BAB 3
METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Metodologi penelitian ini merangkum langkah-langkah yang dilakukan dalam menyusun dan melaksanakan penelitian. Langkah-langkah tersebut mencakup studi literatur, pengumpulan dan analisis data, pengembangan model, pengujian, evaluasi model, hingga penulisan laporan. Setiap langkah dijelaskan secara rinci sebagai berikut.

3.1.1 Studi Literatur

Telaah literatur merupakan tahap pertama dalam penulisan laporan penelitian ini. Telaah literatur memiliki tujuan untuk mencari informasi yang berhubungan dengan topik penelitian yang dilakukan. Dari telaah literatur didapat beberapa informasi yang dibutuhkan dan dikumpulkan dengan cara membaca dan memahami tulisan maupun pembicaraan yang didapat dari berbagai sumber seperti jurnal, dan karya tulis ilmiah.

3.1.2 Pengumpulan dan Pengolahan Data

Pengumpulan dan pengolahan data dari dataset yang diperoleh dari situs UCI dan Open ML. Data yang dicari adalah data yang memiliki lebih dari 50 *feature*. Ada total delapan dataset yang digunakan dalam penelitian ini, yang berasal dari kategori dataset yang berbeda, yaitu *multivariate, univariate*, dan *sequential*. Nama-nama dataset yang digunakan meliputi AP_Breast_Omentum, Musk (Version 2), Internet Advertisements, Bioresponse, Arcene, AP Colon Kidney, Hill Valley, dan Nomao.

3.1.3 Perancangan Model

Dari dataset yang sudah didapatkan dilakukan *feature selection* menggunakan *information gain* lalu dengan nilai *threshold* median. Selanjutnya model klasifikasi *logistic regression* dibangun menggunakan *scikit-learn library*.

3.1.4 Pengujian dan Evaluasi Model

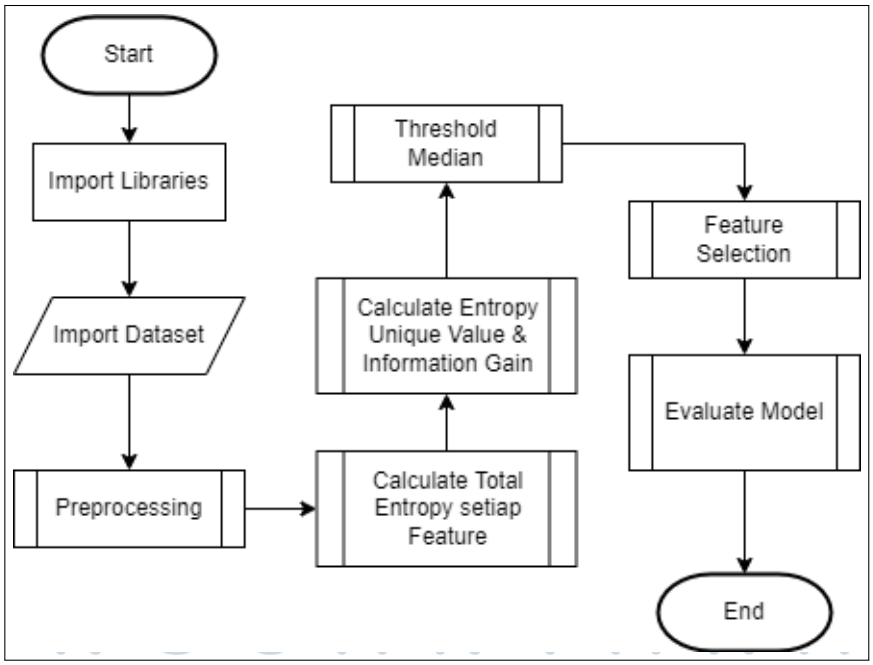
Pengujian dan evaluasi model dilakukan dengan bantuan *confusion matrix* untuk mendapatkan hasil perhitungan dari nilai *accuracy*, *precision*, *recall*,dan F1 *Score* dari model tersebut.

3.1.5 Penulisan Laporan

Penulisan laporan memiliki tujuan untuk mendokumentasikan penelitian, perancangan, serta pembuatan model, sehingga nantinya dapat memberikan informasi untuk penelitian serupa lainnya.

3.2 Perancangan Sistem

Pada penelitian ini dilakukan perancangan sebuah sistem yang digunakan untuk melakukan perhitungan nilai *information gain*, dan melakukan *feature selection* dengan menggunakan *threshold* berdasarkan nilai median. Pada tahap perancangan sistem, alur kerja dari sistem dibuat dalam bentuk *flowchart* seperti pada Gambar 3.1.



Gambar 3.1. Flowchart Utama

Pada Gambar 3.1, terdapat beberapa tahapan sistem yaitu *import libraries,*

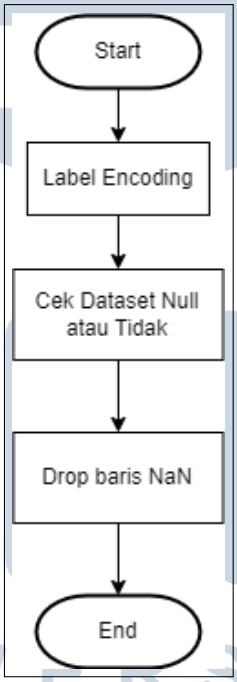
import dataset, preprocessing, calculate total *entropy* setiap *feature*, *calculate entropy unique value & information gain, threshold median, feature selection, evaluate model*.

3.2.1 Import Libraries

Pada tahap ini melakukan *import library* yang diperlukan untuk pembuatan model. *Library* yang dibutuhkan untuk proses *preprocessing*, penggunaan *k-fold cross validation*, pembuatan model klasifikasi, dan evaluasi model.

3.2.2 Preprocessing

Tahap *preprocessing* adalah tahapan untuk pengolahan dataset, agar dataset siap digunakan dalam pemrosesan *feature selection* dan pembuatan model.



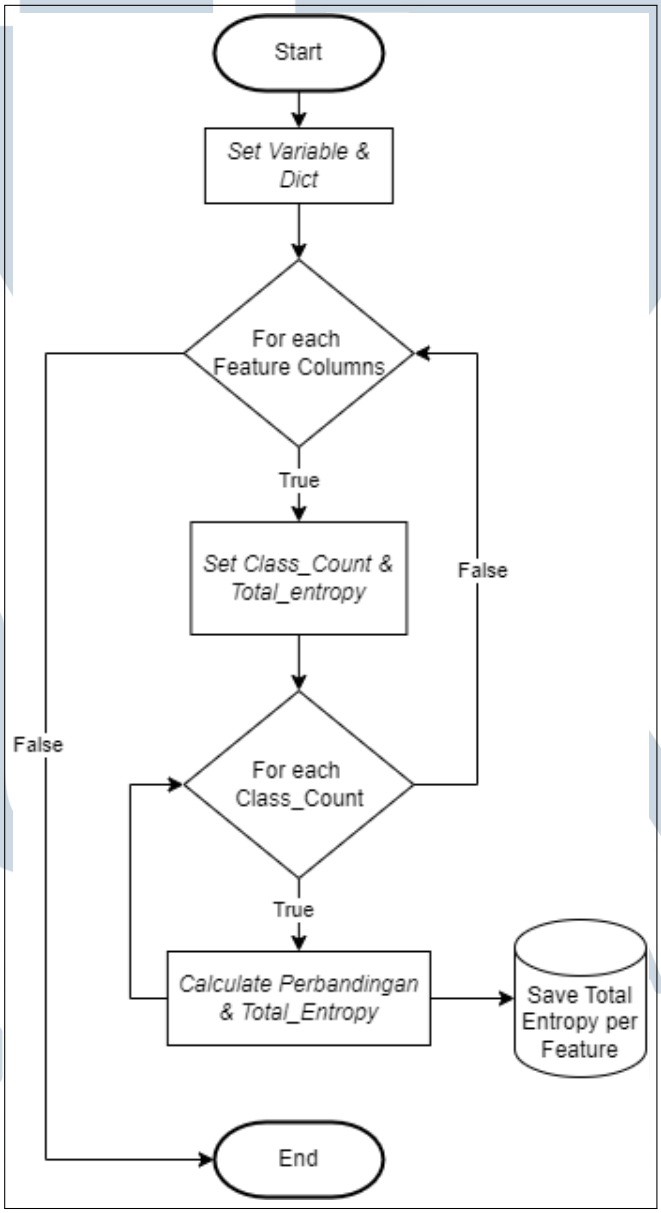
Gambar 3.2. *Preprocessing*

Proses pertama pada Gambar 3.2 adalah melakukan *label encoding* yang mengubah nilai *feature* yang kategorikal menjadi *numerical*. Setelah proses tersebut selesai, maka dilakukan pengecekan apakah terdapat data yang *null* atau NaN. Selanjutnya, apabila terdapat data yang *null* atau NaN maka baris tersebut dihapus.

3.2.3 Calculate Entropy & Information Gain

Pada tahap ini dilakukan perhitungan dari *information gain* setiap *feature* dataset. Langkah-langkah perhitungan *information gain*, yaitu

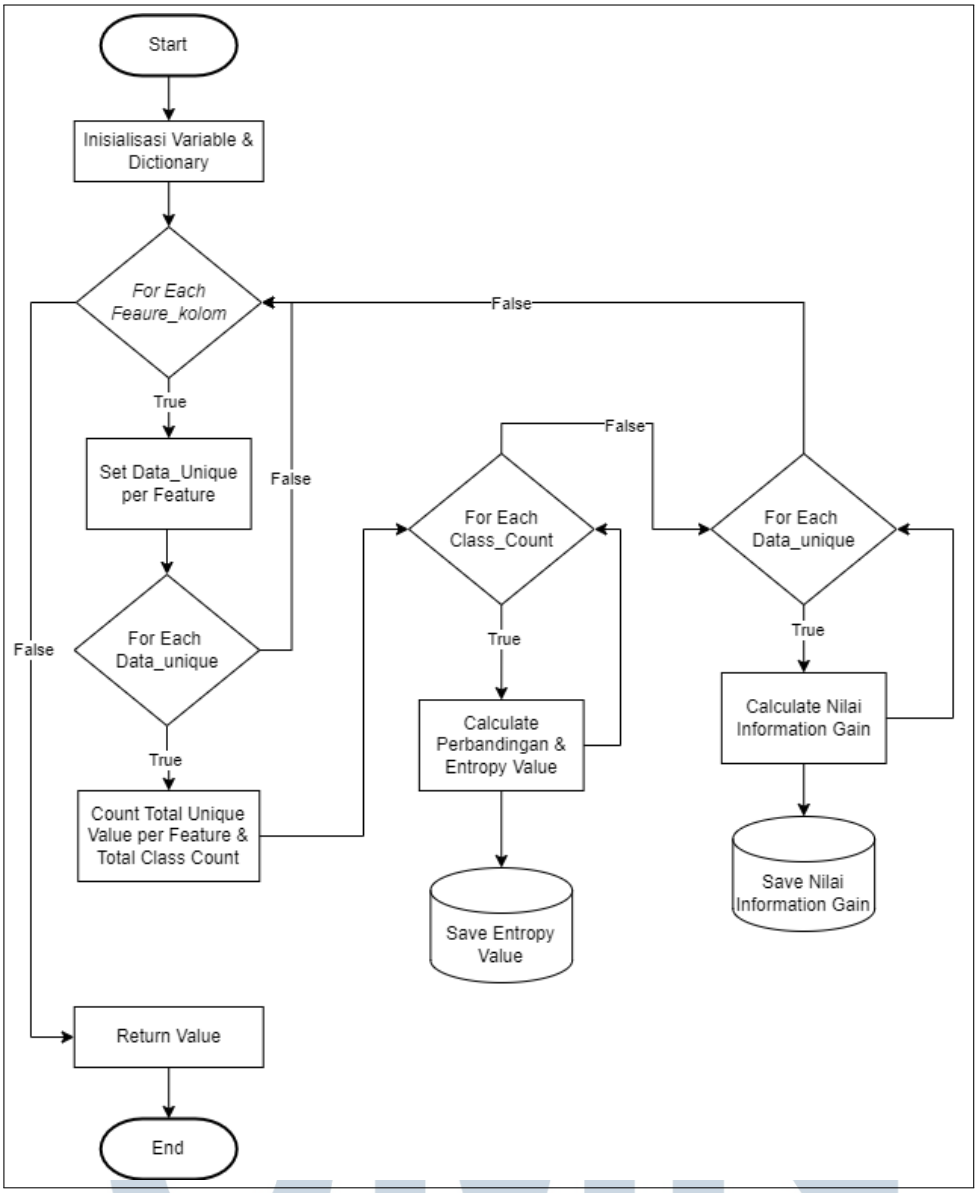
1. Menghitung *entropy* setiap *unique value feature* dan setiap *feature*
2. Menghitung *information gain* setiap *feature*



Gambar 3.3. *Calculate Total Entropy* setiap *Feature*

Pada Gambar 3.3 merupakan *flowchart* untuk menghitung *entropy* setiap *feature*. Berikut langkah-langkah untuk menghitung *entropy* setiap *feature* dataset:

1. Menginisialisasi *variable* untuk menghapus kolom *class*, menghitung total data secara keseluruhan, dan membuat *dictionary* untuk menampung hasil total *entropy* per *feature*.
2. Melakukan perulangan iterasi pertama melalui kolom *feature*.
3. Apabila kondisi iterasi pertama *true*, maka dilakukan perhitungan jumlah *class* yang ada dalam setiap *feature* dan membuat *variable total entropy*.
4. Melakukan perulangan iterasi kedia melalui perhitungan *class* yang sudah dihitung pada proses sebelumnya.
5. Apabila iterasi kedua *true*, langkah selanjutnya adalah menghitung perbandingan dan *total entropy* dari setiap *feature* dataset dengan menggunakan Persamaan 2.1. Lalu dari hasil tersebut disimpan ke dalam sebuah *variable total entropy* per *feature*.
6. Apabila iterasi kedua sudah selesai maka proses tersebut keluar dari perulangan iterasi kedua, dan kembali lagi ke proses perulangan pertama.
7. Apabila iterasi pertama sudah selesai maka proses tersebut keluar dari perulangan iterasi pertama, dan mengakhiri proses perhitungan dari *total entropy* setiap *feature*.



Gambar 3.4. *Calculate Entropy Unique Value & Information Gain*

Pada Gambar 3.4 merupakan *flowchart* untuk menghitung *entropy* setiap *unique value*, dan menghitung nilai *information gain* setiap *feature*. Berikut merupakan penjelasan langkah-langkah pada *flowchart entropy unique value* dan *information gain*:

1. Langkah pertama adalah melakukan inisialisasi variable, dan dictionary yang digunakan dalam proses perhitungan *entropy*, nilai *information gain*, jumlah *feature information gain*, dan membuat variable kolom fitur untuk menghapus kolom *class*.

2. Melakukan perulangan iterasi pertama melalui kolom fitur. Pada iterasi pertama apabila kondisi nya *true* maka membuat sebuah variable dan mencari *unique value* setiap *feature*.

3. Melakukan perulangan iterasi kedua melalui *unique value*. Apabila kondisi iterasi kedua adalah *true*, maka proses selanjutnya adalah memisahkan setiap *feature* dengan *value* per kolom yang sama, menghitung banyak nya *unique value* per *feature*, dan menghitung banyak nya *value class*.

4. Melakukan perulangan iterasi ketiga melalui *value class* yang sudah dihitung pada iterasi kedua. Apabila kondisi iterasi nya *true*, maka proses selanjutnya adalah menghitung hasil perbandingan setiap *feature*, dan dari hasil perbandingan yang sudah didapatkan dihitung nilai *entropy nya* dengan menggunakan Persamaan 2.1. Setelah sudah ada nilai *entropy* dari setiap *feature* maka hasilnya disimpan ke dalam *dictionary entropy value*.

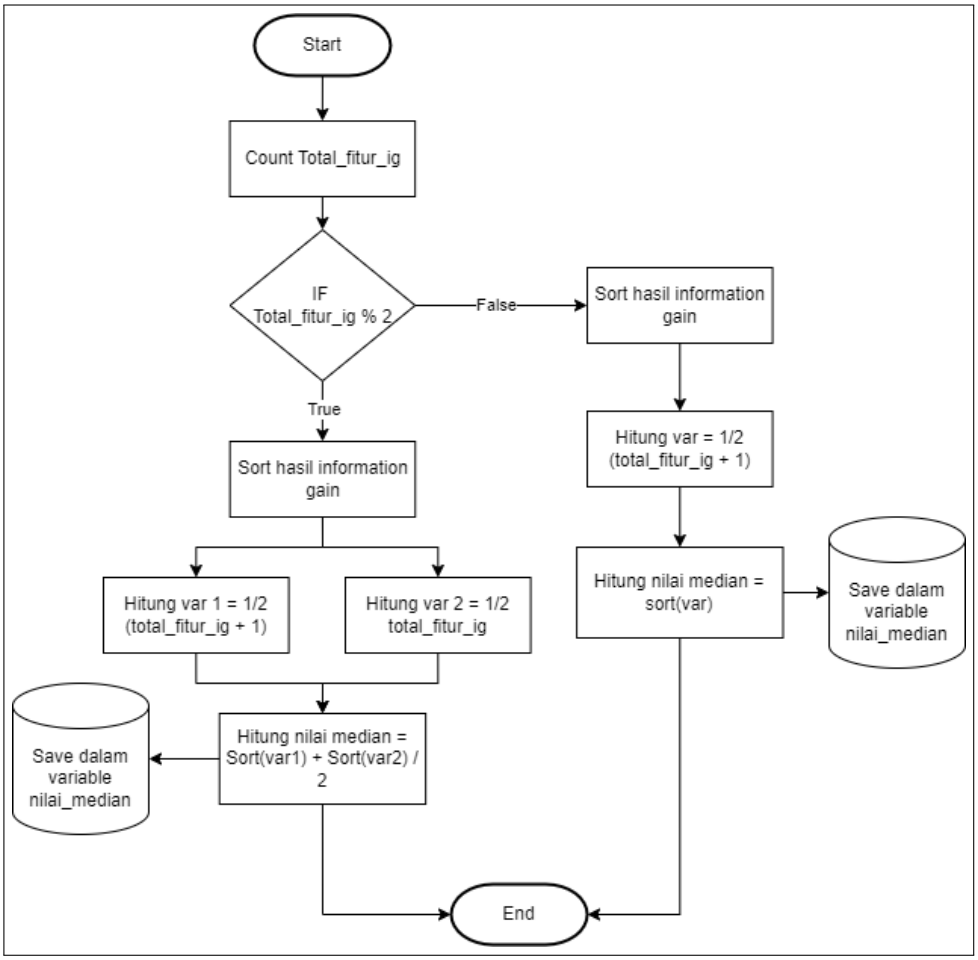
5. Ketika kondisi iterasi ketiga *false*, maka proses selanjutnya adalah melakukan perulangan iterasi keempat.

6. Apabila kondisi iterasi keempat adalah *true*, maka proses selanjutnya adalah melakukan perhitungan nilai *information gain* dari setiap *feature* dengan menggunakan Persamaan 2.2. Lalu hasil perhitungan tersebut disimpan.

7. Apabila kondisi iterasi kedua sudah selesai dan kondisinya sudah *false*, maka kembali lagi ke perulangan iterasi pertama. Dan ketika proses perulangan iterasi pertama sudah selesai dan kondisi sudah *false*, maka proses selanjutnya adalah melakukan *return value* dari *variable* dan *dictionary* yang telah melalui proses perhitungan.

3.2.4 Threshold Median

Tahap *threshold* median adalah tahapan untuk menghitung nilai median dari nilai *information gain* setiap *feature*. Nilai median tersebut dijadikan sebagai nilai akhir *threshold*.

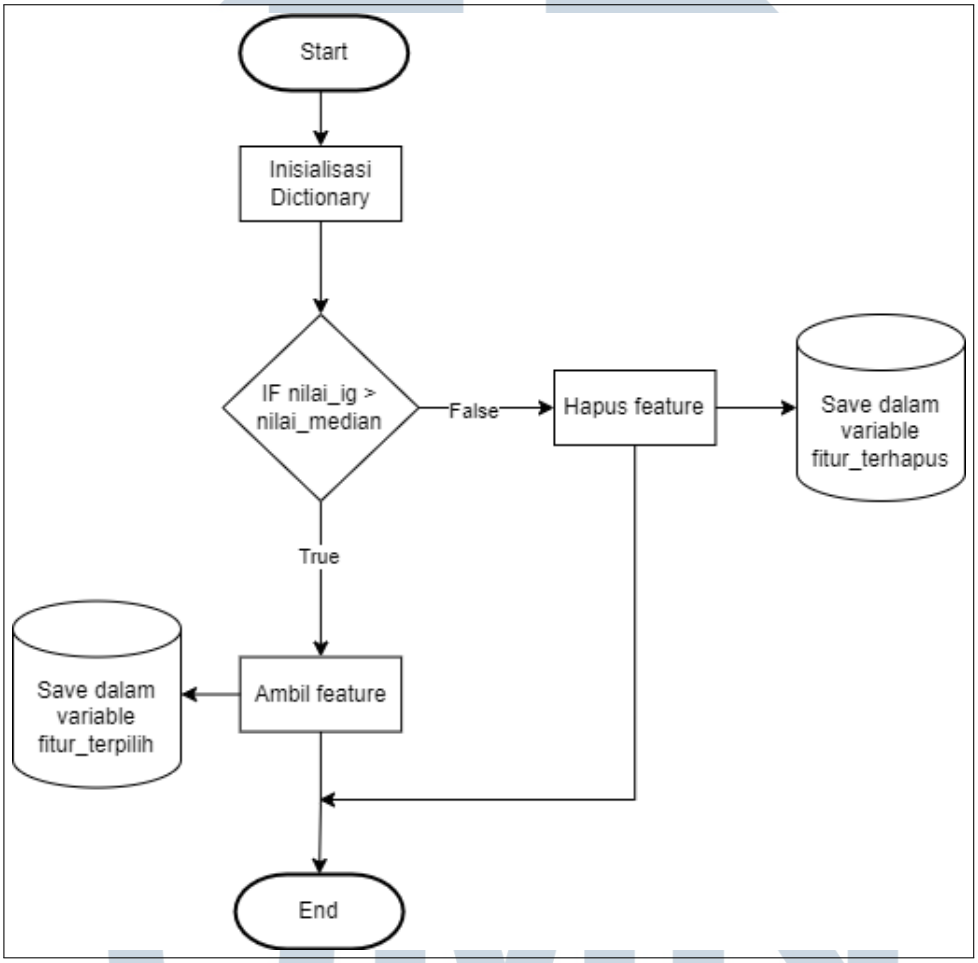


Gambar 3.5. *Threshold* Median

Pada Gambar 3.5 merupakan *flowchart* dari menghitung nilai akhir *threshold* dengan menggunakan Persamaan median. Pada tahap ini dilakukan ketika sudah menghitung semua *information gain* dari *feature* dataset. Langkah pertama adalah menghitung ada berapa banyak jumlah *total feature* yang terdapat ddalam *dictionary* nilai ig, perhitungan jumlah disimpan dalam sebuah *variable* total fitur ig. Apabila nilai total fitur ig berjumlah genap, maka dilakukan *sort* hasil *information gain*, lalu melakukan perhitungan nilai median dengan Persamaan 2.4, dan hasilnya disimpan dalam *variable* nilai median. Apabila nilai total fitur ig berjumlah ganjil, maka dilakukan *sort* hasil *information gain*, lalu melakukan perhitungan nilai median dengan Persamaan 2.3, dan hasilnya disimpan dalam *variable* nilai median.

3.2.5 Feature Selection

Tahap *feature selection* adalah tahapan untuk memilih *feature* yang memiliki nilai *information gain* diatas nilai akhir *threshold*.



Gambar 3.6. *Feature Selection*

Pada Gambar 3.6 merupakan *flowchart feature selection*. Proses dari *feature selection* dilakukan ketika nilai *threshold* median nya sudah didapatkan. Berikut merupakan penjelasan dari *flowchart* proses *feature selection*:

1. Langkah pertama adalah menginisialisasi *dictionary* untuk menampung nilai dari *feature* yang terpilih atau terhapus.
2. Selanjutnya melakukan perulangan iterasi dari *feature* dan *variable* nilai ig.
3. Apabila kondisi iterasi nya *true* maka selanjutnya masuk ke dalam kondisi jika nilai ig lebih besar dari nilai median atau tidak. Jika nilai ig lebih besar

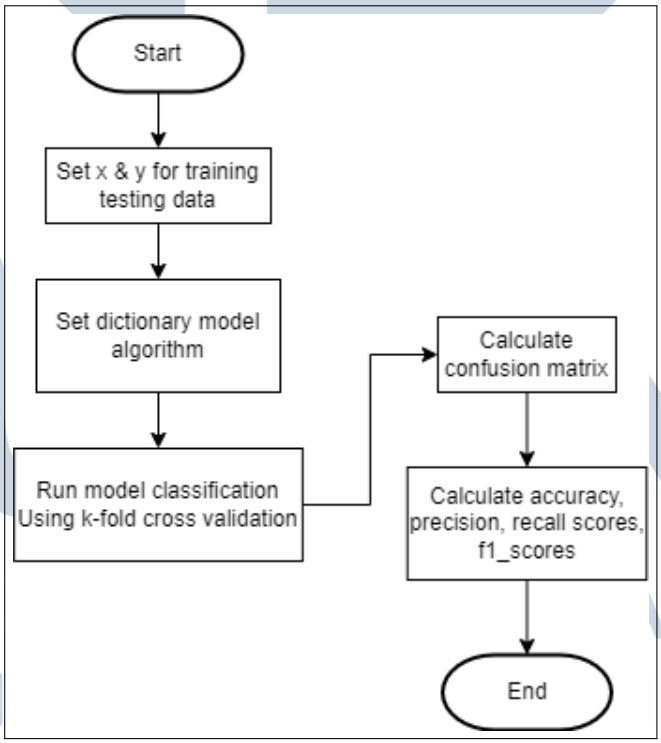
maka *feature* diambil dan disimpan dalam *variable* fitur terpilih. Apabila nilai ig lebih kecil maka *feature* dihapus dan disimpan dalam *variable* fitur terhapus.

4. Ketika semua iterasi selesai, maka perulangan iterasi berakhir.

Setiap *feature* dicek nilai *information gain*, lalu dilakukan *selection* dengan cara apabila nilai *information gain feature* nya lebih besar dari nilai *threshold* mediannya maka *feature* tersebut dipilih, apabila nilai *information gain feature* nya lebih kecil dari nilai *threshold* mediannya maka *feature* tersebut dihapus. *Feature* yang dipilih dilakukan uji coba klasifikasi.

3.2.6 Evaluate Model

Tahap *evaluate model* adalah tahapan untuk melakukan uji coba klasifikasi hasil dataset yang sudah dilakukan *feature selection*.



Gambar 3.7. *Evaluate Model*

Pada Gambar 3.7 merupakan *flowchart evaluate model*. Model klasifikasi diuji dengan *feature* yang sudah dipilih melalui proses *feature selection*. Proses pengujian dilakukan dengan menggunakan metode *k-fold cross validation* dengan

nilai k=10, yang membagi dataset menjadi 10 lipatan. Proses evaluasi dapat dilakukan dengan bantuan *confusion matrix*. Dari hasil tersebut, dapat diketahui nilai *accuracy, precision, recall,* dan *f1-score* dari model klasifikasi yang sudah dilakukan uji coba.