

## **BAB II**

### **PELAKSANAAN UJIAN KOMPETENSI**

#### **2.1. Pendahuluan**

Uji kompetensi Neo4j Certified Professional dilaksanakan secara daring melalui platform resmi Neo4j. Proses sertifikasi ini mencakup pembelajaran teori, latihan praktis (hands-on lab), serta ujian akhir yang terdiri dari teori dan praktik. Tahapan-tahapan ini dirancang untuk memastikan peserta tidak hanya memahami konsep dasar, tetapi juga mampu menerapkannya secara langsung pada kasus nyata.

#### **2.2. Tujuan Sertifikasi**

Tujuan dari mengikuti sertifikasi Neo4j Certified Professional ini adalah sebagai berikut:

1. Meningkatkan kompetensi individu di bidang basis data graf, yaitu agar peserta mampu memahami secara mendalam konsep graph database, perbedaannya dengan basis data relasional, serta keunggulan yang ditawarkan dalam pengelolaan data yang kompleks dan saling terhubung.
2. Menguasai penggunaan Neo4j secara teoritis dan praktis, termasuk penguasaan bahasa kueri Cypher, membangun model data graf, serta mengimplementasikan berbagai kasus penggunaan nyata dalam pengelolaan data.
3. Mendapatkan pengakuan profesional bertaraf internasional sebagai bukti kompetensi yang diakui secara global, sehingga dapat meningkatkan kredibilitas pemegang sertifikat di mata industri.

4. Menunjang proses pembelajaran akademik melalui penguasaan konsep basis data modern yang sesuai dengan kebutuhan penelitian maupun aplikasi industri.

### **2.3. Manfaat Sertifikasi**

Manfaat dari mengikuti sertifikasi Neo4j Certified Professional ini adalah sebagai berikut:

1. Peningkatan daya saing di dunia kerja, karena pemegang sertifikat memiliki nilai tambah saat melamar pekerjaan, terutama di bidang teknologi data dan rekayasa perangkat lunak.
2. Penguatan pemahaman konsep dan praktik basis data graf, baik secara teori maupun aplikasi melalui hands-on lab dan studi kasus nyata.
3. Pengakuan kompetensi oleh komunitas global, mengingat sertifikat ini dikeluarkan langsung oleh Neo4j, Inc. dan diakui secara internasional.
4. Dukungan bagi pengembangan karier maupun riset akademik, khususnya dalam bidang data engineering, analisis data, serta graph-based application development.

### **2.4. Silabus dan Materi yang Dipelajari**

Sebelum penulis mengikuti ujian sertifikasi Neo4j Certified Professional, terdapat rangkaian course yang harus diselesaikan sebagai bagian dari persiapan. Silabus ini dirancang secara sistematis agar penulis memahami konsep dasar hingga penerapan praktis Neo4j dalam berbagai konteks pemrograman. Setiap course memiliki peran penting dalam membangun fondasi pengetahuan, keterampilan teknis, sekaligus kesiapan menghadapi soal-soal berbasis teori maupun praktik.

## Prerequisites

We recommend completing the following courses to learn everything you need to know to pass this certification.

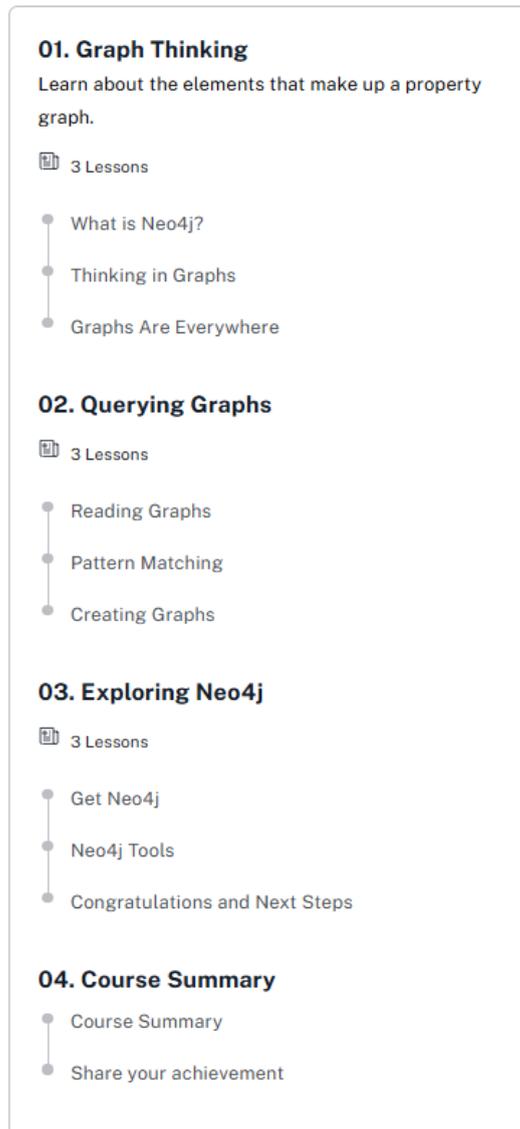
- Neo4j Fundamentals
- Cypher Fundamentals
- Graph Data Modeling Fundamentals
- Importing Data Fundamentals
- Intermediate Cypher Queries
- Using Neo4j with Python

Gambar 2. 1.Silabus Sertifikasi Neo4j Certified Professional

Materi yang dipelajari sebelum mengikuti sertifikasi dapat dilihat pada gambar 2.1 di atas. Berikut rincian silabus yang dipelajari:

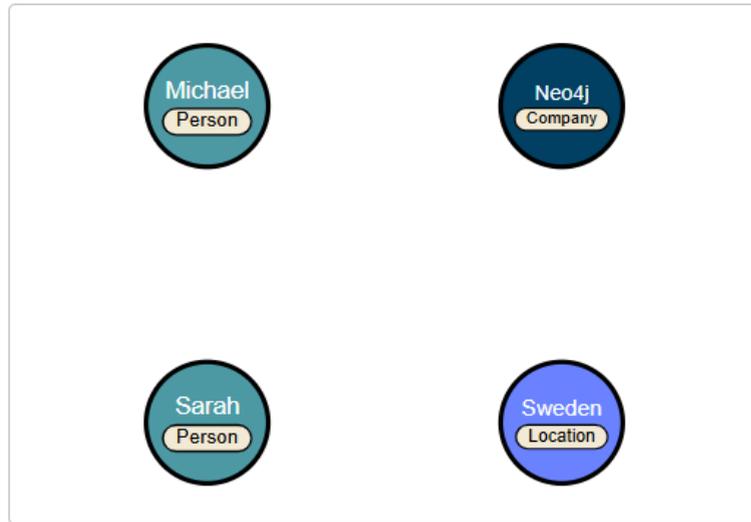
### 2.4.1. Neo4j Fundamentals

Course Neo4j Fundamentals merupakan materi pengantar yang memberikan pemahaman mendasar mengenai konsep graph database serta memperkenalkan Neo4j sebagai salah satu platform basis data graf yang paling banyak digunakan secara global.



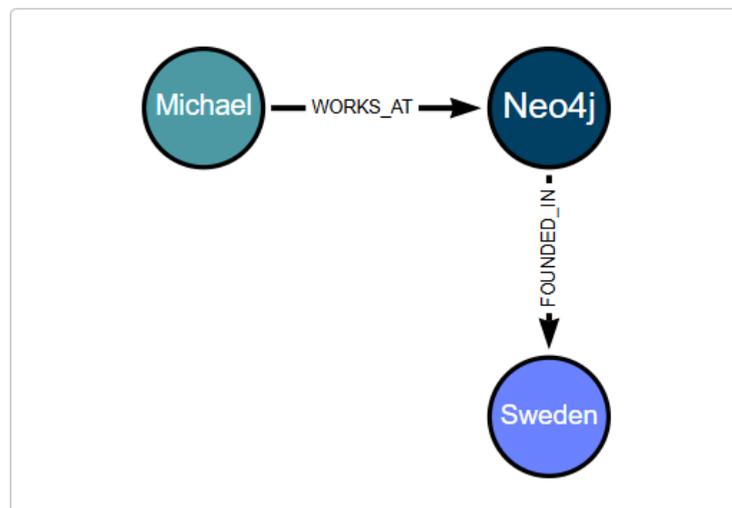
Gambar 2. 2.Silabus Neo4j Fundamentals

Pada gambar 2.2 adalah gambaran secara umum materi yang dipelajari pada course ini. Pada bagian pertama penulis dikenalkan dengan neo4j. Bahwa ia adalah sebuah basis data graf. Berbeda dengan basis data relasional biasa, dimana data disimpan dalam bentuk tabel. Pada basis data graf, data disimpan sebagai node dan relasi.



Gambar 2. 3. Bentuk Node

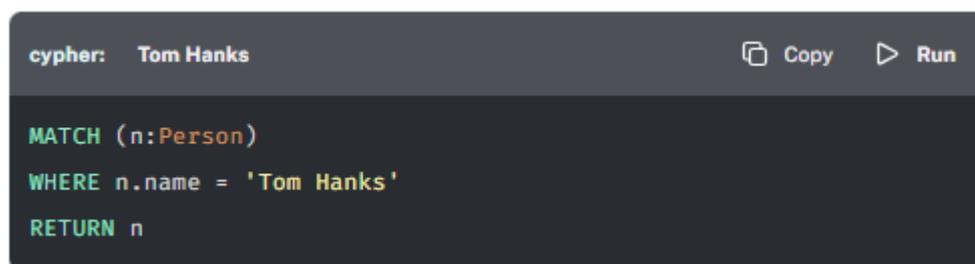
Gambar 2.3 menunjukkan node. Node dikelompokkan berdasarkan dengan label. Pada gambar terdapat label Person, Company, dan Location. Label membuat kita dapat membedakan tipe dari node yang berbeda. Node bisa saja memiliki label lebih dari 1 label.



Gambar 2. 4. Relasi

Gambar 2.4 menunjukkan bagaimana relasi antar node. Relasi merupakan garis di dalam graf. Relasi mendeskripsikan bagaimana node dalam graf terkoneksi dengan lainnya. Misal pada contoh di atas, nama relasinya adalah: WORKS\_AT, maka kita membacanya Michael WORKS\_AT Neo4j.

Pada bagian ini pula kita mulai menjalankan query untuk menampilkan node yang kita inginkan.

A screenshot of a Cypher query editor interface. The top bar shows 'cypher: Tom Hanks' on the left and 'Copy' and 'Run' buttons on the right. The main area contains the following Cypher query:

```
MATCH (n:Person)
WHERE n.name = 'Tom Hanks'
RETURN n
```

Gambar 2. 5. Menggunakan Cypher

Pada gambar 2.5 kita mulai menggunakan bahasa cypher. Bahasa yang digunakan untuk melakukan query terhadap basis data yang kita miliki. Perintah di atas digunakan untuk mencari node dengan label Person yang memiliki properti name bernilai “Tom Hanks”. Bagian MATCH (n:Person) berarti sistem akan mencari semua node dengan label Person dan merepresentasikannya dengan variabel n. Kemudian WHERE n.name = ‘Tom Hanks’ memberikan syarat agar hanya node yang bernama Tom Hanks saja yang dipilih. Akhirnya, RETURN n akan menampilkan node tersebut sebagai hasil query.

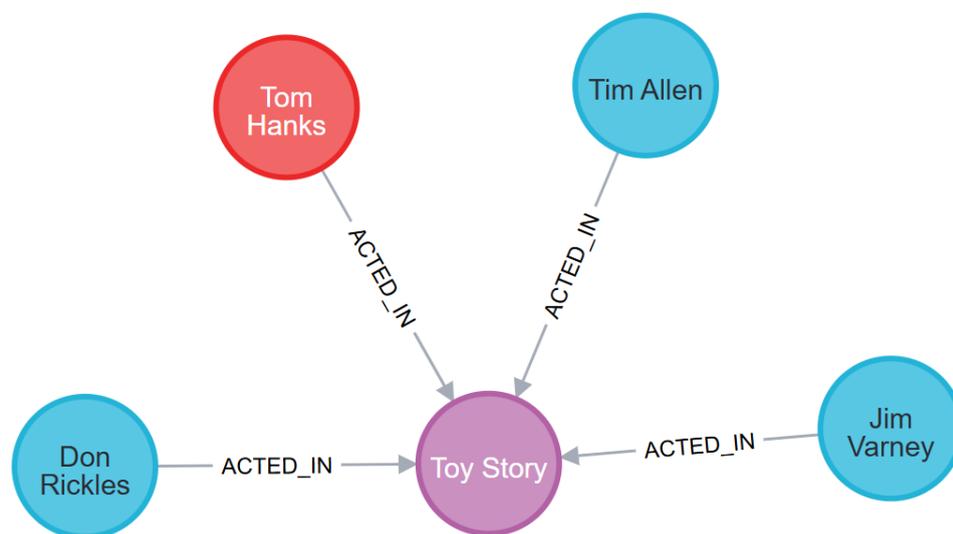
```

cypher: Copy Run
MATCH (m:Movie)<-[r:ACTED_IN]-(p:Person)
WHERE m.title = 'Toy Story'
RETURN m, r, p

```

Gambar 2. 6. Query Film Toy Story

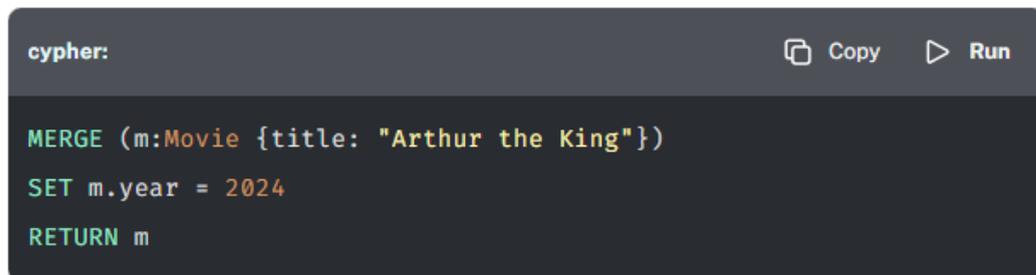
Query pada Gambar 2.6 digunakan untuk menggambarkan bagaimana Neo4j menampilkan hubungan antara sebuah film dengan para pemerannya. Pada contoh ini, penulis menuliskan perintah untuk mencari film berjudul “Toy Story” serta menampilkan aktor-aktor yang terlibat di dalamnya. Sintaks `MATCH (m:Movie)<-[r:ACTED_IN]-(p:Person)` menunjukkan adanya node `Movie` yang berhubungan dengan node `Person` melalui relasi `ACTED_IN`. Selanjutnya, perintah `WHERE m.title = 'Toy Story'` memberikan syarat bahwa film yang dicari adalah “Toy Story”.



Gambar 2. 7. Visualisasi Node

Dapat kita lihat pada Gambar 2.7, Hasil dari query ini berupa visualisasi node

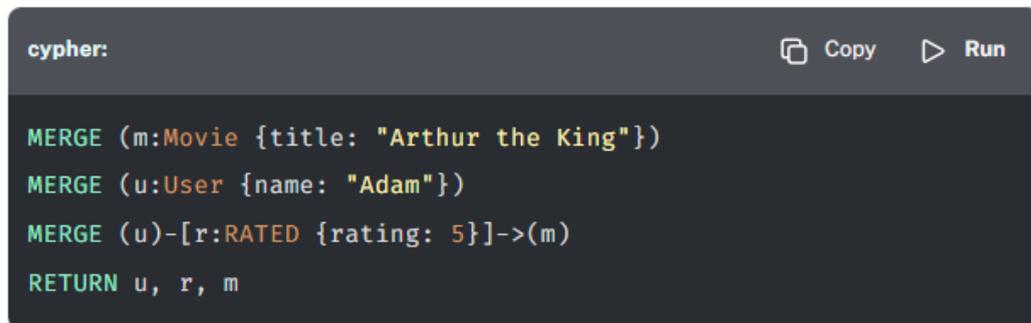
film, node orang (aktor atau aktris), serta relasi yang menghubungkan keduanya. Query ini menampilkan seluruh orang yang berakting di film Toy Story. Dengan demikian, Neo4j tidak hanya menyajikan data film secara individual, tetapi juga memvisualisasikan keterkaitan antardata, sehingga memudahkan pengguna dalam memahami jaringan hubungan antar entitas.

A screenshot of a Cypher query editor interface. The interface has a dark background with light-colored text. At the top left, it says "cypher:". At the top right, there are two buttons: "Copy" with a clipboard icon and "Run" with a play button icon. The main area contains the following Cypher query:

```
MERGE (m:Movie {title: "Arthur the King"})  
SET m.year = 2024  
RETURN m
```

Gambar 2. 8. Membuat Node

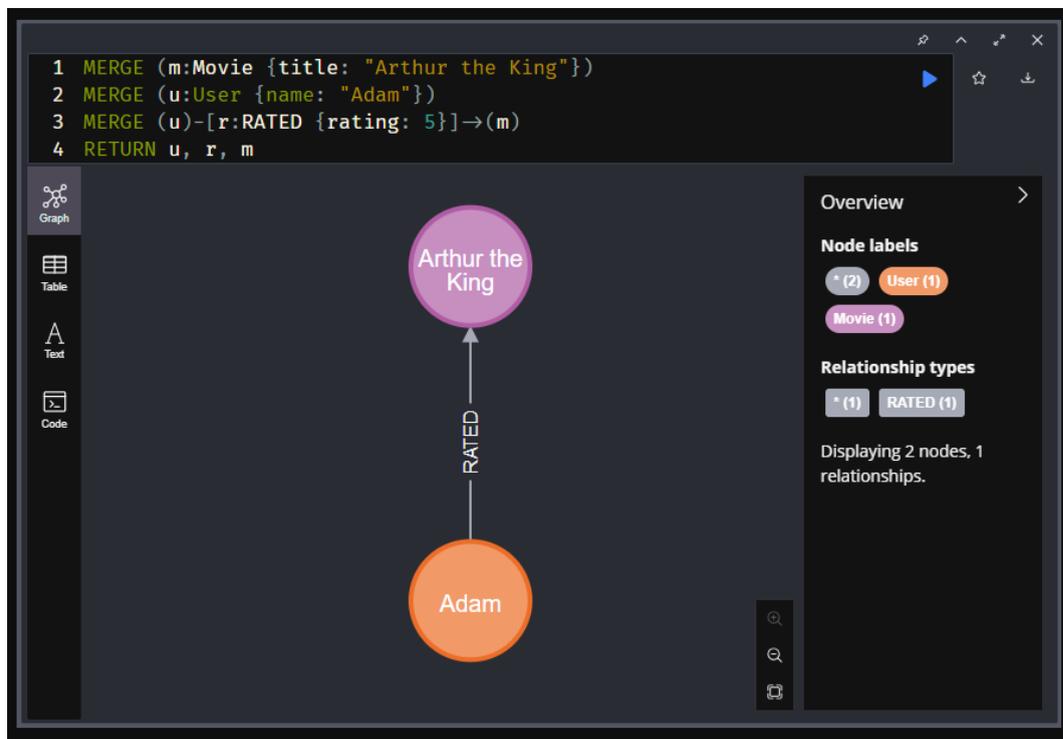
Query di atas digunakan untuk menambahkan sebuah node film baru ke dalam basis data Neo4j. Perintah `MERGE (m:Movie {title: "Arthur the King"})` memastikan bahwa node dengan label `Movie` dan properti `title` bernilai "Arthur the King" akan dibuat jika sebelumnya belum ada, atau menggunakan node yang sudah ada jika ditemukan judul yang sama. Selanjutnya, perintah `SET m.year = 2024` digunakan untuk menambahkan atau memperbarui properti `year` pada node tersebut dengan nilai 2024. `RETURN m` menampilkan node `Movie` yang telah dimasukkan atau diperbarui. Dengan kata lain, perintah ini meminta untuk membuat node baru dengan label `Movie` dan judulnya "Arthur The King", tahun 2004.

A screenshot of a Cypher query editor interface. The interface has a dark background with light-colored text. At the top left, it says "cypher:". At the top right, there are two icons: a copy icon and a play button icon, with the labels "Copy" and "Run" respectively. The main area contains the following Cypher query:

```
MERGE (m:Movie {title: "Arthur the King"})
MERGE (u:User {name: "Adam"})
MERGE (u)-[r:RATED {rating: 5}]->(m)
RETURN u, r, m
```

Gambar 2. 9. Membuat Relasi

Query di atas digunakan untuk membuat node Movie dengan judul “Arthur the King” serta node User dengan nama Adam. Perintah MERGE memastikan bahwa jika node dengan atribut yang sama sudah ada di dalam basis data, maka tidak akan dibuat duplikasi baru, melainkan menggunakan node yang sudah ada. Selanjutnya, dibentuk relasi RATED dari node User menuju node Movie, dengan properti tambahan berupa nilai rating: 5 yang menunjukkan penilaian yang diberikan oleh pengguna terhadap film tersebut. Perintah RETURN u, r, m menampilkan hasil berupa node pengguna, node film, dan relasi di antara keduanya. Dengan kata lain query ini membuat node baru dengan label Movie dan judul “Arthur the King”, lalu membuat node berlabel User, dan namanya Adam. Kemudian antara kedua node itu diberi relasi bernama RATED, dimana arah relasinya dari user ke Movie yang diberikan rating dengan nilai 5. Dengan demikian, query ini tidak hanya mencatat keberadaan film dan pengguna, tetapi juga interaksi berupa pemberian rating.



Gambar 2. 10. Visualisasi Buat Relasi Baru

Gambar di atas memperlihatkan bagaimana tampilan node dan relasi yang baru saja dibuat. Dari gambar ini pembaca bisa langsung melihat dan paham bahwa terdapat 2 buah node dan 1 relasi yaitu RATED pada graf di atas.

Semua contoh ini menunjukkan bagaimana Neo4j memungkinkan kita melakukan pencarian data berdasarkan hubungan antar entitas maupun atribut spesifik, dengan cara yang lebih intuitif dibandingkan basis data relasional.

Proses pembelajaran dilakukan melalui kombinasi materi video, bacaan penjelasan, dan latihan interaktif. Penulis diajak untuk mencoba langsung melakukan eksplorasi terhadap dataset graf sederhana, sehingga pemahaman teori dapat segera diaplikasikan dalam praktik nyata. Selain itu, terdapat kuis evaluasi di akhir sesi yang berfungsi untuk mengukur tingkat pemahaman penulis terhadap materi yang telah dipelajari.

Dengan mengikuti course ini, penulis memperoleh landasan teoretis dan praktis yang kokoh untuk melanjutkan ke tahap berikutnya, seperti penguasaan bahasa query Cypher, hingga teknik pemodelan data graf yang lebih kompleks. Secara keseluruhan, Neo4j Fundamentals menjadi pintu masuk yang sangat penting dalam membangun kompetensi sebagai seorang profesional yang ingin menguasai teknologi basis data graf.

#### **2.4.2. Cypher Fundamentals**

Course Cypher Fundamentals berfokus pada penguasaan dasar-dasar Cypher, yaitu bahasa query deklaratif yang digunakan dalam Neo4j untuk melakukan pencarian, manipulasi, dan analisis data berbasis graf. Cypher dirancang agar mudah dipelajari dengan sintaks yang menyerupai bahasa alami, sehingga memudahkan pengguna dari berbagai latar belakang, baik teknis maupun non-teknis, untuk memahami dan mengeksekusi query terhadap data graf.



**01. Reading Data from Neo4j**

 7 Lessons

- Introduction to Cypher
- Retrieving Nodes
- Finding Relationships
- Traversing Relationships
- Finding Emil
- Filtering Queries
- Finding Specific Actors

**02. Writing Data to Neo4j**

 10 Lessons

- Creating Nodes
- Creating a Node
- Creating Relationships
- Creating a Relationship
- Updating Properties
- Adding Properties to a Movie
- Merge Processing
- Adding or Updating a Movie
- Deleting Data
- Deleting Emil

**03. Course Summary**

- Course Summary
- Share your achievement

Gambar 2. 11. Silabus Cypher Fundamentals

Materi dimulai dengan pengenalan struktur dasar query Cypher, termasuk cara melakukan pencarian sederhana dengan perintah MATCH, penggunaan RETURN untuk menampilkan hasil, serta eksplorasi pola hubungan antar nodes. Penulis juga diperkenalkan pada berbagai klausa penting seperti WHERE untuk penyaringan data.

Selain operasi pencarian, course ini juga membahas perintah yang berkaitan dengan manipulasi data. Hal ini mencakup penggunaan CREATE untuk menambahkan nodes dan relationships, SET untuk memperbarui atribut, serta DELETE untuk menghapus entitas tertentu. Materi ini penting karena memberikan gambaran menyeluruh tentang bagaimana Cypher tidak hanya digunakan untuk membaca data, tetapi juga untuk membangun dan memodifikasi struktur graf sesuai kebutuhan aplikasi.

Pendekatan pembelajaran dilakukan dengan metode hands-on lab yang memungkinkan penulis mencoba langsung menuliskan query pada dataset yang telah disediakan. Visualisasi hasil query di Neo4j Browser membantu memperkuat pemahaman, karena penulis dapat melihat representasi grafis dari relasi yang terbentuk. Latihan-latihan yang diberikan secara bertahap menuntun penulis dari query paling sederhana hingga query dengan kompleksitas menengah.

Di akhir course, penulis diberikan kuis evaluasi untuk mengukur sejauh mana penguasaan penulis terhadap Cypher. Hasil evaluasi ini menjadi tolok ukur awal yang penting sebelum melangkah ke level query yang lebih kompleks pada course selanjutnya. Dengan menguasai Cypher Fundamentals, penulis memperoleh keahlian dasar yang esensial, karena bahasa ini merupakan alat utama dalam

berinteraksi dengan Neo4j dan menjadi kompetensi yang paling sering digunakan dalam praktik profesional sehari-hari.

### **2.4.3. Graph Data Modeling Fundamentals**

Course Graph Data Modeling Fundamentals membekali penulis dengan pemahaman mendasar mengenai bagaimana merancang model data berbasis graf secara efektif serta sesuai dengan kebutuhan sistem informasi. Berbeda dengan basis data relasional yang mengorganisasi data dalam bentuk tabel, kolom, dan baris, basis data graf menekankan representasi entitas sebagai nodes serta hubungan antarentitas sebagai relationships. Dengan pendekatan ini, pemodelan data graf menuntut strategi yang berbeda karena hubungan antardata dipandang sama pentingnya dengan data itu sendiri.

Materi pembelajaran diawali dengan pengenalan prinsip dasar pemodelan data graf. Penulis diperkenalkan pada konsep identifikasi nodes sebagai entitas utama, penggunaan labels untuk mengelompokkan entitas berdasarkan kategori tertentu, serta penambahan properties sebagai atribut pendukung. Selanjutnya, penulis mempelajari bagaimana merancang relationships secara tepat agar mampu merepresentasikan keterhubungan antarentitas secara alami, misalnya hubungan antara pengguna dengan produk atau antara aktor dengan film.

<b>01. Getting Started</b>
3 Lessons
• What is Graph Data Modeling?
• The Domain
• Purpose of the Model
<b>02. Modeling Nodes</b>
4 Lessons
• Modeling Nodes
• Creating Nodes
• Identifying a New Label
• Creating More Nodes
<b>03. Modeling Relationships</b>
4 Lessons
• Modeling Relationships
• Creating Initial Relationships
• Identifying a New Relationship
• Creating More Relationships
<b>04. Testing the Model</b>
2 Lessons
• Testing
• Testing with Instance Model

Gambar 2. 12. Silabus Graph Data Modeling Fundamentals

Dapat dilihat pada gambar di atas bahwa alur pembelajaran dimulai dengan pengenalan dasar mengenai pemodelan data graf. Bagian ini menjelaskan apa yang dimaksud dengan pemodelan data graf, ruang lingkup atau domain yang akan digunakan, serta tujuan dari model yang dibangun. Penjelasan awal ini penting karena memberikan pemahaman konseptual mengenai arah dan manfaat dari rancangan model yang dibuat.

Setelah itu, pembelajaran berlanjut pada tahap pemodelan node dan relationship. Pada tahap pemodelan node, penulis belajar bagaimana merancang

entitas sebagai node, menambahkan node baru, serta mengelompokkan node berdasarkan kategori tertentu. Sementara itu, pada tahap pemodelan relationship, penulis diarahkan untuk memahami bagaimana membangun keterkaitan antar node, baik melalui hubungan awal, hubungan baru yang diidentifikasi, maupun penambahan relasi lebih lanjut. Dengan urutan pembelajaran ini, pemodelan data graf dipahami secara bertahap, mulai dari konsep dasar hingga praktik teknis yang lebih rinci.

Course ini juga menjelaskan bagaimana pemilihan struktur graf yang tepat akan memengaruhi kualitas analisis data dan efisiensi eksekusi query. Berbagai contoh kasus disajikan untuk menunjukkan perbedaan hasil yang muncul apabila pemodelan dilakukan secara tepat dibandingkan dengan pemodelan yang kurang sesuai. Hal ini menegaskan bahwa desain graf yang baik tidak hanya mendukung integritas data, tetapi juga menghasilkan query yang lebih sederhana, efisien, dan mudah dipelihara dalam jangka panjang.

Di akhir course, penulis diberikan evaluasi berupa kuis untuk menguji sejauh mana pemahaman penulis terhadap konsep pemodelan graf. Dengan menyelesaikan course ini, penulis diharapkan mampu merancang struktur data graf yang tidak hanya memenuhi kebutuhan fungsional, tetapi juga dapat mendukung analisis data yang lebih dalam. Keahlian ini menjadi pondasi penting dalam pengembangan aplikasi berbasis graf, sekaligus menjadi kompetensi yang sangat dibutuhkan dalam praktik profesional di industri teknologi informasi.

#### **2.4.4. Importing Data Fundamentals**

Course Importing Data Fundamentals berfokus pada pemahaman dan

keterampilan teknis dalam membawa data dari berbagai sumber eksternal ke dalam basis data graf Neo4j. Kemampuan ini sangat penting karena dalam praktik nyata, data yang akan dianalisis jarang sekali berada dalam format graf sejak awal. Sebagian besar data biasanya tersimpan dalam bentuk CSV files, basis data relasional, ataupun API eksternal yang perlu diolah sebelum dapat dimanfaatkan secara optimal dalam Neo4j.

<b>01. Importing data into Neo4j</b>	
3 Lessons	
Getting started	
How can I import data into Neo4j?	
Tools	
<b>02. Neo4j Data Importer</b>	
8 Lessons	
Data Importer	
Properties and Types	
Add Movie Nodes	
Unique IDs and Constraints	
Creating Relationships	
Add Directed relationship	
Add Users Ratings	
Data Importer Considerations	
<b>03. Source data considerations</b>	
4 Lessons	
Understanding the source data	
Developing a data model	
Import your CSV file	(Optional)
Congratulations and Next Steps	
<b>04. Course Summary</b>	
Course Summary	
Share your achievement	

Gambar 2. 13. Silabus Importing Data Fundamentals

Dapat dilihat pada gambar di atas bahwa materi pembelajaran pada bagian ini

berfokus pada proses impor data ke dalam Neo4j. Alur pembelajaran dimulai dari pengenalan cara dan alat yang digunakan untuk mengimpor data, kemudian dilanjutkan dengan penggunaan Neo4j Data Importer untuk menambahkan node, properti, ID unik, serta membangun relationships antar entitas, termasuk arah hubungan dan nilai rating dari pengguna. Selanjutnya, penulis juga diarahkan untuk memahami karakteristik data sumber, merancang model data yang tepat, hingga mencoba mengimpor data dari file CSV.

Penulis juga diarahkan untuk memahami cara memetakan data tabular atau relasional menjadi struktur graf yang terdiri dari nodes dan relationships. Contoh-contoh nyata ditampilkan, misalnya bagaimana data pelanggan dan transaksi dari sistem relasional dapat dimodelkan menjadi jaringan entitas yang lebih mudah dianalisis menggunakan Neo4j.

Course ini juga menekankan pada aspek performa. Penulis diajarkan teknik optimasi saat mengimpor data, termasuk penggunaan indexes dan constraints untuk memastikan kecepatan query sekaligus menjaga integritas data. Dengan demikian, proses impor tidak hanya berfungsi memindahkan data, tetapi juga memastikan bahwa data yang dihasilkan benar-benar siap digunakan dalam analisis lebih lanjut.

Sebagai bagian dari pembelajaran, terdapat latihan hands-on lab di mana penulis secara langsung melakukan proses impor data dari file CSV ke dalam Neo4j. Latihan ini memberikan pengalaman praktis yang sangat berguna untuk membangun kepercayaan diri dalam mengelola data pada proyek nyata. Di akhir course, evaluasi berupa kuis memastikan bahwa penulis memahami konsep, metode, serta tantangan umum yang muncul dalam proses impor data.

Dengan menyelesaikan course ini, penulis memiliki kompetensi dalam mengintegrasikan data dari berbagai sumber ke dalam basis data graf. Keahlian ini menjadi keterampilan yang krusial di dunia kerja, karena memastikan bahwa Neo4j dapat digunakan secara maksimal untuk mendukung kebutuhan analisis maupun pengembangan aplikasi berbasis data graf.

#### **2.4.5. Intermediate Cypher Queries**

Course Intermediate Cypher Queries dirancang untuk memperdalam keterampilan penulis dalam menggunakan bahasa query Cypher setelah menguasai dasar-dasarnya pada tahap fundamentals. Jika pada tingkat awal penulis hanya berfokus pada query sederhana untuk menambahkan, menampilkan, memperbarui, dan menghapus data, maka pada tahap menengah ini pembelajaran diarahkan pada teknik query yang lebih kompleks, berlapis, serta efisien dalam mengolah data graf yang berskala besar.

Materi dimulai dengan pengenalan cara melakukan penyaringan data menggunakan pola pencocokan yang lebih maju serta klausa WHERE. Pendekatan ini memungkinkan penulis tidak hanya mencari node atau relationship tertentu, tetapi juga menambahkan kondisi logis agar hasil query menjadi lebih spesifik dan relevan. Setelah itu, pembelajaran berlanjut pada cara mengatur urutan hasil dengan klausa ORDER BY, serta teknik paginasi menggunakan SKIP dan LIMIT untuk mengelola hasil query dalam jumlah besar agar lebih mudah dianalisis.

Selain itu, penulis mempelajari teknik eliminasi duplikasi menggunakan perintah DISTINCT agar hasil query lebih bersih dan akurat. Pada tahap selanjutnya, materi berfokus pada agregasi data, seperti menghitung jumlah entitas

dengan COUNT, mencari nilai rata-rata (AVG), maupun menggunakan fungsi-fungsi agregasi lainnya untuk kebutuhan analisis yang lebih dalam. Penulis juga diperkenalkan dengan cara menghasilkan map projections, yaitu teknik untuk menyusun hasil query dalam bentuk representasi yang lebih ringkas dan mudah digunakan pada aplikasi.

Course ini juga memperkenalkan pengelolaan data berbasis waktu untuk mendukung analisis yang melibatkan dimensi temporal, seperti menghitung selisih waktu atau menyusun data berdasarkan periode tertentu. Topik lanjutan lainnya mencakup variable length traversals, yaitu kemampuan menjelajahi hubungan dengan panjang jalur yang bervariasi. Hal ini sangat bermanfaat ketika penulis ingin menemukan koneksi tidak langsung antar node, misalnya dalam analisis jejaring sosial atau sistem rekomendasi.

Dari sisi pemrosesan data, penulis diperkenalkan pada konsep pipelines sebagai alur berurutan dalam query, serta perintah UNWIND yang berfungsi mengubah sebuah list menjadi banyak baris hasil sehingga memudahkan analisis data terstruktur. Untuk efisiensi, course ini menekankan pentingnya subqueries dalam mengoptimalkan pemakaian memori, sekaligus memperkenalkan cara penggunaan parameters agar query lebih aman, fleksibel, dan dapat digunakan kembali pada berbagai konteks.

Sebagaimana course lainnya, materi ini juga dilengkapi dengan latihan hands-on lab yang memungkinkan penulis langsung mempraktikkan query menengah pada skenario nyata. Contoh kasus mencakup penyusunan sistem rekomendasi berbasis hubungan, eksplorasi jalur terpendek antar node, hingga analisis pola

keterhubungan yang kompleks dalam jaringan besar. Melalui latihan ini, penulis tidak hanya menguasai aspek teknis penulisan query, tetapi juga belajar merancang solusi analitis yang efektif menggunakan Neo4j.

Pada bagian akhir, evaluasi berupa kuis diberikan untuk memastikan bahwa penulis benar-benar memahami teknik-teknik tingkat menengah ini. Dengan menyelesaikan course ini, penulis memperoleh kemampuan yang lebih matang dalam memanfaatkan bahasa Cypher, tidak hanya untuk query sederhana, tetapi juga untuk analisis data graf yang lebih kompleks, efisien, dan siap diterapkan pada konteks profesional maupun penelitian akademis.

#### **2.4.6. Neo4j dengan Python**

Course Using Neo4j with Python berfokus pada integrasi antara basis data graf Neo4j dengan bahasa pemrograman Python. Materi dimulai dengan pengenalan Neo4j Python Driver serta cara melakukan koneksi aman ke database. Penulis kemudian belajar mengeksekusi query Cypher langsung dari kode Python, baik untuk mengambil maupun memanipulasi data.

Dengan course ini, penulis tidak hanya memahami teori integrasi database graf, tetapi juga memperoleh keterampilan praktis untuk menghubungkan Neo4j ke dalam aplikasi berbasis Python, sehingga menambah nilai profesional dalam pengembangan perangkat lunak maupun analisis data.

### **2.5. Tahapan Pelaksanaan**

Tahapan pelaksanaan uji kompetensi Neo4j Certified Professional dilakukan melalui beberapa langkah yang sistematis, mulai dari persiapan, pembelajaran, praktik, hingga pelaksanaan ujian dan evaluasi. Setiap tahap dirancang agar penulis

tidak hanya memahami teori, tetapi juga mampu menerapkannya dalam bentuk praktik langsung.

### 2.5.1. Persiapan

Pada tahap awal, penulis melakukan registrasi pada situs resmi Neo4j dan membuat akun untuk mengikuti jalur sertifikasi. Setelah berhasil melakukan pendaftaran, penulis mendapatkan akses ke dashboard pembelajaran yang berisi daftar kursus, materi, dan informasi terkait ujian sertifikasi.

The screenshot shows a web form titled "Complete your account" with the following fields and content:

- Your Details**
  - Introductory text: "In order to personalise your experience, we would like to know a little more about you."
  - Nickname (Required)**: Input field containing "Ridwan".
  - Given Name (Required)**: Input field containing "Ridwan Yusuf Bahry".
  - Labels: "Let us know how to address you" and "This name will appear on your certificates".
- Bio**: Large empty text area.
- Label: "Your bio will appear you on your public profile".
- Country (Required)**: Dropdown menu showing "Indonesia".
- Label: "Where do you live?".
- Job Role (Required)**: Input field containing "College Student".
- Company (Required)**: Input field containing "Universitas Teknologi Digital Indonesia".
- Example text: "Eg. Data Scientist, Software Developer".
- Checkbox: "Unsubscribe from GraphAcademy emails related to your enrollment" (unchecked).

Gambar 2. 14. Pendaftaran Awal

Pada bagian awal penulis mendaftarkan diri dan mengisi data lengkap penulis pada kolom yang disediakan seperti pada gambar di atas. Kemudian baru bisa masuk ke halaman materi pelajaran

Setelah tahap ini penulis bisa mengetahui bagaimana alur sertifikasi, silabus yang akan dipelajari, serta standar kelulusan yang harus dicapai.

### **2.5.2. Pembelajaran**

Tahap pembelajaran dilaksanakan secara daring melalui kursus yang disediakan oleh Neo4j. Materi pembelajaran disusun dalam bentuk learning path yang terdiri atas beberapa topik, antara lain:

1. Pengenalan dasar Neo4j.
2. Pengenalan dasar Cypher.
3. Graph data modeling.
4. Dasar Import Data.
5. Advanced Cypher dan optimasi query.
6. Menggunakan Neo4j di python.

Selama proses pembelajaran, penulis dapat mengakses modul teori, video pembelajaran, serta latihan singkat untuk memperkuat pemahaman.

### **2.5.3. Praktik (Hands-on Lab)**

Setelah mempelajari teori, penulis diwajibkan untuk mengikuti hands-on lab menggunakan Neo4j Browser atau Neo4j Sandbox/Aura. Pada tahap ini, penulis mempraktikkan berbagai perintah query Cypher, seperti:

1. Membuat node dan relationship.
2. Menambahkan properti pada data.
3. Menjalankan query dengan filter dan agregasi.
4. Menganalisis pola hubungan antar data melalui traversal graf.

Tahap praktik ini memberikan pengalaman langsung sehingga penulis dapat mengaplikasikan konsep yang telah dipelajari sebelumnya.

#### **2.5.4. Ujian Teori dan Praktik**

Setelah menyelesaikan tahapan pembelajaran dan praktik, penulis mengikuti ujian sertifikasi. Ujian dilaksanakan secara daring dengan format pilihan ganda untuk menguji pemahaman teori, serta studi kasus praktis untuk menilai kemampuan penggunaan Cypher dalam menyelesaikan masalah.

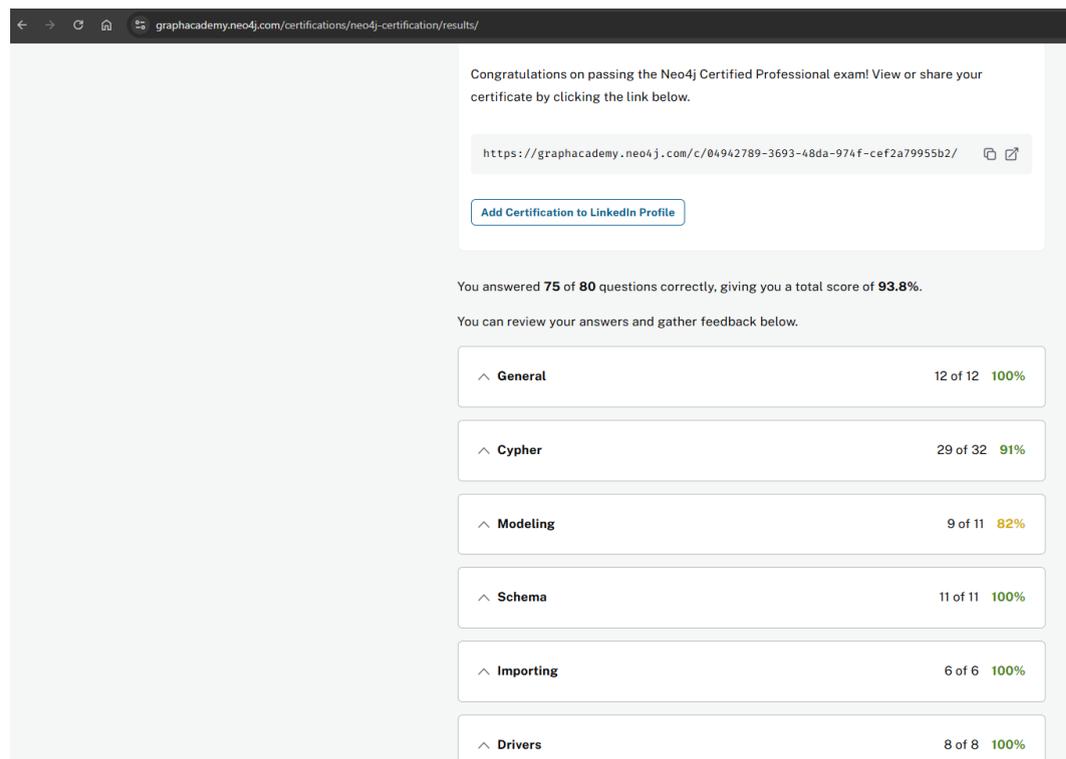
Materi ujian mencakup pemahaman tentang graph database, desain model data, serta keterampilan dalam menulis dan mengoptimalkan query. Ujian ini terdiri dari 80 soal dan diberi waktu mengerjakan 60 menit sehingga menuntut ketelitian dan kecepatan dalam menjawab soal.

Untuk memulai ujian maka penulis menekan tombol “Take Certification” pada halaman neo4j certification. Ketika menekan tombol tersebut, maka waktu 60 menit sudah dimulai. Penulis disuguhkan soal soal terkait neo4j. Namun ketika ujian tidak disebutkan materi tersebut dari bidang yang mana. Hanya ketika sudah selesai ujian dan hasilnya keluar, penulis dapat melihat bahwa ujian terbagi menjadi 6 bidang kompetensi yang terdiri dari: General, Cypher, Modeling, Schema, Importing, Drivers. General artinya bidang secara umum dan dasar terkait neo4j. Bagian cypher lebih membahas khusus terkait query yang ada. Modeling terkait pemahaman tentang model graf itu sendiri. Schema terkait pelajaran skema yang digunakan oleh neo4j. Bagian terakhir, drivers terkait driver yang digunakan oleh neo4j. Soal yang ada terkadang berupa benar atau salah, terkadang berupa pilih salah satu jawaban, terkadang juga pilih seluruh jawaban yang dianggap benar, dan terkadang diminta mengisi query cypher yang tepat pada bagian kosong yang telah disediakan.

Beberapa catatan terkait pengambilan ujian adalah, nilai minimum yang dibutuhkan untuk lulus adalah 80%. Karena ada 80 soal, artinya minimal 64 soal yang benar agar dinyatakan lulus. Jika nilai kurang dari itu, maka dinyatakan gagal, peserta bisa mengulang ujian kembali setelah minimal 24 jam masa tunggu sambil kembali mengulang mempelajari materi yang telah disediakan.

### 2.5.5. Hasil dan Evaluasi

Setelah menyelesaikan ujian, sistem secara otomatis menampilkan hasil evaluasi. Penulis dinyatakan lulus dengan skor yang memenuhi standar kelulusan dan berhak memperoleh sertifikat Neo4j Certified Professional. Sertifikat ini diterbitkan langsung oleh Neo4j dan dapat diverifikasi secara daring.



Congratulations on passing the Neo4j Certified Professional exam! View or share your certificate by clicking the link below.

<https://graphacademy.neo4j.com/c/04942789-3693-48da-974f-cef2a79955b2/>

[Add Certification to LinkedIn Profile](#)

You answered **75 of 80** questions correctly, giving you a total score of **93.8%**.

You can review your answers and gather feedback below.

Category	Score	Percentage
General	12 of 12	100%
Cypher	29 of 32	91%
Modeling	9 of 11	82%
Schema	11 of 11	100%
Importing	6 of 6	100%
Drivers	8 of 8	100%

Gambar 2. 15. Hasil Ujian Sertifikasi

Pada gambar di atas menunjukkan hasil dari ujian yang dilakukan oleh penulis. Dimana dari 80 soal, berhasil menjawab dengan benar 75 soal. Sehingga penulis mendapat nilai 93.8.

Dengan selesainya ujian dan memperoleh sertifikat, penulis telah berhasil menyelesaikan seluruh rangkaian tahapan uji kompetensi, mulai dari persiapan hingga evaluasi akhir.