

Nama : Abdurrahman Al-atsary
NIM : 121450128
Kelas : Praktikum TBD RA (RC)

Tugas Individu Modul 3 (Neo4j Graph Databases)

1. Pembuatan **Node Mahasiswa** dengan nama beserta **NIM-nya**

```
1 CREATE (mahasiswa:Mahasiswa {Name:"Abdurrahman", NIM : "128"}) return mahasiswa;
2 CREATE (mahasiswa:Mahasiswa {Name:"Sasa Rahma Lia", NIM : "119"}) return mahasiswa;
3 CREATE (mahasiswa:Mahasiswa {Name:"Evan", NIM : "102"}) return mahasiswa;
4 CREATE (mahasiswa:Mahasiswa {Name:"Fathir", NIM : "098"}) return mahasiswa;
5 CREATE (mahasiswa:Mahasiswa {Name:"Raditia", NIM : "105"}) return mahasiswa;
6 CREATE (mahasiswa:Mahasiswa {Name:"Nathanael", NIM : "107"}) return mahasiswa;
7 CREATE (mahasiswa:Mahasiswa {Name:"Rangga", NIM : "106"}) return mahasiswa;
8 CREATE (mahasiswa:Mahasiswa {Name:"Saiful", NIM : "115"}) return mahasiswa;
9 CREATE (mahasiswa:Mahasiswa {Name:"Arsal", NIM : "111"}) return mahasiswa;
10 CREATE (mahasiswa:Mahasiswa {Name:"Rafi", NIM : "143"}) return mahasiswa;
```

neo4j\$ CREATE (mahasiswa:Mahasiswa {Name:"Abdurrahman", NIM : "128"}) return mahasiswa
neo4j\$ CREATE (mahasiswa:Mahasiswa {Name:"Sasa Rahma Lia", NIM : "119"}) return mahasiswa
neo4j\$ CREATE (mahasiswa:Mahasiswa {Name:"Evan", NIM : "102"}) return mahasiswa
neo4j\$ CREATE (mahasiswa:Mahasiswa {Name:"Fathir", NIM : "098"}) return mahasiswa
neo4j\$ CREATE (mahasiswa:Mahasiswa {Name:"Raditia", NIM : "105"}) return mahasiswa
neo4j\$ CREATE (mahasiswa:Mahasiswa {Name:"Nathanael", NIM : "107"}) return mahasiswa
neo4j\$ CREATE (mahasiswa:Mahasiswa {Name:"Rangga", NIM : "106"}) return mahasiswa
neo4j\$ CREATE (mahasiswa:Mahasiswa {Name:"Saiful", NIM : "115"}) return mahasiswa
neo4j\$ CREATE (mahasiswa:Mahasiswa {Name:"Arsal", NIM : "111"}) return mahasiswa
neo4j\$ CREATE (mahasiswa:Mahasiswa {Name:"Rafi", NIM : "143"}) return mahasiswa

neo4j\$ MATCH (m:Mahasiswa) RETURN m;

Overview
Node labels
* (20) Mahasiswa (20)
Displaying 20 nodes, 0 relationships.

Disini saya membuat 10 nama terlebih dahulu dengan membuat atribut mahasiswa Node label mahasiswa yang terdiri dari kunci property (**Property Key**) : Name dan NIM sebagai tanda pengenalan di dalam node dan sebagai atribut di dalam node.

2. Membuat **Node Mata Kuliah** dengan property key yang terdiri dari **TPB dan Prodi dan Nama Mata Kuliahnya.**

```

1 CREATE (LMD:Matakuliah{name:"Logika Matematika Diskrit", prodi: "Sains Data", TPB : 4}),
2       (ALPRO:Matakuliah{name:"Algoritma Pemrograman", prodi: "Teknik Informatika", TPB :
3       14}),
4       (SSD:Matakuliah{name:"Statistika Sains Data", prodi: "Sains Data", TPB : 5}),
5       (PEMWEB:Matakuliah{name:"Pemrograman Website", prodi: "Teknik Informatika", TPB:15}),
6       (STRUKDAT:Matakuliah{name:"Struktur Data", prodi: "Sains Data", TPB : 2}),
7       (MatDas:Matakuliah{name:"Matematika Dasar", prodi: "Teknik Lingkungan", TPB : 25}),
8       (ADS:Matakuliah{name:"Analisis Data Statistik", prodi: "Sains Data", TPB:3}),
9       (ANMUL:Matakuliah{name:"Analisis Multivariat", prodi: "Sains Data", TPB: 1}),
10      (INTRANS:Matakuliah{name:"Infrastruktur dan Transportasi", prodi: "PWK", TPB:50}),
11      (PM:Matakuliah{name:"Pembelajaran Mesin", prodi: "Sains Data", TPB: 56});

```

Added 10 labels, created 10 nodes, set 30 properties, completed after 23 ms.



Disini dapat dilihat kita membuat atribut yang terdiri dari 10 Mata Kuliah dengan **Key Property** yang terdiri dari TPB dan Prodi.

3. Membuat **Relasi** Mahasiswa yang mengambil **Mata Kuliah**

```

1 MATCH (mahasiswa:Mahasiswa {Name:"Abdurrahman", NIM : "128"})
2 MATCH (matkul:Matakuliah {TPB: 5, prodi: 'Kimia'})
3 CREATE (mahasiswa)-[:MENGAMBIL]->(matkul)
4 RETURN mahasiswa, matkul;
5
6 MATCH (mahasiswa:Mahasiswa {Name:"Sasa Rahma Lia", NIM : "119"})
7 MATCH (matkul:Matakuliah {TPB: 4, prodi: 'Sains Data'})
8 CREATE (mahasiswa)-[:MENGAMBIL]->(matkul)
9 RETURN mahasiswa, matkul;
10
11 MATCH (mahasiswa:Mahasiswa {Name:"Evan", NIM : "102"})
12 MATCH (matkul:Matakuliah {TPB: 3, prodi: 'Fisika'})

```

neo4j\$ MATCH (mahasiswa:Mahasiswa {Name:"Abdurrahman", NIM : "128"}) MATCH (matkul:Matakuliah {TPB: 5, prodi: 'Kimia'}) ...

neo4j\$ MATCH (mahasiswa:Mahasiswa {Name:"Sasa Rahma Lia", NIM : "119"}) MATCH (matkul:Matakuliah {TPB: 4, prodi: 'Sains Data'}) ...

neo4j\$ MATCH (mahasiswa:Mahasiswa {Name:"Evan", NIM : "102"}) MATCH (matkul:Matakuliah {TPB: 3, prodi: 'Fisika'}) ...

neo4j\$ MATCH (mahasiswa:Mahasiswa {Name:"Fathir", NIM : "098"}) MATCH (matkul:Matakuliah {TPB: 22, prodi: 'Farmasi'}) ...

neo4j\$ MATCH (mahasiswa:Mahasiswa {Name:"Raditia", NIM : "105"}) MATCH (matkul:Matakuliah {TPB: 19, prodi: 'Sains Lingku...'}) ...

neo4j\$ MATCH (mahasiswa:Mahasiswa {Name:"Nathanael", NIM : "107"}) MATCH (matkul:Matakuliah {TPB: 36, prodi: 'Teknik Per...'}) ...

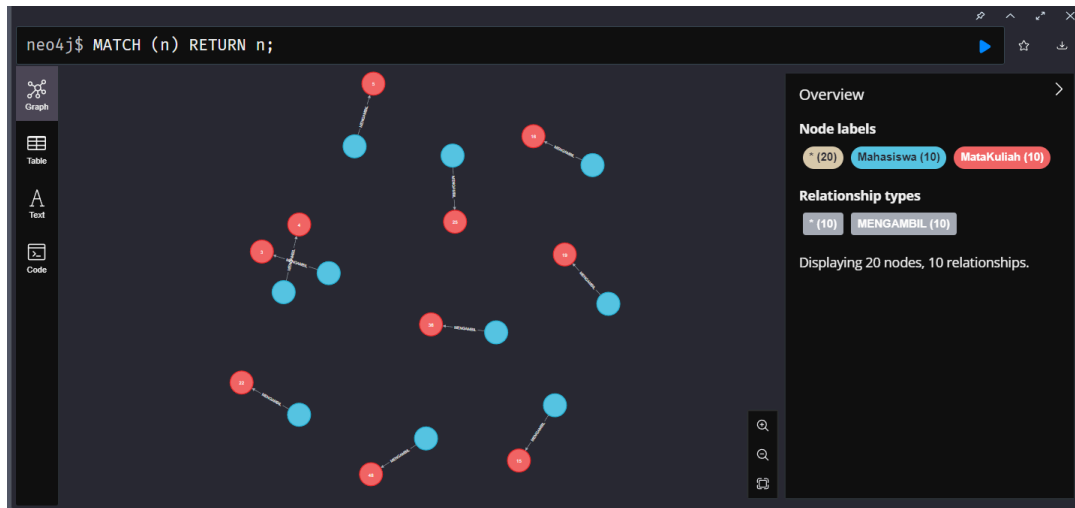
neo4j\$ MATCH (mahasiswa:Mahasiswa {Name:"Rangga", NIM : "106"}) MATCH (matkul:Matakuliah {TPB: 48, prodi: 'Teknik Sipil'}) ...

neo4j\$ MATCH (mahasiswa:Mahasiswa {Name:"Saiful", NIM : "115"}) MATCH (matkul:Matakuliah {TPB: 16, prodi: 'Teknik Inform...'}) ...

neo4j\$ MATCH (mahasiswa:Mahasiswa {Name:"Arsal", NIM : "111"}) MATCH (matkul:Matakuliah {TPB: 25, prodi: 'Teknik Elektro...'}) ...

Membuat relasi antara masing-masing node yang sudah dibuat sebelumnya sesuai urutan dan dapat dilihat bahwa telah membentuk sebuah traceback data sudah terhubung 1 dengan lainnya.

4. Melihat **Relasi antar Kelompok Mahasiswa dengan Matakuliah** dari Graph Method

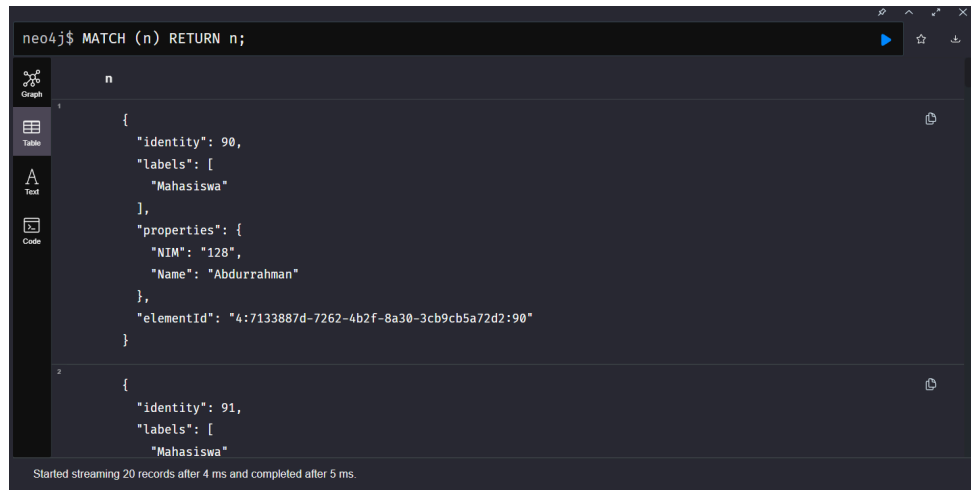


The image shows the Neo4j Table interface. The main area displays a table with the results of the query. The table has two columns: `n` and `id`. The data is as follows:

n	id
(:Mahasiswa {NIM: "128",Name: "Abdurrahman"})	1
(:Mahasiswa {NIM: "119",Name: "Sasa Rahma Lia"})	2
(:Mahasiswa {NIM: "102",Name: "Evan"})	3
(:Mahasiswa {NIM: "098",Name: "Fathir"})	4
(:Mahasiswa {NIM: "105",Name: "Raditia"})	5
(:Mahasiswa {NIM: "107",Name: "Nathanael"})	6
(:Mahasiswa {NIM: "106",Name: "Rangga"})	7
(:Mahasiswa {NIM: "115",Name: "Saiful"})	8
(:Mahasiswa {NIM: "111",Name: "Arsal"})	9
(:Mahasiswa {NIM: "143",Name: "Rafi"})	10

The interface includes a sidebar with icons for Graph, Table, Text, and Code. The top bar shows the query: `neo4j$ MATCH (n) RETURN n;`. The bottom bar shows the MAX COLUMN WIDTH slider.

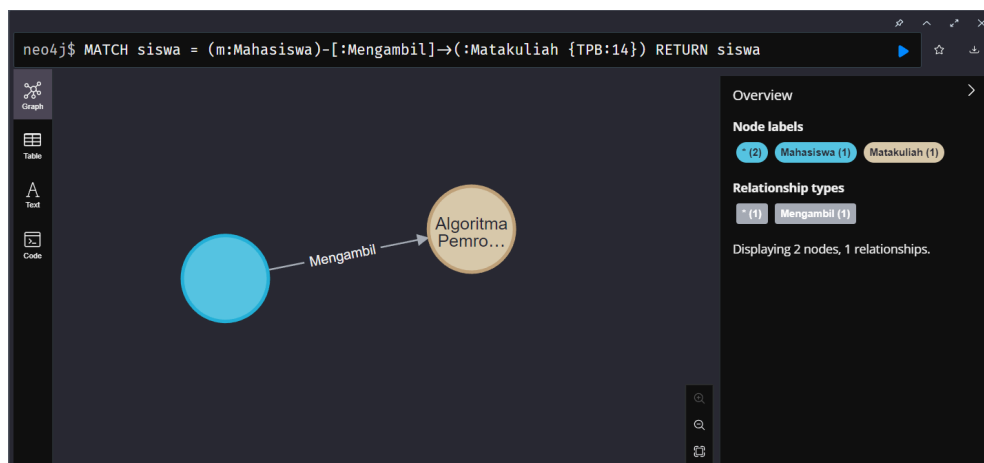
```
neo4j$ MATCH (n) RETURN n;
```




The screenshot shows the Neo4j Cypher query interface. The query `MATCH (n) RETURN n;` has been executed. The results are displayed in a table view, showing a single record with a JSON object. The JSON object contains the following data:

```
{
  "identity": 90,
  "labels": [
    "Mahasiswa"
  ],
  "properties": {
    "NIM": "128",
    "Name": "Abdurrahman"
  },
  "elementId": "4:7133887d-7262-4b2f-8a30-3cb9cb5a72d2:90"
}
```

At the bottom, it states: "Started streaming 20 records after 4 ms and completed after 5 ms."



```
1 MATCH (m:Mahasiswa)-[:Mengambil]->(mk:Matakuliah {name: "Statistika Sains Data"})
2 RETURN m.Name
```



The screenshot shows the Neo4j Cypher query interface. The query `MATCH (m:Mahasiswa)-[:Mengambil]->(mk:Matakuliah {name: "Statistika Sains Data"}) RETURN m.Name` has been executed. The results are displayed in a table view, showing a single record with the name "Evan".

m.Name

m.Name
Evan

Started streaming 1 records after 15 ms and completed after 16 ms.

Diatas merupakan representasi data node dan relasi yang terbentuk dengan property masing masing data dari bentuk yang tabel hingga berbentuk json dapat disimpulkan bahwa telah terbentuk data data dari persoalan yang ada di dalam Tugas Individu dengan requirement yang disebutkan meskipun terlihat singkat dan padat didalam jawabannya dari segi representasi data tabel dan graph disajikan didalam jawaban.