

# KONSEP DASAR DAN RISIKO KEAMANAN NOSQL

---

RPL318 – KEAMANAN BASIS DATA  
TEORI

# Tujuan Pembelajaran

---

Mahasiswa mampu menerapkan praktis dari prinsip-prinsip keamanan pada basis data non-relasional

# Materi

---

1. NoSQL
2. Konsep Dasar Keamanan NoSQL
3. Risiko keamanan pada NoSQL
4. Kebutuhan keamanan di NoSQL

# NoSQL

---

---

## Non-relasional = NoSQL, Not only SQL

NoSQL pada dasarnya adalah istilah umum yang mencakup system basis data, antara lain:

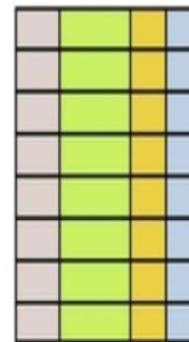
1. Memiliki model data selain tabel relasional konvensional,
2. Mendukung akses terprogram dan bahasa kueri non standar (selain SQL),
3. Dapat mengatasi evolusi skema atau dapat menangani data yang schemaless (tanpa skema),
4. Mendukung distribusi data dalam jaringan server secara desain.

# Penggunaan NoSQL dan SQL

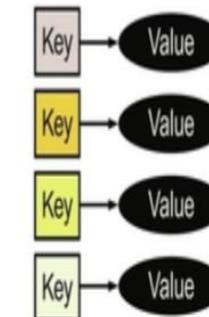
Basis data **NoSQL** dapat dikategorikan menjadi empat jenis organisasi data:

- 1. Key-Value (Kunci/Nilai):** Organisasi data yang sangat sederhana, dimana setiap catatan memiliki satu **atribut data kunci** yang diindeks untuk pencarian berkecepatan tinggi, dan sebuah "nilai", yang sebenarnya hanyalah objek untuk menyimpan data arbitrer yang strukturnya tidak diketahui oleh DBMS.
- 2. Graph (Grafik):** Penyimpanan data yang mendefinisikan **simpul** (node) dan **tepi** (edges) suatu grafik.

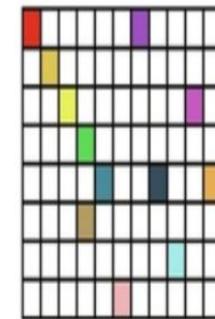
Relational



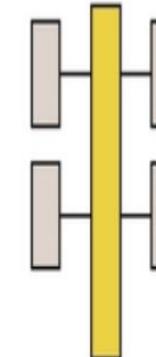
Key-Value



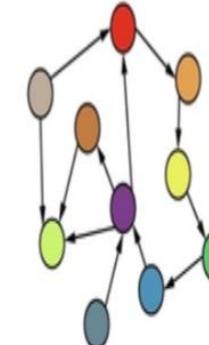
Column-Family



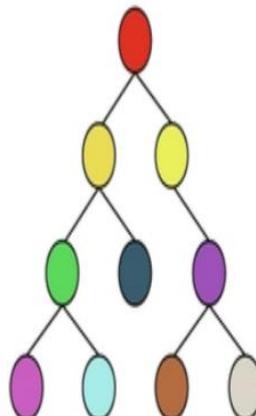
Analytical (OLAP)



Graph



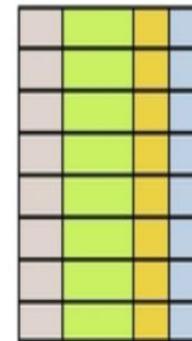
Document



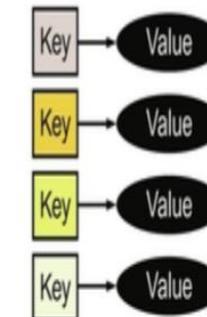
# Penggunaan NoSQL dan SQL

- 3. Column-Family (Berbasis Kolom):**  
Penyimpanan data sebagai **kolom-kolom** sebuah tabel, seperti dalam database relasional baris-baris suatu tabel.
- 4. Document (Dokumen):** Penyimpanan data dalam bentuk yang mirip dengan dokumen teks biasa; biasanya menggunakan banyak **penyusunan bersarang (nesting)**.

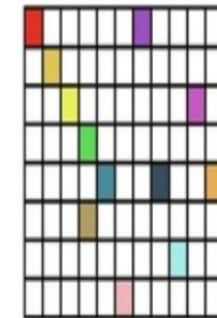
Relational



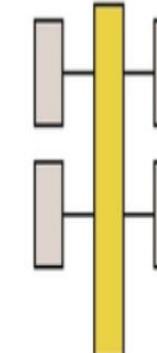
Key-Value



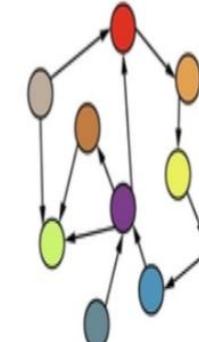
Column-Family



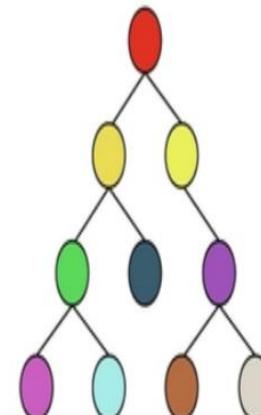
Analytical (OLAP)



Graph



Document



# Tools dari NoSQL berdasarkan kategori

## 1. Key-Value

Tools :Redis, Riak, DynamoDB(in key-value mode), Oracle NoSQL

Use case: session storage, user preferences, shopping carts

## 2. Graph

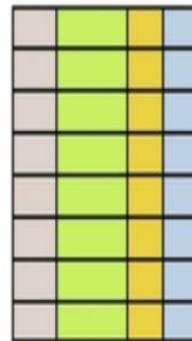
Tools: Neo4j, ArangoDB, Amazon Neptune, OrientDB, YarchData, AllegroGraph

Use case: Supply chain management

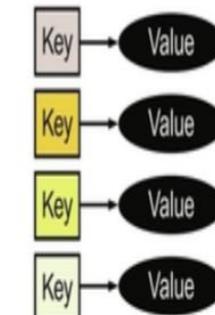
## 1. Column-Family

## 2. Document

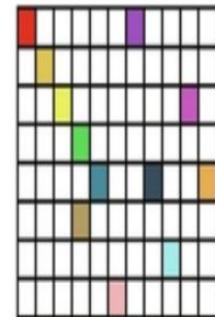
## Relational



## Key-Value



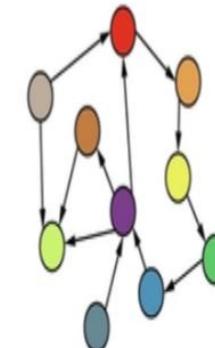
## Column-Family



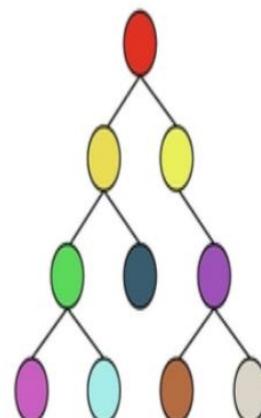
## Analytical (OLAP)



## Graph



## Document



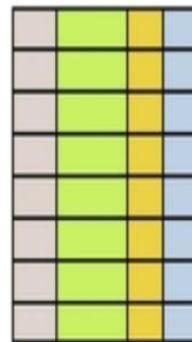
# Tools dari NoSQL berdasarkan kategori

## 3. Column-Family

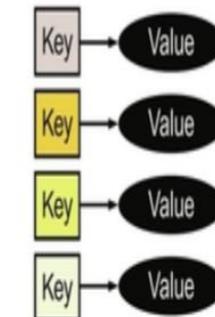
Tools: Apache Cassandra, Hbase

Use case: Time series data, IoT, Analytics workloads

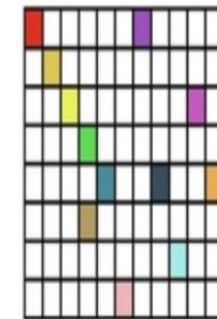
Relational



Key-Value



Column-Family

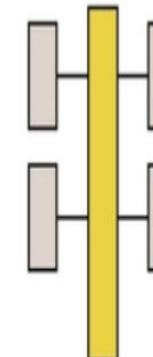


## 4. Document

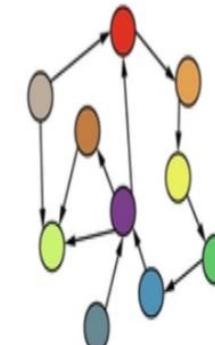
Tools: MongoDB, CouchDB

Use case: Product data, blogs, user profiles

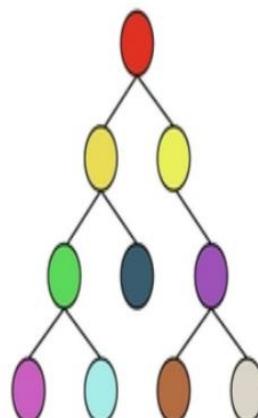
Analytical (OLAP)



Graph



Document



# Perbandingan NoSQL dan SQL

<b>SQL Databases</b>	<b>NoSQL Databases</b>
Centralized applications (ERP, CRM, SCM, etc.)	Decentralized applications (web, mobile, IoT, etc.)
Structured data and moderate data volumes	Multistructured data and high data volumes
High availability	Continuous availability
Moderate velocity data	High velocity data from sensors, machines, phones, etc.
Limited number of data sources	Multiple data sources from many distributed locations
Complex and nested transactions	Simple transactions
Vertical scalability	Horizontal scalability for high performance
Static schema	Dynamic schema

# Konsep Dasar Keamanan NoSQL

---

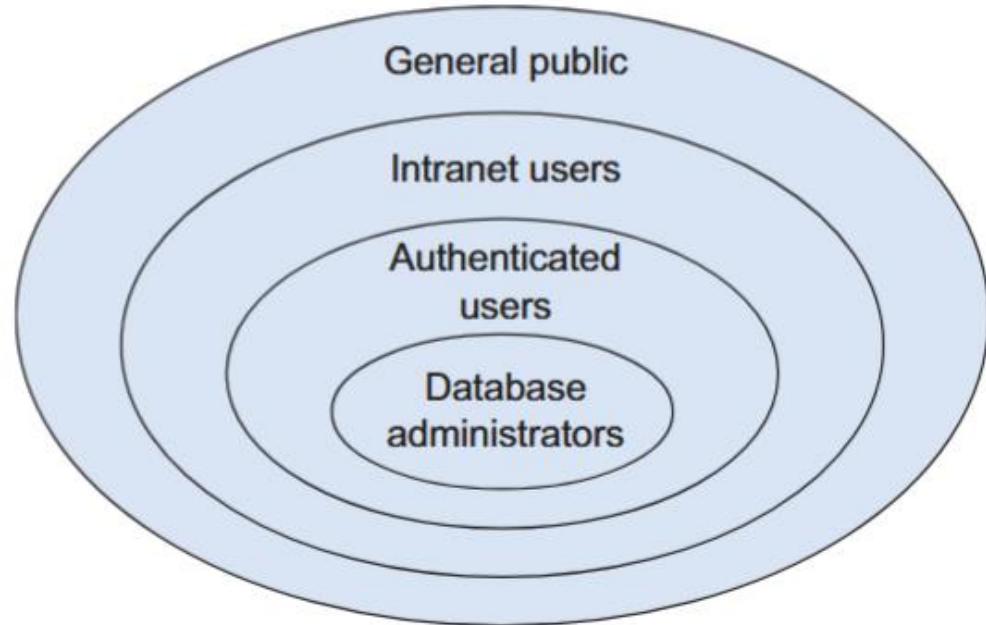
---

Kebutuhan akan basis data NoSQL muncul karena **keterbatasan RDBMS** dalam menghadapi tuntutan aplikasi modern yang cepat atau terprediksi untuk kebutuhan baik struktur data, volume data, maupun kecepatan akses yang ekstrem.

Basis data NoSQL dirancang untuk menangani himpunan data besar guna memenuhi persyaratan Analitik Data (*Data Analytics*) dan penekanan yang lebih kecil diberikan pada keamanan SELAMA FASE DESAIN. Sebagian besar basis data NoSQL tidak menyediakan fitur keamanan yang tertanam (*embedded*) di dalam basis data, dan developer / DBA harus menyisipkan solusi terkait keamanan di middleware (*perangkat lunak perantara*).

# Konsep model keamanan basis data NoSQL

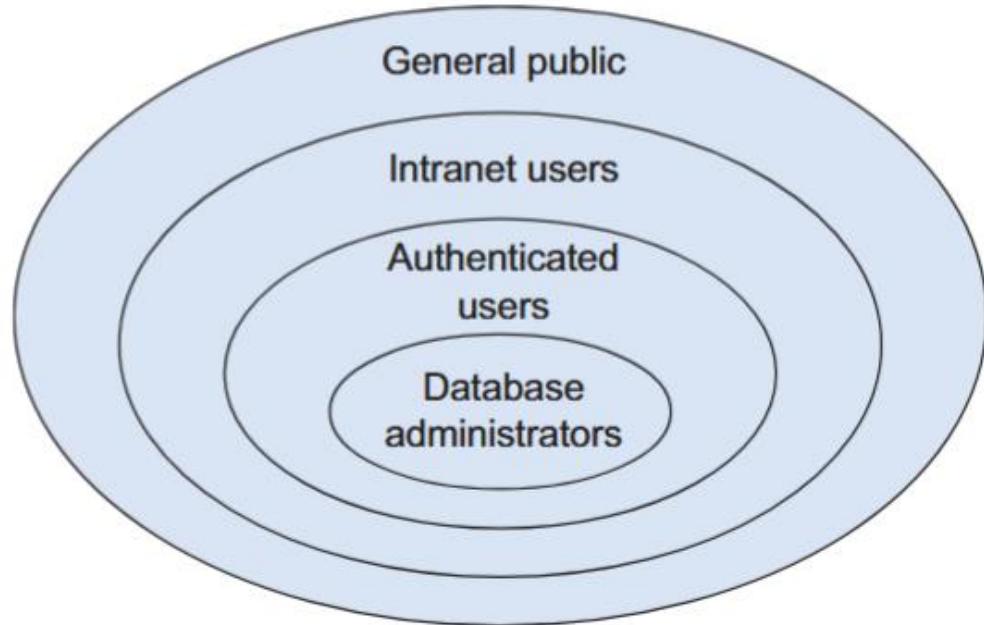
---



1. General Public (Publik Umum),  
Lapisan terluar: siapapun melalui internet yang bisa mengakses sebuah situs publik.
2. Intranet Users (Pengguna Intranet),  
Pengguna internal di jaringan perusahaan (seperti penggunaan LAN), biasanya karyawan yang sudah divalidasi oleh sistem jaringan internal.

# Konsep model keamanan basis data NoSQL

---



3. Authenticated Users (Pengguna Terotentikasi), Pengguna yang memiliki kredensial login (username dan password) untuk mengakses sistem atau basis data tertentu.
4. Database Administrators (Administrator Basis Data), Lapisan paling dalam, dengan hak akses penuh (full privileges) untuk mengatur, memelihara, dan melindungi seluruh sistem basis data.

# Risiko keamanan pada NoSQL

---

---

Setiap Perusahaan / organisasi yang membangun sebuah basis data dapat memilih untuk penempatan keamanan pada **Tingkat aplikasi atau pada Tingkat basis data**, namun harus dengan pertimbangan yang matang yang dilihat dari keuntungan penempatannya. Risiko keamanan dipertimbangkan dan tentukan metode maupun manfaat yang paling sesuai.

# Keuntungan keamanan Tingkat aplikasi

---

1. Kinerja basis data lebih cepat.
2. Penggunaan ruang disk lebih sedikit.
3. Kontrol tambahan melalui API dengan restricted APIs.

# Keuntungan Keamanan Tingkat Basis Data

---

1. Konsistensi kebijakan keamanan.
2. Kemampuan untuk melakukan pembuatan reporting baik rutinitas ataupun mendadak.
3. Audit terpusat (centralized audit).

# Penentuan tools

---

Pedoman untuk menentukan jenis tools untuk basis data NoSQL untuk proyek, yaitu:

- versi open source. Rekomendasi untuk membeli services ataupun versi enterprise / komersil Dimana terdapat fitur keamanan yang tidak ada di versi dasarnya seperti journal logs, disaster recovery log shipping, backup, dan restore.
- Biaya total system. Berapa DBA maupun database developer yang akan dibutuhkan dalam pengelolaan data.
- Prosedur maupun konfigurasi dari keamanan jenis tools dari NoSQL

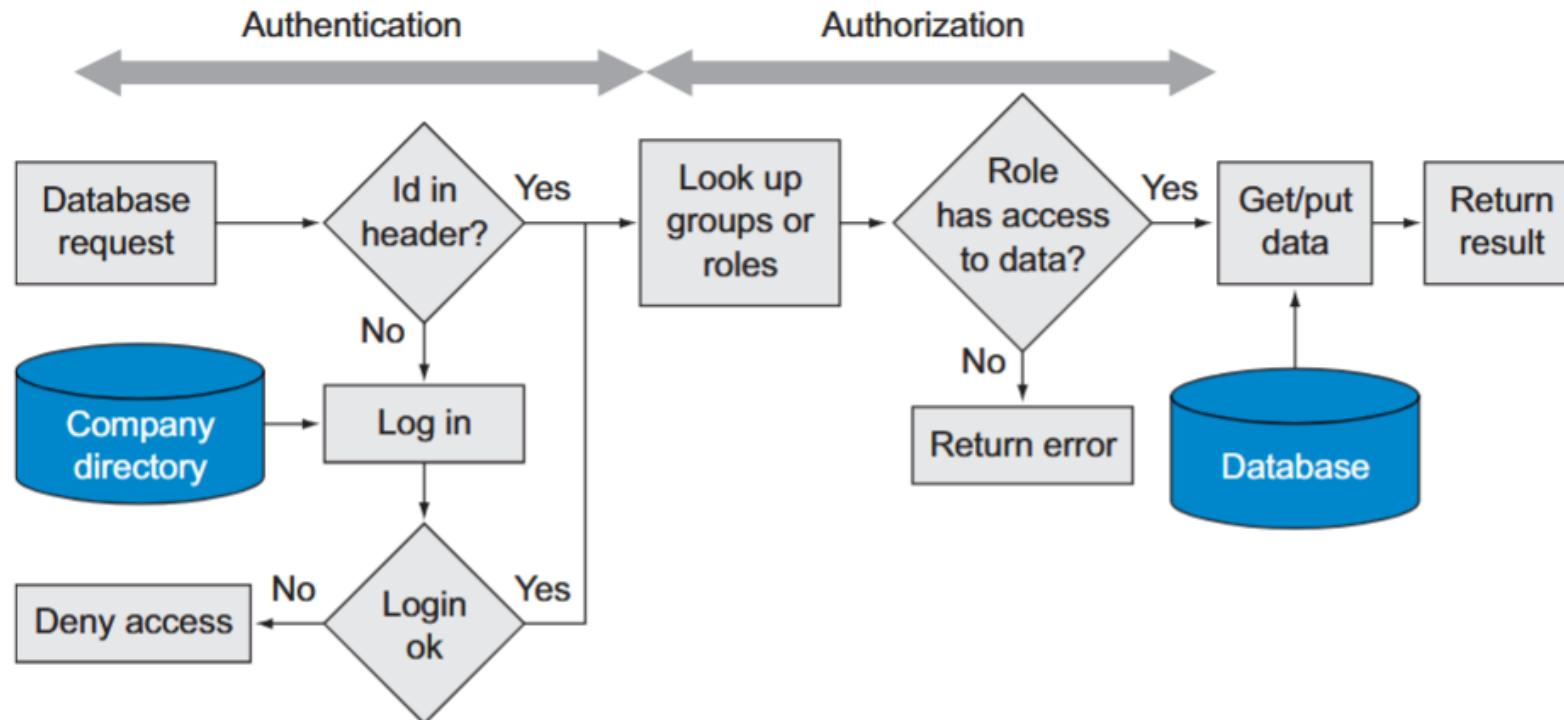
# Kebutuhan keamanan di NoSQL

---

AUTHENTICATION, AUTHORIZATION, AUDIT, ENCRYPTION

# Authentication

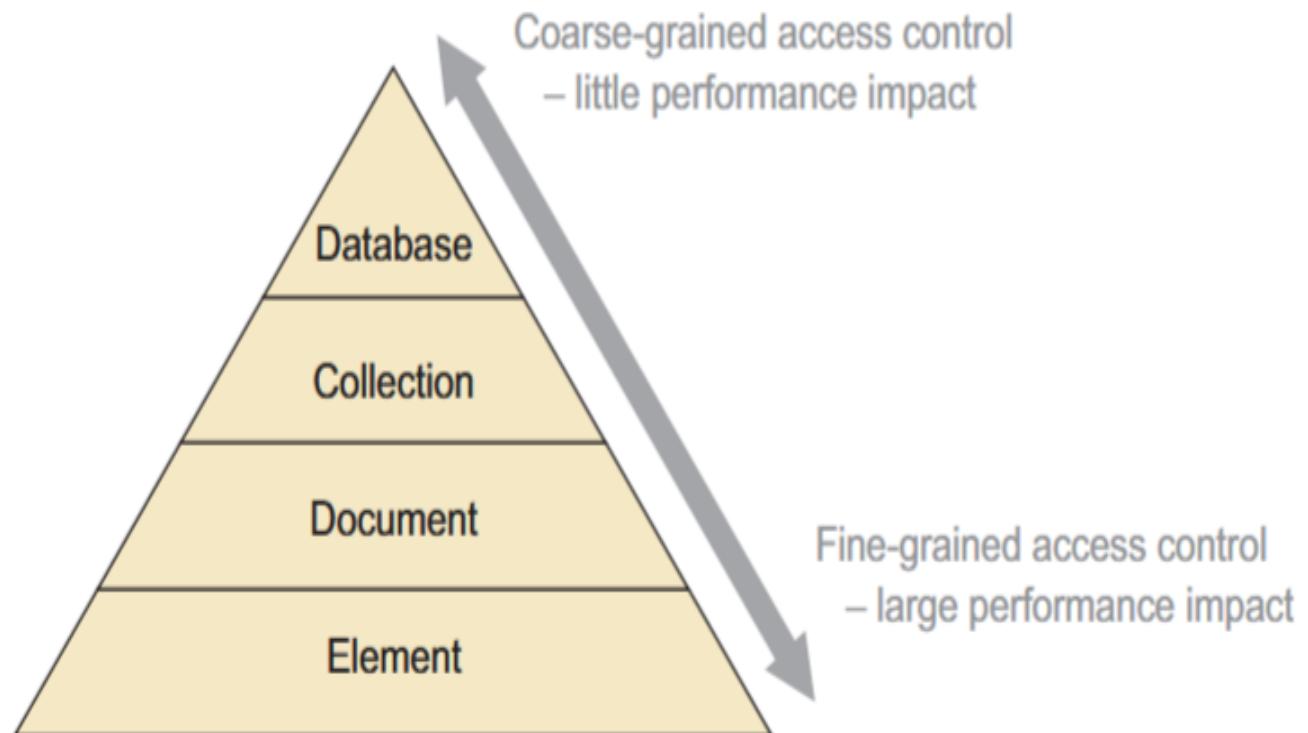
Proses validasi query berbasis web untuk NoSQL:



# Authorization

---

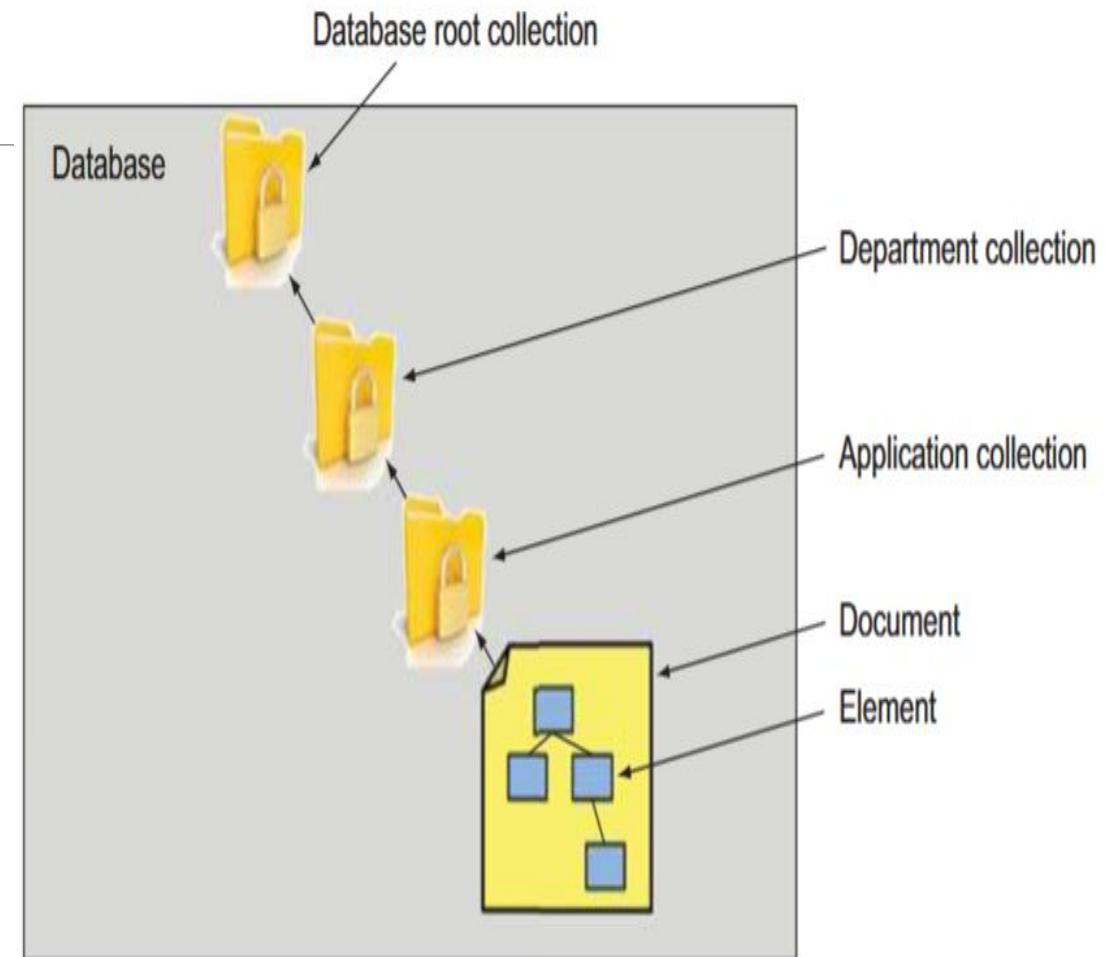
Sebelum membangun basis data yang NoSQL, pertimbangkan sesuai dengan kebutuhan pada kategori NoSQL.



---

Pada penyimpanan dokumen seperti Kumpulan folder berlapis-lapis dengan kuncinya sendiri.

Jadi untuk mengakses elemen dari dokumen tertentu dibutuhkan mengakses berbagai folder.



# Next!

---

- Audit
- Encryption