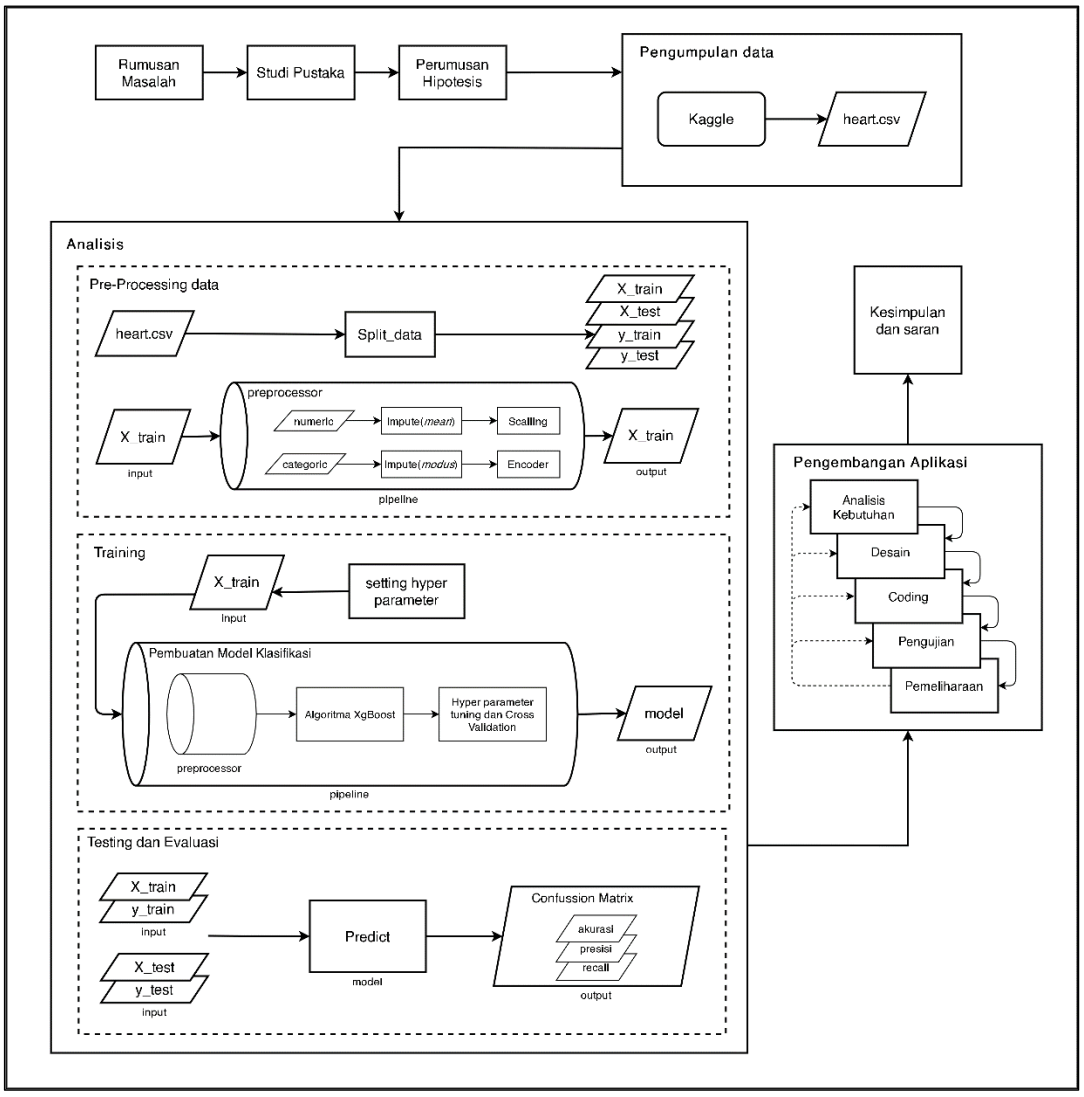
BAB III
METODOLOGI PENELITIAN

3.1.Metode Penelitian

Metode penelitian yang digunakan dalam penelitian ini adalah penelitian kuantitatif. Sedangkan jenis penelitian yang akan dilakukan dalam penelitian ini adalah penelitian implementatif, penelitian ini memuat perancangan dan pengembangan. Perancangan yang dilakukan adalah membuat sistem prediksi penyakit jantung yang dibuat menggunakan bahasa pemrograman Python yang akan ditampilkan dalam bentuk aplikasi berbasis *web*, sedangkan pengembangan yang dilakukan dalam penelitian ini adalah pembuatan model *machine learning* yang akan digunakan untuk melakukan klasifikasi pada prediksi penyakit jantung. Data yang akan digunakan adalah data tabular atau data berbentuk tabel, sedangkan *output* klasifikasi yang dihasilkan adalah label Ya untuk data yang diklasifikasikan terkena penyakit jantung, atau Tidak untuk data yang diklasifikasikan tidak terkena penyakit jantung. Model tersebut dibuat menggunakan algoritma XgBoost dan menggunakan *randomized search optimizer* untuk membantu menentukan *hyper parameter* terbaik yang diperlukan oleh model supaya nantinya menghasilkan model klasifikasi terbaik yang memiliki performa yang maksimal. Pemilihan algoritma XgBoost untuk membuat model klasifikasi karena algoritma tesebut dinilai lebih baik dan lebih cepat dalam hal komputasi dan dapat menghindari *overfitting* karena pada algoritma tersebut juga memiliki *regularization*.

Penelitian ini diawali dengan perumusan rumusan masalah, kemudian dilanjutkan studi literatur yang dilakukan dengan mencari informasi mengenai topik yang bersangkutan lewat sumber-sumber referensi seperti jurnal, buku, dan sumber-sumber lain yang dapat dipercaya. Dalam tahapan ini, peneliti mencari dan mengumpulkan informasi mengenai penelitian-penelitian sebelumnya yang berkaitan dengan prediksi penyakit jantung dan penelitian sebelumnya yang dilakukan menggunakan algoritma XgBoost. Pengumpulan informasi tersebut dilakukan untuk mengetahui perkembangan topik prediksi penyakit jantung seperti algoritma apa saja yang sudah pernah digunakan untuk menyelesaikan kasus prediksi tersebut, dan juga untuk mengetahui kelebihan dan kekurangan algoritma lain dalam meyelesaikan kasus prediksi penyakit jantung. Yang nantinya dijadikan bahan evalusai dan pertimbangan penelitian ini dilakukan dengan algoritma yang dipilih, yaitu algoritma XgBoost. Setelah itu, dilakukan pengumpulan data dan analisis menggunakan algoritma XgBoost. Dilanjutkan dengan pembuatan sistem agar sistem prediksi dapat digunakan oleh pengguna, dan menemukan kesimpulan dan saran dari penelitian yang dilakukan. Tahapan penelitian dapat dilihat pada Gambar 3.1 di bawah ini.



Gambar 3. 1 Tahapan Penelitian

Tahapan-tahapan penelitian yang akan dilakukan digambarkan pada gambar 3.1 diatas. Perincian setiap tahapan penelitian yang dilakukan akan dijelaskan di bawah ini.

3.2.Pengumpulan Data

Data yang akan digunakan dalam penelitian ini adalah data dengan jenis data sekunder. Data sekunder adalah data yang bisa didapatkan secara tidak langsung, misalnya didapatkan lewat orang lain ,lewat sumber dokumen yang sudah memuat data tersebut, melalui website, dan sumber data lainnya (Haqie et al., 2020). Data yang akan digunakan di dalam penelitian ini adalah data *Clevaland Heart Disease* yang didapatkan dengan cara mengunduh data pada *repository* resminya yaitu bersumber dari UCI *machine learning* ataupun dapat juga diunduh dari situs kaggle. Data yang akan digunakan berjumlah sebanyak 303 data dengan rincian 138 data tergolong dalam kelas tidak memiliki penyakit jantung atau sehat, dan sebanyak 165 data tergolong ke dalam memiliki penyakit jantung. Data tersebut memiliki 13 kolom yang memuat atribut data yang akan digunakan untuk klasifikasi. Untuk lebih jelasnya, penjelasan mengenai kolom yang ada pada *dataset* terdapat pada Tabel 3.1 di bawah ini.

Tabel 3. 1 Kolom Pada *Dataset*

Atribut	Deskripsi	Keterangan
Age	Umur pasien	*Numerik*
Sex	Jenis kelamin pasien	0: Wanita, 1: Pria
Cp	*Chest pain type*	1:*typical angina*, 2:*atypical angina*, 3:*non-angina pain*, 4:*asympotomatic*
Trestbps	*Resting blood presure*	*Numerik*
Chol	Serum kolesterol	*Numerik*
Fbs	*Fasting blood sugar*>120 mg/dl	0:*false*, 1:*true*
Restecg	Hasil ECG selama istirahat	0:normal, 1:abnormal (memiliki kelainan gelombang ST-T), 2:*hipertrofil ventrikel*
Thalac	Detak jantung maksimal yang dicapai	*Numerik*
Exang	Ukuran *boolean* yang menunjukan apakah latihan angina industri terjadi	0:*No*, 1:*Yes*
Oldpeak	*Segment* ST yang diperoleh dari latihan relatif terhadap istirahat	*Numerik*
Slope	Kemiringan segmen ST untuk latihan maksimal (puncak)	1:*upsloping*, 2:*flat*, 3:*downsloping*
Ca	Jumlah *vessel* utama yang diwarnai oleh *fluroskopi*	0, 1, 2, dan 3
Thal	*Thal*	1:*normal*, 2:cacat tetap, 3:cacat *reversible*

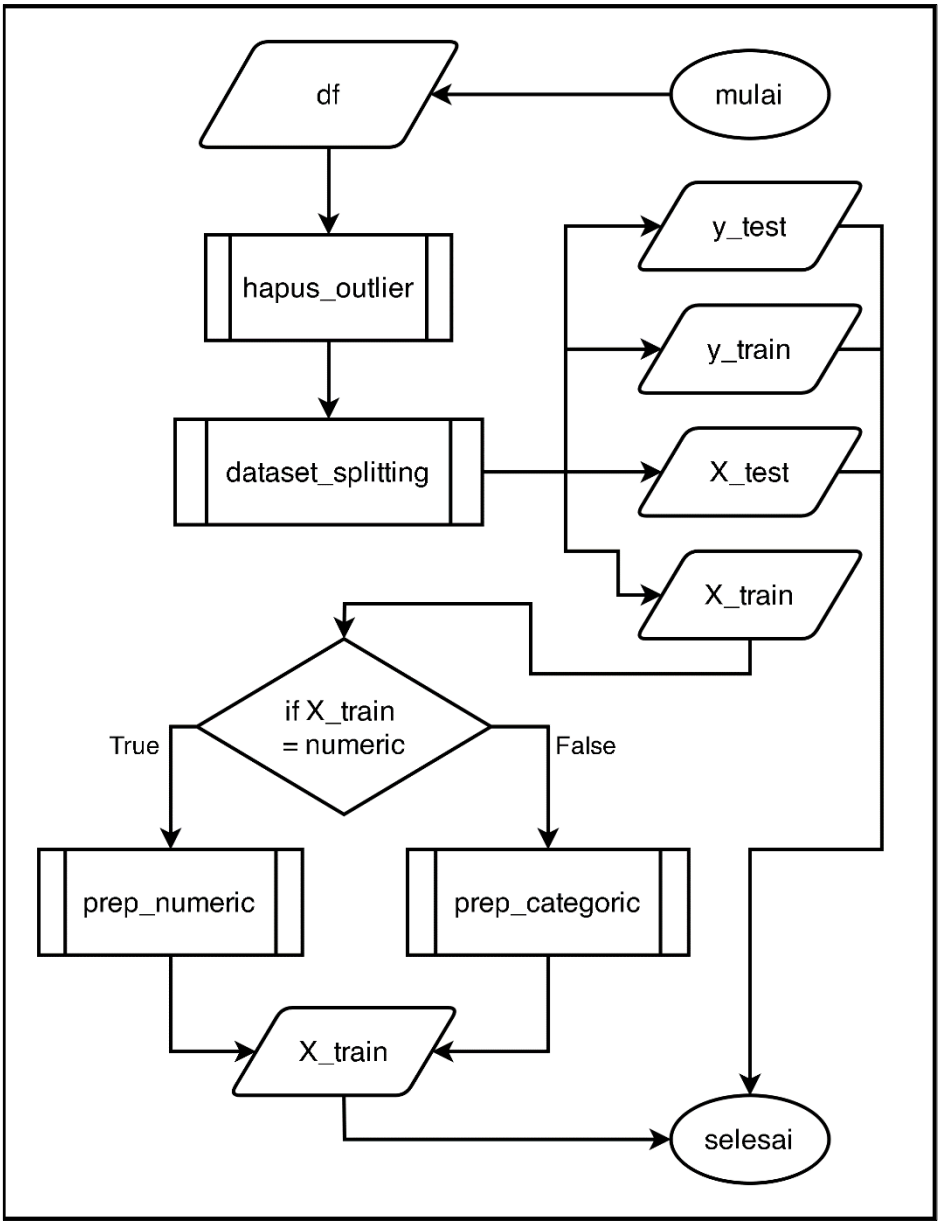
Sedangkan untuk contoh data mentah yang akan digunakan dapat dilihat pada Tabel 3.2 di bawah ini.

Tabel 3. 2 Dataset

Age	Sex	cp	Trestbp	chol	fbs	restecg	thalac	exang	oldpeak	slope	ca	thal	Target
63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
58	1	1	120	284	0	0	160	0	1.8	1	0	2	0
58	1	2	132	224	0	0	173	0	3.2	2	2	3	0

3.3.Preprocessing

Dataset yang akan digunakan setelah diunduh sebelum dilakukan proses *training* pembuatan model dilakukan proses *preprocessing*. *Preprocessing* data adalah sebuah proses yang dilakukan dengan mengubah data ke dalam format data yang lebih sederhana, lebih efektif, sesuai dengan kebutuhan yang ingin digunakan pengguna (Saifullah et al., 2017). *Flowhcart* tahapan *preprocessing* terdapat pada gambar 3.2 dibawah ini.



Gambar 3. 2 Flowchart preprocessing

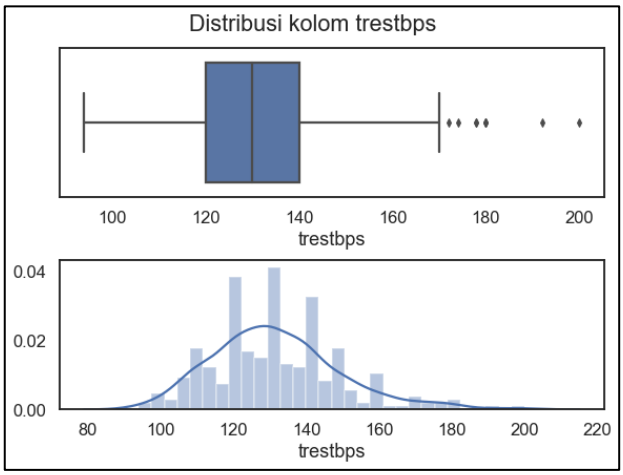
Dalam preprocessing, tahapan yang akan dilakukan adalah sebagai berikut.

1. Menghapus *outlier*.
2. Membagi data menjadi *data training* dan *data testing*.
3. Normalisasi pada data *numeric*.
4. *One hot encoding* pada data *categoric*.

3.3.1. Menghapus Outlier

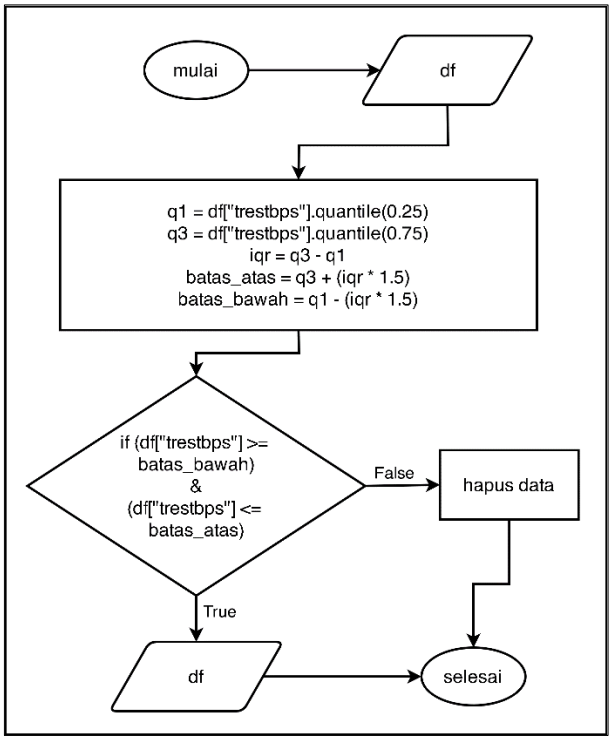
Tahap *preprocessing* yang pertama adalah membersihkan *outlier* data, apabila data yang digunakan memiliki nilai terlalu kecil atau terlalu besar dari dominan rentang datanya, maka data *outlier* tersebut dapat dihilangan supaya data lebih baik dan menghasilkan model yang lebih baik pula. Untuk mencari *outlier* pada data dapat dilihat dengan menggunakan *boxplot*

dan *distplot* seperti yang terlihat pada Gambar 3.3 di bawah ini. Pada gambar di bawah ini terlihat bahwa kolom tersebut memiliki *outlier* pada data di atas, yang terlihat dalam *boxplot*.



Gambar 3. 3 *Boxplot* dan *Distplot* kolom trestbps

Proses membersihkan *outlier* dilakukan dengan menghitung nilai IQR terlebih dahulu, selanjutnya ditentukan batas atas dan batas bawah dengan cara operasi perhitungan $Q1$ atau $Q3$ terhadap nilai IQR yang sudah ditentukan. *Flowchart* dari proses pembersihan oulier dapat dilihat pada Gambar 3.4 di bawah ini.



Gambar 3. 4 Flowchart hapus_outlier

Rumus untuk perhitungan menghapus *outlier* dapat dilihat pada persamaan di bawah ini.

$$Q1 = X\frac{1\times(n+1)}{4} \quad \text{.....(3.1)}$$

$Q3 = X\frac{3\times(n+1)}{4}$ .....(3.2)
$IQR = Q3 - Q1$ .....(3.3)
$batas\_bawah = Q1 - (1.5 \times IQR)$.....(3.4)
$batas\_atas = Q3 + (1.5 \times IQR)$.....(3.5)

Data dalam kolom trestbps dapat dilihat pada Tabel 3.3 di bawah ini. Sesuai dengan tahapan dan rumus yang telah dijelaskan di atas, maka penghapusan *oulier* pada kolom trestbps adalah sebagai berikut.

Q1 = $X\frac{1*(303+1)}{4} = X\frac{304}{4}$ = X ke-76 = 120
Q3 = $X\frac{3*(303+1)}{4} = X\frac{912}{4}$ = X ke 228 = 140
IQR = Q3 – Q1 = 140 – 120 = 20
Batas *outlier* bawah = Q1 – (1,5 * IQR) = 120 – (1,5*20) = 90
Batas *outlier* atas = Q3 + (1,5 * IQR) = 140 + (1,5*IQR) = 170

Tabel 3. 3 Data pada kolom trestbps

No	Data Asli (trestbps)	Data Setelah Diurutkan
1.	145	94
2.	130	94
3.	130	100
4.	120	100
5.	120	100
6.	140	100
7.	140	101
..	...	...
297.	124	178
298.	164	178
299.	140	180
300.	110	180
301.	144	180
302.	130	192
303.	130	200

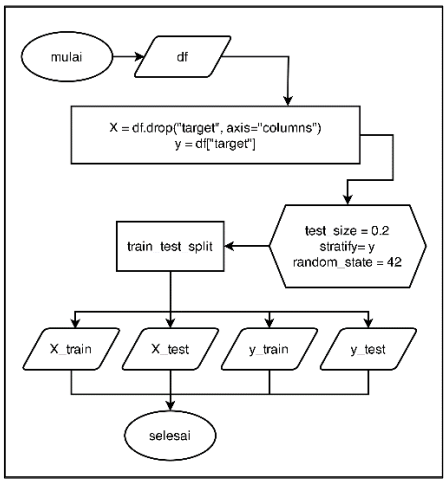
Sesuai dengan perhitungan diatas, kemudian dapat dilakukan *filter* pada kolom trestbps hanya menyisakan data yang memiliki nilai >= 90 dan <= 170. Sehingga nilai min dan max pada kolom tersebut dapat dilihat pada Tabel 3.4 di bawah ini.

Tabel 3. 4 Inforrmasi kolom trestbps

	Trestbps
Min	94
Mean	130
Max	170

3.3.2. Dataset Splitting

Pembagian *dataset* dilakukan untuk membagi data menjadi *data training* dan *data testing*. Pembagian data tersebut menggunakan komposisi 80:20 yaitu 80% akan menjadi *data training* dan 20% akan menjadi *data testing*. Flowchart proses *dataset splitting* dapat dilihat pada Gambar 3.5 di bawah ini.

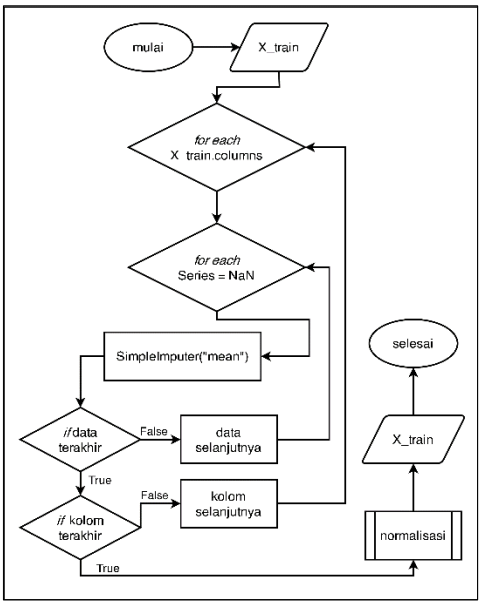


Gambar 3. 5 Flowchart dataset_splitting

Dari Gambar 3.5 di atas dapat dilihat bahwa hasil atau output dari proses tersebut akan menghasilkan 4 *dataframe* baru berupa *X_train*, *X_test*, *y_train*, dan *y_test*. Selanjutnya data yang masih akan dilakukan *preprocessing* hanya data *X_train*.

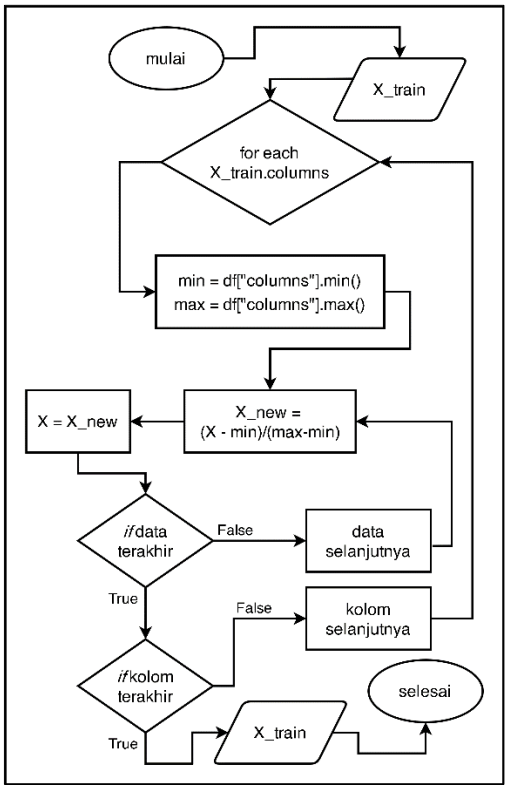
3.3.3. Data Numerik

Selanjutnya data *X_train* akan dibagi menjadi dua, yaitu data yang memiliki nilai numerik, dan data yang memiliki nilai kategorik. Untuk data numerik akan dilakukan proses *impute* menggunakan nilai rata-rata dari kolom tersebut, dan dilanjutnya dengan proses normalisasi berupa *scalling*. Flowchart proses preprocessing pada data numerik dapat dilihat pada gambar 3.6 di bawah ini.



Gambar 3. 6 Flowchart prep_numeric

Setelah data setiap kolom dilakukan impute, maka dilanjutkan dengan normalisasi berupa *scalling*. Proses normalisasi dilakukan supaya proses *training* menjadi lebih cepat, karena dapat memudahkan model dalam memahami data (Hanifa et al., 2017). *Scalling* dalam proses ini dilakukan dengan menggunakan *MinMax*, yaitu merubah data ke dalam range 0 untuk data terkecil dan 1 untuk data terbesar. Flowchart dari proses *scalling* dapat dilihat pada Gambar 3.7 di bawah ini.



Gambar 3. 7 Flowchart Normalisasi MinMax

Rumus dan data yang akan digunakan dapat dilihat pada Tabel 3.5 dan 3.6 di bawah ini.

Tabel 3. 5 Contoh data untuk scalling

No.	Data lama	Data Baru
1.	145	0.67
2.	130	0.47
3.	130	0.47
4.	120	0.34
5.	120	0.34
6.	140	0.6
7.	140	0.6
8.	120	0.34
9.	150	0.73
10.	140	0.6
...	...	
287.	140	0.6
288.	124	0.39

Tabel 3. 6 Lanjutan contoh data untuk scaling

No.	Data lama	Data Baru
289.	164	0.92
290.	140	0.6
291.	110	0.21
292.	144	0.65
293.	130	0.47
294.	130	0.47

$X_{new} = \frac{X - X_{min}}{X_{max} - X_{min}}$ .....(3.6)

Keterangan:

$X_{min} = data\ terkecil$

$X_{max} = data\ terbesar$

Sehingga dari contoh data yang ditampilkan dalam Tabel 3.5 dan Tabel 3.6 dapat diketahui bahwa $X_{min} = 94$ dan nilai $X_{max} = 170$. Contoh perhitungan pada data ke-1 adalah sebagai berikut, untuk dapat lainnya dapat dilihat pada Tabel 3.5 dan Tabel 3.6 di atas.

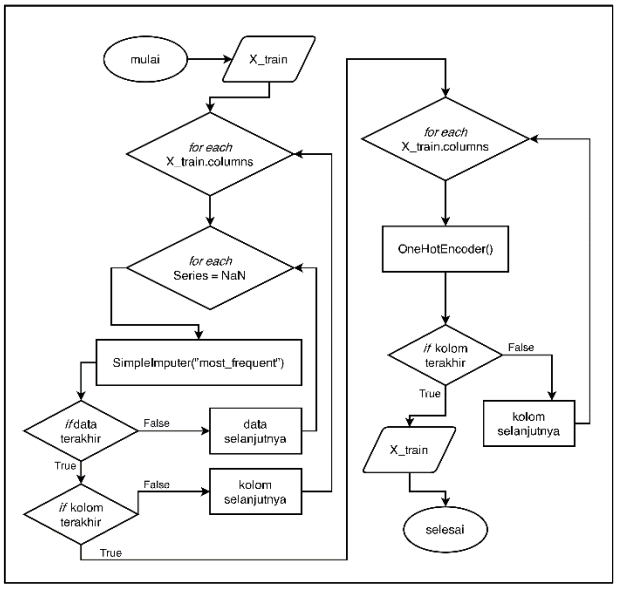
$$X_{new} = \frac{145 - 94}{170 - 94}$$

$$X_{new} = \frac{51}{76}$$

$$X_{new} = 0{,}67$$

3.3.4. Data Kategorik

Kemudian pada data kategorik dilakukan *preprocessing* berupa *impute* data yang kosong dengan *most_frequent* atau modus dari kolom tersebut, dilanjutkan proses *encoding* menggunakan *One Hot Encoding*. Flowchart proses *preprocessing* pada data kategorik dapat dilihat pada Gambar 3.8 di bawah ini.



Gambar 3. 8 Flowchart prep_categoric

Untuk proses *One Hot Encoding* adalah proses mengganti nilai kategorikal dengan nilai *binary* berupa 0 atau 1. Dilakukan dengan cara membuat kolom baru sebanyak kategori yang ada, kemudian kolom baru yang sesuai dengan nilai kategori tersebut akan diberi nilai 1 dan kolom yang tidak sesuai dengan kategori akan diberi nilai 0. Contoh penerapan *One Hot Encoding* dapat dilihat pada Tabel 3.7 dan hasilnya ditampilkan pada Tabel 3.8 di bawah ini.

Tabel 3. 7 Contoh data untuk One Hot Encoding

No.	**Slope**
1.	0
2	0
3.	2
4.	2
5.	2
6.	1

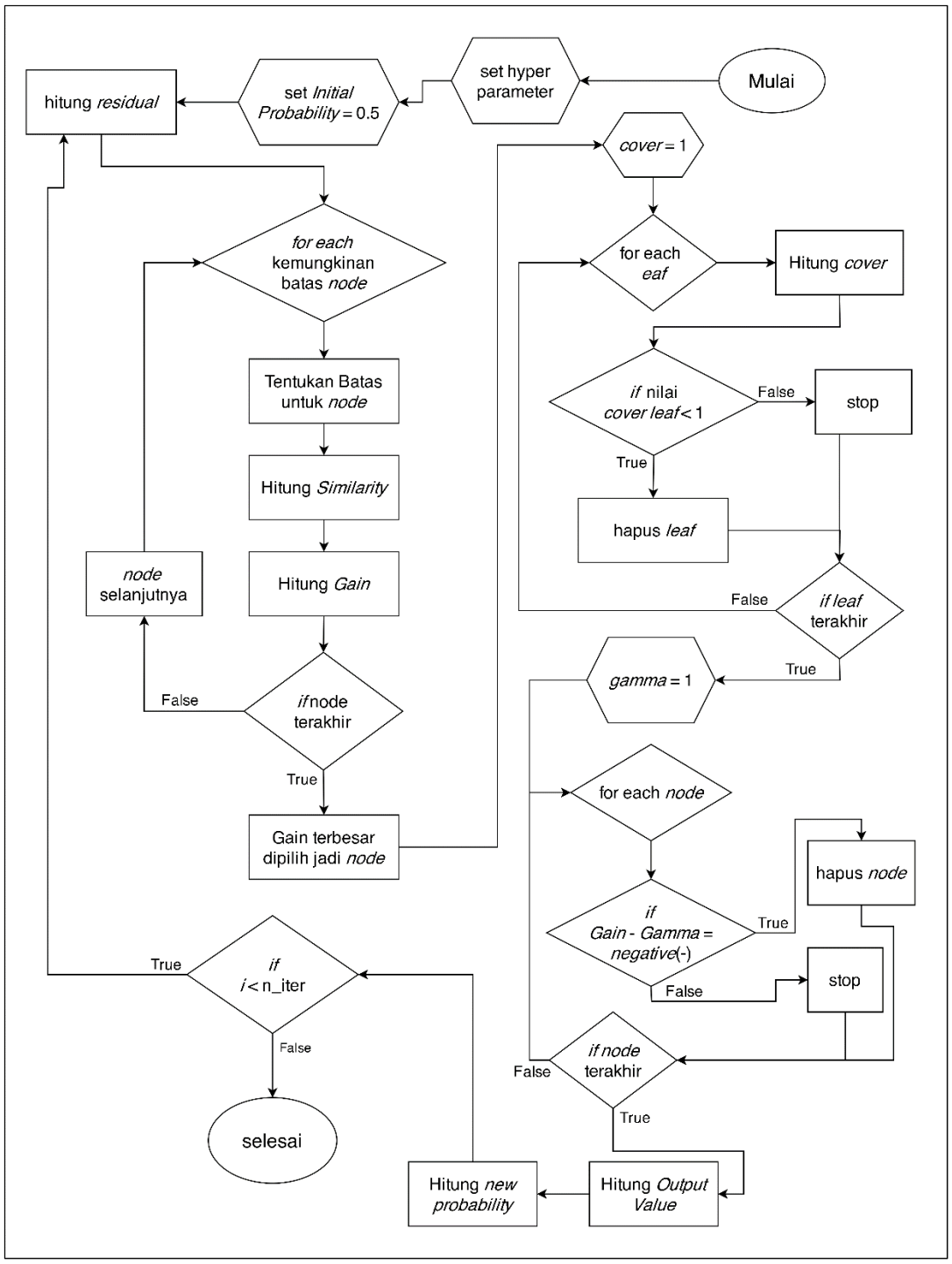
Tabel 3. 8 Hasil proses One Hot Encoding

No.	Slope_0	Slope_1	Slope_2
1.	1	0	0
2.	1	0	0
3.	0	0	1
4.	0	0	1
5.	0	0	1
6.	0	1	0

Dapat dilihat pada Tabel 3.8 di atas, bawah nilai kategori *slope* akan menjadi kolom baru sesuai kategori yang ada. Dalam contoh di atas nilai kategori yang ada adalah 0, 1, 2. Kemudian kolom yang baru akan diberi nilai 1 jika kategori sesuai, dan 0 jika tidak sesuai.

3.4.Training

Proses training yang dilakukan oleh algoritma xgboost dapat dilihat pada flowchart Gambar 3.9 di bawah ini.



Gambar 3. 9 Flowchart XgBoost

Sesuai dengan Gambar 3.9 *flowchart* diatas, algoritma xgboost akan menentukan *initial prediction* atau *initial probability* sebagai prediksi awal dengan nilai yaitu 0,5. Kemudian

dilakukan perhitungan untuk mengetahui nilai *residual* dari masing-masing data dengan rumus yang ditampikan pada persamaan 3.7 di bawah ini.

$$residual = actual\ value - probability \quad \text{.....(3.7)}$$

Setelah mengetahui nilai *residual* dari masing-masing data, kemudian dilakukan proses pembuatan *tree*. Tahap pertama adalah menentukan *root node* untuk *tree* tersebut, untuk menentukan *root note* dilakukan dengan membuat *tree* sebanyak kemungkinan *node* yang ada, kemudian dilakukan perhitungan untuk menentukan nilai *similarity* masing-masing *leaf*. Dari nilai *similarity* tersebut, kemudian dapat diketahui nilai *gain* dari *node* tersebut. node yang memiliki nilai gain terbesarlah yang akan dipilih untuk menjadi root node. Rumus untuk menghitung $similarity$ dan $gain$ dapat dilihat pada persamaan 3.8 dan 3.9 di bawah ini.

$$similarity = \frac{\sum(residual_i)^2}{\sum[prev\ probability_i \times (1 - prev\ probability_i)] + \lambda} \quad \text{.....(3.8)}$$

$$gain = left\ similarity + right\ similarity - root\ similarity \quad \text{.....(3.9)}$$

Kemudian, setelah satu *tree* terbentuk dapat dilakukan pengecekan untuk memastikan apakah *tree* tersebut perlu dipangkas atau tidak dengan menggunakan *cover* dan *gamma*(γ). Apabila nilai c*over leaf* tersebut lebih kecil dari nilai *cover* yang sudah ditentukan, maka *leaf* tersebut dapat dipangkas. Begitu juga dengan *node*, apabila nilai dari ($gain - \gamma$) bernilai negative(-), maka *node* tersebut dapat dipangkas. Rumus untuk menentukan nilai $cover$ untuk *leaf* dapat dilihat pada persamaan 3.10 berikut.

$$cover = \sum[prev\ probability_i \times (1 - prev\ probability_i)] \quad \text{..... (3.10)}$$

Selanjutnya, setelah *tree* terbuat dan *tree* juga sudah dilakukan pengecekan apakah memungkinkan untuk dipangkas atau tidak, satu tree sudah dibuat. Tahap selanjutnya dapat dilakukan dengan menghitung *output value* untuk masing-masing *leaf* dengan rumus berikut.

$$O_{value} = \frac{\sum(residual_i)}{\sum[prev\ probability_i \times (1 - prev\ probability_i)] + \lambda} \quad \text{..... (3.11)}$$

Setelah menentukan O_{value}, kemudian dilakukan perhitungan untuk menentukan *new probability* yang nantinya akan digunakan untuk menghitung *residual* baru dan membuat *tree* pada iterasi selanjutnya. Rumus untuk menentukan *probability* adalah sebagai berikut.

$$\log(odds) = \log\left(\frac{p}{1-p}\right) + \sum[\varepsilon \times O_{value}]_i \quad \text{..... (3.12)}$$

$$probability = \frac{e^{\log(odds)}}{1 + e^{\log(odds)}} \quad \text{..... (3.13)}$$

Contoh *dataset* yang akan digunakan untuk proses *training* terdapat pada Tabel 3.9 di bawah ini.

Tabel 3. 9 Contoh Dataset Untuk Training

Cp	Thalach	Target
3	150	1
3	190	1
0	96	0
1	174	0
1	202	1

Pertama, buat *initial probability* dengan nilai 0,5. Kemudian dapat dihitung nilai *residual* masing-masing data dengan rumus $actual\ value - probability$. Kemudian buat tabel untuk memperbarui data nilai $residual$ yang sudah dihitung. Nilai $residual$ dapat dilihat pada tabel 3.10 di bawah ini.

Tabel 3. 10 Tabel Dengan Nilai Residual

Cp	Thalac	Target	Residual
3	150	1	0.5
3	190	1	0.5
0	96	0	-0.5
1	174	0	-0.5
1	202	1	0.5

Kemudian nilai $residual$ tersebut, diasumsikan sebagai *leaf* dan dihitung $similarity$ yang nantinya akan dijadikan $Root_{similarity}$ pada saat perhitungan nilai $gain$ untuk masing-masing kemunkinan *node* yang ada. Perhitungan $similarity$ adalah sebagai berikut. Untuk perhitungan dengan data di atas, kita asumsikan nilai $lambda(\lambda)$ adalah 0.

$$Root_{similarity} = \frac{(0.5 + 0.5 - 0.5 - 0.5 + 0.5)^2}{\Big(\big(0.5 \times (1 - 0.5)\big) \times 5\Big) + 0}$$

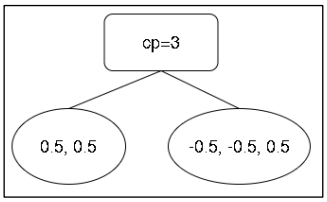
$$Root_{similarity} = \frac{(0.5)^2}{0.25 \times 5}$$

$$Root_{similarity} = \frac{0.25}{1.25}$$

$$Root_{similarity} = 0.2$$

Kemudian, kita lakukan perhitungan untuk mengetahui nilai $gain$ pada masing-masing kemungkinan *node*, nilai $gain$ yang paling tinggi yang nantinya akan dijadikan *node*. Beberapa kemungkinan *node* adalah sebagai berikut.

1. Cp = 3



Gambar 3. 10 Tree Untuk Node Cp=3

Pertama, yang dilakukan adalah menghitung $similarity$ untuk *leaf* sebelah kiri yang memiliki nilai 0,5 dan 0,5. Cara menghitung sama seperti rumus $similarity$ yang sudah dijelaskan di atas, perhitungan $Left_{similarity}$ dapat dilihat di bawah ini.

$$Left_{similarity} = \frac{(0.5 + 0.5)^2}{((0.5 \times (1 - 0.5)) \times 2) + 0}$$
$$Left_{similarity} = \frac{1^2}{(0.25 \times 2) + 0}$$
$$Left_{similarity} = \frac{1}{0.5}$$
$$Left_{similarity} = 2$$

Kemudian dilanjutkan dengan menghitung $similarity$ untuk *leaf* sebelah kanan, yang memiliki tiga nilai yaitu -0.5, -0.5 dan 0.5. Perhitungan $Right_{similarity}$ dapat dilihat di bawah ini.

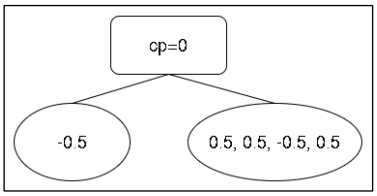
$$Right_{similarity} = \frac{(-0.5 - 0.5 + 0.5)^2}{\left(\left(0.5 \times (1 - 0.5)\right) \times 3\right) + 0}$$
$$Right_{similarity} = \frac{(-0.5)^2}{(0.25 \times 3)}$$
$$Right_{similarity} = \frac{0.25}{0.75}$$
$$Right_{similarity} = 0.33$$

Setelah mengetahui nilai $Left_{similarity}$ dan $Right_{similarity}$, langkah selanjutnya adalah menghitung nilai $gain$. Nilai $gain$ didapatkan dengan menghitung jumlah $similarity$ dikurangi dengan $Root_{similarity}$. Perhitungan nilai $gain$ untuk node cp=3 dapat dilihat di bawah ini.

$$gain = Left_{similarity} + Right_{similarity} - Root_{similarity}$$
$$gain = 2 + 0.33 - 0.2$$
$$gain = 2.13$$

Setelah kita mendapatkan nilai $gain$ dari *probabillity node* ini, simpan nilai $gain$ itu dan kita hitung nilai $gain$ untuk *probabillity node* selanjutnya.

2. Cp = 0



Gambar 3. 11 Tree Untuk Node Cp=0

Selanjutnya adalah perhitungan untuk *probabillity node* berupa cp=0. Pertama kita gambarkan *probabillity node* seperti yang terlihat pada Gambar 3.11 di atas. Kemudian kita hitung nilai $similarity$ untuk masing-masing *leaf*. Perhitungan $Left_{similarity}$ dapat dilihat di bawah ini.

$$Left_{similarity} = \frac{(-0.5)^2}{\big(0.5 \times (1 - 0.5)\big) + 0}$$
$$Left_{similarity} = \frac{0.25}{0.25}$$
$$Left_{similarity} = 1$$

Kemudian menghitung nilai $Right_{similarity}$ yang memiliki nilai $residual$ berupa 0.5, 0.5, -0.5, dan 0.5. perhitungan $Right_{similarity}$ dapat dilihat di bawah ini.

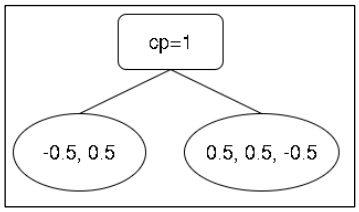
$$Right_{similarity} = \frac{(0.5 + 0.5 - 0.5 + 0.5)^2}{\big(\big(0.5 \times (1 - 0.5)\big) \times 4\big) + 0}$$
$$Right_{similarity} = \frac{(0.5 + 0.5)^2}{(0.25 \times 4) + 0}$$
$$Right_{similarity} = \frac{1}{1}$$
$$Right_{similarity} = 1$$

Setelah menghitung nilai $Left_{similarity}$ dan nilai $Right_{similarity}$, kemudian dilanjutkan menghitung nilai $gain$ dari *probabillity node* ini. Perhitungan nilai $gain$ pada *probabillity node* dilakukan dengan menjumlahkan $similarity$ dan dikurangi dengan nilai $Root_{similarity}$, seperti yang dapat dilihat di bawah ini.

$$gain = Left_{similarity} + Right_{similarity} - Root_{similarity}$$
$$gain = 1 + 1 - 0.2$$
$$gain = 1.8$$

Setelah mendapatkan nilai $gain$, simpan nilai $gain$ tersebut untuk nantinya dibandingkan dengan $gain$ *probabillity node* lainnya. Kemudian dilanjutkan menghitung $similarity$ dari *probabillity node* lainnya.

3. Cp = 1



Gambar 3. 12 Tree Untuk Node Cp=1

Probabillity node selanjutnya adalah cp=1. Pertama kita gambarkan *probabillity node* tersebut seperti pada Gambar 3.12 di atas, kemudian dilanjutkan menghitung nilai $similarity$ baik dari *leaf* sebelah kiri dan sebelah kanan. Untuk perhitungan $Left_{similarity}$ dapat dilihat pada operasi perhitungan di bawah ini.

$$Left_{similarity} = \frac{(-0.5 + 0.5)^2}{\Big(\big(0.5 \times (1 - 0.5)\big) \times 2\Big) + 0}$$

$$Left_{similarity} = \frac{0^2}{0.25 \times 2}$$

$$Left_{similarity} = 0$$

Kemudian setelah mendapatkan nilai $Left_{similarity}$, selanjutnya adalah menghitung nilai $similarity$ untuk leaf sebelah kanan yang berisi nilai $residual$ berupa 0.5, 0.5, dan -0.5. Perhitungan $Right_{similarity}$ dapat dilihat di bawah ini.

$$Right_{similarity} = \frac{(0.5 + 0.5 - 0.5)^2}{\Big(\big(0.5 \times (1 - 0.5)\big) \times 3\Big) + 0}$$

$$Right_{similarity} = \frac{0.5^2}{0.25 \times 3}$$

$$Right_{similarity} = \frac{0.25}{0.75}$$

$$Right_{similarity} = 0.33$$

Setelah mendapatkan nilai $Left_{similarity}$ dan $Right_{similarity}$, selanjutnya menghitung $gain$ dari *probabillity node* tersebut dengan cara menjumlahkan $similarity$ dan dikurangi dengan $Root_{similarity}$ yang sudah dihitung di awal. Perhitungan $gain$ dapat dilihat di bawah ini.

$$gain = Left_{similarity} + Right_{similarity} - Root_{similarity}$$

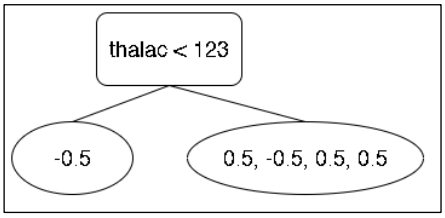
$$gain = 0 + 0.33 - 0.2$$

$$gain = 0.13$$

Nilai $gain$ yang sudah didapat disimpan untuk nantinya dibandingkan dengan $gain$ *probabillity node* lainnya. Selanjutnya adalah menghitung $similarity$ dan $gain$ dari *probabillity node* selanjutnya.

4. Thalac < 123

Untuk menentukan *probabillity node* ini kita melakukan sorting pada nilai numerik kemudian mengambil nilai tengah dari dua nilai numerik yang ada. Diawali dari *node* ini, maka *probabillity node* adalah $96 + (\frac{150-96}{2})$ yaitu 123. Kemudian kita buat *probabillity node* thalac<123. Gambar dari probabillity node ini dapat dilihat pada Gambar 3.13 di bawah ini.



Gambar 3. 13 Tree Untuk Node Thalac<123

Kita menghitung nilai $similarity$ dari leaf sebelah kiri yang hanya berisi satu nilai $residual$. Perhitungan $Left_{similarity}$ dapat dilihat di bawah ini.

$$Left_{similarity} = \frac{(-0.5)^2}{\big(0.5 \times (1-0.5)\big) + 0}$$

$$Left_{similarity} = \frac{0.25}{0.25}$$

$$Left_{similarity} = 1$$

Setelah mendapatkan nilai $Left_{similarity}$, maka kita juga menghitung nilai $Right_{similarity}$ yang berisi empat nilai $residual$ seperti yang terlihat pada Gambar 3.13 di atas. Perhitungan $Right_{similarity}$ dapat dilihat seperti di bawah ini.

$$Right_{similarity} = \frac{(0.5 - 0.5 + 0.5 + 0.5)^2}{\Big(\big(0.5 \times (1-0.5)\big) \times 4\Big) + 0}$$

$$Right_{similarity} = \frac{(0.5 + 0.5)^2}{0.25 \times 4}$$

$$Right_{similarity} = \frac{1}{1}$$

$$Right_{similarity} = 1$$

Kemudian setelah mendapatkan $Left_{similarity}$ dan $Right_{similarity}$ pada *probabillity node* ini, selanjutnya adalah mencari nilai $gain$ dengan cara menjumlahkan nilai $similarity$ dikurangi dengan nilai $Root_{similarity}$ seperti yang dapat dilihat pada perhitungan di bawah ini.

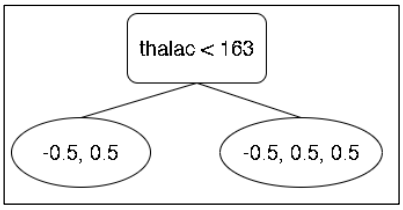
$$gain = Left_{similarity} + Right_{similarity} - Root_{similarity}$$

$$gain = 1 + 1 - 0.2$$

$$gain = 1.8$$

5. Thalac < 163

Untuk menentukan *probabillity node* ini, maka kita mencari nilai tengah dari kedua nilai numerik 150 dan 174. Oleh karena itu dilakukan perhitungan $150 + (\frac{174-150}{2})$ dan mendapatkan nilai 163 sebagai batas *node*. Gambar *probabillity node* dapat dilihat pada Gambar 3.14 di bawah ini.



Gambar 3. 14 Tree Untuk Node Thalac<163

Pertama kita mencari nilai $similarity$ yang terdapat pada *leaf* sebelah kiri yang berisi dua nilai $residual$ yaitu -0.5 dan 0.5. perhitungan $Left_{similarity}$ dapat dilihat di bawah ini.

$$Left_{similarity} = \frac{(-0.5 + 0.5)^2}{\Big(\big(0.5 \times (1 - 0.5)\big) \times 2\Big) + 0}$$

$$Left_{similarity} = \frac{0^2}{0.25 \times 2}$$

$$Left_{similarity} = 0$$

Selanjutnya kita mencari nilai $similarity$ untuk leaf sebelah kanan. Perhitungan $Right_{similarity}$ dapat dilihat seperti di bawah ini.

$$Right_{similarity} = \frac{(-0.5 + 0.5 + 0.5)^2}{\Big(\big(0.5 \times (1 - 0.5)\big) \times 3\Big) + 0}$$

$$Right_{similarity} = \frac{0.5^2}{0.25 \times 3}$$

$$Right_{similarity} = \frac{0.25}{0.75}$$

$$Right_{similarity} = 0.33$$

Setelah kita mendapatkan kedua nilai $similarity$ pada masing-masing *leaf*, maka selanjutnya kita bisa mencari $gain$ untuk *probabillity node* ini. Perhitungan $gain$ dapat dilihat di bawah ini.

$$gain = Left_{similarity} + Right_{similarity} - Root_{similarity}$$

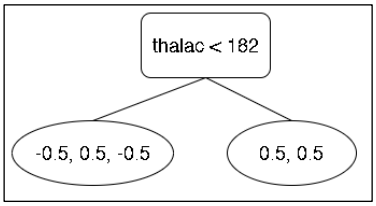
$$gain = 0 + 0.33 - 0.2$$

$$gain = 0.13$$

Setelah mendapatkan nilai $gain$, kita simpan nilai $gain$ pada *probabillity node* ini kemudian dilanjutkan mencari nilai $gain$ pada *probabillity node* lainnya.

6. Thalac < 182

Untuk menentukan batas pada *node* ini, kita mencari nilai tengah dari kedua nilai numerik 174 dan 190 dengan melakukan perhitungan $174 + (\frac{190-174}{2})$ dan didapatkan hasil 182 sebagai batas *probabillity node* ini. Gambar *probabillity node* dapat dilihat pada Gambar 3.15 di bawah ini.



Gambar 3. 15 Tree Untuk Node Thalac<182

Pertama kita mencari nilai $Left_{similarity}$ yang berisi tiga nilai $residual$ seperti yang terlihat pada Gambar 3.15 di atas. Perhitungan $Left_{similarity}$ dapat dilihat di bawah ini.

$$Left_{similarity} = \frac{(-0.5 + 0.5 - 0.5)^2}{\Big(\big(0.5 \times (1 - 0.5)\big) \times 3\Big) + 0}$$

$$Left_{similarity} = \frac{-0.5^2}{0.25 \times 3}$$

$$Left_{similarity} = \frac{0.25}{0.75}$$

$$Left_{similarity} = 0.33$$

Setelah mendapatkan nilai $Left_{similarity}$, maka kita mencari $similarity$ untuk *leaf* selanjutnya yang ada di sebelah kanan. Perhitungan $Right_{similarity}$ dapat dilihat seperti di bawah ini.

$$Right_{similarity} = \frac{(0.5 + 0.5)^2}{\Big(\big(0.5 \times (1 - 0.5)\big) \times 2\Big) + 0}$$

$$Right_{similarity} = \frac{1^2}{0.25 \times 2}$$

$$Right_{similarity} = \frac{1}{0.5}$$

$$Right_{similarity} = 2$$

Setelah mendapatkan nilai $similarity$ dari kedua *leaf*, selanjutnya menjumlahkan $similarity$ tersebut dan dikurangi $Root_{similarity}$ untuk mendapatkan $gain$. Perhitungan nilai $gain$ pada *probabillity node* ini dapat dilihat di bawah ini.

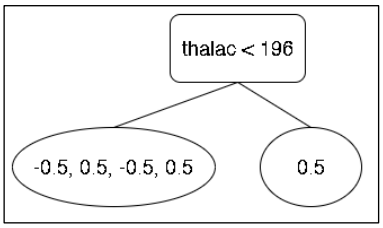
$$gain = Left_{similarity} + Right_{similarity} - Root_{similarity}$$

$$gain = 0.33 + 2 - 0.2$$

$$gain = 2.13$$

7. Thalac < 196

Untuk menentukan batas *probabillity node* ini, kita melakukan perhitungan $190 + (\frac{202-190}{2})$ untuk mencari nilai tengah dari dua nilai numerik yang ada pada kolom tersebut yaitu 190 dan 202, perhitungan tersebut menghasilkan 196 sebagai batas *node*. Gambar *probabillity node* ini dapat dilihat pada Gambar 3.16 di bawah ini.



Gambar 3. 16 Tree Untuk Node Thalac<196

Pertama kita mencari $similarity$ untuk *leaf* yang ada di sebelah kiri. *Leaf* tersebut berisi empat nilai $residual$ yaitu -0.5, 0.5, -0.5 dan 0.5. Perhitungan $Left_{similarity}$ dapat dilihat di bawah ini.

$$Left_{similarity} = \frac{(-0.5 + 0.5 - 0.5 + 0.5)^2}{\Big((0.5 \times (1 - 0.5)) \times 4\Big) + 0}$$

$$Left_{similarity} = \frac{0^2}{1}$$

$$Left_{similarity} = 0$$

Kemudian setelah mendapatkan nilai $Left_{similarity}$, selanjutnya adalah mencari nilai $similarity$ untuk *leaf* belah kanan yang hanya berisi satu nilai $residual$ yaitu 0.5. Perhitungan $Right_{similarity}$ dapat dilihat seperti di bawah ini.

$$Right_{similarity} = \frac{0.5^2}{\big(0.5 \times (1 - 0.5)\big) + 0}$$

$$Right_{similarity} = \frac{0.25}{0.25}$$

$$Right_{similarity} = 1$$

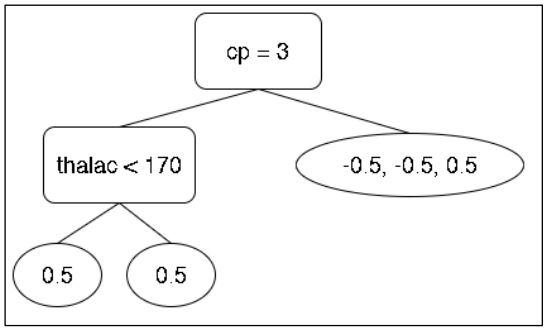
Kemudian setelah mendapatkan nilai $Left_{similarity}$ dan nilai $Right_{similarity}$, maka kita jumlahkan nilai $similarity$ tersebut dan dikurangi dengan nilai $Root_{similarity}$ untuk mendapatkan $gain$ *probabillity node* ini. Perhitungan $gain$ untuk *probabillity node* ini dapat dilihat di bawah ini.

$$gain = Left_{similarity} + Right_{similarity} - Right_{similarity}$$

$$gain = 0 + 1 - 0.2$$

$$gain = 0.8$$

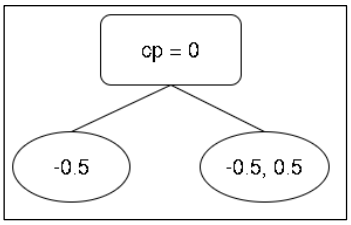
Setelah semua kemungkinan *node* dihitung nilai $gain$, maka node yang memiliki nilai $gain$ yang paling tinggi nantinya yang akan dijadikan *root node*. Sesuai dengan perhitungan di atas, maka yang akan menjadi *root node* adalah cp=3 karena memiliki nilai $gain$ yang paling tinggi yaitu 2.17. Setelah pemilihan *root node* tersebut, maka *tree* sementara yang dapat dibangun dapat dilihat pada Gambar 3.17 di bawah ini.



Gambar 3. 17 Tree Sementara Yang Terbentuk

Kemudian, kita masih melakukan hal yang sama untuk perhitungan *probabillity node* yang tersisa pada *leaf* sebelah kanan yang terlihat pada Gambar 3.17 di atas. *Probabillity node* tersebut berisi tiga nilai $residual$ yang tersisa yaitu -0.5, -0.5 dan 0.5. Perhitungan *probabillity node* yang ada pada leaf tersebut dapat dilihat di bawah ini.

1. Cp=0



Gambar 3. 18 Tree Untuk Node Cp=0

Sesuai dengan ketiga nilai $residual$ yang tersisa, yang menjadi *probabillity node* pertama adalah cp=0. Sama seperti perhitungan untuk menentukan *root node* sebelumnya, kita menentukan nilai $similarity$ untuk *leaf* sebelah kiri terlebih dahulu. Perhitungan $Left_{similarity}$ dapat dilihat di bawah ini.

$$Left_{similarity} = \frac{-0.5^2}{\big(0.5 \times (1-0.5)\big) + 0}$$

$$Left_{similarity} = \frac{0.25}{0.25}$$

$$Left_{similarity} = 1$$

Kemudian kita mencari $similarity$ dari *leaf* selanjutnya, yaitu $Right_{similarity}$ yang memiliki sisa dua nilai $residual$. Perhitungan $Right_{similarity}$ dapat dilihat di bawah ini.

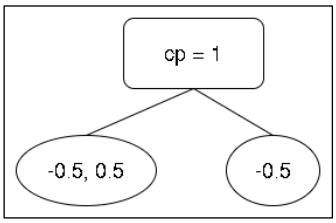
$$Right_{similarity} = \frac{(-0.5 + 0.5)^2}{\left(\left(0.5 \times (1 - 0.5)\right) \times 2\right) + 0}$$
$$Right_{similarity} = \frac{0^2}{0.25 \times 2}$$
$$Right_{similarity} = 0$$

Setelah mendapatkan $Left_{similarity}$ dan $Right_{similarity}$, selanjutnya kita menghitung nilai $gain$ dari *probabillity node* ini. $Root_{similarity}$ yang digunakan dalam perhitungan ini adalah $Right_{similarity}$ dari *node* cp=3 yang sudah menjadi *root node*, dapat dilihat pada Gambar 3.17 *tree* sementara yang dibangun di atas. Perhitungan mencari $gain$ adalah sebagai berikut.

$$gain = Left_{similarity} + Right_{similarity} - Root_{similarity}$$
$$gain = 1 + 0 - 0.33$$
$$gain = 0.67$$

Kemudian $gain$ yang sudah didapat kita simpan dahulu, dilanjutnya mencari $gain$ pada *probabillity node* selanjutnya. Nilai $gain$ dari masing-masing *probabillity node* nantinya akan dibandingkan, *probabillity node* yang memiliki $gain$ tertinggi yang akan menjadi *node*.

2. Cp=1



Gambar 3. 19 Tree untuk node cp=1

Probabillity node selanjutnya adalah cp=1. Pertama kita menghitung $similarity$ dari *leaf* yang berada di kiri yang berisi dua nilai $residual$ yaitu -0,5 dan 0.5. Perhitungan $Left_{similarity}$ dapat dilihat di bawah ini.

$$Left_{similarity} = \frac{(-0.5 + 0.5)^2}{\left(\left(0.5 \times (1 - 0.35)\right) \times 2\right) + 0}$$
$$Left_{similarity} = \frac{0^2}{0.25 \times 2}$$
$$Left_{similarity} = 0$$

Kemudian dilanjutkan dengan mencari nilai $similarity$ dari *leaf* sebelah kanan. Perhitungan nilai $Right_{similarity}$ dapat dilihat seperti di bawah ini.

$$Right_{similarity} = \frac{-0.5^2}{\left(0.5 \times (1 - 0.5)\right) + 0}$$

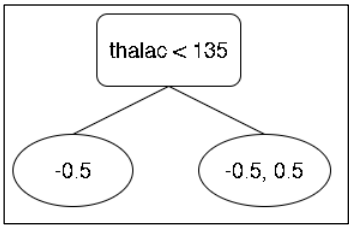
$$Right_{similarity} = \frac{0.25}{0.25}$$
$$Right_{similarity} = 1$$

Setelah mendapatkan kedua nilai $similarity$, langkah selanjutnya adalah mencari nilai $gain$ dari *probabillity node* ini dengan cara menajumlahkan $similarity$ dan dikurangi dengan $Root_{similarity}$ pada *tree* yang telah dibuat. Perhitungan $gain$ pada *probabillity node* ini dapat dilihat di bawah ini.

$$gain = Left_{similarity} + Right_{similarity} - Root_{similarity}$$
$$gain = 0 + 1 - 0.33$$
$$gain = 0.67$$

Setelah mendapatkan $gain$ untuk *probabillity node* ini, maka dilanjutkan dengan mencari nilai $gain$ untuk *probabillity node* yang tersisa. Nantinya jika semua *probabillity node* sudah diketahui nilai $gain$nya, maka $gain$ akan dibandingkan dan dicari $gain$ paling tinggi untuk dijadikan *node*.

3. Thalac<135



Gambar 3. 20 Tree Untuk Node Thalac<135

Untuk menentukan batas pada *probabillity node* ini, nilai numerik pada kolom thalac dari ketiga $residual$ tersebut kita urutkan yaitu 96, 174, dan 202. Kemudian dari masing-masing urutan kita cari nilai tengah diantara dua nilai numerik. Sehingga pada *probabbility node* ini dihitung $96 + (\frac{174-94}{2})$ dan didapatkan nilai 135 sebagai batas *probabillity node*. Gambar *probabillity node* ini dapat dilihat pada Gambar 3.20 di atas. Selanjutnya kita mencari nilai $similarity$ dari masing-masing *leaf*, diawali dari *leaf* sebelah kiri. Perhitungan $Left_{similarity}$ dapat dilihat di bawah ini.

$$Left_{similarity} = \frac{-0.5^2}{\big(0.5 \times (1 - 0.5)\big) + 0}$$
$$Left_{similarity} = \frac{0.25}{0.25}$$
$$Left_{similarity} = 1$$

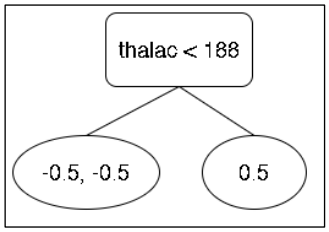
Kemudian dilanjutkan dengan mencari $similarity$ pada *leaf* sebelah kanan yang memiliki dua nilai $residual$. Perhitungan $Right_{similarity}$ dapat dilihat pada operasi matematika di bawah ini.

$$Right_{similarity} = \frac{(-0.5 + 0.5)^2}{\left(\left(0.5 \times (1 - 0.5)\right) \times 2\right) + 0}$$
$$Right_{similarity} = \frac{0^2}{0.25 \times 2}$$
$$Right_{similarity} = 0$$

Setelah semua nilai $similarity$ diketahui, kemudian dilanjutkan mencari nilai $gain$ untuk *probabillity node* ini. Caranya adalah menjumlahkan semua nilai $similarity$ yang tadi sudah dicari dan dikurangi dengan $Root_{similarity}$ yang berisi nilai $Right_{similarity}$ pada level *root node*. Perhitungan mencari $gain$ dapat dilihat di bawah ini.

$$gain = Left_{similarity} + Right_{similarity} - Root_{similarity}$$
$$gain = 1 + 0 - 0.33$$
$$gain = 0.67$$

4. Thalac<188



Gambar 3. 21 Tree Untuk Node Thalac<188

Probabillity node terakhir pada level ini adalah *probabillity node* dengan batas 188. Batas tersebut didapat dengan menghitung nilai tengah diantara dua nilai numerik dengan cara $174 + (\frac{202-174}{2})$ dan mendapatkan nilai 188 sebagai batas. Kemudian digambarkan *tree* berdasarkan batas yang ada dengan *leaf* berupa $residual$ yang tersisa, seperti yang terlihat pada Gambar 3.21 di atas. Selanjutnya kita mencari $similarity$ untuk masing-masing *leaf*, diawali dari *leaf* sebelah kiri yang memiliki dua nilai $residual$. Perhitungan $Left_{similarity}$ dapat dilihat di bawah ini.

$$Left_{similarity} = \frac{(-0.5 - 0.5)^2}{\left(0.5 \times \left((1 - 0.5)\right) \times 2\right) + 0}$$
$$Left_{similarity} = \frac{-1^2}{0.25 \times 2}$$
$$Left_{similarity} = \frac{1}{0.5}$$
$$Left_{similarity} = 2$$

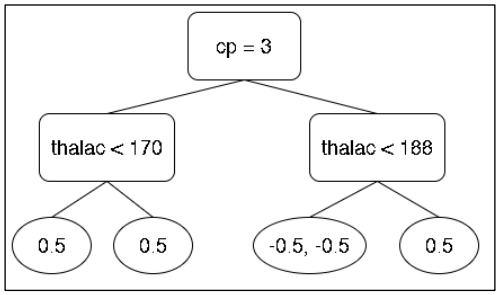
Kemudian setelah kita mendapatkan nilai $Left_{similarity}$, langkah selanjutnya adalah mencari nilai $similarity$ untuk *leaf* sebelah kanan. Perhitungan mencari $Right_{similarity}$ dapat dilihat seperti di bawah ini.

$$Right_{similarity} = \frac{0.5^2}{(0.5 \times (1 - 0.5))}$$
$$Right_{similarity} = \frac{0.25}{0.25}$$
$$Right_{similarity} = 1$$

Setelah mendapatkan semua nilai $similarity$, kita mencari nilai $gain$ untuk *probabillity node* ini. Perhitungan nilai $gain$ dicari dengan menjumlahkan semua nilai $similarity$ dan dikurangi dengan nilai $Root_{node}$ pada level ini. Perhitungan nilai $gain$ dapat dilihat di bawah ini.

$$gain = Left_{similarity} + Right_{similarity} - Root_{similarity}$$
$$gain = 2 + 1 - 0.33$$
$$gain = 2.67$$

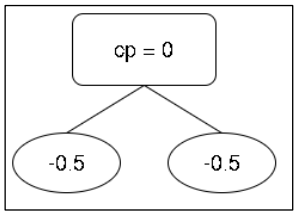
Setelah mengetahui semua nilai $gain$ pada semua *probabillity node* yang ada untuk level ini. Nilai $gain$ dari masing-masing *probabillity node* untuk level ini akan dibandingkan, dan *probabillity node* yang memiliki nilai $gain$ paling tinggi akan menjadi *node* untuk level ini. Sesuai dengan perhitungan nilai $gain$ di atas, maka *node* yang akan dipilih adalah thalac<188 karena memiliki nilai $gain$ paling tinggi diantara *probabillity node* lainnya, yaitu sebesar 2.67. Setelah pemilihan *node* untuk level ini, maka kita dapat menggambarkan *tree* sementara yang bisa dibangun. *Tree* sementara yang sudah dibangun berdasarkan perhitungan di atas dapat dilihat pada Gambar 3.22 di bawah ini.



Gambar 3. 22 Tree Sementara Yang Sudah Dibuat

Berdasarkan Gambar 3.22 di atas, terlihat bahwa node thalac<188 masih memiliki dua sisa nilai $residual$ yaitu -0.5 dan -0.5. Kita masih akan melakukan perhitungan lagi pada *probabillity node* yang ada untuk level atau kedalaman *tree* selanjutnya. Beberapa *probabillity node* yang ada untuk level ini berdasarkan data yang kita gunakan untuk proses *training* adalah cp=0, cp=1, dan thalac<135. Dari *probabillity node* tersebut nantinya kita lakukan perhitungan untuk menemukan masing-masing nilai $gain$, dan *probabillity node* yang memiliki nilai $gain$ paling tinggi akan menajdi *node*, dan dengan dipilihnya *node* tersebut *tree* pertama sudah selesai dibangun dengan tiga level kedalaman. Berikut perhitungan untuk mencari nilai $gain$ pada masing-masing *probabillity node* untuk level ini dapat dilihat di bawah ini.

1. Cp=0



Gambar 3. 23 Tree Untuk Node Cp=0

Probabillity node pertama yang ada untuk level ini adalah cp=0. Pertama kita hitung nilai $similarity$ untuk masing-masing *leaf* yang hanya memiliki sisa satu nilai $residual$. Diawali dengan perhitungan $similarity$ pada *leaf* sebelah kiri atau disebut $Left_{similarity}$. Perhitungan pada $Left_{similarity}$ dapat dilihat di bawah ini.

$$Left_{similarity} = \frac{(-0.5)^2}{\big(0.5 \times (1-0.5)\big) + 0}$$
$$Left_{similarity} = \frac{0.25}{0.25}$$
$$Left_{similarity} = 1$$

Setelah kita mendapatkan nilai $Left_{similarity}$, tahap selanjutnya adalah melakukan perhitungan yang sama untuk leaf sebelah kanan atau disebut $Right_{similarity}$. Perhitungan $Right_{similarity}$ dapat dilihat di bawah ini.

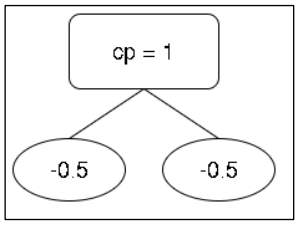
$$Right_{similarity} = \frac{(-0.5)^2}{\big(0.5 \times (1-0.5)\big) + 0}$$
$$Right_{similarity} = \frac{0.25}{0.25}$$
$$Right_{similarity} = 1$$

Setelah kita mendapatkan semua nilai $similarity$ untuk *leaf* yang ada. Selanjutnya adalah menghitung nilai $gain$ pada *probabillity node* ini. Perhitungan $gain$ pada *probabillity node* ini dilakukan dengan menjumlahkan nilai $similarity$ yang sudah kita cari, dan dikurangi dengan $Root_{node}$ yang ada untuk level kedalaman ini. $Root_{node}$ untuk level kedalaman ini adalah $Left_{similarity}$ dari node thalac<188. Perhitungan mencari nilai $gain$ untuk *probabillity node* ini dapat dilihat pada operasi matematika yang ada di bawah ini.

$$gain = Left_{similarity} + Right_{similarity} - Root_{similarity}$$
$$gain = 1 + 1 - 2$$
$$gain = 0$$

Setelah kita mengetahui nilai $gain$ pada *probabillity node* ini, maka kita lanjutkan dengan mencari nilai $gain$ pada *probabillity node* lainnya untuk nantinya dibandingkan dan dicari nilai $gain$ yang paling besar untuk dijadikan *node* untuk level kedalaman ini.

2. Cp=1



Gambar 3. 24 Tree Untuk Node Cp=1

Probabillity node selanjutnya yang akan kita lakukan perhitungan nilai $gain$ adalah *probabillity node* cp=1, sesuai dengan sisa data yang kita gunakan untuk *training*. Pertama kita lakukan perhitungan untuk mencari nilai $similarity$ pada masing-masing *leaf* yang ada. Diawali dari perhitungan *leaf* sebelah kiri atau disebut $Left_{similarity}$ yang dapat dilihat pada operasi matematika di bawah ini.

$$Left_{similairty} = \frac{(-0.5)^2}{\big(0.5 \times (1-0.5)\big) + 0}$$
$$Left_{similarity} = \frac{0.25}{0.25}$$
$$Left_{similarity} = 1$$

Kemudian setelah kita mendapatkan nilai $Left_{similarity}$, maka selanjutnya kita melakukan perhitungan yang sama untuk mencari nilai $gain$ pada *leaf* sebelah kanan atau disebut $Right_{similarity}$. Perhitungan untuk mencari $Right_{similarity}$ dapat dilihat di bawah ini.

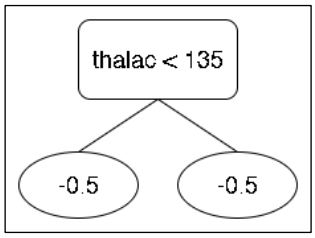
$$Right_{similarity} = \frac{(-0.5)^2}{\big(0.5 \times (1-0.5)\big) + 0}$$
$$Right_{similarity} = \frac{0.25}{0.25}$$
$$Right_{similarity} = 1$$

Setelah kita mengetahui nilai $Right_{similarity}$ dan nilai $Left_{similarity}$, maka kita jumlahkan kedua nilai $similarity$ tersebut dan dikurangi dengan nilai $Root_{nsimilarity}$ yang ada pada level kedalaman diatasnya untuk menemukan nilai $gain$ pada *probabillity node* ini. Perhitungan untuk mencari nilai $gain$ pada *probabillity node* ini dapat dilihat di bawah ini.

$$gain = Left_{similarity} + Right_{similarity} - Root_{similarity}$$
$$gain = 1 + 1 - 2$$
$$gain = 0$$

Selanjutnya kita simpan dahulu nilai $gain$ pada *probabillity node* ini, kita lanjutkan untuk mencari nilai $gain$ pada sisa *probabillity node* yang ada untuk level kedalaman ini.

3. Thalac<135



Gambar 3. 25 Tree Dengan Node Thalac<135

Perhitungan *probabillity node* terakhir adalah untuk thalac<135. Seperti perhitungan pada *probabillity node* sebelumnya, nilai batas untuk *probabillity node* ini didapatkan dengan mencari nilai tengah dari kedua nilai numerik yang ada, kemudian melakukan perhitungan $96 + (\frac{174-96}{2})$. Setelah mendapatkan nilai batas untuk *probabillity node*, maka kita dapat menggambarkan *tree* seperti yang ditampilkan pada Gambar 3.25 di atas. Kemudian kita lakukan perhitungan untuk mencari nilai $similarity$ dari kedua *leaf* yang ada. Diawali dari perhitungan $Left_{similarity}$ yang dapat dilihat di bawah ini.

$$Left_{similarity} = \frac{(-0.5)^2}{\big(0.5 \times (1-0.5)\big) + 0}$$
$$Left_{similarity} = \frac{0.25}{0.25}$$
$$Left_{similarity} = 1$$

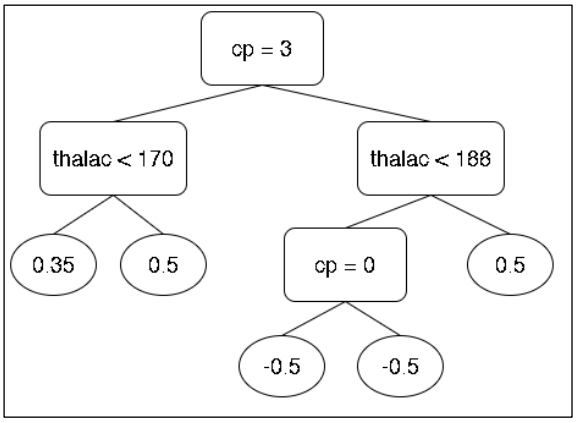
Setelah mengetahui nilai $Left_{similarity}$, maka dilanjutkan dengan mencari nilai $similarity$ pada *leaf* sebelah kanan atau disebut $Right_{similarity}$. Nilai $residual$ yang ada pada leaf sebelah kanan hanya memiliki satu nilai $residual$ yaitu -0.5. Perhitungan mencari nilai $Right_{similarity}$ dapat dilihat di bawah ini.

$$Right_{similarity} = \frac{(-0.5)^2}{\big(0.5 \times (1-0.5)\big) + 0}$$
$$Right_{similarity} = \frac{0.25}{0.25}$$
$$Right_{similarity} = 1$$

Kemudian, setelah kita mengetahui kedua nilai $similarity$ di atas, kita jumlahkan $Left_{similarity}$ dan $Right_{similarity}$ dan dikurangi dengan nilai $Root_{similarity}$ untuk mendapatkan nilai $gain$ pada *probabillity node* ini. Perhitungan untuk mencari nilai $gain$ pada *probabillity node* untuk level kedalam ini dapat dilihat pada operasi matematika di bawah ini.

$$gain = Left_{similarity} + Right_{similarity} - Root_{similarity}$$
$$gain = 1 + 1 - 2$$
$$gain = 0$$

Setelah mengetahui masing-masing nilai $gain$ pada *probabillity node* yang ada untuk level kedalaman ini, kemudian *probabillity node* dengan nilai $gain$ yang paling tinggi akan dipilih untuk menjadi *node*. Namun karena semua *probabillity node* yang sudah kita lakukan perhitungan menghasilkan nilai $gain$ yang sama, yaitu 0. Maka ketiga *probabillity node* di atas boleh dijadikan *node*, dan *node* yang dipilih untuk level kedalaman ini adalah cp=0. Setelah menetapkan *node* untuk level kedalam ini, maka kita dapat menggambar *tree* yang sudah berhasil kita bangun berdasarkan perhitungan yang kita lakukan. Untuk hasil *tree* pertama yang dibangun dapat dilihat pada Gambar 3.26 di bawah ini.



Gambar 3. 26 Hasil Tree Pertama Yang Dibuat

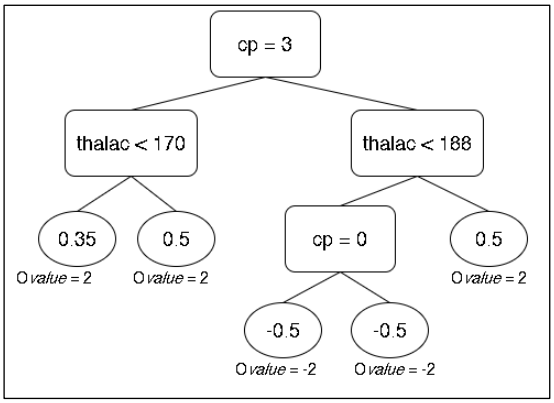
Setelah kita berhasil membangun *tree* pertama. Langkah selanjutnya adalah melakukan pengecekan pada masing-masing *leaf* dan *node* sesuai dengan *flowchart* algoritma Xgboost. Namun, dalam contoh proses *training* ini nilai $cover$ dan $gamma(\gamma)$ kita asumsikan dengan nilai 0, dikarenakan keterbasan contoh data yang digunakan dalam perhitungan manual kali ini. Sehingga tidak ada *leaf* ataupun *node* yang dipangkas. Kemudian dilanjutkan langkah berikutnya yaitu menghitung *output value* (O_{value}) untuk masing-masing *leaf* yang ada pada *tree* yang sudah dibangun. Perhitungan O_{value} pada masing-masing *leaf* dapat dilihat pada operasi matematika di bawah ini, sedangkan hasil perhitungan O_{value} dapat dilihat pada Gambar 3.27 di bawah ini setelah perhitungan. Rumus untuk menghitung O_{value} dapat dilihat pada persamaan 3.14 di bawah ini.

$$O_{value} = \frac{\sum(residual_i)}{\sum[prev\ probability_i \times (1 - prev\ probability_i)] + \lambda} \quad \text{..... (3.14)}$$

Sedangkan perhitungan O_{value} pada masing-masing leaf dapat dilihat di bawah ini.

1. $O_{value} = \frac{0.5}{(0.5 \times (1-0.5)) + 0} = \frac{0.5}{0.25} = 2$
2. $O_{value} = \frac{0.5}{(0.5 \times (1-0.5)) + 0} = \frac{0.5}{0.25} = 2$
3. $O_{value} = \frac{-0.5}{(0.5 \times (1-0.5)) + 0} = \frac{-0.5}{0.25} = -2$
4. $O_{value} = \frac{-0.5}{(0.5 \times (1-0.5)) + 0} = \frac{-0.5}{0.25} = -2$

5. $O_{value} = \frac{0.5}{(0.5 \times (1-0.5))+0} = \frac{0.5}{0.25} = 2$



Gambar 3. 27 Tree Pertama dan O_{Value}

Setelah mengetahui nilai O_{value} dari masing-masing *leaf*. Maka kita dapat menghitung nilai $probabillity$ baru yang nantinya dapat kita gunakan untuk membuat *tree* kedua dengan menghitung nilai $residual$ yang baru dari nilai $probabillity$ yang didapat. Rumus perhitungan $probabillity$ dapat dilihat pada persamaan 3.15 dan persamaan 3.16 di bawah ini dengan nilai $learning\ rate(\varepsilon)$ adalah 0.3.

$\log(odds) = \log\left(\frac{p}{1-p}\right) + \sum[\varepsilon \times O_{valie}]_i$ ..... (3.15)

$probability = \frac{e^{\log(odds)}}{1+ e^{\log(odds)}}$ ..... (3.16)

Keterangan:

$p = 0.5\ (initial\ probability)$

$\log\left(\frac{p}{1-p}\right) = \log\left(\frac{0.5}{1-0.5}\right)$

$\log\left(\frac{p}{1-p}\right) = \log(1)$

$\log\left(\frac{p}{1-p}\right) = 0$

Sedangkan untuk perhitungan nilai $probabillity$ dari setiap data yang digunakan dapat dilihat pada operasi matematika di bawah ini. Urutan pada perhitungan yang ditunjukan di bawah ini sudah disesuaikan dengan urutan data yang digunakan.

1. $\log(odds) = 0 + (0.3 \times 2) = 0.6$

   $probability = \frac{e^{0.6}}{1 + e^{0.6}}$

   $probability = \frac{2.71828^{0.6}}{1 + 2.71828^{0.6}}$

   $probability = \frac{1.822}{2.822}$

   $probability = 0.65$

2. $\log(odds) = 0 + (0.3 \times 2) = 0.6$

   $probability = \frac{e^{0.6}}{1 + e^{0.6}}$

   $probability = 0.65$

3. $\log(odds) = 0 + (0.3 \times -2) = -0.6$

   $probability = \frac{e^{-0.6}}{1 + e^{-0.6}}$

   $probability = \frac{2.71828^{-0.6}}{1 + 2.71828^{-0.6}}$

   $probability = \frac{0.549}{1.549}$

   $probability = 0.35$

4. $\log(odds) = 0 + (0.3 \times -2) = -0.6$

   $probability = \frac{e^{-0.6}}{1 + e^{-0.6}}$

   $probability = 0.35$

5. $\log(odds) = 0 + (0.3 \times 2) = 0.6$

   $probability = \frac{e^{0.6}}{1 + e^{0.6}}$

   $probability = 0.65$

Setelah menentukan semua nilai $probabillity$ untuk masing-masing data, maka kita dapat menghitung nilai $residual$ yang baru. Dan nantinya kita dapat membuat *tree* lagi untuk iterasi berikutnya. Nilai $residual$ yang baru didapatkan dengan menghitung selisih dari target dan $probabillity$ yang baru saja kita hitung. Data yang memuat nilai $probabillity$ dan nilai $residual$ yang baru dapat dilihat pada Tabel 3.11 di bawah ini.

Tabel 3. 11 Data Terbaru Dan Residu

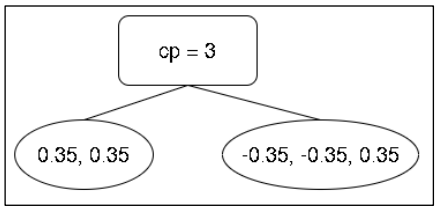
Cp	Thalac	Traget	Probability	Residual
3	150	1	0.65	0.35
3	190	1	0.65	0.35
0	96	0	0.35	-0.35
1	174	0	0.35	-0.35
1	202	1	0.65	0.35

Setelah mengetahui nilai $residual$ pada semua data, kemudian kita dapat menggunakan nilai $residual$ tersebut untuk membangun *tree* pada iterasi kedua. Proses yang dilakukan untuk membangun *tree* kedua sama seperti yang dilakukan pada *tree* pertama. Yang pertama dilakukan adalah mengasumsikan semua nilai $residual$ sebagai satu *leaf* yang akan dihitung nilai $similarity$ dan dijadikan sebagai $Root_{similarity}$. Proses perhitungan $Root_{similarity}$ pada *tree* kedua dapat dilihat di bawah ini. Menggunakan nilai $residual$ yang baru yaitu 0.35 dan -0.35.

$$Root_{similarity} = \frac{(0.35 + 0.35 - 0.35 - 0.35 + 0.35)^2}{\Big(\big(0.65 \times (1 - 0.65)\big) \times 3\Big) + \Big(\big(0.35 \times (1 - 0.35)\big) \times 2\Big) + 0}$$
$$= \frac{0.35^2}{0.6825 + 0.455}$$
$$= \frac{0.1225}{1.1375}$$
$$= 0.1076$$

Kemudian kita lanjutkan dengan melakukan perhitungan nilai $gain$ dari *probabillity node* yang ada. *Node* yang memiliki nilai $gain$ paling tinggi akan dijadikan sebagai *root node* untuk *tree* kedua. Perhitungan nilai $gain$ pada semua *probabillity node* dapat dilihat di bawah ini.

1. Cp=3



Gambar 3. 28 Tree Kedua Untuk Node Cp=3

Setelah kita menentukan *probabillity node* dan menggambar seperti yang terlihat pada Gambar 3.28 di atas. Selanjutnya adalah mulai menghitung nilai $similarity$ pada setiap *leaf* yang ada. Berdasarkan pembagian *probabillity node* tersebut, *leaf* kiri memiliki dua nilai $residual$ yaitu 0.35 dan 0.35. sedangkan *leaf* kanan memiliki tiga nilai $residual$ yaitu -0.35, -0.35, dan 0.35. Perhitungan nilai $similarity$ pada *leaf* kiri atau disebut $Left_{similarity}$ dapat dilihat seperti di bawah ini.

$$Left_{similaroty} = \frac{(0.35 + 0.35)^2}{\Big(\big(0.65 \times (1 - 0.65)\big) \times 2\Big) + 0}$$
$$Left_{similarity} = \frac{0.7^2}{0.2275 \times 2} = \frac{0.49}{0.455}$$
$$Left_{similarity} = 1.0769$$

Setelah kita mendapatkan nilai $Left_{similarity}$, selanjutnya kita lakukan perhitungan nilai $similarity$ yang sama untuk *leaf* sebelah kanan. Proses perhitungan $Right_{similarity}$ dapat dilihat pada operasi matematika di bawah ini.

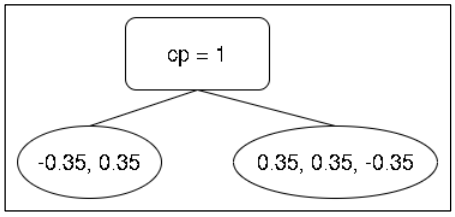
$$Right_{similarity} = \frac{(-0.35 - 0.35 + 0.35)^2}{\big(0.65 \times (1 - 0.65)\big) + \Big(\big(0.35 \times (1 - 0.35)\big) \times 2\Big) + 0}$$
$$Right_{similarity} = \frac{-0.35^2}{0.2275 + 0.455} = \frac{0.1225}{0.6825}$$
$$Right_{similarity} = 0.1794$$

Kemudian, setelah kita mendapatkan semua nilai $similarity$ pada setiap *leaf* yang ada. Kita lakukan perhitungan untuk mencari nilai $gain$ pada *probabillity node* ini dengan cara menjumlahkan nilai $similarity$ tersebut dan dikurangi dengan nilai $Root_{similarity}$ yang sudah kita cari pada perhitungan di atas. Perhitungan nilai $gain$ untuk *probabillity node* ini dapat dilihat di bawah ini.

$$gain = Left_{similarity} + Right_{similarity} - Root_{similarity}$$
$$gain = 1.0769 + 0.1794 - 0.1076$$
$$gain = 1.1487$$

Nilai $gain$ yang sudah kita dapatkan kita simpan terlebih dahulu, kemudian dilanjutkan dengan mencari nilai $gain$ pada *probabillity node* lainnya. Nantinya, setiap nilai $gain$ dari masing-masing *probabillity node* akan dibandingkan, dan *probabillity node* yang memiliki nilai $gain$ paling tinggi akan menjadi *root node*.

2. Cp=1



Gambar 3. 29 Tree Kedua Untuk Node Cp=1

Selanjutnya, *probabillity node* yang akan kita hitung adalan cp=1. Yang akan dilakukan pertama adalah menghitung nilai $similarity$ pada masing-masing *leaf* yang ada. Dari Gambar 3.29 di atas dapat dilihat bahwa *leaf* sebelah kiri memiliki dua nilai $residual$ yaitu -0.35 dan 0.35. sedangkan *leaf* sebelah kanan memiliki tiga nilai $residual$ yaitu 0.35, 0.35, dan -0.35. Kita menghitung nilai $similarity$ untuk *leaf* sebelah kiri terlebih dahulu. Perhitungan $Left_{similarity}$ dapat dilihat di bawah ini.

$$Left_{similarity} = \frac{(-0.35 + 0.35)^2}{\big(0.35 \times (1 - 0.35)\big) + \big(0.65 \times (1 - 0.65)\big) + 0}$$
$$Left_{similarity} = \frac{0^2}{0.2275 + 0.2275}$$
$$Left_{similarity} = 0$$

Setelah kita mendapatkan nilai $Left_{similarity}$ dari hasil perhitungan di atas, kita melakukan perhitungan yang sama lagi untuk *leaf* yang ada di sebelah kanan yang memiliki tiga nilai $residual$. Perhitungan $Right_{similarity}$ dapat dilihat pada operasi matematika di bawah ini.

$$Right_{similarity} = \frac{(0.35 + 0.35 - 0.35)^2}{\Big(\big(0.65 \times (1 - 0.65)\big) \times 2\Big) + \big(0.35 \times (1 - 0.35)\big) + 0}$$

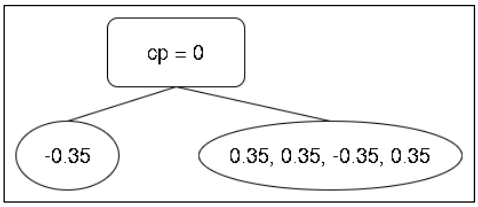
$$Right_{similarity} = \frac{0.35^2}{0.455 + 0.2275} = \frac{0.1225}{0.6825}$$
$$Right_{similarity} = 0.1794$$

Setelah mendapatkan semua nilai $similarity$ pada setiap *leaf* yang ada. Langkah selanjutnya adalah mencari nilai $gain$ untuk *probabillity node* ini. Caranya adalah dengan menjumlahkan nilai $similarity$ dan dikurangi dengan nilai $Root_{similarity}$ yang sudah kita cari pada perhitungan sebelumnya. Proses perhitungan nilai $gain$ pada *probabillity node* ini dapat dilihat di bawah ini.

$$gain = Left_{similarity} + Right_{similarity} - Root_{similarity}$$
$$gain = 0 + 0.1794 - 0.1076$$
$$gain = 0.0718$$

Setelah kita mendapatkan nilai $gain$ pada *probabillity node* ini, selanjutnya kita lakukan perhitungan lagi untuk menentukan nilai $gain$ pada *probabillity node* lainnya.

3. Cp=0



Gambar 3. 30 Tree Kedua Untuk Node Cp=0

Selanjutnya kita melakukan perhitungan untuk menentukan nilai $gain$ pada probabillity node cp=0. Perhitungan untuk mencari nilai $gain$ diawali dengan mencari nilai $similarity$ pada masing-masing leaf yang ada. Perhitungan untuk mencari nilai $Left_{similarity}$ dapat dilihat pada operasi matematika di bawah ini.

$$Left_{similarity} = \frac{(-0.35)^2}{\big(0.35 \times (1 - 0.35)\big) + 0} = \frac{0.1225}{0.2275}$$
$$Left_{similarity} = 0.5384$$

Setelah kita mendapatkan nilai $similarity$ untuk *leaf* sebelah kiri, selanjutnya kita mencari nilai $similarity$ untuk *leaf* sebelah kanan dengan perhitungan yang sama. Perhitungan untuk mencari $Right_{similarity}$ dapat dilihat di bawah ini.

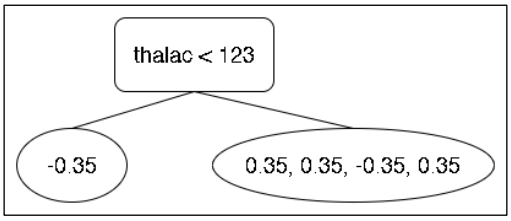
$$Right_{similarity} = \frac{(0.35 + 0.35 - 0.35 + 0.35)^2}{\Big(\big(0.65 \times (1 - 0.65)\big) \times 3\Big) + \big(0.35 \times (1 - 0.35)\big) + 0}$$
$$Right_{similarity} = \frac{(0.35 + 0.35)^2}{0.6825 + 0.2275} = \frac{0.7^2}{0.91} = \frac{0.49}{0.91}$$
$$Right_{similarity} = 0.5384$$

Setelah kita mendapatkan semua nilai $Left_{similarity}$ dan nilai $Right_{similarity}$. Langkah selanjutnya adalah mencari nilai $gain$ untuk *probabillity node* ini. Caranya adalah dengan menjumlahkan kedua nilai $Left_{similarity}$ dan nilai $Right_{similarity}$ dan dikurangi dengan nilai $Root_{similarity}$ untuk *tree* kedua ini. Perhitungan nilai $gain$ untuk *probabillity node* ini dapat dilihat di bawah ini.

$$gain = Left_{similarity} + Right_{similarity} - Root_{similarity}$$
$$gain = 0.5384 + 0.5384 - 0.1076$$
$$gain = 0.9692$$

Setelah mendapatkan nilai $gain$ dari *probabillity node* ini, kemudian kita lanjutkan dengan mencari nilai $gain$ untuk *probabillity node* lainnya hingga tidak ada lagi *probabillity node* yang tersisa.

4. Thalac<123



Gambar 3. 31 Tree Kedua Untuk Node Thalac<123

Selanjutnya kita menghitung nilai $gain$ untuk *probabillity node* yang ada di kolom thalac. Untuk mencari batas pada *probabillity node* tersebut dilakukan dengan mengurutkan nilai numerik yang ada di kolom thalac, kemudian mencari nilai tengah untuk setiap dua nilai numerik yang ada. Pada *probabillity node* ini batas dicari dengan melakukan perhitungan $96 + (\frac{150-96}{2})$ dan mendapatkan hasil 123 sebagai batas. Gambar *tree* pada *probabillity node* ini dapat dilihat pada Gambar 3.31 di atas. Kemudian kita mencari nilai $similarity$ untuk setiap leaf, diawali dari *leaf* sebelah kiri atau $Left_{similarity}$. Perhitungan $Left_{similarity}$ dapat dilihat di bawah ini.

$$Left_{similarity} = \frac{(-0.35)^2}{\big(0.35 \times (1 - 0.35)\big) + 0} = \frac{0.1225}{0.2275}$$
$$Left_{similarity} = 0.5384$$

Setelah mendapatkan nilai $Left_{similarity}$, kita lakukan perhitungan yang sama untuk mencari nilai $similarity$ pada *leaf* sebelah kanan yang memiliki empat nilai $residual$. Perhitungan untuk mencari nilai $Right_{similarity}$ dapat dilihat di bawah ini.

$$Right_{similarity} = \frac{(0.35 + 0.35 - 0.35 + 0.35)^2}{\Big(\big(0.65 \times (1 - 0.65)\big) \times 3\Big) + \big(0.35 \times (1 - 0.35)\big) + 0}$$
$$Right_{similarity} = \frac{(0.35 + 0.35)^2}{0.6825 + 0.2275} = \frac{0.7^2}{0.91} = \frac{0.49}{0.91}$$

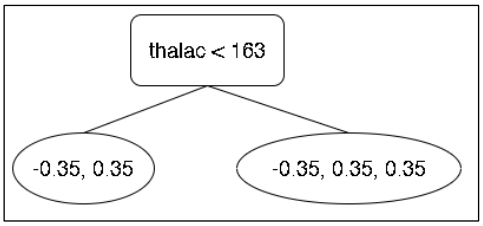
$Right_{similarity} = 0.5384$

Setelah mendapatkan nilai $Left_{similarity}$ dan nilai $Right_{similarity}$, selanjutnya kita menjumlahkan nilai tersebut untuk mendapatkan nilai $gain$ dari *probabillity node* ini. Proses perhitungan nilai $gain$ dari *probabillity node* ini dapat dilihat di bawah ini.

$gain = Left_{similarity} + Right_{similarity} - Root_{similarity}$
$gain = 0.5384 + 0.5384 - 0.1076$
$gain = 0.9692$

Nilai $gain$ dari *probabillity node* ini kita simpan terlebih dahulu dan kita lanjutnkan untuk mencari nilai $gain$ untuk *probabillity node* yang masih tersisa. Nantinya jika semua nilai $gain$ dari *probabillity node* sudah didapatkan, nilai $gain$ tersebut akan dibandingkan dan dicari nilai $gain$ yang tertinggi.

5. Thalac<163



Gambar 3. 32 Tree Kedua Untuk Node Thalac<163

Selanjutnya kita menghitung nilai $gain$ untuk *probabillity node* thalac<163. Untuk menentukan batas tersebut, dilakukan perhitungan untuk mencari nilai tengah dari dua nilai numerik dan mendapat nilai 163 untuk batasnya, perhitungan ini sama seperti menentukan batas pada *probabillity node* sebelumnya atau bahkan *tree* pertama. Kemudian dilanjutkan menghitung nilai $similarity$ dari kedua *leaf* yang ada. Dapat dilihat pada Gambar 3.32 di atas, *leaf* sebelah kiri memiliki dua nilai $residual$ berupa -0.35 dan 0.35 sedangkan *leaf* sebelah kanan memiliki tiga nilai $residual$ berupa -0.35, 0.35, dan 0.35. Pertama yang dilakukan adalah menghitung nilai $similarity$ untuk *leaf* sebelah kiri. Perhitungan $Left_{similarity}$ dapat dilihat di bawah ini.

$$Left_{similarity} = \frac{(-0.35 + 0.35)^2}{\big(0.35 \times (1 - 0.35)\big) + \big(0.65 \times (1 - 0.65)\big) + 0}$$
$Left_{similarity} = 0$

Setelah kita mendapatkan nilai $Left_{similarity}$, selanjutnya kita melakukan perhitungan yang sama untuk mencari nilai $similarity$ pada *leaf* yang ada di sebelah kanan. Perhitungan $Right_{similarity}$ dapat dilihat pada operasi matematika di bawah ini.

$$Right_{similarity} = \frac{(-0.35 + 0.35 + 0.35)^2}{\big(0.35 \times (1 - 0.35)\big) + \Big(\big(0.65 \times (1 - 0.65)\big) \times 2\Big) + 0}$$

$$Right_{similarity} = \frac{0.35^2}{0.2275 + 0.455} = \frac{0.1225}{0.6825}$$

$$Right_{similarity} = 0.1794$$

Kemudian setelah kita mendapatkan kedua nilai $Left_{similarity}$ dan nilai $Right_{similarity}$, maka kita jumlahkan nilai tersebut dan dikurangi dengan nilai $Root_{similarity}$ yang sudah kita hitung pada awal pembuatan *tree* di atas untuk mendapatkan nilai $gain$ dari *probabillity node* ini. Perhitungan untuk mencari nilai $gain$ dapat dilihat di bawah ini.

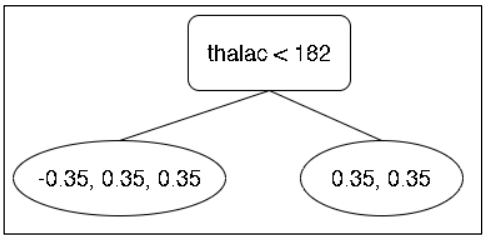
$$gain = Left_{similarity} + Right_{similarity} - Root_{similarity}$$

$$gain = 0 + 0.1794 - 0.1076$$

$$gain = 0.0718$$

Nilai $gain$ yang sudah didapatkan dari perhitungan di atas kita simpah terlebih dahulu untuk nantinya dibandingkan dengan nilai $gain$ dari *probabillity node* lainnya.

6. Thalac<182



Gambar 3. 33 Tree Kedua Untuk Node Thalac<182

Selanjutnya adalah *probabillity node* dengan batas 182. Batas tersebut didapatkan dengan melakukan perhitungan dari dua nilai numerik 174 dan 190. Dari kedua nilai numerik tersebut kita cari nilai tengahnya sehingga perhitungannya menjadi $174 + (\frac{190-174}{2})$ dan menghasilkan 182 sebagai batas. Gambar untuk *probabillity node* ini dapat dilihat pada Gambar 3.33 di atas. Kemudian kita dapat menghitung nilai $gain$ *probabillity node* ini dengan menghitung nilai $similarity$ dari kedua *leaf* terlebih dahulu. Perhitungan nilai $similarity$ untuk *leaf* sebelah kiri atau $Left_{similarity}$ dapat dilihat di bawah ini.

$$Left_{similarity} = \frac{(-0.35 + 0.35 + 0.35)^2}{\big(0.35 \times (1 - 0.35)\big) + \Big(\big(0.65 \times (1 - 0.65)\big) \times 2\Big) + 0}$$

$$Left_{similarity} = \frac{0.35^2}{0.2275 + 0.455} = \frac{0.1225}{0.6825}$$

$$Left_{similarity} = 0.1794$$

Setelah mengetahui nilai $similarity$ untuk *leaf* sebelah kiri, maka kita lakukan perhitungan yang sama untuk mencari nilai $similarity$ pada *leaf* sebelah kanan. Perhitungan untuk mencari $Right_{similarity}$ dapat dilihat pada operasi matematika di bawah ini.

$$Right_{similarity} = \frac{(0.35 + 0.35)^2}{\Big(\big(0.65 \times (1 - 0.65)\big) \times 2\Big) + 0}$$

$$Right_{similarity} = \frac{0.7^2}{0.455} = \frac{0.49}{0.455}$$

$$Right_{similarity} = 1.0769$$

Setelah kita mendapatkan semua nilai $similarity$ baik dari $Left_{similarity}$ maupun $Right_{similarity}$, kita jumlahkan nilai $similarity$ tersebut dikurangi dengan nilai $Root_{similarity}$ yang akan meghasilkan nilai $gain$ untuk *probabillity node* ini. Perhitungan untuk mencari nilai $gain$ dari *probabillity node* ini dapat dilihat pada perhitungan di bawah ini.

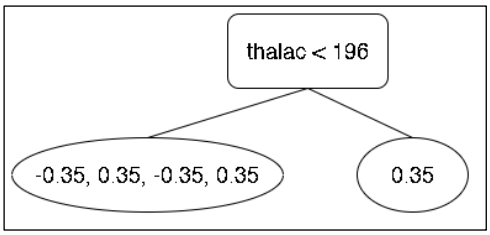
$$gain = Left_{similarity} + Right_{similarity} - Root_{similarity}$$

$$gain = 0.1794 + 1.0769 - 0.1076$$

$$gain = 1.1487$$

Nilai $gain$ milik *probabillity node* yang sudah didapat disimpan terlebuh dahulu, selanjutnya kita mencari nilai $gain$ untuk *probabillity node* lainnya.

7. Thalac<196



Gambar 3. 34 Tree Kedua Untuk Node Thalac<196

Kemudian yang terakhir adalah kita menghitung nilai $gain$ untuk *probabillity node* dengan batas 196. Batas tersebut didapatkan dengan melakukan perhitungan $190 + (\frac{202-190}{2})$ dan mendapatkan nilai 196 sebagai batas. Gambar *probabillity node* yang dibuat dapat dilihat pada Gambar 3.34 di atas. Kemudian kita dapat mencari nilai $similarity$ untuk masing-masing *leaf* yang ada. Perhitungan untuk mencari nilai $Left_{similarity}$ dapat dilihat pada operasi matematika di bawah ini.

$$Left_{similarity} = \frac{(-0.35 - 0.35 + 0.35 + 0.35)^2}{\Big(\big(0.35 \times (1 - 0.35)\big) \times 2\Big) + \Big(\big(0.65 \times (1 - 0.65)\big) \times 2\Big) + 0}$$

$$= 0$$

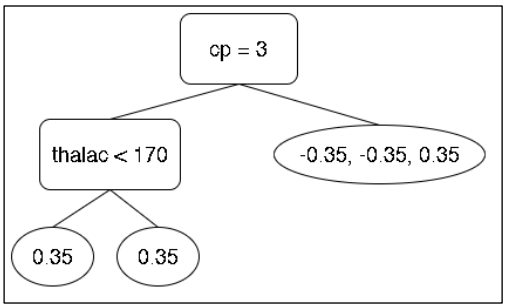
Setelah kita mendapatkan nilai $Left_{similarity}$, selanjutnya kita melakukan perhitungan yang sama untuk mencari $Right_{similarity}$. Perhitungan $Right_{similarity}$ dapat dilihat di bawah ini.

$$Right_{similarity} = \frac{(0.35)^2}{\big(0.65 \times (1 - 0.65)\big) + 0}$$
$$Right_{similarity} = \frac{0.1225}{0.2275}$$
$$Right_{similarity} = 0.5384$$

Setelah semua nilai $similarity$ diketahui, kita dapat menjumlahkan nilai $similarity$ tersebut dan dikurangi dengan nilai $Root_{similarity}$ untuk mendapatkan nilai $gain$ dari *probabillity node* ini. Perhitungan untuk mencari nilai $gain$ pada *probabillity node* ini dapat dilihat pada operasi matematika di bawah ini.

$$gain = Left_{similarity} + Right_{similarity} - Root_{similarity}$$
$$gain = 0 + 0.5384 - 0.1076$$
$$gain = 0.4308$$

Setelah semua nilai $gain$ kita dapatkan dari perhitungan masing-masing *probabillity node* yang ada. *Node* yang memiliki nilai $gain$ paling tinggi akan dijadikan sebagai *root node*. Sehingga, sesuai dengan perhitungan nilai $gain$ diatas, maka yang akan dijadikan *root node* adalah *node* cp=3. Sehingga setelah dipilih *probabillity node* mana yang akan menjadi *root node*, maka *tree* kedua sementara yang dibangun dapat dilihat pada gambar 3.35 di bawah ini.



Gambar 3. 35 Hasil Tree Kedua Sementara Yang Terbentuk

Kemudian, setelah kita menentukan *root node* dan menggambarkan *tree* yang berhasil dibangun seperti yang terlihat pada Gambar 3.35 di atas. Selanjutnya kita akan melakukan perhitungan yang sama lagi untuk menentukan *node* pada level kedalaman selanjutnya, pada Gambar 3.35 di atas terlihat *leaf* sebelah kanan *root node* masih memiliki sisa tiga nilai $residual$. Dari ketiga nilai $residual$ tersbeut akan dipecah untuk mendapatkan *node* yang tepat dengan melakukan perhitungan nilai $gain$ untuk setiap *probabillity node* yang ada. Perhitungan dan $Root_{similarity}$ untuk ketiga $residual$ tersebut dapat dilihat di bawah ini.

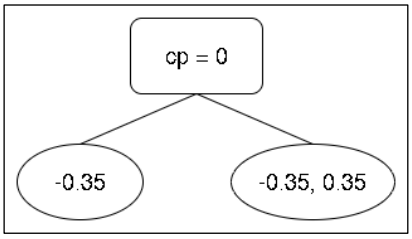
$$Root_{similarity} = \frac{(-0.35 - 0.35 + 0.35)^2}{\Big(\big(0.35 \times (1 - 0.35)\big) \times 2\Big) + \big(0.65 \times (1 - 0.65)\big) + 0}$$

$$Root_{similarity} = \frac{-0.35^2}{0.455 + 0.2275} = \frac{0.1225}{0.6825}$$

$$Root_{similarity} = 0.1794$$

Setelah kita mengetahui nilai $Root_{similarity}$ yang dapat dilihat dari perhitungan di atas. Selanjutnya kita melakukan perhitungan untuk mencari nilai $gain$ dari setiap *probabillity node* yang ada. Masing-masing *probabillity node* dan perhitungan untuk mencari nilai $gain$ dapat dilihat di bawah ini.

1. Cp=0



Gambar 3. 36 Tree Kedua Untuk Node Cp=0

Probabillity node pertama yang ada dalam nilai $residual$ di atas adalah cp=0. Pertama yang kita lakukan adalah mencari nilai $similarity$ untuk setiap *leaf* yang ada. Pertama adalah *leaf* sebelah kiri atau disebut $Left_{similariry}$ perhitungan untuk mencari nilai $Left_{similarity}$ dapat dilihat di bawah ini.

$$Left_{similarity} = \frac{-0.35^2}{\big(0.35 \times (1 - 0.35)\big) + 0} = \frac{0.1225}{0.2275}$$

$$Left_{similarity} = 0.5384$$

Setelah menentukan nilai $Left_{similarity}$ kita dapat melanjutkan perhitungan yang sama yaitu untuk mencari nilai $similarity$ pada *leaf* sebelah kanan atau disebut $Right_{similarity}$. Perhitungan $Right_{similarity}$ dapat dilihat di bawah ini.

$$Right_{similarity} = \frac{(-0.35 + 0.35)^2}{\big(0.35 \times (1 - 0.35)\big) + \big(0.65 \times (1 - 0.65)\big) + 0}$$

$$Right_{similarity} = 0$$

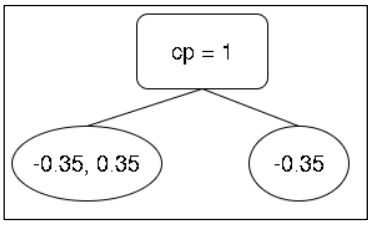
Setelah kita mendapatkan semua nilai $similarity$, kedua nilai tersebut dapat kita jumlahkan untuk mendapatkan nilai $gain$ dari *probabillity node* ini. Perhitungan nilai $gain$ untuk *probabillity node* ini dapat dilihat di bawah ini.

$$gain = Left_{similarity} + Right_{similarity} - Root_{similarity}$$

$$gain = 0.5384 + 0 - 0.1794$$

$$gain = 0.359$$

2. Cp=1



Gambar 3. 37 Tree Kedua Untuk Node Cp=1

Kemudian *probabillity node* berikutnya adalah *probabillity node* cp=1, seseuai dengan sisa nilai $residual$ yang ada di atas. Pertama adalah mencari nilai $similarity$ untuk setiap *leaf* yang ada, mulai dari *leaf* sebelah kiri atau $Left_{similarity}$ yang memiliki dua nilai $residual$ yaitu -0.35, dan 0.35 kemudian dilanjutkan *leaf* sebelah kanan atau $Right_{similarity}$ yang memiliki satu sisa nilai $residual$ yaitu -0.35. Perhitungan untuk mencari $Left_{similarity}$ dapat dilihat pada operasi matematika di bawah ini.

$$Left_{similarity} = \frac{(-0.35 + 0.35)^2}{\big(0.35 \times (1 - 0.35)\big) + \big(0.65 \times (1 - 0.65)\big) + 0}$$
$$Left_{similarity} = 0$$

Setelah kita berhasil menghitung nilai $Similarity$ untuk *leaf* sebelah kiri, maka kita lakukan perhitungan yang sama untuk menentukan nilai $similarity$ untuk *leaf* sebelah kanan. Perhitungan untuk mencari nilai $Right_{similarity}$ dapat dilihat di bawah ini.

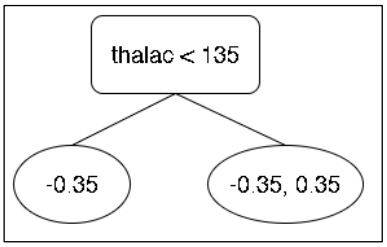
$$Right_{similarity} = \frac{-0.35^2}{\big(0.35 \times (1 - 0.35)\big) + 0} = \frac{0.1225}{0.2275}$$
$$Right_{similarity} = 0.5384$$

Kemudian setelah kita mendapatkan semua nilai $similarity$ baik dari *leaf* sebelah kiri dan *leaf* sebelah kanan, kita dapat menjumlahkan *leaf* tersebut dan mengurangi dengan nilai $Root_{similarity}$ untuk mendapatkan nilai $gain$ dari *probabillity node* ini. Perhitungan untuk mencari nilai $gain$ adalah seperti yang dapat dilihat di bawah ini.

$$gain = Left_{similarity} + Right_{similarity} - Root_{similarity}$$
$$gain = 0 + 0.5384 - 0.1794$$
$$gain = 0.359$$

Setlah semua nilai $similarity$ dan nilai $gain$ untuk probabillity node ini ditemukan, simpan ilai $gain$ untuk probabillity node ini, nilai $gain$ tersebut nantiny akan kita bandingkan dengan nilai $gain$ milik probabillity node lainnya. Selanjutnya kita lanjutkan untuk mencari nilai $gain$ untuk probabillity node lainnya yang masih tersisa.

3. Thalac<135



Gambar 3. 38 Tree Kedua Untuk Node Thalac<135

Probabillity node selanjutnya adalah thalac<135. Cara menentukan batas untuk *probabillity node* ini adalah melakukan perhitungan pada dua nilai numerik yang ada pada kolom tersebut dan mencari nilai tengahnya. Untuk mencari nilai tengahnya dilakukan perhitungan $96 + (\frac{174-96}{2})$ dan mendapatkan nilai 135 sebagai batas *probabillity node* ini. Untuk mencari nilai $gain$ untuk *probabillity node* ini diawali dengan mencari nilai $similarity$ masing-masing *leaf* yang ada. Pertama adalah mencari nilai $Left_{similarity}$ untuk *leaf* sebelah kiri. Perhitungan untuk mencari nilai $Left_{similarity}$ dapat dilihat di bawah ini.

$$Left_{similarity} = \frac{-0.35^2}{\big(0.35 \times (1 - 0.35)\big) + 0} = \frac{0.1225}{0.2275}$$
$$Left_{similarity} = 0.5384$$

Setelah kita mendapatkan nilai $similarity$ untuk *leaf* sebelah kiri, maka selanjutnya kita lakukan perhitungan yang sama untuk *leaf* yang berada di sebelah kanan atau disebut $Right_{similarity}$. Perhitungan untuk mencari nilai $Right_{similarity}$ dapat dilihat di bawah ini.

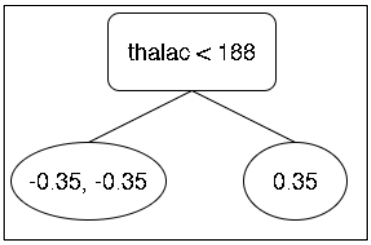
$$Right_{similarity} = \frac{(-0.35 + 0.35)^2}{\big(0.35 \times (1 - 0.35)\big) + \big(0.65 \times (1 - 0.65)\big) + 0}$$
$$Right_{similarity} = 0$$

Kemudian apabila kedua nilai $similarity$ dari kedua *leaf* sudah ditemukan, maka kita dapat menjumlahkan nilai $similarity$ tersebut dan mengurangi dengan nilai $Root_{similarity}$ untuk mendapatkan nilai $gain$ dari *probabillity node* ini. Perhitungan untuk mencari nilai $gain$ dapat dilihat di bawah ini.

$$gain = Left_{similarity} + Right_{similarity} - Root_{similarity}$$
$$gain = 0.5384 + 0 - 0.1794$$
$$gain = 0.359$$

Setelah nilai $gain$ pada *probabillity node* ini ditemukan, nilai $gain$ tersebut disimpan dan kita melanjutkan untuk mencari nilai $gain$ dari sisa *probabillity node* yang ada. Jika semua nilai $gain$ sudah ditemukan, maka kita dapat membandingkan nilai $gain$ tersebut untuk menemukan *probabillity node* yang dipilih untuk menjadi *node*.

4. Thalac<188



Gambar 3. 39 Tree Kedua Untuk Node Thalac<188

Probabillity node terakhir yang ada pada sisa nilai $residual$ di atas adalah probabillity node untuk batas thalac<188. Untuk menemukan batas probabillity node tersebut, dilakukan perhitungan untuk menemukan nilai tengah dari dua nilai numerik yang ada pada data sisa $residual$ yang ada. Untuk menemukan batas probabillity node, dilakukan perhitungan $174 + (\frac{202-174}{2})$ dan mendapatkan nilai 188 sebagai batas dari probabillity node ini. Gambar dari probabillity node ini dapat dilihat pada Gambar 3.39 di atas. Pertama yang kita lakukan adalah mencari nilai $similarity$ untuk leaf sebelah kiri. Perhitungan untuk mencari nilai $Left_{similarity}$ dapat dilihat di bawah ini.

$$Left_{similarity} = \frac{(-0.35 - 0.35)^2}{\Big(\big(0.35 \times (1 - 0.35)\big) \times 2\Big) + 0} = \frac{0.49}{0.455}$$

$$Left_{similarity} = 1.0769$$

Setelah menemukan nilai $Left_{similarity}$, maka kita lakukan perhitungan yang sama untuk mencari nilai $similarity$ untuk *leaf* yang berada di sebelah kanan yang hanya memiliki satu nilai $residual$ berdasarkan Gambar 3.39 di atas. Perhitungan untuk mencari nilai $Right_{similarity}$ dapat dilihat di bawah ini.

$$Right_{similarity} = \frac{0.35^2}{\big(0.65 \times (1 - 0.65)\big) + 0} = \frac{0.1225}{0.2275}$$

$$Right_{similarity} = 0.5384$$

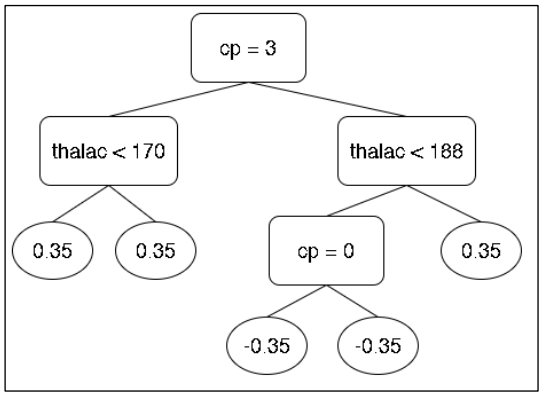
Kemudian dapat dilanjutkan untuk mencari nilai $gain$ dari *probabillity node* ini apabila sudah mendapatkan semua nilai $similarity$ yang ada. Perhitungan untuk mencari nilai $gain$ dari *probabillity node* ini dapat dilihat di bawah ini.

$$gain = left_{similarity} + Right_{similarity} - Root_{similarity}$$

$$gain = 1.0769 + 0.5384 - 0.1794$$

$$gain = 1.4359$$

Setelah nilai $gain$ dari masing-masing *probabillity node* ditemukan, maka kita membandingkan masing-masing nilai gain, dan *probabillity node* yang memiliki nilai $gain$ paling tinggi akan menjadi *node*. Maka *node* yang memiliki $gain$ tertinggi adalah thalac<188. Sehingga hasil *tree* kedua yang dibangun dapat dilihat pada Gambar 3.40 di bawah ini.

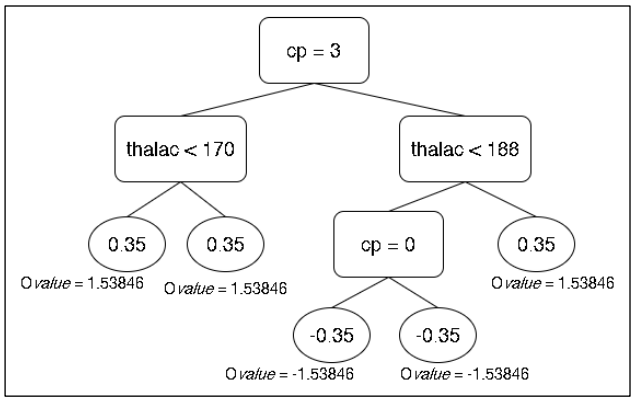


Gambar 3. 40 Hasil Tree Kedua

Setelah *tree* kedua terbentuk, kita menentukan O_{value} untuk masing-masing *leaf* yang ada. Rumus untuk menghitung O_{value} dan perhitungan O_{value} untuk masing-masing *leaf* pada *tree* kedua dapat dilihat di bawah ini, sedangkan *tree* kedua yang sudah mendapatkan nilai O_{value} ditampilkan pada Gambar 3.41 di bawah ini setelah perhitungan selesai.

$$O_{value} = \frac{\sum(residual)_i}{\sum[prev\ probability_i \times (1 - prev\ probability_i)] + \lambda}$$

1. $O_{value} = \frac{0.35}{(0.65\times(1-0.65))+0} = 1.53846$
2. $O_{value} = \frac{0.35}{(0.65\times(1-0.65))+0} = 1.53846$
3. $O_{value} = \frac{-0.35}{(0.35\times(1-0.35))+0} = -1.53846$
4. $O_{value} = \frac{-0.35}{(0.35\times(1-0.35))+0} = -1.53846$
5. $O_{value} = \frac{0.35}{(0.65\times(1-0.65))+0} = 1.53846$



Gambar 3. 41 Tree Kedua Dan O_{Value}

Setelah mendapatkan nilai O_{value} dari masing-masing *leaf* yang ada, dan Gambar tree kedua yang berhasil dibuat dapat dilihat pada Gambar 3.41 di atas. Maka kita dapat menghitung $probabillity$ untuk masing-masing data. Dari hasil $probabillity$ tersebut nantinya akan mendapatkan nilai $residual$ yang baru yang nantinya akan digunakan untuk

membuat *tree* pada iterasi berikutnya yaitu *tree* ketiga, kemudian melakukan perhitungan yang sama sesuai nilai $residual$ yang ada. Namun, pada contoh ini hanya dituliskan proses hingga *tree* kedua. Perhitungan untuk mencari $probabillity$ menggunakan kedua *tree* yang sudah dibangun dapat dilihat di bawah ini, urutan perhitungan sudah disesuaikan dengan urutan data yang digunakan.

1. $\log(odds) = 0 + (0.3 \times 2) + (0.3 \times 1.53846) = 1.0615$
   $probability = \frac{e^{1.0615}}{1 + e^{1.0615}} = \frac{2.891}{3.891}$
   $probability = 0.74299$
2. $\log(odds) = 0 + (0.3 \times 2) + (0.3 \times 1.53846) = 1.0615$
   $probability = \frac{e^{1.0615}}{1 + e^{1.0615}} = \frac{2.891}{3.891}$
   $probability = 0.74299$
3. $\log(odds) = 0 + (0.3 \times -2) + (0.3 \times -1.53846) = -1.0615$
   $probability = \frac{e^{-1.0615}}{1 + e^{-1.0615}} = \frac{0.346}{1.346}$
   $probability = 0.25705$
4. $\log(odds) = 0 + (0.3 \times -2) + (0.3 \times -1.53846) = -1.0615$
   $probability = \frac{e^{-1.0615}}{1 + e^{-1.0615}} = \frac{0.346}{1.346}$
   $probability = 0.25705$
5. $\log(odds) = 0 + (0.3 \times 2) + (0.3 \times 1.53846) = 1.0615$
   $probability = \frac{e^{1.0615}}{1 + e^{1.0615}} = \frac{2.891}{3.891}$
   $probability = 0.74299$

Setelah mengetahui nilai $probabilllity$ maka dapat digunakan untuk mencari nilai $residual$ yang akan digunakan untuk membuat *tree* pada iterasi berikutnya. Dalam algoritma XgBoost, biasanya *tree* yang dibuat 100 *tree* atau lebih. Hasil $probabillity$ dan nilai $residual$ dari kedua *tree* yang dibangun terdapat pada Tabel 3.12 di bawah ini.

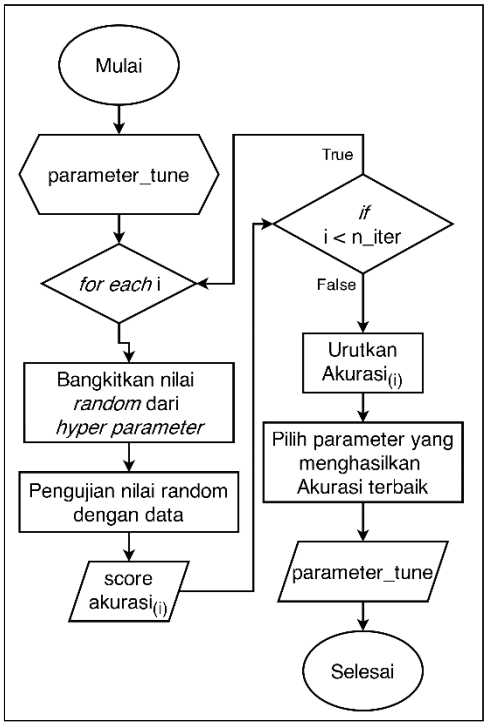
Tabel 3. 12 Data Dan Residual Tree Kedua

Cp	Thalac	Target	Probability	Residual
3	150	1	0.74299	0.25701
3	190	1	0.74299	0.25701
0	96	0	0.25705	-0.25705
1	174	0	0.25705	-0.25705
1	202	1	0.74299	0.25701

Sesuai dengan Tabel 3.12 di atas, nilai $residual$ yang baru dapat digunakan untuk pembuatan *tree* yang ketiga. Dan dari beberapa iterasi pembuatan *tree* akan menghasilkan nilai $residual$ yang semakin kecil hingga mendekati 0 pada label target 0, dan mendekati 1 untuk label target 1. Banyaknya *tree* yang akan dibuat biasanya kita yang menentukan pada

hyper parameter. Pada *hyper parameter*, terdapat beberapa parameter yang dapat kita tentukan untuk mendapatkan nilai parameter yang paling optimal, sehingga model yang dibuat pun memiliki performa yang maksimal dengan akurasi yang tinggi. Untuk menentukan *hyper parameter* yang menghasilkan model yang memiliki akurasi maksimal, maka menggunakan *randomized search optimizer*.

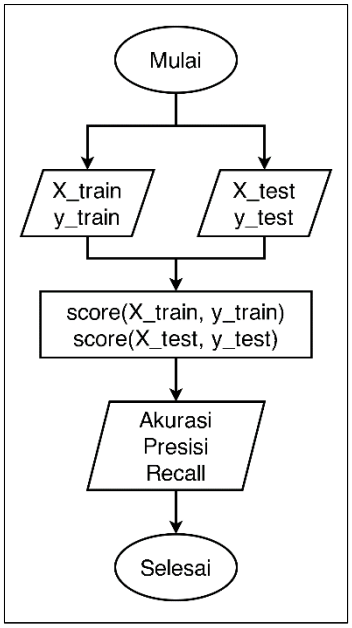
Di dalam *randomized search*, teknik tersebut bekerja dengan cara membangkitkan nilai *random* dari *range* parameter yang sudah kita tentukan dan mencoba nilai parameter tersebut. Teknik *randomized search* akan melakukan pembangkitan nilai *random* dan evaluasi untuk mendapatkan performa dari parameter tersebut secara iteratif sebanyak iterasi yang kita tentukan. Di dalam *randomized search* juga terdapat fitur *cross validation* untuk melakukan validasi terhadap model yang sedang dibuat. Kita dapat menentukan nilai cv, biasanya diatur dengan nilai 3 sampai 5 yang berarti *data training* akan dibagi menjadi sebanyak nilai itu, dan masing-masing bagian akan menjadi *data testing* untuk pengujian model. Apabila iterasi pada *randomized search* berjumlah 100 dan kita mengatur *cross validation* dengan nilai 3, maka jumlah iterasi total adalah 300 iterasi. Flowchart *randomized search* ditampilkan pada gambar 3.42 berikut.



Gambar 3. 42 Flowchart RandomizedSearchCV

3.5.Evaluasi

Untuk melakukan evaluasi dari model klasifikasi xgboost yang dibuat, pada penelitian ini dilakukan dengan menggunakan *confusion matrix*. Dengan menggunakan *confusion matrix* maka dapat diketahui akurasi, presisi, dan recall dari model klasifikasi yang dibuat. Flowchart proses evaluasi dapat dilihat pada Gambar 3.43 di bawah ini.



Gambar 3. 43 Flowchart Proses Evaluasi

Contoh evaluasi yang dilakukan menggunakan data yang sudah ditraining pada proses perhitungan diatas adalah sebagai berikut. model yang digunakan untuk memprediksi adalah model yang dibangun menggunakan dua *tree* dan menggunakan $learning\ rate(\varepsilon)$ adalah 0.3. Rumus untuk menghitung prediksi menggunakan data dapat dilihat pada Persamaan 3.17 dan persamaan 3.18 di bawah ini.

$$\log(odds) = 0 + \sum[0.3 \times O_{value}]_i \quad (3.17)$$

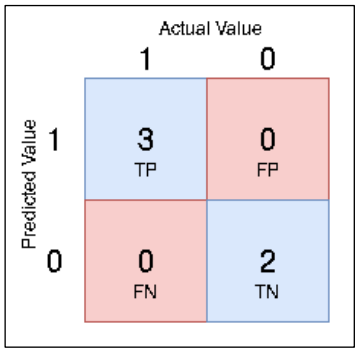
$$probability = \frac{e^{\log(odds)}}{1+e^{\log(odds)}} \quad (3.18)$$

Sesuai dengan rumus yang ada pada Persamaan 3.17 dan Persamaan 3.18 di atas, masing-masing data akan diprediksi menggunakan model yang sudah dibuat, model yang dibuat terdiri dari dua tree yang sudah dibangun. Proses perhitungan dapat dilihat pada proses perhitungan $probabillity$ dan Tabel 3.12 di atas. Sedangkan hasil dari perhitungan $probabillity$ dapat dilihat pada Tabel 3.13 di bawah ini.

Tabel 3. 13 Hasil Prediksi

Cp	Thalac	Target	Prediction
3	150	1	0.74299 (1)
3	190	1	0.74299 (1)
0	96	0	0.25705 (0)
1	174	0	0.25705 (0)
1	202	1	0.74299 (1)

Karena kita menentukan di awal dengan standar 0.5 yang berarti yang memiliki $probabillity \geq 0.5$ adalah label 1, dan $probability < 0.5$ adalah dengan label 0. Sehingga bentuk *confusion matrix* dari hasil evaluasi 5 data di atas dapat dilihat pada Gambar 3.44 di bawah ini.



Gambar 3. 44 Evaluasi Dengan Confusion Matrix

Sesuai dengan *confusion matrix* di atas, maka dapat dilakukan perhitungan utnuk mencari nilai akurasi, recall, dan presisi dari model yang dibangun. Perhitungan tersebut dapat dilihat di bawah ini.

1. $akurasi = \frac{TP+TN}{TP+FP+FN+TN} = \frac{5}{5} = 1.0$
2. $recall = \frac{TP}{TP+FN} = \frac{3}{3} = 1.0$
3. $presisi = \frac{TP}{TP+FP} = \frac{3}{3} = 1.0$

Hasil tersebut hanyalah hasil berdasarkan 5 *dataset* yang digunakan dalam contoh proses *training* dengan catatan model yang dibuat menggunakan nilai $lambda(\lambda)$, $gamma(\gamma)$, dan $cover$ adalah 0. Perhitungan contoh evaluasi di atas juga dilakukan dengan menggunakan data yang sama yang digunakan untuk proses *training*, maka hasil akurasi dapat mencapai 100%. Apabila menggunakan seluruh *dataset* maka hasil akurasi, presisi, dan recall dapat berbeda tidak selalu mencapai 100%. Menggunakan *randomized search* juga akan membantu menemukan parameter terbaik untuk mendapatkan model dengan performa yang baik dan tidak *overfit*.

3.6.Analisis Kebutuhan

Analisis kebutuhan dapat didefinisikan sebuah proses yang dilakukan untuk mencari dan munculkan perbedaan antara tujuan ideal yang ada dan ekspektasi tujuan yang kita harapkan (Briggs, 1991). Definisi lain dari analisis kebutuhan adalah sebuah proses pengumpulan informasi mengenai ketidakseimbangan yang ada dan menentukan prioritas untuk dipecahkan (Sanjaya, 2008). Dalam subbab ini menjelaskan tentang apa aja yang dibutuhkan dan dilakukan oleh sistem yang dibuat. Analisis kebutuhan ini dilakukan untuk mengumpulkan data dan digunakan untuk pengambilan keputusan proses apa saja yang akan terlibat dan menentukan tujuan dari sistem yang dibuat. Terdapat dua jenis kebutuhan, yaitu kebutuhan fungsional dan kebutuhan non fungsional.

3.6.1. Kebutuhan Fungsional

Kebutuhan fungsional adalah informasi mengenai apa saja yang harus ada atau yang akan dilakukan sistem tersebut. Beberapa kebutuhan fungsional yang ada di dalam sistem prediksi ini antara lain adalah:

a. Sistem dapat menerima input data sesuai dengan parameter yang ada.
b. Sistem dapat melakukan prediksi berdasarkan data parameter yang diinput oleh user.

c. Sistem dapat menampilkan hasil prediksi.

3.6.2. Kebutuhan Non Fungsional

Kebutuhan non fungsional adalah kebutuhan yang melibatkan perilaku sistem. Dalam subbab ini akan menjelaskan spefisikasi perangkat yang digunakan untuk membuat sistem, seperti spefisikasi perangkat keras (hardware) ataupun perangkat lunak (software) yang digunakan.

a. Analisis Perangkat Keras (*hardware*)
   Perangkat keras atau hardware yang digunakan untuk membuat sistem prediksi. Spesifikasi hardware dapat dilihat pada Tabel 3.14 dibawah ini.

Tabel 3. 14 Hardware yang digunakan

No	Perangkat Keras	Keterangan
1.	Processor	Intel i7-7700HQ
2.	RAM	8GB DDR5
3.	Storage	1TB
4.	Graphic Card	Nvidia GeForce GTX1050 4GB
5.	Perangkat input dan output	Keyboard, mouse, monitor
6.	Koneksi Internet	Wifi

b. Analisis Kebutuhan Perangkat lunak (*software*)
   Perangkat lunak atau *software* yang digunakan untuk membuat sistem prediksi ini. Daftar software yang digunakan dapat dilihat pada Tabel 3.15 di bawah ini.

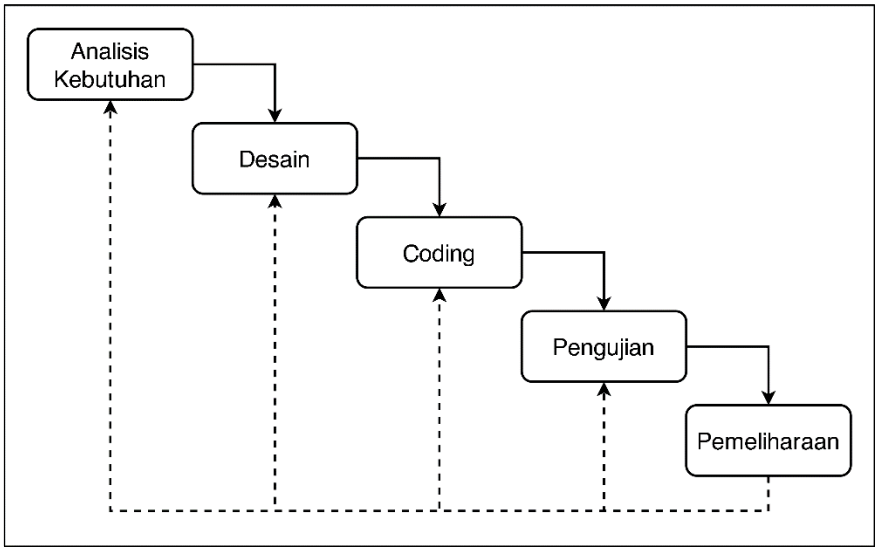
Tabel 3. 15 Software yang digunakan

No.	Perangkat Lunak	Keterangan
1.	Operating System	Windows 10 64bit
2.	Bahasa python 3.7	Bahasa Pemrograman
3.	Jupyter Notebook	*Software* Code Editor
4.	Flask API	*Framework* untuk membuat API
5.	Draw.io	*Software* untuk membuat diagram
6.	Adobe XD	*Software* untuk mendesain UI

3.7.Proses Desain

Di dalam subbab proses desain ini berisi tahap-tahapan yang akan menjelaskan proses pembuatan perangkat lunak sederhana klasifikasi penyakit jantung dengan model klasifikasi yang dibuat menggunakan algoritma XgBoost yang sudah dijelaskan pada subbab sebelumnya. Di dalam proses desain akan dibagi menjadi beberapa subbab lagi yang akan membahas tentang perancangan sistem, perancangan proses, dan desain perancangan antarmuka perangkat lunak yang akan dibangun. Tujuan dari proses desain adalah untuk memberikan gambaran tentang perangkat lunak yang akan dibangun, mulai dari proses kerja perangkat lunak tersebut, hingga tampilan antarmuka yang akan dibangun.

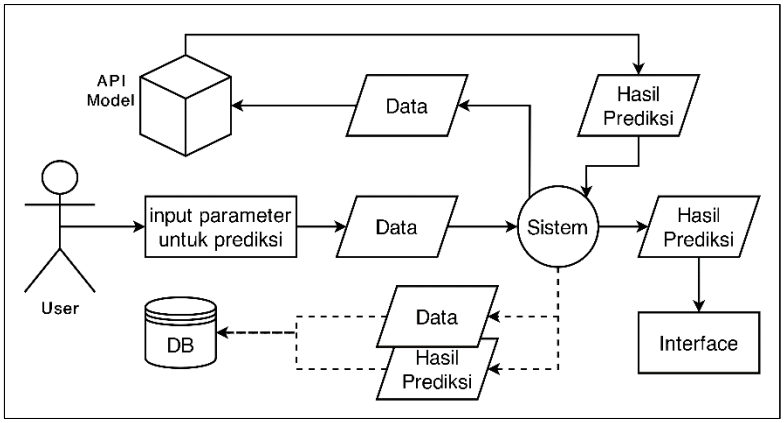
3.7.1. Perancangan Sistem



Gambar 3. 45 Tahapan Waterfall

Pengembangan perangkat lunak pada penelitian ini akan menggunakan model pengembangan *waterfall*. Tahapan model pengembangan *waterfall* dapat dilihat pada gambar 3.45 di atas. Di dalam model pengembangan waterfall dibagi ke dalam beberapa tahapan diantaranya adalah sebagai berikut.

1. Analisis Kebutuhan
   Pada tahap ini dilakukan analisis terhadap apa saja yang akan dibutuhkan dalam pembuatan sistem ini, mulai dari kebutuhan *internal* sistem sampai kebutuhan *external* contohnya laptop atau perangkat yang digunakan untuk membuat sistem.
2. Desain
   Pada tahap ini dilakukan proses perancangan desain meliputi desain arsitektur sistem yang akan dibuat, dfd sistem, hingga desain rancangan *interface* sistem.
3. Coding
   Setelah menyelesaikan tahapan desain dan mendapatkan gambaran desain sistem yang akan dibuat, pada tahap ini dilakukan proses pengkodean atau implementasi dari desain menjadi perangkat lunak.
4. Pengujian
   Pada tahap ini dilakukan pengujian terhadap perangkat lunak yang sudah dibuat.
5. Pemeliharaan
   Tahapan terakhir adalah proses *maintenance* atau pemeliharaan perangkat lunak tersebut. Apabila ditemukan *error* maka dapat dilakukan perbaikan dan dilakukan proses *testing* kembali, dan memungkikan untuk melakukan pengembanga terhadap sistem tersebut.

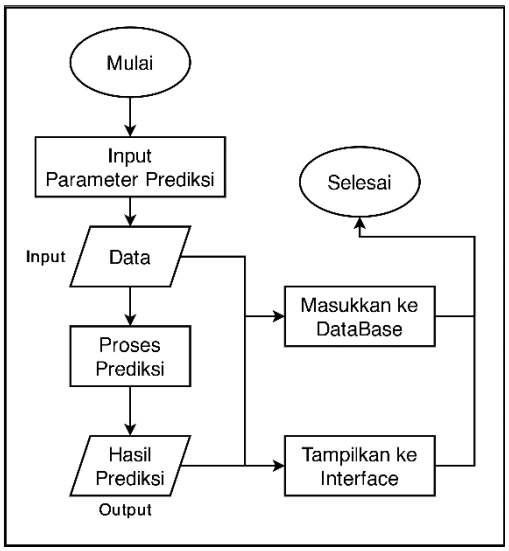


Gambar 3. 46 Arsitektur Sistem Prediksi

Gambaran umum arsitektur sistem prediksi yang akan dibuat adalah seperti Gambar 3.46 di atas. Dimulai dari *user* yang melakukan input parameter berupa *features* data seperti *age*, *cp*, *thalac*, dan lain-lain. Data tersbut kemudian akan dikirimkan sistem dan diprediksi oleh model menggunakan API. Kemudian model akan menerima data yang diinput oleh *user*, dan model akan melakukan prediksi berdasarkan data tersebut. kemudian hasil prediksi akan dikirimkan ke sistem menggunakan API, dan sistem akan menampilkan hasil prediksi tersebut ke *interface* perangkat lunak. Data yang ada meliputi data yang diinput *user* dan data hasil prediksi akan diterima oleh sistem dan akan masuk ke dalam *database* sebagai data baru, yang nantinya dari data baru tersebut dapat digunakan untuk *training* ulang apabila data baru yang didapat sudah banyak.

3.7.2. Perancangan Proses

Dalam perancangan proses ini, akan dijelaskan proses-proses yang terjadi di dalam perangkat lunak yang akan dibuat. *Flowchart* proses sistem terdapat pada gambar 3.47 di bawah ini.

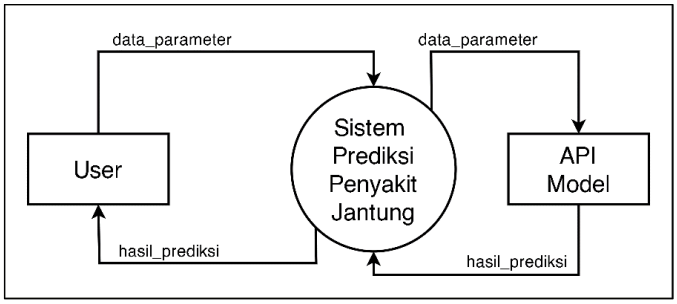


Gambar 3. 47 Flowchart Sistem

Sesuai dengan *flowchart* yang ditampilkan pada Gambar 3.47 di atas, proses yang terjadi pada sistem adalah proses memprediksi penyakit jantung dengan algoritma XgBoost.

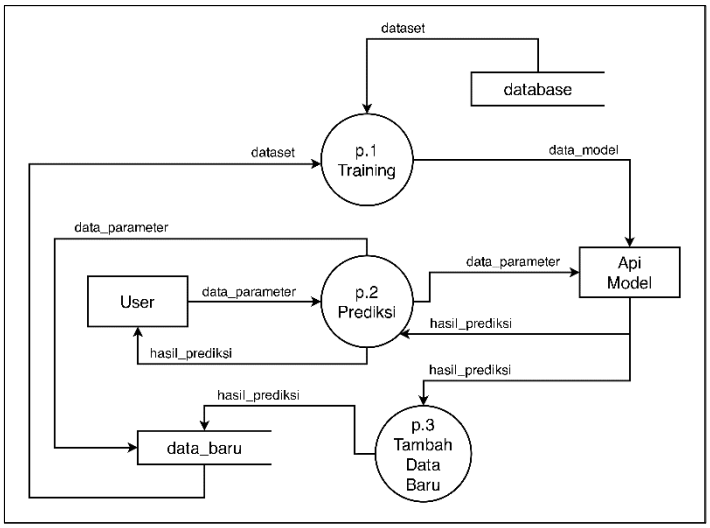
Dimulai dari *user* yang melakukan input parameter, kemudian inputan parameter tersebut akan dibawa menuju model prediksi dengan API, yang nantinya model prediksi akan melakukan prediksi berdasarkan parameter yang diinput oleh *user*. Kemudian setelah model melakukan prediksi, nantinya hasil prediksi yang diberikan oleh model akan dibawa ke sistem melalui API dan akan ditampilkan ke *interface* perangkat lunak. Data dari inputan user dan data hasil prediksi yang diterima oleh sistem akan dimasukkan ke dalam database untuk menjadi data baru, nantinya apabila sudah memiliki banyak data baru dapat dilakukan training untuk memperbaiki model klasifikasi yang ada.

Berikut pada Gambar 3.48 di bawah ini adalah DFD dari sistem yang akan dibuat. Berisi gambaran proses dan data apa saja yang akan berlangsung dalam sistem tersebut.



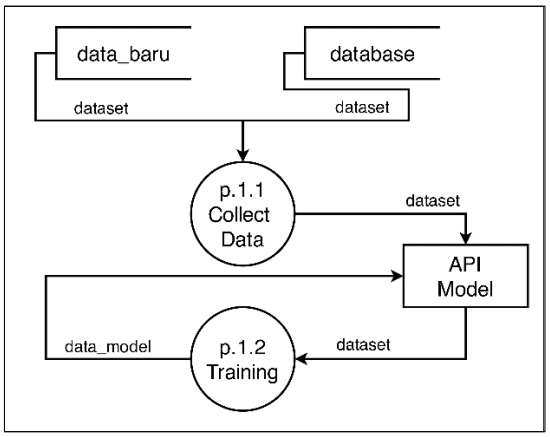
Gambar 3. 48 DFD Level 0

Pada Gambar 3.48 di atas terlihat bahwa sistem yang akan dibuat adalah sistem prediksi yang sudah siap digunakan, *model* yang digunakan untuk memprediksi sudah dibuat dan sudah ditraining sebelum proses pembuatan sistem. Jadi, sistem yang dibuat hanya digunakan untuk memprediksi. Entitas yang ada di dalam dfd tersebut hanya *user* dan API model yang akan menerima *input* dan menghasilkan *output* hasil prediksi. *User* dapat melakukan *input* berupa data parameter yang akan diproses oleh sistem dan diprediksi oleh *model*. Kemudian hasil prediksi yang diberikan oleh *model* akan ditampilkan kepada user. DFD level 1 dari sistem yang akan dibuat dapat dilihat pada Gambar 3.49 di bawah ini.



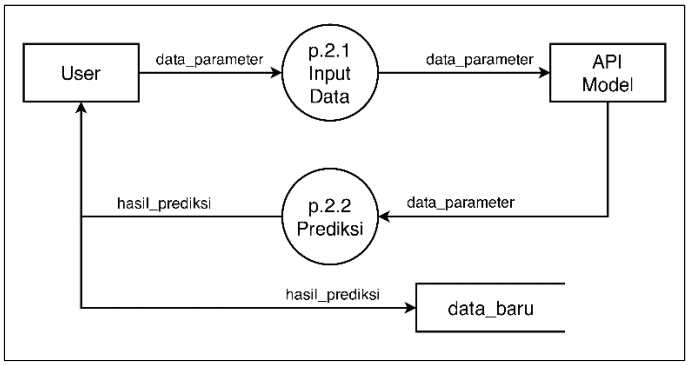
Gambar 3. 49 DFD Level 1

Di dalam dfd level 1 yang ditampilkan pada Gambar 3.49 di atas dapat dipecah menjadi tiga proses utama yaitu proses *training*, proses prediksi, dan yang ketiga adalah proses Menambahkan data baru ke dalam *database* untuk data baru. Hasil dari proses prediksi tersebut nantinya akan ditampilkan ke *user*. Untuk DFD level 2 pada proses *training* dapat dilihat pada Gambar 3.50 di bawah ini.



Gambar 3. 50 DFD Level 2 Proses Training

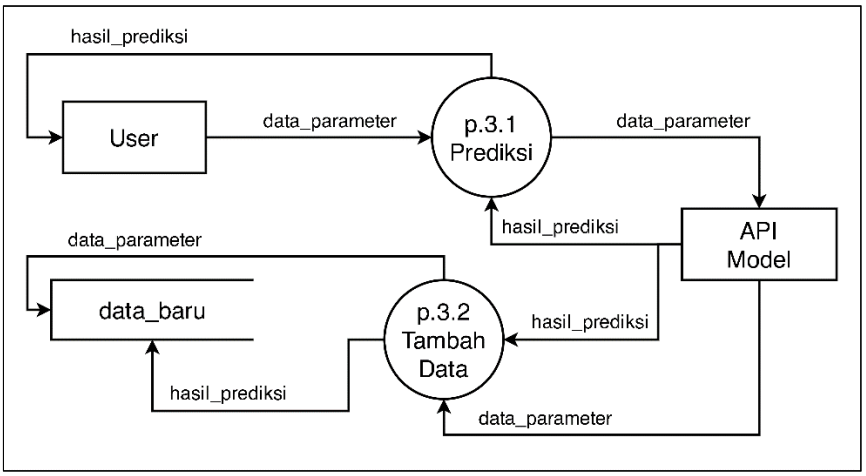
Gambar 3.50 di atas menjelaskan dfd level 2 untuk proses *training*. Proses *training* dilakukan apabila data baru dirasa sudah banyak. Proses *training* diawali dengan pengambilan data dari database, kemudian *database* akan masuk ke *pipeline* dengan total data yang diambil dari *database*, selanjutnya dilakukan proses *training* menggunakan algoritma XgBoost dan pemilihan *hyper parameter* menggunakan RandomizedSearchCV dan akan mendapatkan model klasifikasi. Setelah proses *training* dan model klasifikasi sudah siap, maka sistem prediksi siap digunakan. DFD level 2 untuk proses prediksi dapat dilihat pada Gambar 3.51 di bawah ini.



Gambar 3. 51 DFD Level 2 Proses Prediksi

Gambar 3.51 di atas adalah DFD level 2 untuk prose prediksi, proses prediksi diawali dengan data yang diinputkan oleh *user* dan akan diterima sistem, data yang diinputkan *user* adalah data parameter *features* yang ada pada *dataset* seperti umur, jenis kelamin, tekanan darah, dan lain-lain. Sistem nantinya akan mengirimkan data yang diinputkan tersebut ke model untuk diprediksi. Hasil prediksi yang dihasilkan akan dikirimkan melalui API dan akan ditampilkan ke user melalui *interface* sistem dan data hasil prediksi nantinya juga akan

ditambahkan ke dalam *database*, dimana data baru tersebut apabila sudah banyak dapat digunakan untuk *training* ulang model klasifikasi untuk mendapatkan model klasifikasi yang semakin baik. DFD level 2 untuk proses tambah data baru dapat dilihat pada Gambar 3.52 di bawah ini.

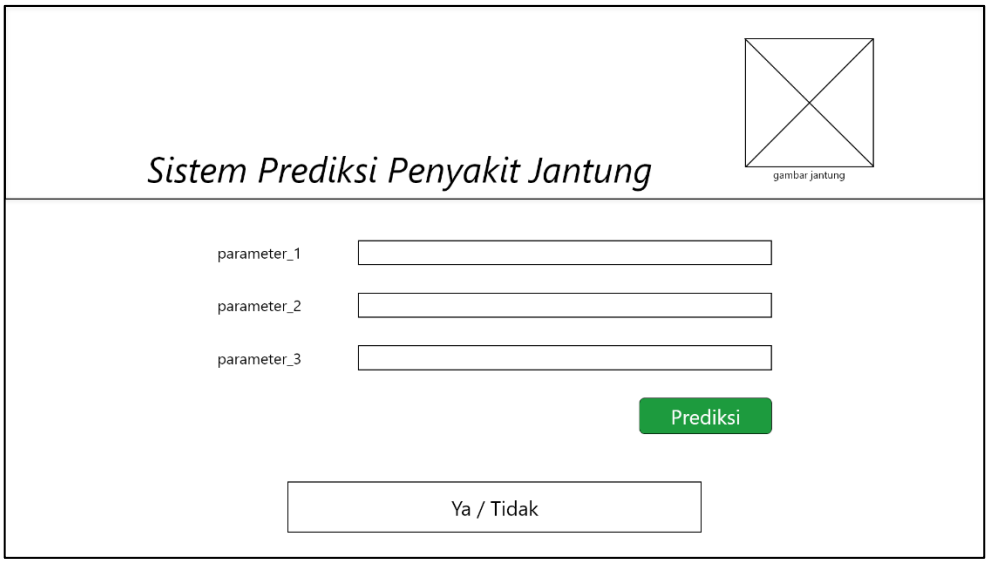


Gambar 3. 52 DFD Level 2 Proses Tambah Data Baru

Gambar 3.52 di atas adalah DFD level 2 untuk proses tambah data baru, dimana setelah proses prediksi dilakukan data yang diinpukan oleh user ke dalam sistem akan ditambahkan ke dalam database sebagai data baru. Data yang ditambahkan adalah data parameter atau features seperti umur, jenis kelamin, tekanan darah, dan lain-lain, dan juga data hasil prediksi yang dihasilkan oleh model klasifikasi.

3.7.3. Perancangan Antarmuka

Tampilan antarmuka perangkat lunak atau disebut *user interface* adalah sebuah media yang digunakan untuk komunikasi antara pengguna dan sistem. Tampilan antarmuka atau *user interface* berisi fungsi-fungsi yang ada di dalam sistem yang dibuat dengan tampilam grafis yang memudahkan pengguna untuk menggunakan sistem tersebut. Dalam sistem prediksi penyakit jantung yang akan dibuat ini, nantinya akan menggunakan satu tampilan yang berisi form inputan untuk memasukkan nilai dari parameter-parameter, terdapat tombol untuk menginput parameter dan terdapat space untuk menampung hasil prediksi sesuai dengan parameter yang diinputkan. Rancangan *user interface* yang akan dibuat adalah seperti Gambar 3.53 di bawah ini.



Gambar 3. 53 User Interface Sistem

DAFTAR PUSTAKA

Afianto, M. F., Faraby, S. Al, & Adiwijaya. (2017). *Kategorisasi Teks pada Hadits Sahih Al-Bukhari menggunakan Random Forest*. *4*(3), 4874–4881.

Aini, S. H. A., Sari, Y. A., & Arwan, A. (2018). Seleksi Fitur Information Gain untuk Klasifikasi Penyakit Jantung Menggunakan Kombinasi Metode K-Nearest Neighbor dan Naïve Bayes. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *2*(9), 2546–2554.

Alpaydin, E. (2010). *Introduction to Machine Learning*. https://books.google.co.id/books?id=tZnSDwAAQBAJ&sitesec=buy&hl=id&source=gbs_vpt_read

Andreanus, J., & Kurniawan, A. (2018). Sejarah , Teori Dasar dan Penerapan Reinforcement Learning : Sebuah Tinjauan Pustaka. *Jurnal Telematika*, *12*(2), 113–118.

Anies. (2015). *Kolesterol dan Penyakit Jantung Koroner*. Ar-Ruzz Media.

Annisa, R. (2019). Analisis Komparasi Algoritma Klasifikasi Data Mining Untuk Prediksi Penderita Penyakit Jantung. *Jurnal Teknik Informatika Kaputama (JTIK)*, *3*(1), 22–28. https://jurnal.kaputama.ac.id/index.php/JTIK/article/view/141/156

Arulkumaran, K., Deisenroth, M., Brundage, M., & Bhatarath, A. (2017). Deep reinforcement learning: A brief survey. *IEEE Signal Process Mag*, *34*, *n*.

Briggs, L. J. (1991). *Instrukticonal Design: Principles and Aplications*. Educational Technology.

Dewi, S. (2016). Komparasi 5 Metode Algoritma Klasifikasi Data Mining Pada Prediksi Keberhasilan Pemasaran Produk Layanan Perbankan. *Techno Nusa Mandiri*, *XIII*(1), 60–66.

Ghani, M. A., & Subekti, A. (2018). Email Spam Filtering Dengan Algoritma Random Forest. *IJCIT (Indonesian Journal on Computer and Information Technology*, *Vol.3*, *No.*(2), 216~221.

Ginting, S. L., Zarman, W., & Hamidah, I. (2014). Analisis dan Penerapan Algoritma C4.5 Dalam Data Mining Untuk Memprediksi Masa Studi Mahasiswa Berdasarkan Data Nilai Akademik. *Snast*, *November*, 159.

Gunawan, V. A., Fitriani, I. I., & Putra, L. S. A. (2020). Sistem Diagnosis Otomatis Identifikasi Penyakit Jantung Coroner Menggunakan Ektraksi Ciri GLCM dan Klasifikasi SVM. *Informatika Mulawarman : Jurnal Ilmiah Ilmu Komputer*, *15*(1), 13. https://doi.org/10.30872/jim.v15i1.2495

Hanifa, T. T., Al-faraby, S., & Adiwijaya. (2017). Analisis Churn Prediction pada Data Pelanggan PT . Telekomunikasi dengan Logistic Regression dan Underbagging. *Universitas Telkom*, *4*(2), 78.

Haqie, Z. A., Nadiah, R. E., & Ariyani, O. P. (2020). Inovasi Pelayanan Publik Suroboyo Bis Di Kota Surabaya. *JPSI (Journal of Public Sector Innovations)*, *5*(1), 23. https://doi.org/10.26740/jpsi.v5n1.p23-30

Ibrahem Ahmed Osman, A., Najah Ahmed, A., Chow, M. F., Feng Huang, Y., & El-Shafie, A. (2020). Extreme gradient boosting (Xgboost) model to predict the groundwater levels in Selangor Malaysia. *Ain Shams Engineering Journal*, *xxxx*. https://doi.org/10.1016/j.asej.2020.11.011

Indriyono, B. V., Utami, E., & Sunyoto, A. (2015). *Pemanfaatan Algoritma Porter Stemmer Untuk Bahasa Indonesia Dalam Proses Klasifikasi Jenis Buku*. 301–310.

Jothikumar, R., & Siva Balan, R. (2016). C4.5 classification algorithm with back-track pruning for accurate prediction of heart disease. *ISSN: 0970-938X (Print)*. https://www.biomedres.info/biomedical-research/c45-classification-algorithm-with-

backtrack-pruning-for-accurate-prediction-of-heart-disease.html
Karo, I. M. K. (2020). *Implementasi Metode XGBoost dan Feature Importance untuk Klasifikasi pada Kebakaran Hutan dan Lahan*. *1*(1), 10–16.
Kusuma, P. P. B., & Srinandi, I. G. A. M. (2013). Prediksi Waktu Ketahanan Hidup Dengan Metode Partial Least Square. *E-Jurnal Matematika*, *2*(1), 49. https://doi.org/10.24843/mtk.2013.v02.i01.p028
Lestari, M. (2014). Penerapan Algoritma Klasifikasi Nearest Neighbor (K-NN) untuk Mendeteksi Penyakit Jantung. *Faktor Exacta*, *7*(September 2010), 366–371.
Lubis, C., & Gondawijaya, F. (2019). Heart Sound Diagnose System with BFCC, MFCC, and Backpropagation Neural Network. *IOP Conference Series: Materials Science and Engineering*, *508*(1). https://doi.org/10.1088/1757-899X/508/1/012119
Marleni, L., & Alhabib, A. (2017). Faktor Risiko Penyakit Jantung Koroner di RSI SITI Khadijah Palembang. *Jurnal Kesehatan*, *8*(3), 478. https://doi.org/10.26630/jk.v8i3.663
Muslim, F. (2019). *Penerapan Brute Force dan Decrease and Conquer pada Parameter Tuning XGBoostClassifier*.
Normawati, D., & Winarti, S. (2017). Seleksi Fitur Menggunakan Penambangan Data Berbasis Variable Precision Rough Set (VPRS) untuk Diagnosis Penyakit Jantung Koroner. *Jurnal Ilmiah Teknik Elektro Komputer Dan Informatika*, *3*(2), 100. https://doi.org/10.26555/jiteki.v3i2.8072
Novandya, A. (2017). Penerapan Algoritma Klasifikasi Data Mining C4.5 Pada Dataset Cuaca Wilayah Bekasi. *KNiST*, *XIV*(2), 368–372.
Nurhayati, Busman, & Iswara, R. P. (2019). Pengembangan Algoritma Unsupervised Learning Technique Pada Big Data Analysis di Media Sosial sebagai media promosi Online Bagi Masyarakat. *Jurnal Teknik Informatika*, *12*(1), 79–96. https://doi.org/10.15408/jti.v12i1.11342
Omer, M. K., Sheta, O. E., Adrees, M. S., Stiawan, D., Riyadi, M. A., & Budiarto, R. (2018). Deep neural network for heart disease medical prescription expert system. *Indonesian Journal of Electrical Engineering and Informatics*, *6*(2), 217–224. https://doi.org/10.11591/ijeei.v6i2.456
Pareza Alam Jusia. (2018). Analisis komparasi pemodelan algoritma decision tree menggunakan metode particle swarm optimization dan metode adaboost untuk prediksi awal penyakit jantung. *Seminar Nasional Sistem Informasi 2018*, 1048–1056.
Peryanto, A., Yudhana, A., & Umar, R. (2020). Klasifikasi Citra Menggunakan Convolutional Neural Network dan K Fold Cross Validation. *Journal of Applied Informatics and Computing*, *4*(1), 45–51. https://doi.org/10.30871/jaic.v4i1.2017
Pinata, N. N. P., Sukarsa, I. M., & Rusjayanthi, N. K. D. (2020). *Prediksi Kecelakaan Lalu Lintas di Bali dengan XGBoost pada Python*. *8*(3), 188–196.
Prasetyo, E., & Prasetiyo, B. (2020). *PENINGKATAN AKURASI KLASIFIKASI ALGORITMA C4.5 MENGGUNAKAN TEKNIK BAGGING PADA DIAGNOSIS PENYAKIT JANTUNG*. *7*(5), 1035–1040. https://doi.org/10.25126/jtiik.202072379
Purnamasari, D., Henrata, J., Sasmita, Y. P., Ihsani, F., & Wicaksana, I. W. S. (2013). *Get Easy Using WEKA*.
Puspitasari, A. M., Ratnawati, D. E., & Widodo, A. W. (2018). Klasifikasi Penyakit Gigi Dan Mulut Menggunakan Metode Support Vector Machine. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *2*(2), 802–810.
Putra, P. D., & Rini, D. P. (2019). Prediksi Penyakit Jantung dengan Algoritma Klasifikasi. *Prosiding Annual Research Seminar 2019*, *5*(1), 978–979.
Rahayu, E. S., Satria, R., & Supriyanto, C. (2015). Penerapan Metode Average Gain,

Threshold Pruning dan Cost Complexity Pruning Untuk Split Atribut Pada Algoritma C4.5. *Journal of Intelligent Systems*, *1*(2), 91–97.

Retnasari, T., & Rahmawati, E. (2017). Diagnosa Prediksi Penyakit Jantung Dengan Model Algoritma Naïve Bayes Dan Algoritma C4.5. *Konferensi Nasional Ilmu Sosial & Teknologi (KNiST)*, 7–12.

Rohman, A., Suhartono, V., & Supriyanto, C. (2017). Penerapan Agoritma C4.5 Berbasis Adaboost Untuk Prediksi Penyakit Jantung. *Jurnal Teknologi Informasi*, *13*, 13–19.

Saifullah, Zarlis, M., Zakaria, & Sembiring, R. W. (2017). Analisa Terhadap Perbandingan Algoritma Decision Tree Dengan Algoritma Random Tree Untuk Pre-Processing Data. *J-SAKTI (Jurnal Sains Komputer Dan Informatika)*, *1*(2), 180. https://doi.org/10.30645/j-sakti.v1i2.41

Sanjaya, W. (2008). *Perencanaan dan Desain Sistem Pembelajaran*. Kencana Group.

Septadaya, A., Dewi, C., & Rahayudi, B. (2019). Implementasi Extreme Learning Machine dan Fast Independent Component Analysis untuk Klasifikasi Aritmia Berdasarkan Rekaman Elektrokardiogram. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer E-ISSN*, *2548*(5), 964X.

Setiawati, D., Taufik, I., Jumadi, J., & Zulfikar, W. B. (2016). Klasifikasi Terjemahan Ayat Al-Quran Tentang Ilmu Sains Menggunakan Algoritma Decision Tree Berbasis Mobile. *Jurnal Online Informatika*, *1*(1), 24. https://doi.org/10.15575/join.v1i1.7

Syukron, M., Santoso, R., & Widiharih, T. (2020). *Perbandingan Metode Smote Random Forest dan Smote XgBoost Untuk Klasifikasi Tingkat Penyakit Hepatitis C Pada Imbalance Class Data*. *9*, 227–236.

Utomo, D. P., & Mesran, M. (2020). Analisis Komparasi Metode Klasifikasi Data Mining dan Reduksi Atribut Pada Data Set Penyakit Jantung. *Jurnal Media Informatika Budidarma*, *4*(2), 437. https://doi.org/10.30865/mib.v4i2.2080

Wibisono, A. B., & Fahrurozi, A. (2019). Perbandingan Algoritma Klasifikasi Dalam Pengklasifikasian Data Penyakit Jantung Koroner. *Jurnal Ilmiah Teknologi Dan Rekayasa*, *24*(3), 161–170. https://doi.org/10.35760/tr.2019.v24i3.2393

Widiastuti, N. A., Santosa, S., & Supriyanto, C. (2014). Algoritma Klasifikasi Data Mining Naïve Bayes Berbasis Particle Swarm Optimization Untuk Deteksi Penyakit Jantung. *Pseudocode*, *1*, 11–14.

Wiharto, W., Suryani, E., & Cahyawati, V. (2019). The methods of duo output neural network ensemble for prediction of coronary heart disease. *Indonesian Journal of Electrical Engineering and Informatics*, *7*(1), 50–57. https://doi.org/10.11591/ijeei.v7i1.458

Yualinda, S., Wijaya, D. R., & Hernawati, E. (2020). Aplikasi Berbasis Dataset E-Commerce Untuk Prediksi Kemiskinan Menggunakan Algoritma Naive Bayes, XgBoost dan Similarity Based Feature Selection. *Jurnal Borneo Cendekia*, *3*(2), 40–46.

Zhang, D., Qian, L., Mao, B., Huang, C., Huang, B., & Si, Y. (2018). A Data-Driven Design for Fault Detection of Wind Turbines Using Random Forests and XGboost. *IEEE Access*, *6*, 21020–21031. https://doi.org/10.1109/ACCESS.2018.2818678

LAMPIRAN

Tabel 5. 1 Lampiran Dataset

age	sex	cp	trestbps	chol	Fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
57	1	0	140	192	0	1	148	0	0.4	1	0	1	1
56	0	1	140	294	0	0	153	0	1.3	1	0	2	1
44	1	1	120	263	0	1	173	0	0	2	0	3	1
52	1	2	172	199	1	1	162	0	0.5	2	0	3	1
57	1	2	150	168	0	1	174	0	1.6	2	0	2	1
54	1	0	140	239	0	1	160	0	1.2	2	0	2	1
48	0	2	130	275	0	1	139	0	0.2	2	0	2	1
49	1	1	130	266	0	1	171	0	0.6	2	0	2	1
64	1	3	110	211	0	0	144	1	1.8	1	0	2	1
58	0	3	150	283	1	0	162	0	1	2	0	2	1
50	0	2	120	219	0	1	158	0	1.6	1	0	2	1
58	0	2	120	340	0	1	172	0	0	2	0	2	1
66	0	3	150	226	0	1	114	0	2.6	0	0	2	1
43	1	0	150	247	0	1	171	0	1.5	2	0	2	1
69	0	3	140	239	0	1	151	0	1.8	2	2	2	1
59	1	0	135	234	0	1	161	0	0.5	1	0	3	1
44	1	2	130	233	0	1	179	1	0.4	2	0	2	1
42	1	0	140	226	0	1	178	0	0	2	0	2	1
61	1	2	150	243	1	1	137	1	1	1	0	2	1
40	1	3	140	199	0	1	178	1	1.4	2	0	3	1
71	0	1	160	302	0	1	162	0	0.4	2	2	2	1
59	1	2	150	212	1	1	157	0	1.6	2	0	2	1
51	1	2	110	175	0	1	123	0	0.6	2	0	2	1
65	0	2	140	417	1	0	157	0	0.8	2	1	2	1
53	1	2	130	197	1	0	152	0	1.2	0	0	2	1
41	0	1	105	198	0	1	168	0	0	2	1	2	1
65	1	0	120	177	0	1	140	0	0.4	2	0	3	1
44	1	1	130	219	0	0	188	0	0	2	0	2	1
54	1	2	125	273	0	0	152	0	0.5	0	1	2	1
51	1	3	125	213	0	0	125	1	1.4	2	1	2	1
46	0	2	142	177	0	0	160	1	1.4	0	0	2	1
54	0	2	135	304	1	1	170	0	0	2	0	2	1
54	1	2	150	232	0	0	165	0	1.6	2	0	3	1
65	0	2	155	269	0	1	148	0	0.8	2	0	2	1
65	0	2	160	360	0	0	151	0	0.8	2	0	2	1

Tabel 5. 2 Lanjutan Lampiran Dataset

Age	Sex	Cp	Trestbps	Chol	Fbs	Restecg	Thalac	Exang	Oldpeak	Slope	Ca	Thal	target
51	0	2	140	308	0	0	142	0	1.5	2	1	2	1
48	1	1	130	245	0	0	180	0	0.2	1	0	2	1
45	1	0	104	208	0	0	148	1	3	1	0	2	1
53	0	0	130	264	0	0	143	0	0.4	1	0	2	1
39	1	2	140	321	0	0	182	0	0	2	0	2	1
52	1	1	120	325	0	1	172	0	0.2	2	0	2	1
44	1	2	140	235	0	0	180	0	0	2	0	2	1
47	1	2	138	257	0	0	156	0	0	2	0	2	1
53	0	2	128	216	0	0	115	0	0	2	0	0	1
53	0	0	138	234	0	0	160	0	0	2	0	2	1
51	0	2	130	256	0	0	149	0	0.5	2	0	2	1
66	1	0	120	302	0	0	151	0	0.4	1	0	2	1
62	1	2	130	231	0	1	146	0	1.8	1	3	3	1
44	0	2	108	141	0	1	175	0	0.6	1	0	2	1
63	0	2	135	252	0	0	172	0	0	2	0	2	1
52	1	1	134	201	0	1	158	0	0.8	2	1	2	1
48	1	0	122	222	0	0	186	0	0	2	0	2	1
45	1	0	115	260	0	0	185	0	0	2	0	2	1
34	1	3	118	182	0	0	174	0	0	2	0	2	1
57	0	0	128	303	0	0	159	0	0	2	1	2	1
71	0	2	110	265	1	0	130	0	0	2	1	2	1
54	1	1	108	309	0	1	156	0	0	2	0	3	1
52	1	3	118	186	0	0	190	0	0	1	0	1	1
41	1	1	135	203	0	1	132	0	0	1	0	1	1
58	1	2	140	211	1	0	165	0	0	2	0	2	1
35	0	0	138	183	0	1	182	0	1.4	2	0	2	1
51	1	2	100	222	0	1	143	1	1.2	1	0	2	1
45	0	1	130	234	0	0	175	0	0.6	1	0	2	1
44	1	1	120	220	0	1	170	0	0	2	0	2	1
62	0	0	124	209	0	1	163	0	0	2	0	2	1
54	1	2	120	258	0	0	147	0	0.4	1	0	3	1
51	1	2	94	227	0	1	154	1	0	2	1	3	1
29	1	1	130	204	0	0	202	0	0	2	0	2	1
51	1	0	140	261	0	0	186	1	0	2	0	2	1
43	0	2	122	213	0	1	165	0	0.2	1	0	2	1
55	0	1	135	250	0	0	161	0	1.4	1	0	2	1
51	1	2	125	245	1	0	166	0	2.4	1	0	2	1
59	1	1	140	221	0	1	164	1	0	2	0	2	1
52	1	1	128	205	1	1	184	0	0	2	0	2	1
58	1	2	105	240	0	0	154	1	0.6	1	0	3	1
41	1	2	112	250	0	1	179	0	0	2	0	2	1

Tabel 5. 3 Lanjutan Lampiran Dataset

Age	Sex	Cp	Trestbps	Chol	Fbs	Restecg	Thalac	Exang	Oldpeak	Slope	Ca	Thal	target
45	1	1	128	308	0	0	170	0	0	2	0	2	1
60	0	2	102	318	0	1	160	0	0	2	1	2	1
52	1	3	152	298	1	1	178	0	1.2	1	0	3	1
42	0	0	102	265	0	0	122	0	0.6	1	0	2	1
67	0	2	115	564	0	0	160	0	1.6	1	0	3	1
68	1	2	118	277	0	1	151	0	1	2	1	3	1
46	1	1	101	197	1	1	156	0	0	2	0	3	1
54	0	2	110	214	0	1	158	0	1.6	1	0	2	1
58	0	0	100	248	0	0	122	0	1	1	0	2	1
48	1	2	124	255	1	1	175	0	0	2	2	2	1
57	1	0	132	207	0	1	168	1	0	2	0	3	1
52	1	2	138	223	0	1	169	0	0	2	4	2	1
54	0	1	132	288	1	0	159	1	0	2	1	2	1
45	0	1	112	160	0	1	138	0	0	1	0	2	1
53	1	0	142	226	0	0	111	1	0	2	0	3	1
62	0	0	140	394	0	0	157	0	1.2	1	0	2	1
52	1	0	108	233	1	1	147	0	0.1	2	3	3	1
43	1	2	130	315	0	1	162	0	1.9	2	1	2	1
53	1	2	130	246	1	0	173	0	0	2	3	2	1
42	1	3	148	244	0	0	178	0	0.8	2	2	2	1
59	1	3	178	270	0	0	145	0	4.2	0	0	3	1
63	0	1	140	195	0	1	179	0	0	2	2	2	1
42	1	2	120	240	1	1	194	0	0.8	0	0	3	1
50	1	2	129	196	0	1	163	0	0	2	0	2	1
68	0	2	120	211	0	0	115	0	1.5	1	0	2	1
69	1	3	160	234	1	0	131	0	0.1	1	1	2	1
45	0	0	138	236	0	0	152	1	0.2	1	0	2	1
50	0	1	120	244	0	1	162	0	1.1	2	0	2	1
50	0	0	110	254	0	0	159	0	0	2	0	2	1
64	0	0	180	325	0	1	154	1	0	2	0	2	1
57	1	2	150	126	1	1	173	0	0.2	2	1	3	1
64	0	2	140	313	0	1	133	0	0.2	2	0	3	1
43	1	0	110	211	0	1	161	0	0	2	0	3	1
55	1	1	130	262	0	1	155	0	0	2	0	2	1
37	0	2	120	215	0	1	170	0	0	2	0	2	1
41	1	2	130	214	0	0	168	0	2	1	0	2	1
56	1	3	120	193	0	0	162	0	1.9	1	0	3	1
46	0	1	105	204	0	1	172	0	0	2	0	2	1
46	0	0	138	243	0	0	152	1	0	1	0	2	1
64	0	0	130	303	0	1	122	0	2	1	2	2	1
59	1	0	138	271	0	0	182	0	0	2	0	2	1
41	0	2	112	268	0	0	172	1	0	2	0	2	1

Tabel 5. 4 Lanjutan Lampiran Dataset

Age	Sex	Cp	Trestbps	Chol	Fbs	Restecg	Thalach	Exang	Oldpeak	Slope	Ca	Thal	target
54	0	2	108	267	0	0	167	0	0	2	0	2	1
39	0	2	94	199	0	1	179	0	0	2	0	2	1
34	0	1	118	210	0	1	192	0	0.7	2	0	2	1
47	1	0	112	204	0	1	143	0	0.1	2	0	2	1
67	0	2	152	277	0	1	172	0	0	2	1	2	1
52	0	2	136	196	0	0	169	0	0.1	1	0	2	1
74	0	1	120	269	0	0	121	1	0.2	2	1	2	1
54	0	2	160	201	0	1	163	0	0	2	1	2	1
49	0	1	134	271	0	1	162	0	0	1	0	2	1
42	1	1	120	295	0	1	162	0	0	2	0	2	1
41	1	1	110	235	0	1	153	0	0	2	0	2	1
41	0	1	126	306	0	1	163	0	0	2	0	2	1
49	0	0	130	269	0	1	163	0	0	2	0	2	1
60	0	2	120	178	1	1	96	0	0	2	0	2	1
62	1	1	128	208	1	0	140	0	0	2	0	2	1
57	1	0	110	201	0	1	126	1	1.5	1	0	1	1
64	1	0	128	263	0	1	105	1	0.2	1	1	3	1
51	0	2	120	295	0	0	157	0	0.6	2	0	2	1
43	1	0	115	303	0	1	181	0	1.2	1	0	2	1
42	0	2	120	209	0	1	173	0	0	1	0	2	1
67	0	0	106	223	0	1	142	0	0.3	2	2	2	1
76	0	2	140	197	0	2	116	0	1.1	1	0	2	1
70	1	1	156	245	0	0	143	0	0	2	0	2	1
44	0	2	118	242	0	1	149	0	0.3	1	1	2	1
60	0	3	150	240	0	1	171	0	0.9	2	0	2	1
44	1	2	120	226	0	1	169	0	0	2	0	2	1
42	1	2	130	180	0	1	150	0	0	2	0	2	1
66	1	0	160	228	0	0	138	0	2.3	2	0	1	1
71	0	0	112	149	0	1	125	0	1.6	1	0	2	1
64	1	3	170	227	0	0	155	0	0.6	1	0	3	1
66	0	2	146	278	0	0	152	0	0	1	1	2	1
39	0	2	138	220	0	1	152	0	0	1	0	2	1
58	0	0	130	197	0	1	131	0	0.6	1	0	2	1
47	1	2	130	253	0	1	179	0	0	2	0	2	1
35	1	1	122	192	0	1	174	0	0	2	0	2	1
58	1	1	125	220	0	1	144	0	0.4	1	4	3	1
56	1	1	130	221	0	0	163	0	0	2	0	3	1
56	1	1	120	240	0	1	169	0	0	0	0	2	1
55	0	1	132	342	0	1	166	0	1.2	2	0	2	1
41	1	1	120	157	0	1	182	0	0	2	0	2	1
38	1	2	138	175	0	1	173	0	0	2	4	2	1
38	1	2	138	175	0	1	173	0	0	2	4	2	1

Tabel 5. 5 Lanjutan Lampiran Dataset

Age	Sex	Cp	Trestbps	Chol	Fbs	Restecg	Thalach	Exang	Oldpeak	Slope	Ca	Thal	target
67	1	0	160	286	0	0	108	1	1.5	1	3	2	0
67	1	0	120	229	0	0	129	1	2.6	1	2	3	0
62	0	0	140	268	0	0	160	0	3.6	0	2	2	0
63	1	0	130	254	0	0	147	0	1.4	1	1	3	0
53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
56	1	2	130	256	1	0	142	1	0.6	1	1	1	0
48	1	1	110	229	0	1	168	0	1	0	0	3	0
58	1	1	120	284	0	0	160	0	1.8	1	0	2	0
58	1	2	132	224	0	0	173	0	3.2	2	2	3	0
60	1	0	130	206	0	0	132	1	2.4	1	2	3	0
40	1	0	110	167	0	0	114	1	2	1	0	3	0
60	1	0	117	230	1	1	160	1	1.4	2	2	3	0
64	1	2	140	335	0	1	158	0	0	2	0	2	0
43	1	0	120	177	0	0	120	1	2.5	1	0	3	0
57	1	0	150	276	0	0	112	1	0.6	1	1	1	0
55	1	0	132	353	0	1	132	1	1.2	1	1	3	0
65	0	0	150	225	0	0	114	0	1	1	3	3	0
61	0	0	130	330	0	0	169	0	0	2	0	2	0
58	1	2	112	230	0	0	165	0	2.5	1	1	3	0
50	1	0	150	243	0	0	128	0	2.6	1	0	3	0
44	1	0	112	290	0	0	153	0	0	2	1	2	0
60	1	0	130	253	0	1	144	1	1.4	2	1	3	0
54	1	0	124	266	0	0	109	1	2.2	1	1	3	0
50	1	2	140	233	0	1	163	0	0.6	1	1	3	0
41	1	0	110	172	0	0	158	0	0	2	0	3	0
51	0	0	130	305	0	1	142	1	1.2	1	0	3	0
58	1	0	128	216	0	0	131	1	2.2	1	3	3	0
54	1	0	120	188	0	1	113	0	1.4	1	1	3	0
60	1	0	145	282	0	0	142	1	2.8	1	2	3	0
60	1	2	140	185	0	0	155	0	3	1	0	2	0
59	1	0	170	326	0	0	140	1	3.4	0	0	3	0
46	1	2	150	231	0	1	147	0	3.6	1	0	2	0
67	1	0	125	254	1	1	163	0	0.2	1	2	3	0
62	1	0	120	267	0	1	99	1	1.8	1	2	3	0
65	1	0	110	248	0	0	158	0	0.6	2	2	1	0
44	1	0	110	197	0	0	177	0	0	2	1	2	0
60	1	0	125	258	0	0	141	1	2.8	1	1	3	0
58	1	0	150	270	0	0	111	1	0.8	2	0	3	0
68	1	2	180	274	1	0	150	1	1.6	1	0	3	0
62	0	0	160	164	0	0	145	0	6.2	0	3	3	0
52	1	0	128	255	0	1	161	1	0	2	1	3	0
59	1	0	110	239	0	0	142	1	1.2	1	1	3	0

Tabel 5. 6 Lanjutan Lampiran Dataset

Age	Sex	Cp	Trestbps	Chol	Fbs	Restecg	Thalac	Exang	Olspeak	Slope	Ca	Thal	target
60	0	0	150	258	0	0	157	0	2.6	1	2	3	0
49	1	2	120	188	0	1	139	0	2	1	3	3	0
59	1	0	140	177	0	1	162	1	0	2	1	3	0
57	1	2	128	229	0	0	150	0	0.4	1	1	3	0
61	1	0	120	260	0	1	140	1	3.6	1	1	3	0
39	1	0	118	219	0	1	140	0	1.2	1	0	3	0
61	0	0	145	307	0	0	146	1	1	1	0	3	0
56	1	0	125	249	1	0	144	1	1.2	1	1	2	0
43	0	0	132	341	1	0	136	1	3	1	0	3	0
62	0	2	130	263	0	1	97	0	1.2	1	1	3	0
63	1	0	130	330	1	0	132	1	1.8	2	3	3	0
65	1	0	135	254	0	0	127	0	2.8	1	1	3	0
48	1	0	130	256	1	0	150	1	0	2	2	3	0
63	0	0	150	407	0	0	154	0	4	1	3	3	0
55	1	0	140	217	0	1	111	1	5.6	0	0	3	0
65	1	3	138	282	1	0	174	0	1.4	1	1	2	0
56	0	0	200	288	1	0	133	1	4	0	2	3	0
54	1	0	110	239	0	1	126	1	2.8	1	1	3	0
70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
62	1	1	120	281	0	0	103	0	1.4	1	1	3	0
35	1	0	120	198	0	1	130	1	1.6	1	0	3	0
59	1	3	170	288	0	0	159	0	0.2	1	0	3	0
64	1	2	125	309	0	1	131	1	1.8	1	0	3	0
47	1	2	108	243	0	1	152	0	0	2	0	2	0
57	1	0	165	289	1	0	124	0	1	1	3	3	0
55	1	0	160	289	0	0	145	1	0.8	1	1	3	0
64	1	0	120	246	0	0	96	1	2.2	0	1	2	0
70	1	0	130	322	0	0	109	0	2.4	1	3	2	0
51	1	0	140	299	0	1	173	1	1.6	2	0	3	0
58	1	0	125	300	0	0	171	0	0	2	2	3	0
60	1	0	140	293	0	0	170	0	1.2	1	2	3	0
77	1	0	125	304	0	0	162	1	0	2	3	2	0
35	1	0	126	282	0	0	156	1	0	2	0	3	0
70	1	2	160	269	0	1	112	1	2.9	1	1	3	0
59	0	0	174	249	0	1	143	1	0	1	0	2	0
64	1	0	145	212	0	0	132	0	2	1	2	1	0
57	1	0	152	274	0	1	88	1	1.2	1	1	3	0
56	1	0	132	184	0	0	105	1	2.1	1	1	1	0
48	1	0	124	274	0	0	166	0	0.5	1	0	3	0
56	0	0	134	409	0	0	150	1	1.9	1	2	3	0
66	1	1	160	246	0	1	120	1	0	1	3	1	0
54	1	1	192	283	0	0	195	0	0	2	1	3	0

Tabel 5. 7 Lanjutan Lampiran Dataset

Age	Sex	cp	trestbps	Chol	Fbs	Restecg	Thalach	Exang	Oldpeak	Slope	Ca	Thal	target
69	1	2	140	254	0	0	146	0	2	1	3	3	0
51	1	0	140	298	0	1	122	1	4.2	1	3	3	0
43	1	0	132	247	1	0	143	1	0.1	1	4	3	0
62	0	0	138	294	1	1	106	0	1.9	1	3	2	0
67	1	0	100	299	0	0	125	1	0.9	1	2	2	0
59	1	3	160	273	0	0	125	0	0	2	0	2	0
45	1	0	142	309	0	0	147	1	0	1	3	3	0
58	1	0	128	259	0	0	130	1	3	1	2	3	0
50	1	0	144	200	0	0	126	1	0.9	1	0	3	0
62	0	0	150	244	0	1	154	1	1.4	1	0	2	0
38	1	3	120	231	0	1	182	1	3.8	1	0	3	0
66	0	0	178	228	1	1	165	1	1	1	2	3	0
52	1	0	112	230	0	1	160	0	0	2	1	2	0
53	1	0	123	282	0	1	95	1	2	1	2	3	0
63	0	0	108	269	0	1	169	1	1.8	1	2	2	0
54	1	0	110	206	0	0	108	1	0	1	1	2	0
66	1	0	112	212	0	0	132	1	0.1	2	1	2	0
55	0	0	180	327	0	2	117	1	3.4	1	0	2	0
49	1	2	118	149	0	0	126	0	0.8	2	3	2	0
54	1	0	122	286	0	0	116	1	3.2	1	2	2	0
56	1	0	130	283	1	0	103	1	1.6	0	0	3	0
46	1	0	120	249	0	0	144	0	0.8	2	0	3	0
61	1	3	134	234	0	1	145	0	2.6	1	2	2	0
67	1	0	120	237	0	1	71	0	1	1	0	2	0
58	1	0	100	234	0	1	156	0	0.1	2	1	3	0
47	1	0	110	275	0	0	118	1	1	1	1	2	0
52	1	0	125	212	0	1	168	0	1	2	2	3	0
58	1	0	146	218	0	1	105	0	2	1	1	3	0
57	1	1	124	261	0	1	141	0	0.3	2	0	3	0
58	0	1	136	319	1	0	152	0	0	2	2	2	0
61	1	0	138	166	0	0	125	1	3.6	1	1	2	0
42	1	0	136	315	0	1	125	1	1.8	1	0	1	0
52	1	0	128	204	1	1	156	1	1	1	0	0	0
59	1	2	126	218	1	1	134	0	2.2	1	1	1	0
40	1	0	152	223	0	1	181	0	0	2	0	3	0
61	1	0	140	207	0	0	138	1	1.9	2	1	3	0
46	1	0	140	311	0	1	120	1	1.8	1	2	3	0
59	1	3	134	204	0	1	162	0	0.8	2	2	2	0
57	1	1	154	232	0	0	164	0	0	2	1	2	0
57	1	0	110	335	0	1	143	1	3	1	1	3	0
55	0	0	128	205	0	2	130	1	2	1	1	3	0
61	1	0	148	203	0	1	161	0	0	2	1	3	0

Tabel 5. 8 Lanjutan Lampiran Dataset

Age	Sex	Cp	Trestbps	Chol	Fbs	Restecg	Thalach	Exang	Oldpeak	Slope	Ca	Thal	target
58	1	0	114	318	0	2	140	0	4.4	0	3	1	0
58	0	0	170	225	1	0	146	1	2.8	1	2	1	0
67	1	2	152	212	0	0	150	0	0.8	1	0	3	0
44	1	0	120	169	0	1	144	1	2.8	0	0	1	0
63	1	0	140	187	0	0	144	1	4	2	2	3	0
63	0	0	124	197	0	1	136	1	0	1	0	2	0
59	1	0	164	176	1	0	90	0	1	1	2	1	0
57	0	0	140	241	0	1	123	1	0.2	1	0	3	0
45	1	3	110	264	0	1	132	0	1.2	1	0	3	0
68	1	0	144	193	1	1	141	0	3.4	1	2	3	0
57	1	0	130	131	0	1	115	1	1.2	1	1	3	0
57	0	1	130	236	0	0	174	0	0	1	1	2	0