

MODUL PRAKTIKUM

SD3203-Teknologi Basis Data

**Program Studi Sains Data
Fakultas Sains
Institut Teknologi Sumatera**

2025

MODUL 8

Graph Database

Graph Database

1. Tujuan Praktikum

- a. Memahami konsep dasar Graph Database dan struktur datanya (Node, Relationship, Property, Path).
- b. Mampu membuat dan memodifikasi data dalam bentuk graf menggunakan Cypher.

2. Konsep Dasar

Basis data graph (graph database) adalah jenis basis data yang menggunakan struktur graph untuk menyimpan data. Salah satu aplikasi yang dapat diterapkan dalam graph database adalah Neo4j. Neo4J adalah sistem manajemen basis data graph yang dikembangkan oleh Neo4J, Inc., sebuah perusahaan yang berbasis di San Francisco Bay Area, Amerika Serikat. Hasil perhitungan peringkat DB-Engines menunjukkan Neo4J menempati peringkat pertama sistem basis data graph paling populer mengalahkan Microsoft Azure Cosmos DB, Arango DB, Orient DB, dan beberapa sistem manajemen basis data graph lain. Neo4J menggunakan bahasa query bernama Cypher. Cypher adalah sebuah bahasa deklaratif yang terinspirasi dari SQL untuk mendeskripsikan pola graph. Cypher pada awalnya dikembangkan eksklusif untuk Neo4J tapi kemudian dengan dirilisnya open Cypher kini Cypher bisa diterapkan pada SAP Hana dan Agens Graph. Cypher Query mendefinisikan node relationship dengan syntax CREATE dan MATCH. Neo4J seperti database graph lainnya bisa digunakan, dalam mempermudah, penyimpanan dan pengambilan data dalam bentuk graph. Misalnya adalah media sosial, system rekomendasi, fraud detection dsb-nya.

Pada Neo4J data direpresentasikan dalam bentuk Node dan Relasi. Node merupakan titik. Relasi adalah penghubung antar dua node. Baik Node dan Relasi memiliki property yang bisa kita sesuaikan. Sebelum menciptakan node atau Relasi, bisa kita definisikan atribut yang ada di tiap node atau Relasi tersebut. Hal ini opsional tidak wajib tapi merupakan best practice untuk dilakukan. Perancangan database di Neo4J hanya sebatas desain node dan relasi. Tidak ada istilah tabel atau database dalam Neo4J. Hal ini berbeda dengan basis data RDBMS

atau NoSQL lainnya. Untuk Neo4J, dalam satu database server Neo4J hanya ada satu database, kita tidak perlu mendefinisikan nama database, juga tidak ada table didalam database tersebut. Sehingga ketika kita ingin mengubah database maka seluruh file di direktori data diNeo4J harus dihapus secara manual. Cara ini yang paling cepat dibanding menghapus node dan relasi. Neo4j telah dikembangkan sejak tahun 2003, dapat ditulis dengan Bahasa Java, JRuby, Scala, Python, Clojure dan sebagainya.

Karakteristik tipe data dari neo4j adalah:

1. Bersifat optional, tidak memiliki struktur data, mudah saat melakukan perubahan skema.
2. Mudah untuk dimodelkan pada domain data sets yang umum seperti pada permodelan CMS (CompactMotion Switch) sebagai access controllist, database objek, Triple stores dll.
3. Penggunaannya pada semantic web dan EDF (External Data Format), linked Data, GIS (Geographic Information System), situs jejaring sosial dan lain sebagainya.

Bagian penting pembentuk graph database dalam Neo4j terdiri dari 6 bagian yaitu, Node, Relationship, Properties, Path, Traversal dan Query



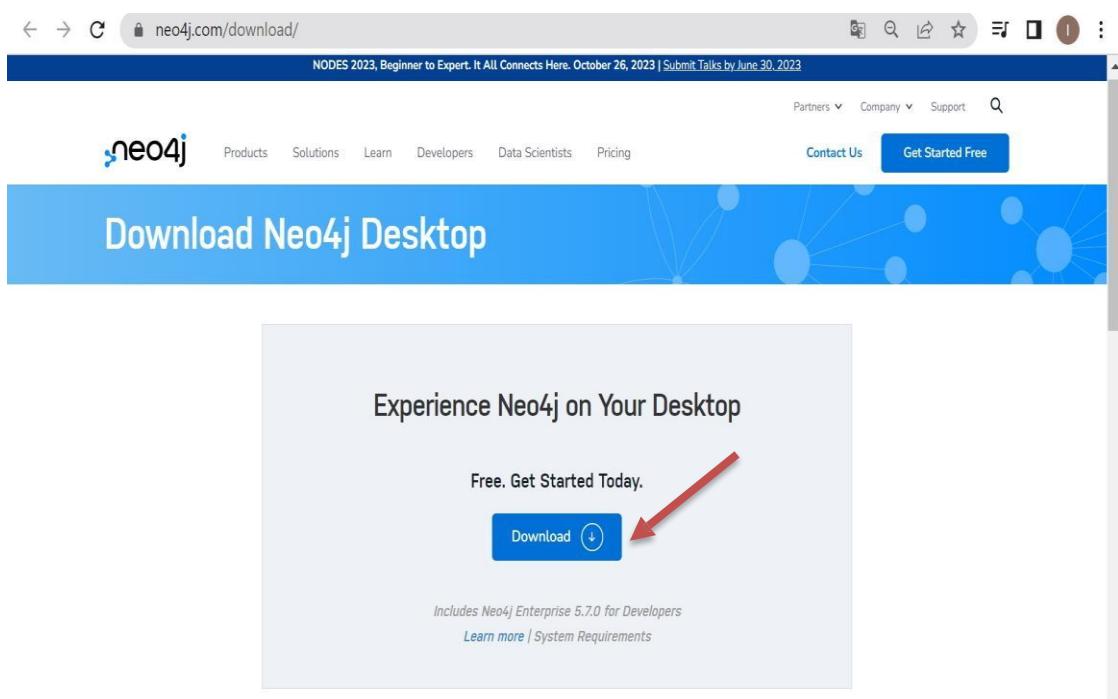
- Node adalah bagian yang paling mendasar yang membentuk sebuah graph. Dalam neo4j node dapat memiliki relationship dan property.
- Relationship merupakan salah satu kunci pembentuk graph database. Dengan adanya relationship memungkinkan untuk mencari data atau informasi yang berkaitan. Seperti halnya node, relationship juga dapat memiliki properti.
- Properti merupakan key pada setiap node yang umumnya adalah sebuah tipe data string. Dan nilai atau isi properti pada setiap key yang bertipe string dapat mencakup semua tipe data dan memungkinkan untuk fungsi array.

- Path merupakan node yang memiliki hubungan (relationship) disebut path. Misalnya hubungan antara node 1 dengan node 2, biasanya path berasal dari pengambilan secara query maupun hasil dari kunjungan traversal.
- Traversal memiliki arti mengunjungi setiap node, mengikuti setiap hubungan satu node ke node yang lain yang tentu saja berdasarkan aturan yang ada. Dalam banyak kasus, tiap node yang dikunjungi merupakan subgraph.
- Query merupakan kemampuan untuk menampilkan suatu data dari database. Bahasa query yang dipakai oleh Neo4j sekarang mendukung kemampuan untuk mengambil seluruh jalur terpendek, mengekstrak dengan mudah nilai pertama dari list dengan null menggunakan COALESCE, alias column dan menangani hubungan panjang variable.

3. Latihan Praktikum

I. Cara Menginstall Aplikasi Neo4j

1. Download aplikasinya terlebih dahulu di <https://neo4j.com/download/>



Kemudian klik download dan isilah data pribadi seperti dibawah ini

neo4j

Products Solutions Learn Developers Data Scientists Pricing

Contact Us Get Started Free

Get Neo4j Desktop

Get Started Now

Please fill out this form to begin your download

ira

Last Name

Email (Business Preferred)

Company Name

Phone Number

Indonesia

By downloading you agree to the [Neo4j License Agreement for Neo4j Desktop Software.](#)

Download Desktop

The information you provide will be used in accordance with the terms of our [privacy policy.](#)



Setelah diisi data tersebut maka klik download desktop. Selanjutnya akan ditampilkan activation key dibawah ini :

NODES 2023, Beginner to Expert. It All Connects Here. October 26, 2023 | [Submit Talks by June 30, 2023](#)

neo4j

Products Solutions Learn Developers Data Scientists Pricing

Contact Us Get Started Free

Thanks for downloading Neo4j Desktop

Your download should begin automatically in a few seconds. If it doesn't, Click one of the links : [Windows](#) • [OSX](#) • [Linux](#)

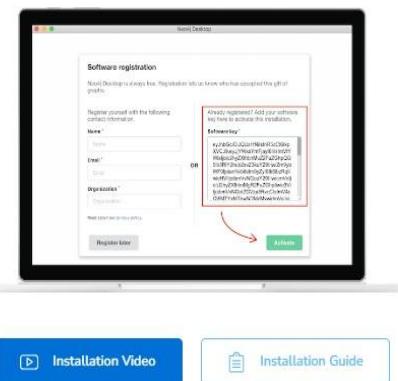
Recommended system requirements: MacOS 10.10 (Yosemite)+, Windows 8.1+ with Powershell 5.0+, Ubuntu 12.04+, Fedora 21, Debian 8.

Neo4j Desktop Activation Key

Use this key to activate your copy of Neo4j Desktop for use.

Copy to clipboard

```
eyJhbGciOiJQUzI1NiIshR5cC16IkpxVCj9eyIjbWFpbCl6nHzhZm0cmplpcmExMzbAZ21haWwuY29tUiwb
Wl4cGFuZWxjZCI6iRkZXpY2U6MTg3YmjOWI4OTMzZDEtMGYyN2E4NmRlZWmzNTMtMjYwMzE4NT
EtOWM0YzgtTg3YmjOWI4OTQ5YTkICltaxhwYW5lbFByb2pY3RjZC16lRizmlyNDE0YW15NzNjNzQ
xYjZmMDY3YmYvNmQ1NTc1Iiwb3NjIjolsIshB1Yi6im5lbzRqlmNbSlslnJZy6ImlyYSBzYWZpdHUpIw
ic3VljoibmVNGoLzGVza3RvcCslmV4cC16lMTcxNDg3NzE4NwidmVyojokIshlzcyl6m5lbzRqlmNbSls
Im5iZl6MTY4Mzl1NDc4NiwhWF0ljoxNjgzMjU0Nzg2LCjqdGkiO1ZF15VHNFnFngfQ.HZlaEr6WO_y17sg
fxSOBi04pEKUajkb2rTwP67APu0xbuM2ExrbGctovsZIKLyt2cbpBJbcQYrrYFh3AmZL08463BM7rc
TTr7EUVmPdsO_OvUmDBDmzzbfk1XCtcdMMAD30mpdbKMsL1RiifhfqbTECsQtu7yrRnh53ylkp
IAFFMBTezyflok-CrnJLwyXa55sR08rZPBYf6kxLI-7sxwR6oCOpFRkVMqfSgJHTqs6IruYorRQxou0o
```

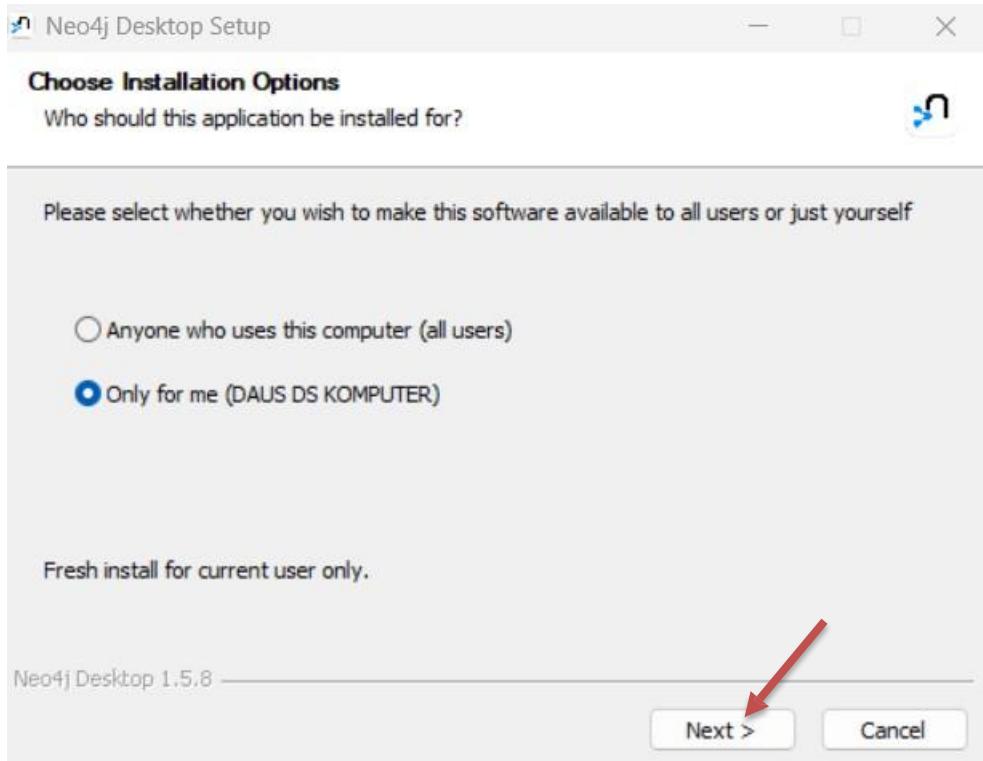


Installation Video

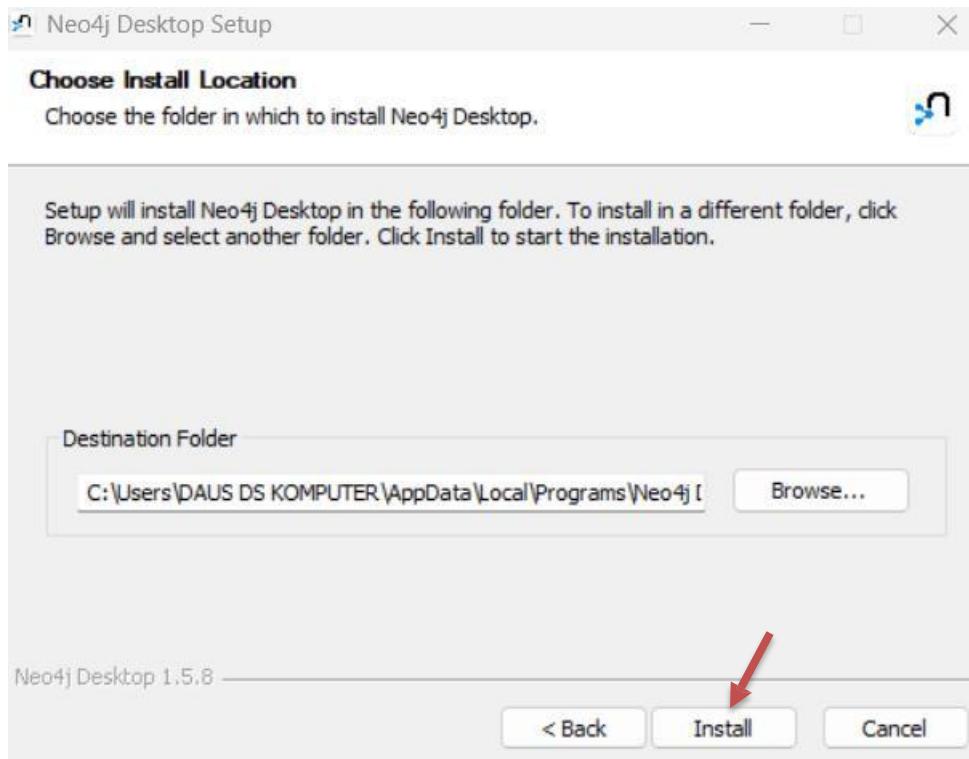
Installation Guide

Copy activation key dan simpan dahulu di notepad++

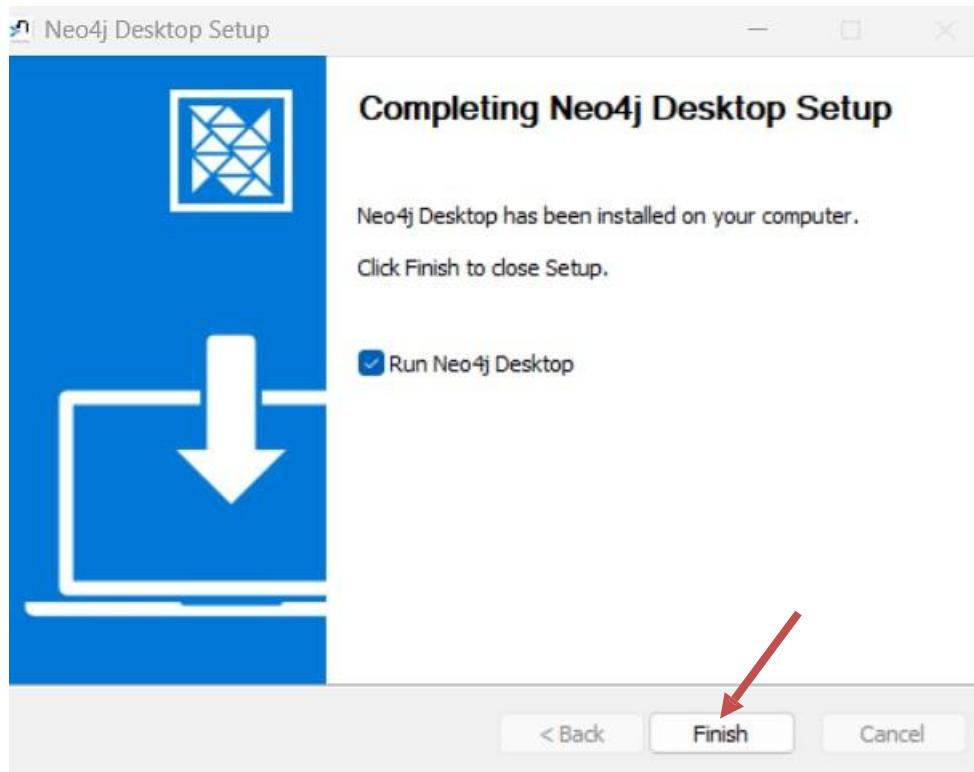
- Setelah download install aplikasi Neo4j Desktop Setup 1.5.8.exe. Pilih only for me.lalu klik next



3. Pilih lokasinya. Lalu klik install



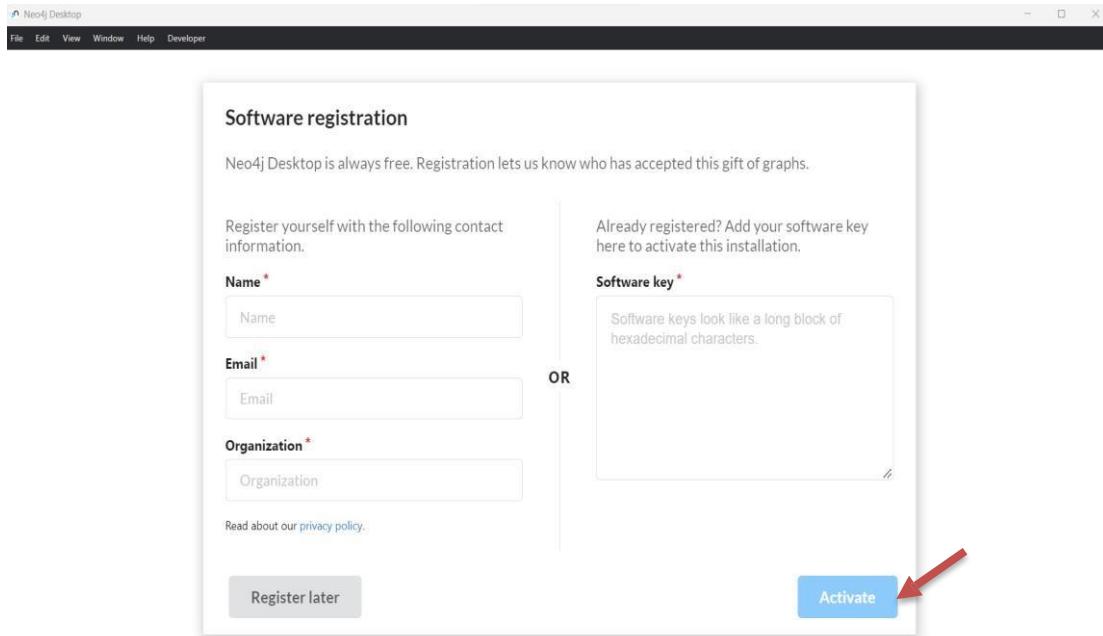
4. Klik finish aplikasi bisa digunakan



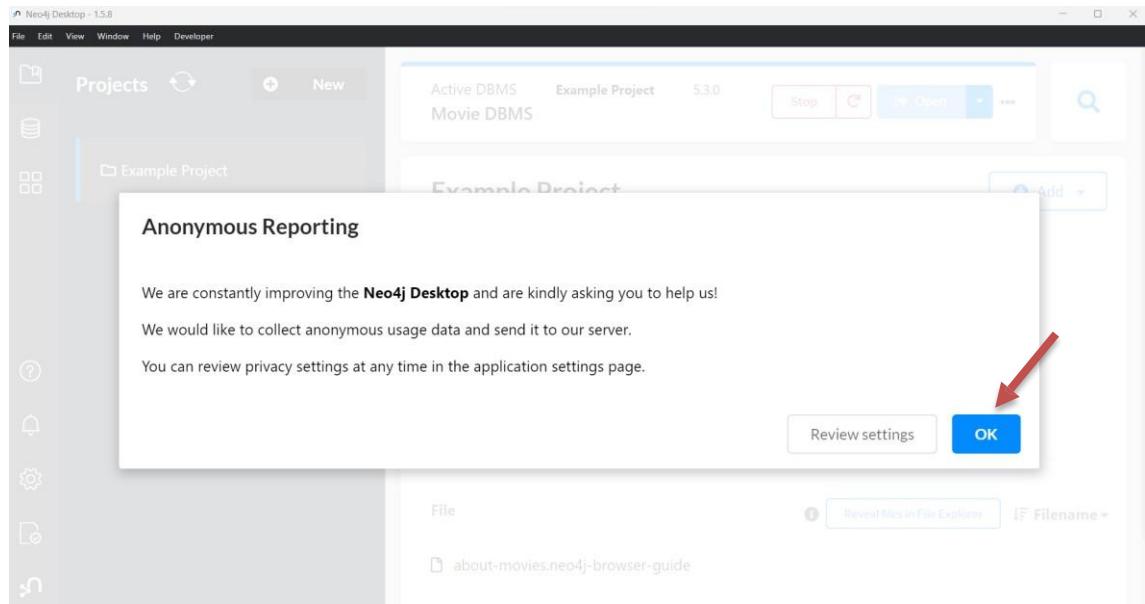
5. Selanjutnya run neo4j secara langsung dengan pilih centang 2 kali dan klik allow access. Kemudian pilih I Agree dan pastikan lokasi aplikasi tersebut sama seperti lokasinya saat di download komputer masing-masing.



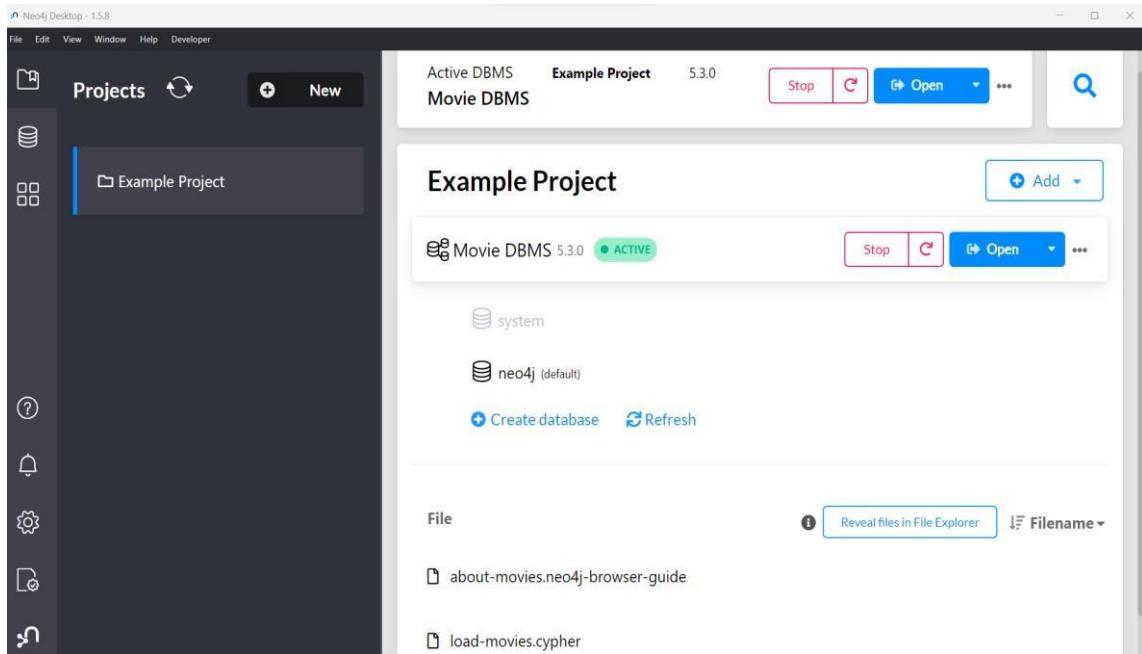
6. Langkah selanjutnya adalah isi data pribadi seperti langkah no.1 dan masukkan activation key yang telah disimpan di notepad++ dibagian software key lalu klik activate



Kemudian klik OK pada anonymous reporting

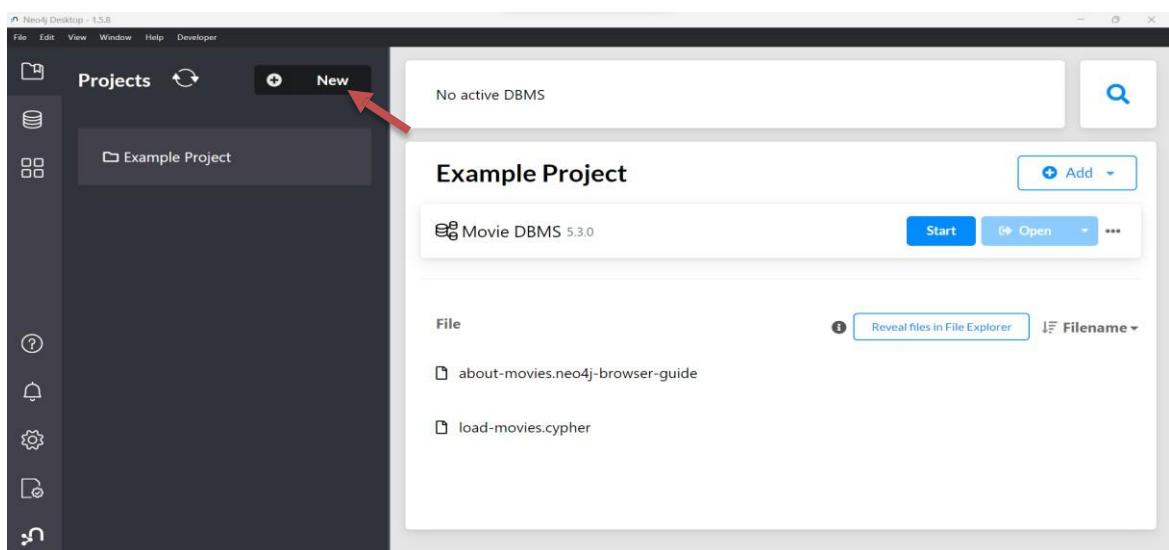


7. Jika sudah berhasil install aplikasi Neo4j maka tampilan layer akan seperti dibawah ini

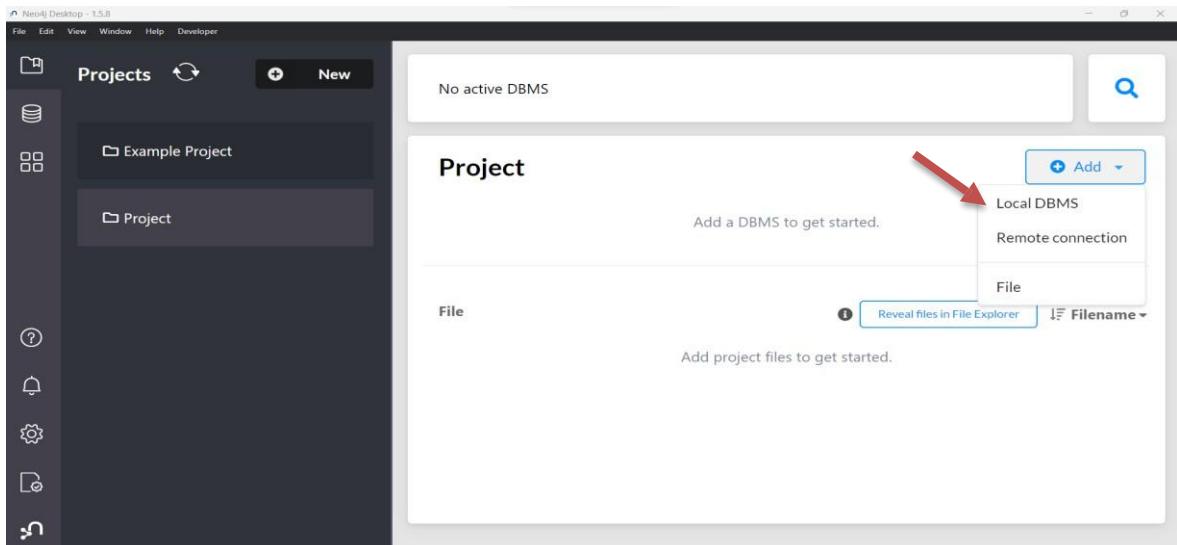


II. Cara Memulai Aplikasi Neo4j Desktop dan Membangun Graph Database

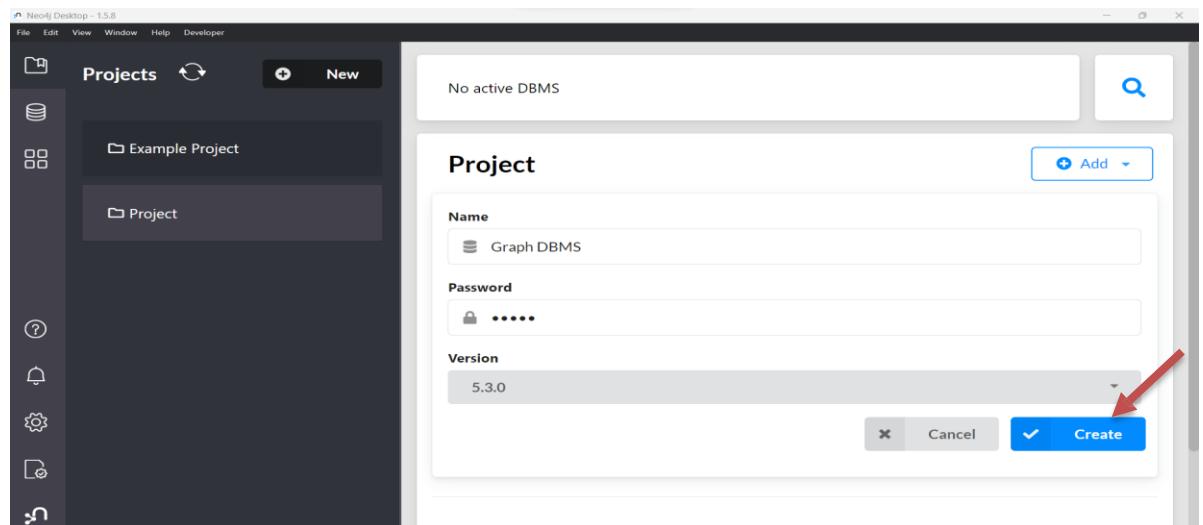
1. Bukalah aplikasi neo4j dan tampilan layar akan muncul, lalu untuk membuat project baru dengan mengklik tombol “New” di bagian kiri atas layar kemudain pilih create project



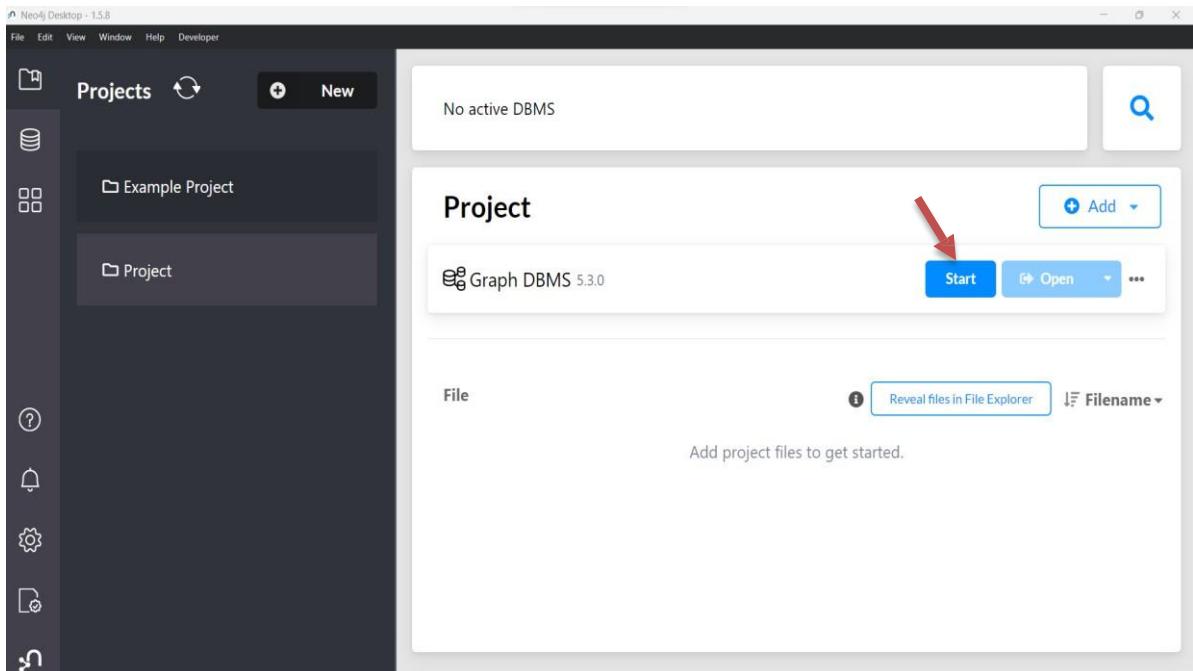
2. Tampilan Neo4j untuk new project seperti dibawah ini. Kemudian klik add lalu pilih local DBMS



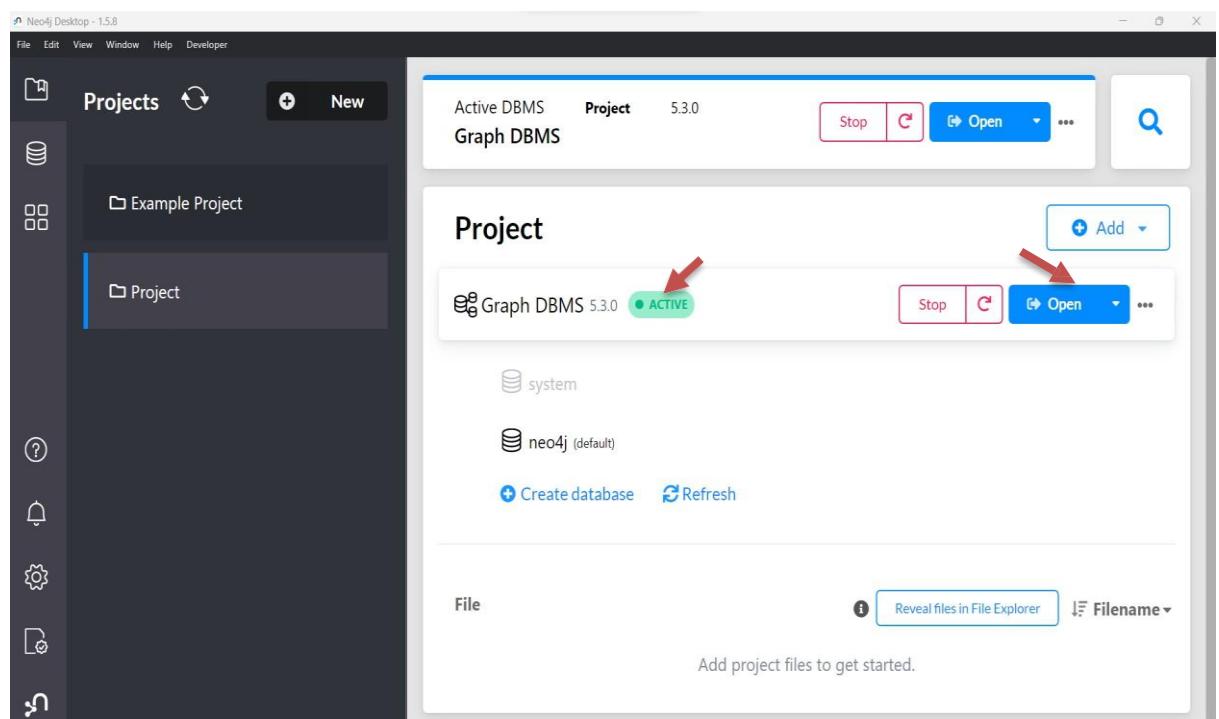
3. Langkah selanjutnya membuat nama database "Graph DBMS" lalu isi password minimal 8 karakter selanjutnya klik create. Password sesuai keinginan masing-masing.



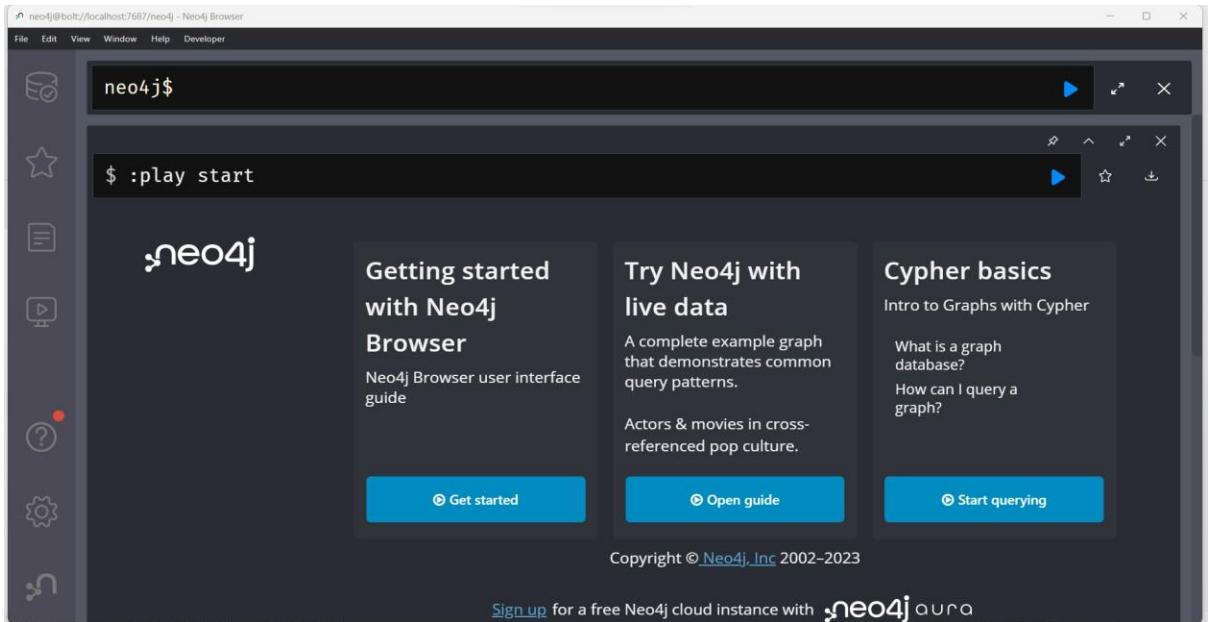
4. Kemudian untuk memulai Graph Database dengan klik start.



5. Tahap selanjutnya pastikan Graph DBMS sudah active kemudian pilih open with Neo4j browser.



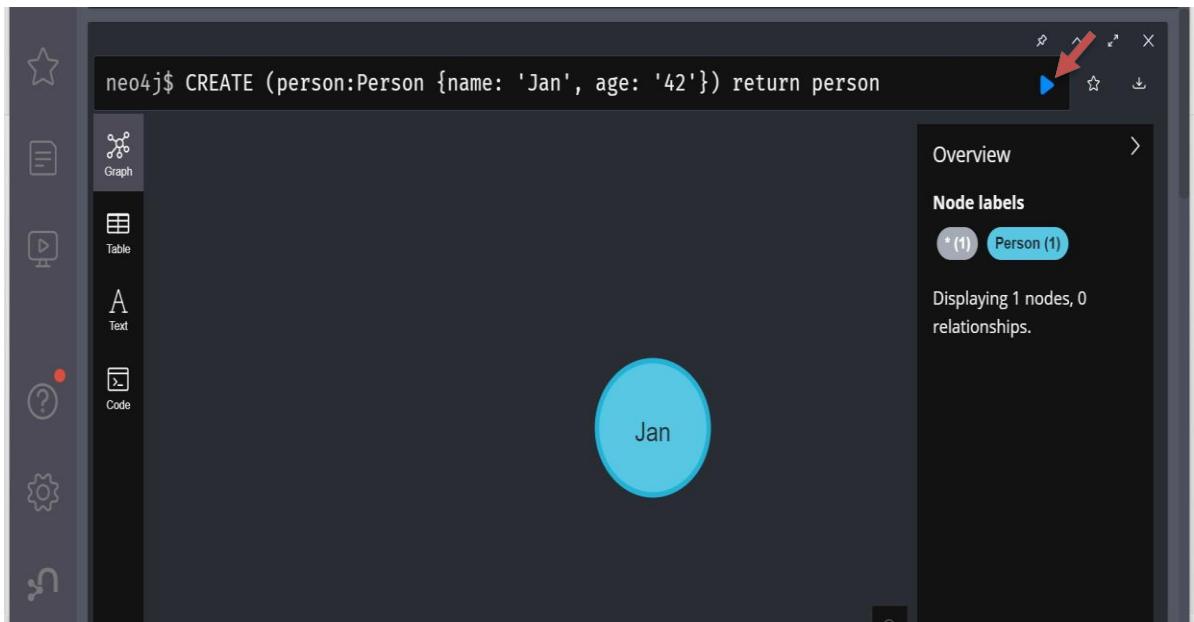
Neo4j browser akan ditampilkan seperti dibawah ini



6. Tahap selanjutnya membuat node terlebih dahulu seperti dibawah ini
➤ **Membuat node tanpa relationships secara manual**

```
CREATE (person:Person {name: 'Jan', age: '42'}) return person
```

buatlah syntax seperti diatas lalu klik tanda panah biru



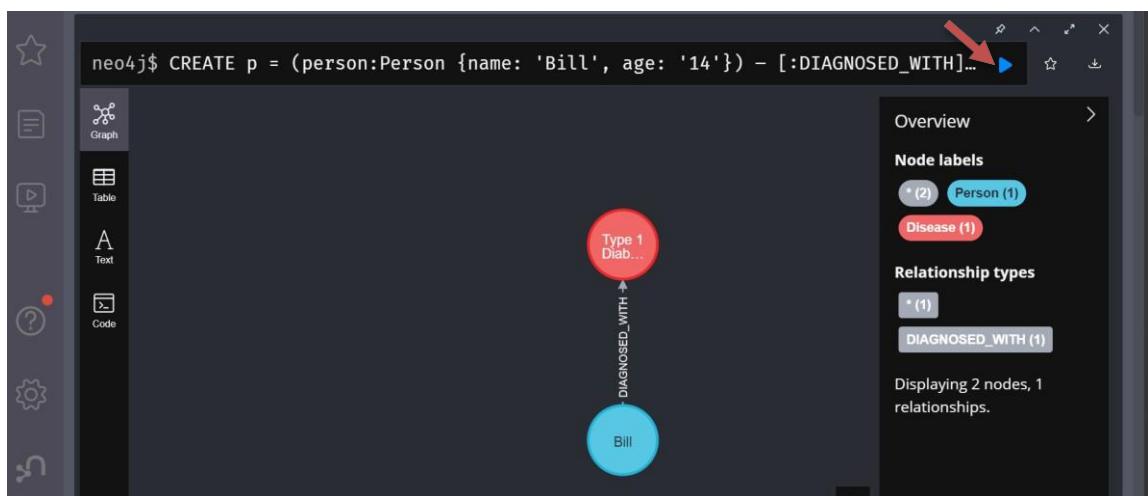
Jika sudah berhasil maka akan ditampilkan seperti ini dengan keterangan added 1 label, created 1 node

```
neo4j$ CREATE (person:Person {name: 'Jan', age: '42'}) return person
```

➤ Membuat node dengan relationships secara manual

```
CREATE p = (person:Person {name: 'Bill', age: '14'}) - [:DIAGNOSED_WITH] -> (disease:Disease { name: 'Type 1 Diabetes' }) RETURN p
```

buatlah syntax seperti diatas lalu klik tanda panah biru



Jika sudah berhasil maka akan ditampilkan seperti ini dengan keterangan added 2 label, created 2 node, set 3 properties dan created 1 relationship.

```

neo4j$ CREATE p = (person:Person {name: 'Bill', age: '14'}) - [:DIAGNOSED_WITH] -> (disease:Disease { name... })
    
```

7. Kemudian step-step selanjutnya dengan **membuat relationships** dengan menambahkan relationships diantara node. Berikut syntax ditampilkan dibawah ini.

```

MATCH (p:Person {name:'Jan'}), (f:Person {name:'Samantha'})
CREATE (p)-[:FRIENDS_WITH {since: 2009}]->(f)
    
```

buatlah syntax seperti diatas lalu klik tanda panah biru. Jika sudah berhasil maka akan ditampilkan seperti ini

```

neo4j$ MATCH (p:Person {name:'Jan'}), (f:Person {name:'Samantha'}) CREATE (p)-[:FRIENDS_WITH {since: 2009}]->(f)
    
```

8. Kita dapat memperbarui atau **update node properties** dengan menggunakan syntax dibawah ini

```

MATCH (person:Person { name: 'Jan' }) SET person.profession = 'Software Engineer' RETURN person
    
```

buatlah syntax seperti diatas lalu klik tanda panah biru.

```

neo4j$ MATCH (person:Person { name: 'Jan' }) SET person.profession = 'Software Engineer' RETURN person
    
```

Jika sudah berhasil maka akan ditampilkan 3 set properties seperti ini

```
neo4j$ MATCH (person:Person { name: 'Jan' }) SET person.profession = 'Software Engineer' RETURN person
```

```
person
```

```
1
{
    "identity": 10,
    "labels": [
        "Person"
    ],
    "properties": {
        "profession": "Software Engineer",
        "name": "Jan",
        "age": "42"
    },
    "elementId": "10"
}
2
{
    "identity": 11,
    "labels": []
}
```

Set 3 properties, started streaming 3 records after 25 ms and completed after 30 ms.

9. Kita dapat **menghapus relationship dan node** dengan cara dibawah ini

➤ **Menghapus relationship**

Gunakanlah syntax untuk menghapus relationship

```
MATCH ()-[r:FRIENDS_WITH]-() DELETE r
```

Maka akan ditampilkan hasilnya seperti ini

```
neo4j$ MATCH ()-[r:FRIENDS_WITH]-() DELETE r
```

(no changes, no records)

Completed after 33 ms.

➤ **Menghapus Node**

Gunakanlah syntax untuk menghapus node

```
MATCH (a:Camp) WHERE a.name='Joselin Diabetes Camp' DELETE a
```

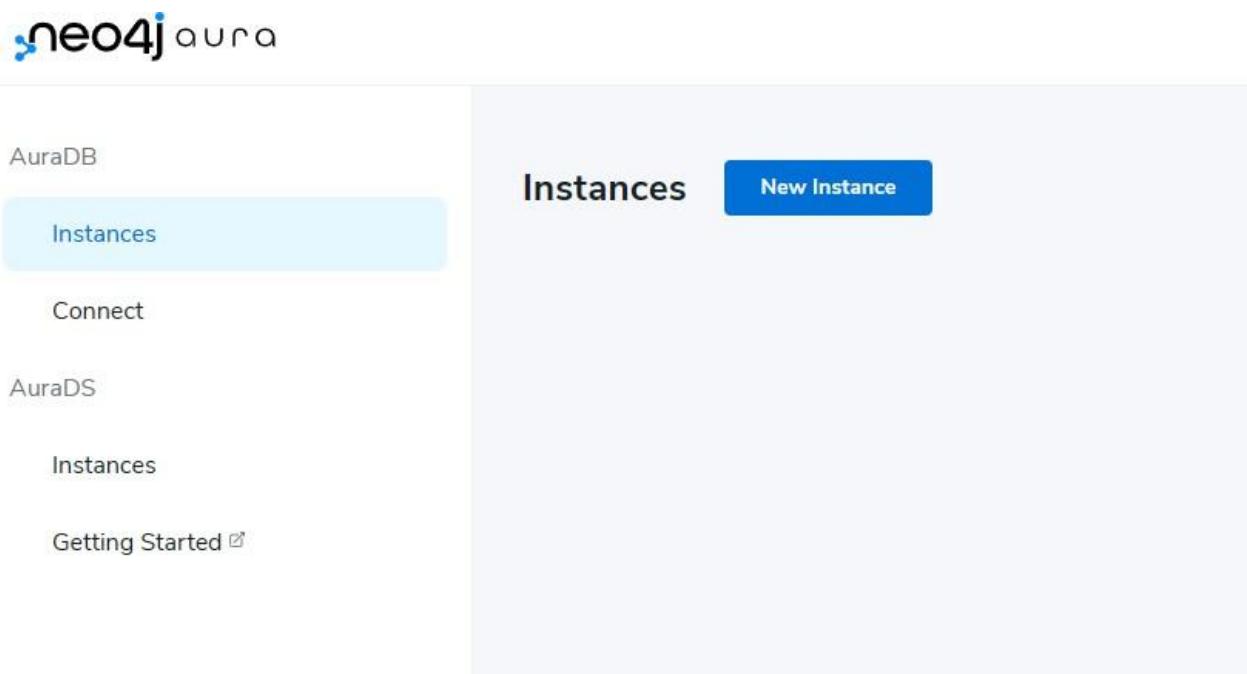
Maka akan ditampilkan hasilnya seperti ini



A screenshot of the Neo4j browser interface. The main window shows a command line input: `neo4j$ MATCH (a:Camp) WHERE a.name='Joselin Diabetes Camp' DELETE a`. Below the command, the output is displayed: `(no changes, no records)`. At the bottom of the window, it says `Completed after 26 ms.`. On the left side, there is a sidebar with three icons: a star, a table, and a code editor. The table icon is currently selected.

III. Penggunaan neo4j aura

- Link untuk neo4j <https://neo4j.com/> masuk ke neo4j free for develop masuk ke neo4j aura
- Siapkan akun google masing-masing



A screenshot of the neo4j aura dashboard. The top navigation bar has the neo4j aura logo. Below the navigation, there are two main sections: "AuraDB" on the left and "Instances" on the right. The "Instances" section is active, indicated by a blue background. It contains a "New Instance" button. On the left sidebar under "AuraDB", there are links for "Instances" (which is highlighted in blue), "Connect", and "Getting Started". On the right panel under "Instances", there is a table with one row, which is collapsed. The table columns are "Name", "Status", "Type", and "Actions". The single row shows "My First Neo4j DB", "Up", "beginner", and a "View" button.

- Buat New Instances pilih beginner dan NOTE penting untuk pastikan mendownload password sebelum mengkonfirmasi.

Credentials for Instance01

Username: neo4j

Generated password

c-vPRJvJzCdc-PB343LJT33zs7qOwTPGqR

[Download](#)

We strongly advise changing this initial password.

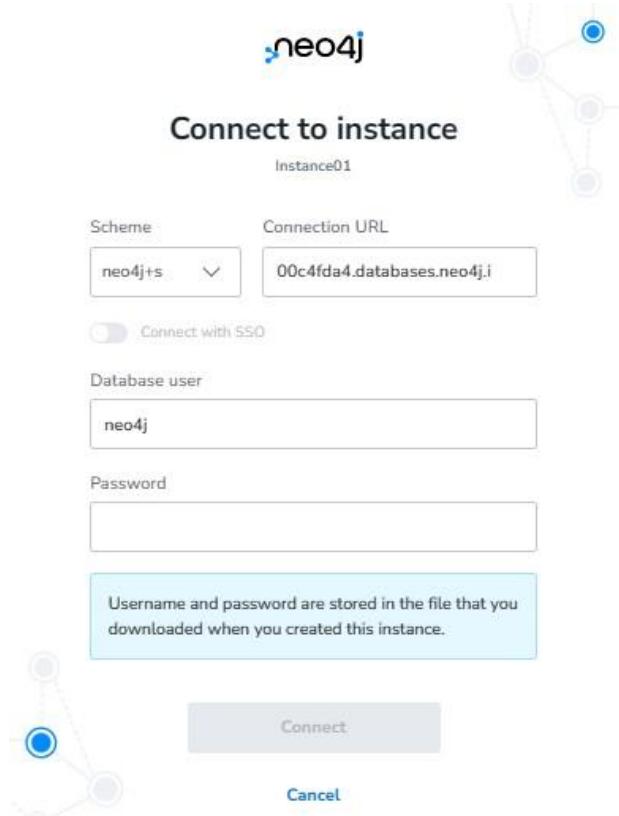
- I confirm I have copied or downloaded the above credentials, as this password will not be available after this point

[Continue](#)

- setelah dikonfirmasi klik open

The screenshot shows the neo4j aura dashboard. On the left sidebar, there are links for AuraDB, Instances (which is highlighted in blue), Connect, and AuraDS. Below these are links for Instances, Getting Started, and a question mark icon. The main content area is titled 'Instances' and shows a list of instances. The first instance listed is 'Instance01' which is 'Free' and 'Running'. It provides details such as Neo4j version 5, 0 nodes (0%), 0 relationships (0%), Region Singapore (asia-southeast1), GCP, and a Connection URI: neo4j+s://00c4fda4.databases.neo4j.io. There are also icons for deleting the instance and more options.

- Siapkan password yang telah didownload untuk tahap dibawah.



- Copy NEO4J_PASSWORD dari dokumen yang didownload

A. CREATING NODES

```

1 CREATE (person:Person {name: 'Jan', age: '42'}) return person
2 person

```

Graph Database

Dengan output:

The screenshot shows the Neo4j browser interface. At the top, there is a query input field containing the text: '\$ MATCH (n:Person) RETURN n LIMIT 25;'. Below the input field are three buttons: 'Graph' (highlighted in blue), 'Table', and 'RAW'. To the right of these buttons is a download icon. The main area displays a single node, represented by a green circle with the label '"Jan"' inside.

- Lakukan cara diatas buatlah:
 - node person dengan Samantha dengan umur 32{masih dalam node person}
 - node untuk camp dengan Joselin Diabetes Camp {node yang berbeda}

B. Manually create nodes with relationships

The screenshot shows the Neo4j browser interface. The 'Query' tab is selected. On the left, there is a sidebar with 'Database Information', 'Nodes (1)', and a 'Person' button. The main area contains a query editor with the following code:

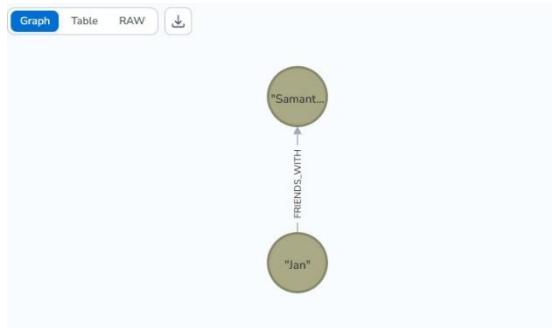
```
1 CREATE p = (person:Person {name: 'Bill', age: '14'}) ->
2 [:DIAGNOSED_WITH] -> (disease:Disease { name: 'Type 1
3 Diabetes' }) RETURN p
```

The screenshot shows the Neo4j browser interface after the query has been executed. The 'Query' tab is still selected. On the left, the sidebar shows 'Database Information', 'Nodes (3)' (with 'Disease' and 'Person' buttons), 'Relationships (1)' (with 'DIAGNOSED_WITH' button), and 'Property keys' (with buttons for 'age', 'data', 'id', 'name', 'nodes', 'relationships', 'style', and 'visualisation'). The main area displays two nodes: a green node labeled '"Bill"' and a blue node labeled '"Type 1 Diab..."' (with an ellipsis). A vertical arrow between them indicates a relationship named 'DIAGNOSED_WITH'. At the bottom of the screen, a message states: 'Created 2 nodes, created 1 relationship, set 3 properties, added 2 labels' and 'Started streaming 1 record after 43ms and completed after 58ms.'

C. ADDING RELATIONSHIPS

Add a relationship between people nodes

```
1 MATCH (p:Person {name:'Jan'}), (f:Person {name:'Samantha'})  
2 CREATE (p)-[:FRIENDS_WITH {since: 2009}]->(f)
```



D. UPDATING NODE PROPERTIES

```
1 MATCH (person:Person { name: 'Jan' })  
2 SET person.profession = 'Software Engineer'  
3 RETURN person
```

- Setelah di run maka dalam tampilan table akan berubah jan adalah sofwere engineer

Graph Table RAW

n

```
{  
  identity: 0,  
  labels: ["Person"],  
  properties: {  
    profession: "Software Engineer",  
    name: "Jan",  
    age: "42"  
  },  
  elementId: "4:a584c95a-8ca7-4535-98de-4355888bf865:0"  
}
```

E. DELETING RELATIONSHIPS AND NODES

- Menghapus relationships

```
$ MATCH ()-[r:FRIENDS_WITH]-() DELETE r
```

base

```
$ MATCH ()-[r:FRIENDS_WITH]-( ) DELETE r
```

□ ⏴ ⏵ ⏷ ×

✓ Deleted 1 relationship

Completed after 42ms

- Menghapus node

```
1 MATCH (a:Camp)
2 WHERE a.name='Joselin Diabetes Camp'
3 DELETE a
```

