

BAB III METODOLOGI PENELITIAN

3.1. Gambaran Umum Objek Penelitian

Objek dari ini adalah Perusahaan X yang merupakan salah satu perusahaan *multi-finance* yang bergerak di daerah Indoensia dimana nama dari perusahaan ini tidak dapat disebutkan. Perusahaan X ini memiliki sistem yang bekerja selama 24/7, Perusahaan X ini juga memiliki berbagai *database* yang digunakan untuk menyimpan data penting. Salah satu *database* yang digunakan adalah PostgreSQL, dimana *database* PostgreSQL ini digunakan untuk menyimpan data transaksi harian para *partner* dari Perusahaan X. *Connection pooling* dan *Load Balancer* akan digunakan untuk 5 *database* PostgreSQL yang tereplikasi sehingga terdapat 10 *database* PostgreSQL dan 5 server untuk *Connection pooling* dan *Load Balancer*.

3.2. Observasi

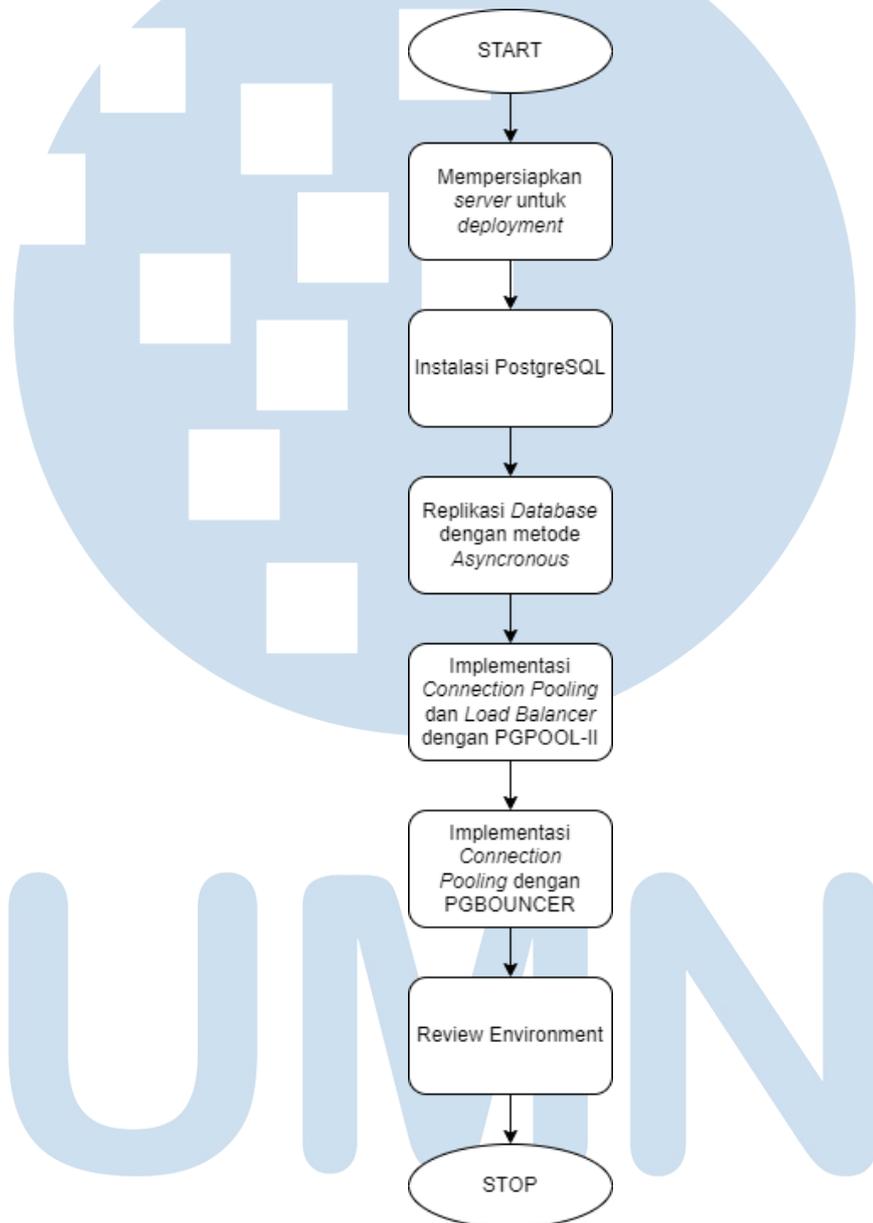
Hal yang pertama dilakukan adalah melakukan pengumpulan informasi mengenai *environment database* PostgreSQL. Dimana dicari tahu mengenai kemampuan dan bentuk replikasi dari PostgreSQL sendiri. Kemudian melakukan pengumpulan informasi mengenai fitur *connection pooling* dan *load balancer* pada PostgreSQL, dengan detail sebagai berikut:

- 1) Penggunaan *pgbouncer* sebagai *connection pooling* dan *pgpool-II* sebagai *load balancer*.
- 2) Penggunaan *pgpool-II* sebagai *connection pooling* dan *load balancer*.

Selain itu dihimpun juga informasi mengenai penggunaan *pgbench* sebagai fitur pada PostgreSQL yang akan digunakan untuk mengumpulkan data TPS pada *environment database* PostgreSQL.

3.3. Konfigurasi Environment

Setelah menghimpun semua informasi kemudian dilakukan pembuatan *environment database* dengan *flowchart* kerja sebagai berikut:



Gambar 3. 1 *Flowchart* proses konfigurasi *Environment Database*

Gambar 3.1 menggambarkan proses konfigurasi *environment database* PostgreSQL. Dengan proses lebih rinci sebagai berikut:

1) Mempersiapkan Server *Deployment*

Mempersiapkan server yang akan digunakan untuk *deployment* dimana dibutuhkan 3 server untuk environment *database* dengan rincian sebagai berikut:

1) Server *Database Master*, dengan spesifikasi:

- a. Memori : 8 GB
- b. Storage : 100 GB
- c. CPU : 4 cores

2) Server *Database Slave*, dengan spesifikasi:

- a. Memori : 8 GB
- b. Storage : 100 GB
- c. CPU : 4 cores

3) Server PGPOOL-II dan PGBOUNCER, dengan spesifikasi:

- a. Memori : 4 GB
- b. Storage : 25 GB
- c. CPU : 4 cores

Spesifikasi yang digunakan merupakan $\frac{1}{2}$ dari besaran spesifikasi yang digunakan dalam *environment production*.

2) Instalasi PostgreSQL

Melakukan instalasi *database* PostgreSQL pada 2 server *database* yaitu untuk *database master* dan *slave* dengan menggunakan versi *database* PostgreSQL yang sama.

3) Replikasi *Database* dengan Metode *Asynchronous*

Melakukan replikasi pada *database* dimana server 1 akan menjadi *database master* dan server 2 akan menjadi *database slave* (menerima replikasi dari *database* server 1. Yang dimana replikasi yang digunakan adalah metode repikasi *Asynchronous*

4) Implementasi *Connection pooling* dan *Load Balancer* dengan PGPOOL-II

Melakukan instalasi pgpool-II pada server 3 dan melakukan konfigurasi parameter untuk setiap *database* dan juga konfigurasi parameter pada *file* konfigurasi dari pgpool.

5) Implementasi *Connection pooling* dengan PGBOUNCER

Melakukan instalasi pgbouncer pada server 3 dan melakukan koneksi pgbouncer ke pgpool-II di server 3.



3.4. Pengumpulan Data

Melakukan pengumpulan data dengan menggunakan `pgbench`. Data yang diambil adalah TPS dan *connection latency* dari *database* dengan metode *load test*. Data diambil dengan menggunakan fitur *benchmarking* pada PostgreSQL yaitu `pgbench`. Inisialisasi *database testing* dengan nama `db_bench` menggunakan `pgbench` dengan *command*:

```
pgbench -U postgres -p *pgbouncer_port* -h *pgbouncer_host*  
-i -s 50 db_bench
```

Yang artinya melakukan inisialisasi pada `db_bench` dengan *scaling factor* 50 yang artinya nantinya isi dari data akan 50 kali lebih besar dari *setting default* `pg_bench`. Kemudian dilakukan *benchmarking* dengan merubah jumlah *client* dan transaksi untuk perbandingan TPS. Data yang diambil berdasarkan pada 3 skenario yang terbentuk dari kombinasi antara *tools* PGBOUNCER dan PGPOOL-II, dimana skenario yang digunakan adalah sebagai berikut:

a) Skenario 1 (Menggunakan PGBOUNCER dan PGPOOL-II)

Pada skenario ini PGBOUNCER akan berperan sebagai *connection pooling* dari *database* PostgreSQL dan disusul dengan PGPOOL-II sebagai *connection pooling* dan *load balancer* dari *database*. Pada skenario ini terdapat 2 fitur *connection pooling* yang berjalan bersamaan yaitu dari PGBOUNCER dan juga dari PGPOOL-II.

b) Skenario 2 (Menggunakan PGPOOL-II)

Pada skenario ini yang digunakan hanya *tools* PGPOOL-II saja. Skenario ini menggunakan fitur *connection pooling* dan *load balancer* dari *tools* PGPOOL-II.

c) Skenario 3 (Menggunakan PGBOUNCER dan PGPOOL-II)

Pada skenario ini digunakan *tools* PGBOUNCER dan PGPOOL-II, dimana PGBOUNCER akan berperan sebagai *connection pooling* dan PGPOOL-II akan berperan sebagai *load balancer* saja. Pada skenario ini fitur *connection pooling* dari *tools* PGPOOL-II akan dinon-aktifkan dan hanya menggunakan fitur *load balancer* nya saja.

3.5. Metode Kuantitatif

Setelah data terhimpun, kemudian data tersebut akan dianalisa untuk melihat TPS rata-rata dari setiap skenario, dan membandingkan TPS terbaik yang bisa didapatkan dengan melihat nilai TPS terbesar pada skenario tertentu. Dilakukan juga analisa pada nilai *latency* dimana pada *latency* dilihat skenario dengan *latency* terkecil.

