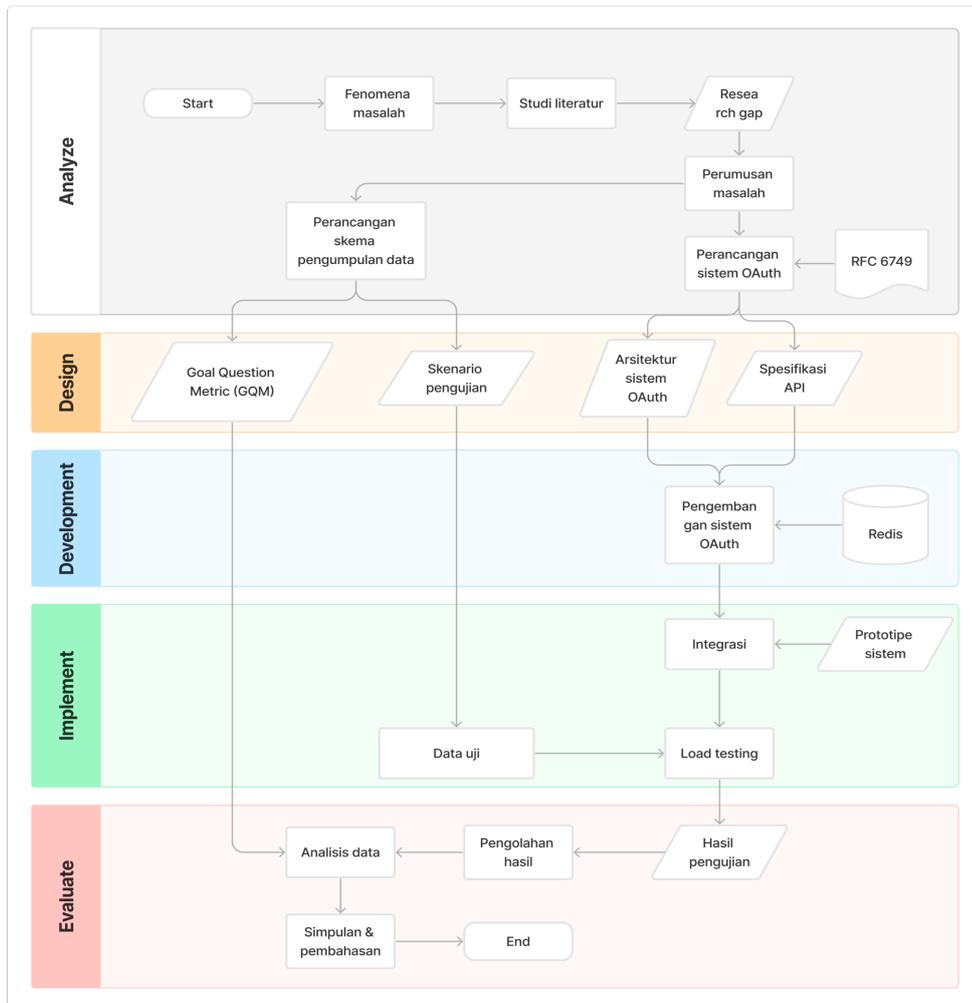


# BAB III

## METODE PENELITIAN

### 3.1 Desain Penelitian

Desain penelitian yang digunakan adalah *Research and Development (R&D)*. Tujuan dari desain R&D adalah untuk menciptakan produk baru atau menyempurnakan produk yang telah ada agar dapat dinilai keefektifannya dan dapat dipertanggungjawabkan (Sugiyono, 2020). Berbagai model penelitian yang dapat digunakan dalam R&D. Model yang digunakan pada penelitian ini adalah ADDIE yang terdiri dari lima tahap, meliputi analisis (*Analyze*), desain (*Design*), pengembangan (*Development*), implementasi (*Implement*), dan evaluasi (*Evaluate*).



Gambar 3. 1 Desain Penelitian R&D Model ADDIE

### 3.1.1 *Analyze*

Tahap pertama adalah mempelajari fenomena keilmuan Rekayasa Perangkat Lunak yang berkaitan dengan dampak penggunaan IMDB Redis sebagai basis data manajemen token OAuth 2.0. Setelah itu, studi literatur dilakukan untuk mengidentifikasi teori-teori yang relevan, perkembangan, tanggapan para peneliti, dan solusi yang sudah diterapkan. Sumber-sumber untuk studi literatur ini termasuk buku, jurnal, artikel ilmiah, dan dokumen elektronik. Studi literatur dapat mengeksplorasi kondisi terkini dan kesenjangan penelitian yang sangat penting untuk memastikan bahwa penelitian tetap relevan. Berdasarkan temuan dari analisis fenomena dan studi literatur, tahap terakhir dilakukan perumusan masalah.

### 3.1.2 *Design*

Tahap desain adalah tahap di mana produk perangkat lunak dibuat berdasarkan perumusan masalah. Langkah pertama melibatkan perancangan sistem OAuth 2.0, yang meliputi:

- Desain arsitektur, menggambarkan teknologi yang diperlukan untuk membangun sistem OAuth 2.0. Hal ini memungkinkan aplikasi klien untuk terintegrasi dengan penyedia sumber daya (*Resource Server*). REST API digunakan sebagai desain arsitektur.
- Desain spesifikasi API, menjelaskan standar penggunaan API termasuk standar untuk URL *endpoint*, metode HTTP, dan format data untuk permintaan dan respons.

Dokumen RFC 6749 digunakan sebagai pedoman perancangan sistem OAuth 2.0. Setelah itu, proses desain pengumpulan data menggunakan pendekatan *Goal Question Metric* (GQM) untuk menentukan skema dan alur pengumpulan data. Terakhir, skema observasi dirancang untuk mendapatkan data melalui alur pengujian dan alat pendukung.

### 3.1.3 *Develop*

Pada tahap ini, sistem dikembangkan sesuai dengan desain yang dihasilkan pada tahap sebelumnya, yaitu sistem OAuth 2.0. Proses pengembangan sistem melibatkan penggunaan dua buah basis data NoSQL. MongoDB berfungsi sebagai

basis data utama, sedangkan Redis berfungsi sebagai basis data pendukung untuk manajemen token. Selain itu, komponen Auth dikembangkan di sisi klien untuk memfasilitasi otorisasi pengguna terhadap aplikasi pihak ketiga.

### 3.1.4 Implement

Sistem OAuth 2.0 yang telah dibuat sebelumnya akan diintegrasikan dengan prototipe sistem yang dikembangkan di luar penelitian ini. Prototipe sistem menggunakan arsitektur *microservices* di sisi *server*, sedangkan aplikasi web di sisi klien berfungsi sebagai aplikasi pihak ketiga yang berperan sebagai aplikasi klien dalam OAuth. Selain itu, pengujian beban dilakukan untuk mengumpulkan data terhadap sistem OAuth 2.0 sehingga menghasilkan metrik evaluasi kinerja sistem.

### 3.1.5 Evaluate

Tahap selanjutnya adalah evaluasi menggunakan model matematika untuk menganalisis data hasil pengujian yang telah diperoleh pada tahap sebelumnya. Model tersebut disesuaikan dengan uji hipotesis yang akan dilakukan. Pada penelitian ini, analisis inferensial dilakukan untuk menguji hipotesis, sementara analisis deskriptif digunakan untuk membahas temuan. Hasil analisis data disajikan dan diinterpretasikan dalam bentuk grafik. Terakhir, temuan dan hasil dirangkum untuk menarik kesimpulan dan implikasi.

## 3.2 Alat dan Bahan Penelitian

### 3.2.1 Alat Penelitian

Alat penelitian meliputi perangkat keras (*hardware*) & perangkat lunak (*software*) dengan spesifikasi sebagai berikut.

Tabel 3. 1

Spesifikasi perangkat keras

Perangkat Keras	Spesifikasi
CPU	Intel Core i5-4300U
RAM	16 GB
<i>Storage</i>	SSD 250 GB

Tabel 3. 2

## Spesifikasi perangkat lunak

<b>Perangkat Lunak</b>	<b>Keterangan</b>
MacOS	Sistem Operasi
Visual Studio Code	<i>Code Editor</i>
NodeJS	Bahasa Pemrograman
MongoDB	NoSQL
Redis	IMDB
JMeter	Alat Uji Kinerja

### 3.2.2 Bahan Penelitian

Pemanfaatan materi penelitian, termasuk jurnal, *e-book*, dokumen internet, dan artikel merupakan aspek mendasar dari proses penelitian. Materi-materi ini berfungsi sebagai sumber informasi utama bagi peneliti untuk memperoleh pemahaman yang lebih dalam tentang subjek penelitian, pengembangan aplikasi yang inovatif, dan melakukan eksperimen.

### 3.3 Instrumen Penelitian

Penelitian berfokus pada hasil pengujian yang dilakukan pada sistem OAuth 2.0. Instrumen dan alat pendukung diperlukan untuk memudahkan proses penentuan dan pengumpulan metrik yang digunakan selama pengujian. Penjelasan mengenai alat dan instrumen yang digunakan dapat dilihat di bawah.

#### 3.3.1 Instrumen Pendukung

Kualitas perangkat lunak berdasarkan efisiensi kinerja dirujuk oleh peneliti sesuai dengan ISO 25010. Hal ini mencakup aspek-aspek seperti perilaku waktu, pemanfaatan sumber daya, dan kapasitas (ISO, 2011). Penjelasan lebih lanjut tentang atribut kualitas perangkat lunak ini dapat ditemukan dalam pedoman ISO 25023 (lihat tabel 3.3).

Tabel 3. 3  
Metrik setiap aspek

Aspek	Kode	Metrik	Satuan
Perilaku waktu	M1	Rata-rata waktu respons	<i>Milisecond</i> (ms)
	M2	Rata-rata <i>throughput</i>	<i>Request/s</i>
Penggunaan sumber daya	M3	Rata-rata penggunaan CPU	% (0-1)
	M4	Rata-rata penggunaan memori	% (0-1)

Tabel di atas menyajikan metrik yang digunakan untuk mengevaluasi kinerja sistem berdasarkan dua aspek utama, yaitu perilaku waktu dan penggunaan sumber daya. Adapun uraian dalam setiap metrik yang digunakan dalam penelitian ini adalah sebagai berikut:

- 1) Rata-rata waktu respons (M1) yang dibutuhkan sistem OAuth 2.0 untuk merespons pengguna atau tugas.

$$X = \sum_{i=1 \text{ ke } n} \frac{A_i}{n} \quad (1)$$

$A_i$  = waktu yang dibutuhkan sistem OAuth 2.0

$n$  = jumlah respons

- 2) Rata-rata *throughput* (M2) yang dapat diproses sistem OAuth 2.0 dalam satuan waktu (detik).

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

$A_i$  = jumlah perintah berhasil diproses

$B_i$  = waktu untuk mengeksekusi perintah

$n$  = jumlah respons

- 3) Rata-rata penggunaan CPU (M3) ketika mengeksekusi perintah dalam kurun waktu tertentu.

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

$A_i$  = ukuran penggunaan CPU untuk mengeksekusi perintah

$B_i$  = waktu operasi untuk mengeksekusi perintah

$n$  = jumlah respons

- 4) Rerata penggunaan memori (M4) ketika mengeksekusi perintah dalam kurun waktu tertentu.

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

$A_i$  = ukuran penggunaan memori untuk mengeksekusi perintah

$B_i$  = waktu penggunaan memori untuk mengeksekusi perintah

$n$  = jumlah respons

### 3.3.2 Alat Pendukung

Proses pengujian pada penelitian ini menggunakan uji beban atau *load testing*. JMeter adalah alat sumber terbuka yang dikembangkan oleh *Apache Software Foundation* untuk melakukan pengujian beban. Penelitian terdahulu menggunakan alat ini untuk menguji dan mengevaluasi kinerja perangkat lunak (Gajewski & Zabierowski, 2019; S. Li dkk., 2017; Saransig & Tapia, 2019).

### 3.4 Goal Question Metric (GQM)

Pendekatan GQM digunakan untuk memetakan instrumen penelitian ke dalam rumusan masalah, tujuan penelitian, dan metrik penelitian. Pendekatan ini bertujuan untuk menghindari pengukuran aspek-aspek yang tidak relevan. Tabel 3.4 menunjukkan bentuk visualisasi tabel GQM.

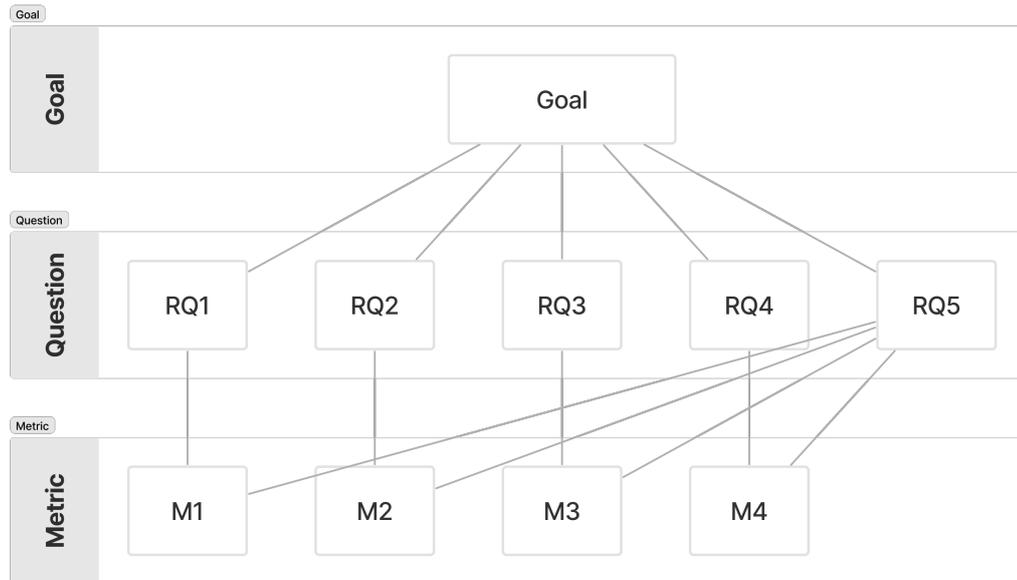
Tabel 3.4

*Goal Question Metric*

<i>Goal</i>	<i>Question</i>	<i>Metric</i>
Menganalisis dampak Redis manajemen terhadap kinerja sistem otentikasi dan otorisasi	Apakah Redis token waktu OAuth 2.0? (RQ1)	Rata-rata waktu respon (M1)

<i>Goal</i>	<i>Question</i>	<i>Metric</i>
berbasis OAuth 2.0 meliputi waktu respons, <i>throughput</i> dan <i>utilization</i>	Apakah implementasi Redis dalam manajemen token mempengaruhi <i>throughput</i> sistem OAuth 2.0? (RQ2)	Rata-rata <i>throughput</i> (M2)
	Apakah implementasi Redis dalam manajemen token mempengaruhi <i>utilization</i> CPU sistem OAuth 2.0? (RQ3)	Rata-rata penggunaan CPU (M3)
	Apakah implementasi Redis dalam manajemen token mempengaruhi <i>utilization</i> memori sistem OAuth 2.0? (RQ4)	Rata-rata penggunaan memori (M4)
	Bagaimana keunggulan Redis dalam manajemen token OAuth 2.0 berdasarkan waktu respons ( <i>time response</i> ), <i>throughput</i> dan <i>utilization</i> ? (RQ5)	Rata-rata waktu respons (M1)
		Rata-rata <i>throughput</i> (M2)
		Rata-rata penggunaan CPU (M3)
		Rata-rata penggunaan memori (M4)

Pada gambar 3.2 di bawah memberikan visualisasi hubungan masing-masing pertanyaan (RQ1-RQ5) dengan metrik (M1-M4) terkait terhadap tujuan penelitian.



Gambar 3.2 *Goal Question Metric*

### 3.5 Hipotesis Penelitian

Rumusan pertama dibagi menjadi empat hipotesis, yang diberi label RQ1 - RQ4. Pertanyaan kelima merupakan rumusan masalah kedua, yang diberi label RQ5, tidak memerlukan pengujian hipotesis. Di bawah ini adalah hipotesis yang diajukan.

- 1) Hipotesis pertama (RQ1) “Apakah implementasi Redis dalam manajemen token mempengaruhi waktu respons sistem OAuth 2.0?” diusulkan sebagai berikut:
  - $H_0$ : implementasi Redis tidak mempengaruhi waktu respons sistem OAuth 2.0.
  - $H_1$ : implementasi Redis mempengaruhi waktu respons sistem OAuth 2.0.
- 2) Hipotesis kedua (RQ2) “Apakah implementasi Redis dalam manajemen token mempengaruhi *throughput* sistem OAuth 2.0?” diusulkan sebagai berikut:
  - $H_0$ : implementasi Redis tidak mempengaruhi *throughput* sistem OAuth 2.0.
  - $H_1$ : implementasi Redis mempengaruhi *throughput* sistem OAuth 2.0.

3) Hipotesis ketiga (RQ3) “Apakah implementasi Redis dalam manajemen token mempengaruhi penggunaan CPU sistem OAuth 2.0?” diusulkan sebagai berikut:

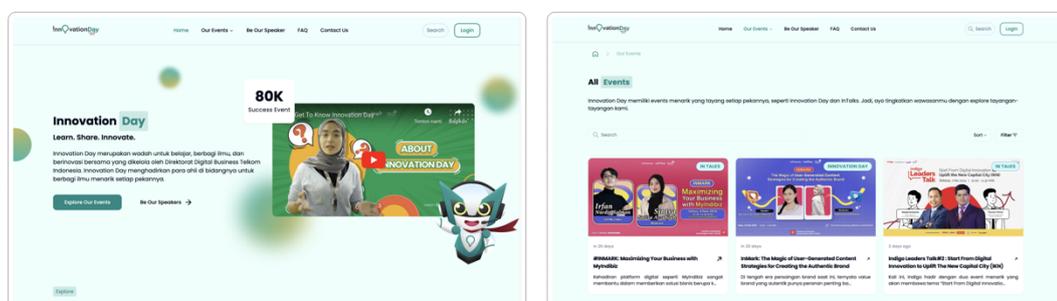
- $H_0$ : implementasi Redis tidak mempengaruhi penggunaan CPU sistem OAuth 2.0.
- $H_1$ : implementasi Redis mempengaruhi penggunaan CPU sistem OAuth 2.0.

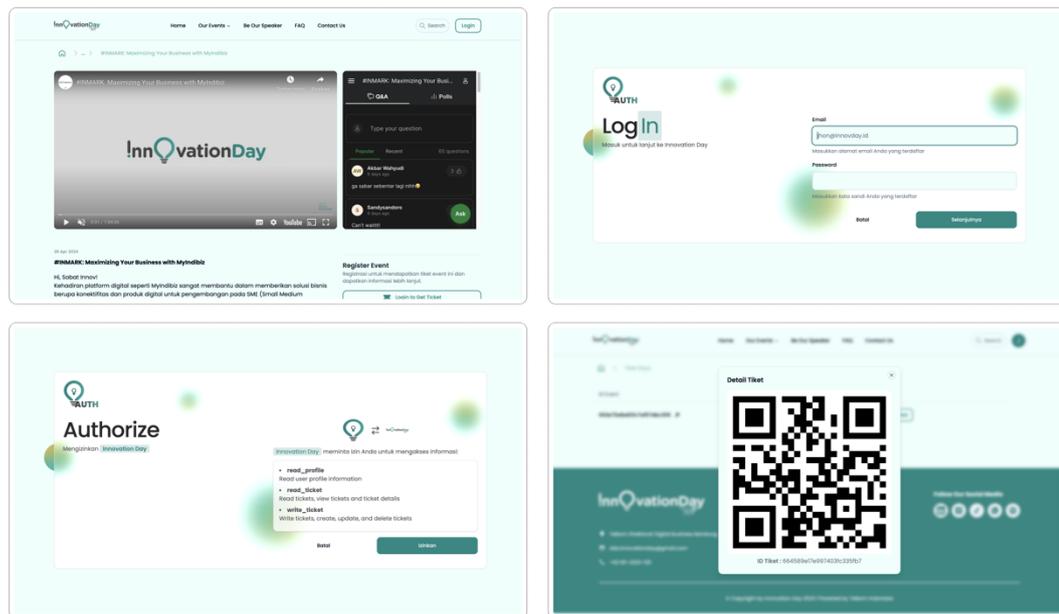
4) Hipotesis keempat (RQ4) “Apakah implementasi Redis dalam manajemen token mempengaruhi penggunaan memori sistem OAuth 2.0?” diusulkan sebagai berikut:

- $H_0$ : implementasi Redis tidak mempengaruhi penggunaan memori sistem OAuth 2.0.
- $H_1$ : implementasi Redis mempengaruhi penggunaan memori sistem OAuth 2.0.

### 3.6 Gambaran Sistem

Penelitian ini menggunakan prototipe sistem berbasis *microservices* di bidang *event management* yang disebut Innovation Day. Proses pengembangan prototipe sistem tidak dibahas secara detail. Selanjutnya, prototipe sistem diintegrasikan dengan aplikasi web yang berfungsi sebagai *frontend* atau aplikasi klien dalam OAuth. Meskipun prototipe sistem tidak akan dilibatkan dalam proses pengujian, namun diharapkan dapat memberikan gambaran kasus nyata penggunaan OAuth 2.0 dari sisi pengguna. Pengujian akan dilakukan pada sistem atau layanan OAuth 2.0 yang akan diintegrasikan dengan prototipe sistem untuk memastikan kesesuaiannya dengan tujuan penelitian.





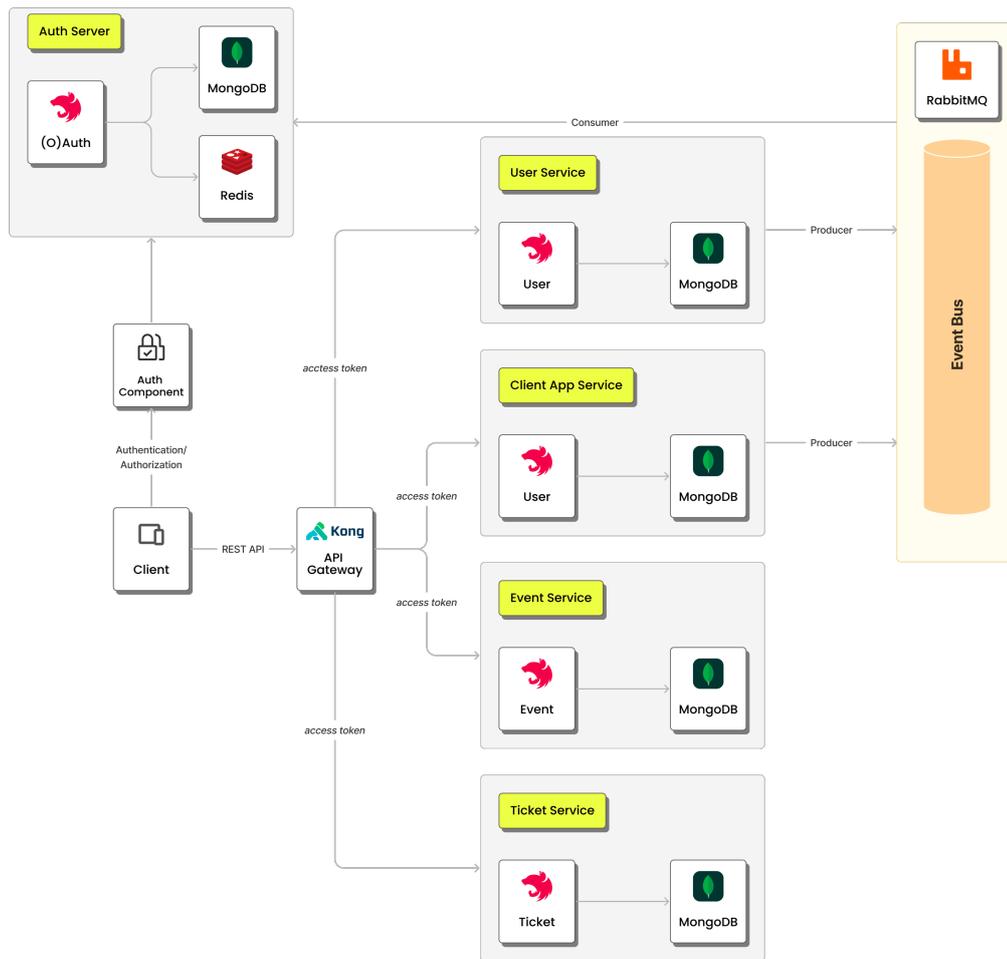
Gambar 3. 3 Antarmuka Prototipe Sistem

### 3.6.1 Validasi Prototipe Sistem

Prototipe sistem adalah produk perangkat lunak eksperimental yang dikembangkan selama masa magang di Direktorat Digital Bisnis Telkom Indonesia untuk mendukung *event* dan *sharing knowledge*. Prototipe ini dibuat berdasarkan penelitian-penelitian terdahulu, khususnya kasus penggunaan OAuth 2.0 yang dominan pada sistem dengan arsitektur *microservices*. Prototipe sistem ini memiliki fitur dekomposisi *user*, *event*, *ticket*, *client*, dan OAuth.

### 3.6.2 Rancangan Sistem

Pada sisi *server* atau *backend* menggunakan arsitektur *microservices*. Semua layanan menggunakan basis data NoSQL MongoDB, sedangkan Auth *server* atau sistem OAuth 2.0 menggunakan IMDB Redis sebagai basis data tambahan untuk manajemen token. Gambar 3.4 di bawah ini menunjukkan visualisasi dari arsitektur tersebut.



Gambar 3. 4 Arsitektur Sistem

### 3.6.3 Spesifikasi API

Masing-masing fungsionalitas layanan dari *microservices* dicantumkan ke dalam dua tabel terpisah. Namun, hanya *endpoint* tertentu dari Auth Server yang akan diuji, karena tidak semua *endpoint* digunakan dalam pengujian. Adapun spesifikasi dari Auth Server dirancang berdasarkan kerangka kerja OAuth 2.0 (Lihat lampiran 2).

Tabel 3. 5  
Spesifikasi API Auth Server

<b>Layanan</b>	<b>Endpoint</b>	<b>Tujuan atau Fungsionalitas</b>
Auth <i>Server</i>	GET /authorize	Mengalihkan <i>request</i> ke Auth <i>Component</i> (halaman <i>login</i> )
	POST /authorize	Mendapatkan kode otorisasi
	POST /token	Mendapatkan <i>access token</i> dan menyimpan <i>refresh token</i>
	DELETE /revoke	Menghapus <i>refresh token</i>
	POST /login	Melakukan proses otentikasi

Setiap *endpoint* memiliki beberapa parameter dan atribut headers untuk mengelola permintaan aplikasi klien. Lihat Lampiran 3 untuk informasi lebih lanjut terkait spesifikasi API Auth *Server*.

Tabel 3. 6  
Spesifikasi API Prototipe Sistem

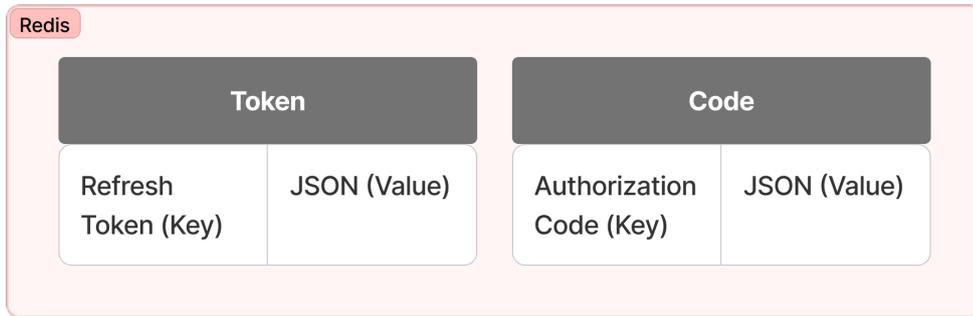
<b>Layanan</b>	<b>Endpoint</b>	<b>Tujuan atau Fungsionalitas</b>
<i>User</i>	POST /users	Mendaftarkan user baru
	GET /users	Memperoleh seluruh <i>user</i>
	GET /users/:id	Memperoleh <i>user</i> yang spesifik
	PATCH /users/:id	Memperbarui <i>user</i> yang spesifik
	DELETE /users/:id	Menghapus <i>user</i> yang spesifik
<i>Client</i>	POST /clients	Mendaftarkan aplikasi klien
	GET /clients	Memperoleh seluruh aplikasi klien

Layanan	Endpoint	Tujuan atau Fungsionalitas
	GET /clients/:id	Memperoleh aplikasi klien yang spesifik
	PATCH /clients/:id	Memperbarui aplikasi klien yang spesifik
	DELETE /clients/:id	Menghapus aplikasi klien yang spesifik
<i>Event</i>	GET /events	Memperoleh seluruh <i>event</i>
	GET /events/:id	Memperoleh <i>event</i> yang spesifik
<i>Ticket</i>	POST /tickets	Mendaftar <i>event</i> dan menyimpan <i>ticket</i>
	GET /tickets	Memperoleh seluruh <i>ticket</i>
	GET /tickets/:id	Memperoleh <i>ticket</i> yang spesifik

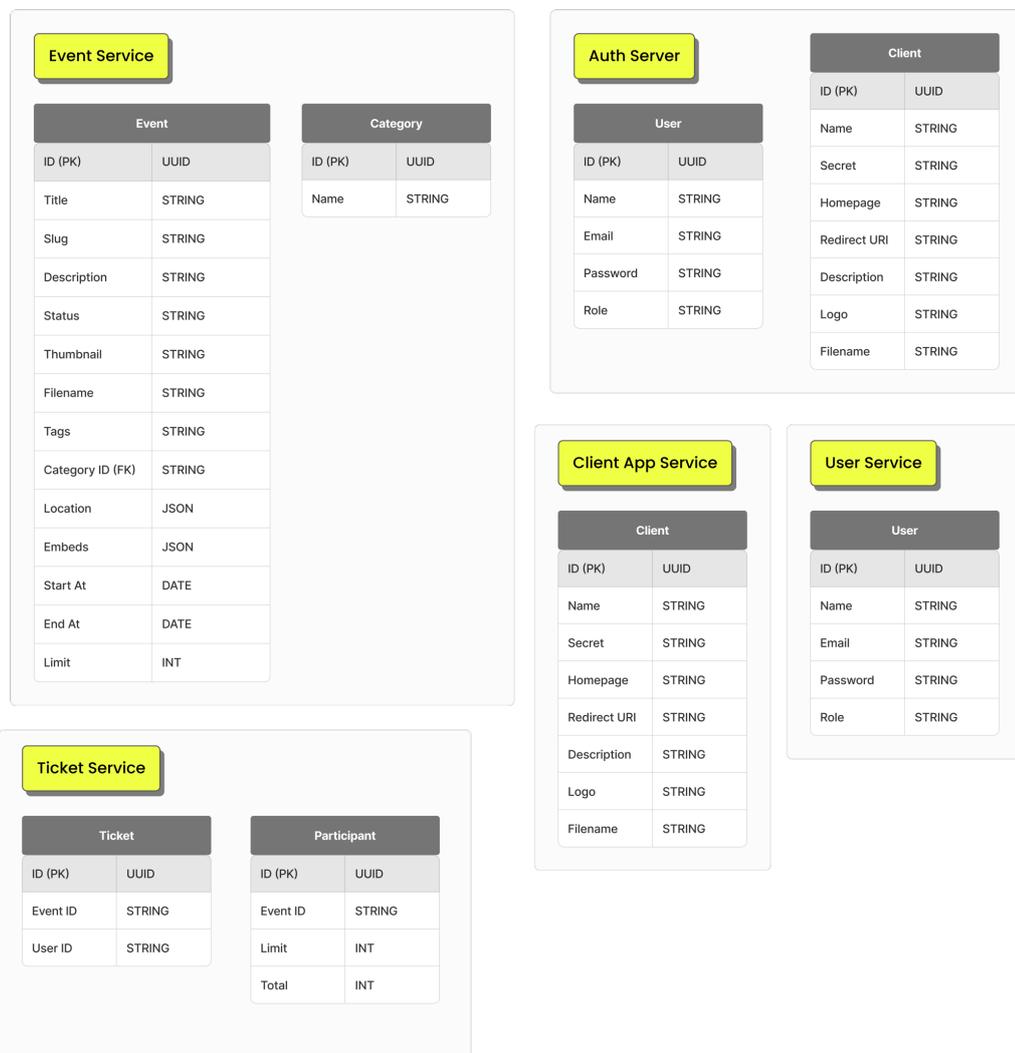
Tabel 3.6 menyajikan daftar layanan API yang tersedia, beserta *endpoint* dan fungsinya. Layanan-layanan ini tidak bergantung satu sama lain dan hanya dapat diakses melalui *API Gateway* dengan token akses pada *endpoint* tertentu.

### 3.6.4 Basis Data

*Entity Relationship Diagram* (ERD) dibuat sebagai rancang struktur basis data sesuai dengan fitur masing-masing layanan yang menggunakan basis data NoSQL. Desain ERD disesuaikan untuk mengakomodasi struktur berbasis dokumen dari basis data NoSQL MongoDB, tanpa menghilangkan atribut atau kolom apa pun. Gambar 3.5 ditujukan untuk mengakomodasi struktur *key-value* dari IMDB Redis. Sedangkan gambar 3.6 menunjukkan masing-masing basis data untuk setiap layanan (*service*) *microservices*.



Gambar 3.5 Entity Relationship Diagram Redis



Gambar 3.6 Entity Relationship Diagram MongoDB

### 3.6.5 Data Seeder

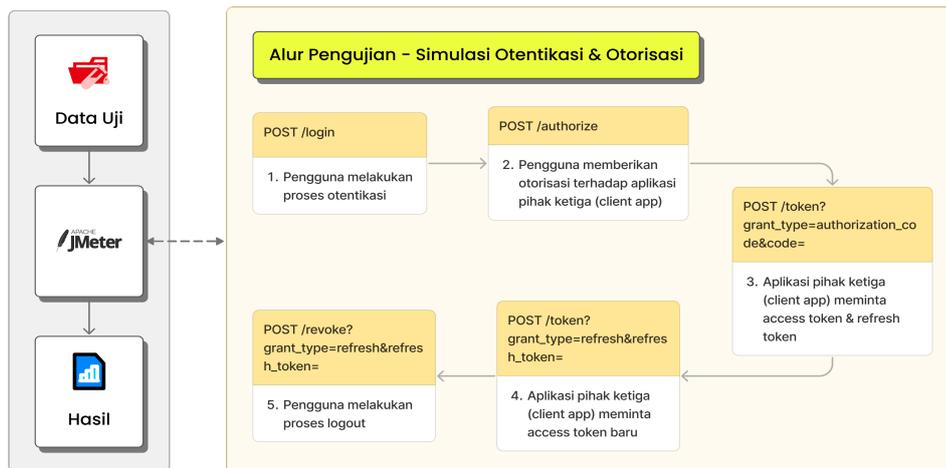
Dalam memenuhi kriteria dan skenario pengujian, data *seeder* digunakan untuk menghasilkan lingkungan pengujian realistis dengan data yang secara akurat mencerminkan kondisi produksi dan telah terisi ke dalam basis data. Adapun tabel yang telah tersisi, yaitu tabel *User* dengan 500 data, tabel *Client App* dengan 1 data, tabel *Event* dengan 20 data, dan tabel *Category* dengan 2 data. Dataset pada tabel *User* didapatkan dari repositori *open source*, sedangkan tabel *Event* dan *Category* menggunakan dataset dari PT Telkom Indonesia. Lampiran 4 – Lampiran 6 berisi informasi tambahan tentang dataset yang diadaptasi secara keseluruhan.

## 3.7 Pengumpulan Data

Skenario pengujian mengumpulkan semua hasil pengukuran untuk setiap metrik atau pengukuran yang dimaksud. Hasilnya kemudian dikirimkan ke sistem OAuth 2.0 untuk dilakukan uji beban (*load testing*). Proses transformasi kemudian digunakan untuk menyesuaikan unit setiap metrik dari hasil uji beban yang telah dilakukan.

### 3.7.1 Skenario Pengujian

Proses otentikasi dan otorisasi pengguna pada aplikasi pihak ketiga (*client app*) disimulasikan. Selain itu, metode pemrosesan data utama seperti menulis, membaca, dan menghapus diimplementasikan dalam skenario pengujian (Kabakus & Kara, 2017; Osemwegie dkk., 2018; Yang dkk., 2021). Alat JMeter digunakan untuk mengurutkan eksekusi setiap tahap. Gambar 3.7 menampilkan alur skenario pengujian untuk setiap permintaan.



Gambar 3. 7 Skenario Pengujian

#### 1) Login (R1)

Ketika mengakses POST /login, identitas pengguna diverifikasi dalam basis data. Tahap ini diasumsikan sebagai tahap awal setelah pengguna mendaftar di sistem dan sebelum mengotorisasi aplikasi pihak ketiga yang telah terdaftar.

#### 2) Create Authorization Code (R2)

Ketika mengakses POST /authorize, aplikasi pihak ketiga akan memperoleh kode otorisasi. Tahap ini diasumsikan sebagai tahap pengguna memberikan otorisasi kepada aplikasi pihak ketiga (*client app*) dengan menyetujui syarat dan ketentuan yang diberikan melalui komponen Auth. Setelah mendapatkan kode otorisasi, pengguna dikembalikan ke aplikasi pihak ketiga.

#### 3) Create Access Token & Refresh Token (R3)

Ketika mengakses POST /token?grant\_type=authorization\_code&code= (berserta kode otorisasi), aplikasi pihak ketiga akan memperoleh *access token* dan *refresh token*. Tahap ini diasumsikan ketika pengguna diarahkan kembali ke aplikasi pihak ketiga, secara otomatis akan mengakses *endpoint* untuk memperoleh token.

#### 4) Create New Access Token (R4)

Ketika mengakses POST `/token?grant_type=refresh&refresh_token=` (beserta *refresh token*), aplikasi pihak ketiga akan memperoleh *access token* baru. Tahap ini diasumsikan ketika pengguna mengunjungi kembali aplikasi pihak ketiga tetapi *access token* pengguna berakhir, maka aplikasi pihak ketiga akan memperbarui *access token* menggunakan *refresh token* yang telah tersimpan.

#### 5) *Delete Refresh Token* (R5)

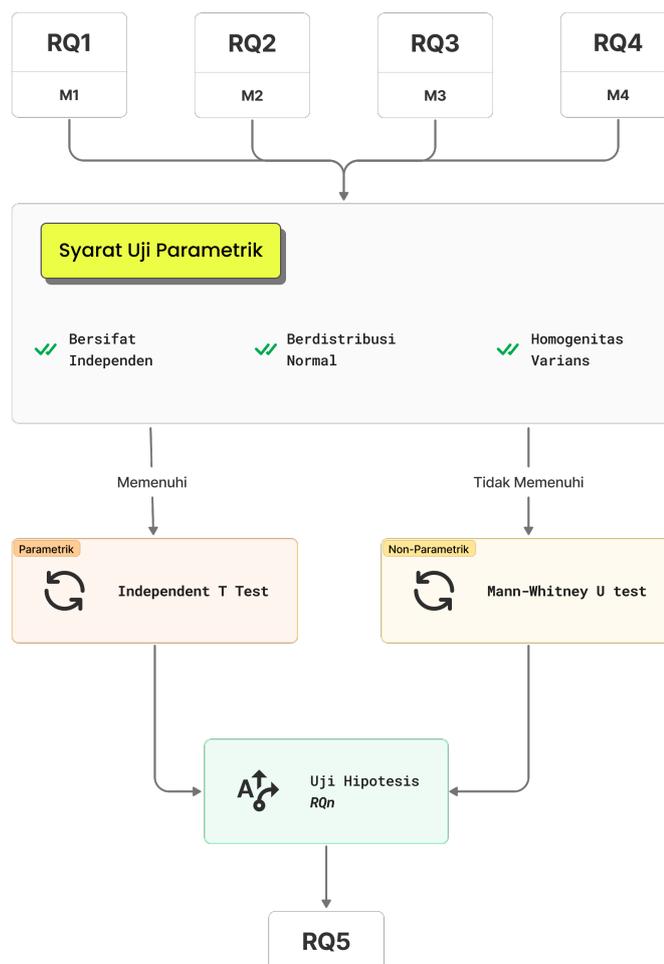
Ketika mengakses POST `/revoke?grant_type=refresh&refresh_token=` (beserta *refresh token*), memproses penghapusan *refresh token* dalam basis data. Tahap ini diasumsikan ketika pengguna melakukan *logout* dari aplikasi pihak ketiga sebagai antisipasi penyalahgunaan *refresh token*.

### 3.7.2 Kriteria Pengujian

Dataset *User* terlebih dahulu dimasukkan ke dalam basis data utama sebelum pengujian beban dimulai. Setiap tahap pengujian memerlukan penyisipan *payload* dalam bentuk *request body*, *header*, dan atribut pada *endpoint*. Lampiran 4 menguraikan atribut dari fragmen dataset. Pengujian dilakukan dengan meningkatkan beban secara bertahap berdasarkan jumlah pengguna virtual, yaitu 100, 300, dan 500 pengguna secara bersamaan dalam waktu 300 detik (Suryawan & Muliantara, 2024). Jumlah yang dipilih ditentukan melalui penerapan praktik terbaik dalam pengujian beban awal dengan tujuan untuk memahami kinerja dasar sistem. Berdasarkan penelitian sebelumnya, jumlah ini dapat mengidentifikasi dan mengatasi potensi *bottleneck* dan masalah kinerja tanpa membebani sistem secara berlebihan pada tahap awal pengujian (Hendayun dkk., 2023).

### 3.8 Analisis Data

Analisis awal dilakukan secara deskriptif dengan membandingkan hasil uji beban (*load testing*) pada tiga jenis beban yang berbeda berdasarkan desain dan skenario di atas. Analisis lebih lanjut dilakukan secara inferensial menggunakan hasil jenis beban terbesar untuk menguji hipotesis. Alur analisis digambarkan pada gambar 3.8. Sehubungan dengan RQ5, pengujian hipotesis diperlukan pada M1 hingga M4.



Gambar 3. 8 Alur Analisis Data

### 3.8.1 Tanggapan Terhadap *Outliers*

Sebelum memulai tahap prasyarat dan pengujian hipotesis, setiap anomali (*outliers*) yang teridentifikasi dalam dataset akan dihilangkan. Hal ini direkomendasikan karena sering kali mengakibatkan perubahan signifikan pada substansi kesimpulan. Formulasi rentang interkuartil (IQR) berikut ini digunakan untuk mengilustrasikan *outliers* pada diagram boxplot (Aguinis dkk., 2013).

$$IQR = Q3 - Q1 \tag{5}$$

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

$$outliers_{high} = Q3 + (1.5 IQR) \tag{7}$$

### 3.8.2 Prasyarat Uji Parametrik

Ketika memastikan dataset yang dimaksud dapat diperoleh melalui pendekatan parametrik, perlu dipastikan bahwa dataset yang dimaksud memenuhi sejumlah syarat yang ditentukan (Usmadi, 2020).

- 1) Sampel data tidak saling mempengaruhi atau harus independen satu sama lain.
- 2) Populasi data yang dikumpulkan harus terdistribusi secara normal, sehingga memerlukan uji normalitas menggunakan *Kolmogorov Smirnov*.

$$Z \text{ score} = \frac{X - \bar{X}}{\sigma} \quad (8)$$

$$\sigma = \sqrt{\frac{\sum(x_i - \bar{x})^2}{n - 1}} \quad (9)$$

Dimana:

- $\bar{X}$  = rata-rata
- $\sigma$  = simpangan baku

- 3) Varians populasi antar kelompok harus sama (homogenitas varians) *Levene*.

$$W = \frac{(n - k)}{k - 1} \frac{\sum_{i=1}^k n_i (\bar{Z}_i - \bar{Z}_{..})^2}{\sum_{i=1}^k \sum_{j=1}^{n_i} n_i (\bar{Z}_{ij} - \bar{Z}_i)^2} \quad (10)$$

$$Z_{ij} = |Y_{ij} - \bar{Y}_i| \quad (11)$$

Dimana:

- $n$  = jumlah perlakuan
- $k$  = banyak kelompok
- $\bar{Y}_i$  = rata-rata dari kelompok ke-i
- $\bar{Z}_i$  = rata-rata kelompok dari  $\bar{Z}_i$
- $\bar{Z}_{..}$  = rata-rata menyeluruh dari  $Z_{ij}$

### 3.8.3 Teknik Analisis Data

Jika dataset memenuhi kriteria yang diperlukan untuk pengujian parametrik, dataset tersebut akan dianalisis melalui *Independent t-test*. Jika tidak demikian, analisis dilakukan secara non-parametrik melalui *Mann-Whitney U Test (Wilcoxon)*

*rank-sum test*). Bagian berikut ini memberikan penjelasan mengenai kedua teknik analisis tersebut.

- 1) *Independent t-test*, dikenal juga sebagai uji-t tidak berpasangan adalah uji statistik untuk mengetahui apakah ada perbedaan statistik yang signifikan antara rata-rata dua kelompok yang tidak berhubungan (*independen*). (Mishra dkk., 2019).

$$t_{hitung} = \frac{\bar{X}_1 - \bar{X}_2}{S_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \quad (12)$$

$$S_p = \sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}} \quad (13)$$

$$t_{table} = \alpha; df = n_1 + n_2 - 2 \quad (14)$$

Dimana:

- $\bar{X}_1$  dan  $\bar{X}_2$  = Rata-rata dari dua sampel
- $n_1$  dan  $n_2$  = Ukuran dari dua sampel
- $S_p$  = Standar deviasi gabungan
- $s_1$  dan  $s_2$  = Standar deviasi dari dua sampel
- $\alpha$  = nilai signifikansi
- $df$  = derajat kebebasan (*degree of freedom*)

Keputusan:

- $t_{hitung} > t_{table}$  maka  $H_0$  ditolak
- 2) *Mann-Whitney U Test*, dikenal juga sebagai *Wilcoxon rank-sum test* merupakan alternatif non-parametrik yang sering digunakan untuk uji-t dua sampel. Uji *Mann-Whitney U* menggunakan data yang diperingkat sebagai pengganti nilai asli, sehingga lebih tahan terhadap outlier dan distribusi non-normal (Perme & Manevski, 2019).

$$U_1 = n_1 n_2 + \frac{n_1(n_1 + 1)}{2} - R_1 \quad (15)$$

$$U_2 = n_1 n_2 + \frac{n_2(n_2 + 1)}{2} - R_2 \quad (16)$$

Dimana:

- $n_1$  dan  $n_2$  = Jumlah sampel 1 dan 2
- $R_1$  dan  $R_2$  = Jumlah rangking pada sampel  $n_1$  dan  $n_2$
- $U_1$  dan  $U_2$  = Jumlah peringkat 1 dan 2