Abstrak

Database as a Service (DBaaS) merupakan model transaksional *database* yang mempunyai kelebihan diantaranya: biaya yang dikeluarkan oleh pengguna akan lebih murah karena tidak membeli *hardware* secara langsung dan tidak membutuhkan banyak sumber daya untuk mengelola *database*, melainkan hanya membeli layanan *database*. Secara umum, DBaaS sering diakses oleh banyak pengguna yang mengakibatkan terjadinya *over-load*. Salah satu solusi untuk mencegah *over-load* adalah membagi-bagi *request* atau beban ke beberapa *database server* sehingga tidak terpusat ke satu *database server* saja. Proses tersebut dinamakan *load balancing*. Pada penelitian ini, metode *load balancing* yang diimplementasikan adalah *static load balancing* dengan algoritma *random* dan *round robin.* Dalam era teknologi modern, pengimplementasian *load balancing* membutuhkan perancangan yang tepat dalam *database server.* Arsitektur *database server* yang mendukung untuk diterapkannya *load balancing* adalah arsitektur *shared-nothing cluster* (SN-Cluster). Hasil pada penelitian ini adalah *random load balancing* dengan SN-Cluster dapat meningkatkan performansi sistem DBaaS berupa *response time*, *throughput*, dan *error rate* pada skenario pengujian pengambilaan dan penginputan data, sedangkan *round robin load balancing* dengan SN-Cluster dapat meningkatkan performansi sistem DBaaS pada skenario pengujian perubahan dan penghapusan data. Di lain sisi, *load balancing* dengan SN-Cluster dapat menangani *concurrency* pada jumlah *client* 100 dan 200, akan tetapi tidak dapat menangani *consistency* untuk semua jumlah *client.*

Kata kunci : DBaaS, *over-load, load balancing, shared-nothing cluster, response time, throughput, error rate, concurrency, consistency.*