

BAB III

METODOLOGI PENELITIAN

3.1 Gambaran Umum Objek Penelitian

Data yang akan digunakan dalam penelitian kali ini adalah data dari PT. GKT. Beberapa data yang akan digunakan adalah data produk, data *suppliers*, data *customers*, dan data transaksi penjualan produk dari perusahaan. PT. GKT adalah salah satu perusahaan *retail* produk-produk yang berkaitan dengan listrik, seperti lampu, kipas angin, kabel, steker, dan lain-lain. Sampai saat ini, sudah ada lebih dari ratusan produk listrik yang dijual ke pelanggan setiap harinya.

PT. GKT saat ini berlokasi di daerah Senen, Jakarta Pusat dan sudah berdiri sejak tahun 2022. Secara garis besar, proses bisnis di perusahaan ini dilakukan dengan membeli produk-produk listrik terlebih dahulu ke perusahaan-perusahaan *supplier* barang-barang listrik. Setelah itu, produk-produk tersebut akan dijual kembali dalam bentuk eceran dan grosir. Proses pencatatan dan penyimpanan transaksi penjualan saat ini masih dilakukan secara manual dan belum disimpan ke dalam *database*. Jumlah transaksi penjualan saat ini setiap harinya berkisar di 400 sampai dengan 500 transaksi penjualan produk.

Saat ini, dengan semakin berkembangnya proses bisnis di perusahaan, penggunaan *database* diperlukan untuk membantu memudahkan penyimpanan transaksi penjualan dan pembelian yang semakin banyak. Penggunaan *database* pada perusahaan juga nantinya dapat membantu untuk melakukan analisis data transaksi penjualan. Penggunaan *database* ini diharapkan juga bisa digunakan untuk keperluan jangka panjang perusahaan dalam menyimpan data.

3.2 Metode Penelitian

3.2.1 *System and Hardware Requirements*

Setiap DBMS memiliki minimal *requirements* yang perlu untuk dimiliki oleh setiap komputer atau *laptop* agar DBMS dapat beroperasi dengan baik. *Database* seperti MySQL, PostgreSQL, maupun Microsoft SQL Server juga memiliki *requirements* yang berbeda agar dapat berjalan dengan efektif di

laptop. Tabel 3.1 merupakan perbandingan minimum *system* dan *hardware requirements* yang perlu dimiliki sebelum mengunduh DBMS yang dibutuhkan.

Tabel 3. 1 Perbandingan Minimum *System* dan *Hardware Requirements Database* Penelitian

Requirements	MySQL 8.0.41 [55]	PostgreSQL 17.2 [56]	Microsoft SQL Server 2022 [57]
Operation System (OS) Version	Windows 11.	Windows 10.	Windows 10 1607.
RAM	Minimal 4 GB RAM.	Minimal 2 GB RAM.	Minimal 512 MB untuk Express Edition dan minimal 1 GB untuk edisi lainnya. Untuk Express Edition lebih baik 1 GB, sedangkan untuk edisi lainnya lebih baik 4 GB.
Processor Type	MySQL hanya mendukung <i>processor</i> x64 saja.	Mendukung <i>processor</i> x64.	Mendukung <i>processor</i> x64, seperti AMD Opteron, AMD Athlon 64, Intel Xeon didukung dengan Intel EM64T, dan Intel Pentium IV didukung dengan EM64T.
Screen Resolution	1024 x 768.	800 x 600	800 x 600 (Super-VGA).
Processor Speed	Xeon 3GHz atau Dual Core 2GHz, setara dengan AMD CPU.	Minimal 1 GHz.	Minimal 1.4 GHz, lebih baik jika mencapai 2.0 GHz.

Tabel 3.1 merupakan informasi terkait *system* dan *hardware requirements* pada masing-masing RDBMS sesuai dengan versi *database* yang akan digunakan untuk pengujian *query*. Berdasarkan informasi dari Tabel 3.1, *laptop* yang akan digunakan adalah *laptop* Dell dengan model berupa Dell Vostro 3405. Spesifikasi *laptop* memiliki OS berupa Microsoft Windows 11, *processor* AMD Ryzen 7 3700U with Radeon Vega Mobile Gfx. Selain itu, *memory* RAM yang digunakan di *laptop* sebesar 16 GB.

3.2.2 Alur Penelitian

Setiap DBMS memiliki minimal *requirements* yang perlu untuk dimiliki oleh setiap komputer atau *laptop* agar DBMS dapat beroperasi dengan baik.

Database seperti MySQL, PostgreSQL, maupun Microsoft SQL Server juga memiliki *requirements* yang berbeda agar dapat berjalan dengan efektif di *laptop*. Tabel 3.1 merupakan perbandingan minimum *system* dan *hardware requirements* yang perlu dimiliki sebelum mengunduh DBMS yang dibutuhkan.

Alur penelitian kali ini akan mengikuti siklus *database system development lifecycle* (DSDLC). Walaupun begitu, alur pengembangan *database* akan dibuat lebih sederhana dibandingkan siklus awal. Ada dua pendekatan yang umum digunakan dalam DSDLC, yaitu pendekatan *top-down* dan pendekatan *bottom-up* [19]. Tabel 3.2 adalah perbandingan pendekatan antara *top-down* dan *bottom-up* pada DSDLC.

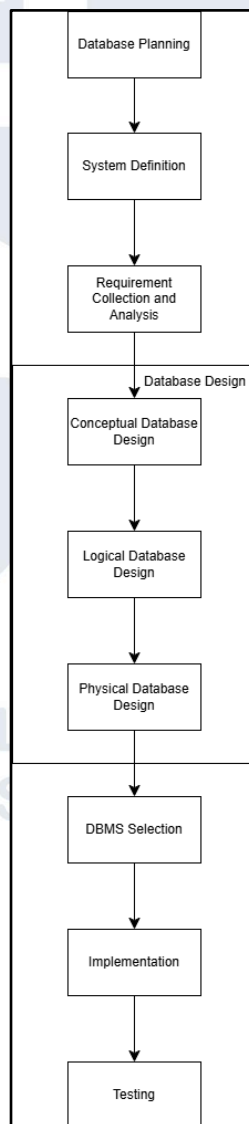
Tabel 3. 2 Perbandingan Pendekatan *Top-down* dan *Bottom-up* [19], [58]

Indeks Perbandingan	<i>Top-down</i>	<i>Bottom-up</i>
Definisi	Perancangan <i>database</i> dari <i>level</i> yang tinggi ke <i>detail</i> yang lebih spesifik.	Perancangan <i>database</i> dari tahapan paling dasar, lalu dilanjutkan dengan tahapan pembuatan struktur <i>database</i> yang lebih kompleks.
Proses	Tahapan awal dari pendekatan <i>top-down</i> dimulai dengan mengidentifikasi entitas terlebih dahulu, baru diikuti dengan <i>detail</i> yang lebih spesifik, seperti atribut yang digunakan dari masing-masing entitas.	Pendekatan <i>bottom-up</i> dimulai dari mengidentifikasi atribut pada tabel, lalu dilanjutkan dengan tahapan pembuatan struktur <i>database</i> yang lebih kompleks, seperti pengelompokan yang merepresentasikan jenis-jenis entitas dan hubungan antar entitas.
Kompleksitas Penggunaan	Lebih mudah untuk sistem yang lebih kompleks.	Lebih mudah untuk sistem yang lebih sederhana, namun kurang bagus untuk sistem yang sudah kompleks.
Skala Penggunaan	Diperuntukkan untuk <i>database</i> besar dengan atribut yang banyak.	Diperuntukkan untuk <i>database</i> sederhana dengan atribut tidak banyak.

Berdasarkan hasil perbandingan ini, pendekatan yang dipilih dalam penelitian ini adalah pendekatan *bottom-up* karena atribut yang digunakan di dalam setiap tabel di *database* tidak terlalu banyak, hanya ada beberapa atribut saja yang memang berkaitan dengan entitasnya. Selain itu, skala penggunaan *database* ini masih masuk ke dalam *database* sederhana karena skema yang dibuat lebih berkaitan dengan transaksi penjualan saja. Oleh karena itu,

pendekatan *bottom-up* dinilai lebih cocok untuk perancangan *database* penelitian.

Setelah menentukan pendekatan yang dilakukan dalam perancangan *database*, selanjutnya adalah penjelasan terkait rangkaian proses DSDLC yang akan dijalankan di dalam penelitian. DSDLC dipilih sebagai alur penelitian ini karena penelitian ini berfokus pada pengembangan *database*, sehingga pendekatan siklus pengembangan *database* lebih sesuai dengan kebutuhan. Gambar 3.1 merupakan rangkaian alur untuk pengembangan DSDLC pada penelitian.



Gambar 3. 1 Alur Penelitian *Database System Development Lifecycle* (DSDLC)

Berdasarkan gambar alur penelitian pada Gambar 3.1, ada tujuh tahapan siklus pengembangan *database* yang akan dilakukan di dalam penelitian, sedikit berbeda dengan siklus awal yang memiliki sembilan tahapan. Berikut adalah penjelasan masing-masing tahapan yang akan dilakukan di dalam penelitian.

1) *Database Planning*

Database planning adalah tahapan untuk merencanakan tujuan dari pengembangan *database* dan alur rencana pembuatan *database* dengan cara yang paling efektif dan efisien. Mengetahui tujuan dari pengembangan *database* bertujuan agar hasil dari perancangan *database* nantinya bisa sesuai dengan tujuan penelitian.

2) *System Definition*

Setelah mengetahui tujuan dari pengembangan *database*, tahapan selanjutnya adalah tahapan *system definition*. Tahapan ini dilakukan untuk menentukan batasan-batasan dari sistem *database* yang ingin dirancang.

3) *Requirement Collection and Analysis*

Setelah menentukan tujuan dan batasan dari sistem *database* yang akan dibuat, tahapan selanjutnya adalah tahapan *requirement collection and analysis*. Tahap ini adalah tahapan untuk mengumpulkan *requirements* yang diperlukan untuk mengembangkan *database*. *Requirements* yang dikumpulkan adalah informasi dan data yang ada di perusahaan untuk melakukan pengembangan *database*.

4) *Database Design*

Pembuatan *database design* akan dilakukan dengan tiga tahapan, yaitu *conceptual database design*, *logical database design*, dan *physical database design*. Berikut adalah penjelasan ketiga tahapan *database design* yang akan dilakukan dalam proses penelitian.

1) *Conceptual database design*

Tahapan perancangan *database* dimulai dari mengidentifikasi atribut yang dimiliki terlebih dahulu pada setiap data, lalu mengidentifikasi entitas yang sesuai dan mewakili atribut-atributnya.

2) *Logical database design*

Tahapan perancangan *database* selanjutnya adalah mengelompokkan atribut ke dalam masing-masing entitas, menentukan relasi antara entitas yang akan ditambahkan, dan memilih beberapa atribut yang dapat dijadikan *primary key candidate* dalam satu entitas. Selain itu, proses normalisasi *database* juga akan dilakukan pada tahapan *logical database design*, mulai dari *unnormalized forms* sampai dengan *third normal form* (3NF).

3) *Physical database design*

Hasil *logical database design* yang sudah dibuat akan diubah menjadi *physical database design* yang akan disesuaikan dengan DBMS. Pada *physical database design*, akan ditampilkan nama entitas, atribut dari entitas, tipe data setiap atributnya, dan *primary key* dari setiap entitas. Terakhir, hasil dari entitas tersebut akan dibuat ke bentuk *Entity Relationship Diagram* (ERD) untuk menunjukkan relasi semua entitas yang sudah dibuat dari awal proses *design*.

5) **DBMS Selection**

Tahapan selanjutnya adalah menentukan DBMS yang akan diimplementasikan ke dalam penelitian. Ada tiga RDBMS yang akan digunakan untuk penelitian, yaitu MySQL, PostgreSQL, dan Microsoft SQL Server. Dalam penelitian ini, *database* yang akan dibandingkan adalah MySQL, PostgreSQL, dan Microsoft SQL Server. Versi MySQL yang digunakan adalah MySQL 8.0.41 dengan bantuan *tools* XAMPP, phpMyAdmin, dan MySQL Workbench. Versi PostgreSQL yang digunakan adalah PostgreSQL 17.2 dengan

bantuan GUI berupa pgAdmin 4. Terakhir, untuk Microsoft SQL Server akan menggunakan Microsoft SQL Server 2022 - 16.0.1000.6 (x64) dengan bantuan GUI berupa SQL Server Management Studio (SSMS).

6) *Implementation*

Tahapan *implementation* adalah tahapan untuk melakukan implementasi *database* yang telah dirancang ke RDBMS yang telah ditentukan. Sebelum menuliskan *query* SQL ke *database*, tahapan implementasi yang perlu dilakukan adalah membuat *database* MySQL, PostgreSQL, dan Microsoft SQL Server sesuai dengan rancangan *database design* yang sudah dibuat sebelumnya pada tahap keempat. Hasil akhir dari pembuatan skema *database* akan dibuat di *website* dbdiagram.io, sehingga untuk pembuatan *database* dan tabel *database* bisa langsung didapatkan dengan *export query* SQL sesuai dengan tipe *database*.

Setelah struktur tabel berhasil dibuat, nantinya, kolom data akan disesuaikan terlebih dahulu sesuai dengan nama atribut pada hasil ERD dan disimpan ke dalam *file* .CSV terlebih dahulu agar memudahkan untuk memetakan nama kolom ke masing-masing atribut *database* dalam proses *import*. *File* yang digunakan adalah .CSV karena pada ketiga RDBMS yang digunakan dalam penelitian, salah satu tipe *import* yang tersedia adalah dalam bentuk *file* .CSV. Nama *file* .CSV juga disamakan terlebih dahulu sesuai dengan nama entitas pada ERD agar memudahkan ketika proses *import* data berlangsung.

Dalam implementasi *query* SQL, nantinya akan dilakukan pengujian dengan total sepuluh *query* menggunakan *SELECT statement*. Struktur penulisan yang akan diujikan juga beragam, mulai dari *single table*, *join tables*, *multiple join tables*, *subquery*, dan *common table expression* (CTE). Setelah mendapatkan data hasil pengujian sebelum dilakukan optimasi struktur penulisan *query*,

barulah optimasi *database* dibuat dan diujikan kembali dengan *query* yang sama.

7) *Testing*

Setelah melakukan penulisan *query* pada MySQL, PostgreSQL, dan Microsoft SQL Server, tahapan terakhir adalah melakukan pengujian *query* sebelum dioptimalkan struktur penulisan *query* dan pengujian *query* sesudah dioptimalkan struktur penulisan SQL-nya. Ada beberapa *tools* yang dapat digunakan untuk melakukan pengujian *database*, contohnya seperti Apache JMeter dan SysBench. Tabel 3.3 adalah tabel penjelasan komparasi antara *tools* pengujian dengan Apache JMeter dengan SysBench.

Tabel 3. 3 Perbandingan Apache JMeter dan SysBench

Indeks Perbandingan	Apache JMeter	SysBench
Kompabilitas <i>Operating System</i> (OS)	Windows dan Linux.	Lebih dikhususkan untuk Linux dan MacOS.
Jenis <i>Database</i> yang Dapat Digunakan	MySQL, PostgreSQL, Microsoft SQL Server, dan <i>database-database</i> lainnya.	Hanya MySQL dan PostgreSQL saja.
Kemudahan Penggunaan	Dokumentasi <i>online</i> lengkap dan mudah digunakan.	Ada dokumentasi <i>online</i> dan mudah digunakan.

Berdasarkan penjelasan dari Tabel 3.3, *tools* Apache JMeter lebih baik untuk digunakan dalam pengujian penelitian. Apache JMeter dipilih karena proses pengujian dilakukan dengan menggunakan *operating system* (OS) Windows, sehingga SysBench kurang cocok digunakan dalam penelitian ini. Selain itu, pilihan *database* yang digunakan pada Apache JMeter juga lebih banyak dan variatif dibandingkan dengan SysBench, sehingga Apache JMeter menjadi pilihan *tools* pengujian yang paling cocok dalam penelitian.

Proses pengujian nantinya akan dijalankan sebanyak lima kali dengan jumlah *thread* yang akan ditingkatkan. Pada pengujian pertama, jumlah *thread* yang akan digunakan adalah 10, lalu pengujian kedua ditingkatkan menjadi 30 *thread*, selanjutnya pada pengujian ketiga, jumlah *thread* yang digunakan adalah 50 *thread*.

Pada pengujian keempat, jumlah *thread* yang akan digunakan adalah 75 *thread* dan pada pengujian kelima, jumlah *thread* yang akan digunakan adalah 100 *thread*. Proses pengujian nantinya akan dilakukan secara *looping* selama satu menit untuk mendapatkan hasil *average response time*, *throughput*, dan juga *error rate* dari ketiga RDBMS yang dijadikan objek penelitian.

3.3 Teknik Pengumpulan Data

Data yang akan digunakan dalam penelitian ini berasal dari PT. GKT. Sumber data ini dipilih karena perusahaan ini merupakan perusahaan yang menjual produk dalam bentuk barang, sehingga data menjadi lebih mudah untuk diimplementasikan dalam bentuk ERD dan diproses ke dalam RDBMS. Berikut adalah penjelasan terkait beberapa teknik pengumpulan data yang digunakan untuk kebutuhan penelitian.

3.3.1 Studi Pustaka

Pengumpulan data pada penelitian dimulai dengan melakukan studi pustaka. Proses pengumpulan data dengan studi pustaka dilakukan dengan mencari beberapa jurnal penelitian yang memiliki kesesuaian dengan penelitian yang berkaitan dengan performa *database*, terutama dari sisi struktur penulisan *query* SQL. Pencarian dari beberapa sumber ini dilakukan untuk memahami referensi terkait analisis *performance database* yang dapat membantu untuk melakukan penelitian.

3.3.2 Wawancara

Selain studi pustaka, pengumpulan data yang dilakukan juga adalah wawancara. Wawancara dilakukan bersama dengan salah satu narasumber yang bekerja di perusahaan tersebut sebagai *admin*, bernama Ibu Imel. Wawancara ini dilakukan dengan tujuan untuk mendapatkan informasi yang lebih mendalam terkait aspek yang berkaitan dengan penelitian, seperti alur proses bisnis yang ada di perusahaan sehingga dapat memudahkan untuk membuat proses perancangan *database* yang sesuai dengan kebutuhan penelitian. Wawancara ini juga bertujuan dapat lebih memahami terkait kebutuhan sampel data yang akan dipakai dalam penelitian. Hasil dari

wawancara dapat digunakan untuk kebutuhan perancangan *Entity Relationship Diagram* (ERD) pada bab selanjutnya agar hasilnya bisa sesuai dengan proses bisnis dari perusahaan.

3.4 Teknik Analisis Data

Proses analisis data akan dimulai dari pembuatan *database* terlebih dahulu sesuai dengan hasil rancangan *Entity Relationship Diagram* (ERD) yang sudah dibuat. Tahapan selanjutnya adalah mempersiapkan dua tipe *query* dengan *result* yang sama untuk *database* MySQL, PostgreSQL, dan Microsoft SQL Server sebagai RDBMS untuk penelitian. Setelah menyiapkan kedua jenis *query* SQL, nantinya *query* tersebut akan dipindahkan ke Apache JMeter untuk dilakukan pengujian performa.

Proses pengecekan performa yang akan dicek adalah performa dari *response time* dan *throughput* di Apache JMeter. *Query* juga akan dijalankan lima kali untuk mendapatkan nilai *average response time* untuk kedua jenis *query* agar hasilnya dapat dibandingkan untuk mengetahui struktur penulisan *query* yang lebih efektif. Hasil dari pengujian di Apache JMeter yang akan dilihat adalah di bagian *Listener*. Pada bagian *Listener*, ada beberapa fitur yang dapat dilihat, yaitu *View Results Tree*, *View Results in Table*, dan *Summary Report*. Beberapa fitur pada *Listener* ini dapat membantu menganalisis kinerja *database* yang akan diujikan.

Masing-masing *Listener* pada Apache JMeter memiliki fungsinya tersendiri. *View Results Tree* dan *View Results in Table* dapat digunakan untuk melihat hasil *sample response time* dari *sample query* SQL yang sudah dimasukkan ke masing-masing *JDBC Request*. Hasil akhir dari kedua *Listener* ini adalah dalam bentuk tabel. Selanjutnya, untuk *Summary Report* dapat digunakan untuk melihat hasil dari *response time* dan *throughput* pada masing-masing *query*, *average response time*, dan *average throughput*. Untuk melihat hasil *Summary Report* dalam bentuk grafik, dapat menggunakan *Listener* seperti *Aggregate Graph* agar dapat terlihat *minimal*, *average*, dan *maximal* dari *response time* maupun *throughput* pada *sample query*.