BASIC REDIS

# Apa Itu Redis?

**Redis** adalah singkatan dari Remote Dictionary Server. Merupakan sistem penyimpanan data open-source yang berfokus pada kinerja tinggi dan kesederhanaan penggunaan. Redis menyimpan data dalam memori, sehingga menawarkan akses data yang sangat cepat.

Redis dapat digunakan sebagai:

* **Database**: Menyimpan data terstruktur dalam format key-value, seperti nama pengguna dan password, atau data produk dan harga.
* **Cache**: Menyimpan data yang sering diakses dari database utama untuk meningkatkan performa aplikasi.
* **Message broker**: Mengirim dan menerima pesan antara aplikasi yang berbeda.
* **Pub/sub**: Menerbitkan pesan ke beberapa subscriber yang berlangganan.
* **Session store**: Menyimpan data sesi pengguna untuk aplikasi web.

Mengapa Harus Menggunakan Redis? Berikut beberapa alasan mengapa Anda harus menggunakan Redis:

* **Performa tinggi**: Redis menawarkan akses data yang sangat cepat karena data disimpan dalam memori.
* **Kesederhanaan**: Redis mudah digunakan dan dipelajari.
* **Serbaguna**: Redis dapat digunakan untuk berbagai macam aplikasi, seperti database, cache, message broker, pub/sub, dan session store.
* **Skalabilitas**: Redis dapat di-scale secara horizontal untuk menangani beban kerja yang besar.
* **Open-source**: Redis adalah open-source dan gratis untuk digunakan.

Keunggulan Menggunakan Redis:

* Performa tinggi
* Kesederhanaan
* Serbaguna
* Skalabilitas
* Open-source
* Dukungan komunitas yang besar

Kekurangan Menggunakan Redis:

* **Tidak tahan lama**: Data yang disimpan dalam memori akan hilang jika server restart.
* **Tidak cocok untuk data yang besar**: Redis tidak ideal untuk menyimpan data yang sangat besar.
* **Keamanan**: Redis perlu dikonfigurasi dengan benar untuk memastikan keamanan data.

Hal yang Harus Diperhatikan Ketika Menggunakan Redis:

* **Penting untuk memilih struktur data yang tepat**: Redis menawarkan berbagai macam struktur data, seperti string, hash, list, set, dan sorted set. Pilih struktur data yang tepat untuk aplikasi Anda untuk memaksimalkan performa.
* **Pertimbangkan ketahanan data**: Jika data Anda harus tahan lama, Anda perlu menggunakan mekanisme persistensi Redis, seperti RDB atau AOF.
* **Awasi penggunaan memori**: Redis menyimpan data dalam memori, jadi penting untuk memantau penggunaan memori dan memastikan server Anda memiliki cukup memori untuk menangani beban kerja.
* **Keamanan**: Konfigurasikan Redis dengan benar untuk memastikan keamanan data. Gunakan autentikasi dan otorisasi untuk mengontrol akses ke server Redis Anda.
* **Gunakan Redis dengan tepat**: Redis adalah alat yang ampuh, tetapi tidak cocok untuk semua aplikasi. Pastikan Anda memahami keterbatasan Redis sebelum menggunakannya dalam aplikasi Anda.

Jangan terburu-buru menggunakan Redis, ada beberapa pertimbangkan yang perlu diperhatikan sebelum menggunakan Redis:

* Gunakan Redis sebagai alternatif terakhir setelah anda melakukan optimalisasi koding dan query pada aplikasi anda.
* Gunakan Redis untuk data yang sering digunakan/diambil tapi jarang diupdate.
* Jangan gunakan Redis untuk data yang terlalu besar karena itu memberikan beban yang berat pada memory (RAM).

# Install Redis

Redis mendukung OS berikut: Linux, macOS, dan Windows. Di sini saya akan menunjukkan cara install pada Windows, silahkan kunjungi link berikut ini untuk info instalasi pada OS lainnya.

<https://redis.io/docs/latest/operate/oss_and_stack/install/install-redis/>

Install Redis pada Windows:

1. Install atau enable WSL2.

Microsoft menyediakan detail intruksi untuk Install WSL pada link berikut: <https://learn.microsoft.com/en-us/windows/wsl/install>.

1. Install Redis.

Setelah install atau mengaktifkan WSL ketikan perintah berikut ini pada WSL anda:

curl -fsSL https://packages.redis.io/gpg | sudo gpg --dearmor -o /usr/share/keyrings/redis-archive-keyring.gpg

echo "deb [signed-by=/usr/share/keyrings/redis-archive-keyring.gpg] https://packages.redis.io/deb $(lsb\_release -cs) main" | sudo tee /etc/apt/sources.list.d/redis.list

sudo apt-get update

sudo apt-get install redis

Ada dua aplikasi yang akan terinstall yaitu **redis-server** dan **redis-cli**.

1. Jalankan Redis Server

sudo service redis-server start

1. Hubungkan ke Redis dengan redis-cli dan lakukan testing dengan mengetik ping.

redis-cli

127.0.0.1:6379> ping

PONG

# Redis Config

Ketika menjalankan Redis menggunakan command **redis-cli** maka **redis-server** akan menggunakan default configurasi redis. Pada configurasi redis kita dapat mengatur berbagai hal mulai dari memory, memory-policy, authenticate, authorize, port, persistence, dan configurasi lainnya.

Bagaimana mengubah config Redis? Kita dapat mengubah config dengan membuat sebuah file config baru **redis.conf** lalu sesuaikan configurasi kalian dan jalankan perintah berikut:

redis-server <your-path-redis-config>

Jika kalian bingung dengan config apa saja yang harus dimasukkan kalian bisa menyalin default config pada link berikut ini:

<https://redis.io/docs/latest/operate/oss_and_stack/management/config/>

Pastikan config nya di sesuaikan dengan versi Redis yang kalian pakai.

# Struktur dan Tipe Data Redis

Redis menyimpan data dalam struktur data **key-value**, di mana setiap kunci unik dipetakan ke nilai tertentu. Redis mendukung berbagai tipe data yang dapat digunakan untuk menyimpan berbagai jenis informasi.

Tipe Data Redis:

* **String**: Tipe data paling sederhana di Redis, digunakan untuk menyimpan teks biasa. Contoh penggunaan: menyimpan nama pengguna, pesan, URL, JSON, dll.
* **Hash**: Tipe data yang mirip dengan dictionary di Python, digunakan untuk menyimpan kumpulan key-value yang terkait. Contoh penggunaan: menyimpan data pengguna, seperti alamat email, nomor telepon, dan alamat.
* **List**: Tipe data yang mirip dengan array di Python, digunakan untuk menyimpan daftar nilai yang berurutan. Contoh penggunaan: menyimpan daftar item dalam keranjang belanja, daftar tweet, atau daftar riwayat penelusuran.
* **Set**: Tipe data yang mirip dengan set di Python, digunakan untuk menyimpan kumpulan nilai yang unik dan tidak berurutan. Contoh penggunaan: menyimpan daftar tag untuk sebuah artikel, daftar pengikut pengguna, atau daftar ID sesi.
* **Sorted Set**: Tipe data yang mirip dengan set, tetapi dengan nilai yang diurutkan berdasarkan skor yang terkait. Contoh penggunaan: menyimpan daftar skor pemain game, daftar tugas dengan prioritas, atau daftar tweet berdasarkan popularitas.
* **HyperLogLog**: Tipe data yang digunakan untuk memperkirakan jumlah elemen unik dalam kumpulan besar data dengan efisiensi memori yang tinggi. Contoh penggunaan: menghitung jumlah pengunjung unik situs web, menghitung jumlah produk yang dilihat pengguna, atau menghitung jumlah tweet yang dibagikan.
* **Geospatial**: Tipe data yang digunakan untuk menyimpan data geospasial, seperti koordinat GPS dan bentuk geometris. Contoh penggunaan: menyimpan lokasi toko, melacak pergerakan pengguna, atau menghitung jarak antara dua titik.

Pada pembahasan kali ini kita hanya akan mempelajari Tipe Data String, untuk penjelasan tipe data lainnya dapat kalian lihat di link dokumentasi redis berikut: <https://redis.io/docs/latest/develop/data-types/>.

# Tipe Data String

Kita menggunakan **redis-cli** untuk memanipulasi data pada Redis. Redis CLI adalah aplikasi command line interface default yang ada Ketika kita menginstall redis. Untuk memanipulasi data redis, jalankan perintah **redis-cli** atau **redis-cli -h localhost -p 6379** untuk menyesuaikan host dan port pada config kalian.

Pada Redis Ketika ingin menyimpan data kita perlu menentukan pada database apa data ingin di simpan. Berbeda dengan database lainnya yang nama database nya bisa kita sesuaikan, pada Redis database-nya berupa index yang jumlah nya dapat kita atur pada config redis dan secara default jumlahnya ada 16 (index 0 - 15). Untuk select database ketikan command **select <index-database>**.

## Common Operation

1. Mengubah string value dari key.

**set key value**

127.0.0.1:6379> set yusril "M Yusril Aprial"

OK

1. Mendapakan value menggunakan key.

**get key**

127.0.0.1:6379> get yusril

"M Yusril Aprial"

1. Mengecek apakah key memiliki value.

**exists key**

127.0.0.1:6379> exists yusril

(integer) 1

1. Menambahkan data value ke key.

**append key value**

127.0.0.1:6379> append yusril " Aprial"

(integer) 22

127.0.0.1:6379> get yusril

"M Yusril Aprial Aprial"

1. Mencari key menggunakan pattern (Regex), cara ini kurang direkomendasikan karena akan mengambil semua key yang ada lalu mencari dengan pattern yang diberikan.

**keys pattern**

127.0.0.1:6379> keys \*

1) "ayu"

2) "naruto"

3) "zidan"

4) "yusril"

5) "anto"

6) "ado"

1. Menghapus Menggunakan key.

**del key [key ...]**

127.0.0.1:6379> del naruto anto ado

(integer) 3

127.0.0.1:6379> keys \*

1) "ayu"

2) "zidan"

3) "yusril"

## Range Operation

1. Mengubah value dari offset yang ditentukan.

**setrange key offset value**

127.0.0.1:6379> setrange yusril 9 "ini data yang diganti"

(integer) 30

127.0.0.1:6379> get yusril

"M Yusril ini data yang diganti"

1. Mengambil value dari range yang ditentukan.

**getrange key start end**

127.0.0.1:6379> getrange yusril 0 7

"M Yusril"

## Multiple Operation

1. Mengambil beberapa value berdasarkan key yang diberikan.

**mget key [key ...]**

mget yusril ayu zidan

1) "M Yusril ini data yang diganti"

2) "Ayu Zahra"

3) "Zidan"

1. Mengubah beberapa key sekaligus.

**mset key value [key value ...]**

127.0.0.1:6379> mset yusril "Yusril" ayu "Ayu" zidan "Zidan"

OK

127.0.0.1:6379> mget yusril ayu zidan

1) "Yusril"

2) "Ayu"

3) "Zidan"

## Expiration Operation

Ketika menggunakan Redis sebaiknya selalu menetukan kapan waktu expired nya, untuk mengurangi beban memory bagi data yang jarang di ambil.

1. Mengatur waktu pada key yang sudah ada.

**expire key seconds**

127.0.0.1:6379> expire yusril 60

(integer) 1

1. Megatur waktu sekaligus value dari key diberikan.

**setex key seconds value**

127.0.0.1:6379> setex dian 60 "Dian"

OK

1. Melihat sisa waktu dari key.

**ttl key**

127.0.0.1:6379> ttl ronaldo

(integer) 334

## Increment & Decrement Operation

Ketika client melakukan set pada waktu bersamaan, memungkinkan untuk terjadinya Race Condition. Untuk mengatasi hal tersebut redis menyediakan Increment & Decrement operation yang akan berjalan secara atomik.

1. Menambah satu angka berdasarkan key.

**Incr key**

127.0.0.1:6379> incr count

(integer) 1

127.0.0.1:6379> get count

"1"

1. Mengurangi satu angka berdasarkan key.

**decr key**

127.0.0.1:6379> decr count

(integer) 0

127.0.0.1:6379> get count

"0"

1. Menambahkan berdasarkan jumlah yang diberikan.

**incrby key increment**

127.0.0.1:6379> incrby count 10

(integer) 10

127.0.0.1:6379> get count

"10"

1. Mengurangi berdasarkan jumlah yang diberikan.

**decrby key decrement**

127.0.0.1:6379> decrby count 5

(integer) 0

127.0.0.1:6379> get count

"0"

## Flush Operation

1. Menghapus semua key pada database yang sedang diakses.

**flushdb**

127.0.0.1:6379> flushdb

OK

1. Menghapus semua key pada seluruh database.

**flushall**

127.0.0.1:6379> flushall

OK

## Pipeline Operation

Operasi ini memungkinkan untuk mengeksekusi beberapa operasi sekaligus dengan membuat file txt dan menuliskan operasi-operasi yang ingin dieksekusi.

**redis-cli -h <host> -p <port> -n <index-database> --pipe < <path-file>**

Contoh isi file:

set yusril "M Yusril Aprial"

set anto "Anto Putradewa"

set zidan "Zidan"

set ayu "Ayu Zahra"

Contoh perintah:

redis-cli -h localhost -p 6379 -n 0 --pipe < input-file.txt

## Transaction Operation

Redis transactions memungkinkan Anda mengeksekusi serangkaian perintah sebagai satu unit kerja, memastikan integritas data. Ini berguna saat Anda perlu melakukan beberapa perubahan pada data secara atomis, artinya semuanya berhasil atau semuanya gagal.

1. **MULTI**: Menandai dimulainya transaksi. Perintah ini memberi tahu Redis untuk mulai merekam perintah yang akan menjadi bagian dari transaksi.
2. **EXEC**: Menjalankan semua perintah yang direkam sejak MULTI dan menerapkan perubahan ke database. Jika ada kesalahan pada salah satu perintah, seluruh transaksi dibatalkan.
3. **DISCARD**: Membuang semua perintah yang direkam sejak MULTI. Transaksi dibatalkan dan tidak ada perubahan yang diterapkan ke database.

Contoh yang disetujui (EXEC):

127.0.0.1:6379> MULTI

OK

127.0.0.1:6379> set joko "Joko"

QUEUED

127.0.0.1:6379> set dini "Dini"

QUEUED

127.0.0.1:6379> EXEC

1) OK

2) OK

Contoh yang ditolak (DISCARD):

127.0.0.1:6379> MULTI

OK

127.0.0.1:6379> set intan "Intan"

QUEUED

127.0.0.1:6379> set gaia "Gaia"

QUEUED

127.0.0.1:6379> DISCARD

OK

## Server Operation

1. Memonitor Redis Server.

**monitor**

1. Menampilkan semua config.

**info**

1. Menampilkan config berdasarkan key-config.

**config get <key-config>**

1. Menampilkan list client yang mengakses server.

**client list**

1. Menampilkan id client saat ini.

**client id**

1. Mematikan client yang sedang aktif berdasarkan address.

**client kill <ip:port>**

# Referensi

Programmer Zaman Now: <https://www.youtube.com/watch?v=5N0fD0GMm2g&list=PL-CtdCApEFH8dCbx_jnFR2SKqUWAw-HQX&index=1>

Redis Docs: <https://redis.io/docs/latest/>