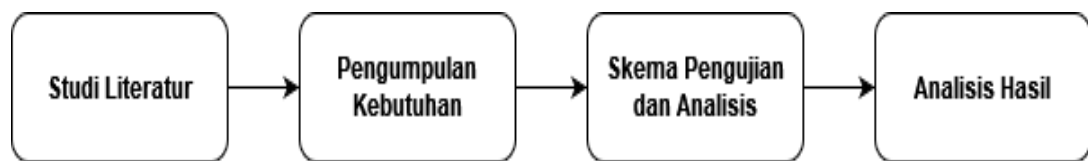


BAB III

METODOLOGI

Alur metode penelitian yang digunakan, dimulai dari hasil studi literatur, pengumpulan kebutuhan, perancangan program, implementasi program, dan kesimpulan hasil pengujian. Diagram alur penelitian dapat dilihat pada gambar 3.1



Gambar 3. 1 Metodologi Penelitian

3.1 Studi Literatur

Tahap ini dilakukan Mengumpulkan dan mempelajari penelitian-penelitian terdahulu yang relevan dengan topik ini untuk memahami dasar teoretis dan temuan sebelumnya.

3.2 Pengumpulan Kebutuhan

Alat-alat yang digunakan dalam penelitian ini terdiri dari perangkat keras dan perangkat lunak.

3.2.1 Perangkat Keras

Spesifikasi perangkat keras (*hardware*) yang digunakan sebagai *web server* pada penelitian ini dapat dilihat pada tabel 3.1.

Tabel 3. 1 Alat dan Perangkat Pengujian

No	Perangkat Keras	Spesifikasi
1	Prosesor	i3 5005u
2	Grafis	nvidia geforce 930m
3	RAM	6 GB

3.2.2 Perangkat Lunak

Perangkat lunak yang digunakan pada proses analisis perbandingan kinerja *Redis* dan *Memcached* dijabarkan pada tabel 3.2.

Tabel 3. 2 Spesifikasi *Software*

No	Nama <i>Software</i>	Keterangan
1	<i>Memtear Benchmark</i>	<i>Software</i> yang digunakan untuk menguji kinerja <i>Redis</i> dan <i>Memcached</i> .
2	Ubuntu	<i>Operating System</i> yang digunakan pada server pengujian kinerja <i>Redis</i> dan <i>Memcached</i> .
3	<i>Redis</i>	Sistem Penyimpanan struktur data dalam <i>memory</i> yang digunakan sebagai <i>cache</i> yang selanjutnya akan diuji menggunakan Memtier dan nantinya menjadi bahan perbandingan dengan <i>Memcached</i> .
4	<i>Memcached</i>	Sistem Penyimpanan struktur data dalam <i>memory</i> yang digunakan sebagai <i>cache</i> yang selanjutnya akan diuji menggunakan Memtier dan nantinya menjadi bahan perbandingan dengan <i>Redis</i> .

3.3 Persiapan Lingkungan Pengujian

Tahap ini dilakukan untuk menyiapkan lingkungan laboratorium yang diperlukan untuk menjalankan pengujian kinerja *Redis* dan *Memcached*. Tahapan persiapan lingkungan pengujian terdiri dari beberapa tahapan sebagai berikut :

1. Menyediakan perangkat yang digunakan sebagai server dengan spesifikasi yang cukup untuk menjalankan *Redis* dan *Memcached*.
2. Menginstal dan mengonfigurasi *Redis* dan *Memcached* sesuai dengan spesifikasi yang telah ditentukan.
3. Menyiapkan alat atau *software* atau *tool* yang akan digunakan untuk pengukuran kinerja yaitu Memtier.

3.4 Skenario Pengujian

3.4.1 Throughput

Tujuan: Mengukur Jumlah *request* yang dapat ditangani per detik.

Langkah-langkah:

Tahap lingkungan pengujian disiapkan dengan instalasi *Redis* dan *Memcached* pada masing masing server. Operasi dasar (*GET*, *SET*) dibuat menggunakan skrip dan ditujukan pada kedua server. Skrip dengan kombinasi beban (25 *client* dengan 5000 *request*, 25 *client* dengan 10000 *request*, 25 *client* dengan 20000 *request*, 50 *client* dengan 5000 *request*, 50 *client* dengan 10000 *request*, 50 *client* dengan 20000 *request*, 100 *client* dengan 5000 *request*, 100 *client* dengan 10000 *request*, dan 100 *client* dengan 20000 *request*) dan catat berapa *request* yang dapat ditangani.

Bandingkan *throughput* (jumlah permintaan per detik) untuk kedua sistem.

Tujuan: Mengukur kemampuan sistem untuk menangani peningkatan beban.

Langkah-langkah:

Tingkatkan jumlah permintaan secara bertahap sesuai dengan kombinasi beban yang akan diujikan. Catat kinerja kedua sistem saat beban ditingkatkan.

3.4.2 *Response time*

Tujuan: Mengukur waktu yang dibutuhkan untuk merespon *request* dari *client*..

Langkah-langkah:

Tahap lingkungan pengujian disiapkan dengan instalasi *Redis* dan *Memcached* pada masing masing server. Operasi dasar (*GET*, *SET*) dibuat menggunakan skrip dan ditujukan pada kedua server. Skrip dengan kombinasi beban (25 *client* dengan 5000 *request*, 25 *client* dengan 10000 *request*, 25 *client* dengan 20000 *request*, 50 *client* dengan 5000 *request*, 50 *client* dengan 10000 *request*, 50 *client* dengan 20000 *request*, 100 *client* dengan 5000 *request*, 100 *client* dengan 10000 *request*, dan 100 *client* dengan 20000 *request*) dan catat berapa Waktu yang dibutuhkan untuk merespon *request* dari *client*. Bandingkan *response time* untuk kedua sistem.

Tujuan: Mengukur kemampuan sistem untuk menangani peningkatan beban.

Langkah-langkah:

Tingkatkan jumlah permintaan secara bertahap sesuai dengan kombinasi beban yang akan diujikan. Catat kinerja kedua sistem saat beban ditingkatkan.

3.4.3 *Memory Usage*

Tujuan: Mengukur jumlah *memory* yang digunakan oleh *cache*.

Langkah-langkah:

Tahap lingkungan pengujian disiapkan dengan instalasi *Redis* dan *Memcached* pada masing masing server. Operasi dasar (*GET*, *SET*) dibuat menggunakan skrip dan ditujukan pada kedua server. Skrip dengan kombinasi beban (25 *client* dengan 5000 *request*, 25 *client* dengan 10000 *request*, 25 *client* dengan 20000 *request*, 50 *client* dengan 5000 *request*, 50 *client* dengan 10000 *request*, 50 *client* dengan 20000 *request*, 100 *client* dengan 5000 *request*, 100 *client* dengan 10000 *request*, dan 100 *client* dengan 20000 *request*) dan catat berapa jumlah *memory* yang digunakan oleh *cache*. Bandingkan *memory usage* untuk kedua sistem.

Tujuan: Mengukur kemampuan sistem untuk menangani peningkatan beban.

Langkah-langkah:

Tingkatkan jumlah permintaan secara bertahap sesuai dengan kombinasi beban yang akan diujikan. Catat kinerja kedua sistem saat beban ditingkatkan.

3.4.4 CPU Usage

Tujuan: Mengukur peningkatan *CPU Usage* saat *load test* dijalankan..

Langkah-langkah:

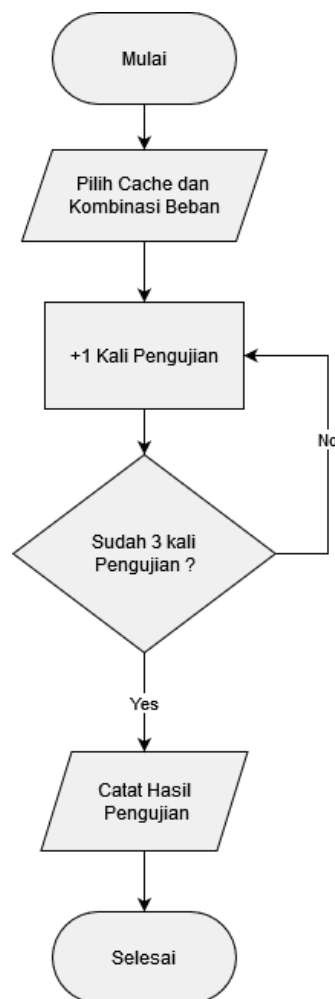
Lingkungan pengujian disiapkan dengan instalasi *Redis* dan *Memcached* pada masing masing server. Operasi dasar (*GET*, *SET*) dibuat menggunakan skrip dan ditujukan pada kedua server. Skrip dengan kombinasi beban (25 *client* dengan 5000 *request*, 25 *client* dengan 10000 *request*, 25 *client* dengan 20000 *request*, 50 *client* dengan 5000 *request*, 50 *client* dengan 10000 *request*, 50 *client* dengan 20000 *request*, 100 *client* dengan 5000 *request*, 100 *client* dengan 10000 *request*, dan 100 *client* dengan 20000 *request*) dan catat berapa peningkatan *CPU Usage* saat *load test* dijalankan.. Bandingkan *CPU usage* untuk kedua sistem.

Tujuan: Mengukur kemampuan sistem untuk menangani peningkatan beban.

Langkah-langkah:

Tingkatkan jumlah permintaan secara bertahap sesuai dengan kombinasi beban yang akan diujikan. Catat kinerja kedua sistem saat beban ditingkatkan.

3.5 Pengujian



Gambar 3. 2 Skenario Pengujian

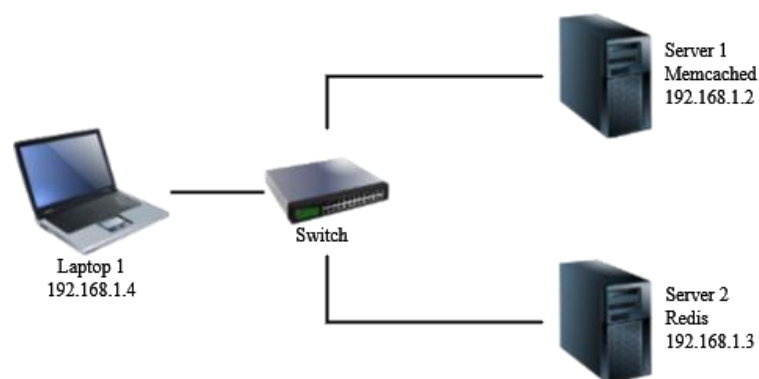
Gambar 3.2 merupakan proses pengujian yang dilakukan pada penelitian ini. Proses pengujian pada penelitian ini dilakukan dengan beberapa tahapan yang di jabarkan sebagai berikut :

1. Melakukan pengujian awal untuk memastikan semua sistem berfungsi dengan baik dan melakukan kalibrasi jika diperlukan.
2. Menjalankan skenario pengujian pada *Redis* dan *Memcached* secara bergantian untuk setiap parameter kinerja.
3. Melakukan pengulangan pengujian beberapa kali untuk mendapatkan data yang konsisten dan valid.
4. Mencatat hasil pengujian secara detail, termasuk waktu respon, *throughput*, *memory usage*, dan CPU

Parameter yang di uji pada penelitian ini terdiri dari 4 parameter. Berikut penjelasan dari parameter yang sudah di jelaskan pada tabel 3.3.

Tabel 3. 3 Parameter

No.	Parameter	Penjelasan	Satuan
1.	<i>Response time</i>	Waktu yang dibutuhkan untuk merespon <i>request</i> dari <i>client</i> .	<i>Ms</i>
2.	<i>Throughput</i>	Jumlah <i>request</i> yang dapat ditangani per detik.	<i>Ops/sec</i>
3.	Memory Usage	Jumlah <i>memory</i> yang digunakan oleh <i>cache</i> .	<i>Mb</i>
4.	<i>CPU Usage</i>	Peningkatan <i>CPU Usage</i> saat <i>load test</i> dijalankan.	<i>Percent</i>



Gambar 3. 3 Skema Pengujian

Gambar 3.3 menunjukkan skema pengujian yang terdiri dari desktop *client*, *switch*, dan dua *server caching*, yaitu *Redis* dan *Memcached*. Laptop 1 terhubung ke *switch*, *switch* berfungsi untuk menghubungkan Laptop 1 dengan server. Server *Memcached* menggunakan alamat IP 192.168.1.2 dengan port 11211, sedangkan server *Redis* menggunakan alamat IP 192.168.1.3 dengan port 6379.

Laptop 1 berfungsi sebagai pengirim *request* yang dihasilkan oleh *Mementier*, yang mensimulasikan berbagai kombinasi beban (*client* dan *request*) ke server *Redis* dan *Memcached*. *Mementier* bekerja dengan mengirimkan *request* ke kedua server, yang kemudian masing-masing server akan merespon *request* tersebut. Output dari *Mementier* berupa file *Excel* yang berisi data performa dari masing-masing server yang terdiri dari *response time*, *throughput*, *memory usage*, dan *CPU usage* untuk setiap request yang diproses.

3.6 Analisis Hasil

Tahapan ini mencakup analisis hasil pengujian *Redis* dan *Memcached* menggunakan *Mementier*. Analisis perbandingan kinerja dilakukan berdasarkan skor parameter seperti *response time*, *throughput*, *memory usage* dan *CPU Usage*. Skor parameter kinerja tersebut dibandingkan untuk menentukan sistem *cache* mana yang lebih baik antara *Redis* dan *Memcached*, berdasarkan variasi beban yang diberikan oleh *Mementier*.