

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Subjek dan Objek Penelitian**

Penelitian yang dilakukan adalah perancangan sistem untuk mengklasifikasikan penyakit stroke hemoragik melalui gambar *CT Scan* otak menggunakan algoritma *Convolutional Neural Network* (CNN) dengan hasil yang diharapkan berupa sistem yang dapat berjalan sesuai dengan kebutuhan fungsionalnya dengan tingkat akurasi yang baik dan bekerja dalam waktu yang efektif.

Data yang digunakan adalah data sekunder yang diambil dari dataset penelitian [29] versi 1.3.1 yang telah memiliki lisensi dari PhysioNet (link dataset <https://physionet.org/content/ct-ich/1.3.1/>). Data tersebut terdiri dari CT Scan 82 pasien, 35 pasien di antaranya adalah penderita stroke hemoragik dan 47 pasien lainnya adalah normal. Setiap pasien memiliki sekitar 30-60 irisan gambar dengan usia rata-rata pasien adalah 27.8 tahun serta 46 pasien adalah laki-laki sedangkan 36 pasien lainnya adalah perempuan. Data orisinal yang didapatkan masih memiliki format NIfTI, maka data tersebut akan diubah menjadi gambar berformat png.

Pengumpulan data pada penelitian ini menggunakan teknik dokumentasi. Dokumentasi adalah sebuah teknik pengumpulan data yang menggunakan beberapa sumber yang berkaitan dengan data yang ingin diteliti baik berupa dokumen tertulis maupun elektronik. Teknik dokumentasi dilaksanakan dengan meneliti catatan-catatan penting yang memiliki hubungan dengan objek penelitian.

### 3.2 Alat dan Bahan Penelitian

Alat yang digunakan dalam penelitian adalah Laptop / PC dengan spesifikasi:

- Merek : Asus
- Type : A407UA-BV320T
- OS : Windows 10
- Processor : Intel Core i3
- RAM : DDR4 4 GB
- Storage : 1 TB HDD & 512 GB SSD

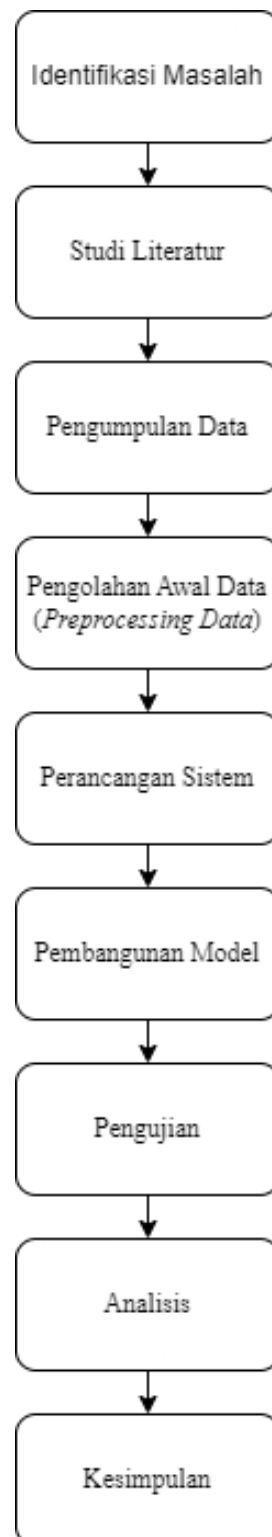
Terdapat beberapa *software* atau *tools* yang digunakan dalam penelitian ini antara lain :

1. Google Colaboratory
2. Jupyter Notebook
3. Microsoft Word (versi : 2013)

Bahan yang digunakan dalam penelitian adalah *dataset* berupa gambar (format NIFTI) CT *Scan* otak normal serta otak penderita stroke hemoragik yang akan diubah menjadi gambar berformat png.

### 3.3 Diagram Alir Penelitian

Penelitian akan dilaksanakan mengacu pada diagram alir penelitian yang telah dibuat, agar seluruh tahap penelitian dilaksanakan secara berurutan dan berkelanjutan tanpa menunda dan mengganggu tahap penelitian lainnya. Gambar 3.1 menampilkan diagram alir penelitian.



Gambar 3.1 Diagram Alir Penelitian

### **3.3.1 Identifikasi masalah**

Penelitian dimulai dari identifikasi masalah yang ada dan ingin diselesaikan. Identifikasi masalah merupakan tahap awal yang mampu memberikan gambaran bagi peneliti mengenai tujuan dan manfaat dari penelitian yang akan dilaksanakan.

### **3.3.2 Studi Literatur**

Penelitian dilanjutkan dengan melakukan studi literatur penelitian sebelumnya yang memiliki keterkaitan atau kesamaan objek penelitian maupun metode penelitian. Studi literatur merupakan tahap penting dalam melakukan penelitian untuk mendapatkan informasi dan gambaran mengenai objek atau metode yang akan digunakan, dimana hasil dari tahap studi literatur dapat digunakan sebagai pijakan atau rujukan penelitian yang akan dilaksanakan.

### **3.3.3 Pengumpulan Data**

Data yang digunakan dalam penelitian ini merupakan gambar berformat NIFTI dari CT Scan otak normal dan CT Scan otak penderita stroke hemoragik, yang diambil dari dataset penelitian [29] versi 1.3.1 yang telah memiliki lisensi dari PhysioNet (link dataset <https://physionet.org/content/ct-ich/1.3.1/>). Data yang digunakan berupa CT Scan 82 pasien, 35 pasien di antaranya adalah penderita stroke hemoragik dan 47 pasien lainnya adalah normal.

Pengumpulan data pada penelitian ini menggunakan teknik dokumentasi. Dokumentasi adalah sebuah teknik pengumpulan data yang menggunakan beberapa sumber yang berkaitan dengan data yang ingin diteliti baik berupa dokumen tertulis maupun elektronik. Teknik dokumentasi dilaksanakan dengan meneliti catatan-catatan penting yang memiliki hubungan dengan objek penelitian. Data yang terkumpul dari sumber dataset berformat ZIP dengan beberapa file di dalamnya yaitu

patient\_demographics.csv, hemorrhage\_diagnosis\_raw\_ct.csv, folder ct\_scans dan folder masks. Dalam penelitian data yang digunakan adalah hemorrhage\_diagnosis\_raw\_ct.csv dan file CT Scan pada folder ct\_scans.

- hemorrhage\_diagnosis\_raw\_ct.csv

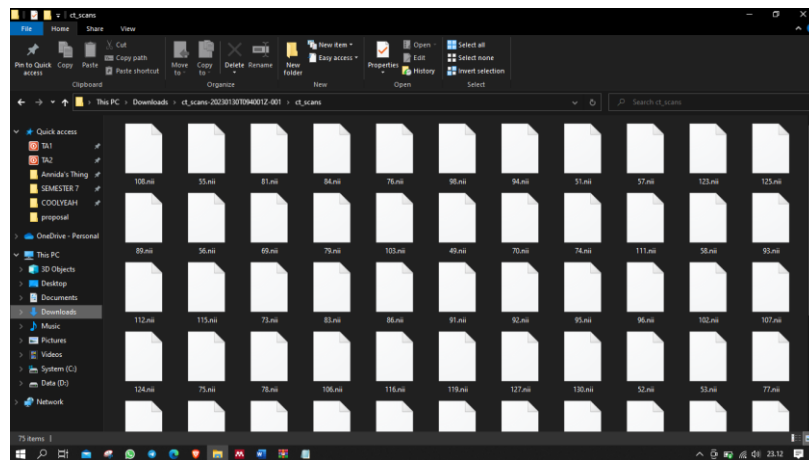
Merupakan file demografis pasien yang berisi data irisan dan label tipe yang telah didiagnosis untuk setiap irisan dari hasil pemindaian atau ekstraksi file CT Scan NIFTI. Data yang ada pada file ini adalah 2814 baris dengan 9 atribut yaitu PatientNumber, SliceNumber, Intraventricular, Intraparenchymal, Subarachnoid, Epidural, Subdural, No\_Hemorrhage, dan Fracture\_Yes\_No. Gambar 3.2 menampilkan beberapa baris pertama dari file hemorrhage\_diagnosis\_raw\_ct.csv.

PatientNumber	SliceNumber	Intraventricular	Intraparenchymal	Subarachnoid	Epidural	Subdural	No_Hemorrhage	Fracture_Yes_No
49	1	0	0	0	0	0	0	1
49	2	0	0	0	0	0	0	1
49	3	0	0	0	0	0	0	1
49	4	0	0	0	0	0	0	1
49	5	0	0	0	0	0	0	1
49	6	0	0	0	0	0	0	1
49	7	0	0	0	0	0	0	1
49	8	0	0	0	0	0	0	1
49	9	0	0	0	0	0	0	1
49	10	0	0	0	0	0	0	1
49	11	0	0	0	0	0	0	1
49	12	0	0	0	0	0	0	1
49	13	0	0	0	0	0	0	1
49	14	0	0	0	0	1	0	0
49	15	0	0	0	0	1	0	0
49	16	0	0	0	0	1	0	0
49	17	0	0	0	0	1	0	0
49	18	0	0	0	0	0	0	1
49	19	0	0	0	0	0	0	1

Gambar 3.2 File hemorrhage\_diagnosis\_raw\_ct.csv

- Folder ct\_scans

Berisi file CT Scan dalam format NIFTI dari setiap pasien. File-file pada folder ini telah disesuaikan dengan nomor pasien pada file demografis pasien hemorrhage\_diagnosis\_raw\_ct.csv. Jumlah total file CT Scan pada folder ini adalah 75 file NIFTI, dengan ukuran file dalam rentang 15 mb sampai dengan 29 mb. Gambar 3.3 menampilkan file-file yang terdalem dalam folder ct\_scans.



Gambar 3.3 File pada folder ct\_scans

### 3.3.4 Pengolahan Awal Data

#### 3.3.4.1 Ekstraksi NIFTI

Data yang telah terkumpul akan dilakukan pengolahan awal data (*preprocessing data*). Dimana data CT Scan berformat NIFTI akan diekstraksi terlebih dahulu data citra dari setiap slice menjadi data gambar berformat PNG. Proses ekstraksi dilakukan dengan library nibabel untuk melakukan load data NIFTI.

```
def nifti2jpg(wkwk):
    folder = wkwk[139:]
    folder = folder.split('.')[0]
    ini = nib.load(wkwk)
    ini = ini.get_data()
    os.mkdir(f'/content/drive/MyDrive/Colab Notebooks/TA/TA2/output/{folder}/')
    for i in range(0, ini.shape[2]):
        fig = plt.imshow(ini[:, :, i])
        fig.set_cmap('gray')
        fig.axes.get_xaxis().set_visible(False)
        fig.axes.get_yaxis().set_visible(False)
        plt.savefig(f'/content/drive/MyDrive/Colab Notebooks/TA/TA2/output/{folder}/{i+1}.png', transparent=True)

for i in range(len(res)):
    nifti = res[i]
    data = '/content/drive/MyDrive/Colab Notebooks/TA/TA2/ct_scans/'+res[i]
    nifti2jpg(data)
```

Gambar 3.4 Ekstraksi data citra setiap slice CT Scan

Hasil dari proses ekstraksi di atas berupa folder-folder yang berupa citra setiap slice dari CT Scan pasien dengan total 2.814 data

gambar PNG. Jumlah slice dari setiap CT Scan pasien sebagai berikut :

Tabel 3.1 Jumlah Slice CT Scan Dataset

No Pasien	Jumlah Slice CT Scan	No Pasien	Jumlah Slice CT Scan
49	39	94	47
50	32	95	37
51	46	96	37
52	35	97	33
53	35	98	48
54	33	99	33
55	56	100	34
56	40	101	35
57	43	102	37
58	38	103	40
66	34	104	31
67	31	105	34
68	32	106	36
69	40	107	37
70	39	108	58
71	34	109	33
72	33	110	33
73	37	111	39
74	39	112	38
75	36	113	33
76	51	114	32
77	35	115	38
78	36	116	36
79	40	117	33
80	33	118	35
81	56	119	36
82	34	120	35

No Pasien	Jumlah Slice CT Scan
83	37
84	52
85	34
86	37
87	32
88	34
89	42
90	33
91	37
92	37
93	38

No Pasien	Jumlah Slice CT Scan
121	35
122	32
123	43
124	37
125	43
126	34
127	36
128	35
129	35
130	36



Gambar 3.5 Contoh Output Proses Ekstraksi data citra setiap slice CT Scan

Proses ekstraksi CT Scan berformat NIFTI menjadi gambar-gambar slice CT Scan berformat PNG akan menghasilkan jumlah gambar yang berbeda-beda, karena dipengaruhi oleh faktor slice thickness yang digunakan pada setiap CT Scan. Gambar pada setiap slice atau irisan CT Scan ditransformasi menggunakan indeks pada ukuran CT Scan yang terbaca.

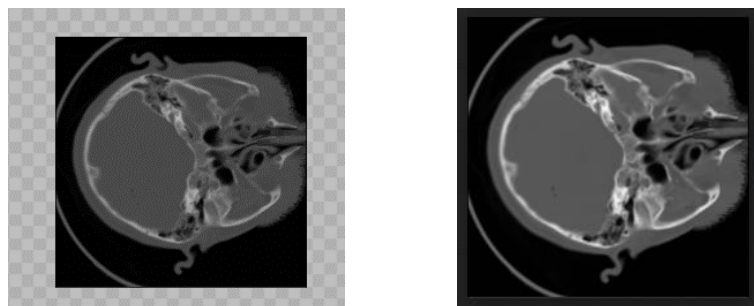


### 3.3.4.2 Removing transparency

Tahap *preprocessing* selanjutnya adalah menghapus latar belakang transparans pada setiap citra. Output dari tahap *preprocessing* ini adalah citra tanpa latar belakang transparans.

```
def autocrop_image(image, output_dir, border = 0):
    # Get the bounding box
    bbox = image.getbbox()
    # Crop the image to the contents of the bounding box
    image = image.crop(bbox)
    # Determine the width and height of the cropped image
    (width, height) = image.size
    # Add border
    width += border * 2
    height += border * 2
    # Create a new image object for the output image
    cropped_image = Image.new("RGBA", (width, height), (0,0,0,0))
    # Paste the cropped image onto the new image
    cropped_image.paste(image, (border, border))
    # Done!
    return cropped_image.save(output_dir)
```

Gambar 3.6 Fungsi Menghapus Latar Belakang Transparans pada Citra



(a)

(b)

Gambar 3.7 (a) citra dengan background (b) citra tanpa background

Citra sebelum dilakukan removing transparency background memiliki alpha channel atau background transparan di sekitar objek, dimana alpha channel ini akan terbaca saat proses pelatihan model dan mempengaruhi performa proses training sehingga alpha channel

ini harus dihilangkan. Citra hasil removing hanya menampilkan objek saja tanpa background transparans atau alpha channel di sekitarnya,

### 3.3.4.3 Labeling

Pelabelan setiap slice dilakukan dengan berpacu pada data demografis pasien `hemorrhage_diagnosis_raw_ct.csv` atribut `PatientNumber`, `SliceNumber`, dan `No_Hemorrhage`. Output dari pelabelan berupa slice citra CT Scan yang sudah terbagi menjadi 2 kelas yaitu hemoragik dan normal.

```
for i in range(data.shape[0]):
    row = data.iloc[i]
    ct = row['PatientNumber']
    category = row['No_Hemorrhage']
    sli = row['SliceNumber']

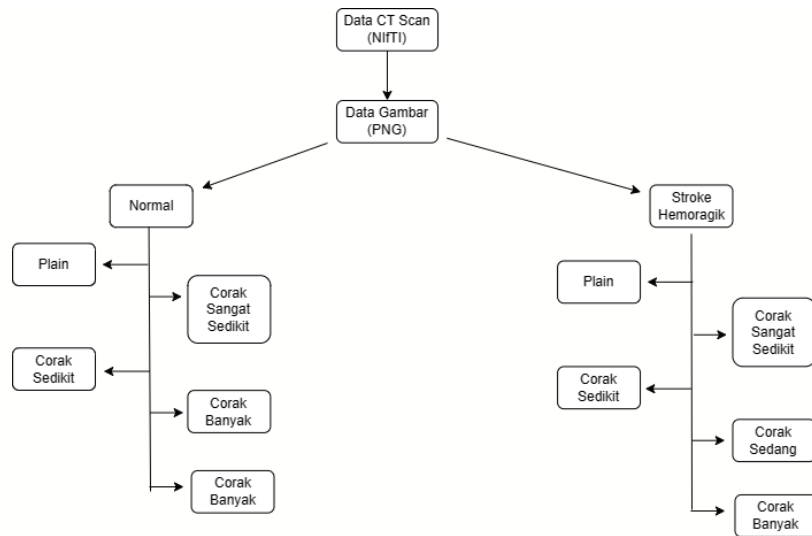
    res = os.listdir(f'/content/drive/MyDrive/Colab Notebooks/TA/TA2/output/{ct}')
    obj = f'/content/drive/MyDrive/Colab Notebooks/TA/TA2/output/{ct}/{sli}.png'
    dir = f'/content/drive/MyDrive/Colab Notebooks/TA/TA2/dataset/{category}/{ct}/{sli}.png'

    shutil.copy(obj,dir)
```

Gambar 3.8 Proses Pelabelan

### 3.3.4.4 Augmentasi

Untuk mempermudah proses klasifikasi setiap data pada setiap kelas, peneliti membagi data berdasarkan kuantitas corak pada CT Scan yaitu plain, corak sangat sedikit, corak sedikit, corak sedang, dan corak banyak. Gambar 3.9 menampilkan ilustrasi pembagian data berdasarkan corak.

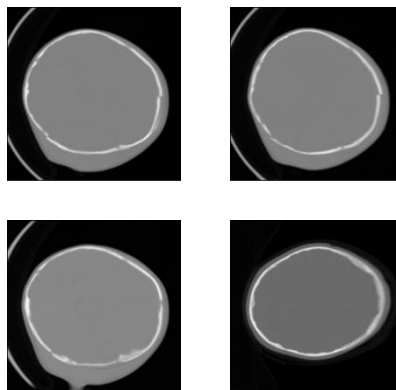


Gambar 3.9 Pembagian Data Berdasarkan Corak

Peneliti menetapkan indikator untuk pembagian corak pada gambar CT Scan :

#### 1. Corak Plain (Tidak Bercorak)

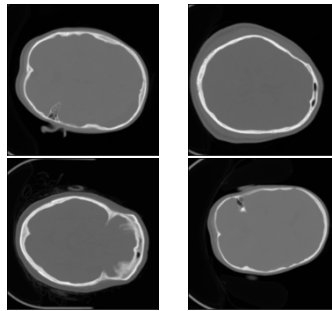
Data citra gambar CT Scan peneliti kategorikan sebagai corak plain atau tidak bercorak jika pada gambar CT Scan tidak ada bercak di sekitar otak sebagai objek utamanya. Gambar 3.10 menampilkan contoh gambar CT Scan yang dikategorikan sebagai corak plain atau tidak bercorak.



Gambar 3.10 CT Scan Plain

## 2. Corak Sangat Sedikit

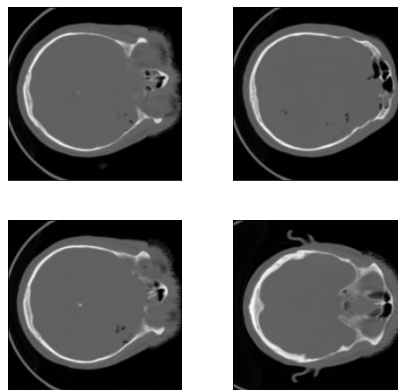
Data citra gambar CT Scan peneliti kategorikan sebagai corak sangat sedikit jika pada gambar CT Scan terdapat bercak kecil di sekitar garis tepi. Gambar 3.11 menampilkan contoh gambar CT Scan yang dikategorikan sebagai corak sangat sedikit.



Gambar 3.11 CT Scan Corak Sangat Sedikit

## 3. Corak Sedikit

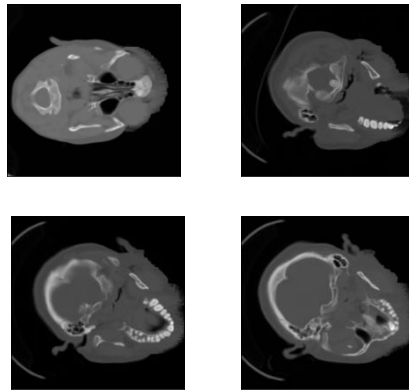
Data citra gambar CT Scan peneliti kategorikan sebagai corak sedikit jika pada gambar CT Scan terdapat bercak besar di sekitar garis tepi dan bercak kecil di tengah objek. Gambar 3.12 menampilkan contoh gambar CT Scan yang dikategorikan sebagai corak sedikit.



Gambar 3.12 CT Scan Corak Sedikit

#### 4. Corak Sedang

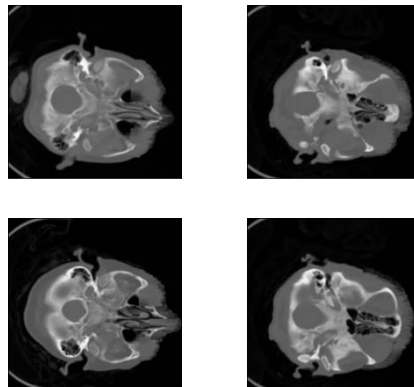
Data citra gambar CT Scan peneliti kategorikan sebagai corak sedang jika pada gambar CT Scan terdapat bercak besar di sekitar garis tepi dan bercak besar di tengah objek. Gambar 3.13 menampilkan contoh gambar CT Scan yang dikategorikan sebagai corak sedang.



Gambar 3.13 CT Scan Corak Sedang

#### 5. Corak Banyak

Data citra gambar CT Scan peneliti kategorikan sebagai corak banyak jika pada gambar CT Scan memiliki bercak di seluruh bagian otak sebagai objek utamanya. Gambar 3.14 menampilkan contoh gambar CT Scan yang dikategorikan sebagai corak banyak.



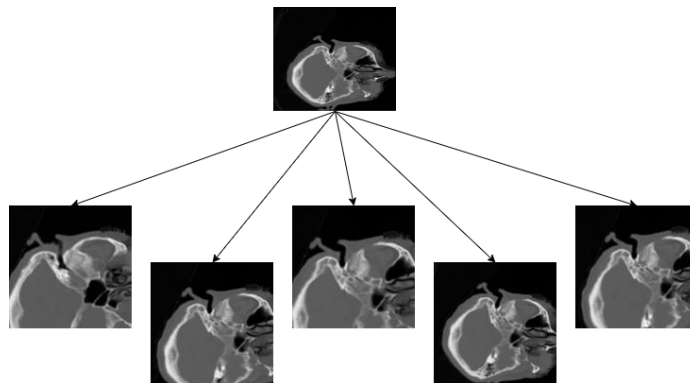
Gambar 3.14 CT Scan Corak Banyak

Tahap augmentasi dilakukan agar data seimbang, dengan memperbanyak data menggunakan transformasi data seperti pemutaran (root), pembesaran (zoom), pemotongan (crop), dan pengubahan ukuran (resize) [50]. Gambar 3.15 menampilkan fungsi augmentasi yang dilakukan.

```
def perbanyak_(ini, sebanyak_ini):
    source_dir = ini
    output_dir = "."
    p = Augmentor.Pipeline(source_directory=source_dir, output_directory=output_dir)
    #merotasi inputan
    p.rotate(probability=0.7, max_left_rotation=13, max_right_rotation=13)
    #memperbesar inputan
    p.zoom_random(probability=0.5, percentage_area=0.9)
    #memotong inputan
    p.crop_random(probability=0.6, percentage_area=0.9)
    #mengubah ukuran inputan
    p.resize(probability=1.0, width=218, height=218)
    #mengambil sample sebanyak x dari inputan
    p.sample(sebanyak_ini)
```

Gambar 3.15 Fungsi Augmentasi

Ilustrasi salah satu output augmentasi terlihat pada gambar 3.16.

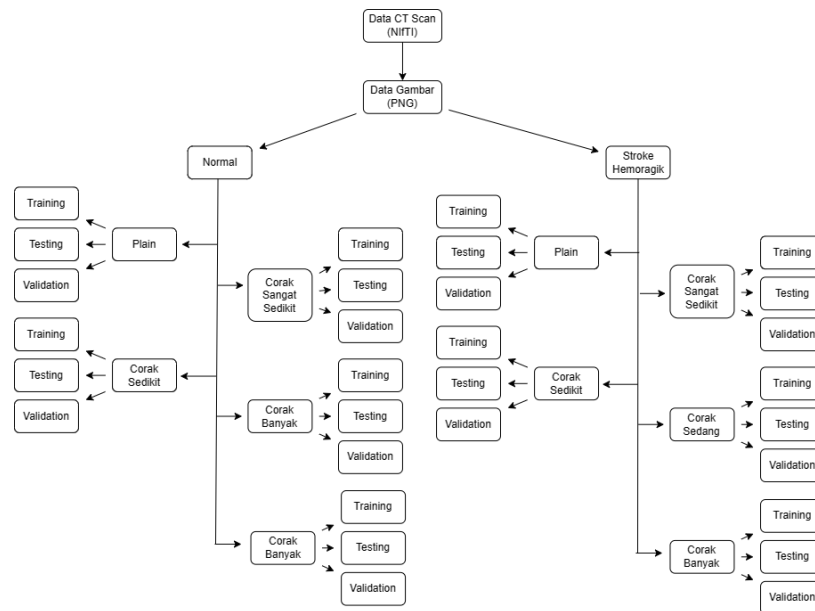


Gambar 3.16 Output Proses Augmentasi

### 3.3.4.5 *Splitting Dataset*

Gambar 3.17 menampilkan ilustrasi *splitting* data yang dilakukan. Data menjadi data untuk proses pelatihan (*training*), validasi (*validation*), dan pengujian (*testing*). Adapun perbandingan

yang digunakan dalam pembagian data untuk pelatihan (*training*), validasi (*validation*), dan pengujian (*testing*) adalah 8:1:1.



Gambar 3.17 Tahap *Preprocessing* Data

Proses *splitting* dataset dilakukan menggunakan teknik split folder, dimana folder yang akan dipisahkan adalah folder yang memiliki folder-folder kelas dengan seluruh datanya yang tersimpan di dalamnya. Gambar 3.18 menampilkan *source code* yang digunakan dalam tahap *splitting* dataset.

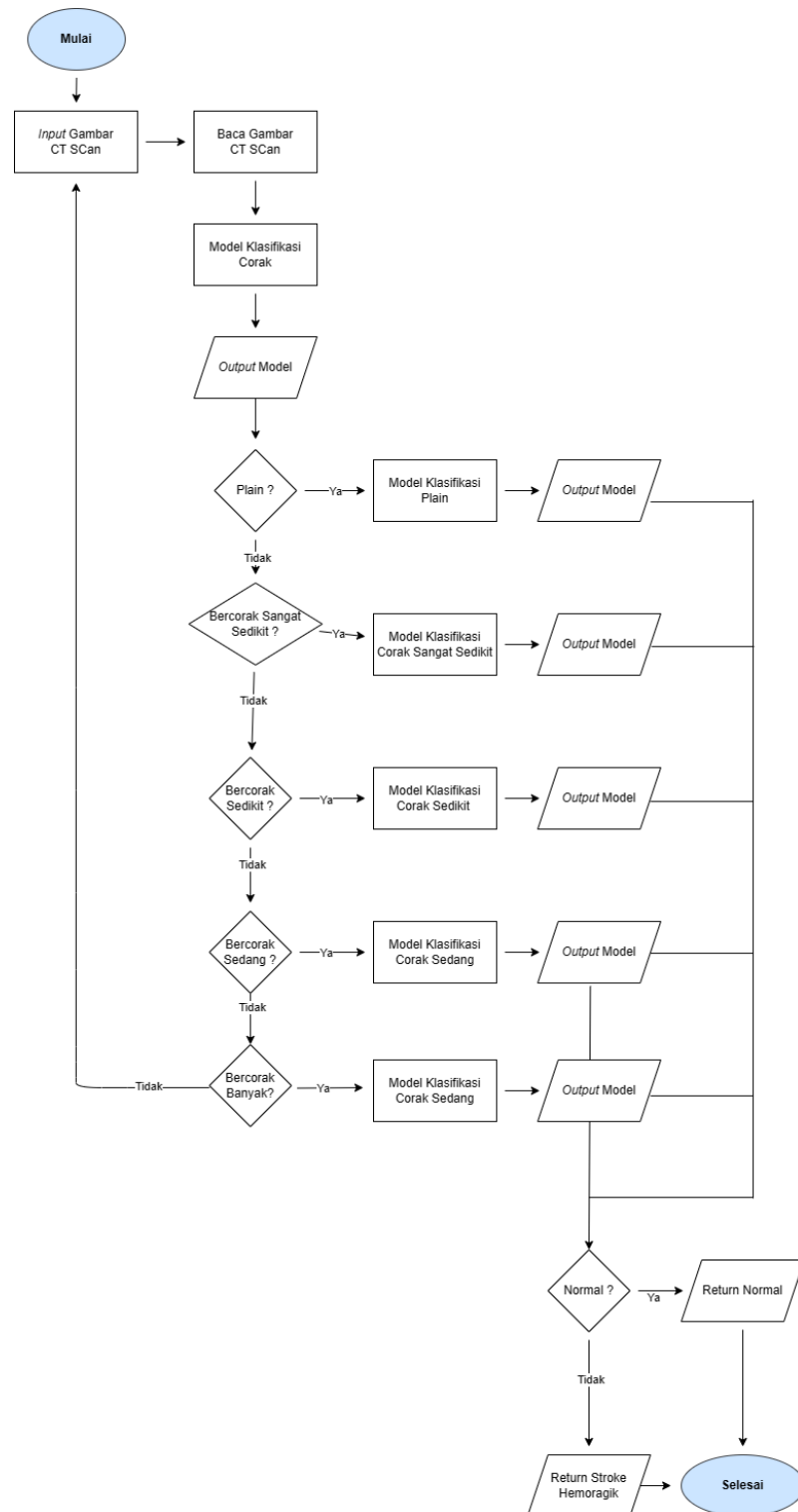
```

splitfolders.ratio(
  "/content/drive/myDrive/Colab Notebooks/1. TA/1. TA2/2. dataset_ready/Dataset1(Normal+Hemoragik)",
  output="/content/drive/myDrive/Colab Notebooks/1. TA/1. TA2/2. dataset_ready/fix ready/Dataset1(Normal+Hemoragik)",
  seed=1337, ratio=(.8, .1, .1), group_prefix=None, move=False) # default values
  
```

Gambar 3.0-18 *Source code* Tahap *Splitting*

### 3.3.5 Perancangan Sistem

Gambar 3.19 menampilkan *flowchart* dari sistem klasifikasi penyakit stroke hemoragik yang akan dibuat.

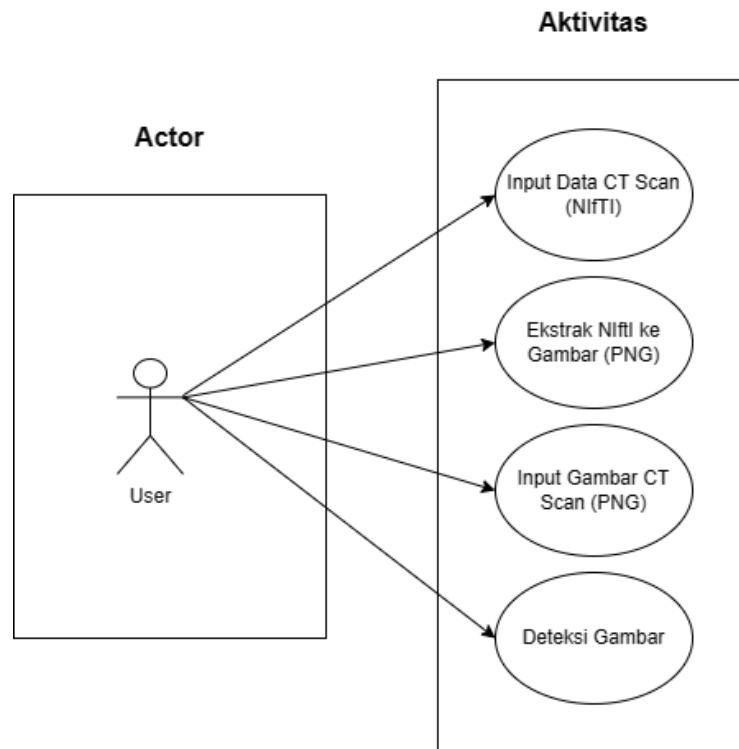


Gambar 3.19 *Flowchart* Sistem Klasifikasi Penyakit Stroke Hemoragik



Sistem klasifikasi stroke hemoragik akan dirancang dengan mengintegrasikan 6 model *Convolutional Neural Network* (CNN) yaitu model klasifikasi corak gambar CT Scan (*plain*, corak sangat sedikit, corak sedikit, corak sedang, atau corak banyak), model klasifikasi *plain* (normal atau stroke hemoragik), model klasifikasi corak sangat sedikit (normal atau stroke hemoragik), model klasifikasi corak sedikit (normal atau stroke hemoragik), model klasifikasi corak sedang (normal atau stroke hemoragik), dan model klasifikasi corak banyak (normal atau stroke hemoragik). Ketika sebuah *input*-an masuk maka model yang dijalankan pertama kali adalah model klasifikasi corak pada gambar CT Scan dengan *output*: apakah gambar yang di-*input*-kan termasuk corak *plain*, corak sangat sedikit, corak sedikit, corak sedang, atau corak banyak. Jika pada model klasifikasi pertama sudah menghasilkan *output*, maka model selanjutnya yang akan dijalankan sesuai dengan output pada model klasifikasi pertama, dimana model selanjutnya akan menghasilkan *output*: apakah gambar yang di-*input*-kan termasuk normal atau stroke hemoragik.

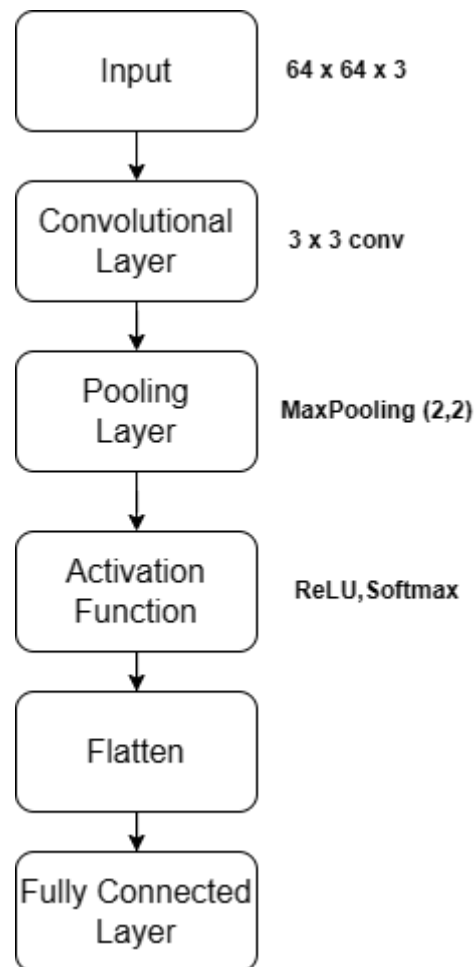
Sistem klasifikasi penyakit stroke hemoragik yang dirancang akan diimplementasikan dalam bentuk web sederhana dengan mengintegrasikan kode program dan streamlit, bertujuan untuk mempermudah pengguna dalam penggunaannya. Untuk mengetahui hubungan antara pengguna dengan sistem atau hal apa saja yang dapat dilakukan pengguna di sistem yang telah dirancang, gambar 3.20 menampilkan use case diagram.



Gambar 3.20 Use case Diagram Sistem

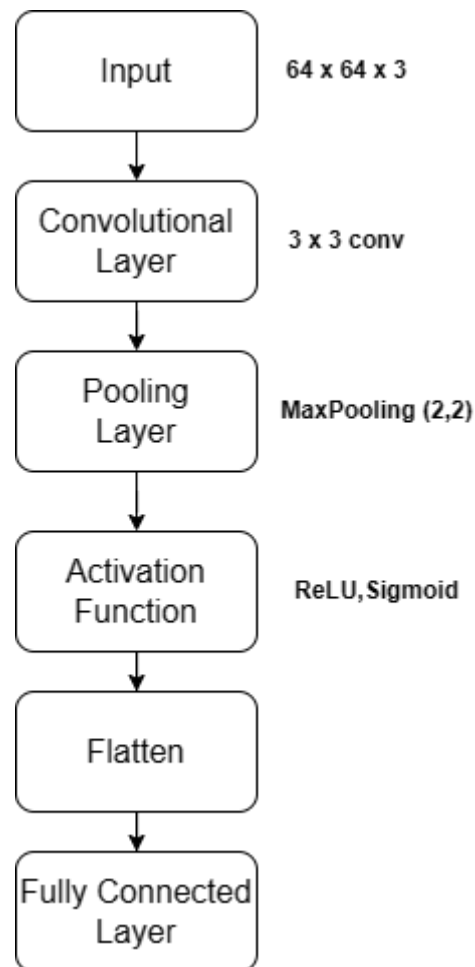
### 3.3.6 Pembangunan Model

Model yang digunakan dalam penelitian ini adalah Convolutional Neural Network (CNN). Jumlah model yang dibangun ada 6 yaitu model klasifikasi corak menggunakan dataset 1, model klasifikasi hemoragik/normal corak banyak menggunakan dataset 2, model klasifikasi hemoragik/normal corak sedang menggunakan dataset 3, model klasifikasi hemoragik/normal corak sedikit menggunakan dataset 4, model klasifikasi hemoragik/normal corak sangat sedikit menggunakan dataset 5, dan model klasifikasi hemoragik/normal plain menggunakan dataset 6.



Gambar 3.21 Proses *Training* Model 1 (Klasifikasi Corak)

Proses *training* model 1 yaitu klasifikasi corak dimulai pada dari proses *training* adalah tahap *rescale*, yaitu mengubah ukuran input menjadi 64x64. Kemudian diproses di *convolutional layer* dengan ukuran 3x3, *pooling layer* jenis max pooling ukuran 2x2, *activation function* jenis softmax, *flatten layer*, dan *fully-connected layer*.



Gambar 3.22 Proses *Training* Model 2,3,4,5,6

Proses *training* model 2,3,4,5 yaitu klasifikasi corak dimulai pada dari proses *training* adalah tahap *rescale*, yaitu mengubah ukuran input menjadi 64x64. Kemudian diproses di *convolutional layer* dengan ukuran 3x3, *pooling layer* jenis max pooling ukuran 2x2, *activation function* jenis sigmoid, *flatten layer*, dan *fully-connected layer*.

### 3.3.7 Pengujian

Langkah selanjutnya adalah pengujian model klasifikasi penyakit stroke hemoragik yang telah dirancang. Pengujian dapat membantu mengevaluasi ketahanan dan mendeteksi kerentanan lebih awal model deep

learning yang dibangun [51]. Pengujian dilakukan dengan melakukan uji coba klasifikasi menggunakan *dataset testing*. Dimana kelas hasil uji coba tersebut akan dibandingkan dengan kelas aktualnya, menggunakan *confusion matrix*. Adapun jenis parameter evaluasi *confusion matrix* yang digunakan berupa *accuracy* atau nilai akurasi.

### **3.3.8 Analisis**

Langkah selanjutnya adalah analisis hasil pengujian. Pada tahap ini masing-masing hasil pengujian model akan dibandingkan untuk menentukan performa terbaik dari masing-masing model. Tahap ini akan menghasilkan model dengan performa terbaik beserta akurasinya. Untuk dilanjutkan ke tahap implementasi sistem.

### **3.3.9 Kesimpulan**

Penarikan kesimpulan akan dilakukan berdasarkan tahap sebelumnya yaitu tahap pengujian. Kesimpulan yang dihasilkan dapat menjadi jawaban atas pertanyaan penelitian.