

Sistem Rekomendasi Pembelian Mobil Bekas



Anggota Kelompok

Arkananta Masarief 5025231115

Rafi Faheem Aziz 5025231116

Kemal Tangguh Aji Rajasa 5025231263

I. Knowledge Acquisition (Kelompok)

a. Dataset

Dataset Prediksi Harga Mobil Bekas adalah kumpulan informasi otomotif yang komprehensif yang diambil dari situs web pasar otomotif populer, <https://www.cars.com>. Dataset ini terdiri dari 4.009 titik data, masing-masing mewakili daftar kendaraan yang unik, dan mencakup sembilan fitur berbeda yang memberikan wawasan berharga ke dalam dunia otomotif.

- **Merek & Model:** Mengidentifikasi nama merek atau perusahaan beserta model spesifik dari setiap kendaraan.
- **Tahun Model:** Mengetahui tahun pembuatan kendaraan, yang penting untuk menilai depresiasi dan kemajuan teknologi.
- **Jarak Tempuh:** Memperoleh data jarak tempuh setiap kendaraan, sebuah indikator utama tingkat keausan dan potensi kebutuhan perawatan.
- **Jenis Bahan Bakar:** Mengetahui jenis bahan bakar yang digunakan kendaraan, apakah itu bensin, diesel, listrik, atau hibrida.
- **Tipe Mesin:** Memahami spesifikasi mesin, memberikan gambaran tentang performa dan efisiensi.
- **Transmisi:** Menentukan jenis transmisi, apakah otomatis, manual, atau varian lainnya.
- **Warna Eksterior & Interior:** Menjelajahi aspek estetika kendaraan, termasuk pilihan warna eksterior dan interior.
- **Riwayat Kecelakaan:** Mengetahui apakah kendaraan memiliki riwayat kecelakaan atau kerusakan sebelumnya, yang sangat penting untuk pengambilan keputusan yang tepat.
- **Surat-Surat Bersih:** Mengevaluasi ketersediaan surat-surat yang bersih dan lengkap (clean title), yang dapat mempengaruhi nilai jual kembali dan status hukum kendaraan.
- **Harga:** Mengakses harga yang tercantum untuk setiap kendaraan, membantu dalam perbandingan harga dan penyusunan anggaran.

b. EDA

Analisis Distribusi Merek Mobil : Persebaran jumlah mobil berdasarkan mereknya. Informasi ini penting untuk mengetahui merek mobil apa yang paling umum dalam dataset ini.

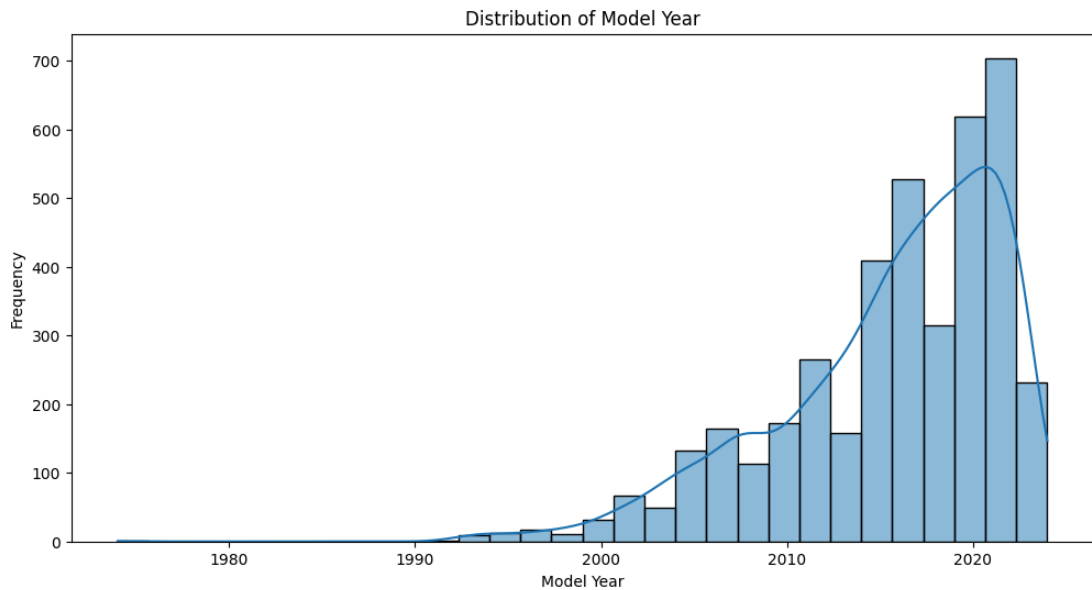
```
brand_counts = df['brand'].value_counts()
display(brand_counts)
```

brand	count
Ford	386
BMW	375
Mercedes-Benz	315
Chevrolet	292
Porsche	201
dst	dst

Analisis Distribusi Manufaktur Mobil : Tahun manufaktur (kolom *model_year*) adalah salah satu faktor terpenting dalam membeli mobil bekas atau *second-hand*.

```
import matplotlib.pyplot as plt
import seaborn as sns

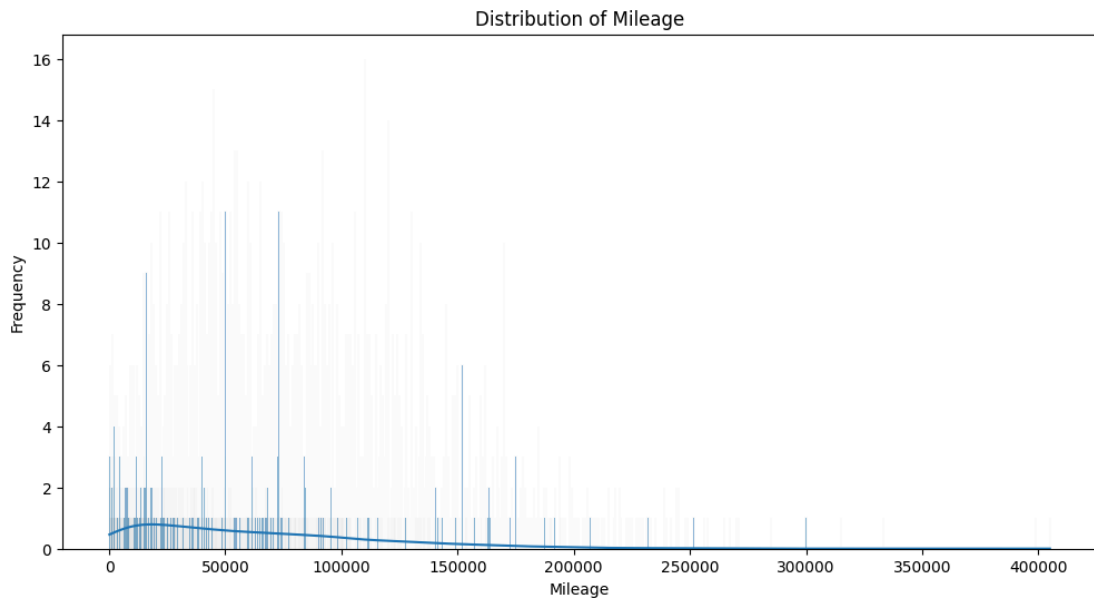
plt.figure(figsize=(12, 6))
sns.histplot(df['model_year'], kde=True, bins=30)
plt.title('Distribution of Model Year')
plt.xlabel('Model Year')
plt.ylabel('Frequency')
plt.show()
```



Analisis Distribusi Jarak Tempuh Mobil: Jarak tempuh mobil bekas menjadi indikator kuat kelayakan mobil tersebut (dalam satuan mil).

```
# Normalisasi
print("Tipe data sebelum cleaning:", df['milage'].dtype)
print("Contoh data sebelum cleaning:", df['milage'].head(3).to_list())
print("-" * 30)
df['milage'] = df['milage'].str.replace(' mi.', "", regex=False)
df['milage'] = df['milage'].str.replace(',', "", regex=False)
df['milage'] = pd.to_numeric(df['milage'], errors='coerce')

# Visualisasi Graph
plt.figure(figsize=(12, 6))
sns.histplot(df['milage'], kde=True, bins=20000)
plt.title('Distribution of Mileage')
plt.xlabel('Mileage')
plt.ylabel('Frequency')
plt.show()
```



Analisis Jenis Bahan Bakar:

```
fuel_type_counts_counts = df['fuel_type'].value_counts()
display(fuel_type_counts_counts)
```

fuel_type	count
gasoline	3309
hybrid	194
E85 Flex Fuel	139
Diesel	116
Plug-in hybrid	34
not supported	2
not defined	215

Normalisasi Fitur Transmission: Dalam dataset ini terdapat lebih dari 2566 jenis transmisi mesin (2566 *unique value*), seperti 6 Speed A/T, 3 Speed A/T dst. Untuk kemudahan filter dan query maka tipe seperti 6 Speed A/T, 3 Speed A/T akan dinormalisasi menjadi Automatic.

```
def normalize_transmission(transmission):
    if 'A/T' in transmission:
```

```

    return 'Automatic'
elif 'M/T' in transmission:
    return 'Manual'
elif 'Automatic' in transmission:
    return 'Automatic'
elif 'Manual' in transmission:
    return 'Manual'
elif 'CVT' in transmission:
    return 'Automatic'
elif 'Dual Shift' in transmission:
    return 'Dual Shift'
elif 'Overdrive' in transmission:
    return 'Dual Shift'
elif '8-SPEED AT' in transmission:
    return 'Automatic'
elif '6 Speed At/Mt' in transmission:
    return 'Dual Shift'
elif 'Auto' in transmission:
    return 'Dual Shift'
elif 'Mt' in transmission:
    return 'Dual Shift'
else:
    return;

```

```

df['transmission'] = df['transmission'].apply(normalize_transmission)
display(df['transmission'].value_counts())

```

transmission	count
Automatic	3212
Dual-shift	408
Manual	372

Normalisasi Fitur Fuel Type

```

def normalize_fuel(fuel):
    if 'not supported' in fuel:
        return 'not defined'
    else:
        return fuel;

cleaned_df['fuel_type'] = cleaned_df['fuel_type'].apply(normalize_fuel)

display(cleaned_df['fuel_type'].value_counts())

```

fuel_type	count
Gasoline	3297
Not defined	214
Hybrid	192
E85 Flex Fuel	139
Diesel	116
Plug-in Hybrid	34

Hasil Normalisasi Fitur dan Penentuan Fitur sebagai Node atau Properti:

Node	Properti (node car)
brand	id
model	mileage
fuel_type	year
transmission	price
accident	exterior_color
clean_title	interior_color
Car (node sentral)	engine_description

Penentuan fitur milage, year, dst sebagai properti didasarkan karena fitur tersebut adalah fitur yang melekat pada satu entitas di node car, analogi sederhananya adalah ketika pembeli mencari suatu mobil, ia akan menjelaskan ‘goals’ nya terlebih dulu seperti “saya membutuhkan mobil matic, kalau bisa yang jarak tempuh nya tidak terlalu tinggi”. Logika saat query adalah mencari di node Car yang memiliki properti mileage dengan batas tertentu, lalu dihubungkan dengan node ‘*transmission*’ dengan value ‘*matic*’.

c. Pengetahuan Baru yang Didapatkan

1. **Dampak Riwayat Kecelakaan:**

Mobil yang memiliki riwayat kecelakaan("At least 1 accident or damage reported") memiliki nilai jual yang lebih rendah. Hal ini bisa dilihat jika mobil yang tidak pernah mengalami kecelakaan

2. **Pengaruh Jarak Tempuh (Mileage) terhadap Harga:**

Semakin tinggi jarak tempuh sebuah mobil, semakin drastis penurunan harganya. Namun, tingkat penurunan harga (depresiasi) ini berbeda antar merek. Merek mobil mewah (seperti BMW atau Mercedes-Benz) mengalami persentase penurunan harga yang lebih tajam per milnya dibandingkan merek mobil non-mewah (seperti Ford atau Hyundai) karena biaya perawatan yang lebih tinggi dan persepsi penurunan nilai premium.

3. **Hierarki Harga Berdasarkan Merek:**

Pengetahuan Baru: Terdapat hierarki harga yang jelas antar merek mobil. Data menunjukkan bahwa merek-merek mewah seperti Bentley, Mercedes-Benz, dan Lexus secara konsisten menempati posisi teratas dalam harga rata-rata mobil bekas. Pengetahuan ini memungkinkan sistem untuk melakukan segmentasi harga awal bahkan sebelum menganalisis atribut lain.

4. **Pentingnya Status "Clean Title":**

Pengetahuan Baru: Atribut clean_title merupakan faktor krusial. Ketidadaan clean title bisa menjadi indikator adanya masalah hukum atau riwayat kendaraan yang serius (misalnya, pernah terendam banjir atau ditarik oleh perusahaan pembiayaan), yang secara drastis akan menurunkan nilainya, terlepas dari kondisi fisiknya.

5.

d. Knowledge Representation dengan Graph (Neo4j)

Pembuatan Unique Values untuk Nodes Brand, Fuel Type, Transmission, Accident Status, dan Clean Title

```
CREATE CONSTRAINT unique_brand IF NOT EXISTS FOR (b:Brand)
  REQUIRE b.name IS UNIQUE;
CREATE CONSTRAINT unique_fuel IF NOT EXISTS FOR (f:FuelType)
  REQUIRE f.name IS UNIQUE;
CREATE CONSTRAINT unique_transmission IF NOT EXISTS FOR
  (t:Transmission) REQUIRE t.type IS UNIQUE;
CREATE CONSTRAINT unique_accident IF NOT EXISTS FOR
  (a:AccidentStatus) REQUIRE a.status IS UNIQUE;
CREATE CONSTRAINT unique_title IF NOT EXISTS FOR (t:TitleStatus)
  REQUIRE t.status IS UNIQUE;
```

Load Data dari CSV dan MERGE untuk semua node, serta sisa kolom akan menjadi Atribut bagi Node Car

```
LOAD CSV WITH HEADERS FROM 'file:///used_car_final_clean.csv' AS row
  FIELDTERMINATOR ',';

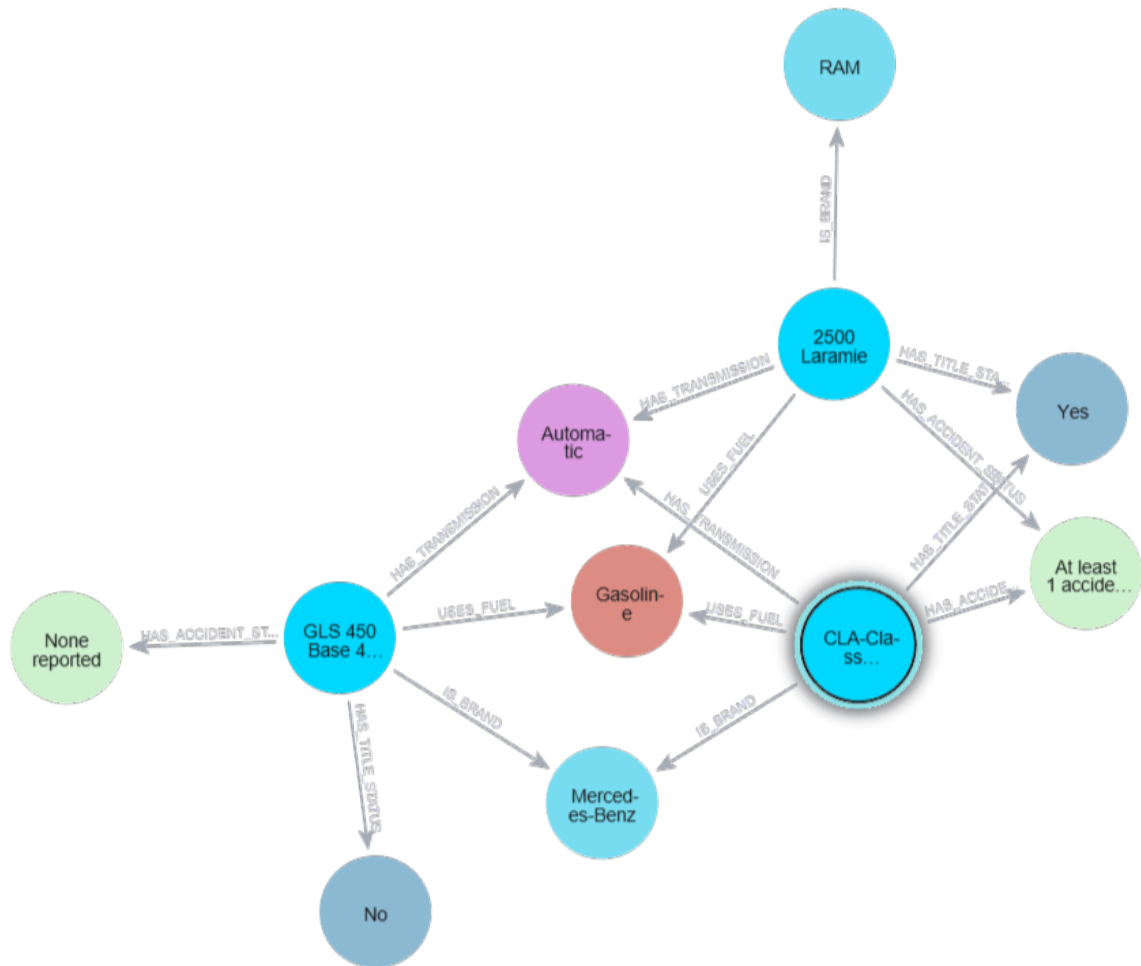
MERGE (brand:Brand {name: row.brand})
MERGE (fuel:FuelType {name: row.fuel_type})
MERGE (transmission:Transmission {type: row.transmission})
MERGE (accident:AccidentStatus {status: row.accident})
MERGE (title:CleanTitleStatus {status: row.clean_title})

CREATE (car:Car {
  id: linenumbers(),
  model: row.model,
  year: toInteger(row.model_year),
  price: toFloat(replace(replace(row.price, '$', ''), ',', '')),
  milage: toFloat(row.milage),
  engine_description: row.engine,
  exterior_color: row.ext_col,
  interior_color: row.int_col
})
```

Buat Relationship

```
CREATE (car)-[:IS_BRAND]->(brand)
CREATE (car)-[:USES_FUEL]->(fuel)
CREATE (car)-[:HAS_TRANSMISSION]->(transmission)
CREATE (car)-[:HAS_ACCIDENT_STATUS]->(accident)
CREATE (car)-[:HAS_TITLE_STATUS]->(title)
```

Contoh Graph



II. Knowledge Reasoning (Individu)

a. Arkananta Masarief

Pencarian Mobil Alternatif

Penciptaan sebuah pencarian untuk mencari alternatif mobil yang berbeda merek sebagai perbandingan saat ingin membeli sebuah mobil. Pencarian ini didasarkan dengan kemiripan tipe bahan bakar, harga, transmisi, dan usia. Perbedaan tahun harus dibawah 2 tahun untuk dianggap mirip, serta harga mobil tidak boleh lebih dari 15%.

```
MATCH (c1:Car)-[:IS_BRAND]->(b1:Brand),
      (c1)-[:USES_FUEL]->(f1:FuelType),
      (c1)-[:HAS_TRANSMISSION]->(t1:Transmission)

MATCH (c2:Car)-[:IS_BRAND]->(b2:Brand),
      (c2)-[:USES_FUEL]->(f2:FuelType),
      (c2)-[:HAS_TRANSMISSION]->(t2:Transmission)

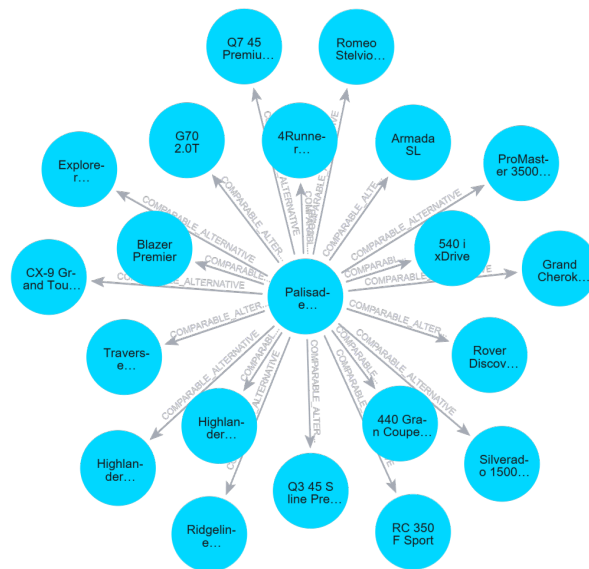
WHERE id(c1) < id(c2)
      AND b1 <> b2
      AND f1 = f2
      AND t1 = t2

WITH c1, c2,
      abs(c1.year - c2.year) AS yearDiff,
      abs(c1.price - c2.price) AS priceDiff

WHERE yearDiff <= 2
      AND priceDiff <= (c1.price * 0.15)
MERGE (c1)-[:COMPARABLE_ALTERNATIVE]-(c2)
SET r.price_difference = priceDiff,
     r.year_difference = yearDiff

RETURN count(r) AS new_alternatives_found
```

Bisa terlihat bahwa Palisade memiliki kemiripan dengan beberapa model mobil merek lain.



Untuk pencarian di penggunaannya bisa dilakukan dengan seperti dibawah ini.

```

MATCH (car:Car)
WHERE car.model CONTAINS 'Supra' AND car.price > 40000
LIMIT 1

MATCH (car)-[r:COMPARABLE_ALTERNATIVE]-(alternative:Car)
MATCH (alternative)-[:IS_BRAND]->(altBrand:Brand)

RETURN alternative.model AS Alternative_Model,
       altBrand.name AS Alternative_Brand,
       alternative.year AS Year,
       alternative.price AS Price,
       r.price_difference AS Price_Difference
ORDER BY
       Price_Difference ASC

```

	Alternative_Model	Alternative_Brand	Year	Price	Price_Difference
1	"F-150 Lariat"	"Ford"	2022	53500.0	0.0
2	"Expedition Max King Ranch"	"Ford"	2020	53722.0	222.0
3	"GLS 450 Base 4 MATIC"	"Mercedes-Benz"	2019	53800.0	300.0
4	"Challenger R/T Scat Pack"	"Dodge"	2022	53900.0	400.0
5	"Romeo Stelvio Quadrifoglio"	"Alfa"	2019	53900.0	400.0
6	"G90 3.3T Premium"	"Genesis"	2021	53950.0	450.0
7	"X7 xDrive40i"	"BMW"	2019	53998.0	498.0

Kode tersebut mencari alternatif dari Supra yang berbeda merek dan ditampilkan berdasarkan harga termiripnya dahulu. Bisa terlihat bahwa Supra memiliki kemiripan dengan F-150 Lariat milik Ford. Jika dataset dilakukan cleaning pada bagian engine, maka fungsi search bisa ditingkatkan dengan memasukkan mesin yang mirip.

b. Rafi Faheem Aziz

Pencarian mobil paling handal dari setiap merek

Pencarian untuk mencari keandalan model dari setiap merek agar bisa mencari mobil terbaik yang bisa dibeli. Parameter yang diperhitungkan dalam pencarian ini adalah status accident yang nantinya akan diproses menjadi skor keandalan. Mobil akan dianggap baik jika skor keandalan lebih dari 80.

```

MATCH (c:Car)-[:IS_BRAND]->(b:Brand)
WITH b, c.model AS modelName, count(c) AS totalCars

MATCH (car:Car {model: modelName})-[:IS_BRAND]->(b)
MATCH (car)-[:HAS_ACCIDENT_STATUS]->(a:AccidentStatus)
WHERE a.status = 'None reported'
WITH b, modelName, totalCars, count(car) AS healthyCars

WITH b, modelName, totalCars, healthyCars,
     round(100.0 * healthyCars / totalCars) AS reliabilityScore

MERGE (mp:ModelProfile {name: modelName})
SET mp.reliability_score = reliabilityScore,
     mp.cars_in_dataset = totalCars
MERGE (b)-[:HAS_MODEL]->(mp)

```

RETURN

b.name AS Brand,
modelName AS Model,
totalCars AS CarsInDataset,
reliabilityScore AS ReliabilityScore

Dapat dilihat tiap merek memiliki beberapa model dengan tiap modelnya sudah memiliki skor keandalan, sebagai contoh merek miotsubishi..



Lalu, dapat dilakukan pencarian model mobil yang termasuk baik yaitu model mobil dengan skor keandalan lebih dari 80.

```
MATCH (b:Brand {name: 'Mitsubishi'})-[:HAS_MODEL]->(mp:ModelProfile)
WHERE mp.reliability_score > 80
RETURN
  mp.name AS Model,
  mp.reliability_score AS Skor_Keandalan
ORDER BY mp.reliability_score DESC
```

	Model	Skor_Keandalan
1	"Lancer Evolusi on GSR"	100.0
2	"Eclipse Spyder GS"	100.0
3	"Eclipse GS"	100.0
4	"Outlander Sport 2.4 SE"	100.0
5	"Lancer Evolusi on IX"	100.0
6	"Mirage ES"	100.0
7	"Outlander SEL"	100.0

Dari hasil tersebut dapat terlihat merek Mitsubishi memiliki beberapa model mobil yang memiliki skor keandalan di atas 80, sehingga model-model mobil tersebut dianggap baik.

c. Kemal Tangguh Aji Rajasa

Scenario based knowledge reasoning:

Pak Luhut Pandjaitan ingin membeli sebuah mobil untuk keluarganya untuk dipakai sehari-hari di dalam kota, keluarganya beranggotakan 4 orang: 2 dewasa dan 2 remaja.

Penyelesaian:

1. Diketahui jika goals adalah membeli sebuah mobil yang cocok untuk dipakai sehari-hari bersama keluarganya yang beranggotakan empat orang,
2. Keputusan diambil secara '*goal-driven*' dimana goals ditetapkan terlebih dahulu, proses reasoning tidak dimulai dari data mobil melainkan dari sebuah goals lalu bekerja mundur dari tujuan atau goals.
3. Ada beberapa ketentuan yang dibutuhkan untuk mencapai tujuan yaitu:
 - a. Bisa mengakomodasi seluruh anggota keluarga Pak Luhut:
 - i. karena tidak ada kolom khusus yang menyatakan bahwa sebuah mobil adalah mobil keluarga maka,
 - ii. penentuan awal apakah cocok sebagai mobil keluarga (sumber berdasarkan internet)
 - iii. *transmission* preferensi adalah *automatic* untuk kenyamanan pemakaian.
 - iv. karena pemakaian di dalam kota maka *fuel type* preferensi adalah *gasoline*.
 - v. keluarga Pak Luhut beranggotakan empat orang, maka brand sports car seperti lamborghini, ferrari, aston martin, dst. tidak sesuai dengan kebutuhan Pak Luhut
 - b. Dapat digunakan sebagai transportasi sehari-hari dan efisien
 - i. tidak ada kolom khusus yang menyatakan sebuah mobil layak digunakan sehari-hari tanpa perawatan extra
 - ii. tahun terdekat akan menjadi opsi terbaik
 - iii. jarak tempuh sebagai indikator seberapa 'terpakai' nya mobil itu
 - iv. memiliki histori *accident* yang bersih untuk menghindari kerusakan yang terbawa dari pemilik sebelumnya
4. Query di Neo4j digunakan untuk mencari data untuk merepresentasikan ketentuan mobil sesuai goals

cypher query ketentuan 3a
// Buat node kategori "Family Car" MERGE (fc:CarCategory {name: 'Family Car'})

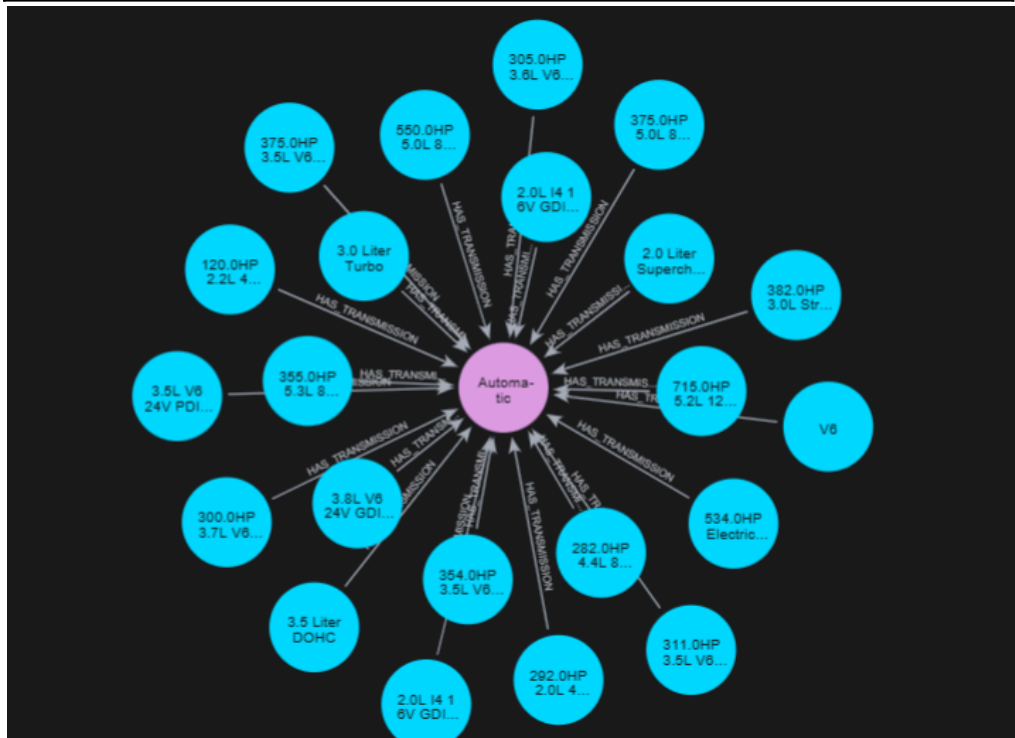

```

// list mobil yang relevan (misalnya SUV, Sedan, Minivan dari merek umum)
MATCH (c:Car)-[:OF_BRAND]->(b:Brand)
WHERE b.name IN [
    'Hyundai', 'Kia', 'Toyota', 'Honda', 'Nissan', 'Ford', 'Chevrolet',
    'Subaru', 'Volkswagen', 'Mazda', 'Jeep', 'GMC', 'Dodge', 'Chrysler',
    'Audi', 'BMW', 'Mercedes-Benz', 'Lexus', 'Acura', 'Volvo'
]

// Buat hubungan dari mobil yang relevan ke kategori "Family Car"
WITH c, fc
MERGE (c)-[:IS_CATEGORY]->(fc);

MATCH (car:Car)-[r:IS_CATEGORY]->(category:CarCategory {name: 'Family Car'})
RETURN car, r, category
LIMIT 25

```



cypher query ketentuan 3b

```

// Temukan mobil yang memenuhi ketentuan:
MATCH
    // - Merupakan "Mobil Keluarga"
    (car:Car)-[:IS_CATEGORY]->(:CarCategory {name: 'Family Car'}),
    // - DAN memiliki surat-surat bersih
    (car)-[:HAS_TITLE_STATUS]->(:CleanTitleStatus {status: 'Yes'}),
    // - DAN tidak memiliki riwayat kecelakaan

```

```
(car)-[:HAS_ACCIDENT_STATUS]->(:AccidentStatus {status: 'None reported'})
```

```
//tampilkan hasil relevan, diurutkan berdasarkan tahun terbaru dan jarak tempuh terendah
```

```
RETURN
```

```
    car.year AS Tahun,  
    car.model AS Model,  
    car.mileage AS Jarak_Tempuh
```

```
ORDER BY Tahun DESC, Jarak_Tempuh ASC
```

```
LIMIT 15;
```

	Tahun	Model	Jarak_Tempuh
1	2024	"M3 CS"	124.0
2	2024	"M3 CS"	124.0
3	2024	"M3 CS"	124.0
4	2024	"M3 CS"	124.0
5	2024	"NX 350h Premium"	300.0
6	2024	"NX 350h Premium"	300.0
7	2024	"NX 350h Premium"	300.0
8	2024	"NX 350h Premium"	300.0

Didapatkan delapan hasil teratas mobil dengan ketentuan pada poin 3a dan 3b, disimpulkan model M3 CS memenuhi semua ketentuan dengan jarak tempuh yang minimum dan tahun manufaktur yang terbilang sangat baru.

Referensi

<https://github.com/KemalRajasa/rsbp/tree/main/ets> link github kelompok 7 beserta source code

<https://www.kaggle.com/search?q=used+car+in%3Adatasets>