**Syntax pada Analisis Data Kereta Api dan Stasiun pada Daerah Operasi VIII Surabaya Menggunakan SPARQL dengan Algoritma Betweenness centrality**

1. **Syntax visualisasi pada Neo4J**

*LOAD CSV WITH HEADERS FROM "file:///CLEAR\_stasiun-kota-kereta" AS csvLine*

*MERGE (nama\_stasiun:Stasiun {name: csvLine.nama\_stasiun})*

*MERGE (kota\_kabupaten:KotaKabupaten {name: csvLine.kota\_kabupaten})*

*MERGE (nama\_kereta:Kereta {name: csvLine.nama\_kereta})*

*CREATE (nama\_stasiun)-[:MENGOPERASIKAN]->(nama\_kereta)*

*CREATE (nama\_stasiun)-[:BERLOKASI\_DI]->(kota\_kabupaten)*

1. **Syntax Pembuatan Model Node2Vec**

*node2vec = Node2Vec(g, dimensions=85,*

*walk\_length=16, num\_walks=100, workers=2)*

*model = node2vec.fit(window=10, min\_count=1)*

1. **Syntax visualisasi t-SNE**

*#TSNE Kereta dan Stasiun*

*tsne = TSNE(n\_components=2, random\_state=7, perplexity=15)*

*embeddings\_2d = tsne.fit\_transform(embeddings)*

*#Mulai Visualisasi*

*warna\_stasiun* ***=*** *{*

*'Stasiun Blitar': 'lightblue',*

*'Stasiun Surabaya Gubeng': 'b',*

*'Stasiun Surabaya Kota': 'r',*

*'Stasiun Bangil': 'teal',*

*'Stasiun Gembong': 'cyan',*

*'Stasiun Porong': 'g',*

*'Stasiun Benowo': 'y',*

*'Stasiun Kandangan': 'magenta',*

*'Stasiun Tandes': 'darkblue',*

*'Stasiun Cerme': 'purple',*

*'Stasiun Duduk': 'lime',*

*'Stasiun Lamongan': 'silver',*

*'Stasiun Sumlaran': 'grey',*

*'Stasiun Pucuk': 'black',*

*'Stasiun Babat': 'orange',*

*'Stasiun Bowerno': 'brown',*

*'Stasiun Sumberrejo': 'maroon',*

*'Stasiun Kapas': 'olive',*

*'Stasiun Indro': 'lightpink',*

*'Stasiun Gresik': 'violet',*

*'Stasiun Sepanjang': 'lightpink',*

*'Stasiun Boharan': 'darkviolet',*

*'Stasiun Kedinding': 'darkorange',*

*'Stasiun Gedangan (Jawa Timur)': 'pink',*

*'Stasiun Gununggangsir': 'indigo'*

*}*

*df['color']* ***=*** *df['nama\_stasiun'].apply(****lambda*** *x: warna\_stasiun[x])*

*warna\_kereta****=*** *dict(zip(df['nama\_kereta'], df['color']))*

*colors* ***=*** *[warna\_kereta[x]* ***for*** *x* ***in*** *df.nama\_kereta]­­­*

*#Visualisasi Embeding*

*import matplotlib.patches as mpatches*

*figure = plt.figure(figsize=(11, 9))*

*ax = figure.add\_subplot(111)*

*ax.scatter(embeddings\_2d[:, 0], embeddings\_2d[:, 1], c=colors)*

*i=0*

*for keretaku in nodekereta:*

*ax.annotate(keretaku, (embeddings\_2d[i, 0], embeddings\_2d[i, 1]))*

*i=i+1*

*stasiun\_legend = [mpatches.Patch(color=color, label=df.nama\_stasiun.unique()) for df.nama\_stasiun, color in warna\_stasiun.items()]*

*ax.legend(handles=stasiun\_legend);*

1. **Syntax visualisasi Algoritma Betweeness Centrality**

*#Algoritma Betweenness*

*pos = nx.spring\_layout(g)*

*node\_color = [20000.0 \* betweenness\_dict[v] for v in g]*

*node\_size = [v \* 10000 for v in betweenness\_dict.values()]*

*plt.figure(figsize=(20,20))*

*nx.draw\_networkx(g, pos=pos, with\_labels=True,*

*node\_color=node\_color,*

*node\_size=node\_size )*

*plt.axis('off')*

1. **Syntax menghitung density graph pada Jupyter**

*#Density*

*density = nx.density(g)*

*print("Network density:", density)*