

BAB 3

METODOLOGI PENELITIAN

3.1 Telaah Literatur

Pada tahap telaah literatur, dilakukan pencarian dan pemahaman teori-teori dan metode-metode yang digunakan pada penelitian ini seperti protokol LEACH, berbagai macam serangan *denial-of-service* pada protokol LEACH, metode SMOTE, dan algoritma *Random Forest*. Pencarian dan pemahaman informasi ini didapatkan dari berbagai jurnal yang membahas hal-hal terkait dengan topik penelitian ini.

3.2 Pengumpulan Data

Adapun *dataset* yang digunakan pada penelitian ini didapat dari *dataset* yang dikembangkan oleh Almomani dkk. (2016). *Dataset* berisi 374.661 data lalu lintas sensor jaringan nirkabel dari sejumlah 100 buah sensor yang terbagi dalam 5 buah *cluster* dan terdiri dari 2 kategori yaitu serangan dan bukan serangan. Jumlah data lalu lintas jaringan yang merupakan serangan berjumlah 34.595, sedangkan jumlah data lalu lintas jaringan yang bukan merupakan serangan berjumlah 340.066. Adapun kategori serangan juga dapat dibagi menjadi 4 tipe serangan DoS yang dapat terjadi pada lapisan jaringan pada sistem jaringan sensor nirkabel.

Dataset yang digunakan terdiri dari lima belas fitur utama yang menjadi parameter untuk pelatihan dan pengujian aplikasi yang dirancang, yaitu status *Cluster Head*, jarak antar simpul dengan *Cluster Head*, jumlah pengiriman dan penerimaan pesan ADV, jumlah pengiriman dan penerimaan pesan *join*, jumlah

pengiriman dan penerimaan pesan TDMA *schedule*, urutan (*rank*) suatu simpul pada TDMA *schedule* tertentu, jumlah pengiriman dan penerimaan paket data, jumlah paket data yang dikirim ke *Base Station*, jarak antar *Cluster Head* dengan *Base Station*, kode pengiriman *cluster*, dan besar konsumsi energi setiap simpul.

Untuk memberi gambaran lebih terkait *dataset* yang digunakan pada penelitian ini, contoh bentuk *dataset* yang tersimpan dalam *file* .csv dapat dilihat pada Gambar 3.1 sampai dengan Gambar 3.3.

id	Time	Is_CH	who CH	Dist_To_CH	ADV_S	ADV_R	JOIN_S
102041	103	1	102041	0	1	3	0
103003	153	1	103003	0	1	1	0
108078	403	1	108078	0	1	3	0
111029	553	1	111029	0	1	5	0
112040	603	1	112040	0	1	5	0
114005	703	1	114005	0	1	3	0
114065	703	1	114065	0	1	3	0
115021	753	1	115021	0	1	5	0
117061	853	1	117061	0	1	9	0
201021	1003	1	201021	0	1	7	0
201061	1003	1	201061	0	1	7	0
101021	53	1	101021	0	1	0	0
101041	53	1	101041	0	1	0	0
102040	103	1	102040	0	1	2	0
103029	153	1	103029	0	1	3	0
103078	153	1	103078	0	1	3	0
105005	253	1	105005	0	1	2	0
105040	253	1	105040	0	1	2	0
106003	303	1	106003	0	1	5	0
106005	303	1	106005	0	1	5	0
106041	303	1	106041	0	1	5	0
106058	303	1	106058	0	1	5	0
107003	353	1	107003	0	1	5	0
107005	353	1	107005	0	1	5	0
107041	353	1	107041	0	1	5	0
107058	353	1	107058	0	1	5	0
107061	353	1	107061	0	1	5	0
108003	403	1	108003	0	1	0	0

Gambar 3.1 Potongan *Dataset* Pertama

JOIN_R	SCH_S	SCH_R	Rank	DATA_S	DATA_R	Data_Sent_To_BS
93	1	0	0	0	1302	0
72	1	0	0	0	1296	0
73	1	0	0	0	1314	0
36	1	0	0	0	1260	0
48	1	0	0	0	1296	0
38	1	0	0	0	1254	0
40	1	0	0	0	1280	0
59	1	0	0	0	1298	0
20	1	0	0	0	1131	0
20	1	0	0	0	1140	0
31	1	0	0	0	1240	0
17	1	0	0	0	1105	0
34	1	0	0	0	1258	0
31	1	0	0	0	1240	0
10	1	0	0	0	960	0
45	1	0	0	0	1260	0
10	1	0	0	0	927	0
31	1	0	0	0	1155	0
19	1	0	0	0	1140	0
15	1	0	0	0	1080	0
14	1	0	0	0	990	0
5	1	0	0	0	720	0
9	1	0	0	0	927	0
16	1	0	0	0	1088	0
13	1	0	0	0	996	0
13	1	0	0	0	1040	0
4	1	0	0	0	640	0
0	0	0	0	0	1	0

Gambar 3.2 Potongan *Dataset* Kedua

dist_CH_To_BS	send_code	Consumed Energy	Attack type
0	0	0.00726	Blackhole
0	0	0.00745	Blackhole
0	0	0.00725	Blackhole
0	0	0.00722	Blackhole
0	0	0.00726	Blackhole
0	0	0.00732	Blackhole
0	0	0.0073	Blackhole
0	0	0.00721	Blackhole
0	0	0.00728	Blackhole
0	0	0.0072	Blackhole
0	0	0.00719	Blackhole
0	0	0.00225	Blackhole
0	0	0.00225	Blackhole
0	0	0.00728	Blackhole
0	0	0.00724	Blackhole
0	0	0.00722	Blackhole
0	0	0.05273	Blackhole
0	0	1.82509	Blackhole
0	0	0.00734	Blackhole
0	0	2.26538	Blackhole
0	0	0.27124	Blackhole
0	0	0.18777	Blackhole
0	0	2.09484	Blackhole
0	0	2.19525	Blackhole
0	0	2.29338	Blackhole
0	0	2.21863	Blackhole
0	0	0.00685	Blackhole
0	0	2.20725	Blackhole

Gambar 3.3 Potongan *Dataset* Ketiga

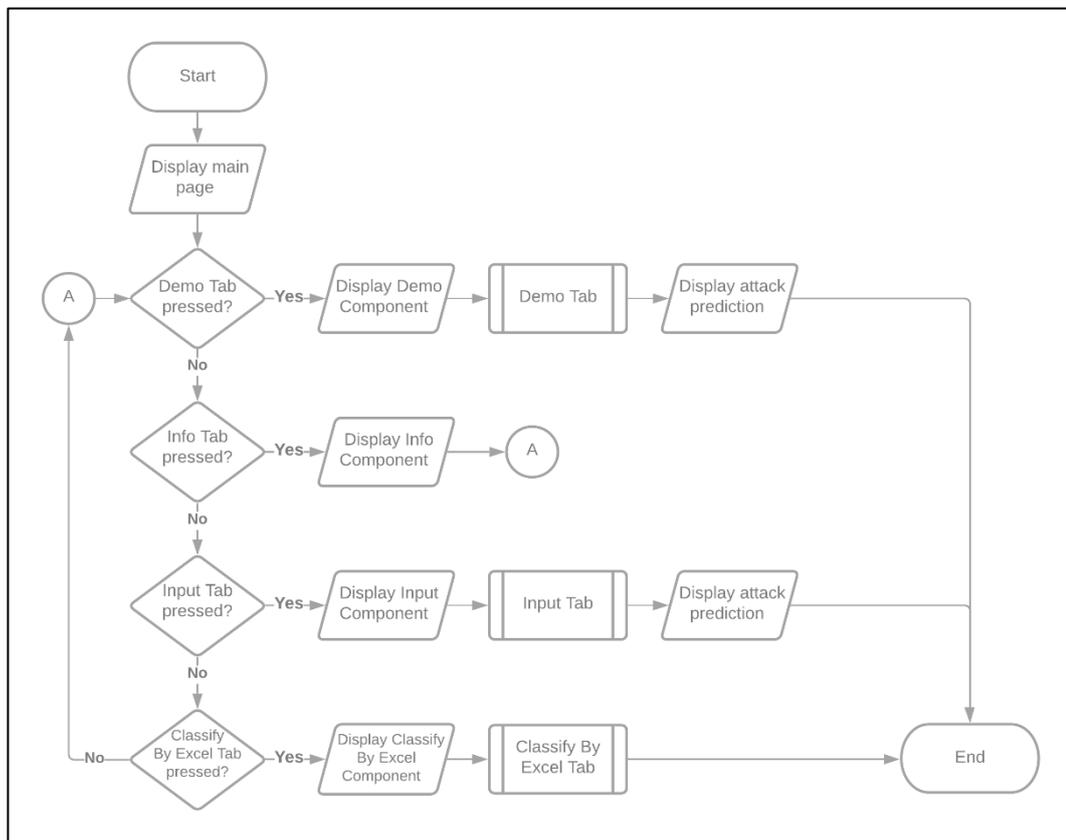
3.3 Perancangan Aplikasi

Perancangan aplikasi terdiri dari *flowchart* aplikasi dan algoritma, serta rancangan antarmuka aplikasi yang digunakan pada penelitian ini. *Flowchart* dibagi menjadi dua bagian, yaitu *flowchart* aplikasi *web* yang dibangun untuk mengimplementasikan penelitian dan *flowchart* pembuatan model pembelajaran

mesin yang didapatkan dari implementasi metode SMOTE dan algoritma Random Forest pada data latih penelitian.

3.3.1 Flowchart Aplikasi Web

Gambar 3.4 merupakan *flowchart* utama dari aplikasi *web* yang dibangun. Pada halaman utama aplikasi *web*, diperlihatkan informasi singkat terkait penjelasan Jaringan Sensor Nirkabel (JSN), protokol pada JSN, dan serangan-serangan *denial-of-service* yang dapat terjadi pada JSN.

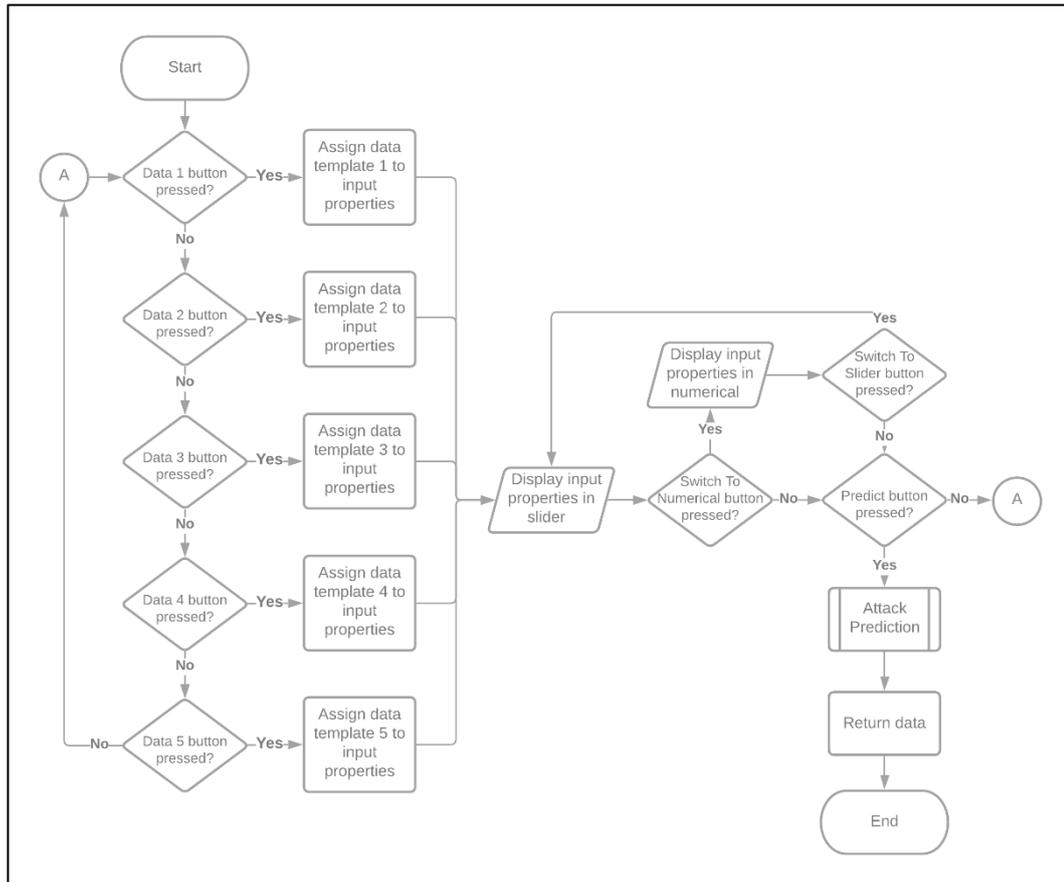


Gambar 3.4 *Flowchart* Utama Aplikasi Web

Pada aplikasi *web* ini, bagian-bagian lain dibuat dalam bentuk *tabs*. Terdapat empat buah *tabs* pada aplikasi *web* ini yang dapat digunakan, yaitu *Demo*

Tab, *Info Tab*, *Input Tab*, dan *Classify By Excel Tab*. *Demo Tab* berfungsi untuk memberikan contoh penggunaan aplikasi *web* dengan menyediakan contoh masukan data sebuah simpul pada JSON dan hasil prediksi serangan yang terjadi pada simpul tersebut. *Info Tab* memberikan informasi lebih lengkap terkait cara penggunaan aplikasi *web*, tata cara memasukkan data, dan penjelasan terkait makna dari masing-masing data yang dapat dimasukkan. *Input Tab* berfungsi untuk melakukan prediksi terhadap data simpul yang dimasukkan. Pada *tab* ini, pengguna dapat memasukkan data sebuah simpul JSON dan mendapatkan prediksi serangan yang terjadi pada simpul tersebut. *Classify By Excel Tab* berfungsi untuk menggunakan fitur prediksi serangan dengan menggunakan pengunggahan *file* Excel.

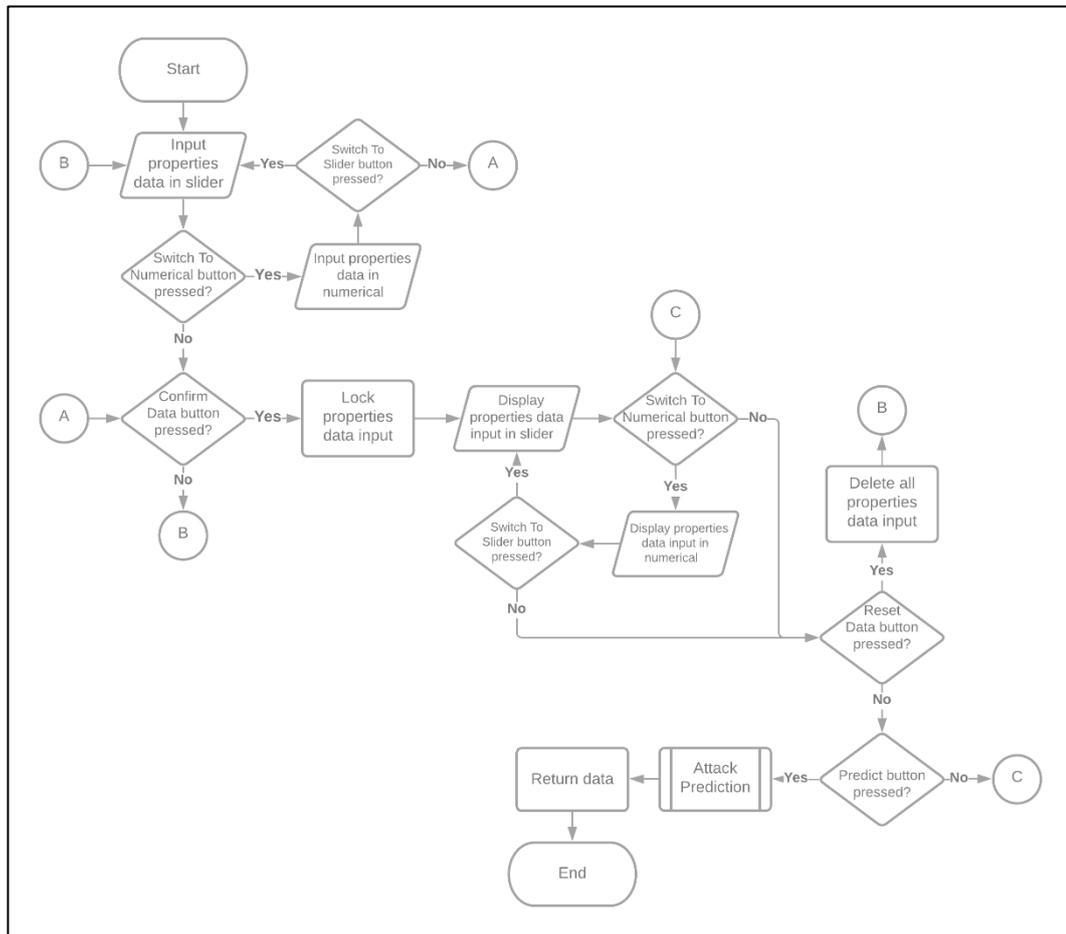
Gambar 3.5 merupakan *flowchart* dari *Demo Tab*. Pada *tab* ini, pengguna dapat memilih satu dari lima contoh data yang telah disediakan. Setiap contoh data yang telah disediakan mewakili setiap hasil prediksi, yaitu serangan *Blackhole*, *Grayhole*, *Flooding*, TDMA, dan bukan serangan. Contoh data dapat disajikan dalam dua bentuk, yaitu dalam bentuk numerik dan dalam bentuk *slider*. Setelah pengguna memilih contoh data, pengguna dapat menekan tombol “Prediksi” yang berfungsi untuk memproses data dan menghasilkan keluaran prediksi atas data tersebut.



Gambar 3.5 *Flowchart* Demo Tab

Gambar 3.6 merupakan *flowchart* dari *Input Tab*. Pada *tab* ini, pengguna dapat memasukkan data sebuah simpul JSON secara manual. Pengguna juga dapat memilih antara bentuk numerik dan bentuk *slider* dalam melakukan pemasukkan data simpul JSON. Setelah pengguna memasukkan data, pengguna dapat menekan tombol “Confirm” untuk mengunci data dan menyatakan bahwa data telah dimasukkan secara benar dan siap untuk diprediksi, atau menekan tombol “Reset” untuk menghapus seluruh data yang telah dimasukkan dan mengulang proses pemasukkan data simpul JSON. Jika pengguna menekan tombol “Confirm”, tombol

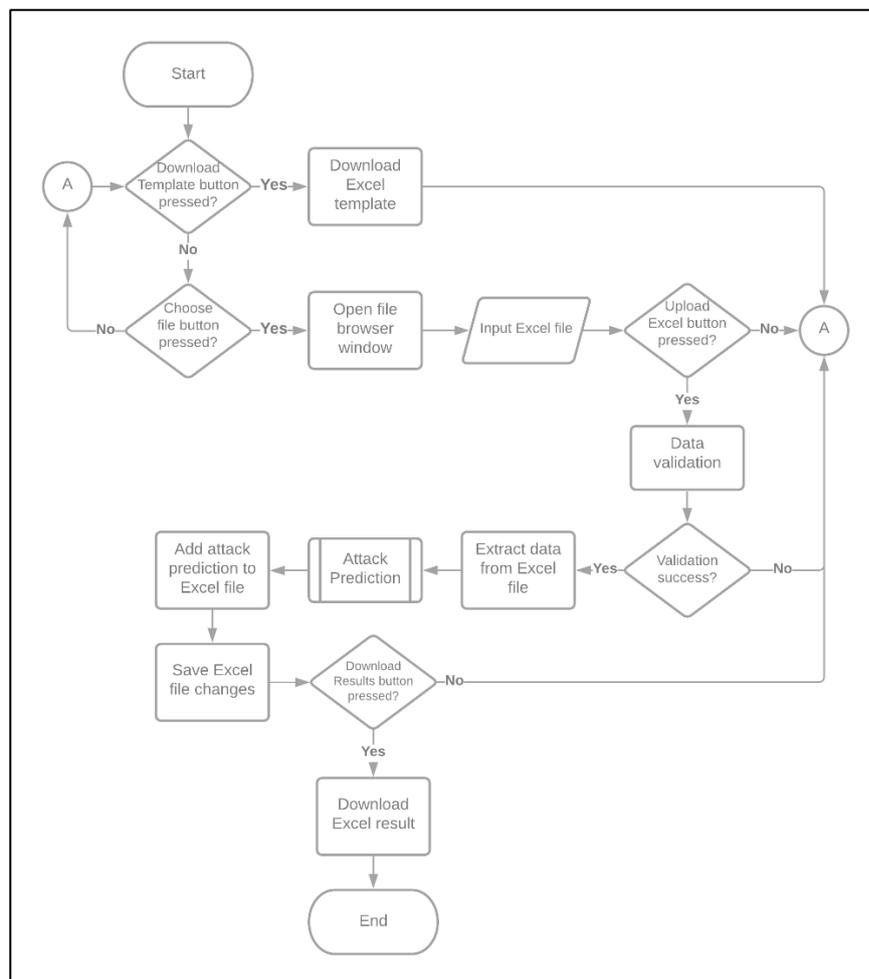
“Prediksi” akan dapat ditekan. Jika tombol “Prediksi” ditekan, maka data akan diproses dan menghasilkan keluaran prediksi atas data tersebut.



Gambar 3.6 Flowchart Input Tab

Gambar 3.7 merupakan flowchart dari *Classify By Excel Tab*. Pada tab ini, pengguna dapat mengunduh *file template* Excel yang tersedia, lalu mengisi template tersebut dengan lebih dari satu data simpul yang ingin diprediksi. Template tersebut kemudian dapat diunggah kembali ke aplikasi *web* untuk diproses.

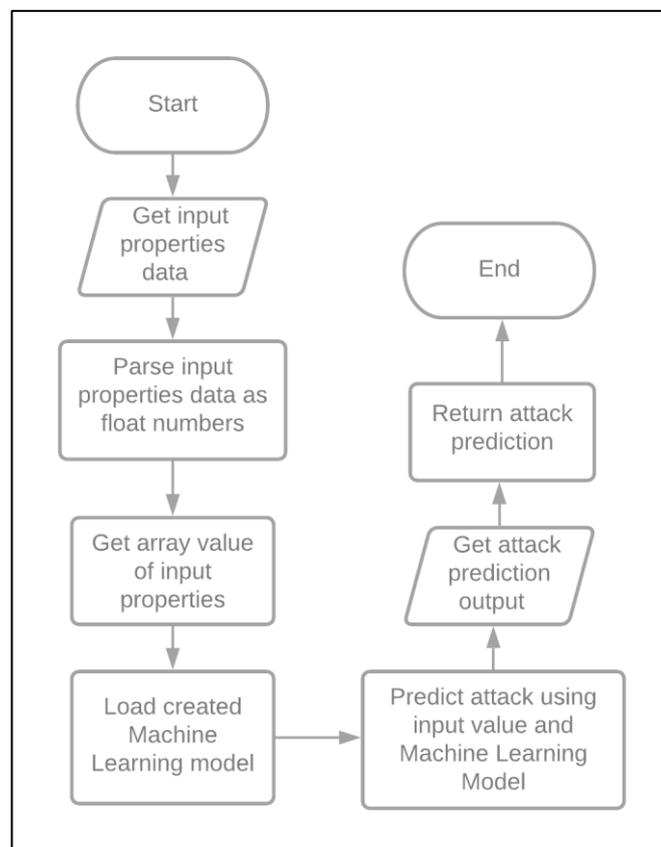
Pemrosesan *file* Excel meliputi validasi data yang telah dimasukkan pengguna dan ekstraksi data dari setiap baris *file* Excel untuk diprediksi. Setelah proses prediksi selesai, hasil prediksi masing-masing data akan kembali dituliskan pada setiap baris data *file* Excel tersebut pada kolom yang telah disediakan. Kemudian pengguna dapat mengunduh kembali *file* tersebut dan melihat hasil yang telah ditambahkan dengan prediksi tipe serangan yang terjadi untuk masing-masing data simpel yang dituliskan. Fitur *Classify By Excel* ini berfungsi agar pengguna dapat melakukan prediksi banyak data simpel JSN secara mudah dan cepat.



Gambar 3.7 Flowchart Classify By Excel Tab

Gambar 3.8 merupakan flowchart dari proses prediksi serangan yang dilakukan oleh aplikasi web. Setelah aplikasi web mendapatkan data masukan dari pengguna, data tersebut akan diurai sebagai tipe data float dan diubah menjadi bentuk array. Hal ini perlu dilakukan agar data masukan siap untuk diprediksi dengan menggunakan model pembelajaran mesin yang telah dibuat. Perancangan pembuatan model pembelajaran mesin dijelaskan pada bagian setelah ini.

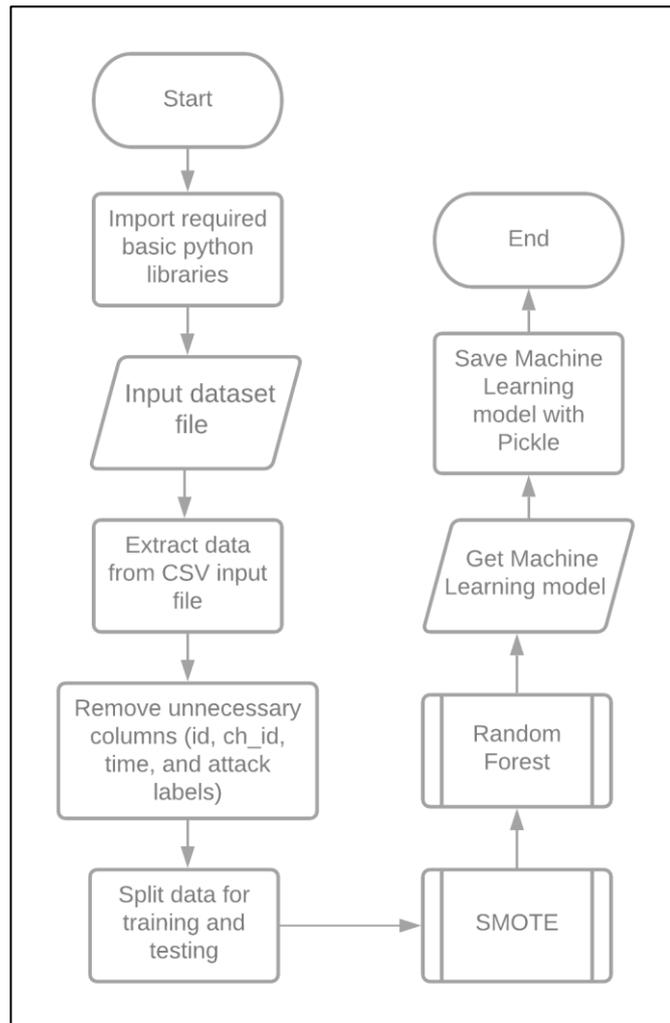
Setelah data masukan telah siap, model pembelajaran mesin akan dimuat ke dalam aplikasi web. Data masukan tersebut pun dapat digunakan sebagai masukan pada model pembelajaran mesin, dan hasil prediksi serangan dapat dikeluarkan. Hasil prediksi tersebut kemudian akan dikembalikan ke tampilan web untuk diperlihatkan kepada pengguna.



Gambar 3.8 *Flowchart* Attack Prediction

3.3.2 Flowchart Pembuatan Model Pembelajaran Mesin

Gambar 3.9 merupakan *flowchart* utama dari pembuatan model pembelajaran mesin. Teknik utama dalam pembuatan model pembelajaran mesin adalah mengimplementasikan metode SMOTE untuk melakukan *oversampling* pada data latih dan algoritma Random Forest untuk melakukan pelatihan data dan klasifikasi serangan.

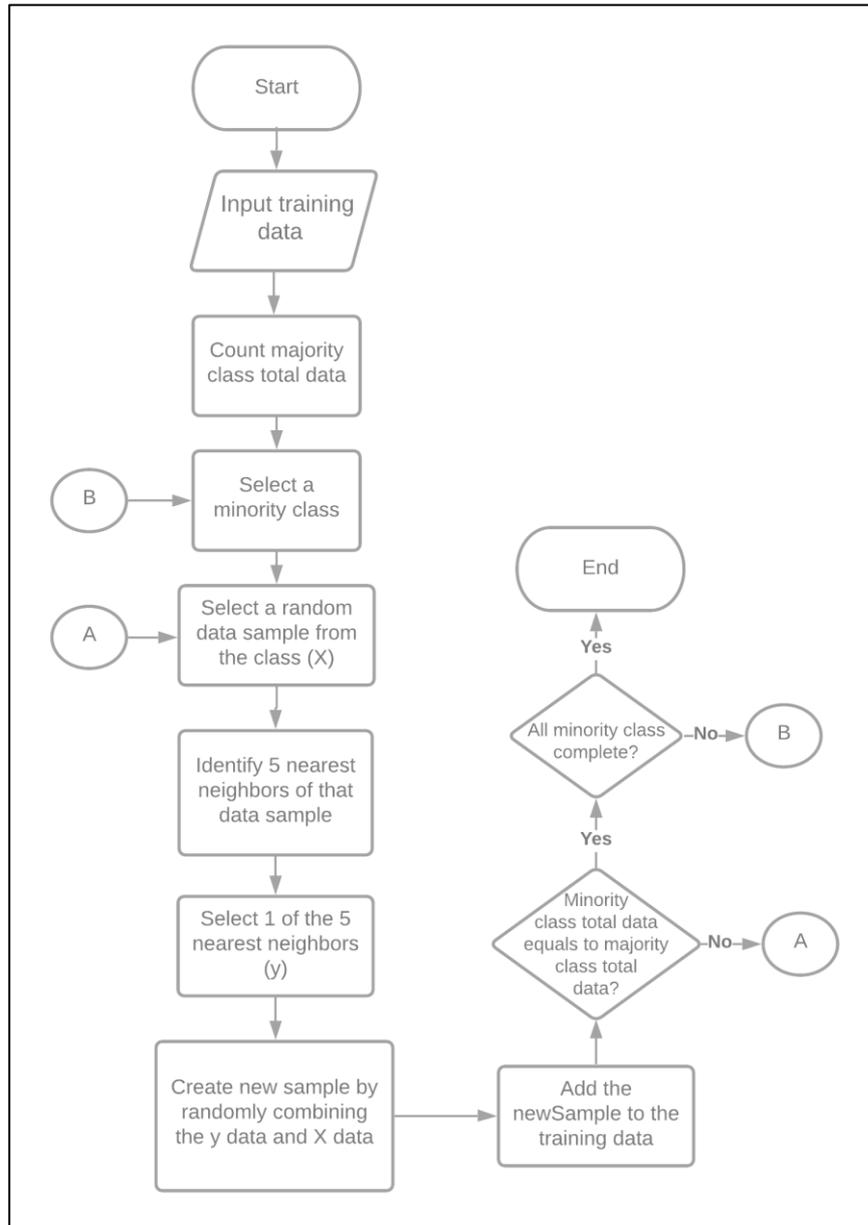


Gambar 3.9 *Flowchart* Utama Pembuatan Model Pembelajaran Mesin

Hal pertama yang perlu dilakukan sebelum melakukan proses pembuatan model lebih lanjut adalah melakukan *import library* dasar yang diperlukan untuk mendukung proses pembuatan model pembelajaran mesin. Setelah itu, dilakukan *import dataset* yang digunakan untuk proses pelatihan dan pengujian model. Data yang ada pada file *.csv dataset* kemudian diekstraksi ke dalam variabel-variabel agar bisa diolah lebih lanjut. Setelah data selesai diekstraksi, dilakukan praproses data. Praproses data dilakukan dengan menghilangkan kolom data yang tidak dibutuhkan seperti data *identifier* dan label-label serangan.

Praproses data dilanjutkan dengan pembagian data tersebut ke dalam kategori data latih dan data uji dengan rasio yang telah ditentukan sebelumnya. Data latih digunakan untuk melatih model pembelajaran mesin, sedangkan data uji digunakan untuk mengevaluasi model pembelajaran mesin untuk memastikan kualitas model tersebut. Setelah pembagian data selesai dilakukan, data latih digunakan pada metode SMOTE dan algoritma Random Forest untuk membangun model pembelajaran mesin, sebelum akhirnya disimpan pada mesin.

Gambar 3.10 merupakan *flowchart* dari metode SMOTE yang digunakan untuk melakukan proses *oversampling* pada data latih. Teknik *oversampling* yang diimplementasikan berfungsi untuk memperbanyak data latih pada kelas data minoritas agar data latih menjadi seimbang. Jumlah data pada kelas minoritas diperbanyak sehingga serupa dengan jumlah data pada kelas mayoritas. Metode SMOTE memanfaatkan algoritma *K-Nearest Neighbors* (KNN) untuk menentukan data baru yang ditambahkan pada dataset.



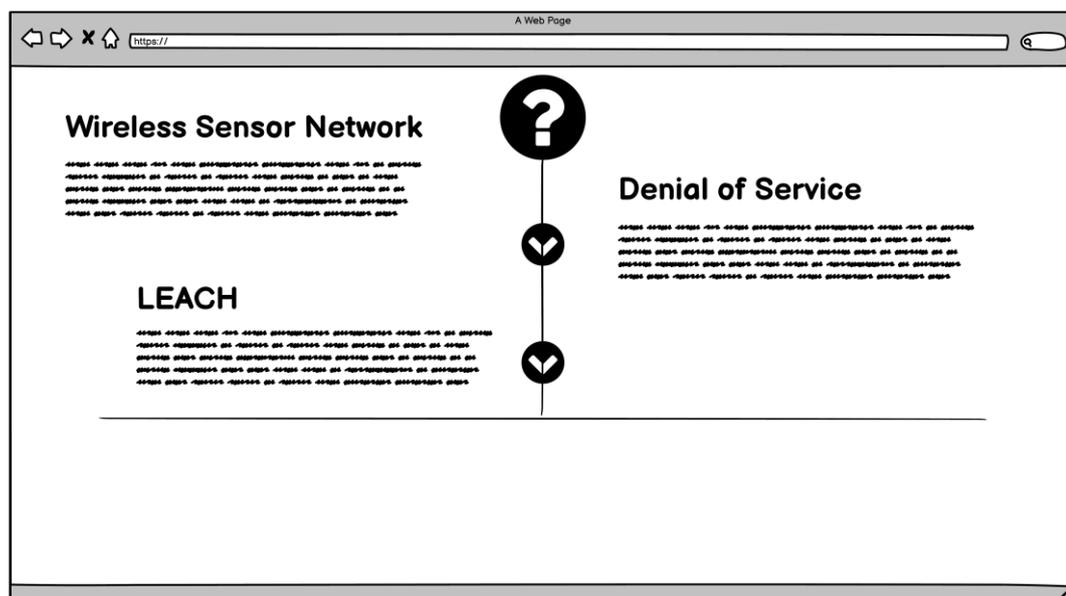
Gambar 3.10 Flowchart SMOTE

Metode SMOTE yang diimplementasikan dimulai dari menghitung jumlah data pada kelas mayoritas atau kelas yang paling banyak datanya. Jumlah ini akan menjadi batas pembuatan data baru pada kelas-kelas minoritas. Setelah itu, sampel data acak dipilih dari data kelas minoritas tersebut. Kemudian, sejumlah lima data *nearest neighbors* dari data terpilih tersebut ditentukan dengan menggunakan

algoritma *K-Nearest Neighbors*. Dari lima data *nearest neighbors* tersebut, dipilih kembali satu secara acak untuk digabungkan ke sampel data acak yang sebelumnya telah dipilih. Penggabungan dilakukan dengan cara menggabungkan data pada kolom-kolom yang juga dipilih secara acak antara kedua data tersebut. Setelah proses penggabungan selesai, maka data hasil penggabungan tersebut merupakan data baru yang dapat ditambahkan ke dalam data latih. Proses ini kemudian dilakukan berulang kali hingga jumlah data setiap kelas minoritas serupa dengan jumlah data kelas mayoritas.

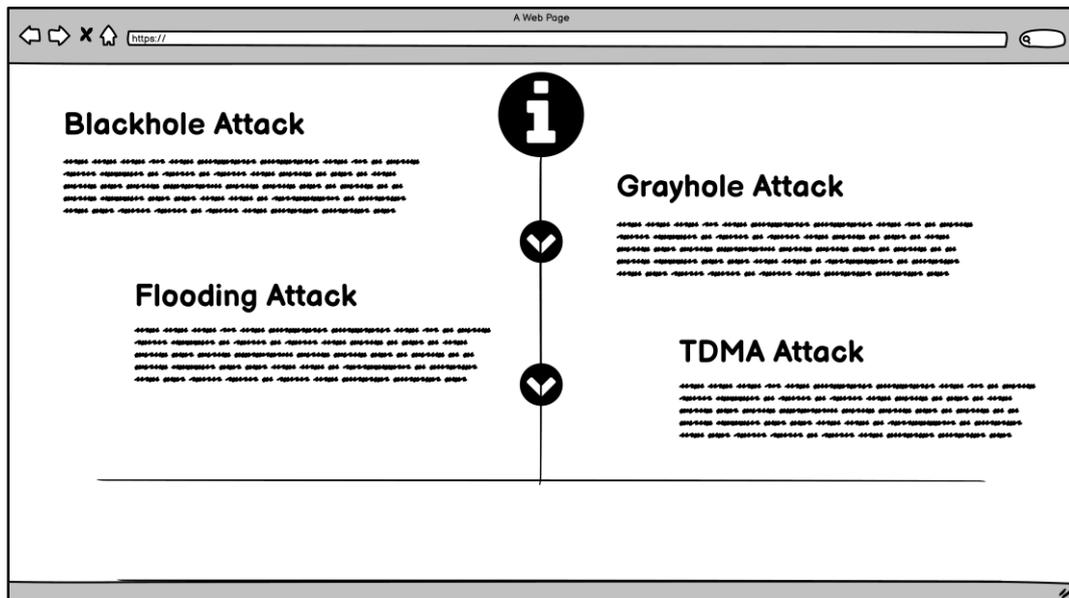
3.3.3 Rancangan Tampilan Antarmuka

Gambar 3.11 merupakan rancangan tampilan antarmuka halaman awal pada aplikasi *web* yang menampilkan informasi umum mengenai Jaringan Sensor Nirkabel, serangan *Denial-of-Service*, dan protokol LEACH pada Jaringan Sensor Nirkabel. Informasi disajikan dalam bentuk paragraf-paragraf.



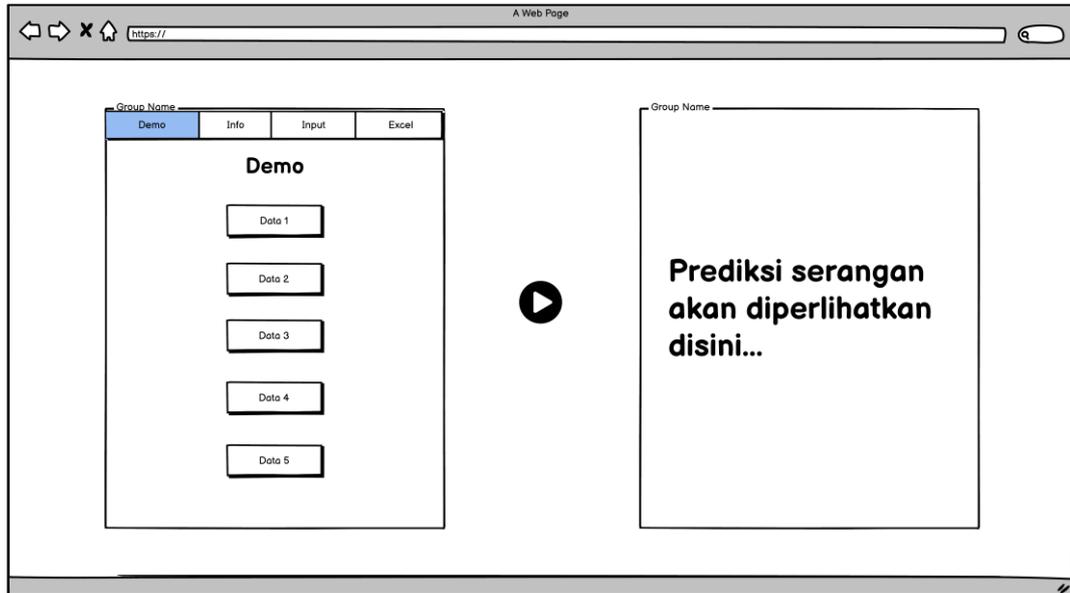
Gambar 3.11 Halaman Awal (Informasi Umum)

Gambar 3.12 merupakan rancangan tampilan antarmuka halaman awal pada aplikasi *web* yang menampilkan informasi terkait jenis-jenis serangan *Denial-of-Service* yang dapat terjadi pada Jaringan Sensor Nirkabel dengan protokol LEACH. Informasi disajikan dalam bentuk paragraf-paragraf.

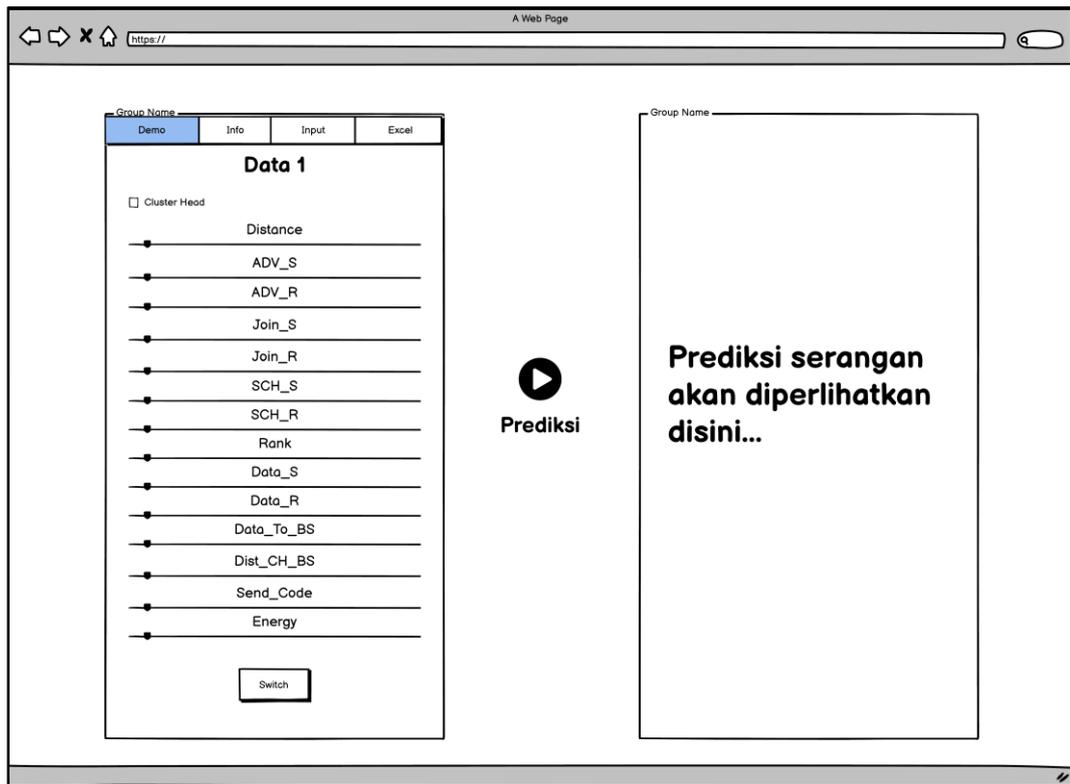


Gambar 3.12 Halaman Awal (Informasi Serangan)

Gambar 3.13 merupakan rancangan tampilan antarmuka *tab* demo pada aplikasi *web*. Pada *tab* demo, setiap pilihan contoh data demo direpresentasikan dengan tombol yang dapat ditekan. Jika pengguna menekan salah satu tombol contoh data, maka aplikasi *web* akan langsung menampilkan contoh data yang dipilih dalam bentuk *slider* ataupun numerik. Rancangan tampilan antarmuka setelah pengguna menekan tombol contoh data dapat dilihat pada Gambar 3.14. Pengguna dapat merubah bentuk data yang disajikan dengan menekan tombol “Switch”.

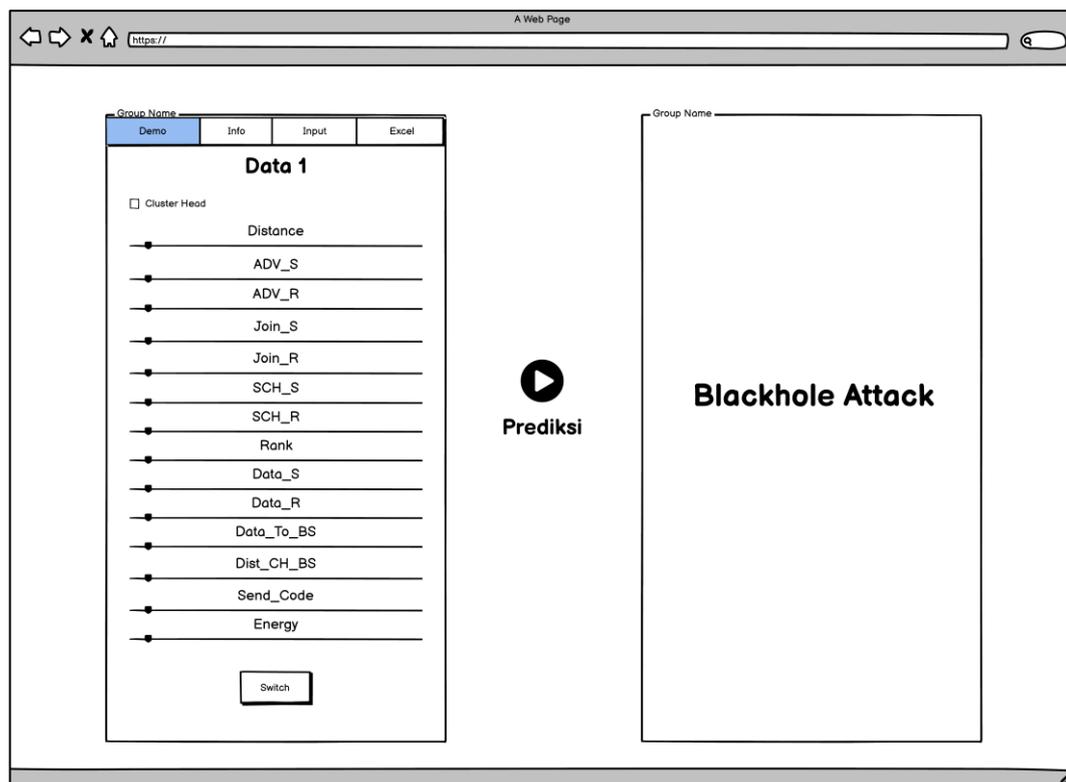


Gambar 3.13 *Tab Demo*



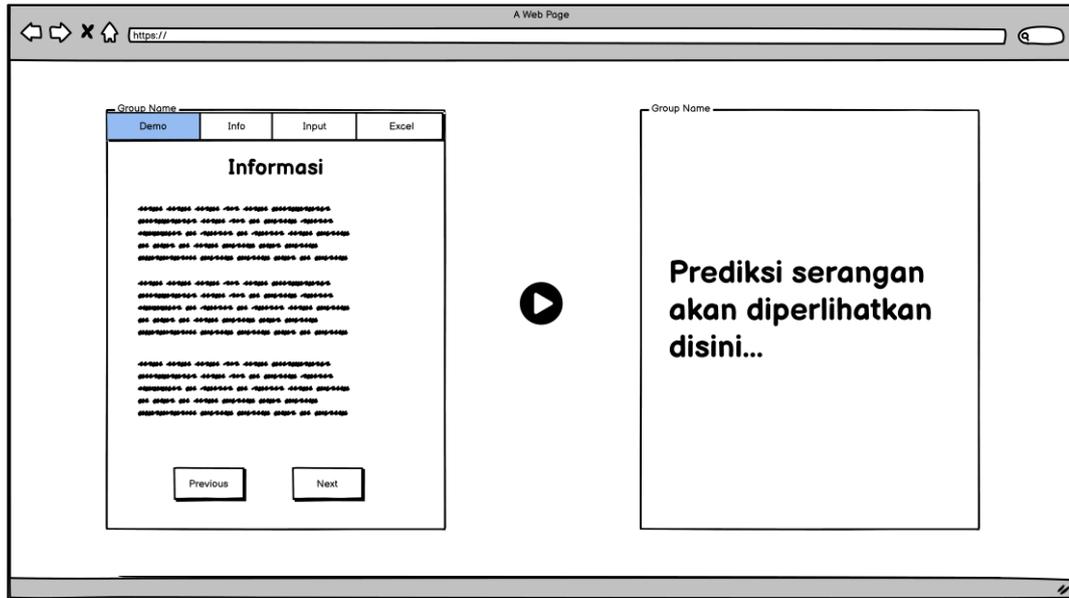
Gambar 3.14 *Tab Demo (Contoh Data)*

Gambar 3.15 merupakan rancangan tampilan antarmuka *tab* demo jika pengguna telah menekan tombol “Prediksi”. Setelah pengguna menekan tombol ”Prediksi”, maka aplikasi *web* akan memproses data yang telah dipilih pengguna dan menampilkan prediksi serangan berdasarkan data tersebut.



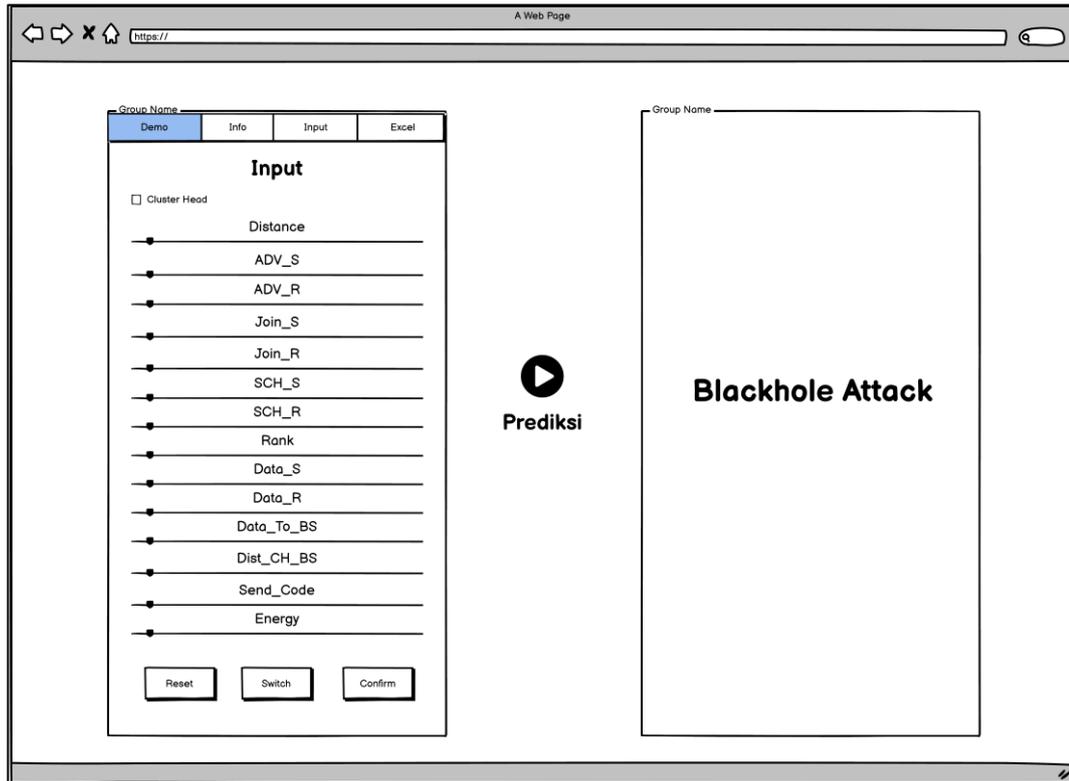
Gambar 3.15 *Tab* Demo (Hasil Prediksi)

Gambar 3.16 merupakan rancangan tampilan antarmuka *tab* informasi. Pada *tab* ini pengguna dapat mengetahui informasi lebih lengkap terkait cara penggunaan aplikasi *web*, tata cara memasukkan data, dan penjelasan terkait makna dari masing-masing data yang dapat dimasukkan. Tombol “Next” dan “Previous” dapat digunakan untuk bernavigasi antar beberapa halaman informasi yang terdapat pada *tab* ini.



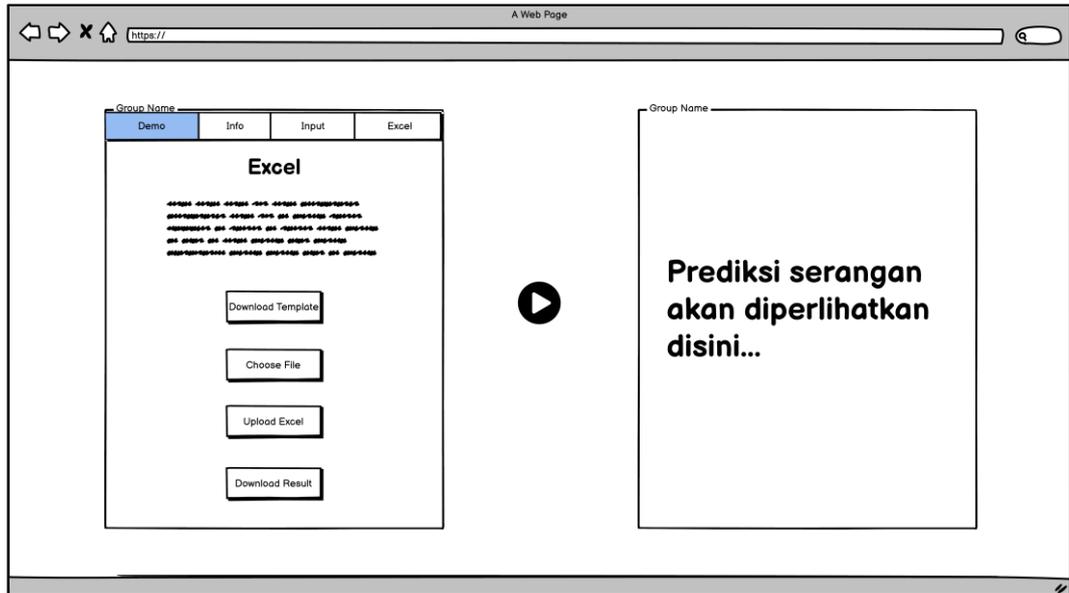
Gambar 3.16 *Tab Informasi*

Gambar 3.17 merupakan rancangan tampilan antarmuka *tab* input. Pada *tab* ini, pengguna dapat memasukkan data sebuah simpul Jaringan Sensor Nirkabel kemudian memproses data tersebut untuk mengetahui prediksi serangan yang terjadi pada simpul tersebut. Pengguna dapat memasukkan data dengan cara mengetikkan nilai-nilai atribut pada kolom input yang telah disediakan jika bentuk data yang dipilih adalah numerik, ataupun dengan cara menggeser *slider* jika bentuk data adalah *slider*. Pengguna dapat mengubah bentuk data input dengan cara menekan tombol “Switch”. Setelah pengguna memasukkan data, pengguna dapat menekan tombol “Confirm” untuk mengunci data dan menyatakan bahwa data telah dimasukkan secara benar dan siap untuk diprediksi, atau menekan tombol “Reset” untuk menghapus seluruh data yang telah dimasukkan dan mengulang proses pemasukkan data simpul JSN. Jika pengguna menekan tombol “Confirm”, tombol “Prediksi” akan dapat ditekan. Jika tombol “Prediksi” ditekan, maka data akan diproses dan menghasilkan keluaran prediksi atas data tersebut.



Gambar 3.17 Tab Input

Gambar 3.18 merupakan rancangan tampilan antarmuka *tab classify by excel*. Tab ini berfungsi agar pengguna dapat menggunakan fitur *Classify by Excel*. Tombol “Download Template” berfungsi untuk melakukan pengunduhan *file template*. Tombol “Choose File” berfungsi untuk memilih *file* yang akan diunggah ke aplikasi *web*. Tombol “Upload File” berfungsi untuk mengunggah *file* yang telah dipilih dan memulai proses *classify by excel*. Setelah proses selesai, tombol “Download Result” akan muncul dan dapat digunakan untuk mengunduh hasil klasifikasi yang telah dibuat.



Gambar 3.18 Tab Classify by Excel

3.4 Implementasi

Pada tahap implementasi, dilakukan penulisan *source code* implementasi metode SMOTE, implementasi algoritma *Random Forest*, dan aplikasi berbasis web. Metode SMOTE digunakan pada data latih awal yang berasal dari *dataset* untuk menghasilkan data latih baru yang seimbang. Algoritma *Random Forest* digunakan untuk melatih model pembelajaran mesin menggunakan data latih baru tersebut agar dapat melakukan klasifikasi yang lebih baik. Selain itu, *debugging* pada kode program juga dilakukan untuk mengetahui dan memperbaiki kesalahan dan kekurangan yang ada dalam aplikasi yang dibangun. Pembuatan model pembelajaran mesin dan pelatihan data dilakukan menggunakan bahasa pemrograman Python. Untuk membuat aplikasi berbasis web, digunakan bahasa pemrograman JavaScript dan Python yang ada pada framework React dan Flask.

Implementasi juga dilakukan dengan bantuan perangkat lunak dan perangkat keras. Spesifikasi dari perangkat lunak dan perangkat keras yang digunakan untuk perancangan dan pembangunan aplikasi *web* dan penelitian ini adalah sebagai berikut.

a. Perangkat Lunak

- Windows 10 Home 64-bit
- Jupyter Notebook
- Visual Studio Code

b. Perangkat Keras

- Manufaktur : Acer Inc.
- Model : Aspire 3 – A315
- Prosesor : AMD Ryzen™ 5 2500U with Vega⁸ Graphics
- RAM : 8 GB
- Penyimpanan : 1 TB HDD

3.5 Uji Coba dan Evaluasi

Pada tahap ini, dilakukan pengujian terhadap model pembelajaran mesin. Uji coba dilakukan dengan menggunakan data uji yang telah ditentukan sebelumnya. Data uji tersebut dimasukkan pada model pembelajaran mesin untuk mendapatkan keluaran berupa hasil klasifikasi. Hasil klasifikasi tersebut kemudian dievaluasi menggunakan berbagai macam parameter evaluasi performa, yaitu *True Positive Rate* (TPR), *True Negative Rate* (TNR), *False Positive Rate* (FPR), *False Negative Rate* (FNR), *accuracy*, dan *precision*.

3.6 Dokumentasi

Pada tahap dokumentasi, dilakukan konsultasi pengembangan penelitian dan penulisan laporan penelitian. Hal ini dilakukan agar penelitian dapat bergerak ke arah yang sesuai serta mendapatkan masukan dan saran yang berguna untuk pengembangan penelitian.