

LAMPIRAN

LISTING PROGRAM

```
import tkinter
import cv2
import numpy
import pandas
from tkinter import ttk
from PIL import Image, ImageTk

import tensorflow as tf
from tensorflow import keras

class Aplikasi:
    def interface(self):
        self.window = tkinter.Tk()

        self.window.title("Aplikasi")

        self.window.columnconfigure(0, weight=1)
        self.window.columnconfigure(1, weight=1)
        self.window.rowconfigure(0, weight=4)
        self.window.rowconfigure(1, weight=1)

        self.frame1 = tkinter.Frame(self.window)
        self.frame2 = tkinter.Frame(self.window)
        self.frame3 = tkinter.Frame(self.window)
```

```

self.frame1.grid(row=0, column=0, rowspan=2)

self.frame2.grid(row=0, column=1, sticky="news", pady=10)

self.frame3.grid(row=1, column=1, sticky="s", pady=20)


self.labelframe1 = tkinter.LabelFrame(self.frame2, text="Hasil Deteksi",
labelanchor="n", padx=10, pady=10)

self.labelframe2 = tkinter.LabelFrame(self.frame2, text="Informasi", labelanchor="n",
padx=10, pady=10)


self.labelframe1.grid(row=0, column=0, sticky="ns", padx=10, pady=10)

self.labelframe2.grid(row=1, column=0, sticky="ns", padx=10, pady=10)


self.canvas = tkinter.Canvas(self.frame1,
height=self.vid.get(cv2.CAP_PROP_FRAME_HEIGHT), width=600)

self.canvas.grid()


self.label1 = tkinter.Label(self.labelframe1, text=self.plate_number, padx=20,
font=("Helvetica", 12, "bold"), width=40)

self.label1.pack(expand = True)


self.treeview = ttk.Treeview(self.labelframe2, columns = ('key', 'value'), show = ")

self.treeview.pack(expand = True)


self.button1 = tkinter.Button(self.frame3, text="Quit", font=("Helvetica", 10, "bold"),
width=40, height=2, command=self.cmd_quit)

self.button1.grid()


def __init__(self):

self.df = pandas.read_csv("data.csv")

```

```

self.vid = cv2.VideoCapture(0)
self.vid.set(cv2.CAP_PROP_FRAME_WIDTH, 512)
self.vid.set(cv2.CAP_PROP_FRAME_HEIGHT, 1280)

self.plate_number = "..."
self.interface()

self.count_interval = 0
self.startvideo()

self.window.mainloop()

def startvideo(self):
    ret, frame = self.vid.read()
    if ret:
        frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

        a,b,c = self.normalisasi(frame)

        self.photo = ImageTk.PhotoImage(image=Image.fromarray(frame))

        d,e,f,g = self.cariplat(frame, a, b)

        if f is not None:
            self.photo = ImageTk.PhotoImage(image=Image.fromarray(g))

            (thresh, img_plate_bw) = cv2.threshold(f, 0, 255, cv2.THRESH_BINARY +
cv2.THRESH_OTSU)

            kernel = cv2.getStructuringElement(cv2.MORPH_CROSS, (3,3))
            img_plate_bw = cv2.morphologyEx(img_plate_bw, cv2.MORPH_OPEN, kernel)

```

```
contours_plate, hierarchy = cv2.findContours(img_plate_bw, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)
```

```
index_chars_candidate = []
```

```
index_counter_contour_plate = 0
```

```
img_plate_rgb = cv2.cvtColor(f,cv2.COLOR_GRAY2BGR)
```

```
img_plate_bw_rgb = cv2.cvtColor(img_plate_bw, cv2.COLOR_GRAY2RGB)
```

```
for contour_plate in contours_plate:
```

```
    x_char,y_char,w_char,h_char = cv2.boundingRect(contour_plate)
```

```
    if h_char >= 40 and h_char <= 60 and w_char >=10:
```

```
        index_chars_candidate.append(index_counter_contour_plate)
```

```
cv2.rectangle(img_plate_rgb,(x_char,y_char),(x_char+w_char,y_char+h_char),(0,255,0),5)
```

```
cv2.rectangle(img_plate_bw_rgb,(x_char,y_char),(x_char+w_char,y_char+h_char),(0,255,0),
5)
```

```
    index_counter_contour_plate += 1
```

```
if index_chars_candidate == []:
```

```
    print('Karakter tidak tersegmentasi')
```

```
else:
```

```
    h,i,j = self.carikarakter(contours_plate, index_chars_candidate, f, g)
```

```
    self.photo = ImageTk.PhotoImage(image=Image.fromarray(i))
```

```
    self.count_interval = self.count_interval + 1
```

```
    if self.count_interval == 60:
```

```
        if len(j) > 2:
```

```
            self.kenalikarakter(contours_plate, img_plate_bw, j)
```

```
            self.update_info()
```

```
self.count_interval = 0
```

```
self.canvas.create_image(0, 0, image=self.photo, anchor="nw")
```

```
self.window.after(10, self.startvideo)
```

```
def cmd_quit(self):
```

```
    if self.vid.isOpened():
```

```
        self.vid.release()
```

```
    self.vid = None
```

```
    self.window.quit()
```

```
    self.window = None
```

```
def update_info(self):
```

```
    self.label1.config(text=self.plate_number)
```

```
    try:
```

```
        data = self.df.loc[self.df['plat']==self.plate_number]
```

```
        data = data.values[0]
```

```
        # index = data[0]
```

```
        plat = data[1]
```

```
        nama_pemilik = data[2]
```

```
        merek_kendaraan = data[3]
```

```
        tahun = data[4]
```

```
        self.treeview.delete(*self.treeview.get_children())
```

```
        self.treeview.insert(parent="", index=tkinter.END, values=('Nomor Plat', plat))
```

```
        self.treeview.insert(parent="", index=tkinter.END, values=('Nama Pemilik',  
nama_pemilik))
```

```

        self.treeview.insert(parent="", index=tkinter.END, values=('Merk Kendaraan',
merek_kendaraan))

        self.treeview.insert(parent="", index=tkinter.END, values=('Tahun', tahun))
except Exception as e:

    self.treeview.delete(*self.treeview.get_children())

    self.treeview.insert(parent="", index=tkinter.END, values=('Nomor Plat', "..."))
    self.treeview.insert(parent="", index=tkinter.END, values=('Nama Pemilik', "..."))
    self.treeview.insert(parent="", index=tkinter.END, values=('Merk Kendaraan', "..."))
    self.treeview.insert(parent="", index=tkinter.END, values=('Tahun', "..."))

```

```

def normalisasi(self, img):
    img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE,(20,20))
    img_opening = cv2.morphologyEx(img_gray, cv2.MORPH_OPEN, kernel)
    img_norm = img_gray - img_opening
    (thresh, img_norm_bw) = cv2.threshold(img_norm, 0, 255, cv2.THRESH_BINARY +
cv2.THRESH_OTSU)

    (thresh, img_without_norm_bw) = cv2.threshold(img_gray, 0, 255,
cv2.THRESH_BINARY + cv2.THRESH_OTSU)
    return (img_gray, img_without_norm_bw, img_norm_bw)

```

```

def cariplat(self, img_show_plate, img_gray, img_norm_bw):
    contours_vehicle, hierarchy = cv2.findContours(img_norm_bw, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE) # get the contour for every area

    index_plate_candidate = []
    index_counter_contour_vehicle = 0
    for contour_vehicle in contours_vehicle:
        x,y,w,h = cv2.boundingRect(contour_vehicle)
        aspect_ratio = w/h

```

```

if w >= 200 and aspect_ratio <= 4 :

    index_plate_candidate.append(index_counter_contour_vehicle)

    index_counter_contour_vehicle += 1

img_show_plate_bw = cv2.cvtColor(img_norm_bw, cv2.COLOR_GRAY2RGB)

if len(index_plate_candidate) == 0:

    print("Plat nomor tidak ditemukan")

    return (None, None, None)

elif len(index_plate_candidate) == 1:

    x_plate,y_plate,w_plate,h_plate =
cv2.boundingRect(contours_vehicle[index_plate_candidate[0]])

cv2.rectangle(img_show_plate,(x_plate,y_plate),(x_plate+w_plate,y_plate+h_plate),(0,255,0)
,5)

cv2.rectangle(img_show_plate_bw,(x_plate,y_plate),(x_plate+w_plate,y_plate+h_plate),(0,2
55,0),5)

    img_plate_gray = img_gray[y_plate:y_plate+h_plate, x_plate:x_plate+w_plate]

    img_plate_rgb = img_show_plate[y_plate:y_plate+h_plate, x_plate:x_plate+w_plate]

else:

    print('Dapat dua lokasi plat, pilih lokasi plat kedua')

    x_plate,y_plate,w_plate,h_plate =
cv2.boundingRect(contours_vehicle[index_plate_candidate[1]])

cv2.rectangle(img_show_plate,(x_plate,y_plate),(x_plate+w_plate,y_plate+h_plate),(0,255,0)
,5)

cv2.rectangle(img_show_plate_bw,(x_plate,y_plate),(x_plate+w_plate,y_plate+h_plate),(0,2
55,0),5)

    img_plate_gray = img_gray[y_plate:y_plate+h_plate, x_plate:x_plate+w_plate]

    img_plate_rgb = img_show_plate[y_plate:y_plate+h_plate, x_plate:x_plate+w_plate]

return (img_show_plate, img_show_plate_bw, img_plate_gray, img_plate_rgb)

```

```

def carikarakter(self, contours_plate, index_chars_candidate, img_plate_gray,
img_plate_rgb1):
    score_chars_candidate = numpy.zeros(len(index_chars_candidate))
    counter_index_chars_candidate = 0
    for chars_candidateA in index_chars_candidate:
        xA,yA,wA,hA = cv2.boundingRect(contours_plate[chars_candidateA])
        for chars_candidateB in index_chars_candidate:
            if chars_candidateA == chars_candidateB:
                continue
            else:
                xB,yB,wB,hB = cv2.boundingRect(contours_plate[chars_candidateB])
                y_difference = abs(yA - yB)
                if y_difference < 11:
                    score_chars_candidate[counter_index_chars_candidate] =
score_chars_candidate[counter_index_chars_candidate] + 1
                    counter_index_chars_candidate += 1
    index_chars = []
    chars_counter = 0
    for score in score_chars_candidate:
        if score == max(score_chars_candidate):
            index_chars.append(index_chars_candidate[chars_counter])
            chars_counter += 1
    img_plate_rgb2 = cv2.cvtColor(img_plate_gray, cv2.COLOR_GRAY2BGR)
    for char in index_chars:
        x, y, w, h = cv2.boundingRect(contours_plate[char])
        cv2.rectangle(img_plate_rgb2,(x,y),(x+w,y+h),(0,255,0),5)
        cv2.putText(img_plate_rgb2, str(index_chars.index(char)),(x, y + h + 50),
cv2.FONT_ITALIC, 2.0, (0,0,255), 3)
    x_coors = []
    for char in index_chars:

```

```

        x, y, w, h = cv2.boundingRect(contours_plate[char])
        x_coors.append(x)
x_coors = sorted(x_coors)
index_chars_sorted = []
for x_coor in x_coors:
    for char in index_chars:
        x, y, w, h = cv2.boundingRect(contours_plate[char])
        if x_coors[x_coors.index(x_coor)] == x:
            index_chars_sorted.append(char)
img_plate_rgb3 = cv2.cvtColor(img_plate_gray, cv2.COLOR_GRAY2BGR)
for char_sorted in index_chars_sorted:
    x,y,w,h = cv2.boundingRect(contours_plate[char_sorted])
    cv2.rectangle(img_plate_rgb3,(x,y),(x+w,y+h),(0,255,0),5)
    cv2.putText(img_plate_rgb3, str(index_chars_sorted.index(char_sorted)),(x, y + h +
50), cv2.FONT_ITALIC, 2.0, (0,0,255), 3)
    cv2.rectangle(img_plate_rgb1,(x,y),(x+w,y+h),(0,255,0),5)
    cv2.putText(img_plate_rgb1, str(index_chars_sorted.index(char_sorted)),(x, y + h +
50), cv2.FONT_ITALIC, 2.0, (0,0,255), 3)
    return (img_plate_rgb3, img_plate_rgb1, index_chars_sorted)

```

```

def kenalikarakter(self, contours_plate, img_plate_bw, index_chars_sorted):
    img_height = 40
    img_width = 40
    class_names = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J',
'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z']
    model = keras.models.load_model('my_model')
    num_plate = []
    for char_sorted in index_chars_sorted:
        x,y,w,h = cv2.boundingRect(contours_plate[char_sorted])
        char_crop = cv2.cvtColor(img_plate_bw[y:y+h,x:x+w], cv2.COLOR_GRAY2BGR)

```

```

char_crop = cv2.resize(char_crop, (img_width, img_height))
img_array = keras.preprocessing.image.img_to_array(char_crop)
img_array = tf.expand_dims(img_array, 0)
predictions = model.predict(img_array)
score = tf.nn.softmax(predictions[0])
num_plate.append(class_names[numpy.argmax(score)])
plate_number = "
for a in num_plate:
    plate_number += a

self.plate_number = plate_number
jalankan = Aplikasi()
jalankan = None

```

@echo off

```

if defined _OLD_VIRTUAL_PROMPT (
    set "PROMPT=%_OLD_VIRTUAL_PROMPT%"
)
set _OLD_VIRTUAL_PROMPT=

if defined _OLD_VIRTUAL_PYTHONHOME (
    set "PYTHONHOME=%_OLD_VIRTUAL_PYTHONHOME%"
    set _OLD_VIRTUAL_PYTHONHOME=
)

if defined _OLD_VIRTUAL_PATH (
    set "PATH=%_OLD_VIRTUAL_PATH%"
)

```

```
set _OLD_VIRTUAL_PATH=
```

```
set VIRTUAL_ENV=
```

```
set VIRTUAL_ENV_PROMPT=
```

```
:END
```

```
@echo off
```

```
rem This file is UTF-8 encoded, so we need to update the current code page while executing it
```

```
for /f "tokens=2 delims=:." %%a in ("%SystemRoot%\System32\chcp.com") do (
```

```
    set _OLD_CODEPAGE=%%a
```

```
)
```

```
if defined _OLD_CODEPAGE (
```

```
    "%SystemRoot%\System32\chcp.com" 65001 > nul
```

```
)
```

```
set VIRTUAL_ENV=C:\Users\MSI14-B4MW\Desktop\deteksiplat\env
```

```
if not defined PROMPT set PROMPT=$P$G
```

```
if defined _OLD_VIRTUAL_PROMPT set PROMPT=%_OLD_VIRTUAL_PROMPT%
```

```
if defined _OLD_VIRTUAL_PYTHONHOME set  
PYTHONHOME=%_OLD_VIRTUAL_PYTHONHOME%
```

```
set _OLD_VIRTUAL_PROMPT=%PROMPT%
```

```
set PROMPT=(env) %PROMPT%
```

```
if defined PYTHONHOME set _OLD_VIRTUAL_PYTHONHOME=%PYTHONHOME%
```

```
set PYTHONHOME=
```

```
if defined _OLD_VIRTUAL_PATH set PATH=%_OLD_VIRTUAL_PATH%
```

```
if not defined _OLD_VIRTUAL_PATH set _OLD_VIRTUAL_PATH=%PATH%
```

```
set PATH=%VIRTUAL_ENV%\Scripts;%PATH%
```

```
set VIRTUAL_ENV_PROMPT=(env)
```

```
:END
```

```
if defined _OLD_CODEPAGE (
```

```
    "%SystemRoot%\System32\chcp.com" %_OLD_CODEPAGE% > nul
```

```
    set _OLD_CODEPAGE=
```

```
)
```



KARTU MONITORING BIMBINGAN
MAHASISWA PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH PAREPARE

PROPOSAL

Mahasiswa : Rezki Dwi Putra	Pembimbing I : Hj. A. Irmayani P, S.T., M.T
NIM : 219280124	Pembimbing II : Wahyuddin, S.kom., M.kom
Judul Skripsi : APLIKASI PENGENALAN PLAT KENDARAAN BERBASIS ANDROID UNTUK IDENTITAS KEPEMILIKAN KENDARAAN	

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 1 Perbaiki BAB II	<i>[Signature]</i>	Konsultasi 1 Perbaiki dokumen	<i>[Signature]</i>
Konsultasi 2 Perbaiki daftar pustaka	<i>[Signature]</i>	Konsultasi 2 Ikuti format penulisan	<i>[Signature]</i>
Konsultasi 3 acc	<i>[Signature]</i>	Konsultasi 3 Tambahkan teori	<i>[Signature]</i>
Konsultasi 4		Konsultasi 4 Acc	<i>[Signature]</i>
Konsultasi 5		Konsultasi 5	

Lanjut ke halaman sebelah...

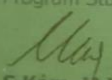
Perhatian :

1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini sudah dikuasai oleh mahasiswa, dicatat konsultasi dan diisi oleh Pembimbing

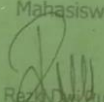
Lampiran

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 6		Konsultasi 6	
Konsultasi 7		Konsultasi 7	
Konsultasi 8		Konsultasi 8	
Konsultasi 9		Konsultasi 9	
Konsultasi 10		Konsultasi 10	

Mengetahui
Ketua Program Studi


Marlina, S.Kom., M.Kom.
NBM. 1162 680

Parepare,

Mahasiswa

Rizki Dwi Putra
NIM. 219280124

Perhatian :

1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini wajib dibawa oleh mahasiswa disetiap konsultasi dan diisi oleh Pembimbing
3. Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal/ujian skripsi
4. Kartu ini dicetak di atas kertas karton berwarna hijau muda dan dicetak timbal balik



KARTU MONITORING BIMBINGAN
MAHASISWA PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH PAREPARE

HASIL

Mahasiswa : Rezi Dwi Putra	Pembimbing I : Hj. A. Irmayani P. S.T.M.T
NIM : 219280124	Pembimbing II : Wahyuddin. S.kom.M.kom
Judul Skripsi : APLIKASI PENGENALAN PLAT KENDARAAN BERBASIS PYTHON UNTUK IDENTITAS KEPEMILIKAN KENDARAAN	

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 1 ~ Hasil pengujian = ... C (kesimpulan)	28/10/2021 A.I.	Konsultasi 1 - Perbaiki - format - penulisan	
Konsultasi 2 ~ pergeseran Plat (jarak kamera ke Plat (10 cm - 35 cm) = ?	A.I.	Konsultasi 2 - Use Case - Tabel Use Case	
Konsultasi 3 ~ perbaiki abstrak $\Rightarrow$ dan jarak ?	A.I.	Konsultasi 3 - Activity Diagram	
Konsultasi 4 Bawa DD Asli (Motor)	A.I.	Konsultasi 4 - Abstrak pengujian sistem	
Konsultasi 5 Campuran $\Rightarrow$ hosting Program	A.I.	Konsultasi 5 Acc	

Lanjut ke halaman sebelah...

Perhatian :

1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini wajib dibawa oleh mahasiswa saat konsultasi dan diisi oleh Pembimbing

Lanjutan

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 6 <i>Ace</i>	<i>25 Nov 2023</i> <i>Stru/</i>	Konsultasi 6	
Konsultasi 7		Konsultasi 7	
Konsultasi 8		Konsultasi 8	
Konsultasi 9		Konsultasi 9	
Konsultasi 10		Konsultasi 10	

Parepare,

Mengetahui
Ketua Program Studi

Marlina, S.Kom., M.Kom.
NBM. 1162 680

Mahasiswa

Rezki Dwi Putra
NIM. 219280124

Perhatian :

1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini wajib dibawa oleh mahasiswa disetiap konsultasi dan diisi oleh Pembimbing.
3. Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal/ujian skripsi
4. Kartu ini dicetak di atas kertas karton berwarna hijau muda dan dicetak timbal balik



KARTU MONITORING BIMBINGAN
MAHASISWA PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH PAREPARE

HASIL

Mahasiswa : Rezki Dwi Putra	Pembimbing I : Hj. A. Imayani P, S.T.,M.T
NIM : 219280124	Pembimbing II : Wahyuddin, S.kom.,M.kom
Judul Skripsi : APLIKASI PENGENALAN PLAT KENDARAAN BERBASIS PYTHON UNTUK IDENTITAS KEPEMILIKAN KENDARAAN	

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 1 ~ Hasil pengujian = ...? C (kesimpulan)	28/10/2023 A.I.	Konsultasi 1 - Perbaiki → format penulisan	A
Konsultasi 2 ~ pergerakan Plat (Jarak kamera ke Plat (10 cm - 35 cm) = ?	A.I.	Konsultasi 2 - Use Case - Tabel Use Case	A
Konsultasi 3 ~ perbaiki abstrak ⇒? dan jarak ?	A.I.	Konsultasi 3 - Activity Diagram	A
Konsultasi 4 Bawa DD Asli (Motor)	A.I.	Konsultasi 4 - Abstrak pengujian sistem	A
Konsultasi 5 Lampiran ⇒ listing Program	A.I.	Konsultasi 5 Acc	A

Lanjut ke halaman sebelah ..

Perhatian :

1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini wajib dibawa oleh mahasiswa saat konsultasi dan tidak boleh diserahkan

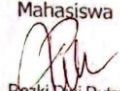
Lanjutan ...

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 6 Ace	25 Nov 2023 Arj	Konsultasi 6 Acci	
Konsultasi 7 Jurnal & Revisi fol 14.	12/2/2024 Arj	Konsultasi 7	
Konsultasi 8 ASTRAL Khs hygiene → Tegal	Arj	Konsultasi 8	
Konsultasi 9 Ace	Arj	Konsultasi 9	
Konsultasi 10		Konsultasi 10	

Mengetahui
Ketua Program Studi

Marlina, S.Kom., M.Kom.
NBM. 1162 680

Parepare,

Mahasiswa

Rezki Dwi Putra
NIM. 219280124

Perhatian :

1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini wajib dibawa oleh mahasiswa disetiap konsultasi dan diisi oleh Pembimbing
3. Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal/ujian skripsi
4. Kartu ini dicetak di atas kertas karton berwarna hijau muda dan dicetak timbal balik