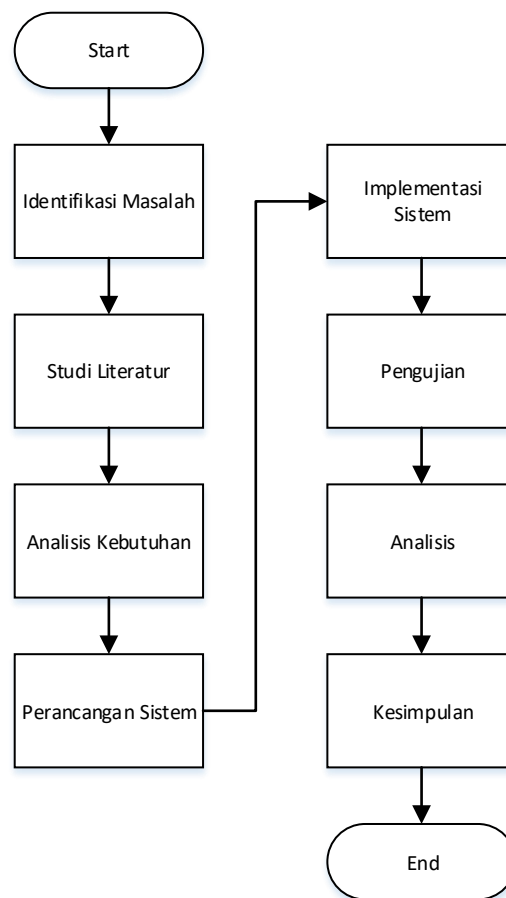


BAB 3 METODOLOGI PENELITIAN

Pada bagian ini dijelaskan tentang metode dan langkah-langkah yang dilakukan dalam penelitian ini.

3.1 Metodologi Penelitian

Pada bagian ini menjelaskan langkah-langkah yang akan dilakukan dalam pengerjaan tugas akhir seperti pada Gambar 3.1 berikut.



Gambar 3.1 Diagram Alir Metode Penelitian

Berdasarkan Gambar 3.1, dapat diuraikan langkah-langkah metodologi penelitian yang akan dilakukan, yaitu:

1. Identifikasi Masalah yang harus diselesaikan dalam penelitian ini.

2. Studi literatur penelitian sebelumnya yang terkait, *Software-defined networking, Mininet, Ryu Controller, Routing, Algoritme Dijkstra* dan logika *fuzzy Tsukamoto dan Mamdani*.
3. Analisis kebutuhan sistem yang meliputi kebutuhan fungsional dan non-fungsional.
4. Perancangan atau desain sistem, topologi, dan lingkungan untuk menerapkan logika *fuzzy* pada SDN.
5. Implementasi sistem pada SDN.
6. Pengujian sistem logika *fuzzy* pada SDN.
7. Analisis hasil pengujian berdasarkan parameter uji yang telah ditentukan.
8. Penarikan kesimpulan berdasarkan hasil analisis pengujian yang dilakukan terhadap sistem.

3.2 Studi Literatur

Pada penelitian ini, dilakukan studi literatur untuk menjadi dasar dan landasan terhadap perancangan dan implementasi. Berbagai studi literatur yang dilakukan terhadap dasar-dasar SDN beserta aplikasi dan penerapannya. Kemudian dilakukan studi literatur tentang *routing* dengan algoritme *Dijkstra* dan logika. Setelah itu, dilakukan studi literatur tentang *Ryu Controller* untuk pengembangan aplikasi-aplikasi SDN dan penerapan *routing* dengan logika *fuzzy* dan algoritme *Dijkstra* menggunakan *Ryu Controller*. Studi literatur terakhir dilakukan untuk mengetahui berbagai parameter uji yang cocok digunakan dengan penelitian ini.

3.3 Analisis Kebutuhan

Analisis kebutuhan diperlukan untuk mengidentifikasi dan menganalisis apa saja yang dibutuhkan dalam melakukan penelitian ini. Kebutuhan diklasifikasikan dalam kebutuhan fungsional dan kebutuhan non-fungsional.

3.3.1 Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan tentang apa saja yang dapat dilakukan oleh sistem, informasi apa saja yang harus ada dan bisa dihasilkan oleh sistem. Berikut ini adalah kebutuhan fungsional dari sistem ini, yaitu:

- 1) Sistem dapat memonitoring *traffic* dan *delay* yang digunakan sebagai *parameter* pada logika *fuzzy*.
- 2) Sistem dapat menentukan bobot *link* dengan menggunakan logika *fuzzy*.
- 3) Sistem dapat menentukan rute terpendek menggunakan algoritme *Dijkstra* sesuai dengan bobotnya.

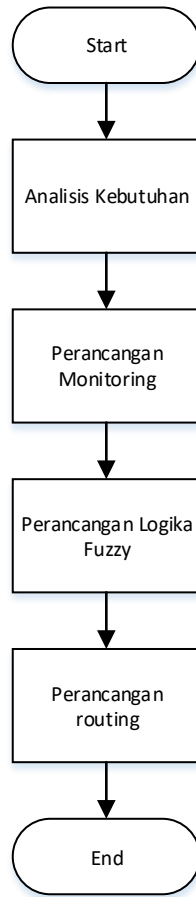
3.3.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional merupakan kebutuhan perangkat keras maupun perangkat lunak untuk mendukung pembangunan sistem dan melancarkan penelitian ini. Dalam penelitian ini menggunakan Mininet sebagai emulator SDN dikarenakan Mininet memiliki tampilan yang sederhana dan mudah digunakan oleh orang awam. Selain itu hardware teknologi SDN sendiri memiliki harga yang relatif mahal dan juga membutuhkan banyak switch untuk mengimplementasikan fungsi routing agar berjalan dengan optimal. Berikut merupakan kebutuhan non-fungsional pada penelitian ini.

- a. Perangkat keras:
Komputer/laptop dengan spesifikasi:
 - CPU: Intel Core i3.
 - RAM: 4096 MB.
 - Hard disk: 500 GB.
- b. Perangkat lunak:
 - *Ubuntu Linux 16.04 64-bit*: sebagai sistem operasi.
 - *Mininet*: sebagai emulator dalam membangun topologi.
 - *Ryu*: sebagai *SDN Controller*.
 - *sFlow-RT*: sebagai *monitor traffic* jaringan.
 - *Iperf, ping*: sebagai alat/aplikasi untuk melakukan pengujian.

3.4 Perancangan Sistem

Perancangan sistem mengacu bagaimana cara membuat sistem agar mampu mencapai tujuan dari penulisan penelitian ini. Sistem yang diharapkan adalah sistem yang mampu mengatasi kelemahan dari algoritme *routing* yang ada saat ini. Digunakan Logika *Fuzzy* untuk menentukan bobot tiap *link* dengan parameternya yaitu *traffic* dan *delay*. Langkah-langkah perancangan seperti pada gambar di bawah ini.

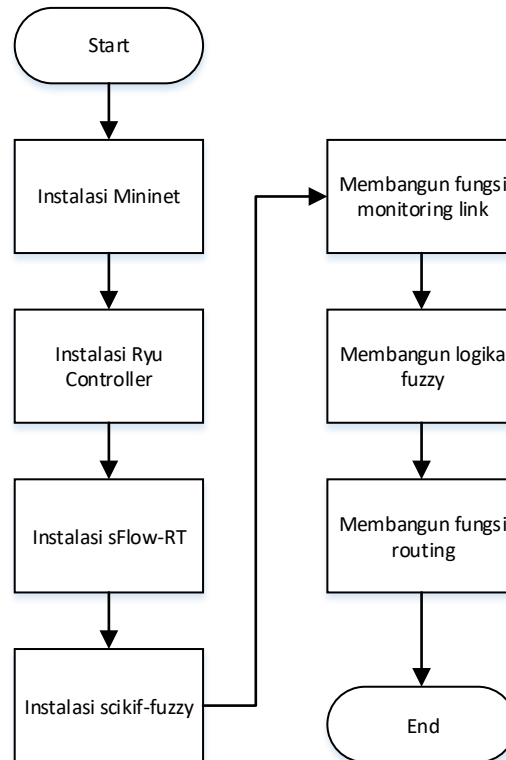


Gambar 3.2 Diagram Alir Perancangan Sistem Secara Umum

Pertama adalah analisis kebutuhan sistem yaitu tentang apa saja yang dibutuhkan dalam perancangan sistem. Selanjutnya adalah perancangan fungsi monitoring sebagai parameter dalam menentukan bobot. Kemudian dilakukan perancangan Logika *Fuzzy* untuk menentukan bobot *link*. Terakhir adalah melakukan perancangan *routing* untuk meneruskan paket dari pengirim ke penerima.

3.5 Implementasi

Implementasi sistem yang dibuat berdasarkan perancangan kebutuhan dan perancangan sistem yang telah dibuat sebelumnya seperti pada gambar berikut ini.



Gambar 3.3 Diagram Alir Implementasi Sistem

Implementasi dimulai dengan menginstal *Mininet*, *Ryu*, *sFlow-RT* dan *scikit-fuzzy*. Setelah semua kebutuhan perangkat lunak berhasil terinstal, maka dilanjutkan dengan membuat fungsi untuk memonitoring *traffic* dan *delay*. Hasil dari monitoring tersebut akan dijadikan sebagai masukan dalam perhitungan Logika Fuzzy. Selanjutnya dilakukan perhitungan bobot *link* menggunakan Logika Fuzzy. Setelah diketahui bobotnya, maka akan dilakukan perhitungan *shortest path* dengan menggunakan Algoritme Dijkstra.

3.6 Pengujian

Rencana pengujian akan dilakukan dengan cara membandingkan model algoritme *routing* diantaranya adalah algoritme *Dijkstra* dengan menggunakan logika *fuzzy* dan algoritme *Static Cost Dijkstra*. Pengujian berfokus pada pengujian fungsional. Pengujian fungsional berfungsi untuk melihat apakah kebutuhan fungsional sistem berjalan sesuai dengan perancangan atau tidak. Pengujian dilakukan untuk melakukan penilaian dan evaluasi mekanisme sebelum dan sesudah penerapan routing adaptif berdasarkan kepadatan lalu lintas jaringan menggunakan logika *fuzzy*. Parameter yang diujikan adalah *throughput*, *packet loss*, *convergence time* dan pencarian jalur. Untuk melakukan pengujian *throughput*, didapatkan dengan cara mengirimkan paket TCP sebesar-besarnya menuju server dengan variabel pembedanya berupa jumlah *client*. Untuk

mendapatkan nilai *packet loss* yaitu dengan cara mengirimkan paket UDP menuju server dengan variabel pembedanya berupa jumlah *client*. Kemudian untuk mendapatkan nilai *convergence time* dapat dilakukan dengan cara memberi timer pada Aplikasi *Controller*. Sedangkan untuk melakukan pengujian pencarian jalur, dapat dilakukan dengan cara menggunakan aplikasi MiniNAM dan bwm-ng.

3.7 Analisis

Setelah memperoleh data dari pengujian, analisis akan dilakukan untuk mengukur kinerja sistem apakah sistem sudah berjalan dengan baik dan sesuai dengan tujuan penelitian.

3.8 Kesimpulan

Pengambilan kesimpulan dilakukan berdasarkan pengujian dan analisis terhadap performa dari algoritme *Dijkstra* dengan logika *fuzzy* dibandingkan dengan algoritme routing jaringan konvensional. Berdasarkan hasil pengujian, dapat ditentukan apakah algoritme *Dijkstra* dengan logika *fuzzy* tersebut lebih unggul dibandingkan dengan algoritme *Static Cost Dijkstra*.