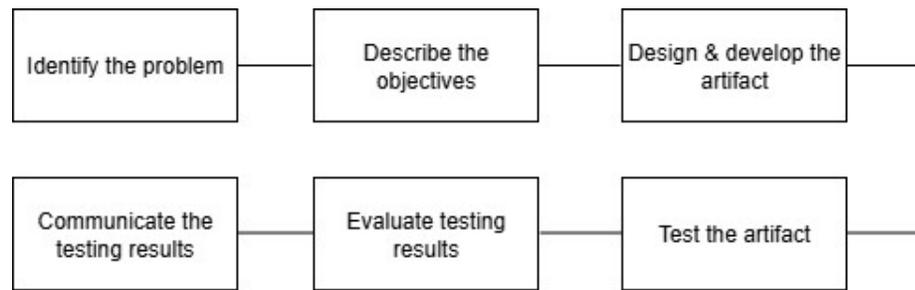


BAB III METODE PENELITIAN

Penelitian ini merupakan penelitian yang memakai metode D&D (*Design and Development*). Metode D&D merupakan metode yang umumnya bertujuan untuk mengidentifikasi permasalahan dan menghasilkan solusi yang dapat digunakan untuk memecahkan permasalahan tertentu (Nurul Fazriah dkk., 2024).

Menurut J. Ellis & Levy (2010), metode D&D terdiri dari enam tahapan. Tahapan pertama yang dilakukan adalah identifikasi masalah (*identify the problem*), yaitu proses untuk mengkaji secara mendalam permasalahan yang melatarbelakangi penelitian sehingga dapat memastikan bahwa masalah tersebut membutuhkan solusi penelitian. Lalu tahap kedua adalah mendeskripsikan tujuan (*describe the objectives*), tahapan ini dilakukan dengan cara menetapkan sasaran yang jelas dan relevan, sehingga penelitian dapat terfokuskan pada pemecahan masalah yang tepat. Tahapan berikutnya yaitu desain dan pengembangan sistem (*design and develop the artifact*), dalam hal ini peneliti melakukan perancangan sekaligus mengembangkan sistem. Lalu tahap keempat adalah melakukan pengujian sistem (*test the artifact*), yaitu melakukan pengujian solusi yang telah dirancang dan dikembangkan untuk memastikan fungsi dan kinerjanya sesuai dengan kebutuhan pengguna maupun tujuan penelitian. Tahap berikutnya yaitu evaluasi hasil pengujian (*evaluate testing results*), bertujuan untuk menilai kelebihan serta kekurangan dari sistem yang dikembangkan. Lalu tahapan keenam adalah menyampaikan hasil pengujian (*communicate the testing results*), pada tahapan terakhir ini dilakukan untuk mendokumentasikan serta menyampaikan hasil penelitian. Berikut merupakan alur dari metode D&D pada gambar 3.1.



Gambar 3.1 Alur Metodologi D&D

3.1. Identifikasi Masalah (*Identify the problem*)

Pada sebuah jaringan yang terhubung ke akses internet, keamanan data tentu menjadi hal utama yang perlu diperhatikan, terutama terkait akses ke situs berbahaya atau yang melanggar kebijakan, yang berisiko bagi pengguna. Meskipun terdapat aplikasi seperti Winbox yang mampu memblokir situs tertentu, kurangnya mekanisme pemantauan dan notifikasi peringatan membuat administrator jaringan sering terlambat mengetahui percobaan akses, sehingga berpotensi menimbulkan kebocoran data atau pelanggaran kebijakan. Proses pengecekan manual pada *log router* yang umum digunakan juga memakan waktu dan tidak efisien, sehingga diperlukan solusi yang mengintegrasikan perangkat Mikrotik dengan layanan notifikasi instan, seperti bot Telegram, untuk memantau dan memberikan pemberitahuan secara langsung kepada administrator.

3.2. Mendeskripsikan Tujuan (*Describe the objectives*)

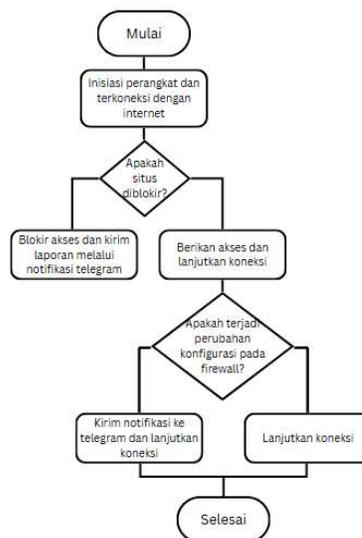
Berdasarkan permasalahan yang sudah diuraikan sebelumnya, pengembangan sebuah *website* dapat membantu untuk mengatasi permasalahan tersebut. *Website* yang terintegrasi dengan perangkat Mikrotik dapat memungkinkan administrator jaringan untuk melakukan konfigurasi *Layer 7 Protocol* dan *Simple Queue* secara lebih praktis dan efisien. Serta dengan penambahan fitur notifikasi pada aplikasi Telegram dapat membantu administrator untuk memantau kondisi jaringan.

3.3. Desain dan Mengembangkan Sistem (*Design & develop the artifact*)

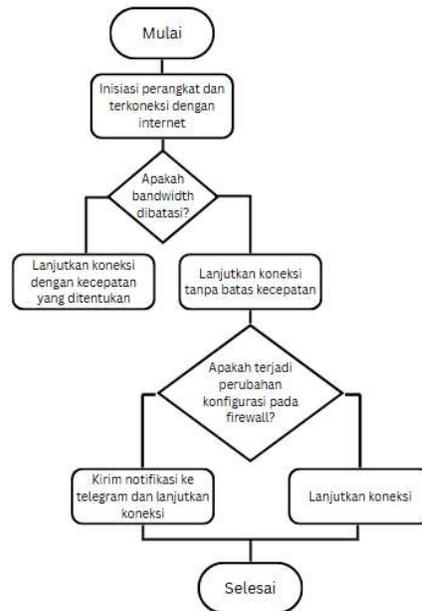
Pada tahap ini, sudah dimulai untuk perancangan yang meliputi desain flowchart, diagram arsitektur, diagram use case, perencanaan desain untuk website, perencanaan integrasi antara perangkat Mikrotik, *website*, dan bot Telegram, serta perencanaan penggunaan *tools* yang akan digunakan.

3.3.1. *Flowchart*

Pada gambar 3.2 menjelaskan tentang alur kerja fitur *firewall Layer 7 Protocol* pada website yang dikembangkan, dimulai dari proses inisiasi perangkat sampai mengirimkan notifikasi peringatan ke bot Telegram. Sedangkan pada gambar 3.3 menjelaskan tentang alur kerja fitur *Simple Queue* pada website yang dikembangkan, dimulai dari inisiasi perangkat sampai mengirimkan notifikasi perubahan konfigurasi ke bot Telegram.



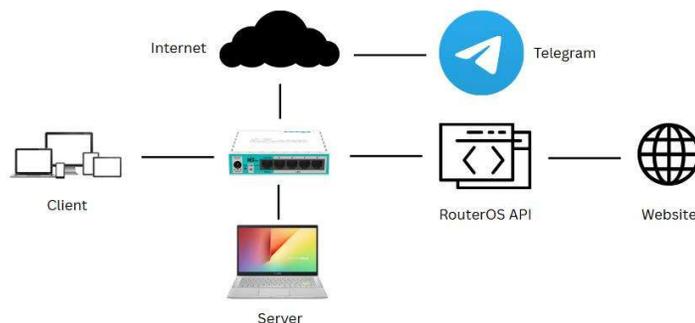
Gambar 3.2 Flowchart Firewall Layer 7 Protocol



Gambar 3.3 Flowchart Bandwidth Simple Queue

3.3.2. Diagram Arsitektur

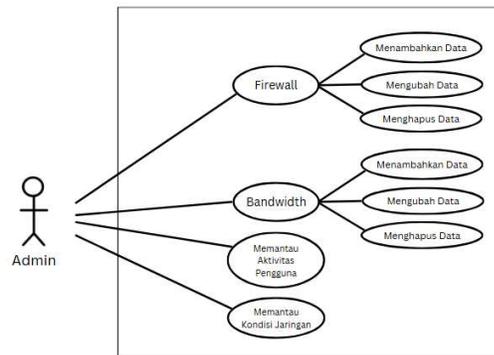
Pada gambar 3.4 menjelaskan tentang diagram arsitektur sistem pada penelitian ini. Pada perangkat Mikrotik, *interface* ether1 digunakan untuk terhubung ke jaringan internet, lalu pada ether2 digunakan pada *pc server* untuk konfigurasi *router*, sedangkan pada interface lainnya digunakan untuk dihubungkan pada perangkat *client*. Perangkat Mikrotik dapat terintegrasi dengan website dengan bantuan RouterOS API.



Gambar 3.4 Diagram Arsitektur Sistem

3.3.3. Diagram Use Case

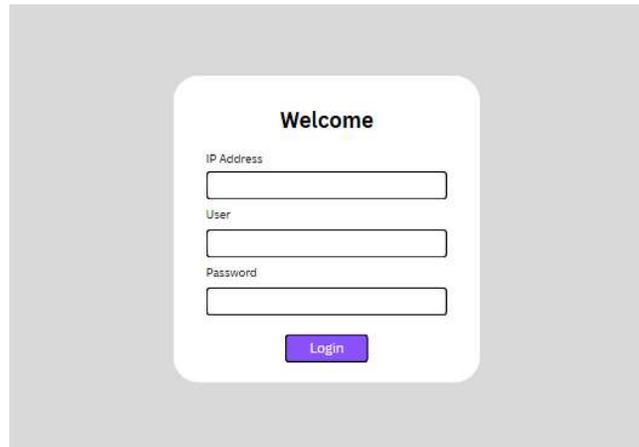
Pada gambar 3.5 menjelaskan tentang *use case* pada penelitian ini. Pada penelitian ini hanya terdapat satu *role* saja yaitu seorang administrator jaringan. Dalam hal ini, seorang admin dapat mengelola data pada fitur *firewall* dan *bandwidth*. Selain itu, admin dapat memantau aktivitas pengguna dan dapat memantau kondisi jaringan.



Gambar 3.5 Diagram Use Case

3.3.4. Mockup Website

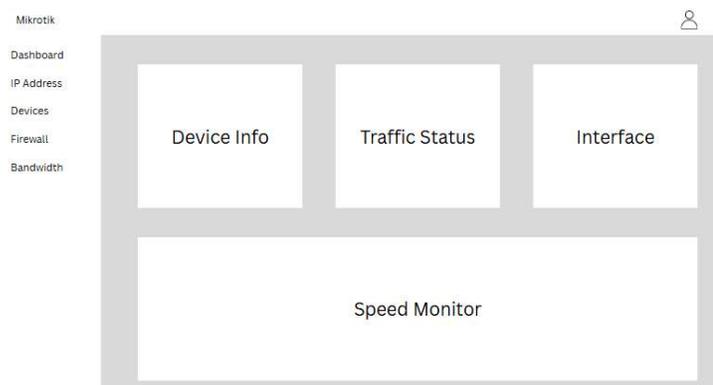
Berikut ini merupakan bentuk desain awal website dalam bentuk gambaran sketsa sebelum akhirnya dikembangkan untuk menjadi sebuah website yang terintegrasi dengan *router* Mikrotik. Gambar 3.6 berikut merupakan Gambaran halaman login sebelum admin masuk ke halaman dashboard.



The image shows a login form with a white background and rounded corners, centered on a gray background. At the top, the word "Welcome" is written in bold. Below it are three input fields labeled "IP Address", "User", and "Password". At the bottom of the form is a purple button labeled "Login".

Gambar 3.6 Mockup halaman login aplikasi web yang dikembangkan

Gambar 3.7 di bawah ini merupakan gambaran pada halaman dashboard setelah admin berhasil mengisi data pada halaman login.



Gambar 3.7 Mockup Halaman Dashboard aplikasi web yang dikembangkan

Gambar 3.8 di bawah ini merupakan gambaran halaman *ip address* yang berisikan tentang *ip address* dan *network* yang terdaftar pada *interface*

Address	Network	Interface

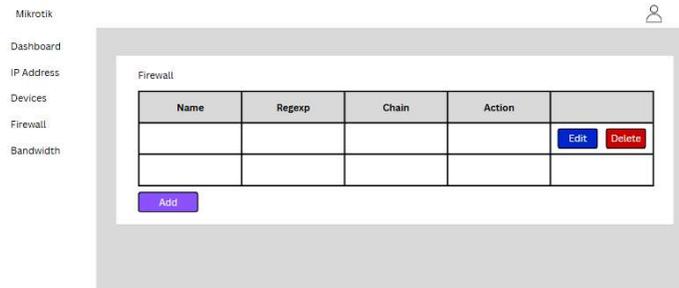
Gambar 3.8 Mockup Halaman IP Address aplikasi web yang dikembangkan

Gambar 3.9 di bawah ini merupakan gambaran halaman *devices*. Pada halaman ini admin dapat melihat *nama*, *mac address*, *ip address*, serta aktivitas dari perangkat yang terhubung.

MAC Address	Name	IP Address	Activity

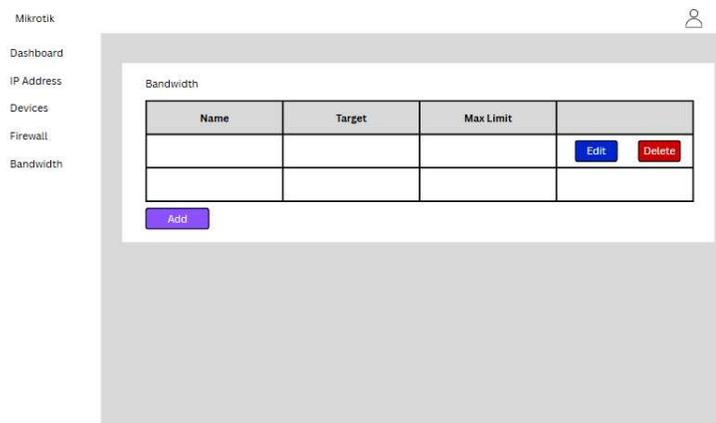
Gambar 3.9 Mockup Halaman Devices aplikasi web yang dikembangkan

Gambar 3.10 di bawah ini merupakan gambaran halaman *firewall*. Pada halaman ini admin dapat menambahkan, mengubah, dan menghapus data konfigurasi pada *firewall*.



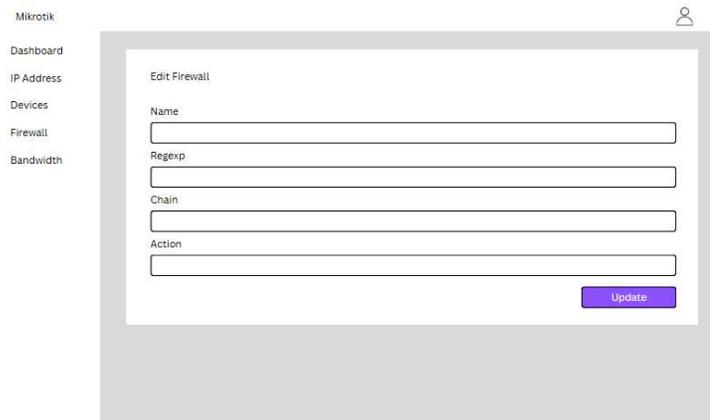
Gambar 3.10 Mockup Halaman Firewall aplikasi web yang dikembangkan

Gambar 3.11 di bawah ini merupakan gambaran pada halaman *bandwidth*. Pada halaman ini admin dapat melakukan penambahan, pengubahan, dan menghapus data konfigurasi pada *bandwidth*.



Gambar 3.11 Mockup Halaman Bandwidth aplikasi web yang dikembangkan

Gambar 3.12 di bawah ini merupakan gambaran pada halaman *edit firewall*. Halaman ini akan muncul setelah admin klik tombol “*edit*” di sebelah kanan item.



Mikrotik

Dashboard

IP Address

Devices

Firewall

Bandwidth

Edit Firewall

Name

Regexp

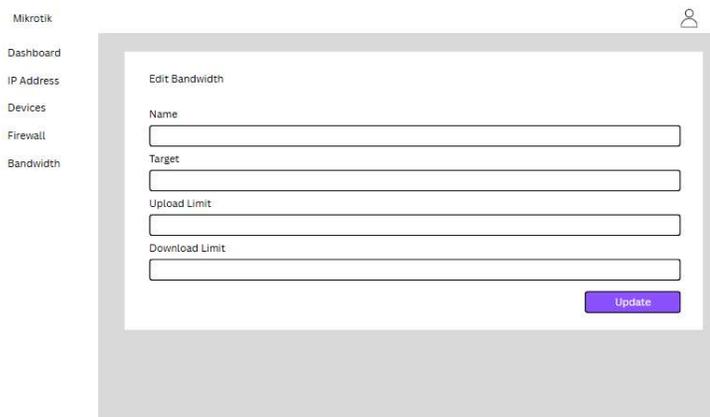
Chain

Action

Update

Gambar 3.12 Mockup Halaman Edit Item Firewall aplikasi web yang dikembangkan

Gambar 3.13 di bawah ini merupakan gambaran pada halaman *edit bandwidth*. Halaman ini akan muncul setelah admin klik tombol “*edit*” di sebelah kanan item.



Mikrotik

Dashboard

IP Address

Devices

Firewall

Bandwidth

Edit Bandwidth

Name

Target

Upload Limit

Download Limit

Update

Gambar 3.13 Mockup Halaman Edit Item Bandwidth aplikasi web yang dikembangkan

Pada penelitian ini menggunakan *framework* Laravel, lalu untuk penghubung antara perangkat Mikrotik dengan *website* yang dikembangkan adalah menggunakan RouterOS API. Pada sistem ini juga menggunakan *script* tertentu untuk meneruskan *log* yang sudah tercatat sebelumnya untuk dikirimkan ke bot Telegram.

Metode pengembangan sistem yang digunakan pada penelitian ini adalah metode Agile. Hal ini dikarenakan sistem yang dirancang memiliki beberapa komponen seperti RouterOS API, Laravel, dan bot Telegram yang dikembangkan secara bertahap. Dengan metode Agile, tim administrator dapat memperbaikinya dengan cepat ketika terjadi kendala ataupun perubahan pada sistem secara tidak terduga. Selain itu, dengan metode ini tim administrator dapat menyesuaikan dengan segera ketika mendapati adanya kebaruan pada versi sistem ataupun fitur yang ingin ditambahkan.

3.4. Pengujian Sistem (*Test the artifact*)

Setelah melakukan pengembangan pada sistem, dilakukan implementasi serta pengujian secara menyeluruh untuk memastikan agar fitur yang sudah diterapkan sebelumnya dapat berjalan dengan baik. Pengujian ini diantaranya:

1. Pengujian *Blackbox*, dengan instrumen pengujian berupa tabel yang berisikan fitur yang diuji, *test case*, dan hasil.
2. Pengujian laporan aktivitas pada bot telegram, dengan instrumen pengujian berupa tabel yang berisikan tentang skenario pengujian, hasil, dan contoh output dari bot telegram.
3. Pengujian pada sebuah jaringan, dengan instrumen pengujian berupa perangkat yang terhubung dengan *router*, lalu diuji hasil konfigurasi dari *firewall* dan *bandwidth* melalui *website* yang sudah dikembangkan.
4. Pengujian dengan perangkat Mikrotik yang berbeda, dengan instrumen pengujian berupa seri *router* Mikrotik yang berbeda (RB750r2, RB941-2nd, RB941-2nd-TC).

3.5. Evaluasi Hasil Pengujian (*Evaluate testing results*)

Setelah pengujian, tahapan berikutnya adalah melakukan evaluasi kinerja sistem dan mengumpulkan hasil dokumentasi pada saat proses pengembangan. Evaluasi ini dilakukan untuk menilai kelebihan, kekurangan, dan potensi yang dapat dikembangkan pada sistem lebih lanjut.

3.6. Menyampaikan Hasil Pengujian (*Communicate the testing results*)

Pada tahap akhir di pengujian ini, hasil dari pengujian yang sudah dilakukan sebelumnya dituliskan ke dalam sebuah laporan tugas akhir skripsi.