

# Modul Praktikum Data Mining



## **Tim Penyusun:**

Dr. Rakhmat Arianto, S.ST., M.Kom

Ir. Rudy Ariyanto, ST., M.Cs

Prof. Dr. Eng. Rosa Andrie Asmara, ST., MT

**Jurusan Teknologi Informasi**

**Sistem Informasi Bisnis**

**Politeknik Negeri Malang**

**Maret 2025**

## DAFTAR ISI

DAFTAR ISI .....	2
JOBSHEET 6 Pelabelan Data.....	3
Pendahuluan .....	3
Tujuan Praktikum.....	3
Peralatan yang dibutuhkan .....	3
Praktikum .....	6

## JOBSHEET 6

### Pelabelan Data

#### Pendahuluan

Modul ini menjelaskan proses perbaikan data secara manual.

#### Tujuan Praktikum

Setelah menyelesaikan praktikum ini, mahasiswa mampu:

- Memahami tentang pelabelan data.

#### Peralatan yang dibutuhkan

Beberapa peralatan yang dibutuhkan dalam menyelesaikan praktikum ini adalah:

- Aplikasi Microsoft Excel
- Google Colab
- Google Drive
- Koneksi Internet
- Browser Web

Proses melakukan pelabelan data pada dasarnya dibagi menjadi 2 cara umum, yaitu secara otomatis menggunakan model Machine Learning atau secara manual. Pada materi kali ini kita akan berfokus untuk melakukan pelabelan data secara manual. Pada pelabelan secara manual sendiri, terbagi menjadi tiga metode yang digunakan pada dataset berukuran kecil, data dengan pola jelas, atau dataset besar.

1. Pada dataset berukuran kecil, kita dapat menggunakan pelabelan secara manual. Tahapan yang dilakukan adalah dengan melihat, membaca, mendengar data dan memberi data secara langsung. Alat yang digunakan biasanya menggunakan tool worksheet seperti Excel, Google Sheets, Labelbox, Roboflow, atau alat open-source seperti LabelImg (untuk citra). Sebagai contoh, seorang editor membaca 100 artikel berita dan memberi label apakah berita itu "politik", "hiburan", atau "olahraga".
2. Pada dataset dengan pola yang jelas, kita dapat gunakan Rule-Based Labelling. Caranya adalah dengan Membuat aturan sederhana berbasis logika atau keyword untuk menentukan label. Sebagai contoh Jika ingin memberi label pada email sebagai "spam" atau "bukan spam", kita bisa menggunakan aturan:

- a. Jika email mengandung kata seperti "gratis", "diskon besar", "transfer uang", maka labelkan sebagai "spam".
- b. Jika tidak, labelkan sebagai "non-spam".

```
def label_email(text):
    spam_keywords = ["gratis", "diskon besar", "transfer uang", "hadiah"]
    for word in spam_keywords:
        if word in text.lower():
            return "spam"
    return "non-spam"

emails = ["Dapatkan diskon besar sekarang!", "Halo, bagaimana kabarmu?", "Anda memenangkan hadiah"]
labels = [label_email(email) for email in emails]

for email, label in zip(emails, labels):
    print(f"Email: {email} -> Label: {label}")
```

3. Metode manual berikutnya adalah melakukan Clustering secara mandiri. Caranya adalah dengan mengelompokkan data yang mirip satu sama lain, lalu memberi label pada tiap kelompok. Untuk mempermudah dapat dilakukan secara manual dengan bantuan alat visualisasi seperti Excel atau Tableau.
4. Crowdsourcing (Pelabelan oleh banyak orang). Caranya adalah dengan Menggunakan banyak orang untuk memberi label pada data. Bisa dilakukan dengan platform seperti: Amazon Mechanical Turk, Appen, Toloka. Setiap label diberikan oleh beberapa orang, lalu diambil mayoritasnya. Sebagai contoh, perusahaan e-commerce ingin memberi label pada 10.000 gambar produk. Mereka membayar pekerja online untuk memberi label apakah gambar itu baju, sepatu, atau aksesoris.

Berikut ini beberapa contoh Crowdsourcing populer yang pernah dilakukan:

1. ReCaptcha: Crowdsourcing untuk digitalisasi buku. Awalnya dikembangkan oleh Luis von Ahn (pencipta CAPTCHA), reCAPTCHA digunakan untuk membedakan manusia dan bot di internet. Google kemudian mengakuisisi reCAPTCHA dan menggunakannya untuk membantu digitalisasi buku melalui crowdsourcing. reCAPTCHA membantu mendigitalisasi jutaan buku dari Google Books dan arsip New York Times. Lebih dari 200 juta CAPTCHA diketik oleh manusia setiap hari, membantu membaca teks yang sulit dikenali komputer. Berikut adalah tahapannya:
  - Banyak buku lama yang dipindai ke bentuk digital, tetapi beberapa kata sulit dikenali oleh sistem OCR (Optical Character Recognition).
  - Setiap kali pengguna internet diminta untuk mengisi CAPTCHA, mereka sebenarnya membantu mengetik ulang kata-kata dari buku yang tidak dapat dibaca oleh komputer.

- Jika banyak orang mengetik jawaban yang sama, kata tersebut dianggap benar dan masuk ke dalam versi digital buku tersebut.
2. Galaxy Zoo: Crowdsourcing untuk Astronomi. Proyek ini adalah crowdsourcing untuk mengklasifikasikan galaksi dalam gambar yang diambil oleh teleskop luar angkasa. Astronom memiliki jutaan gambar galaksi, tetapi terlalu banyak untuk dianalisis oleh segelintir ilmuwan. Ribuan orang membantu mengklasifikasikan lebih dari 50 juta galaksi. Beberapa penemuan baru, termasuk galaksi aneh seperti Hanny's Voorwerp, ditemukan oleh sukarelawan. Setiap orang biasa dapat berkontribusi dengan melihat gambar galaksi dan memberi label bentuknya (spiral, elips, dll.). Website Galaxy Zoo memungkinkan siapa saja membantu sains hanya dengan melihat gambar dan mengklik pilihan yang sesuai.
  3. Wikipedia: Ensiklopedia Crowdsourcing Terbesar. Wikipedia adalah ensiklopedia yang dibuat sepenuhnya oleh pengguna internet. Siapa saja dapat menulis, mengedit, atau memperbaiki artikel. Wikipedia kini memiliki lebih dari 6 juta artikel dalam bahasa Inggris saja. Saat ini menjadi salah satu sumber pengetahuan terbesar yang gratis dan terbuka untuk semua orang. Orang-orang dari seluruh dunia dapat berkontribusi dalam berbagai bahasa. Komunitas editor menjaga kualitas dengan mengoreksi kesalahan dan menambahkan sumber terpercaya.
  4. ImageNet adalah kumpulan lebih dari 14 juta gambar yang dikategorikan ke dalam lebih dari 20.000 kategori objek, seperti anjing, mobil, pesawat, dan lainnya. Project ini dinisiasi oleh Fei-Fei Li dan timnya di Stanford University pada tahun 2009. Tujuan utama: Membantu melatih dan mengembangkan algoritma pengenalan gambar berbasis machine learning. Gambar dikumpulkan dari internet menggunakan search engine seperti Google dan Flickr. Pekerja crowdsourcing dari Amazon Mechanical Turk (MTurk) diberi tugas untuk memberi label pada gambar berdasarkan kategori yang telah ditentukan. Setiap gambar dicek oleh beberapa pekerja untuk memastikan akurasi. Dengan adanya dataset ini, memunculkan kompetisi ImageNet Challenge (ILSVRC - ImageNet Large Scale Visual Recognition Challenge) untuk membuat model terbaik, Peneliti AI berlomba untuk membuat model terbaik dalam mengenali dan mengklasifikasikan gambar. Pada tahun 2012, tim dari University of Toronto menggunakan deep learning (CNN - Convolutional Neural Networks) yang disebut AlexNet, yang mengalahkan metode lain dengan akurasi jauh lebih tinggi. Keberhasilan AlexNet memicu revolusi deep learning yang kita kenal sekarang.

## Praktikum

### LATIHAN

1. Anda diberikan dataset berisi 10 ulasan pelanggan dalam format teks. Gunakan Excel, Google Sheets, atau Python (Pandas) untuk menambahkan label di kolom baru. Simpan hasil dalam format CSV atau XLSX. Tugas Anda adalah memberi label secara manual untuk mengklasifikasikan ulasan tersebut ke dalam kategori berikut:

- a. Positif
- b. Negatif
- c. Netral

No	Ulasan
1	"Produk ini sangat bagus! Saya sangat puas dengan kualitasnya."
2	"Pengiriman sangat lambat, saya kecewa dengan layanan ini."
3	"Barang sesuai dengan deskripsi, cukup oke."
4	"Kualitas jelek dan tidak sesuai gambar!"
5	"Lumayan, tapi saya berharap lebih baik lagi."
6	"Sangat direkomendasikan! Barang bagus dan harga terjangkau."
7	"Mengecewakan, produk yang saya terima berbeda dari yang dijanjikan."
8	"Biasa saja, tidak terlalu bagus tapi juga tidak jelek."
9	"Terlalu mahal untuk kualitas yang didapat, tidak worth it!"
10	"Fast response, packing rapi, dan produk berkualitas tinggi!"

2. Buat aturan sederhana untuk memberi label pada email sebagai "spam" atau "non-spam" berdasarkan kata kunci tertentu. Gunakan Python (Regex atau Pandas) untuk menerapkan aturan berbasis kata kunci seperti "hadiah", "diskon", "pinjaman". Jika email mengandung kata-kata tersebut, beri label "spam", jika tidak, beri label "non-spam". Simpan hasilnya dalam format CSV atau JSON. Anda diberikan dataset dengan kolom berikut:

No	Email
1	"Selamat! Anda memenangkan hadiah besar! Klik link ini untuk klaim hadiah Anda."
2	"Rapat tim akan diadakan besok pukul 10 pagi di ruang konferensi."
3	"Dapatkan diskon besar hingga 70%! Penawaran terbatas!"
4	"Mohon konfirmasi kehadiran Anda dalam acara minggu depan."
5	"Anda terpilih untuk mendapatkan pinjaman tanpa bunga! Hubungi kami segera!"
6	"Akun Anda terdeteksi mengalami aktivitas mencurigakan. Segera periksa keamanan!"
7	"Kartu kredit Anda mendapatkan penawaran khusus hari ini! Jangan lewatkan kesemua ini."
8	"Saya butuh laporan keuangan untuk rapat sore ini, bisa dikirim sekarang?"
9	"Tawaran eksklusif: Beli 1 gratis 1 hanya hari ini!"

10

"Berikut adalah notulen rapat kemarin. Mohon ditinjau dan berikan masukan"


3.

Dataset berikut berisi **data transaksi pelanggan**. Kelompokkan transaksi berdasarkan pola pembelian yang mirip. Gunakan Excel atau Google Sheets untuk mengelompokkan data ke dalam kategori (misalnya: "Pelanggan Besar", "Pelanggan Reguler", "Pelanggan Kecil"). Buat tabel baru dengan label kategori yang diberikan berdasarkan analisis Anda. Simpan hasil dalam format CSV atau XLSX.

No	Nama	Produk	Total Belanja (Rp)	Frekuensi Belanja (per bulan)
1	Andi	Elektronik	5.000.000	2
2	Budi	Pakaian	500.000	6
3	Citra	Elektronik	7.500.000	1
4	Dedi	Makanan	200.000	10
5	Ema	Pakaian	1.000.000	4
6	Fajar	Elektronik	4.500.000	3
7	Gina	Makanan	300.000	8
8	Hadi	Pakaian	700.000	5
9	Intan	Makanan	250.000	12
10	Joko	Elektronik	6.000.000	2

4.

Buat 10 gambar hewan (anjing dan kucing) yang perlu dilabeli, gambar dapat diambil dari Google Images, Pexels, Unsplash, atau platform berbagi gambar lainnya yang gratis. Berikan pelabelan yang akan dilakukan dengan bantuan LabelImg atau Roboflow. Berikan dua kategori yaitu "Anjing" dan "Kucing". Setiap gambar harus dilabeli oleh minimal 3 orang untuk meningkatkan akurasi. Simpan hasil dalam format Pascal VOC XML atau COCO JSON. Contoh table:

No	Gambar	Label
1		Anjing / Kucing

## 1. Pelebelan Ulasan

```
# Import library yang diperlukan
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from google.colab import drive

# Mount Google Drive untuk menyimpan hasil
drive.mount('/content/drive')

# Buat DataFrame dengan data ulasan
data_ulasan = {
    'No': range(1, 11),
    'Ulasan': [
        "Produk ini sangat bagus! Saya sangat puas dengan kualitasnya.",
        "Pengiriman sangat lambat, saya kecewa dengan layanan ini.",
        "Barang sesuai dengan deskripsi, cukup oke.",
        "Kualitas jelek dan tidak sesuai gambar!",
        "Lumayan, tapi saya berharap lebih baik lagi.",
        "Sangat direkomendasikan! Barang bagus dan harga terjangkau.",
        "Mengecewakan, produk yang saya terima berbeda dari yang dijanjikan.",
        "Biasa saja, tidak terlalu bagus tapi juga tidak jelek.",
        "Terlalu mahal untuk kualitas yang didapat, tidak worth it!",
        "Fast response, packing rapi, dan produk berkualitas tinggi!"
    ]
}

# Buat DataFrame
df_ulasan = pd.DataFrame(data_ulasan)

# Lakukan pelabelan manual
df_ulasan['Label'] = ['Positif', 'Negatif', 'Netral', 'Negatif', 'Netral',
                     'Positif', 'Negatif', 'Netral', 'Negatif', 'Positif']

# Tampilkan hasil
print("Hasil Pelabelan Manual Ulasan Pelanggan:")
print(df_ulasan)

# Simpan hasil ke CSV di Google Drive
df_ulasan.to_csv('/content/drive/MyDrive/hasil_pelabelan_ulasan.csv', index=False)
print("\nHasil telah disimpan di Google Drive sebagai 'hasil_pelabelan_ulasan.csv'")
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount,
Hasil Pelabelan Manual Ulasan Pelanggan:
```

No	Ulasan	Label
0 1	Produk ini sangat bagus! Saya sangat puas deng...	Positif
1 2	Pengiriman sangat lambat, saya kecewa dengan l...	Negatif
2 3	Barang sesuai dengan deskripsi, cukup oke.	Netral
3 4	Kualitas jelek dan tidak sesuai gambar!	Negatif
4 5	Lumayan, tapi saya berharap lebih baik lagi.	Netral
5 6	Sangat direkomendasikan! Barang bagus dan harg...	Positif
6 7	Mengecewakan, produk yang saya terima berbeda ...	Negatif
7 8	Biasa saja, tidak terlalu bagus tapi juga tida...	Netral
8 9	Terlalu mahal untuk kualitas yang didapat, tid...	Negatif
9 10	Fast response, packing rapi, dan produk berkua...	Positif

Hasil telah disimpan di Google Drive sebagai 'hasil\_pelabelan\_ulasan.csv'



hasil\_pelabelan\_ulasan - Excel

File Home Insert Page Layout Formulas Data Review View Help

Paste Clipboard Font Alignment

B18

No	Ulasan	Label
1	Produk ini sangat bagus! Saya sangat puas dengan kualitasnya.	Positif
2	Pengiriman sangat lambat, saya kecewa dengan layanan ini.	Negatif
3	Barang sesuai dengan deskripsi, cukup oke.	Netral
4	Kualitas jelek dan tidak sesuai gambar!	Negatif
5	Lumayan, tapi saya berharap lebih baik lagi.	Netral
6	Sangat direkomendasikan! Barang bagus dan harga terjangkau.	Positif
7	Mengecewakan, produk yang saya terima berbeda dari yang dijanjikan.	Negatif
8	Biasa saja, tidak terlalu bagus tapi juga tidak jelek.	Netral
9	Terlalu mahal untuk kualitas yang didapat, tidak worth it!	Negatif
10	Fast response, packing rapi, dan produk berkualitas tinggi!	Positif

## 2. Pelebelan Email

```
# Import library yang diperlukan
import pandas as pd
import re
from google.colab import drive

# Mount Google Drive jika belum di-mount
try:
    drive.mount('/content/drive')
except:
    print("Google Drive sudah di-mount")

# Buat DataFrame dengan data email
data_email = {
    'No': range(1, 11),
    'Email': [
        "Selamat! Anda memenangkan hadiah besar! Klik link ini untuk klaim hadiah Anda.",
        "Rapat tim akan diadakan besok pukul 10 pagi di ruang konferensi.",
        "Dapatkan diskon besar hingga 70%! Penawaran terbatas!",
        "Mohon konfirmasi kehadiran Anda dalam acara minggu depan.",
        "Anda terpilih untuk mendapatkan pinjaman tanpa bunga! Hubungi kami segera!",
        "Akun Anda terdeteksi mengalami aktivitas mencurigakan. Segera periksa keamanan!",
        "Kartu kredit Anda mendapatkan penawaran khusus hari ini! Jangan lewatkan kesempatan ini.",
        "Saya butuh laporan keuangan untuk rapat sore ini, bisa dikirim sekarang?",
        "Tawaran eksklusif: Beli 1 gratis 1 hanya hari ini!",
        "Berikut adalah notulen rapat kemarin. Mohon ditinjau dan berikan masukan."
    ]
}

# Buat DataFrame
df_email = pd.DataFrame(data_email)

# Definisikan kata kunci spam
kata_kunci_spam = ["hadiah", "diskon", "pinjaman", "tawaran", "gratis", "penawaran", "eksklusif", "kartu kredit"]

# Fungsi untuk memeriksa apakah email mengandung kata kunci spam
def cek_spam(email):
    email_lower = email.lower()
    return any(kata in email_lower for kata in kata_kunci_spam)

# Terapkan fungsi untuk melabeli email
df_email['Label'] = df_email['Email'].apply(lambda x: "Spam" if cek_spam(x) else "Non-Spam")

# Tampilkan hasil
print("Hasil Pelabelan Email Berbasis Aturan:")
print(df_email)

# Simpan hasil ke CSV di Google Drive
df_email.to_csv('/content/drive/MyDrive/hasil_deteksi_spam.csv', index=False)
print("\nHasil telah disimpan di Google Drive sebagai 'hasil_deteksi_spam.csv'")
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount
Hasil Pelabelan Email Berbasis Aturan:
```

No	Email	Label
0 1	Selamat! Anda memenangkan hadiah besar! Klik l...	Spam
1 2	Rapat tim akan diadakan besok pukul 10 pagi di...	Non-Spam
2 3	Dapatkan diskon besar hingga 70%! Penawaran te...	Spam
3 4	Mohon konfirmasi kehadiran Anda dalam acara mi...	Non-Spam
4 5	Anda terpilih untuk mendapatkan pinjaman tanpa...	Spam
5 6	Akun Anda terdeteksi mengalami aktivitas mencu...	Non-Spam
6 7	Kartu kredit Anda mendapatkan penawaran khusus...	Spam
7 8	Saya butuh laporan keuangan untuk rapat sore i...	Non-Spam
8 9	Tawaran eksklusif: Beli 1 gratis 1 hanya hari ...	Spam
9 10	Berikut adalah notulen rapat kemarin. Mohon di...	Non-Spam

Hasil telah disimpan di Google Drive sebagai 'hasil\_deteksi\_spam.csv'

hasil\_deteksi\_spam - Excel

Search (Alt+Q)

File Home Insert Page Layout Formulas Data Review View Help

Clipboard Font Alignment Number

No	Email	Label
1	Selamat! Anda memenangkan hadiah besar! Klik link ini untuk klaim hadiah Anda.	Spam
2	Rapat tim akan diadakan besok pukul 10 pagi di ruang konferensi.	Non-Spam
3	Dapatkan diskon besar hingga 70%! Penawaran terbatas!	Spam
4	Mohon konfirmasi kehadiran Anda dalam acara minggu depan.	Non-Spam
5	Anda terpilih untuk mendapatkan pinjaman tanpa bunga! Hubungi kami segera!	Spam
6	Akun Anda terdeteksi mengalami aktivitas mencurigakan. Segera periksa keamanan!	Non-Spam
7	Kartu kredit Anda mendapatkan penawaran khusus hari ini! Jangan lewatkan kesempatan ini.	Spam
8	Saya butuh laporan keuangan untuk rapat sore ini, bisa dikirim sekarang?	Non-Spam
9	Tawaran eksklusif: Beli 1 gratis 1 hanya hari ini!	Spam
10	Berikut adalah notulen rapat kemarin. Mohon ditinjau dan berikan masukan.	Non-Spam

### 3. Pengelompokan Pelanggan

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from google.colab import drive

# Mount Google Drive jika belum di-mount
try:
    drive.mount('/content/drive')
except:
    print("Google Drive sudah di-mount")

# Buat DataFrame dengan data transaksi pelanggan
data_pelanggan = {
    'No': range(1, 11),
    'Nama': ["Andi", "Budi", "Citra", "Dedi", "Ema", "Fajar", "Gina", "Hadi", "Intan", "Joko"],
    'Produk': ["Elektronik", "Pakaian", "Elektronik", "Makanan", "Pakaian",
               "Elektronik", "Makanan", "Pakaian", "Makanan", "Elektronik"],
    'Total_Belanja': [5000000, 500000, 7500000, 200000, 1000000,
                      4500000, 300000, 700000, 250000, 6000000],
    'Frekuensi': [2, 6, 1, 10, 4, 3, 8, 5, 12, 2]
}
```

```
# Buat DataFrame
df_pelanggan = pd.DataFrame(data_pelanggan)

# Definisikan fungsi untuk menentukan kategori pelanggan
def tentukan_kategori(row):
    # Pelanggan dengan belanja tinggi (> 3.000.000)
    if row['Total_Belanja'] > 3000000:
        return "Pelanggan Besar"
    # Pelanggan dengan belanja rendah (< 500.000) tapi frekuensi tinggi (> 7)
    elif row['Total_Belanja'] < 500000 and row['Frekuensi'] > 7:
        return "Pelanggan Loyal"
    # Pelanggan dengan belanja menengah dan frekuensi menengah
    elif 500000 <= row['Total_Belanja'] <= 3000000:
        return "Pelanggan Reguler"
    # Pelanggan dengan belanja rendah dan frekuensi rendah
    else:
        return "Pelanggan Kecil"

# Terapkan fungsi untuk melabeli pelanggan
df_pelanggan['kategori'] = df_pelanggan.apply(tentukan_kategori, axis=1)
print("Hasil Pengelompokan Pelanggan:")
print(df_pelanggan)

# Simpan hasil ke CSV di Google Drive
df_pelanggan.to_csv('/content/drive/MyDrive/hasil_pengelompokan_pelanggan.csv', index=False)
print("\nHasil telah disimpan di Google Drive sebagai 'hasil_pengelompokan_pelanggan.csv'")
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount('/content/drive')

Hasil Pengelompokan Pelanggan:

No	Nama	Produk	Total_Belanja	Frekuensi	Kategori
0	1	Andi Elektronik	5000000	2	Pelanggan Besar
1	2	Budi Pakaian	500000	6	Pelanggan Reguler
2	3	Citra Elektronik	7500000	1	Pelanggan Besar
3	4	Dedi Makanan	200000	10	Pelanggan Loyal
4	5	Ema Pakaian	1000000	4	Pelanggan Reguler
5	6	Fajar Elektronik	4500000	3	Pelanggan Besar
6	7	Gina Makanan	300000	8	Pelanggan Loyal
7	8	Hadi Pakaian	700000	5	Pelanggan Reguler
8	9	Intan Makanan	250000	12	Pelanggan Loyal
9	10	Joko Elektronik	6000000	2	Pelanggan Besar

Hasil telah disimpan di Google Drive sebagai 'hasil\_pengelompokan\_pelanggan.csv'

hasil\_pengelompokan\_pelanggan - Excel

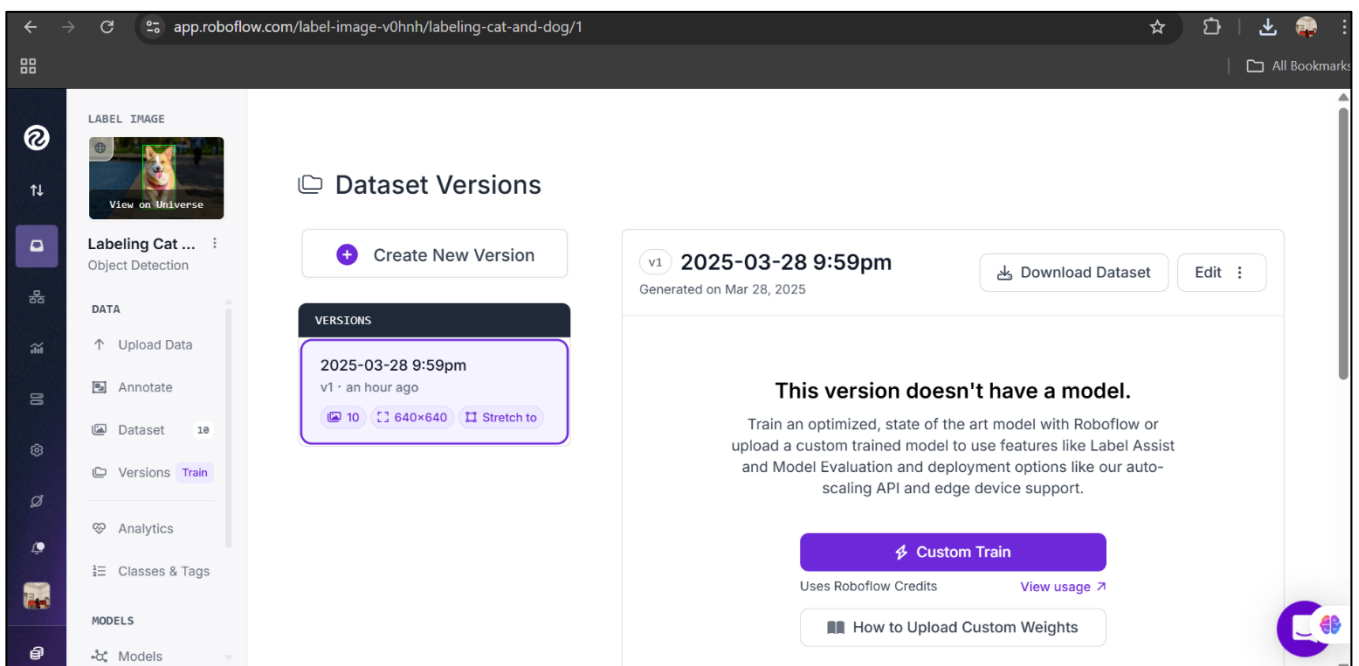
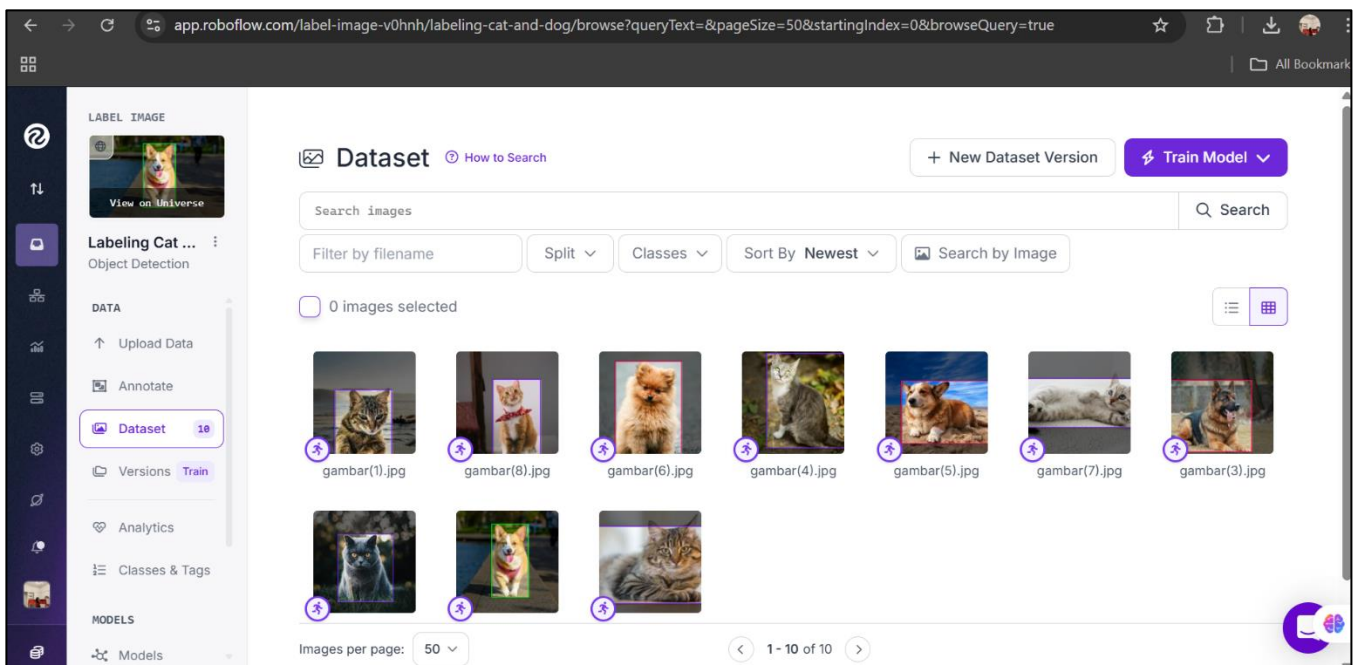
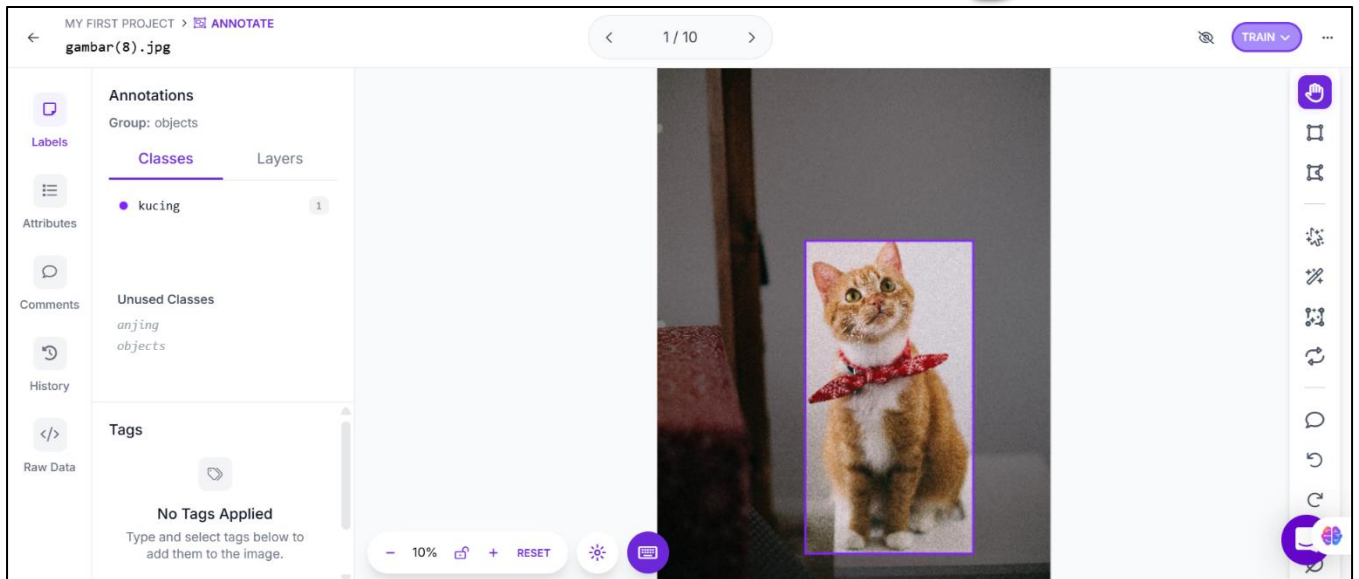
File Home Insert Page Layout Formulas Data Review View

Clipboard Font Alignment

M13

	A	B	C	D	E	F	G
1	No	Nama	Produk	Total_Bela	Frekuensi	Kategori	
2	1	Andi	Elektronik	5000000	2	Pelanggan Besar	
3	2	Budi	Pakaian	500000	6	Pelanggan Reguler	
4	3	Citra	Elektronik	7500000	1	Pelanggan Besar	
5	4	Dedi	Makanan	200000	10	Pelanggan Loyal	
6	5	Ema	Pakaian	1000000	4	Pelanggan Reguler	
7	6	Fajar	Elektronik	4500000	3	Pelanggan Besar	
8	7	Gina	Makanan	300000	8	Pelanggan Loyal	
9	8	Hadi	Pakaian	700000	5	Pelanggan Reguler	
10	9	Intan	Makanan	250000	12	Pelanggan Loyal	
11	10	Joko	Elektronik	6000000	2	Pelanggan Besar	
12							

#### 4. Pelebelan Gambar Anjing dan Kucing



```

# Install Roboflow (jika belum)
!pip install roboflow

# Import library yang diperlukan
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from roboflow import Roboflow
from google.colab import drive
import json
import os
from PIL import Image
import requests
from io import BytesIO

# Mount Google Drive untuk menyimpan hasil
drive.mount('/content/drive')

# Inisialisasi Roboflow dengan API key Anda
rf = Roboflow(api_key="Tf08487FIhlsAgj0bT9e")
project = rf.workspace("label-image-v0hnh").project("labeling-cat-and-dog")
version = project.version(1)
dataset = version.download("coco")

# Path ke file anotasi COCO JSON
annotations_path = f"{dataset.location}/train/_annotations.coco.json"

# Baca file anotasi
with open(annotations_path, 'r') as f:
    coco_data = json.load(f)

# Ekstrak informasi gambar
images_info = {img['id']: img for img in coco_data['images']}

# Ekstrak informasi kategori (label)
categories_info = {cat['id']: cat['name'] for cat in coco_data['categories']}

# Buat dictionary untuk menyimpan anotasi per gambar
image_annotations = {}

# Kelompokkan anotasi berdasarkan gambar
for ann in coco_data['annotations']:
    image_id = ann['image_id']
    if image_id not in image_annotations:
        image_annotations[image_id] = []

    category_name = categories_info[ann['category_id']]
    bbox = ann['bbox'] # [x, y, width, height]

    image_annotations[image_id].append({
        'label': category_name,
        'bbox': bbox
    })

# Buat data untuk tabel
table_data = []
for img_id, img_info in images_info.items():
    annotations = image_annotations.get(img_id, [])
    labels = [ann['label'] for ann in annotations]

    table_data.append({
        'Image ID': img_id,
        'Filename': img_info['file_name'],
        'Width': img_info['width'],
        'Height': img_info['height'],
        'Labels': ', '.join(labels),
        'Annotation Count': len(annotations)
    })

# Buat DataFrame
df_images = pd.DataFrame(table_data)

# Tampilkan tabel
print("Tabel Hasil Pelabelan Gambar:")
display(df_images)

```



Tabel Hasil Pelabelan Gambar:

	Image ID	Filename	Width	Height	Labels	Annotation Count
0	0	gambar-10-.jpg.rf.6d68f8e50343cdda018bf15f8e77...	640	640	kucing	1
1	1	gambar-2-.jpg.rf.b6e7598bea8fe211442ea04307a7c...	640	640	kucing	1
2	2	gambar-6-.jpg.rf.c9923fcd66371993af03230518716...	640	640	anjing	1
3	3	gambar-7-.jpg.rf.138917749f8606d2da8a2679075a8...	640	640	kucing	1
4	4	gambar-3-.jpg.rf.e51435372d6abe99b28599b65520c...	640	640	anjing	1
5	5	gambar-9-.jpg.rf.7319e36bfc4225986e666c7fda29c...	640	640	anjing	1
6	6	gambar-8-.jpg.rf.088f9e7528766ec33aa74b280dee8...	640	640	kucing	1
7	7	gambar-1-.jpg.rf.cf7574208740de5cd82557e60677d...	640	640	kucing	1
8	8	gambar-5-.jpg.rf.378a7e6700246d1f3de821deeb177...	640	640	anjing	1
9	9	gambar-4-.jpg.rf.5afeec33b2be53a718f43e636414b...	640	640	kucing	1

```
# Fungsi untuk menampilkan gambar dengan bounding box
def display_image_with_annotations(image_path, annotations):
    # Baca gambar
    img = Image.open(image_path)
    plt.figure(figsize=(10, 8))
    plt.imshow(img)

    # Plot bounding box
    for ann in annotations:
        bbox = ann['bbox'] # [x, y, width, height]
        x, y, w, h = bbox

        # Buat rectangle
        rect = plt.Rectangle((x, y), w, h, linewidth=2, edgecolor='r', facecolor='none')
        plt.gca().add_patch(rect)

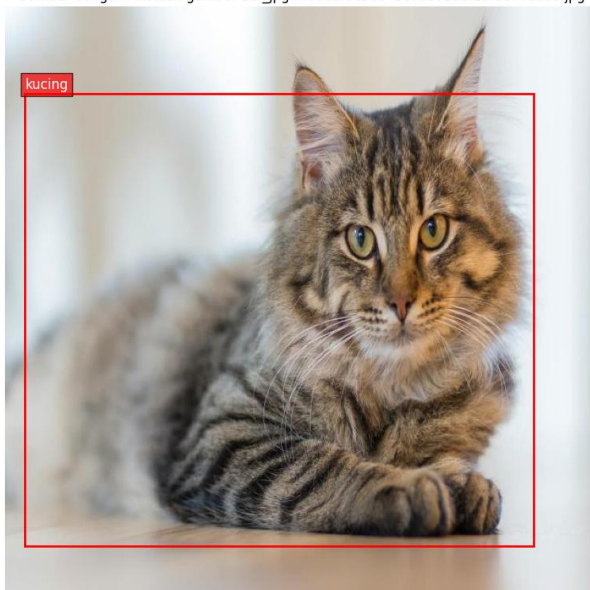
        # Tambahkan label
        plt.text(x, y-5, ann['label'], color='white', fontsize=12,
                bbox=dict(facecolor='red', alpha=0.7))

    plt.axis('off')
    plt.title(f"Gambar dengan Anotasi: {os.path.basename(image_path)}")
    plt.tight_layout()
    plt.show()

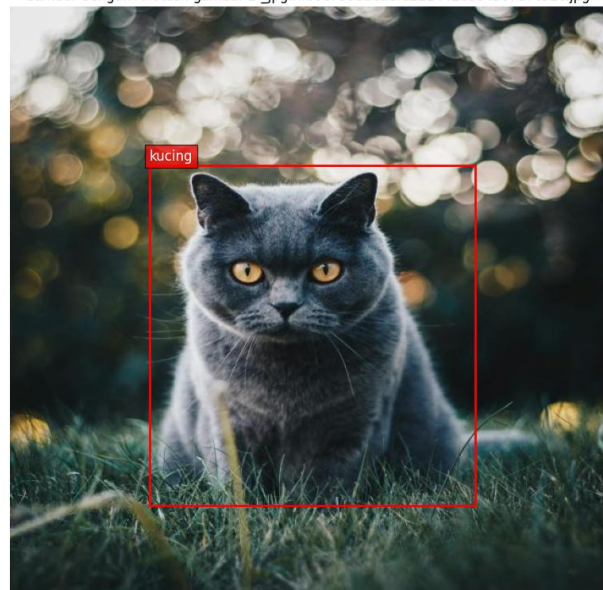
# Tampilkan beberapa gambar dengan anotasi
for i, (img_id, img_info) in enumerate(list(images_info.items())[:10]):
    image_path = f"{dataset.location}/train/{img_info['file_name']}"
    annotations = image_annotations.get(img_id, [])

    if os.path.exists(image_path) and annotations:
        display_image_with_annotations(image_path, annotations)
```

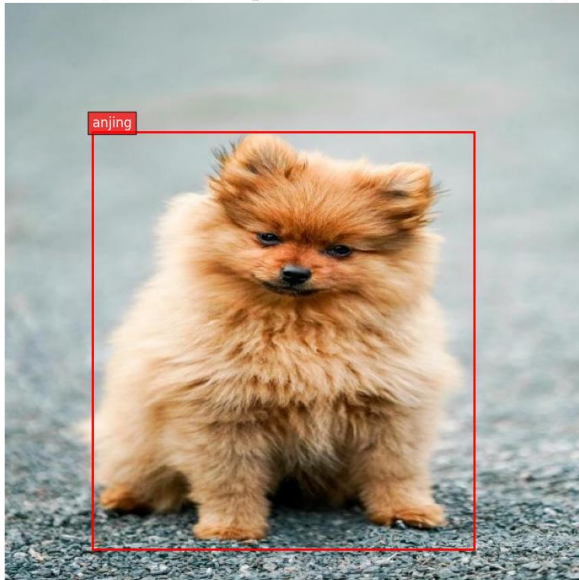
Gambar dengan Anotasi: gambar-10-.jpg.rf.6d68f8e50343cdda018bf15f8e77de5b.jpg



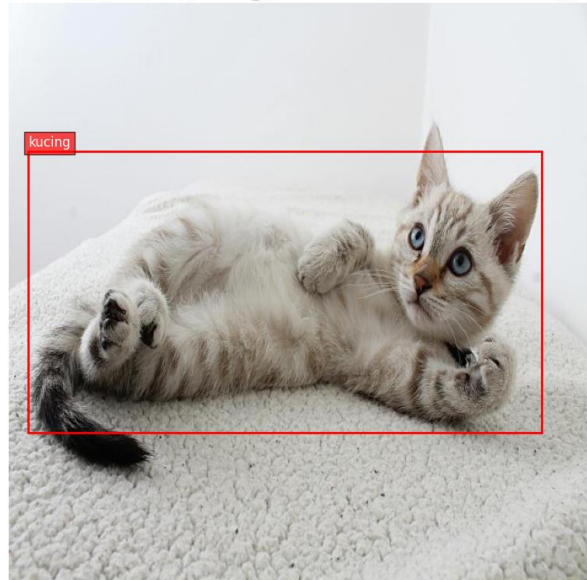
Gambar dengan Anotasi: gambar-2-.jpg.rf.b6e7598bea8fe211442ea04307a7c52c.jpg



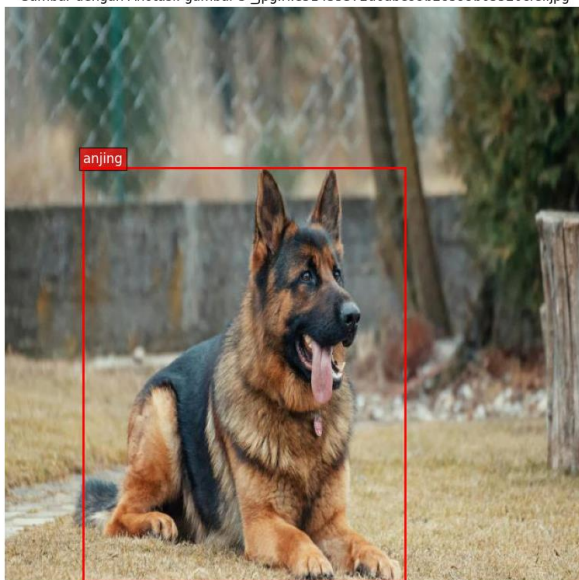
Gambar dengan Anotasi: gambar-6\_.jpg.rf.c9923fcd66371993af032305187166f6.jpg



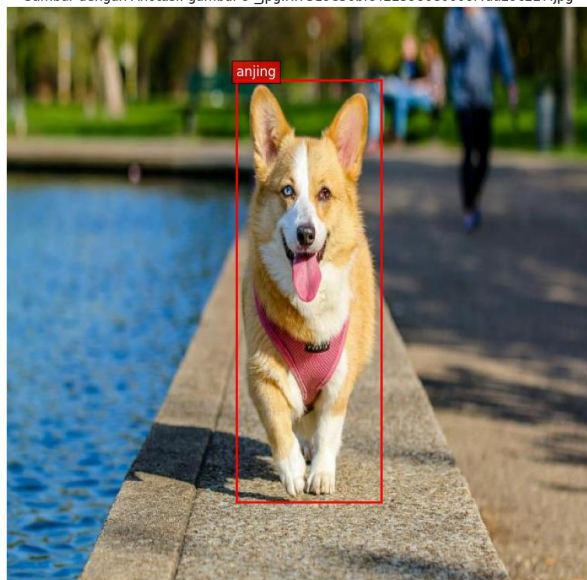
Gambar dengan Anotasi: gambar-7-.jpg.rf.138917749f8606d2da8a2679075a8145.jpg



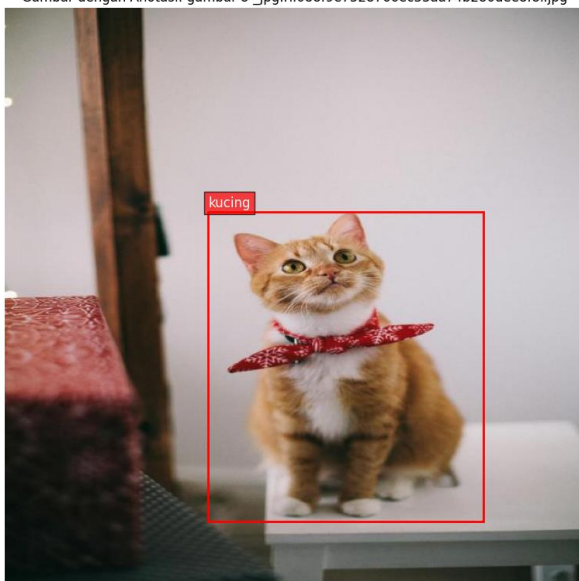
Gambar dengan Anotasi: gambar-3\_.jpg.rf.e51435372d6abe99b28599b65520cfef.jpg



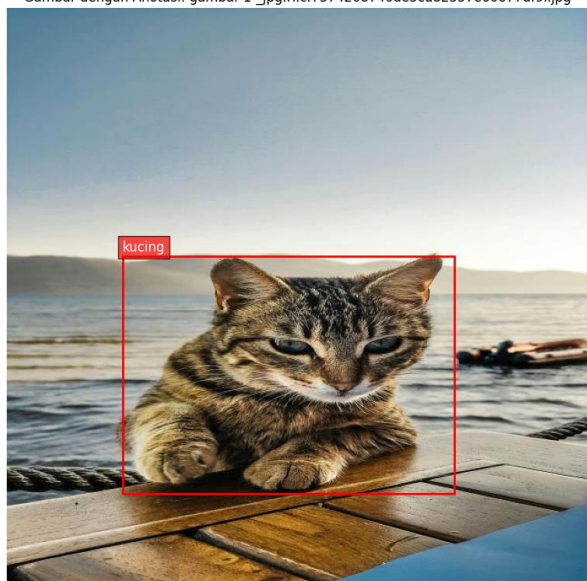
Gambar dengan Anotasi: gambar-9-.jpg.rf.7319e36bfc4225986e666c7fda29c227.jpg



Gambar dengan Anotasi: gambar-8\_.jpg.rf.088f9e7528766ec33aa74b280dee8f8f.jpg

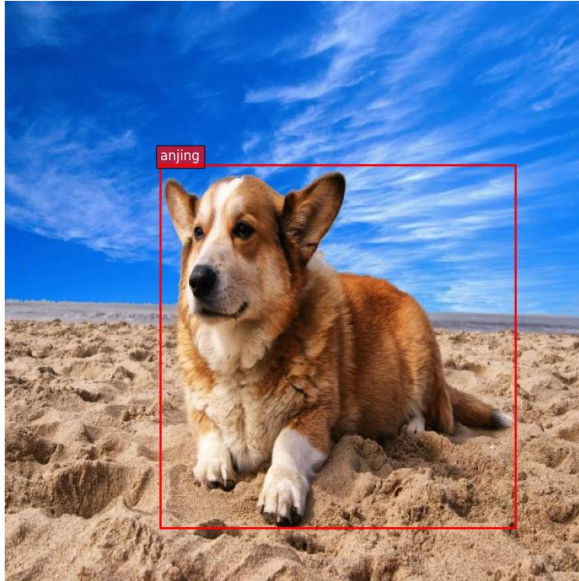


Gambar dengan Anotasi: gambar-1\_.jpg.rf.cf7574208740de5cd82557e60677df9f.jpg

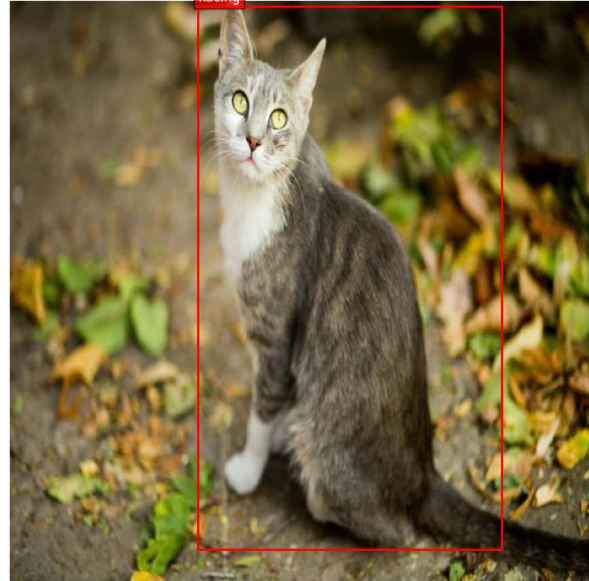




Gambar dengan Anotasi: gambar-5- \_jpg.rf.378a7e6700246d1f3de821deeb17776d.jpg



Gambar dengan Anotasi: gambar-4- \_jpg.rf.5afeec33b2be53a718f43e636414b29c.jpg



hasil\_pelabelan\_robotflow - Excel

File Home Insert Page Layout Formulas Data Review View Help

Clipboard Font Alignment Number Conditional Formatting

J13

	A	B	C	D	E	F
	Image ID	Filename	Width	Height	Labels	Annotation Count
1	0	gambar-10- _jpg.rf.6d68f8e50343cdda018bf15f8e77de5b.jpg	640	640	kucing	1
2	1	gambar-2- _jpg.rf.b6e7598bea8fe211442ea04307a7c52c.jpg	640	640	kucing	1
3	2	gambar-6- _jpg.rf.c9923fcd66371993af032305187166f6.jpg	640	640	anjing	1
4	3	gambar-7- _jpg.rf.138917749f8606d2da8a2679075a8145.jpg	640	640	kucing	1
5	4	gambar-3- _jpg.rf.e51435372d6abe99b28599b65520cfef.jpg	640	640	anjing	1
6	5	gambar-9- _jpg.rf.7319e36bfc4225986e666c7fda29c227.jpg	640	640	anjing	1
7	6	gambar-8- _jpg.rf.088f9e7528766ec33aa74b280dee8f8f.jpg	640	640	kucing	1
8	7	gambar-1- _jpg.rf.cf7574208740de5cd82557e60677df9f.jpg	640	640	kucing	1
9	8	gambar-5- _jpg.rf.378a7e6700246d1f3de821deeb17776d.jpg	640	640	anjing	1
10	9	gambar-4- _jpg.rf.5afeec33b2be53a718f43e636414b29c.jpg	640	640	kucing	1
11						
12						