Perbandingan antara Redis dan Graph database (ArangoDB)

Redis dan ArangoDB masing-masing memiliki keunggulan dan kelemahan tergantung pada kebutuhan pengguna tertentu, keduanya adalah pilihan yang sepertinya cukup bagus dan banyak penggunanya. Redis dan ArangoDB berbeda dalam hal arsitektur, jenis penyimpanan data, dan fitur, tetapi keduanya cocok untuk penggunaan yang berbeda. Disini saya, akan membandingkan dan mengontraskan kedua database ini berdasarkan beberapa hal. Ini termasuk arsitektur, fitur, dan contoh penggunaan sintaksis untuk operasi dasar.

a. **Arsitektur dan Tipe Penyimpanan Data**

Redis adalah basis data in-memory (di memori) yang terkenal dengan kecepatannya yang luar biasa. menyimpan semua data di RAM, yang memungkinkan waktu respons yang sangat cepat untuk operasi baca dan tulis. Namun, Redis juga dikenal sebagai penyimpanan data sementara atau cache, yang digunakan untuk aplikasi yang membutuhkan akses cepat dan latensi rendah. Struktur data seperti string, set, hash, dan list dapat digunakan dalam berbagai aplikasi dengan dukungan Redis.

Berbeda dengan Redis, **ArangoDB** adalah basis data multi-model yang mendukung tiga model data utama: dokumen, graf, dan key-value. Selain itu, redis menggunakan model dokumen untuk menyimpan data dalam format **JSON** dan memungkinkan pengelolaan data dalam bentuk graf menggunakan konsep seperti node dan edge. membuatnya sangat fleksibel dan cocok untuk berbagai jenis aplikasi, termasuk yang memerlukan analisis graf atau pencarian yang lebih kompleks.

b. **Skalabilitas dan Kinerja**

Clustering dan partitioning memungkinkan Redis untuk mencapai skalabilitas horizontal, yang sangat penting untuk aplikasi dengan tingkat lalu lintas tinggi yang memerlukan skalabilitas yang efisien. Oleh karena itu, Redis dapat menangani jutaan operasi per detik dengan latensi rendah, menjadikannya pilihan utama untuk aplikasi yang memerlukan kecepatan respons yang sangat cepat, sepersepuluh detik atau lebih.

Meskipun ArangoDB mendukung skalabilitas horizontal, ia lebih fokus pada integrasi berbagai model data dalam satu database. Ini membuatnya lebih cocok untuk aplikasi yang memerlukan analisis data terstruktur dan terhubung, seperti sistem rekomendasi berbasis graf atau jejaring sosial. Meskipun ArangoDB bekerja dengan baik, ia mungkin tidak secepat Redis dalam operasi baca/tulis dasar karena ia menangani lebih banyak masalah pengelolaan data.

c. **Keuntungan dan Kekurangan**

Keuntungan Redis:

- Kecepatannya tinggi karena disimpan didalam memori.
- Operasi yang sangat cepat untuk struktur data sederhana.
- Cocok untuk caching dan penyimpanan data sementara.
- Dukungan untuk berbagai jenis struktur data yang memudahkan implementasi.

Kekurangan Redis:

- Ada batasan kapasitas karena penyimpanan datanya di RAM.
- Kurang cocok untuk kueri kompleks atau hubungan antar data yang rumit.
- Tidak mendukung model data graf atau hubungan antar entitas secara native.

Keuntungan ArangoDB:

- Bagus untuk multi-model seperti dokumen, graf, dan key-value.
- Query kompleks dan analisis graf dapat dilakukan secara efisien.
- Fleksibilitasnya lebih besar untuk aplikasi yang membutuhkan struktur data yang lebih kaya.
- Skalabilitas horizontal yang baik dan integrasi berbagai model dalam satu database.

Kekurangan ArangoDB:

- Lebih kompleks untuk diatur dan dikelola daripada Redis.
- Kinerja mungkin lebih lambat dibandingkan Redis dalam operasi sederhana.
- Mungkin lebih cocok untuk kasus penggunaan jangka panjang, bukan caching atau penyimpanan sementara.

Jika saya harus memilih salah satu antara **Redis** dan **ArangoDB**, saya akan memilih **Redis.**

Alasan saya memilih Redis karena Redis lebih unggul dalam hal kecepatan dan kesederhanaan. Jika seandainya aplikasi saya membutuhkan performa tinggi dengan latensi rendah, terutama untuk operasi dasar seperti caching, antrian pesan, atau penyimpanan data sementara, Redis adalah pilihan yang lebih tepat. Redis dirancang untuk menangani jutaan operasi per detik dengan kecepatan sangat tinggi karena penyimpanan data di memori.

Redis sangat idea jika fokusnya pada efisensi dan kecepatan. Selain itu, Redis lebih mudah diatur dan dikelola, yang mempermudah pengembangan aplikasi dengna beban yang membutuhkan skalabilitas dan kecepatan tinggi.

Namun, jika aplikasi yang anda inginkan membutuhkan Query atau penyimpanan data yang lebih kompleks, ArangoDB lebih cocok. Tetapi karena saya ingin untuk penggunaan umum dan mengutamakan performa dan kecepatan maka saya memilih Redis.

CONTOH PENGGUNAAN REDIS DALAM DOCKER

Instalasi Redis didalam docker

1. **Membuat container Redis dalam docker :**

```
PS C:\Users\hp> docker exec -it redis-server redis-cli
127.0.0.1:6379>
```

2. **Jika sudah masuk kedalam container Redis:**

```
PS C:\Users\hp> docker exec -it redis-server redis-cli
127.0.0.1:6379>
```

Penggunaan redis dalam docker

1. Menyisipkan data menggunakan **SET :**

   SET adalah perintah untuk menyimpan data, dan user:1001 adalah kunci (kunci) yang digunakan untuk mengidentifikasi data yang kita simpan. kunci ini dapat menjadi nama apa pun, tetapi disarankan untuk menggunakan nama yang mendeskripsikan, seperti user:1001, di mana nilai yang kita simpan adalah nama pengguna dengan ID 1001. Jika data dengan key yang sama sudah ada, maka nilai yang lama akan diupdate dengan nilai baru.

```
127.0.0.1:6379> SET user:1001 "John Doe"
OK
127.0.0.1:6379> SET user:1002 "Johnig"
OK
127.0.0.1:6379>
```

```
127.0.0.1:6379> GET user:1001
"John Doe"
127.0.0.1:6379>
```

```
127.0.0.1:6379> GET user:1002
"Johnig"
127.0.0.1:6379>
```

```
127.0.0.1:6379> KEYS user:*
1) "user:1002"
2) "user:1001"
127.0.0.1:6379>
```

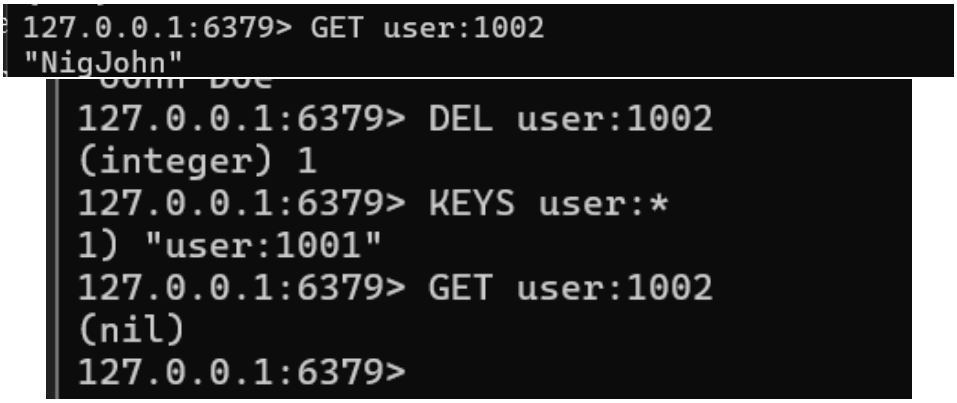
2. Memperbarui data menggunakan **SET** juga :

```
127.0.0.1:6379> SET user:1002 "NigJohn"
OK
127.0.0.1:6379> KEYS user:*
1) "user:1002"
2) "user:1001"
```

```
127.0.0.1:6379> GET user:1002
"NigJohn"
```

3. Menghapus data menggunakan **DEL** :

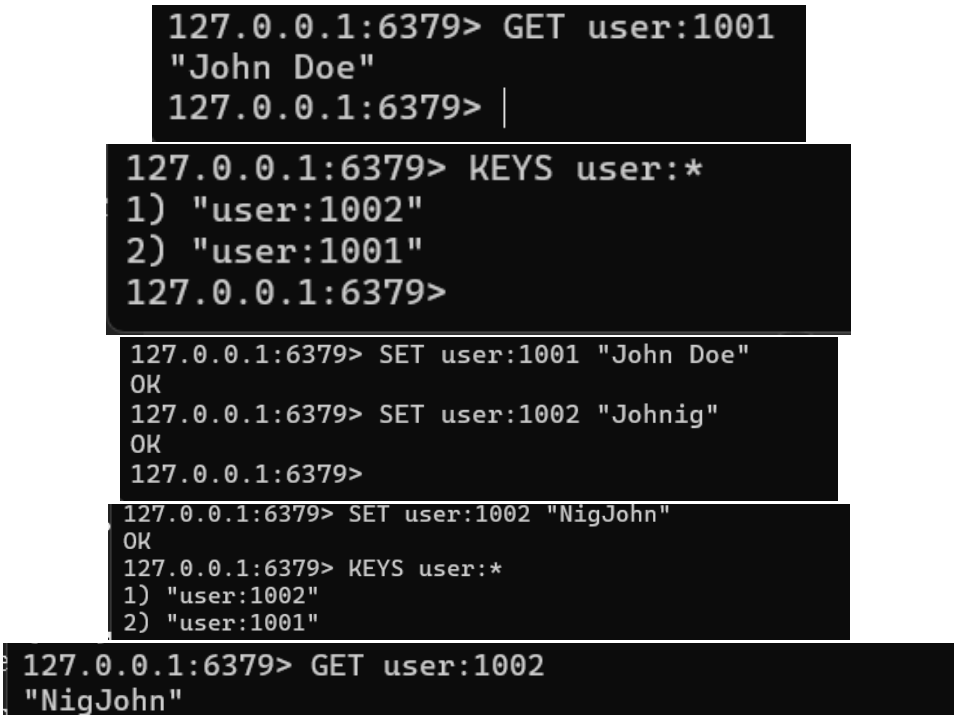
Perintah **DEL** digunakan untuk menghapus key dan nilainya. key user:1002 adalah key yang akan dihapus, dan setelah perintah ini digunakan, tombol user:1002 dan nilainya akan hilang dari Redis.



4. Melakukan Query dapat menggunakan **GET** berdasarkan nilay key, dan **KEYS** untuk mendapatkan user yang ada:

GET adalah perintah untuk mengambil nilai berdasarkan key yang ada.
user:1001 adalah key yang kita ingin ambil nilainya.
Redis akan mengembalikan nilai yang terkait dengan key tersebut, dalam hal ini, "Johnig" (karena sebelumnya kita sudah mengupdate menjadi NigJohn).

Perintah **KEYS** akan menampilkan semua key di Redis, dengan tanda bintang (*) sebagai wildcard yang berarti "semua key". Jika ada beberapa key di Redis, perintah ini akan mengembalikan daftar semua key tersebut.



Implemntasi penggunaan python untuk menampilkan data yang sudah ada dalam Redis

1. Pastikan untuk menginstall modulnya terlebih dahulu menggunakan **pip install redis.**
2. Buat kode untuk mengakses dan menampilkan data di Redis:

redis-implemntation.py

```
redis-implementation > redis-implemntation.py > ...
1   import redis
2
3   r = redis.Redis(host='localhost', port=6379, db=0)
4
5   user_data = r.get('user:1001')
6   if user_data:
7       print(f"Data user:1001: {user_data.decode('utf-8')}")
8   else:
9       print("Data tidak ditemukan")
10
11  user_data = r.get('user:100')
12  if user_data:
13      print(f"Data user:1001: {user_data.decode('utf-8')}")
14  else:
15      print("Data tidak ditemukan")
16
17  all_keys = r.keys('*')
18  print(f"Semua keys di Redis: {[key.decode('utf-8') for key in all_keys]}")
19
20  keys_user = r.keys('user:*')
21  print(f"Keys yang dimulai dengan 'user:': {[key.decode('utf-8') for key in keys_user]}")
22
```

3. Output kode diatas sebagari berikut :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS

PS C:\Users\hp\OneDrive\Dokumen\roni\semester 3 kuliah\keamanan basis data\redis-implementation> & C:
Drive/Dokumen/roni/semester 3 kuliah/keamanan basis data/redis-implementation/redis-implemntation.py"
Data user:1001: John Doe
Data tidak ditemukan
Semua keys di Redis: ['user:1002', 'user:1001']
Keys yang dimulai dengan 'user:': ['user:1002', 'user:1001']
PS C:\Users\hp\OneDrive\Dokumen\roni\semester 3 kuliah\keamanan basis data\redis-implementation>
```