**SQL vs No SQL**

Sebelum masuk ke perbedaan dari sql dan no sql, saya akan membahas terlebih dahulu mengenai data. Data adalah sebuah bukti nyata dari sebuah peristiwa atau objek yang sudah dikaji. Secara umum, data adalah sebuah bukti yang biasanya berbentuk tulisan atau bentuk yang lain. Dalam pengembangan aplikasi atau software, data sangat dibutuhkan untuk proses pengembangan yang memungkinkan untuk menampilkan data ke user atau meminta data dari user. Data dalam pengembangan software menjadi informasi antar software dengan user. Data biasanya disimpan ke dalam database. Database adalah sebuah tempat penyimpanan data yang digunakan untuk pengembangan software, di dalam database terdapat beberapa dataset yang dibuat untuk mengelompokkan data berdasarkan ketentuan yang sudah dibuat. Dataset yang ada di dalam database biasanya berbentuk tabel, satu tabel ada baris dan kolom dan satu kolom biasanya berisi data tunggal dan satu baris bisanya berisi data tuple. Data tunggal yaitu data yang bersifat satu maksudnya data tersebut hanya berisi satu informasi, misal satu kolom data tunggal adalah kolom nama, maka data tunggal dari kolom tersebut adalah masing – masing nama yang terdapat pada kolom tersebut. Sedangkan data tuple adalah data yang berisi satu set data, misal dalam satu tabel terdiri dari kolom nama, Alamat, dan nomor hp, maka data tuple dari satu set data adalah satu set informasi dari satu baris, contoh nama = aan, Alamat = undaan, nomor hp = 329.

Pada pengembangan software, terdapat 2 istilah yang sering digunakan untuk menyimpan data, yaitu SQL dan No SQL. Keduanya digunakan dengan studi kasus yang berbeda. No SQL digunakan jika:

1. Basis data relasional tidak restriktif.

Ketika kita ingin mengembangkan software yang mana awalnya memang software tersebut tidak menyimpan data yang terdistribusi atau data yang bervariasi, Ketika software yang kita kembangkan sudah menjadi software yang besar, maka akan banyak variasi data yang harus kita simpan ke dalam system kita, Ketika hal tersebut terjadi maka saatnya kita mengubah system database kita dari SQL menjadi NoSQL. Jadi, **NoSQL digunakan untuk menyimpan data terdistribusi atau data yang mempunyai variasi yang banyak.**

1. Ketika ACID *(Atomicity Consistency Isolation Durability)* sudah tidak dibutuhkan.

Dalam konteks pemrosesan transaksi, terdapat ACID yang mengacu pada empat sifat utama transaksi.

1. **Atomicity**

Semua perubahan data dilakukan seolah-olah itu adalah satu operasi. Misal semua transaksi terjadi atau tidak sama sekali. Jadi istilahnya kalo dijalnkan satu ya jalan semua kalo tidak dijalankan ya tidak dijalankan semua. Jika kita ingin memindahkan uang dari Bank A ke Bank B, lalu terjadi error pada server Bank B padahal kita sudah memencet tombol kirim, apa yang terjadi? Jika system bank A dan Bank B tidak menggunakan Atomicity, maka uang tersebut akan hilang, Atomicity mencegah terjadinya hal tersebut. Maka dari itu istilahnya jika dijalankan ya jalan semua, jika dijalankan ke server Bank B ya dijalankan keduanya Bank A dan Bank B, jika tidak dijalankan ya tidak jalan semua, jika dijalankan ke Bank B ternyata server Bank B mengalami error, maka transaksi ke Bank B tidak dapat dijalankan dan perpindahan uang dari Bank A ke Bank B pun dibatalkan. Atomicity juga bersifat keterkaitan.

1. **Consistency**

Consistency digunakan untuk memeriksa apakah setiap operasi transaksi mematuhi Batasan integritas data dan aturan bisnis yang telah ditetapkan dalam transaksi basis data. Misal suatu website menjual barang dengan stok yang terbatas yaitu 100 stok, jika ada pembeli 1 yang ingin membeli 100 stok dan ada pembeli 2 yang membeli 20 stok dalam waktu yang sama. Jika website tersebut tidak menggunakan prinsip consistency maka kedua transaksi tersebut akan dieksekusi dan membuat nilai stok yang ada di database menjadi minus menjadi (-20). Jika menggunakan prinsip consistency biasanya hanya satu transaksi yang akan dijalankan jika ada transaksi dalam waktu yang sama. Istilah ini dikenal sebagai *“Serial Execution”*. Misal transaksi yang dijalankan dulu adalah transaksi pembeli 1 yang membeli 100 stok, prinsip consistency juga dapat mengecek ketersediaan stok yang ada di dalam database, maka transaksi dari pembeli 1 dapat berjalan dengan lancar, lalu transaksi pembeli 2 yang membeli 20 stok akan dieksekusi, sebelum dieksekusi akan dicek terleih dahulu ketersediaan stok yang ada di dalam database, jika stoknya 0, maka transaksi akan dibatalkan. Prinsip consistency bersifat relative yang menyesuaikan aturan dari bisnis yang telah ditetapkan.

1. **Isolation**
2. **Durability**