# Crawl data dari twitter @TMCPoldaMetro

twitter\_auth\_token = 'b33af24ed78827356029e26971c6159a4ea5cb5d'

# Import required Python package  
!pip install pandas  
  
# Install Node.js (because tweet-harvest built using Node.js)  
!sudo apt-get update  
!sudo apt-get install -y ca-certificates curl gnupg  
!sudo mkdir -p /etc/apt/keyrings  
!curl -fsSL https://deb.nodesource.com/gpgkey/nodesource-repo.gpg.key | sudo gpg --dearmor -o /etc/apt/keyrings/nodesource.gpg  
  
!NODE\_MAJOR=20 && echo "deb [signed-by=/etc/apt/keyrings/nodesource.gpg] https://deb.nodesource.com/node\_$NODE\_MAJOR.x nodistro main" | sudo tee /etc/apt/sources.list.d/nodesource.list  
  
!sudo apt-get update  
!sudo apt-get install nodejs -y  
  
!node -v

## filter berdasarkan kata kunci "Situasi arus lalu lintas di"

from datetime import datetime, timedelta  
  
# Generate date range: today and 365 days ago  
until\_date = datetime.now().date()  
since\_date = until\_date - timedelta(days=365)  
  
# Define target  
username = "TMCPoldaMetro"  
keyword = '"Situasi arus lalu lintas di"'  
filename = 'tmc\_polda.csv'  
  
search\_keyword = f'from:{username} {keyword} since:{since\_date} until:{until\_date} lang:id'  
limit = 800 # adjust this depending on how much data you want  
  
# Run tweet-harvest  
!npx -y tweet-harvest@2.6.1 -o "{filename}" -s "{search\_keyword}" --tab "LATEST" -l {limit} --token {twitter\_auth\_token}

## filter data yang memiliki pola "HH.MM Situasi arus lalu lintas di"

import pandas as pd  
import re  
  
# 1. Memuat data  
df = pd.read\_csv('tweets-data/tmc\_polda.csv')  
  
# 2. Definisikan pola regex yang akan dicari  
# r'\d{2}:\d{2}' mencari pola waktu (jam:menit (HH:MM))  
# diikuti dengan spasi dan teks "Situasi arus lalu lintas"  
pattern = r'\d{2}.\d{2} Situasi arus lalu lintas di'  
  
# 3. Filter DataFrame  
# - df['full\_text'].str.contains() digunakan untuk mencari string/pola dalam Series/kolom.  
# - regex=True memberitahu pandas bahwa 'pattern' adalah sebuah regular expression.  
# - na=False akan memperlakukan nilai NaN (jika ada) sebagai False (tidak cocok).  
mask = df['full\_text'].str.contains(pattern, regex=True, na=False)  
  
# 4. Mengambil kolom 'created\_at' dan 'full\_text' dari data yang sudah terfilter  
# Kita ambil kedua kolom yang kita butuhkan.  
# Gunakan .copy() untuk menghindari SettingWithCopyWarning saat kita memodifikasi DataFrame nanti.  
lalu\_lintas = df.loc[mask, ['created\_at', 'full\_text']].copy()  
  
# 5. Konversi kolom 'created\_at' ke format datetime, ubah timezone, dan format ulang  
# a. pd.to\_datetime(): Mengubah string tanggal menjadi objek datetime. Pandas cerdas  
# dan akan secara otomatis mengenali format "+0000" sebagai UTC.  
# b. .dt.tz\_convert('Asia/Jakarta'): Mengonversi timezone dari UTC ke WIB (UTC+7).  
# c. .dt.strftime('%d %b %Y'): Memformat ulang objek datetime menjadi string dengan format  
# Hari(angka) Bulan(singkat) Tahun(lengkap). Contoh: "23 Jun 2025"  
lalu\_lintas['created\_at'] = pd.to\_datetime(lalu\_lintas['created\_at']).dt.tz\_convert('Asia/Jakarta').dt.strftime('%d %b %Y')  
  
# 6. Mengganti nama kolom  
lalu\_lintas = lalu\_lintas.rename(columns={  
 'created\_at': 'date',  
 'full\_text': 'text'  
})  
  
# 7. Menyimpan DataFrame yang sudah berisi dua kolom ke file csv baru  
lalu\_lintas.to\_csv('lalu\_lintas.csv', index=False)

# Ekstraksi menggunakan metode In-NER

import pandas as pd  
import re  
import json

# Memecah sebuah kalimat (teks) menjadi daftar kata-kata atau "token".  
def custom\_tokenizer(text):  
 if not isinstance(text, str): return []  
 # Menghapus url dari text  
 urls = re.findall(r'https://t.co/\S+', text)  
 text\_no\_urls = re.sub(r'https://t.co/\S+', '', text)  
 tokens = [token for token in re.split(r'([,.()])?\s+', text\_no\_urls) if token]  
 return tokens

# Membersihkan string lokasi yang sudah diekstraksi agar formatnya konsisten.  
def clean\_location(loc\_str):  
 if not loc\_str: return None  
 # Menghapus kata depan seperti di, dari, ke, menuju.  
 loc\_str = re.sub(r'^(di|dari|ke|menuju)\s+', '', loc\_str.strip())  
 # Menghapus spasi atau tanda baca yang tidak perlu di awal dan akhir string.  
 loc\_str = loc\_str.strip(' ,.()')  
 # Mengubah string menjadi format Title Case (setiap kata diawali huruf kapital), kecuali jika string tersebut sudah mengandung "Jl."  
 return loc\_str if 'Jl.' in loc\_str else loc\_str.title()

# Pola untuk mengenali waktu  
TIME\_PATTERN = re.compile(r'^\d{2}\.\d{2}$')  
  
# Kata-kata yang biasanya mengawali sebuah frasa lokasi  
LOC\_STARTERS = {'di', 'dari', 'depan', 'kawasan', 'on', 'exit'}  
  
# Kata-kata kunci yang menandakan lokasi  
LOC\_KEYWORDS = {  
 'arteri', 'bundaran', 'csw', 'exit', 'gbk', 'gerbang', 'graha', 'gt', 'interchange',  
 'jakbar', 'jakpus', 'jaksel', 'jaktim', 'jakut', 'jembatan', 'jl', 'jlnt', 'junction',  
 'km', 'kolong', 'lampu', 'light', 'merah', 'monas', 'mrt', 'off', 'pasar', 'pgc',  
 'pintu', 'pospol', 'ramp', 'ruas', 'semanggi', 'simpang', 'susun', 'terminal', 'tl',  
 'tmii', 'tugu', 'tol', 'traffic', 'underpass', 'wisata'  
}  
  
# Kata-kata yang menggambarkan kondisi lalu lintas  
STATUS\_KEYWORDS = {'cukup', 'cenderung', 'lancar', 'mengalir', 'normal', 'padat', 'ramai'}  
  
# Kata kunci arah  
DIRECTION\_KEYWORD = {'arah', 'mengarah', 'menuju'}  
  
# Kata kunci dua arah  
BIDIRECTIONAL\_KEYWORDS = {'maupun', 'sebaliknya', 'dan'}  
  
# Kata kunci penyebab hambatan  
OBSTACLE\_CAUSE\_KEYWORDS = {'karena', 'imbas', 'dikarenakan'}  
  
# Kata kunci cuaca  
WEATHER = {'hujan','berawan','gerimis','cerah'}

'''  
Memberikan "tag" atau label pada setiap token yang dihasilkan oleh custom\_tokenizer.  
Label ini mengikuti skema BIO (Beginning, Inside, Outside).  
'''  
# B- (Beginning): Menandai awal dari sebuah entitas.  
# I- (Inside): Menandai bagian dalam atau kelanjutan dari sebuah entitas.  
# O (Outside): Menandai token yang tidak termasuk dalam kategori manapun.  
'''  
menggunakan variabel state (is\_location, is\_status) untuk melacak konteks.  
Jika token sebelumnya diberi label B-LOC, token berikutnya kemungkinan besar adalah I-LOC  
(kecuali jika token tersebut adalah kata pemutus seperti 'dan' atau 'terpantau').  
'''  
def label\_tokens(text):  
 tokens = custom\_tokenizer(text)  
 labels = []  
 is\_location, is\_status = False, False  
  
 for i, tok in enumerate(tokens):  
 lower\_tok = tok.lower()  
 if i == 0: is\_location = is\_status = False  
  
 if tok.startswith('https://t.co/'):  
 labels.append("O")  
 is\_location = is\_status = False  
 elif TIME\_PATTERN.match(tok):  
 labels.append("B-TIME")  
 is\_location = is\_status = False  
 elif lower\_tok in BIDIRECTIONAL\_KEYWORDS:  
 labels.append("B-BIDIR")  
 is\_location = False  
 elif lower\_tok in DIRECTION\_KEYWORD:  
 labels.append("B-DIR")  
 is\_location = False  
 elif lower\_tok in OBSTACLE\_CAUSE\_KEYWORDS:  
 labels.append("B-OBSTACLE-CAUSE")  
 is\_location = is\_status = False  
 elif lower\_tok in LOC\_STARTERS or any(kw in lower\_tok for kw in LOC\_KEYWORDS):  
 labels.append("B-LOC" if not is\_location else "I-LOC")  
 is\_location = True; is\_status = False  
 elif lower\_tok in STATUS\_KEYWORDS:  
 labels.append("B-STATUS" if not is\_status else "I-STATUS")  
 is\_status = True; is\_location = False  
 elif is\_location:  
 if lower\_tok in ['terpantau', 'sedangkan', 'diimbau', 'dan', 'yang']:  
 labels.append("O"); is\_location = False  
 else:  
 labels.append("I-LOC")  
 elif lower\_tok in WEATHER:  
 labels.append("B-WEATHER")  
 is\_location = is\_status = False  
 else:  
 labels.append("O")  
 is\_location = is\_status = False  
 return tokens, labels

'''  
Mengambil hasil dari label\_tokens (daftar token dan labelnya)  
dan mengubahnya menjadi format data yang rapi (dictionary).  
'''  
# Fungsi ini mencari token dengan label B- (Beginning) untuk menemukan awal dari suatu informasi.  
'''  
Ketika B- ditemukan, ia akan terus maju untuk mengumpulkan semua token I- yang mengikutinya,  
lalu menggabungkannya menjadi satu frasa utuh (misalnya, [B-LOC: 'Jl.', I-LOC: 'Gatot', I-LOC: 'Subroto']  
menjadi "Jl. Gatot Subroto").  
'''  
# Setelah semua potongan informasi (waktu, status, lokasi, dll.) terkumpul, fungsi ini mulai membangun record data.  
'''  
Logika utamanya adalah menangani berbagai skenario lokasi:  
Satu Lokasi: Hanya ada satu lokasi yang disebutkan.  
Dua Arah (maupun): "dari A maupun ke B" akan dipecah menjadi dua record: (dari: A, ke: B).  
Bolak-balik (sebaliknya): "A ke B dan arah sebaliknya" akan dipecah menjadi dua record: (dari: A, ke: B) dan (dari: B, ke: A).  
Standar (Dari-Ke): Jika ada dua lokasi, lokasi pertama dianggap from dan yang kedua to.  
'''  
def extract\_from\_tagged(tokens, labels, date\_from\_csv, base\_from=None):  
 time, status, obstacle, weather = None, None, None, None  
 locations, bidir\_flags = [], []  
  
 i = 0  
 while i < len(tokens):  
 tag = labels[i]  
 if tag.startswith("B-"):  
 chunk\_tokens = []; chunk\_type = tag[2:]  
 j = i  
 while j < len(tokens) and (labels[j] == f"B-{chunk\_type}" or labels[j] == f"I-{chunk\_type}"):  
 chunk\_tokens.append(tokens[j]); j += 1  
 full\_chunk = " ".join(chunk\_tokens)  
 if chunk\_type == "TIME": time = full\_chunk.replace('.', ':')  
 elif chunk\_type == "STATUS": status = full\_chunk  
 elif chunk\_type == "LOC": locations.append(clean\_location(full\_chunk))  
 elif chunk\_type == "BIDIR": bidir\_flags.append(tokens[i].lower())  
 elif chunk\_type == "OBSTACLE-CAUSE":  
 obs\_tokens = [tokens[k] for k in range(j, len(tokens)) if tokens[k].lower() not in ['diimbau', 'agar']]  
 if obs\_tokens: obstacle = ' '.join(obs\_tokens).strip(" .")  
 elif chunk\_type == "WEATHER": weather = full\_chunk  
 i = j  
 else: i += 1  
  
 results = []  
  
 if base\_from and not locations: return []  
 if base\_from and not any(re.search(r'di |dari ', loc, re.I) for loc in locations):  
 locations.insert(0, base\_from)  
  
 if not time or not status or not locations: return []  
  
 base\_record = {'time': time, 'date': date\_from\_csv, 'from': None, 'to': None,  
 'status': status, 'obstacle': obstacle, 'weather': weather}  
  
 if len(locations) == 1:  
 results.append({\*\*base\_record, 'from': locations[0], 'to': None})  
 elif 'maupun' in bidir\_flags and len(locations) > 1:  
 origin = locations[0]  
 for dest in locations[1:]:  
 results.append({\*\*base\_record, 'from': origin, 'to': dest})  
 elif 'sebaliknya' in bidir\_flags and len(locations) > 1:  
 loc1, loc2 = locations[0], locations[1]  
 results.append({\*\*base\_record, 'from': loc1, 'to': loc2})  
 results.append({\*\*base\_record, 'from': loc2, 'to': loc1})  
 elif len(locations) >= 2:  
 results.append({\*\*base\_record, 'from': locations[0], 'to': locations[1]})  
  
 return results

# Membaca file lalu\_lintas.csv ke dalam sebuah DataFrame pandas.  
df\_input = pd.read\_csv('lalu\_lintas.csv')  
all\_extracted\_data = []  
base\_origin\_for\_sedangkan = None  
  
# Melakukan loop untuk setiap baris dalam file CSV.  
for index, row in df\_input.iterrows():  
 if not isinstance(row['text'], str): continue  
  
 '''  
 Menangani "sedangkan" kalimat seperti "Tol Cikampek ramai, sedangkan Tol Jagorawi lancar" mengandung dua laporan terpisah.  
 re.split(r'\s+sedangkan\s+', ...) memecah kalimat tersebut menjadi dua bagian, yang kemudian diproses secara terpisah.  
 '''  
 parts = re.split(r'\s+sedangkan\s+', row['text'], flags=re.IGNORECASE)  
  
  
 for i, part in enumerate(parts):  
 tokens, labels = label\_tokens(part)  
  
 '''  
 digunakan untuk menangani kasus seperti "Tol Cikampek arah Jakarta lancar, sedangkan arah sebaliknya padat".  
 Lokasi "Tol Cikampek" dari bagian pertama kalimat akan digunakan kembali sebagai konteks untuk bagian kedua ("arah sebaliknya").  
 '''  
 contextual\_origin = base\_origin\_for\_sedangkan if i > 0 else None  
 extracted\_records = extract\_from\_tagged(tokens, labels, row['date'], contextual\_origin)  
  
 if extracted\_records:  
 # Setiap record data yang berhasil diekstraksi akan ditambahkan ke dalam list  
 all\_extracted\_data.extend(extracted\_records)  
 if i == 0 and extracted\_records[0]['from']:  
 base\_origin\_for\_sedangkan = extracted\_records[0]['from']  
 base\_origin\_for\_sedangkan = None  
  
# Sebelum disimpan, data disaring. Hanya record yang memiliki semua kunci wajib (time, date, from, to, status) yang akan disimpan  
MANDATORY\_KEYS = ['time', 'date', 'from', 'to', 'status']  
final\_data = [  
 rec for rec in all\_extracted\_data if all(rec.get(key) for key in MANDATORY\_KEYS)  
]  
  
# Menyimpan hasil ekstraksi ke JSON  
output\_filename = 'traffic\_data\_inner.json'  
with open(output\_filename, 'w', encoding='utf-8') as f:  
 json.dump(final\_data, f, ensure\_ascii=False, indent=4)

# Ekstraksi menggunakan metode regex

import csv  
import re  
import json  
from datetime import datetime  
  
# --- 1. DEFINISI ENTITAS DAN KATA KUNCI (DENGAN LOKASI YANG DIPERBANYAK) ---  
  
# Tabel Bantu Lokasi yang sudah diperkaya secara signifikan  
LOKASI\_JAKARTA = [  
 # Jakarta Pusat  
 {'nama\_jalan': 'sudirman', 'keywords': ['sudirman', 'jl jend sudirman', 'jendral sudirman', 'fx sudirman'], 'wilayah': 'jakarta pusat'},  
 {'nama\_jalan': 'thamrin', 'keywords': ['thamrin', 'mh thamrin', 'jl mh thamrin'], 'wilayah': 'jakarta pusat'},  
 {'nama\_jalan': 'bundaran hi', 'keywords': ['bundaran hi', 'bunderan hi', 'hotel indonesia', 'plaza gi'], 'wilayah': 'jakarta pusat'},  
 {'nama\_jalan': 'monas', 'keywords': ['monas', 'silang monas', 'monas timur', 'kawasan monas'], 'wilayah': 'jakarta pusat'},  
 {'nama\_jalan': 'patung kuda', 'keywords': ['patung kuda', 'bundaran patung kuda', 'monas barat daya'], 'wilayah': 'jakarta pusat'},  
 {'nama\_jalan': 'gbk', 'keywords': ['gbk', 'gelora bung karno', 'pintu 10 gbk', 'gbk senayan'], 'wilayah': 'jakarta pusat'},  
 {'nama\_jalan': 'senayan', 'keywords': ['senayan', 'bundaran senayan', 'senayan city', 'plaza senayan', 'lapangan tembak senayan'], 'wilayah': 'jakarta pusat'},  
 {'nama\_jalan': 'simpang lima senen', 'keywords': ['simpang lima senen', 'senen', 'stasiun senen', 'ps senen'], 'wilayah': 'jakarta pusat'},  
 {'nama\_jalan': 'cempaka putih', 'keywords': ['cempaka putih'], 'wilayah': 'jakarta pusat'},  
 {'nama\_jalan': 'salemba', 'keywords': ['salemba', 'salemba raya', 'tl carolus'], 'wilayah': 'jakarta pusat'},  
 {'nama\_jalan': 'tugu tani', 'keywords': ['tugu tani'], 'wilayah': 'jakarta pusat'},  
 {'nama\_jalan': 'gambir', 'keywords': ['gambir'], 'wilayah': 'jakarta pusat'},  
 {'nama\_jalan': 'kwitang', 'keywords': ['kwitang'], 'wilayah': 'jakarta pusat'},  
 {'nama\_jalan': 'harmoni', 'keywords': ['harmoni'], 'wilayah': 'jakarta pusat'},  
 {'nama\_jalan': 'kemayoran', 'keywords': ['kemayoran', 'jiexpo kemayoran'], 'wilayah': 'jakarta pusat'},  
 {'nama\_jalan': 'palmerah', 'keywords': ['palmerah'], 'wilayah': 'jakarta pusat'},  
 {'nama\_jalan': 'parman', 'keywords': ['parman', 's parman', 'jl s parman', 'letjend s parman'], 'wilayah': 'jakarta pusat'},  
 {'nama\_jalan': 'pejompongan', 'keywords': ['pejompongan', 'penjernihan'], 'wilayah': 'jakarta pusat'},  
 {'nama\_jalan': 'suryopranoto', 'keywords': ['suryopranoto', 'jl suryopranoto'], 'wilayah': 'jakarta pusat'},  
 {'nama\_jalan': 'gunung sahari', 'keywords': ['gunung sahari', 'jl gunung sahari'], 'wilayah': 'jakarta pusat'},  
 {'nama\_jalan': 'pintu besi', 'keywords': ['pintu besi'], 'wilayah': 'jakarta pusat'},  
 {'nama\_jalan': 'kramat raya', 'keywords': ['kramat raya', 'jl kramat raya'], 'wilayah': 'jakarta pusat'},  
 {'nama\_jalan': 'kramat bunder', 'keywords': ['kramat bunder', 'jl kramat bunder'], 'wilayah': 'jakarta pusat'},  
 {'nama\_jalan': 'letjen suprapto', 'keywords': ['letjen suprapto', 'jl letjen suprapto'], 'wilayah': 'jakarta pusat'},  
 {'nama\_jalan': 'asia afrika', 'keywords': ['asia afrika', 'jl asia afrika'], 'wilayah': 'jakarta pusat'},  
 {'nama\_jalan': 'gerbang pemuda', 'keywords': ['gerbang pemuda', 'jl gerbang pemuda'], 'wilayah': 'jakarta pusat'},  
 {'nama\_jalan': 'biak', 'keywords': ['biak', 'tl biak'], 'wilayah': 'jakarta pusat'},  
 {'nama\_jalan': 'dukuh atas', 'keywords': ['dukuh atas'], 'wilayah': 'jakarta pusat'},  
 {'nama\_jalan': 'tanah abang', 'keywords': ['tanah abang'], 'wilayah': 'jakarta pusat'},  
 {'nama\_jalan': 'sarinah', 'keywords': ['sarinah'], 'wilayah': 'jakarta pusat'},  
 {'nama\_jalan': 'kebon sirih', 'keywords': ['kebon sirih'], 'wilayah': 'jakarta pusat'},  
 {'nama\_jalan': 'wahidin', 'keywords': ['wahidin', 'jl dr wahidin'], 'wilayah': 'jakarta pusat'},  
 {'nama\_jalan': 'karet bivak', 'keywords': ['karet bivak'], 'wilayah': 'jakarta pusat'},  
 {'nama\_jalan': 'dpr/mpr', 'keywords': ['dpr ri', 'dpr/mpr ri', 'gedung dpr/mpr'], 'wilayah': 'jakarta pusat'},  
 {'nama\_jalan': 'lapangan banteng', 'keywords': ['lapangan banteng'], 'wilayah': 'jakarta pusat'},  
  
 # Jakarta Selatan  
 {'nama\_jalan': 'gatot subroto', 'keywords': ['gatot subroto', 'gatsu', 'jl gatot subroto'], 'wilayah': 'jakarta selatan'},  
 {'nama\_jalan': 'kuningan', 'keywords': ['kuningan', 'gt kuningan', 'mall kuningan city', 'rasuna said', 'jl rasuna said', 'hr rasuna said'], 'wilayah': 'jakarta selatan'},  
 {'nama\_jalan': 'pancoran', 'keywords': ['pancoran'], 'wilayah': 'jakarta selatan'},  
 {'nama\_jalan': 'semanggi', 'keywords': ['semanggi', 'plaza semanggi', 'jembatan semanggi', 'gt semanggi', 'off ramp semanggi'], 'wilayah': 'jakarta selatan'},  
 {'nama\_jalan': 'tendean', 'keywords': ['tendean'], 'wilayah': 'jakarta selatan'},  
 {'nama\_jalan': 'fatmawati', 'keywords': ['fatmawati', 'on ramp fatmawati', 'mrt fatmawati'], 'wilayah': 'jakarta selatan'},  
 {'nama\_jalan': 'lebak bulus', 'keywords': ['lebak bulus'], 'wilayah': 'jakarta selatan'},  
 {'nama\_jalan': 'cilandak', 'keywords': ['cilandak', 'gt cilandak'], 'wilayah': 'jakarta selatan'},  
 {'nama\_jalan': 'pasar minggu', 'keywords': ['pasar minggu', 'ps minggu'], 'wilayah': 'jakarta selatan'},  
 {'nama\_jalan': 'kalibata', 'keywords': ['kalibata', 'tl kalibata'], 'wilayah': 'jakarta selatan'},  
 {'nama\_jalan': 'ragunan', 'keywords': ['ragunan', 'taman margasatwa ragunan'], 'wilayah': 'jakarta selatan'},  
 {'nama\_jalan': 'blok m', 'keywords': ['blok m'], 'wilayah': 'jakarta selatan'},  
 {'nama\_jalan': 'mt haryono', 'keywords': ['mt haryono', 'jl mt haryono'], 'wilayah': 'jakarta selatan'},  
 {'nama\_jalan': 'tb simatupang', 'keywords': ['tb simatupang', 'jl tb simatupang'], 'wilayah': 'jakarta selatan'},  
 {'nama\_jalan': 'pondok indah', 'keywords': ['pondok indah', 'pim', 'underpass pim'], 'wilayah': 'jakarta selatan'},  
 {'nama\_jalan': 'bintaro', 'keywords': ['bintaro', 'hankam bintaro'], 'wilayah': 'jakarta selatan'},  
 {'nama\_jalan': 'antasari', 'keywords': ['antasari', 'jlnt antasari'], 'wilayah': 'jakarta selatan'},  
 {'nama\_jalan': 'prapanca', 'keywords': ['prapanca raya'], 'wilayah': 'jakarta selatan'},  
 {'nama\_jalan': 'satrio', 'keywords': ['jl dr satrio', 'jl satrio', 'casablanca', 'jlnt casablanca'], 'wilayah': 'jakarta selatan'},  
 {'nama\_jalan': 'mampang', 'keywords': ['mampang', 'mampang prapatan'], 'wilayah': 'jakarta selatan'},  
 {'nama\_jalan': 'tebet', 'keywords': ['tebet', 'off ramp tebet'], 'wilayah': 'jakarta selatan'},  
 {'nama\_jalan': 'cipete', 'keywords': ['cipete', 'off ramp cipete'], 'wilayah': 'jakarta selatan'},  
 {'nama\_jalan': 'ampera', 'keywords': ['ampera', 'gt ampera'], 'wilayah': 'jakarta selatan'},  
 {'nama\_jalan': 'pakubuwono', 'keywords': ['pakubuwono'], 'wilayah': 'jakarta selatan'},  
 {'nama\_jalan': 'iskandarsyah', 'keywords': ['iskandarsyah raya'], 'wilayah': 'jakarta selatan'},  
 {'nama\_jalan': 'melawai', 'keywords': ['melawai'], 'wilayah': 'jakarta selatan'},  
 {'nama\_jalan': 'panglima polim', 'keywords': ['panglima polim'], 'wilayah': 'jakarta selatan'},  
  
 # Jakarta Timur  
 {'nama\_jalan': 'cawang', 'keywords': ['cawang', 'cawang kompor', 'uki cawang', 'cawang interchange'], 'wilayah': 'jakarta timur'},  
 {'nama\_jalan': 'ahmad yani', 'keywords': ['ahmad yani', 'jl ahmad yani'], 'wilayah': 'jakarta timur'},  
 {'nama\_jalan': 'rawamangun', 'keywords': ['rawamangun', 'mega rawamangun'], 'wilayah': 'jakarta timur'},  
 {'nama\_jalan': 'panjaitan', 'keywords': ['panjaitan', 'jl di panjaitan'], 'wilayah': 'jakarta timur'},  
 {'nama\_jalan': 'matraman', 'keywords': ['matraman'], 'wilayah': 'jakarta timur'},  
 {'nama\_jalan': 'jatinegara', 'keywords': ['jatinegara', 'kodim lama jatinegara'], 'wilayah': 'jakarta timur'},  
 {'nama\_jalan': 'kampung rambutan', 'keywords': ['kampung rambutan', 'kp rambutan'], 'wilayah': 'jakarta timur'},  
 {'nama\_jalan': 'pasar rebo', 'keywords': ['pasar rebo', 'ps rebo'], 'wilayah': 'jakarta timur'},  
 {'nama\_jalan': 'tmii', 'keywords': ['tmii', 'taman mini', 'pintu 1 tmii'], 'wilayah': 'jakarta timur'},  
 {'nama\_jalan': 'pgc', 'keywords': ['pgc', 'pgc cililitan'], 'wilayah': 'jakarta timur'},  
 {'nama\_jalan': 'cililitan', 'keywords': ['cililitan'], 'wilayah': 'jakarta timur'},  
 {'nama\_jalan': 'condet', 'keywords': ['condet'], 'wilayah': 'jakarta timur'},  
 {'nama\_jalan': 'cakung', 'keywords': ['cakung', 'kolong cakung', 'pospol cakung'], 'wilayah': 'jakarta timur'},  
 {'nama\_jalan': 'pulogadung', 'keywords': ['pulogadung', 'pulo gadung'], 'wilayah': 'jakarta timur'},  
 {'nama\_jalan': 'bekasi raya', 'keywords': ['jl raya bekasi', 'bekasi raya'], 'wilayah': 'jakarta timur'},  
 {'nama\_jalan': 'halim', 'keywords': ['halim', 'halim baru', 'halim lama'], 'wilayah': 'jakarta timur'},  
 {'nama\_jalan': 'dewi sartika', 'keywords': ['dewi sartika', 'jl dewi sartika'], 'wilayah': 'jakarta timur'},  
 {'nama\_jalan': 'bogor raya', 'keywords': ['bogor raya', 'jl raya bogor'], 'wilayah': 'jakarta timur'},  
 {'nama\_jalan': 'arion', 'keywords': ['arion', 'lampu merah arion'], 'wilayah': 'jakarta timur'},  
 {'nama\_jalan': 'pramuka', 'keywords': ['jl pramuka raya'], 'wilayah': 'jakarta timur'},  
 {'nama\_jalan': 'bambu apus', 'keywords': ['bambu apus'], 'wilayah': 'jakarta timur'},  
 {'nama\_jalan': 'cibubur', 'keywords': ['cibubur', 'cibubur junction'], 'wilayah': 'jakarta timur'},  
 {'nama\_jalan': 'tamini square', 'keywords': ['tamini square', 'lampu merah garuda', 'tmi square'], 'wilayah': 'jakarta timur'},  
 {'nama\_jalan': 'cijantung', 'keywords': ['cijantung'], 'wilayah': 'jakarta timur'},  
 {'nama\_jalan': 'pinang ranti', 'keywords': ['pinang ranti'], 'wilayah': 'jakarta timur'},  
 {'nama\_jalan': 'otista', 'keywords': ['otista raya'], 'wilayah': 'jakarta timur'},  
 {'nama\_jalan': 'utan kayu', 'keywords': ['utan kayu'], 'wilayah': 'jakarta timur'},  
 {'nama\_jalan': 'caglak', 'keywords': ['caglak', 'tl caglak'], 'wilayah': 'jakarta timur'},  
 {'nama\_jalan': 'rindam', 'keywords': ['rindam'], 'wilayah': 'jakarta timur'},  
 {'nama\_jalan': 'kiwi', 'keywords': ['kiwi', 'tl kiwi'], 'wilayah': 'jakarta timur'},  
  
 # Jakarta Barat  
 {'nama\_jalan': 'daan mogot', 'keywords': ['daan mogot', 'jl daan mogot', 'daanmogot baru'], 'wilayah': 'jakarta barat'},  
 {'nama\_jalan': 'slipi', 'keywords': ['slipi', 'lampu merah slipi', 'tl slipi'], 'wilayah': 'jakarta barat'},  
 {'nama\_jalan': 'tomang', 'keywords': ['tomang', 'underpass tomang'], 'wilayah': 'jakarta barat'},  
 {'nama\_jalan': 'grogol', 'keywords': ['grogol', 'central park', 'rs darmais'], 'wilayah': 'jakarta barat'},  
 {'nama\_jalan': 'cengkareng', 'keywords': ['cengkareng', 'tl cengkareng'], 'wilayah': 'jakarta barat'},  
 {'nama\_jalan': 'kalideres', 'keywords': ['kalideres'], 'wilayah': 'jakarta barat'},  
 {'nama\_jalan': 'pesing', 'keywords': ['pesing'], 'wilayah': 'jakarta barat'},  
 {'nama\_jalan': 'kota', 'keywords': ['kota', 'stasiun kota', 'kota tua'], 'wilayah': 'jakarta barat'},  
 {'nama\_jalan': 'asemka', 'keywords': ['asemka', 'tl asemka'], 'wilayah': 'jakarta barat'},  
 {'nama\_jalan': 'jembatan lima', 'keywords': ['jembatan lima'], 'wilayah': 'jakarta barat'},  
 {'nama\_jalan': 'jalan panjang', 'keywords': ['jl panjang', 'jalan panjang'], 'wilayah': 'jakarta barat'},  
 {'nama\_jalan': 'puri kembangan', 'keywords': ['puri kembangan', 'ringroad puri'], 'wilayah': 'jakarta barat'},  
 {'nama\_jalan': 'kedoya', 'keywords': ['kedoya'], 'wilayah': 'jakarta barat'},  
 {'nama\_jalan': 'latumenten', 'keywords': ['latumenten'], 'wilayah': 'jakarta barat'},  
 {'nama\_jalan': 'meruya', 'keywords': ['meruya', 'gt meruya'], 'wilayah': 'jakarta barat'},  
 {'nama\_jalan': 'roxy', 'keywords': ['roxy'], 'wilayah': 'jakarta barat'},  
  
 # Jakarta Utara  
 {'nama\_jalan': 'priok', 'keywords': ['priok', 'tanjung priok', 'tj priok', 'pelabuhan tj priok'], 'wilayah': 'jakarta utara'},  
 {'nama\_jalan': 'ancol', 'keywords': ['ancol', 'taman impian jaya ancol', 'bintang mas ancol'], 'wilayah': 'jakarta utara'},  
 {'nama\_jalan': 'pluit', 'keywords': ['pluit', 'emporium pluit', 'gt pluit'], 'wilayah': 'jakarta utara'},  
 {'nama\_jalan': 'kelapa gading', 'keywords': ['kelapa gading', 'moi', 'mall of indonesia'], 'wilayah': 'jakarta utara'},  
 {'nama\_jalan': 'sunter', 'keywords': ['sunter'], 'wilayah': 'jakarta utara'},  
 {'nama\_jalan': 'cilincing', 'keywords': ['cilincing', 'jl cakung cilincing'], 'wilayah': 'jakarta utara'},  
 {'nama\_jalan': 'koja', 'keywords': ['koja', 'tl jaya koja'], 'wilayah': 'jakarta utara'},  
 {'nama\_jalan': 'marunda', 'keywords': ['marunda'], 'wilayah': 'jakarta utara'},  
 {'nama\_jalan': 'pademangan', 'keywords': ['pademangan'], 'wilayah': 'jakarta utara'},  
 {'nama\_jalan': 'muara angke', 'keywords': ['muara angke'], 'wilayah': 'jakarta utara'},  
 {'nama\_jalan': 'jembatan tiga', 'keywords': ['jembatan tiga'], 'wilayah': 'jakarta utara'},  
 {'nama\_jalan': 'perintis', 'keywords': ['perintis', 'tl perintis'], 'wilayah': 'jakarta utara'},  
 {'nama\_jalan': 'kebon baru', 'keywords': ['kebon baru', 'tl kebon baru'], 'wilayah': 'jakarta utara'},  
  
 # Ruas Tol  
 {'nama\_jalan': 'tol dalam kota', 'keywords': ['tol dalam kota', 'dalkot', 'ruas tol dalam kota'], 'wilayah': 'lintas wilayah'},  
 {'nama\_jalan': 'tol jakarta-cikampek', 'keywords': ['tol jakarta-cikampek', 'tol japek', 'ruas tol japek'], 'wilayah': 'lintas wilayah'},  
 {'nama\_jalan': 'tol jagorawi', 'keywords': ['tol jagorawi', 'ruas tol jagorawi'], 'wilayah': 'lintas wilayah'},  
 {'nama\_jalan': 'tol jakarta-tangerang', 'keywords': ['tol jakarta-tangerang', 'ruas tol jakarta-tangerang', 'tol janger'], 'wilayah': 'lintas wilayah'},  
 {'nama\_jalan': 'tol jorr', 'keywords': ['tol jorr', 'ruas tol jorr', 'jorr w2s'], 'wilayah': 'lintas wilayah'},  
 {'nama\_jalan': 'tol sedyatmo', 'keywords': ['tol sedyatmo', 'ruas tol sedyatmo', 'tol bandara'], 'wilayah': 'lintas wilayah'},  
 {'nama\_jalan': 'tol cawang-priok', 'keywords': ['tol cawang - tanjung priok', 'tol wiyoto wiyono'], 'wilayah': 'lintas wilayah'},  
 {'nama\_jalan': 'layang mbz', 'keywords': ['layang mbz'], 'wilayah': 'lintas wilayah'},  
 {'nama\_jalan': 'cikunir', 'keywords': ['cikunir', 'off ramp cikunir', 'gt cikunir', 'km 09+750'], 'wilayah': 'bekasi'},  
  
 # Lainnya  
 {'nama\_jalan': 'bandara soetta', 'keywords': ['bandara soetta', 'bandara soekarno hatta'], 'wilayah': 'tangerang'},  
 {'nama\_jalan': 'bekasi', 'keywords': ['bekasi'], 'wilayah': 'bekasi'},  
 {'nama\_jalan': 'depok', 'keywords': ['depok'], 'wilayah': 'depok'},  
 {'nama\_jalan': 'tangerang', 'keywords': ['tangerang'], 'wilayah': 'tangerang'},  
 {'nama\_jalan': 'jatiasih', 'keywords': ['jatiasih'], 'wilayah': 'bekasi'},  
]  
  
  
# Urutkan berdasarkan panjang keyword, dari terpanjang ke terpendek  
ALL\_LOC\_KEYWORDS = []  
for loc in LOKASI\_JAKARTA:  
 for keyword in loc['keywords']:  
 ALL\_LOC\_KEYWORDS.append(keyword)  
ALL\_LOC\_KEYWORDS = sorted(list(set(ALL\_LOC\_KEYWORDS)), key=len, reverse=True)  
  
  
STATUS\_KEYWORDS = {  
 'padat merayap': ['padat merayap'],  
 'ramai cenderung padat': ['ramai cenderung padat'],  
 'cukup padat': ['cukup padat', 'agak padat'],  
 'padat': ['padat'],  
 'tersendat': ['tersendat', 'terhambat', 'sedikit terhambat'],  
 'ramai lancar': ['ramai lancar', 'ramai lancer'],  
 'lancar': ['lancar'],  
 'kondusif': ['kondusif'],  
}  
  
OBSTACLE\_KEYWORDS = {  
 'proyek': ['proyek', 'pengerjaan', 'betonisasi jalan', 'pembetonan', 'perbaikan jalan', 'perbaikan', 'pengaspalan', 'pembangunan lrt', 'galian'],  
 'kecelakaan': ['kecelakaan', 'laka lantas', 'gangguan ban', 'ban pecah'],  
 'kendaraan gangguan': ['gangguan', 'mogok', 'kendala'],  
 'demonstrasi': ['penyampaian pendapat', 'aksi masyarakat', 'aliansi mahasiswa'],  
 'kegiatan': ['kegiatan', 'fun run', 'konser', 'pertandingan sepak bola', 'hbkb'],  
 'balap liar': ['balap liar'],  
 'banjir': ['banjir', 'genangan'],  
 'rekayasa lalin': ['rekayasa lalu lintas', 'buka tutup', 'pengalihan arus', 'penutupan arus'],  
 'antrian': ['antrian kendaraan', 'antrian'],  
 'volume kendaraan': ['volume kendaraan'],  
}  
  
WEATHER\_KEYWORDS = {  
 'hujan': ['hujan'],  
 'gerimis': ['gerimis'],  
 'berawan': ['berawan'],  
 'cerah': ['cerah'],  
}  
  
BIDIRECTIONAL\_KEYWORDS = [  
 'arah sebaliknya', 'kedua arah', 'dan sebaliknya', 'maupun arah sebaliknya',  
 'arah berlawanan', 'maupun arah', 'arah ... maupun arah', 'dari ... menuju ... maupun'  
]  
  
# --- 2. FUNGSI-FUNGSI EKSTRAKSI ---  
  
def find\_keywords(text, keyword\_map):  
 for canonical, variations in keyword\_map.items():  
 for var in variations:  
 if var in text:  
 return canonical  
 return None  
  
def ner\_locations(text, location\_keywords):  
 found\_locations = []  
 temp\_text = " " + text + " "  
 for loc in location\_keywords:  
 if f" {loc} " in temp\_text:  
 found\_locations.append(loc)  
 temp\_text = temp\_text.replace(f" {loc} ", " <LOC\_FOUND> ", 1)  
 # Urutkan berdasarkan posisi kemunculan di teks asli  
 found\_locations.sort(key=lambda x: text.find(x))  
 # Kembalikan daftar unik sambil mempertahankan urutan  
 return list(dict.fromkeys(found\_locations))  
  
def determine\_from\_to(text, locations):  
 if len(locations) < 2:  
 return []  
 pairs = []  
  
 # Pola: di [loc1] arah [loc2] maupun arah [loc3]  
 maupun\_match = re.search(r'di (.\*?) arah (.\*?) maupun arah (.\*)', text)  
 if maupun\_match and len(locations) > 0:  
 source = locations[0]  
 for dest in locations[1:]:  
 if dest in maupun\_match.group(0):  
 pairs.append((source, dest))  
 if pairs: return pairs # Langsung kembalikan jika pola ini cocok  
  
 # Pola: [loc1] ... arah/menuju/ke ... [loc2]  
 for i in range(len(locations) - 1):  
 loc1\_re = re.escape(locations[i])  
 loc2\_re = re.escape(locations[i+1])  
 pattern = re.compile(f"{loc1\_re}.\*?(?:arah|menuju|ke|menuju arah){{1,2}}.\*{loc2\_re}")  
 if pattern.search(text):  
 pairs.append((locations[i], locations[i+1]))  
  
 # Pola: dari [loc1] menuju [loc2]  
 for i in range(len(locations) - 1):  
 loc1\_re = re.escape(locations[i])  
 loc2\_re = re.escape(locations[i+1])  
 pattern = re.compile(f"dari {loc1\_re} (?:menuju|ke) {loc2\_re}")  
 if pattern.search(text):  
 pairs.append((locations[i], locations[i+1]))  
  
 # Jika tidak ada pola jelas, asumsikan urutan kemunculan  
 if not pairs and len(locations) >= 2:  
 pairs.append((locations[0], locations[1]))  
  
 return list(dict.fromkeys(pairs))  
  
# --- 3. PROSES UTAMA ---  
  
def process\_traffic\_data(csv\_filepath):  
 """  
 Memproses file CSV lalu lintas, mengekstrak tanggal dari kolom 'date'  
 dan waktu dari kolom 'text'.  
 """  
 all\_extracted\_data = []  
 try:  
 with open(csv\_filepath, mode='r', encoding='utf-8') as csvfile:  
 reader = csv.DictReader(csvfile)  
 for row in reader:  
 # Ambil data dari kolom 'date' dan 'text'  
 date\_from\_csv = row.get('date', '').strip()  
 text\_from\_csv = row.get('text', '').strip()  
  
 if not date\_from\_csv or not text\_from\_csv:  
 continue  
  
 # Ekstrak waktu dari awal kolom 'text' (misal: "19.42 ...")  
 time\_match = re.search(r'^(\d{2}[.:]\d{2})', text\_from\_csv)  
 if not time\_match:  
 continue # Lewati baris jika tidak ada format waktu di awal  
  
 time = time\_match.group(1).replace('.', ':')  
 date = date\_from\_csv  
  
 # Bersihkan teks dari stempel waktu untuk diproses lebih lanjut  
 cleaned\_text = re.sub(r'^\d{2}[.:]\d{2}\s\*', '', text\_from\_csv).lower()  
  
 # Ekstrak entitas lainnya menggunakan teks yang sudah bersih  
 status = find\_keywords(cleaned\_text, STATUS\_KEYWORDS)  
 obstacle = find\_keywords(cleaned\_text, OBSTACLE\_KEYWORDS)  
 weather = find\_keywords(cleaned\_text, WEATHER\_KEYWORDS)  
 locations\_found = ner\_locations(cleaned\_text, ALL\_LOC\_KEYWORDS)  
 from\_to\_pairs = determine\_from\_to(cleaned\_text, locations\_found)  
  
 if not from\_to\_pairs:  
 continue  
  
 is\_bidirectional = any(keyword in cleaned\_text for keyword in BIDIRECTIONAL\_KEYWORDS)  
  
 for from\_loc, to\_loc in from\_to\_pairs:  
 if not all([time, date, from\_loc, to\_loc, status]):  
 continue  
  
 data\_entry = {  
 "time": time,  
 "date": date,  
 "from": from\_loc.title(),  
 "to": to\_loc.title(),  
 "status": status.title(),  
 "obstacle": obstacle.title() if obstacle else None,  
 "weather": weather.title() if weather else None  
 }  
 all\_extracted\_data.append(data\_entry)  
  
 if is\_bidirectional:  
 reversed\_entry = data\_entry.copy()  
 reversed\_entry["from"] = to\_loc.title()  
 reversed\_entry["to"] = from\_loc.title()  
 all\_extracted\_data.append(reversed\_entry)  
  
 except FileNotFoundError:  
 print(f"Error: File tidak ditemukan di '{csv\_filepath}'")  
 return None  
 return all\_extracted\_data  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 csv\_file = 'lalu\_lintas.csv'  
 output\_json\_file = 'traffic\_data\_regex.json'  
  
 extracted\_data = process\_traffic\_data(csv\_file)  
  
 if extracted\_data:  
 # Menghapus duplikat entri sebelum menyimpan  
 unique\_data = [dict(t) for t in {tuple(d.items()) for d in extracted\_data}]  
 print(f"Berhasil mengekstrak {len(unique\_data)} data lalu lintas unik.")  
  
 try:  
 with open(output\_json\_file, 'w', encoding='utf-8') as f:  
 json.dump(unique\_data, f, indent=2, ensure\_ascii=False)  
  
 print(f"Hasil ekstraksi telah berhasil disimpan ke file: '{output\_json\_file}'")  
  
 except IOError as e:  
 print(f"Gagal menyimpan file: {e}")  
  
 else:  
 print("Tidak ada data yang dapat diekstrak atau file CSV kosong/tidak ditemukan.")

Berhasil mengekstrak 677 data lalu lintas unik.  
Hasil ekstraksi telah berhasil disimpan ke file: 'traffic\_data\_regex.json'