

BAB II

LANDASAN TEORI

2.1 Jaringan Komputer

Jaringan komputer merupakan kumpulan dari beberapa komputer yang berjumlah banyak yang terpisah-pisah namun dapat saling terkoneksi atau saling terhubung dalam melaksanakan tugasnya. Jaringan komputer seperti dua buah komputer dapat dikatakan saling terhubung apabila keduanya dapat saling berbagi data, bertukar informasi, program-program, dan sebagainya. LAN (*Local Area Network*), MAN (*Metropolitan Area Network*), WAN (*Wide Area Network*) adalah contoh dari jaringan komputer. Jaringan komputer dapat dihubungkan melalui berbagai kabel, seperti kabel tembaga, kabel *coaxial*, kabel *twisted pair*, kabel serat optik dan berbagai teknologi *wireless* (Purba dan Efendi, 2020).

2.2 Cache

Cache merupakan perangkat keras atau perangkat lunak yang digunakan untuk menyimpan sesuatu seperti data untuk sementara waktu di lingkungan komputasi (Rahman dan Ikbali, 2019).

Berikut adalah beberapa jenis *cache* :

a. Browser/HTTP Cache

Browser cache membantu mempermudah dan membuat *loading* lebih cepat (Manhas, 2013). *Browser cache*, *web server* tidak perlu lagi melakukan *request* dan transmisi data untuk menampilkan situs yang ingin dikunjungi di *browser* (Ridhalri, 2022).

b. ***OpCode Cache***

OpCode Cache adalah *cache* pada *script* PHP dengan menyimpan hasil kompilasi dari PHP. Proses eksekusi PHP jadi lebih singkat dengan adanya *OpCode Cache*, karena tidak perlu melalui tahapan *parsing* dan kompilasi dan disimpan didalam *memory* (Rahman dan Ikbali, 2019).

c. ***Object Cache***

Object cache, data objek dapat disimpan secara lokal sehingga tidak perlu diambil secara konstan untuk *request* tambahan. *Object cache* dapat membantu meningkatkan kecepatan dan kinerja aplikasi *web* (Rahman dan Ikbali, 2019).

d. ***Page Cache***

Page cache memiliki kemiripan dengan *cache* lain. *Page cache* memiliki manfaat untuk meningkatkan kecepatan waktu *loading* sebuah halaman *website* untuk memberikan *user experience* yang lebih baik (Luthfi Muhammad, Data Mahendra, 2018). *Page cache* menyimpan halaman *web* lengkap untuk ditampilkan di lain waktu kepada pengunjung (Rahman dan Ikbali, 2019).

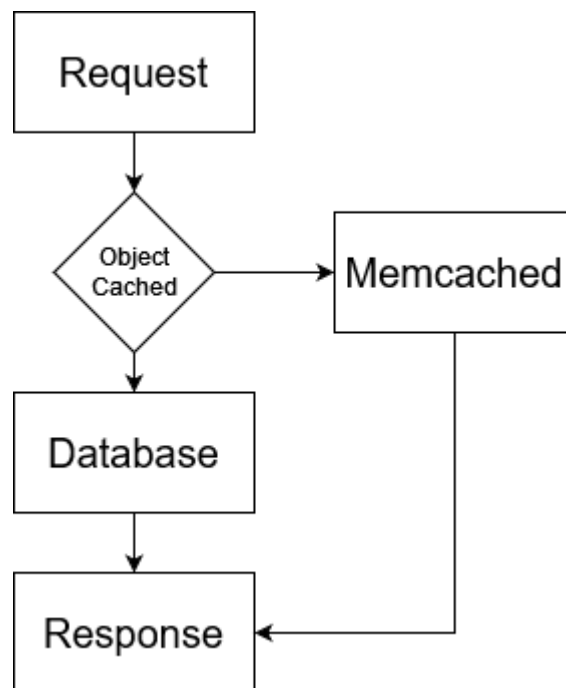
e. ***Content Delivery Network Cache***

CDN cache adalah bentuk penyimpanan data yang lebih luas. *CDN Cache* menambahkan konten situs *web* statis ke *server proxy* yang didistribusikan secara global. Pengunjung dari seluruh dunia memungkinkan untuk mengunduh konten situs anda lebih cepat sehingga mempercepat waktu buka situs (Rahman dan Ikbali, 2019).

2.3 *Memcached*

Memcached adalah sistem *caching memory* terdistribusi yang bersifat umum. *Memcached* digunakan untuk mempercepat *website* dinamis yang menggunakan *database* dengan menyimpan data dan objek di RAM untuk mengurangi jumlah pembacaan ke sumber data eksternal (seperti *database* atau API) (Hidayat dan Rahmatulloh, 2019). *Memcached* adalah perangkat lunak sumber terbuka yang gratis, dilisensikan di bawah lisensi *Revised BSD*. *Memcached* berjalan pada sistem operasi mirip Unix (*Linux* dan *macOS*) serta *Microsoft Windows* (Hidayat & Rahmatulloh, 2019).

Memcached merupakan proses *caching* sederhana untuk mempercepat aplikasi *web*. *Memcached* menyimpan banyak data yang sering diakses dan mengurangi kebutuhan *query* basis data yang berulang, sehingga menghasilkan waktu respon yang lebih cepat. *Memcached* memiliki kelemahan data hanya disimpan sementara dan akan hilang jika instan *Memcached* gagal.



Gambar 2. 1 Proses *Memcached* (Hidayat dan Rahmatulloh, 2019).

2.4 *Redis*

Redis adalah *NoSQL database* berbasis *key-value* dalam menyimpan data dan memiliki popularitas yang tinggi di komunitas pengembangan perangkat (Bustamin, 2021).

Redis memiliki kelebihan *throughput* yang besar dan rendah *response time* karena data disimpan didalan *memory*, selain itu *Redis* juga dapat dijadikan sebagai sistem yang terdistribusi (Kusuma, 2019). *Redis* memiliki kelemahan yaitu dapat menyimpan data seluruhnya di *memory*, yang berarti sensitif terhadap kehilangan data jika terjadi *crash* atau *shutdown*.

2.5 Sistem Terdistribusi

Sistem terdistribusi adalah sebuah sistem yang komponennya berada pada jaringan komputer. Komponen-komponen ini saling berkomunikasi dan

berkoordinasi hanya dengan pengiriman pesan (*message passing*). *Redis* melibatkan lebih dari satu komputer dalam suatu infrastruktur jaringan, baik lokal, internet, maupun nirkabel. Sistem terdistribusi tidak hanya melakukan komunikasi antara satu proses pada satu komputer dengan proses pada komputer yang lain, tetapi juga mempertimbangkan ketersediaan infrastruktur jaringan yang memadai serta dukungan standarisasi sistem yang terbuka (Syahrir, Zulfikri, dan Azwar, 2022).

2.6 *Memtear*

Memtear adalah alat *benchmarking* berperforma tinggi yang di kembangkan oleh *Redis Labs*. *Memtear* memiliki fitur kustomisasi untuk menghasilkan beberapa *workload pattern*. *Memtier* dapat mengatur rasio antara *read and write operations*. *Memtier* memberikan pilihan untuk mengatur jumlah *request* yang akan dilakukan oleh setiap *client* atau durasi waktu untuk menjalankan suatu pengujian. *Memtier* telah digunakan dalam beberapa karya ilmiah sebagai alat perbandingan (Magalhaes, Monteiro, Brayner, dan Moraes, 2021).

2.7 Penelitian Terkait

Tabel 2. 1 Penelitian Sejenis

No	Peneliti	Judul	Metode	Tools yang digunakan	State of The Art
1.	(Febriyani, Sakti Pramukantoro, dan Bakhtiar, 2019)	Perbandingan Kinerja <i>Redis</i> , <i>Mosquitto</i> , dan <i>MongoDB</i> sebagai <i>Message Broker</i> pada <i>IoT Middleware</i>	<i>Load Testing</i>	<i>Wireshark</i>	Pengujian kinerja <i>IoT middleware</i> menunjukkan bahwa <i>Redis</i> unggul dalam <i>CPU Usage</i> saat menulis data, efisiensi <i>memory</i> , <i>runtime</i> , dan kecepatan penanganan data melalui MQTT. <i>MongoDB</i> memiliki kinerja terbaik dalam disk I/O dan skalabilitas, serta unggul dalam <i>time subscribe</i> dan <i>concurrent subscribe</i> . Sementara itu, <i>mosquitto</i> menunjukkan performa terbaik dalam kecepatan menulis data, menerima data melalui CoAP, dan <i>time publish</i> melalui CoAP serta <i>concurrent publish</i> .
2.	(M. L. Cahyadi, Setiawan Heri, dan Mair, 2023)	Analisis Perbandingan Kinerja <i>Web Server Nginx</i> dan <i>Litespeed</i> Menggunakan	<i>Load Testing</i>	<i>Httpperf</i>	Pengujian kinerja <i>web server</i> <i>Nginx</i> dan <i>Litespeed</i> pada VPS dengan OS <i>debian</i> menunjukkan bahwa <i>Nginx</i> lebih unggul dalam hal <i>throughput</i> , <i>connection</i> , dan <i>reply time</i> . Dengan menggunakan <i>httperf</i> , ditemukan

No	Peneliti	Judul	Metode	Tools yang digunakan	State of The Art
		<i>Httpperf</i> Dengan Sistem Operasi Debian			bahwa Nginx memiliki rata-rata <i>throughput</i> 128719.2 KB/s, <i>connection</i> 0.33 ms, dan <i>reply time</i> 0.15 ms, sementara Litespeed memiliki <i>throughput</i> 57405.6 KB/s, <i>connection</i> 0.34 ms, dan <i>reply time</i> 0.23 ms. Hasil ini mengindikasikan bahwa Nginx menawarkan performa yang lebih baik untuk kecepatan dan responsivitas dibandingkan Litespeed.
3.	(Alam dan Dewi, 2022)	<i>Performance Testing Analysis Of Bandungtanginas Application With Jmeter</i>	<i>Load Testing</i>	JMeter	Aplikasi bandungtanginas.id dapat bekerja melayani <i>request</i> tanpa <i>error</i> saat diberikan beban oleh 50 pengguna secara bersamaan. Hasil tes dikategorikan baik jika mengacu pada target SLA (<i>Service Level Agreement</i>).
4.	(Prasetyo dan Salimin, 2021)	Analisis Perbandingan Performa <i>Web Server</i> Docker Swarm dengan <i>Kubernetes Cluster</i>	<i>Load Testing</i>	JMeter	Kubernetes cluster memberikan performa yang bagus dengan <i>response time</i> yang lebih cepat dibanding docker swarm dari <i>load testing</i> 300 <i>thread</i> dan 2000 <i>thread</i> untuk <i>web server</i> nginx yang dilakukan. Docker swarm dapat memberikan

No	Peneliti	Judul	Metode	Tools yang digunakan	State of The Art
					performa yang tidak jauh beda dari Kubernetes cluster dengan menggunakan <i>resource</i> yang lebih efisien.
5.	(Busran dan Ridwan Ahmad, 2020)	Analisis Perbandingan Performa Apache Web Server Dan Nginx Menggunakan Apache Jmeter	Load Testing	JMeter	Penelitian ini menunjukkan bahwa Nginx lebih efisien dan cepat daripada Apache dalam <i>response time</i> dan <i>throughput</i> . Pengujian <i>response time</i> dengan 100, 1000, dan 10000 <i>request</i> , Nginx memiliki rata-rata masing-masing 155ms, 225ms, dan 266ms, dibandingkan Apache dengan 280ms, 365ms, dan 400ms. Hasil pengujian <i>throughput</i> , Nginx juga unggul, mengeksekusi lebih banyak <i>request</i> dalam 100, 500, dan 1000 detik dibandingkan Apache, menunjukkan kinerja yang lebih baik secara keseluruhan.
6.	(Alzaidi dan Vagner, 2022)	<i>Benchmarking Redis and HBase NoSQL Databases using Yahoo Cloud Service Benchmarking tool</i>	Load Testing	Yahoo Cloud Service Benchmark	Penelitian ini menunjukkan bahwa basis data HBase menawarkan kelebihan yang signifikan dalam hal skalabilitas dan fleksibilitas struktural, yang mempermudah pengembangan perangkat lunak.

No	Peneliti	Judul	Metode	Tools yang digunakan	State of The Art
					Pengujian menggunakan Yahoo Cloud Service <i>Benchmarking</i> menunjukkan bahwa HBase memiliki kinerja yang hampir sama dengan <i>Redis</i> saat menggunakan kurang dari 7 <i>thread</i> . HBase menunjukkan <i>throughput</i> yang lebih tinggi dibandingkan <i>Redis</i> ketika jumlah thread ditingkatkan.
7.	(Kabakus, 2020)	<i>A Performance Comparison of Java Cache Memory Implementations</i>	<i>Load Testing</i>	Eksekusi Query	Penelitian ini menilai kinerja berbagai implementasi <i>memory cache</i> . Hasil dari pengujian menunjukkan bahwa arsitektur <i>memory cache</i> sangat mempengaruhi kinerja dan tidak ada satu pun <i>cache</i> Java yang unggul di semua aspek. Kombinasi beberapa <i>cache</i> Java mungkin memberikan kinerja terbaik. <i>caching</i> terdistribusi meningkatkan waktu eksekusi. Perlu dievaluasi <i>memory cache</i> yang mendukung mode terdistribusi dan desain arsitektur <i>cache</i> secara rinci, serta berbagai skenario penghapusan data untuk memahami kinerja dalam skenario spesifik.

No	Peneliti	Judul	Metode	Tools yang digunakan	State of The Art
8.	(Kausar, Nasar, dan Soosaimanickam, 2022)	<i>A Study of Performance and Comparison of NoSQL Databases: MongoDB, Cassandra, and Redis Using YCSB</i>	<i>Load Testing</i>	Yahoo Cloud Service Benchmark	Uji coba pada <i>MongoDB</i> 4.4, <i>Cassandra</i> 4.0.3, dan <i>Redis</i> 6.2.6 dengan berbagai beban kerja hingga 1.000.000 operasi menunjukkan bahwa optimasi NoSQL seperti pemanfaatan <i>cache memory</i> mempercepat waktu eksekusi. <i>Redis</i> unggul dalam kinerja baca karena menggunakan memori volatile. <i>MongoDB</i> lebih baik dalam operasi baca dan pemindaian dibandingkan <i>Cassandra</i> . <i>Cassandra</i> mengungguli <i>Redis</i> dalam pemindaian. <i>MongoDB</i> menunjukkan <i>response time</i> lebih rendah kecuali dalam beban kerja tinggi, menjadikannya basis data <i>NoSQL</i> dengan performa terbaik. Evaluasi <i>NoSQL</i> di cloud dan pengujian basis data lainnya dapat memperluas penelitian ini.
9.	(Banja, Ilić, Kopanja, Zlatković, dkk., 2021)	<i>Microsoft Sql Server And Oracle:Comparativ Performance Analysis</i>	<i>Load Testing</i>	Eksekusi Query	Penelitian ini membandingkan kinerja Microsoft SQL Server dan Oracle. SQL Server mengeksekusi <i>query</i> lebih cepat. Oracle, mendukung berbagai sistem operasi dan bahasa pemrograman, menawarkan keamanan

No	Peneliti	Judul	Metode	Tools yang digunakan	State of The Art
					multi-layer yang lebih kompleks dan fleksibilitas, membuatnya cocok untuk aplikasi yang memerlukan keamanan tinggi dan dukungan <i>multi-platform</i> . Penelitian ini membantu memilih DBMS yang sesuai berdasarkan kebutuhan kinerja, fleksibilitas, dan keamanan.
10.	(Pandey, 2020)	<i>Performance Benchmarking and Comparison of Cloud-Based Databases MongoDB (NoSQL) Vs MySQL (Relational) using YCSB</i>	<i>Load Testing</i>	Yahoo Cloud Service Benchmark	Penelitian ini membandingkan kinerja <i>cloud database</i> MongoDB (NoSQL) dan MySQL (Relasional) menggunakan YCSB. MongoDB unggul dalam penanganan data tidak terstruktur dengan performa lebih baik pada beban kerja baca-tulis tinggi, sementara MySQL lebih baik untuk transaksi dengan integritas data tinggi, cocok untuk aplikasi yang memerlukan konsistensi dan struktur data.

[illegible]

No	Penulis	Judul	Tools						Object																	
			JMeter	Mentear	HTTPERF	YCSB	Wireshark	Eksekusi Query	MongoDB	Apache	Oracle	Microsoft SQL	Nginx	HBase	Redis	Mosquito	Litespeed	Docker Swarm	Kubernetes Cluster	Website	Cassandra	Hazecast	Memcached	Ignite	Cache 2K	Guava
		Menggunakan Apache Jmeter																								
6.	(Alzaidi dan Vagner, 2022)	Benchmarking Redis and HBase NoSQL Databases using Yahoo Cloud Service Benchmarking tool				✓							✓	✓												
7.	(Kabakus, 2020)	A Performance Comparison of Java Cache Memory Implementations					✓														✓	✓	✓	✓	✓	✓
8.	(Kausar, Nasar, dan Soosaimani ckam, 2022)	A Study of Performance and Comparison of NoSQL Databases: MongoDB,				✓		✓						✓						✓						

[illegible]

No	Penulis	Judul	Tools						Object																	
			JMeter	Memtestar	HTTPERF	YCSB	Wireshark	Eksekusi Query	MongoDB	Apache	Oracle	Microsoft SQL	Nginx	HBase	Redis	Mosquitto	Litespeed	Docker Swarm	Kubernetes Cluster	Website	Cassandra	Hazelcast	Memcached	Ignite	Cache 2K	Guava
11.	Nisrina Khairunisa Anwar	Analisis Perbandingan Kinerja Redis dan Memcached sebagai Solusi Server Side Caching dalam Mengoptimalkan Infrastruktur server		✓										✓								✓				

Berdasarkan matriks penelitian pada tabel 2.2, perbedaan penelitian ini dengan beberapa penelitian sebelumnya terletak pada objek pengujian yaitu sistem *cache Redis* dan *Memcached*. Penelitian sebelumnya belum ada yang membandingkan kinerja dari kedua sistem, maka dari itu penelitian ini bertujuan untuk menguji kinerja sistem *cache* lalu membandingkan sistem *cache* mana yang lebih unggul pada berbagai variasi beban pengujian yang diberikan.