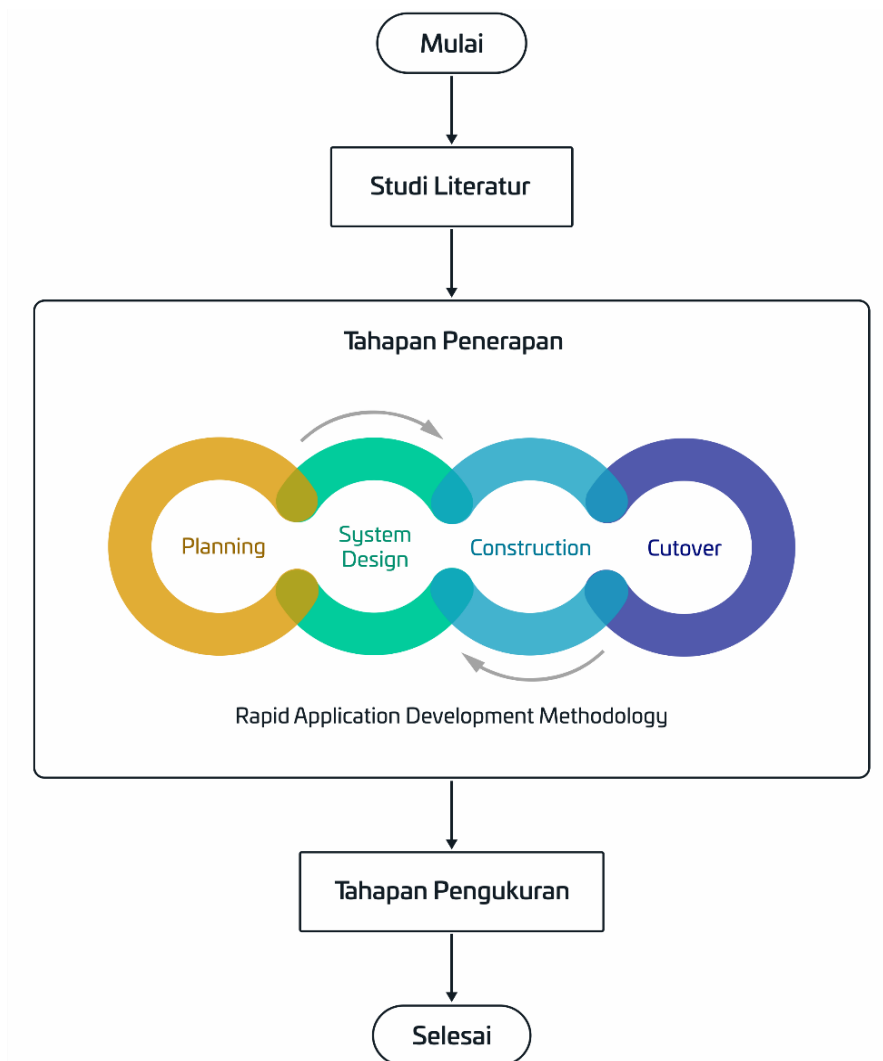


## BAB III

### METODE PENELITIAN

#### 3.1 Tahapan Penelitian

Gambar 3.1 merupakan tahapan penelitian yang dimulai dari proses studi literatur, tahapan penerapan (RAD Methodology) yang memiliki 4 tahapan yaitu *planning*, *system design*, *construction*, dan *cutover*, tahapan pengukuran, dan penarikan kesimpulan.



Gambar 3.1 Tahapan Penelitian

### 3.1.1 Studi Literatur

Pada tahap ini, mempelajari semua hal yang berkaitan dengan *Microservice*, *Containerization*, dan *Event-Driven Architecture* dari berbagai sumber pustaka berupa buku, jurnal, laporan penelitian, skripsi, dan tesis yang telah dilakukan serta hasil pencarian pustaka di internet.

### 3.1.2 Planning

Pada tahap ini dilakukan perencanaan alur data, pembangunan dan penerapan arsitektur pada aplikasi, termasuk perencanaan kebutuhan *software development* (bahasa pemrograman, *tools*, *library*, *framework*, dan *resources*).

Arsitektur aplikasi akan diterapkan menggunakan Kubernetes untuk mengotomatisasi, menskalakan, dan mengelola *container* aplikasi, dan dalam menangani proses menjalankan *container* aplikasi akan dibantu Skaffold, serta *server* alur komunikasi atau *event bus* akan dibantu dengan NATS Streaming. Selain itu, terdapat beberapa Kubernetes *object* yang akan digunakan dalam penerapan arsitektur aplikasi yaitu ClusterIP Service, Deployment, Pod, Persistent Volume, Ingress/Ingress Nginx. Semua Kubernetes *object* yang didefinisikan akan membantu dalam proses komunikasi, pengelolaan layanan, koneksi dengan *database*, *router*, dan lain sebagainya.

Aplikasi yang akan dibangun menggunakan bahasa pemrograman Typescript, Node.js sebagai *runtime environment*, Express.js sebagai *framework*, Jest untuk melakukan *unit test* pada aplikasi, dan MongoDB sebagai *database* NoSQL, serta *resources/libraries* lain untuk kebutuhan pembangunan aplikasi.

### **3.1.3 System Design**

*System Design* merupakan tahapan perancangan arsitektur sistem sesuai dengan tahapan *planning* yang telah dilakukan sebelumnya. Perancangan arsitektur akan dilakukan menggunakan *diagrams.net* yang merupakan *tool* untuk membuat diagram *data flow*, *wireframe*, UML dan lain sebagainya.

### **3.1.4 Construction**

*Construction* merupakan tahapan konstruksi sesuai dengan tahapan *system design*. Selain itu, pada tahap ini juga dilakukan pengujian pada aplikasi juga komunikasi antar layanan, jika masih terdapat ketidaksesuaian maka dapat kembali ke tahap *system design* lalu kembali ke tahap *construction*. Hal tersebut dilakukan hingga sesuai dengan tahap *planning* yang dilakukan sebelumnya.

### **3.1.5 Cutover**

Tahap ini merupakan tahap *final* dimana arsitektur sistem telah berjalan dengan baik dan sesuai dengan *planning* di awal. Pada penelitian ini, jika arsitektur sistem telah selesai diimplementasikan akan dilakukan pengukuran untuk kebutuhan penelitian.

### **3.1.6 Pengukuran**

Pada tahap ini akan dilakukan pengukuran dengan mengevaluasi performa arsitektur yang diimplementasikan. Pengukuran yang dilakukan meliputi penggunaan CPU (*CPU usage*) (Dewi et al. 2019) dalam satuan *milliseconds* ketika

memproses *request* dari *client*, waktu respon (*response time*) dalam satuan *milliseconds* ketika menangani sejumlah *request* dari *client*, dan tingkat kesalahan (*error rate*) dalam satuan persentase (*percentage*) (Hong, Sik Yang, and Kim 2018) ketika menangani peningkatan sejumlah *request* diluar batas kemampuan sistem. Tahapan pengukuran dibantu menggunakan JMeter untuk mengetahui hasil *response time* dan *error rate*, serta dibantu dengan API *process.cpuUsage()* dari Node.js untuk mengetahui hasil dari CPU *usage*.

### **3.1.7 Penarikan Kesimpulan**

Penarikan kesimpulan adalah tahap akhir dari proses penelitian dimana pada tahap ini hasil dari tahap pengukuran akan diambil kesimpulan sebagai hasil penelitian terhadap solusi meningkatkan kinerja pada arsitektur *microservice*.