**ANALISIS GAMBAR KE TEKS KARAKTER RECOGNITION**

**(Tugas Proyek Khusus)**

**Oleh**

**PUTRA PRIBOWO**

**1617051005**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS LAMPUNG**

**BANDAR LAMPUNG**

**2019**

**DATA GAMBAR YANG DIGUNAKAN**

**GAMBAR 1**

****

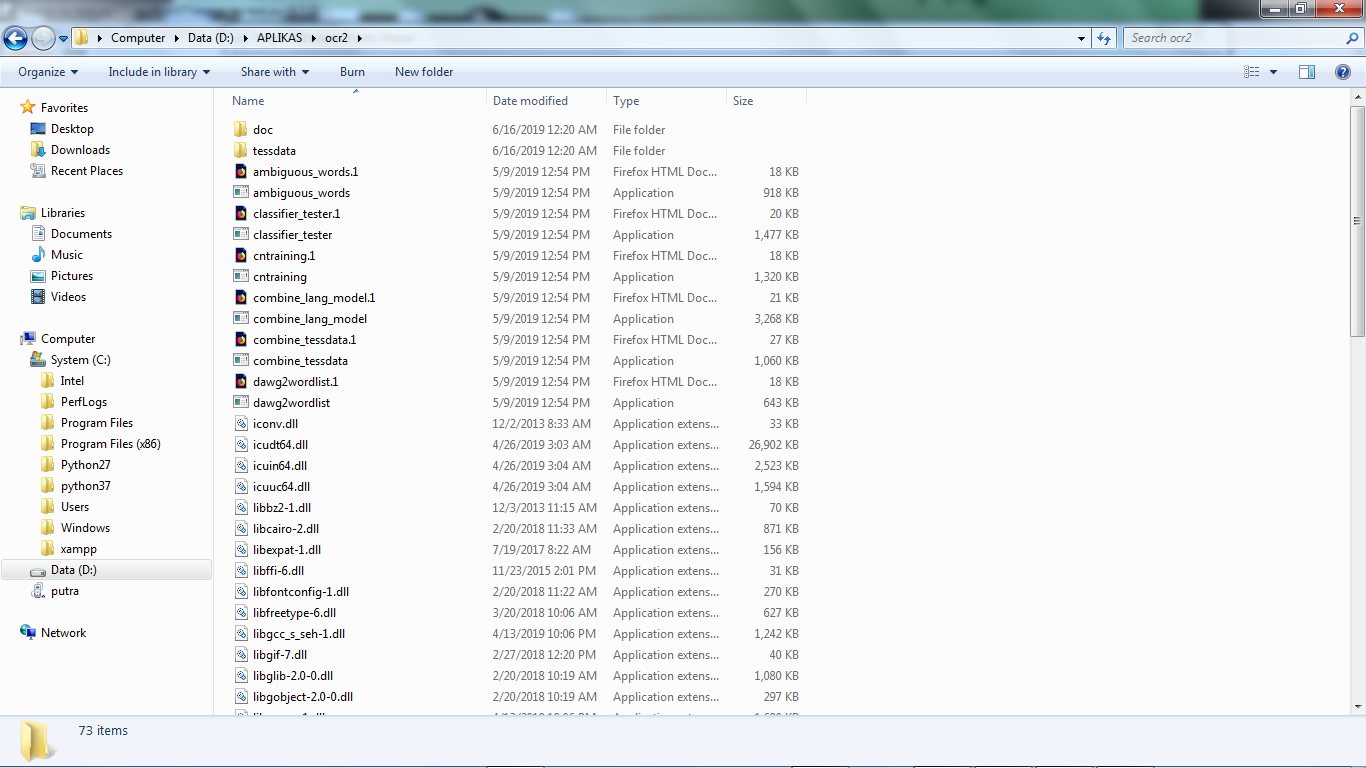
**GAMBAR 2**

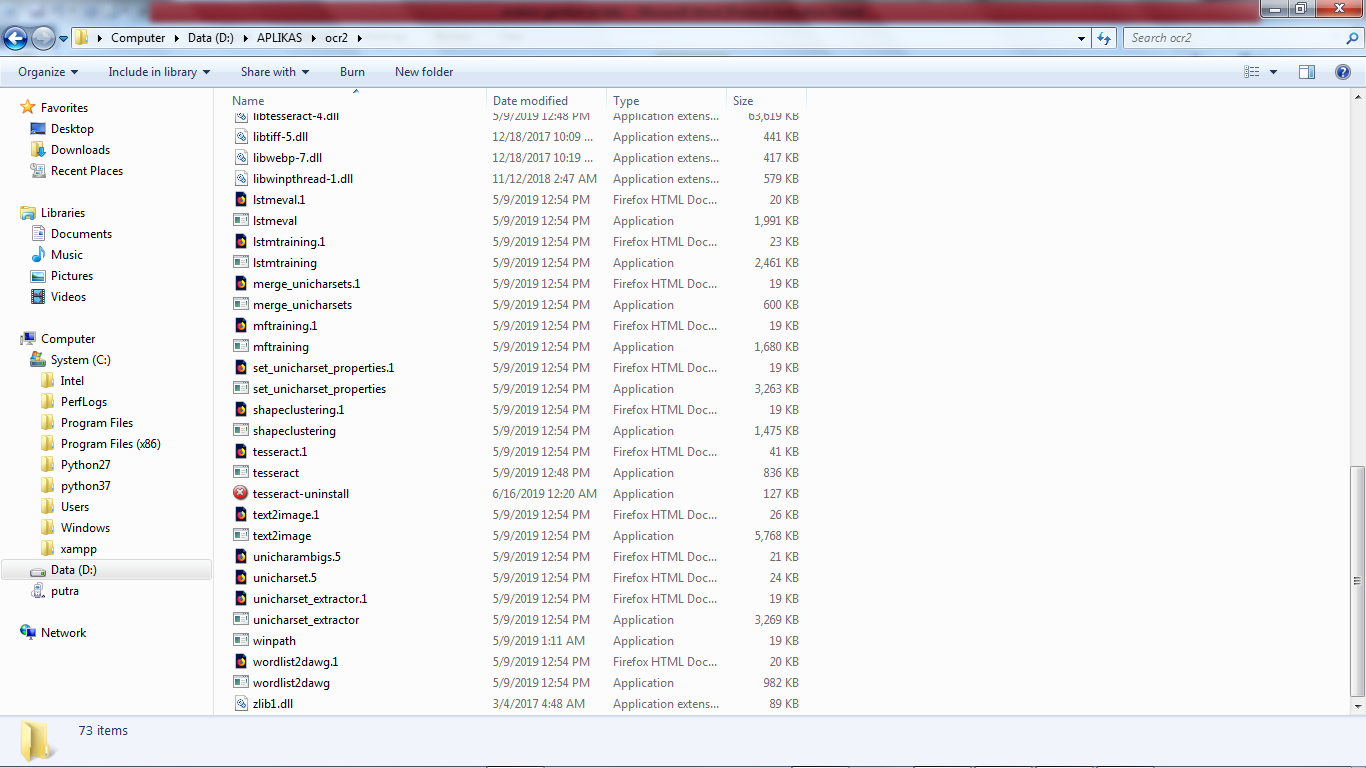
****

**GAMBAR 3**

****

**GAMBAR PATH OCR YANG DIGUNAKAN DIDALAM PROGRAM**





**SOURCODE**

import cv2

import pytesseract

import numpy as np

import argparse

import os

from PIL import Image

import sys

#untuk ekstrak path tesseract ocr

pytesseract.pytesseract.tesseract\_cmd = r'D:\APLIKAS\ocr2\tesseract'

#untuk memasukkan path gambarnya

def get\_string(img\_path):

# membaca gambar dan menjadi RGB

img = cv2.imread(img\_path, cv2.IMREAD\_COLOR)

# merubah ke grayscale

img\_c = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

# untuk mengurangi noise

kernel = np.ones((1, 1), np.uint8)

img\_nd = cv2.dilate(img\_c, kernel, iterations=1)

img\_ne = cv2.erode(img\_nd, kernel, iterations=1)

# menulis image sebelum merubahnya ke teks

cv2.imwrite("thres.png", img\_ne)

# menggunakan tesseract

result = pytesseract.image\_to\_string(Image.open("thres.png"))

return result

if \_\_name\_\_ == '\_\_main\_\_':

from sys import argv

if len(argv)<2:

print("python image-to-text.py nama file")

else:

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print('Nama: Putra Pribowo')

print('Npm : 1617051005')

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print('--- Memulai ---')

for i in range(1,len(argv)):

print(argv[i])

# memberikan informasi gambar ke teks

print("\n")

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print("hasil string saja")

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print(get\_string(argv[i]))

print("\n")

# memberikan informasi kotak estimasi

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print("gambar ke boxes")

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print(pytesseract.image\_to\_boxes(Image.open(argv[i])))

print("\n")

# memberikan informasi including boxes, confidences, baris and nomor baris

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print("gambar ke data")

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print(pytesseract.image\_to\_data(Image.open(argv[i])))

print("\n")

# memberikan informasi tentang orientasi dan skrip deteksi

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print("gambar ke osd")

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

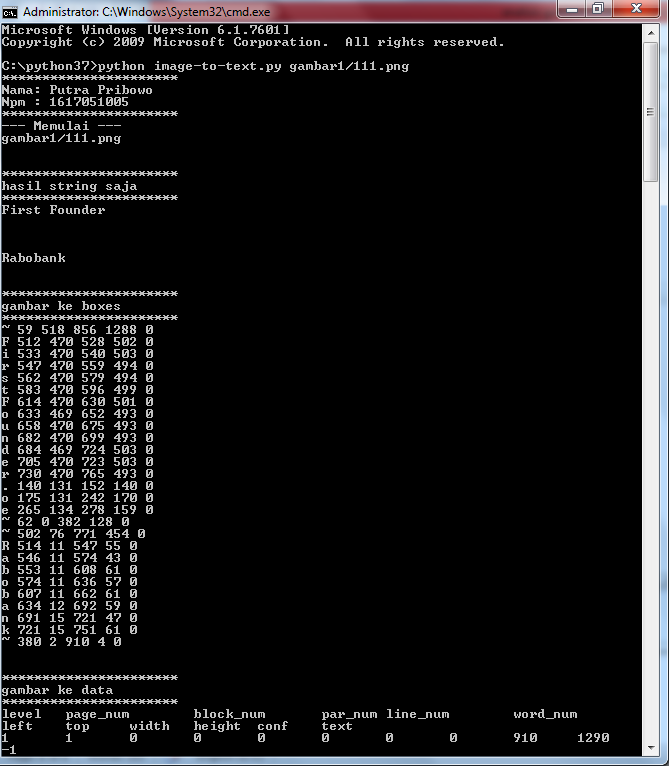
print(pytesseract.image\_to\_osd(Image.open(argv[i])))

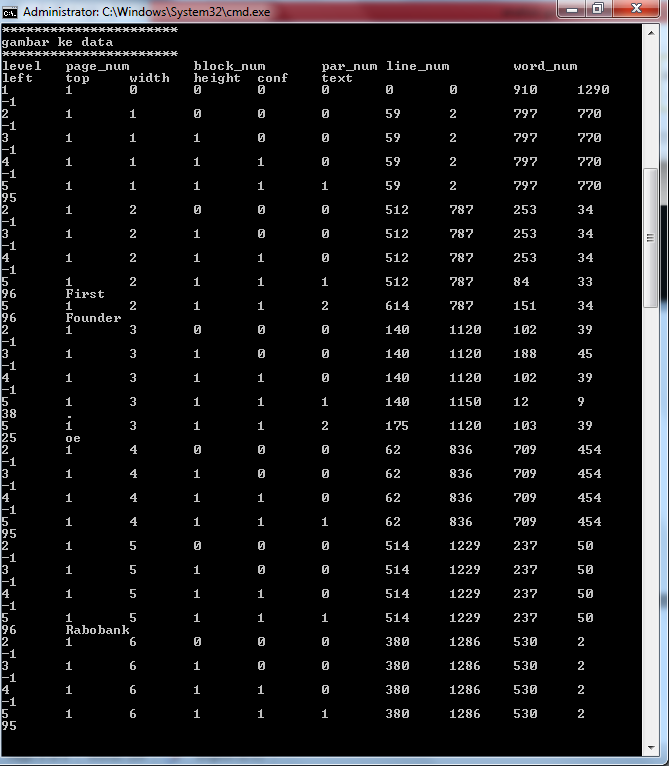
print()

print()

print('------ Sukses -------')

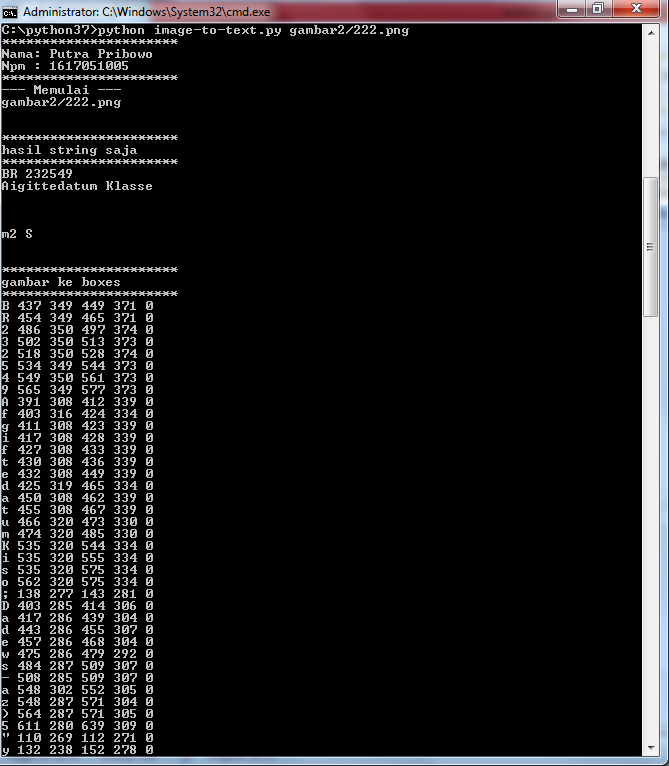
**HASIL GAMBAR 1**

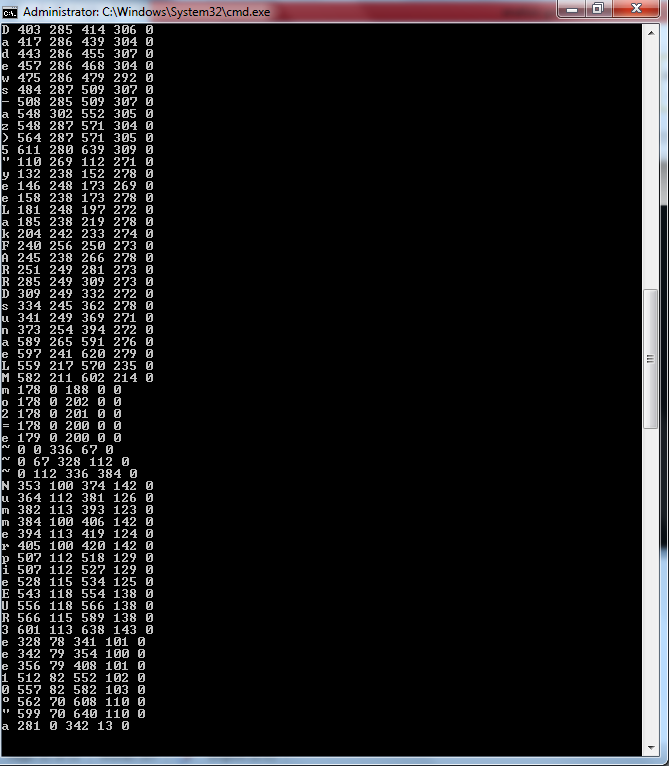


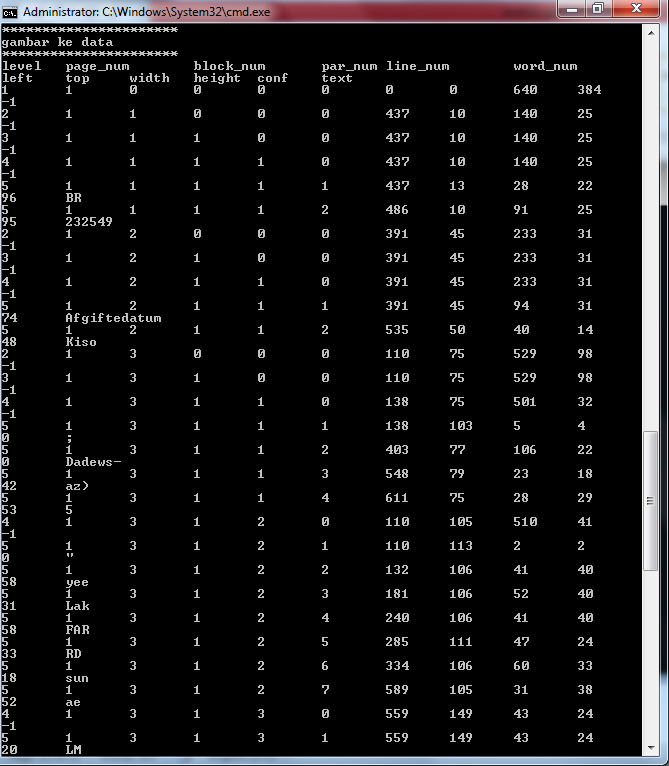


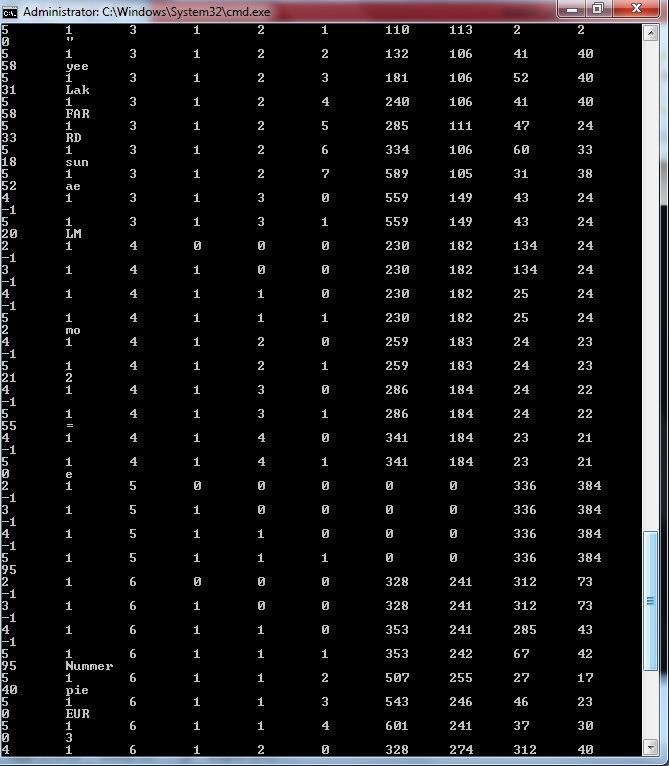


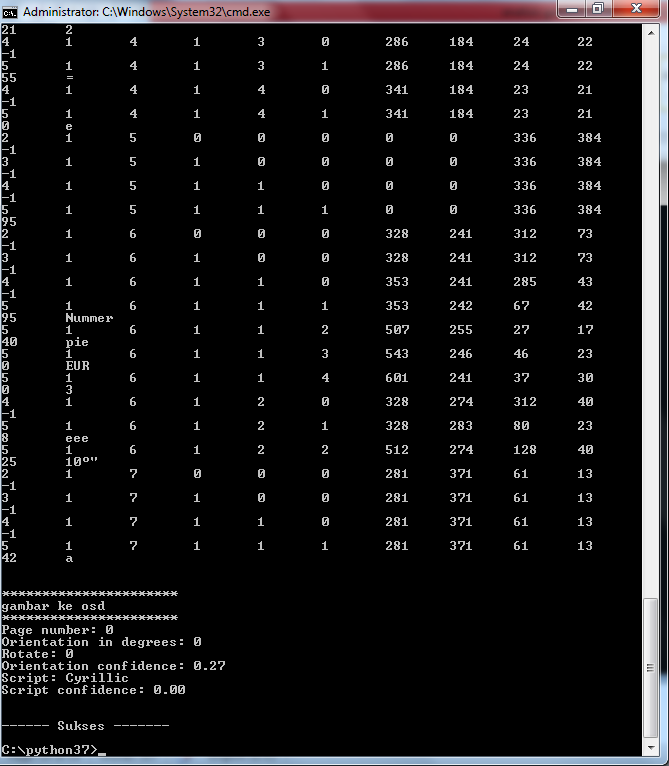
**HASIL GAMBAR 2**



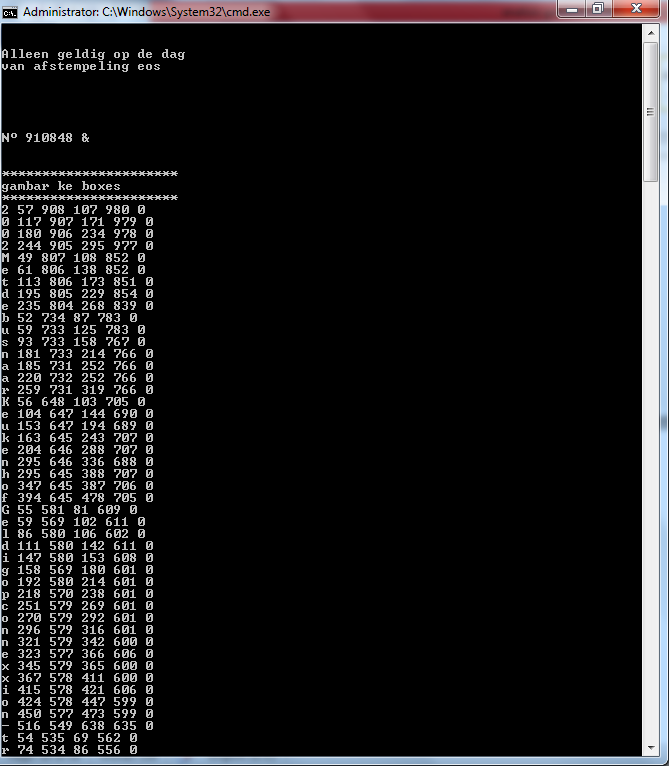


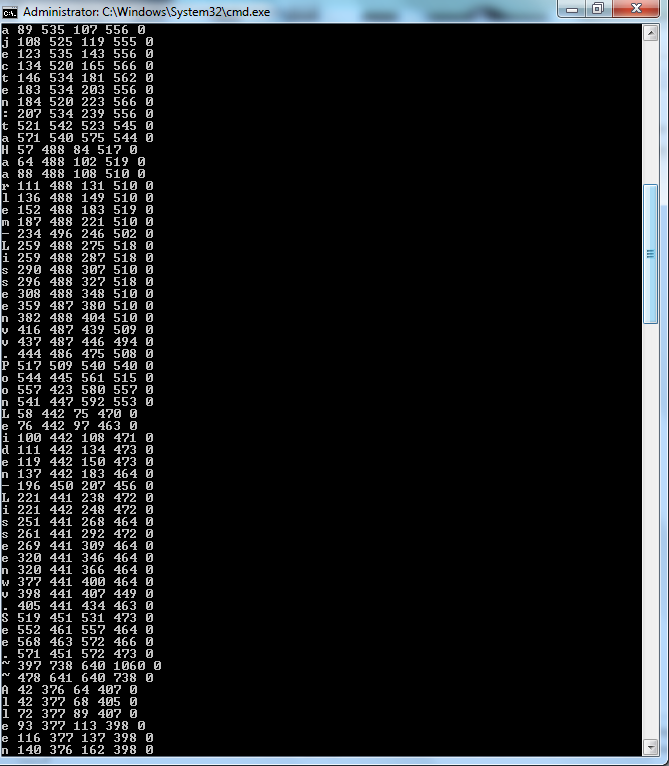


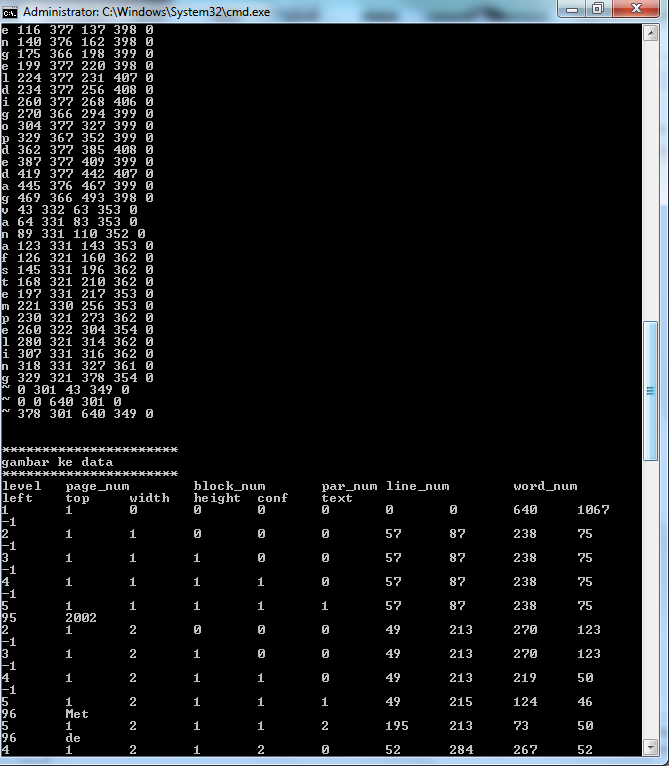


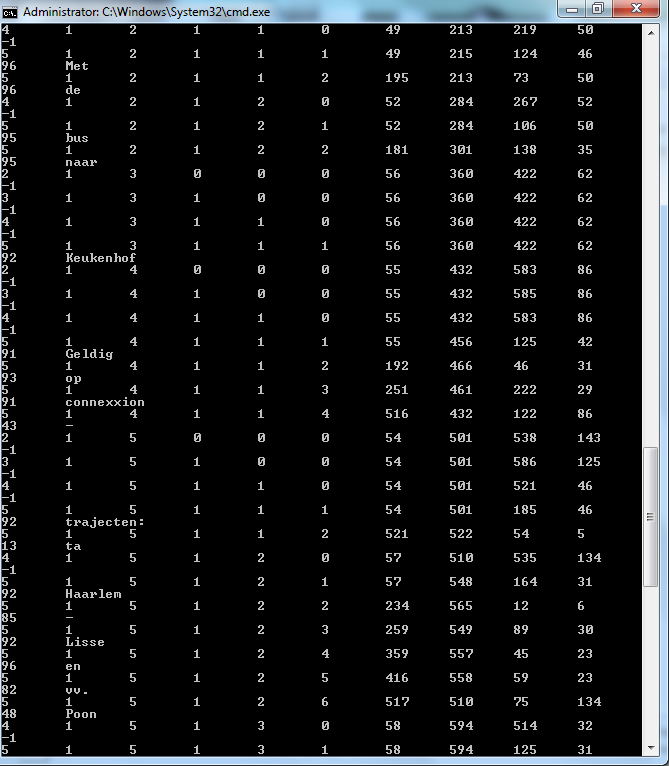


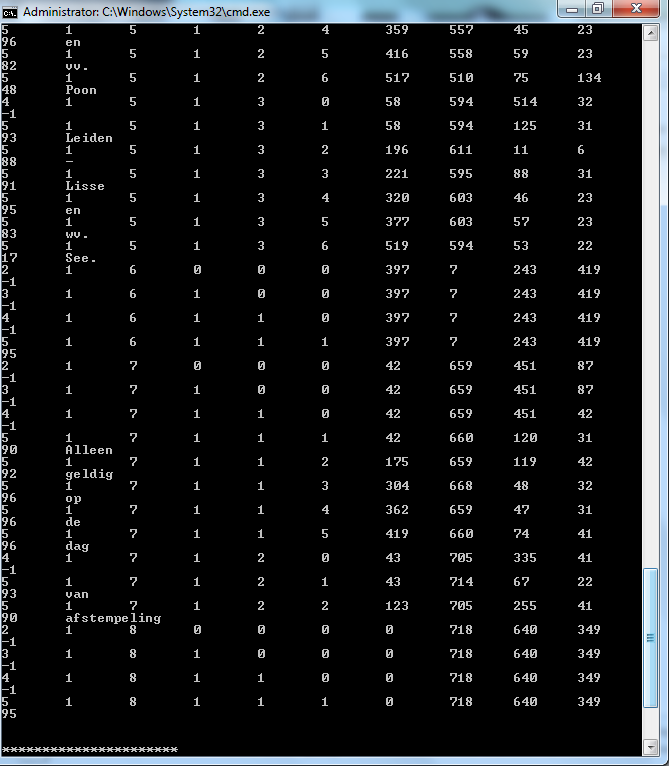
**HASIL GAMBAR 3**

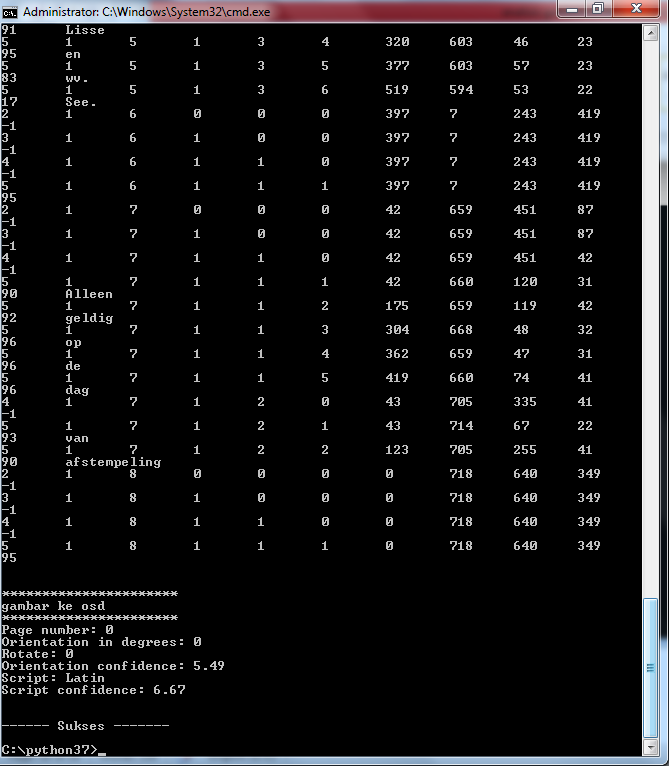












**HASIL PENURUNAN NOISE GAMBAR 1**

****

**HASIL PENURUNAN NOISE GAMBAR 2**

****

**HASIL PENURUNAN NOISE GAMBAR 3**

****

**DALAM MENGGUNAKAN SORCODENYA PADA GAMBAR 1**

**Python image-to-text.py gambar1/111.png**

**ANALISIS GAMBAR 1**

Dalam analisis gambar tersebut sebelumnya membuat ukuran gambar 900x1500 pixel dan mengurangi noise beberapa kali, dan hasilnya tidak semua terbaca tulisan hanya sebagian, dikarenakan kurangnya pixel yang lebih tinggi didalam yang dikonvert dan tingkat akurasi pembacaannya mencapai 25 persen.

Tulisan dalam gambar tersebut mendapatkan hasil first founder dan Rabobank tulisannya dapat terbaca. Dan terdapat suatu nilai yang ditampung yang ditampilkan pada sistem dan menghasilkan nilai akurasi yang menggunakan tulisan apa saja yang digunakan didalam gambar.

**DALAM MENGGUNAKAN SORCODENYA PADA GAMBAR 2**

**Python image-to-text.py gambar2/222.png**

**ANALISIS GAMBAR 2**

Saya menggunakan ukuran gambar 1500x900 pixel dan mengurangi noise beberapa kali dengan converter online dari sedikit kebaca tulisan menjadi agak lebih dari sebelumnya, dan hasilnya tidak semua terbaca tulisan hanya sebagian, dikarenakan kurangnya pixel yang lebih tinggi didalam gambar atau kurang besar.

Tulisan dalam gambar tersebut mendapatkan hasil BR232549 aigittedatum klasse tulisannya dapat terbaca mencapai 35 persen. Dan terdapat suatu nilai yang ditampung berbentuk boxes yang ditampilkan pada sistem dan menghasilkan nilai akurasi yang menggunakan tulisan apa saja yang digunakan didalam gambar yang dapat terlihat pada gambar 2 diatas hasil running programnya.

**DALAM MENGGUNAKAN SORCODENYA PADA GAMBAR 3**

**Python image-to-text.py gambar3/333.png**

**ANALISIS GAMBAR 3**

Dalam analisis gambar 3 ini saya menggunakan ukuran pixel gambar 640x1067 pixel. Didalam gambar saya mengurangi noise pada gambar 1 x di convert di online. Dan hasil dari tulisan pada gambar tersebut dapat terbaca 80 persen tulisan pada gambar tersebut. Tulisan yang terbaca adalah 2002 Met de bus naar Keukenhof gelding op connexxion trajecten: haarlem – lisse en vv leiden – lisse en vv Allen geldig op de dag van afstempeling no 010848.

Yang dapat dilihat pada hasil gambar 3 dan dapat dilihat nilai dari boxes dan akurasi tulisan yang digunakan pada gambar tersebut.