BAB II
PELAKSANAAN UJIAN KOMPETENSI

2.1. Pendahuluan

Pelaksanaan uji kompetensi sertifikasi *Neo4j Certified Professional* merupakan bagian dari upaya untuk memperoleh pengakuan resmi atas kemampuan dalam mengelola dan menggunakan database graph Neo4j. Sertifikasi ini diselenggarakan oleh Neo4j Inc. dan dapat diakses secara daring secara global. Proses sertifikasi ini mencakup tahapan-tahapan pembelajaran, pelatihan, hingga pelaksanaan ujian yang menguji baik aspek teori maupun praktik.

2.2. Tujuan Sertifikasi

Tujuan dari mengikuti sertifikasi *Neo4j Certified Professional* adalah untuk:

1. Memvalidasi penguasaan terhadap konsep dan implementasi basis data graph menggunakan Neo4j.
2. Membekali peserta dengan keterampilan praktis yang dapat langsung diterapkan di dunia industri.
3. Menambah nilai profesionalisme dan daya saing lulusan dalam bidang data dan pemrograman.
4. Memperkuat portofolio individu sebagai bukti kompetensi di bidang teknologi graph.

2.3. Manfaat Sertifikasi

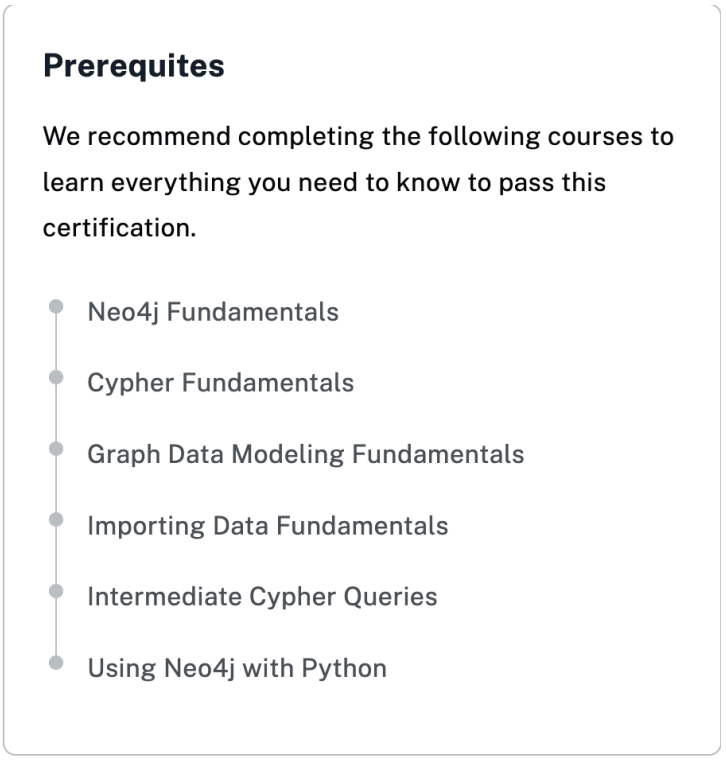
Manfaat yang diperoleh dari pelaksanaan sertifikasi ini antara lain:

1. Mendapatkan pengakuan resmi dari Neo4j Inc. sebagai profesional yang kompeten.
2. Meningkatkan peluang karir di bidang data engineering, data analysis,

dan pengembangan aplikasi berbasis graph.

3. Menambah kredibilitas dalam portofolio dan profil profesional. Memperluas wawasan serta pemahaman mendalam tentang manajemen data relasional yang kompleks.

2.4. Silabus dan Materi yang Dipelajari



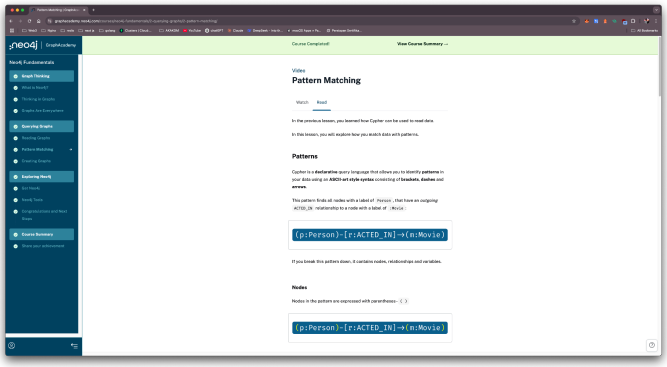
Gambar 2.1. Dokumentasi Silabus Sertifikasi *Neo4j Certified Professional*

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.

Berikut adalah rincian silabus yang dipelajari, disertai contoh *query* nyata dan studi kasus penerapan:

2.4.1. Dasar-Dasar *Graph Database*

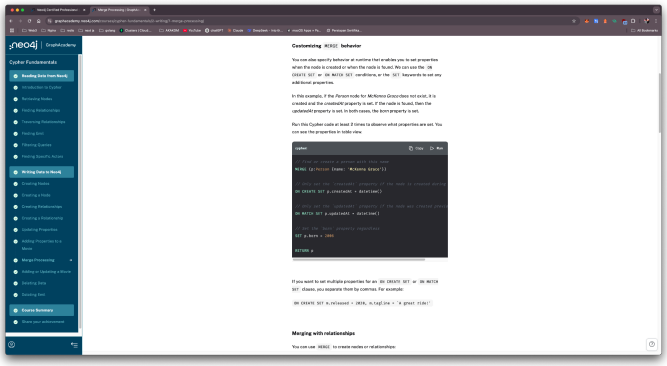
Materi dimulai dengan pengenalan dasar-dasar graph database, termasuk pemahaman tentang node, relationship, dan properties sebagai struktur inti dalam Neo4j. Konsep ini dibandingkan secara langsung dengan model relasional tradisional untuk memberikan pemahaman menyeluruh mengenai keunggulan graph dalam menangani relasi data yang kompleks. Contoh dokumentasi hasil pelatihan dasar-dasar graph database ditunjukkan pada Gambar 2.2.



Gambar 2.2. Dokumentasi pelatihan Neo4j Fundamentals.

2.4.2. Bahasa Query Cypher

Pada materi ini peserta mempelajari bahasa *query Cypher* yang merupakan bahasa utama dalam mengakses dan memanipulasi data di Neo4j. Pembelajaran mencakup sintaks dasar seperti MATCH, CREATE, WHERE, RETURN, serta penggunaan fungsi agregasi seperti COUNT dan AVG. Peserta juga diberikan latihan untuk menggabungkan berbagai klausa guna membangun query yang lebih kompleks. Gambar 2.3 menunjukkan tangkapan layar dari pelatihan *Cypher Fundamentals* yang menjadi bagian penting dalam membangun keterampilan dasar menulis *query*.

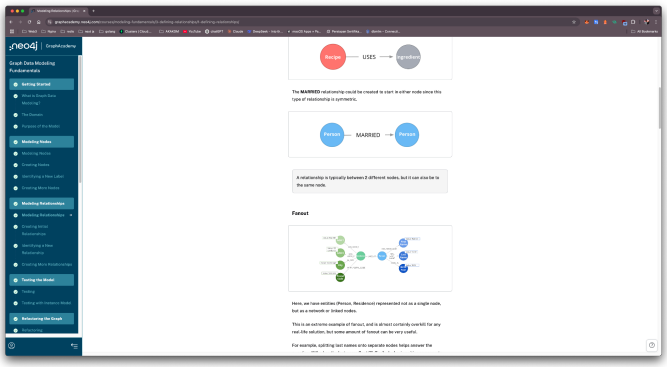


Gambar 2.3. Dokumentasi Pelatihan *Cypher Fundamentals*.

2.4.3. Modeling Data di Neo4j

Materi dilanjutkan dengan topik pemodelan data graph. Peserta diajak untuk merancang skema database berdasarkan skenario nyata dengan mempertimbangkan efisiensi *query* dan keterbacaan data. Latihan mencakup bagaimana menentukan entitas dan relasi yang tepat, serta penerapan prinsip-prinsip desain *graph* yang baik.

Dalam konteks performa, pelatihan juga membahas pentingnya penggunaan *index* dan *constraint*. Peserta diajarkan cara membuat *indeks* untuk mempercepat pencarian dan *constraint* untuk menjaga integritas data, termasuk penerapan pada properti unik seperti NIM mahasiswa. Ilustrasi hasil pelatihan dalam modul *Graph Data Modeling Fundamentals* dapat dilihat pada Gambar 2.4.



Gambar 2.4. Dokumentasi Pelatihan *Graph Data Modeling Fundamentals*.

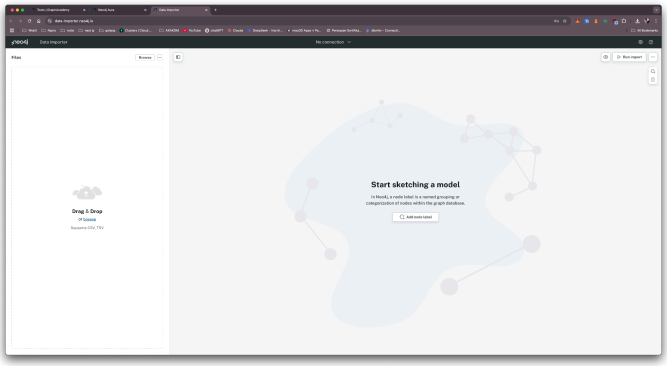
2.4.4. Data Import dan Neo4j AuraDB

Pada materi ini peserta diperkenalkan pada cara mengimpor data dari file eksternal menggunakan fitur Neo4j Data Importer maupun melalui *query Cypher* seperti LOAD CSV. Proses penggunaan fitur Neo4j *Data Importer* ditunjukkan pada Gambar 2.5. Kegiatan ini dirancang untuk memberikan pemahaman praktis dalam menyiapkan dan migrasi data dari sumber luar ke Neo4j.

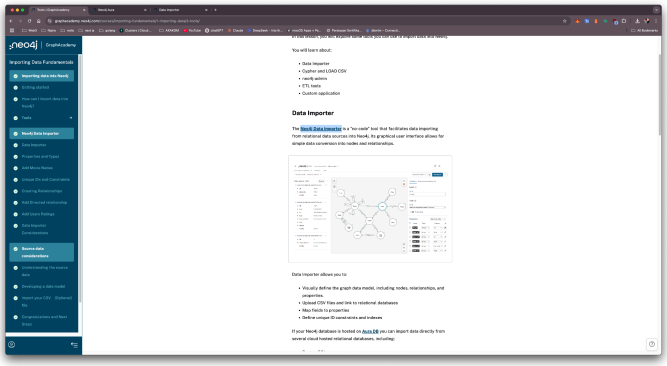
Dokumentasi pembelajaran dari modul *Importing Data Fundamentals* dapat dilihat pada Gambar 2.6.

Gambar 2.7 menampilkan antarmuka dan aktivitas dalam penggunaan layanan Neo4j AuraDB.

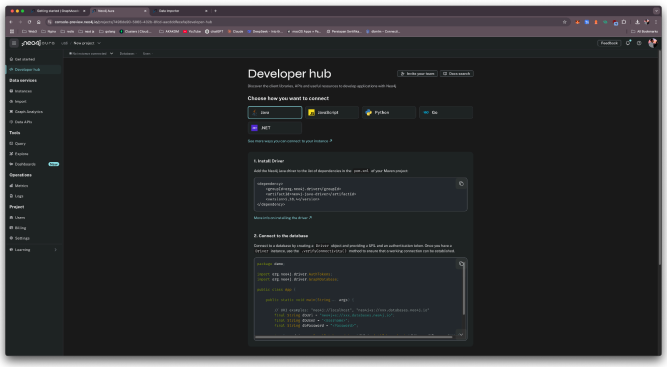
Untuk memperkuat aspek aplikatif, pelatihan juga menyertakan studi kasus nyata dan praktik optimasi query. Peserta diajak untuk mengevaluasi performa query menggunakan perintah PROFILE dan merancang solusi berbasis graph untuk permasalahan seperti sistem rekomendasi dan deteksi penipuan.



Gambar 2.5. Dokumentasi Pelatihan Penggunaan Neo4j *Data Importer*.



Gambar 2.6. Dokumentasi Pelatihan *Importing Data Fundamentals*.



Gambar 2.7. Dokumentasi Penggunaan Neo4j Aura

2.4.5. Bahasa Query Cypher tingkat lanjut

Setelah memahami dasar-dasar bahasa *query Cypher* seperti MATCH, CREATE, WHERE, dan RETURN, peserta sertifikasi juga dibekali dengan kemampuan lanjutan dalam menyusun kueri yang lebih kompleks dan efisien. Bahasa *query Cypher* tingkat lanjut sangat penting dalam praktik penggunaan Neo4j, terutama ketika berhadapan dengan volume data yang besar dan relasi yang kompleks.

Beberapa topik lanjutan yang dipelajari dalam pelatihan meliputi:

2.4.5.1. *Pattern Comprehension* dan *Filtering*

Cypher memungkinkan pencocokan pola relasi yang lebih fleksibel dengan menggunakan *pattern comprehension*, serta teknik penyaringan hasil yang lebih spesifik.

```
MATCH (p:Person)
RETURN p.name, [(p)-[:FRIEND_WITH]->(f) | f.name] AS friends
```

Gambar 2.8. Contoh *Cypher Query Pattern Comprehension* dan *Filtering*

2.4.5.2. Penggunaan Subquery (CALL { ... })

Subquery memungkinkan penyusunan logika *query* bersarang untuk memproses bagian data tertentu tanpa mencemari main query context.

```
MATCH (p:Product)
CALL {
  WITH p
  MATCH (p)<-[:BOUGHT]-(u:User)
  RETURN count(u) AS buyerCount
}
RETURN p.name, buyerCount
```

Gambar 2.9. Contoh Penggunaan Subquery

2.4.5.3. Agregasi dan Proyeksi Lanjutan

Dalam *Cypher* tingkat lanjut, pengguna dapat menggabungkan WITH, ORDER BY, SKIP, LIMIT, serta COLLECT() untuk merangkum dan memproyeksikan data dalam bentuk agregat.

```
MATCH (a:Author)-[:WROTE]->(b:Book)
RETURN a.name, COLLECT(b.title) AS booksWritten
```

Gambar 2.10. Contoh *Cypher Query* Agregasi

2.4.5.4. *Conditional Expressions* dan *Case Logic*

Cypher mendukung ekspresi bersyarat menggunakan CASE untuk menyesuaikan hasil berdasarkan kondisi tertentu.

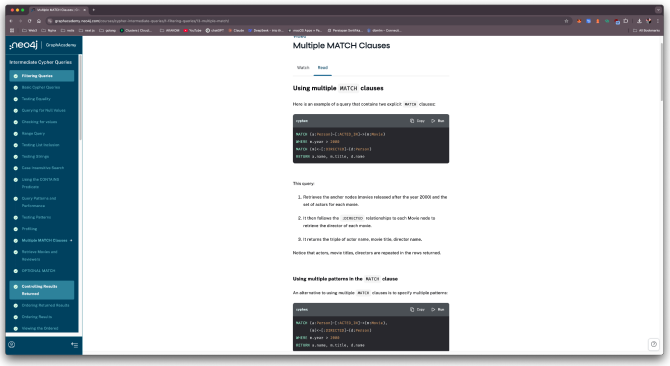
```
MATCH (p:Person)
RETURN p.name,
       CASE
         WHEN p.age >= 60 THEN 'Senior'
         WHEN p.age >= 18 THEN 'Adult'
         ELSE 'Minor'
       END AS category
```

Gambar 2.11. Contoh *Cypher Query Conditional Expressions* dan *Case Logic*

2.4.5.5. *Profiling* dan Optimasi Query untuk menganalisis performa dan efisiensi query.

```
PROFILE MATCH (p:Person)-[:FRIEND_WITH]->(f:Person)
WHERE p.name = 'Andi'
RETURN f.name
```

Gambar 2.12. Contoh *Cypher Query Profiling*



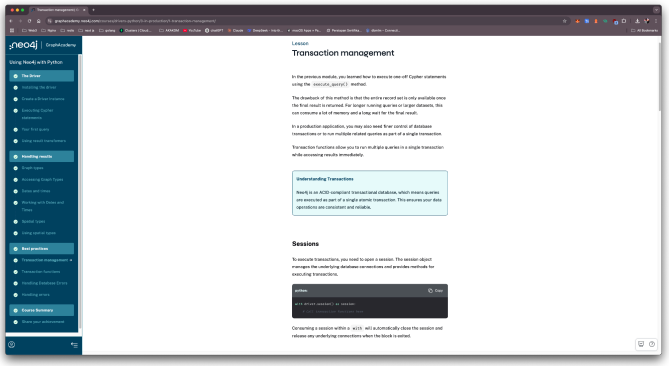
Gambar 2.13. Dokumentasi Pelatihan *Intermediate Cypher Queries*

Contoh hasil pembelajaran untuk materi query lanjutan seperti *pattern comprehension*, subquery, dan *profiling* ditunjukkan pada Gambar 2.13.

2.4.6. Implementasi Neo4j dengan menggunakan Python

Salah satu keunggulan Neo4j sebagai basis data graph adalah kemampuannya untuk diintegrasikan dengan berbagai bahasa pemrograman, salah satunya Python. Dalam pelatihan yang diikuti sebagai bagian dari proses sertifikasi, peserta juga mempelajari bagaimana membangun koneksi dan menjalankan query Cypher menggunakan Python.

Python dipilih karena merupakan bahasa pemrograman yang populer di kalangan data engineer dan data scientist, serta memiliki pustaka resmi bernama neo4j yang mendukung akses langsung ke database Neo4j. Pelatihan penggunaan Neo4j melalui bahasa Python terdokumentasi dalam Gambar 2.14.



Gambar 2.14. Dokumentasi Pelatihan Using Neo4j with Python

Gambar 2.15 menunjukkan potongan kode Python yang digunakan untuk melakukan koneksi ke database Neo4j menggunakan pustaka neo4j. Pada potongan kode ini, ditampilkan proses inisialisasi driver yang menggunakan URI database, username, dan password. Langkah ini merupakan dasar dari integrasi aplikasi Python dengan Neo4j, yang memungkinkan eksekusi kueri Cypher dari aplikasi Python secara langsung.

```python
from neo4j import GraphDatabase

# koneksi driver
uri = "bolt://localhost:7687"
username = "neo4j"
password = "neo4j"

driver = GraphDatabase.driver(uri, auth=(username, password))
```

Gambar 2.15. Python Code Connection ke Neo4J

Pada gambar 2.16 ditampilkan contoh kode Python untuk membuat node bertipe `Mahasiswa` dan `MataKuliah` di dalam graph database Neo4j. Masing-masing node diberi atribut seperti nama mahasiswa atau nama mata kuliah. Proses ini merupakan bagian dari tahap data modeling secara praktis dalam konteks dunia pendidikan.

```python
# Cypher Query untuk create node Mahasiswa & Mata Kuliah

from neo4j import GraphDatabase

def create_data(tx):

    tx.run("
        CREATE (m1:Mahasiswa {nama: 'Andi', nim: 'A001'})
        CREATE (m2:Mahasiswa {nama: 'Budi', nim: 'A002'})
        CREATE (mk1:MataKuliah {kode: 'MK001', nama: 'Basis Data'})
        CREATE (mk2:MataKuliah {kode: 'MK002', nama: 'Jaringan Komputer'})
    ")

with driver.session() as session:
    session.write_transaction(create_data)
```

Gambar 2.16. Python Code Untuk Create Node Mahasiswa dan Mata Kuliah

Gambar 2.17 memperlihatkan cuplikan kode Python untuk membuat relasi `:MENGAMBIL` antara node `Mahasiswa` dan `MataKuliah`. Relasi ini menyatakan bahwa seorang mahasiswa mengambil suatu mata kuliah tertentu, yang merupakan contoh sederhana dari pemodelan hubungan antar entitas dalam sistem akademik.

```python
#Cypher Query untuk Membuat Relationship :MENGAMBIL

def create_relationship(tx):
    tx.run("
        MATCH (m:Mahasiswa {nama: 'Andi'}), (mk:MataKuliah {nama: 'Basis Data'})
        CREATE (m)-[:MENGAMBIL]->(mk)

        MATCH (m:Mahasiswa {nama: 'Budi'}), (mk:MataKuliah {nama: 'Jaringan Komputer'})
        CREATE (m)-[:MENGAMBIL]->(mk)
    ")

with driver.session() as session:
    session.write_transaction(create_relationship)
```

Gambar 2.17. Python Code Untuk Membuat Relationship :`MENGAMBIL`

Gambar 2.18 menampilkan contoh kode Python yang menjalankan query Cypher untuk menampilkan daftar mahasiswa beserta mata kuliah yang mereka ambil. Hasil dari query ini digunakan untuk menampilkan hubungan antar entitas dalam bentuk data tabular atau JSON, yang umum digunakan dalam aplikasi data analitik.

```python
#Query Data Mahasiswa yang Mengambil Mata Kuliah

def query_data(tx):
    result = tx.run("
        MATCH (m:Mahasiswa)-[:MENGAMBIL]->(mk:MataKuliah)
        RETURN m.nama AS nama_mahasiswa, mk.nama AS nama_matakuliah
    ")
    for record in result:
        print(f"{record['nama_mahasiswa']} mengambil {record['nama_matakuliah']}")

with driver.session() as session:
    session.read_transaction(query_data)
```

Gambar 2.18. Python Code Untuk Menampilkan Data Mahasiswa yang Mengambil Mata Kuliah

2.5. Tahapan Pelaksanaan Ujian

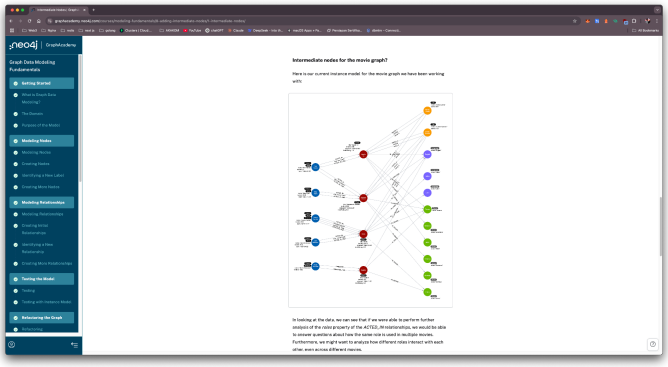
Proses untuk memperoleh sertifikasi *Neo4j Certified Professional* dilakukan dalam beberapa tahap sebagai berikut:

2.5.1. Persiapan Belajar Mandiri

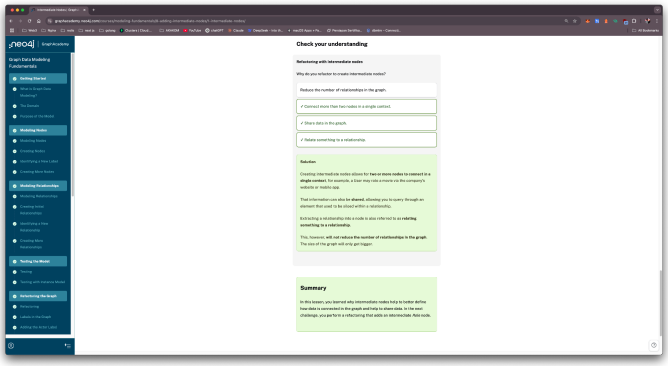
Sebelum mengikuti ujian sertifikasi *Neo4j Certified Professional*, dilakukan persiapan belajar mandiri secara intensif melalui berbagai sumber resmi yang disediakan oleh Neo4j. Tahapan ini mencakup eksplorasi materi pada platform Neo4j GraphAcademy, yang menyediakan pelatihan daring gratis dalam bentuk video, teks interaktif, serta hands-on labs. Materi-materi seperti *Neo4j Fundamentals*, *Cypher Fundamentals*, dan *Graph Data Modeling* menjadi dasar penting yang harus dikuasai untuk memahami konsep *graph database* dan bahasa *query Cypher*. Selain itu, dokumentasi resmi Neo4j dan forum komunitas digunakan sebagai referensi tambahan untuk memperdalam pemahaman dan mencari solusi dari permasalahan yang muncul selama latihan mandiri.

Aktivitas belajar mandiri juga mencakup praktik langsung menggunakan Neo4j Sandbox dan Neo4j AuraDB untuk menguji *query-query* yang telah dipelajari secara teori. Modul latihan dan kuis evaluasi pada akhir setiap pelatihan diselesaikan secara rutin sebagai tolak ukur penguasaan materi. Pendekatan ini tidak hanya melatih pemahaman konseptual, tetapi juga memperkuat keterampilan teknis dalam mengelola *graph database* secara nyata. Komitmen untuk menyelesaikan semua modul pelatihan serta konsistensi dalam praktik langsung

menjadi fondasi utama dalam menghadapi ujian sertifikasi, sekaligus sebagai bentuk tanggung jawab profesional dalam proses pembelajaran mandiri yang terstruktur.



Gambar 2.19. Dokumentasi Belajar Mandiri *Graph Data Modeling Fundamentals*



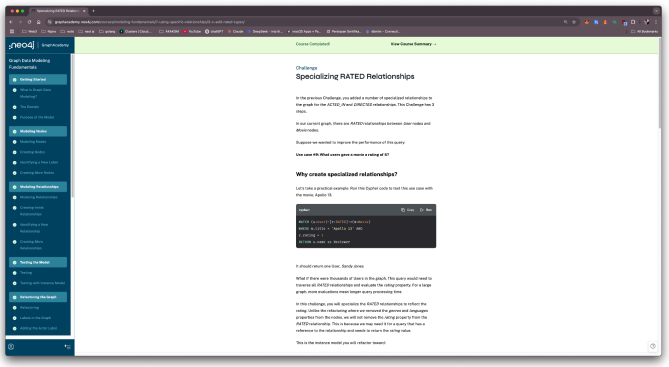
Gambar 2.20. Dokumentasi Quiz Belajar Mandiri

2.5.2. Simulasi dan Latihan Soal

Setelah tahap belajar mandiri diselesaikan, proses dilanjutkan ke tahap simulasi dan latihan soal guna mengasah pemahaman serta meningkatkan kesiapan dalam menghadapi format ujian sertifikasi. Simulasi dilakukan melalui kuis interaktif yang tersedia pada akhir setiap modul pelatihan di GraphAcademy, dengan soal-soal yang dirancang menyerupai bentuk ujian sesungguhnya. Kuis-kuis tersebut memberikan umpan balik secara langsung, sehingga kelemahan

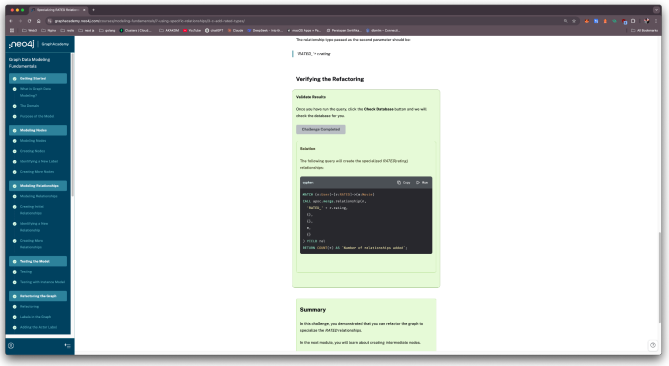
konsep yang belum dikuasai dapat diidentifikasi, dan kesalahan pemahaman dapat diperbaiki secara cepat dan efektif.

Selain itu, *sample dataset* dari modul pelatihan turut dimanfaatkan untuk melakukan eksplorasi lebih lanjut terhadap berbagai kasus nyata yang berpotensi muncul dalam ujian sertifikasi. Melalui pendekatan ini, penyusunan *query Cypher* berdasarkan skenario data yang kompleks dilakukan dan hasilnya dievaluasi secara mandiri. Proses ini mencakup analisis hasil *query*, interpretasi struktur *graph*, serta pemahaman terhadap logika di balik pola relasi data. Latihan intensif dengan pendekatan *problem-based learning* terbukti efektif dalam membangun kepercayaan diri dan meningkatkan ketepatan dalam menjawab soal ujian, yang sebagian besar menuntut kemampuan interpretatif dan konseptual dalam memahami struktur *graph* serta hasil eksekusi *query*.



Gambar 2.21. Dokumentasi *Challenge* Sebagai Latihan Soal

Gambar 2.21 menunjukkan dokumentasi salah satu *challenge* (tantangan) dalam GraphAcademy yang berfungsi sebagai latihan soal sebelum mengikuti ujian sertifikasi. *Challenge* ini membantu peserta menguji pemahaman melalui studi kasus dan query nyata yang mencerminkan soal-soal dalam ujian.



Gambar 2.22. Dokumentasi Jawaban Pada *Challange* Sebagai Latihan Soal

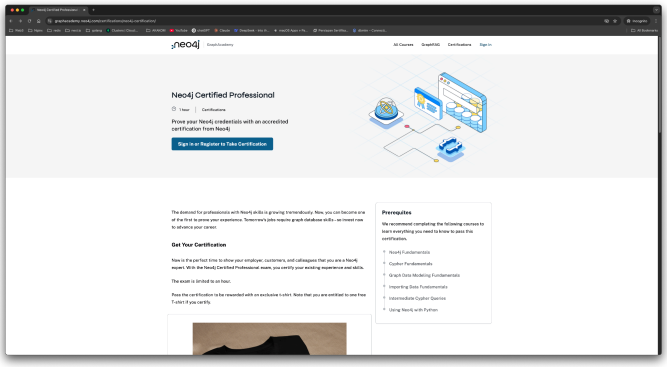
Gambar 2.22 menampilkan hasil atau tangkapan layar dari jawaban yang diberikan pada salah satu *challenge* di GraphAcademy. Response langsung dari sistem digunakan untuk mengevaluasi keakuratan query dan pemahaman peserta terhadap pola relasi data.

2.5.3. Pendaftaran Ujian Sertifikasi

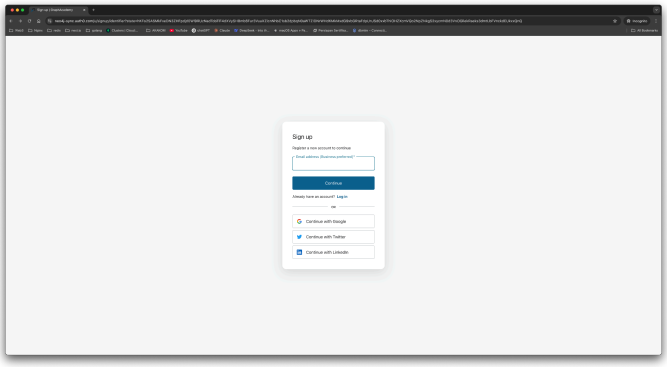
Setelah kesiapan materi dan teknis dianggap memadai melalui pelatihan dan latihan soal, tahap selanjutnya adalah melakukan pendaftaran untuk mengikuti ujian sertifikasi *Neo4j Certified Professional*. Proses pendaftaran dilakukan secara daring melalui platform resmi GraphAcademy. Untuk mengikuti ujian ini, peserta diwajibkan memiliki akun pada situs https://graphacademy.neo4j.com dan telah menyelesaikan beberapa kursus pengantar yang direkomendasikan. Tidak terdapat biaya pendaftaran, karena ujian ini disediakan secara gratis sebagai bagian dari komitmen Neo4j dalam membangun komunitas pengguna dan profesional di bidang 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.



Gambar 2.23. Dokumentasi Halaman Untuk Mendaftar Sertifikasi



Gambar 2.24. Dokumentasi Form Untuk Mendaftar Sertifikasi

Gambar 2.23 memperlihatkan antarmuka halaman pendaftaran untuk mengikuti ujian sertifikasi *Neo4j Certified Professional*.

Gambar 2.24 merupakan formulir yang harus diisi oleh peserta untuk mendaftarkan diri secara resmi. Proses ini dilakukan secara daring dan tidak dipungut biaya.

2.5.4. Pelaksanaan Ujian

Ujian sertifikasi *Neo4j Certified Professional* dilaksanakan secara daring dan dapat diakses melalui akun peserta pada platform GraphAcademy. Ujian diikuti setelah seluruh modul pelatihan dan latihan soal diselesaikan sebagai bentuk kesiapan akhir. Ujian ini terdiri atas sekitar 80 soal pilihan ganda yang dirancang untuk menguji pemahaman konsep dasar hingga lanjutan dalam penggunaan Neo4j serta bahasa *query Cypher*. Seluruh soal disusun dalam konteks studi kasus dunia nyata, sehingga peserta dituntut tidak hanya memahami sintaksis, tetapi juga mampu menganalisis pola hubungan data, memilih pendekatan pemodelan yang sesuai, serta memahami logika eksekusi *query*.

Durasi pengerjaan ujian adalah sekitar 60 menit, dan tidak ada fitur "pause" yang memperbolehkan jeda di tengah sesi, sehingga fokus dan konsentrasi penuh sangat dibutuhkan. Sebagian besar soal menampilkan cuplikan *query Cypher* yang perlu dianalisis, hasil eksekusi *query* yang harus diinterpretasikan, atau skenario penggunaan graph database dalam permasalahan dunia nyata seperti sistem rekomendasi, deteksi penipuan, dan relasi antar-entitas kompleks. Selama ujian berlangsung, tidak ada pengawasan langsung (proctored), namun integritas dan kejujuran peserta tetap dijunjung tinggi, sesuai dengan etika profesional. Hasil ujian langsung ditampilkan begitu semua soal diselesaikan dan dikirim, memungkinkan peserta segera mengetahui apakah mereka berhasil memperoleh sertifikasi.

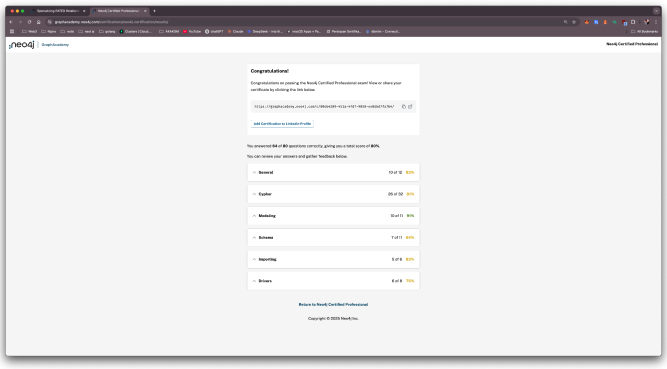
2.5.5. Evaluasi dan Pengumuman Hasil

Setelah seluruh soal ujian diselesaikan dan dikirimkan, sistem secara otomatis melakukan evaluasi terhadap jawaban peserta. Proses evaluasi ini bersifat instan karena setiap soal telah terintegrasi dengan sistem penilaian berbasis platform. Hasil ujian dapat diperoleh dalam hitungan detik setelah mengirimkan jawaban akhir. Jika peserta mencapai nilai minimum kelulusan sebesar 80%, maka status "Lulus" akan langsung ditampilkan di layar beserta informasi sertifikat digital yang dapat diakses dan diverifikasi secara daring.

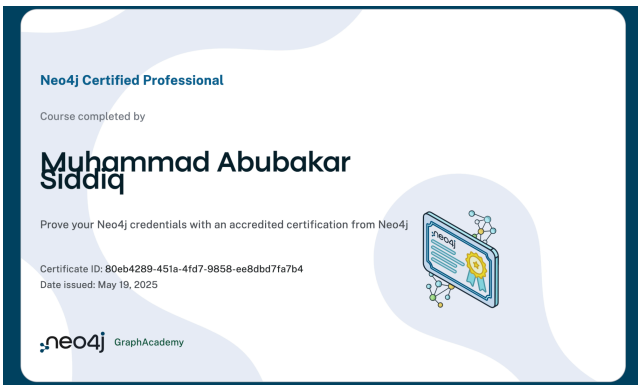
Kecepatan dalam proses evaluasi ini menjadi keunggulan dari sistem sertifikasi *Neo4j Certified Professional*, karena peserta tidak perlu menunggu lama untuk mengetahui hasil uji kompetensi mereka.

Selain sertifikat digital resmi yang dapat diunduh dalam format PDF, peserta yang lulus juga memperoleh tautan unik menuju halaman verifikasi sertifikat yang dapat dibagikan secara publik atau dicantumkan dalam portofolio profesional, seperti LinkedIn atau CV. Sebagai bentuk apresiasi tambahan, Neo4j menyediakan *reward* berupa kaos kelulusan yang dapat diklaim secara gratis oleh peserta yang lulus, melalui pengisian formulir pengiriman alamat. *Reward* tersebut telah berhasil diklaim dan diterima sebagai simbol keberhasilan dalam menyelesaikan seluruh proses sertifikasi. Keseluruhan pengalaman ini memberikan nilai pencapaian tersendiri serta memperkuat motivasi untuk terus mengembangkan kompetensi di bidang teknologi data terhubung.

Gambar 2.25 menampilkan hasil evaluasi akhir dari pelaksanaan ujian sertifikasi. Sistem secara otomatis menunjukkan hasil lulus atau tidaknya peserta beserta skor akhir setelah seluruh soal dikerjakan.



Gambar 2.25. Dokumentasi Hasil Evaluasi Sertifikasi



Gambar 2.26. Dokumentasi Sertifikat *Neo4j Certified Professional*

Gambar 2.26 merupakan tampilan visual dari sertifikat resmi yang diberikan oleh Neo4j Inc. Sertifikat ini dilengkapi dengan tautan verifikasi dan menjadi bukti resmi atas penguasaan kompetensi dalam penggunaan graph database Neo4j.

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*