

DAFTAR PUSTAKA

- Aritonang, L. W. (2020). Rancang Bangun Aplikasi Deteksi Kemiripan Dua Gambar Menggunakan Algoritma Ratcliff/Obershelp. *Journal of Computer System and Informatics (JoSYC)*, 1(3), 191-198.
- Armansyah, Kurniawan R. (2021). Model Prediksi Kelulusan Mahasiswa Tepat Waktu Menggunakan Metode Naïve Bayes.
- Billhaqqi, T. T. I., Wicaksono, G. W., & Aditya, C. S. K. (2020). Analisis Perbandingan Algoritma Rabin-Karp Dan Winnowing Dalam Penilaian Jawaban Otomatis. In *Prosiding SENTRA (Seminar Teknologi dan Rekayasa)* (No. 6, pp. 269-276).
- Faisal, M., Nugroho, F., M El Sulthan, M., Amini, F., Hariyadi, M. A., & Sedayu, A. (2020). Plagiarism detection using manber and winnowing algorithm. *International Journal of Advanced Science and Technology*, 29(6s), 2130-2136.
- Februariyanti, H., & Santoso, D. B. (2017). Hierarchical Agglomerative Clustering Untuk Pengelompokan Skripsi Mahasiswa.
- Hendini, A. 2016. Pemodelan UML Sistem Informasi Monitoring Penjualan Dan Stok Barang (Studi Kasus: Distro Zhezha Pontianak). *Jurnal Khatulistiwa Informatika*, VOL. IV, NO. 2, 107-116.
- Hidayat, W., Utami, E., & Hartanto, A. D. (2020, November). Effect of Stemming Nazief & Adriani on the Ratcliff/Obershelp algorithm in identifying level of similarity between slang and formal words. In *2020 3rd International Conference on Information and Communications Technology (ICOIACT)* (pp. 22-27). IEEE.
- Izzah, N., Yusliani, N., & Roodiah, D. (2022). Sistem Deteksi Kemiripan Teks Pada Berita Berbahasa Indonesia Menggunakan algoritma Ratcliff/Obershelp. *Jurnal Linguistik Komputasional*, 5(1), 1-6.
- Kharisman, O., Susanto, B., & Suwarno, S. (2018). IMPLEMENTASI ALGORITMA WINNOWING UNTUK MENDETEKSI KEMIRIPAN

- PADA DOKUKEN TEKS. *Jurnal Informatika*, 9(1).
- L. J. Yudhy, Alicia S, Agustinus J. Rancang Bangun Aplikasi Deteksi Kemiripan Dokumen Teks Menggunakan Algoritma Ratcliff/Obershelp. *E-journal Teknik Informatika Vol 11, No.1* (2017) ISSN: 2301-8364
- Nurdin, N., & Munthoha, A. (2017). Sistem Pendeteksian Kemiripan Judul Skripsi Menggunakan Algoritma WInnowing. *InfoTekJar: Jurnal Nasional Informatika dan Teknologi Jaringan*, 2(1).
- Pratomo, A., Irawan, A., & Risa, M. (2020, February). Similarity detection design using WInnowing Algorithm as an effort to apply green computing. In *Journal of Physics: Conference Series* (Vol. 1450, No. 1, p. 012065). IOP Publishing.
- Radili, A., & Sanjaya, S. (2018). Penerapan Metode WInnowing Fingerprint dan Naive Bayes untuk Pengelompokan Dokumen. *Jurnal CoreIT: Jurnal Hasil Penelitian Ilmu Komputer dan Teknologi Informasi*, 3(2)
- Rangkuti, Y.M. and Idrus, S.I.A. and Tarigan, 2021. "Pengantar Pemrograman Python". *Media Sains Indonesia*
- Riki, R., Edy, E., & Maryanto, M. (2019). Plagiarism Detection Application Uses WInnowing Algorithm with Synonym Recognition for Indonesian Text Documents. *Selangor Science & Technology Review (SeSTeR)*, 3(1),
- Setiawan, A. (2017). Implementasi Algoritma WInnowing Untuk Deteksi Kemiripan Judul Skripsi Studi Kasus Stmik Budidarma. *Informasi dan Teknologi Ilmiah (INTI)*, 4(2).
- Sunardi, S., Yudhana, A., & Mukaromah, I. A. (2018). Implementasi Deteksi Plagiarisme Menggunakan Metode N-Gram Dan Jaccard Similarity Terhadap Algoritma WInnowing. *Transmisi: Jurnal Ilmiah Teknik Elektro*, 20(3)
- Sunardi, S., Yudhana, A., & Mukaromah, I. A. (2017). Perancangan aplikasi deteksi plagiarisme karya ilmiah menggunakan algoritma winnowing.
- Usman, "Kajian Plagiarisme: Studi Perbandingan Hukum Islam dan Hukum Positif di Indonesia", *Jurna Hukum dan syariah Vol. 9 No.1* 2018

Yudhana, A., Djayali, A. D., & Sunardi, S. (2017). Sistem Deteksi Plagiarisme Dokumen Karya Ilmiah dengan Algoritma Pencocokan Pola. *Jurnal Rekayasa Teknologi Informasi (JURTI)*, 1(2)

Yusuf, B., Vivianie, S., Marsya, J. M., & Sofyan, Z. (2019, November). Analisis Perbandingan Algoritma Rabin-Karp dan Ratcliff/Obershelp untuk Menghitung Kesamaan Teks dalam Bahasa Indonesia. In *SEMINAR NASIONAL APTIKOM (SEMNASTIK) 2019* (pp. 61-69).



UNIVERSITAS ISLAM NEGERI
SUMATERA UTARA MEDAN

LAMPIRAN

Lampiran 1

Source Code

```
#import library
import os
from turtle import end_fill
from typing_extensions import Self
import docx
from docx import Document
import nltk # Library nltk
from nltk.tokenize import word_tokenize           from
nltk.corpus import stopwords

import string # Library string
import numpy as np
import prettytable
from prettytable import PrettyTable
from cProfile import label
from cgitb import text
from fileinput import filename
from tabnanny import filename_only
from tkinter import *
from tkinter import filedialog
from tkinter.ttk import *

def ekstrak_document(read_doc):
    #read_doc2 = docx.Document(doc2)
    document = ""
    for para in read_doc.paragraphs:
        document += para.text
    return document
```

```

def preprocessing(doc):
    lowercase = doc.lower()
    cleansing =
lowercase.translate(str.maketrans("", "", string.punctuation))
    tokens = word_tokenize(cleansing)
    stop_words = set(stopwords.words('indonesian'))
    filtering = [i for i in tokens if not i in stop_words]
    result = ("".join(filtering))
    return result

def preprocessing_ratclif(doc):
    lowercase = doc.lower()
    cleansing =
lowercase.translate(str.maketrans("", "", string.punctuation))
    tokens = word_tokenize(cleansing)
    stop_words = set(stopwords.words('indonesian'))
    filtering = [i for i in tokens if not i in stop_words]
    return filtering

def make_kgrams(s, t):      #membuat kgram
    grams = []
    start, end = 0, t
    while start < len(s) - t + 1:
        grams.append(s[start:end])
        start += 1
        end += 1
    return grams

def make_hash(grams):
    k = textareak.get("1.0", "end-1c")
    b = textareab.get("1.0", "end-1c")

```

```

k = int(k)
b = int(b)
hashpertama = []
rolling_hash = 0
for i in range(k):
    rolling_hash = (ord(grams[0][i]) * (b ** (k-(i+1))))
    hashpertama.append(rolling_hash)
return hashpertama

def rolling_hash(hashpertama,k_grams):
    list_rollinghash = [hashpertama]
    k = textareak.get("1.0","end-1c")
    b = textareab.get("1.0","end-1c")
    k = int(k)
    b = int(b)
    for i in range(len(k_grams)-1):
        hasher = (((list_rollinghash[i] -
(ord(k_grams[i][0]) * (b ** (k-1)))) * b) +
ord(k_grams[i+1][k-1]))
        list_rollinghash.append(hasher)
    return list_rollinghash

def window (k_grams,final_hash):
    w = textareaw.get("1.0","end-1c")
    w = int(w)
    a = -1
    prewindow = []
    for i in range(len(k_grams)):
        a += 1
        #print (final_hash[0+a : w+a],min(final_hash[0+a :
w+a]))
        prewindow.append(final_hash[0+a : w+a])
        if w+a >= len(k_grams):

```



```

File",
                                                    filetypes = (("Word
Document",
                                                    "*.docx*"),
                                                    ("all
files",
                                                    "*.*"))
    properties2.append(filename)
    textarea2.delete(1.0,END)
    raw_document2 = open(filename,"rb")
    read_doc2 = docx.Document(raw_document2)
    document2 = ekstrak_document(read_doc2)
    textarea2.insert(END,document2)
    return
def mainproses():
    dokumen1 = textarea1.get("1.0","end-1c")
    dokumen2 = textarea2.get("1.0","end-1c")
    k = textareak.get("1.0","end-1c")
    b = textareab.get("1.0","end-1c")
    w = textareaw.get("1.0","end-1c")
    result1 = preprocessing(dokumen1)
    result2 = preprocessing(dokumen2)
    k = int(k)
    b = int(b)
    w = int(w)
    k_grams1 = make_kgrams(result1, k)
    k_grams2 = make_kgrams(result2, k)
    kgram1.append(k_grams1)
    kgram2.append(k_grams2)

```



```

hashpertama1 = (sum(make_hash(k_grams1)))
hashpertama2 = (sum(make_hash(k_grams2)))
final_hash1 = rolling_hash(hashpertama1,k_grams1)
final_hash2 = rolling_hash(hashpertama2,k_grams2)
hash1.append(final_hash1)
hash2.append(final_hash2)

windows1 = window(k_grams1,final_hash1)
windows2 = window(k_grams2,final_hash2)
windowsvar1.append(windows1)
windowsvar2.append(windows2)

min_window1 = selected_hash(windows1)
min_window2 = selected_hash(windows2)
min_window1 = list(dict.fromkeys(min_window1))
min_window2 = list(dict.fromkeys(min_window2))
fingerprint1.append(min_window1)
fingerprint2.append(min_window2)

jaccard_similarity =
(len(set(min_window1).intersection(set(min_window2)))) /
(len(set(min_window1).union(set(min_window2)))) * 100
textarea3.delete(1.0,END)
textarea3.tag_configure("center", justify='center')
textarea3.insert(END, '{:0.2f}'.format(jaccard_similarity), '%
')
textarea3.insert(END, ' %')
textarea3.tag_add("center", "1.0", "end")

jumlah_karakter1 = len(result1)
jumlah_karakter2 = len(result2)
total_karakter = jumlah_karakter1 + jumlah_karakter2

```

```

same_word = []

i = 0
filtering1 = preprocessing_ratclif(dokumen1)
filtering2 = preprocessing_ratclif(dokumen2)
for i in filtering1 :
    if i in filtering2 :
        same_word.append(i)

kata_sama = np.array(same_word, dtype=list)
katasama.append(kata_sama)

total_subsequence = "".join(kata_sama)
RO_similarity = ((2*len(total_subsequence)) /
(total_karakter))*100
textarea4.delete(1.0,END)
textarea4.tag_configure("center", justify='center')
textarea4.insert(END, '{:0.2f}'.format(RO_similarity), '%')
textarea4.insert(END, ' %')
textarea4.tag_add("center", "1.0", "end")
raw_document1 = open(properties1[0], "rb")
raw_document2 = open(properties2[0], "rb")
read_doc1 = docx.Document(raw_document1)
read_doc2 = docx.Document(raw_document2)

prop1 = read_doc1.core_properties
prop2 = read_doc2.core_properties

tabelproperties = PrettyTable(["properties", "dokumen
pertama", "dokumen kedua"])
tabelproperties.add_row(["author",
prop1.author, prop2.author])

```

```

        tabelproperties.add_row(["last_modified_by",
prop1.last_modified_by, prop2.last_modified_by])
        tabelproperties.add_row(["last_modified_time",
prop1.modified, prop2.modified])
        tabelproperties.add_row(["created_time",
prop1.created, prop2.created])
        tabelproperties.add_row(["word_count",
len(dokumen1), len(dokumen2)])
        tabelproperties.add_row(["revision",
prop1.revision, prop2.revision])
        tabelproperties.add_row(["last_printed",
prop1.last_printed, prop2.last_printed])
        textarea5.delete(1.0,END)
        textarea5.insert(END,tabelproperties)
        return

def new_window():
    new_win = Toplevel()
    new_win.title ('Detail Output')
    new_win.iconbitmap('C:/Users/Rambee/Pictures/docx.ico')

    lebar = 1200
    tinggi = 850

    new_win.resizable(0,0)
    screen_width = new_win.winfo_screenwidth()
    screen_height = new_win.winfo_screenheight()

    newx = int((screen_width/2) - (lebar/2))
    newy = int((screen_height/2) - (tinggi/2) - 100)

    new_win.geometry(f"{lebar}x{tinggi}+{newx}+{newy}")

    label_judulnew = Label(new_win,text="DETAIL OUTPUT

```

```
PROGRAM",
    font=("Source Sans Pro", 15)
    ).place(x = 470, y = 10 )

    label_proses_dokumen1 = Label(new_win,text="Dokumen
Pertama",
    font=("Source Sans Pro", 15)
    ).place(x = 250, y = 50 )

    label_proses_dokumen1 = Label(new_win,text="Dokumen
Kedua",
    font=("Source Sans Pro", 15)
    ).place(x = 760, y = 50 )

    label_proses_kgram = Label(new_win,text="k-gram",
    font=("Source Sans Pro", 15)
    ).place(x = 560, y = 60 )

    label_proses_hash = Label(new_win,text="rolling hash",
    font=("Source Sans Pro", 15)
    ).place(x = 540, y = 210 )

    label_proses_window = Label(new_win,text="window",
    font=("Source Sans Pro", 15)
    ).place(x = 555, y = 360 )

    label_proses_fingerprint =
Label(new_win,text="fingerprint",
    font=("Source Sans Pro", 15)
    ).place(x = 545, y = 510 )

    label_proses_katasama = Label(new_win,text="kata yang
sama",
```

```
font=("Source Sans Pro", 15)
).place(x = 520, y = 510+150 )
```

```
framew1 = Frame(new_win)
framew1.place(x = 100, y = 100)
textareanew1 = Text(framew1, height=5, width=60)
textareanew1.pack(side=LEFT)
textareanew1.insert(END,kgram1[0])
```

```
framew2 = Frame(new_win)
framew2.place(x = 600, y = 100)
textareanew2 = Text(framew2, height=5, width=60)
textareanew2.pack(side=LEFT)
textareanew2.insert(END,kgram2[0])
```

```
framew3 = Frame(new_win)
framew3.place(x = 100, y = 250)
textareanew3 = Text(framew3, height=5, width=60)
textareanew3.pack(side=LEFT)
textareanew3.insert(END,hash1[0])
```

```
framew4 = Frame(new_win)
framew4.place(x = 600, y = 250)
textareanew4 = Text(framew4, height=5, width=60)
textareanew4.pack(side=LEFT)
textareanew4.insert(END,hash2[0])
```

```
framew5 = Frame(new_win)
framew5.place(x = 100, y = 400)
textareanew5 = Text(framew5, height=5, width=60)
textareanew5.pack(side=LEFT)
textareanew5.insert(END,windowvar1[0])
```

```
framenew6 = Frame(new_win)
framenew6.place(x = 600, y = 400)
textareanew6 = Text(framenew6, height=5, width=60)
textareanew6.pack(side=LEFT)
textareanew6.insert(END, windowsvar2[0])

framenew7 = Frame(new_win)
framenew7.place(x = 100, y = 550)
textareanew7 = Text(framenew7, height=5, width=60)
textareanew7.pack(side=LEFT)
textareanew7.insert(END, fingerprint1[0])

framenew8 = Frame(new_win)
framenew8.place(x = 600, y = 550)
textareanew8 = Text(framenew8, height=5, width=60)
textareanew8.pack(side=LEFT)
textareanew8.insert(END, fingerprint2[0])

framenew9 = Frame(new_win)
framenew9.place(x = 270, y = 700)
textareanew9 = Text(framenew9, height=5, width=80)
textareanew9.pack(side=LEFT)
textareanew9.tag_configure("center", justify='center')
textareanew9.insert(END, katasama[0])
textareanew9.tag_add("center", "1.0", "end")

button_out =
Button(new_win, text='KELUAR', command=new_win.destroy)
button_out.pack(side=BOTTOM)

root = Tk()
root.title("Winnowing-Ratcliff")
```

```

root.iconbitmap('C:/Users/Rambee/Pictures/docx.ico')

lebar = 1200
tinggi = 800

root.resizable(0,0)
screen_width = root.winfo_screenwidth()
screen_height = root.winfo_screenheight()

newx = int((screen_width/2) - (lebar/2))
newy = int((screen_height/2) - (tinggi/2) - 100)

root.geometry(f"{lebar}x{tinggi}+{newx}+{newy}")

properties1 = []
properties2 = []
kgram1 = []
kgram2 = []
hash1 = []
hash2 = []
windowsvar1 = []
windowsvar2 = []
fingerprint1 = []
fingerprint2 = []
katasama = []

label_judul = Label(root,text="  Identifikasi Tingkat
Kemiripan Dokumen Teks Menggunakan Fungsi Hash Pada
Algoritma Winnowing \n
Pattern Recognition pada Algoritma Ratcliff/Obershelp",
font=("Source Sans Pro", 15)
).place(x = 140, y = 10 )

button_explore1 = Button(root,

```

```
        text = "pilih dokumen 1",
        command = browsefiles1)
button_explore1.place(x = 340, y = 100 )

button_explore2= Button(root,
        text = "pilih dokumen 2",
        command = browsefiles2)
button_explore2.place(x = 760, y = 100 )

frame1 = Frame(root)
frame1.place(x = 190, y = 150)

textarea1 = Text(frame1, height=6, width=50)
textarea1.pack(side=LEFT)

frame2 = Frame(root)
frame2.place(x = 1200, y = 150)

textarea2 = Text(frame1, height=6, width=50)
textarea2.pack(side=LEFT)

labelwinnowing = Label(root,text="hasil identifikasi
menggunakan algoritma winnowing",
        font=("Source Sans Pro", 15)
).place(x = 380, y = 260 )

labelk = Label(root,text="masukkan nilai k=",
        font=("Source Sans Pro", 15)
).place(x = 380, y = 290 )

framek = Frame(root)
framek.place(x = 540, y =295)
```



```
textareak = Text(framek, height=1, width=3)
textareak.pack()

labelb = Label(root, text="b=",
               font=("Source Sans Pro", 15)
               ).place(x = 580, y = 290 )

frameb = Frame(root)
frameb.place(x = 610, y = 295)

textareab = Text(frameb, height=1, width=3)
textareab.pack()

labelw = Label(root, text="w=",
               font=("Source Sans Pro", 15)
               ).place(x = 650, y = 290 )

framew = Frame(root)
framew.place(x = 685, y = 295)

textareaw = Text(framew, height=1, width=3)
textareaw.pack()

frame3 = Frame(root)
frame3.place(x = 500, y = 330)

textarea3 = Text(frame3, height=1, width=25)
textarea3.tag_configure("center", justify='center')
textarea3.insert(END, "hasil algoritma winnowing")
textarea3.tag_add("center", "1.0", "end")
textarea3.pack()
```

```

labelratcliff = Label(root,text="hasil identifikasi
menggunakan algoritma Ratcliff/Obershelp",
                      font=("Source Sans Pro", 15)
                      ).place(x = 345, y = 360 )
frame4 = Frame(root)
frame4.place(x = 500, y =400)

textarea4 = Text(frame4, height=1, width=25)
textarea4.pack()
textarea4.insert(END,"hasil algoritma Ratcliff/Obershelp")

labelgetinfo = Label(root,text="Get Document Information",
                      font=("Source Sans Pro", 15)
                      ).place(x = 485, y = 430 )

frame5 = Frame(root)
frame5.place(x = 340, y =470)

textarea5 = Text(frame5, height=10, width=66)
textarea5.pack()
textarea5.insert(END,"          properties information dokumen
akan ditampilkan disini")

button_proses = Button(root,text = "Proses",command =
mainproses)
button_proses.place(x = 540, y =700)

button_detail = Button(root,text = "Detail
Output",command=new_window)
button_detail.place(x = 620, y =700)

root.mainloop()

```

Lampiran 2

Kartu Bimbingan Skripsi

KARTU BIMBINGAN SKRIPSI

Semester Gasal/Genap Tahun Akademik Genap /2021/2022

Nama : Yusuf Karim Rambe	Pembimbing I : Abdul Halim Hasugian M. Kom
NIM : 0701178069	Pembimbing II : Muhammad Siddik Hasibuan M. Kom
Prog. Studi : ILMU KOMPUTER	SK Pembimbing :

Judul Skripsi :

Identifikasi Forensik Tingkat Kemiripan Dokumen Teks
Menggunakan Fungsi Hash Pada Algoritma Windowing Dan
Pattern Recognition Pada Algoritma Rascliff /obershelp

P E R I O D E	PEMBIMBING I			PEMBIMBING II		
	Tgl.	Materi Bimbingan	Tanda Tangan	Tgl.	Materi Bimbingan	Tanda Tangan
I	24/10/2021	Perbaikan Rumusan masalah Bab 1 dan Latar belakang.		22/10/2021	Perbaikan format dan pembahasan masalah	
II	10/12/2021	Penambahan materi pada bab 2 dan penambahan referensi.		08/11/2021	Perbaikan judul skripsi dan perubahan rumusan masalah	
III	13/09/2022	Perbaikan format penulisan dan perbaikan TYPO.		07/12/2021	Penambahan materi bab 2 dan penambahan referensi.	
IV	24/05/2022	ACC Sempurna.		18/01/2022	Penambahan materi algoritma windowing dan Rascliff/obershelp	
V	25/07/2022	Bimbingan kode program		15/02/2022	Perbaikan flowchart dan penambahan skema algoritma	

VI	28/07/2022	Perbaikan format penulisan		21/04/2022	ACC Sempro.	
VII	1/08/2022	Bimbingan BAB 4		28/07/2022	Perbaikan format penulisan.	
VIII	8/08/2022	Bimbingan Bab 5		1/08/2022	Bimbingan BAB 4	
IX	15/08/2022	ACC Sidang munaqasah		8/08/2022	Bimbingan BAB 5.	
X		DAFTAR PRAKTIKUM		15/08/2022	ACC Sidang munaqasah	

Medan, 24 - 05 - 2022

An. Dekan

Ketua Jurusan/Program Studi

Ilmu Komputer

Ilka Letyra, M.kom

NIP. 198506042015031006

Catatan: Pada saat bimbingan, kartu ini harus diisi dan ditandatangani oleh pembimbing

Lampiran 3

Curriculum Vitae

DAFTAR RIWAYAT HIDUP

Curriculum Vitae



I. Data Pribadi

1. Nama : YUSUF KARIM RAMBE
2. Tempat/Tanggal Lahir : TANJUNG PASIR/31-05-200
3. Jenis Kelamin : PRIA
4. Agama : ISLAM
5. Status Pernikahan : LAJANG
6. Warga Negara : INDONESIA
7. Alamat KTP : Jl. Prof. Dr. Midian Sirait
RT/RW : 000
Kel/Desa : Lumban Manurung
Kecamatan : PARMAKSIAN
8. Alamat Sekarang : Jl. Besar Tanjung Anom, Kota Medan
9. Nomor Telepon/Wa : 082294087902
10. E-mail : ykrambe@gmail.com

II. Pendidikan Formal

Periode (Tahun)	Sekolah / Institusi / Universitas	Jurusan	Jenjang Pendidikan	IPK / UAN / RAPOR
2005 - 2011	MIN Lumban Gurning Porsea		SD	8.75
2011 - 2014	MTSN Lumban Gurning Porsea		SMP	8.65
2014 - 2017	SMAN 1 Siantar Narumonda	SMA / IPA	SMA/SMK	72.3
2017 - 2022	Universitas Islam Negeri Sumatera Utara	ILMU KOMPUTER	S1	3.47

III. Pendidikan Non Formal / Training – Seminar

Tahun	Lembaga / Instansi	Keterampilan
2022	Coursera	IT support google
2022	LKP CCE Institute	Toefl
2022	DQLAB	Junior Data Analyst

IV. Pengalaman Bekerja

Tahun	Lembaga / Instansi	Jabatan
2021	Kominfo Padang Sidempuan	Magang

V. Penguasaan Bahasa

NO	Bahasa	Kemampuan			
		membaca	menulis	berbicara	mendengar
1	DONESIA	AIK	AIK	AIK	AIK
2	GGRIS	AIK	AIK	AIK	AIK

VI. Lain-lain : - Hardskill : Install ulang komputer

- softskill : Microsoft Office, Bahasa C, Python, MySQL

Demikian CV ini saya buat dengan sebenarnya.

Medan, 05 November 2022

(Yusuf Karim Rambe)

UNIVERSITAS ISLAM NEGERI
SUMATERA UTARA MEDAN