

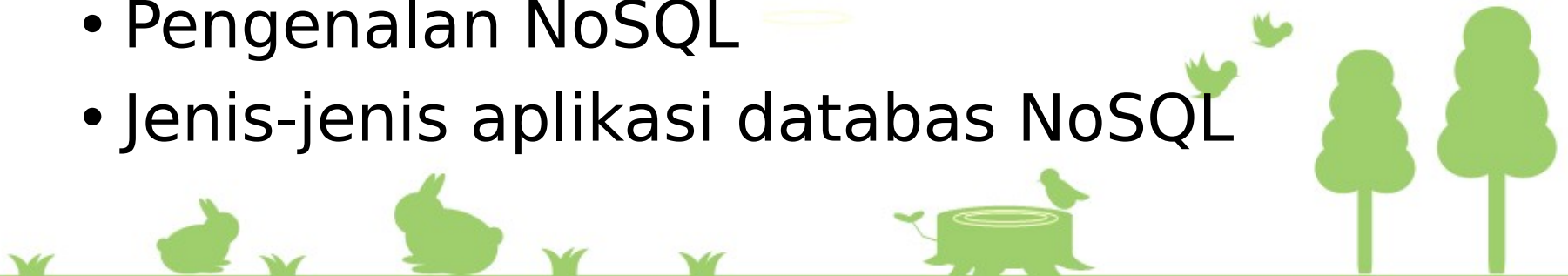
# SQL dan NoSQL

Big Data Analytics



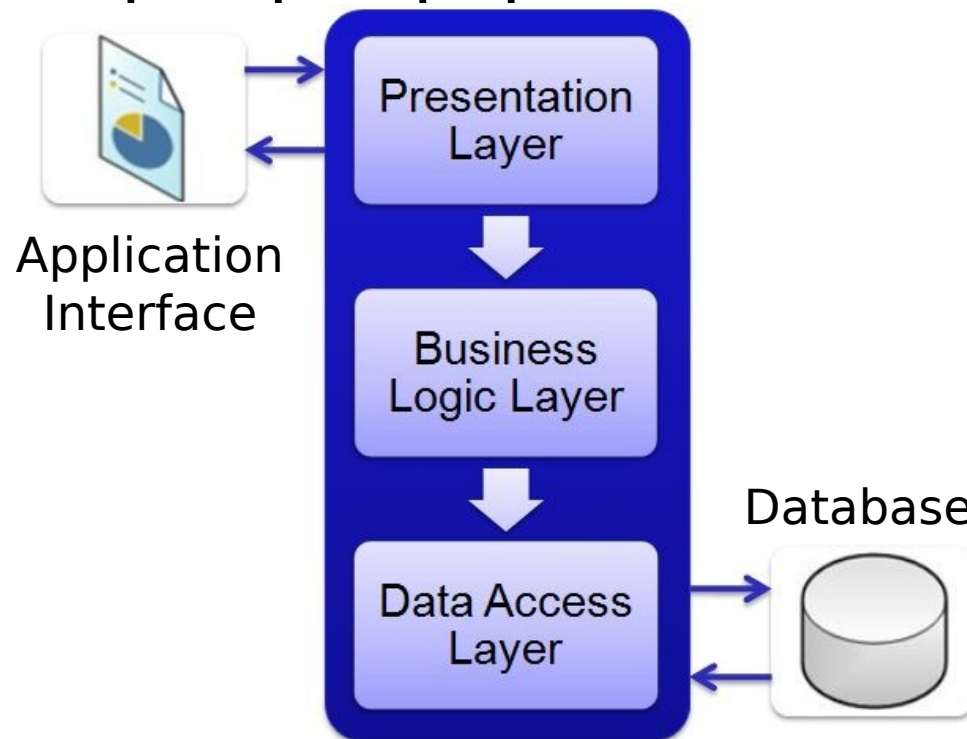
# Daftar Isi

- Pendahuluan
- Pengenalan RDBMS
- Penggunaan SQL pada RDBMS
  - Data Definition Language
  - Data Manipulation Language
- Contoh aplikasi RDBMS
- Pengenalan NoSQL
- Jenis-jenis aplikasi databas NoSQL



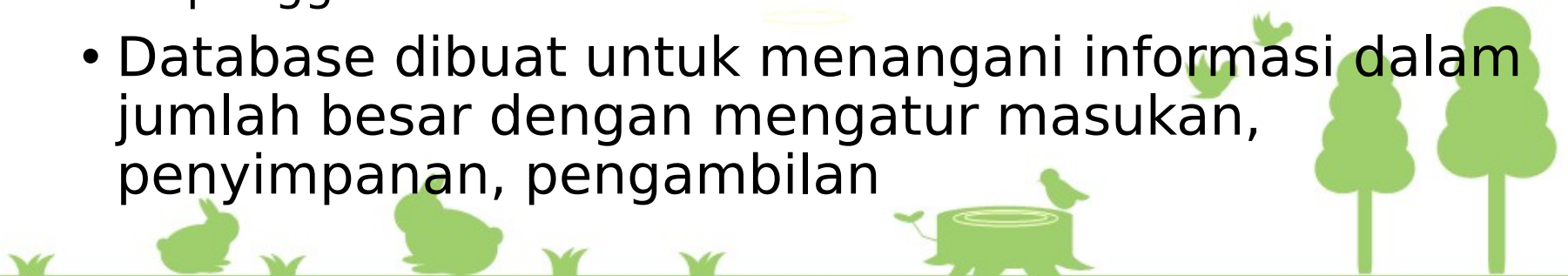
# Pendahuluan (1)

- Aplikasi-aplikasi modern umumnya menggunakan arsitektur tiga layer seperti pada gambar



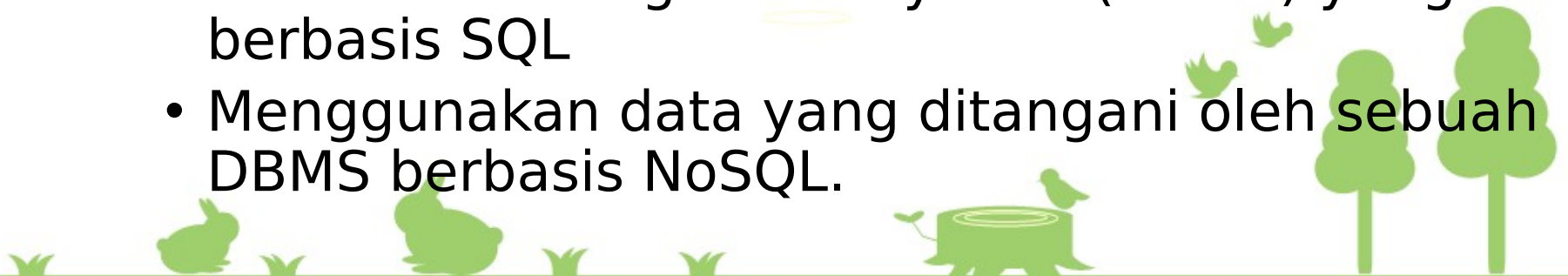
# Pendahuluan (2)

- Dari gambar arsitektur tersebut, terlihat bahwa penyimpanan basis data merupakan salah satu komponen yang umumnya ada pada aplikasi modern.
- Terminologi yang terkait dengan penyimpanan basis data:
  - Database: Koleksi data yang sudah terorganisir.
  - DBMS – Database Management System: Sebuah paket aplikasi yang mengontrol pembuatan, perawatan dan penggunaan sebuah database.
- Database dibuat untuk menangani informasi dalam jumlah besar dengan mengatur masukan, penyimpanan, pengambilan



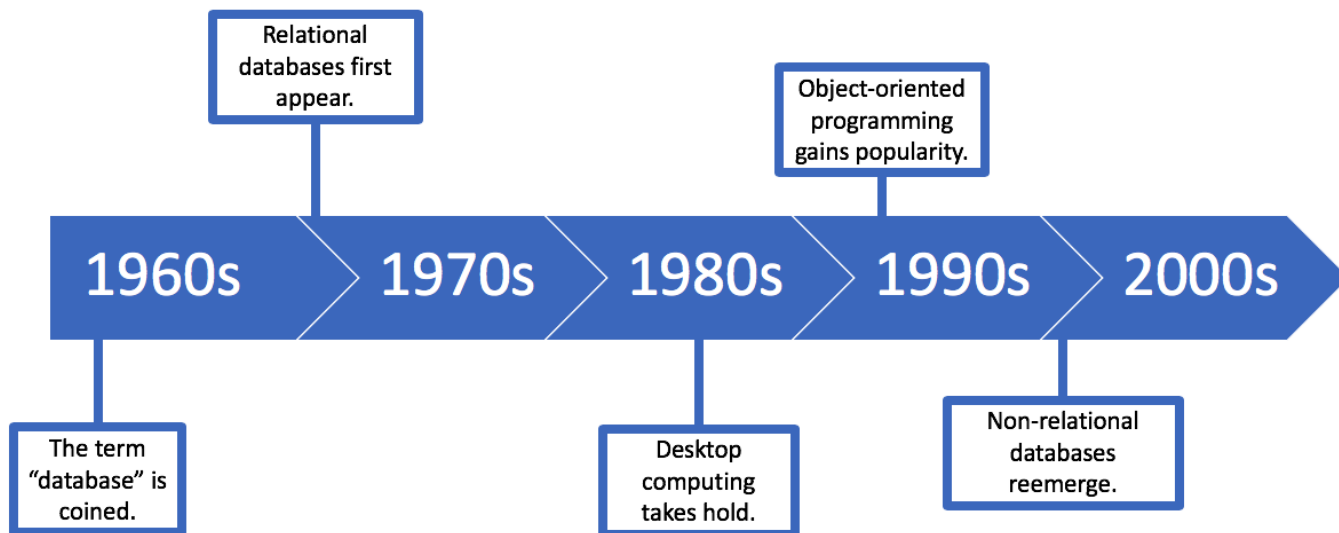
# Pendahuluan (3)

- Penyimpanan basis data memiliki sejarah yang panjang:
  - Menggunakan struktur data yang terpasang di dalam sumber kode aplikasi.
  - Menggunakan file eksternal untuk menyimpan struktur data di atas.
  - Menggunakan data yang ditangani oleh sebuah database management system (DBMS) yang berbasis SQL
  - Menggunakan data yang ditangani oleh sebuah DBMS berbasis NoSQL.



# Pendahuluan (4)

- Perkembangan database ditunjukkan seperti pada gambar berikut:



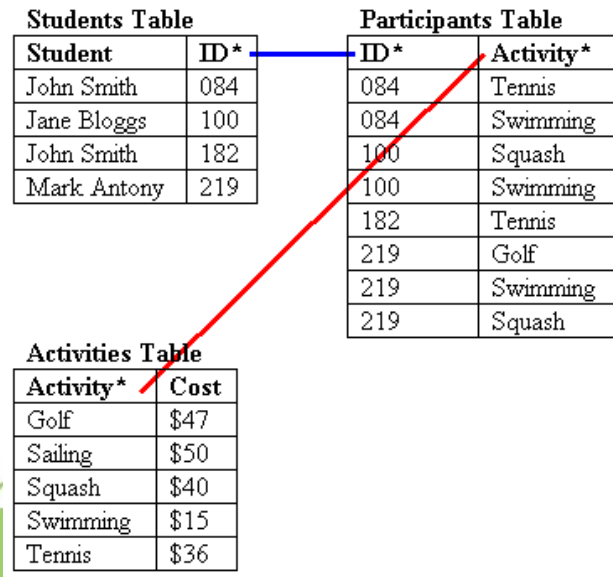
# Pendahuluan (5)

- Dalam sesi ini, akan didiskusikan implementasi SQL dan NoSQL DBMS untuk big data.



# Pengenalan RDBMS (1)

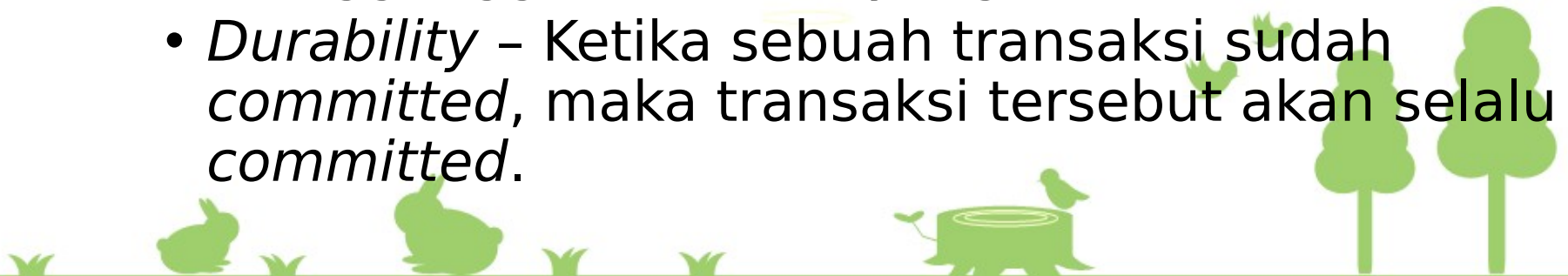
- RDBMS merupakan singkatan dari Relational Database Management System.
- Data pada RDBMS disimpan dalam bentuk tabel-tabel dan terdapat keterhubungan (*relationship*) diantara tabel-tabel tersebut.





# Pengenalan RDBMS (2)

- Fokus utama RDBMS adalah ACID:
  - *Atomicity* – Setiap transaksi adalah atomic. Jika satu bagian mengalami kegagalan, maka keseluruhan transaksi akan gagal.
  - *Consistency* – Setiap transaksi adalah sekumpulan aturan yang konsisten.
  - *Isolation* – Tidak ada transaksi yang dapat mengganggu transaksi yang lain.
  - *Durability* – Ketika sebuah transaksi sudah *committed*, maka transaksi tersebut akan selalu *committed*.




# Pengenalan RDBMS (3)

- Fitur-fitur yang terdapat pada RDBMS:
  - Identitas Data diatur dalam bentuk kolom-kolom tabel dan setiap data tersimpan dalam bentuk baris dari tabel.
  - Data tersusun dalam bentuk normalisasi.
  - Didukung secara luas.
  - Metode akses data yang konsisten menggunakan SQL.



# Pengenalan SQL (1)

- SQL adalah kepanjangan dari *Structured Query Language*, merupakan bahasa standard yang dipakai pada DBMS berbasis Relational, atau yang biasa disebut RDBMS – Relational Database Management System.
  - SQL merupakan sekumpulan perintah yang digunakan untuk memberikan input, mengakses dan memanipulasi data pada RDBMS.
  - SQL mengikuti standard ANSI (American National Standard Institute) untuk manajemen RDBMS.
- 

# Pengenalan SQL (2)

- Terdapat tiga jenis perintah SQL, yaitu:
  - DDL atau Data Definition Language. DDL merupakan perintah SQL yang berhubungan dengan pendefinisian suatu struktur database, dalam hal ini database dan table. Perintah SQL yang termasuk dalam DDL antara lain : CREATE, ALTER, RENAME, DROP.
  - DML atau Data Manipulation Language. DML merupakan perintah SQL yang berhubungan dengan manipulasi atau pengolahan data atau record dalam table. Perintah SQL yang termasuk dalam DML antara lain : SELECT, INSERT, UPDATE, DELETE.
  - DCL atau Data Control Language. DCL merupakan perintah SQL yang berhubungan dengan pengaturan hak akses user, baik terhadap server, database, table maupun field. Perintah SQL yang termasuk dalam DCL antara lain : GRANT, REVOKE.

# Pengenalan SQL (3)

- Contoh perintah pada SQL

TABLE  
7.1

SQL Data Definition Commands

COMMAND OR OPTION	DESCRIPTION
CREATE SCHEMA AUTHORIZATION	Creates a database schema
CREATE TABLE	Creates a new table in the user's database schema
NOT NULL	Ensures that a column will not have null values
UNIQUE	Ensures that a column will not have duplicate values
PRIMARY KEY	Defines a primary key for a table
FOREIGN KEY	Defines a foreign key for a table
DEFAULT	Defines a default value for a column (when no value is given)
CHECK	Constraint used to validate data in an attribute
CREATE INDEX	Creates an index for a table
CREATE VIEW	Creates a dynamic subset of rows/columns from one or more tables
ALTER TABLE	Modifies a table's definition (adds, modifies, or deletes attributes or constraints)
CREATE TABLE AS	Creates a new table based on a query in the user's database schema
DROP TABLE	Permanently deletes a table (and thus its data)
DROP INDEX	Permanently deletes an index
DROP VIEW	Permanently deletes a view

# Pengenalan SQL (4)

- Contoh perintah pada SQL

TABLE  
7.2

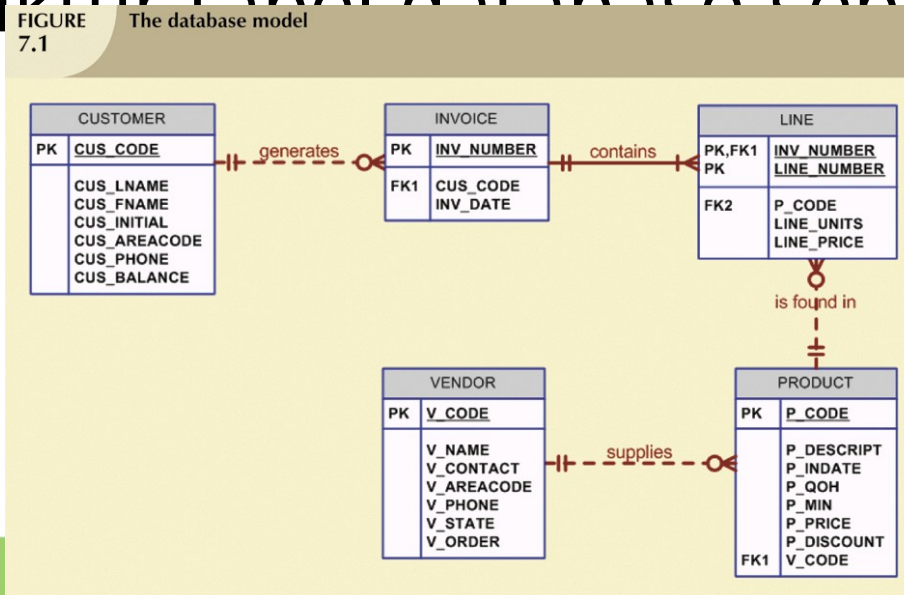
SQL Data Manipulation Commands

COMMAND OR OPTION	DESCRIPTION
INSERT	Inserts row(s) into a table
SELECT	Selects attributes from rows in one or more tables or views
WHERE	Restricts the selection of rows based on a conditional expression
GROUP BY	Groups the selected rows based on one or more attributes
HAVING	Restricts the selection of grouped rows based on a condition
ORDER BY	Orders the selected rows based on one or more attributes
UPDATE	Modifies an attribute's values in one or more table's rows
DELETE	Deletes one or more rows from a table
COMMIT	Permanently saves data changes
ROLLBACK	Restores data to their original values



# Data Definition Language (1)

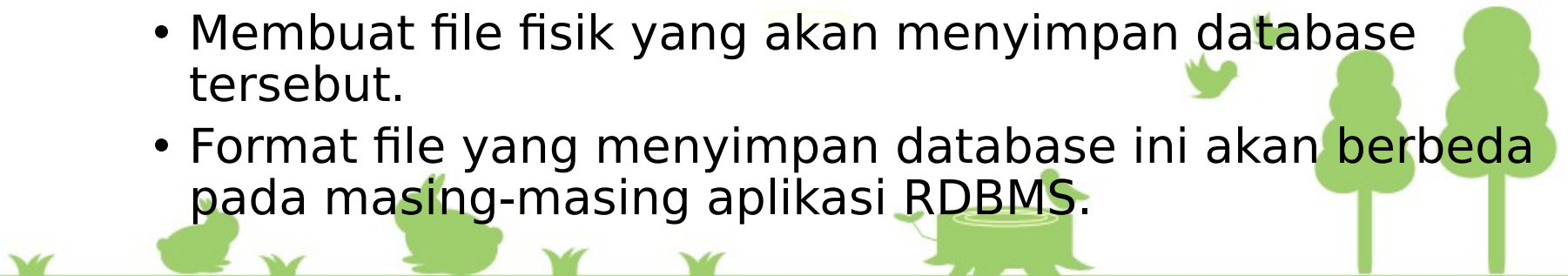
- Perintah SQL yang termasuk pada Data Definition Language (DDL) adalah CREATE, ALTER, RENAME, DROP.
- Penggunaan DDL mengikuti model database dan struktur tabel database seperti pada gambar





# Data Definition Language (2)

- Dari gambar model database di atas, langkah-langkah selanjutnya adalah:
  - Membuat struktur database (perintah: `CREATE DATABASE Nama_Database;`).
  - Membuat struktur tabel yang akan digunakan untuk menyimpan data. Dengan perintah:
    - `CREATE TABLE nama_tabel ( column1 datatype, column2 datatype, column3 datatype, ... );`
- Pada prakteknya, saat DDL dieksekusi, RDBMS akan:
  - Membuat file fisik yang akan menyimpan database tersebut.
  - Format file yang menyimpan database ini akan berbeda pada masing-masing aplikasi RDBMS.





# Data Definition Language (3)

- Tipe data menentukan jenis data yang dapat disimpan dalam satu kolom pada tabel database.
- Pemilihan tipe data tergantung dari tujuan penggunaan kolom tersebut.
- Pemilihan tipe data perlu memperhatikan jenis pengurutan data dan pengambilan data dari database.



# Data Definition Language (4)

- Tipe data yang umum pada database:

TABLE  
7.4

Some Common SQL Data Types

DATA TYPE	FORMAT	COMMENTS
Numeric	NUMBER(L,D)	The declaration NUMBER(7,2) indicates numbers that will be stored with two decimal places and may be up to six digits long, including the sign and the decimal place. Examples: 12.32, -134.99.
	INTEGER	May be abbreviated as INT. Integers are (whole) counting numbers, so they cannot be used if you want to store numbers that require decimal places.
	SMALLINT	Like INTEGER, but limited to integer values up to six digits. If your integer values are relatively small, use SMALLINT instead of INT.
	DECIMAL(L,D)	Like the NUMBER specification, but the storage length is a <i>minimum</i> specification. That is, greater lengths are acceptable, but smaller ones are not. DECIMAL(9,2), DECIMAL(9), and DECIMAL are all acceptable.
Character	CHAR(L)	Fixed-length character data for up to 255 characters. If you store strings that are not as long as the CHAR parameter value, the remaining spaces are left unused. Therefore, if you specify CHAR(25), strings such as "Smith" and "Katzenjammer" are each stored as 25 characters. However, a U.S. area code is always three digits long, so CHAR(3) would be appropriate if you wanted to store such codes.
	VARCHAR(L) or VARCHAR2(L)	Variable-length character data. The designation VARCHAR2(25) will let you store characters up to 25 characters long. However, VARCHAR will not leave unused spaces. Oracle users may use VARCHAR2 as well as VARCHAR.
Date	DATE	Stores dates in the Julian date format.

# Data Manipulation Language (1)

- Perintah pada Data Manipulation Language (DML) meliputi:
  - Penambahan baris pada tabel
  - Menyimpan perubahan tabel
  - Menampilkan isi baris tabel
  - Memperbaharui baris tabel
  - Menghapus baris tabel
  - Menyisipkan baris tabel dengan kondisi tertentu.

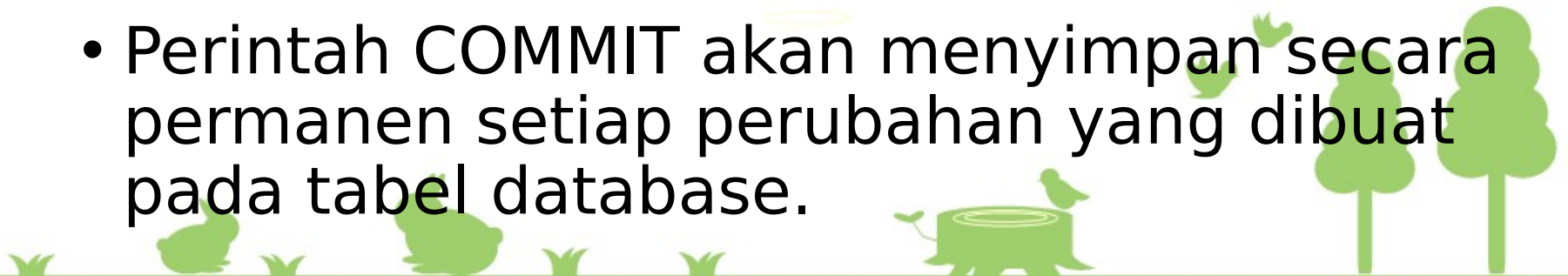


# Data Manipulation Language (2)

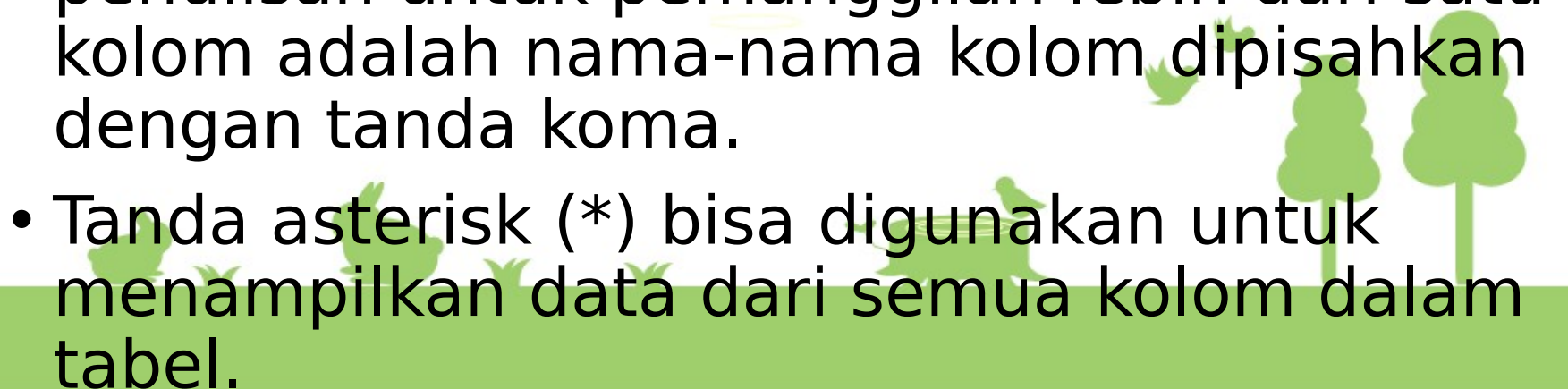
- Perintah untuk menambahkan baris data pada tabel adalah:
  - INSERT. Sintaksnya adalah:
    - INSERT INTO nama\_tabel (kolom1, kolom2, ..., kolomN) VALUES (value1, value2, ..., valueN);
  - Ketika menambahkan data, perhatikan bahwa:
    - Isi baris berada di dalam tanda kurung ( )
    - Karakter dan tanggal dimasukkan menggunakan tanda apostrof ("")
    - Data angka tidak menggunakan tanda apostrof
    - Nama atribut dipisahkan oleh tanda koma ( , )
    - Sebuah data diperlukan untuk setiap kolom
  - Gunakan nilai NULL untuk data yang tidak diketahui.

# Data Manipulation Language (3)

- Perubahan pada isi tabel tidak akan tersimpan secara fisik pada hardisk hingga salah satu kondisi berikut terpenuhi:
  - Database ditutup.
  - Program ditutup.
  - Perintah COMMIT dieksekusi.
- Sintaksnya: COMMIT [WORK];
- Perintah COMMIT akan menyimpan secara permanen setiap perubahan yang dibuat pada tabel database.



# Data Manipulation Language (4)

- Menampilkan isi tabel, menggunakan perintah SELECT.
  - Sintaksnya:
    - `SELECT daftar_nama_kolom FROM nama_tabel;`
  - Daftar\_nama\_kolom mewakili satu atau lebih kolom yang akan ditampilkan. Format penulisan untuk pemanggilan lebih dari satu kolom adalah nama-nama kolom dipisahkan dengan tanda koma.
  - Tanda asterisk (\*) bisa digunakan untuk menampilkan data dari semua kolom dalam tabel.
- 

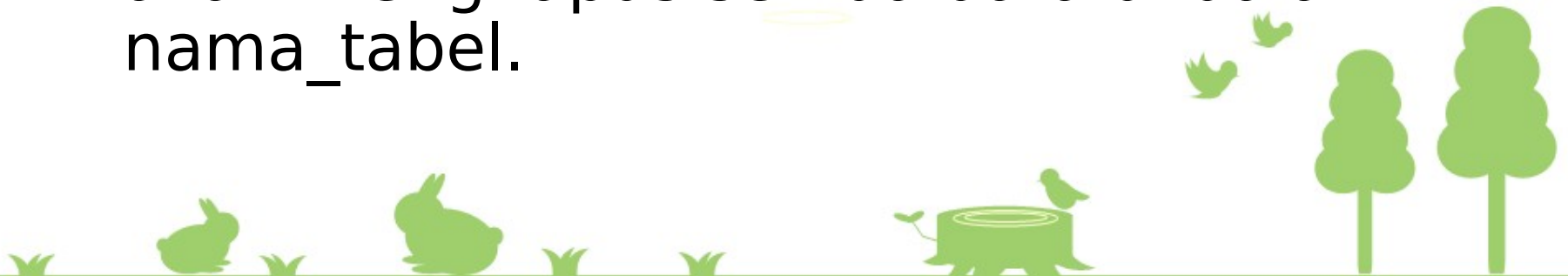
# Data Manipulation Language (5)

- Mengubah dan memperbaharui isi tabel menggunakan perintah UPDATE.
- Sintaks:
  - UPDATE nama\_tabel SET namakolom = data [, namakolom = data] [WHERE daftar\_kondisi];
- Jika lebih dari satu kolom yang diperbaharui dalam satu baris, pemisahannya menggunakan tanda koma.



# Data Manipulation Language (6)

- Menghapus isi tabel menggunakan perintah DELETE
- Sintaksnya:
  - DELETE FROM nama\_tabel [WHERE kondisi];
- [WHERE kondisi] bersifat optional. Tetapi bila opsi ini tidak digunakan, perintah DELETE akan menghapus semua data di dalam nama\_tabel.





# Data Manipulation Language (7)

- Menyisipkan baris tabel juga dapat dilakukan dengan subquery pada perintah SELECT.
- Sintaksnya:
  - INSERT INTO nama\_tabel SELECT daftar\_kolom FROM nama\_tabel;

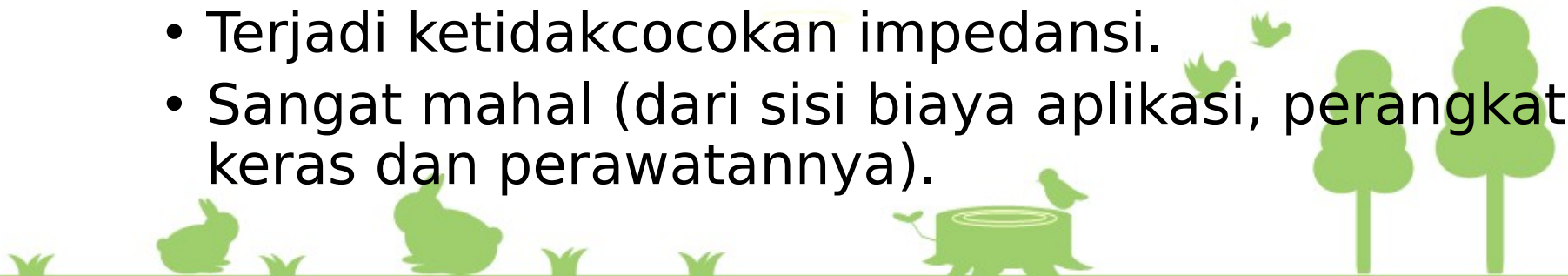


# Contoh Aplikasi RDBMS



# Pengenalan NoSQL (1)

- RDBMS sudah digunakan sangat luas hingga saat ini.
- Namun, RDBMS tidak dikembangkan dalam konsep aplikasi terdistribusi. Hal ini dikarenakan:
  - Perintah join pada RDBMS sangat berat.
  - Susah untuk dikembangkan secara horizontal.
  - Terjadi ketidakcocokan impedansi.
  - Sangat mahal (dari sisi biaya aplikasi, perangkat keras dan perawatannya).



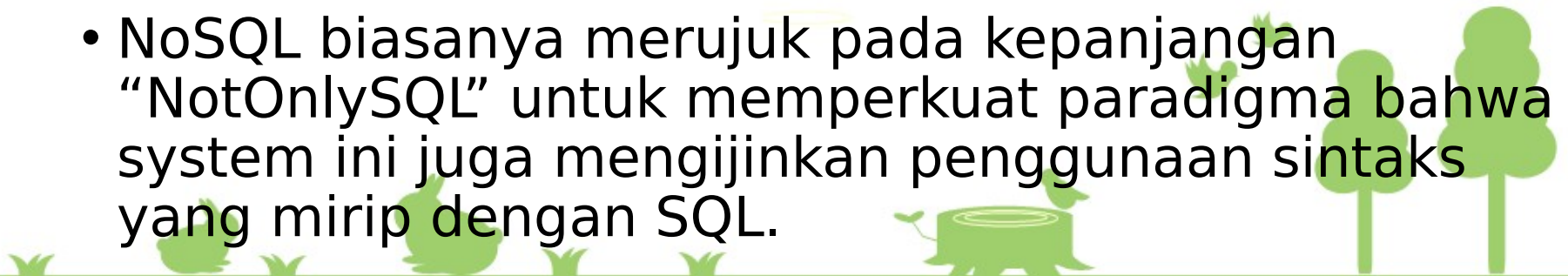
# Pengenalan NoSQL (2)

- Perkembangan teknologi saat ini menimbulkan trend-trend



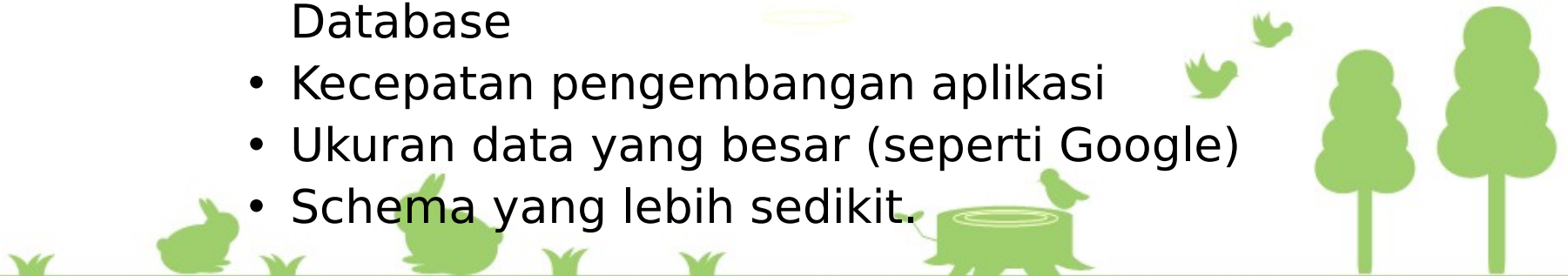
# Pengenalan NoSQL (3)

- Trend-trend tersebut semakin susah dipenuhi dengan menggunakan RDBMS.
- Kemudian diperkenalkan teknologi database yang baru berbasis pada NoSQL.
- NoSQL adalah sebuah DBMS yang menyediakan mekanisme untuk menyimpan dan memanggil data yang memiliki batas konsistensi yang rendah disbanding dengan RDBMS.
- NoSQL biasanya merujuk pada kepanjangan “NotOnlySQL” untuk memperkuat paradigma bahwa system ini juga mengizinkan penggunaan sintaks yang mirip dengan SQL.



# Pengenalan NoSQL (4)

- Karakteristik dari database NoSQL:
  - NoSQL menghindari:
    - Overhead pada transaksi ACID
    - Kompleksitas query SQL
    - Keberadaan DBA
    - Adanya proses transaksi
  - NoSQL menyediakan:
    - Kemudahan dan perubahan yang lebih sering pada Database
    - Kecepatan pengembangan aplikasi
    - Ukuran data yang besar (seperti Google)
    - Schema yang lebih sedikit.



# Pengenalan NoSQL (5)

- Penggunaan NoSQL jika:
  - Model RDBMS terlalu membatasi
  - Dukungan ACID tidak terlalu dibutuhkan
  - Impedansi Object-to-Relational (O/R)
  - Aplikasi bersifat terdistribusi
  - Pencatatan data berasal dari sumber terdistribusi
  - Menyimpan even atau temporal data
  - Data memiliki skema yang fleksibel
- NoSQL tidak dapat digunakan jika:
  - Menggunakan data finansial
  - Membutuhkan kesesuaian dengan ACID
  - Merupakan data bisnis yang kritikal



# Pengenalan NoSQL (6)

- Database NoSQL diklasifikasikan ke dalam 4 data model:
  - Key-value
  - Document
  - Column Family
  - Graph
- Setiap data model memiliki bahasa query masing-masing.

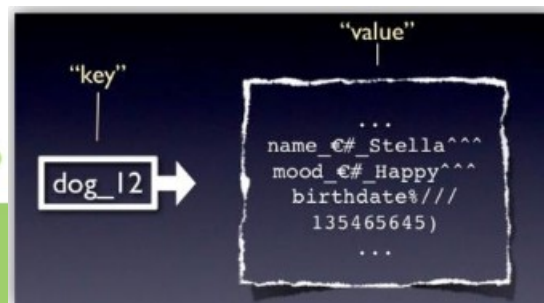




# Pengenalan NoSQL (7)

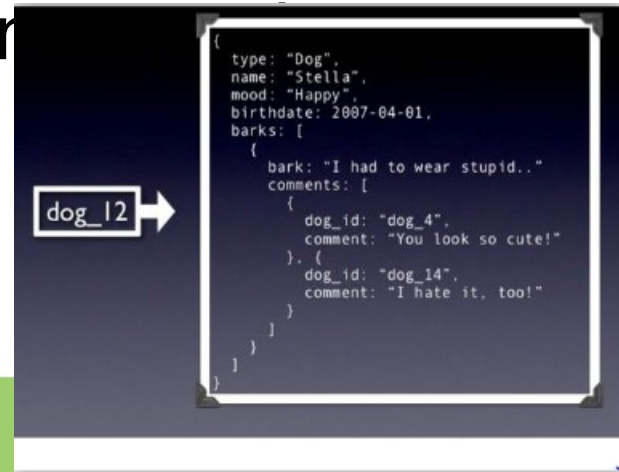
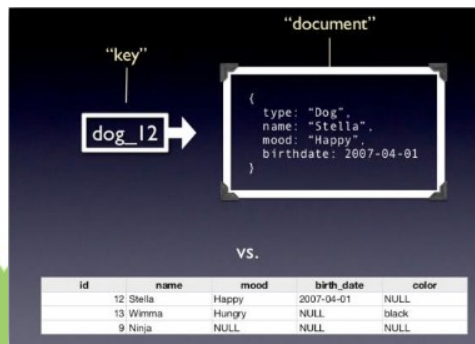
- Database NoSQL dengan model Key-Value merupakan database NoSQL yang paling umum.
- Ide utamanya adalah menggunakan sebuah tabel hash.
- Data akses (value) menggunakan string yang disebut key.
- Data tidak memiliki format khusus.
- Model Data: pasangan (key, value)
- Operasi dasar:
  - Insert(key, value),
  - Fetch(key)
  - Update(key)
  - Delete(key)

Car	
Key	Attributes
1	Make: Nissan Model: Pathfinder Color: Green Year: 2003
2	Make: Nissan Model: Pathfinder Color: Blue Color: Green Year: 2005 Transmission: Auto




# Pengenalan NoSQL (8)

- Database NoSQL berbasis Document memasangkan sebuah key dengan data yang memiliki struktur yang kompleks.
- Data indeks mengikuti konsep B-Trees.
- Document bisa mengandung banyak pasangan key-value atau pasangan key-array atau dokumen.



# Pengenalan NoSQL (9)

- Contoh Database NoSQL berbasis Document



The image shows a screenshot of a Facebook interface with several SQL queries overlaid on different sections, illustrating how a document-based NoSQL database might store and retrieve data. The queries are as follows:

- Top Left (User Profile):**  

```
SELECT name, pic, profile_url  
FROM user  
WHERE uid = me()
```
- Top Right (Stream):**  

```
SELECT message, attachment  
FROM stream  
WHERE source_id = me() AND type = 80
```
- Middle Left (Friendlist):**  

```
SELECT name  
FROM friendlist  
WHERE owner = me()
```
- Middle Right (Activity Feed):**  

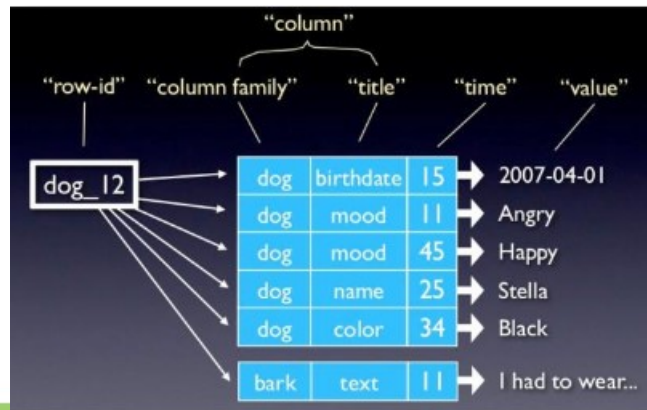
```
SELECT name, pic  
FROM user  
WHERE online_presence = "active"  
AND uid IN ( SELECT uid2  
FROM friend  
WHERE uid1 = me() )
```
- Bottom Left (Groups):**  

```
SELECT name  
FROM group  
WHERE gid IN ( SELECT gid  
FROM group_member  
WHERE uid = me() )
```

At the bottom left, there is a URL: <https://developers.facebook.com/docs/reference/fql/>

# Pengenalan NoSQL (10)

- Database NoSQL berbasis Column memiliki instan data yang paling kecil.
- Database NoSQL berbasis Column mengandung nama, data dan timestamp



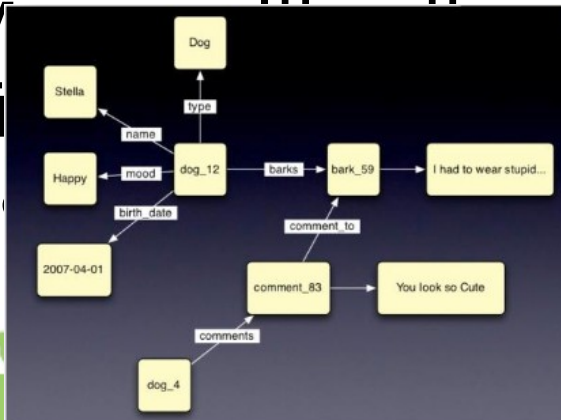
ColumnFamily: Authors		
Key	Value	
"Eric Long"	Columns	
	Name	Value
	"email"	"eric (at) long.com"
	"country"	"United Kingdom"
	"registeredSince"	"01/01/2002"
"John Steward"	Columns	
	Name	Value
	"email"	"john.steward (at) somedomain.com"
	"country"	"Australia"
	"registeredSince"	"01/01/2009"
"Ronald Mathies"	Columns	
	Name	Value
	"email"	"ronald (at) sodeso.nl"
	"country"	"Netherlands, The"
	"registeredSince"	"01/01/2010"

# Pengenalan NoSQL (11)

- Database NoSQL berbasis Graph mengikuti konsep Graph Theory.
- Bisa dikembangkan secara vertical dan tidak memiliki cluster.



- Memenuhi konsep





# Perbandingan SQL dan NoSQL

	SQL Databases	No SQL Database
Example	Oracle , mysql	Mondo DB, CouchDB, Neo4J
Storage Model	Rows and tables	Key-value. Data stored as single document in JSON, XML
Schemas	Static	Dynamic
Scaling	Vertical & Horizontal	Horizontal
Transactions	Yes	Certain levels
Data Manipulation	Select, Insert , Update	Through Object Oriented API's

# Jenis-jenis Aplikasi Database NoSQL (1)

 Apache Jackrabbit	Apache	2010	Document	Apache Open Source	Consistency High Availability Persistence	
 Big Table	Google	2005	Column	Proprietary	Consistency High Availability Partition Tolerance Persistence	designed to scale across hundreds or thousands of machines
 Cassandra	Apache Software Foundation	2008	Column	Apache Open Source	High Availability Partition Tolerance Persistence	write often, read less
Couchbase	Various	2011	Document		Consistency High Availability Persistence	
 CouchDB	Apache Software Foundation	2005	Document	Apache Open Source	High Availability Partition Tolerance Persistence	accumulating, occasionally changing data with pre-defined queries
 HBase	Apache Software Foundation	2010	Column	Apache Open Source	Consistency Partition Tolerance Persistence	random read write to large database
 Hilbari	Gemini Mobile Technologies	2010	Key-Value	Apache Open Source		

# Jenis-jenis Aplikasi Database NoSQL (2)

 <b>Hypertable</b>	Hypertable Inc	2010	Key-Value	GPL Open Source	Consistency Partition Tolerance Persistence	
 <b>MemcacheDB</b>	Steve Chu	2008	Key-Value	BSD Open Source	Consistency Partition Tolerance	small pieces of data with many concurrent connections. Transient data
 <b>MongoDB</b>	10gen	2009	Document	AGPL Open Source	Consistency Partition Tolerance Persistence	dynamic queries, frequently written, rarely read statistical data
 <b>Neo4j</b>	Neo Technology	2007	Graph	AGPL GPL Open Source	High Availability Partition Tolerance Persistence	complex data relationships and queries.
 <b>Redis</b>	Salvatore Sanfilippo	2009	Key-Value	BSD Open Source	Consistency Partition Tolerance Persistence	rapidly changing data, frequently written, rarely read statistical data