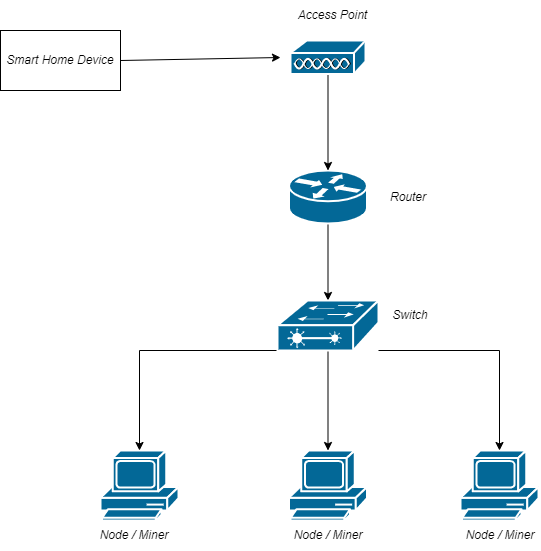
**BAB III**

**METODE PENELITIAN**

1. **Alur Pengerjaan Tugas Akhir**
2. **Uraian Metodelogi** 
   * 1. **Studi Literatur**
     2. **Topologi Jaringan**

Pada penelitian ini menggunakan topologi jaringan, yaitu terdapat *access point* untuk mejadi jaringan local antara perangkat *smart home* dengan PC (*node* / *miner*) serta terdapat 3 *node* yang dapat berperan sebagai *miner* dan saling terhubung antara *node* yang satu dengan *node* yang lain. Perangkat *smart home* yang akan disimulasikan dalam bentuk program *Python* yang akan mengirimkan data kepada PC yang berperan sebagai miner. Topologi jaringan yang dibuat dapat dilihat pada Gambar 3.1 berikut ini :



Gambar 3.1 Topologi Jaringan

PC yang berperan sebagai *node* dalam jaringan dan dapat juga berperan sebagai *miner*. *Miner* bertanggung jawab untuk melakukan pengecekan dan validasi terhadap aktivitas yang terjadi di dalam sistem, misalnya sebuah *smart home devices* yang hendak mengirimkan dan menyimpan data ke dalam sistem apakah diizinkan atau tidak dan apakah data yang dikirimkan merupakan data yang valid atau tidak. Penelitian ini menggunakan platform MultiChain yang akan dipasang pada semua node yang terdapat di dalam sistem.

* + 1. **Perangkat *Smart Home***

Pada penelitian ini dilakukan pembuatan perangkat *smart home* berbasis program yang dimana pembuatan program menggunakan Bahasa *python*. *Smart Home* berbasis program ini akan dijalankan dengan tujuan untuk dapat mengirimkan data ke *node admin* yang kemudian akan diproses

* + 1. ***Node Admin***

Berdasarkan Gambar.3.1 dari tiga *node* akan dipilih satu *node* yang akan berperan sebagai *node admin* dan dua *node* lainnya akan berperan sebagai *node client*. *Node admin* akan berperan untuk menginisiasi *blockchain*, seperti memberi izin *node* yang diperbolehkan untuk terhubung ke dalam *blockchain*, memberi izin *node* untuk melakukan *mining* ke dalam *blockchain*, menjalankan sebuah program yang berfungsi sebagai penguhubung antara *blockchain* pada MultiChain dengan perangkat *Smart Home* berbasis program yang akan mengirimkan paket data kemudian meneruskannya ke dalam *blockchain*. Untuk dapat menerima paket yang dikirim dari perangkat *smart home* perlu dibuat jalur komunikasi dengan menggunakan metode *socket programming* yang berfungsi sebagai penghubung antara *node admin* dengan perangkat *smart home*

* + 1. **Rancangan Pengiriman Data**

Data atau pesan dikirimkan dari perangkat *smart home* yang telah dihubungkan dengan *node admin*. Perangkat *smart home* yang disimulasikan dengan menggunakan program Python mengirimkan data kepada *node admin*. Setelah terhubung, perangkat *smart home* dapat mengirim data ke *node* Multichain menggunakan *socket*. Data akan diserialisasi menjadi format yang sesuai .JSON atau format khusus, lalu kirim data melalui koneksi socket yang sudah dibuat. Dengan menggunakan perintah send() untuk mengirim data melalui *socket*. Data yang dikirimkan berbentuk kalimat acak atau random string dengan dua jenis besaran data yaitu 32 bytes dan 128 bytes. Untuk mengirim pesan atau data dengan besaran 32 bytes maka pengguna harus mengetikkan “iot32” pada input field program dan menekan tombol enter, sedangkan untuk mengirim data atau pesan dengan ukuran 128 bytes maka pengguna harus mengetikkan “iot128” pada input field program. Data akan dikirimkan sebanyak 30 kali dengan waktu jeda tiap pengiriman data adalah 25 detik karena harus menunggu proses mining dari blockchain terlebih dahulu.

* + 1. **Rancangan Penerimaan Data**

Data yang telah dikirim oleh perangkat *smart home* kemudian akan diterima oleh sebuah program yang dijalankan pada *node admin*. Program ini memiliki fungsi untuk menerima data yang dikirim oleh perangkat *smart home* kemudian meneruskannya kepada platform MultiChain.

* + 1. **Koneksi dengan MultiChain**

Setelah pesan atau data diterima oleh node admin, maka tahap selanjutnya adalah meneruskannya ke dalam blockchain dengan menggunakan API yang telah disediakan oleh MultiChain sehingga pesan tesebut dapat disimpan ke dalam blockchain. Untuk dapat berkomunikasi dengan API milik MultiChain dengan menggunakan bahasa pemrogaman Python, maka dapat memanfaatkan library Savoir.

* + 1. **Penyimpanan Data**

Setelah terhubung dengan API milik MultiChain, maka untuk selanjutnya data dapat diterima dan disimpan ke dalam blockchain. Poses penyimpanan data dalam laporan ini untuk seterusnya disebut dengan proses create. Data yang telah diterima dan masih dalam format hexadecimal akan diteruskan ke dalam blockchain untuk disimpan ke dalam blok baru. MultiChain memiliki fitur stream yang berfungsi sebagai tempat untuk menyimpan data secara umum, dan istilah menyimpan data di dalam stream dikenal dengan istilah publish.

* + 1. **Validasi Data**

Ada dua jenis media yang akan mencoba mengirimkan data ke dalam blockchain yaitu permissioned device dan permissionless device. Permissioned devices adalah peralatan yang diizinkan oleh sistem untuk menyimpan dan melihat data. Sedangkan permissionless devices adalah peralatan yang berada di dalam maupun di luar sistem yang tidak memiliki izin apapun untuk melakukan aktivitas yang sama seperti permissioned devices.

1. **Persiapan *Hardware* dan *Software***
2. *Hardware*

Laptop HP Notebook 14-am013TU. Intel® Celeron® CPU N3060@ 1.60GHz with 8.0GB RAM

1. *Software*

*Software* yang digunakan pada penelitian ini sebagai berikut:

1. Visual Studio Code sebagai *text* *editor* yang membantu proses develop program simulasi peranti IoT.
2. MongoDB sebagai *web* *server* yang menyimpan *local database* pada *localhost* tanpa akses internet.
3. MulthiChain sebagai *tools* yang dapat memberikan akses ke *server* *blockchain*.
4. **Node *Admin***
5. **Peranti IoT**
6. **Rancangan Pengiriman Data**
7. **Rancangan Penerimaan Data**
8. **Koneksi ke MulthiChain API**
9. **Penyimpanan Data**
10. **Proses Validasi Data**