

## BAB V

### SIMPULAN DAN SARAN

#### 5.1. Simpulan

Sesuai dengan kebutuhan Perusahaan X untuk pembangunan *environment database* yang dimiliki, implementasi *connection pooling* dan *load balancing* pada *environment* yang digunakan oleh Perusahaan X dapat menjawab permasalahan yang dihadapi. Implementasi *tools* PGPOOL-II dan juga PGBOUNCER berhasil merealisasikan fitur *load balancer* dan juga *connection pooling* untuk *environment database* PostgreSQL dari Perusahaan X. Antrian dari koneksi dan manajemen buka/tutup koneksi dari *database* dapat ditangani oleh *tools* PGBOUNCER. Sedangkan untuk pembagian *load* transaksi (*query select*) antara *Database Master* dan *Database Master* dapat ditangani oleh *tools* PGPOOL-II, meskipun PGPOOL-II ini bisa digunakan sebagai *tools* untuk *connection pooling* juga tetapi performa *connection pooling* dari PGBOUNCER lebih baik dari PGPOOL-II. Adanya PGPOOL-II juga berhasil mencegah adanya penggunaan *resource* yang berlebih dari 1 *database* saja karena *load* dari transaksinya dapat dibagi antara *Database Master* dan juga *Database Slave*.

Performa dari implementasi *connection pooling* dan *load balancing* pada *environment database* PostgreSQL ini dapat diukur dengan bantuan *tools* pgbench. *Tools* pgbench ini dapat mendapatkan *latency* dan juga TPS (*Transaction per Second*) dari kombinasi *tools* PGPOOL-II dan PGBOUNCER. *Latency* yang rendah menggambarkan bahwa *response* pada transaksi dari *user* ke dalam *database* hingga akhirnya mengirimkan data yang ingin diakses ke *user* cepat dan berbanding terbalik dengan TPS, TPS yang tinggi menggambarkan kekuatan dari *database* untuk memproses transaksi yang masuk dalam 1 detik. Nilai *latency* dan juga TPS yang didapatkan dari *tools* pgbench bisa digunakan untuk

membandingkan skenario-skenario yang terbentuk dari kombinasi PGPOOL-II dan PGBOUNCER. Didapatkan dari 3 skenario yang terbentuk berdasarkan kombinasi PGPOOL-II dan PGBOUNCER skenario dengan TPS tertinggi dan *latency* terendah adalah Skenario Dua. Perbandingan 3:7 juga lebih tepat untuk melakukan *load balancing* jika dibandingkan dengan perbandingan 2:8.

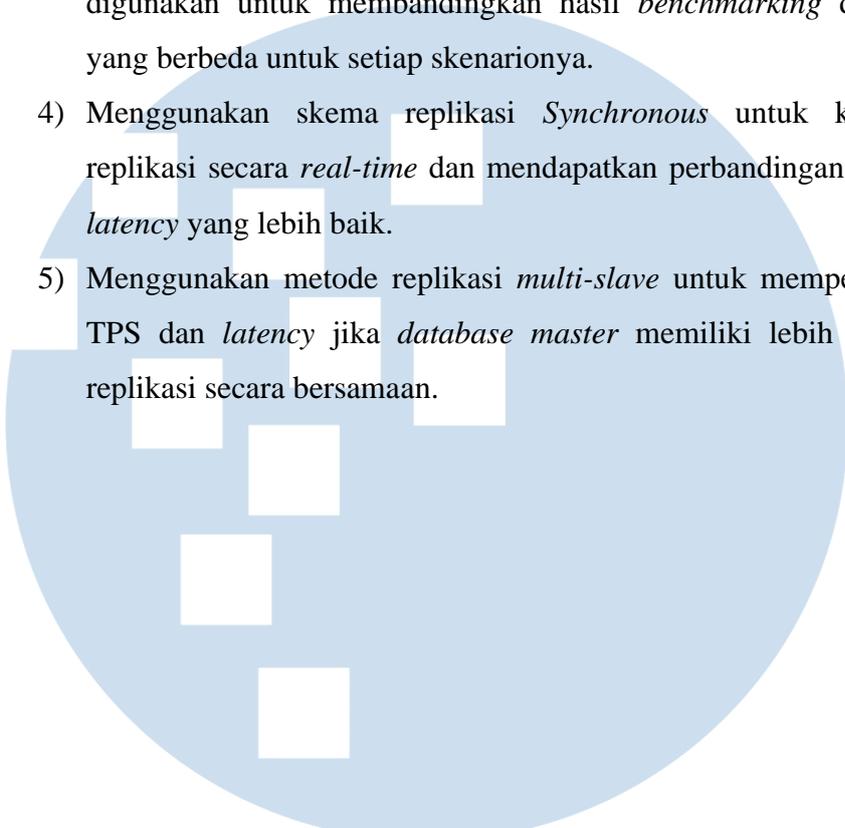
Berdasarkan hasil pengukuran performa dengan *tools* *pgbench*, didapatkan skenario terbaik adalah Skenario Dua, dimana PGBOUNCER berperan sebagai *connection pooling* dan PGPOOL-II hanya berperan sebagai *load balancer*. Nilai dari *latency* setiap percobaan untuk Skenario Dua selalu memiliki nilai terendah dan disusul dengan nilai TPS yang selalu paling tinggi baik TPS *including* dan juga TPS *excludingnya*. Dapat direkomendasikan bahwa penggunaan PGBOUNCER sebagai *connection pooling* dan PGPOOL-II hanya sebagai *load balancer* tanpa mengaktifkan *connection pooling* dari PGPOOL-II paling baik untuk environment *database* Perusahaan X.

## 5.2. Saran

Berdasarkan dengan hasil pembangunan dan juga pengolahan data dari implementasi *connection pooling* dan *load balancing* dengan menggunakan *tools* PGPOOL-II dan PGBOUNCER yang telah dilakukan untuk memenuhi kebutuhan Perusahaan X terdapat beberapa saran untuk penelitian selanjutnya, yaitu:

- 1) Pembangunan *environment database* PostgreSQL ditambahkan fitur *automatic failover* sebagai salah satu bentuk usaha untuk *disaster recovery* (DR) dan melihat apakah adanya fitur *automatic failover* akan mempengaruhi TPS dan *latency* dari *database*.
- 2) Menggunakan spesifikasi yang sama dengan *environment production* untuk memperlihatkan kondisi sebenarnya dari *environment database* yang akan digunakan.

- 3) Menggunakan *tools benchmarking* lainnya diluar *tools pgbench* yang digunakan untuk membandingkan hasil *benchmarking* dari *tools* yang berbeda untuk setiap skenarionya.
- 4) Menggunakan skema replikasi *Synchronous* untuk kebutuhan replikasi secara *real-time* dan mendapatkan perbandingan TPS dan *latency* yang lebih baik.
- 5) Menggunakan metode replikasi *multi-slave* untuk memperlihatkan TPS dan *latency* jika *database master* memiliki lebih dari satu replikasi secara bersamaan.



# UMMN

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA