

## LAMPIRAN

### Lampiran 1: Potongan Kode *Crawling* Judul Publikasi

```
dataset.py

## Melakukan pengecekan apakah file berkenstensi csv
## berisi data judul publikasi hasil proses crawling atau tidak
## jika file berekstensi csv tersebut kosong
## maka dilakukan proses crawling

if df2.empty:
    with open('dataset_naive_bayes/file_utama.csv', 'w',
newline='') as file:

## Membuat baris pada kolom yang berisi nomor, judul jurnal
## bidang, grup riset, kata judul jurnal hasil filter pada file
## berekstensi csv yang digunakan untuk menyimpan data judul
## publikasi hasil proses crawling

        writer = csv.writer(file)
        writer.writerow(["No", "Judul Jurnal", "Bidang", "Grup
Riset", "Kata Judul Jurnal Hasil Filter"])
        try:
            while no_grup_riset < len(GrupRiset):
                print('$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$')
                nama_grup_riset = NamaGrupRiset[no_grup_riset]
                print(f'Grup Riset {nama_grup_riset}')
                while no_bidang < len(GrupRiset[no_grup_riset]):

## Melakukan proses crawling dengan setiap bidang penelitian
## dilakukan proses crawling untuk 50 judul publikasi

                    if no_halaman <= 40:
                        bidang =
GrupRiset[no_grup_riset][no_bidang]
                        nama_bidang = bidang.replace(" ", "+")
                        print("Nama Bidang " + bidang)

## Memasukkan url untuk halaman yang dilakukan proses crawling
```

```

## dengan menginisiasi kata kunci yaitu berupa bidang penelitian

        url =
f'https://scholar.google.com/scholar?start={no_halaman}&q={nama_bi
dang)+Teknik+Informatika&hl=id&as_sdt=0,5'
        print('-----
-')

        print(bidang)
        print(url)
        no_halaman += 10

        response = requests.get(url,
headers=headers_2)

        print(response.status_code)

## melakukan pengecekan apakah status code pada url sama dengan
## 200, jika sama dengan 200 bertanda proses request terhadap
## url tersebut sukses, sehingga dapat dilakukan proses crawling
## jika status code tidak sama dengan 200 maka proses crawling
## tidak dilanjutkan

        if response.status_code == 200:
            soup = BeautifulSoup(response.content,
'lxml')

            for item in soup.select('[data-lid]'):
                try:

print('+++++')

                    nomor_judul += 1

## Mengambil judul publikasi yang terdapat didaftar judul
## publikasi pada halaman dengan html tag adalah h3

                    judul_jurnal =
item.select('h3')[0].get_text()

                    hitung_jurnal_bidang += 1

```

Lampiran 1 Potongan Kode *Crawling* Judul Publikasi

## Lampiran 2: Potongan Kode *Case Folding*

```
skripsi_filter/dataset.py
```

```
## Proses mengubah huruf pada kata judul publikasi seluruh  
## hurufnya menjadi huruf kecil  
## menggunakan fungsi pada library sastrawi  
  
judul = stemmer.stem(judul_jurnal)
```

## Lampiran 2 Potongan Kode *Case Folding*

### Lampiran 3: Potongan Kode *Stopword Removal*

```
skripsi_filter/dataset.py

## Melakukan pengecekan setiap kata pada judul
## yang sudah dilakukan proses case folding
## terhadap kata pada file filter.csv
## yang berisi kata-kata yang bersifat umum
## jika terdapat kata yang sama antara kata pada judul
## dan kata pada file filter.csv
## maka kata tersebut akan dihilangkan

while no_split < len(judul_jurnal_split):
    kata = judul_jurnal_split[no_split]
    array_kata = list(df1['Kata'])

    panjang_data_kata = 1
    panjang_array_kata = len(array_kata)

    s = dict()

    for i in range(panjang_array_kata):
        s[array_kata[i]] = 1
    for i in range(panjang_data_kata):
        if kata not in s.keys():
            print(kata)
```

### Lampiran 3 Potongan Kode *Stopword Removal*

#### **Lampiran 4: Potongan Kode *Stemming***

skripsi\_filter/dataset.py

```
## proses menghilangkan imbuhan pada kata
## agar setiap kata pada judul berubah menjadi kata dasar
## proses tersebut menggunakan fungsi pada library sastrawi

judul = stemmer.stem(judul_jurnal)
print(judul)
```

Lampiran 4 Potongan Kode *Stemming*

## Lampiran 5: Potongan Kode *Tokenization*

```
skripsi_filter/dataset.py

## menghilangkan tanda baca pada setiap kata pada judul

judul = (re.sub('[():,.]', "", judul_jurnal))

## menghilangkan tanda yang tidak diperlukan pada setiap kata
pada ## judul

judul = judul.replace('[PDF][PDF]', '')
judul = judul.replace('❖❖', '')
judul = judul.replace('[PERNYATAAN][C]', '')
judul = judul.replace('[BUKU][B]', '')

print(judul)

kata_dasar = stemmer.stem(judul)

## memisahkan setiap kata menjadi kalimat sendiri-sendiri

judul_jurnal_split = kata_dasar.split()
```

## Lampiran 5 Potongan Kode *Tokenization*

## Lampiran 6: Potongan Kode Pembagian *Dataset*

skripsi\_filter/split.py

```
## Fungsi untuk melakukan pembagian dataset
## menggunakan teknik k-fold cross validation dengan nilai k=5
## memasukkan parameter berupa nomor dokumen
## hasil pembagian dataset
## nilai a yaitu nomor urutan awal pada dataset
## nilai b yaitu nomor urutan akhir pada dataset
## nilai a dan b menandakan batasan
## terhadap nomor urutan dataset yang dijadikan patokan untuk
## membagi dataset menjadi 5 dokumen

def split_data(no_dok, nilai_a, nilai_b):
    ## ----- awal -----

    # dilakukan pengecekan jika nilai a sebagai nomor urutan awal sama
    # dengan 1

        if nilai_a == 1:

            ## pada bagian ini dilakukan pembagian dataset
            ## untuk dokumen ke 1 dengan nilai a sebagai nomor urutan awal
            ## sama dengan 1
            ## dan nilai b sebagai nomor urutan akhir sama dengan 10

                with
            open(f'dataset_naive_bayes/data_testing/data_testing_{no_dok}.csv'
, 'w', newline='', encoding='ISO-8859-1') as file:
                writer = csv.writer(file)
                writer.writerow(['No', 'Bidang', 'Grup Riset', 'Judul
Jurnal', 'Kata Judul Jurnal Hasil Filter'])
                a = nilai_a
                b = nilai_b
                c = 50

                e = 38

                no_split = 0
```

```

no_header = 0

header_array = ['No', 'Bidang', 'Grup Riset', 'Judul
Jurnal', 'Kata Judul Jurnal Hasil Filter']

array_coba = ['list_1', 'list_2', 'list_3', 'list_4',
'list_5']

no_array_coba = 0

while no_split < e:
    while no_header < len(header_array):
        array_tabel =
list(df1[f'{header_array[no_header]}'].loc[(df1['No'] >= a) &
(df1['No'] <= b)])

        array_coba[no_header] = array_tabel
        no_header += 1
        while no_array_coba < len(array_coba[0]):
            writer.writerow(
                [array_coba[0][no_array_coba],
array_coba[1][no_array_coba], array_coba[2][no_array_coba],
                array_coba[3][no_array_coba],
array_coba[4][no_array_coba]])
            no_array_coba += 1
        no_header = 0
        no_array_coba = 0
        a += 50
        b += 50
        no_split += 1

    with
open(f'dataset_naive_bayes/data_training/data_training_{no_dok}.cs
v', 'w', newline='', encoding='ISO-8859-1') as file:
        writer = csv.writer(file)
        writer.writerow(['No', 'Bidang', 'Grup Riset', 'Judul
Jurnal', 'Kata Judul Jurnal Hasil Filter'])
        b = nilai_b
        c = 50

```

```

e = 38

no_split = 0

no_header = 0

header_array = ['No', 'Bidang', 'Grup Riset', 'Judul
Jurnal', 'Kata Judul Jurnal Hasil Filter']

array_coba = ['list_1', 'list_2', 'list_3', 'list_4',
'list_5']

no_array_coba = 0

while no_split < e:
    while no_header < len(header_array):
        array_tabel =
list(df1[f'{header_array[no_header]}'].loc[(df1['No'] > b) &
(df1['No'] <= c)])

        array_coba[no_header] = array_tabel
        no_header += 1
    while no_array_coba < len(array_coba[0]):
        writer.writerow(
            [array_coba[0][no_array_coba],
array_coba[1][no_array_coba], array_coba[2][no_array_coba],
            array_coba[3][no_array_coba],
array_coba[4][no_array_coba]])
        no_array_coba += 1
    no_header = 0
    no_array_coba = 0
    b += 50
    c += 50
    no_split += 1

## ----- tengah -----

# dilakukan pengecekan jika nilai a sebagai nomor urutan awal
# lebih dari 10 dan kurang dari 41

```

```

elif nilai_a > 10 & nilai_a < 41:

## pada bagian ini dilakukan pembagian dataset
## untuk dokumen ke 2 dengan nilai a sebagai nomor urutan awal
## sama dengan 11
## dan nilai b sebagai nomor urutan akhir sama dengan 20

        with
open(f'dataset_naive_bayes/data_testing/data_testing_{no_dok}.csv'
, 'w', newline='', encoding='ISO-8859-1') as file:
            writer = csv.writer(file)
            writer.writerow(['No', 'Bidang', 'Grup Riset', 'Judul
Jurnal', 'Kata Judul Jurnal Hasil Filter'])
            a = nilai_a
            b = nilai_b
            c = 50

            e = 38

            no_split = 0

            no_header = 0

            header_array = ['No', 'Bidang', 'Grup Riset', 'Judul
Jurnal', 'Kata Judul Jurnal Hasil Filter']

            array_coba = ['list_1', 'list_2', 'list_3', 'list_4',
'list_5']

            no_array_coba = 0

            while no_split < e:
                while no_header < len(header_array):
                    array_tabel =
list(df1[f'{header_array[no_header]}'].loc[(df1['No'] >= a) &
(df1['No'] <= b)])
                    array_coba[no_header] = array_tabel
                    no_header += 1
                while no_array_coba < len(array_coba[0]):

```

```

        writer.writerow(
            [array_coba[0][no_array_coba],
array_coba[1][no_array_coba], array_coba[2][no_array_coba],
            array_coba[3][no_array_coba],
array_coba[4][no_array_coba]])
        no_array_coba += 1
        no_header = 0
        no_array_coba = 0
        a += 50
        b += 50
        no_split += 1

## pada bagian ini dilakukan pembagian dataset
## untuk dokumen ke 3 dengan nilai a sebagai nomor urutan awal
## sama dengan 21
## dan nilai b sebagai nomor urutan akhir sama dengan 30

        with
open(f'dataset_naive_bayes/data_training/data_training_{no_dok}.cs
v', 'w', newline='', encoding='ISO-8859-1') as file:
        writer = csv.writer(file)
        writer.writerow(['No', 'Bidang', 'Grup Riset', 'Judul
Jurnal', 'Kata Judul Jurnal Hasil Filter'])
        a = nilai_a
        b = nilai_b
        c = 50
        d = c - c

        e = 38

        no_split = 0

        no_header = 0

        header_array = ['No', 'Bidang', 'Grup Riset', 'Judul
Jurnal', 'Kata Judul Jurnal Hasil Filter']

        array_coba = ['list_1', 'list_2', 'list_3', 'list_4',
'list_5']

```

```

        array_coba_2 = ['list_1a', 'list_2b', 'list_3c',
'list_4d', 'list_5e']

        no_array_coba = 0

        no_array_coba_2 = 0

        while no_split < e:
            while no_header < len(header_array):
                array_tabel =
list(df1[f'{header_array[no_header]}'].loc[(df1['No'] < a) &
(df1['No'] > d)])
                array_tabel_2 =
list(df1[f'{header_array[no_header]}'].loc[(df1['No'] > b) &
(df1['No'] <= c)])
                array_coba[no_header] = array_tabel
                array_coba_2[no_header] = array_tabel_2
                no_header += 1
            while no_array_coba < len(array_coba[0]):
                writer.writerow(
                    [array_coba[0][no_array_coba],
array_coba[1][no_array_coba], array_coba[2][no_array_coba],
                    array_coba[3][no_array_coba],
array_coba[4][no_array_coba]])
                no_array_coba += 1
            while no_array_coba_2 < len(array_coba_2[0]):
writer.writerow([array_coba_2[0][no_array_coba_2],
array_coba_2[1][no_array_coba_2],
array_coba_2[2][no_array_coba_2],
array_coba_2[3][no_array_coba_2],
array_coba_2[4][no_array_coba_2]])
                no_array_coba_2 += 1
            no_header = 0
            no_array_coba = 0
            no_array_coba_2 = 0

```

```

        a += 50
        b += 50
        c += 50
        d += 50
        no_split += 1

## ----- akhir -----

# dilakukan pengecekan jika nilai b sebagai nomor urutan akhir
# sama dengan 50

    elif nilai_b == 50:

## pada bagian ini dilakukan pembagian dataset
## untuk dokumen ke 4 dengan nilai a sebagai nomor urutan awal
## sama dengan 31
## dan nilai b sebagai nomor urutan akhir sama dengan 40

        with
open(f'dataset_naive_bayes/data_testing/data_testing_{no_dok}.csv'
, 'w', newline='', encoding='ISO-8859-1') as file:
            writer = csv.writer(file)
            writer.writerow(['No', 'Bidang', 'Grup Riset', 'Judul
Jurnal', 'Kata Judul Jurnal Hasil Filter'])
            a = nilai_a
            b = nilai_b
            c = 50

            e = 38

            no_split = 0

            no_header = 0

            header_array = ['No', 'Bidang', 'Grup Riset', 'Judul
Jurnal', 'Kata Judul Jurnal Hasil Filter']

            array_coba = ['list_1', 'list_2', 'list_3', 'list_4',
'list_5']

```

```

        no_array_coba = 0

        while no_split < e:
            while no_header < len(header_array):
                array_tabel =
list(df1[f'{header_array[no_header]}'].loc[(df1['No'] >= a) &
(df1['No'] <= b)])
                array_coba[no_header] = array_tabel
                no_header += 1
            while no_array_coba < len(array_coba[0]):
                writer.writerow(
                    [array_coba[0][no_array_coba],
array_coba[1][no_array_coba], array_coba[2][no_array_coba],
                    array_coba[3][no_array_coba],
array_coba[4][no_array_coba]])
                no_array_coba += 1
            no_header = 0
            no_array_coba = 0
            a += 50
            b += 50
            no_split += 1

## pada bagian ini dilakukan pembagian dataset
## untuk dokumen ke 5 dengan nilai a sebagai nomor urutan awal
## sama dengan 41
## dan nilai b sebagai nomor urutan akhir sama dengan 50

        with
open(f'dataset_naive_bayes/data_training/data_training_{no_dok}.cs
v', 'w', newline='', encoding='ISO-8859-1') as file:
            writer = csv.writer(file)
            writer.writerow(['No', 'Bidang', 'Grup Riset', 'Judul
Jurnal', 'Kata Judul Jurnal Hasil Filter'])
            a = nilai_a
            c = 50
            d = c - c

            e = 38

```

```

no_split = 0

no_header = 0

header_array = ['No', 'Bidang', 'Grup Riset', 'Judul
Jurnal', 'Kata Judul Jurnal Hasil Filter']

array_coba = ['list_1', 'list_2', 'list_3', 'list_4',
'list_5']

no_array_coba = 0

while no_split < e:
    while no_header < len(header_array):
        array_tabel =
list(df1[f'{header_array[no_header]}'].loc[(df1['No'] < a) &
(df1['No'] > d)])
        array_coba[no_header] = array_tabel
        no_header += 1
        while no_array_coba < len(array_coba[0]):
            writer.writerow(
                [array_coba[0][no_array_coba],
array_coba[1][no_array_coba], array_coba[2][no_array_coba],
                array_coba[3][no_array_coba],
array_coba[4][no_array_coba]])
            no_array_coba += 1
        no_header = 0
        no_array_coba = 0
        a += 50
        d += 50
        no_split += 1

no_jumlah_data_bagi = 0

jumlah_data_bagi = 5

nilai_a = 1

```

```
nilai_b = 10

## memanggil fungsi pembagian dataset dengan
## k-fold cross validation dengan nilai k = 5
## maka variabel jumlah_data_bagi sama dengan 5
## sehingga dilakukan 5 kali perulangan
## serta nilai awal a sama dengan 1 dan b sama dengan 10

while no_jumlah_data_bagi < jumlah_data_bagi:
    no_dokumen = no_jumlah_data_bagi + 1
    split_data(no_dokumen, nilai_a, nilai_b)
    nilai_a += 10
    nilai_b += 10
    no_jumlah_data_bagi += 1
    print('iterasi')
```

#### Lampiran 6 Potongan Kode Pembagian *Dataset*

## Lampiran 7: Potongan Kode *Weighting Data Training*

skripsi\_filter/weighting.py

```
## ----- Fungsi Weighting -----  
  
def weighting(no_dokumen):  
  
## ----- Melakukan Perhitungan TF -----  
    with  
open(f'hasil_weighting/data_training_{no_dokumen}/tf_{no_dokumen}.  
csv', 'w', newline='',  
      encoding='ISO-8859-1') as file:  
    writer = csv.writer(file)  
  
    df1 =  
pd.read_csv(f'dataset/data_training/data_training_{no_dokumen}.csv  
, encoding='ISO-8859-1')  
  
    kata = list(df1['Kata Judul Jurnal Hasil Filter'])  
  
    kata_dokumen = []  
  
    array_header = ['Grup Riset', 'Bidang', 'Judul']  
  
    for i in kata:  
        if i not in kata_dokumen:  
            kata_dokumen.append(i)  
            array_header.append(i)  
  
    writer.writerow(array_header)  
  
    no_kata = 0  
  
    hitung_tf = 0  
  
    while no_grup < len(grup_dokumen):  
        for data_bidang in grup_dokumen[no_grup]:  
            print(data_bidang)  
            buat_judul_dokumen = []  
            no = list(df1['No']).loc[(df1['Bidang'] ==
```

```

data_bidang]))

        no_awal = no[0]
        no_akhir = no[len(no) - 1]
        no_hitung = no_awal
        while no_hitung <= no_akhir:
            judul = list(df1['Judul
Jurnal'].loc[(df1['No'] == no_hitung)])
            buat_judul_dokumen.append(judul[0])
            array_row = []
            array_row.append(nama_grup_dokumen[no_grup])
            array_row.append(data_bidang)
            array_row.append(judul[0])
            kata = list(df1['Kata Judul Jurnal Hasil
Filter'].loc[(df1['No'] == no_hitung)])
            for i in range(len(kata_dokumen)):
                TF = kata.count(kata_dokumen[i])
                array_row.append(TF)
            writer.writerow(array_row)
            no_hitung += 1
        print('')
        print(len(buat_judul_dokumen))
        print('')
        no_grup += 1

## ----- Melakukan Perhitungan DF -----

        with
open(f'hasil_weighting/data_training_{no_dokumen}/df_{no_dokumen}.
csv', 'w', newline='', encoding='ISO-8859-1') as file:
            writer = csv.writer(file)
            writer.writerow(['kata', 'df'])

            df2 =
pd.read_csv(f'dataset/data_training/data_training_{no_dokumen}.csv
', encoding='ISO-8859-1')

            print('')

            katanya = []

```

```

while no_grup < len(grup_dokumen):
    for data_bidang in grup_dokumen[no_grup]:
        print(data_bidang)
        no = list(df1['No'].loc[(df1['Bidang'] ==
data_bidang)])
        no_awal = no[0]
        no_akhir = no[len(no) - 1]
        no_hitung = no_awal
        while no_hitung <= no_akhir:
            kata = list(df1['Kata Judul Jurnal Hasil
Filter'].loc[(df1['No'] == no_hitung)])
            kata_pisah = []
            for data_kata in kata:
                if data_kata not in kata_pisah:
                    kata_pisah.append(data_kata)
            print(kata_pisah)
            for a in kata_pisah:
                katanya.append(a)
            no_hitung += 1
        no_grup += 1

print('')

print(katanya)

print('')

str3 = []

for i in katanya:
    if i not in str3:
        str3.append(i)
for i in range(0, len(str3)):
    print(str3[i])
    print(katanya.count(str3[i]))
    writer.writerow([str3[i], katanya.count(str3[i])])

## ----- Melakukan Perhitungan IDF -----

```

```

with
open(f'hasil_weighting/data_training_{no_dokumen}/idf_{no_dokumen}
.csv', 'w', newline='', encoding='ISO-8859-1') as file:
    writer = csv.writer(file)
    writer.writerow(['kata', 'df', 'idf'])

df2 =
pd.read_csv(f'dataset/data_training/data_training_{no_dokumen}.csv
', encoding='ISO-8859-1')

judul = list(df2['Judul Jurnal'])

judul_dokumen = []

for i in judul:
    if i not in judul_dokumen:
        judul_dokumen.append(i)

jumlah_dokumen = len(judul_dokumen)

print('')

katanya = []

while no_grup < len(grup_dokumen):
    for data_bidang in grup_dokumen[no_grup]:
        print(data_bidang)
        no = list(df2['No'].loc[(df2['Bidang'] ==
data_bidang)])
        no_awal = no[0]
        no_akhir = no[len(no) - 1]
        no_hitung = no_awal
        while no_hitung <= no_akhir:
            kata = list(df2['Kata Judul Jurnal Hasil
Filter'].loc[(df2['No'] == no_hitung)])
            kata_pisah = []
            for data_kata in kata:
                if data_kata not in kata_pisah:

```

```

        kata_pisah.append(data_kata)
    print(kata_pisah)
    for a in kata_pisah:
        katanya.append(a)
    no_hitung += 1
no_grup += 1

print('')

print(katanya)

print('')

str3 = []

for i in katanya:
    if i not in str3:
        str3.append(i)
for i in range(0, len(str3)):
    print(str3[i])
    print(katanya.count(str3[i]))
    hitung_idf = math.log10(jumlah_dokumen /
katanya.count(str3[i]))
    writer.writerow([str3[i], katanya.count(str3[i]),
hitung_idf])

## ----- Melakukan Perhitungan TF-IDF -----

    with
open(f'hasil_weighting/data_training_{no_dokumen}/tf_idf_{no_dokum
en}.csv', 'w', newline='',
        encoding='ISO-8859-1') as file:
    writer = csv.writer(file)

    df1 =
pd.read_csv(f'dataset/data_training/data_training_{no_dokumen}.csv
', encoding='ISO-8859-1')

    df2 =

```

```

pd.read_csv(f'hasil_weighting/data_training_{no_dokumen}/idf_{no_dokumen}.csv')

kata = list(df1['Kata Judul Jurnal Hasil Filter'])

kata_dokumen = []

array_header = ['Grup Riset', 'Bidang', 'Judul']

for i in kata:
    if i not in kata_dokumen:
        kata_dokumen.append(i)
        array_header.append(i)

writer.writerow(array_header)

no_kata = 0

hitung_tf = 0

idf_1 = list(df2['kata'])
idf_2 = list(df2['df'])
idf_3 = list(df2['idf'])

while no_grup < len(grup_dokumen):
    for data_bidang in grup_dokumen[no_grup]:
        print(data_bidang)
        buat_judul_dokumen = []
        no = list(df1['No']).loc[(df1['Bidang'] ==
data_bidang)])
        no_awal = no[0]
        no_akhir = no[len(no) - 1]
        no_hitung = no_awal
        while no_hitung <= no_akhir:
            judul = list(df1['Judul
Jurnal']).loc[(df1['No'] == no_hitung)])
            buat_judul_dokumen.append(judul[0])
            array_row = []
            array_row.append(nama_grup_dokumen[no_grup])

```

```

        array_row.append(data_bidang)
        array_row.append(judul[0])
        kata = list(df1['Kata Judul Jurnal Hasil
Filter'].loc[(df1['No'] == no_hitung)])
        for i in range(len(kata_dokumen)):
            index_kata = idf_1.index(kata_dokumen[i])
            idf = idf_3[index_kata]
            perhitungan = idf *
kata.count(kata_dokumen[i])
            array_row.append(perhitungan)
            writer.writerow(array_row)
            no_hitung += 1
        no_grup += 1
## ----- Panggil Fungsi Weighting -----
no_dokumen = 0
total_dokumen = 5

while no_dokumen < total_dokumen:
    nomor = no_dokumen + 1
    weighting(nomor)
    no_dokumen += 1

```

### Lampiran 7 Potongan Kode *Weighting Data Training*

## Lampiran 8: Potongan Kode Klasifikasi *Naive Bayes*

```
skripsi_filter/naive_bayes.php

## ----- Fungsi Naive Bayes -----

def naive_bayes(no_dokumen):

## ----- Hitung NB Training -----

    with
open(f'hasil_naive_bayes/data_{no_dokumen}/nilai_nb_training_{no_dokumen}.csv', 'w', newline='', encoding='ISO-8859-1') as file:
    writer = csv.writer(file)

    df1 =
pd.read_csv(f'hasil_weighting/data_training_{no_dokumen}/tf_idf_{no_dokumen}.csv', encoding='ISO-8859-1')

    df2 =
pd.read_csv(f'dataset/data_training/data_training_{no_dokumen}.csv', encoding='ISO-8859-1')

    kata = list(df2['Kata Judul Jurnal Hasil Filter'])

    kata_dokumen = []

    array_header = ['Grup Riset', 'Bidang']

    for i in kata:
        if i not in kata_dokumen:
            kata_dokumen.append(i)
            array_header.append(i)

    writer.writerow(array_header)

    no_kata = 0

    hitung_kata_muncul = 0

    jumlah_kata_muncul = 0
```

```

jumlah_kata_unik = len(kata_dokumen)

hitung_kata = 0

hitung_naive = 0

while no_grup < len(grup_dokumen):
    while no_bidang < len(grup_dokumen[no_grup]):
        print(grup_dokumen[no_grup][no_bidang])
        print('')
        array_row = []
        array_row.append(nama_grup_dokumen[no_grup])
        array_row.append(grup_dokumen[no_grup][no_bidang])
        for data_kata in kata_dokumen:
            daftar_kata =
list(df1[data_kata].loc[(df1['Bidang'] ==
grup_dokumen[no_grup][no_bidang])])
            for tf_kata in daftar_kata:
                jumlah_kata_muncul = jumlah_kata_muncul +
tf_kata

            for data_kata in kata_dokumen:
                print(data_kata)
                daftar_kata =
list(df1[data_kata].loc[(df1['Bidang'] ==
grup_dokumen[no_grup][no_bidang])])
                print(daftar_kata)
                for tf_kata in daftar_kata:
                    hitung_kata_muncul = hitung_kata_muncul +
tf_kata

                    print(hitung_kata_muncul)
                    print(jumlah_kata_muncul)
                    print(jumlah_kata_unik)
                    hitung_naive = (hitung_kata_muncul + 1) /
(jumlah_kata_muncul + jumlah_kata_unik)
                    print(hitung_naive)
                    array_row.append(hitung_naive)
                    hitung_kata_muncul = 0
                    print('')

```

```

        writer.writerow(array_row)
        jumlah_kata_muncul = 0
        no_bidang += 1
    no_bidang = 0
    no_grup += 1

## ----- Hitung Hasil Prediksi -----

    with
open(f'hasil_naive_bayes/data_{no_dokumen}/hasil_prediksi_{no_dokumen}.csv', 'w', newline='', encoding='ISO-8859-1') as file:
        writer = csv.writer(file)

        df1 =
pd.read_csv(f'dataset/data_testing/data_testing_{no_dokumen}.csv',
encoding='ISO-8859-1')
        df2 =
pd.read_csv(f'hasil_naive_bayes/data_{no_dokumen}/nilai_nb_training_{no_dokumen}.csv', encoding='ISO-8859-1')
        df3 =
pd.read_csv(f'hasil_weighting/data_training_{no_dokumen}/tf_idf_{no_dokumen}.csv', encoding='ISO-8859-1')

        judul_dokumen = []

        while no_grup < len(grup_dokumen):
            for data_bidang in grup_dokumen[no_grup]:
                print(data_bidang)
                buat_judul_dokumen = []
                no = list(df1['No'].loc[(df1['Bidang'] ==
data_bidang)])
                no_awal = no[0]
                no_akhir = no[len(no) - 1]
                no_hitung = no_awal
                while no_hitung <= no_akhir:
                    judul = list(df1['Judul
Jurnal'].loc[(df1['No'] == no_hitung)])
                    buat_judul_dokumen.append(judul[0])
                    no_hitung += 1

```

```

        print(len(buat_judul_dokumen))
        judul_dokumen.extend(buat_judul_dokumen)
        no_grup += 1

print('')

print(len(judul_dokumen))

no_judul = 0

no_grup = 0

array_header = ['Judul']

while no_grup < len(grup_dokumen):
    while no_bidang < len(grup_dokumen[no_grup]):
array_header.append(grup_dokumen[no_grup][no_bidang])
        no_bidang += 1
        no_bidang = 0
        no_grup += 1

writer.writerow(array_header)

hitung_total_judul = 0

judul_hitung = list(df3['Judul'])

judul_hitung_dokumen = []

for z in judul_hitung:
    judul_hitung_dokumen.append(z)
for z in range(0, len(judul_hitung_dokumen)):
    hitung_total_judul = hitung_total_judul + 1

print(hitung_total_judul)

nilai_prob_bidang = []

```

```

hitung_judul_bidang = 0

no_grup = 0

while no_grup < len(grup_dokumen):
    while no_bidang < len(grup_dokumen[no_grup]):
        judul_training =
list(df3['Judul'].loc[(df3['Bidang']) ==
grup_dokumen[no_grup][no_bidang]])
        judul_training_dokumen = []
        for data_judul_training in judul_training:
            if data_judul_training not in
judul_training_dokumen:

judul_training_dokumen.append(data_judul_training)
            for data_judul_training in range(0,
len(judul_training_dokumen)):
                hitung_judul_bidang = hitung_judul_bidang + 1
                nilai_prob = hitung_judul_bidang /
hitung_total_judul
                nilai_prob_bidang.append(nilai_prob)
                hitung_judul_bidang = 0
                no_bidang += 1
            no_bidang = 0
            no_grup += 1

no_grup = 0

hitung_prediksi = 0

hitung_prob_kata = 1

print(nilai_prob_bidang)

for data_judul in judul_dokumen:
    array_row = []
    array_row.append(data_judul)
    kata_dokumen = list(df1['Kata Judul Jurnal Hasil
Filter'].loc[(df1['Judul Jurnal'] == data_judul)])

```

```

        while no_grup < len(grup_dokumen):
            for a in grup_dokumen[no_grup]:
                print(f'Judul Dokumen: {data_judul}')
                print(f'Bidang Dokumen: {a}')
                for data_kata_dokumen in kata_dokumen:
                    print(f'Kata Dokumen:
{data_kata_dokumen}')
                    try:
                        nilai_nb =
list(df2[data_kata_dokumen].loc[df2['Bidang'] == a])
                        print(f'Nilai Nb Kata :
{nilai_nb[0]}')
                        hitung_prob_kata = hitung_prob_kata *
nilai_nb[0]
                    except:
                        print('data kosong')
                    if hitung_prob_kata == 1:
                        hitung_prob_kata = 0
                        hitung_prediksi = nilai_prob_bidang[0] *
hitung_prob_kata
                        print(f'nilai prob bidang:
{nilai_prob_bidang[0]}')
                        print(f'Prob Kata: {hitung_prob_kata}')
                        print(f'Hasil Prediksi: {hitung_prediksi}')
                        array_row.append(hitung_prediksi)
                        hitung_prob_kata = 1
                        print('')
                    no_grup += 1
                writer.writerow(array_row)
            no_grup = 0

## ----- Tampilkan Bidang Hasil Prediksi -----

        with
open(f'hasil_naive_bayes/data_{no_dokumen}/prediksi_{no_dokumen}.c
sv', 'w', newline='', encoding='ISO-8859-1') as file:
            writer = csv.writer(file)
            writer.writerow(['Judul', 'Nama Bidang', 'Nilai
Prediksi'])

```

```

df1 =
pd.read_csv(f'hasil_naive_bayes/data_{no_dokumen}/hasil_prediksi_{
no_dokumen}.csv', encoding='ISO-8859-1')

grup_dokumen = [SistemCerdas, ComputerNetwork,
ComputerVision, InformationSystem, MultimediaDanGame]

nama_grup_dokumen = ['Sistem Cerdas', 'Computer Network',
'Computer Vision', 'Information System',
'Multimedia dan Game']

no_grup = 0

judul_dokumen = list(df1['Judul'])

nilai_terbesar = 0

bidang = 'bidang'

for data_judul in judul_dokumen:
    print(data_judul)
    while no_grup < len(grup_dokumen):
        for data_bidang in grup_dokumen[no_grup]:
            print(data_bidang)
            try:
                nilai_prediksi_dokumen =
list(df1[data_bidang].loc[(df1['Judul'] == data_judul)])
                print(nilai_prediksi_dokumen[0])
                if nilai_prediksi_dokumen[0] >
nilai_terbesar:
                    nilai_terbesar =
nilai_prediksi_dokumen[0]
                    bidang = data_bidang
            except:
                print('data kosong')
        no_grup += 1
    print('')
    print(nilai_terbesar)

```

```
print(bidang)
print('')
writer.writerow([data_judul, bidang, nilai_terbesar])
nilai_terbesar = 0
no_grup = 0

## ----- Panggil Fungsi Naive Bayes -----

no_dokumen = 0
total_dokumen = 5

while no_dokumen < total_dokumen:
    nomor = no_dokumen + 1
    naive_bayes(nomor)
    no_dokumen += 1
```

### Lampiran 8 Potongan Kode Klasifikasi *Naive Bayes*

## Lampiran 9: Potongan Kode Pengujian Model Klasifikasi *Naive Bayes*

skripsi\_filter/confusion\_matrix.py

```
## ----- Fungsi Confusion Matrix -----

def confusion_matrix(no_dokumen):

## memasukkan data bidang penelitian hasil klasifikasi
## ke dalam variabel array prediksi

    for data_prediksi_bidang_dokumen in prediksi_bidang_dokumen:
        bidang_prediksi = data_prediksi_bidang_dokumen
        prediksi.append(bidang_prediksi)

    print(f'Jumlah Hasil Prediksi: {len(prediksi)}')

    asli = []

## memasukkan data bidang penelitian pada data testing
## ke dalam variabel array asli

    while no_grup < len(grup_dokumen):
        for data_bidang in grup_dokumen[no_grup]:
            print(data_bidang)
            buat_bidang_dokumen = []
            no = list(df2['No'].loc[(df2['Bidang'] ==
data_bidang)])
            no_awal = no[0]
            no_akhir = no[len(no) - 1]
            no_hitung = no_awal
            while no_hitung <= no_akhir:
                bidang = list(df2['Bidang'].loc[(df2['No'] ==
no_hitung)])
                buat_bidang_dokumen.append(bidang[0])
                no_hitung += 1
            print(len(buat_bidang_dokumen))
            asli.extend(buat_bidang_dokumen)
        no_grup += 1
    print('')
```

```

print(f'Jumlah Data Asli: {len(asli)}')

print('')

print(metrics.confusion_matrix(asli, prediksi))

print('')

## Menghitung precision, recall, accuracy menggunakan
## library sklearn

print(metrics.classification_report(asli, prediksi,
digits=3))

print('')

## ----- Panggil Fungsi Pengujian Model Naive

no_dokumen = 0
total_dokumen = 5

while no_dokumen < total_dokumen:
    nomor = no_dokumen + 1
    confusion_matrix(nomor)
    no_dokumen += 1

```

Lampiran 9 Potongan Kode Pengujian Model Klasifikasi *Naive Bayes*

## Lampiran 10: Potongan Kode *Crawling* Judul Publikasi Dosen

data\_dosen\_sistem/data\_dosen\_2.py

```
with
open('../..//Program_Klasifikasi/data_dosen_sistem/data/data_judul_
dosen/data_judul_dosen.csv', 'w', newline='') as file:
    writer = csv.writer(file)
    writer.writerow(['No Dosen', 'Nama Dosen', 'No Judul', 'Judul
Jurnal', 'Kata Judul Jurnal Hasil Filter'])

    while no_dosen_dokumen < len(nama_dosen_dokumen):
        print(f'Nama Dosen:
{name_dosen_dokumen[no_dosen_dokumen]}')

        nomornya_dosen = list(df1['No Dosen'].loc[(df1['Nama
Dosen'] == nama_dosen_dokumen[no_dosen_dokumen])])
        no_dosen = nomornya_dosen[0]

        mulai = 0
        while mulai <= 100:
            url_2 =
f'{link_dosen_dokumen[no_dosen_dokumen]}&cstart={mulai}&pagesize=1
00'

            response = requests.get(url_2, headers=headers_1)
            print(response.status_code)

            if response.status_code == 200:
                soup = BeautifulSoup(response.content, 'lxml')
                for td in soup.select('td.gsc_a_t'):
                    try:
                        print('-----')
                        print('-----')

                        judul_jurnal =
td.select('a')[0].get_text()

                        judul = stemmer.stem(judul_jurnal)
                        judul = (re.sub('[:,.]', "",
```

```

judul_jurnal))

        judul = judul.replace('[PDF][PDF]', '')
        judul = judul.replace('❖❖', '')
        judul = judul.replace('[PERNYATAAN][C]',
''))

        judul = judul.replace('[BUKU][B]', '')

        print(judul)

        kata_dasar = stemmer.stem(judul)
        judul_jurnal_split = kata_dasar.split()

        cek_judul_jurnal_split =
kata_dasar.split()

        no_split = 0

        no_split_cek = 0

        while no_split_cek <
len(cek_judul_jurnal_split):
            if
cek_judul_jurnal_split[no_split_cek] in filter_judul:
                judul_terverifikasi = 1
                while no_split <
len(judul_jurnal_split):
                    kata =
judul_jurnal_split[no_split]

                    panjang_data_kata = 1
                    panjang_array_kata =
len(filter_kata)

                    s = dict()

                    for i in
range(panjang_array_kata):
                        s[filter_kata[i]] = 1

```

```

                                for i in
range(panjang_data_kata):
                                if kata not in s.keys():
                                    print(kata)

array_no_dosen.append(no_dosen)

array_no.append(nomor_judul)

array_nama_dosen.append(nama_dosen_dokumen[no_dosen_dokumen])

array_judul.append(judul)

array_filter_kata.append(kata)

                                no_split += 1
                                no_split_cek = 0
                                no_split_cek += 1

                                if judul_terverifikasi == 1:
                                    nomor_judul += 1
                                    judul_terverifikasi = 0

                                except Exception as e:
                                    print('Data tidak terdeteksi')

                                while no_array_list_data < len(array_no):

writer.writerow([array_no_dosen[no_array_list_data],
array_nama_dosen[no_array_list_data],
array_no[no_array_list_data],

array_judul[no_array_list_data],
array_filter_kata[no_array_list_data]])
                                no_array_list_data += 1

                                no_array_list_data = 0

                                array_no_dosen = []
                                array_no = []

```

```
        array_nama_dosen = []
        array_judul = []
        array_filter_kata = []
    else:
        print('Belum bisa di crawling in my skin')
        no_dosen_dokumen = len(nama_dosen_dokumen)

    mulai += 100

no_dosen_dokumen += 1
```

#### Lampiran 10 Potongan Kode *Crawling* Judul Publikasi Dosen

## Lampiran 11: Potongan Kode Klasifikasi Judul Publikasi Dosen

```
data_dosen_sistem/naive_bayes.py
# ----- Hitung NB Training -----
--

while no_grup < len(grup_dokumen):
    while no_bidang < len(grup_dokumen[no_grup]):
        print(grup_dokumen[no_grup][no_bidang])
        print('')
        array_row = []
        array_row.append(nama_grup_dokumen[no_grup])
        array_row.append(grup_dokumen[no_grup][no_bidang])
        for data_kata in kata_dokumen:
            daftar_kata = list(df1[data_kata].loc[(df1['Bidang']
== grup_dokumen[no_grup][no_bidang])])
            for tf_kata in daftar_kata:
                jumlah_kata_muncul = jumlah_kata_muncul +
tf_kata
            for data_kata in kata_dokumen:
                print(data_kata)
                daftar_kata = list(df1[data_kata].loc[(df1['Bidang']
== grup_dokumen[no_grup][no_bidang])])
                print(daftar_kata)
                for tf_kata in daftar_kata:
                    hitung_kata_muncul = hitung_kata_muncul +
tf_kata
                print(hitung_kata_muncul)
                print(jumlah_kata_muncul)
                print(jumlah_kata_unik)
                hitung_naive = (hitung_kata_muncul + 1) /
(jumlah_kata_muncul + jumlah_kata_unik)
                print(hitung_naive)
                array_row.append(hitung_naive)
                hitung_kata_muncul = 0
                print('')
            writer.writerow(array_row)
            jumlah_kata_muncul = 0
            no_bidang += 1
        no_bidang = 0
```

```

no_grup += 1

# ----- Hitung Hasil Prediksi -----
--

for datanya_nama_dosen in nama_dosen_dokumen:
    print(datanya_nama_dosen)
    judul_dokumen = []
    no_dosen = list(df1['No Dosen'].loc[(df1['Nama Dosen'] ==
datanya_nama_dosen)])
    no = list(df1['No Judul'].loc[(df1['Nama Dosen'] ==
datanya_nama_dosen)])
    no_awal = no[0]
    no_akhir = no[len(no) - 1]
    no_hitung = no_awal
    while no_hitung <= no_akhir:
        judul = list(df1['Judul Jurnal'].loc[(df1['No Judul'] ==
no_hitung)])
        judul_dokumen.append(judul[0])
        print(judul[0].encode("utf-8"))
        no_hitung += 1
    for data_judul in judul_dokumen:
        nomor_judul += 1
        array_row = []
        array_row.append(no_dosen[0])
        array_row.append(datanya_nama_dosen)
        array_row.append(nomor_judul)
        array_row.append(data_judul)
        kata_dokumen = list(df1['Kata Judul Jurnal Hasil
Filter'].loc[(df1['Judul Jurnal'] == data_judul)])
        while no_grup < len(grup_dokumen):
            while no_bidang < len(grup_dokumen[no_grup]):
                print(f'Judul Dokumen: {data_judul.encode("utf-
8")}')

                print(f'Bidang Dokumen:
{grup_dokumen[no_grup][no_bidang]}')
                for data_kata_dokumen in kata_dokumen:
                    print(f'Kata Dokumen: {data_kata_dokumen}')
                    try:

```

```

        nilai_nb =
list(df2[data_kata_dokumen].loc[df2['Bidang'] ==
grup_dokumen[no_grup][no_bidang]])
        print(f'Nilai Nb Kata : {nilai_nb[0]}')
        hitung_prob_kata = hitung_prob_kata *
nilai_nb[0]
        except:
            print('data kosong')
            if hitung_prob_kata == 1:
                hitung_prob_kata = 0
                hitung_prediksi = nilai_prob_bidang[no_bidang] *
hitung_prob_kata
                print(f'nilai prob bidang:
{nilai_prob_bidang[no_bidang]}')
                print(f'Prob Kata: {hitung_prob_kata}')
                print(f'Hasil Prediksi: {hitung_prediksi}')
                array_row.append(hitung_prediksi)
                hitung_prob_kata = 1
                no_bidang += 1
                print('')
            no_bidang = 0
            no_grup += 1
            writer.writerow(array_row)
            no_grup = 0

# ----- Tampilkan Bidang Hasil Prediksi -----
--
for datanya_nama_dosen in nama_dosen_dokumen:
    nomor_dosen += 1
    print(datanya_nama_dosen)
    judul_dokumen = []
    no_dosen = list(df1['No Dosen'].loc[(df1['Nama Dosen'] ==
datanya_nama_dosen)])
    no = list(df1['No Judul'].loc[(df1['Nama Dosen'] ==
datanya_nama_dosen)])
    no_awal = no[0]
    no_akhir = no[len(no) - 1]
    no_hitung = no_awal
    while no_hitung <= no_akhir:

```

```

        judul = list(df1['Judul'].loc[(df1['No Judul'] ==
no_hitung)])
        judul_dokumen.append(judul[0])
        print(judul[0].encode("utf-8"))
        no_hitung += 1
    for data_judul in judul_dokumen:
        nomor_judul += 1
        print(data_judul.encode("utf-8"))
        while no_grup < len(grup_dokumen):
            for data_bidang in grup_dokumen[no_grup]:
                print(data_bidang)
                try:
                    nilai_prediksi_dokumen =
list(df1[data_bidang].loc[(df1['Judul'] == data_judul)])
                    print(nilai_prediksi_dokumen[0])
                    if nilai_prediksi_dokumen[0] >
nilai_terbesar:
                        nilai_terbesar =
nilai_prediksi_dokumen[0]
                    no_bidang_penelitian = list(df2['No
Bidang'].loc[(df2['Bidang Penelitian'] == data_bidang)])
                    no_grup_riset = list(df2['No Grup
Riset'].loc[(df2['Bidang Penelitian'] == data_bidang)])
                    bidang = data_bidang
                    grup = nama_grup_dokumen[no_grup]
                except:
                    print('data kosong')
                    no_grup += 1
            print('')
            print(nilai_terbesar)
            print(bidang)
            print('')
            writer.writerow([no_dosen[0], datanya_nama_dosen,
nomor_judul, data_judul, no_grup_riset[0], grup,
no_bidang_penelitian[0], bidang, nilai_terbesar, 0])
            nilai_terbesar = 0
            no_grup = 0

```

#### Lampiran 11 Potongan Kode Klasifikasi Judul Publikasi Dosen

## Lampiran 12: Potongan Kode Menentukan Kecenderungan Bidang

```
data_dosen_sistem/kecenderungan_bidang.py

for data_dosen in nama_dosen_dokumen:
    print(data_dosen)
    print('')
    jumlah_publicasi = list(df3['No Judul'].loc[(df3['Nama
Dosen'] == data_dosen)])
    nomor_dosen = list(df1['No Dosen'].loc[(df1['Nama Dosen'] ==
data_dosen)])
    no_awal = nomor_dosen[0]
    no_akhir = nomor_dosen[len(nomor_dosen) - 1]
    no_hitung = no_awal
    while no_hitung <= no_akhir:
        bidang = list(df1['Nama Bidang'].loc[(df1['No Dosen'] ==
no_hitung)])
        jumlah_bidang_muncul = list(df1['Jumlah Bidang
Muncul'].loc[(df1['No Dosen'] == no_hitung)])
        no_hitung += 1
        for data_jumlah_bidang in jumlah_bidang_muncul:
            if no_bidang_tertinggi < data_jumlah_bidang:
                no_bidang_tertinggi = data_jumlah_bidang
        print(f'No Bidang Tertinggi: {no_bidang_tertinggi}')
        index_kecenderungan_bidang =
jumlah_bidang_muncul.index(no_bidang_tertinggi)
        kecenderungan_bidang = bidang[index_kecenderungan_bidang]
        no_kecenderungan_bidang = list(df2['No
Bidang'].loc[(df2['Bidang Penelitian'] ==
kecenderungan_bidang)])
        print(f'Kecenderungan Bidang: {kecenderungan_bidang}')
        persentase = (no_bidang_tertinggi / len(jumlah_publicasi)) *
100
        writer.writerow([nomor_dosen[0], data_dosen,
no_kecenderungan_bidang[0], kecenderungan_bidang,
no_bidang_tertinggi, int(persentase)])
        no_bidang_tertinggi = 0
    print('')
```

## Lampiran 12 Potongan Kode Menentukan Kecenderungan Bidang

### Lampiran 13: Potongan Kode Menampilkan Daftar Profil Peneliti

server/index.js

```
app.get("/api/get", (req, res) => {
  const sqlInsert =
    "SELECT kecenderungan_bidang_dosen.ID_DOSEN AS ID_DOSEN, kecen
    derungan_bidang_dosen.NAMA_DOSEN AS NAMA_DOSEN, group_riset_1.GROU
    P_RISET_DOSEN_1 AS GROUP_RISET_ASAL_1, group_riset_2.GROUP_RISET_D
    OSEN_2 AS GROUP_RISET_ASAL_2, IFNULL(minat_group_riset_1.KECEN_GRO
    UP_RISET_DOSEN_1, '-
    ') AS KECEN_GROUP_RISET_1, IFNULL(minat_group_riset_1.PERSENTASE_G
    RUP_RISET_1, 0) AS PERSENTASE_KECEN_GROUP_RISET_1, IFNULL(minat_gr
    oup_riset_2.KECEN_GROUP_RISET_DOSEN_2, '-
    ') AS KECEN_GROUP_RISET_2, IFNULL(minat_group_riset_2.PERSENTASE_G
    RUP_RISET_2, 0) AS PERSENTASE_KECEN_GROUP_RISET_2, IFNULL(kecender
    ungan_bidang_dosen.KECEN_BIDANG_PENELITIAN_DOSEN, '-
    ') AS KECEN_BIDANG_PENELITIAN, IFNULL(kecenderungan_bidang_dosen.P
    ERSENTASE, 0) AS PERSENTASE_KECEN_BIDANG FROM kecenderungan_bidang
    _dosen INNER JOIN group_riset_1 ON kecenderungan_bidang_dosen.ID_D
    OSEN = group_riset_1.ID_DOSEN INNER JOIN group_riset_2 ON kecender
    ungan_bidang_dosen.ID_DOSEN = group_riset_2.ID_DOSEN INNER JOIN mi
    nat_group_riset_1 ON kecenderungan_bidang_dosen.ID_DOSEN = minat_g
    roup_riset_1.ID_DOSEN INNER JOIN minat_group_riset_2 ON kecenderun
    gan_bidang_dosen.ID_DOSEN = minat_group_riset_2.ID_DOSEN";
  db.query(sqlInsert, (err, result) => {
    res.send(result);
  });
});
```

### Lampiran 13 Potongan Kode Menampilkan Daftar Profil Peneliti

## Lampiran 14: Potongan Kode Menampilkan Kecenderungan Bidang

server/index.js

```
app.get("/api/get", (req, res) => {
  const sqlInsert =
    "SELECT kecenderungan_bidang_dosen.ID_DOSEN AS ID_DOSEN, kecen
    derungan_bidang_dosen.NAMA_DOSEN AS NAMA_DOSEN, group_riset_1.GROU
    P_RISET_DOSEN_1 AS GROUP_RISET_ASAL_1, group_riset_2.GROUP_RISET_D
    OSEN_2 AS GROUP_RISET_ASAL_2, IFNULL(minat_group_riset_1.KECEN_GRO
    UP_RISET_DOSEN_1, '-
    ') AS KECEN_GROUP_RISET_1, IFNULL(minat_group_riset_1.PERSENTASE_G
    RUP_RISET_1, 0) AS PERSENTASE_KECEN_GROUP_RISET_1, IFNULL(minat_gr
    oup_riset_2.KECEN_GROUP_RISET_DOSEN_2, '-
    ') AS KECEN_GROUP_RISET_2, IFNULL(minat_group_riset_2.PERSENTASE_G
    RUP_RISET_2, 0) AS PERSENTASE_KECEN_GROUP_RISET_2, IFNULL(kecender
    ungan_bidang_dosen.KECEN_BIDANG_PENELITIAN_DOSEN, '-
    ') AS KECEN_BIDANG_PENELITIAN, IFNULL(kecenderungan_bidang_dosen.P
    ERSENTASE, 0) AS PERSENTASE_KECEN_BIDANG FROM kecenderungan_bidang
    _dosen INNER JOIN group_riset_1 ON kecenderungan_bidang_dosen.ID_D
    OSEN = group_riset_1.ID_DOSEN INNER JOIN group_riset_2 ON kecender
    ungan_bidang_dosen.ID_DOSEN = group_riset_2.ID_DOSEN INNER JOIN mi
    nat_group_riset_1 ON kecenderungan_bidang_dosen.ID_DOSEN = minat_g
    roup_riset_1.ID_DOSEN INNER JOIN minat_group_riset_2 ON kecenderun
    gan_bidang_dosen.ID_DOSEN = minat_group_riset_2.ID_DOSEN";
  db.query(sqlInsert, (err, result) => {
    res.send(result);
  });
});
```

## Lampiran 14 Potongan Kode Menampilkan Kecenderungan Bidang

## Lampiran 15: Potongan Kode Menampilkan Judul Publikasi Dosen

server/index.js

```
app.get("/api/get/:id", (req, res) => {
  const id = req.params.id;
  const sqlUpdate =
    "SELECT (ROW_NUMBER() OVER (ORDER BY publikasi_dosen.JUDUL)) A
S Nomor, dosen.ID_DOSEN, dosen.NAMA_DOSEN, publikasi_dosen.JUDUL,
group_riset.GROUP_RISET, bidang_penelitian.BIDANG_PENELITIAN, publ
ikasi_dosen.NILAI_PREDIKSI FROM publikasi_dosen RIGHT JOIN dosen O
N publikasi_dosen.ID_DOSEN = dosen.ID_DOSEN LEFT JOIN bidang_penel
itian ON publikasi_dosen.ID_BIDANG_PENELITIAN = bidang_penelitian.
ID_BIDANG_PENELITIAN LEFT JOIN group_riset ON bidang_penelitian.ID
_GROUP_RISET = group_riset.ID_GROUP_RISET WHERE dosen.ID_DOSEN = ?
";
  db.query(sqlUpdate, id, (err, result) => {
    res.send(result);
  });
});
```

Lampiran 15 Potongan Kode Menampilkan Judul Publikasi Dosen

## Lampiran 16: Potongan Kode Menampilkan Grafik

server/index.js

```
app.get("/api/get_kecen_grup", (req, res) => {
  const sqlUpdate =
    "WITH group_1 AS ( SELECT gr.ID_GROUP_RISET, gr.GROUP_RISET, C
OUNT(*) AS total FROM minat_group_riset mgr JOIN group_riset gr ON
mgr.ID_GROUP_RISET_1 = gr.ID_GROUP_RISET GROUP BY gr.ID_GROUP_RIS
ET ), group_2 AS ( SELECT gr.ID_GROUP_RISET, gr.GROUP_RISET, COUNT
(*) AS total FROM minat_group_riset mgr JOIN group_riset gr ON mgr
.ID_GROUP_RISET_2 = gr.ID_GROUP_RISET GROUP BY gr.ID_GROUP_RISET )
SELECT g1.ID_GROUP_RISET, g1.GROUP_RISET, ROUND(g1.total/(SELECT
COUNT(minat_group_riset.ID_DOSEN) FROM minat_group_riset) * 100, 0
) AS persentase_mayor, ROUND(g2.total/(SELECT COUNT(minat_group_ri
set.ID_DOSEN) FROM minat_group_riset) * 100, 0) AS persentase_mino
r, g1.total AS mayor, g2.total AS minor FROM group_1 g1 JOIN group
_2 g2 ON g1.ID_GROUP_RISET = g2.ID_GROUP_RISET;";
  db.query(sqlUpdate, (err, result) => {
    res.send(result);
  });
});

app.get("/api/get_grup_1", (req, res) => {
  const sqlUpdate =
    "WITH bidang AS ( SELECT bp.ID_BIDANG_PENELITIAN, bp.BIDANG_PE
NELITIAN, COUNT(mbp.ID_DOSEN) AS total FROM minat_bidang_penelitia
n mbp RIGHT JOIN bidang_penelitian bp ON mbp.ID_BIDANG_PENELITIAN
= bp.ID_BIDANG_PENELITIAN RIGHT JOIN group_riset gr ON bp.ID_GROUP
_RISET= gr.ID_GROUP_RISET WHERE bp.ID_GROUP_RISET = 1 GROUP BY bp.
ID_BIDANG_PENELITIAN ) SELECT (ROW_NUMBER() OVER (ORDER BY b.ID_BI
DANG_PENELITIAN)) AS Nomor, b.ID_BIDANG_PENELITIAN, b.BIDANG_PENEL
ITIAN, b.total AS mayor, ROUND(b.total/(SELECT COUNT(minat_bidang_
penelitian.ID_DOSEN) FROM minat_bidang_penelitian) * 100, 0) AS pe
rsentase_mayor FROM bidang b;";
  db.query(sqlUpdate, (err, result) => {
    res.send(result);
  });
});

app.get("/api/get_grup_2", (req, res) => {
  const sqlUpdate =
    "WITH bidang AS ( SELECT bp.ID_BIDANG_PENELITIAN, bp.BIDANG_PE
NELITIAN, COUNT(mbp.ID_DOSEN) AS total FROM minat_bidang_penelitia
n mbp RIGHT JOIN bidang_penelitian bp ON mbp.ID_BIDANG_PENELITIAN
= bp.ID_BIDANG_PENELITIAN RIGHT JOIN group_riset gr ON bp.ID_GROUP
_RISET= gr.ID_GROUP_RISET WHERE bp.ID_GROUP_RISET = 2 GROUP BY bp.
ID_BIDANG_PENELITIAN ) SELECT (ROW_NUMBER() OVER (ORDER BY b.ID_BI
```

```

DANG_PENELITIAN)) AS Nomor, b.ID_BIDANG_PENELITIAN, b.BIDANG_PENEL
ITIAN, b.total AS mayor, ROUND(b.total/(SELECT COUNT(minat_bidang_
penelitian.ID_DOSEN) FROM minat_bidang_penelitian) * 100, 0) AS pe
rsentase_mayor FROM bidang b;";
    db.query(sqlUpdate, (err, result) => {
        res.send(result);
    });
});

app.get("/api/get_grup_3", (req, res) => {
    const sqlUpdate =
        "WITH bidang AS ( SELECT bp.ID_BIDANG_PENELITIAN, bp.BIDANG_PE
NELITIAN, COUNT(mbp.ID_DOSEN) AS total FROM minat_bidang_penelitia
n mbp RIGHT JOIN bidang_penelitian bp ON mbp.ID_BIDANG_PENELITIAN
= bp.ID_BIDANG_PENELITIAN RIGHT JOIN group_riset gr ON bp.ID_GROUP
_RISET= gr.ID_GROUP_RISET WHERE bp.ID_GROUP_RISET = 3 GROUP BY bp.
ID_BIDANG_PENELITIAN ) SELECT (ROW_NUMBER() OVER (ORDER BY b.ID_BI
DANG_PENELITIAN)) AS Nomor, b.ID_BIDANG_PENELITIAN, b.BIDANG_PENEL
ITIAN, b.total AS mayor, ROUND(b.total/(SELECT COUNT(minat_bidang_
penelitian.ID_DOSEN) FROM minat_bidang_penelitian) * 100, 0) AS pe
rsentase_mayor FROM bidang b;";
    db.query(sqlUpdate, (err, result) => {
        res.send(result);
    });
});

app.get("/api/get_grup_4", (req, res) => {
    const sqlUpdate =
        "WITH bidang AS ( SELECT bp.ID_BIDANG_PENELITIAN, bp.BIDANG_PE
NELITIAN, COUNT(mbp.ID_DOSEN) AS total FROM minat_bidang_penelitia
n mbp RIGHT JOIN bidang_penelitian bp ON mbp.ID_BIDANG_PENELITIAN
= bp.ID BIDANG PENELITIAN RIGHT JOIN group_riset gr ON bp.ID GROUP
_RISET= gr.ID_GROUP_RISET WHERE bp.ID_GROUP_RISET = 4 GROUP BY bp.
ID_BIDANG_PENELITIAN ) SELECT (ROW_NUMBER() OVER (ORDER BY b.ID_BI
DANG PENELITIAN)) AS Nomor, b.ID BIDANG PENELITIAN, b.BIDANG PENEL
ITIAN, b.total AS mayor, ROUND(b.total/(SELECT COUNT(minat_bidang_
penelitian.ID_DOSEN) FROM minat_bidang_penelitian) * 100, 0) AS pe
rsentase mayor FROM bidang b;";
    db.query(sqlUpdate, (err, result) => {
        res.send(result);
    });
});

app.get("/api/get_grup_5", (req, res) => {
    const sqlUpdate =
        "WITH bidang AS ( SELECT bp.ID_BIDANG_PENELITIAN, bp.BIDANG_PE
NELITIAN, COUNT(mbp.ID DOSEN) AS total FROM minat_bidang_penelitia

```

```

n mbp RIGHT JOIN bidang_penelitian bp ON mbp.ID_BIDANG_PENELITIAN
= bp.ID_BIDANG_PENELITIAN RIGHT JOIN group_riset gr ON bp.ID_GROUP
_RISET= gr.ID_GROUP_RISET WHERE bp.ID_GROUP_RISET = 5 GROUP BY bp.
ID_BIDANG_PENELITIAN ) SELECT (ROW_NUMBER() OVER (ORDER BY b.ID_BI
DANG_PENELITIAN)) AS Nomor, b.ID_BIDANG_PENELITIAN, b.BIDANG_PENEL
ITIAN, b.total AS mayor, ROUND(b.total/(SELECT COUNT(minat_bidang_
penelitian.ID_DOSEN) FROM minat_bidang_penelitian) * 100, 0) AS pe
rsentase_mayor FROM bidang b;";
    db.query(sqlUpdate, (err, result) => {
        res.send(result);
    });
});

```

### Lampiran 16 Potongan Kode Menampilkan Grafik

### Lampiran 17: Potongan Kode Menampilkan Daftar *Link Scholar* Dosen

```
server/index.js

app.get("/api/get_link_dosen", (req, res) => {
  const sqlUpdate =
    "SELECT kecenderungan_bidang_dosen.ID_DOSEN AS ID_DOSEN, kecen
    derungan_bidang_dosen.NAMA_DOSEN AS NAMA_DOSEN, group_riset_dosen.
    ID_GROUP_RISET_1 AS ID_GRUP_ASAL_1, group_riset_1.GROUP_RISET_DOSE
    N_1 AS GROUP_RISET_ASAL_1, group_riset_dosen.ID_GROUP_RISET_2 AS I
    D_GRUP_ASAL_2, group_riset_2.GROUP_RISET_DOSEN_2 AS GROUP_RISET_AS
    AL_2, dosen.LINK_SCHOLAR AS link_scholar_dosen FROM kecenderungan_
    bidang_dosen INNER JOIN group_riset_1 ON kecenderungan_bidang_dose
    n.ID_DOSEN = group_riset_1.ID_DOSEN INNER JOIN group_riset_2 ON ke
    cenderungan_bidang_dosen.ID_DOSEN = group_riset_2.ID_DOSEN INNER J
    OIN minat_group_riset_1 ON kecenderungan_bidang_dosen.ID_DOSEN = m
    inat_group_riset_1.ID_DOSEN INNER JOIN minat_group_riset_2 ON kece
    nderungan_bidang_dosen.ID_DOSEN = minat_group_riset_2.ID_DOSEN INN
    ER JOIN dosen ON kecenderungan_bidang_dosen.ID_DOSEN = dosen.ID_DO
    SEN INNER JOIN group_riset_dosen ON kecenderungan_bidang_dosen.ID_
    DOSEN = group_riset_dosen.ID_DOSEN;";
  db.query(sqlUpdate, (err, result) => {
    res.send(result);
  });
});
```

Lampiran 17 Potongan Kode Menampilkan Daftar *Link Scholar* Dosen

## Lampiran 18: Potongan Kode Menambahkan Data Dosen

```
server/index.js

app.post("/api/insert_data_dosen", (req, res) => {
  const idDosen = req.body.idDosen;
  const namaDosen = req.body.namaDosen;
  const linkScholar = req.body.linkScholar;

  const sqlInsert =
    "INSERT INTO dosen (ID_DOSEN, NAMA_DOSEN, LINK_SCHOLAR) VALUES
    (?, ?, ?);";
  db.query(sqlInsert, [idDosen, namaDosen, linkScholar], (err, res
ult) => {
    console.log(result);
  });
});

app.post("/api/insert_data_grup_riset_dosen", (req, res) => {
  const idGroupRisetDosen = req.body.idDosen;
  const idDosen = req.body.idDosen;
  const grupRisetAsal1 = req.body.grupRisetAsal1;
  const grupRisetAsal2 = req.body.grupRisetAsal2;

  const sqlInsert =
    "INSERT INTO group_riset_dosen (ID_GROUP_RISET_DOSEN, ID_DOSEN
, ID GROUP RISET 1, ID GROUP RISET 2) VALUES (?, ?, ?, ?);";
  db.query(
    sqlInsert,
    [idGroupRisetDosen, idDosen, grupRisetAsal1, grupRisetAsal2],
    (err, result) => {
      console.log(result);
    }
  );
});
```

## Lampiran 18 Potongan Kode Menambahkan Data Dosen

## Lampiran 19: Potongan Kode Mengedit Data Dosen

server/index.js

```
app.post("/api/update_data_dosen", (req, res) => {
  const idDosen = req.body.idDosen;
  const namaDosen = req.body.namaDosen;
  const linkScholar = req.body.linkScholar;
  const sqlUpdate =
    "UPDATE dosen SET NAMA_DOSEN = ?, LINK_SCHOLAR = ? WHERE ID_DO
    SEN = ?;";
  db.query(sqlUpdate, [namaDosen, linkScholar, idDosen], (err, res
    ult) => {
    if (err) console.log(err);
  });
});

app.post("/api/update_data_grup_riset_dosen", (req, res) => {
  const grupRisetAsal1 = req.body.grupRisetAsal1;
  const grupRisetAsal2 = req.body.grupRisetAsal2;
  const idDosen = req.body.idDosen;
  const sqlUpdate =
    "UPDATE group_riset_dosen SET ID_GROUP_RISET_1 = ?, ID_GROUP_R
    ISET_2 = ? WHERE ID_DOSEN = ?;";
  db.query(
    sqlUpdate,
    [grupRisetAsal1, grupRisetAsal2, idDosen],
    (err, result) => {
      if (err) console.log(err);
    }
  );
});
```

Lampiran 19 Potongan Kode Mengedit Data Dosen

## Lampiran 20: Potongan Kode Jalankan Klasifikasi Di Halaman *Website*

server/index.js

```
const { PythonShell } = require("python-shell");

app.get("/coba/hello", (req, res, next) => {
  let options = {
    mode: "text",
    pythonOptions: ["-u"],
    scriptPath: "../..//Program_Klasifikasi/data_dosen_sistem",
  };
  PythonShell.run("hello.py", options, function (err, result) {
    if (err) throw err;
    console.log("result: ", result.toString());
    res.send(result.toString());
  });
});

app.get("/coba/ambil_database", (req, res, next) => {
  let options = {
    mode: "text",
    pythonOptions: ["-u"],
    scriptPath: "../..//Program_Klasifikasi/data_dosen_sistem",
  };
  PythonShell.run("ambil_database.py", options, function (err, result) {
    if (err) throw err;
    console.log("result: ", result.toString());
    res.send(result.toString());
  });
});

app.get("/coba/data_dosen_2", (req, res, next) => {
  let options = {
    mode: "text",
    pythonOptions: ["-u"],
    scriptPath: "../..//Program_Klasifikasi/data_dosen_sistem",
  };
  PythonShell.run("data_dosen_2.py", options, function (err, result) {
    if (err) throw err;
    console.log("result: ", result.toString());
    res.send(result.toString());
  });
});

app.get("/coba/naive_bayes", (req, res, next) => {
```

```

let options = {
  mode: "text",
  pythonOptions: ["-u"],
  scriptPath: "../..//Program_Klasifikasi/data_dosen_sistem",
};
PythonShell.run("naive_bayes.py", options, function (err, result
) {
  if (err) throw err;
  console.log("result: ", result.toString());
  res.send(result.toString());
});
});

app.get("/coba/kecenderungan_bidang", (req, res, next) => {
  let options = {
    mode: "text",
    pythonOptions: ["-u"],
    scriptPath: "../..//Program_Klasifikasi/data_dosen_sistem",
  };
  PythonShell.run("kecenderungan_bidang.py", options, function (er
r, result) {
    if (err) throw err;
    console.log("result: ", result.toString());
    res.send(result.toString());
  });
});

app.get("/coba/insert_database", (req, res, next) => {
  let options = {
    mode: "text",
    pythonOptions: ["-u"],
    scriptPath: "../..//Program_Klasifikasi/data dosen sistem",
  };
  PythonShell.run("insert_database.py", options, function (err, re
sult) {
    if (err) throw err;
    console.log("result: ", result.toString());
    res.send(result.toString());
  });
});
});

```

## Lampiran 20 Potongan Kode Jalankan Klasifikasi Di Halaman *Website*



No. Skripsi : 654

**FORM VERIFIKASI**

**ABSTRAK BAHASA INGGRIS DAN TATA TULIS BUKU LAPORAN SKRIPSI**

**Nama Mahasiswa** : Sulthan Rafif NIM : 1741720158  
**Tanggal Ujian** : 16 Juli 2021  
**Judul** : SISTEM INFORMASI KLASIFIKASI BIDANG PENELITIAN DOSEN  
JURUSAN TEKNOLOGI INFORMASI POLITEKNIK NEGERI  
MALANG DENGAN METODE NAIVE BAYES

NO	BAGIAN YANG DIVERIFIKASI	NAMA VERIFIKATOR	TANGGAL VERIFIKASI	TTD
1	Abstrak Berbahasa Inggris	Faiz Ushbah Mubarak S.Pd, M.Pd	31/8/21	
2	Tata Tulis Buku Laporan Skripsi	Pramana Yoga Saputra, S.Kom., MMT.	31/08/2021	