

BAB II

PELAKSANAAN UJIAN KOMPETENSI

2.1. Pendahuluan

Di tengah pesatnya transformasi digital, data telah menjadi elemen kunci dalam mendukung proses pengambilan keputusan, pengembangan sistem informasi, serta analisis bisnis yang berbasis bukti. Khususnya, jenis data yang saling terhubung atau *connected data* menjadi semakin relevan dalam berbagai bidang seperti media sosial, sistem rekomendasi, manajemen jaringan, dan analisis keuangan. Namun, pendekatan basis data relasional konvensional sering kali kurang efisien dalam menangani kompleksitas hubungan antar data tersebut.

Untuk mengatasi tantangan tersebut, teknologi basis data graf (graph database) hadir sebagai solusi yang lebih alami dan optimal dalam memodelkan serta menganalisis relasi antar entitas. Graph database memungkinkan visualisasi dan pencarian hubungan yang lebih cepat dan intuitif, menjadikannya sangat berguna dalam konteks data yang kompleks dan dinamis.

Salah satu platform graph database yang paling dikenal dan banyak digunakan secara global adalah Neo4j. Dengan arsitektur berbasis graf dan bahasa kueri Cypher yang mudah dipahami, Neo4j mendukung pengguna dalam membangun sistem yang membutuhkan pencarian relasi data yang cepat dan akurat. Teknologi ini telah digunakan oleh berbagai perusahaan besar di dunia dalam mendukung operasional mereka.

Sebagai bentuk penguatan kompetensi di bidang teknologi data modern, penulis mengikuti program Neo4j Certified Professional, yang disediakan secara resmi dan daring oleh Neo4j, Inc. melalui platform pembelajaran Neo4j GraphAcademy. Sertifikasi ini menguji penguasaan konsep dasar graph database, pemodelan data, serta penggunaan bahasa kueri Cypher dalam konteks aplikasi nyata.

Keikutsertaan penulis dalam program sertifikasi ini tidak hanya sebagai upaya pengembangan keahlian pribadi, tetapi juga menjadi bagian penting dalam penyusunan tugas akhir. Sertifikasi ini merepresentasikan integrasi antara pembelajaran akademik dan penerapan praktis, serta menjadi landasan yang relevan dengan perkembangan industri teknologi informasi saat ini.

2.2. Tujuan Sertifikasi

Tujuan dari mengikuti sertifikasi Neo4j Certified Professional adalah untuk:

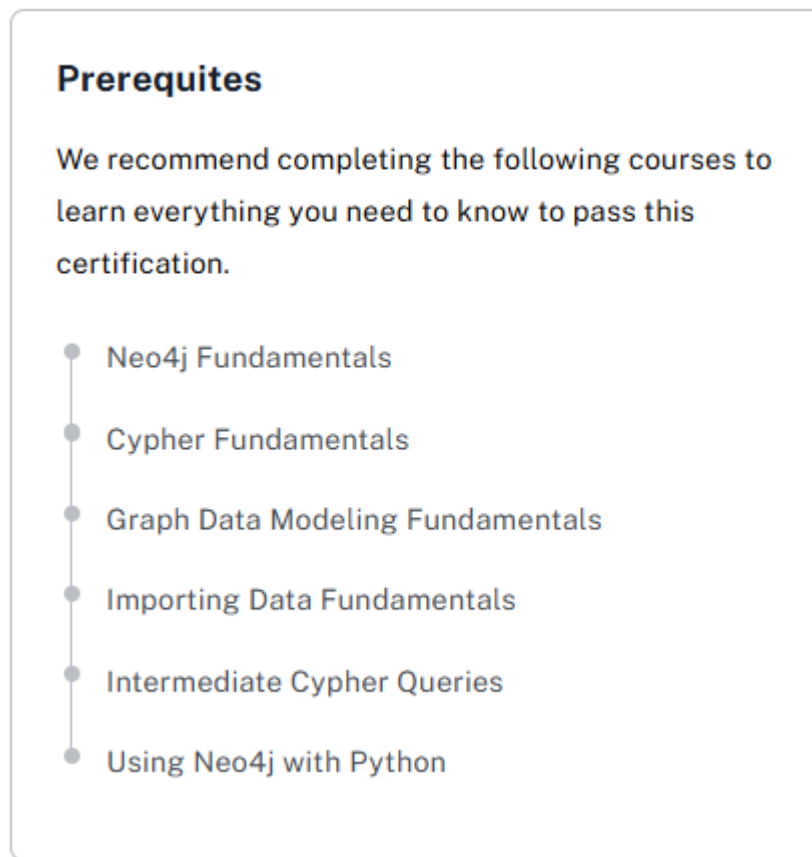
1. Memvalidasi kemampuan dalam menggunakan Neo4j dan memahami prinsip dasar basis data graf.
2. Meningkatkan kompetensi teknis dalam menulis dan mengoptimalkan kueri menggunakan bahasa Cypher.
3. Memahami cara membangun dan mengelola skema graf yang kompleks untuk berbagai kebutuhan aplikasi.
4. Mendapatkan pengakuan resmi dari Neo4j sebagai profesional yang kompeten dalam teknologi graf.
5. Menjadi bekal untuk mengembangkan sistem informasi berbasis graf sebagai bagian dari topik tugas akhir.

2.3. Manfaat Sertifikasi

Sertifikasi Neo4j Certified Professional memberikan sejumlah manfaat, baik secara akademik maupun profesional, antara lain:

1. Pengakuan Kompetensi Global: Sertifikat ini diakui secara internasional sebagai bukti keterampilan dalam basis data graf dan Cypher.
2. Mendukung Penelitian Akademik: Sertifikasi mendukung pelaksanaan skripsi yang menggunakan model graf atau teknologi basis data non-relasional.
3. Peningkatan Portofolio Profesional: Sertifikat ini memperkuat CV dan portofolio dalam menghadapi persaingan di dunia kerja.
4. Akses ke Komunitas Neo4j: Peserta sertifikasi menjadi bagian dari komunitas profesional Neo4j yang luas dan aktif, sehingga membuka peluang kolaborasi dan pengembangan karier.
5. Kesiapan untuk Proyek Nyata: Materi yang dipelajari relevan dengan praktik dunia nyata, termasuk pengembangan aplikasi dan analisis data.

2.4. Silabus dan Materi yang Dipelajari



Gambar 2. 1 Dokumentasi Silabus Sertifikasi Neo4j Certified Professional

Pelatihan dan sertifikasi Neo4j Certified Professional mencakup pemahaman dasar hingga lanjutan mengenai konsep graph database, penggunaan bahasa query Cypher, serta praktik terbaik dalam pemodelan dan optimasi data berbasis graf. Silabus lengkap yang mencakup seluruh topik pembelajaran dalam sertifikasi ini dapat dilihat pada Gambar 2.1.

Materi pembelajaran untuk sertifikasi Neo4j Certified Professional mencakup topik-topik esensial berikut:

1. Dasar-Dasar Database Graph

a. Konsep graf: node, relasi, dan property

Database graf menggunakan tiga elemen utama: node untuk mewakili entitas (seperti orang atau produk), relationship untuk menunjukkan koneksi antar node, dan property sebagai atribut tambahan yang menyimpan informasi pada node atau hubungan.

b. Arsitektur dasar Neo4j

Neo4j dirancang dengan arsitektur native graph, di mana hubungan antar node tersimpan secara langsung di level storage. Hal ini memungkinkan traversal data sangat cepat dibandingkan sistem lain yang menggunakan pendekatan relasional.

c. Keunggulan basis data graf dibandingkan relasional

Graph database lebih unggul dalam menangani data yang kompleks dan saling terhubung. Tidak perlu JOIN yang rumit seperti di relational database, sehingga performa tetap stabil walau data dan hubungan bertambah banyak.

2. Bahasa Query Cypher

a. Struktur dasar kueri

Cypher adalah bahasa query khusus Neo4j. Perintah utamanya antara lain:

- MATCH (mencari pola),
- RETURN (menampilkan hasil),
- WHERE (penyaring kondisi),

- CREATE (membuat node/relasi),
- MERGE (cari atau buat),
- SET (ubah data),
- DELETE (hapus node/relasi).

b. Menggunakan filter, logika, dan ekspresi dalam Cypher

Cypher mendukung berbagai ekspresi logika (AND, OR, NOT) dan fungsi bawaan untuk menyaring dan memanipulasi data, sehingga hasil query bisa lebih spesifik dan dinamis.

c. Operasi kompleks seperti path finding dan pattern matching

Neo4j mampu melakukan pencarian jalur terpendek, analisis jalur tertentu, serta pencocokan pola hubungan (pattern matching), yang sangat berguna untuk analisis jaringan atau rekomendasi.

3. Modeling Data di Neo4j

a. Merancang skema graf yang efisien

Pemodelan graf harus mencerminkan struktur dunia nyata dengan jelas. Penggunaan label, properti, dan relasi yang tepat akan mempermudah query dan pengelolaan data.

b. Praktik terbaik dalam pemodelan node dan relasi

Gunakan label bermakna, relasi yang deskriptif, dan hindari membuat node yang terlalu terhubung tanpa klasifikasi. Relasi harus dimodelkan secara eksplisit, bukan hanya sebagai atribut.

c. Penanganan data redundan dan optimalisasi performa

Kurangi pengulangan data yang tidak perlu, dan gunakan pendekatan graf untuk menyimpan data yang bisa dirujuk lewat hubungan antar node. Hal ini membantu menjaga konsistensi dan efisiensi.

4. Index dan Constraint

a. Pembuatan dan penggunaan index

Index digunakan untuk mempercepat pencarian node berdasarkan properti tertentu. Misalnya, mencari User berdasarkan email bisa lebih cepat jika ada index pada properti email.

b. Definisi constraint untuk menjaga integritas data dan meningkatkan performa

Constraint memastikan data tetap valid dan konsisten, misalnya dengan mencegah duplikasi data (UNIQUE) atau mewajibkan properti tertentu ada (NOT NULL), serta membantu Neo4j dalam optimasi query.

5. Optimasi dan Praktik Penggunaan di Dunia Nyata

Neo4j telah banyak digunakan di berbagai industri karena kemampuannya dalam menangani data yang kompleks dan saling terhubung. Salah satu penerapan utamanya adalah pada sistem rekomendasi, di mana pola hubungan antara pengguna dan produk dianalisis untuk memberikan rekomendasi yang relevan. Di bidang jejaring sosial, Neo4j digunakan untuk memetakan dan menganalisis hubungan antar pengguna, seperti pertemanan, interaksi, atau pengaruh sosial. Selain itu, dalam deteksi penipuan (fraud detection), Neo4j mampu mengidentifikasi pola transaksi yang mencurigakan dengan melacak koneksi tidak langsung antar entitas.

Semua penerapan ini memperlihatkan keunggulan basis data graf dalam memproses dan menganalisis data yang bersifat dinamis serta sangat terhubung.

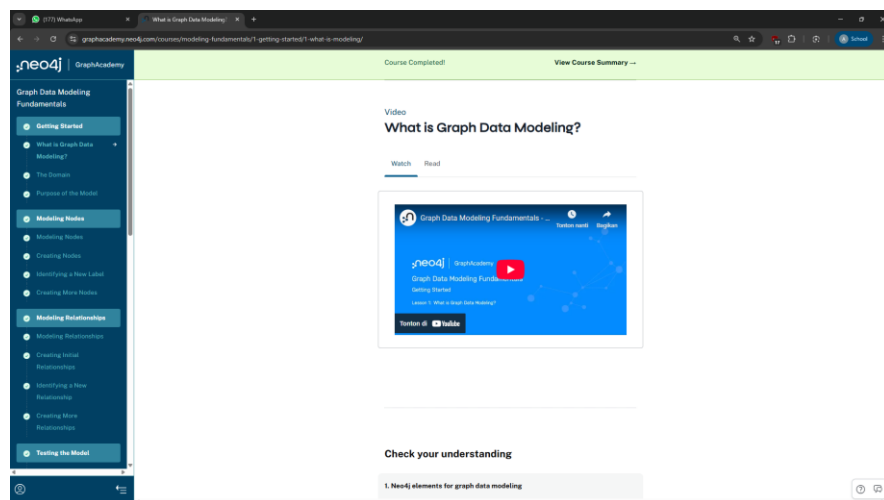
2.5. Tahapan Pelaksanaan Ujian

Proses untuk memperoleh sertifikasi *Neo4j Certified Professional* dilakukan dalam beberapa tahap sebagai berikut:

2.5.1. Persiapan Belajar Mandiri

a. Mengikuti pelatihan daring gratis dari Neo4j Academy

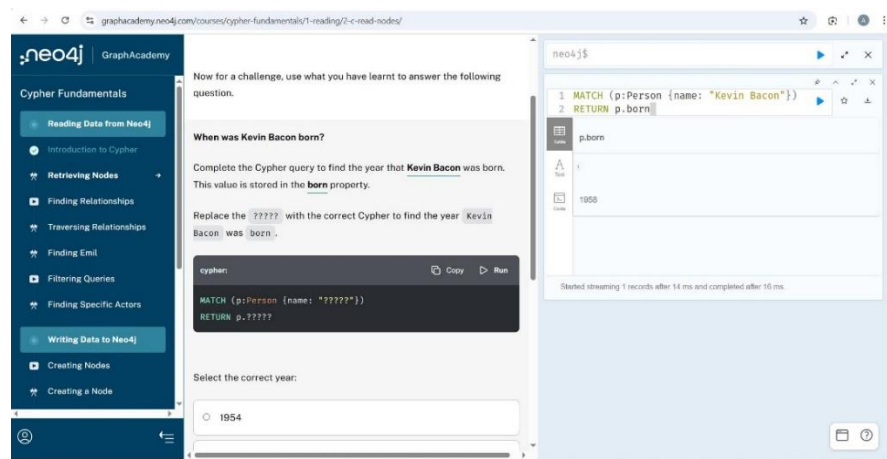
Tahap pertama dalam proses sertifikasi Neo4j Certified Professional adalah melakukan belajar mandiri untuk memahami konsep dasar dan praktik penggunaan Neo4j. Salah satu sumber utama yang sangat direkomendasikan adalah pelatihan daring gratis dari Neo4j Academy yang dapat diakses melalui <https://graphacademy.neo4j.com>. Platform ini menyediakan berbagai modul pembelajaran yang terstruktur dan interaktif, mulai dari pengenalan hingga tingkat lanjutan.



Gambar 2. 2 Dokumentasi Belajar Mandiri Graph Data Modeling Fundamentals

b. Membaca dokumentasi resmi Neo4j dan tutorial penggunaan Cypher

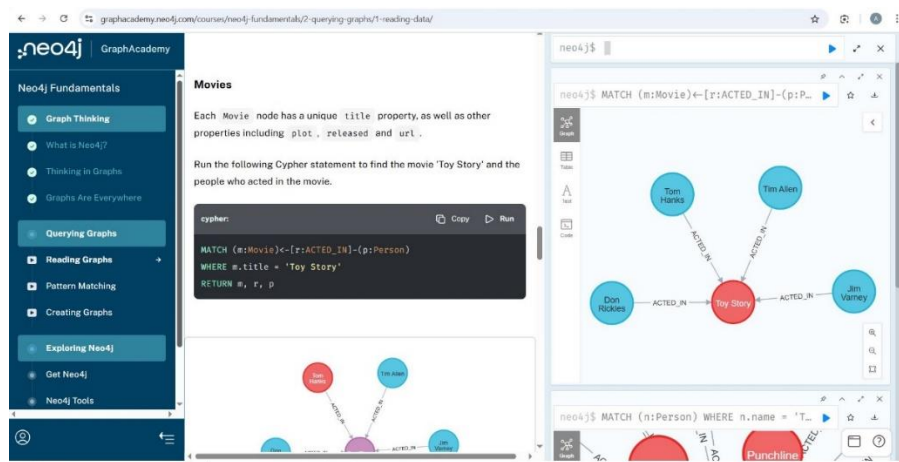
Selain itu, peserta juga disarankan untuk membaca dokumentasi resmi Neo4j guna memperdalam pemahaman teknis, khususnya tentang struktur graf dan penggunaan bahasa query Cypher. Dokumentasi ini mencakup referensi sintaks, penjelasan fitur, serta contoh-contoh penggunaan yang aplikatif.



Gambar 2. 3 Dokumentasi Penggunaan Cypher

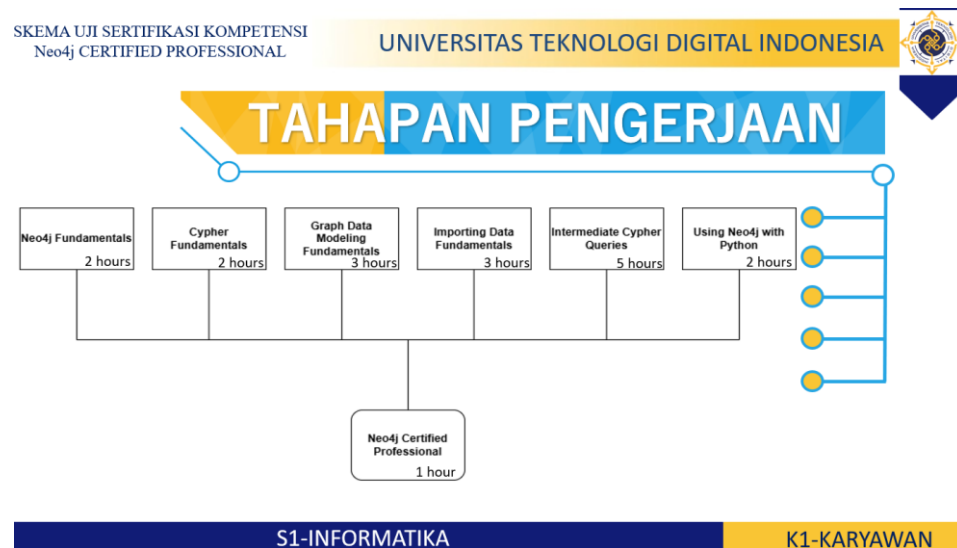
c. Praktik langsung melalui *hands-on labs* yang disediakan dalam pelatihan (sandbox)

Sebagai pelengkap, peserta dapat berlatih langsung menggunakan hands-on labs yang tersedia dalam pelatihan, seperti melalui sandbox atau lingkungan latihan terintegrasi. Dengan praktik langsung ini, peserta dapat menguji pengetahuan mereka secara real-time dan membangun kepercayaan diri sebelum mengikuti ujian sertifikasi.



Gambar 2. 4 Dokumentasi implementasi langsung pada sand box

2.5.2. Simulasi dan Latihan Soal pada Pengerjaan Modul Neo4j

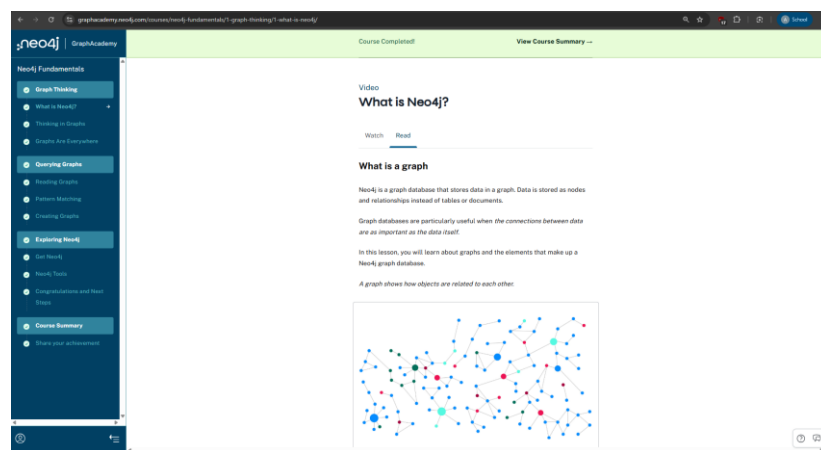


Gambar 2. 5 Tahapan Pengerjaan Sertifikasi Neo4j

Dalam menempuh proses uji sertifikasi kompetensi Neo4j ini, peserta diharuskan menempuh 6 modul yang wajib diselesaikan agar dapat menuju ke proses selanjutnya yaitu mendapatkan akses untuk mengikuti ujian Neo4j Certified Professional. Berikut penjelasan mendetail dari 6 modul yang direkomendasikan sebagai prasyarat sebelum mengikuti ujian Neo4j Certified Professional:

1. Neo4j Fundamentals

Modul pertama adalah Neo4j Fundamentals, yang memperkenalkan konsep dasar dari graph database dan bagaimana Neo4j bekerja sebagai platform penyimpanan dan analisis data terhubung. Materi mencakup pengenalan terhadap komponen utama seperti node, relationship, dan properti, serta perbandingan dengan basis data relasional dan penerapan Neo4j di dunia nyata. Modul ini cocok untuk pemula yang belum familiar dengan konsep graph.



Gambar 2. 6 Dokumentasi Pelatihan Neo4j Fundamentals

Terdapat 5 poin materi penting yang dapat kita pelajari pada modul ini, yaitu:

a. Komponen yang Mengatur Penyimpanan Data dalam Neo4j

Neo4j menggunakan arsitektur native graph yang dirancang khusus untuk menyimpan dan mengakses data dalam bentuk graf. Komponen penyimpanan utama terdiri dari node store (untuk menyimpan entitas), relationship store (untuk menyimpan relasi antar node), dan property store (untuk menyimpan atribut dari node dan relasi). Dengan desain ini, traversal antar node menjadi sangat cepat karena relasi disimpan secara langsung di level penyimpanan, bukan melalui indeks seperti pada basis data relasional.

b. Masalah Notasi $O(n)$ Terkait dengan Basis Data Relasional

Pada database relasional, pencarian hubungan antar data sering kali melibatkan operasi JOIN yang kompleks, dan waktu prosesnya dapat meningkat secara linier ($O(n)$) atau bahkan lebih buruk seiring bertambahnya data. Hal ini menyebabkan penurunan performa pada skala besar. Neo4j mengatasi masalah ini dengan menyimpan relasi sebagai bagian langsung dari data, sehingga pencarian antar entitas tidak bergantung pada operasi JOIN dan memiliki performa yang lebih stabil.

c. Label Node (Label Simpul) yang Digunakan pada Grafik

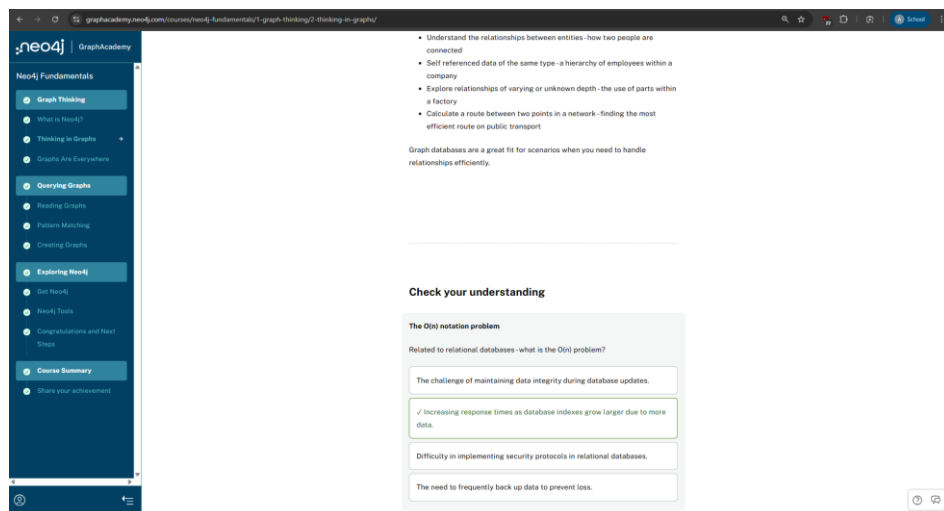
Label pada node digunakan untuk mengelompokkan dan mengidentifikasi jenis entitas dalam graph. Misalnya, node dengan label :Person menunjukkan bahwa node tersebut mewakili seseorang. Label ini sangat berguna untuk menyaring dan menjalankan query secara lebih spesifik serta meningkatkan performa pencarian, karena sistem hanya akan memproses node yang relevan dengan label yang ditentukan.

d. Fitur yang Disertakan dalam Edisi Komunitas Neo4j Sumber Terbuka

Neo4j menyediakan Community Edition sebagai versi sumber terbuka yang bisa digunakan secara gratis. Meskipun fitur-fitur seperti keamanan tingkat lanjut, monitoring, dan clustering tidak tersedia, edisi ini tetap menyediakan semua fitur utama untuk membuat, mengelola, dan menjalankan database graf termasuk kemampuan query dengan Cypher, penggunaan indeks dan constraint dasar, serta import data.

e. Duplicate Constraint, Importir Data

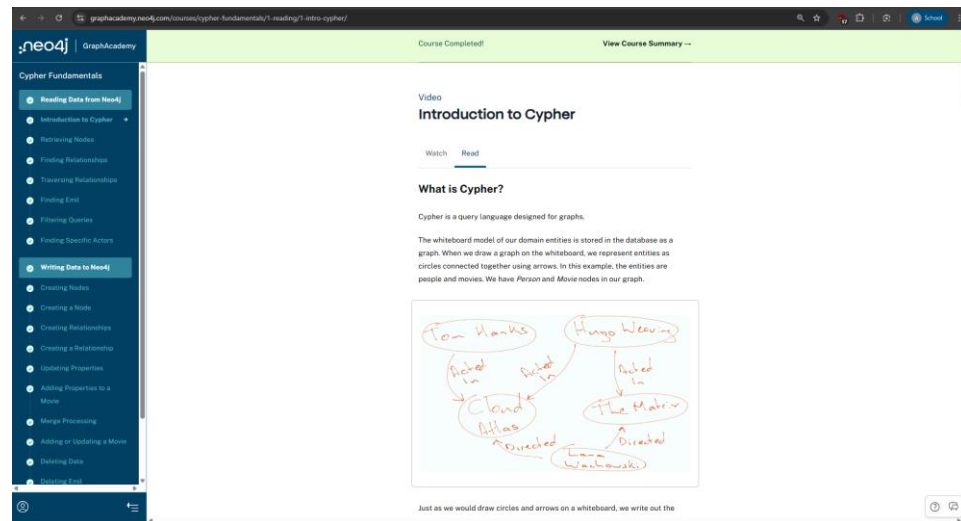
Duplicate constraint digunakan untuk memastikan bahwa tidak ada dua node yang memiliki nilai properti yang sama pada kolom tertentu, misalnya email atau ID unik. Ini berguna untuk menjaga integritas data dan mencegah duplikasi. Sementara itu, importir data (seperti LOAD CSV) memungkinkan pengguna membawa data dari file eksternal (misalnya CSV atau JSON) ke dalam Neo4j dengan mudah, sehingga proses migrasi atau integrasi data dapat dilakukan dengan cepat dan efisien.



Gambar 2. 7 Dokumentasi Latihan Soal Neo4j Fundamentals

2. Cypher Fundamentals

Selanjutnya, Cypher Fundamentals membahas bahasa query Cypher, yaitu bahasa utama yang digunakan untuk berinteraksi dengan database Neo4j. Modul ini mencakup sintaks dasar seperti MATCH, RETURN, CREATE, WHERE, MERGE, dan DELETE, serta cara menulis query untuk membaca, menambahkan, dan memodifikasi data. Pemahaman Cypher sangat penting karena seluruh interaksi dengan Neo4j dilakukan melalui bahasa ini.



Gambar 2. 8 Dokumentasi Pelatihan Cypher Fundamentals

Terdapat 25 poin materi penting yang dapat kita pelajari pada modul ini, yaitu:

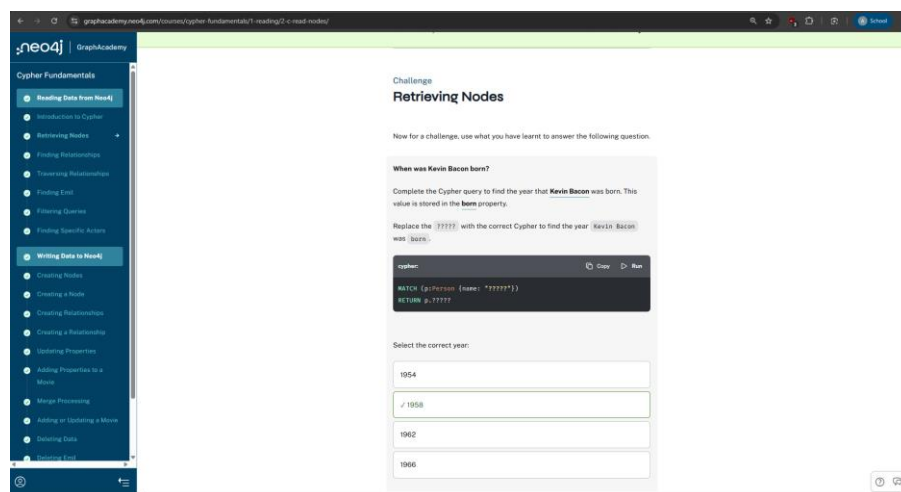
Modul Cypher Fundamentals				
Klausul Cypher untuk membaca data Neo4j	Filtering. Kata kunci Cypher untuk menyaring node	Kata Kunci Cypher untuk mengambil properti	Query Cypher untuk menampilkan informasi	Menentukan arah hubungan dengan query Cypher
Kueri menggunakan pola. Klausul MATCH	Penggunaan hubungan DIRECTED	Query Cypher untuk memfilter data yang akan ditampilkan	Memfilter nilai dalam daftar	Klausu MATCH untuk mengambil data tertentu
Klausu WHERE untuk memfilter data	Membuat node dengan kata kunci Cypher	Penggunaan MERGE untuk membuat node	Membuat simpul label untuk node	Klausu untuk membuat hubungan pada Cypher
Menggunakan MERGE untuk hubungan	Buat hubungan antara dua node yang ada	Buat hubungan ACTED_IN antara 2 data	Membuat, menambahkan, menghapus properti	Cara menambahkan property ke node atau hubungan
Menggunakan klausa SET untuk menambahkan properti	Pemrosesan MERGE saat tidak ada simpul dalam grafik	Update query Cypher sesuai data yang diinginkan	Penggunaan klausa ON CREATE & ON MATCH	Hapus simpul yang ada dalam grafik
S1-INFORMATIKA			K1-KARYAWAN	

Gambar 2. 9 Materi pada Modul Cypher Fundamentals

Modul ini dimulai dengan pemahaman tentang klausul MATCH yang digunakan untuk membaca data dari Neo4j dengan pola tertentu, diikuti oleh WHERE yang berfungsi untuk memfilter data hasil pencarian. Untuk membuat node atau relasi, digunakan MERGE dan CREATE, sedangkan SET digunakan untuk menambahkan properti pada node atau relasi yang sudah ada. Filter tambahan dapat dilakukan melalui penggunaan hubungan DIRECTED dan logika dalam WHERE, memungkinkan kontrol lebih detail terhadap arah relasi dan isi data.

Cypher juga mendukung pengambilan informasi melalui kata kunci properti, yang dapat disesuaikan dengan filter untuk hanya menampilkan data yang relevan. Label digunakan untuk memberi identitas pada node, sementara relasi seperti `ACTED_IN` digunakan untuk menghubungkan node yang sudah ada. Untuk hubungan dua arah atau relasi spesifik antar node, Cypher menyediakan cara mengatur arah hubungan dengan simbol panah ($\rightarrow$ atau $\leftarrow$) dan mendukung penggunaan `ON CREATE` dan `ON MATCH` dalam `MERGE` untuk menentukan aksi sesuai kondisi node.

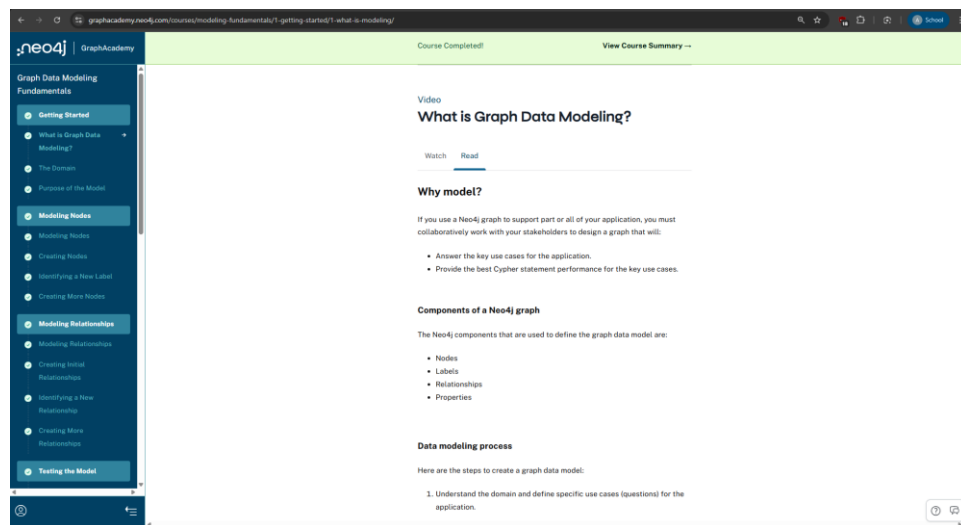
Selain membuat dan membaca, Cypher juga memungkinkan untuk memperbarui atau menghapus data, termasuk menghapus node dan menyesuaikan query agar sesuai dengan kebutuhan data saat ini. Dengan semua kemampuan tersebut, pengguna bisa melakukan manipulasi data secara menyeluruh, mulai dari pencarian, pembuatan, pembaruan, hingga penghapusan data dalam bentuk graf, secara efisien dan terstruktur.



Gambar 2. 10 Dokumentasi Latihan Soal Cypher Fundamentals

3. Graph Data Modeling Fundamentals

Modul ketiga, Graph Data Modeling Fundamentals, mengajarkan cara memodelkan data dunia nyata ke dalam struktur graf yang efisien. Peserta belajar bagaimana merancang node, relasi, dan properti dengan baik, menggunakan label dan arah hubungan yang tepat, serta menghindari kesalahan umum dalam pemodelan. Modul ini juga menekankan pentingnya representasi relasi yang akurat agar sistem tetap mudah dikelola dan performa terjaga.



Gambar 2. 11 Dokumentasi Pelatihan Graph Data Modeling Fundamentals

Terdapat 33 poin materi penting yang dapat kita pelajari pada modul ini, yaitu:

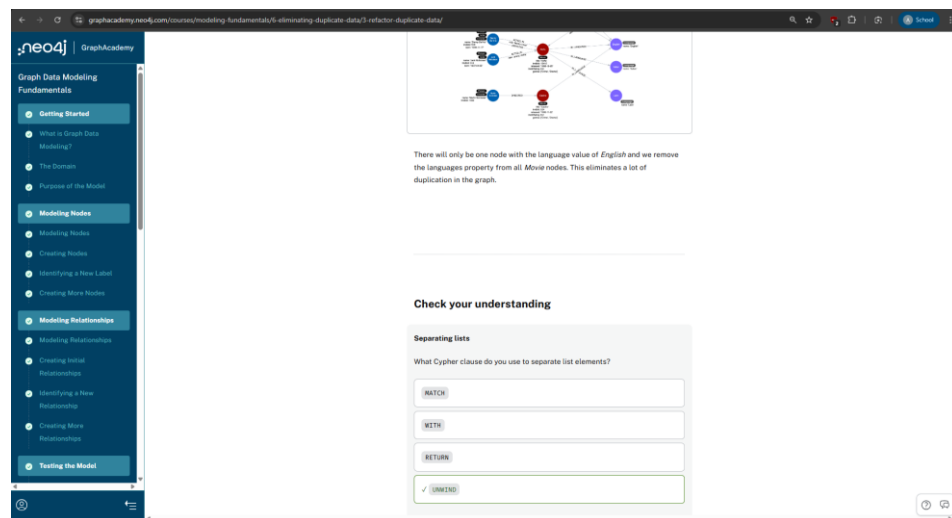
Modul Graph Data Modeling Fundamentals					
Elemen Neo4j untuk pemodelan data grafik	Pemodelan berulang	Memahami domain	Tujuan model	Menentukan label node	Menentukan properti node
Penulisan Kode Cypher yang baik dan benar	Menambahkan label baru ke model	Membuat lebih banyak node	Penggunaan elemen kasus pada hub. data	Koneksi antar entitas	Menentukan hubungan
Menambah hubungan baru ke model	Properti Hubungan	Menciptakan lebih banyak hubungan	Hasil pengujian	Refaktor pada model data grafik dan grafik	Menambah label ke node
Jumlah label maksimum pada node	Mengubah grafik dengan Cypher	Langkah setelah refactoring	Menambah label yang memiliki hubungan	Hal yang dihindari saat membuat model grafik	Menghilangkan duplikasi untuk apa?
Memisahkan elemen daftar	Menambah genre nodes dalam 2 bagian	Melakukan refactor data kompleks	Mengapa mengkhususkan hub.?	Menciptakan hubungan dinamis	Verifikasi pada relasi dan refactoring
Modifikasi query menjadi query untuk membuat relasi khusus berdasarkan data yang diinginkan		Mengapa melakukan refactoring untuk membuat simpul perantara?		Refactoring dan menambahkan node Role berdasarkan properti role dari relasi ACTED_IN	
S1-INFORMATIKA				K1-KARYAWAN	

Gambar 2. 12 Materi pada Modul Graph Data Modeling Fundamentals

Modul ini membahas prinsip dasar pemodelan data graf di Neo4j. Dimulai dari pentingnya menulis kode Cypher yang baik dan benar, termasuk dalam menambahkan hubungan dan label baru ke model graf. Setiap entitas harus memiliki label node yang sesuai, dan hubungan antar entitas ditentukan melalui properti hubungan. Dalam satu graf, jumlah label maksimum juga perlu dipertimbangkan agar struktur tetap efisien dan tidak membingungkan. Elemen-elemen seperti daftar, genre, dan role harus dikelompokkan dengan tepat agar data mudah ditelusuri.

Pada tahap pemodelan berulang, kita mungkin perlu menambahkan lebih banyak node dan hubungan saat model berkembang. Misalnya, menambah label baru atau memecah satu entitas menjadi dua bagian, seperti memisahkan genre nodes. Proses ini juga mencakup refactor grafik menggunakan Cypher untuk menyesuaikan dengan kebutuhan baru. Tujuan dari semua ini adalah untuk mendapatkan model graf yang sesuai dengan domain masalah, menghasilkan data yang mudah diuji, dan mengkhususkan hubungan untuk analisis tertentu.

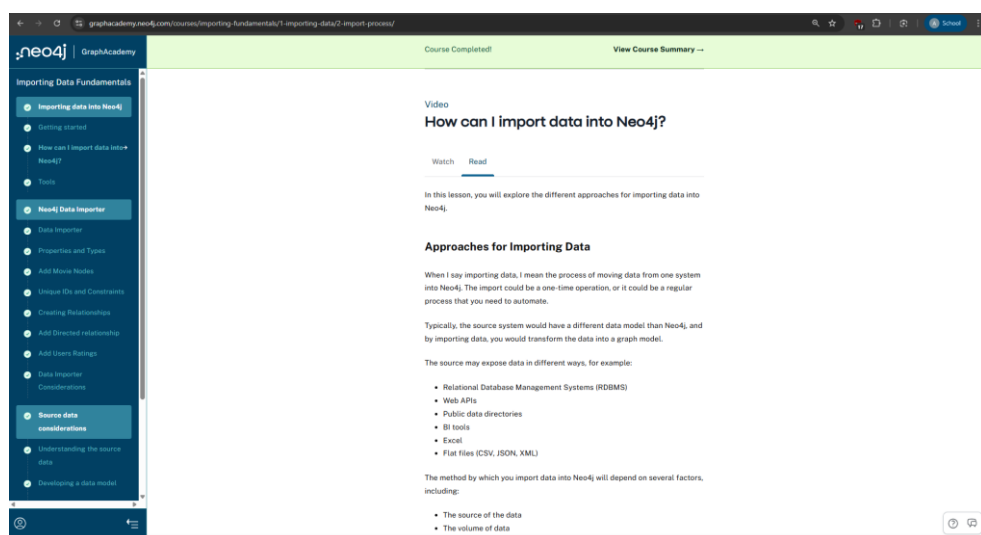
Saat menentukan model, penting memahami koneksi antar entitas, menghindari label yang tidak relevan, serta menyesuaikan properti agar tidak terjadi duplikasi. Penggunaan label juga harus disesuaikan dengan relasi dinamis, dan kadang diperlukan verifikasi terhadap struktur hubungan seperti `ACTED_IN`. Refactoring digunakan untuk memperbaiki atau menyederhanakan model, misalnya dengan menambahkan node Role berdasarkan properti relasi, atau membuat simpul perantara untuk memisahkan relasi kompleks agar lebih mudah dikelola dan dianalisis.



Gambar 2. 13 Dokumentasi Latihan Soal Graph Data Modeling Fundamentals

4. Importing Data Fundamentals

Kemudian ada Importing Data Fundamentals, yang membahas cara mengimpor data dari sumber eksternal seperti file CSV ke dalam Neo4j. Modul ini mengajarkan penggunaan perintah LOAD CSV, pemetaan kolom ke struktur graf, serta tips menangani data besar agar proses impor berjalan lancar. Modul ini penting bagi peserta yang ingin membangun graph dari data yang sudah ada.



Gambar 2. 14 Dokumentasi Pelatihan Importing Data Fundamentals

Terdapat 8 poin materi penting yang dapat kita pelajari pada modul ini, yaitu:

a. Penggunaan Jalur ETL untuk Mengimpor Data ke Neo4j

ETL (Extract, Transform, Load) adalah proses standar untuk mengimpor data ke Neo4j. Data diekstrak dari sumber eksternal seperti file CSV, diubah strukturnya agar sesuai dengan model graf (transform), lalu dimuat (load) ke dalam database menggunakan perintah seperti `LOAD CSV`. Proses ini penting untuk menjaga kesesuaian dan efisiensi data dalam struktur graf.

b. Membuat Node dengan Data Importer

Neo4j menyediakan Data Importer Tool yang memudahkan pengguna membuat node secara visual dari file data. Pengguna dapat mengimpor file seperti CSV, memetakan kolom menjadi properti node, dan menentukan label yang sesuai. Ini mempercepat proses pemodelan tanpa harus menulis query Cypher secara manual.

c. Jumlah Properti untuk Setiap Node saat Menggunakan Pengimpor Data

Saat mengimpor data, setiap node dapat memiliki banyak properti (atribut) tergantung pada isi file. Namun, penting untuk menyesuaikan jumlah properti agar tidak berlebihan dan tetap relevan dengan konteks penggunaannya, sehingga performa dan keterbacaan data tetap terjaga.

d. Menetapkan ID Unik dan Fungsinya

Setiap node sebaiknya memiliki ID unik (misalnya `user_id`, `product_id`) untuk menghindari duplikasi dan menjaga integritas data. ID ini berfungsi sebagai pengenalan utama saat melakukan merge atau pembaruan data, serta membantu dalam pencarian dan pemetaan relasi antar entitas.

e. Properti Hubungan

Tidak hanya node, relasi antar node juga bisa memiliki properti, misalnya tanggal transaksi, rating, atau durasi. Properti ini memberi makna tambahan pada relasi dan dapat digunakan dalam query untuk analisis lebih dalam, seperti mencari relasi berdasarkan nilai atau waktu tertentu.

f. Importir Data

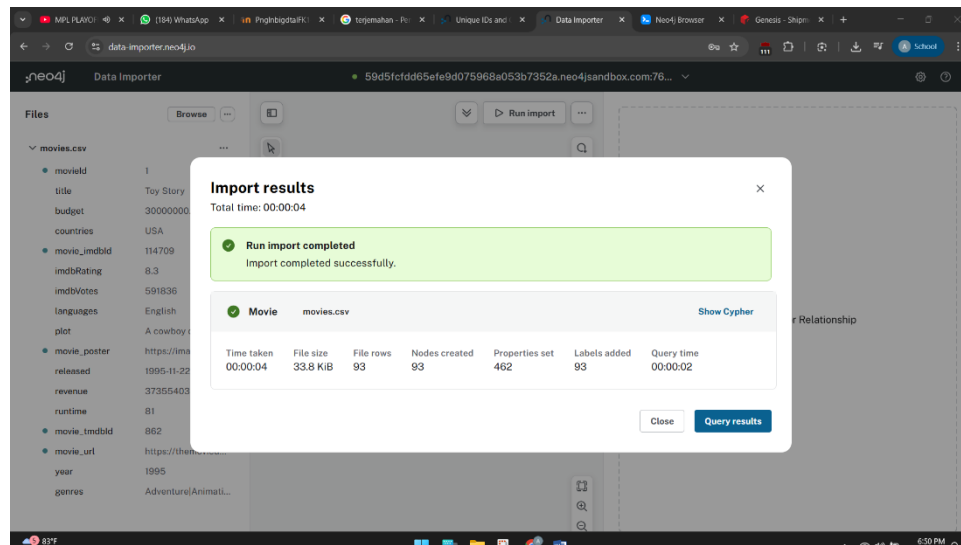
Neo4j mendukung beberapa metode impor data, mulai dari LOAD CSV, Neo4j Data Importer Tool, hingga API atau skrip Python. Pilihan metode bergantung pada sumber data dan kebutuhan pengguna, baik untuk skala kecil maupun besar. Tool-tool ini membantu mempercepat integrasi data dari sistem luar ke dalam graf Neo4j.

g. Aspek Kualitas Data yang Memastikan Dapat Diterapkan Sesuai Konteks

Kualitas data sangat penting dalam proses impor. Data harus bersih, tidak duplikatif, konsisten, dan sesuai dengan model graf. Jika tidak, hasil analisis bisa menyesatkan. Oleh karena itu, validasi data sebelum impor sangat disarankan agar hasil pemodelan akurat dan bermanfaat.

h. Model Data

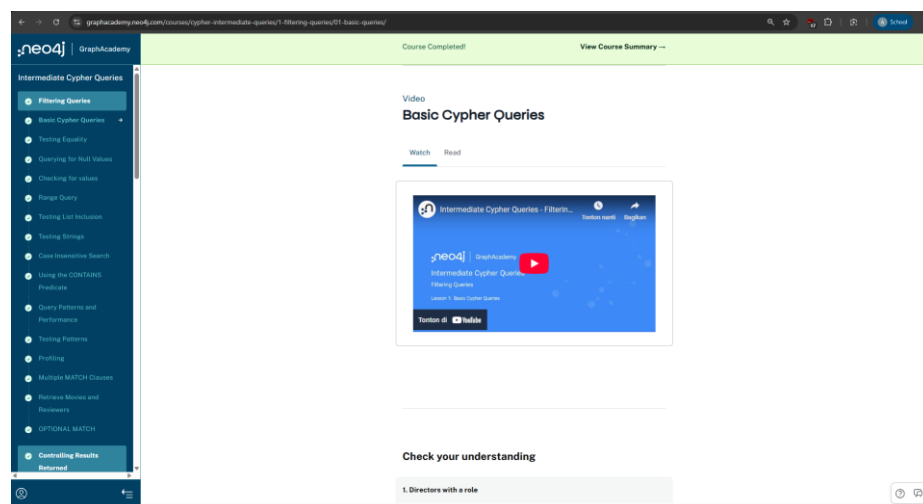
Sebelum mengimpor data, kita harus memiliki model data graf yang jelas: apa saja nodenya, label yang digunakan, relasi antar node, serta properti yang akan ditambahkan. Model ini menjadi pedoman dalam memetakan data mentah menjadi struktur graf yang bermakna, terstruktur, dan siap digunakan untuk query.



Gambar 2. 15 Dokumentasi Latihan Soal Importing Data Fundamental

5. Intermediate Cypher Queries

Modul kelima, Intermediate Cypher Queries, merupakan lanjutan dari modul Cypher dasar. Di sini peserta akan mempelajari query yang lebih kompleks menggunakan klausa lanjutan seperti WITH, COLLECT, UNWIND, ORDER BY, dan LIMIT. Modul ini juga membahas pencocokan pola lanjutan, pengelolaan jalur (path), serta tips optimasi penulisan query agar lebih efisien dan efektif.



Gambar 2. 16 Dokumentasi Pelatihan Intermediate Cypher Queries

Terdapat 70 poin materi penting yang dapat kita pelajari pada modul ini, yaitu:

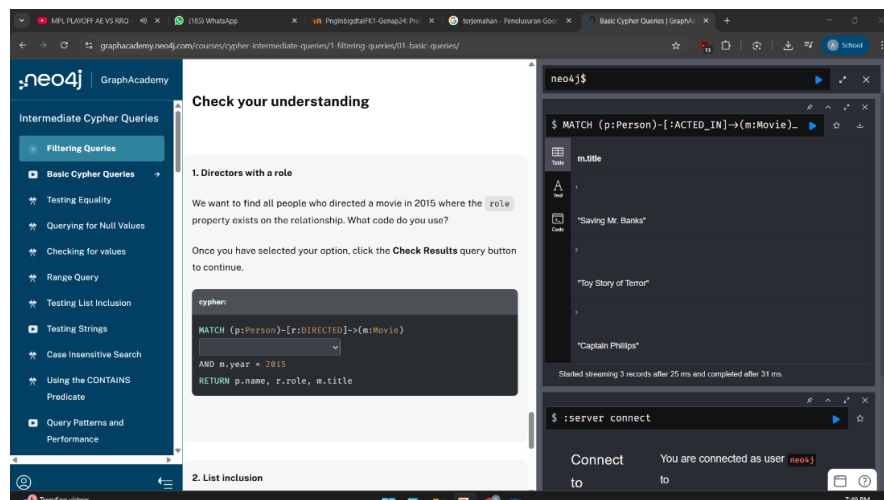
Modul Intermediate Cypher Queries									
Kode panggil data	Penyertaan daftar	Menemukan data diminta	Periksa tidak ada nilai null	Update klausa Where	Next update klausa Where	Uji sertaaan daftar	Uji properti string	Keyword Cypher	Pencarian dg perbedaan
Predikat CONTAINS	Uji ada pola dalam grafik	Performa kueri	Pola pengujian	Buat profil kueri	Mencocokkan pola optional	Pencocokan nilai optional	Ambil data dgn syarat	Optional Match	Mengurutkan baris hasil
Memesan Hasil	Mengurutkan Hasil	Melihat hasil yg diurutkan	Mengurutkan beberpa nilai	Membatasi Hasil	Hilangkan Duplikat	Mengembalik an properti	Mengembalik an data	Mengembalik an informasi	Mengembalik an nilai
Mengembalikn data bersyarat	Menampilkan daftar data	Mengumpuk an data	Menghitung Hasil	Membuat Daftar	Menampilkan jumlah data	Menghitung data durasi	Data temporal dalam grafik	Hitung hari antara 2 tgl	Durasi dalam Menit
Performa kueri terbaik	Melintasi Grafik	Menelusuri Grafik	Hubungan yang dilalui	Menampilkan data berjarak	Temukan jalur terpendek	Mengembalik an data	Menampilkan data hasil	Mengembalik an data berjarak	Variabel Cakupan
Cakupan Cypher	Penggunaan WITH	Penggunaan WITH dg syrt	Membatasi Hasil dg syrt	Cypher untuk agregasi	WITH untuk hasil	Menggunaka n UNWIND	Jenis data pda UNWIND	UNWIND utk mencari hasil	Teruskan hasil dgn UNWIND
Menggunaka n subkueri	Gabung hasil klausa Match	Tambahkan subkueri	Gabungkan 2 data hasil	Parameter yg digunakan	Parameter untuk sesi	Menetapkan bil. bulat	Menetapkan parameter	Menggunaka n parameter	Karakter awal parameter
S1-INFORMATIKA							K1-KARYAWAN		

Gambar 2. 17 Materi pada Modul Intermediate Cypher Queries

Modul ini membahas prinsip dasar pemodelan data graf di Neo4j. Dimulai dari pentingnya menulis kode Cypher yang baik dan benar, termasuk dalam menambahkan hubungan dan label baru ke model graf. Setiap entitas harus memiliki label node yang sesuai, dan hubungan antar entitas ditentukan melalui properti hubungan. Dalam satu graf, jumlah label maksimum juga perlu dipertimbangkan agar struktur tetap efisien dan tidak membingungkan. Elemen-elemen seperti daftar, genre, dan role harus dikelompokkan dengan tepat agar data mudah ditelusuri.

Pada tahap pemodelan berulang, kita mungkin perlu menambahkan lebih banyak node dan hubungan saat model berkembang. Misalnya, menambah label baru atau memecah satu entitas menjadi dua bagian, seperti memisahkan genre nodes. Proses ini juga mencakup refactor grafik menggunakan Cypher untuk menyesuaikan dengan kebutuhan baru. Tujuan dari semua ini adalah untuk mendapatkan model graf yang sesuai dengan domain masalah, menghasilkan data yang mudah diuji, dan mengkhususkan hubungan untuk analisis tertentu.

Saat menentukan model, penting memahami koneksi antar entitas, menghindari label yang tidak relevan, serta menyesuaikan properti agar tidak terjadi duplikasi. Penggunaan label juga harus disesuaikan dengan relasi dinamis, dan kadang diperlukan verifikasi terhadap struktur hubungan seperti `ACTED_IN`. Refactoring digunakan untuk memperbaiki atau menyederhanakan model, misalnya dengan menambahkan node `Role` berdasarkan properti relasi, atau membuat simpul perantara untuk memisahkan relasi kompleks agar lebih mudah dikelola dan dianalisis.



Gambar 2. 18 Dokumentasi Latihan Soal Intermediate Cypher Queries

6. Using Neo4j with Python

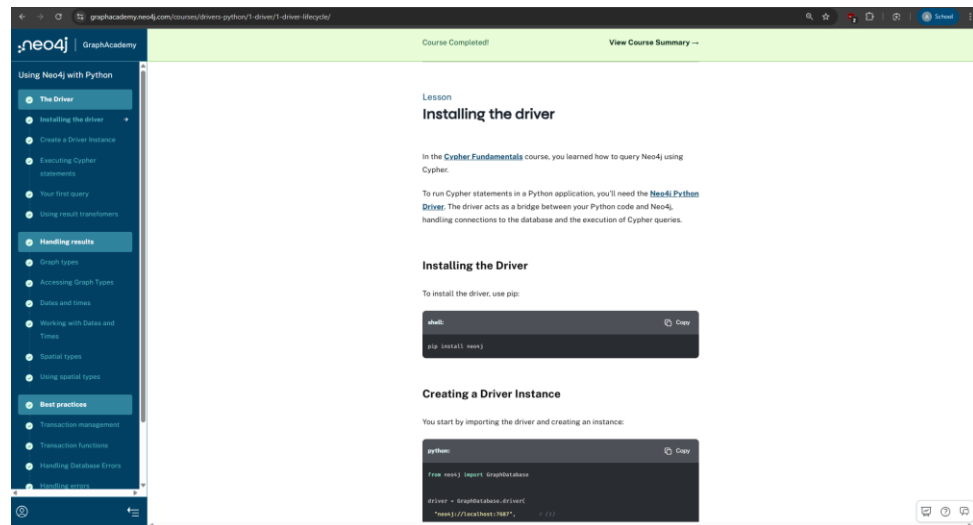
Modul Using Neo4j with Python dirancang khusus bagi peserta yang ingin mengembangkan aplikasi berbasis Python yang terintegrasi langsung dengan Neo4j, sebuah database graf. Modul ini dimulai dengan pengenalan cara menginstal pustaka resmi neo4j menggunakan pip dan dilanjutkan dengan membuat driver instansi, yaitu objek koneksi utama yang digunakan untuk berkomunikasi dengan database. Dengan driver ini, peserta dapat membuka sesi koneksi (session) untuk

menjalankan berbagai query Cypher secara programatis.

Selain menjalankan query, peserta juga diajarkan bagaimana mengambil dan memproses hasil (records) dari database, baik berupa data node, relasi, maupun properti tertentu. Modul ini memberikan teknik membaca hasil secara efisien, mencetak informasi spesifik, dan memanipulasi data hasil query menjadi bentuk Python-friendly seperti dictionary atau list. Peserta juga diperkenalkan pada konsep transaksi (transactions) dalam Neo4j—misalnya, menjalankan batch query secara atomik atau mengalirkan hasil query besar ke klien dengan efisien tanpa membebani memori.

Modul ini juga mencakup materi lanjutan seperti akses properti node dan relasi, penggunaan data waktu (datetime), serta data spasial, misalnya titik koordinat 3D untuk pemetaan atau pelacakan posisi. Tak kalah penting, peserta dipandu untuk menangani kesalahan (error handling) dalam koneksi dan eksekusi query agar aplikasi yang dibangun lebih tangguh dan stabil.

Dengan mengikuti modul ini, peserta akan mampu membangun aplikasi Python yang memanfaatkan kekuatan analisis hubungan dari database graf Neo4j—misalnya untuk sistem rekomendasi, analisis jaringan sosial, pelacakan alur proses, atau mendeteksi pola penipuan. Modul ini menjadi penghubung praktis antara konsep graf dan implementasi langsung dalam pemrograman modern.



Gambar 2. 19 Dokumentasi Pelatihan Using Neo4j with Python

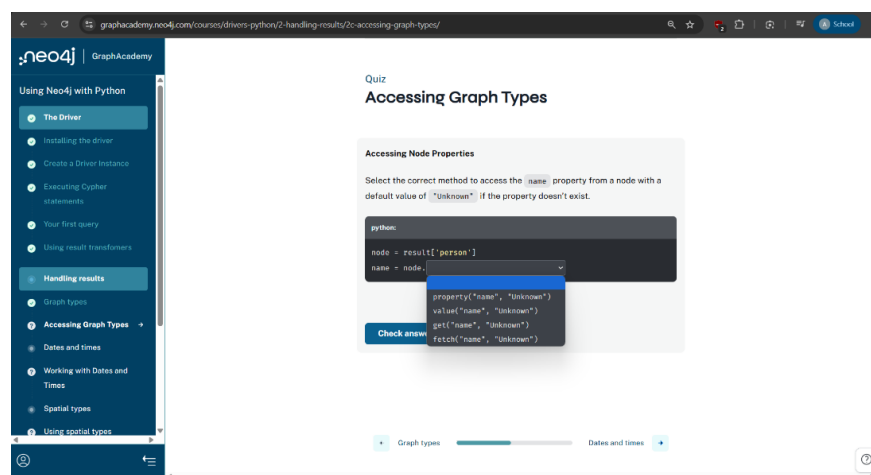
Terdapat 11 poin materi penting yang dapat kita pelajari pada modul ini, yaitu:

Modul Using Neo4j with Python		
Membuat Instansi Driver Cara menginstal Neo4j	Membuat contoh driver baru	Menjalankan pernyataan Cypher
Cetak nama daftar data (records)	Menggunakan transformator hasil	Mengakses Jenis Grafik Mengakses Properti Node
Mengakses Jenis Hubungan Properti mengakses jenis Hub	Membuat objek DateTime	Menggunakan tipe spasial Pembuatan titik 3D
Fungsi Transaksi, mengalirkan hasil pernyataan Cypher yang berjalan lama ke klien (menjalankan kueri baca dalam transaksi)		Penanganan Kesalahan
S1-INFORMATIKA		K1-KARYAWAN

Gambar 2. 20 Materi pada Modul Using Neo4j with Python

Modul ini membahas integrasi Neo4j dengan Python melalui penggunaan Neo4j Driver, dimulai dari cara menginstal driver dan membuat instansinya, hingga menjalankan pernyataan Cypher langsung dari Python. Untuk mengakses dan menampilkan data, pengguna dapat mencetak records (baris data) serta menggunakan transformator hasil agar data yang dikembalikan lebih mudah diolah. Modul ini juga menjelaskan cara mengakses jenis grafik dan properti node, serta mengakses dan membedakan jenis hubungan (relationship).

Dalam pengolahan data lanjutan, pengguna dapat membuat objek `DateTime` untuk memanipulasi data berbasis waktu, serta menggunakan tipe spasial untuk membuat titik koordinat 3D, yang bermanfaat dalam analisis geografis. Modul ini turut membahas fungsi transaksi, termasuk bagaimana menjalankan kueri dalam transaksi jangka panjang agar hasil tetap dapat dikirim ke klien secara efisien. Terakhir, penanganan kesalahan (error handling) juga dijelaskan untuk menjaga stabilitas program Python saat berinteraksi dengan Neo4j.



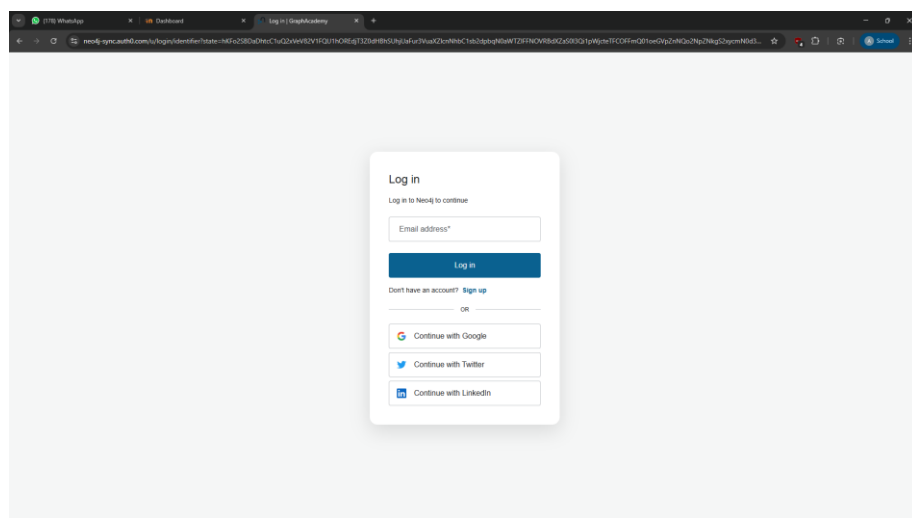
Gambar 2. 21 Dokumentasi Latihan Soal Using Neo4j with Python

2.5.3. Pendaftaran Ujian Sertifikasi

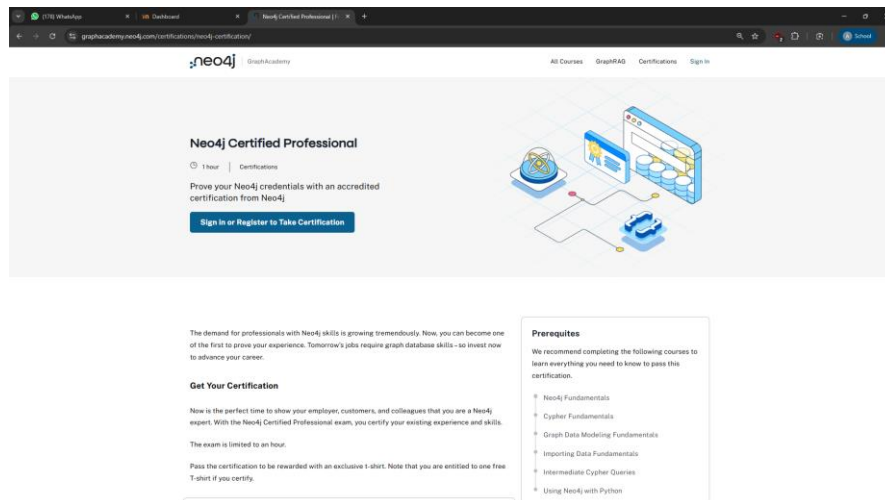
Setelah merasa cukup siap secara materi dan teknis melalui pelatihan dan latihan soal, tahap berikutnya yang dilakukan penulis adalah mendaftarkan diri untuk mengikuti ujian sertifikasi Neo4j Certified Professional. Proses pendaftaran dilakukan secara daring melalui platform resmi GraphAcademy. Untuk dapat mengikuti ujian ini, peserta hanya perlu memiliki akun di situs <https://graphacademy.neo4j.com> dan telah menyelesaikan beberapa kursus

pengantar yang direkomendasikan. Tidak ada biaya yang dikenakan, karena ujian ini disediakan secara gratis sebagai bentuk komitmen Neo4j dalam membangun komunitas pengguna dan profesional teknologi graph database.

Formulir pendaftaran cukup sederhana dan hanya memerlukan data dasar seperti nama lengkap, alamat email, serta kesediaan peserta untuk mengikuti ujian dalam satu sesi tanpa jeda. Setelah pendaftaran dikonfirmasi, peserta dapat langsung mengakses ujian kapan saja melalui dasbor akun GraphAcademy. Tidak ada jadwal khusus atau sesi terjadwal, sehingga peserta memiliki fleksibilitas penuh dalam menentukan waktu pelaksanaan ujian sesuai kesiapan masing-masing. Sistem ini sangat mendukung pembelajaran mandiri, karena memberikan keleluasaan bagi peserta untuk mempersiapkan diri secara optimal sebelum menghadapi ujian kompetensi yang bersifat internasional tersebut.

A screenshot of a web browser displaying the login page of GraphAcademy. The page has a light gray background. In the center, there is a white login form with a blue header 'Log in'. Below the header, it says 'Log in to Neo4j to continue'. There is a text input field for 'Email address*'. Below this is a blue 'Log in' button. Under the button, it says 'Don't have an account? Sign up' with a blue link. Below this is an 'OR' separator. There are three social login options: 'Continue with Google' (with a Google logo), 'Continue with Twitter' (with a Twitter logo), and 'Continue with LinkedIn' (with a LinkedIn logo). The browser's address bar shows a long URL starting with 'neo4j-gynaetha.com'.

Gambar 2. 22 Dokumentasi Form Untuk Mendaftar Sertifikasi



Gambar 2. 23 Dokumentasi Halaman Untuk Mendaftar Sertifikasi

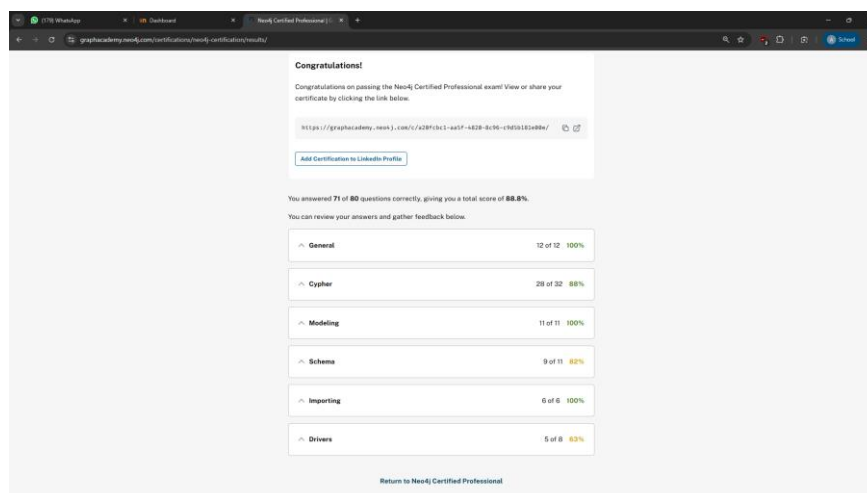
2.5.4. Pelaksanaan Ujian

Pelaksanaan ujian sertifikasi Neo4j Certified Professional dirancang untuk menguji pemahaman mendalam peserta terhadap konsep graph database, bahasa query Cypher, serta penerapan praktis model graph dalam skenario nyata. Ujian ini berdurasi sekitar 60 menit dan terdiri dari sekitar 80 soal pilihan ganda. Seluruh soal disusun dalam format teori dan studi kasus, di mana peserta diminta untuk menafsirkan hasil kueri Cypher, menganalisis struktur graph, serta menentukan solusi yang paling tepat berdasarkan konteks tertentu. Meskipun tidak terdapat praktik langsung seperti menulis kode atau query secara live, banyak pertanyaan yang menuntut peserta untuk memahami hasil output dari Cypher, struktur relasi antar node, serta bagaimana model graph sebaiknya dibentuk untuk menyelesaikan suatu kasus bisnis atau teknis. Oleh karena itu, peserta disarankan untuk memiliki pengalaman hands-on melalui latihan atau sandbox Neo4j sebelum mengikuti ujian, agar lebih siap menghadapi soal-soal yang bersifat aplikatif dan berbasis skenario.

2.5.5. Evaluasi dan Pengumuman Hasil

Pada tahap evaluasi dan pengumuman hasil ujian sertifikasi Neo4j Certified Professional, peserta akan mendapatkan hasil ujian secara langsung setelah proses submit selesai dilakukan. Sistem penilaian yang digunakan bersifat otomatis, sehingga tidak ada jeda waktu untuk proses koreksi manual. Apabila peserta berhasil lulus, maka sertifikat digital akan langsung tersedia dan dapat diakses melalui link yang disediakan oleh sistem—biasanya dalam bentuk PDF yang telah terverifikasi oleh Neo4j dan dapat dibagikan secara profesional, misalnya ke LinkedIn atau CV.

Selain sertifikat, peserta yang lulus juga akan mendapatkan reward berupa kaos eksklusif kelulusan dari Neo4j. Untuk mendapatkan hadiah ini, peserta harus mengisi dan mengirimkan data alamat pengiriman melalui formulir resmi yang ditampilkan setelah kelulusan. Kaos akan dikirimkan ke alamat tersebut sebagai bentuk apresiasi dan simbol keberhasilan dalam menyelesaikan ujian. Proses ini tidak hanya memberi pengakuan formal, tetapi juga meningkatkan motivasi dan kebanggaan peserta sebagai bagian dari komunitas profesional pengguna Neo4j.



Gambar 2. 24 Dokumentasi Hasil Evaluasi Sertifikasi



Gambar 2. 25 Dokumentasi Sertifikat Neo4j Certified Professional

2.6. Bukti Pendukung dan Dokumentasi Proses

Selama proses menuju sertifikasi, beberapa bukti pendukung yang dapat ditampilkan dalam lampiran atau portofolio antara lain:

1. Sertifikat kelulusan dari Neo4j Certified Professional
2. Screenshot hasil pelatihan di Neo4j Academy (Graph Fundamentals, Cypher Fundamentals, dsb.)
3. Catatan hasil kuis atau latihan
4. Dokumentasi penggunaan Cypher untuk menyelesaikan soal studi kasus
5. Screenshot hasil kerja *hands-on lab* atau proyek mini berbasis graph