



Proyek Kelompok Knowledge Graph Gasal 2025/2026



Deskripsi

Pada proyek akhir ini, kalian diminta untuk mendesain dan mengembangkan aplikasi berbasis teknologi **Knowledge Graph (KG)**.

1. 🕸 Aplikasi yang dikembangkan merupakan aplikasi berbasis **Knowledge Graph**. Keindahan tampilan (UI) tidak menjadi fokus utama, namun akan diberikan **nilai bonus** bagi tim yang menghadirkan desain yang menarik.
2. 🛠 Data Knowledge Graph dapat dibangun menggunakan format **RDF** ataupun **Property Graph**.
 - Artinya, kalian juga perlu menyiapkan **Triple Store** (jika menggunakan RDF) atau **Neo4j Backend API** (jika menggunakan Property Graph) sebagai bagian dari infrastruktur aplikasi kalian.
3. 📁 Sumber data yang digunakan dalam Knowledge Graph harus berasal dari **dua jenis** sumber berikut:
 - a. 🔒 **Sumber Internal (Disediakan oleh Asisten Dosen)**
 - i. Data internal bersifat wajib dan akan disediakan dalam bentuk **CSV**.
 - ii. Kalian diminta untuk mengubah data tersebut menjadi **RDF** atau **Property Graph**, termasuk menyusun ontologi data internal yang sesuai.
 - b. 🔓 **Sumber Eksternal (Open-Access Knowledge Graph)**
Data eksternal diambil dari sumber-sumber Knowledge Graph terbuka, seperti:

- i. SPARQL Endpoint: [DBpedia](#), [WikiData](#), [YAGO](#), [KBpedia](#), [BIO2RDF](#), [Cultural KG](#), dan Open KG lainnya yang relevan dengan konteks data kalian.

Catatan: Ini yang utama

- ii. API: [GoogleKG](#) (hanya jika masih tersedia kuota di Google Cloud).

Catatan: Penggunaan API Knowledge Graph (misalnya GoogleKG) tidak dianjurkan secara berlebihan, karena dianggap terlalu mudah. Diperbolehkan hanya untuk **1–2 use case** sebagai pelengkap, bukan sumber utama data.

- iii. Open-KG yang menyediakan data tanpa endpoint query

Contoh: Hasil konstruksi KG dari suatu paper penelitian

Catatan: Boleh digunakan sebagai pelengkap dari point i.

c. Catatan

- i. Data eksternal diharapkan dapat menjadi sarana bagi kalian untuk mengeksplorasi dan memperkaya data internal dengan informasi tambahan yang relevan dari berbagai *Open Knowledge Base* yang sudah tersedia.
- ii. Sebagai bagian dari laporan proyek, kumpulkan **seluruh query** (misalnya SPARQL atau Cypher) yang kalian gunakan untuk **mengambil** dan **mengintegrasikan** data eksternal ke dalam Knowledge Graph kalian.

4. ● **Fitur-Fitur Utama Aplikasi**

a. **Searching**

Aplikasi diharapkan dapat menampilkan **fitur pencarian data** melalui antarmuka pengguna (GUI), seperti **halaman utama (homepage)** atau **kolom pencarian (search bar)**. Fitur ini memungkinkan pengguna untuk dengan mudah menemukan entitas dalam Knowledge Graph, ibaratnya seperti fungsi *Ctrl + F* atau mesin pencari internal di dalam aplikasi.

b. **SPARQL/Cypher Query**

Aplikasi kalian diharapkan menyediakan fitur query bebas (*free-form query editor*) yang terhubung langsung dengan *endpoint* Knowledge Graph. Melalui fitur ini, pengguna dapat menuliskan dan menjalankan *query* secara langsung di antarmuka *Front-End*, mirip seperti *query console* pada Wikidata atau Neo4j Browser, seperti gambar di bawah ini.

The screenshot shows the Wikidata Query Service interface. At the top, there's a header with 'Wikidata Query Service', 'Examples', 'Help', 'More tools', and 'Query Builder'. Below this is a text area containing a SPARQL query:

```

1 SELECT ?anime ?animeLabel ?malID WHERE {
2   ?anime wdt:P4886 ?malID .
3   SERVICE wikibase:label { bd:serviceParam wikibase:language "en,ja,id". }
4 }
5 LIMIT 50

```

Below the query editor, there's a 'Table' view showing 50 results in 3029 ms. The table has three columns: 'anime', 'animeLabel', and 'malID'. The first few rows are:

anime	animeLabel	malID
Q105144908	Cowboy Bebop	1
Q123122452	Ashtamuze Bally	102
Q1050738	Milennium Actress	1033
Q123487938	Career, Calendar Legend	104
Q123653653	Boys Be...	105
Q10485007	Revere no Tensai Bakabon	10607

Fitur ini memungkinkan pengguna untuk:

- Mengeksplorasi data secara mandiri melalui SPARQL query (untuk Knowledge Graph berbasis RDF) atau Cypher query (untuk Property Graph).
- Menampilkan hasil query secara interaktif, misalnya dalam bentuk tabel (seperti contoh pada gambar di atas), daftar entitas, atau visualisasi sederhana.

Dengan demikian, pengguna tidak hanya dapat mencari data melalui *search bar*, tetapi juga melakukan eksplorasi pengetahuan yang lebih mendalam menggunakan bahasa *query* yang sesuai dengan jenis *graph* yang digunakan.

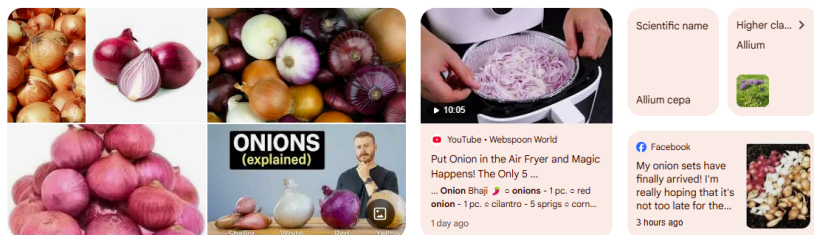
c. Info Box

Aplikasi juga perlu menampilkan **Info Box**, yaitu tampilan sederhana berisi **atribut** dan **informasi lengkap** dari suatu **entitas** dalam Knowledge Graph. Ketika pengguna mengklik hasil dari fitur *Search*, aplikasi akan menampilkan halaman Info Box dari entitas yang dipilih. Tampilan dapat dibuat minimalis, namun harus **informatif**.

Untuk gambaran lebih jelas mengenai struktur dan tampilannya, silakan merujuk pada **gambar dan penjelasan di bawah ini**.

Onion

Vegetable



Pada dataset **MBG (Meal Balance Grub)**, terdapat sebuah entitas bernama 🧅 [Onion](#) atau 🧄 bawang bombay. Ketika kalian melakukan pencarian di Google, akan muncul **infobox** seperti yang terlihat pada gambar di atas atau di samping.

Infobox tersebut sangat membantu dalam memahami makna dan konteks suatu entitas, karena biasanya berisi berbagai atribut penting, antara lain:

- Gambar
- Kelas
- Abstraksi/Penjelasan umum
- Sumber
- Informasi yang sering dicari, seperti fakta nutrisinya
- Dan rekomendasi pencarian lainnya

Jika diperhatikan lebih lanjut, dalam infobox tersebut juga terdapat entitas lain yang dapat ditelusuri lebih dalam. Sebagai contoh, terdapat entitas [USDA](#), yaitu sumber data untuk fakta nutrisi si bawang bombay.

Selain itu, ada pula entitas lain yang berhubungan secara semantik dengan entitas bawang tersebut, seperti [Allium](#), yang merupakan genus atau klasifikasi taksonomi dari Onion.

About

The onion, also known as the bulb onion or common onion, is a vegetable that is the most widely cultivated species of the genus *Allium*. The shallot is a botanical variety of the onion which was classified as a separate species until 2011. The onion's close relatives include garlic, scallion, leek, and chives. [Wikipedia](#)

Scientific name: *Allium cepa*

Higher classification: [Allium](#)

[Feedback](#)

Nutrition facts

Sources include: [USDA](#)

Onions

40 Calories - 100 grams

Nutrient	Amount(g)	DV(%)
Total Fat	0,1 g	0%
Cholesterol	0 mg	0%
Sodium	4 mg	0%
Potassium	146 mg	4%
Total Carbohydrate	9 g	3%
Protein	1,1 g	2%

Percent Daily Values are based on a 2,000 calorie diet.
Your daily values may be higher or lower depending on your calorie needs.

People also search for



Tomato



Potato



Welsh onion



Cucumber

[See more](#) >

5. **Fitur-Fitur Advanced (Opsional / Bonus)**

Fitur-fitur berikut bersifat **opsional**, namun **sangat direkomendasikan** untuk kalian yang ingin mengasah kemampuan teknis lebih jauh, sekaligus menambah nilai pada **CV** kalian 📁 (terutama bagi yang tertarik pada bidang *AI Engineering*).

a. Contoh Fitur *Advanced* yang Dapat Diimplementasikan (namun tidak terbatas pada)

i. **Vector Embedding**

Neo4j, sudah memiliki dukungan native untuk penyimpanan dan pencarian berbasis vektor, namun kalian juga dapat menggunakan alternatif *vector database* lainnya.

Contoh: Menyimpan embedding *entity* pada Neo4j (yang memiliki dukungan *native vector index*), atau menggunakan alternatif seperti Qdrant, Pinecone, atau FAISS untuk integrasi eksternal. Fitur ini dapat digunakan untuk *semantic search* dan *re-ranking* hasil pencarian.

ii. **Reranking**

Mengurutkan hasil pencarian berdasarkan *semantic similarity* atau relevansi kontekstual.

Contoh: Setelah hasil dicari berdasarkan nama entitas, gunakan model seperti **cross-encoder/ms-marco-MiniLM-L-6-v2** untuk menentukan relevansi paling tinggi terhadap *query* pengguna.

iii. **Semantic Search**

Pencarian berbasis makna (*meaning-aware search*), bukan sekadar pencocokan kata kunci.

Contoh: Saat pengguna mencari kata “**mobil**”, sistem juga dapat menampilkan hasil yang memuat kata “**kendaraan**” atau “**otomobil**”, karena mengenali bahwa kata-kata tersebut memiliki **makna sinonim**. Begitu pula, pencarian untuk “**dosen**” dapat menampilkan entitas yang berisi kata “**pengajar**” atau “**instructor**”, meskipun istilah yang digunakan berbeda.

iv. **Question Answering**

Sistem tanya-jawab menggunakan bahasa alami (*natural language*), tanpa perlu menulis query manual.

Contoh: Pengguna menulis pertanyaan seperti “*Apa kandungan gizi dari bawang bombay?*”, dan sistem mengambil jawaban dari triple RDF atau relasi dalam Neo4j.

v. **Chatbot berbasis RAG**

Menggabungkan data dari Knowledge Graph dengan *language model* untuk menghasilkan jawaban yang kontekstual dan kaya informasi.

Contoh: Chatbot yang menjawab pertanyaan seputar entitas pada graph (misalnya resep makanan, nutrisi, atau klasifikasi tanaman) dengan merujuk pada data internal maupun eksternal (DBpedia/Wikidata).

- b. Catatan
- Pada Neo4J, terdapat fitur *native* untuk melakukan *Semantic Search*, jika kalian berhasil mengimplementasikan Vector Embedding. Sehingga, sangat disayangkan jika kalian tidak mengerjakan bagian yang ini, jika memakai Neo4J
 - Untuk **Vector Embedding** dan **Reranking**, kalian dapat menjalankan model *embedding* atau *reranker* di Google Colab atau Kaggle, yang menyediakan akses GPU gratis. Jika dataset kalian berukuran besar, cukup gunakan subset data untuk demonstrasi, atau jalankan serta demokan secara lokal khusus untuk fitur ini.
 - Untuk **RAG**, kalian dapat mendemonstrasikan fitur ini melalui Colab atau Kaggle, dengan data Knowledge Graph yang diambil secara langsung dari API backend aplikasi kalian. Dengan demikian, integrasi antara model RAG dan sistem Knowledge Graph tetap berjalan secara terhubung (*end-to-end*).

Further Reading Advance Feature

- Model Embedding (Tiny): <https://huggingface.co/Qwen/Qwen3-Embedding-0.6B>
- Model Reranker (Tiny): <https://huggingface.co/Qwen/Qwen3-Reranker-0.6B>
- [Semantic Search & RAG using Neo4J](#)

Daftar Dataset

Kalian disarankan untuk melihat dataset tersebut, dan membayangkan bentuk graf hasilnya seperti apa

No	Inisial	Judul	Link Dataset	Notes
1	MBG	Meal Balance Grub (Calorie Meal Planner)	https://www.kaggle.com/datasets/hugodarwood/epirecipes	Ingredient disimpan dalam one-hot-encoding
			https://www.kaggle.com/datasets/kkhandekar/calories-in-food-items-per-100-grams	
2	HEP	Historical Event and People: BC to AD, Zoro to Nero	https://www.kaggle.com/datasets/saketk511/world-important-events-ancient-to-modern	Data Internal banyak tapi kurang informatif, sila ambil data External yang banyak
			https://www.kaggle.com/datasets/mit/pantheon-project	
3	TC	Tourism Cuy	https://www.kaggle.com/datasets/aprabowo/indonesia-tourism-destination?select=tourism_wit_h_id.csv	

			https://www.kaggle.com/datasets/aprabowo/indonesia-tourism-destination?select=tourism_rating.csv	
4	HPD	Historical Pandemic to Deaths Relations	https://www.kaggle.com/datasets/mtkaggleteam/pandemics-in-world?select=1-+the-number-of-cases-of-infectious-diseases.csv https://www.kaggle.com/datasets/mtkaggleteam/pandemics-in-world?select=2-+the-worlds-number-of-vaccinated-one-year-olds.csv	
5	AD	Anime Dataset 2025	https://www.kaggle.com/datasets/syahrulapriansyah2/myanimelist-2025 https://www.kaggle.com/datasets/syahrulapriansyah2/anime-characters/	<p>Data hasil <i>scraping</i>, <i>compatible</i> jika ingin ingin menambahkan fitur <i>semantic search</i> karena ada kolom <i>description</i> yang kaya akan makna dan juga <i>compatible</i> jika ingin diintegrasikan dengan WikiData</p> <p>Dua dataset ini saling berhubungan</p> <p>(jangan lupa upvote datasetnya di kaggle yaa👉karena saya udah cape ngescrapenya :V)</p>
6	HA	Historical Art: Ranging from 3rd to 19th century	https://www.kaggle.com/datasets/ansonnnnn/historic-art https://www.kaggle.com/datasets/ikarus777/best-artworks-of-all-time?select=artists.csv	<p>Download csv-nya saja</p> <p>Download csv-nya saja, fokus dengan biografi untuk fitur embedding (Skip kalau tanpa embedding)</p>

Lini Masa Pengerjaan

No	Aktivitas	Deadline
1	Tahap 1: Pemilihan Dataset	November 8, 2025
2	Tahap 2: Final	Desember 5, 2025
3	Tahap 3: Demo	Desember 6, 2025

Tahap 1 (Pembuatan Kelompok dan Pemilihan Dataset)

Pada tahap ini, Anda diminta untuk membentuk satu kelompok yang terdiri dari **maksimal empat orang**, silakan daftarkan kelompok anda di sini: [\[Link Pendaftaran Kelompok\]](#). Selain itu, setiap kelompok menentukan dataset dan mendaftarkannya pada tautan berikut [TBD]. Jika suatu kelompok submit lebih dari sekali, maka akan dipilih submit-an yang terbaru.

Tahap 2 (Final)

Pada tahap ini, setiap kelompok diwajibkan menyelesaikan **aplikasi final**, **laporan lengkap**, serta **berkas pendukung** lainnya. Seluruh berkas final akan dikumpulkan melalui SCELE.

A. Komponen Laporan Final

Laporan harus memuat:

1. **Judul Proyek dan Dataset yang dipilih**
2. **Nama Kelompok**, nama anggota, dan masing-masing **NPM**
3. **Deskripsi aplikasi** serta **tautan aplikasi (deploy URL)**
4. **Penjelasan Ontologi/Vocabulary**, termasuk:
 - daftar class dan property
 - relasi antar class
 - *graph visualization* (opsional namun direkomendasikan)
5. **Penjelasan Data RDF/Property Graph**
 - bagaimana data internal dikonversi
 - struktur *triple store* / *Neo4j store*
6. **Integrasi Data Internal & Eksternal**
 - data eksternal apa yang diambil
 - dari sumber apa (contoh: Wikidata, DBpedia, YAGO, dsb.)
 - relasi apa yang ditambahkan ke *graph* internal
 - penjelasan mekanisme *enrichment*
7. **Kumpulan Query Data Eksternal**
 - seluruh SPARQL atau Cypher yang digunakan untuk mengambil data eksternal
8. **Penjelasan Arsitektur Aplikasi dan API**
 - arsitektur backend-frontend
 - *endpoint* utama
 - penyimpanan data
9. **Penjelasan masing-masing fitur aplikasi**, dilengkapi screenshot
 - *Searching*
 - SPARQL/Cypher Console
 - Info Box
 - Fitur lain yang diimplementasikan
10. **Penjelasan Fitur Advanced** (jika dikerjakan)
 - *vector embedding*
 - *semantic search*
 - *reranking*
 - QA atau RAG

Format laporan disesuaikan dengan format laporan pada Tahap 2. Format penamaan dokumen adalah: **[NamaKelompok]_ProyekKelompok_Final_Laporan.pdf**

contoh

SemWeebs_ProyekKelompok_Final_Laporan.pdf

B. Slide Presentasi

Isi ringkas yang dipakai untuk demo & presentasi 5-15 menit:

- Overview proyek
- Ontologi
- Arsitektur aplikasi
- Fitur utama
- Integrasi data eksternal
- Fitur advanced (jika ada)

Note: setelah *deadline* Slide masih boleh di-*update* sampai dengan presentasi namun tidak perlu dikumpulkan ulang

Format: **[NamaKelompok]_ProyekKelompok_Final_Slide.pdf**

C. Source Code

ZIP berisi:

- backend API
- frontend aplikasi
- script konversi data internal (jika ada)
- notebook embedding (jika ada)
- dll yang dirasa perlu untuk dicantumkan

Format ZIP: **[NamaKelompok]_ProyekKelompok_Final_Code.zip**

Demo

Pelaksanaan demo proyek final wajib dihadiri oleh setiap anggota kelompok. Demo akan dilaksanakan dengan ketentuan sebagai berikut:



Tanggal: **6 Desember 2025**



Sesi Waktu:

- **Sesi 1: 08.30 – 12.00**
- **Sesi 2: 13.00 – 16.00**

🕒 Durasi per Kelompok:




- **20 menit presentasi**
- **10 menit sesi tanya jawab (QA)**

Catatan:

- Waktu 20 menit presentasi mencakup presentasi *slide* dan demo aplikasi.

- Durasi presentasi maksimal 20 menit sehingga **dilarang melebihi batas waktu** tersebut. Oleh karena itu, setiap kelompok dianjurkan berlatih sebelum presentasi agar tidak melebihi batas waktu 20 menit
- Pembagian waktu antara presentasi slide dan demo aplikasi diserahkan sepenuhnya kepada masing-masing kelompok.
- Link pembagian jadwal kelompok akan diumumkan melalui tautan berikut:
https://docs.google.com/spreadsheets/d/1ANBWWWh-AHltn1d_RE9EkkfYon0AG_Ji5xnFpSavAEwU/edit?gid=0#gid=0

Apabila terdapat anggota kelompok yang **berhalangan hadir pada seluruh slot yang tersedia**, silakan **menghubungi asisten dosen** untuk koordinasi lebih lanjut.

 Catatan penting saat Demo: Hanya presentasikan konsep besar dan penting.  Jangan habiskan waktu kalian dan kelompok setelahnya . Karena informasi detail sudah ada di laporan. Jika dirasa dosen perlu penjelasan lebih, akan disanggah atau ditanya saat QA.

Yang terpenting !!! Have Fun & Be Yourself !!! Anggap presentasi sebagai latihan: Waktu mepet tapi materinya serapat

Komponen Penilaian (Draft, subjected to change)

No.	Komponen	Persentase
2	Laporan Final <ul style="list-style-type: none"> • Kelengkapan dan kejelasan laporan • Ketepatan ontologi/vocabulary dan triple store yang dibangun • Implementasi Triple Store atau Property Graph Database 	30%
3	Demo Program <ul style="list-style-type: none"> • API berfungsi dengan baik • Pengelolaan data <i>Local</i> dan <i>Remote</i> • Kesesuaian fitur aplikasi (searching, query, info box) • Deploy Aplikasi ke Server 	70%
4	Bonus: <ul style="list-style-type: none"> • Keindahan Front End • Data External, yang melebihi ekspektasi atau sangat memperkaya data Internal • Fitur Advance 	BONUS

Tools

Berikut adalah beberapa tools yang dapat ditelusuri dan digunakan dalam pengerjaan proyek akhir.

- Working with RDF Data
 - Python: [RDFLib](#)
 - Java : [Apache Jena](#)
 - Golang: [rdf2go](#)
 - C/C++ : [Redland](#)
- Triple Store
 - [Apache Jena TDB](#)
 - [Apache Jena Fuseki](#)
 - [GraphDB](#)
 - [Virtuoso](#)
 - [Blazegraph](#)
- Property Graph
 - [Neo4J](#)