

TESIS
EKSTRAKSI NIK CITRA E-KTP MENGGUNAKAN
METODE CONVOLUTION NEURAL NETWORK DENGAN
DATASET MNIST DIGIT



Disusun oleh:

Nama : Yusda Helmani
NIM : 21.55.1061
Konsentrasi : Digital Transformation Intelligence

PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA
2023

TESIS
EKSTRAKSI NIK CITRA E-KTP MENGGUNAKAN
METODE CONVOLUTION NEURAL NETWORK DENGAN
DATASET MNIST DIGIT

EXTRACTION NIK OF E-KTP IMAGE USING THE CONVOLUTION
NEURAL NETWORK METHOD WITH DATASET MNIST DIGITS

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister



Disusun oleh:

Nama : Yusda Helmani
NIM : 21.55.1061
Konsentrasi : Digital Transformastion Intelligence

PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA
2023

HALAMAN PENGESAHAN

**EKSTRAKSI NIK CITRA E-KTP MENGGUNAKAN
METODE CONVOLUTION NEURAL NETWORK DENGAN
DATASET MNIST DIGIT**

**EXTRACTION NIK OF E-KTP IMAGE USING THE CONVOLUTION
NEURAL NETWORK METHOD WITH DATASET MNIST DIGITS**

Dipersiapkan dan Disusun oleh

Yusda Helmani

21.55.1061

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Rabu, 01 Februari 2023

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 01 Februari 2023
Rektor

Prof. Dr. M. Suyanto, M.M.
NIK. 190302001

HALAMAN PERSETUJUAN

**EKSTRAKSI NIK CITRA E-KTP MENGGUNAKAN
METODE CONVOLUTION NEURAL NETWORK DENGAN
DATASET MNIST DIGIT**

**EXTRACTION NIK OF E-KTP IMAGE USING THE CONVOLUTION
NEURAL NETWORK METHOD WITH DATASET MNIST DIGITS**

Dipersiapkan dan Disusun oleh

Yusda Helmani

21.55.1061

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Rabu, 01 Februari 2023

Pembimbing Utama

Anggota Tim Penguji

Prof. Dr. Ema Utami, S.Si, M.Kom
NIK. 190302037

Dr. Andi Sunvoto, M.Kom,
NIK. 190302052

Pembimbing Pendamping

Alva Hendi Muhammad, S.T., M.Eng., Ph.D.
NIK. 190302493

Dr. Hanafi, S.Kom, M.Eng
NIK. 190302024

Prof. Dr. Ema Utami, S.Si, M.Kom
NIK. 190302037

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 01-Februari 2023

Direktur Program Pascasarjana

Dr. Kusrini, M.Kom.
NIK. 190302106

HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : Yusda Helmani

NIM : 21.55.1061

Konsentrasi : Digital Transformasion Intelligence

Menyatakan bahwa Tesis dengan judul berikut:

Ekstraksi NIK citra e-KTP menggunakan metode Convolution Neural Network dengan dataset MNIST DIGIT

Dosen Pembimbing Utama : Prof. Dr. Ema Utami, S.Si, M.Kom

Dosen Pembimbing Pendamping : Dr. Hanafi, S.Kom, M.Eng

1. Karya tulis ini adalah benar-benar **ASLI** dan **BELUM PERNAH** diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima **SANKSI AKADEMIK** dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi

Yogyakarta, 01 Februari 2023

Yang Menyatakan,



Yusda Helmani

HALAMAN PERSEMBAHAN

Puji syukur kehadiran Allah SWT dan berkat rahmat Nabi Besar Muhammad SAW, penulis dapat menyelesaikan karya tulis laporan akhir berupa Tesis. Pada kesempatan ini penulis ingin menyampaikan ucapan terimakasih yang sebesar-besarnya kepada :

1. Kepada kedua orang tua (Bapak M. Yusuf, (Alm) dan Ibu Hamdanah) yang tiada hentinya memberikan nasehat dan menjaga serta membesarkan saya hingga saat ini, dan kepada istri dan anak tercinta (Fetty Haryati Asriani Nasution, S.A. dan Yusrizal Falah Alhanan) yang telah memberikan semangat yang lebih untuk menempuh pendidikan ini.
2. Bapak Prof. Dr. M. Suyanto, MM. Selaku Rektor Universitas Amikom Yogyakarta.
3. Ibu Prof. Dr. Kusriani, M.Kom, selaku Direktur Program Pascasarjana Universitas Amikom Yogyakarta.
4. Ibu Prof. Dr. Ema Utami, S.Si, M.Kom, selaku dosen pembimbing 1 dan Bapak Dr. Hanafi, S.Kom, M.Eng, selaku dosen pembimbing 2.
5. Kepada Bapak Farezi Indra Kasugi, M.Kom, selaku pimpinan kerja, teman kuliah dan teman bertukar pendapat.
6. Seluruh Keluarga, kerabat dekat dan teman-teman yang telah membantu dalam pengumpulan data e-KTP.

HALAMAN MOTTO

“Raihlah ilmu dan untuk meraih ilmu belajarlah tenang dan sabar”

(Umar Bin Khattab)

“Barang siapa keluar untuk mencari sebuah ilmu, maka ia akan berada di jalan

Allah hingga ia kembali ”

(HR. Tirmidzi)

“Hanya pendidikan yang bisa menyelamatkan masa depan, tanpa pendidikan

indonesia tak mungkin bertahan”

(Najwa Shihab)

“The whole purpose of education is to turn mirrors into windows”

(Sydney J. Harris)

KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadiran Allah SWT karena atas rahmat dan hidayah yang telah diberikan, penulis masih diberikan kesempatan dan kemudahan untuk menyelesaikan Tesis ini.

Dalam penulisan Tesis dalam rangka memenuhi salah satu syarat kelulusan perguruan tinggi Program Studi Strata-2 Magister Teknik Informatika di Universitas AMIKOM Yogyakarta dan meraih gelar Magister Komputer (M.Kom).

Dalam penyusunan laporan ini, saya banyak berterima kasih atas bantuan dan masukan dari berbagai pihak, terutama dari Prof. Dr. Ema Utami, S.Si, M.Kom selaku pembimbing utama dan Dr. Hanafi, S.Kom, M.Eng selaku pembimbing pendamping yang bersedia meluangkan waktu dalam proses menyelesaikan penulisan ini.

Akhir kata, disadari bahwa tesis ini belum dapat dikatakan sempurna seperti yang diharapkan, namun semoga dapat bermanfaat yang optimal bagi pembaca maupun dari penulis sendiri serta dapat digunakan sebagai referensi untuk penelitian lain.

Yogyakarta, 01 Februari 2023

Penulis

DAFTAR ISI

JUDUL	ii
HALAMAN PENGESAHAN	iii
HALAMAN PERSETUJUAN	iv
HALAMAN PERNYATAAN KEASLIAN TESIS	v
HALAMAN PERSEMBAHAN	vi
HALAMAN MOTTO	vii
KATA PENGANTAR	viii
DAFTAR ISI	ix
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiii
INTISARI	xv
<i>ABSTRACT</i>	xvi
BAB I PENDAHULUAN	1
1.1. Latar Belakang Masalah	1
1.2. Rumusan Masalah	8
1.3. Batasan Masalah	8
1.4. Tujuan Penelitian	9
1.5. Manfaat Penelitian	9
BAB II TINJAUAN PUSTAKA	11
2.1. Tinjauan Pustaka	11
2.2. Keaslian Penelitian	16
2.3. Landasan Teori	21

2.3.1. Kartu Tanda Penduduk Elektronik	21
2.3.2. Machine Learning.....	21
2.3.3. Deep Learning	23
2.3.4. Optical Character Recognition	24
2.3.5. Convolution Neural Network	26
2.3.6. Convolution Layer.....	28
2.3.7. Pooling Layer	31
2.3.8. Library TensorFlow	32
2.3.9. Dataset MNIST DIGITS.....	33
BAB III METODE PENELITIAN.....	35
3.1. Jenis, Sifat, dan Pendekatan Penelitian.....	35
3.1.1. Jenis Penelitian	35
3.1.2. Sifat Penelitian.....	35
3.1.3. Pendekatan Penelitian.....	36
3.2. Metode Pengumpulan Data.....	36
3.2.1. Wawancara	36
3.2.2. Observasi dan Pengamatan.....	36
3.2.3. Studi Pustaka	37
3.2.4. Dokumentasi.....	37
3.3. Metode Analisis Data.....	37

3.4. Alur Penelitian	38
3.5. Metode Evaluasi.....	40
BAB IV HASIL PENELITIAN DAN PEMBAHASAN	43
4.1. Membangun Model CNN dataset EMNIST	43
4.1.1. Pengumpulan Data MNIST Digits	43
4.1.2. Pembuatan Model MLP dan CNN	46
4.1.3. Evaluasi Model MLP dan CNN	54
4.2. Implementasi Recognition citra E-KTP.....	60
4.2.1. Pengumpulan data E-KTP.....	60
4.2.2. Image Preprocessing dan Segmentation.....	60
4.2.3. Ekstraksi Citra menggunakan CNN	66
4.2.4. Evaluasi Hasil Ekstraksi.....	69
BAB V PENUTUP	73
5.1. Kesimpulan	73
5.2. Saran	75
Daftar Pustaka.....	76
LAMPIRAN.....	83

DAFTAR TABEL

Tabel 3. 1 Confusion Matrix	41
Tabel 4. 1 Jumlah dataset training Emnist Digit	45
Tabel 4. 2 Ringkasan arsitektur layer model MLP	48
Tabel 4. 3 Contoh hasil pelatihan dataset Model MLP	49
Tabel 4. 4 Ringkasan arsitektur layer model CNN	53
Tabel 4. 5 Confusion Matrix dataset	55
Tabel 4. 6 Confusion Matrix dataset	58
Tabel 4. 7 Hasil Predict dan Confidence Model CNN	68
Tabel 4. 8 Hasil Ekstraksi dengan E-KTP berwarna	69
Tabel 4. 9 Hasil Ekstraksi dengan E-KTP Fotocopy	70
Tabel 5. 1 Evaluasi Confusion Matrix Model MLP	74
Tabel 5. 2 Evaluasi Confusion Matrix Model CNN	74

DAFTAR GAMBAR

Gambar 2. 1 Contoh citra e-KTP	21
Gambar 2. 2 Proses Machine Learning	22
Gambar 2. 3 Ruang Lingkup Deep Learning	23
Gambar 2. 4 Optical Character Recognition	25
Gambar 2. 5 Proses OCR secara umum.	25
Gambar 2. 6 Arsitektur MLP Sederhana	27
Gambar 2. 7 Proses Konvolusi pada CNN	28
Gambar 2. 8 Convolutin layer dengan satu filter berukuran 5x5	29
Gambar 2. 9 Teknik Zero padding dengan stride = 1	29
Gambar 2. 10 Contoh konvolusi 1x1 dengan 4 filter	31
Gambar 2. 11 Proses pooling untuk mereduksi dimensi data	31
Gambar 2. 12 Teknik max pool dan average pool	32
Gambar 2. 13 Sampel Citra pada MNIST Digits	34
Gambar 3. 1 Diagram Alur Penelitian.....	38
Gambar 4. 1 Contoh angka dan hurupf data EMNIST.....	44
Gambar 4. 2 Grafik dataset MNIST Digits	44
Gambar 4. 3 Contoh ilustrasi dataset MNIST Digit.....	45
Gambar 4. 4 Contoh layer MLP.....	47
Gambar 4. 5 Alur Model MLP.....	48
Gambar 4. 6 Contoh Layer CNN	50
Gambar 4. 7 Alur Model CNN.....	51
Gambar 4. 8 Grafik Kurva Model MLP.....	55

Gambar 4. 9 Confusion Matrix dengan Dataset MNIST Digits.....	56
Gambar 4. 10 Hasil Kesalahan Prediksi Model MPL.....	57
Gambar 4. 11 Grafik Kurva Model CNN.....	58
Gambar 4. 12 Confusion Matrix dengan Dataset MNIST Digits.....	59
Gambar 4. 13 Hasil Kesalahan Prediksi Model CNN.....	60
Gambar 4. 14 Citra Grayscalee.....	62
Gambar 4. 15 Binerisasi Citra.....	62
Gambar 4. 16 Visuali Sgmentation Line Citra E-KTP.....	64
Gambar 4. 17 Nilai NIK pada Citra E-KTP.....	65
Gambar 4. 18 Proses Segmentation Angka NIK Citra E-KTP.....	65
Gambar 4. 19 Nilai Acak NIK hasil Proses Proyeksi Horizontal.....	66
Gambar 4. 20 Hasil Predict dan Confidence Model CNN.....	68

INTISARI

Ekstraksi Nomor Induk Kependudukan (NIK) dari Kartu Tanda Penduduk (KTP) merupakan tugas penting dalam berbagai tujuan identifikasi. Penggunaan Convolution Neural Network (CNN) dalam pengenalan citra telah menunjukkan hasil yang menjanjikan dalam beberapa tahun terakhir, termasuk dalam ekstraksi teks dari citra. Dalam penelitian ini bertujuan untuk mengevaluasi keefektifan metode CNN dalam melakukan ekstraksi NIK pada citra e-KTP.

Metode yang diusulkan melibatkan model CNN untuk mempelajari fitur-fitur Angka dari Dataset MNIST Digit yang terdiri dari gambar digit 0-9 yang diolah untuk merepresentasikan digit pada citra E-KTP. Hasil penelitian menunjukkan bahwa metode CNN efektif dalam melakukan ekstraksi NIK pada citra e-KTP dengan tingkat akurasi yang tinggi dengan nilai *test loss* sebesar 0.01%, nilai *test score* sebesar 99.59% dan *error* sebesar 0.41%. Berdasarkan hasil dari evaluasi metode CNN dapat diimplementasikan sebagai sarana untuk ekstraksi NIK pada citra e-KTP terlihat hasil pembacaan angka mendapatkan tingkat akurasi sebesar 96.91% dengan 10 sampel citra e-KTP berwarna sedangkan tingkat akurasi untuk e-KTP fotocopy mendapatkan nilai akurasi sebesar 93.25%. Implikasi dari penelitian ini adalah dapat digunakan sebagai solusi untuk mempermudah proses verifikasi identitas dan validasi data.

Secara keseluruhan, pendekatan berbasis CNN untuk ekstraksi NIK dari citra e-KTP dengan dataset MNIST Digit menunjukkan hasil yang menjanjikan dan dapat diterapkan pada berbagai tujuan identifikasi. Penelitian lebih lanjut dapat dilakukan untuk memperluas dataset seperti MNIST Alfabet untuk melakukan ekstraksi keseluruhan citra e-KTP dan mengoptimalkan parameter-parameter model CNN untuk kinerja yang lebih baik.

Kata kunci : CNN, NIK, E-KTP, MNIST Digit

ABSTRACT

Extracting the Population Identification Number (NIK) from the Identity Card (KTP) is an important task in various identification purposes. The use of Convolutional Neural Network (CNN) in image recognition has shown promising results in recent years, including in text extraction from images. This study aims to evaluate the effectiveness of the CNN method in extracting NIK on e-KTP images.

The proposed method involves training a CNN model to learn the numerical features from the MNIST Digit Dataset, which consists of images of digits 0-9 processed to represent the digits on the e-KTP image. The research results show that the CNN method is effective in extracting NIK on e-KTP images with a high level of accuracy, - with a test loss value of 0.01%, a test score value of 99.59%, and an error value of 0.41%. Based on the evaluation results of the CNN method, it can be implemented as a means of extracting NIK from e-KTP images, with a reading accuracy level of 96.91% for 10 colored e-KTP image samples, while the accuracy level for e-KTP photocopies obtained an accuracy value of 93.25%. The implication of this research is that it can be used as a solution to simplify the identity verification process and data validation.

Overall, the CNN-based approach for NIK extraction from e-KTP images with the MNIST Digit Dataset shows promising results and can be applied to various identification purposes. Further research can be conducted to expand the dataset, such as the MNIST Alphabet, to extract the entire e-KTP image and optimize the CNN model parameters for better performance.

Keyword: CNN, NIK, E-KTP, EMNIST Digit

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Kartu identitas tradisional akan digantikan dengan kartu tanda penduduk elektronek (E-KTP) untuk tahun 2009 pada wilayah Padang, Makassar, Yogyakarta dan Denpasar, sementara di daerah dan kota lainnya resmi diluncurkan pada tahun 2011. Hampir seluruh penduduk Indonesia telah memiliki E-KTP (UU RI No. 24 Tahun 2013 Tentang Perubahan Atas UU No. 23 Tahun 2006 Tentang Administrasi Kependudukan, 2013) pasal 27 menyatakan setiap penduduk harus memberitahukan kepada otoritas administrasi setempat dalam waktu sekurang-kurangnya 60 (enam puluh) hari sejak kelahiran, pendaftaran dan penerbitan akta kelahiran harus dilakukan setelah menerima keputusan. Pada pasal 63 UU tersebut menyatakan bahwa setiap warga Negara Indonesia dan warga negara asing pemegang izin tinggal tetap yang telah berusia 17 (tujuh belas) tahun atau pernah menikah wajib memiliki KTP Elektronik (E-KTP) dan Pasal 63 ayat 5 menyatakan bahwa setiap warga negara harus membawa E-KTP pada saat berpergian sementara, sedangkan pada ayat 6 menyebutkan bahwa warga negara atau penduduk hanya memiliki 1 (satu) E-KTP.

E-KTP merupakan salah satu tanda bahwa seseorang diakui secara sah bertempat tinggal di wilayah administratif Indonesia yang tercantum dalam (Peraturan Pemerintah No. 40 Tahun 2019 Tentang Administrasi Kependudukan, 2019) Tentang Administrasi Kependudukan. E-KTP berisi tentang informasi

tertentu seperti Nomor Induk Kependudukan (NIK), nama, alamat, dll. Di sisi lain, pada aplikasi-aplikasi yang telah memerlukan pengambilan data E-KTP selalu melakukan pengetikan ulang sehingga dapat memakan waktu untuk proses input data karena proses integrasi sistem dari pemerintah pusat yang dinaungi oleh Kementerian Dalam Negeri Republik Indonesia tidak membuka akses untuk pertukaran data. Dalam hal tersebut banyak permasalahan-permasalahan yang terjadi seperti *human error* pada saat melakukan input data pada kesempatan ini penulis memperkenalkan salah satu teknologi *Machine Learning* yang terus dikembangkan dalam proses pengenalan karakter yakni Jaringan Saraf Tiruan (JST) dengan metode *Optical Character Recognition* (OCR).

Proses pengambilan data dari E-KTP dapat dilakukan melalui sistem dengan cara mengenali huruf dan angka pada citra E-KTP yaitu dengan melakukan ekstraksi data/informasi. *Optical Character Recognition* adalah metode pengenalan huruf dan angka pada citra atau gambar menjadi tulisan dengan karakter ASCII yang dikenal komputer. Gambar tersebut bisa berasal dari tulisan tangan, printer, scan dokumen, hasil print screen halaman web, foto, dll. (Abdullah, S. S., & Muhammad, F. D., 2020). Dari proses *Optical Character Recognition* akan menghasilkan teks yang sesuai dengan gambar *output scanner*. Akurasi dari teks yang dihasilkan bergantung pada metode pengenalan yang akan digunakan dan kejelasan citra atau gambar yang dijadikan objek untuk dilakukan pengambilan informasi atau data. Ekstraksi informasi (*Information Extraction* atau IE) adalah proses pengambilan fakta dan informasi yang terstruktur dari dokumen tidak terstruktur

atau semi terstruktur. Fakta dan informasi disini merupakan entitas atau informasi yang terdapat pada dokumen.

Penelitian mengenai OCR telah dilakukan sebelumnya oleh banyak peneliti dengan berbagai macam metode (Hartanto et al., 2012) melakukan OCR menggunakan algoritma *Template Matching Correlation*. Pada penelitian ini pengujian dilakukan sebanyak tiga kali dengan ukuran font yang berbeda yaitu 12pt, 20pt dan 36pt. Pada hasil pengujian pertama dilakukan dengan ukuran huruf 12pt menghasilkan tingkat keberhasilan sebesar 92,24%, pengujian kedua dilakukan dengan ukuran huruf 20pt menghasilkan tingkat keberhasilan sebesar 93,54% dan pada pengujian ketiga dilakukan dengan ukuran huruf 36pt menghasilkan tingkat keberhasilan sebesar 92,93%. Tingkat keberhasilan dari 3 pengujian tersebut terdapat akurasi rata-rata sebesar 92,90%.

Penelitian tentang ekstraksi karakter citra dengan metode *Tesseract* yang dilakukan oleh (Ibnutama & Gilang Suryanata, 2020) melakukan ekstraksi karakter citra menggunakan OCR untuk pencetakan nomor kendaraan pada struk parkir. Tujuan pengujian adalah untuk mencegah pendeteksian saat kendaraan meninggalkan tempat parkir. Hasil pengujian terhadap 10 sampel pelat nomor kendaraan menunjukkan tingkat akurasi pengenalan pelat menggunakan *Tesseract*. Hasil penerapan sistem pada 10 sampel pelat nomor memberikan rata-rata nilai presentase akurasi sebesar 96,39 %. Dari hasil tersebut, metode menggunakan *Tesseract* dapat mengenali karakter teks pada gambar dengan baik. Penyebab utama terjadinya kesalahan pengenalan adalah banyaknya noise dan objek lain pada citra

pelat nomor kendaraan yang diperoleh, sehingga proses segmentasi membedakan objek yang bukan pelat nomor.

Kajian selanjutnya adalah Ekstraksi data E-KTP menggunakan *Optical Character Recognition Convolution Neural Network*. (Sugiarta et al., 2021) untuk pengujian pengambilan informasi atau data yang terkandung pada Kartu Tanda Penduduk Elektronik (e-KTP) pada tampilan depan kartu dalam bentuk gambar yang berisi data pribadi pemilik. Ekstraksi karakter memanfaatkan CNN OCR dilakukan sebanyak 35 gambar E-KTP menggunakan metode ORC, dan empat layer CNN dapat membantu mengekstraksi informasi atau data dari E-KTP dengan kecepatan rata-rata 30 detik dan tingkat kesalahan 5%.

Penelitian selanjutnya merupakan Implementasi *Optical Character Recognition (OCR)* (Setiawan et al., 2017), yang menggunakan platform Android untuk menginput kata-kata penerjemah. Sistem dibangun menggunakan pengenalan teks dengan *mobile vision* dan diuji dengan lima teks gambar dan 30 halaman teks kalimat untuk masukan kaya yang akan diterjemahkan dengan material gambar teks. Berdasarkan hasil pengujian *black box* pada *orientation portrait* dan *landscape* disimpulkan bahwa program tidak dapat digunakan dalam *orientation landscape*. Berdasarkan hasil perhitungan *recall* dan *precision* pada gambar teks mendapatkan nilai *recall* sebesar 1,00, nilai *precision* sebesar 1,00 dan nilai *f-measure* sebesar 1,00. berdasarkan perhitungan akurasi penerjemah, diperoleh nilai *recall* sebesar 0,60, nilai *precision* sebesar 0,81 dan *f-measure* sebesar 0,86. Berdasarkan hasil akurasi yang diperoleh, dapat disimpulkan bahwa penerapan beberapa font dan ukuran yang berbeda cukup baik.

Penelitian selanjutnya tentang konversi gambar kartu nama menjadi teks menggunakan teknik OCR dan Jaro-Winkler Distance (Zulhida Putri et al., 2018), dalam penelitian ini kartu nama memiliki informasi seperti nama, jabatan, instansi, nomor ponsel dan alamat yang disimpan ke dalam database. Penelitian ini bertujuan untuk membangun aplikasi yang mampu mentransfer data kartu dengan konversi teks dan pengelompokan data. Algoritma yang digunakan untuk mengekstraksi teks adalah optical character recognition (OCR). Teks yang diekstraksi kemudian dikelompokkan menggunakan algoritma Jaro-Winkler Distance. Hasil pengujian dijalankan sebanyak 5 kali dengan nilai akurasi rata-rata adalah 67% untuk informasi nama, 74% untuk informasi jabatan, 56% untuk informasi instansi, 82% untuk nomor handphone dan 78% untuk informasi alamat.

Pada penelitian selanjutnya tentang analisis sistem pengemalan karakter pada dokumen digital dengan menggunakan metode tesseract (Rahmawati et al., 2021), penggunaan metode tesseract pada dokumen digital pada penelitian ini cocok untuk digunakan dalam sistem optical character recognition, misalnya untuk deteksi karakter pada suatu objek. Sistem dirancang dengan menggunakan Bahasa Pemrograman *Python*, pengujian dilakukan pada 10 gambar dokumen digital, konfigurasi sistem pengujian adalah konfigurasi sistem terbaik berdasarkan parameter kinerja yang digunakan yaitu parameter jarak, rotasi dan opacity. Parameter kinerja (performation) terbaik didapatkan pada jarak 30 cm pada kondisi outdoor sebesar 85,1%. Akurasi rata-rata terendah dicapai pada kondisi outdoor pada jarak 50 cm yaitu sebesar 78,39%. Semakin jauh jarak gambarnya, semakin mempengaruhi akurasi. Hal ini dipengaruhi oleh kemiringan garis, spasi pada garis

yang baru dan cahaya pada foto. Selanjutnya, untuk parameter rotasi tidak ditemukan perbedaan rata-rata akurasi dari masing-masing parameter. Akurasi rata-rata terbaik diperoleh di dalam ruangan yaitu 85,1%. Akurasi terendah diperoleh di dalam ruangan yaitu 84,7% dengan parameter opacity, akurasi rata-rata terbaik diperoleh di dalam ruangan dengan transparansi 25% yaitu 84,5%, sedangkan akurasi rata-rata terendah dengan keadaan di luar ruangan dengan transparansi 75% yaitu sebesar 76%.

Penerapan *Covolution Neural Network* untuk pengenalan karakter pada aksara lampung (Rahmawati et al., 2021). Pada penelitian ini, implementasi aksara lampung disebut juga Had Lampung atau KaGaNa yang merupakan aksara asli dari lampung. *Covolution Neural Network (CNN)* digunakan untuk pengenalan gambar aksara lampung dan klasifikasi pola. CNN memiliki lapisan *convolutional* yang terdiri dari beberapa kombinasi lapisan *convolutional layer*, *pooling layer* dan *fully connected layer*. Dataset dikembangkan dengan mengumpulkan hasil tulisan tangan dari responden yang telah ditentukan dan kemudian dilakukan pemindaian gambar. Kemudian proses tagging dijalankan dan disimpan dalam format YOLO yaitu format TXT. Dari arsitektur CNN yang dibangun berdasarkan hasil evaluasi menunjukkan *loss*, *accuracy* menghasilkan nilai training *accuracy* mendapatkan nilai sebesar 0,57 dan *precision* sebesar 0,87. Hasil yang *accuracy* dan *precision* menunjukkan bahwa model pelatihan baik mendekati angka 1.

Sebuah studi yang menggambarkan tentang perbandingan akurasi pengenalan karakter plat nomor menggunakan tesseract dan data latih emnist (Cahyani et al., 2022). Dalam penelitian tersebut, ekstraksi plat nomor dapat

digunakan untuk berbagai kebutuhan seperti sistem parkir, pengaturan lalu lintas, dan identitas untuk memeriksa jika terjadi kecelakaan. Pengenalan karakter menerapkan metode *template matching* pada huruf dan angka. Menggunakan convolutional neural network untuk melakukan pengenalan karakter dengan data latih EMNIST. Tujuan dari penelitian ini adalah untuk membandingkan penggunaan metode OCR dalam melakukan pengenalan karakter menggunakan Tesseract dan CNN. Data yang diuji adalah 58 gambar mobil dengan 36 kategori kelas karakter yang terdiri dari huruf dan angka. Hasil uji deteksi label menggunakan CNN pada data latih EMINST menunjukkan kinerja yang buruk pada 11 citra dengan akurasi di atas 75%. Diuji dengan Tesseract menggunakan segmentasi tanda plat nomor pada 44 citra memiliki akurasi diatas 75%.

Metode OCR untuk pengenalan citra merupakan metode dalam *deep learning* yang memiliki beberapa keunggulan, efisiensi dan efektifitas. Dengan demikian, banyak penelitian terdahulu yang menggunakan metode OCR sebagai metode pengenalan pola khususnya ekstraksi karakter pada gambar, berdasarkan studi literatur penelitian sebelumnya, penulis menggunakan metode OCR dengan metode *Convolution Neural Network* dengan data latih MNIST Digits untuk ekstraksi NIK pada citra E-KTP menjadi teks, yang nantinya dapat digunakan untuk konfirmasi data pada sistem transaksi pembayaran atau sistem informasi terkait pendaftaran yang memuat data E-KTP serta menjadikan solusi validasi identitas dan validasi data.

1.2. Rumusan Masalah

Berdasarkan latar belakang permasalahan yang telah dijelaskan di atas, maka permasalahan utama dari penelitian ini adalah

1. Membangun model *Multilayer Perceptron* dan model *Convolution Neural Network* dengan data latih MNITS Digit.
2. Membuat pengujian ekstraksi E-KTP menggunakan model *Multilayer Perceptron* dan *Convolution Neural Network*
3. Evaluasi model *Multilayer Perceptron* dan model *Convolution Neural Network* dan tingkat akurasi hasil ekstraksi E-KTP.

1.3. Batasan Masalah

Pembatasan masalah untuk menghindari penyimpangan atau memperluas topik sehingga penelitian lebih terarah dan memudahkan dalam pembahasan sehingga tujuan penelitian akan tercapai. Beberapa hal yang membatasi masalah dalam penulisan ini adalah :

1. Penelitian ini hanya melakukan pengambilan data citra E-KTP yaitu NIK dengan pengambilan angka.
2. Spesial karakter tidak dilakukan proses pengenalan.
3. Foto dan Tanda tangan tidak dilakukan proses pengenalan.
4. Bahasa pemrograman yang dipakai adalah Python.
5. Dataset yang dipakai untuk pelatihan MNIST Digits Recognition.
6. 10 Sampel Pengujian Ekstraksi citra E-KTP.

7. Pengelolaan data, pembuatan model akan dilakukan dengan menggunakan Bahasa Pemrograman Python yang akan dijalankan melalui *Tools* pembantu *Jupyter Notebook*

1.4. Tujuan Penelitian

Mengenai tujuan penulis selama melakukan penelitian, tujuan penelitian ini ada beberapa hal yaitu :

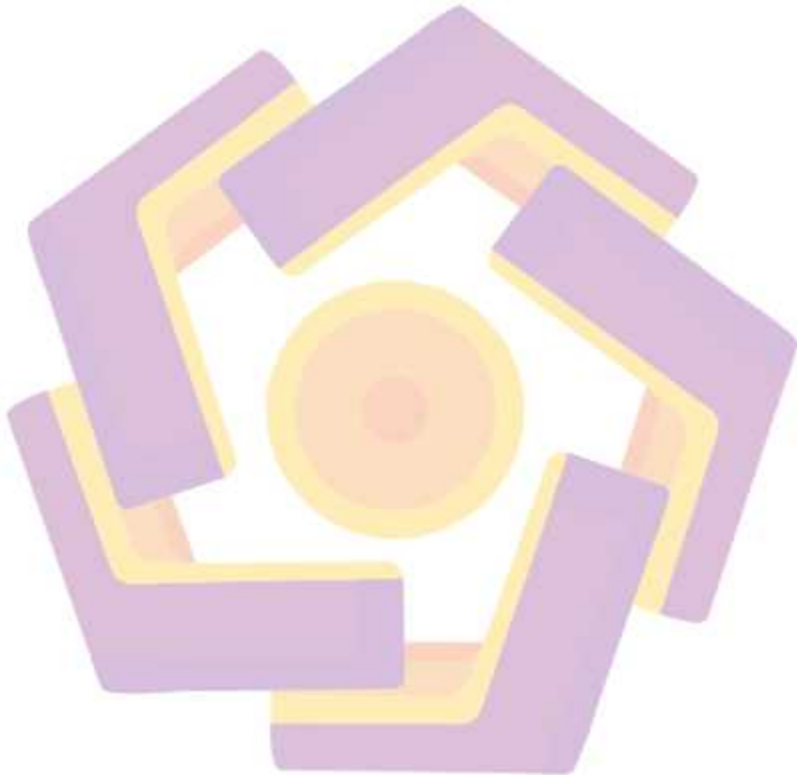
1. Membangun Model *Convolution Neural Network* dengan dataset EMNIST Digits Recognition.
2. Melakukan implementasi ekstraksi NIK pada citra E-KTP menggunakan Model *Convolution Neural Network* dengan preprocessing images.
3. Mengetahui hasil evaluasi perbandingan Model *Multilayer Perceptron* dengan Model *Convolution Neural Network* dan Evaluasi tingkat akurasi hasil ekstraksi.

1.5. Manfaat Penelitian

Mengenai manfaat dari penelitian ini penulis dapat menyimpulkan beberapa hal yang akan dibahas yaitu :

1. Implementasi teknologi ini diharapkan dapat menjadi referensi untuk penelitian selanjutnya tentang model *Multilayer Perceptron* dan model *Convolution Neural Network* dengan data latih MNITS Digit.
2. Bagi peneliti dapat mengetahui tingkat akurasi model *Multilayer Perceptron* dan model *Convolution Neural Network*.

3. Mengetahui nilai evaluasi model *Multilayer Perceptron* dan model *Convolution Neural Network* dengan data latih mnist digits.
4. Dengan implementasi sistem tentunya akan membuka peluang riset lebih jauh lagi baik.



BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Penggunaan E-KTP dalam registrasi menggunakan OCR dengan metode *Template Matching Correlation* (TMC) (Abdullah & Muhammad, 2020). Penelitian teknik pengelolaan citra digital untuk menentukan sebagian kecil citra cocok dengan template citra. Metode penelitian ini melakukan observasi, analisis, desain dan implementasi. Pengamatan dilakukan untuk mengidentifikasi proses pendaftaran pada *marketplace*, mengidentifikasi konten yang termasuk E-KTP, mengidentifikasi OCR dan mengidentifikasi proses *template matching*, dilakukan dengan menganalisa perpektif *as is* dan *to be*. Berdasarkan observasi yang dilakukan, registrasi dilakukan berdasarkan hasil input pengguna. *Username* dimasukkan melalui pengetikan, kemudian mengunduh E-KTP, proses verifikasi tidak dapat dilakukan secara langsung. Hasil selanjutnya mengidentifikasi pemindaian data E-KTP menggunakan OCR dan langkah-langkah OCR dicocokkan dengan algoritma *template matching correlation*. Pada proses selanjutnya digunakan metode *recognition pattern* yaitu menggunakan metode *template matching recognition* dengan citra yang membagi biner menjadi 5 track yang masing-masing track memiliki 8 sektor. Berdasarkan hasil dari penelitian ini, tingkat keberhasilan pemindaian identitas yang terdapat dalam E-KTP dengan memanfaatkan OCR menunjukkan skor akurasi 85%. Dalam pelaksanaannya tidak semua identitas dapat terbaca seperti informasi tentang alamat masih belum sesuai,

Penerapan pengenalan Nomor Induk Kependudukan (NIK) pada e-KTP menggunakan partisi fitur dengan invarian Moment Hu dan *Intensity Of Character* (Mirah, 2018). Penelitian ini membangun sebuah sistem yang dapat membaca dan mengenali NIK secara otomatis pada E-KTP, dengan data latih sebanyak 70 data E-KTP, data latih dan pertukaran data uji melalui *3-fold cross validation*. *Preprocessing* pertama adalah *resize*, *grayscale*, binarisasi, dan segmentasi. Langkah selanjutnya adalah mengekstrak fitur menggunakan metode invarian human Hu dan *Intensity of Character* (IoC), melakukan pengujian data dengan model klasifikasi menggunakan metode *template matching* dan *euclidean distance*. Pada pengujian *3-fold cross validation* terdapat 3 data latih dan 3 data uji, langkah segmentasi akan menghitung akurasi seluruh data, akurasi yang diperoleh sebesar 98,57%, pengujian pada *fold* pertama memiliki tingkat akurasi sebesar 86,96%, pada *fold* kedua tercapai akurasi sebesar 91,30%, pada *fold* ketiga akurasi 78,26%, dengan nilai rata-rata identifikasi NIK E-KTP dengan akurasi sebesar 85,51%.

Implementasi Verifikasi dan Matching E-KTP dengan Faster R-CNN dan ORB (Hudaya et al., 2021), membuat model pendeteksi menggunakan Faster R-CNN dan metode pencocokan menggunakan OBR (Oriented FAST dan Rotated BRIEF) dan KNN-BFN (K-Nearest Neighbor Brute Force Matcher). Tujuan dari implementasi ini adalah untuk mencapai akurasi 80% dan membuktikan pencocokan menggunakan OBR hanya dapat menggantikan teknik OCR. Hasil akurasi implementasi pada model pendeteksi mencapai mAP (Mean Average Precision) sebesar 94%. Namun proses pencocokan hanya mencapai akurasi sebesar 43,46%. Proses pencocokan hanya menggunakan fitur gambar yang kurang

baik dari teknik OCR sebelumnya tetapi meningkatkan waktu pemrosesan dari 4510ms menjadi 60ms.

Implementasi optical character recognition (OCR) selama Pandemi Covid-19 (Firdaus et al., 2021). Untuk mengubah gambar huruf menjadi karakter ASCII yang dikenali komputer, file gambar relatif lebih besar dari pada file teks ASCII berdasarkan kesimpulan bahwa program OCR dapat digunakan untuk memindai dokumen tersebut hilang di Kantor Pelayanan Perbendaharaan Negara (KPPN) Palembang dan program OCR dapat menghemat waktu dalam pembuatan, pemrosesan dan penulisan dokumen.

Menerapkan model fountain pada pengembangan aplikasi *text recognition* dan *text-to-speech* berbasis Android menggunakan Flutter (Ernawati et al., 2021), penelitian ini bertujuan untuk membuat sebuah aplikasi bernama ICSApp (*I Can See Application*) untuk mengubah atau mengonversi gambar menjadi teks menjadi ucapan. Penerapan model *fountain* sangat efisien karena model bersifat reotitif dan hanya terfokus pada satu langkah. Kesimpulan dari penelitian ini adalah dikembangkan menggunakan *Flutter Framework* dengan bahasa pemrograman Dart dan sistem dapat melakukan penerjemahan ke dalam beberapa bahasa seperti bahasa Indonesia, Inggris dan Arab. Pengujian dilakukan dengan menggunakan *Black Box Testing*.

Implementasi pengenalan teks yang transliterasi menggunakan toolkit firebase ML Android (Aditia Muclis & Somantri, 2021), tujuan penelitian adalah menterjemahkan bahasa lain yang terkandung dalam informasi, pengumuman dan pesan dalam gambar, brosur dan poster. Membuat aplikasi untuk membantu

masyarakat desa Gunungendut untuk mengatasi masalah pemahaman bahasa asing agar mudah dipahami melalui aplikasi *smartphone* karena banyak yang tidak mengerti bahasa asing karena kurangnya literasi dalam pembelajaran bahasa asing untuk komunikasi. Penelitian ini menggunakan observasi dan literatur review untuk mengumpulkan informasi dan bahan untuk penelitian, sedangkan metode pengembangan sistem menggunakan *prototype* dimana sistem dapat dikembangkan lebih cepat. Kesimpulan dari penelitian ini adalah perhitungan skor akurasi pengenalan teks adalah sebesar 91,1% dan akurasi yang lebih rendah sebesar 8,9% dari 68 sampel kata.

Desain aplikasi *Optical Character Recognition* Berbasis Backpropagation pada perangkat seluler (Zulkarnain et al., 2020). Tujuan dari penelitian ini adalah alat alternatif untuk merubah teks gambar kedalam teks digital dengan cepat dan menghitung akurasi persamaan karakter. Metode pengumpulan data menggunakan observasi dan literatur review. Sedangkan metode pengembangan sistem menggunakan model *waterfall* dengan *class life cycle* (CLC), metode *waterfall* menggunakan *Communication, Planning, Modeling, Construction* dan *Deployment*. Dari hasil pembahasan mengenai metode yang digunakan, dapat disimpulkan bahwa yang telah diuraikan berupa aplikasi yang dibangun adalah sebuah aplikasi yang terdiri dari teks dalam gambar. Menggunakan kemampuan teknologi pengenalan teks *firebase ML-Kit*. Hasil pengujian dilakukan dengan 5 sampel gambar mencapai rata-rata akurasi sebesar 99% dari persamaan simbol yang dihasilkan.

Pengenalan karakter Hiragana menggunakan 2D *Convolution Neurat Network* (Margarita Susilo et al., 2017). Penelitian ini dilakukan untuk mengklasifikasikan tokoh dalam komik Jepang atau yang sering disebut Manga karena tidak semua orang bisa membaca bahasa Jepang. Oleh karena itu, pengenalan bentuk dibuat menggunakan algoritma CNN. Proses pengambilan tulisan Jepang dari *text bubble* komik kemudian dilakukan proses ekstraksi fitur sehingga dapat dilakukan analisis pola. Hasil yang diperoleh merupakan suatu aplikasi *Element Extrator From Manga* (Komik Jepang) dengan bahasa pemrograman C# dan *library AForge*. Kemudian aplikasi pengenalan pola menggunakan MATLAB untuk mendapatkan akurasi pada *Element Extrator From Manga* didapatkan skor sebesar 88% *frame* komik, 91% teks dan 46% karakter. Selain itu, dalam studi yang menggunakan CNN mencapai akurasi sebesar 96,2% dan *test cross validation* mencapai akurasi sebesar 86%.

2.2. Keaslian Penelitian

Tabel 2.1 Matriks tinjauan literatur dan posisi penelitian
Eksraksi NIK citra E-KTP menggunakan Convolution Neuran Network dengan dataset MNIST Digit

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
1.	Ipteks Sistem Informasi Pembayaran ak Kendaraan Bermotor pada Badan Pendapatan Daerah Provinsi Sulawesi Utara	Wiswati Dalonto, Sianly Lovely Kentey, Jantjen J. Tinangon, Jurnal Ipteks Akuntansi, 2018	Meneliti tentang penerapan sistem informasi pajak kendaraan bermotor untuk mempermudah proses pembayaran pajak kendaraan.	Aplikasi Infomasi pajak kendaraan sulawesi utara dapat digunakan sebagai panduan untuk membayar pajak kendaraan, dan aplikasi juga dapat memberikan informasi kendaraan, informasi pajak dan informasi wajib pajak. Aplikasi ini tidak hanya memudahkan wajib pajak untuk mengetahui jumlah pajak kendaraan yang harus dibayar, tetapi juga sebagai bentuk sosialisasi kepada wajib pajak dalam hal kesadaran membayar pajak.	Berludaskan penerapan ilmu pengetahuan dan teknologi diharapkan aplikasi ini terus berkembang dan harus semakin meluas di kalangan masyarakat.	Penelitian membangun aplikasi untuk validasi pembayaran pajak kendaraan bermotor yang akan digunakan oleh pemerintah daerah. Validasi yang dimaksud menggunakan data citra e-KTP

Tabel 2.1 Lanjutan

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
2	Ekstraksi Informasi/data e-KTP menggunakan Optical Character Recognition Convolution Neural Network	Gunawan Sugiarta, Dianthika Puteri Andini, Syarief Hidayatullah, Jurnal Teknologi Rekyasa, 2021	Meneliti tentang Ekstraksi e-KTP, tampilan depan kartu berupa gambar yang memuat informasi seperti Nomor Induk Kependudukan (NIK), nama lengkap, alamat dan identitas pemilik.	Dengan menggunakan metode OCR dan empat lapis CNN dapat membantu penggalan informasi atau data dalam e-KTP dengan rata-rata kecepatan 30 Detik dan tingkat kesalahan 5%. Data yang dikumpulkan dapat digunakan untuk keperluan lain seperti input data kependudukan, verifikasi alat pembayaran dan keamanan.	Usulan untuk melanjutkan hasil dari ekstraksi e-KTP dapat dikembangkan menjadi keamanan sehingga dapat dibangun secara efisien dari segi waktu dan tidak perlu verifikasi secara manual.	Penulis menggunakan algoritma yang sama namun dalam penelitian yang ditulis menggunakan tujuan yang berbeda. Penulis melanjutkan penelitian terdahulu dengan tujuan validasi pembayaran pajak kendaraan bermotor dengan algoritma yang diterapkan oleh penelitian terdahulu.
3	Klasifikasi citra menggunakan Convolution Neural Network dan K Fold Cross Validation	Ari Peryanto, Anton Yudhana, Rusydi Umar, Journal of Applied Informatics and Computing (JAIC), 2020	Meneliti tentang pengklasifikasian citra pada gambar bunga seperti mawar, tulip dan matahari menggunakan metode CNN dan K-Fold Cross Validation	Menggunakan metode CNN dengan library keras dan tensorflow dalam bahasa pemrograman python yang memiliki rasio kecocokan/akurasi tertinggi sebesar 80,36%, rata-rata akurasi tertinggi sebesar 76,49%, dan untuk nilai terendah sebesar 66,07% dengan rata-rata sebesar 60,31%	Dengan membandingkan metode CNN dengan algoritma klasifikasi lainnya seperti Support Vector Machine (SVM), dapat diberikan saran untuk penelitian tahap selanjutnya	Pada penelitian ini memanfaatkan metode CNN dengan menggunakan library keras dan tensorflow dan K-Fold Cross Validation. Berbeda dengan penulis melakukan metode OCR dengan memanfaatkan Library Firebase ML-Kit untuk menghitung accuracy, recall, percision dan f-score penulis menggunakan metode <i>Confusion Matrix</i>

Tabel 2.1 Lanjutan

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
4	Penerapan Convolution Neural Network pada Pengenalan Aksara Lampung Berbasis Optical Character Recognition	Agus Mulyanto, Erlina Susanti, Farli Rosi, Wajiran, Rohmat Indra Borman, Jurnal Edukasi dan Penelitian Informatika, 2021	Investigasi pendeteksi skrip lumpung yang menangkap data gambar yang disimpan dalam file TXT dalam format YOLO menggunakan metode OCR-CNN	Pembuatan dataset dilakukan dengan mengumpulkan hasil tulisan tangan dan memindai citra, tahap pre-processing diberikan label dan disimpan dalam format YOLO, berdasarkan hasil <i>accuracy</i> mendapat nilai 0,57, <i>precision</i> mendapatkan nilai 0,87, dan <i>recall</i> mendapatkan nilai 0,47. Hasil evaluasi menunjukkan <i>accuracy</i> dan <i>precision</i> yang baik. Tetapi untuk hasil <i>recall</i> memiliki nilai yang rendah.	Saran untuk tahap penelitian selanjutnya pelabelan dilakukan untuk menghasilkan laporan yang harus membuat prediksi yang baik dan harus ditambahkan ke bahan.	Penulis memanfaatkan algoritma yang sama namun dari sisi proses OCR memiliki perbedaan. Penelitian terdahulu membuat dataset untuk mengenali citra yaitu aksara lumpung. Sedangkan penulis hanya melakukan pengambilan karakter pada citra e-KTP dengan metode yang berbeda.
5	Ekstraksi Karakter Citra Menggunakan Optical Character Recognition untuk Pencetakan Nomor Kendaraan Bermotor	Khairi Ibtnutama, Mhd. Gilang Suryanata, Jurnal Media Informatika Budidarma, 2020	Menceliti tentang keamanan pada sarana parkir dengan menambahkan nomor pelat ke tanda terima parkir menggunakan OCR dengan Tesseract.	Hasil penerapan sistem pada 10 contoh nomor pelat kendaraan menghasilkan rata-rata persentase akurasi hasil ekstraksi sebesar 96,39%, yang menunjukkan bahwa metode OCR menggunakan Tesseract dapat melakukan pengenalan karakter teks pada gambar dengan sangat baik.	Lakukan penelitian dengan sedikit informasi, sehingga mendapatkan akurasi yang lebih jelas, dan selama segmentasi, sehingga menentukan karakter yang akan diambil, karena banyak benda lain pada pelat nomor kendaraan, yang akan menyebabkan banyak kesalahan.	Penulis melakukan ekstraksi data pada citra e-KTP, berbeda dengan penelitian terdahulu yang melakukan ekstraksi citra pada plat Kendaraan Bermotor. Penerapan pada penelitian terdahulu berbeda dengan yang dilakukan penulis, penelitian terdahulu menerapkan untuk pencetakan struk parkir. Namun penulis melakukan penerapan untuk validasi pembayaran pajak kendaraan bermotor.

Tabel 2.1 Lanjutan

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
8.	Implementasi Text Reconniter yang diterjemahkan Bahasa Lain dengan Firebase ML-Kit Berbasis Android	Pascal Aditia Muclis, Somantri, Jurnal Informatika Universitas Pamulang, 2021	Menceliti tentang penerapan aplikasi seluler yang memfasilitasi pemahaman bahasa asing.	Sistem mampu mengenali teks berbahasa asing (Inggris), yang kemudian diterjemahkan ke dalam bahasa Indonesia menggunakan <i>library firebase ML Kit</i> . Hasil perhitungan akurasi pengenalan teks mendapatkan akurasi sebesar 91,1%, dan akurasi yang kurang sebesar 8,9% dari 68 sampel kata.	Saran dari penelitian ini adalah pertukaran bahasa tidak hanya satu bahasa (inggris - indonesia) tetapi beberapa bahasa.	Penulis dengan penelitian terdahulu menggunakan <i>library firebase ML-Kit</i> dan teknologi OCR yang sama sama, namun pada penelitian terdahulu fungsi <i>library</i> untuk menerjemahkan teks bahasa asing ke bahasa indonesia dan pada peneliti <i>library</i> yang digunakan untuk ekstraksi data c-KTP
9.	Perancangan Aplikasi Optical Character Recognition Berbasis Backpropagation pada Perangkat Android	Toby Zulkarnain, Suharyanto, Anton, Inti Nusa Mandiri, 2020	Penerjemah gambar menjadi teks dengan berbasis mobile memanfaatkan	Aplikasi OCR berbasis backpropagation pada perangkat mobile dengan menerapkan teknologi <i>firebase ml-kit</i> dapat mempercepat proses perhitungan. Hasil yang didapatkan dari hasil pengujian 5 contoh gambar memberikan akurasi 99% kemiripan karakter gambar dengan hasil teks.	Kelemahan penelitian ini adalah pembahasannya terlalu singkat sehingga tidak mudah dan perhitungan untuk mendapatkan nilai tidak dibahas. Usulan dari penelitian ini adalah untuk menambahkan jumlah sampel dan membuat perhitungan yang akurat dengan menggunakan metode <i>Confusion Matrix</i> ...	Penulis dengan penelitian terdahulu menggunakan <i>Library</i> yang sama yaitu <i>firebase ML-Kit</i> namun pada penelitian terdahulu menggunakan metode yang berbeda yaitu menggunakan metode <i>backpropagation</i> sedangkan pada peneliti menggunakan metode <i>confusion matrix</i>

Tabel 2.1 Lanjutan

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
10.	Pengenalan Plat Nomor Menggunakan Optical Character Recognition Berbasis Android untuk meningkatkan keamanan kendaraan di Universitas Nurul Jadid	Fathorazi Nur Fajri, Akhmad Khairi, Subdil ibadi, Agung Maulana, Jurnal Kecerdasan buatan, Komputasi dan Teknologi Informasi	Tujuan dari penelitian ini adalah untuk meningkatkan keamanan kendaraan di Universitas Nurul Jadid	Aplikasi pengenalan plat nomor menggunakan teknologi optical character recognition berbasis Android membuat prototype aplikasi. Akurasi yang dihasilkan masih sangat rendah karena adanya variasi font pada plat nomor kendaraan. Sulitnya membedakan antara Q dan O, I dan 1, dikarenakan warna plat yang putih dan penggunaan variasi atau lesins pada plat seperti bautm stiker dan penutup plat.	Kelemahan dari penelitian ini adalah tidak bisa menghitung akurasi yang didapat, hanya penjelasan akurasi yang sangat kecil. Saran penelitian ini untuk perbaikan dapat menggunakan metode deep learning agar pembacaan plat nomor lebih akurat dan menyebutkan library atau metode apa yang digunakan dalam penelitian ini.	Paada penelitian ini menggunakan OCR Berbasis android untuk pendeteksi plat kendaraan, sedangkan penulis membuat penelitian menggunakan OCR dengan ekstraksi citra e-KTP dengan memanfaatkan Library Firebase ML-Kit dan metode perhitungan Accuracy, Recall, Percision dan F-Score menggunakan Confusion Matrix

2.3. Landasan Teori

2.3.1. Kartu Tanda Penduduk Elektronik

Kartu tanda penduduk elektronik (e-KTP) adalah dokumen identitas resmi penduduk, yang dikeluarkan oleh lembaga penegak hukum dan berlaku di seluruh Indonesia. Dalam Peraturan (UU RI No. 24 Tahun 2013 Tentang Perubahan Atas UU No. 23 Tahun 2006 Tentang Administrasi Kependudukan, 2013), dalam peraturan ini, informasi perseorangan yang tertulis dalam e-KTP berupa NIK, nama lengkap, alamat, dan yang lain. E-KTP memiliki chip yang dapat menyimpan berbagai macam data. Penyimpanan data chip sesuai dengan standar internasional NISTIR 7123 dan dengan dokumen perjalanan yang dapat dibaca mesin ICAO 9303 dan dengan spesifikasi UE Passport Specification 2006. E-KTP diunduh dalam format standar ISO 7810, dengan *Form Factor* ukuran kartu kredit yaitu 53,98 mm x 85,60 mm.

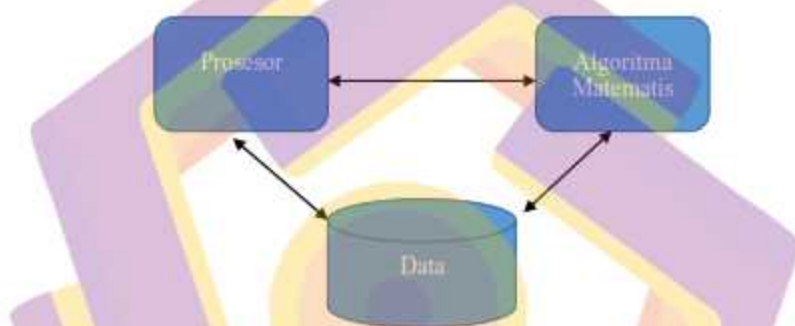


Gambar 2. 1 Contoh citra e-KTP

2.3.2. Machine Learning

Penjelasan tentang *Machine Learning* dalam buku yang berjudul pengenalan machine learning (Kurniawan, 2020) adalah ilmu yang mempelajari algoritma komputer yang mampu mengenali pola dalam data dengan tujuan

memahami berbagai jenis data dengan sedikit mungkin campur tangan manusia dalam tindakan nyata. Dengan pembelajaran mesin, dapat menciptakan mesin yang belajar dari informasi yang ada dan membuat keputusan sendiri tanpa harus diprogram ulang. Secara umum, machine learning berada di bawah payung Artificial Intelligence (AI). Sekarang manufaktur mesin dijelaskan, tiga komponen dasar diperlukan yaitu matematis, prosesor komputer, dan data.

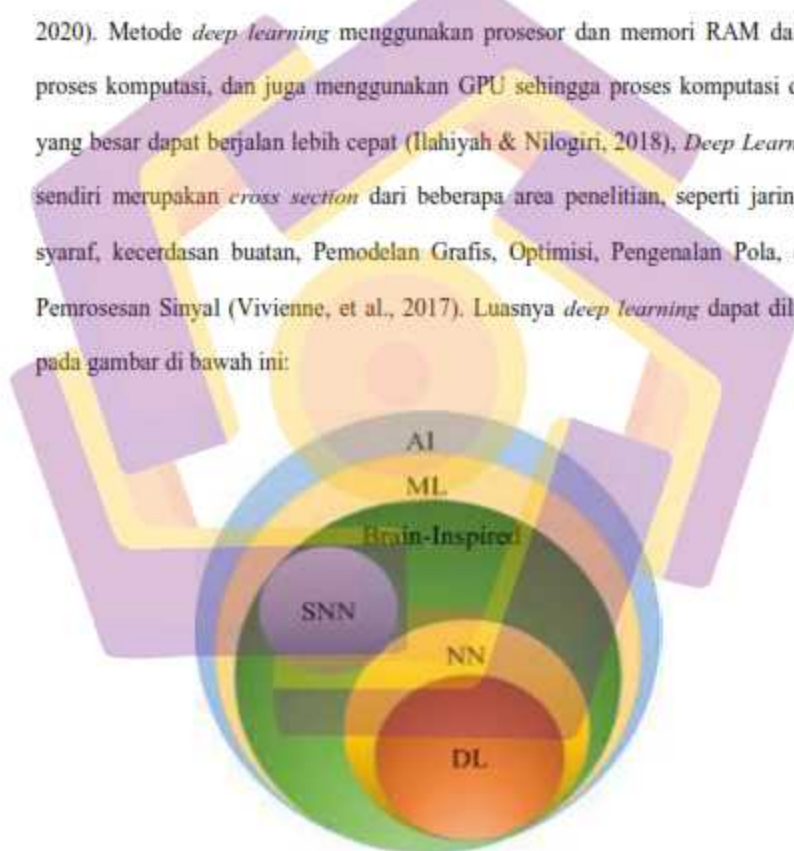


Gambar 2. 2 Proses Machine Learning
(sumber : Kurniawan, 2020)

Istilah *machine learning* pertama kali didefinisikan oleh Arthur Samuel ditahun 1959. Menurut Arthur, *machine learning* adalah cabang ilmu komputer yang memberikan kemampuan pembelajaran kepada komputer mempelajari sesuatu tanpa pemrograman eksplisit. *machine learning* adalah cabang terapan dari Artificial Intelligence (kecerdasan buatan) yang berfokus pada pengembangan sistem yang dapat belajar “dengan sendirinya” tanpa perlu manusia memprogramnya berulang kali.

2.3.3. Deep Learning

Deep learning (DL) adalah kelas teknik pembelajaran mesin (*machine learning*) yang menggunakan pemrosesan data berlapis untuk mengekstraksi, mengubah, mengubah, menganalisis, dan mengklasifikasikan pola, khususnya untuk memahami data seperti gambar, audiom dan teks (Awangga & Batubara, 2020). Metode *deep learning* menggunakan prosesor dan memori RAM dalam proses komputasi, dan juga menggunakan GPU sehingga proses komputasi data yang besar dapat berjalan lebih cepat (Ilahiyah & Nilogiri, 2018), *Deep Learning* sendiri merupakan *cross section* dari beberapa area penelitian, seperti jaringan syaraf, kecerdasan buatan, Pemodelan Grafis, Optimisi, Pengenalan Pola, dan Pemrosesan Sinyal (Vivienne, et al., 2017). Luasnya *deep learning* dapat dilihat pada gambar di bawah ini:



Gambar 2. 3 Ruang Lingkup Deep Learning
(Sumber : Vivienne, et al., 2017)

Jenis *deep learning* yang banyak digunakan adalah *Deep Supervised Learning* (DSL) membangkitkan suatu fungsi yang memetakan *input* (dengan sejumlah atribut) ke *output* (label kelas) berdasarkan data berlabel yang diberikan, kualitas hasil pembelajaran sangat bergantung pada validitas label pada data latih (kesesuaian *input* dan *output*) (Alom et al., 2018). Dalam penelitiannya juga menyebutkan, Ada beberapa model *deep learning* yang termasuk ke dalam *deep supervised learning* berupa *Deep Neural Network* (DNN), *Convolution Neural Network* (CNN), *Recurrent Neural Network* (RNN), *Long Short Term Memory* (LSTM), *Gated Recurrent Unit* (GRU), *Restricted Boltzmann Machines* (RBM), dan *Recursive Neural Tensor Network* (RNTN).

Algoritma *deep learning* memiliki fitur yang unik yaitu sebuah fitur yang mampu mengekstraksi secara otomatis. Algoritma ini sangat penting dalam sebuah kecerdasan buatan karena mampu mengurangi beban pemrograman dalam memilih fitur yang eksplisit dan algoritma ini dapat digunakan untuk memecahkan permasalahan yang perlu pengawasan (*supervised*), tanpa pengawasan (*unsupervised*), dan semi terawasi (*semi supervised*) dalam berbagai aplikasi seperti pengenalan citra, pengenalan suara, klasifikasi teks dan sebagainya (Lecun et al., 2015).

2.3.4. Optical Character Recognition

Optical character recognition (OCR) adalah proses pengenalan teks pada gambar untuk digunakan sebagai data baru yaitu teks. Saat menggunakan pengenalan teks, gambar tulisan tangan, gambar komputerm dan font dapat menjadi

informasi. OCR memiliki banyak aplikasim seperti mengimplementasikan mesin penerjemahan dari bahasa Indonesia ke bahasa Inggris (Setiawan et al., 2017).



Gambar 2. 4 Optical Character Recognition
(sumber : (Fathorazi Nur Fajri et al., n.d.))

Penelitian (Hartanto et al., 2012) menjelaskan bahwa *Optical Character Recognition* (OCR) adalah aplikasi komputer yang digunakan untuk mengenali gambar huruf dan angka yang diubah menjadi file tetulis. Secara umum, proses OCR ditunjukkan pada gambar di bawah ini :



Gambar 2. 5 Proses OCR secara umum.
(sumber :Hartanto et al., 2012)

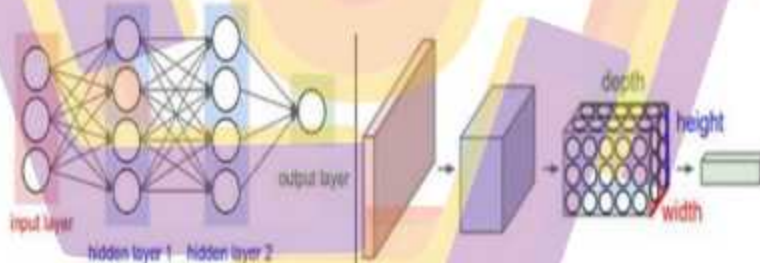
OCR terdiri dari delapan proses yaitu pemindaian, segmentasi, pra-, pemrosesan, visualisasi, ekstraksi fitur, pelatihan dan pengenalan atau pasca-pemrosesan (Sugiarta et al., 2021). Dalam penelitian ini, proses pelatihan dan pengenalan menggunakan CNN dimanfaatkan untuk proses training terhadap karakter pada citra gambar yang terdiri dari *convolution layer*, *pooling layer* dan *fully connected layer*.

OCR dikembangkan oleh Emanuel Goldberg pada tahun 1914. Sistem OCR terus berkembang sehingga dapat mencapai akurasi yang lebih baik bahkan dalam situasi dimana karakter sulit untuk dikenali. Menerapkan OCR ke program membutuhkan *library* yang memungkinkan program menerapkan OCR untuk menerjemahkan algoritma pembacaan (Firdaus et al., 2021). Ini digunakan di berbagai bidang seperti membaca data secara otomatis dan memasukkan langsung ke database. Misalnya, memindai paspor, catatan bank, sekuritas, mengidentifikasi plat nomor dari video atau gambar yang diambil kamera, dan merupakan penyimpanan konten paling populer. Buku sumber primer dan naskah sejarah (Firdaus et al., 2021).

2.3.5. Convolution Neural Network

Convolution Neural Network (CNN) adalah salah satu *neural network* yang biasa digunakan dalam pemrosesan citra digital, *convolution* adalah matriks untuk menyaring citra (Abhirawa et al., 2017). CNN terdiri dari banyak neuron dengan bobot dan bias yang dapat dilatih. Setiap neuron dalam CNN dapat menerima banyak *input*, melakukan operasi aritmatika untuk menghasilkan *output*, dan

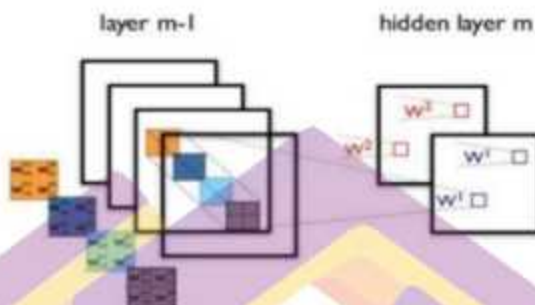
terhubung ke *neuron* lain. Nama konvolusi sendiri merupakan operasi aljabar linear dan merupakan salah satu dari sekian banyak layer yang dimiliki CNN (Suyanto et al., 2019). Arsitektur CNN terdiri atas satu lapis masukan, lapisan keluaran dan beberapa lapisan tersembunyi. Lapis tersembunyi biasanya mencakup *Convolution Layer*, *Pooling Layer*, *Normalization Layer*, *ReLU Layer*, *Fully Connected Layer*, dan *Loss Layer* (Alom et al., 2018). Studi tersebut juga menjelaskan bahwa CNN menggunakan arsitektur tiga dimensi (3D) yaitu lebar, tinggi, dan kedalaman, setiap lapisan mengubah volume input 3D dari dalam volume output 3D. Masukan berupa gambar berwarna, dengan lebar dan tinggi mewakili dimensi *Red*, *Green*, *Blue* (RGB). Konsep dan pengoperasian CNN memiliki kemiripan dengan MLP (*Multilayer Preception*), di CNN setiap *neuron* ditampilkan dalam dua dimensi, berbeda dengan di MLP yang setiap *neuron* hanya memiliki satu dimensi ukuran



Gambar 2. 6 Arsitektur MLP Sederhana
(Sumber : (Pratiwi et al., 2021))

Pada CNN, data yang dipropagasikan pada jaringan adalah ada dua dimensi, sehingga operasi linear dan parameter bobot pada CNN berbeda. Pada CNN operasi linear menggunakan konvolusi, sedangkan bobot tidak lagi satu dimensi saja,

namun berbentuk empat dimensi yaitu merupakan kumpulan *kernel* konvolusi terlihat pada gambar berikut deminsi bobot pada CNN :

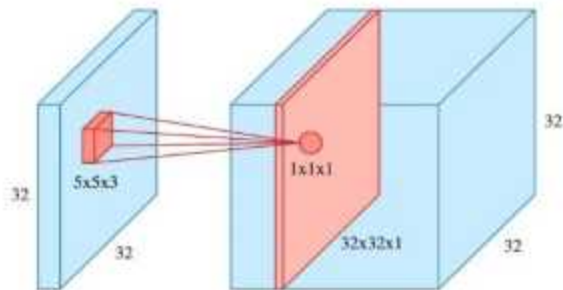


Gambar 2.7 Proses Konvolusi pada CNN
(Sumber : (Pratiwi et al., 2021))

2.3.6. Convolution Layer

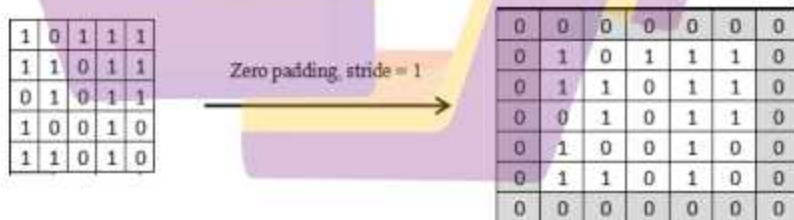
Convolution layer (lapisan konvolusional) adalah langkah pertama setelah lapisan masukan dalam arsitektur CNN. Lapisan ini digunakan sebagai filter untuk mengekstraksi objek dari gambar masukan. Filter ini berisi bobot untuk mendeteksi fitur objek seperti tepi, kurva atau warna (Arrofiqoh & Harintaka, 2018). Filter digunakan berulang kali untuk menghasilkan rangkaian bidang terima, dalam hal ini hasil bidang *receptive*, dalam hal ini hasil bidang *receptive* berupa data tunggal.

Dalam buku (Suyanto et al., 2019) terdapat contoh membangun sebuah *convolution layer* dengan satu lembaran *neuron* berukuran 32 x 32 piksel, setiap lembar terhubung ke area kecil di gambar input, misalnya bidang reseptif 5 x 5 piksel. Untuk setiap *neuron*, dan dicatat bahwa *filter* yang digunakan memiliki ukuran 5 x 5, semua bidang reseptif diuji sebagian tumpang tindih secara parsial, dalam hal ini semua *neuron* ini harus pasti berbagi bobot koneksi (*weight sharing*).



Gambar 2. 8 Convolutin layer dengan satu filter berukuran 5x5
(sumber :(Suyanto et al., 2019))

Convolution Layer dari arsitektur CNN biasanya menggunakan lebih dari satu *filter*. Ketika empat *filter* digunakan, maka *convolution layer* berisi sekumpulan *neuron* yang disusun dalam kisi berukuran $32 \times 32 \times 4$. Input (digambarkan) memiliki empat *neuron* yang melihat area yang sama pada gambar input, sehingga *overfitting* mudah dikontrol (karena ruang solusinya relatif kecil) dan setiap filter mencoba mencocokkan fitur di semua posisi spasial. CNN biasanya menggunakan ukuran lebar atau *stride* = 1 dengan *zero padding* besar $P = \frac{(F-1)}{2}$ dimana P adalah ukuran *padding* dan F bidang reseptif atau terpotong secara spasial, yang tentu saja sesuai dengan ukuran *filter*. Contoh *zero padding* = 1 :



Gambar 2. 9 Teknik Zero padding dengan stride = 1
(sumber :(Suyanto et al., 2019))

Selain itu, karena gambar input menghasilkan nilai yaitu $(1 \times 1) + (1 \times 1) + (1 \times 1) = 3$, proses konvolusi pada dasarnya hanyalah perkalian titik antara filter dan larik

berukuran 3×3 yang dapat diterima. Bidang reseptif *neuron* tumpang tindih secara parsial, sehingga transmisi dilakukan dengan ukuran langkah atau *stride* = 1 hingga semua bidang terisi. Berikutnya adalah ukuran atau skala dalam arsitektur membahas tentang sebuah ukuran dimensi dalam arsitektur CNN. Ukuran gambar input dapat dinyatakan sebagai $[W_1 \times H_1 \times D_1]$ dimana W_1 dan H_1 adalah lebar dan tinggi gambar sedangkan D_1 adalah jumlah kanal *Red Green Blue* (RGB). Menjelaskan dalam bukunya sebagian besar arsitektur CNN dengan empat *hyperparameters* (parameter yang harus ditentukan oleh pengguna), yaitu jumlah filter K , ukuran atau jangkauan spasial bidang reseptif F , lebar langkah atau *stride* S , dan *zero padding* P . Volume awal dapat dihitung menggunakan formula $[W_2 \times H_2 \times D_2]$ dimana :

$$W_2 = \frac{(W_1 - F + 2P)}{S}$$

$$H_2 = \frac{(H_1 - F + 2P)}{S}$$

$$D_2 = K$$

Sebagian besar arsitektur CNN umumnya menggunakan setelan *hyperparameters* sebagai berikut :

- a. Jumlah filter K berupa bilangan pangkat 2, misal 32, 46, 128, 51
- b. Ukuran filter yang digunakan adalah
 - $F = 3, S = 1, P = 1$
 - $F = 5, S = 1, P = 2$
 - $F = 5, S = 2, P =$ berapapun yang sesuai
 - $F = 1, S = 1, P =$ (yang berarti menggunakan konvolusi 1×1)

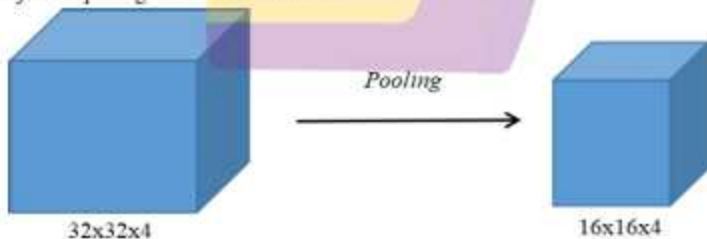
Dalam kebanyakan kasus, CNN di dunia nyata dirancang untuk mengurangi ukuran data dengan terlalu cepat. Dalam praktiknya, mengurangi ukuran data terlalu cepat akan mempengaruhi kinerja CNN. CNN terkadang menggunakan konvolusi 1x1 hanya untuk mengurangi kedalaman data sambil mempertahankan hal-hal seperti lebar dan tinggi. sebagai contoh pada gambar berikut :



Gambar 2. 10 Contoh konvolusi 1x1 dengan 4 filter (sumber : (Suyanto et al., 2019))

2.3.7. Pooling Layer

Pooling layer adalah merupakan layer yang mengambil fitur sebagai input *feature map* dan mengolahnya dengan berbagai macam operasi statistika berdasarkan nilai piksel terdekat (Ilahiyah & Nilogiri, 2018). Agregasi membuat data menjadi lebih kecil, mudah untuk mengontrol *overfitting*, seperti yang ditunjukkan pada gambar di bawah ini :



Gambar 2. 11 Proses pooling untuk mereduksi dimensi data (sumber : (Suyanto et al., 2019))

- A. Umumnya proses *pooling* dilakukan dengan menggunakan *max pooling* atau *average pooling* dengan memilih nilai maksimum atau nilai rata-rata dari range. Contoh ilustrasinya ada pada gambar berikut



Gambar 2. 12 Teknik max pool dan average pool
(sumber : (Suyanto et al., 2019))

Pada bagan *max pooling*, empat nilai di kiri atas memberikan nilai maksimum yaitu 7, untuk mengisi data *pooling* sedangkan empat nilai di kanan atas, kiri bawah dan kanan bawah memberikan nilai maksimum sebesar 6, 5, dan 8 sedangkan pada citra fungsi rata-rata. Sementara itu pada gambar *average pooling*, dari empat nilai di bawah ini, dan untuk melengkapi data hasil *pooling* di bagian kanan atas, kiri bawah dan kanan bawah dihasilkan oleh 3 dan 4 secara berturut-turut. Untuk banyak masalah praktis, teknik *max pooling* umumnya lebih baik dari pada *average pooling*, *L2-norm pooling*, atau teknik lainnya (Suyanto et al., 2019).

2.3.8. Library TensorFlow

Pada tahun 2015, perusahaan teknologi terbesar di dunia yaitu Google merilis *Library TensorFlow*, yang digunakan di sebagian besar produk Google seperti Google Search, Deteksi Spam, Pengenalan Ucapan (Speech Recognition), Google Assistant, Google Now, dan Google Foto. TensorFlow memiliki

kemampuan unik untuk menangani subgraf parsial, yang memungkinkan pelatihan terdistribusi memulai partisi *neural network*. Dengan kata lain, TensorFlow memungkinkan paralelisme model dan paralelisme data (Raharjo, 2022). TensorFlow adalah antarmuka untuk mengekspresikan algoritma pembelajaran mesin dan menjalankan perintah dengan menggunakan informasi dimilikinya tentang suatu objek atau objek yang dikenali dan mampu membedakan satu objek yang dikenali dan mampu membedakan suatu objek dari yang lain (Darma Nurfitra & Ariyanto, 2020) dan TensorFlow berfungsi melakukan pelatihan model menggunakan *Central Processing Unit (CPU)* dan *Graphic Processing Unit (GPU)*.

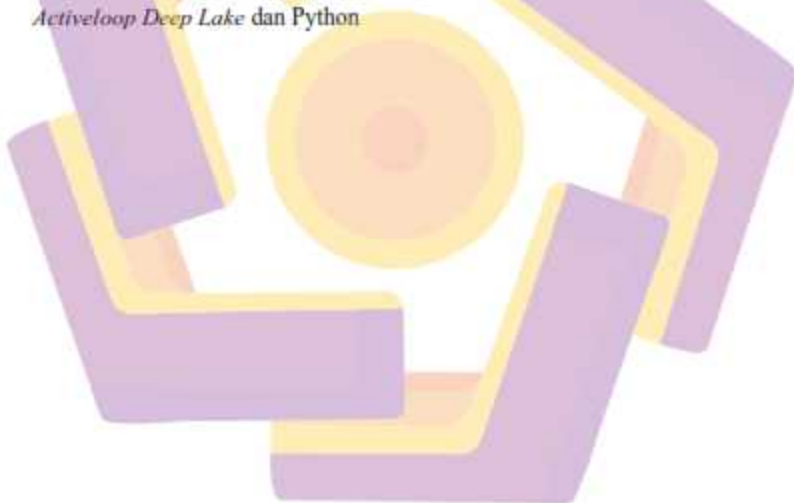
2.3.9. Dataset MNIST DIGITS

Situs web (National Institute Of Standards and Technology, 2019) menjelaskan kumpulan data EMNIST, yang merupakan kumpulan nomor karakter tulisan tangan berasal dari MNIST Spacial Database 19 dan diubah menjadi format gambar 28 x 28 piksel dan struktur kumpulan data yang langsung tersedia untuk MNIST. Dataset MNIST berisai 145.000 gambar alfabet dalam 26 kategori dan 280.000 gambar angka dalam 10 kategori. Sebagai contoh dataset MNIST dapat dilihat dari gambar di bawah ini :



Gambar 2. 13 Sampel Citra pada MNIST Digits

Pengoprasian kumpulan data MNIST diajarkan dengan model PyTorch atau Tensorflow dalam satu baris kode menggunakan paket *Open Source Activeloop Deep Lake* dan Python.



BAB III

METODE PENELITIAN

3.1. Jenis, Sifat, dan Pendekatan Penelitian

3.1.1. Jenis Penelitian

Penelitian yang digunakan peneliti adalah penelitian kualitatif, yang memerlukan pemahaman yang mendalam dan menyeluruh terhadap subjek penelitian untuk menjawab permasalahan, memperoleh informasi, menganalisis, dan memperoleh hasil penelitian, serta situasi dan kondisi yang tertentu. Metode penelitian ini dapat disebut dengan naturalistik karena penelitian dilakukan dalam kondisi alamiah (*natural environment*) disebut juga sebagai metode etnografi karena metode ini sering digunakan dalam bidang antropologi budaya (Sugiyono, 2011:12).

Metode penelitian kualitatif adalah metode penelitian untuk mempelajari kondisi obyek yang alamiah, dimana peneliti sebagai instrumen kuncinya, teknik pengumpulan data dilakukan secara *triangulasi* (gabungan), analisis data bersifat induktif, dan hasil penelitian kualitatif lebih menekankan relevansi daripada generalisasi (Sugiyono, 2011:13).

3.1.2. Sifat Penelitian

Sifat penelitian yang akan dilakukan adalah eksperimen dimana peneliti melakukan sebuah eksperimen guna melakukan sinkronisasi dari data ekstraksi Citra E-KTP yang mana mengambil data tulisan yang terdapat di dalamnya.

3.1.3. Pendekatan Penelitian

Pendekatan kualitatif digunakan oleh peneliti dimana penelitian akan melakukan penelitian sesuai alur yang telah peneliti ini dibuat

3.2. Metode Pengumpulan Data

Metode pengumpulan data digunakan untuk memperoleh informasi yang diperlukan untuk mencapai tujuan penelitian dan merupakan langkah strategis dalam penelitian. Data yang dikumpulkan dengan diberikan sampel, sampel terdiri dari data MNITS sebagai data latih untuk membuat model dan E-KTP sebagai sasaran penelitian. Variabel-variabel yang diteliti adalah informasi NIK pada citra E-KTP dalam sampel penelitian.

3.2.1. Wawancara

Wawancara digunakan sebagai teknik pengumpulan data ketika peneliti ingin melakukan studi pendahuluan untuk menemukan-subyek yang akan diteliti. Wawancara harus mendapatkan informasi yang tidak dapat diperoleh melalui observasi atau survei. Dalam hal ini, wawancara terstruktur adalah wawancara yang digunakan sebagai teknik pengumpulan data. Pada bagian ini penulis melakukan wawancara ke masyarakat dengan sampel acak dan meminta penggunaan data E-KTP dalam penelitian ini.

3.2.2. Observasi dan Pengamatan

Observasi ini bertujuan untuk mengumpulkan informasi melalui pengamatan secara langsung terhadap objek penelitian untuk mendapatkan hasil yang diinginkan. Pengamatan akan mendapatkan pengalaman dan pengetahuan yang sangat personal yang terkadang sulit diungkapkan dengan kata-kata.

3.2.3. Studi Pustaka

Tinjauan literatur dilakukan dengan mengumpulkan data dan informasi untuk digunakan dalam model ekstraksi teks dan gambar. Kartu ID (KTP) untuk mengekstrak teks, yang menggunakan teknik metode *optical character recognition* dengan membandingkan teknik convolution neural network dengan data pelatihan MNIST dan library Tesseract Python.

3.2.4. Dokumentasi

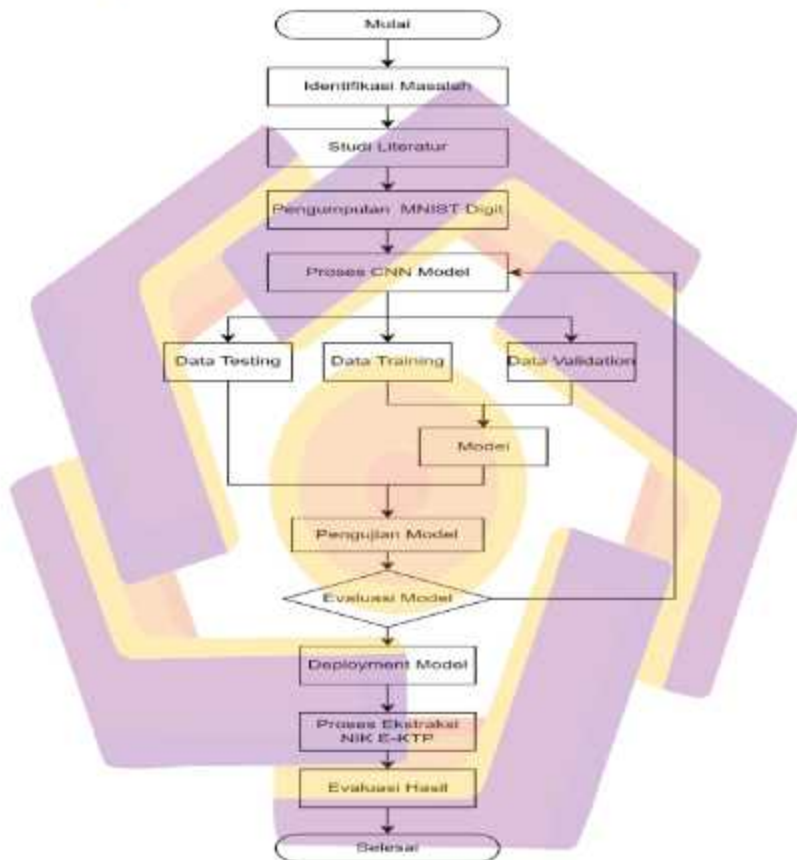
Studi ini juga membutuhkan informasi resmi yang dirinci dalam beberapa dokumen. Untuk keperluan memperoleh informasi tersebut, penelitian ini menggunakan teknik dokumenter. Teknik dokumentasi yang relevan dilakukan dengan cara penggalian informasi berupa buku, arsip, dokumen, tulisan, angka dan gambar, serta keterangan-keterangan dan informasi lainnya, sebagaimana dijelaskan (Sugiyono, 2011). Dokumen yang dibutuhkan informasinya dalam penelitian ini adalah dokumen nasional berupa Kartu Tanda Penduduk Elektronik (E-KTP).

3.3. Metode Analisis Data

Metode analisis data pada penelitian ini menggunakan metode kualitatif yaitu pembahasan dan analisis yang kesimpulannya merupakan hasil penelitian yang telah dilakukan sebelumnya dengan menggunakan kalimat-kalimat berdasarkan pengamatan dan penelitian. Metode penelitian ini bersifat induktif, artinya analisis yang dilakukan dikembangkan menjadi hipotesis menurut skema tertentu.

3.4. Alur Penelitian

Alur penelitian dapat dilihat dari *flow chat* penelitian yang ditunjukkan pada gambar 3.4 dibawah ini.



Gambar 3. 1 Diagram Alur Penelitian

Alur penelitian secara sistematis tahapan demi tahapan dapat dilihat pada gambar 3.1 dan akan dijelaskan sebagai berikut :

1. Identifikasi Masalah

Pada proses ini, semua permasalahan yang di dapat pada saat survei permasalahan akan dirumuskan dan dijadikan tujuan penelitian.

2. Studi Literatur

Dalam proses ini, penulis mencari informasi dengan membaca jurnal penelitian dan buku-buku yang dianggap relevan dengan pertanyaan. Proses ini juga menjadi acuan penelitian untuk memilih metode atau algoritma yang tepat untuk masalah tersebut.

3. Pengumpulan Citra E-KTP

Setelah tahap penentuan algoritma yang digunakan tahap berikutnya yang akan dilakukan adalah mengumpulkan data yang diperlukan dan relevan dengan algoritma yang akan digunakan. Data pada penelitian ini dibagi menjadi dua bagian yaitu data E-KTP dataset.

4. Prossesing Model CNN

Pada tahapan pembuatan Model CNN melibatkan Prossesing data MNIST Digit yang telah dikumpulkan. Dalam pembuatan Model CNN melalui 3 tahapan yaitu Data Testing, Data Training dan Data Validation. Setelah melalui 3 tahap tersebut proses pengujian model dilakukan untuk mendapatkan akurasi.

5. Pengujian Model

Pada tahapan proses pengujian model dilakukan untuk mendapatkan nilai akurasi model.

6. Evaluasi Model

Pada tahapan ini model yang telah dilakukan preprocessing dan tahap latih akan dilakukan setelah pengujian model. Dalam tahapan ini, akan mendapatkan nilai confusion matrix dari pengujian model untuk melihat nilai presisi dan recall.

7. Deployment Model

Pada tahapan ini, model yang sudah selesai akan dilakukan ekstraksi citra.

8. Proses Ekstraksi NIK E-KTP

Pada tahapan proses ekstraksi NIK dilakukan image preprocessing dengan cara resizing image, grayscale, thresholding, dan segmentation. Pada tahapan segmentation dilakukan dengan memisahkan data citra menjadi line yang mana akan diambil nilai NIK.

9. Evaluasi hasil

Pada tahapan evaluasi hasil ini dilakukan setelah proses ekstraksi yang mana akan mendapatkan nilai akurasi pada setiap angka yang terdapat pada NIK.

10. Kesimpulan

Pada tahapan kesimpulan akan membahas hasil evaluasi model CNN dengan data EMNIST Digit dan evaluasi hasil dari ekstraksi NIK E-KTP.

3.5. Metode Evaluasi

Metode scoring adalah metode untuk mengukur seberapa baik metode Convolution Neural Network bekerja dengan data pelatihan MNIST. Metode

skoring yang digunakan dalam penelitian ini adalah *confusion matrix* yang memiliki empat parameter, yaitu *true positive* (TP), *true negative* (TN), *false positive* (FP), *false negative* (FN), dalam mengukur *accuracy*, *percision*, *recall* dan *f1-score* (Widhi Saputro & Wulan Sari, 2019). *Confusion matrix* dapat dilihat sebagai berikut:

Tabel 3. 1 Confusion Matrix

	Prediksi	Dikenal	Tidak Dikenal
Aktual			
Dikenal		TP	FN
Tidak dikenal		FP	TN

True Positive (TP) adalah nilai dimana kategori hasil prediksinya positif dan nilai kategori aktualnya positif. *False positive* (FP) adalah nilai dimana kategori hasil prediksi bernilai positif. *True negative* (TN) adalah nilai kelas hasil prediksi negatif dan nilai kelas aktual negatif. Sedangkan, *False Negative* (FN) adalah nilai dimana kategori hasil prediksi bernilai negatif dan nilai kategori yang sebenarnya bernilai positif. *Confusion matrix* digunakan untuk mengukur *accuracy* kinerja model CNN dengan data latih MNIST melakukan ekstraksi teks dengan persamaan (3.1).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.1)$$

Accuracy menunjukkan kedekatan hasil pengenalan kata dengan nilai sebenarnya. Metrik umum digunakan untuk mengukur kualitas dari data MNIST adalah kombinasi *percision* dan *recall*. *Percision* mengukur kemampuan sistem informasi untuk mendapatkan kembali informasi yang paling relevan dan didefinisikan

sebagai persentase informasi yang dikembalikan. *Precision* adalah proporsi kalimat yang diperoleh, dapat dirumuskan dengan menggunakan persamaan, dan *recall* dirumuskan melalui persamaan.

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

Precision berarti kesesuaian antara informasi yang diterima oleh pengguna dengan jawaban yang diberikan oleh sistem. *Recall* hanyalah hit sistem yang menentukan ulang data yang diperlukan. Kombinasi *precision* dan *recall* sebagai rata-rata harmonik dapat disebut sebagai *F1-Score*, yang dapat dirumuskan persamaan.

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

F1-Score adalah perhitungan evaluatif dari informasi yang dikembalikan, menggabungkan *recall* dan *precision*.

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

Bab ini menjelaskan analisa dan implementasi tahapan perencanaan alur penelitian. Prosesnya melibatkan beberapa langkah, mulai dari pengumpulan data, pendefinisian arsitektur, dan skenario penelitian.

4.1. Membangun Model CNN dataset EMNIST

4.1.1. Pengumpulan Data MNIST Digits

Tahapan pertama yang akan dilakukan peneliti dalam perancangan sistem adalah pengambilan Dataset EMNIST didapatkan oleh penulis dari hasil eksplorasi dataset publik yang diperoleh dari situs kaggle.com, sekumpulan digit karakter tersegmentasi masing-masing menempati 128x128 piksel per raster dan diberi label dari 62 kelas ASCII format heksa desimal sesuai dengan format "0"- "9". "A"- "Z" dan "a"- "z" (Grother et al., 2016). Dataset EMNIST terbagi menjadi enam sub-dataset adalah EMNIST Balanced, EMNIST by merge, EMNIST by class, EMNIST letters, EMNIST digits dan EMNIST MNIST (Mustafa et al., 2022). Dari enam sub-dataset EMNIST, penelitian ini hanya menggunakan EMNIST balanced yang terdiri dari hurup dan angka, model yang dihasilkan dari data pelatihan EMNIST akan diimplementasikan untuk melakukan ekstraksi citra E-KTP. Berikut contoh dari citra dataset EMNIST pada gambar 4.1 dibawah ini.



Gambar 4. 1 Contoh angka dan huruf data EMNIST

Pengambilan dataset MNIST Digits menggunakan format csv yang dibagi menjadi dua bagian, data latihan (*train*) dan data uji (*test*). Pencarian record diimplementasikan dalam kode program berikut.

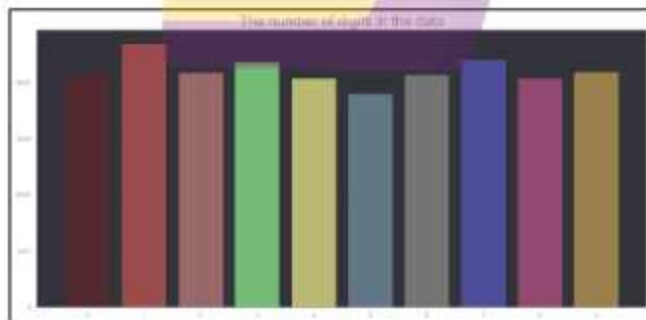
```
train = pd.read_csv("../input/digit-raw/train.csv")
test = pd.read_csv("../input/digit-raw/test.csv")

X_train = train.drop(labels = ["label"], axis = 1)
y_train = train["label"]

X_test = test

print(X_train.shape, X_test.shape)
```

Ouput dari kode program membagi data training sebesar 42.000 dan data testing sebesar 28.000. pada data training pembagian digit angka 0 – 9 memiliki jumlah yang cukup untuk dilakukan pengujian. Hasil ilustrasi data training dapat dilihat pada gambar di bawah ini :



Gambar 4. 2 Grafik dataset MNIST Digits

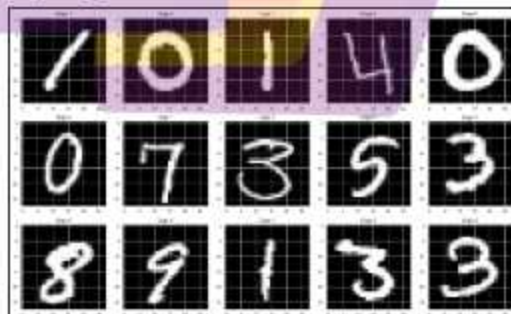
Tabel 4. 1 Jumlah dataset training Emnist Digit

Angka	Jumlah
0	4.132
1	4.684
2	4.177
3	4.351
4	4.072
5	3.795
6	4.137
7	4.401
8	4.063
9	4.188
Total	42.000

Pada tahapan ini pengambilan dataset yang akan digunakan untuk training dan testing model untuk mengenali angka.

```
X_train_plot = X_train.values.reshape(-1, 28, 28)
def Show_example_digits(mono = 'gray'):
    fig = plt.figure(figsize = (10, 14))
    for idx in range(15):
        plt.subplot(3, 5, idx+1)
        plt.imshow(X_train_plot[idx], cmap = mono)
        plt.title("Digit {}".format(y_train[idx]))
    plt.tight_layout()
Show_example_digits()
```

Pada kode program diatas menampilkan contoh ilustrasi dataset yang akan digunakan terlihat pada gambar dibawah ini.



Gambar 4. 3 Contoh ilustrasi dataset MNIST Digit

Tahapan selanjutnya melakukan pembagian data training menjadi dua bagian yaitu bagian validation dan bagian training.

```
X_train, X_val, y_train, y_val = train_test_split(X_train,
y_train, test_size = 0.2, random_state = 2, stratify=y_train)
```

Penjelasan dari kode program diatas X_train dan y_train adalah fitur dan target data training yang sebelumnya sudah dibagi dari dataset utama dengan menggunakan fungsi `train_test_split`, `test_size` sebesar 0.2 menunjukkan bahwa 20% dari data training akan digunakan sebagai data validation, sisanya 80% akan digunakan sebagai data training, `random_state` sebesar 2 seed untuk memastikan hasil pembagian data tetap sama setiap kali dijalankan, `stratify=y_train` menentukan bahwa pembagian data validation akan dilakukan berdasarkan distribusi target, sehingga proporsi target pada data training dan data validation tetap sama. Hasil dari kode ini adalah x_val dan y_val yang merupakan fitur dan x_train dan y_train yang merupakan fitur dan target data training yang baru.

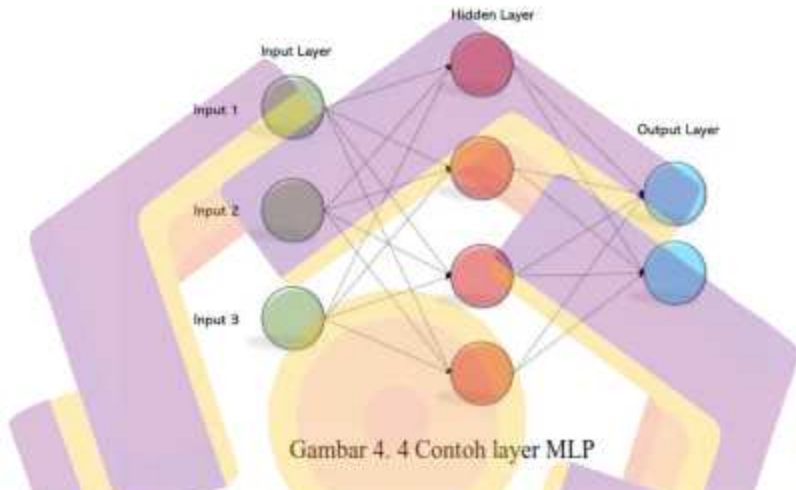
4.1.2. Pembuatan Model MLP dan CNN

Pada fase ini, penulis akan melakukan beberapa skenario pengujian model termasuk skenario pengujian dataset asli. Selama proses, setiap skenario menentukan parameter model input, yang ditulis pada bahasa pemrograman *Python* dengan *Tools Managed Jupyter Server* yang menjalankan *PyCham 2022.3 Professional Edition*.

4.1.2.1. Pembuatan Model MLP (Multilayer Perceptron)

Model MLP memiliki beberapa layer yaitu input layer, hidden layer dan output layer. Setiap layer memiliki beberapa node atau neuron yang bertugas mengolah input dan mengirim output ke layer berikutnya. Kerja dari MLP adalah

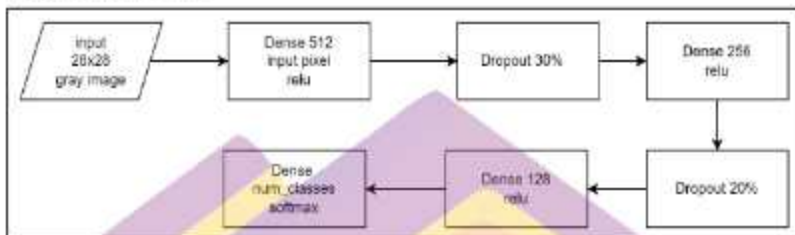
melalui proses forward propagation dimana input data diteruskan melalui setiap layer dan diaktivasi oleh fungsi aktivasi seperti sigmoid, ReLU, than, dll. Kemudian dilakukan proses backpropagation untuk memperbaharui bobot dan bias agar prediksi yang dihasilkan lebih baik.



```
def def_mlp():
    return Sequential([
        Dense(512, input_dim=num_pixels, activation='relu'),
        Dropout(0.5),
        Dense(256, activation='relu'),
        Dropout(0.2),
        Dense(128, activation='relu'),
        Dense(num_classes, activation='softmax')])
```

penjelasan kode program diatas model neural network terdiri dari beberapa layer dense (Fully Connected Layer), layer dropout dan layer aktivasi penjelasan tentang fungsi dense(512, input_dim = num_pixels, activation='relu'): Ini adalah lapisan fully-connected dengan 512 unit dan input_dim = num_pixels. "Relu" adalah fungsi aktivasi dan lapisan Dense(num_classes, activation='softmax'): Ini adalah lapisan output dengan num_classes unit dan fungsi aktivasi "softmax". Fungsi ini digunakan untuk klasifikasi multi-kelas. Sequential mengindikasikan bahwa

lapisan dalam model ini terhubung secara berurutan, dari input sampai dengan output. Skenario alur pembuatan model MLP dapat di ilustarasikan pada diagram berikut dibawah ini.



Gambar 4. 5 Alur Model MLP

```

model.compile(loss='categorical_crossentropy', optimizer='adam',
              metrics=['f1'])
model.summary()
  
```

kode program diatas digunakan untuk mengkompilasi model neural network yang didefinisikan sebelumnya. `loss=categorical_crossentropy` adalah fungsi loss yang digunakan untuk mengevaluasi model, `optimizer=adam` adalah fungsi optimizer yang digunakan untuk melakukan pembaharuan bobot dalam proses pelatihan, `metrics=[f1]` adalah metric yang digunakan untuk mengevaluasi performa model.

Tabel 4. 2 Ringkasan arsitektur layer model MLP

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 512)	401920
dropout (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 256)	131328
dropout_1 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 128)	32896
dense_3 (Dense)	(None, 10)	1290
Total params: 567,434		
Trainable params: 567,434		
Non-trainable params: 0		


```
learning_history = model.fit(X_train, y_train, batch_size=1024,
epochs=40, verbose=2, callbacks=callbacks('simple_mip'),
validation_data=(X_val, y_val));
```

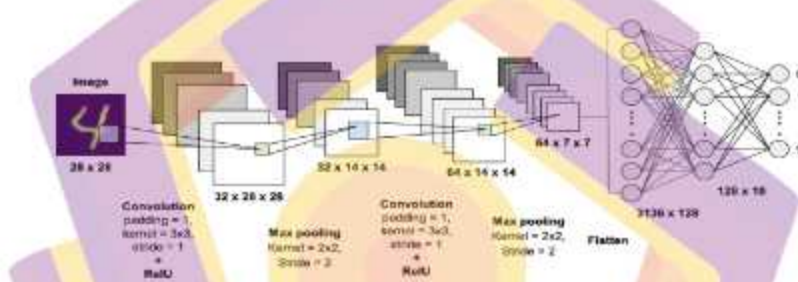
pada kode program diatas digunakan untuk melatih model dengan menggunakan data (X_train, Y_train). Fungsi dari batch_size untuk menentukan jumlah data yang digunakan dalam satu iterasi pelatihan, fungsi epoch digunakan untuk jumlah iterasi pelatihan yang dilakukan, fungsi verbose digunakan untuk seberapa banyak informasi yang akan ditampilkan saat proses pelatihan berlangsung, fungsi callbacks adalah fungsi untuk memodifikasi proses pelatihan selama berlangsung dan fungsi validation_data digunakan untuk data validasi proses pelatihan dan membantu mencegah overfitting.

Tabel 4.3 Contoh hasil pelatihan dataset Model MLP

Epoch 1/40
33/33 - 1s - loss: 0.0040 - f1: 0.9988 - val_loss: 0.0830 - val_f1: 0.9804 - lr: 1.0000e-04 - 802ms/epoch - 24ms/step
Epoch 2/40
33/33 - 1s - loss: 0.0040 - f1: 0.9989 - val_loss: 0.0831 - val_f1: 0.9802 - lr: 1.0000e-04 - 662ms/epoch - 20ms/step
Epoch 3/40
33/33 - 1s - loss: 0.0035 - f1: 0.9990 - val_loss: 0.0847 - val_f1: 0.9798 - lr: 1.0000e-04 - 691ms/epoch - 21ms/step
Epoch 4/40
33/33 - 1s - loss: 0.0033 - f1: 0.9992 - val_loss: 0.0849 - val_f1: 0.9802 - lr: 1.0000e-04 - 691ms/epoch - 21ms/step
Epoch 5/40
33/33 - 1s - loss: 0.0031 - f1: 0.9992 - val_loss: 0.0845 - val_f1: 0.9801 - lr: 1.0000e-04 - 695ms/epoch - 21ms/step
Epoch 6/40
33/33 - 1s - loss: 0.0032 - f1: 0.9991 - val_loss: 0.0851 - val_f1: 0.9796 - lr: 1.0000e-04 - 692ms/epoch - 21ms/step
Epoch 7/40
33/33 - 1s - loss: 0.0031 - f1: 0.9990 - val_loss: 0.0861 - val_f1: 0.9806 - lr: 1.0000e-04 - 678ms/epoch - 21ms/step
Epoch 8/40
33/33 - 1s - loss: 0.0027 - f1: 0.9993 - val_loss: 0.0873 - val_f1: 0.9798 - lr: 1.0000e-04 - 699ms/epoch - 21ms/step
Epoch 9/40
33/33 - 1s - loss: 0.0035 - f1: 0.9991 - val_loss: 0.0873 - val_f1: 0.9801 - lr: 1.0000e-04 - 685ms/epoch - 21ms/step
Epoch 10/40

4.1.2.2. Pembuatan Model CNN dengan Augmentation

membuat model CNN menggunakan TensorFlow karena menawarkan fleksibilitas untuk melakukan iterasi yang cepat untuk melakukan lebih banyak eksperimen guna menghasilkan model dan berjalan pada GPU dan TPU besar (Raharjo, 2022). Untuk model CNN, melatih 3 lapisan CNN dengan dengan ukuran filter 3×3 dan meningkatkan ukuran lapisan dari 16 dan 32 menjadi 64 dan juga menggunakan pembuangan di setiap lapisan untuk mencegah *overfitting*.

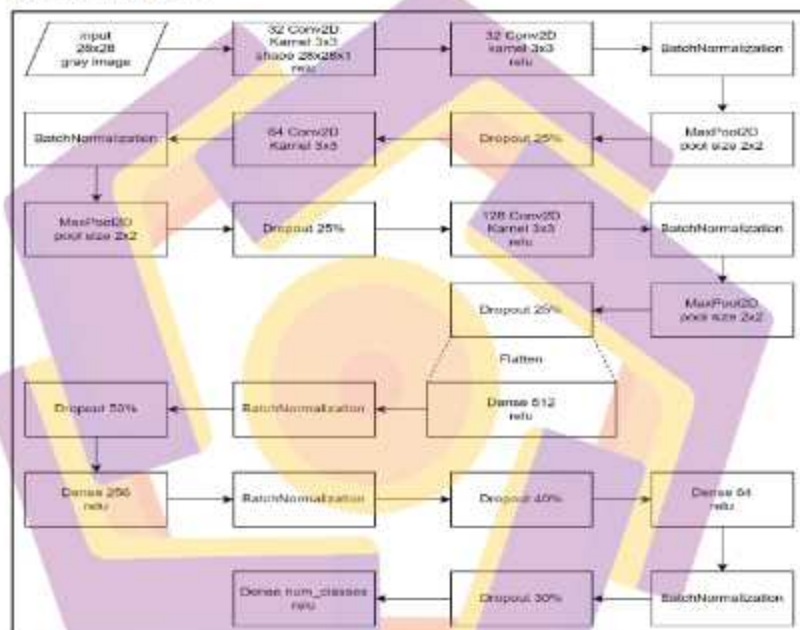


Gambar 4. 6 Contoh Layer CNN

Data sederhana untuk MNIST dengan ukuran 28×28 cukup besar, tetapi untuk mengurangi distorsi gambar diproses terlebih dahulu dan dilakukan *preprocessing* serta pelabelan pada gambar dengan menggunakan *NumPy Array* berukuran (m, n_x) dan $(m, 1)$ yang sesuai dengan jumlah data yang dikumpulkan dan $(n_x = 28 * 28)$ pada jumlah piksel suatu gambar, kemudian format ulang gambar menjadi $(m, 28, 28, 1)$, langkah terakhir *preprocessing* terakhir adalah membagi nilai piksel dengan 255 untuk mendapatkan distribusi. Dari nilai yang diperoleh antara 0 dan 1 (Meany & Arola, 2017). Untuk model dioptimalkan dengan algoritma *Adaptive Moment Estimation (Adam Optimizer)* (Kingma & Ba, 2014), ADAM adalah algoritma pengoptimalan pengganti *stochastic gradient descent* untuk

training model dalam *deep learning* yang menggabungkan sifat-sifat terbaik dari algoritma *AdaGrad* dan *RMSProp* menyediakan algoritma yang lebih optimal dimana dapat menangani gradien yang merambat dan mengandung *noise*.

Skenario alur pembuatan model CNN dapat diilustrasikan pada diagram berikut dibawah ini.



Gambar 4.7 Alur Model CNN

```
def get_cnn_v2():
    return Sequential([
        Conv2D(32, kernel_size=(3, 3), activation='relu',
padding='same', input_shape = input_shape),
        Conv2D(32, kernel_size=(3, 3), activation='relu',
padding='same'),
        BatchNormalization(),
        MaxPool2D(pool_size=(2, 2)),
        Dropout(0.25),
        Conv2D(64, kernel_size=(3, 3), activation='relu',
padding='same'),
        BatchNormalization(),
        MaxPool2D(pool_size=(2, 2)),
        Dropout(0.25),
        Conv2D(128, kernel_size=(3, 3), activation='relu',
```

```
padding='same' ),
    BatchNormalization(),
    MaxPool2D(pool_size=(2, 2)),
    Dropout(0.25),
    Flatten(),
    Dense(512, activation='relu'),
    BatchNormalization(),
    Dropout(0.5),
    Dense(256, activation='relu'),
    BatchNormalization(),
    Dropout(0.4),
    Dense(64, activation='relu'),
    BatchNormalization(),
    Dropout(0.3),
    Dense(num_classes, activation = "softmax")
})
```

kode program diatas menjelaskan tentang pembuatan model CNN menggunakan bibliotek keras, yang mampu melakukan ekstraksi fitur dari gambar dan melakukan klasifikasi. model ini terdiri dari beberapa layer seperti Conv2D untuk melakukan operasi konvolusi pada setiap patch gambar dan memperoleh fitur lokal dengan argumen yang digunakan seperti filter (32), kernel (3x3) aktivasi relu, padding same, dan input_shape. Fungsi layer MaxPool2D untuk melakukan downsampling pada setiap fitur mmenjadi fitur yang lebih besar, argumen yang digunakan ukuran pooling (2x2), fungsi layer Dropout melakukan regulasi pada model dengan mengeset beberapa node dalam model menjadi 0 selama pelatihan, fungsi layer Flatten digunakan untuk mengubah dimensi gambar menjadi 3D ke 1D, dan fungsi layer Densi digunakan untuk fully connected dan melakukan klasifikasi, argumen yang digunakan adalah jumlah node (256) dan aktivasi relu.

```
model.compile(loss='categorical_crossentropy', optimizer='Adam',
metrics=['f1'])
model.summary()
```

kode program diatas digunakan untuk mengkompilasi model neural network yang didefinisikan sebelumnya. loss=categorical_crossentropy adalah fungsi loss yang digunakan untuk mengevaluasi model, optimizer=adam adalah fungsi optimizer

yang digunakan untuk melakukan pembaharuan bobot dalam proses pelatihan, `metrics=[f1]` adalah metric yang digunakan untuk mengevaluasi performa model. Hasil dari fungsi `summary` dapat dilihat pada tabel berikut.

Tabel 4. 4 Ringkasan arsitektur layer model CNN

Layer (type)	Output Shape	Param #
Model: "sequential"		
conv2d_6 (Conv2D)	(None, 28, 28, 32)	320
conv2d_7 (Conv2D)	(None, 28, 28, 32)	9248
batch_normalization (Batch Normalization)	(None, 28, 28, 32)	128
max_pooling2d_3 (MaxPooling2D)	(None, 14, 14, 32)	0
dropout_8 (Dropout)	(None, 14, 14, 32)	0
conv2d_8 (Conv2D)	(None, 14, 14, 64)	18496
conv2d_9 (Conv2D)	(None, 14, 14, 64)	36928
batch_normalization_1 (Batch Normalization)	(None, 14, 14, 64)	256
max_pooling2d_4 (MaxPooling2D)	(None, 7, 7, 64)	0
dropout_9 (Dropout)	(None, 7, 7, 64)	0
conv2d_10 (Conv2D)	(None, 7, 7, 128)	73856
conv2d_11 (Conv2D)	(None, 7, 7, 128)	147584
batch_normalization_2 (Batch Normalization)	(None, 7, 7, 128)	512
max_pooling2d_5 (MaxPooling2D)	(None, 3, 3, 128)	0
dropout_10 (Dropout)	(None, 3, 3, 128)	0
flatten_1 (Flatten)	(None, 1152)	0
dense_10 (Dense)	(None, 512)	590336
batch_normalization_3 (Batch Normalization)	(None, 512)	2048
dropout_11 (Dropout)	(None, 512)	0
dense_11 (Dense)	(None, 256)	131328
batch_normalization_4 (Batch Normalization)	(None, 256)	1024
dropout_12 (Dropout)	(None, 256)	0
dense_12 (Dense)	(None, 64)	16448
batch_normalization_5 (Batch Normalization)	(None, 64)	256
dropout_13 (Dropout)	(None, 64)	0
dense_13 (Dense)	(None, 10)	650
Total params: 1,029,418		
Trainable params: 1,027,306		
Non-trainable params: 2,112		

```
train_aug = ImageDataGenerator(
    featurewise_center = False,
    samplewise_center = False,
```

```

featurewise_std_normalization = False,
samplewise_std_normalization = False,
zca_whitening = False,
horizontal_flip = False,
vertical_flip = False,
fill_mode = 'nearest',
rotation_range = 10,
zoom_range = 0.1,
width_shift_range = 0.1,
height_shift_range = 0.1)

```

pada kode program diatas menjelaskan tentang fungsi ImageDataGenerator yang digunakan untuk melakukan data augmentation pada dataset gambar yaitu memperbanyak jumlah data dengan memuat variasi dari data asli. Variasi data tersebut dapat membantu model mengatasi overfitting dan membantu model lebih generalis, beberapa argumen yang terdapat didalamnya seperti `featurewise_center`, `samplewise_center`, `featurewise_std_normalization`, `zca_whitening`, `horizontal_flip`, `samplewise_std_normalization`, `vertical_flip` yang berisi nilai false, `fill_mode` digunakan untuk mengisi pixel baru setelah melakukan rotasi, dan `rotation_range` sebesar 10 derajat.

```

learning history = model.fit(train_gen, epochs =
40, steps_per_epoch=X_train.shape[0] // 64, validation_data=(X_val,
y_val), callbacks=[callbacks], best_val)

```

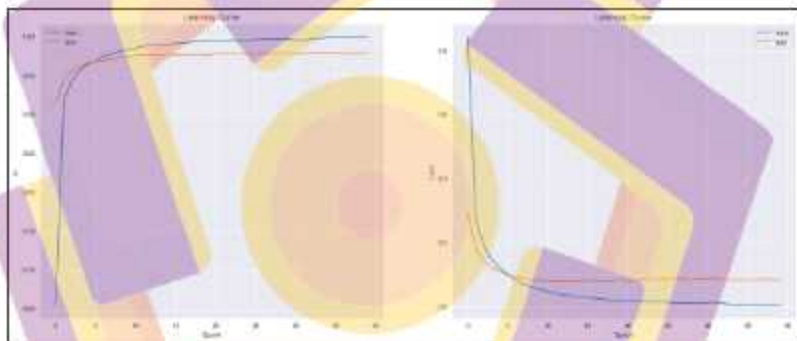
kode program diatas untuk melakukan pelatihan model CNN, fungsi `fit` memiliki beberapa argumen yaitu `train_gen` adalah fungsi augmentation yang digunakan sebagai sumber data untuk melatih model, `epochs` digunakan untuk iterasi pelatihan, `steps_per_epoch` digunakan untuk jumlah batch yang akan diproses pada setiap epoch dan `validation_data` digunakan data validasi saat melatih model.

4.1.3. Evaluasi Model MLP dan CNN

Pada tahapan evaluasi model adalah yang mana cara untuk mengetahui seberapa baik model tersebut digunakan atau mampu melakukan prediksi dan mengklasifikasi data.

4.1.3.1. Evaluasi Model MLP

Hasil dari pembuatan model MLP dengan melakukan 40 epochs menghasilkan nilai *Test Loss* sebesar 8.39% dan nilai *Test Score* sebesar 98.02% dengan nilai *error* sebesar 1.97%, dari hasil tersebut memperoleh kurva grafik gambar sebagai berikut.



Gambar 4. 8 Grafik Kurva Model MLP

Akurasi validasi lebih besar daripada akurasi training itu berarti model yang dibangun sangat terlatih. Berikut ini penyajian hasil prediksi model MPL dengan dataset MNITS Digits terlihat pada tabel berikut ini :

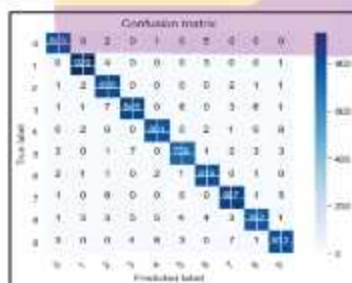
Tabel 4. 5 Confusion Matrix dataset

Dataset	precision	recall	f1-score	Support
0	0.99	0.99	0.99	827
1	0.99	0.99	0.99	937
2	0.97	0.99	0.98	835
3	0.98	0.97	0.98	870

Tabel 4.5 Lanjutan

Dataset	precision	recall	f1-score	Support
4	0.98	0.98	0.98	814
5	0.98	0.97	0.98	759
6	0.98	0.99	0.99	827
7	0.98	0.99	0.98	880
8	0.99	0.96	0.97	813
9	0.98	0.97	0.97	838
Accuracy			0.98	8400
Macro Avg	0.98	0.98	0.98	8400
Weighted Avg	0.98	0.98	0.98	8400

terlihat dari tabel diatas, dataset yang dilatih pada pengujian model MLP sebesar 8.400 data validation yaitu dengan persentase angka 20% dari data training. Dari hasil tersebut, total *accuracy* mendapatkan nilai sebesar 98.0%, rata-rata *precision* mendapatkan nilai sebesar 98.0%, *recall* mendapatkan nilai sebesar 98.0% dan *f1-score* mendapatkan nilai sebesar 98.0% yang artinya hasil pengujian model pada 40 Epochs menghasilkan nilai yang sangat bagus pada semua digit angka. Untuk melihat lebih detail penulis akan menyajikan *plot confusion matrix* dapat dilihat pada gambar berikut ini :



Gambar 4. 9 Confusion Matrix dengan Dataset MNIST Digits

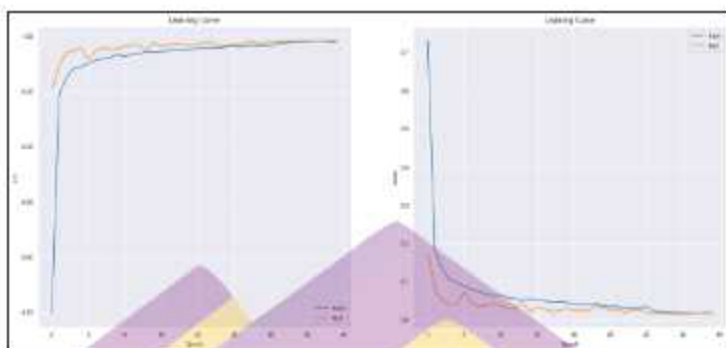
Berdasarkan hasil plot confusion matrix diatas kita dapat melihat masih banyak terdapat nilai *false negative* karena kesulitan terutama pada angka 4 dan 9. Pada kesempatan ini penulis akan menyajikan hasil prediksi sekaligus untuk menyelidiki kesalahan-kesalahan yang paling penting dan perlu mendapatkan perbedaan probabilitas nilai riil dan diprediksi dalam hasil kesalahan yang terjadi, diambil dari top 16 kesalahan yang sangat sering terjadi pada saat training dataset. Berikut ini hasil visual data top 16 dapat dilihat pada gambar berikut ini :



Gambar 4. 10 Hasil Kesalahan Prediksi Model MPL

4.1.3.2. Evaluasi Model CNN

Hasil pembuatan model CNN dengan melakukan 40 epochs menghasilkan nilai *Test Loss* sebesar 0.01% dan nilai *Test Score* sebesar 99.59% dengan nilai *error* sebesar 0.41%. dari hasil tersebut memperoleh kurva grafik gambar sebagai berikut.



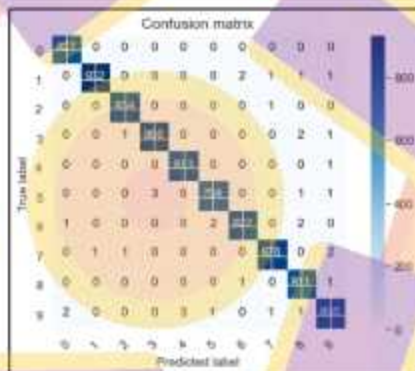
Gambar 4. 11 Grafik Kurva Model CNN

Akurasi validasi lebih besar daripada akurasi training itu berarti model yang dibangun sangat terlatih. Berikut ini penyajian hasil prediksi model CNN dengan dataset MNITS Digits terlihat pada tabel berikut ini :

Tabel 4. 6 Confusion Matrix dataset

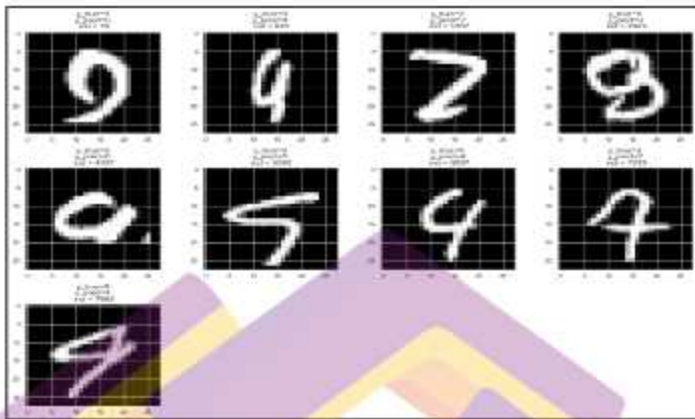
Dataset	precision	recall	f1-score	Support
0	1.00	1.00	1.00	827
1	1.00	0.99	1.00	937
2	1.00	1.00	1.00	835
3	1.00	1.00	1.00	870
4	1.00	1.00	1.00	814
5	1.00	0.99	0.99	759
6	1.00	0.99	1.00	827
7	1.00	1.00	1.00	880
8	0.99	1.00	0.99	813
9	0.99	0.99	0.99	838
Accuracy			0.99	8400
Macro Avg	0.99	0.99	0.99	8400
Weighted Avg	0.99	0.99	0.99	8400

terlihat dari tabel diatas, dataset yang dilatih pada pengujian model MLP sebesar 8.400 data validation yaitu dengan persentase angka 20% dari data training. Dari hasil tersebut, total *accuracy* mendapatkan nilai sebesar 99.0%, rata-rata *precision* mendapatkan nilai sebesar 99.0%, *recall* mendapatkan nilai sebesar 99.0% dan *f1-score* mendapatkan nilai sebesar 99.0% yang artinya hasil pengujian model pada 40 Epochs menghasilkan nilai yang sangat bagus pada semua digit angka. Untuk melihat lebih detail penulis akan menyajikan *plot confusion matrix* dapat dilihat pada gambar berikut ini :



Gambar 4. 12 Confusion Matrix dengan Dataset MNIST Digits

Berdasarkan hasil plot confusion matrix diatas kita dapat melihat masih banyak terdapat nilai *false negative* karena kesulitan terutama pada angka 4 dan 9. Pada kesempatan ini penulis akan menyajikan hasil prediksi sekaligus untuk menyelidiki kesalahan-kesalahan yang paling penting dan perlu mendapatkan perbedaan probabilitas nilai riil dan diprediksi dalam hasil kesalahan yang terjadi, diambil dari top 9 kesalahan yang sangat sering terjadi pada saat training dataset. Berikut ini hasil visual data top 9 dapat dilihat pada gambar berikut ini :



Gambar 4. 13 Hasil Kesalahan Prediksi Model CNN

4.2. Implementasi Recognition citra E-KTP

4.2.1. Pengumpulan data E-KTP

Tahapan pertama yang akan dilakukan peneliti dalam implementasi sistem recognition citra E-KTP adalah mengumpulkan data citra E-KTP sebanyak 10 sampel pengujian untuk E-KTP berwarna dan sebanyak 10 pengujian untuk E-KTP hasil dari fotocopy.

4.2.2. Image Preprocessing dan Segmentation

Proses *pre-processing* merupakan tahapan yang dilakukan sebelum proses utama dari untuk mengenali karakter numerik pada E-KTP. Tahap *pre-processing* yang dilakukan antara meliputi *resizing*, *grayscale*, *thresholding* dan *segmentation*.

a. Resizing

Proses yang dilakukan adalah mengubah ukuran gambar sehingga semua gambar berukuran sama atau lebih kecil. Ukuran masing masing E-KTP

cukup besar yaitu 2448 x 3264. Setelah menjalankan pengujian, dibutuhkan waktu lebih lama untuk menjalankan data dengan ukuran frame yang cukup besar. Oleh karena itu, proses *resize* dilakukan untuk mempercepat proses perhitungan. Proses *resize* menggunakan ukuran 300 x 500, ukuran ini dianggap ideal karena masih menghasilkan gambar yang bagus. Proses *resize* menggunakan library Python dengan OpenCV, yang menyediakan fungsi *resize* dengan kode program sebagai berikut.

```
cv2.resize
cv2.resize(img, (300, 500))
```

b. *Grayscale*

Proses *grayscale* merupakan konversi suatu citra menjadi citra *grayscale* yang digunakan untuk menyederhanakan model citra (Candra et al., 2011). Mengubah citra menjadi *grayscale* dapat dilakukan dengan mengambil semua piksel pada citra, kemudian informasi warna piksel dari tiga warna primer yaitu red, green, dan blue (RGB), menjumlahkannya, lalu membaginya menjadi tiga. Untuk mendapatkan rata-rata yang dapat dipakai menjadi *grayscale*. Proses *grayscale* menggunakan menggunakan OpenCV dengan fungsi `COLOR_BGR2GRAY`. Untuk gambaran yang lebih jelas, lihat kode program sebagai di bawah ini.

```
# Convert to grayscale
img = cv2.imread('./images/1.jpg')
cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

Kode program diatas akan menghasilkan citra menjadi *grayscale*, terlihat contoh pada gambar E-KTP berikut :



Gambar 4. 14 Citra Grayscalee

c. Thresholding

Setelah tahap *grayscale* selesai, proses selanjutnya akan menggunakan proses *thresholding* (biner). proses biner merupakan tahap awal (*pre-processing*) yang bertujuan untuk memisahkan piksel yang termasuk bagian teks dari bagian *background* (Jawas, 2015). *Thresholding* biasanya sebagai nilai ambang batas piksel mana yang masuk kategori latar depan dan latar belakang. Proses binary menggunakan OpenCV dengan memanggil fungsi *threshold*, untuk lebih jelas melihat kode program sebagai di bawah ini.

```

image1 = cv2.imread('images/1.jpg')
img = cv2.cvtColor(image1, cv2.COLOR_BGR2GRAY)
ret, thresh1 = cv2.threshold(img, 120, 255,
cv2.THRESH_BINARY + cv2.THRESH_OTSU)
cv2.imshow('otsu Threshold', thresh1)

```

pada kode prgram diatas akan menghasilkan gambar sebagai berikut :



Gambar 4. 15 Binerisasi Citra

d. Segmentation

Proses segmentation merupakan proses yang sangat penting sebelum melakukan pengenalan pada suatu karakter. Segmentation pada citra merupakan suatu proses yang membagi citra ke dalam beberapa bagian yaitu bagian yang diperlukan dan bagian yang tidak diperlukan oleh sistem (Munir, 2004). Pada kesempatan ini penulis akan melakukan segmentasi pada citra E-KTP dengan mengambil data NIK, yang mana citra tersebut sudah berupa citra biner (hitam-putih) untuk memudahkan dalam melakukan pemisahan pada karakter. Proses identifikasi NIK pada E-KTP harus bisa mendapatkan letak NIK pada objek. Pada proses ini juga akan mendapatkan NIK secara otomatis.

Pada penelitian ini segmentation dilakukan dengan menggunakan metode profil proyeksi vertikal dan horizontal. Pada awal metode yang dipakai adalah proses vertikal proyeksi, melakukan crop line teks citra dengan membuat fungsi sebagai berikut :

```
def crop_text_to_lines(text, blanks):
    x1 = 0
    y = 0
    lines = []
    for i, blank in enumerate(blanks):
        x2 = blank
        print("x1=", x1, ", x2=", x2, ", Diff=", x2-x1)
        line = text[:, x1:x2]
        lines.append(line)
        x1 = blank
    return lines
```

pada fungsi diatas akan menghasilkan nilai sebagai berikut :

Tabel 4. 7 Hasil Crop Citra E-KTP

x1= 0 , x2= 81 , Diff= 81
x1= 81 , x2= 116 , Diff= 35
x1= 116 , x2= 175 , Diff= 59
x1= 175 , x2= 235 , Diff= 60
x1= 235 , x2= 301 , Diff= 66
x1= 301 , x2= 334 , Diff= 33
x1= 334 , x2= 366 , Diff= 32
x1= 366 , x2= 398 , Diff= 32
x1= 398 , x2= 431 , Diff= 33
x1= 431 , x2= 466 , Diff= 35
x1= 466 , x2= 499 , Diff= 33
x1= 499 , x2= 535 , Diff= 36
x1= 535 , x2= 568 , Diff= 33
x1= 568 , x2= 599 , Diff= 31
x1= 599 , x2= 664 , Diff= 65

Ilustrasi nilai-nilai diatas akan dibuat menjadi gambar berikut ini :



Gambar 4. 16 Visuali Sgmentation Line Citra E-KTP

Pada gambar diatas merupakan contoh hasil segmentasi menggunakan vertikal proyeksi, selanjutnya penulis akan mengambil line yang berisi nilai NIK terdapat pada baris ke 3. Proses selanjutnya menggunakan horizontal proyeksi yang mana nilai-nilai teks pada NIK akan dipecah satu persatu. Tahap awal melakukan crop secara vertikal untuk

mengambil nilai angka pada line NIK. Sebagai contoh dapat terlihat pada gambar berikut ini :

3578236109870001

Gambar 4. 17 Nilai NIK pada Citra E-KTP

Setelah dilakukan pengambilan nilai NIK akan dilakukan proses proyeksi secara horizontal yaitu dengan memisahkan setiap angka pada NIK dengan kode program sebagai berikut.

```
ctr, hier = cv2.findContours(in_img, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
rects = [cv2.boundingRect(ctr) for ctr in ctrs]
```

hasil dari kode program Dapat dilihat pada gambar berikut :

3 5 7 8 2 3 6 1 0 9 8 7 0 0 0 1

Gambar 4. 18 Proses Segmentation Angka NIK Citra E-KTP
Pada untuk memisahkan angka pada gambar diatas dilakukan kode program sebagai berikut.

```
images = []
no = 0
for rect in rects:
    cv2.rectangle(in, [rect[0], rect[1]], [rect[0] +
rect[2], rect[1] + rect[3]], (0, 255, 0), 1)
    leng = int(rect[3] * 1.2)
    pt1 = int(rect[1] + rect[3] // 2 - leng // 2)
    pt2 = int(rect[0] + rect[2] // 2 - leng // 2)
    roi = in[pt1:pt1 + leng, pt2:pt2 + leng]
    roi = cv2.resize(roi, (28, 28))
    roi = cv2.dilate(roi, (4, 4))
    images.append(roi)

plt.figure(figsize=(26, 26))
for i in range(16):
    ax = plt.subplot(4, 4, i + 1)
    plt.imshow(images[i])
    plt.axis("off")
```

pada kode diatas akan menghasilkan gambar dengan nilai angka acak paca nilai angka NIK, untuk lebih jelas silahkan lihat gambar berikut :



Gambar 4. 19 Nilai Acak NIK hasil Proses Proyeksi Horisontal

4.2.3. Ekstraksi Citra menggunakan CNN

Proses ekstraksi citra pertama-tama penulis akan melakukan pengambilan model CNN yang sudah disimpan kedalam folder project menggunakan library `tensorflow.keras.models` dengan mengimport fungsi `load_model`. Langkah berikutnya melakukan persiapan data gambar yang sudah melalui preprocessing dan segmentation. untuk dapat melakukan predict model pada citra dilakukan proses penerjemah gambar menjadi angka dengan menggunakan library `tensorflow.keras.utils` dengan mengimport fungsi `img_to_array` untuk melakukan penerjemahan citra menjadi angka yang tersusun menjadi array, selanjutnya dilakukan reshape dengan nilai 28x28 piksel. Untuk lebih jelas lihat kode program sebagai berikut.

```
def load_image(filename):
    img = img_to_array(filename)
    img = img.reshape(1, 28, 28, 1)
    img = img.astype('float32')
    img = img / 255.0
    return img

def run(filename):
    img = load_image(filename)
```

```

model = load_model('./models/final_model.h5')
predict_value = model.predict(img)
digit = argmax(predict_value)
confidence = round(100 * (np.max(predict_value[0])), 2)
print(f"Predict: {digit},\n Confidence: {confidence}%")
return digit, confidence

```

pada kode diatas dengan run akan melakukan predict model dan melakukan perhitungan confidence nilai akurasi ekstraksi. proses selanjutnya menjalankan fungsi-fungsi pada kode program diatas dengan melakukan looping gambar dengan menghitung banyaknya gambar yang dijadikan array untuk lebih silahkan lihat kode program sebagai berikut.

```

plt.figure(figsize=(12, 24))
for i in range(len(images)):
    ax = plt.subplot(4, 1 + 1)
    plt.imshow(images[i])
    digit, confidence = run(images[i])
    plt.title(f"Predict: {digit},\n Confidence: {confidence}%")

```

pada kode diatas akan menghasilkan plot gambar dengan nilai predict dan nilai confidence disetiap gambar. Untuk lebih jelas hasil ekstraksi model CNN dapat dilihat pada tabel dan gambar berikut ini :

Tabel 4. 8 Hasil Predict dan Confidence Model CNN

1/1 [=====] - 0s 47ms/step Predict: 0, Confidence: 99.99%
1/1 [=====] - 0s 47ms/step Predict: 8, Confidence: 93.38%
1/1 [=====] - 0s 47ms/step Predict: 9, Confidence: 92.02%
1/1 [=====] - 0s 54ms/step Predict: 0, Confidence: 100.0%
1/1 [=====] - 0s 50ms/step Predict: 1, Confidence: 98.86%

Tabel 4.8 Lanjutan

1/1 [=====]	- 0s 54ms/step
Predict: 6,	
Confidence: 100.0%	
1/1 [=====]	- 0s 56ms/step
Predict: 3,	
Confidence: 100.0%	
1/1 [=====]	- 0s 32ms/step
Predict: 2,	
Confidence: 99.97%	
1/1 [=====]	- 0s 32ms/step
Predict: 8,	
Confidence: 76.21%	
1/1 [=====]	- 0s 83ms/step
Predict: 7,	
Confidence: 99.99%	
1/1 [=====]	- 0s 50ms/step
Predict: 5,	
Confidence: 100.0%	
1/1 [=====]	- 0s 49ms/step
Predict: 3,	
Confidence: 100.0%	
1/1 [=====]	- 0s 49ms/step
Predict: 1,	
Confidence: 96.93%	
1/1 [=====]	- 0s 48ms/step
Predict: 0,	
Confidence: 99.98%	
1/1 [=====]	- 0s 49ms/step
Predict: 0,	
Confidence: 100.0%	
1/1 [=====]	- 0s 41ms/step
Predict: 7,	
Confidence: 100.0%	



Gambar 4. 20 Hasil Predict dan Confidence Model CNN

4.2.4. Evaluasi Hasil Ekstraksi

Tahapan evaluasi hasil ekstraksi yang mana pengujian dilakukan sebanyak 10 sampel E-KTP berwarna dan 10 sampel E-KTP fotocopy untuk proses Ekstraksi dengan size ratio image 500x800 piksel. Hasil pengujian dengan menggunakan E-KTP berwarna dapat dilihat pada tabel berikut :




Tabel 4.9 Hasil Ekstraksi dengan E-KTP berwarna

No	Citra E-KTP	Predik NIK	Model MLP	Model CNN
1.		[8, 9, 0, 3, 6, 3, 2, 8, 7, 5, 3, 3, 0, 0, 0, 7] 	78.10 %	93.62 %
2.		[0, 0, 7, 0, 9, 4, 0, 2, 0, 7, 8, 5, 2, 7, 3, 6] 	90.87 %	96.84 %
3.		[2, 0, 0, 0, 0, 9, 4, 5, 0, 2, 5, 0, 2, 7, 3, 6] 	84.03 %	98.16 %
4.		[5, 0, 0, 0, 5, 8, 5, 0, 5, 2, 2, 0, 8, 7, 6, 3] 	76.14 %	95.64 %
5.		[0, 0, 9, 3, 2, 0, 0, 0, 6, 0, 4, 6, 7, 0, 3, 6] 	96.47 %	97.41 %




Tabel 4. 10 Lanjutan

No	Citra E-KTP	Predik NIK	Model MLP	Model CNN
6.		[0, 2, 0, 0, 0, 8, 0, 6, 6, 5, 0, 2, 2, 7, 3, 6] 0000 0000 0000 0000	72.52 %	99.62 %
7.		[2, 0, 0, 0, 9, 8, 2, 6, 3, 5, 6, 0, 2, 7, 3, 6] 0000 0000 0000 0000	76.13 %	96.74 %
8.		[9, 2, 0, 0, 4, 8, 4, 0, 8, 0, 4, 0, 2, 3, 7, 6] 0000 0000 0000 0000	83.45 %	95.90 %
9.		[0, 3, 6, 4, 6, 8, 9, 8, 5, 0, 3, 0, 0, 0, 9, 8] 0000 0000 0000 0000	86.91 %	95.72 %
10.		[3, 0, 0, 0, 8, 0, 9, 3, 5, 2, 6, 0, 7, 0, 3, 6] 0000 0000 0000 0000	85.33 %	99.50 %

Tabel 4. 11 Hasil Ekstraksi dengan E-KTP Fotocopy

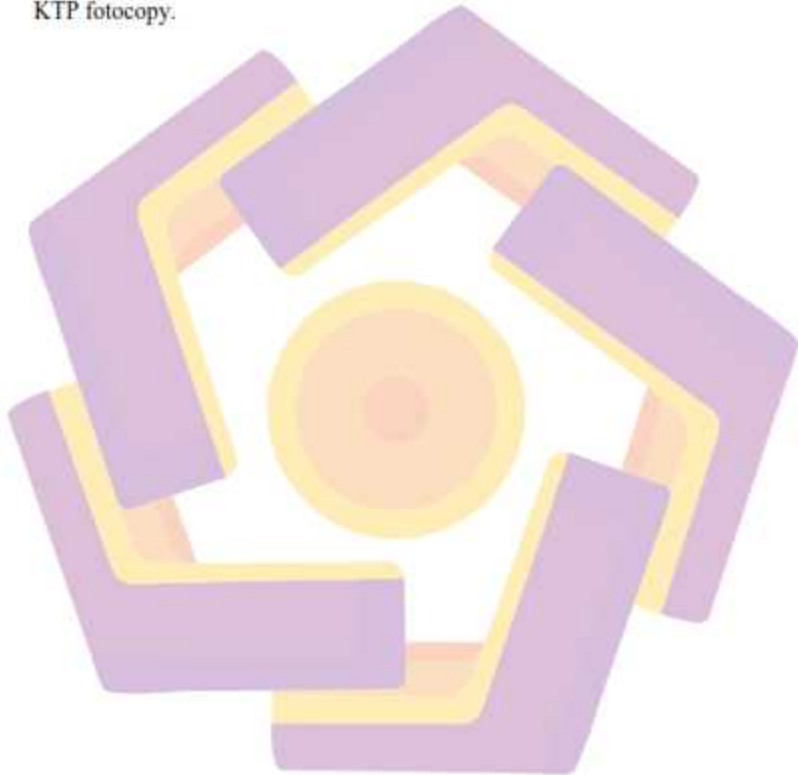
No	Citra E-KTP	Predik NIK	Model MLP	Model CNN
1.		[6,3,5,0,3,0,4,8,0,5,2,0,5,6,0,0] 0000 0000 0000 0000	75.72 %	90.25 %
2.		[6,8,8,8,8,3,0,4,7,3,0,7,9,8,2,8] 0000 0000 0000 0000	77.09 %	80.10 %
3.		[6,4,6,3,6,0,0,7,0,7,9,6,2,2,0,0] 0000 0000 0000 0000	69.06 %	97.41 %

Tabel 4. 12 Lanjutan

No	Citra E-KTP	Predik NIK	Model MLP	Model CNN
4.		[5,0,0,5,8,0,5,0,5,6,6,2,0,2,7,3] 	78.29 %	98.36 %
5.		[0,0,2,0,0,9,2,8,6,0,6,0,7,0,3,6] 	76.04 %	89.57 %
6.		[6,0,2,7,3,5,2,0,3,7,0,2,5,0,2,0] 	82.57 %	99.66 %
7.		[4,8,7,3,6,0,0,8,9,0,8,8,5,0,2,0] 	82.39 %	95.41 %
8.		[6,6,2,0,5,0,8,0,2,7,3,0,0,0,5,7] 	79.97 %	91.85 %
9.		[4,4,8,0,7,8,7,3,6,8,0,0,5,2,0,0] 	76.77 %	90.84 %
10.		[6,0,8,2,2,2,0,2,0,2,7,3,3,0,0,3] 	82.71 %	99.07 %

Dari hasil pengujian pada tabel diatas menunjukkan bahwa model MLP dan CNN dengan data latih MNIST Digit dapat melakukan ekstraksi NIK pada citra E-KTP dengan melakukan preprocessing image dan sigmentation untuk memisahkan digit angka pada variabel NIK secara random sampel. Dari data sampel citra warna pada E-KTP menunjukkan bahwa model CNN lebih baik dalam melakukan ekstraksi dari pada model MLP, terlihat pada hasil rata-rata akurasi model CNN sebesar 96.91% dan hasil rata-rata akurasi model MLP sebesar 82.99% dari 10 sampel E-

KTP berwarna. Sedangkan dengan menggunakan sampel citra hasil dari fotocopy E-KTP, model CNN tetap menunjukkan nilai yang lebih baik daripada model MLP. Dari hasil ekstraksi NIK E-KTP nilai rata-rata akurasi sebesar 93.25% dan nilai rata-rata akurasi model MLP sebesar 78.06% dengan menggunakan 10 sampel E-KTP fotocopy.



BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan hasil kajian ekstraksi NIK pada citra E-KTP menggunakan Convolution Neural Network dengan data training Mnist Digits, dapat disimpulkan bahwa:

1. Rancangan arsitektur model MLP yang menghasilkan *test loss* sebesar 8.39%, *test score* sebesar 98.02% dan *error* 1.98% dengan jumlah 30 *epoch* dan model *Convolution Neural Network* dengan *augmentation* yang menghasilkan *test loss* sebesar 0.01%, *test score* sebesar 99.59% dan *error* sebesar 0.41% dengan jumlah 30 *epoch*.
2. Implementasi text recognition NIK pada citra E-KTP sebanyak 10 sampel E-KTP berwarna dan 10 sampel fotocopy berhasil dilakukan dengan melakukan *preprocessing image* dan *sigmentation* secara proyeksi vertikal dan horizontal untuk memperoleh setiap angka pada NIK yang berjumlah 16 karakter angka. Menghitung akurasi hasil ekstraksi pada E-KTP warna dengan model MLP memperoleh nilai rata-rata sebesar 82.99% dan model CNN memperoleh rata-rata akurasi sebesar 96.91%, Sedangkan dengan menggunakan E-KTP dari hasil fotocopy model MLP memperoleh nilai rata-rata sebesar 78.06% dan model CNN memperoleh nilai rata-rata sebesar 93.25%
3. Hasil evaluasi dengan *confusion matrix* pada model MLP dan model CNN ditunjukkan pada tabel berikut:

Tabel 5. 1 Evaluasi Confusion Matrix Model MLP

Angka	precision	recall	f1-score	Support
0	0.99	0.99	0.99	827
1	0.99	0.99	0.99	937
2	0.97	0.99	0.98	835
3	0.98	0.97	0.98	870
4	0.98	0.98	0.98	814
5	0.98	0.97	0.98	759
6	0.98	0.99	0.99	827
7	0.98	0.99	0.98	880
8	0.99	0.96	0.97	813
9	0.98	0.97	0.97	838

Dari hasil tabel diatas model MLP mendapatkan nilai *macro avarage precision* sebesar 0.98 atau 98%, *recall* 0.98 atau 98%, *f1-score* 0.98 atau 98% dan nilai *support* sebesar 8.400 data training.

Tabel 5. 2 Evaluasi Confusion Matrix Model CNN

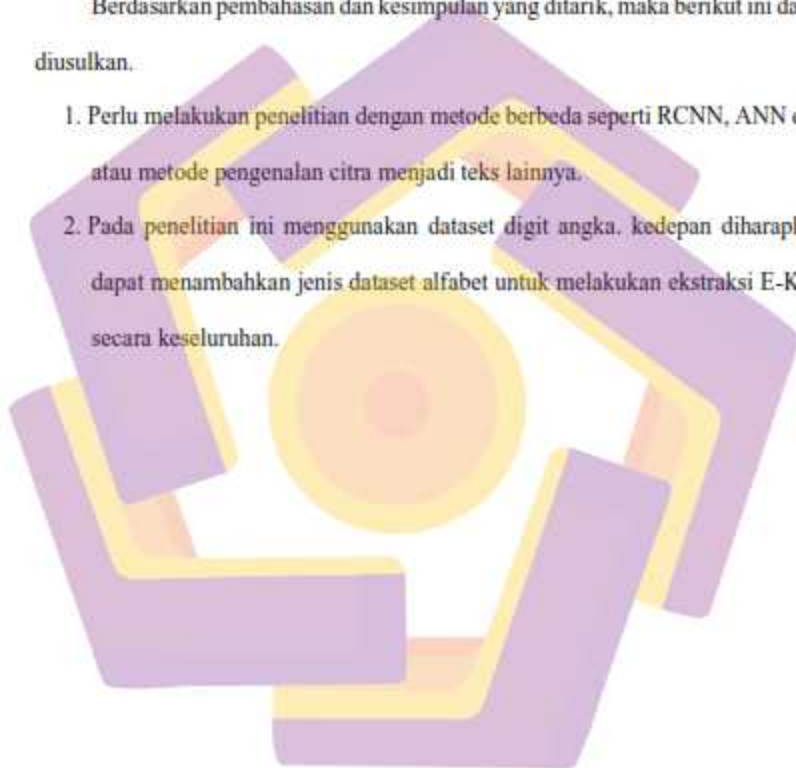
Angka	precision	recall	f1-score	Support
0	1.00	1.00	1.00	827
1	1.00	0.99	1.00	937
2	1.00	1.00	1.00	835
3	1.00	1.00	1.00	870
4	1.00	1.00	1.00	814
5	1.00	0.99	0.99	759
6	1.00	0.99	1.00	827
7	1.00	1.00	1.00	880
8	0.99	1.00	0.99	813
9	0.99	0.99	0.99	838

Dari hasil tabel diatas model CNN mendapatkan nilai *macro average precision* sebesar 0.99 atau 99%, *recall* 0.99 atau 99%, *f1-score* 0.99 atau 99% dan nilai *support* sebesar 8.400 data training.

5.2. Saran

Berdasarkan pembahasan dan kesimpulan yang ditarik, maka berikut ini dapat diusulkan.

1. Perlu melakukan penelitian dengan metode berbeda seperti RCNN, ANN dan atau metode pengenalan citra menjadi teks lainnya.
2. Pada penelitian ini menggunakan dataset digit angka. kedepan diharapkan dapat menambahkan jenis dataset alfabet untuk melakukan ekstraksi E-KTP secara keseluruhan.



DAFTAR PUSTAKA

PUSTAKA BUKU

- Dr. Suyanto, S.T, M.Sc, Kurniawan Nur Ramadhani, S.T, M.T., Satria Mandala, Ph.D. (2019) *Deep Learning Modernisasi Mechine Learning untuk Big Data*. Informatika Bandung, Bandung.
- Sugiyono, P. (2011). *Metode Penelitian Kuantitatif, Kualitatif dan kombinasi (mixed methods)*. Bandung: Alfabeta, ISBN: 979-602-9328-06-6.
- Sosihan Ari Yuana. (2019) *Konsep & Implementasi Pemrograman Python Kasus Big Data*. Lokomedia, Yogyakarta.
- Kurniawan, D. (2020). *Pengenalan Machine Learning dengan Python* (D. Kurniawan, Ed.; 1st ed.). PT. Elex Media Komputindo.
- Raharjo, B. (2022). *Deep Learning dengan Python*.

PUSTAKA MAJALAH, JURNAL ILMIAH ATAU PROSIDING

- Abdullah, S. S., & Muhammad, F. D. (2020). Penggunaan e-KTP untuk Registrasi Otomatis Memanfaatkan Sistem OCR Dengan Metode Template Matching Correlation. *Media Jurnal Informatika*, 12(2).
<https://doi.org/10.35194/mji.v12i2.1224.g1147>
- Abhirawa, H., Jondri, M. S., & Arifianto, A. (2017). Pengenalan Wajah Menggunakan Convolutional Neural Network. *E-Proceeding of Engineering*, 4(2355–9365).

- Aditia Muclis, P., & Somantri. (2021). Implementasi Text Recogniter yang Diterjemahkan ke Bahasa Lain dengan Firebase MI-Kit Berbasis Android. *Jurnal Informatika Universitas Pamulang*, 6(2), 2622–4615. <https://doi.org/10.32493/informatika.v6i2.12104>
- Alom, M. Z., Tarek, M. T., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., van Essen, B., Awwal, A. A. S., & Vijayan, K. A. (2018). *Alom et al.*
- Arrofiqoh, E. N., & Harintaka, H. (2018). IMPLEMENTASI METODE CONVOLUTIONAL NEURAL NETWORK UNTUK KLASIFIKASI TANAMAN PADA CITRA RESOLUSI TINGGI. *GEOMATIKA*, 24(2), 61. <https://doi.org/10.24895/jig.2018.24-2.810>
- Awangga, R. M., & Batubara, N. A. (2020). *Neural networks, perceptrons, connectionism, neural computers*. Kreatif Industri Nusantara.
- Cahyani, T. I., Zakiyamani, M., Riana, D., Hardianti, S., & Magister, P. (2022). PERBANDINGAN AKURASI PENGENALAN KARAKTER PLAT NOMOR MENGGUNAKAN TESSERACT DAN DATA LATIH EMNIST COMPARISON ACCURACY CHARACTER RECOGNITION IN PLATE NUMBER USING TESSERACT AND EMNIST DATASET. *Journal of Information Technology and Computer Science (INTECOMS)*, 5(2).
- Candra, R., Santi, N., Pd, S., & Kom, M. (2011). Mengubah Citra Berwarna Menjadi GrayScale dan Citra biner. *Jurnal Teknologi Informasi DINAMIK*, 16(1), 14–19.

- Darma Nurfitra, R., & Ariyanto, G. (2020). IMPLEMENTASI DEEP LEARNING BERBASIS TENSORFLOW UNTUK PENGENALAN SIDIK JARI. *Jurnal Teknik Elektro* . <http://bias.csr.unibo.it/fvc2004/databases.asp>
- Darmanto, H. (2019). *PENGENALAN SPESIES IKAN BERDASARKAN KONTUR OTOLITH MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK* (Vol. 2, Issue 1).
- Ernawati, S., Wati, R., & Maulana, I. (2021). PENERAPAN MODEL FOUNTAIN UNTUK PENGEMBANGAN APLIKASI TEXT RECOGNITION DAN TEXT TO SPEECH BERBASIS ANDROID MENGGUNAKAN FLUTTER. *Prosiding Seminar Nasional Aplikasi Sains & Teknologi*.
- Fathorazi Nur Fajri, Ahmad Khairi, Suhdil ibadi, & Agung Maulana. (n.d). *Pengenalan Plat Nomor Menggunakan Optical Character Recognition Berbasis Android Untuk Meningkatkan Keamanan Kendaraan Di Universitas Nurul Jadid*. 2, 2021. <https://ejournal.unuja.ac.id/index.php/core>
- Firdaus, A., Syamsu Kurnia, M., Shafera, T., Firdaus, W. I., Teknik, J., Politeknik, K., & Sriwijaya -Palembang, N. (2021). Implementasi Optical Character Recognition (OCR) Pada Masa Pandemi Covid-19. In *Jurnal JUPITER* (Vol. 13, Issue 2).
- Grother, P., Hanaoka, K., & Pritzker, P. (2016). *NIST Special Database 19 Handprinted Forms and Characters 2 nd Edition*.
- Hartanto, S., Sugiharto, A., & Sukmawati, N. E. (2012). Optical Character Recognition menggunakan Algoritma Template Matching Correlation. *Journal*

- of Informatics and Technology*, 1(1), 11–20. <http://ejournal-s1.undip.ac.id/index.php/joint>
- Hudaya, M. M., Siti Saadah, & Hendy Irawan. (2021). Implementation of Verification and Matching E-KTP with Faster R-CNN and ORB. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 5(4), 783–793. <https://doi.org/10.29207/resti.v5i4.3175>
- Ibnutama, K., & Gilang Suryanata, M. (2020). Ekstraksi Karakter Citra Menggunakan Optical Character Recognition Untuk Pencetakan Nomor Kendaraan Pada Struk Parkir. 4, 1119–1125. <https://doi.org/10.30865/mib.v4i4.2432>
- Ilahiyah, S., & Nilogiri, A. (2018). Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network. *Jurnal Sistem & Teknologi Informasi Indonesia*, 3(2502–5724).
- Jawas, N. (2015). BENERISASI CITRA DOKUMEN DENGAN FILTERISASI HOMOMORPHIC. In *Seminar Nasional Informatika*.
- Kingma, D. P., & Ba, J. (2014). *Adam: A Method for Stochastic Optimization*. <http://arxiv.org/abs/1412.6980>
- Kurniawan, D. (2020). *Pengenalan Machine Learning dengan Python* (D. Kurniawan, Ed.; 1st ed.). PT. Elex Media Komputindo.
- Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. In *Nature* (Vol. 521, Issue 7553, pp. 436–444). Nature Publishing Group. <https://doi.org/10.1038/nature14539>

- Margarita Susilo, M., Pola Karakter Bahasa, P., Martomanggolo Wonohadidjojo, D., & Sugianto, N. (2017). Pengenalan Pola Karakter Bahasa Jepang Hiragana Menggunakan 2D Convolutional Neural Network. *28 JUI SI, 03(02)*.
- Meany, C., & Arola, M. (2017). *Optical Character Recognition via Deep Learning*.
- Mirah, S. (2018). *Pengenalan Nomor Induk Kependudukan (NIK) pada e-KTP menggunakan ekstraksi ciri dengan invarian Momen Hu dan Intensity Of Character*.
- Munir, R. (2004). *Pengolahan citra digital dengan pendekatan algoritmik*. Informatika.
- Mustafa, Y. F., Ridho, F., & Mariyah, S. (2022). Study of Handwriting Recognition Implementation in Data Entry of Survei Angkatan Kerja Nasional (SAKERNAS) using CNN. *International Conferene on Data Science and Official Statistic*.
- Pratiwi, H. A., Cahyanti, M., & Lamsani, M. (2021). IMPLEMENTASI DEEP LEARNING FLOWER SCANNER MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK. *Sebatik, 25(1)*.
<https://doi.org/10.46984/sebatik.v25i1.1297>
- Rahmawati, A. N., Wibowo, S. A., & Sunarya, U. (2021). *ANALISIS SISTEM OPTICAL CHARACTER RECOGNITION (OCR) PADA DOKUMEN DIGITAL MENGGUNAKAN METODE TESSERACT PERFORMANCE ANALYSIS OF OPTICAL CHARACTER RECOGNITION (OCR) SYSTEM ON DIGITAL DOCUMENTS USING TESSERACT METHOD*.

- Setiawan, A., Sujaini, H., & Bijaksana PN, A. (2017). Implementasi Optical Character Recognition (OCR) pada Mesin Penerjemah Bahasa Indonesia ke Bahasa Inggris. *Jurnal. Sistem Dan Teknologi Informasi (JUSTIN)*, 5.
- Sugiarta, G., Andini, D. P., & Hidayatullah, S. (2021). Ekstraksi Informasi/Data e-KTP Menggunakan Optical Character Recognition Convolutional Neural Network. *JTERA (Jurnal Teknologi Rekayasa)*, 6(1), 1. <https://doi.org/10.31544/jtera.v6.i1.2021.1-6>
- Suyanto, Kurniawan Nur Ramadhani, & Satria Mandala. (2019). *Deep Learning Modernisasi Mechine Learning untuk Big Data*. Informatika.
- Widhi Saputro, I., & Wulan Sari, B. (2019). Uji Performa Algoritma Naïve Bayes untuk Prediksi Masa Studi Mahasiswa Naïve Bayes Algorithm Performance Test for Student Study Prediction. *Citec Journal*, 6(1).
- Zulhida Putri, D., Setiawan, Y., Supratman, J. W., Limun, K., & Bengkulu, K. (2018). KONVERSI CITRA KARTU NAMA KE TEKS MENGGUNAKAN TEKNIK OCR DAN JARO-WINKLER DISTANCE. *Jurnal TEKNOINFO*, 12(1), 1-6. <http://code.google.com/p/tesseract-ocr>
- Zulkarnain, T., Suharyanto, & Anton. (2020). INTI NUSA MANDIRI PERANCANGAN APLIKASI OPTICAL CHARACTER RECOGNITION BERBASIS BACKPROPAGATION PADA PERANGKAT MOBILE. *INTI NUSA MANDIRI*, 14. <https://doi.org/10.33480/inti.v14i2.1134>

PUSTAKA LAPORAN PENELITIAN

Lecun Y., Bengio Y., & Hinton G., (2015), Deep Learning, Nature, 521(7553), 436-444.

Sekar Mirah, 2018, Pengenalan NIK Pada e-KTP Menggunakan Segmentasi Profil Proyeksi dan Ekstraksi Diri Menggunakan Invarian Momen HU dan Intensity of Character, Skripsi, Program Studi Teknik Informatika, Universitas Sanata Dharma, Yogyakarta.

UU RI No. 24 Tahun 2013 Tentang Perubahan Atas UU No. 23 Tahun 2006 Tentang Administrasi Kependudukan, (2013).

Peraturan Pemerintah No. 40 Tahun 2019 Tentang Administrasi Kependudukan, (2019).

PUSTAKA ELEKTRONIK

Alom, M. Z. et al. 2018, The History Bagan From AlexNet: A Comprehensive Survey on Deep Learning Approaches, Cornell University Library's, <https://arxiv.org/abs/1803.01164>

National Institute Of Standards and Technology. (2019). *The EMNIST Dataset*. <https://www.nist.gov/itl/products-and-services/emnist-dataset>.

Anonim, 2007. Introducing JSON. [Online]. Tersedia : <http://json.org>, (Diakses pada tanggal 07 Oktober 2022)

LAMPIRAN

