

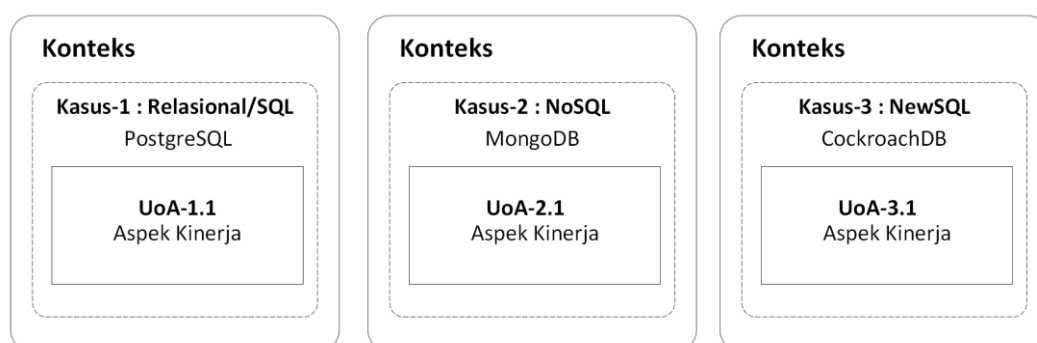
## BAB III

### METODE PENELITIAN

#### 3.1 Desain Penelitian

Desain penelitian yang digunakan dalam penelitian ini adalah *Case Study* atau Studi Kasus, yakni sebuah desain penelitian yang menyelidiki secara empiris fenomena sementara dalam kasus nyata ketika batasan antara fenomena dan konteks yang nyata tidak jelas terlihat, menghimpun dari berbagai sumber serta memanfaatkan pengembangan teoritis sebelumnya sebagai panduan dalam menganalisis (Yin, 2003). Batasan yang dimaksud yaitu tidak adanya kontrol eksperimen dan manipulasi yang diterapkan, sehingga hal inilah yang membedakan dengan desain penelitian eksperimental (Benbasat dkk., 1987). Lebih lanjut fenomena tersebut dapat berupa fenomena yang umum terjadi pada keilmuan rekayasa perangkat lunak (Runeson dkk., 2012).

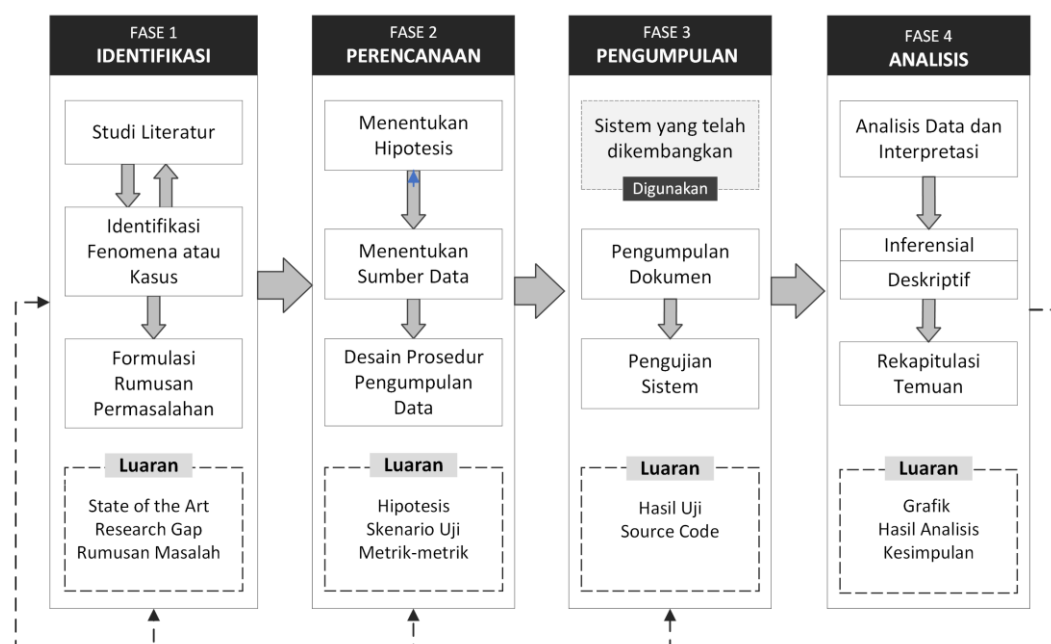
Adapun fenomena yang diangkat pada penelitian ini yaitu berbagai jenis basis data yang telah ada dan banyak digunakan dalam pengembangan perangkat lunak, antara lain jenis relasional menggunakan PostgreSQL, NoSQL menggunakan MongoDB dan NewSQL menggunakan CockroachDB. Sehingga penelitian ini termasuk ke dalam jenis *multiple case* dengan pendekatan holistik (Yin, 2003), karena hanya memiliki satu unit yang dianalisis setiap kasusnya (*Unit of Analysis* atau UoA) yaitu aspek kinerja. Selanjutnya sampel dan populasi pada desain penelitian *case study* umumnya tidak ditentukan, sehingga jumlah sampel dianggap sebagai tunggal (Schoch, 2020). Untuk mempermudah pemahaman terkait desain penelitian ini, digunakan visualisasi dalam bentuk diagram pada **Gambar 3.1** ini.



**Gambar 3.1** Skema Desain Penelitian

### 3.2 Tahapan Penelitian

Proses atau tahapan penelitian yang digunakan dalam penelitian ini diadaptasi dan dimodifikasi dari sumber literatur (Runeson dkk., 2012). Terdapat empat fase yang dilakukan dengan masing-masing fase memiliki sub tahapan dan luaran yang dihasilkan. Setiap fase dapat bersifat iteratif, namun dalam penelitian ini jumlah iteratif yang dilakukan hanya satu kali. Untuk mempermudah pemahaman terhadap tahapan yang dilakukan, maka disajikan sebuah diagram pada Gambar 3.2 dilanjutkan dengan penjelasan secara deskriptif dari masing-masing fase.



Gambar 3.2 Tahapan Penelitian

#### 3.2.1 Fase 1: Identifikasi

Fase pertama diawali dengan mengidentifikasi fenomena dalam keilmuan rekayasa perangkat lunak sesuai dengan topik yang diangkat. Selanjutnya yaitu tahapan studi literatur untuk mengetahui perkembangan terkini, tanggapan dari para peneliti terhadap fenomena tersebut hingga solusi-solusi yang sudah diterapkan. Harapannya melalui studi literatur dapat memperoleh luaran berupa kesenjangan penelitian dan *state of the art* yang memiliki peran penting untuk memastikan adanya hasil dengan unsur keterbaruan (*novelty*) pada penelitian yang akan dilakukan. Tahapan studi literatur memiliki alur dua arah sehingga apabila

pengetahuan atau wawasan yang diperoleh belum terpenuhi, maka tahapan dapat kembali lagi menuju tahapan identifikasi fenomena untuk mencari fenomena lainnya atau yang lebih umum. Kemudian tahapan yang terakhir yaitu formulasi rumusan permasalahan yang akan diangkat dalam penelitian ini. Pertanyaan atau rumusan masalah penelitian pun disusun pada tahapan terakhir tersebut.

### **3.2.2 Fase 2: Perencanaan**

Rumusan masalah penelitian yang dihasilkan dari fase sebelumnya menjadi landasan utama untuk menyusun hipotesis-hipotesis. Objek penelitian yang diangkat kemudian dijabarkan untuk diketahui masing-masing kasus serta *Unit of Analysis*. Pada tahapan selanjutnya yaitu memetakan darimana sumber data yang akan dianalisis dapat diperoleh, seperti metrik kinerja hingga sumber kode (*source code*) dari prototipe aplikasi. Tahapan terakhir yaitu merancang skema dan alur pengumpulan data, yang dimana penelitian ini menggunakan pendekatan *Goal Question Metric* (GQM). Selain itu juga dirancang skema observasi langsung yang akan dilakukan melalui pengujian. Hal yang perlu dirancang antara lain seperti alur skenario uji dan alat pendukung yang digunakan untuk memperoleh data.

### **3.2.3 Fase 3: Pengumpulan**

Fase ketiga melibatkan sebuah sistem atau prototipe aplikasi yang telah dikembangkan diluar penelitian ini, adapun spesifikasi sistem yang digunakan yaitu *microservices* dengan pola CQRS. Sistem tersebut dikembangkan dengan tiga jenis basis data yang berbeda. Proses pengumpulan data dilakukan terhadap sistem *microservices* tersebut sehingga akan dihasilkan dokumen sumber kode serta metrik evaluasi kinerja yang dilakukan. Data yang dihasilkan selama proses pengujian kemudian dikumpulkan dan didokumentasikan dengan baik sebelum dianalisis lebih lanjut di tahapan berikutnya.

### **3.2.4 Fase 4: Analisis**

Data yang telah diperoleh di fase sebelumnya meliputi sumber kode dan hasil pengujian, kemudian dianalisis menggunakan metode ataupun model matematis yang bersesuaian terhadap pengujian hipotesis yang akan dilakukan. Selain itu juga, data yang diperoleh tentunya diinterpretasikan dalam bentuk grafik seperti histogram ataupun diagram batang. Dalam penelitian ini, analisis menggunakan dua pendekatan yaitu secara inferensial untuk menguji hipotesis dan secara deskriptif

untuk membahas temuan-temuan serta keterkaitannya dengan teori dan perkembangan penelitian yang sudah ada. Terakhir hasil analisis dan temuan tersebut kemudian dirangkum untuk memperoleh kesimpulan serta implikasinya.

### 3.3 Instrumen Penelitian

Penelitian ini condong pada observasi terhadap hasil pengujian yang dilakukan pada sistem. Sebagai upaya untuk mempermudah proses dalam menentukan dan mengumpulkan metrik-metrik yang digunakan selama pengujian, maka diperlukan instrumen pendukung serta alat pendukung. Dibawah ini merupakan penjelasan lengkap dari instrumen dan alat yang digunakan :

#### 3.3.1 Instrumen Pendukung

Untuk menentukan atribut apa saja yang berkaitan erat dengan kinerja sebuah perangkat lunak, maka penelitian ini mengacu pada atribut kualitas perangkat lunak mengenai efisiensi kinerja atau “*performance efficiency*” dari ISO 25010. Aspek yang berkaitan antara lain terdiri dari perilaku waktu, kapasitas dan penggunaan sumber daya (ISO, 2011). Penjelasan serta strategi yang dilakukan untuk memenuhi kriteria setiap aspeknya dijabarkan dalam **Tabel 3.1**.

**Tabel 3.1** Penjelasan Atribut Efisiensi Kinerja dari ISO 25010

No.	Sub-Aspek	Penjelasan	Strategi
1	Perilaku waktu ( <i>time behaviour</i> )	Sejauh mana respon dan waktu pemrosesan dari sistem yang diujikan	Melakukan <i>load testing</i> dengan beberapa skenario dengan perlakuan yang sama
2	Penggunaan sumber daya ( <i>resource utilization</i> )	Sejauh mana penggunaan sumber daya ketika sistem digunakan	Melakukan <i>load testing</i> dengan beberapa skenario dengan perlakuan yang sama

Namun ISO 25010 hanya memberikan gambaran umum mengenai atribut kualitas perangkat lunak, sehingga dalam penelitian ini diperlukan dokumen acuan tambahan lainnya yaitu ISO 25023 yang memberikan penjelasan lebih lengkap

mengenai pedoman evaluasi serta pengukuran dari masing-masing atribut tersebut. Beberapa pengukuran yang digunakan dalam penelitian ini untuk masing-masing aspek pada atribut efisiensi kinerja sesuai dengan pedoman ISO 25023 tercantum pada **Tabel 3.2** (ISO, 2016). Namun tidak seluruh pengukuran diadaptasi, melainkan hanya pengukuran yang sangat direkomendasikan oleh ISO dan memungkinkan diukur melalui alat yang digunakan.

**Tabel 3.2** Metrik Setiap Aspek

Aspek	Kode	Metrik	Satuan
Perilaku waktu ( <i>time behaviour</i> )	M1	Rerata Waktu Respons	Waktu (ms)
	M2	Rerata <i>Throughput</i>	<i>Request</i> /detik
Penggunaan sumber daya ( <i>resource utilization</i> )	M3	Rerata Penggunaan Prosesor	% (0-1)
	M4	Rerata Penggunaan Memori	% (0-1)

- 1) **Rerata Waktu Respons (M1)**, seberapa lama rerata waktu yang diperlukan sistem untuk merespon pengguna atau task. Adapun formulasi yang digunakan sebagai berikut :

$$X = \sum_{i=1 \text{ to } n} (A_i)/n \quad (1)$$

$A_i$  = Waktu yang diperlukan sistem dalam pengukuran ke  $I$

$n$  = jumlah respon yang diukur

- 2) **Rerata *Throughput* (M2)**, seberapa banyak *request* yang dapat diproses oleh sistem dalam satuan waktu (detik). Adapun formulasi yang digunakan sebagai berikut :

$$X = \sum_{i=1 \text{ to } n} (A_i/B_i)/n \quad (2)$$

$A_i$  = Jumlah perintah yang berhasil diproses dalam pengamatan ke  $I$

$B_i$  = Waktu operasi untuk mengeksekusi perintah pengamatan ke  $I$

$n$  = jumlah respon yang diamati

- 3) **Rerata Penggunaan Prosesor (M3)**, seberapa besar penggunaan prosesor digunakan ketika mengeksekusi perintah dalam kurun waktu tertentu. Adapun formulasi yang digunakan sebagai berikut :

$$X = \sum_{i=1 \text{ to } n} (A_i/B_i)/n \quad (3)$$

$A_i$  = Waktu prosesor yang digunakan untuk mengeksekusi perintah dalam pengamatan ke  $I$

$B_i$  = Waktu operasi untuk mengeksekusi perintah pengamatan ke  $I$

$n$  = jumlah respon yang diamati

- 4) **Rerata Penggunaan Memori (M4)**, seberapa besar penggunaan memori digunakan ketika mengeksekusi perintah dalam kurun waktu tertentu. Adapun formulasi yang digunakan sebagai berikut :

$$X = \sum_{i=1 \text{ to } n} (A_i/B_i)/n \quad (4)$$

$A_i$  = Ukuran memori yang digunakan untuk mengeksekusi perintah dalam pengamatan ke  $I$

$B_i$  = Ukuran memori untuk mengeksekusi perintah pengamatan ke  $I$

$n$  = jumlah respon yang diamati

### 3.3.2 Alat Pendukung

Alat yang digunakan untuk membantu proses *load testing* dan pengumpulan hasil pengujian dalam penelitian ini adalah JMeter, alat ini dikembangkan oleh Apache secara sumber terbuka (*open-source*). Penggunaan alat ini telah banyak digunakan terutama pada ranah akademisi, hal ini ditunjukkan dengan berbagai penelitian sebelumnya yang telah menggunakan alat ini dalam menguji atau mengevaluasi kinerja dari perangkat lunak (Fansha dkk., 2021; Langsari dkk., 2017; Sutino dkk., 2018). Alat ini memiliki fitur yang melimpah dan mendukung banyak *library* eksternal sehingga sejalan dengan berbagai skenario pengujian dan metrik yang diperlukan dalam penelitian ini.

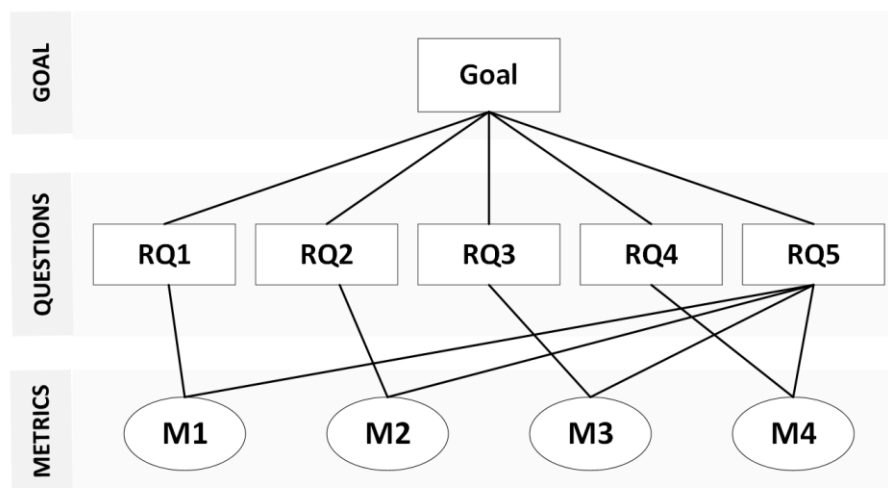
### 3.4 Goal Question Metric (GQM)

Sesuai dengan metrik yang diperoleh dari instrumen penelitian (**Subbab 3.3.1**), maka dapat diturunkan dan dipetakan antara tujuan, rumusan permasalahan dan metrik yang berkaitan melalui pendekatan Goal Question Metric (GQM). Pendekatan ini memiliki keuntungan yaitu dapat meminimalisir adanya data yang dikoleksi tidak relevan sesuai tujuan dan permasalahan (Runeson dkk., 2012). GQM yang dirancang divisualisasikan dalam bentuk **Tabel 3.3** berikut. Sedangkan visualisasi dalam bentuk diagram dapat dilihat pada **Gambar 3.3**.

**Tabel 3.3** Goal Question Metric

<b>Goal</b>	<b>Question</b>	<b>Metric</b>
Menganalisis dan membandingkan kinerja sistem ( <i>microservices</i> dengan CQRS)	Apakah terdapat perbedaan yang signifikan dari masing-masing jenis basis data terhadap waktu respons sistem ? ( <b>RQ1</b> )	Rerata Waktu Respons ( <b>M1</b> )
meliputi waktu respon, <i>throughput</i> , penggunaan prosesor dan memori pada setiap jenis basis data	Apakah terdapat perbedaan yang signifikan dari masing-masing jenis basis data terhadap <i>throughput</i> sistem ? ( <b>RQ2</b> )	Rerata <i>Throughput</i> ( <b>M2</b> )
	Apakah terdapat perbedaan yang signifikan dari masing-masing jenis basis data terhadap penggunaan prosesor sistem ? ( <b>RQ3</b> )	Rerata Penggunaan Prozessor ( <b>M3</b> )
	Apakah terdapat perbedaan yang signifikan dari masing-masing jenis basis data terhadap penggunaan memori sistem ? ( <b>RQ4</b> )	Rerata Penggunaan Memori ( <b>M4</b> )

Goal	Question	Metric
	Bagaimanakah keunggulan dari masing-masing kelompok sistem pada seluruh aspek yang diuji ? <b>(RQ5)</b>	Rerata Waktu Respons <b>(M1)</b>
		Rerata <i>Throughput</i> <b>(M2)</b>
		Rerata Penggunaan Prosesor <b>(M3)</b>
		Rerata Penggunaan Memori <b>(M4)</b>



Gambar 3.3 Goal Question Metric

### 3.5 Hipotesis Penelitian

Terdapat lima rumusan masalah penelitian yang diajukan, namun hanya ada empat rumusan yang memiliki hipotesis yaitu **RQ1** sampai **RQ4**. Mengingat **RQ5** merupakan pertanyaan yang tidak memerlukan adanya uji hipotesis. Adapun masing-masing rumusan masalah terdapat hipotesis nol ( $H_0$ ) dan hipotesis sementara ( $H_1$ ) yang diusulkan.

- 1) Hipotesis terkait **rumusan masalah pertama (RQ1)** “Apakah terdapat perbedaan yang signifikan dari masing-masing jenis basis data terhadap waktu respons sistem ?” diusulkan sebagai berikut :

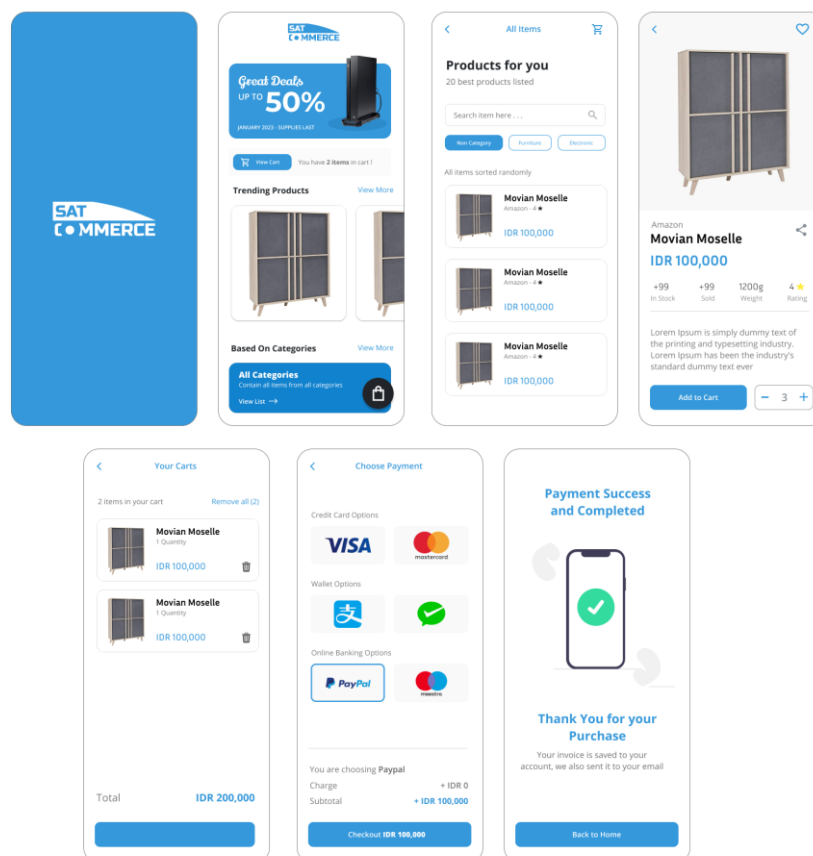


- $H_0$  = Tidak ada perbedaan waktu respon sistem yang signifikan terhadap masing-masing jenis basis data
  - $H_1$  = Terdapat perbedaan waktu respon sistem yang signifikan terhadap masing-masing jenis basis data
- 2) Hipotesis terkait **rumusan masalah kedua (RQ2)** “Apakah terdapat perbedaan yang signifikan dari masing-masing jenis basis data terhadap *throughput* sistem ?” diusulkan sebagai berikut :
- $H_0$  = Tidak ada perbedaan *throughput* sistem yang signifikan terhadap masing-masing jenis basis data
  - $H_1$  = Terdapat perbedaan *throughput* sistem yang signifikan terhadap masing-masing jenis basis data
- 3) Hipotesis terkait **rumusan masalah ketiga (RQ3)** “Apakah terdapat perbedaan yang signifikan dari masing-masing jenis basis data terhadap penggunaan prosesor sistem ?” diusulkan sebagai berikut :
- $H_0$  = Tidak ada perbedaan penggunaan prosesor sistem yang signifikan terhadap masing-masing jenis basis data
  - $H_1$  = Terdapat perbedaan penggunaan prosesor sistem yang signifikan terhadap masing-masing jenis basis data
- 4) Hipotesis terkait **rumusan masalah keempat (RQ4)** “Apakah terdapat perbedaan yang signifikan dari masing-masing jenis basis data terhadap penggunaan memori sistem ?” diusulkan sebagai berikut :
- $H_0$  = Tidak ada perbedaan penggunaan memori sistem yang signifikan terhadap masing-masing jenis basis data
  - $H_1$  = Terdapat perbedaan penggunaan memori sistem yang signifikan terhadap masing-masing jenis basis data

### 3.6 Gambaran Prototipe Sistem

Prototipe yang dievaluasi pada penelitian ini yaitu sebuah *microservices* dalam bidang *e-commerce* bernama SatCommerce (Pamungkas, 2023). Diharapkan melalui prototipe tersebut dapat memberikan gambaran serta mensimulasikan kasus nyata yang terjadi pada umumnya. Sebagai catatan bahwa penelitian ini tidak membahas secara menyeluruh proses pengembangan prototipe. Selain itu juga,

prototipe sistem terintegrasi dengan aplikasi android yang berperan sebagai sisi *front-end*. Meskipun dalam pengujian ini tidak melibatkan sisi *front-end* tetapi sebagai gambaran dari tampilan antarmuka aplikasi dapat dilihat pada **Gambar 3.4**. Masing-masing bagian subbab ini terdapat gambaran serta penjelasan untuk memastikan prototipe telah sesuai dengan konteks dan tujuan penelitian.



**Gambar 3.4** Antarmuka Aplikasi SatCommerce

### 3.6.1 Validasi Prototipe Sistem

Prototipe yang dikembangkan diupayakan dengan maksimal untuk mengacu pada penelitian-penelitian sebelumnya. Melihat kasus penggunaan *microservices* yang secara dominan terdapat pada bidang *e-commerce* (Li dkk., 2022), hal ini telah bersesuaian dengan prototipe sistem pada bidang *e-commerce*. Selain itu teknologi yang digunakan pada prototipe pun telah banyak digunakan pada penelitian sejenis *microservices* lainnya seperti penggunaan bahasa pemrograman C# (Malić, 2019; Rados & Hajnic, 2022). Dekomposisi fitur atau domain pada prototipe yang terdiri

Muhammad Raihan Satrio Putra Pamungkas, 2023

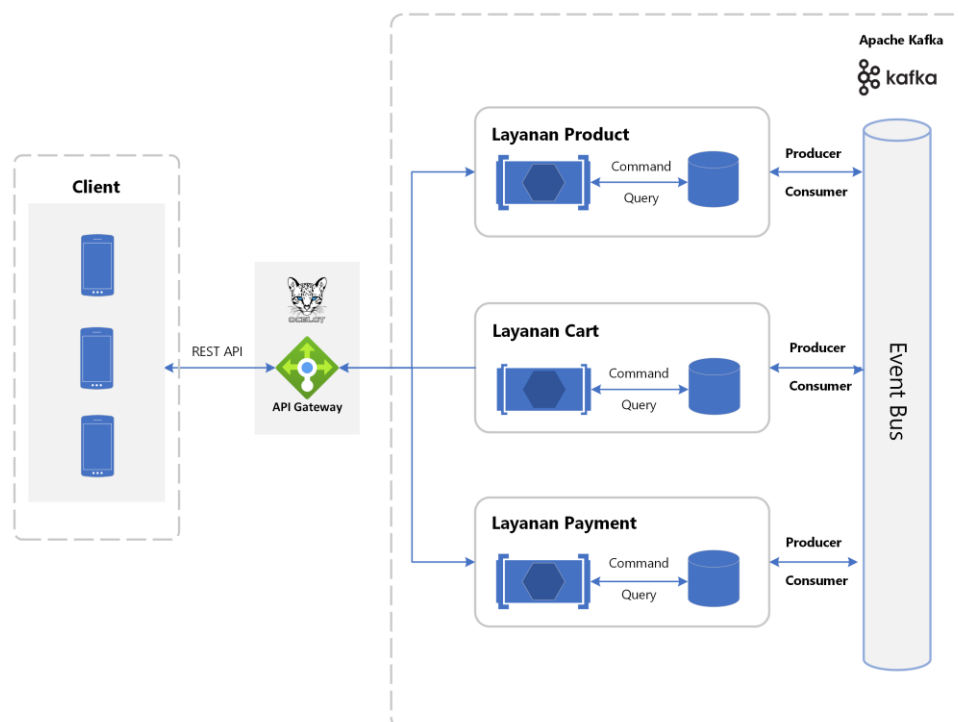
**ANALISIS KINERJA MICROSERVICES DENGAN POLA COMMAND QUERY RESPONSIBILITY SEGREGATION PADA BERAGAM JENIS BASIS DATA**

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

atas *product*, *cart* dan *payment* juga telah mengacu pada *Software Product Line* pada bidang *e-commerce* (Rodas-Silva dkk., 2019). Lebih lanjut, *Software Product Line* atau SPL adalah seperangkat sistem intensif perangkat lunak yang berbagi rangkaian fitur umum secara bersama untuk kebutuhan segmen pasar ataupun stakeholder (Babar dkk., 2010).

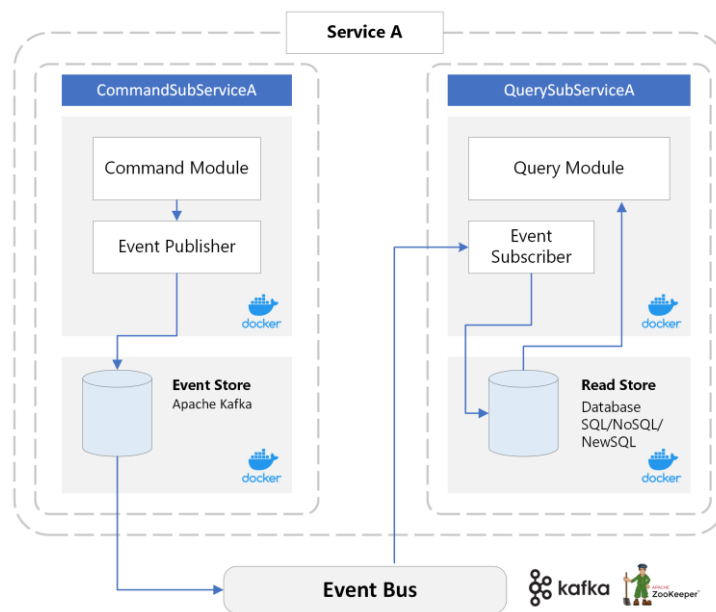
### 3.6.2 Rancangan Sistem

Secara umum prototipe yang digunakan pada sisi *back-end* memiliki arsitektur *microservices* dengan pola CQRS dan beberapa tambahan pola pendukung seperti *event sourcing*, *database per services* dan *API Gateway*. Terdapat tiga prototipe sistem yang mengimplementasikan basis data relasional menggunakan PostgreSQL, basis data NoSQL menggunakan MongoDB dan basis data NewSQL menggunakan CockroachDB. Meskipun begitu setiap prototipe memiliki arsitektur dan konteks (*e-commerce*) yang serupa. Visualisasi arsitektur secara garis besar dari prototipe sistem ini dapat ditinjau pada **Gambar 3.5** berikut.



**Gambar 3.5** Gambaran Umum Arsitektur Sistem

Jika ditinjau, terdapat empat layanan yang berperan sebagai *microservices* meliputi layanan *product*, *cart* dan *payment*. Masing-masing layanan memiliki fungsionalitas yang independen seperti layanan *product* untuk memproses produk-produk *e-commerce*, layanan *cart* untuk memproses keranjang pengguna, dan layanan *payment* untuk memproses layanan pembayaran. Teknologi dan alat yang digunakan dalam prototipe sistem dibahas pada **Subbab 3.6.2**. Prototipe sistem yang dikembangkan mengimplementasikan pola CQRS, hal ini dapat ditinjau dengan menelaah arsitektur dari tingkatan setiap layanan yang tercantum pada **Gambar 3.6**. Dapat dilihat terdapat modul dan penyimpanan yang terpisahkan sehingga sesuai dengan prinsip CQRS.



**Gambar 3.6** Arsitektur Pada Tingkatan Setiap Layanan

### 3.6.3 Spesifikasi Sistem

Pada bagian ini dijelaskan sistem dari level fungsionalitas, teknologi perangkat lunak serta level perangkat keras yang digunakan dalam proses pengujian. Masing-masing dicantumkan dalam tiga tabel yang terpisah. Meskipun tidak seluruh *endpoint* berikut digunakan (lihat **Subbab 3.7**)

**Tabel 3.4** Spesifikasi API dari Prototipe Sistem

Layanan	Endpoint	Tujuan atau Fungsionalitas
<i>Product</i>	GET /product	Memperoleh seluruh produk yang ada
	GET /product/:ID	Memperoleh produk yang spesifik
	POST /product	Menyimpan produk pada basis data
	DELETE /products/:ID	Menghapus produk yang spesifik
<i>Cart</i>	POST /cart	Menyimpan data produk pada keranjang
	GET /cart	Memperoleh seluruh keranjang
	DELETE /cart/:ID	Menghapus data keranjang yang spesifik
<i>Payment</i>	POST /payment	Membuat metode pembayaran
	POST /payment/cart	Memproses pembayaran keranjang
	GET /payment	Memperoleh seluruh metode pembayaran
	DELETE /payment	Menghapus seluruh metode pembayaran

Selanjutnya pada **Tabel 3.5** yaitu mengenai daftar teknologi yang digunakan dalam prototipe sistem, setiap layanan memiliki teknologi yang sama apabila ditinjau dalam satu prototipe sistem.

**Tabel 3.5** Spesifikasi Teknologi dari Prototipe Sistem

Jenis	Keterangan	Versi
Bahasa Pemrograman	C#	7.0
<i>Framework</i> API	.NET Core 6	6.0.12
Kontainer	Docker	~
<i>Event Bus</i>	Apache Kafka	~
<i>API Gateway</i>	Ocelot	~

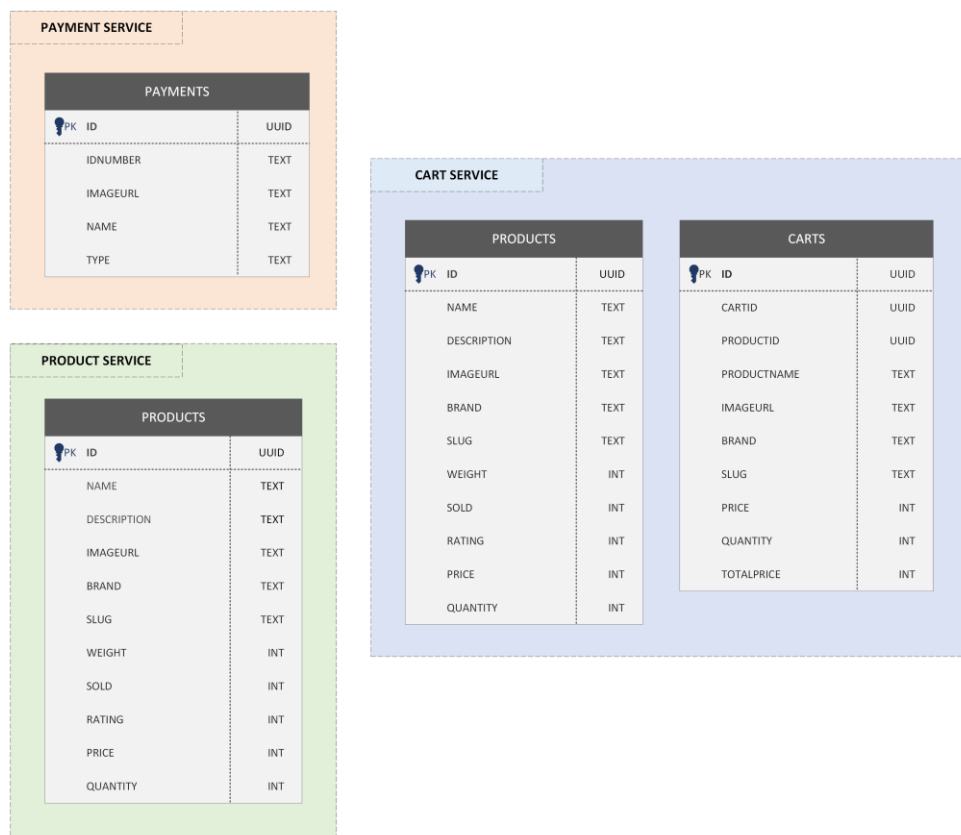
Terakhir yaitu **Tabel 3.6** mengenai spesifikasi dari perangkat pengujian yang digunakan untuk proses *host* dari prototipe dan lingkungan pengujian, sehingga keduanya berada pada sebuah perangkat yang sama.

**Tabel 3.6** Spesifikasi Sistem di Perangkat Pengujian

Jenis	Spesifikasi
Sistem Operasi	Ubuntu 22.04.1 LTS
Prosesor	Intel Core i3-6006U @ 2.00 GHz – 2 Core
RAM	12 GB DDR4 2133 MHz (8 + 4 Dual Channel)
Disk	500GB HDD

### 3.6.4 Basis Data

Dalam prototipe ini terdapat tiga jenis basis data yang diujicobakan yaitu relasional, NoSQL dan NewSQL. Untuk mendukung penyimpanan data yang sesuai dengan konteks fungsionalitas dari masing-masing layanan yang ada, maka dirancang sebuah *Entity Relationship Diagram (ERD)*. **Gambar 3.7** ditujukan untuk masing-masing basis data dalam setiap layanannya. Namun mengingat bentuk basis data dari jenis NoSQL adalah dokumen, maka terdapat penyesuaian lebih lanjut tanpa menghilangkan atribut dan kolom.

**Gambar 3.7** Entity Relationship Diagram Sistem

Muhammad Raihan Satrio Putra Pamungkas, 2023

ANALISIS KINERJA MICROSERVICES DENGAN POLA COMMAND QUERY RESPONSIBILITY SEGREGATION PADA BERAGAM JENIS BASIS DATA

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

### 3.6.5 Data Seeder

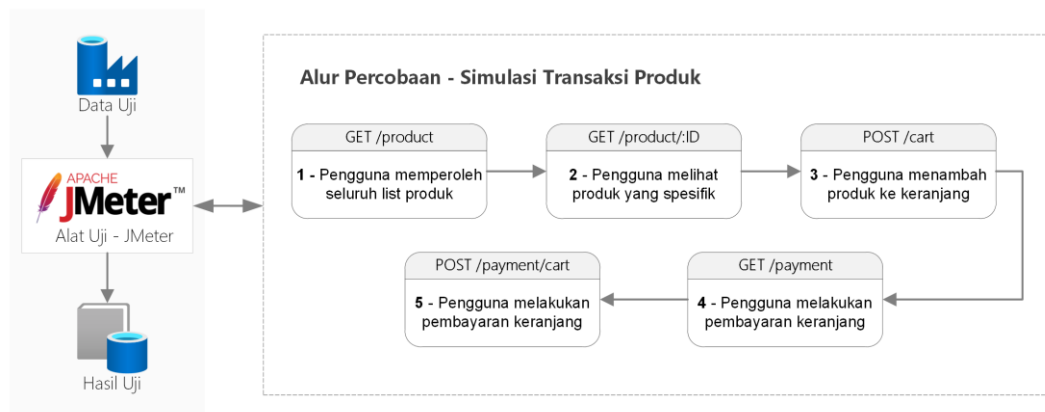
Untuk memenuhi skenario dan kriteria pengujian, maka diperlukan terlebih dahulu data yang telah terhimpun pada basis data atau yang biasa dikenal dengan istilah data *seeder*. Terdapat dua tabel yang telah terisi yaitu tabel Products dengan jumlah data sebanyak 20 dan tabel Payments sebanyak 6 data. Dataset yang digunakan pada tabel Products diadaptasi dari penelitian Amazon Berkeley Object (ABO) yaitu sebuah dataset besar mengenai katalog produk (Collins dkk., 2022). Sedangkan dataset tabel Payments diadaptasi dari sebuah repositori sumber terbuka. Selengkapnya dataset hasil adaptasi secara keseluruhan dapat dilihat pada **Lampiran 1** dan **Lampiran 2**.

## 3.7 Pengumpulan Data Melalui Pengujian

Sumber data dihimpun dari hasil pengujian terhadap sistem yang dilakukan, sehingga terdapat lima sumber data sesuai dengan jumlah metrik atau pengukuran yang dirancang. Penggunaan berbagai sumber telah sesuai dengan definisi atau ciri khas dari desain penelitian *case study* yang memerlukan banyaknya sumber data (Yin, 2003), pendekatan ini dikenal dengan istilah *triangulation* sebagai upaya untuk meningkatkan validitas penelitian (Runeson dkk., 2012). Seluruh hasil pengukuran dapat diperoleh melalui satu skenario pengujian yang kemudian dilanjutkan pada proses transformasi untuk penyesuaian satuan setiap metrik.

### 3.7.1 Alur Skenario Pengujian

Skenario pengujian mencoba untuk mensimulasikan penjelajahan pengguna (*user journey*) ketika berbelanja di situs *e-commerce*. Berdasarkan literatur yang ditemukan (Tupikovskaja-Omovie, 2022), penelitian ini mengadaptasi dan menyederhanakan *user journey* ketika berbelanja di situs *e-commerce* untuk memperoleh tahapan skenario pengujian. Selain itu juga metode utama dalam pemrosesan data yaitu *create*, *read*, *update* dan *delete* juga diimplementasikan dalam skenario pengujian ini. Setiap tahapan bersifat prosedural dan telah diatur melalui alat JMeter untuk mengeksekusi tahapan secara berurutan. Alur skenario divisualisasikan pada **Gambar 3.8**, sedangkan penjelasan masing-masing *request* atau sub skenario menyertai dibawah ini.



**Gambar 3.8** Alur Skenario Pengujian

1) ***Gdttl Products (R1)***

Dengan mengakses `GET /product`, yaitu untuk memperoleh daftar seluruh produk yang tersedia pada basis data, jika diasumsikan tahapan ini merupakan tahapan awal ketika pengguna mengunjungi suatu situs *e-commerce* dan memperoleh seluruh produk pada etalase digital

2) ***Get Specific Product (R2)***

Dengan mengakses `GET /product/:ID` (berserta parameter ID produk), yaitu untuk memperoleh identitas produk dengan ID tersebut secara mendetail . Jika diasumsikan tahapan ini merupakan tahapan ketika pengguna mengklik produk untuk mencari tahu secara lebih jelas mengenai deskripsi, penilaian serta identitas produk lainnya.

3) ***Create Cart (R3)***

Dengan mengakses `POST /cart`, yaitu untuk menambah satu produk dari daftar seluruh produk yang telah dipilih kedalam keranjang. Jika diasumsikan tahapan ini serupa ketika pengguna memilih produk kemudian sistem secara otomatis memasukkannya ke dalam keranjang pengguna yang bersesuaian.

4) ***Gdttl Payment (R4)***

Dengan mengakses `GET /payment`, yaitu untuk memperoleh metode pembayaran yang tersedia untuk digunakan dalam pembayaran. Jika diasumsikan tahapan ini ketika pengguna ingin memilih metode pembayaran.



### 5) *Make a Payment for Specific Cart (R5)*

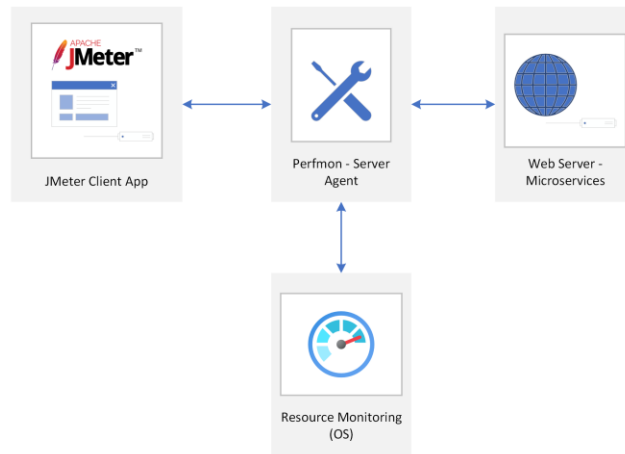
Dengan mengakses POST /payment/cart, yaitu untuk memproses transaksi dari pesanan yang telah dibuat. Dalam kasus nyata, proses transaksi terkadang membutuhkan waktu yang relatif lama akibat faktor eksternal seperti proses validasi metode pembayaran ataupun faktor lainnya. Namun dalam pengujian ini faktor-faktor tersebut tidak diikutsertakan, sehingga proses berlangsung secara langsung.

### 3.7.2 Kriteria Pengujian

Sebelum pengujian dilakukan, terlebih dahulu data telah dihimpun dan tersedia pada basis data. Dataset yang digunakan merupakan data *dummy* yang di dengan lokalisasi Bahasa Indonesia. Untuk meninjau atribut dan bagaimana penggalan dataset, maka dapat dilihat pada **Lampiran 1** dan **Lampiran 2**. Selain itu juga setiap tahapan memerlukan *payload* berupa *request body* ataupun *identifier* pada *endpoint*. Supaya memperoleh nilai yang lebih presisi, maka dilakukanlah proses replikasi (Montgomery, 2013). Adapun jumlah replikasi yang dilakukan untuk pengujian skenario ini yaitu sebanyak 200 kali yang berperan juga sebagai jumlah *request*. Dimana jumlah *request* ini didasari dari penelitian sejenis yang telah dilakukan sebelumnya (Fansha dkk., 2021).

### 3.7.3 Pengaturan Pengujian

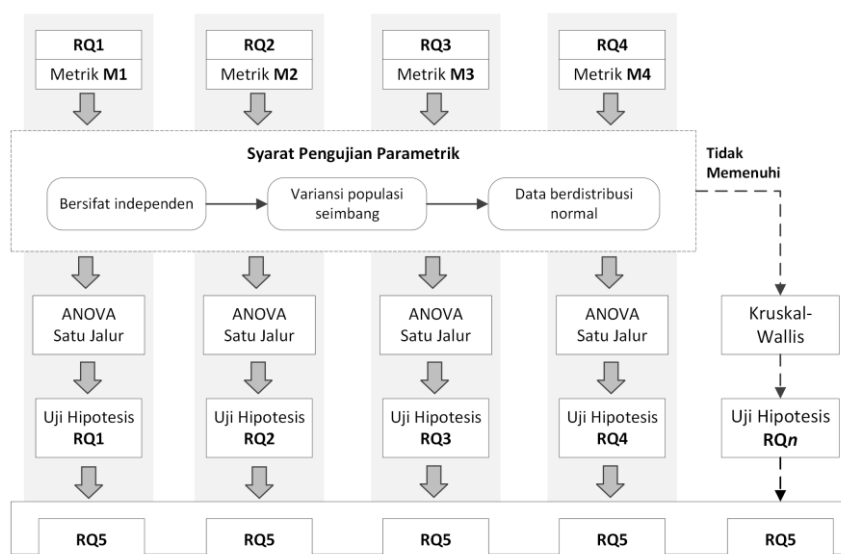
Secara *default* alat Jmeter dapat mengukur waktu respon dan *throughput*, namun untuk mengukur penggunaan memori dan prosesor diperlukan *plugin* tambahan bernama “*perfmon*” dengan *script* tambahan yang berperan sebagai agen. Peran dari agen disini untuk mengetahui *port* alamat yang dituju, setelah *port* alamat diperoleh maka sistem operasi memberikan keluwesan untuk mengetahui *process id* (PID) yang memantau secara *realtime* penggunaan memori dan prosesor, seperti alat bernama “*htop*” pada sistem operasi Linux.



**Gambar 3.9** Pengaturan Pengukuran

### 3.8 Teknik Analisis Data

Hasil pengukuran dan pengujian yang telah dilakukan sesuai desain dan skenario diatas, kemudian dianalisis lebih lanjut untuk menguji hipotesis yang bersesuaian. Data yang dihimpun terlebih dahulu dilakukan pengecekan prasyarat untuk uji secara parametrik. Jika seluruh syarat memenuhi maka pengujian dapat dilakukan menggunakan pendekatan secara parametrik. Namun, jika syarat tidak terpenuhi maka pendekatan dilakukan secara non-parametrik. Alur analisis divisualisasikan pada **Gambar 3.10**. Dapat ditinjau bahwa untuk memperoleh **RQ5** diperlukan pengujian hipotesis pada **M1** hingga **M4**. Penjelasan mengenai prasyarat serta teknik pengujian yang dilakukan pada masing-masing Subbab.



**Gambar 3.10** Skema Analisis Data

### 3.8.1 Tanggapan terhadap *Outliers*

Apabila hasil pengujian terdapat data yang terindikasi sebagai pencilan (*outliers*), maka *outliers* tersebut dihapus sebelum dilanjutkan kepada tahapan prasyarat dan pengujian hipotesis. Hal ini direkomendasikan jika memang adanya kesalahan pengukuran ataupun anomali yang terjadi (Osborne & Overbay, 2004). Sehingga dataset yang dianalisis dan diuji merupakan dataset yang telah bersih dari adanya *outliers*. Selain itu juga untuk memvisualisasikan *outliers* dalam bentuk diagram BoxPlot maka digunakan metode Tukey dengan formulasi *Inter Quartilerange* (IQR) sebagai berikut (Kannan dkk., 2015) :

$$IQR = Q3 - Q1 \quad (5)$$

$$outliers_{low} = Q1 - (1.5 \times IQR) \quad (6)$$

$$outliers_{high} = Q1 + (1.5 \times IQR) \quad (7)$$

(Dimana nilai Q3 adalah Kuartil atas dan Q1 adalah Kuartil bawah)

### 3.8.2 Prasyarat Pengujian Parametrik

Untuk mengetahui bahwa koleksi data yang telah diperoleh dapat dianalisis menggunakan pendekatan parametrik, maka terdapat beberapa syarat yang perlu dipenuhi oleh koleksi data tersebut (Bluman, 2012) :

- 1) Variansi dari populasi harus memiliki jumlah yang seimbang
- 2) Populasi dari data yang dihimpun harus memiliki sifat yang berdistribusi normal. Sehingga untuk mengecek hal tersebut diperlukan uji normalitas yang dilakukan.
- 3) Setiap sampel data harus bersifat independen satu sama lainnya

### 3.8.3 Teknik Analisis dan Pengujian

Jika hasil pengujian syarat parametrik terpenuhi maka data dapat dianalisis menggunakan teknik *one-way ANOVA* untuk metrik tunggal. Namun apabila pengujian syarat parametrik tidak terpenuhi maka perlu dianalisis menggunakan teknik *Kruskal-Wallis*. Berikut ini penjelasan dari masing-masing teknik yang berpotensi digunakan :

- 1) **One Way ANOVA** (*Analysis of Variance*), teknik ini dapat menguji pengaruh variabel bebas atau faktor terhadap satu variabel terikat (Bluman, 2012). Sesuai dengan desain pengujian, terdapat satu faktor atau tingkatan mengenai jumlah *request* yang diberikan (Lihat **Tabel 3.8**) maka teknik dengan *one-way* digunakan untuk menguji hipotesis. Formulasi dan prosedur ANOVA dapat dilihat pada **Lampiran 9**.
- 2) **Kruskal-Wallis**, teknik ini menjadi alternatif ketika data yang diperoleh tidak memenuhi syarat parametrik (seperti tidak berdistribusi normal). Sehingga teknik ini memiliki keserupaan yaitu melakukan Uji-F untuk kasus non parametrik (Bluman, 2012). Adapun formulasi dari Kruskal-Wallis dapat ditinjau pada **Lampiran 10**.