstraksi entitas data, dan *CNN* untuk klasifikasi jenis tanda tangan, sehingga mampu mengoptimalkan proses verifikasi.

### 3. Metodologi

Penelitian ini menggunakan *Framework PyTorch* dan *HuggingFace Transformers* untuk melatih model *LayoutLMv3* (ekstraksi entitas/NER) serta model *CNN ResNet-18* untuk klasifikasi jenis tanda tangan (TTD). Ekstraksi teks memanfaatkan *OCR hybrid* yaitu *Tesseract* sebagai mesin utama dan *TrOCR* sebagai *fallback* untuk tulisan tangan. Data masukan berupa PDF empat jenis dokumen, yaitu Danom, BAST Perwakilan, BAST Pengganti, dan SPTJM. Eksperimen dijalankan pada lingkungan Google Colab dengan *GPU NVIDIA Tesla T4* (VRAM 16 GB) dan *Python 3.x*. Pelatihan model menggunakan *PyTorch 2.x* (*CUDA* kompatibel bawaan Colab) serta pustaka *Transformers*, *pdf2image*, *OpenCV*, dan *Tesseract*. Penelitian ini ditempuh melalui enam fase, yaitu akuisisi dan anotasi data, persiapan data, pemodelan, evaluasi, integrasi model, serta implementasi aplikasi web. Skema alurnya ditunjukkan pada Gambar 1.



**Gambar 1.** Tahapan Penelitian

### 1) Akuisisi & Anotasi Data

Dataset diperoleh dari dokumen operasional Banpang milik Perum BULOG Kanwil Yogyakarta yang mencakup Danom, BAST Perwakilan, BAST Pengganti, dan SPTJM dalam format PDF. Proses anotasi dilakukan dua jalur, pertama ada *Anotasi NER* untuk *LayoutLMv3*. Setiap halaman diberi *bounding box* pada area informasi kunci (Nama, NIK, Alamat, No Urut, Pekerjaan, Sebab Penggantian, dan TTD). Teks kata (*words*) diperoleh dari OCR awal lalu dikurasi manual. Hasilnya disimpan terlebih dahulu sebagai CSV dan dikonversi ke JSON dengan skema, *no*, *page\_id*, *word*, *label (BIO)*, *koordinat (x1, y1, x2, y2)* dan *doc\_type*. Lalu yang kedua ada *Anotasi ROI TTD* untuk *ResNet-18*. Setiap kotak TTD dicrop menjadi citra kecil dan disertai *.txt* perhalaman dengan format YOLO untuk pelatihan klasifikasi isi kotak. Skema kelas yang digunakan adalah 1 = *signature*, 2 = *w* (wakil), 3 = *g* (ganti), 4 = kosong, 5 = *ttd\_dtt*. Sebagai acuan verifikasi lintas dokumen, disusun *ground truth relational* yang memetakan setiap baris Danom ke padanannya pada BAST Perwakilan / Pengganti serta SPTJM menggunakan kunci No urut dan kecocokan Nama-NIK. Data yang berlabel ini menjadi dasar evaluasi ekstraksi entitas dan kebenaran aturan verifikasi pada tahap berikutnya.

### 2) Persiapan Data

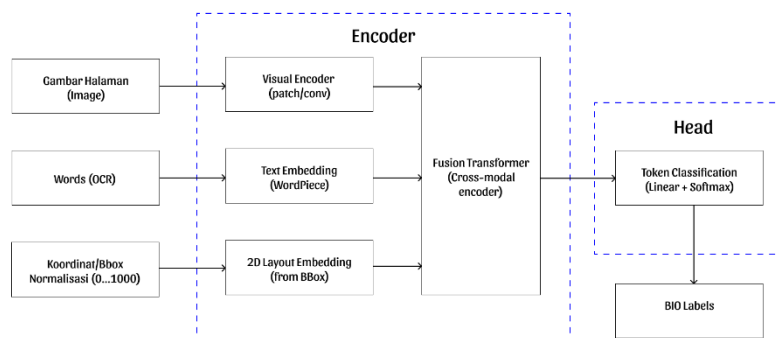
Tahap ini menyiapkan data berlabel agar konsisten dan siap digunakan pada dua model yaitu: ekstraksi entitas *LayoutLMv3 (NER)* dan klasifikasi tanda tangan *CNN-ResNet-18*. Seluruh PDF dikonversi menjadi citra per halaman ( $\pm 300$  DPI), lalu dilakukan *deskew*, *denoise* ringan, dan koreksi orientasi (*OpenCV*). Ekstraksi teks memakai *OCR hybrid*, dengan *tesseract* sebagai mesin utama, dan *TrOCR* sebagai *fallback* pada segmen berkepastian rendah, sehingga tiap kata memiliki *word* dan *bounding box (bbox)*. Koordinat *bbox* dinormalisasi menjadi 0-1000 dan divalidasi  $[x_0 \leq x_1, y_0 \leq y_1]$ . Label *BIO* diselaraskan dengan urutan kata/*bbox*, lalu token khusus/*overflow* diberi *labels = -100*. Dataset dibagi 80% data latih dan 20% data uji dengan stratifikasi per *doc\_type* (Danom, BAST Perwakilan, BAST Pengganti, SPTJM). Dari porsi data latih (80%), disisihkan 10% sebagai data validasi (*hold-out*) menggunakan *random seed* tetap untuk memantau *val-loss* dan *macro-F1*. *Class weights* dihitung dari distribusi label pada data latih saja, untuk *CNN* ditambah *WeightedRandomSampler*. Format akhir, dataset *NER* disimpan sebagai JSON berisi *image\_path*, *words*, *bbox(0..1000)*, *ner\_tags*, *doc\_type*. Untuk modul *CNN TTD*, setiap kotak TTD di-*crop*, di-*resize* 224×224, tidak dilakukan augmentasi ringan seperti *flip/rotate/color jitter*, dan anotasi koordinat disimpan dalam format YOLO dengan kelas (*signature*, *W*, *G*, kosong, *ttd\_dtt*). Seluruh paket data ini menjadi dasar pelatihan dan evaluasi pada tahap berikutnya.

### 3) Pemodelan

#### a. *LayoutLMv3 (Token Classification/NER)*

Model *LayoutLMv3* digunakan untuk mengekstrak entitas terstruktur dari dokumen. Model menerima tiga masukan serentak, citra halaman (hasil konversi PDF ke *image*  $\pm 300$  DPI), *words* hasil OCR dan koordinat *bbox* tiap kata yang telah dinormalisasi ke rentang 0...1000, *words* dan *bbox* menjadi inputan dalam bentuk satu halaman utuh, bukan per *words* agar model paham informasi global dokumen. Dengan *apply\_ocr=False*, kata dan *bbox* disuplai dari tahap

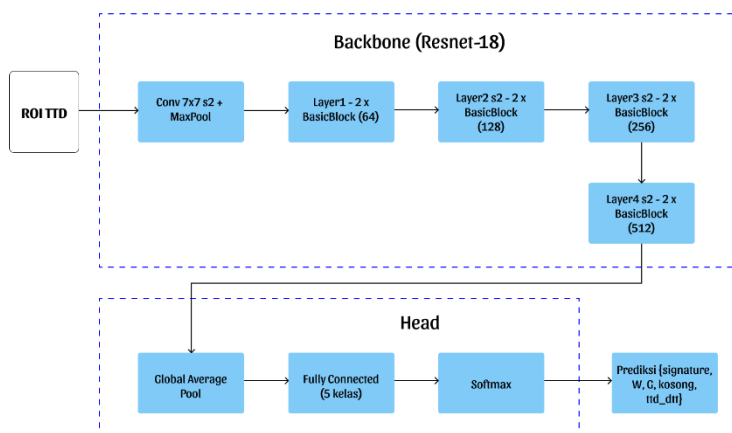
persiapan data. Di dalam *Encoder*, fitur visual diekstrak oleh *visual encoder*, kata ditransformasikan menjadi *Text Embedding (WordPiece)*, dan posisi spasial dikodekan sebagai *2D Layout Embedding* dari bbox. Ketiga sumber informasi difusikan melalui *Fusion Transformer (cross-modal encoder)*. Pada *Head*, *token classification (linear + softmax)* memetakan setiap token pada skema *BIO (B-ENT, I-ENT, O)*. Hasil token-level digabungkan menjadi *span* entitas *BIO* yang utuh. Kemudian semua *words* disimpan dalam bentuk halaman, meliputi Nama, NIK, Alamat, No urut, Pekerjaan dan lainnya. Arsitektur dapat dilihat pada Gambar 2.



**Gambar 2.** Arsitektur *LayoutLMv3*

b. *CNN (ResNet-18)*

Model *Resnet\_18* ini mengklasifikasikan isi kotak TTD per baris menjadi {signature, W, G, kosong, ttd\_dtt}. Masukan berupa ROI TTD ( $224 \times 224$ ) yaitu *crop* dari koordinat kotak TTD per halaman. Bagian *Backbone* mengikuti *ResNet-18*, yaitu *stem Conv 7×7 s2 + MaxPool*, diikuti *Layer1–Layer4* (masing-masing  $2 \times$  *BasicBlock* dengan kanal 64/128/256/512). Pada *Head*, fitur diringkas oleh *Global Average Pool*, kemudian *Fully Connected* (5 kelas) dan *Softmax* menghasilkan probabilitas kelas {signature, W, G, kosong, ttd\_dtt} untuk setiap ROI. Arsitektur model dapat dilihat pada Gambar 3.



**Gambar 3.** Arsitektur *ResNet-18*

c. *Training Model*

Pelatihan model dilaksanakan secara terpisah dengan pengaturan *hyperparameter* serupa. Data dibagi 80% latih / 20% uji, lalu sebagian kecil dari latih disisakan sebagai validasi (hold-out) dengan *seed* tetap. Pada eksperimen ini jumlah *epoch* ditetapkan (tanpa *early stopping*), *checkpoint* disimpan tiap *epoch* dan model terbaik dipilih berdasarkan *minimum validation loss*. Seluruh proses mencatat akurasi dan *macro-F1* pada validasi, serta *logging* ke CSV untuk pelacakan. Secara skematis, *LayoutLMv3* dilatih dalam dua fase, pertama *head-only* dimana *backbone* dibekukan dan hanya *classifier head* yang dilatih, kedua *fine-tuning end-to-end* dimana *backbone* dibuka dan seluruh parameter dilatih. Sedangkan *ResNet-18* untuk TTD juga dua fase, *head-only* lalu *fine-tuning end-to-end*, tanpa dilakukan augmentasi ringan pada ROI seperti *rotate/flip/color jitter*. Penelitian ini melibatkan dua komponen model, *LayoutLMv3 (NER)* dan *ResNet-18 (TTD)*. Rancangan pelatihan dirangkum pada Tabel 1 sebagai berikut.

Tabel 1 Inisialisasi Model Pelatihan

Komponen	Fase	Optimizer	LR	Epoch	Batch	Scheduler	Loss
LayoutLMv3	Freeze	AdamW	1e-5	5	2	-	Weighted CE
LayoutLMv3	Fine-tune	AdamW	5e-5	10	2	Linear warmup	Weighted CE
ResNet-18	Initial	AdamW	1e-3	10	32	-	CE
ResNet-18	Fine-tune	AdamW	3e-4	10	32	CosineAnnealing LR	berbobot CE

#### 4) Evaluasi

Model dilatih terpisah dengan pengaturan *hyperparameter* yang serupa dan dievaluasi pada data uji (20%) menggunakan *checkpoint* dengan *validation loss* terendah. Metrik juga dibedakan karena tugas yang berbeda, untuk *LayoutLMv3 (NER)* evaluasi dilakukan pada token valid (label -100 diabaikan). Metrik yang dilaporkan meliputi, akurasi token untuk melihat proporsi token yang benar, *precision* dan *recall* per-kelas entitas untuk menilai ketepatan dan ketercakupan prediksi, *F1* per-kelas untuk rata-rata harmonik *precision-recall*, serta *macro-f1* agar rata-rata *f1* tidak bias ke kelas *d=dominan*. Perhitungan diberikan pada Persamaan (1)-(5).

$$Accuracy_{token} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Precision_k = \frac{TP_k}{TP_k + FP_k} \quad (2)$$

$$Recall_k = \frac{TP_k}{TP_k + FN_k} \quad (3)$$

$$F1_k = \frac{2 Precision_k Recall_k}{Precision_k + Recall_k} \quad (4)$$

$$Macro-F1 = \frac{1}{K} \sum_{k=1}^K rF1_k \quad (5)$$

Untuk *CNN ResNet-18* (klasifikasi kotak TTD), digunakan evaluasi menggunakan akurasi keseluruhan, yaitu *precision/recall/F1* per-kelas, dan *macro-f1*, untuk analisis kesalahan didukung oleh *confusion matrix*. Perhitungan diberikan pada Persamaan (6) dan (7).

$$Accuracy = \frac{\sum_{k=1}^K rTP_k}{\sum_{k=1}^K (TP_k + FP_k + FN_k)} \quad (7)$$

$$Macro-F1 = \frac{1}{K} \sum_{k=1}^K \frac{2 TP_k}{2 TP_k + FP_k + FN_k} \quad (8)$$

#### 5) Integrasi Model

Integrasi dilakukan untuk mengubah keluaran *LayoutLMv3 (NER)* dan *ResNet-18* (TTD) menjadi keputusan verifikasi lintas dokumen. Entitas hasil *NER* pada tiap halaman (Nama, NIK, Alamat, No Urut, Tanda tangan, Sebab Penggantian, Pekerjaan, dll) dinormalisasi lalu digabung per baris danom dan dipautkan ke baris terkait pada BAST Perwakilan, BAST Pengganti, dan SPTJM menggunakan kunci NIK/No Urut (*upper-case*, *trim spasi*, hapus tanda baca). Pada saat yang sama, prediksi *CNN* pada ROI TTD (kelas: *signature*, W, G, kosong, *ttd\_dtt*) diletakkan ke baris tersebut. Selanjutnya aturan bisnis dievaluasi dengan logika sederhana, misalnya penerima asli valid jika *ttd\_cls = signature*, dan tidak memiliki padanan di BAST Perwakilan / Pengganti.

Lalu penerima wakil valid jika  $ket\_ttd = W$  dan  $ttd\_cls = W$  dan terdapat baris cocok di BAST Perwakilan dengan identitas penerima wakil lengkap. Sedangkan penerima ganti valid jika  $ket\_ttd = G$  dan  $ttd\_cls = G$  dan ada baris cocok di BAST Pengganti serta sepadan di SPTJM, dicap *inkonsistensi*. Setiap baris menghasilkan status (*Valid/Error*) dan deskripsi singkat yang ditampilkan pada antarmuka web, lalu daftar kesalahan verifikasi di simpan berupa *excel* dan *json* sehingga transporter dapat menindaklanjuti temuan dengan cepat.

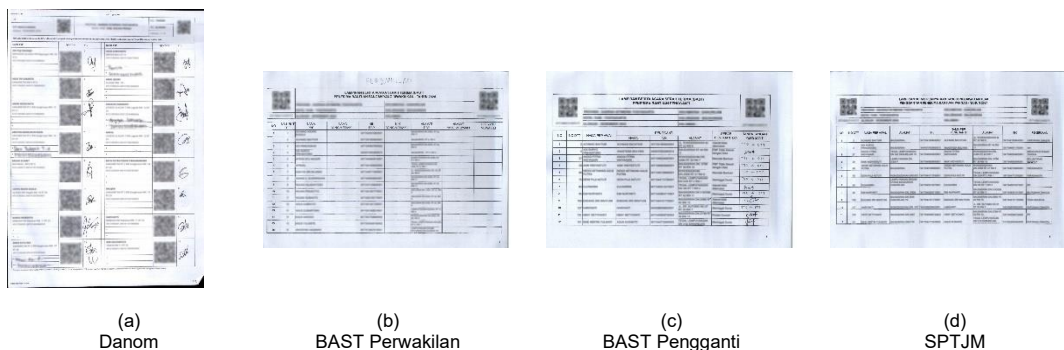
## 6) Implementasi Aplikasi Model

Implementasi sistem sebagai *website* berbasis *Flask (Python 3.x)*. Model *LayoutLMv3* dan *ResNet-18* disimpan sebagai file terpisah dan dimuat saat server mulai, lalu dipanggil pada proses unggah/olah dokumen. Alurnya pengguna mengunggah PDF, dimana halaman dikonversi ke gambar 300 DPI, penerapan *Hybrid OCR* menghasilkan *words* dan *bbox* terstandar (0...1000), kemudian *NER* dengan *LayoutLMv3* memetik entitas target (*BIO*). Klasifikasi TTD, dengan ROI TTD perbaris dicrop dengan format *YOLO* kemudian dinormalisasi 224x224, lalu diprediksi *ResNet-18* menjadi kelas *signature*, *W*, *G*, kosong, *ttd\_dtt*. Terakhir *Rule-engine* menyatukan keluaran *NER* + *CNN* lintas dokumen dan Danom ↔ BAST Perwakilan/Pengganti ↔ SPTJM dan menghasilkan status (*Valid/Error*). Sementara penyimpanan data, berkas PDF, citra halaman 300 DPI, dan crop ROI TTD disimpan di file sistem local dengan struktur teratur. *MySQL* menyimpan data terstruktur dan metadata, *words* hasil OCR, *bbox* 0...1000, entitas *NER*, prediksi *CNN*, keputusan aturan, serta *path* ke berkas di *disk*, bukan menyimpan CSV sebagai file. Format *CSV/JSON* hanya dipakai untuk *impor/ekspor*, sementara isinya (khususnya *words*) langsung di-*insert* ke tabel DB. Setiap record menyertakan *confidence* dan *timestamp* untuk kebutuhan audit. Antarmuka pengguna menampilkan progres verifikasi, tabel verifikasi dengan filter/pencarian, serta tombol unduh laporan. Mekanisme *auth* sederhana (pegawai/admin), pembatasan ukuran berkas dan *logging* kesalahan diterapkan untuk menjaga keamanan dan privasi. Dengan implementasi ini, model berjalan terintegrasi di layanan web, mulai dari unggah PDF hingga keluaran verifikasi yang siap ditindaklanjuti, sehingga mendukung operasi lapangan melalui tampilan yang ringkas, cepat, dan dapat diaudit.

## 4. Hasil dan Pembahasan

### 4.1. Hasil

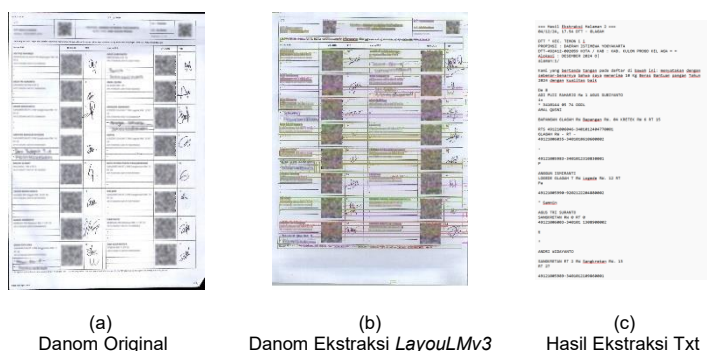
Penelitian ini mengakuisisi dataset secara mandiri dari berkas PDF dokumen Banpang (Danom, BAST Perwakilan, BAST Pengganti, dan SPTJM) yang berasal dari Kantor Perum BULOG Yogyakarta. Seluruh PDF dirender menjadi citra 300 DPI dan dilakukan *deskew* serta *denoise* ringan. Anotasi dilakukan manual pada dua level. Anotasi *NER (LayoutLMv3)* dimana setiap kata pada halaman diberi kotak pembatas (*bbox*) dan tag *BIO* untuk entitas target seperti, Nama, NIK, Alamat, No Urut, Pekerjaan, Sebab Pengantian, dll. Koordinat dinormalisasi ke skala 0-1000 dan diurutkan per baris/kolom agar selaras dengan urutan baca. Kemudian Anotasi ROI TTD (*ResNet-18*) setiap sel tanda tangan per baris Danom/BAST dicrop menjadi ROI 224x224 dan diberi label kelas {*signature*, *W*, *G*, kosong, *ttd\_dtt*}. Tiap sampel menyertakan meta informasi *doc\_type* dan *page\_id* untuk mendukung verifikasi lintas dokumen. Adapun contoh dari dokumen Banpang yang ditampilkan pada Gambar 4.



**Gambar 4.** Contoh Dokumen Banpang

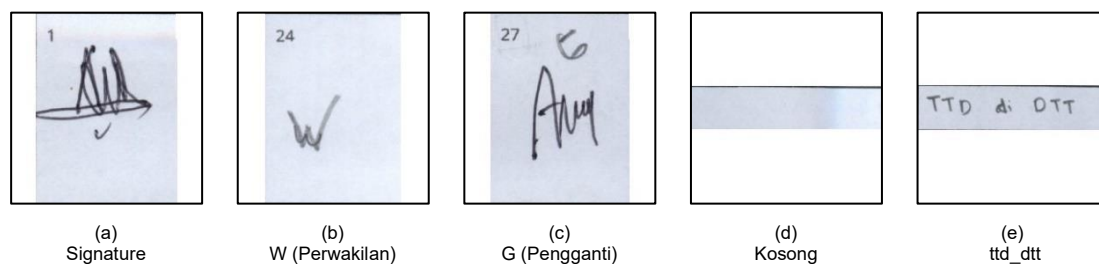


Untuk data *NER (LayoutLMv3)*, keluaran *OCR (Tesseract)* pada rancangan akhir disiapkan *fallback TrOCR* untuk kasus kepercayaan rendah / hasil ekstraksi dari *OCR* kosong dengan *single* penyimpanan. *Output* dipecah menjadi token, setiap token diberi *bounding box*, lalu koordinatnya dinormalisasi ke rentang 0-1000 agar sesuai format masukan *LayoutLMv3*. Dari total 232 halaman *Danom* dan *BAST*, 84 halaman terpilih dianotasi secara manual, setiap kata di ketik ulang berserta labelnya, serta koordinat kotak ( $x_1$ ,  $y_1$ ,  $x_2$ ,  $y_2$ ). Anotasi tersebut kemudian diekstraksi menjadi berkas *JSON*, sehingga terkumpul 15.825 token beranotasi, token berlabel **O** tetap disimpan supaya konteks baris tidak hilang. Dokumen yang struktur tabelnya bergeser atau baris terpotong terlebih dahulu dibersihkan agar urutan token dan koordinat tidak meloncat. Adapun hasil anotasi ditampilkan pada Gambar 5.



**Gambar 5.** Hasil Anotasi *LayoutLMv3*

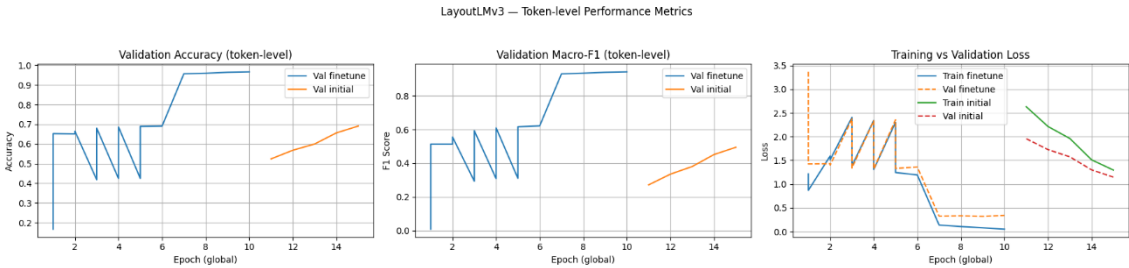
Untuk data klasifikasi *ROI TTD (ResNet-18)*, setiap baris penerima memiliki satu kotak tanda tangan yang diambil dari koordinat template / hasil deteksi *YOLO*. Kotak ini dicrop lalu di *resize* ke 224x224 dan diberi salah satu dari lima label (*signature*, *W*, *G*, *kosong*, *ttd\_dtt*). Pada percobaan ini tidak digunakan *augmentasi* (*rotasi*, *flip*, atau *color jitter*) karena pola goresan pada kelas *W* dan *G* sensitive terhadap perubahan bentuk dan dikhawatirkan mengaburkan ciri yang ingin dipelajari model. Karena itu, penanganan ketidakseimbangan kelas dilakukan di tahap pelatihan (*loss* berbobot / *sampler*), bukan di tahap pembentukan dataset. Adapun hasil anotasi ditampilkan pada Gambar 6.



**Gambar 6.** Hasil Anotasi *ResNet-18*

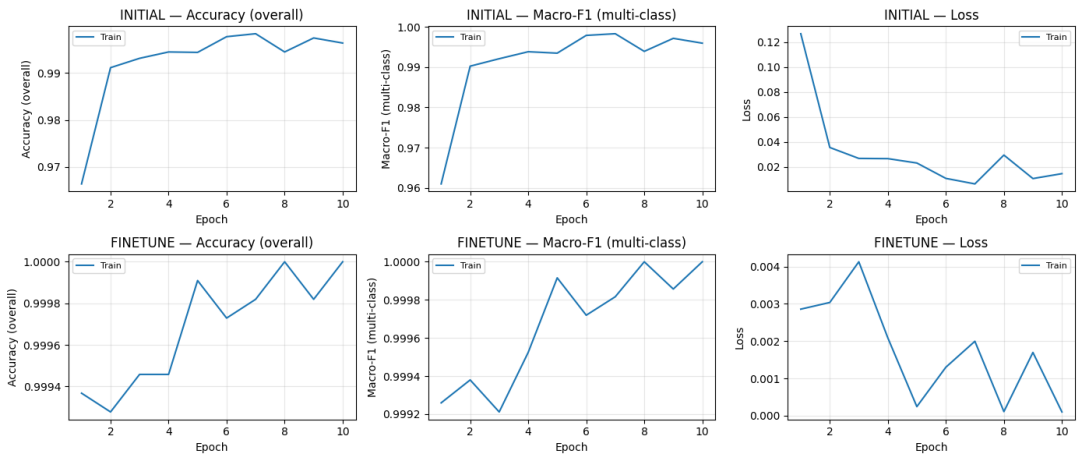
Pada *LayoutLMv3* data dibagi 80% pelatihan dan 20% pengujian, dari porsi latih disisihkan ~10% sebagai validasi. Pelatihan dilakukan dua fase, fase pertama *head-only* selama 5 *epoch* (*backbone* dibekukan) dengan *AdamW*,  $LR = 1e-5$ ,  $batch = 2$ . Fase kedua *fine-tuning end-to-end* selama 10 *epoch* dengan  $LR = 5e-5$ ,  $weight-decay = 0,01$ , serta *linear warmup* 10% + *decay*, dimana model belajar pada satu halaman *full*, agar informasi global tidak hilang. Ketidakseimbangan label ditangani dengan *Weighted Cross-Entropy* (berbasis distribusi label latih) dengan  $ignore\_index = -100$  untuk token yang tidak dievaluasi. *Checkpoint* dipilih berdasarkan *validation loss minimum*. Kinerja dilaporkan pada token valid menggunakan *Accuracy (token-level)* dan *Macro-F1 (token-level)*, kurva *train/val loss* dan *val Accuracy/Macro-F1* ditampilkan pada Gambar 7.





Gambar 7. Laporan Kinerja LayoutLMv3

Sementara *ResNet-18* setiap kotak TTD per baris dicrop dan diresize 224x224, dinormalisasi mengikuti overall *ImageNet*, lalu dilabeli salah satu dari 5 kelas (*signature*, *W*, *G*, kosong, *ttd\_dtt*). Pembagian data 80% pelatihan dan 20% pengujian (dengan validasi dari porsi latih). Tidak dilakukan augmentasi pada pelatihan ini, ketidakseimbangan kelas ditangani melalui *class weight / sampler*. Optimizer menggunakan *AdamW* ( $\approx 3 \times 10^{-4}$ , *weight\_decay* = 0,01 dengan *CosineAnnealingLR*, *checkpoint* dipilih berdasarkan *validation loss minimum*. Metrik yang dilaporkan adalah *Accuracy (overall)* dan *Macro-F1* (multi-kelas). Perkembangan *train/ val loss* serta *val Accuracy/Macro-F1* diringkas pada Gambar 8.



Gambar 8. Laporan Kinerja ResNet-18

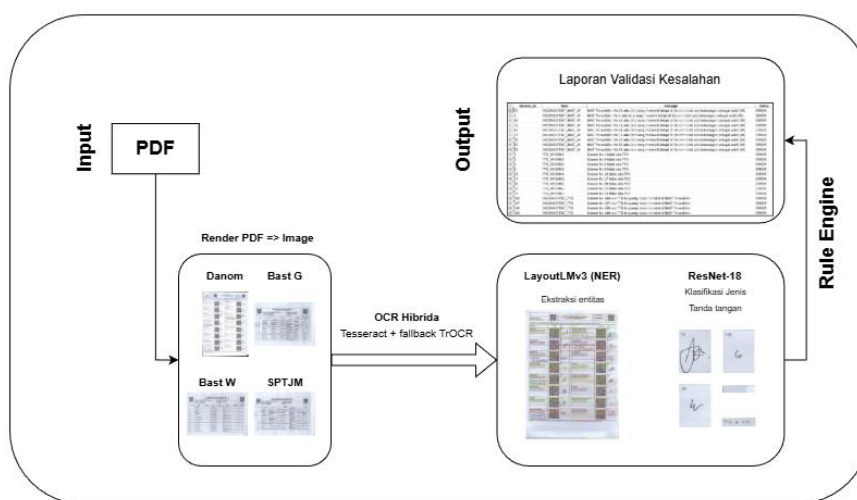
*LayoutLMv3* (NER) dievaluasi pada token valid dengan *Accuracy token* dan *Macro-F1*, *checkpoint* terbaik dipilih berdasarkan *validation loss minimum*. Skema dua fase pada *Weighted Cross-Entropy* dimana fase *freeze*: Optimizer *AdamW*, LR  $1 \times 10^{-5}$ , batch 2, dan fase *fine-tune*: Optimizer *AdamW*, LR  $5 \times 10^{-5}$ , *weight decay* 0,01, *linear warm-up*, memberikan konvergensi stabil dan kenaikan *Macro-F1* pada validasi. Sedangkan untuk *ResNet-18* (Klasifikasi ROI), ketidakseimbangan kelas ditangani dengan *class weight* pada *Cross-Entropy* berbobot. Pelatihan dibagi dua fase, pada *initial*: Optimizer *AdamW*, LR  $1 \times 10^{-3}$ , batch 32, dan fase *fine-tune*: Optimizer *AdamW*, LR  $3 \times 10^{-4}$ , *CosineAnnealingLR*. Model terbaik dipilih via *validation loss minimum* dan dilaporkan dengan *Accuracy* multi-kelas serta *Macro-F1* pada data uji yang dapat dilihat pada Tabel 2.

Tabel 2. Evaluasi Performa Best Model

Metrics	LayoutLMv3	ResNet-18
Val_f1_score (Macro-F1)	0,9455	0,9753
Val_accuracy	0,9678	0,9904
Val_loss	0,353	0,0890
Train_f1_score	0,9918	0,9992
Train_accuracy	0,993	0,9994
Learning_rate	2.22e-05	0,00023

Model terbaik pada masing-masing komponen dijalankan pada berkas yang sama dan bekerja secara paralel. Adapun visualisasi untuk proses yang dapat dilihat pada Gambar 9. Hasil

OCR pada sistem ini diperoleh lewat pendekatan hibrida. *Tesseract* dipakai secara default untuk teks tercetak pada dokumen, sedangkan *TrOCR* diaktifkan sebagai *fallback* pada potongan yang kualitas bacanya rendah atau mengandung unsur tulisan tangan. Dari keluaran OCR ini, *LayoutLMv3* mengekstrak entitas per baris, No Urut, Nama, NIK, Alamat, dll beserta koordinat token. Pada saat yang sama *ResNet-18* mengklasifikasi ROI kotak tanda tangan tiap baris ke kelas (*signature*, *W*, *G*, *kosong*, *ttt\_dtt*). Kedua keluaran kemudian dipadukan pada kunci (dokumen, halaman, baris) sehingga setiap baris memiliki pasangan (entitas dengan kelas TTD) yang konsisten. Di atas pasangan ini, *rule engine* mengeksekusi verifikasi lintas dokumen, seperti pada Danom ada keterangan W, maka wajib memiliki padanan di BAST Perwakilan dengan identitas cocok dan data perwakilan + TTD terisi). Jika Danom ada keterangan G harus konsisten dengan BAST Pengganti dan muncul di SPTJM (NIK, Alamat, Pekerjaan pengganti lengkap), sistem menolak duplikasi (muncul di BAST W dan G sekaligus, atau BAST tanpa penanda W/G di Danom), serta memeriksa kelengkapan/kesesuaian *field*.



Gambar 9. Proses Interseksi

Keluaran integrasi disajikan sebagai lembar laporan validasi bentuk CSV dan JSON, sesuai struktur kolom dapat dilihat pada Gambar 10. Tabel merangkum temuan per baris identitas dokumen dan lokasinya, aturan yang dicek/terlanggar, serta ringkasan hasil pemeriksaan tanpa penambahan kolom di luar yang tercantum pada laporan tersebut. Dengan mekanisme ini, kasus seperti W/G tanpa dokumen pendamping, BAST tanpa penanda di Danom, TTD kosong, ketidakcocokan NIK>Nama/Alamat, atau baris ganda terdeteksi dini sehingga operator menerima laporan siap tindak, bukan sekedar hasil ekstraksi.

	danom_no	type	message	status
2	67	INCONSISTENT_BAST_W	BAST Perwakilan No 67 ada data yang mewakili tetapi di Danom tidak ada keterangan sebagai wakil (W)	ERROR
3	5	INCONSISTENT_BAST_W	BAST Perwakilan No 5 ada data yang mewakili tetapi di Danom tidak ada keterangan sebagai wakil (W)	ERROR
4	23	INCONSISTENT_BAST_W	BAST Perwakilan No 23 ada data yang mewakili tetapi di Danom tidak ada keterangan sebagai wakil (W)	ERROR
5	52	INCONSISTENT_BAST_W	BAST Perwakilan No 52 ada data yang mewakili tetapi di Danom tidak ada keterangan sebagai wakil (W)	ERROR
6	41	INCONSISTENT_BAST_W	BAST Perwakilan No 41 ada data yang mewakili tetapi di Danom tidak ada keterangan sebagai wakil (W)	ERROR
7	37	INCONSISTENT_BAST_W	BAST Perwakilan No 37 ada data yang mewakili tetapi di Danom tidak ada keterangan sebagai wakil (W)	ERROR
8	78	INCONSISTENT_BAST_W	BAST Perwakilan No 78 ada data yang mewakili tetapi di Danom tidak ada keterangan sebagai wakil (W)	ERROR
9	82	INCONSISTENT_BAST_W	BAST Perwakilan No 82 ada data yang mewakili tetapi di Danom tidak ada keterangan sebagai wakil (W)	ERROR
10	83	INCONSISTENT_BAST_W	BAST Perwakilan No 83 ada data yang mewakili tetapi di Danom tidak ada keterangan sebagai wakil (W)	ERROR
11	1	TTD_MISSING	Danom No 1 tidak ada TTD	ERROR
12	4	TTD_MISSING	Danom No 4 tidak ada TTD	ERROR
13	6	TTD_MISSING	Danom No 6 tidak ada TTD	ERROR
14	6	TTD_MISSING	Danom No 6 tidak ada TTD	ERROR
15	13	TTD_MISSING	Danom No 13 tidak ada TTD	ERROR
16	17	TTD_MISSING	Danom No 17 tidak ada TTD	ERROR
17	20	TTD_MISSING	Danom No 20 tidak ada TTD	ERROR
18	22	TTD_MISSING	Danom No 22 tidak ada TTD	ERROR
19	33	TTD_MISSING	Danom No 33 tidak ada TTD	ERROR
20	100	INCONSISTENT_TTD	Danom No 100 ada TTD langsung tetapi tercatat di BAST Perwakilan	ERROR
21	137	INCONSISTENT_TTD	Danom No 137 ada TTD langsung tetapi tercatat di BAST Perwakilan	ERROR
22	139	INCONSISTENT_TTD	Danom No 139 ada TTD langsung tetapi tercatat di BAST Perwakilan	ERROR
23	140	INCONSISTENT_TTD	Danom No 140 ada TTD langsung tetapi tercatat di BAST Perwakilan	ERROR

(a) Laporan Validasi Excel

```

{
  "timestamp": "2023-08-08 10:52:40",
  "version": "1",
  "total_message": 242,
  "total_error": 95,
  "total_warning": 45,
  "total_info": 102,
  "total_success": 0,
  "total_failure": 0,
  "total_cancel": 0,
  "total_retry": 0,
  "total_timeout": 0,
  "total_error_detail": [
    {
      "error_detail": [
        {
          "danom_no": "67",
          "type": "INCONSISTENT_BAST_W",
          "message": "BAST Perwakilan No 67 ada data yang mewakili tetapi di Danom tidak ada keterangan sebagai wakil (W)",
          "status": "ERROR"
        },
        {
          "danom_no": "5",
          "type": "INCONSISTENT_BAST_W",
          "message": "BAST Perwakilan No 5 ada data yang mewakili tetapi di Danom tidak ada keterangan sebagai wakil (W)",
          "status": "ERROR"
        },
        {
          "danom_no": "23",
          "type": "INCONSISTENT_BAST_W",
          "message": "BAST Perwakilan No 23 ada data yang mewakili tetapi di Danom tidak ada keterangan sebagai wakil (W)",
          "status": "ERROR"
        },
        {
          "danom_no": "52",
          "type": "INCONSISTENT_BAST_W",
          "message": "BAST Perwakilan No 52 ada data yang mewakili tetapi di Danom tidak ada keterangan sebagai wakil (W)",
          "status": "ERROR"
        },
        {
          "danom_no": "41",
          "type": "INCONSISTENT_BAST_W",
          "message": "BAST Perwakilan No 41 ada data yang mewakili tetapi di Danom tidak ada keterangan sebagai wakil (W)",
          "status": "ERROR"
        },
        {
          "danom_no": "37",
          "type": "INCONSISTENT_BAST_W",
          "message": "BAST Perwakilan No 37 ada data yang mewakili tetapi di Danom tidak ada keterangan sebagai wakil (W)",
          "status": "ERROR"
        },
        {
          "danom_no": "78",
          "type": "INCONSISTENT_BAST_W",
          "message": "BAST Perwakilan No 78 ada data yang mewakili tetapi di Danom tidak ada keterangan sebagai wakil (W)",
          "status": "ERROR"
        },
        {
          "danom_no": "82",
          "type": "INCONSISTENT_BAST_W",
          "message": "BAST Perwakilan No 82 ada data yang mewakili tetapi di Danom tidak ada keterangan sebagai wakil (W)",
          "status": "ERROR"
        },
        {
          "danom_no": "83",
          "type": "INCONSISTENT_BAST_W",
          "message": "BAST Perwakilan No 83 ada data yang mewakili tetapi di Danom tidak ada keterangan sebagai wakil (W)",
          "status": "ERROR"
        },
        {
          "danom_no": "1",
          "type": "TTD_MISSING",
          "message": "Danom No 1 tidak ada TTD",
          "status": "ERROR"
        },
        {
          "danom_no": "4",
          "type": "TTD_MISSING",
          "message": "Danom No 4 tidak ada TTD",
          "status": "ERROR"
        },
        {
          "danom_no": "6",
          "type": "TTD_MISSING",
          "message": "Danom No 6 tidak ada TTD",
          "status": "ERROR"
        },
        {
          "danom_no": "6",
          "type": "TTD_MISSING",
          "message": "Danom No 6 tidak ada TTD",
          "status": "ERROR"
        },
        {
          "danom_no": "13",
          "type": "TTD_MISSING",
          "message": "Danom No 13 tidak ada TTD",
          "status": "ERROR"
        },
        {
          "danom_no": "17",
          "type": "TTD_MISSING",
          "message": "Danom No 17 tidak ada TTD",
          "status": "ERROR"
        },
        {
          "danom_no": "20",
          "type": "TTD_MISSING",
          "message": "Danom No 20 tidak ada TTD",
          "status": "ERROR"
        },
        {
          "danom_no": "22",
          "type": "TTD_MISSING",
          "message": "Danom No 22 tidak ada TTD",
          "status": "ERROR"
        },
        {
          "danom_no": "33",
          "type": "TTD_MISSING",
          "message": "Danom No 33 tidak ada TTD",
          "status": "ERROR"
        },
        {
          "danom_no": "100",
          "type": "INCONSISTENT_TTD",
          "message": "Danom No 100 ada TTD langsung tetapi tercatat di BAST Perwakilan",
          "status": "ERROR"
        },
        {
          "danom_no": "137",
          "type": "INCONSISTENT_TTD",
          "message": "Danom No 137 ada TTD langsung tetapi tercatat di BAST Perwakilan",
          "status": "ERROR"
        },
        {
          "danom_no": "139",
          "type": "INCONSISTENT_TTD",
          "message": "Danom No 139 ada TTD langsung tetapi tercatat di BAST Perwakilan",
          "status": "ERROR"
        },
        {
          "danom_no": "140",
          "type": "INCONSISTENT_TTD",
          "message": "Danom No 140 ada TTD langsung tetapi tercatat di BAST Perwakilan",
          "status": "ERROR"
        }
      ]
    }
  ]
}

```

(b) Laporan Validasi JSON

Gambar 10. Hasil Laporan Validasi Kesalahan Dokumen

#### 4.2. Pembahasan

Berdasarkan Tabel evaluasi performa, *LayoutLMv3* menunjukkan performa yang stabil pada tugas *labeling* entitas. Model ini mencapai *val\_f1\_score* (*macro-f1*) sebesar 0.9455 dengan *val\_accuracy* 0,9678 dan *val\_loss* sekitar 0,353. Nilai *train\_f1\_score* (0,9918) dan *train\_accuracy*

(0,993) yang sedikit lebih tinggi tetapi masih sejalan dengan metrik validasi mengindikasikan bahwa *fine-tuning* berjalan cukup baik. Dengan *learning rate* akhir sekitar  $2,22 \times 10^{-5}$ , *LayoutLMv3* dapat mengenali entitas Nama, NIK, Alamat, No Urut dll secara cukup andal sebagai dasar *labeling* entitas [3]. Sedangkan untuk klasifikasi kotak tanda tangan, *ResNet-18* secara angka tampak sangat baik, dengan *val\_f1\_score* 0,9753, *val\_accuracy* 0,9904, dan *val\_loss* hanya 0,0890. Metrik pelatihan bahkan mendekati sempurna (*train\_f1\_score* 0,9992, *train\_accuracy* 0,9994) [6]. Namun, hasil ini perlu dibaca hati-hati karena distribusi kelas pada dataset TTD sangat tidak seimbang, kelas *signature* dan kosong jauh lebih banyak dibanding *W*, *G* dan *ttd\_dtt*. Akurasi dan *macro-F1* yang tinggi belum tentu mencerminkan kemampuan model membedakan kelas minoritas secara konsisten, sehingga masih menunjukkan bahwa prediksi pada kelas *W*, *G* dan *ttd\_dtt* lebih rentan salah, terutama ketika contoh latihnya sedikit atau kualitas citranya rendah.

Secara keseluruhan, pendekatan dua model pada penelitian ini menunjukkan hasil yang berbeda, *LayoutLMv3* sudah cukup andal untuk tugas *labeling* entitas dokumen, sedangkan *ResNet-18* untuk klasifikasi kotak TTD masih belum dapat dianggap berhasil. Nilai akurasi dan *F1* memang tampak tinggi pada tabel evaluasi, tetapi analisis lebih lanjut menunjukkan model cenderung bias terhadap kelas mayoritas (*signature* dan kosong) dan sering salah memprediksi kelas minoritas, terutama *W*, *G* dan *ttd\_dtt*. Kondisi ini mengindikasikan adanya *overfitting* akibat distribusi dataset yang sangat tidak seimbang, sehingga kinerja di dunia nyata tidak sebaik yang tercermin pada metrik validasi. Oleh karena itu, pengembangan berikutnya perlu difokuskan pada penambahan contoh untuk kelas *W*, *G* dan *ttd\_dtt*, penerapan teknik penyeimbangan kelas (misalnya *oversampling* atau *loss* berbobot), serta augmentasi yang lebih terarah sebelum *ResNet-18* dapat digunakan sebagai komponen verifikasi yang benar-benar reliabel.

Penelitian ini memperkuat temuan L. Wang et al. [8], S. Huang et al. [7], Z. Yan et al. [3], yang menunjukkan sistem mampu memahami struktur dasar dari dokumen dan dapat meningkatkan akurasi dan efisiensi verifikasi data penerima bantuan pangan. *LayoutLMv3* cukup baik dalam melabeli setiap data, apakah termasuk kelas Nama, Alamat, NIK, dll. *LayoutLMv3* yang dibatasi 512 token, dapat diatasi dengan membagi dokumen menjadi 4 – 6 bagian, sehingga tidak kehilangan informasi global. *Hybird OCR* yang dijalankan juga memiliki hasil yang baik, dengan *pytesseract* yang mampu mengekstrak teks *printed* dan *TrOCR* yang dapat mengekstrak teks *handwriting* mampu mengoptimalkan proses ekstraksi teks. Selain itu, penelitian ini juga sejalan dengan K. Wijaya dan E.P. Widiyanto [11], N. Khasanah [6] yang mencoba melakukan klasifikasi pada sebuah citra tanda tangan. Dengan jumlah dataset yang terbatas, *ResNet-18* mampu melakukan klasifikasi jenis tanda tangan. Untuk mengatasi *inbalance* dataset pada penelitian ini, diatasi dengan *WeightedRandomSampler*, sehingga kelas yang sedikit akan lebih sering dilihat. Pemisahan kedua model untuk ekstraksi teks dan klasifikasi jenis tanda tangan, mampu meningkatkan fokus model dalam satu tugas, sehingga dapat dimaksimalkan dengan baik. Dengan pendekatan ini, kontribusi utama penelitian dalam memperkaya pada ranah *intelligent document processing* untuk layanan pemerintah, dengan menghadirkan rancangan sistem verifikasi dokumen bantuan pangan yang lebih cepat, konsisten, dan dapat diaudit, efektif dalam mendukung manajemen operasional verifikasi data. Sistem ini menjadi dasar bagi pengembangan lanjutan, dan perlu adanya perluasan struktur dokumen, perbaikan dan penyeimbangan dataset tanda tangan.

## 5. Simpulan

Penelitian ini menghasilkan prototipe sistem verifikasi dokumen Bantuan Pangan yang menggabungkan *OCR hibrida*, *LayoutLMv3*, dan *ResNet-18* dalam satu alur *end-to-end*. Model *LayoutLMv3* untuk *NER* mencapai *val\_accuracy* sekitar 96,78% dan *macro-F1* 0,9455, sehingga cukup andal untuk mengekstrak dan *labeling* entitas kunci (Nama, NIK, Alamat, No Urut, Ket\_TTD) dari PDF hasil scan dan mengubahnya menjadi data terstruktur. *ResNet-18* untuk klasifikasi kotak tanda tangan secara angka juga tampak tinggi (*val\_accuracy* 99,04%, *macro-F1* 0,9753), namun analisis lebih lanjut menunjukkan model masih bias ke kelas mayoritas (*signature* dan kosong) dan sering keliru pada kelas minoritas (*W*, *G*, *ttd\_dtt*), sehingga belum dapat dianggap reliabel di kondisi nyata. Hal ini membuktikan tujuan utama penelitian merancang alur verifikasi dokumen otomatis yang lebih cepat dan konsisten telah tercapai di sisi ekstraksi teks dan *rule-based checking*, sementara sisi klasifikasi TTD masih memerlukan pengayaan data, penyeimbangan kelas, dan eksperimen lanjutan sebelum siap dioperasionalkan.

**Referensi**

- [1] B. P. Nasional, "Jangan Keliru, Ini Perbedaan Bantuan Pangan Beras dengan Bantuan Sosial Lainnya," *Badan Pangan Nasional*, 2024. <https://badanpangan.go.id/blog/post/jangan-keliru-ini-perbedaan-bantuan-pangan-beras-dengan-bantuan-sosial-lainnya> (accessed Jun. 13, 2024).
- [2] B. P. Statistik, "Persentase Penduduk Miskin Maret 2024 turun menjadi 9,03 persen.," *Badan Pusat Statistik*, 2024. <https://www.bps.go.id/id/pressrelease/2024/07/01/2370/persentase-penduduk-miskin-maret-2024-turun-menjadi-9-03-persen-.html> (accessed Jun. 13, 2024).
- [3] Z. Yan *et al.*, "DocExtractNet: A novel framework for enhanced information extraction from business documents," in *Information Processing and Management*, Elsevier Ltd, 2025, p. 104046. doi: 10.1016/j.ipm.2024.104046.
- [4] M. R. Wardani, S. Sudin, and G. Mandar, "Implementasi Teknologi OCR Berbasis Artificial Intelligence Dalam Perancangan Aplikasi Bansos di Desa Daruba Pantai," *J. Ilm. Multidisiplin*, vol. 2, no. 2, pp. 19–29, 2024, doi: <https://doi.org/10.62017/merdeka>.
- [5] M. Li *et al.*, "TrOCR: Transformer-Based Optical Character Recognition with Pre-trained Models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Washington DC, 2023, pp. 13094–13101.
- [6] N. Khasanah, "Komparasi Arsitektur Resnet50 Dan Vgg16 Untuk Untuk Klasifikasi Citra Tanda Tangan," *JSI J. Sist. Inf.*, vol. 14, no. 1, pp. 2611–2621, 2022.
- [7] S. Huang, Y. Xiong, and G. Wu, "LayoutPointer: A Spatial-Context Adaptive Pointer Network for Visual Information Extraction," in *Association for Computational Linguistics*, 2024, pp. 3737–3748. doi: 10.18653/v1/2024.naacl-long.207.
- [8] L. Wang, J. He, X. Xu, N. Liu, and H. Liu, "Alignment-Enriched Tuning for Patch-Level Pre-trained Document Image Models," in *AAAI Conference on Artificial Intelligence*, 2023, pp. 2590–2598. doi: 10.1609/aaai.v37i2.25357.
- [9] M. T. Toha and A. Triayudi, "Penerapan Membaca Tulisan Di Dalam Gambar Menggunakan Metode OCR Berbasis Website (Studi Kasus: e-KTP) Penerapan Membaca Tulisan di Dalam Gambar Menggunakan Metode OCR Berbasis Website (Studi Kasus: e-KTP)," *JST (Jurnal Sains dan Teknol.)*, vol. 11, no. 1, pp. 175–183, 2022, doi: 10.23887/jstundiksha.v11i1.42279.
- [10] B. M. Sujatmiko, E. Yudaningtyas, and P. Mudji Raharjo, "Convolution Neural Network Dengan Desain Jaringan Resnet Sebagai Metode Klasifikasi Tumor Kulit," *J. Simantec*, vol. 11, no. 1, pp. 53–64, 2022, doi: 10.21107/simantec.v11i1.14083.
- [11] K. Wijaya and E. P. Widiyanto, "Klasifikasi Kepemilikan Tanda Tangan Menggunakan Convolutional Neural Network Dengan Arsitektur Alexnet Khrisnaldi," In *Mdp Student Conference (MSC) 2023*, Palembang, 2023, pp. 133–143.
- [12] T. Heriyanto, Y. Sholva, and R. D. Nyoto, "Implementasi Optical Character Recognition (OCR) untuk Verifikasi Berkas pada Digital Library Program Studi Informatika Universitas Tanjungpura," *J. Ris. Sains dan Teknol. Inform.*, vol. 1, no. 1, pp. 26–29, 2023, doi: 10.26418/juristi.v1i1.60916.
- [13] P. A. Septio and S. Y. J. Prasetyo, "Pembuatan Aplikasi Validasi Document Tagihan Pembelian Barang Secara Digital Menggunakan OCR dengan tool tesseract pada System Portal Perusahaan," *J. Sains Komput. Inform.*, vol. 7, no. September, pp. 650–662, 2023.
- [14] S. S. Nurhaliza, M. Subali, L. Etp, and Rozi, "Analisis Kinerja Optical Character Recognition Untuk Membaca Dokumen Secara Otomatis," in *Seminar Nasional Teknologi Informasi dan Komunikasi STI&K (SeNTIK)*, SeNTIK, Ed., Jakarta Selatan: STMIK Jakarta STI&K, 2022, pp. 2581–2327.
- [15] L. Huttner *et al.*, "Low-Rank Adaptation vs. Fine-Tuning for Handwritten Text Recognition," 2025. doi: 10.1109/WACVW65960.2025.00146.
- [16] M. Jungo *et al.*, "Impact of the ground truth quality for handwriting recognition," in *ACM International Conference Proceeding Series*, 2023, pp. 135–140. doi: 10.1145/3628797.3628976.
- [17] H. Zhang, E. Whittaker, and I. Kitagishi, "Extending TrOCR for Text Localization-Free OCR of Full-Page Scanned Receipt Images," 2023. doi: 10.1109/ICCVW60793.2023.00160.
- [18] P. Hu, Z. Zhang, J. Ma, S. Liu, J. Du, and J. Zhang, "DocMamba: Efficient Document Pre-training with State Space Model," in *AAAI Conference on Artificial Intelligence*, 2025, pp. 24095–24103. doi: 10.1609/aaai.v39i22.34584.