Nama : Agung Rizky S

Kelas : TI-3C

Nim : 2241720187

Github : https://github.com/agungrizkysetiawan/PembelajaranMesin.git

JOBSHEET 12

Optical Character Recognition (OCR)

Instalasi dan Import Library

```
!sudo apt install tesseract-ocr
!pip install pytesseract
!pip install opencv-python
```

```
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  tesseract-ocr-eng tesseract-ocr-osd
The following NEW packages will be installed:
  tesseract-ocr tesseract-ocr-eng tesseract-ocr-osd
0 upgraded, 3 newly installed, 0 to remove and 49 not upgraded.
Need to get 4,816 kB of archives.
After this operation, 15.6 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tesseract-
ocr-eng all 1:4.00~git30-7274cfa-1.1 [1,591 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tesseract-
ocr-osd all 1:4.00~git30-7274cfa-1.1 [2,990 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tesseract-
ocr amd64 4.1.1-2.1build1 [236 kB]
Fetched 4,816 kB in 0s (26.1 MB/s)
debconf: unable to initialize frontend: Dialog
debconf: (No usable dialog-like program is installed, so the dialog
based frontend cannot be used. at
/usr/share/perl5/Debconf/FrontEnd/Dialog.pm line 78, <> line 3.)
debconf: falling back to frontend: Readline
debconf: unable to initialize frontend: Readline
debconf: (This frontend requires a controlling tty.)
debconf: falling back to frontend: Teletype
dpkg-preconfigure: unable to re-open stdin:
Selecting previously unselected package tesseract-ocr-eng.
(Reading database ... 123633 files and directories currently
installed.)
Preparing to unpack .../tesseract-ocr-eng_1%3a4.00~git30-7274cfa-
```

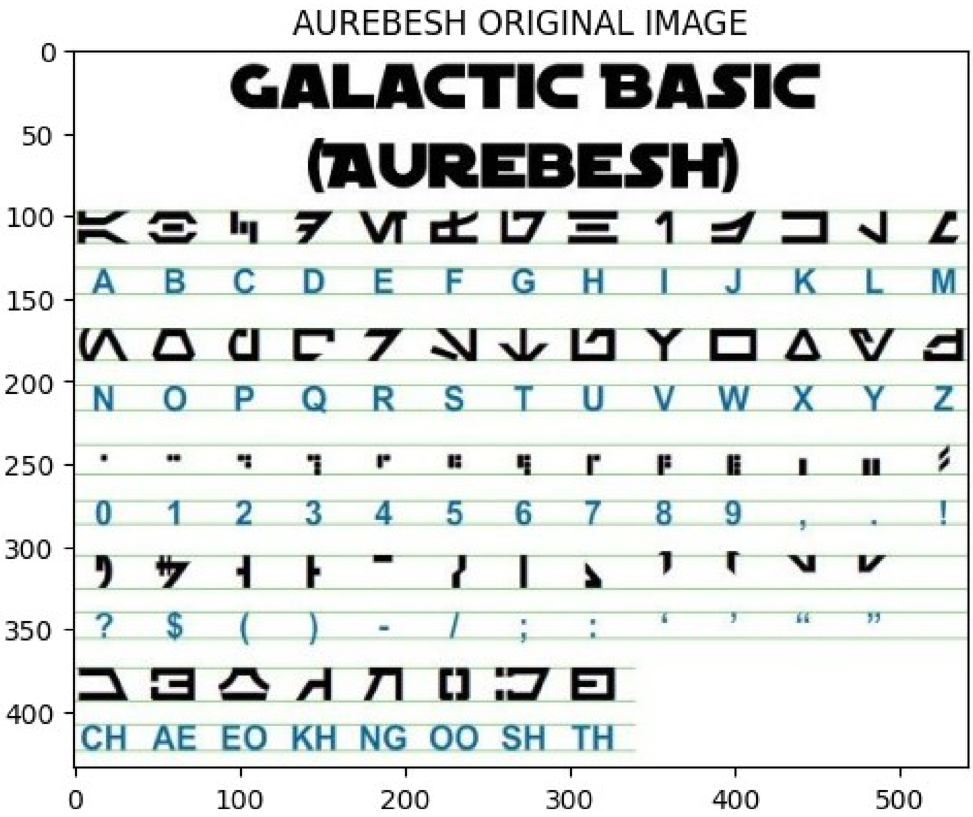
```
1.1_all.deb ...
Unpacking tesseract-ocr-eng (1:4.00~git30-7274cfa-1.1) ...
Selecting previously unselected package tesseract-ocr-osd.
Preparing to unpack .../tesseract-ocr-osd_1%3a4.00~git30-7274cfa-
1.1_all.deb ...
Unpacking tesseract-ocr-osd (1:4.00~git30-7274cfa-1.1) ...
Selecting previously unselected package tesseract-ocr.
Preparing to unpack .../tesseract-ocr_4.1.1-2.1build1_amd64.deb ...
Unpacking tesseract-ocr (4.1.1-2.1build1) ...
Setting up tesseract-ocr-eng (1:4.00~git30-7274cfa-1.1) ...
Setting up tesseract-ocr-osd (1:4.00~git30-7274cfa-1.1) ...
Setting up tesseract-ocr (4.1.1-2.1build1) ...
Processing triggers for man-db (2.10.2-1) ...
Collecting pytesseract
  Downloading pytesseract-0.3.13-py3-none-any.whl.metadata (11 kB)
Requirement already satisfied: packaging>=21.3 in
/usr/local/lib/python3.10/dist-packages (from pytesseract) (24.2)
Requirement already satisfied: Pillow>=8.0.0 in
/usr/local/lib/python3.10/dist-packages (from pytesseract) (11.0.0)
Downloading pytesseract-0.3.13-py3-none-any.whl (14 kB)
Installing collected packages: pytesseract
Successfully installed pytesseract-0.3.13
Requirement already satisfied: opencv-python in
/usr/local/lib/python3.10/dist-packages (4.10.0.84)
Requirement already satisfied: numpy>=1.21.2 in
/usr/local/lib/python3.10/dist-packages (from opencv-python) (1.26.4)
```

```python
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```python
import re
import cv2
import numpy as np
import pytesseract
from pytesseract import Output
from matplotlib import pyplot as plt

IMG_DIR = '/content/drive/MyDrive/ML/images-ocr/images'


# get grayscale image
def get_grayscale(image):
    return cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# noise removal
def remove_noise(image):
    return cv2.medianBlur(image,5)

#thresholding
```

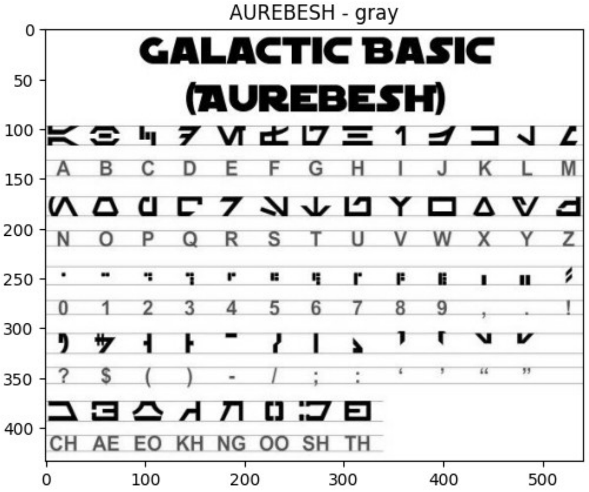
```python
def thresholding(image):
    return cv2.threshold(image, 0, 255, cv2.THRESH_BINARY +
cv2.THRESH_OTSU)[1]

#dilation
def dilate(image):
    kernel = np.ones((5,5),np.uint8)
    return cv2.dilate(image, kernel, iterations = 1)

#erosion
def erode(image):
    kernel = np.ones((5,5),np.uint8)
    return cv2.erode(image, kernel, iterations = 1)

#opening - erosion followed by dilation
def opening(image):
    kernel = np.ones((5,5),np.uint8)
    return cv2.morphologyEx(image, cv2.MORPH_OPEN, kernel)

#canny edge detection
def canny(image):
    return cv2.Canny(image, 100, 200)

#skew correction
def deskew(image):
    coords = np.column_stack(np.where(image > 0))
    angle = cv2.minAreaRect(coords)[-1]
    if angle < -45:
        angle = -(90 + angle)
    else:
        angle = -angle
    (h, w) = image.shape[:2]
    center = (w // 2, h // 2)
    M = cv2.getRotationMatrix2D(center, angle, 1.0)
    rotated = cv2.warpAffine(image, M, (w, h), flags=cv2.INTER_CUBIC,
borderMode=cv2.BORDER_REPLICATE)
    return rotated

#template matching
def match_template(image, template):
    return cv2.matchTemplate(image, template, cv2.TM_CCOEFF_NORMED)

# Plot original image

image = cv2.imread(IMG_DIR + '/aurebesh.jpg')
b,g,r = cv2.split(image)
rgb_img = cv2.merge([r,g,b])
plt.imshow(rgb_img)
plt.title('AUREBESH ORIGINAL IMAGE')
plt.show()
```



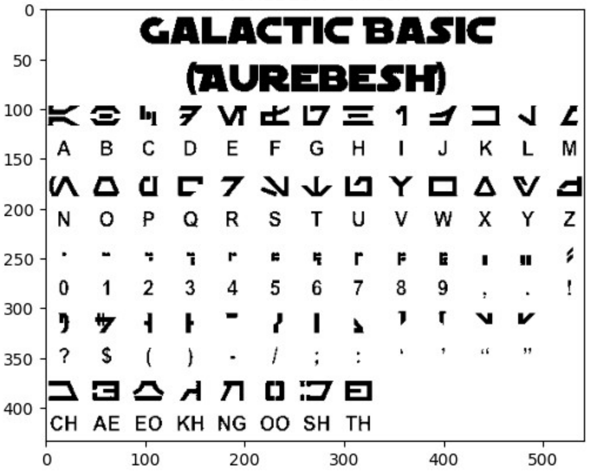
Pre-processing Image

```
# Preprocess image

gray = get_grayscale(image)
thresh = thresholding(gray)
opening = opening(gray)
canny = canny(gray)
images = {'gray': gray,
          'thresh': thresh,
          'opening': opening,
          'canny': canny}

# Plot images after preprocessing

fig = plt.figure(figsize=(13,13))
ax = []

rows = 2
columns = 2
keys = list(images.keys())
for i in range(rows*columns):
    ax.append( fig.add_subplot(rows, columns, i+1) )
    ax[-1].set_title('AUREBESH - ' + keys[i])
    plt.imshow(images[keys[i]], cmap='gray')
```

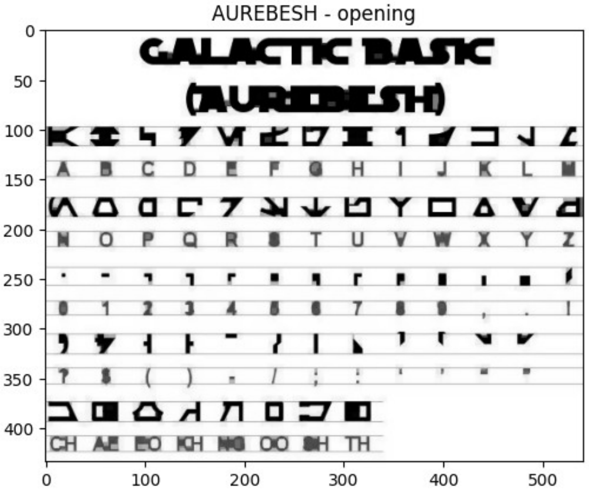
AUREBESH - gray



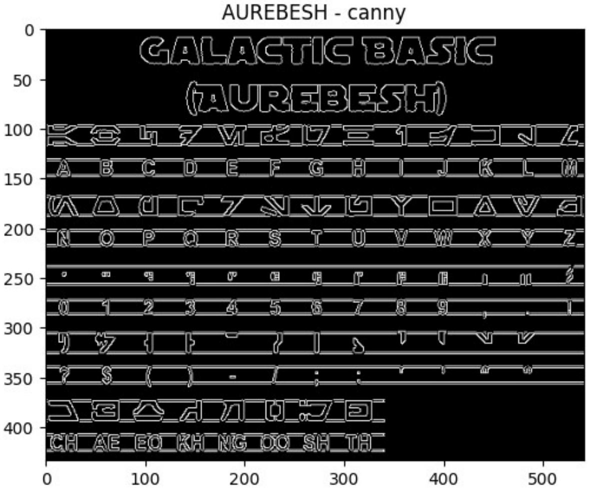
AUREBESH - thresh



AUREBESH - opening



AUREBESH - canny



Ektstraksi Data

```
# Get OCR output using Pytesseract

custom_config = r'--oem 3 --psm 6'
print('-----------------------------------------')
print('TESSERACT OUTPUT --> ORIGINAL IMAGE')
print('-----------------------------------------')
print(pytesseract.image_to_string(image, config=custom_config))
print('\n-----------------------------------------')
print('TESSERACT OUTPUT --> THRESHOLDED IMAGE')
print('-----------------------------------------')
print(pytesseract.image_to_string(image, config=custom_config))
print('\n-----------------------------------------')
print('TESSERACT OUTPUT --> OPENED IMAGE')
print('-----------------------------------------')
print(pytesseract.image_to_string(image, config=custom_config))
print('\n-----------------------------------------')
print('TESSERACT OUTPUT --> CANNY EDGE IMAGE')
print('-----------------------------------------')
print(pytesseract.image_to_string(image, config=custom_config))
```

```
-----------------------------------------
TESSERACT OUTPUT --> ORIGINAL IMAGE
-----------------------------------------
GALACTIC BASIC
(AUREBESH)

RE TFVMVEVEStZIoNe
AB CD EF Ga KL
AOderT7NVYoYoOoOAVA
N_ Oo. 2 _ HG: Re SS Ty wee Ve
Ss eg ei
ed
i a a Sy ee ee ee
st
ASaSAnNADIE
CH AE EO KH NG OO SH TH


-----------------------------------------
TESSERACT OUTPUT --> THRESHOLDED IMAGE
-----------------------------------------
GALACTIC BASIC
(AUREBESH)

RE TFVMVEVEStZIoNe
AB CD EF Ga KL
AOderT7NVYoYoOoOAVA
N_ Oo. 2 _ HG: Re SS Ty wee Ve
Ss eg ei
ed
i a a Sy ee ee ee
st
ASaSAnNADIE
CH AE EO KH NG OO SH TH


-----------------------------------------
TESSERACT OUTPUT --> OPENED IMAGE
-----------------------------------------
GALACTIC BASIC
(AUREBESH)

RE TFVMVEVEStZIoNe
```

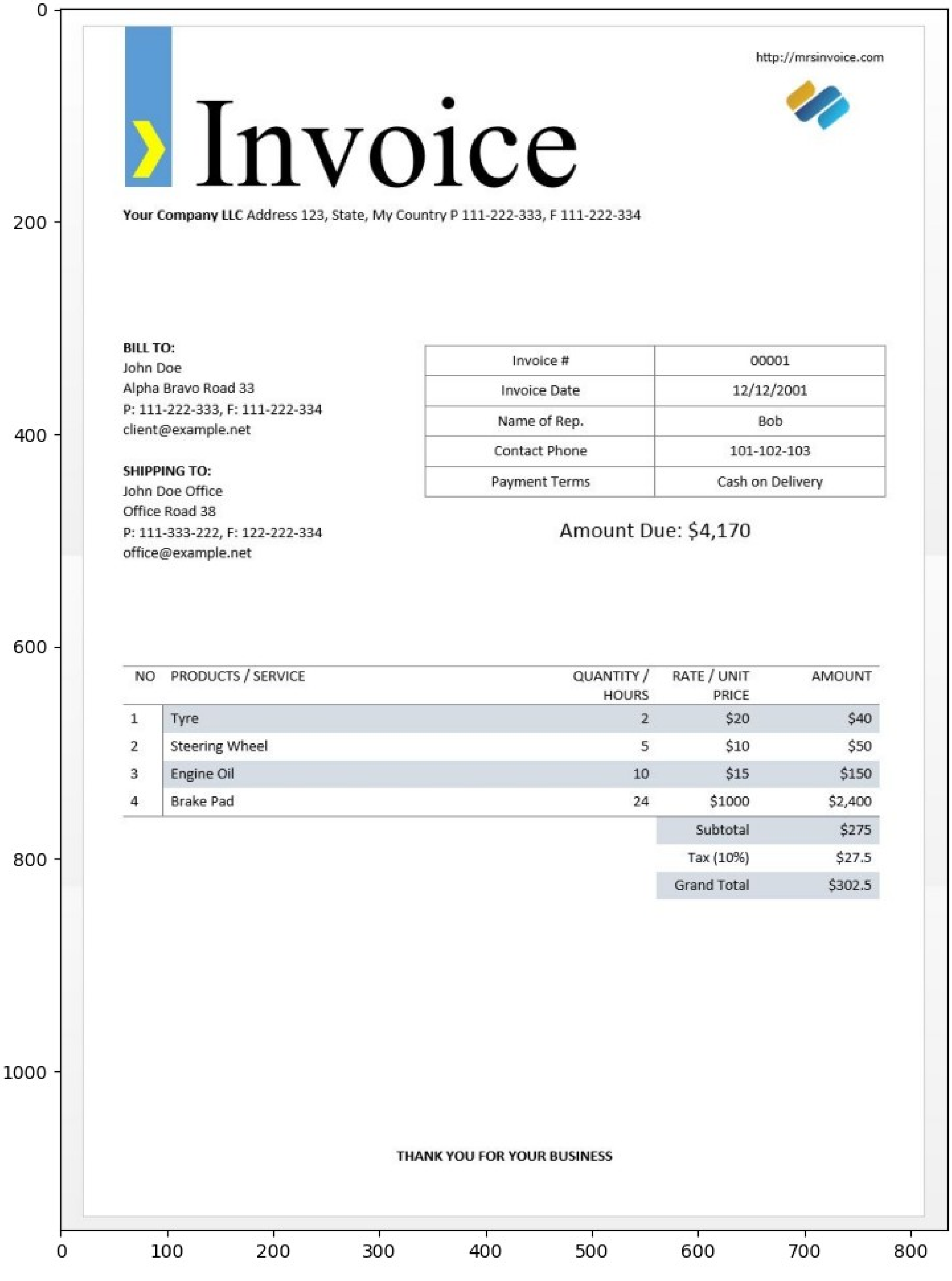
```
AB CD EF Ga KL
AOderT7NVYoYoOoOAVA
N_ Oo. 2 _ HG: Re SS Ty wee Ve
Ss eg ei
ed
i a a Sy ee ee ee
st
ASaSAnNADIE
CH AE EO KH NG OO SH TH


-----------------------------------------
TESSERACT OUTPUT --> CANNY EDGE IMAGE
-----------------------------------------
GALACTIC BASIC
(AUREBESH)

RE TFVMVEVEStZIoNe
AB CD EF Ga KL
AOderT7NVYoYoOoOAVA
N_ Oo. 2 _ HG: Re SS Ty wee Ve
Ss eg ei
ed
i a a Sy ee ee ee
st
ASaSAnNADIE
CH AE EO KH NG OO SH TH
```

Praktikum 2

Bounding Box - Level Karakter

```
# Plot gambar original

# Membaca gambar dari direktori
image = cv2.imread(IMG_DIR + '/invoice-sample.jpg')

# Memisahkan saluran warna (blue, green, red) karena matplotlib
menggunakan skema warna RGB
b, g, r = cv2.split(image)
rgb_img = cv2.merge([r, g, b])

# Menampilkan gambar dalam ukuran tertentu
plt.figure(figsize=(16, 12))
plt.imshow(rgb_img)
plt.title('CONTOH GAMBAR INVOICE')
plt.show()
```

CONTOH GAMBAR INVOICE


0
200
400
600
800
1000
0
100
200
300
400
500
600
700
800

http://mrsinvoice.com

Invoice

Your Company LLC Address 123, State, My Country P 111-222-333, F 111-222-334

BILL TO:
John Doe
Alpha Bravo Road 33
P: 111-222-333, F: 111-222-334
client@example.net

SHIPPING TO:
John Doe Office
Office Road 38
P: 111-333-222, F: 122-222-334
office@example.net

Invoice #	00001
Invoice Date	12/12/2001
Name of Rep.	Bob
Contact Phone	101-102-103
Payment Terms	Cash on Delivery

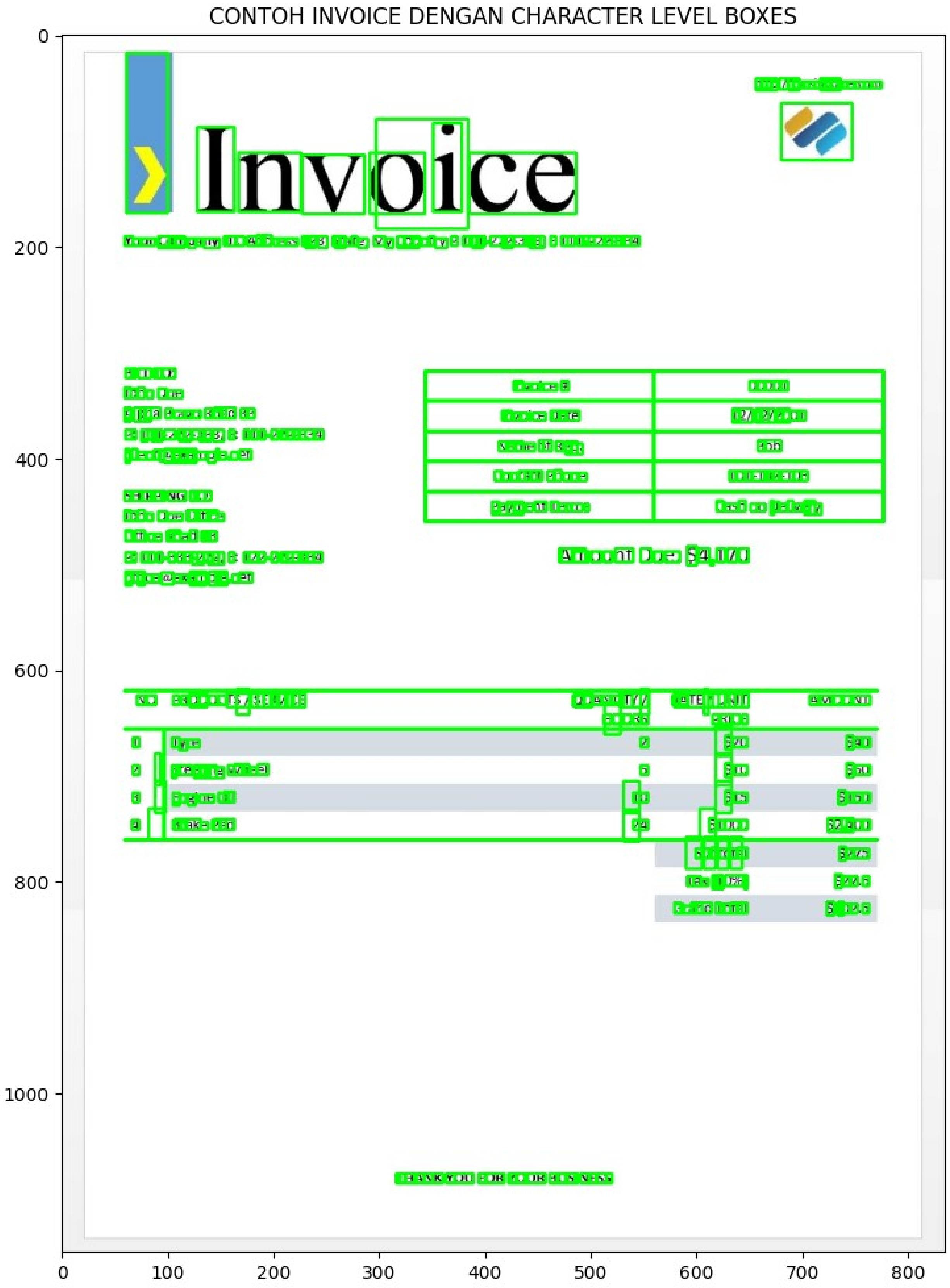
Amount Due: $4,170

NO	PRODUCTS / SERVICE	QUANTITY / HOURS	RATE / UNIT PRICE	AMOUNT
1	Tyre	2	$20	$40
2	Steering Wheel	5	$10	$50
3	Engine Oil	10	$15	$150
4	Brake Pad	24	$1000	$2,400
			Subtotal	$275
			Tax (10%)	$27.5
			Grand Total	$302.5

THANK YOU FOR YOUR BUSINESS

Plot karakter boxes pada gambar menggunakan fungsi pytesseract.image_to_boxes()

```python
# Membaca gambar dari direktori
image = cv2.imread(IMG_DIR + '/invoice-sample.jpg')

# Mendapatkan dimensi tinggi (h), lebar (w), dan channel warna (c) dari gambar
h, w, c = image.shape

# Menggunakan pytesseract.image_to_boxes() untuk mendapatkan informasi kotak karakter
boxes = pytesseract.image_to_boxes(image)

# Iterasi melalui setiap baris hasil dan membuat kotak pada gambar menggunakan OpenCV
for b in boxes.splitlines():
    b = b.split(' ')
    image = cv2.rectangle(image, (int(b[1]), h - int(b[2])), (int(b[3]), h - int(b[4])), (0, 255, 0), 2)

# Memisahkan channel warna untuk mengonversi dari BGR ke RGB
b, g, r = cv2.split(image)
rgb_img = cv2.merge([r, g, b])

# Menampilkan gambar dengan kotak karakter
plt.figure(figsize=(16, 12))
plt.imshow(rgb_img)
plt.title('CONTOH INVOICE DENGAN CHARACTER LEVEL BOXES')
plt.show()
```


CONTOH INVOICE DENGAN CHARACTER LEVEL BOXES

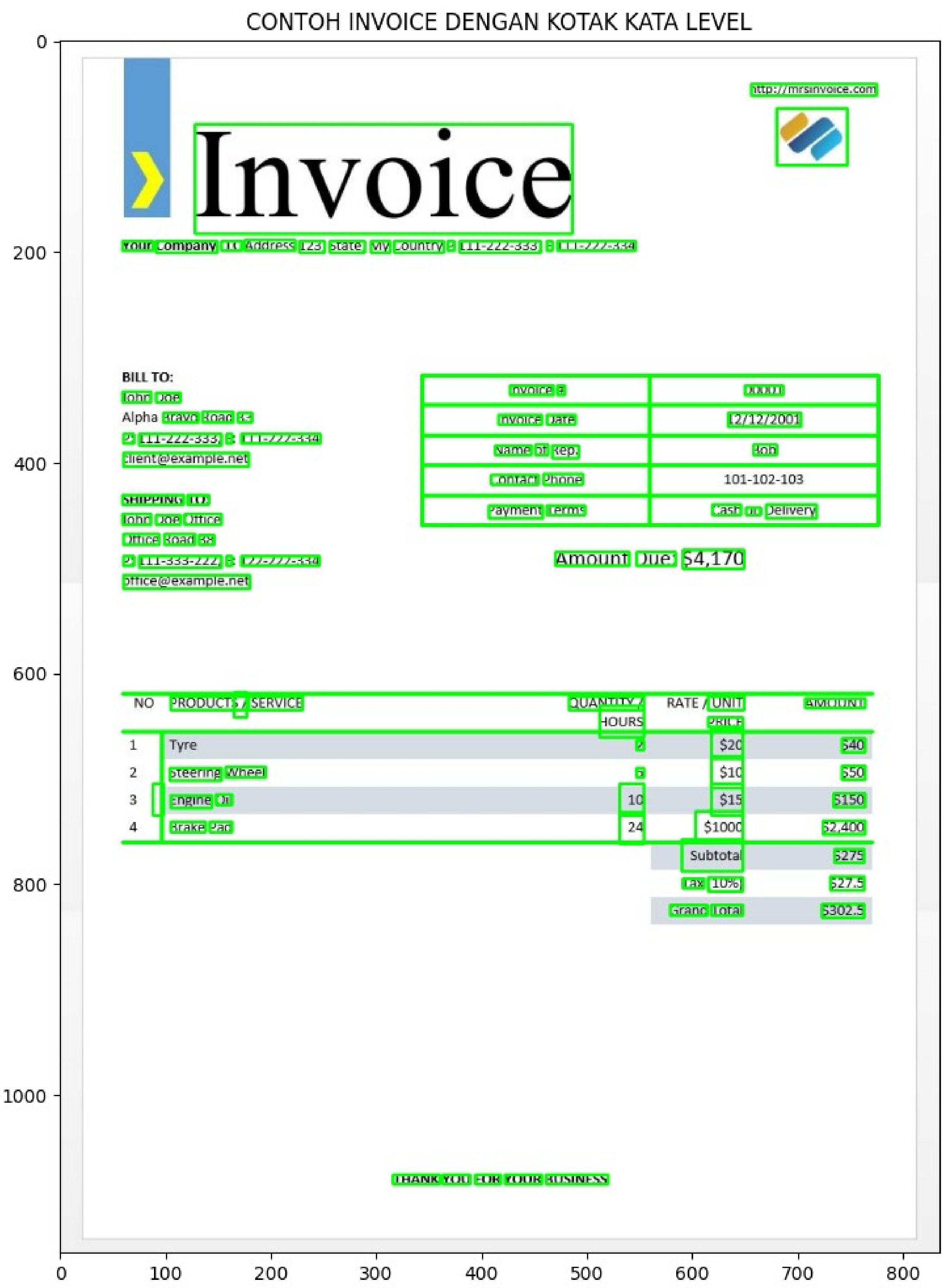
Bounding Box - Level Kata

```
# Membaca gambar contoh invoice
image = cv2.imread(IMG_DIR + '/invoice-sample.jpg')

# Menggunakan pytesseract.image_to_data() untuk mendapatkan data teks
dari gambar
d = pytesseract.image_to_data(image, output_type=Output.DICT)

# Menampilkan kunci-kunci data yang diperoleh dari hasil OCR
print('DATA KEYS: \n', d.keys())
```

```
DATA KEYS:
 dict_keys(['level', 'page_num', 'block_num', 'par_num', 'line_num',
'word_num', 'left', 'top', 'width', 'height', 'conf', 'text'])
```

```
n_boxes = len(d['text'])
for i in range(n_boxes):
    # Kondisi untuk hanya memilih kotak dengan kepercayaan > 60%
    if int(d['conf'][i]) > 60:
        # Mendapatkan koordinat dan ukuran kotak kata
        (x, y, w, h) = (d['left'][i], d['top'][i], d['width'][i],
d['height'][i])
        # Membuat kotak pada gambar untuk kata dengan kepercayaan >
60%
        image = cv2.rectangle(image, (x, y), (x + w, y + h), (0, 255,
0), 2)

# Memisahkan channel warna untuk mengonversi dari BGR ke RGB
b, g, r = cv2.split(image)
rgb_img = cv2.merge([r, g, b])

# Menampilkan gambar dengan kotak kata berdasarkan kepercayaan > 60%
plt.figure(figsize=(16, 12))
plt.imshow(rgb_img)
plt.title('CONTOH INVOICE DENGAN KOTAK KATA LEVEL')
plt.show()
```

CONTOH INVOICE DENGAN KOTAK KATA LEVEL



http://mrsinvoice.com

Invoice

Your Company LLC Address 123, State, My Country P: 111-222-333, F: 111-222-334

BILL TO:
John Doe
Alpha Bravo Road 33
P: 111-222-333, F: 111-222-334
client@example.net

SHIPPING TO:
John Doe Office
Office Road 38
P: 111-333-222, F: 122-222-334
office@example.net

Invoice #	00001
Invoice Date	12/12/2001
Name of Rep.	Bob
Contact Phone	101-102-103
Payment Terms	Cash on Delivery

Amount Due: $4,170

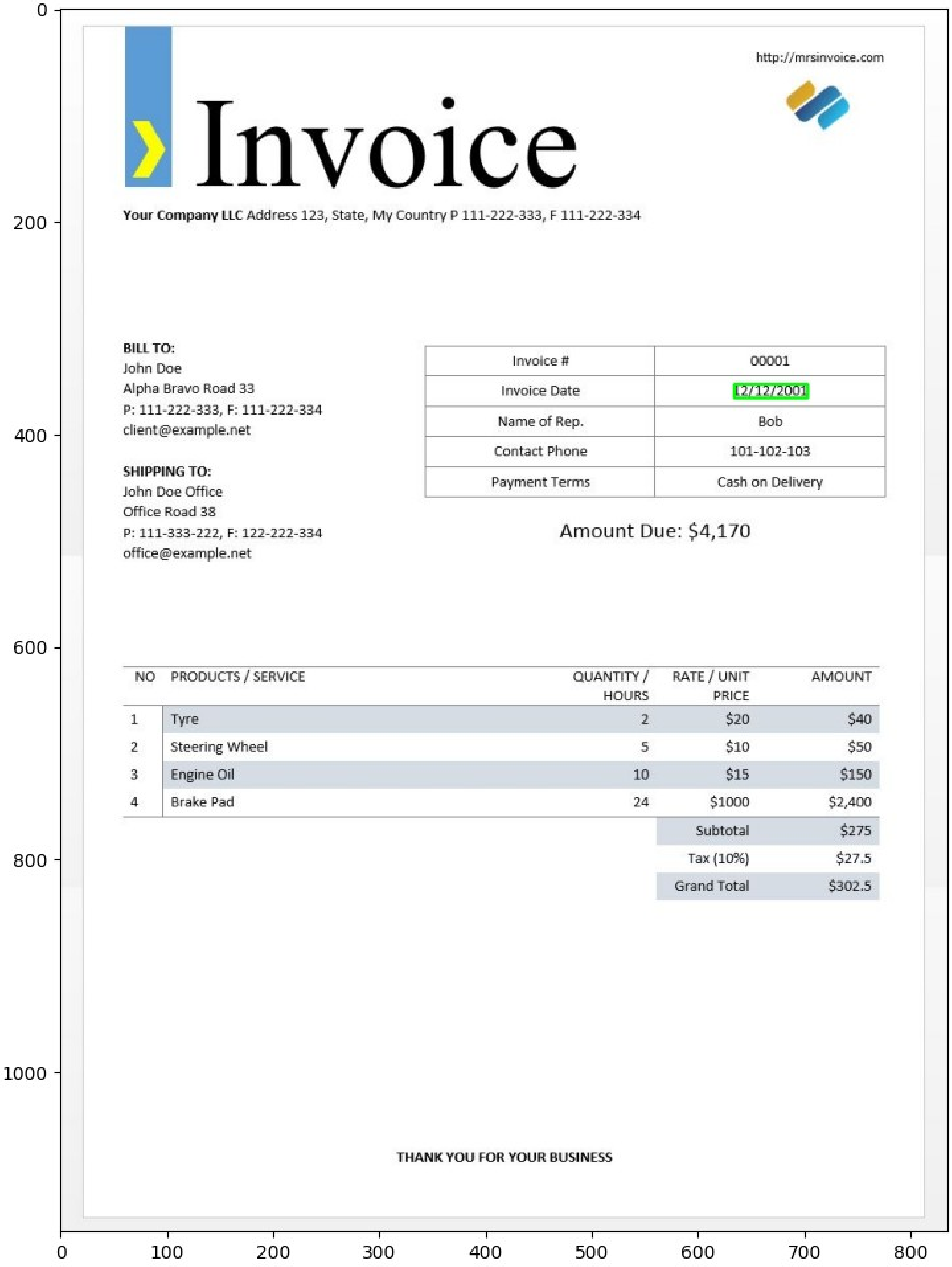
NO	PRODUCTS / SERVICE	QUANTITY / HOURS	RATE / UNIT PRICE	AMOUNT
1	Tyre	2	$20	$40
2	Steering Wheel	5	$10	$50
3	Engine Oil	10	$15	$150
4	Brake Pad	24	$1000	$2,400
			Subtotal	$275
			Tax 10%	$27.5
			Grand Total	$302.5

THANK YOU FOR YOUR BUSINESS

Text template matching - Pola Regex

```
image = cv2.imread(IMG_DIR + '/invoice-sample.jpg')

# Pola tanggal dalam format dd/mm/yyyy
date_pattern = '^(0[1-9]|[12][0-9]|3[01])/(0[1-9]|1[012])/(19|20)\d\
d$'

n_boxes = len(d['text'])
for i in range(n_boxes):
    # Memeriksa apakah kotak memiliki tingkat kepercayaan lebih dari
60%
    if int(d['conf'][i]) > 60:
        # Memeriksa apakah teks di dalam kotak sesuai dengan pola
tanggal
        if re.match(date_pattern, d['text'][i]):
            # Mendapatkan koordinat dan ukuran kotak kata
            (x, y, w, h) = (d['left'][i], d['top'][i], d['width'][i],
d['height'][i])
            # Membuat kotak pada gambar untuk tanggal yang sesuai
dengan pola
            image = cv2.rectangle(image, (x, y), (x + w, y + h), (0,
255, 0), 2)

# Memisahkan channel warna untuk mengonversi dari BGR ke RGB
b, g, r = cv2.split(image)
rgb_img = cv2.merge([r, g, b])

# Menampilkan gambar dengan kotak-kotak yang menandai lokasi tanggal
plt.figure(figsize=(16, 12))
plt.imshow(rgb_img)
plt.title('CONTOH INVOICE DENGAN KOTAK UNTUK TANGGAL')
plt.show()
```

CONTOH INVOICE DENGAN KOTAK UNTUK TANGGAL


0
200
400
600
800
1000
0
100
200
300
400
500
600
700
800

http://mrsinvoice.com

Invoice

Your Company LLC Address 123, State, My Country P 111-222-333, F 111-222-334

BILL TO:
John Doe
Alpha Bravo Road 33
P: 111-222-333, F: 111-222-334
client@example.net

SHIPPING TO:
John Doe Office
Office Road 38
P: 111-333-222, F: 122-222-334
office@example.net

Invoice #	00001
Invoice Date	12/12/2001
Name of Rep.	Bob
Contact Phone	101-102-103
Payment Terms	Cash on Delivery

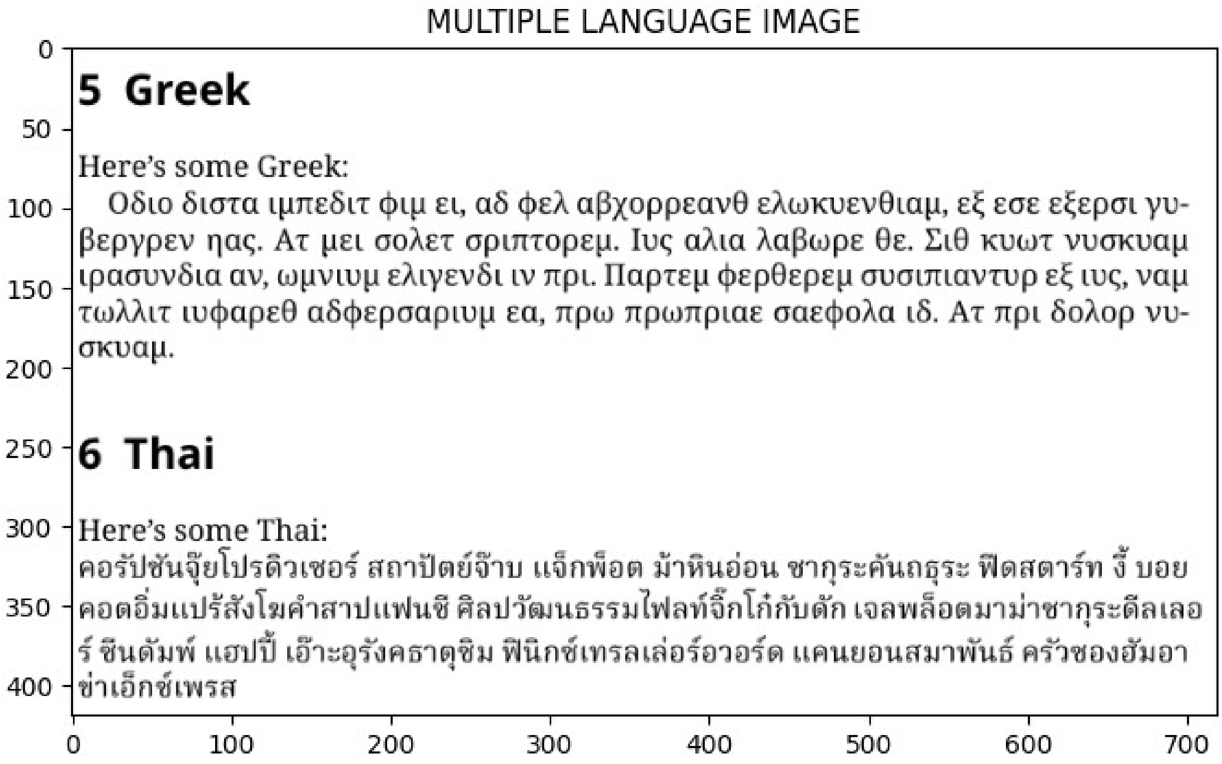
Amount Due: $4,170

NO	PRODUCTS / SERVICE	QUANTITY / HOURS	RATE / UNIT PRICE	AMOUNT
1	Tyre	2	$20	$40
2	Steering Wheel	5	$10	$50
3	Engine Oil	10	$15	$150
4	Brake Pad	24	$1000	$2,400
			Subtotal	$275
			Tax (10%)	$27.5
			Grand Total	$302.5

THANK YOU FOR YOUR BUSINESS

Deteksi berbagai bahasa - OCR

```
# Membaca gambar asli
image = cv2.imread(IMG_DIR + '/greek-thai.png')

# Memisahkan channel warna untuk mengonversi dari BGR ke RGB
b, g, r = cv2.split(image)
rgb_img = cv2.merge([r, g, b])

# Menampilkan gambar asli
plt.figure(figsize=(8, 16))
plt.imshow(rgb_img, cmap='gray')
plt.title('MULTIPLE LANGUAGE IMAGE')
plt.show()
```


MULTIPLE LANGUAGE IMAGE
0
50
100
150
200
250
300
350
400
0
100
200
300
400
500
600
700

5 Greek

Here’s some Greek:

Οδιο διστα ιμπεδιτ φιμ ει, αδ φελ αβχορρεανθ ελωκυενθιαμ, εξ εσε εξερσι γυβεργρεν ηας. Ατ μει σολετ σριπτορεμ. Ιυς αλια λαβωρε θε. Σιθ κυωτ νυσκυαμ ιρασυνδια αν, ωμνιυμ ελιγενδι ιν πρι. Παρτεμ φερθερεμ συσιπιαντυρ εξ ιυς, ναμ τωλλιτ ιυφαρεθ αδφερσαριυμ εα, πρω πρωπριαε σαεφολα ιδ. Ατ πρι δολορ νυσκυαμ.

6 Thai

Here’s some Thai:

คอรัปชันจุ๊ยโปรดิวเซอร์ สถาปัตย์จ๊าบ แจ็กพ็อต ม้าหินอ่อน ซากุระคันถธุระ ฟิดสตาร์ท งี้ บอยคอตอิ่มแปร้สังโฆคำสาปแฟนซี ศิลปวัฒนธรรมไฟลท์จิ๊กโก๋กับดัก เจลพล็อตมาม่าซากุระดีลเลอร์ ซีนดัมพ์ แฮปปี้ เอ๊าะอุรังคธาตุซิม ฟินิกซ์เทรลเล่อร์อวอร์ด แคนยอนสมาพันธ์ ครัวซองฮัมอาข่าเอ็กซ์เพรส

Deteksi berbagai bahasa - OCR

```
# Output dengan hanya bahasa Inggris yang ditentukan

# Konfigurasi khusus dengan hanya bahasa Inggris yang diatur
custom_config = r'-l eng --oem 3 --psm 6'

# Menggunakan pytesseract.image_to_string() untuk mendapatkan teks
dari gambar dengan konfigurasi khusus
print(pytesseract.image_to_string(image, config=custom_config))
```

```
5 Greek
Here’s some Greek:

OSto Stota tuMedit Huy et, aS ea aByoppeave edAwkvevOuay, e§ ece
efepot yu-
Bepypev nas. AT Wel GoAET apiTtopen. Tug aAta AaBwpe Ve. LO KUWT
VUoKLaL
(paovvéia av, WUVLUU eAtyevil tv mpL TMaptep bepSepey GvaTLAaVTUp e€€
LUG, Va
TWAALT LUdaped ASdepoapluy Ea, TOW TpwTplae Gaedoda 16. AT mpt SoAop
vv-
oxvau.

6 Thai

Here’s some Thai: .
aosUsugulushawos amiaddw usntioa sinfudou winszduagss Haaonsn 3 vos
AonduusaladrawunudAavTausssulwavianlAdudn wandoamnsiwinsedataa
$ Guduvi woud rvaseasiadu Windinsadosor0sa uausouanrwus aswouduer
didadinsa
```

Tugas

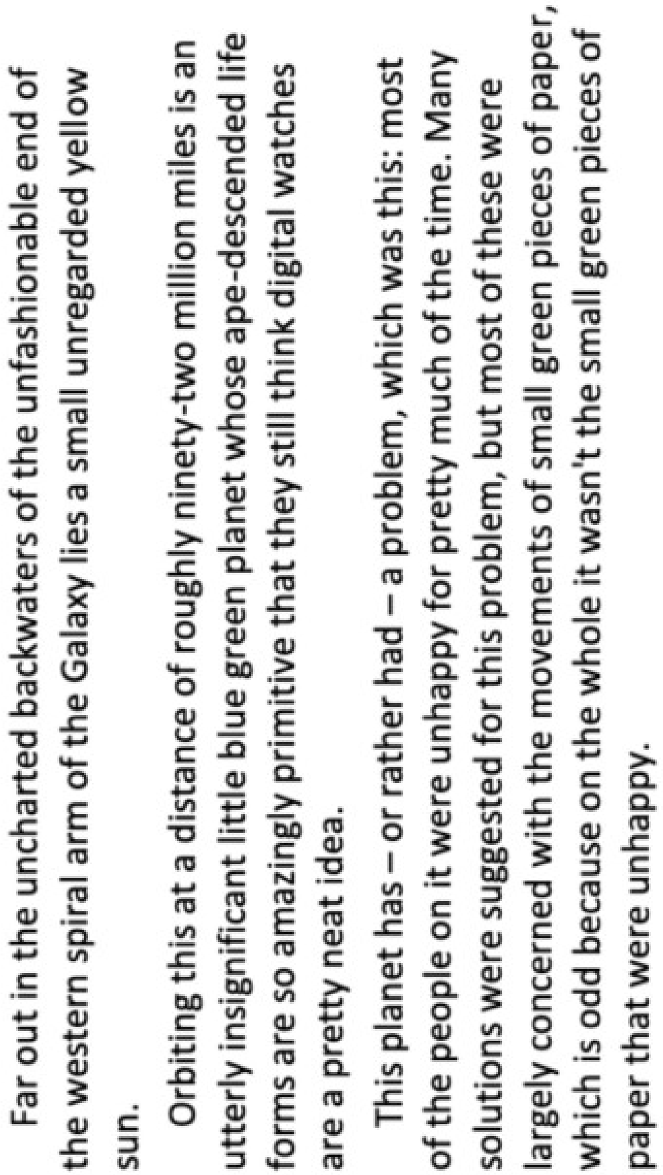
1. Persiapan Gambar:
- Gunakan gambar contoh yang disediakan ('hitchhikers-rotated.png').
- Tampilkan gambar asli menggunakan Python dan OpenCV.

1. Deteksi Orientasi dan Skrip:
- Implementasikan skrip Python untuk mendeteksi orientasi teks dalam gambar.
- Gunakan Tesseract untuk mendapatkan sudut rotasi (angle) dan jenis skrip (script).
- Tampilkan hasil orientasi dan jenis skrip.

```python
import cv2
import pytesseract
from matplotlib import pyplot as plt

# Path ke gambar contoh (gunakan nama file yang diberikan pada tugas)
image_path = '/content/drive/MyDrive/ML/images-ocr/images/hitchhikers-rotated.png'

# Membaca gambar
image = cv2.imread(image_path)

# Menampilkan gambar asli
```

```python
plt.figure(figsize=(8, 8))
plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
plt.title("Gambar Asli")
plt.axis('off')
plt.show()

# Menggunakan Tesseract untuk mendeteksi orientasi dan jenis skrip
# Konfigurasi untuk mendapatkan data orientasi dan skrip
config = "--psm 0"
detection_data = pytesseract.image_to_osd(image, config=config)

# Menampilkan hasil deteksi orientasi dan jenis skrip
print("Hasil Deteksi Tesseract:")
print(detection_data)
```

Gambar Asli



Far out in the uncharted backwaters of the unfashionable end of the western spiral arm of the Galaxy lies a small unregarded yellow sun.

Orbiting this at a distance of roughly ninety-two million miles is an utterly insignificant little blue green planet whose ape-descended life forms are so amazingly primitive that they still think digital watches are a pretty neat idea.

This planet has – or rather had – a problem, which was this: most of the people on it were unhappy for pretty much of the time. Many solutions were suggested for this problem, but most of these were largely concerned with the movements of small green pieces of paper, which is odd because on the whole it wasn't the small green pieces of paper that were unhappy.

```
Hasil Deteksi Tesseract:
Page number: 0
Orientation in degrees: 270
Rotate: 90
Orientation confidence: 17.90
Script: Latin
Script confidence: 2.25
```