

BAB 5 KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, setiap pengujian telah memberikan jawaban untuk menjawab rumusan masalah yang telah dirumuskan sebelumnya. Kesimpulan dari masing-masing hasil tersebut adalah sebagai berikut:

1. Perancangan sistem *caching* pada aplikasi web menggunakan Redis *in-memory database* dengan *framework* Laravel telah berhasil dilakukan. Sistem *caching* diintegrasikan secara efektif ke dalam aplikasi, Hal ini memungkinkan data yang sering diakses disimpan di Redis, sehingga dapat mengurangi beban *query* ke *database* MySQL. Redis sebagai *cache* terbukti efektif dalam mendukung arsitektur aplikasi yang responsif dan efisien.
2. Pengujian performa sistem *caching* berdasarkan *hit ratio*, *response time*, dan *throughput* menunjukkan bahwa penerapan *cache* mampu secara signifikan meningkatkan performa aplikasi web. *Response time* menurun drastis, bahkan hingga tiga kali lebih cepat pada skenario data besar 19000 entri, dan *throughput* meningkat secara signifikan. Selain itu, dari sisi efisiensi algoritma *cache replacement*, algoritma RR (*Random Replacement*) mencatat rata-rata *hit ratio* tertinggi sebesar 62,06%, diikuti oleh LFU, LRU, dan FIFO. Berdasarkan hasil tersebut menandakan bahwa strategi penghapusan *cache* yang tepat dapat mempengaruhi efektivitas *caching* secara langsung.

5.2 Saran

Berdasarkan hasil penelitian, terdapat beberapa aspek yang dapat dikembangkan lebih lanjut untuk menyempurnakan pendekatan yang telah dirancang. Saran untuk penelitian lanjutan adalah sebagai berikut:

1. Mengenai penelitian lanjutan pada cakupan sistem *cache*, terdapat banyak aspek yang dapat ditingkatkan seperti melakukan pengujian pada rentang *dataset* yang lebih besar serta skenario beban kerja yang lebih beragam, seperti menambah beban *user(threads)* dan mengurangi kapasitas *maxmemory*, supaya data lebih sering ter *eviction* dan algoritma yang bisa bekerja lebih maksimal.
2. Penelitian lanjutan dapat menambahkan algoritma *cache replacement* seperti GDSF (*Greedy Dual Size Frequency*), *Similar Cache*, dan lainnya, atau dengan pendekatan *machine learning* untuk prediksi data yang paling sering diakses.
3. Selain Redis, teknologi *caching* lain seperti Memcached atau Varnish dapat digunakan untuk dibandingkan efektivitasnya dalam hal *response time*, *throughput*, dan efisiensi penggunaan memori.