**MENDETEKSI TULISAN TANGAN DOKTER DENGAN ALGORITMA *CONVOLUTIONAL NEURAL NETWORK* DAN *RECURRENT NEURAL NETWORK* BERBASIS *DESKTOP***

**SKRIPSI**

Sebagai Salah Satu Syarat untuk Meraih

Gelar Sarjana Komputer

Oleh:

Raka Rasell

32160036



Fakultas Teknologi Dan Desain

Program Studi Teknik Informatika

Universitas Bunda Mulia

Jakarta

2020

|  |
| --- |
| **UNIVERSITAS BUNDA MULIA**  **FAKULTAS TEKNOLOGI DAN DESAIN**  **PROGRAM STUDI TEKNIK INFORMATIKA** |
| Pernyataan Kesiapan Ujian Pendadaran Skripsi  Saya Raka Rasell, dengan ini menyatakan bahwa Skripsi yang berjudul :  **MENDETEKSI TULISAN TANGAN DOKTER DENGAN ALGORITMA *CONVOLUTIONAL NEURAL NETWORK* DAN *RECURRENT NEURAL NETWORK* BERBASIS *DESKTOP***  Merupakan hasil karya saya dan belum pernah diajukan sebagai karya ilmiah, sebagian seluruhnya, atas nama saya atau pihak lain  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Raka Rasell  32160036  Disetujui oleh Pembimbing,  Kami setuju Skripsi tersebut diajukan untuk Ujian Pendadaran  \_\_\_\_\_\_\_\_\_\_\_\_\_\_  Halim Agung, S.Kom, M.Kom 1 Januari 2020  Disetujui oleh Ketua Program Studi,  \_\_\_\_\_\_\_\_\_\_\_\_\_\_  Lukman Hakim, ST., M.Kom 1 Januari 2020 |

|  |
| --- |
| **UNIVERSITAS BUNDA MULIA**  **FAKULTAS TEKNOLOGI DAN DESAIN**  **PROGRAM STUDI TEKNIK INFORMATIKA** |
| **Persetujuan Skripsi**  Yang bertanda tangan di bawah ini, menyatakan bahwa Skripsi dengan judul  **MENDETEKSI TULISAN TANGAN DOKTER DENGAN ALGORITMA *CONVOLUTIONAL NEURAL NETWORK* DAN *RECURRENT NEURAL NETWORK* BERBASIS *WEBSITE***  Disusun oleh :  Raka Rasell 32160036  Telah disetujui dan diterima sebagai salah satu karya ilmiah mahasiswa yang bersangkutan pada Fakultas Teknologi dan Desain – Program Studi Teknik Informatika Universitas Bunda Mulia  Jakarta, 1 Januari 2020  Mengetahui  Ketua Program Studi Dosen Pembimbing  (Lukman Hakim, ST., M.Kom) (Halim Agung, S.Kom, M.Kom) |

**PERNYATAAN**

Saya menyatakan bahwa Skripsi yang berjudul **“MENDETEKSI TULISAN TANGAN DOKTER DENGAN ALGORITMA *CONVOLUTIONAL NEURAL NETWORK* DAN *RECURRENT NEURAL NETWORK* BERBASIS *DESKTOP*”**, sepenuhnya karya saya sendiri. Tidak ada bagian di dalamnya yang merupakan plagiat dari karya orang lain dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan. Atas pernyataan ini, saya siap menanggung resiko / sanksi yang dijatuhkan kepada saya, atau klaim dari pihak lain terhadap keaslian karya saya ini.

Jakarta, 1 Januari 2020

Yang membuat pernyataan

Raka Rasell

# ABSTRAK

Membaca resep dokter terkadang menyulitkan untuk seorang pasien. Kesalahan membaca resep terkadang juga terjadi oleh seorang apoteker yang dapat menyebabkan hilangnya nyawa seorang pasien.

Deteksi tulisan adalah salah satu cabang dari kecerdasan buatan. Deteksi tulisan sudah banyak diimplementasikan diberbagai bidang salah satunya yang populer di Indonesia adalah mendeteksi plat nomor kendaraan. Salah satu implementasi dari deteksi tulisan adalah, mendeteksi tulisan tangan seorang dokter.

Untuk mendeteksi sebuah tulisan tangan dibutuhkan juga dataset yang mendukung untuk training, salah satu dataset yang mendukung adalah IAM Dataset. Penggunaan dataset ini juga mempercepat proses dari implementasi algoritma *Convolutional Neural Network* dan *Recurrent Neural Network.*

Penggunaan algoritma tersebut dikarenakan input yang diterima adalah citra yang berupa tulisan tangan. Dari penerapan algoritma tersebut didapat akurasi sebesar 26% dengan pengujian sebanyak 30 kali dan berbagai jenis tulisan tangan, terutama tulisan pada resep dokter.

**Kata Kunci :**

*Computer Vision, Neural Network, Object Detection*

# PRAKATA

# DAFTAR ISI

# BAB 1 PENDAHULUAN

## Latar Belakang

Tulisan tangan dokter yang kurang dapat dibaca sangat menyulitkan sehingga berpotensi menimbulkan kesalahan dalam membaca nama obat, dosis, aturan pakai, dan cara pemberian yang selanjutnnya dapat menyebabkan kesalahan pengobatan. Kesalahan dalam membaca tulisan tangan dokter disebut *medication error.* *Medication Error* dapat membahayakan pasien dan memaparkan tenaga kesehatan pada tuntukan kriminal [5][7].

*Convolutional Neural Network (CNN)* dan *Recurrent Neural Network* *(RNN)* merupakan metode yang banyak digunakan untuk mendeteksi dan klasifikasi objek berupa gambar. Keduanya dimodelkan berdasarkan jaringan syaraf biologi manusia untuk mendapatkan informasi dari data yang rumit dan memecahkan masalah dengan sendirinya berdasarkan informasi yang melewati jaringan[16]. Untuk mendapatkan informasi maka diperlukan proses *training* yaitu mempelajari fitur dari data yang sudah ada.

Berdasarkan latar belakang yang terjadi, maka dibuat skripsi dengan judul “**Mendeteksi Tulisan Tangan Dokter Dengan Menggunakan Algoritma *Convolutional Neural Network* dan *Recurrent Neural Network* Berbasis *Website***”. Dengan tujuan agar aplikasi berikut dapat bermanfaat di dunia kesehatan dan memudahkan seseorang membaca tulisan tangan dokter.

## Rumusan Masalah

Berdasarkan latar belakang skripsi maka didapatkan suatu rumusan masalah yaitu :

1. Bagaimana mendeteksi tulisan dokter dengan algoritma *CNN* dan *RNN*?
2. Seberapa besar tingkat akurasi menggunakan *CNN* dan *RNN*

## 1.3 Tujuan dan Manfaat

### Tujuan

Tujuan yang didapatkan dalam skripsi ini adalah:

* + - 1. Menerapkan algoritma *CNN* dan *RNN* pada deteksi tulisan tangan dokter.
      2. Mendapatkan nilai akurasi untuk deteksi tulisan tangan dokter dengan menggunakan algoritma *CNN* dan *RNN.*

### Manfaat

Manfaat yang didapatkan dalam skripsi ini adalah sebagai berikut:

1. Memudahkan membaca tulisan tangan dokter.

## Ruang Lingkup

Ruang lingkup yang ada pada skripsi ini adalah sebagai berikut:

1. Deteksi tulisan tangan hanya dilakukan per kata.
2. *Dataset* yang digunakan kurang dari *1GB (1024MB)*.

## Metodologi Penelitian

### Teknik Pengumpulan *Data*

*Data* yang digunakan dalam skripsi ini didapatkan dari gambar dan *text* dengan format *.png* dan .*txt* yang bersifat *open source.*

### Teknik Perancangan Aplikasi

Perancangan aplikasi dibuat dengan *Visual Studio Code* sebagai aplikasi perancang utama, dan juga Algoritma *CNN* dan *RNN* dan juga menggunakan *Connectionist Temporal Classification* *(CTC)* untuk *decode text.*

### Pemodelan Sistem

Pemodelan sistem pada aplikasi informasi ini menggunakan *UML* karena pemograman ini bersifat berorientasi objek.

## Sistematika Penulisan

### BAB 1 PENDAHULUAN

Bab 1 berisi latar belakang masalah, rumusan masalah, tujuan, manfaat, ruang lingkup, metodologi penelitian, dan sistematika penulisan.

### BAB 2 LANDASAN TEORI

Bab 2 berisi teori yang berupa pengertian dan definisi yang diambil dari sumber terpercaya untuk menjadi dasar untuk digunakan dalam menyelesaikan permasalahan yang sudah dijelaskan pada bab sebelumnya.

### BAB 3 ANALISIS DAN PERANCANGAN

Bab 3 berisi tentang analisis kebutuhan yang diperlukan dalam melakukan skripsi serta rancangan sistem yang dipresentasikan menggunakan *flowchart* atau *UML*.

### BAB 4 IMPLEMENTASI

Bab 4 berisi penerapan teori kedalam bahasa pemrograman dan pengujian terhadap pengklasifikasian yang akan dilakukan.

### BAB 5 SIMPULAN DAN SARAN

Bab 5 berisi tentang simpulan dari skripsi yang dilakukan dan saran dalam melakukan skripsi terkait.

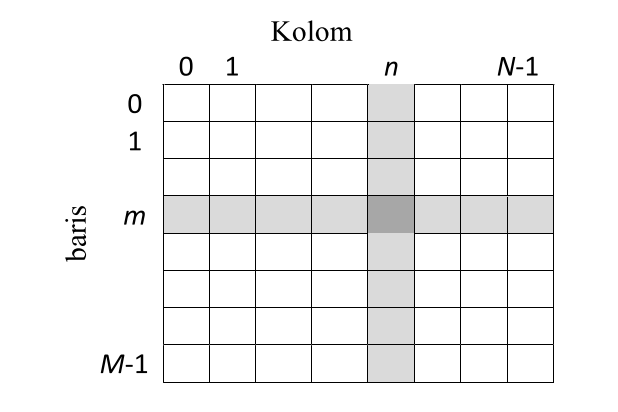
# **BAB 2** LANDASAN TEORI

## *Computer Vision*

Computer vision adalah sebuah kemampuan komputer yang didesain agar mampu melihat sebuah objek sehingga mampu menampilkan objek digital dan mengoleksi data secara visual. Computer Vision mampu mengevaluasi sebuah data yang bersumber dari gambar dengan teknik menggunakan algoritma tertentu. Agar mampu melihat layaknya manusia, maka computer vision harus memiliki fungsi pendukung seperti, proses penangkapan citra, proses pengolahan citra, dan proses pemahaman citra. [13].

## *Image Processing*

Pengolahan citra merupakan sebuah ilmu yang mempelajari Teknik mengolah citra. Citra digital dapat diwakili oleh sebuah matriks dua dimensi yang terdiri dari kolom dan baris, dimana perpotongan kolom dan baris disebut *pixel* atau elemen terkecil dari sebuah citra.[14]



Gambar 2. 1 *Layer* Pada *Convolutional Neural Network*

## Segmentasi Citra

Segmentasi citra merupakan proses untuk memisahkan antara objek satu dengan yang lain sehingga dapat digunakan sebagai input pada proses lain. Secara sederhana teknik yang sering digunakan untuk memisahkan objek adalah thresholding. Thresholding merupakan sebuah teknik binerisasi yang prosesnya dilakukan berdasarkan nilai batas ambang. Nilai ini dapat berupa informasi RGB, HSV, dari suatu pixel. Namun pada umumnya nilai yang sering digunakan adalah tingkat grayscale. Suatu pixel dengan tingkat grayscale yang tinggi dapat dikategorikan sebagai objek dan pixel yang nilainya rendah dapat disebut sebagai background. Secara matematis, metode ini dapat dituliskan dengan persamaan:

Gambar 2. 2 *Layer* Pada *Convolutional Neural Network*

g(x,y) adalah citra biner dari citra grayscale f(x,y), dan T menyatakan nilai threshold.

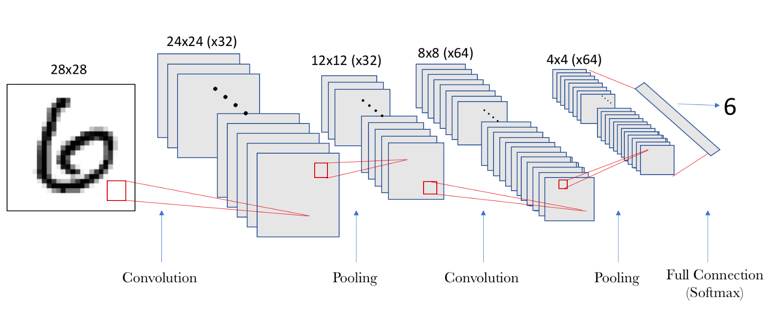
## *Deep Learning*

*Deep Learning* adalah cabang ilmu *machine learning* berbasis *Neural Network.* Dalam *deep learning*, sebuah komputer belajar mengklasifikasi secara langsung dari gambar atau suara. *Convolutional Neural Network (CNN)* adalah salah satu algoritma *deep learning* yang merupakan pengembangan dari *Multilayer Perceptron (MPL)* yang dirancang untuk mengolah *data* dalam bentuk dua dimensi, misalnya gambar. *CNN* dapat belajar langsung dari citra sehingga mengurangi beban dari pemrograman .

Selain itu, metode *machine learning* konvensional hanya mengandalkan *CPU* dan *RAM* dalam proses komputasi, sehingga spesifikasi *CPU* dan *RAM* menentukan kecepatan komputasi. Sedangkan metode *deep learning*, selain menggunakan *CPU* dan *RAM* dalam proses komputasi, metode ini juga memanfaatkan kemampuan *GPU* sehingga proses komputasi *data* yang besar dapat berlangsung lebih cepat [9].

## *Convolution Neural Network (CNN)*

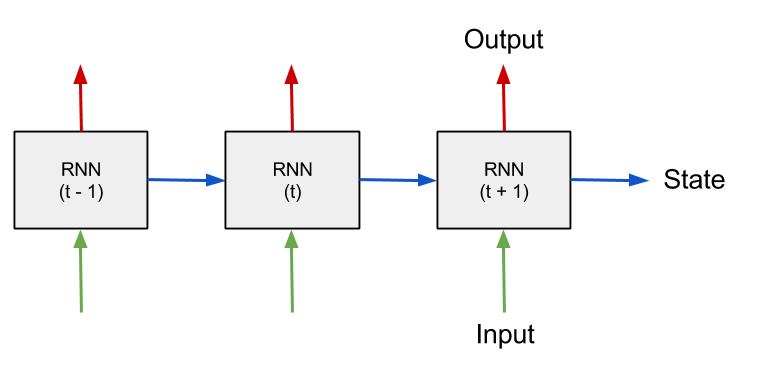
*Convolutional Neural Network* yang juga dikenal dengan *CNN* adalah pengembangan dari *Multilayer Perceptron (MLP)* untuk mengolah *data* dua dimensi. *Multilayer Perceptron* sendiri adalah pengembangan dari *Artificial Neural Network (ANN)* yang ditujukan untuk menutupi kekurangan *Artificial Neural Network (ANN)* dengan *Single-layer Perceptron* dalam menyelesaikan operasi logika yang kompleks. Hal tersebut dimungkinkan dengan menambahkan *hidden layer* yang membuat *ANN* kuat untuk memecahkan operasi logika yang kompleks *(universal approximation)* dan sering digunakan untuk permasalahan-permasalahan klasifikasi, *recognition*, dan prediksi [11].



Gambar 2. 3 *Layer* Pada *Convolutional Neural Network*

## *Recurrent Neural Network (RNN)*

*Recurrent Neural Network* memiliki sedikit perbedaann dibandingkan *Neural Network* biasa. Jika pada Neural Network, setiap input akan masuk ke hidden layer dan mengeluarkan output, sedangkan Recurrent Neural Network terdapat perulangan. Setiap input yang masuk dan menghasilkan output akan masuk kembali menjadi input untuk diproses di dalam hidden layer. Begitu seterusnya hingga memperoleh output yang sesuai dengan target. Jenis Recurrent Neural Network yang digunakan adalah *Long Short Term Memory (LSTM)* yang mampu mengelola informasi lebih akurat. *LSTM* mampu mempertahankan error yang terjadi ketika melakukan backpropagation sehingga tidak memungkinkan kesalahan meningkat.



Gambar 2. 4 *Layer* Pada *Convolutional Neural Network*

## *Python*

*Python* merupakan bahasa pemrograman yang berorientasi obyek dinamis, dapat digunakan untuk bermacam macam pengembangan perangkat lunak. *Python* menyediakan dukungan yang kuat untuk integrasi dengan bahasa pemrograman lain dan alat-alat bantu lainnya. *Python* hadir dengan pustaka-pustaka standar yang dapat diperluas serta dapat dipelajari hanya dalam beberapa hari. Bahasa pemrograman yang interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode [5].

*Python* diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif [5].

## *Visual Studio Code*

*Visual Studio Code* atau disebut *VS Code* adalah sebuah *text editor* ringan dan handal yang dibuat oleh *Microsoft* untuk sistem operasi *multiplatform*, artinya tersedia juga untuk versi *Linux*, *Mac*, dan *Windows*. *Text editor* ini secara langsung mendukung bahasa pemrograman *JavaScript*, *Typescript*, dan *Node.js*, serta bahasa pemrograman lainnya dengan bantuan *plugin* yang dapat dipasang di *marketplace Visual Studio Code* (seperti *C++*, *C#*, *Python*, *Go*, *Java*, dst).

Banyak sekali fitur-fitur yang disediakan oleh *Visual Studio Code*, diantaranya *Intellisense*, *Git Integration*, *Debugging*, dan fitur ekstensi yang menambah kemampuan *text editor*. Fitur-fitur tersebut akan terus bertambah seiring dengan bertambahnya versi *Visual Studio Code*. Pembaruan versi *Visual Studio Code* ini juga dilakukan berkala setiap bulan, dan inilah yang membedakan *VS Code* dengan *text editor* yang lain.

## *Tkinter*

Ada banyak *library* untuk untuk membuat user interface pada python. Salah satu yang digunakan untuk aplikasi ini adalah Tkinter. Tkinter adalah library untuk membuat user interface yang sudah tersedia langsung pada saat instalasi python. Tkinter sangat mudah dan cepat dibandingkan beberapa library lainnya.

## *Command Line Interface (CLI)*

*CLI* adalah tipe antarmuka dimana pengguna berinteraksi dengan sistem operasi melalui *text-terminal*. Pengguna menjalankan perintah dan *program* di sistem operasi tersebut dengan cara mengetikkan baris-baris tertentu [8].

Meskipun konsepnya sama, tiap-tiap sistem operasi memiliki nama atau istilah yang berbeda untuk *CLI*-nya. *UNIX* memberi nama *CLI*-nya sebagai *bash*, *ash*, *ksh*, dan lain sebagainya. *Microsoft Disk Operating System (MS-DOS)* memberi nama *command.com* atau *Command Prompt*. Sedangkan pada *Windows Vista*, *Microsoft* menamakannya *PowerShell*. Pengguna *Linux* mengenal *CLI* pada *Linux* sebagai *terminal*, sedangkan pada *Apple* adalah *commandshell* [8].

## *Unified Modeling Language (UML)*

*Unified Modeling Language* *(UML)* bukanlah suatu proses melainkan bahasa pemodelan secara grafis untuk menspesifikasikan, memvisualisasikan, membangun, dan mendokumentasikan seluruh artifak sistem perangkat lunak. Penggunaan model ini bertujuan untuk mengidentifikasikan bagian-bagian yang termasuk dalam lingkup sistem yang dibahas dan bagaimana hubungan antara sistem dengan subsistem maupun sistem lain di luarnya.

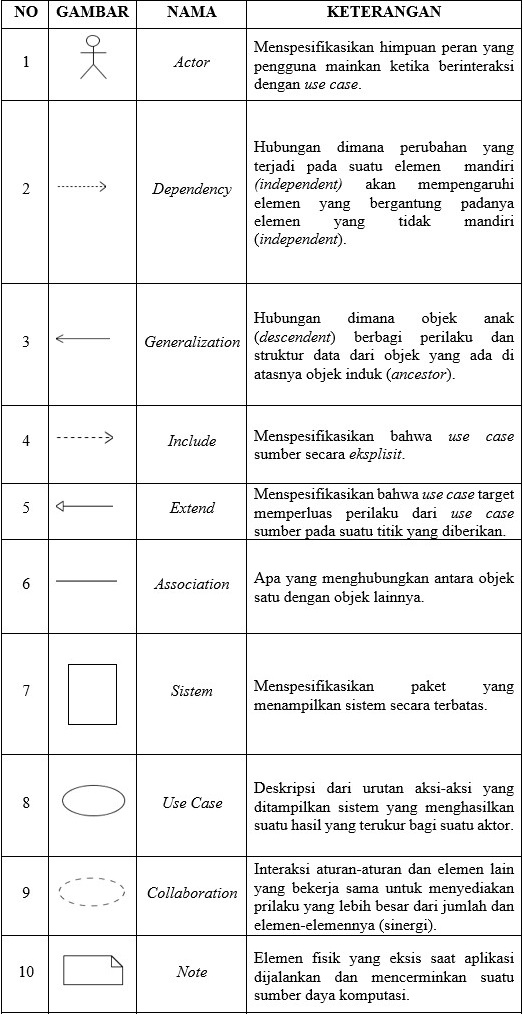
*Unified Modeling Language (UML)* adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan dari sebuah sistem pengembangan *software* berbasis *object oriented*.

Dari Pengertian diatas penulis menyimpulkan bahwa *Unified Modeling Language (UML)* merupakan bahasa pemodelan yang berbentuk grafis yang digunakan untuk memvisualisasi, menspesifikasikan suatu sistem perangkat lunak.

### *Use Case Diagram*

*Use Case Diagram* adalah sesuatu atau proses merepresentasikan hal-hal yang dapat dilakukan oleh aktor dalam menyelesaikan sebuah pekerjaan. Menurut *Shalahuddin* dalam jurnal *(Umar Al Faruq, 2015)* mengungkapkan : “Diagram *use case* merupakan pemodelan untuk kelakuan *(behavior)* sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut” [7].

Berikut adalah simbol-simbol yang ada pada *Use Case Diagram* :



Gambar 2. 2 *Use Case Diagram Symbols*

### *Activity Diagram*

Menurut *Rosa* dalam jurnal *(Sari dan David)* mengungkapkan : “*Activity Diagram* menggambarkan *work flow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas dapat dilakukan oleh sistem” [7].

Simbol-simbol yang digunakan dalam *activity diagram* sebagai berikut :

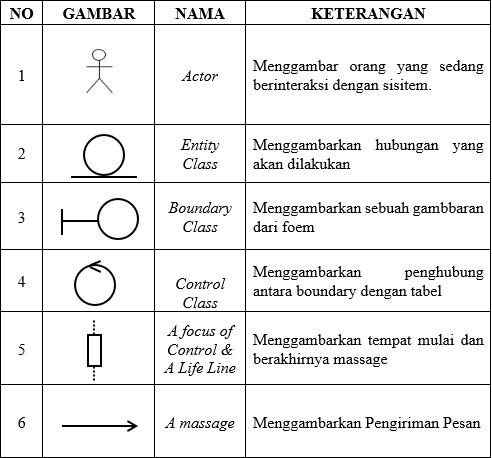


Gambar 2. 3 *Activity Diagram Symbols*

### *Sequence Diagram*

*Sequence Diagram* adalah *tool* yang sangat populer dalam pengembangan sistem informasi secara *object-oriented* untuk menampilkan interaksi antar objek. Berdasarkan definisi tersebut, dapat disimpulkan bahwa *Sequence Diagram* adalah *tool* yang digunakan dalam pengembangan sistem [7].

Berikut adalah simbol-simbol yang digunakan dalam *Sequence diagram* adalah :



Gambar 2. 4 *Sequence Diagram Symbols*

## Penelitian Terdahulu

Pada penelitian terdahulu, dapat disimpulkan bahwa:

Table 2. 1 Penelitian Terdahulu

|  |  |  |  |
| --- | --- | --- | --- |
| No | Penulis | Judul | Kesimpulan |
| 1 | Ivanedra, Kasyfi, Mustikasari, Metty | Implementasi Metode Recurrent Neural Network Pada Text Summarization Dengan Teknik Abstraktif | Perbedaan jumlah dataset yang digunakan sangat berpengaruh, karena pembendaharaan kata yang lebih banyak tentunya menjadikan program lebih akurat. |
| 2 | Soselia, Davit, Amashukeli Shota, Irakli Koberidze, Levan Shugliashvili | *﻿RNN-based Online Handwritten Character Recognition Using Accelerometer and Gyroscope Data* | RNN berhasil digunakan untuk menemukan pola karakter yang unik di antara berbagai gaya tulisan tangan dari beberapa individu. Deteksi tulisan tanga menggunakan perangkat ini menghasilkan akurasi sebesar 88%. |
| 3 | ﻿R. Reeve Ingle, Yasuhisa Fujii, Thomas Deselaers, Jonathan Baccash, Ashok C. Popa | ﻿*A Scalable Handwritten Text Recognition System* | Dari dataset yang diperbaharui maka menghasilkan model yang lebih baik dan juga *low cost.* |
| 4 | ﻿Dina Tri Amalia, Asep Sukohar | ﻿*Rational Drug Prescription Writing* | ﻿Beberapa faktor yang mempengaruhi buruknya penulisan resep salah satunya kurangnya pengetahuan akan standard format penulisan resep. Format penulian resep yang berlaku di Indonesia terdiri dari  *inscriptio, invocatio,*  *prescriptio, signatura, subscriptio* dan *pro.* |
| 5 | ﻿Chammas, Edgard  Mokbel, Chafic  Likforman-Sulem, Laurence | ﻿*Handwriting recognition of historical documents with few labeled data* | - |
| 6 | ﻿Fadhilla, Mutia  Saf, Maksum Ro’is Adin  Sahid, Dadang Syarif Sihabudin | *﻿*Pengenalan Kepribadian Seseorang Berdasarkan Pola Tulisan Tangan Menggunakan Jaringan Saraf Tiruan | Dapat disimpulkan bahwa metode LVQ memiliki tingkat akurasi sebesar 90% dibandingkan dengan *back propagation* sebesar 82%. |
| 7 | ﻿Bluche, Théodore | ﻿*Joint line segmentation and transcription for end-to-end handwritten paragraph recognition* | Disarankan untuk mengubah sedikit arsitektur MDLSTM agar berfokus pada 1 baris segmentasi. |
| 8 | ﻿E.Kamalanaban, Dr  Gopinath, M  Premkumar, S | ﻿*Medicine Box: Doctor’s Prescription Recognition Using Deep Machine Learning* | Dengan penggunaan RNN, maka karakter dari tulisan dapat terprediksi. Kunci dari prediksi kata adalah training data yang panjang. |
| 9 | ﻿Such, Felipe Petroski  Peri, Dheeraj  Brockler, Frank  Hutkowski, Paul  Ptucha, Raymond | ﻿*Fully Convolutional Networks for Handwriting Recognition* | Berbeda dengann metode lexicon, neural network yang dibangun dapat membaca sebuah simbol seperti *underline, infinite*, dan lainnya. |
| 10 | ﻿Achkar, Roger  Ghayad, Khodor  Haidar, Rayan  Saleh, Sawsan  Al Hajj, Rana | *Medical Handwritten Prescription Recognition Using CRNN* | Algoritma butuh mempelajari sebuah paragraph dan mengubahnya ke bentuk teks kecil. Akurasi yang didapat sebesar 95% dengan mempertimbangkan dataset dan waktu *training.* |
| 11 | ﻿Yuwitaning, Eka Farda  Hidayat, Bambang  Andini, Nur | ﻿Implementasi Metode *Hidden Markov Model* Untuk Deteksi Tulisan Tangan. | ﻿Berdasarkan pengujian yang dilakukan pada penelitian ini, penggunaan MDF untuk ekstraksi ciri dan digabungkan dengan HMM sebagai metode klasifikasinya dapat digunakan untuk melakukan pengenalan karakter huruf dan angka tulisan tangan dengan rata-rata CER 25,28% atau rata-rata akurasi karakter 74,72% |
| 12 | ﻿Arrahman  Purwanto  Nurtantio, Pulung | Klasifikasi Nama Obat Tulisan Tangan Dokter Dengan Metode GLCM dan *Backpropagation Neural Network* | ﻿Backpropagation Neural Network secara umum dapat memberikan nilai akurasi tertingi dibanding dengan Naive Bayes dan C 4.5 |

# **BAB 3** ANALISIS DAN PERANCANGAN

## Analisis Kebutuhan Fungsional dan Non Fungsional

### Kebutuhan Fungsional

Kebutuhan Fungsional merupakan rangkaian kebutuhan yang memiliki keterkaitan dengan sistem. Kebutuhan fungsional dari aplikasi yang dirancang ini adalah sebagai berikut:

1. Aplikasi dapat mendeteksi tulisan tangan dokter.
2. Aplikasi dapat melakukan training dari hasil training sebelumnya supaya tingkat akurasi deteksi meningkat.

### Kebutuhan Non-Fungsional

Kebutuhan non fungsional merupakan kebutuhan yang secara tidak langsung terkait dengan sistem.

1. Perangkat keras (*hardware*)

Perangkat keras (*hardware*) yang digunakan dalam perancangan aplikasi ini adalah laptop dengan spesifikasi:

1. *Operating System* : *MacOS Catalina Version 10.15.4*
2. *Processor*  : *Intel® Core™ i5 Skylake 6267U*
3. *Memory*  : *8 GB*
4. *VGA* : *Intel Iris Graphics 550*
5. Perangkat lunak (*software*)

Perangkat lunak (*software*) yang digunakan dalam perancangan aplikasi ini adalah:

1. *Visual Studio Code*
2. *Terminal*
3. *Brainware*

*Brainware* merupakan setiap orang yang memiliki keterlibatan dalam perancangan maupun penggunaan aplikasi. Untuk penggunaan aplikasi, *brainware* tidak memerlukan keahlian khusus namun setidaknya mengerti bagaimana menggunakan aplikasi berbasis *website*.

1. Bahasa Pemograman

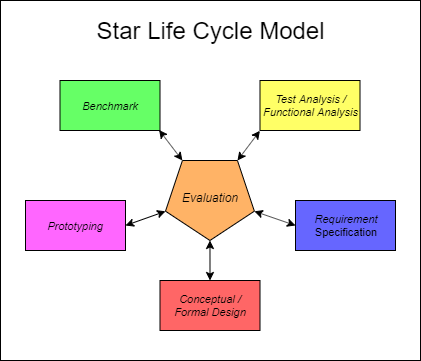
Bahasa Pemograman yang digunakan pada pembuatan *program* deteksi tulisan tangan dokter adalah *Python* versi 3.6. *Module* *python* yang digunakan dalam paket pembuatan *program* ini adalah:

|  |  |
| --- | --- |
| **Modul** | **Modul** |
| Werkzeug | six |
| opencv-python | numpy |
| h5py | grpcio |
| editdistance | astor |
| absl-py |  |

Gambar 3. 1 *Module Python*

## Pemilihan Algoritma/Teknik/Metode

### Model Perancangan Sistem



Gambar 3. 2 *Star Life Cycle Model*

Model perancangan sistem yang digunakan dalam membangun atau merancang aplikasi ini adalah *Star Lifecycle Model*. Penggunaan model *Star Lifecycle* ini dikarenakan pengujian terhadap struktur *program* dilakukan pada banyak faktor sehingga proses pengujian dilakukan secara terus menerus dan banyak dalam satu waktu. Dengan menerapkan model *Star Lifecycle* ini juga menghemat waktu karena tidak perlu menunggu tahapan awal selesai untuk dapat melakukan tahapan selanjutnya sehingga tidak ada hirarki dalam perancangan.

Berikut proses-proses yang ada dan dilakukan pada perancangan sistem dengan *Star Lifecycle Model* ini:

1. *Task Analysis or Functional Analysis*

*Task Analysis* atau *Functional Analysis* adalah identifikasi kemampuan *user*, strategi yang digunakan untuk meningkatkan keterampilannya, alat yang saat ini dipakai, masalah-masalah yang dialami, perubahan yang diinginkan baik dalam ketrampilan maupun peralatan. Metode yang dapat dilakukan adalah dengan mengetahui kemampuan *user*, membuat daftar dengan skala prioritas, dan observasi keterampilan di lapangan.

1. *Requirement Specification*

*Requirement Specification* adalah menentukan kekuatan dan kelemahan suatu spesifikasi yang ada. Metode ini dapat dilakukan dengan meminta *user* untuk mencoba menggunakan berbagai produk yang ada dalam spesifikasi lalu memberitahukan apa kelebihan dan kelemahan dari masing-masing produk.

1. *Conceptual or Formal Design*

*Conceptual or Formal Design* adalah menggunakan hasil analisa untuk membuat alternatif solusi dan saran sampai dengan penentuan pilihan yang terbaik. Metode yang dapat dilakukan adalah dengan bertanya pada *user* sehubungan dengan pengalaman *user* dalam menggunakan prototipe-prototipe yang telah dibuat.

1. Evaluasi dan Validasi

Evaluasi dan Validasi merupakan pemberian masukan kepada *user* secara periodik selama pengembangan dan perancangan dan akan dilakukan terus menerus berdasarkan masukan tadi. Metode yang dapat ditempuh adalah dengan mengamati kebutuhan pokok *user* dalam menggunakan sistem.

1. *Benchmark*

Memadukan hal-hal terbaik yang dimiliki pesaing untuk diterapkan dalam sistem yang dibangun. Metode yang dapat dilakukan adalah dengan menggali informasi dari *user* hal-hal yang sebaiknya ada dan dibandingkan dengan kompetitor.

### Pemilihan Algoritma / Metode

Algoritma yang digunakan pada penulisan skripsi ini adalah

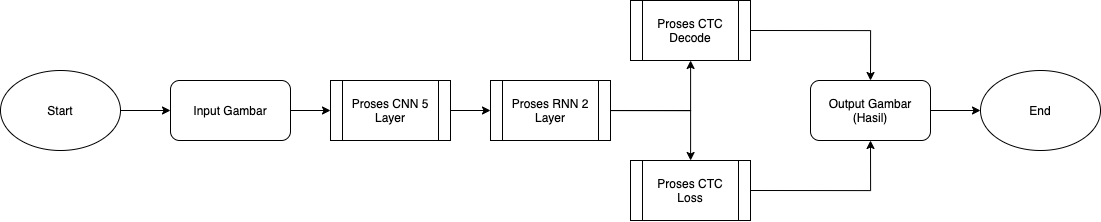
1. *Convolutional Neural Network*
2. *Recurrent Neural Network*

## Pemilihan Algoritma/Teknik/Metode

### *Flowchart Diagram*

#### **Flowchart Deteksi Tulisan Tangan Dokter**

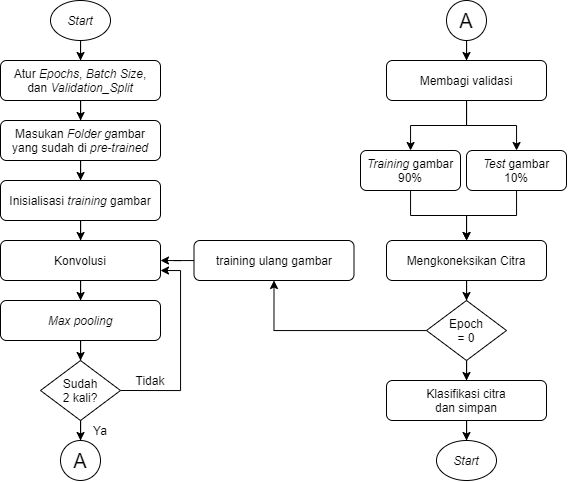
Perancangan *flowchart* *diagram* berikut bertujuan untuk menggambarkan bagaimana aliran proses deteksi tulisan tangan dokter. *Flowchart diagram* dari aplikasi ini ditunjukkan pada Gambar 3.3.



Gambar 3. 3 *Flowchart* Deteksi Tulisan Tangan Dokter

#### **Flowchart Training Gambar**

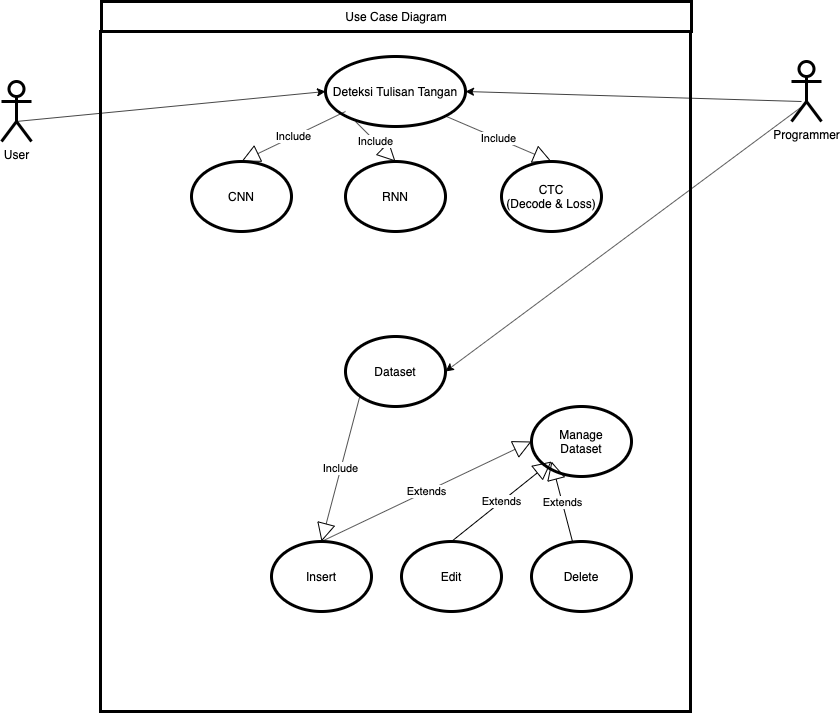
Perancangan *flowchart* *diagram* berikut bertujuan untuk menggambarkan bagaimana sistem melakukan pelatihan gambar. *Flowchart diagram* dari aplikasi ini ditunjukkan pada Gambar 3.4.



Gambar 3. 4 *Flowchart* *Data Training*

### *Use Case Diagram*

*Use* *case* *diagram* berikut mengambarkan apa saja peranan dari *user* dan *programmer* :

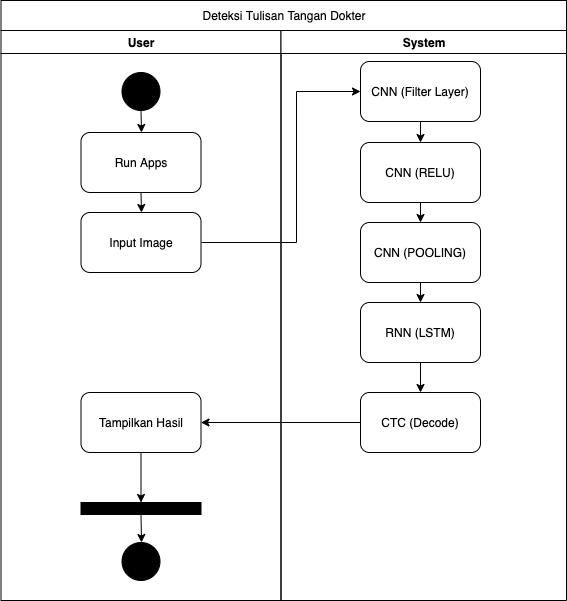


Gambar 3. 5 *Use Case Diagram*

Seperti yang terlihat pada Gambar 3.4, terdapat 2 aktor yaitu *user* dan *programmer* yang berperan aktif dalam *use-case diagram* diatas. *User* disini hanya dapat melakukan deteksi usia yang didalamnya meliputi Memperkecil Citra*, Thresholding,* dan *Classifing*. Sedangkan *programmer* dapat melakukan 3 hal yaitu, deteksi usia seperti *user*, melatih *data,* dan mengatur *database*. *Programmer* dalam melatih *data* dapat memasukan *data* baru ke *database*, dan dalam mengatur *database* dapat memasukan *data* secara manual, merevisi *data*, dan menghapus *data*.

### *Activity Diagram*

Perancangan *activity* *diagram* ini bertujuan untuk menggambarkan aktivitas aktivitas yang terjadi di dalam sistem/aplikasi secara sistematis.

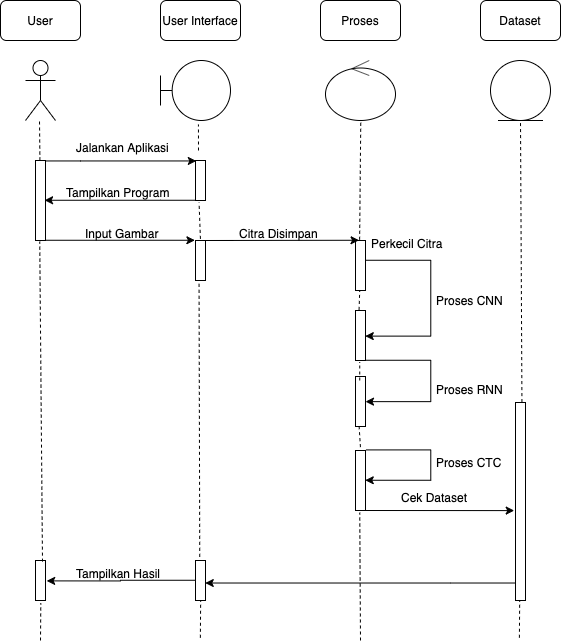


Gambar 3. 6 *Activity Diagram:* Deteksi Tulisan Tangan Dokter

Aktivitas yang dilakukan oleh *user* dengan sistem dalam penggunaan aplikasi digambarkan dengan *activity diagram* yang ditunjukkan pada Gambar 3.6.

1. Saat deteksi, pertama *user* harus menjalankan aplikasi terlebih dahulu.
2. Setelah aplikasi telah dibuka, selanjutnya *user* memilih file input gambar.
3. Citra kemudian akan direkam oleh sistem dan di *preprocessing* terlebih dulu,
4. Citra selanjutnya akan di perkecil untuk mempercepat proses perhitungan.
5. Setelah diperkecil, maka akan menghasilkan output berupa *features map.*
6. *Features Map* akan diproses oleh *RNN* menggunakan *LSTM* untuk mendapatkan informasi yang lebih banyak lagi. *Output RNN* dipetakan dalam bentuk matriks.
7. Selanjutnya *CTC* akan memberikan informasi *loss* dan juga hasil dari *text* tersebut.
8. Proses deteksi selesai.

### *Sequence Diagram*



Gambar 3. 7 *Sequence Diagram:* Deteksi Usia

*Sequence* *diagram* pada Gambar 3.6 menggambarkan proses aplikasi yang dilakukan oleh *user* dalam mendeteksi usia dimana terdapat 4 objek yang saling berkaitan yaitu *user* sebagai aktor, *program interface* sebagai *object boundary*, *calculation process* sebagai *object control*, dan age *database* sebagai *object entity*.

### *Gantt Chart*



Gambar 3. 8 *Gantt Chart*



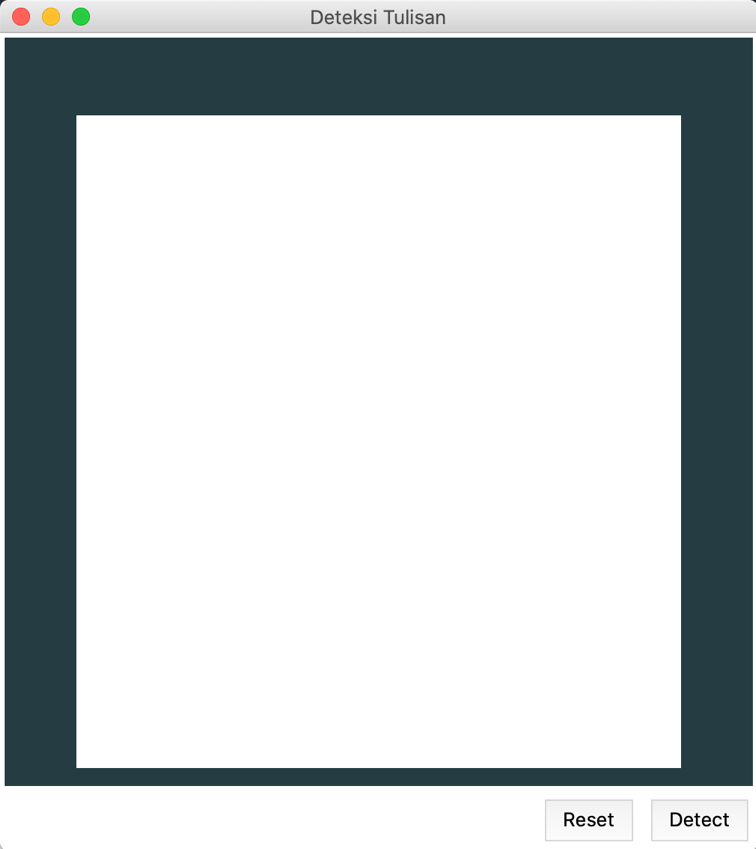
Penjadwalan dalam melakukan segala aktivitas sangatlah menjadi perhatian penting. Jika penjadwalan tidak dilakukan dengan baik, maka hasil yang akan didapatkan tidak akan sesuai dengan harapan. Oleh karena itu kami membuat penjadwalan dalam bentuk *Gantt* *Chart* untuk pencarangan aplikasi berikut.

**BAB 4  
IMPLEMENTASI**

**4.1 Implementasi Antarmuka Pengguna**

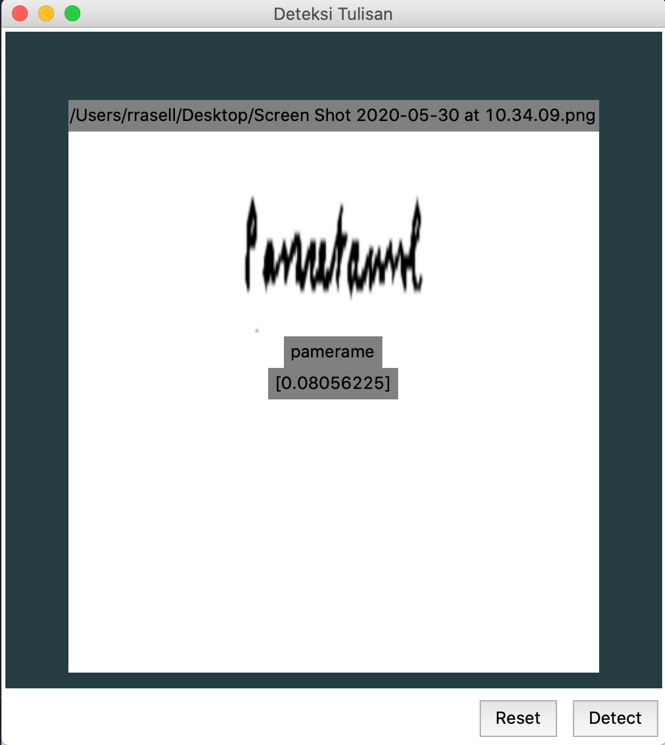
Tampilan antarmuka menggunakan library dari Tkinter untuk mempercepat proses pembuatan, tampilan seperti pada gambar 4.1. Aplikasi memiliki fungsi seperti Open File, Detect, dan juga Reset. Berikut ini adalah alur untuk mendeteksi tulisan tangan dokter:

1. Open File: memilih file berekstensi .png, .jpg, yang akan di detect.
2. Detect: setelah memilih, maka file gambar akan di detect dan keluar hasilnya.
3. Reset: berfungsi untuk mendeteksi ulang gambar lain yang ingin dideteksi.



Gambar 4. 1 Tampilan Aplikasi.

Hasil dari deteksi gambar seperti pada gambar 4.2 yang dimana hasil ditampilkan berupa tulisan yang terdeteksi dan juga probabilitas dari keberhasilan. Probabilitas ditentukan dari hasil training dan sumber data training.



Gambar 4. 2 Tampilan hasil deteksi.

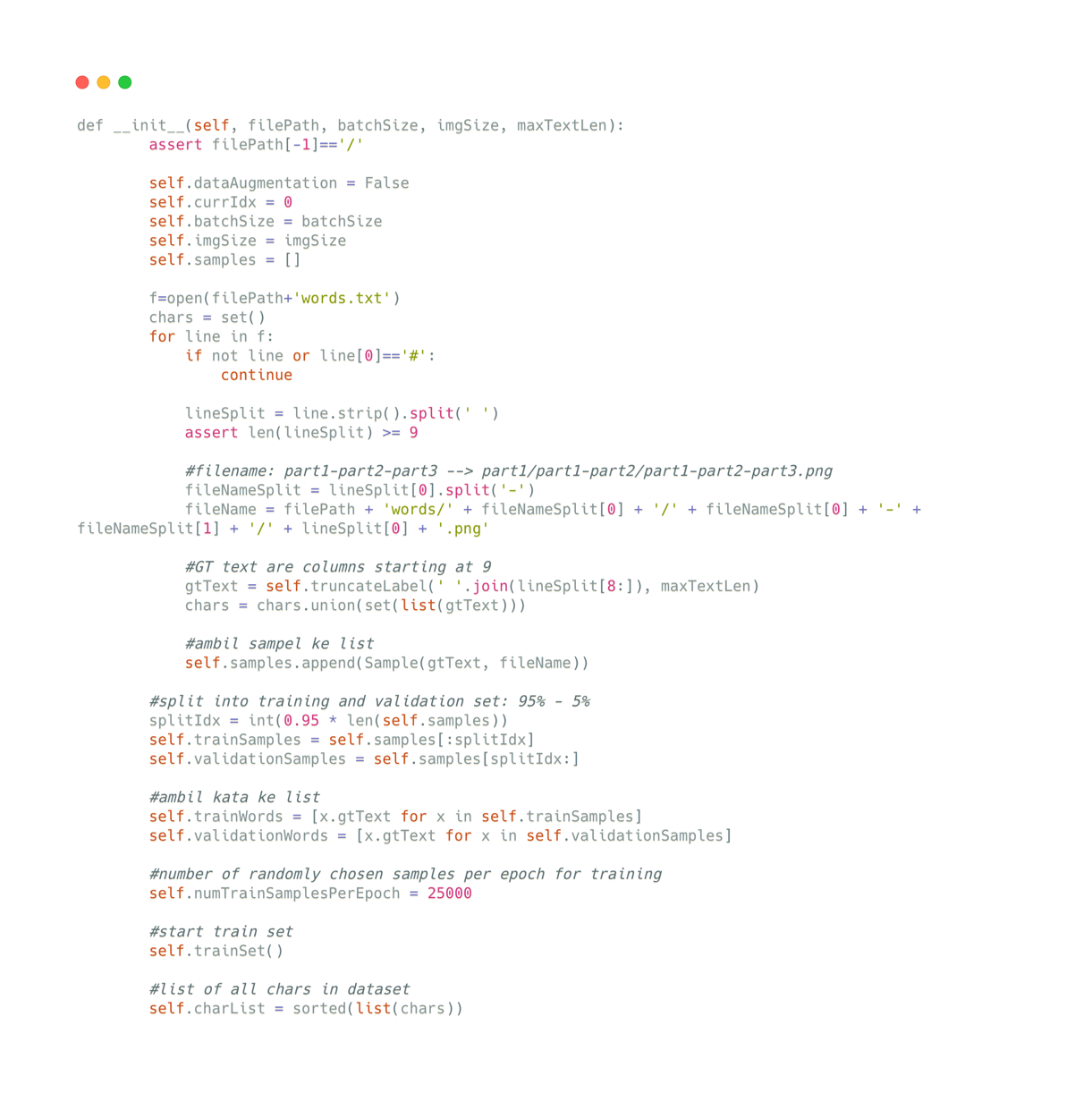
**4.2 Implementasi Proses dan Algoritma**

**4.2.1 Proses Deteksi Tulisan**



Gambar 4. 3 Source Code Deteksi Tulisan

**4.2.2 Proses Training Data**



Gambar 4. 4 Source Code Training Data

**4.2.3 Proses Pengenalan Tulisan**



Gambar 4. 5 Source Code Pengenalan Tulisan

**4.3 Pengujian Aplikasi dan Algoritma**

Pengujian aplikasi dan algoritma menggunakan tulisan tangan dokter yang diambil dari Google dan juga beberapa tulisan tangan pribadi dengan menggunakan penulisan huruf sambung. Semua gambar yang diuji sudah dilakukan proses *cropping* agar proses pengujian bisa lebih cepat lagi.



Gambar 4. 6 Tulisan Paracetamol



Gambar 4. 7 Hasil Deteksi Tulisan “paracetamol”

Pengujian berikut menggunakan sample dari Google, sample yang bertuliskan paracetamol ini terdeteksi dengan tulisan pamerame dengan probabilitas sebesar 0.08056225%. Probabilitas tersebut didasari berdasarkan hasil data training. Lain halnya dengan tulisan dengan bahasa Inggris, probabilitas yang didapat 0.97190386% seperti pada gambar 4.8.



Gambar 4. 8 Hasil Deteksi Tulisan “are”

Tabel 4. 1 Hasil pengujian secara detail.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Gambar | Tulisan | Hasil Tulisan | Probabilitas | Terindentifikasi |
|  | Paracetamol | pamerame | 0.080 | tidak |
|  | Patient | latiaent | 0.05 | tidak |
|  | before | bafore | 0.42 | tidak |
|  | Amoxicillin | amroxicialin | 0.14 | tidak |
|  | little | little | 0.96 | iya |
|  | morphine | uorplsine | 0.12 | tidak |
|  | Ferrous | ferrous | 0.86 | iya |
|  | sulfate | suilfate | 0.21 | tidak |
|  | Dispense | Dispense | 0.74 | iya |
|  | bottle | bottle | 0.86 | iya |
|  | Dextrometorpan | Devrnethorpan | 0.008 | tidak |
|  | Alcohol | Alookal | 0.02 | tidak |
|  | Guaifenesin | Guabfenesin | 0.006 | tidak |
|  | Colace | Colace | 0.84 | iya |
|  | Penicillin | Penierllen | 0.17 | tidak |
|  | Aretaminopen | Ataminopen | 0.12 | tidak |
|  | Aspirin | Aspirin | 0.30 | iya |
|  | Clopidrogel | Clopidoel | 0.05 | tidak |
|  | tablets | tatlets | 0.41 | tidak |
|  | Disoxin | Digosin | 0.12 | tidak |
|  | Antibiotic | untibiotic | 0.11 | tidak |
|  | coconut | coconut | 0.30 | iya |
|  | oil | sil | 0.58 | tidak |
|  | using | using | 0.72 | iya |
|  | vaccum | naccum | 0.59 | tidak |
|  | paracetamol | pavacetnnaf | 0.001 | tidak |
|  | paracetamol | paractano | 0.001 | tidak |
|  | tablet | taorer | 0.03 | tidak |
|  | fever | peron | 0.04 | tidak |
|  | amoxidan | amoridar | 0.05 | tidak |

Pengujian pada tabel 4.1 menggunakan gambar dan tulisan dari pulpen dan spidol. Tingkat akurasi dari pengujian dihitung menggunakan rumus sebagai berikut.

Rumus 4. 1 Perhitungan akurasi.

Berdasarkan rumus 4.1 maka hasil yang didapat yaitu sebesar 26%. Tingkat akurasi yang kecil tersebut dikarenakan dataset yang digunakan sedikit, dan perangkat untuk training data kurang mendukung.

**BAB 5  
KESIMPULAN DAN SARAN**

**5.1 Kesimpulan**

Dari hasil pengujian yang dilakukan maka dapat disimpulkan sebagai berikut:

1. Identifikasi wajah pada pengujian dapat dipengaruhi jarak dan cahaya.
2. Identifikasi wajah dengan objek wajah yang lebih besar dapat menghasilkan akurasi tinggi dan untuk objek wajah yang kecil menghasilkan akurasi yang rendah.
3. Dari hasil pengujian, tingkat akurasi dari metode Local Binary Pattern Histogram mencapai 52,8%.

**5.2 Saran**

Aplikasi deteksi tulisan tangan dokter diharapkan menjadi lebih baik dengan beberapa saran yang dapat digunakan:

1. Untuk proses deteksi dan training data menggunakan framework seperti *Tensorflow* atau *Keras.*
2. Untuk proses training disarankan menggunakan *GPU* yang tergolong *high-end* untuk mempercepat proses *training* data.
3. Dari model yang sudah ada, bisa dikembangkan ke platform lainnya untuk merealisasikannya ke publik.

# DAFTAR PUSTAKA

[1] E. Chammas, C. Mokbel, and L. Likforman-Sulem, “Handwriting recognition of historical documents with few labeled data,” *Proc. - 13th IAPR Int. Work. Doc. Anal. Syst. DAS 2018*, no. section IV, pp. 43–48, 2018, doi: 10.1109/DAS.2018.15.

[2] E. F. Yuwitaning, B. Hidayat, and N. Andini, “Implementasi Metode Hidden Markov Model untuk Deteksi Tulisan Tangan,” *e-Proceeding Eng.*, vol. 1, no. 1, pp. 396–402, 2014.

[3] R. Reeve Ingle, Y. Fujii, T. Deselaers, J. Baccash, and A. C. Popat, “A scalable handwritten text recognition system,” *Proc. Int. Conf. Doc. Anal. Recognition, ICDAR*, pp. 17–24, 2019, doi: 10.1109/ICDAR.2019.00013.

[4] T. Bluche, “Joint line segmentation and transcription for end-to-end handwritten paragraph recognition,” *Adv. Neural Inf. Process. Syst.*, pp. 838–846, 2016.

[5] H. Rahayu *et al.*, “Akurasi Kode Diagnosis Utama Pada Rm 1 Dokumen Rekam Medis Ruang Karmel Dan Karakteristik Petugas Koding Rawat Inap Rumah Sakit Mardi Rahayu Kudus Periode Desember 2009,” *Jl.Nakula I No*, vol. 10, no. 1, pp. 5–11, 2011.

[6] D. E.Kamalanaban, M. Gopinath, and S. Premkumar, “Medicine Box: Doctor’s Prescription Recognition Using Deep Machine Learning,” *Int. J. Eng. Technol.*, vol. 7, no. 3.34, p. 114, 2018, doi: 10.14419/ijet.v7i3.34.18785.

[7] Arrahman, Purwanto, and P. Nurtantio, “Klasifikasi Nama Obat Tulisan Tangan Dokter dengan Metode GLCM dan Backpropagation Neural Network,” *J. Teknol. Inf.*, vol. 12, pp. 123–134, 2016.

[8] D. T. Amalia and A. Sukohar, “Rational Drug Prescription Writing,” *Juke*, vol. 4, no. 7, pp. 22–30, 2014.

[9] I. Sukarsih, “R h -t d k k,” vol. 7, no. 1, pp. 9–15, 2007.

[10] A. Jafar, A. Kadafi, and F. Utaminingrum, “Deteksi Objek Penghalang Secara Real-Time Berbasis Mobile Bagi Penyandang Tunanetra Menggunakan Analisis Blob,” *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 1, pp. 423–432, 2018.

[11] R. Achkar, K. Ghayad, R. Haidar, S. Saleh, and R. Al Hajj, “Medical handwritten prescription recognition using CRNN,” *CITS 2019 - Proceeding 2019 Int. Conf. Comput. Inf. Telecommun. Syst.*, pp. 1–5, 2019, doi: 10.1109/CITS.2019.8862004.

[12] A. Purno and W. Wibowo, “Implementasi Teknik Computer Vision Dengan Metode Colored Markers Trajectory Secara Real Time,” *J. Tek. Inform.*, vol. 8, no. 1, pp. 45–48, 2016.

[13] D. Soselia, S. Amashukeli, I. Koberidze, and L. Shugliashvili, “RNN-based Online Handwritten Character Recognition Using Accelerometer and Gyroscope Data,” 2019.

[14] P. Sahare and S. B. Dhok, “Multilingual Character Segmentation and Recognition Schemes for Indian Document Images,” *IEEE Access*, vol. 6, no. ii, pp. 10603–10617, 2018, doi: 10.1109/ACCESS.2018.2795104.

[15] F. P. Such, D. Peri, F. Brockler, P. Hutkowski, and R. Ptucha, “Fully Convolutional Networks for Handwriting Recognition Workflow- Word Extraction Workflow- Word Recognition,” pp. 1–20, 2019.

[16] M. Fadhilla, M. R. A. Saf, and D. S. S. Sahid, “Pengenalan Kepribadian Seseorang Berdasarkan Pola Tulisan Tangan Menggunakan Jaringan Saraf Tiruan,” *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 6, no. 3, 2017, doi: 10.22146/jnteti.v6i3.340.

[17] Erik, M. A. H. A. S. I. S. W. A. EKSPLORITASI ANTARMUKA GRAFIS PEMAKAIAN TKINTER PADA LINGKUNGAN BAHASA PYTHON. Diss. Fakultas Teknik Unpas, 2008.