BAB IV
ANALISA DAN PEMBAHASAN

4.1 Diskritisasi Data

Pada bagian ini akan di jelaskan secara terinci terkait proses diskritisasi variabel di dataset Malware yang telah melewati proses pra-processing data sebelumnya, dengan tahapan langkah-langkah sebagai berikut :

a. Data masing-masing instance di urutkan dari terkecil s/d terbesar (ascending)
b. Lakukakan partisi ke dalam bin dengan teknik equal-width yaitu dengan cara membagi data kedalam interval **k** dengan ukuran yang sama, selanjutnya lebar interval dihitung dengan w = (min-max)/k, sedangkan batas interval dihitung dengan min+w, min+2w,….,min+(k-1)w
c. Lakukan smoothing berdasarkan batasan (by Boundaries) dengan cara setiap nilai bin diganti dengan nilai yang paling dekat dari batas nilai, dimana batasan nilai terbentuk dari [min, max] tiap bin.

Berdasarkan langkah-langkah tahapan binning yang telah diuraikan di atas, peneliti akan melakukan proses diskritisasi terhadap semua variabel dengan cara membuat 2 (dua) jenis percobaan diskritisasi variabel pada dataset malware yaitu

a. Binning Tanpa Menetapkan Parameter Batasan (*no define boundaries*)
   Binning dengan cara mempartisi setiap nilai-nilai yang ada pada setiap atribut dengan menggunakan nilai maksimal (max) dan nilai minimal (min) yang ada pada range nilai setiap atributnya, sehingga bin yang terbentuk pada setiap atribut dataset sebagai berikut :
   1. Atribut millisecond (Rentang Nilai -1.73 s/d 1.73)

      Untuk k = 3

      Ditentukan max = 1,73, min = -1,73

      sehingga nilai w = (max-min)/3 = 1.153

      Batas interval : min+w, min+(k-1)w sehingga nilai batas intervalnya -0.577 dan 0.577 sehingga batas partisi yang terbentuk pada batas nilai -0.577

dan 0.577 kemudian terbentuk batasan interval pada masing-masing bin yaitu

Range 1 : -∞ - -0.577

Range 2 : -0.577 - 0.577

Range 3 : 0.577 - ∞

Untuk k = 5

Ditentukan max = 1,73, min = -1,73

sehingga nilai w = (max-min)/5 = 0.692

Batas interval : min+w, min+(k-1)w sehingga

nilai batas intervalnya -1.038, -0.346, 0.346 dan 1.038 sehingga batas partisi yang terbentuk pada batas nilai -1.038, -0.346, 0.346 dan 1.038 kemudian terbentuk batasan interval pada masing-masing bin yaitu

Range 1 : -∞ - -1.038

Range 2 : -1.038 - - 0.346

Range 3 : -0.346 - 0.346

Range 4 : 0.346 - 1.038

Range 5 : 1.038 - ∞

2. Atribut state (Rentang Nilai -0.17 s/d 46.05)

Untuk k = 3

Ditentukan max = 46.05, min = -0.17

sehingga nilai w = (max-min)/3 = 15.41

Batas interval : min+w, min+(k-1)w sehingga

nilai batas intervalnya 15.237 dan 30.642 sehingga batas partisi yang terbentuk pada batas nilai 15.237 dan 30.642 kemudian terbentuk batasan interval pada masing-masing bin yaitu

Range 1 : -∞ - 15.237

Range 2 : 15.237 - 30.642

Range 3 : 30.642 - ∞

Untuk k = 5

Ditentukan max = 46.05, min = -0.17

sehingga nilai w = (max-min)/3 = 9.24

Batas interval : min+w, min+(k-1)w sehingga

nilai batas intervalnya 9.075, 18.318, 27.561 dan 36.804 sehingga batas partisi yang terbentuk pada batas nilai 9.075, 18.318, 27.561 dan 36.804 kemudian terbentuk batasan interval pada masing-masing bin yaitu

Range 1 : -∞ - 9.075

Range 2 : 9.075 - 18.318

Range 3 : 18.318 - 27.561

Range 4 : 27.561 - 36.804

Range 5 : 36.804 - ∞

Sehingga terbentuk nilai-nilai partisi 3 Bin dan 5 Bin tanpa parameter batasan (*no define boundaries*) untuk penghitungan manual semua atribut seperti tabel di bawah ini :

Tabel 4.1 Hasil Diskritisasi 3 Bin dan 5 Bin Tanpa Parameter Batasan (Manual)

No	Atribut	Nilai			Nilai Batas Partisi (Bin)	
		Range	Min	Max	k = 3	k = 5
1.	millisecond	-1.73 s/d 1.73	-1.73	1.73	-0.577, 0.577	-1.038, -0.346, 0.346, 1.038
2.	state	-0.17 s/d 46.05	-0.17	46.05	15.237, 30.642	9.075, 18.318, 27.561, 36.804
3.	prio	-1.74 s/d 1.74	-1.74	1.74	-0.580, 0.581	-1.044, -0.348, 0.349, 1.046
4.	static_prio	-0.91 s/d 2.97	-0.91	2.7	0.382, 1.674	-0.135, 0.640, 1.415, 2.190
5.	vm_truncate_count	-1.73 s/d 3.64	-1.73	3.64	0.062, 1.850	-0.653, 0.420, 1.492, 2.565
6.	free_area_cache	0.21 s/d 10.87	0.21	10.87	3.485, 7.177	2.009, 4.223, 6.438, 8.653
7.	mm_users	-1.38 s/d 3.16	-1.38	3.16	0.133, 1.644	-0.472, 0.435, 1.342, 2.248
8.	map_count	-1.63 s/d 5.13	-1.63	5.13	0.621, 2.875	-0, 281, 1.071, 2.424, 3.776
9.	total_vm	-0.84 s/d 8.15	-0.84	8.15	2.157, 5.154	0.957, 2.756, 4.555, 6.354
10.	shared_vm	-1.89 s/d 0.67	-1.89	0.67	-1.044, -0.188	-1.386, -0.873, -0.359, 1.54

No	Atribut	Nilai			Nilai Batas Partisi (Bin)	
		Range	Min	Max	k = 3	k = 5
11.	exec_vm	-1.60 s/d 3.07	-1.60	3.07	-0.045, 1.511	-0.668, 0.266, 1.199, 2.133
12.	reserved_vm	-1.56 s/d 4.88	-1.56	4.88	0.583, 2.730	-0.276, 1.012, 2.300, 3.588
13.	end_data	-1.90 s/d 0.67	-1.90	0.67	-1.044, -0.188	-1.386, -0.873, -0.359, 1.54
14.	last_interval	-0.93 s/d 2.72	-0.93	2.72	0.284, 1.502	-0.204, 0.527, 1.259, 1.990
15.	nvcsw	-1.17 s/d 3.97	-1.17	3.97	0.547, 2.259	-0.138, 0.889, 1.917, 2.944
16.	nivcsw	-0.63 s/d 6.30	-0.63	6.30	1.682, 3.989	0.759, 2.143, 3.528, 4.912
17.	min_flt	-0.15 s/d 18.30	-0.15	18.30	5.999, 12.417	3.540, 7.229, 10.917, 14.606
18.	maj_flt	-1.9 s/d 0.67	-1.9	0.67	-1.044, -0.188	-1.386, -0,873, -0,359, 0.154
19.	fs_excl_counter	-0.51 s/d 7.82	-0.51	7.82	2.264, 5.041	1.153, 2.819, 4.486, 6.152
20.	utime	-1.34 s/d 3.60	-1.34	3.60	0.303, 1.951	-0.356, 0.633, 1.621, 2.610
21.	stime	-1.29 s/d 3.57	-1.29	3.57	0.333, 1.953	-0.315, 0.657, 1.629, 2.602
22.	gtime	-0.51 s/d 4.09	-0.51	4.09	1.023, 2.555	0.410, 1.330, 2.249, 3.168

Berikut hasil lengkap proses diskritisasi 3 dan 5 variabel tanpa input parameter batasan (*no define boundaries*) dengan tool rapidminer, seperti tabel di bawah ini :

Tabel 4.2 Hasil Diskritisasi 3 Bin dan 5 Bin Tanpa Parameter Batasan (Rapidminer)

Atribut	Diskritisasi (Tanpa Parameter Batasan)	
	3-Variabel	5-Variabel
millisecond	[- ∞ - -0.6], [-0.6 - 0.6], [0.6 - ∞]	[- ∞ - -1.0] , [-1.0 - - 0.3] , [-0.3 – 0.3], [0.3 – 1.0] , [1.0 - ∞]
State	[- ∞ - 15.2], [15.2 - 30.6], [30.6 - ∞]	[- ∞ - 9.1] , [9.1 - 18.3] , [18.3 – 27.6] , [27.6 – 36.8] , [36.8 - ∞]
Prio	[- ∞ - -0.6], [-0.6 - 0.6], [0.6 - ∞]	[- ∞ - -1.0] , [-1.0 - - 0.3] , [-0.3 – 0.3], [0.3 – 1.0] , [1.0 - ∞]
static_prio	[- ∞ - 0.4], [0.4 - 1.7], [1.7 - ∞]	[- ∞ - -0.1] , [-0.1 - 0.6] , [0.6 – 1.4], [1.4 – 2.2] , [2.2 - ∞]
vm_truncate_count	[- ∞ - 0.1], [0.1 - 1.8], [1.8 - ∞]	[- ∞ - -0.7] , [-0.7 - 0.4] , [0.4 – 1.5], [1.5 – 2.6] , [2.6 - ∞]
free_area_cache	[- ∞ - 3.5], [3.5 - 7.2], [7.2 - ∞]	[- ∞ - 2.0] , [2.0 - 4.2] , [4.2 – 6.4], [6.4 – 8.7] , [8.7 - ∞]
mm_users	[- ∞ - 0.1], [0.1 - 1.6], [1.6 - ∞]	[- ∞ - -0.5] , [-0.5 - 0.4] , [0.4 – 1.3], [1.3 – 2.2] , [2.2 - ∞]

Atribut	Diskritisasi (Tanpa Parameter Batasan)	
	3-Variabel	5-Variabel
map_count	[- ∞ - 0.6], [0.6 - 2.9], [2.9 - ∞]	[- ∞ - -0.3] , [-0.3 - 1.1] , [1.1 – 2.4], [2.4 – 3.8] , [3.8 - ∞]
total_vm	[- ∞ - 2.2], [2.2 - 5.2], [5.2 - ∞]	[- ∞ - 1.0] , [1.0 - 2.8] , [2.8 – 4.6], [4.6 – 6.4] , [6.4 - ∞]
shared_vm	[- ∞ - - 1.0], [-1.0 - -0.2], -[0.2 - ∞]	[- ∞ - -1.4] , [-1.4 - -0.9] , [-0.9 – -0.4], [-0.4 – 1.6] , [1.6 - ∞]
exec_vm	[- ∞ - 0.0], [-0.0 - 1.5], [1.5 - ∞]	[- ∞ - -0.7] , [-0.7 - 0.3] , [0.3 – 1.2], [1.2 – 2.1] , [2.1 - ∞]
reserved_vm	[- ∞ - 0.6], [0.6 - 2.7], [2.7 - ∞]	[- ∞ - -0.3] , [-0.3 - 1.0] , [1.0 – 2.3], [2.3 – 3.6] , [3.6 - ∞]
end_data	[- ∞ - - 1.0], [-1.0 - -0.2], [-0.2 - ∞]	[- ∞ - -1.4] , [-1.4 - -0.9] , [-0.9 – -0.4], [-0.4 – 1.6] , [1.6 - ∞]
last_interval	[- ∞ - 0.3], [0.3 - 1.5], [1.5 - ∞]	[- ∞ - -0.2] , [-0.2 - 0.5] , [0.5 – 1.3], [1.3 – 2.0] , [2.0 - ∞]
nvcsw	[- ∞ - 0.5], [0.5 - 2.3], [2.3 - ∞]	[- ∞ - -0.1] , [-0.1 - 0.9] , [0.9 – 1.9], [1.9 – 2.9] , [2.9 - ∞]
nivcsw	[- ∞ - 1.7], [1.7 - 4.0], [4.0 - ∞]	[- ∞ - 0.8] , [0.8 - 2.1] , [2.1 – 3.5], [3.5 – 4.9] , [4.9 - ∞]
min_flt	[- ∞ - 6.0], [6.0 - 12.1], [12.1 - ∞]	[- ∞ - - 3.5] , [-3.5 - 7.2] , [7.2 – 10.9], [10.9 – 14.6] , [14.6 - ∞]
maj_flt	[- ∞ - -1.0], [-1.0 - -0.2], [-0.2 - ∞]	[- ∞ - -1.4] , [-1.4 - - 0.9] , [-0.9 – - 0.35], [-0.35 – 0.2] , [0.2 - ∞]
fs_excl_counter	[- ∞ - 2.3], [2.3 - 5.0], [5.0 - ∞]	[- ∞ - 1.2] , [1.2 - 2.8] , [2.8 – 4.5], [4.5 – 6.2] , [6.2 - ∞]
utime	[- ∞ - 0.3], [0.3 - 2.0], [2.0 - ∞]	[- ∞ - - 0.4] , [-0.4 - 0.6] , [0.6 – 1.6], [1.6 – 2.6] , [2.6 - ∞]
stime	[- ∞ - 0.3], [0.3 - 2.0], [2.0 - ∞]	[- ∞ - - 0.3] , [-0.3 - 0.7] , [0.7 – 1.6], [1.6 – 2.6] , [2.6 - ∞]
gtime	[- ∞ - 1.0], [1.0 - 2.6], [2.6 - ∞]	[- ∞ - 0.4] , [0.4 - 1.3] , [1.3 – 2.2], [2.2 – 3.2] , [3.2 - ∞]

b. Binning Dengan Menetapkan Parameter Batasan (define boundaries)

Binning dengan cara menetapkan batasan nilai yang di akan di bin, nilai yang akan kita gunakan sebagai batasan yaitu -1 dan 1, sehingga bin yang terbentuk pada setiap atribut dataset sebagai berikut :

1. Atribut millisecond (Rentang Nilai -1.73 s/d 1.73)

   Untuk k = 3

   Ditentukan max = 1, min = -1

   sehingga nilai w = (max-min)/3 = 0.667

   Batas interval : min+w, min+(k-1)w sehingga

   nilai batas intervalnya -0.333 dan 0.333 sehingga batas partisi yang terbentuk pada batas nilai -1, -0.333, 0.333 dan 1 kemudian terbentuk batasan interval pada masing-masing bin yaitu

   Range 1 : -∞ - 1

Range 2 : -1 - -0.333

Range 3 : -0.333 – 0.333

Range 4 : 0.333 – 1

Range 5 : 1 - ∞

Untuk k = 5

Ditentukan max = 1, min = -1

sehingga nilai w = (max-min)/5 = 0.4

Batas interval : min+w, min+2w, min+3w, min+(k-1)w sehingga

nilai batas intervalnya -0.6, -0.2, 0.2 dan 0.6 sehingga batas partisi yang terbentuk pada batas nilai -1, -0.6, -0.2, 0.2, 0.6 dan 1 kemudian terbentuk batasan interval pada masing-masing bin yaitu

Range 1 : -∞ - 1

Range 2 : -1 - -0.6

Range 3 : -0.6 – - 0.2

Range 4 : -0.2 – - 0.2

Range 5 : 0.2 – 0.6

Range 6 : 0.6 – 1

Range 7 : 1 - ∞

2. Atribut state (Rentang Nilai -0.17 s/d 46.05)

Untuk k = 3

Ditentukan max = 1, min = -1

sehingga nilai w = (max-min)/3 = 0.667

Batas interval : min+w, min+(k-1)w sehingga

nilai batas intervalnya -0.333 dan 0.333 sehingga batas partisi yang terbentuk pada batas nilai -1, -0.333, 0.333 dan 1 kemudian terbentuk batasan interval pada masing-masing bin yaitu

Range 1 : -∞ - -1

Range 2 : -1 - -0.333

Range 3 : -0.333 – 0.333

Range 4 : 0.333 – 1

Range 5 : 1 - ∞

Untuk k = 5

Ditentukan max = 1, min = -1

sehingga nilai w = (max-min)/5 = 0.4

Batas interval : min+w, min+2w, min+3w, min+(k-1)w sehingga

nilai batas intervalnya -0.6, -0.2, 0.2 dan 0.6 sehingga batas partisi yang terbentuk pada batas nilai -1, -0.6, -0.2, 0.2, 0.6 dan 1 kemudian terbentuk batasan interval pada masing-masing bin yaitu

Range 1 : -∞ - -1

Range 2 : -1 - -0.6

Range 3 : -0.6 – - 0.2

Range 4 : -0.2 – - 0.2

Range 5 : 0.2 – 0.6

Range 6 : 0.6 – 1

Range 7 : 1 - ∞

Sehingga terbentuk nilai-nilai partisi 3 Bin dan 5 Bin dengan parameter batasan min max[-1,1] hasil penghitungan manual untuk semua atribut seperti tabel di bawah ini :

Tabel 4.3 Hasil Diskritisasi 3 Bin dan 5 Bin dengan Parameter Batasan (Manual)

No	Atribut	Nilai			Nilai Batas Partisi (Bin)	
		Range	Min	Max	k = 3	k = 5
1.	millisecond	-1.73 s/d 1.73	-1	1	-1, -0.333, 0.333, 1	-1, -0.6, -0.2, 0.2, 0.6, 1
2.	state	-0.17 s/d 40.05	-1	1	-1, -0.333, 0.333, 1	-1, -0.6, -0.2, 0.2, 0.6, 1
3.	prio	-1.74 s/d 1.74	-1	1	-1, -0.333, 0.333, 1	-1, -0.6, -0.2, 0.2, 0.6, 1
4.	static_prio	-0.91 s/d 2.97	-1	1	-1, -0.333, 0.333, 1	-1, -0.6, -0.2, 0.2, 0.6, 1
5.	vm_truncate_count	-1.73 s/d 3.64	-1	1	-1, -0.333, 0.333, 1	-1, -0.6, -0.2, 0.2, 0.6, 1
6.	free_area_cache	0.21 s/d 10.87	-1	1	-1, -0.333, 0.333, 1	-1, -0.6, -0.2, 0.2, 0.6, 1

No	Atribut	Nilai			Nilai Batas Partisi (Bin)	
		Range	Min	Max	k = 3	k = 5
7.	mm_users	-1.38 s/d 3.16	-1	1	-1, -0.333, 0.333, 1	-1, -0.6, -0.2, 0.2, 0.6, 1
8.	map_count	-1.63 s/d 5.13	-1	1	-1, -0.333, 0.333, 1	-1, -0.6, -0.2, 0.2, 0.6, 1
9.	total_vm	-0.84 s/d 8.15	-1	1	-1, -0.333, 0.333, 1	-1, -0.6, -0.2, 0.2, 0.6, 1
10.	shared_vm	-1.89 s/d 0.67	-1	1	-1, -0.333, 0.333, 1	-1, -0.6, -0.2, 0.2, 0.6, 1
11.	exec_vm	-1.60 s/d 3.07	-1	1	-1, -0.333, 0.333, 1	-1, -0.6, -0.2, 0.2, 0.6, 1
12.	reserved_vm	-1.56 s/d 4.88	-1	1	-1, -0.333, 0.333, 1	-1, -0.6, -0.2, 0.2, 0.6, 1
13.	end_data	-1.90 s/d 0.67	-1	1	-1, -0.333, 0.333, 1	-1, -0.6, -0.2, 0.2, 0.6, 1
14.	last_interval	-0.93 s/d 2.72	-1	1	-1, -0.333, 0.333, 1	-1, -0.6, -0.2, 0.2, 0.6, 1
15.	nvcsw	-1.17 s/d 3.97	-1	1	-1, -0.333, 0.333, 1	-1, -0.6, -0.2, 0.2, 0.6, 1
16.	nivcsw	-0.63 s/d 6.30	-1	1	-1, -0.333, 0.333, 1	-1, -0.6, -0.2, 0.2, 0.6, 1
17.	min_flt	-0.15 s/d 18.30	-1	1	-1, -0.333, 0.333, 1	-1, -0.6, -0.2, 0.2, 0.6, 1
18.	maj_flt	-1.9 s/d 0.67	-1	1	-1, -0.333, 0.333, 1	-1, -0.6, -0.2, 0.2, 0.6, 1
19.	fs_excl_counter	-0.51 s/d 7.82	-1	1	-1, -0.333, 0.333, 1	-1, -0.6, -0.2, 0.2, 0.6, 1
20.	utime	-1.34 s/d 3.60	-1	1	-1, -0.333, 0.333, 1	-1, -0.6, -0.2, 0.2, 0.6, 1
21.	stime	-1.29 s/d 3.57	-1	1	-1, -0.333, 0.333, 1	-1, -0.6, -0.2, 0.2, 0.6, 1
22.	gtime	-0.51 s/d 4.09	-1	1	-1, -0.333, 0.333, 1	-1, -0.6, -0.2, 0.2, 0.6, 1

Berikut hasil lengkap proses diskritisasi 3 dan 5 variabel dengan input parameter batasan (*define boundaries*) min max[-1,1] dengan tool rapidminer, seperti tabel di bawah ini :

Tabel 4.4 Hasil Diskritisasi 3 Bin dan 5 Bin Input Parameter Batasan -1 dan 1 (Rapidminer)

Atribut	Diskritisasi (Tanpa Parameter Batasan)	
	3-Variabel	5-Variabel
millisecond	[- ∞ - -1.0], [-1.0 - - 0.3], [-0.3 – 0.3], [0.3 - 1.0], [1.0 - ∞]	[- ∞ - -1.0] , [-1.0 - - 0.6] , [-0.6 – -0.2], [-0.2 – 0.2] , [0.2 – 0.6] , [0.6 – 1.0] , [1.0 - ∞]
state	[- ∞ - -1.0], [-1.0 - - 0.3], [-0.3 – 0.3], [0.3 - 1.0], [1.0 - ∞]	[- ∞ - -1.0] , [-1.0 - - 0.6] , [-0.6 – -0.2], [-0.2 – 0.2] , [0.2 – 0.6] , [0.6 – 1.0] , [1.0 - ∞]
prio	[- ∞ - -1.0], [-1.0 - - 0.3], [-0.3 – 0.3], [0.3 - 1.0], [1.0 - ∞]	[- ∞ - -1.0] , [-1.0 - - 0.6] , [-0.6 – -0.2], [-0.2 – 0.2] , [0.2 – 0.6] , [0.6 – 1.0] , [1.0 - ∞]

Atribut	Diskritisasi (Tanpa Parameter Batasan)	
	3-Variabel	5-Variabel
static_prio	[- ∞ - -1.0], [-1.0 - - 0.3], [-0.3 – 0.3], [0.3 - 1.0], [1.0 - ∞]	[- ∞ - -1.0] , [-1.0 - - 0.6] , [-0.6 – -0.2], [-0.2 – 0.2] , [0.2 – 0.6] , [0.6 – 1.0] , [1.0 - ∞]
vm_truncate_count	[- ∞ - -1.0], [-1.0 - - 0.3], [-0.3 – 0.3], [0.3 - 1.0], [1.0 - ∞]	[- ∞ - -1.0] , [-1.0 - - 0.6] , [-0.6 – -0.2], [-0.2 – 0.2] , [0.2 – 0.6] , [0.6 – 1.0] , [1.0 - ∞]
free_area_cache	[- ∞ - -1.0], [-1.0 - - 0.3], [-0.3 – 0.3], [0.3 - 1.0], [1.0 - ∞]	[- ∞ - -1.0] , [-1.0 - - 0.6] , [-0.6 – -0.2], [-0.2 – 0.2] , [0.2 – 0.6] , [0.6 – 1.0] , [1.0 - ∞]
mm_users	[- ∞ - -1.0], [-1.0 - - 0.3], [-0.3 – 0.3], [0.3 - 1.0], [1.0 - ∞]	[- ∞ - -1.0] , [-1.0 - - 0.6] , [-0.6 – -0.2], [-0.2 – 0.2] , [0.2 – 0.6] , [0.6 – 1.0] , [1.0 - ∞]
map_count	[- ∞ - -1.0], [-1.0 - - 0.3], [-0.3 – 0.3], [0.3 - 1.0], [1.0 - ∞]	[- ∞ - -1.0] , [-1.0 - - 0.6] , [-0.6 – -0.2], [-0.2 – 0.2] , [0.2 – 0.6] , [0.6 – 1.0] , [1.0 - ∞]
total_vm	[- ∞ - -1.0], [-1.0 - - 0.3], [-0.3 – 0.3], [0.3 - 1.0], [1.0 - ∞]	[- ∞ - -1.0] , [-1.0 - - 0.6] , [-0.6 – -0.2], [-0.2 – 0.2] , [0.2 – 0.6] , [0.6 – 1.0] , [1.0 - ∞]
shared_vm	[- ∞ - -1.0], [-1.0 - - 0.3], [-0.3 – 0.3], [0.3 - 1.0], [1.0 - ∞]	[- ∞ - -1.0] , [-1.0 - - 0.6] , [-0.6 – -0.2], [-0.2 – 0.2] , [0.2 – 0.6] , [0.6 – 1.0] , [1.0 - ∞]
exec_vm	[- ∞ - -1.0], [-1.0 - - 0.3], [-0.3 – 0.3], [0.3 - 1.0], [1.0 - ∞]	[- ∞ - -1.0] , [-1.0 - - 0.6] , [-0.6 – -0.2], [-0.2 – 0.2] , [0.2 – 0.6] , [0.6 – 1.0] , [1.0 - ∞]
reserved_vm	[- ∞ - -1.0], [-1.0 - - 0.3], [-0.3 – 0.3], [0.3 - 1.0], [1.0 - ∞]	[- ∞ - -1.0] , [-1.0 - - 0.6] , [-0.6 – -0.2], [-0.2 – 0.2] , [0.2 – 0.6] , [0.6 – 1.0] , [1.0 - ∞]
end_data	[- ∞ - -1.0], [-1.0 - - 0.3], [-0.3 – 0.3], [0.3 - 1.0], [1.0 - ∞]	[- ∞ - -1.0] , [-1.0 - - 0.6] , [-0.6 – -0.2], [-0.2 – 0.2] , [0.2 – 0.6] , [0.6 – 1.0] , [1.0 - ∞]
last_interval	[- ∞ - -1.0], [-1.0 - - 0.3], [-0.3 – 0.3], [0.3 - 1.0], [1.0 - ∞]	[- ∞ - -1.0] , [-1.0 - - 0.6] , [-0.6 – -0.2], [-0.2 – 0.2] , [0.2 – 0.6] , [0.6 – 1.0] , [1.0 - ∞]
nvcsw	[- ∞ - -1.0], [-1.0 - - 0.3], [-0.3 – 0.3], [0.3 - 1.0], [1.0 - ∞]	[- ∞ - -1.0] , [-1.0 - - 0.6] , [-0.6 – -0.2], [-0.2 – 0.2] , [0.2 – 0.6] , [0.6 – 1.0] , [1.0 - ∞]
nivcsw	[- ∞ - -1.0], [-1.0 - - 0.3], [-0.3 – 0.3], [0.3 - 1.0], [1.0 - ∞]	[- ∞ - -1.0] , [-1.0 - - 0.6] , [-0.6 – -0.2], [-0.2 – 0.2] , [0.2 – 0.6] , [0.6 – 1.0] , [1.0 - ∞]
min_flt	[- ∞ - -1.0], [-1.0 - - 0.3], [-0.3 – 0.3], [0.3 - 1.0], [1.0 - ∞]	[- ∞ - -1.0] , [-1.0 - - 0.6] , [-0.6 – -0.2], [-0.2 – 0.2] , [0.2 – 0.6] , [0.6 – 1.0] , [1.0 - ∞]
maj_flt	[- ∞ - -1.0], [-1.0 - - 0.3], [-0.3 – 0.3], [0.3 - 1.0], [1.0 - ∞]	[- ∞ - -1.0] , [-1.0 - - 0.6] , [-0.6 – -0.2], [-0.2 – 0.2] , [0.2 – 0.6] , [0.6 – 1.0] , [1.0 - ∞]
fs_excl_counter	[- ∞ - -1.0], [-1.0 - - 0.3], [-0.3 – 0.3], [0.3 - 1.0], [1.0 - ∞]	[- ∞ - -1.0] , [-1.0 - - 0.6] , [-0.6 – -0.2], [-0.2 – 0.2] , [0.2 – 0.6] , [0.6 – 1.0] , [1.0 - ∞]
utime	[- ∞ - -1.0], [-1.0 - - 0.3], [-0.3 – 0.3], [0.3 - 1.0], [1.0 - ∞]	[- ∞ - -1.0] , [-1.0 - - 0.6] , [-0.6 – -0.2], [-0.2 – 0.2] , [0.2 – 0.6] , [0.6 – 1.0] , [1.0 - ∞]
stime	[- ∞ - -1.0], [-1.0 - - 0.3], [-0.3 – 0.3], [0.3 - 1.0], [1.0 - ∞]	[- ∞ - -1.0] , [-1.0 - - 0.6] , [-0.6 – -0.2], [-0.2 – 0.2] , [0.2 – 0.6] , [0.6 – 1.0] , [1.0 - ∞]
gtime	[- ∞ - -1.0], [-1.0 - - 0.3], [-0.3 – 0.3], [0.3 - 1.0], [1.0 - ∞]	[- ∞ - -1.0] , [-1.0 - - 0.6] , [-0.6 – -0.2], [-0.2 – 0.2] , [0.2 – 0.6] , [0.6 – 1.0] , [1.0 - ∞]

4.2 Decision Tree C4.5

Salah satu algoritma yang digunakan dalam penelitian ini yaitu decision tree C4.5 yang menggunakan kriteria Gini Index data kontinyu, dengan tahapan prosesnya secara rinci dapat dijelaskan sebagai berikut :

a. Menghitung gini index masing-masing atribut dengan menggunakan rumus 2.5 sebagai berikut :

   Gini = 1- $\sum_{i=1}^{C}(p_i)^2$

   Dimana

   C : jumlah nilai dari masing-masing atribut

p_i : jumlah atribut dari masing-masing kelas atau labelnya

Langkah-langkah menghitung gini index setiap atribut :

1) Urutkan nilai-nilai pada atribut dari yang terkecil sampai terbesar
2) Hitung rata-rata (mean) dari setiap nilai yang bersebelahan yang telah diurutkan
3) Menghitung masing-masing Gini untuk semua nilai rata-rata pada atribut tersebut
4) Setelah semua penghitungan Gini dilakukan, nilai Gini terkecil ditentukan sebagai split untuk atribut tersebut

b. Memilih split atribut dengan gini index terkecil untuk dijadikan node(simpul)
c. Melakukan pengecekan value pada atribut di atas, apakah masih ada yang berada di kelas/label yang berbeda?
d. Jika value masih masih ada yang berada pada kelas/label yang berbeda, lakukan penghitungan lagi gini index masing-masing atributnya
e. Ulangi proses tersebut dan proses akan berakhir jika value pada atribut sudah berada pada kelas/label yang sama.

Berdasarkan langkah-langkah tahapan decision tree C4.5 dengan kriteria gini index yang telah diuraikan di atas, peneliti akan menggunakan data sampel untuk melakukan proses penghitungan gini index terhadap atribut yang ada pada dataset malware sebanyak 50 record, penghitungan manual jika diketahui data sampel sebagai berikut :

Tabel 4.5 Dataset Malware Sampel 50 Record

no	shared_vm	min_flt	static_prio	gtime	state	classification
1	-1.257740098	-0.147905307	1.336524274	-0.509163234	-0.168524841	benign
2	0.66725444	-0.075866368	-0.506508685	-0.509163234	-0.168524841	benign
3	0.66725444	-0.075866368	0.526292624	-0.509163234	-0.168524841	benign
4	0.66725444	-0.075866368	-0.506508685	-0.509163234	-0.168524841	benign
5	-1.257740098	-0.147905307	2.717280487	-0.202700549	-0.168524841	benign
6	-1.257740098	-0.147905307	0.865787278	-0.509163234	-0.168524841	malware

no	shared_vm	min_flt	static_prio	gtime	state	classification
7	0.66725444	-0.075866368	-0.910214468	-0.509163234	-0.164149579	malware
8	0.66725444	-0.075866368	-0.022972848	-0.509163234	-0.168524841	benign
9	0.66725444	-0.147905307	-0.665301072	-0.202700549	-0.168524841	benign
10	0.66725444	-0.147905307	-0.542952839	-0.509163234	-0.168524841	benign
11	0.66725444	-0.075866368	-0.901537288	-0.509163234	-0.168524841	malware
12	0.66725444	-0.147905307	-0.633195508	-0.509163234	-0.168524841	benign
13	-1.257740098	-0.147905307	0.867305784	-0.202700549	-0.168524841	malware
14	0.66725444	-0.147905307	-0.189357768	-0.509163234	-0.168524841	benign
15	-1.257740098	-0.147905307	0.867305784	-0.202700549	-0.168524841	malware
16	0.66725444	-0.147905307	0.867305784	-0.509163234	-0.168524841	malware
17	0.66725444	-0.075866368	-0.2271035	-0.509163234	-0.120396962	benign
18	0.66725444	-0.075866368	-0.907394385	-0.509163234	-0.164149579	malware
19	0.66725444	-0.075866368	-0.898500276	-0.509163234	-0.168524841	malware
20	-1.257740098	-0.147905307	0.86925815	-0.509163234	-0.168524841	malware
21	0.66725444	-0.147905307	-0.456831831	-0.202700549	-0.168524841	benign
22	-1.899404944	-0.075866368	-0.852945082	1.329612875	-0.168524841	malware
23	0.66725444	-0.075866368	-0.909997539	-0.509163234	-0.164149579	malware
24	-1.899404944	-0.075866368	-0.86270691	1.02315019	-0.168524841	malware
25	0.66725444	-0.075866368	-0.898717205	-0.509163234	-0.168524841	malware
26	-1.257740098	-0.075866368	0.888781804	-0.509163234	-0.168524841	malware
27	0.66725444	-0.147905307	-0.848172634	-0.509163234	-0.164149579	malware
28	0.66725444	-0.075866368	-0.224066487	-0.202700549	-0.168524841	benign
29	0.66725444	-0.147905307	-0.848172634	-0.509163234	-0.164149579	malware
30	0.66725444	-0.075866368	1.267974555	-0.202700549	-0.168524841	benign
31	0.66725444	-0.147905307	1.62417278	-0.202700549	-0.168524841	benign
32	0.66725444	-0.075866368	-0.593063552	0.716687506	-0.168524841	benign
33	0.66725444	-0.075866368	-0.901537288	-0.509163234	-0.168524841	malware
34	0.66725444	-0.075866368	-0.633195508	-0.509163234	-0.168524841	benign
35	0.66725444	-0.075866368	-0.848172634	-0.509163234	-0.164149579	malware
36	0.66725444	-0.075866368	-0.84556948	-0.509163234	-0.168524841	malware
37	-1.257740098	-0.147905307	0.867739643	-0.509163234	-0.168524841	malware
38	0.66725444	-0.075866368	-0.831252133	-0.509163234	-0.168524841	malware
39	0.66725444	-0.147905307	0.135819539	-0.202700549	-0.168524841	benign
40	0.66725444	-0.147905307	0.135819539	-0.202700549	-0.168524841	benign
41	0.66725444	-0.075866368	-0.633195508	-0.509163234	-0.168524841	benign
42	0.66725444	-0.147905307	-0.901537288	-0.509163234	-0.168524841	malware
43	0.66725444	-0.075866368	-0.844267903	-0.509163234	-0.168524841	malware
44	0.66725444	-0.075866368	0.292008773	-0.509163234	-0.168524841	benign
45	0.66725444	-0.075866368	-0.246410224	-0.509163234	-0.168524841	benign
46	0.66725444	-0.147905307	0.309363132	-0.509163234	-0.168524841	benign

no	shared_vm	min_flt	static_prio	gtime	state	classification
47	0.66725444	-0.075866368	-0.745564984	-0.202700549	-0.168524841	malware
48	-1.899404944	-0.075866368	-0.604777744	2.8619263	-0.155399056	malware
49	0.66725444	-0.075866368	-0.620179738	-0.202700549	-0.168524841	malware
50	0.66725444	-0.147905307	-0.189357768	0.103762136	-0.164149579	benign

***Hanya sebagian atribut yang dapat di tampilkan**

- Langkah 1 menghitung gini index untuk semua atribut guna memilih node 1 yang terdiri dari :
  - Langkah 1.1 menghitung nilai gini index atribut millisecond
    - Mengurutkan nilai-nilai pada atribut millisecond dari terkecil sampai terbesar
    - Menghitung rata-rata dari setiap nilai yang bersebelahan dari data yang telah diurutkan
    - Menghitung Gini Index atribut millisecond dari Tabel pemisahan seperti dibawah ini :

Tabel 4.6 Tabel Pemisahan Nilai Atribut Millisecond

Urutan Nilai	Rata-Rata	Pemisahan Nilai	Jumlah Kasifikasi	
			benign	malware
-1.730	-1.431	<=	1	0
		>	23	26
-1.131	-1.129	<=	2	0
		>	22	26
-1.128	-1.126	<=	3	0
		>	21	26
-1.124	-1.105	<=	4	0
		>	20	26
-1.086	-1.017	<=	5	0
		>	19	26
-0.947	-0.901	<=	5	1
		>	19	25
-0.854	-0.826	<=	5	2
		>	19	24
-0.798	-0.793	<=	6	2
		>	18	24
-0.788	-0.781	<=	7	2

Urutan Nilai	Rata-Rata	Pemisahan Nilai	Jumlah Kasifikasi	
			benign	malware
		>	17	24
-0.774	-0.771	<=	8	2
		>	16	24
-0.767	-0.753	<=	8	3
		>	16	23
-0.740	-0.733	<=	9	3
		>	15	23
-0.726	-0.714	<=	9	4
		>	15	22
-0.701	-0.686	<=	10	4
		>	14	22
-0.670	-0.658	<=	10	5
		>	14	21
-0.646	-0.617	<=	10	6
		>	14	20
-0.587	-0.566	<=	11	6
		>	13	20
-0.546	-0.539	<=	11	7
		>	13	19
-0.532	-0.523	<=	11	8
		>	13	18
-0.514	-0.509	<=	11	9
		>	13	17
-0.504	-0.502	<=	12	9
		>	12	17
-0.501	-0.499	<=	12	10
		>	12	16
-0.497	-0.490	<=	12	11
		>	12	15
-0.483	-0.482	<=	12	12
		>	12	14
-0.480	-0.471	<=	12	13
		>	12	13
-0.462	-0.452	<=	12	14
		>	12	12
-0.442	-0.435	<=	12	15
		>	12	11
-0.428	-0.423	<=	13	15
		>	11	11

Urutan Nilai	Rata-Rata	Pemisahan Nilai	Jumlah Kasifikasi	
			benign	malware
-0.417	-0.405	<=	13	16
		>	11	10
-0.393	-0.355	<=	14	16
		>	10	10
-0.317	-0.308	<=	15	16
		>	9	10
-0.300	-0.296	<=	16	16
		>	8	10
-0.293	-0.291	<=	16	17
		>	8	9
-0.289	-0.288	<=	17	17
		>	7	9
-0.286	-0.281	<=	17	18
		>	7	8
-0.275	-0.274	<=	17	19
		>	7	7
-0.272	-0.270	<=	17	20
		>	7	6
-0.268	-0.265	<=	17	21
		>	7	5
-0.262	-0.260	<=	18	21
		>	6	5
-0.258	-0.253	<=	19	21
		>	5	5
-0.248	-0.244	<=	20	21
		>	4	5
-0.241	-0.239	<=	20	22
		>	4	4
-0.237	-0.232	<=	20	23
		>	4	3
-0.227	-0.223	<=	21	23
		>	3	3
-0.220	-0.218	<=	22	23
		>	2	3
-0.217	-0.211	<=	23	23
		>	1	3
-0.206	-0.204	<=	23	24
		>	1	2
-0.203	-0.199	<=	23	25

Urutan Nilai	Rata-Rata	Pemisahan Nilai	Jumlah Kasifikasi	
			benign	malware
		>	1	1
-0.196	-0.184	<=	23	26
		>	1	0
-0.171				

✓ Menghitung Gini untuk nilai rata-rata -1.431

Rata-Rata	Pemisahan Nilai	Jumlah Kasifikasi	
		benign	malware
-1.431	<=	1	0
	>	23	26

Gini (<=-1.431) = 1-(1/1) ^2-(0/1) ^2=0

Gini (>-1.431) = 1-(23/49) ^2-(26/49) ^2=0.49

Gini (-1.431) = ((1/50) *0)+((49/50) *0.49)=0.49

✓ Lakukan penghitungan semua nilai Gini untuk rata-rata lainnya pada atribut millisecond

✓ Diperoleh hasil lengkap penghitungan nilai Gini untuk atribut millisecond seperti tabel dibawah ini :

Tabel 4.7 Nilai Gini Atribut Millisecond

Urutan Nilai	Rata-Rata	Pemisahan Nilai	Nilai Gini
-1.730	-1.431	<=	
		>	0.488
-1.131	-1.129	<=	
		>	0.477
-1.128	-1.126	<=	
		>	0.465
-1.124	-1.105	<=	
		>	0.452
-1.086	-1.017	<=	
		>	0.439
-0.947	-0.901	<=	
		>	0.465
-0.854	-0.826	<=	

Urutan Nilai	Rata-Rata	Pemisahan Nilai	Nilai Gini
		>	0.481
-0.798	-0.793	<=	
		>	0.471
-0.788	-0.781	<=	
		>	0.460
-0.774	-0.771	<=	
		>	0.448
-0.767	-0.753	<=	
		>	0.465
-0.740	-0.733	<=	
		>	0.453
-0.726	-0.714	<=	
		>	0.468
-0.701	-0.686	<=	
		>	0.457
-0.670	-0.658	<=	
		>	0.469
-0.646	-0.617	<=	
		>	0.479
-0.587	-0.566	<=	
		>	0.470
-0.546	-0.539	<=	
		>	0.480
-0.532	-0.523	<=	
		>	0.487
-0.514	-0.509	<=	
		>	0.493
-0.504	-0.502	<=	
		>	0.487
-0.501	-0.499	<=	
		>	0.492
-0.497	-0.490	<=	
		>	0.496
-0.483	-0.482	<=	
		>	0.498
-0.480	-0.471	<=	
		>	0.499
-0.462	-0.452	<=	
		>	0.498

Urutan Nilai	Rata-Rata	Pemisahan Nilai	Nilai Gini
-0.442	-0.435	<=	
		>	0.496
-0.428	-0.423	<=	
		>	0.499
-0.417	-0.405	<=	
		>	0.496
-0.393	-0.355	<=	
		>	0.499
-0.317	-0.308	<=	
		>	0.499
-0.300	-0.296	<=	
		>	0.498
-0.293	-0.291	<=	
		>	0.499
-0.289	-0.288	<=	
		>	0.498
-0.286	-0.281	<=	
		>	0.499
-0.275	-0.274	<=	
		>	0.499
-0.272	-0.270	<=	
		>	0.497
-0.268	-0.265	<=	
		>	0.492
-0.262	-0.260	<=	
		>	0.497
-0.258	-0.253	<=	
		>	0.499
-0.248	-0.244	<=	
		>	0.499
-0.241	-0.239	<=	
		>	0.499
-0.237	-0.232	<=	
		>	0.496
-0.227	-0.223	<=	
		>	0.499
-0.220	-0.218	<=	
		>	0.498
-0.217	-0.211	<=	

Urutan Nilai	Rata-Rata	Pemisahan Nilai	Nilai Gini
		>	0.490
-0.206	-0.204	<=	
		>	0.496
-0.203	-0.199	<=	
		>	0.499
-0.196	-0.184	<=	
		>	0.488
-0.171			

Sehingga diperoleh gini index terkecil 0.439 sebagai split untuk atribut millisecond

- Langkah 1.2 s/d 1.22 menghitung nilai gini index untuk 21 atribut lainnya, dari proses penghitungan dengan cara yang sama seperti langkah 1.1 diperoleh hasil lengkap kandidat split atribut pada node 1 berdasarkan nilai gini terkecil pada masing-masing atributnya seperti tabel di bawah ini :

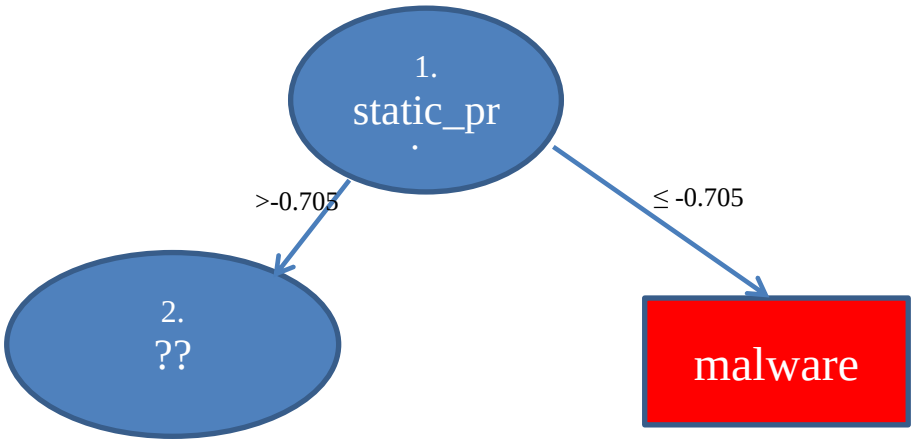
Tabel 4.8 Tabel Kandidat Split Atribut Pada Node 1

No	Atribut	Rata-Rata	Pemisahan Nilai	Gini	Total Gini
1	millisecond	-1.017	<=	0.000	0.439
			>	0.488	
2	state	-0.169	<=	0.497	0.470
			>	0.346	
3	prio	1.473	<=	0.500	0.470
			>	0.000	
4	static_prio	-0.705	<=	0.000	0.262
			>	0.397	
5	vm_truncate_count	-0.276	<=	0.375	0.402
			>	0.426	
6	free_area_cache	-0.152	<=	0.472	0.419
			>	0.305	
7	mm_users	0.687	<=	0.499	0.459
			>	0.000	
8	map_count	-0.411	<=	0.408	0.445
			>	0.471	
9	total_vm	-0.718	<=	0.000	0.448

No	Atribut	Rata-Rata	Pemisahan Nilai	Gini	Total Gini
			>	0.498	
10	shared_vm	-0.295	<=	0.298	0.449
			>	0.492	
11	exec_vm	-0.906	<=	0.278	0.437
			>	0.488	
12	reserved_vm	1.625	<=	0.497	0.477
			>	0.000	
13	end_data	-1.258	<=	0.298	0.449
			>	0.492	
14	last_interval	-0.935	<=	0.432	0.448
			>	0.458	
15	nvcsw	-0.648	<=	0.266	0.372
			>	0.437	
16	nivcsw	-0.597	<=	0.391	0.444
			>	0.467	
17	min_flt	-0.148	<=	0.490	0.487
			>	0.485	
18	maj_flt	-1.258	<=	0.298	0.449
			>	0.492	
19	fs_excl_counter	1.338	<=	0.494	0.465
			>	0.000	
20	utime	-0.318	<=	0.311	0.320
			>	0.330	
21	stime	-1.287	<=	0.432	0.448
			>	0.458	
22	gtime	0.870	<=	0.500	0.470
			>	0.000	

Berdasarkan kandidat split atribut Sesuai tabel di atas diperoleh nilai Gini index terkecil yaitu atribut static_prio dengan nilai 0.262 pada nilai rata-rata -0.705 ditentukan sebagai split position,

sehingga split position (node 1) yang terbentuk seperti gambar dibawah ini :



Gambar 4.1 Pohon Keputusan Node 1

- Langkah 2 menghitung gini index pada tabel yang tersisa dari proses pada node 1 untuk semua atribut guna memilih node 2 yang dengan cara yang sama seperti pada langkah 1.1 s/d 1.22 dengan menjalankan langkah 2.1 s./d 22 sehingga diperoleh hasil lengkap kandidat split atribut pada node 2 berdasarkan nilai gini terkecil pada masing-masing atributnya seperti tabel di bawah ini :

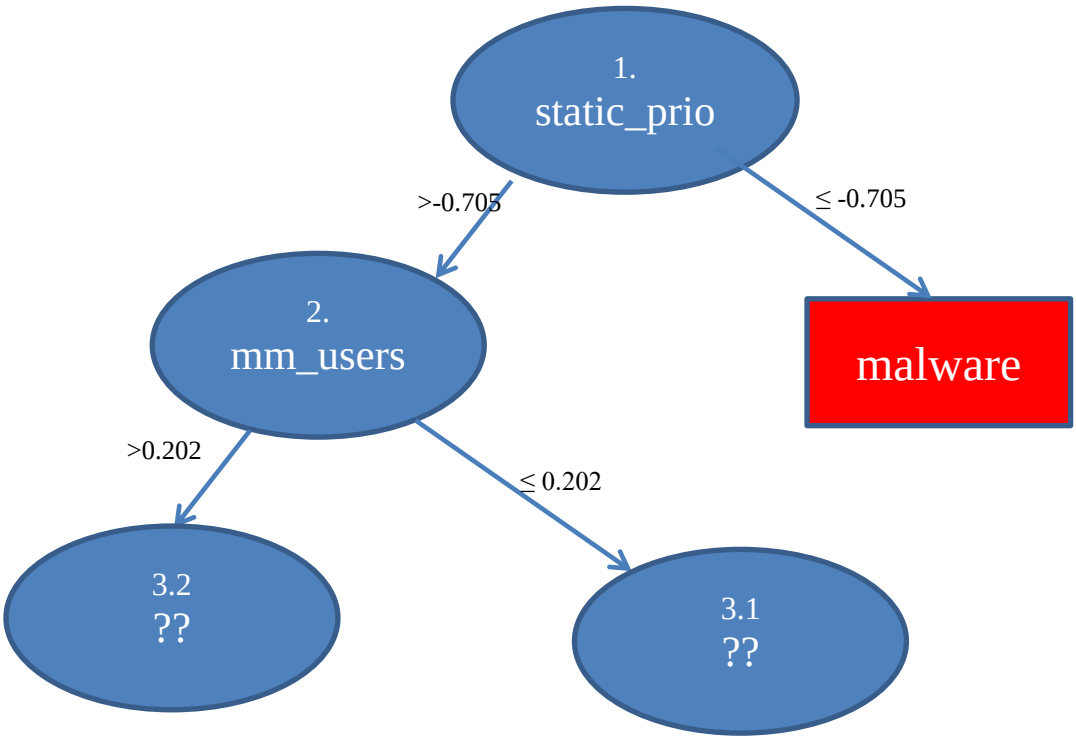
Tabel 4.9 Tabel Kandidat Split Atribut Pada Node 2

No	Atribut	Rata-Rata	Pemisahan Nilai	Gini	Total Gini
1	millisecond	-0.210	<=	0.358	0.366
			>	0.444	
2	state	-0.160	<=	0.383	0.390
			>	0.500	
3	prio	0.886	<=	0.346	0.374
			>	0.500	
4	static_prio	0.696	<=	0.165	0.264
			>	0.463	
5	vm_truncate_count	-0.468	<=	0.000	0.364
			>	0.444	
6	free_area_cache	-0.152	<=	0.159	0.238
			>	0.420	
7	mm_users	0.202	<=	0.087	0.190

No	Atribut	Rata-Rata	Pemisahan Nilai	Gini	Total Gini
			>	0.397	
8	map_count	0.156	<=	0.198	0.334
			>	0.498	
9	total_vm	-0.317	<=	0.000	0.300
			>	0.495	
10	shared_vm	-1.258	<=	0.346	0.205
			>	0.153	
11	exec_vm	0.037	<=	0.124	0.326
			>	0.494	
12	reserved_vm	-0.509	<=	0.000	0.341
			>	0.469	
13	end_data	-1.258	<=	0.346	0.205
			>	0.153	
14	last_interval	1.173	<=	0.320	0.291
			>	0.000	
15	nvcsw	-0.063	<=	0.000	0.287
			>	0.499	
16	nivcsw	-0.569	<=	0.000	0.300
			>	0.495	
17	min_flt	-0.148	<=	0.444	0.388
			>	0.320	
18	maj_flt	-1.258	<=	0.346	0.205
			>	0.153	
19	fs_excl_counter	-0.513	<=	0.320	0.380
			>	0.473	
20	utime	2.011	<=	0.375	0.364
			>	0.000	
21	stime	-1.287	<=	0.000	0.300
			>	0.495	
22	gtime	1.789	<=	0.375	0.364
			>	0.000	

Berdasarkan kandidat split atribut sesuai tabel di atas diperoleh nilai Gini index terkecil yaitu atribut mm_users dengan nilai 0.190 pada nilai rata-rata 0.202 ditentukan sebagai split position,

shingga split position (node 2) yang terbentuk seperti gambar dibawah ini :



Gambar 4.2 Pohon Keputusan Node 2

- Langkah 3 menghitung gini index pada tabel yang tersisa dari proses pada node 2, karena pada node 2 hasilnya terbentuk node 3.1 dan 3.2 maka dilakukan proses penghitungan gini untuk semua atribut guna memilih node 3.1 dan 3.2 dengan cara yang sama seperti langkah-langkah sebelumnya sehingga diperoleh hasil lengkap kandidat split atribut pada node 3.1 dan 3.2 berdasarkan nilai gini terkecil pada masing-masing atributnya seperti tabel di bawah ini :

Tabel 4.10 Tabel Kandidat Split Atribut Pada Node 3.1

No	Atribut	Rata-Rata	Pemisahan Nilai	Gini	Total Gini
1	millisecond	-0.617	<=	0.180	0.082
			>	0.000	
2	state	-0.169	<=	0.095	0.086
			>	0.000	
3	prio	-1.395	<=	0.000	0.086
			>	0.111	
4	static_prio	0.697	<=	0.000	0.068
			>	0.375	
5	vm_truncate_count	-0.276	<=	0.245	0.078
			>	0.000	
6	free_area_cache	-0.206	<=	0.000	0.073
			>	0.320	
7	mm_users	-0.384	<=	0.000	0.082
			>	0.180	
8	map_count	0.156	<=	0.000	0.078
			>	0.245	
9	total_vm	0.123	<=	0.000	0.078
			>	0.245	
10	shared_vm	-0.295	<=	0.000	0.087
			>	0.091	
11	exec_vm	-0.098	<=	0.180	0.082
			>	0.000	
12	reserved_vm	-0.158	<=	0.180	0.082
			>	0.000	
13	end_data	-0.295	<=	0.000	0.087
			>	0.091	
14	last_interval	-0.935	<=	0.000	0.081
			>	0.198	
15	nvcsw	0.018	<=	0.000	0.080
			>	0.219	
16	nivcsw	-0.294	<=	0.000	0.080
			>	0.219	
17	min_flt	-0.148	<=	0.153	0.083
			>	0.153	
18	maj_flt	-0.295	<=	0.000	0.087
			>	0.091	
19	fs_excl_counter	-0.513	<=	0.000	0.080

No	Atribut	Rata-Rata	Pemisahan Nilai	Gini	Total Gini
			>	0.219	
20	utime	-0.500	<=	0.320	0.073
			>	0.000	
21	stime	-1.287	<=	0.000	0.081
			>	0.198	
22	gtime	-0.509	<=	0.133	0.084
			>	0.000	

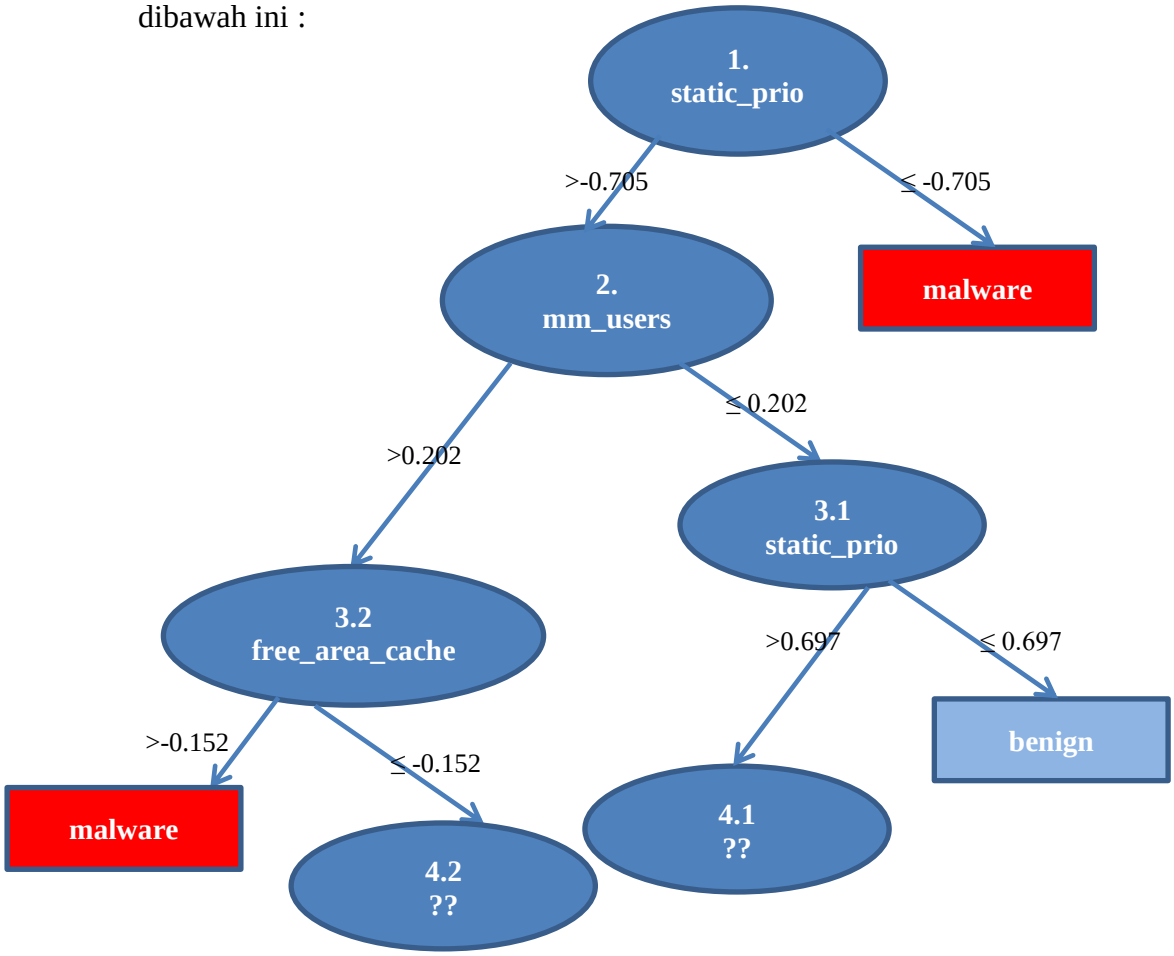
Tabel 4.11 Tabel Kandidat Split Atribut Pada Node 3.2

No	Atribut	Rata-Rata	Pemisahan Nilai	Gini	Total Gini
1	millisecond	-1.017	<=	0.000	0.291
			>	0.320	
2	state	-0.169	<=	0.420	0.382
			>	0.000	
3	prio	0.886	<=	0.469	0.341
			>	0.000	
4	static_prio	1.803	<=	0.320	0.291
			>	0.000	
5	vm_truncate_count	1.792	<=	0.320	0.291
			>	0.000	
6	free_area_cache	-0.152	<=	0.375	0.136
			>	0.000	
7	mm_users	0.687	<=	0.490	0.312
			>	0.000	
8	map_count	-0.397	<=	0.000	0.364
			>	0.444	
9	total_vm	-0.317	<=	0.000	0.162
			>	0.198	
10	shared_vm	-1.258	<=	0.219	0.280
			>	0.444	
11	exec_vm	0.935	<=	0.245	0.338
			>	0.500	
12	reserved_vm	0.396	<=	0.500	0.273
			>	0.000	

No	Atribut	Rata-Rata	Pemisahan Nilai	Gini	Total Gini
13	end_data	-1.258	<=	0.219	0.280
			>	0.444	
14	last_interval	1.173	<=	0.469	0.341
			>	0.000	
15	nvcsw	0.092	<=	0.000	0.273
			>	0.500	
16	nivcsw	-0.559	<=	0.000	0.291
			>	0.320	
17	min_flt	-0.148	<=	0.278	0.370
			>	0.480	
18	maj_flt	-1.258	<=	0.219	0.159
			>	0.000	
19	fs_excl_counter	-0.051	<=	0.346	0.374
			>	0.500	
20	utime	0.275	<=	0.000	0.273
			>	0.500	
21	stime	-0.072	<=	0.375	0.396
			>	0.408	
22	gtime	-0.203	<=	0.420	0.382
			>	0.000	

Berdasarkan kandidat split atribut sesuai 2 tabel di atas diperoleh nilai Gini index terkecil pada node 3.1 yaitu atribut static_prio dengan nilai 0.068 pada nilai rata-rata 0.697 ditentukan sebagai split positionnya, nilai Gini index terkecil pada node 3.2 yaitu atribut free_area_cache dengan nilai 0.136 pada nilai rata-rata -0.152 ditentukan sebagai split positionnya,

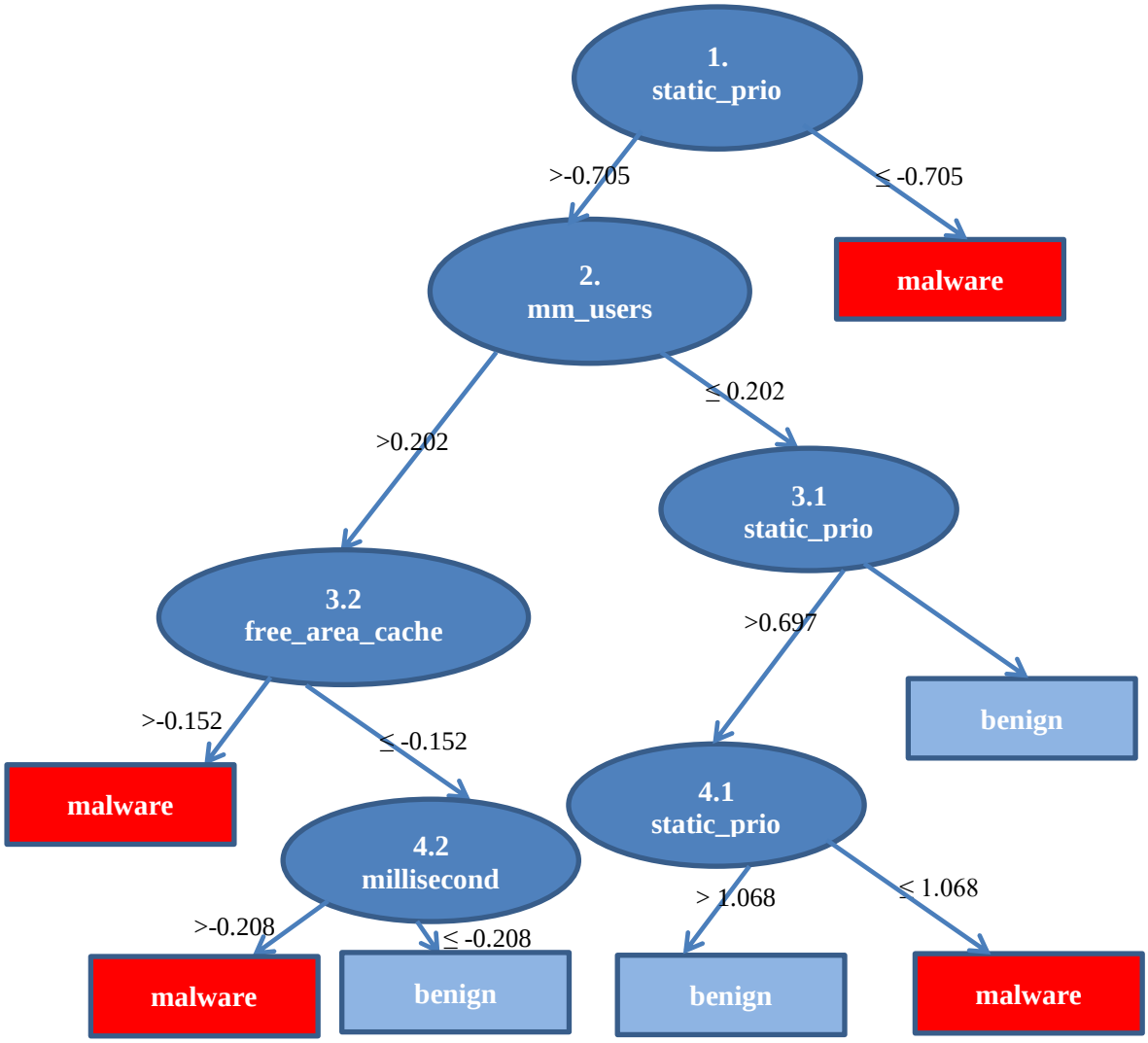
sehingga split position (node 3.1 dan 3.2) yang terbentuk seperti gambar dibawah ini :



Gambar 4.3 Pohon Keputusan Node 3

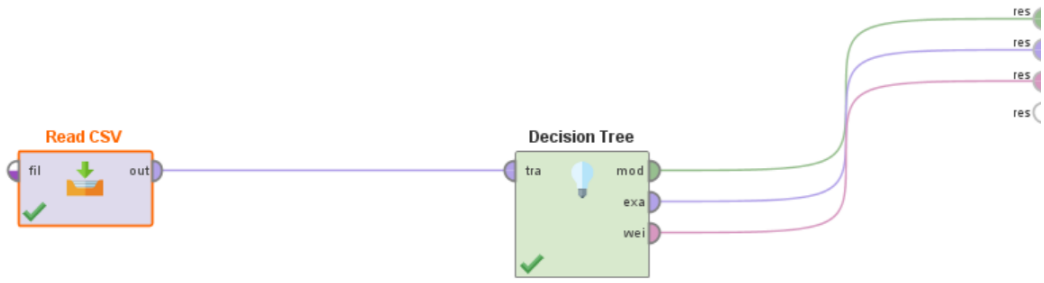
- Langkah 4 menghitung gini index pada tabel yang tersisa dari proses pada node, karena pada node 4 hasilnya juga terbentuk 2 node yaitu node 4.1 dan 4.2 maka dilakukan proses penghitungan gini untuk semua atribut guna memilih node 4.1 dan 4.2 dengan cara yang sama seperti langkah-langkah sebelumnya sehingga diperoleh split atribut pada node 4.1 yaitu static_prio dengan nilai gini 0 pada rata-rata 1.068 sedangkan split atribut pada node 4.2 yaitu millisecond dengan nilai gini 0 pada rata-rata -0.208. Pada proses ini atribut

sudah berada pada kelas/label yang sama, sehingga akan terbentuk pohon keputusan akhir seperti gambar dibawah ini :

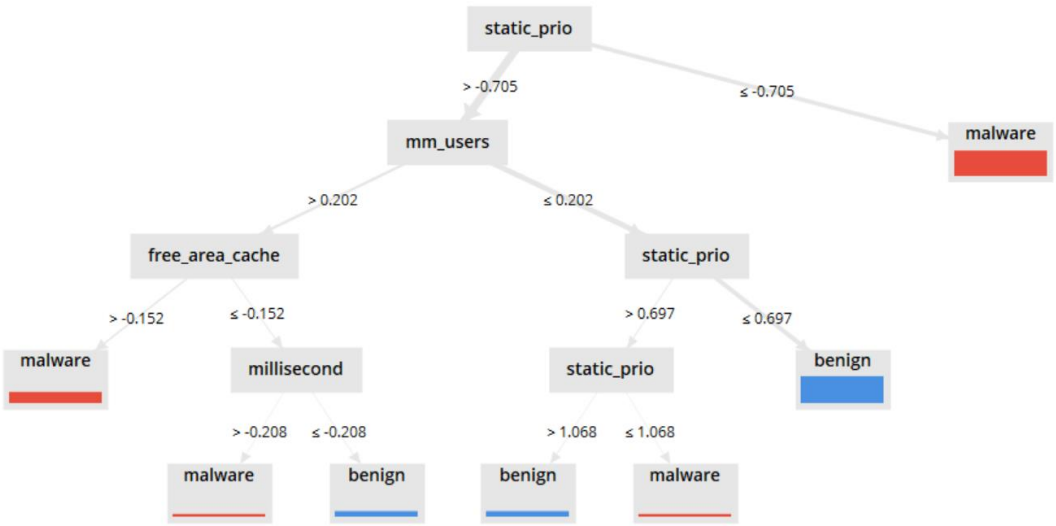


Gambar 4.4 Pohon Keputusan Akhir

Implementasi pemodelan klasifikasi decision tree (C4.5) dengan data sampel lengkap menggunakan tools Rapidminer seperti terlihat pada gambar di bawah ini :



Gambar 4.5 Pemodelan klasifikasi decision tree (Rapidminer)



Gambar 4.6 Hasil Pemodelan klasifikasi decision tree (Rapidminer)

Setelah pohon keputusan terbentuk sesuai pada gambar di atas, maka akan dihasilkan rule dari pohon keputusan tersebut adalah sebagai berikut :

a. If static_prio <= -0.705 Then hasil = malware
b. If static_prio > -0.705 and mm_users <= 0.202 and static_prio <= 0.697 Then hasil = benign

c. If static_prio > 0.697 and mm_users <= 0.202 and static_prio <= 1.068 Then hasil = malware
d. If static_prio > 1.068 and mm_users <= 0.202 Then hasil = benign
e. If static_prio > -0.705 and mm_users > 0.202 and free_area_chace >-0.152 Then hasil = malware
f. If static_prio > -0.705 and mm_users > 0.202 and free_area_chace <=-0.152 and millisecond <=-0.208 Then hasil = benign
g. If static_prio > -0.705 and mm_users > 0.202 and free_area_chace <=-0.152 and millisecond >-0.208 Then hasil = malware

4.3 Naïve Bayes

Algoritma lainnya yang digunakan dalam penelitian ini yaitu naïve bayes, dengan langkah-langkah tahapan prosesnya adalah sebagai berikut :

a. Menghitung *mean* dan *standar deviasi* untuk setiap data atribut.

   Mean dengan menggunakan rumus :

   $Mean = \frac{jumlah\ nilai}{banyaknya\ data}$

   Sementara untuk *standar deviasi* setiap atribut menggunakan rumus :

   $S = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}}$

   Dimana :

   S : standar deviasi

   x_i : nilai x ke-i

   $\bar{x}$: mean

   n : ukuran sampel

b. Menghitung probabilitas prior untuk tiap kelas dengan rumus :

   $P(C) = \frac{Nj}{N}$

   Dimana :

   Nj : jumlah data pada suatu class

   N : jumlah total data

c. Menghitung probabilitas Distribusi Gaussian setiap kategori kelas dengan acuan nilai mean dan standar deviasi sebelumnya untuk setiap atributnya dengan rumus :

$$P = (X_i = x_i \mid Y_i = y_i) = \frac{1}{\sqrt{2\pi\sigma ij}} e^{\frac{(x_i - u_{ij})^2}{2\,\sigma^2 ij}}$$

Dimana :

P : peluang

X_i : atribut ke-i

x_i : nilai atribut ke-i

Y : kelas yang dicari

yj : sub kelas Y yang dicari

u : rata-rata seluruh atribut

σ : deviasi standar dari seluruh atribut

d. Menghitung probabilitas posterior dengan rumus :

$$P(C|X_{1,\ldots,}X_{n)} = P(C) \prod_{i=1}^{n} P(x_i|C)$$

$$P(C|X) = \frac{P(x|C)P(c)}{P(x)}$$

Dimana :

x : data dengan kelas yang belum diketahui

c : hipotesis data yang merupakan kelas spesifik

P(C|X) : probabilitas hipotesis berdasarkan kondisi (*posterior probability)*

P(c) : Probabilitas hipotesis (class prior probability)

P(x|c) : Probabilitas berdasarkan kondisi pada hipotesis (likelihood)

e. Klasifikasi hasil akhir dari data testing

Berdasarkan langkah-langkah tahapan naiive bayes yang telah diuraikan di atas, peneliti akan menggunakan data sampel yang sama yang telah digunakan sebelumnya pada penghitungan Algoritma C4.5 untuk melakukan proses penghitungan dengan algoritma naïve bayes, dengan tahapan sebagai berikut :

1) Menghitung Mean dan Standar Deviasi dari setiap atribut pada setiap kelas/label dengan hasil seperti tabel di bawah ini :

Tabel 4.12 Tabel Mean dan Standard Deviasi

NO	ATRIBUT	MEAN		STD DEVIASI	
		BENIGN	MALWARE	BENIGN	MALWARE
1	millisecond	-0.600874589	-0.460057271	0.406590986	0.211738742
2	state	-0.16633721	-0.167010327	0.009825827	0.003016662
3	prio	-0.2313082	-0.256520328	0.988088114	1.109403602
4	static_prio	0.08646808	-0.380297438	0.859208764	0.778015588
5	vm_truncate_count	0.185620593	-0.4753515	0.924808387	0.767366973
6	free_area_cache	0.063517955	-0.034141987	0.894900856	0.269124071
7	mm_users	-0.343379377	-0.228606113	0.526319187	1.0129685
8	map_count	0.154276999	-0.275647185	1.301910413	0.752718622
9	total_vm	-0.144203796	-0.2515767	0.541374914	0.512568528
10	shared_vm	0.506838229	-0.073128075	0.543483422	1.053461381
11	exec_vm	-0.030440351	-0.37227473	0.643438194	0.89090481
12	reserved_vm	-0.062765998	-0.128109535	0.850768314	0.791643981
13	end_data	0.506838229	-0.073128075	0.543483422	1.053461381
14	last_interval	-0.229946424	-0.114493326	0.969957209	0.870482567
15	nvcsw	0.154220204	-0.48161017	0.842923529	0.658398695
16	nivcsw	-0.247950875	-0.266793781	0.529114755	0.484776535
17	min_flt	-0.111885838	-0.100802924	0.03679417	0.034950738
18	maj_flt	0.506838229	-0.073128075	0.543483422	1.053461381
19	fs_excl_counter	0.026603826	-0.175156995	0.961913698	0.499509537
20	utime	0.109338266	-0.491655866	0.729469936	0.794209638
21	stime	-0.325264546	-0.118820875	1.132083643	0.804722324
22	gtime	-0.330393334	-0.202700549	0.284584695	0.775296082

2) Menghitung probabilitas prior pada masing-masing kelas yang ada

Kelas Benign = $\frac{Jumlah\ Kelas\ Benigh}{Jumlah\ (Kelas\ Benign+Kelas\ Malware)} = \frac{24}{50} = 0.48$

Kelas Malware = $\frac{Jumlah\ Kelas\ Malware}{Jumlah\ (Kelas\ Benign+Kelas\ Malware)} = \frac{26}{50} = 0.52$

3) Menghitung probabilitas Distribusi Gaussian setiap atribut kategori kelas dengan acuan nilai mean dan standar deviasi sebelumnya berdasarkan data testing yang belum diketahui kelas/labelnya.

Jika diketahui data testing :

millisecond : -1.047886023
state : -0.168524841
prio : -0.607451877

static_prio	:	0.888781804
vm_truncate_count	:	1.270472053
free_area_cache	:	-0.120154414
mm_users	:	2.539611178
map_count	:	0.495828062
total_vm	:	-0.139396054
shared_vm	:	-1.257740098
exec_vm	:	2.034377411
reserved_vm	:	0.893160464
end_data	:	-1.257740098
last_interval	:	1.946996922
nvcsw	:	1.390471251
nivcsw	:	-0.549802069
min_flt	:	-0.075866368
maj_flt	:	-1.257740098
fs_excl_counter	:	-0.050540018
utime	:	2.007341758
stime	:	1.143212133
gtime	:	-0.509163234
classification	:	???

Penghitungan probabilitas Distribusi Gaussian pada kelas benign :

$$P(\text{millisecond} \mid \text{benign}) = \frac{1}{\sqrt{2*3.14*0.406590986}} e^{\frac{((-1.047886023)-(-0.600874589))^2}{2*0.406590986^2}} =$$

0.341958552778149

Penghitungan lengkap untuk 21 atribut yang lain pada kelas benign, dengan hasil sebagai berikut :

P (state | benign)	=	3.927099064
P (Prio | benign)	=	0.373382743
P (static_prio | benign)	=	0.278374549
P (vm_truncate_count | benign)	=	0.208538812
P (free_area_cache | benign)	=	0.413033648
P (mm_users | benign)	=	1.67871E-07
P (map_count | benign)	=	0.337897027
P (total_vm | benign)	=	0.542317962
P (shared_vm | benign)	=	0.002781847
P (exec_vm | benign)	=	0.002888022
P (reserved_vm | benign)	=	0.230127919
P (end_data | benign)	=	0.002781847
P (last_interval | benign)	=	0.032646005
P (Nvcsw | benign)	=	0.14826753
P (Nivcsw | benign)	=	0.46620227

P (min_flt | benign) = 1.288342197
P (maj_flt | benign) = 0.002781847
P (fs_excl_counter | benign) = 0.405560344
P (Utime | benign) = 0.015829298
P (Stime | benign) = 0.161700803
P (gtime | benign) = 0.614082051

Selanjutnya Penghitungan probabilitas Distribusi Gaussian pada kelas malware :

$$P(\text{millisecond} \mid \text{malware}) = \frac{1}{\sqrt{2*3.14*0.211738742}} e^{\frac{((-1.047886023)-(-0.600874589))^2}{2*0.211738742^2}} =$$

0.0183867930803045

Penghitungan lengkap untuk 21 atribut yang lain pada kelas malware, dengan hasil sebagai berikut :

P (state | malware) = 6.405073125
P (Prio | malware) = 0.360368963
P (static_prio | malware) = 0.119606766
P (vm_truncate_count | malware) = 0.034242379
P (free_area_cache | malware) = 0.730909137
P (mm_users | malware) = 0.009474386
P (map_count | malware) = 0.272019353
P (total_vm | malware) = 0.544180457
P (shared_vm | malware) = 0.206600127
P (exec_vm | malware) = 0.011002953
P (reserved_vm | malware) = 0.195148687
P (end_data | malware) = 0.206600127
P (last_interval | malware) = 0.025898936
P (Nvcsw | malware) = 0.008633254
P (Nivcsw | malware) = 0.483329112
P (min_flt | malware) = 1.65482949
P (maj_flt | malware) = 0.206600127
P (fs_excl_counter | malware) = 0.547309677
P (Utime | malware) = 0.003170774
P (Stime | malware) = 0.130053019
P (gtime | malware) = 0.419138087

4) Menghitung probabilitas posterior setiap kelas

Probabilitas posterior benign = P (millisecond | benign) * P (state | benign) * P (Prio | benign) * P (static_prio | benign) * P (vm_truncate_count | benign)

* P (free_area_cache | benign) * P (mm_users | benign) * P (map_count | benign) * P (total_vm | benign) * P (shared_vm | benign) * P (exec_vm | benign) * P (reserved_vm | benign) * P (end_data | benign) * P (last_interval | benign) * P (Nivcsw | benign) * P (min_flt | benign) * P (maj_flt | benign) * P (fs_excl_counter | benign) * P (Utime | benign) * P (Stime | benign) * P (gtime | benign) =

0.341958552778149 * 3.927099064 * 0.373382743 * 0.278374549 * 0.208538812 * 0.413033648 * 1.67871E-07 * 0.337897027 * 0.542317962 * 0.002781847 * 0.002888022 * 0.230127919 * 0.002781847 * 0.032646005 * 0.14826753 * 0.46620227 * 1.288342197 * 0.002781847 * 0.405560344 * 0.015829298 * 0.161700803 * 0.614082051 = **1.59620520213715E-27**

Probabilitas posterior malware = P (millisecond | malware) * P (state | malware) * P (Prio | malware) * P (static_prio | malware) * P (vm_truncate_count | malware) * P (free_area_cache | malware) * P (mm_users | malware) * P (map_count | malware) * P (total_vm | malware) * P (shared_vm | malware) * P (exec_vm | malware) * P (reserved_vm | malware) * P (end_data | malware) * P (last_interval | malware) * P (Nivcsw | malware) * P (min_flt | malware) * P (maj_flt | malware) * P (fs_excl_counter | malware) * P (Utime | malware) * P (Stime | malware) * P (gtime | malware) = 0.0183867930803045 * 6.405073125 * 0.360368963 * 0.119606766 * 0.034242379 *0.730909137 * 0.009474386 * 0.272019353 *0.544180457 * 0.206600127 * 0.011002953 * 0.195148687 * 0.206600127 * 0.025898936 * 0.008633254 * 0.483329112 * 1.65482949 * 0.206600127 * 0.547309677 * 0.003170774 * 0.130053019 * 0.419138087 = **1.95164667547195E-20**

5) Penentuan klasifikasi akhir dari data testing

Bandingkan hasil dari probabilitas “malware” dan “benign” yaitu

Probailitas “malware” = 1.95164667547195E-2

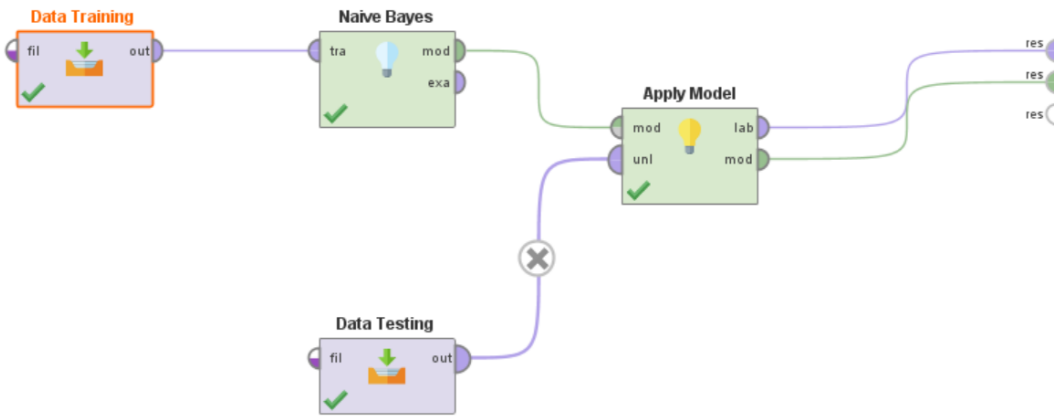
Probalitas “benign” = 1.59620520213715E-27

Dikarenakan 1.95164667547195E-2 > 1.59620520213715E-27, maka dapat disimpulkan bahwa data testing tersebut termasuk klasifikasi “malware”.

Sehingga rule yang dihasilkan sebagai berikut :

a) Jika probabilitas “malware” lebih besar dari probabilitas “benign” maka hasil adaalah “malware”
b) Jika probabilitas “benign” lebih besar dari probabilitas “malware” maka hasil adaalah “benign”

Implementasi pemodelan klasifikasi naïve bayes dengan data sampel menggunakan tools Rapidminer seperti terlihat pada gambar di bawah ini :



Gambar 4.6 Pemodelan klasifikasi naïve bayes (Rapidminer)

SimpleDistribution

```
Distribution model for label attribute classification


Class benign (0.480)
22 distributions

Class malware (0.520)
22 distributions
```

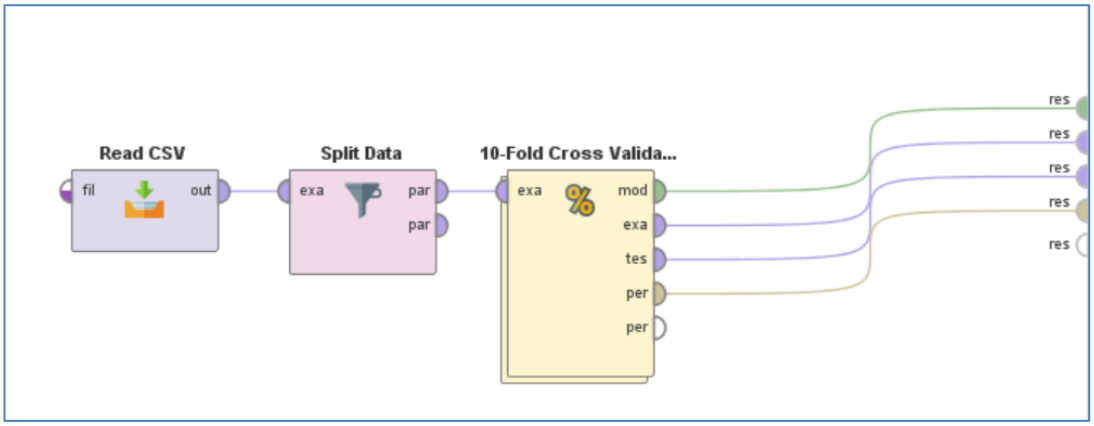
Gambar 4.7 Hasil Distribusi klasifikasi naïve bayes (Rapidminer)

4.4 Pengujian Model

Pengujian yang diusulkan dalam penelitian ini dimaksudkan untuk membandingkan metode mana yang lebih baik antara metode Decision Tree C4.5 kriteria gini index dengan Gaussian Naive Bayes. Metode analisis data dalam penelitian ini mengacu pada tahapan proses CRISDP-DM (Cross –Industry Standard Process for Data Mining). Data set yang telah ditentukan 80 % sebagai data training dan 20 % sebagai data testing selanjutnya akan divalidasi menggunakan 10-folds cross validation, di mana data secara acak akan dibagi menjadi 10 bagian. Pembagian 10 bagian merupakan metode yang paling tepat untuk mendapatkan estimasi terbaik menentukan kesalahan. Hasil dari validasi akan menghasilkan data yang diukur yaitu Accuracy. Evaluasi terhadap model mengukur akurasi dengan confussion matrix yang menitikberatkan pada class secara umum, sedangkan untuk AUC menggunakan ROC Curve dan proses dengan menggunakan 10 fold cross validation. Pengujian akan dilakukan dengan 5 kali percobaan untuk masing-masing algoritma C4.5 dan naïve bayes seperti tabel di bawah ini :

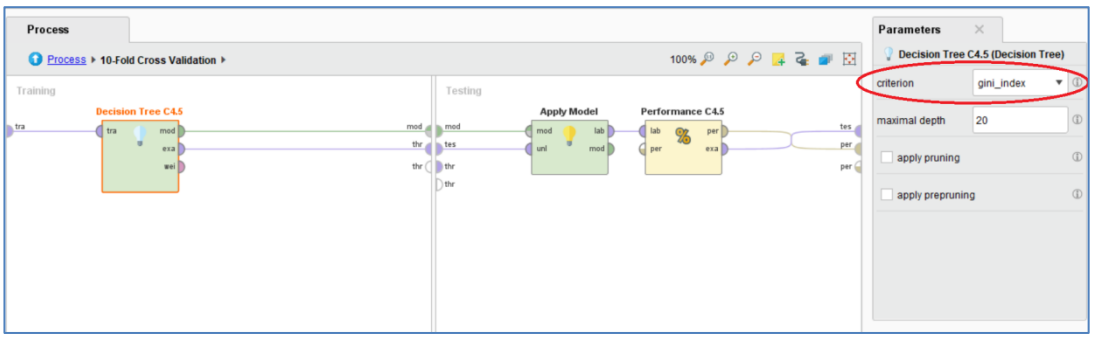
Tabel 4.13 Pengujian Model Algoritma C4.5 Gini Index dan Gaussian Naïve Bayes

Pengujian	Algoritma	
	C4.5	Naïve Bayes
1	Tanpa diskritisasi	Tanpa diskritisasi
2	Diskritisasi 3 Variabel Tanpa Parameter Batasan -1 dan 1	Diskritisasi 3 Variabel Tanpa Parameter Batasan -1 dan 1
3	Diskritisasi 3 Variabel dengan Parameter Batasan -1 dan 1	Diskritisasi 3 Variabel dengan Parameter Batasan -1 dan 1
4	Diskritisasi 5 Variabel Tanpa Parameter Batasan -1 dan 1	Diskritisasi 5 Variabel Tanpa Parameter Batasan -1 dan 1
5	Diskritisasi 5 Variabel Dengan Parameter Batasan -1 dan 1	Diskritisasi 5 Variabel Dengan Parameter Batasan -1 dan 1



Gambar 4.9 Struktur Proses Pengujian dengan 10-Fold Cross Validation

4.4.1 Pengujian Model Decision Tree C4.5 Kriteria Gini Index



Gambar 4.10 Struktur Decision Tree (C4.5) Kriteria Gini Index

a. Tanpa Diskritisasi

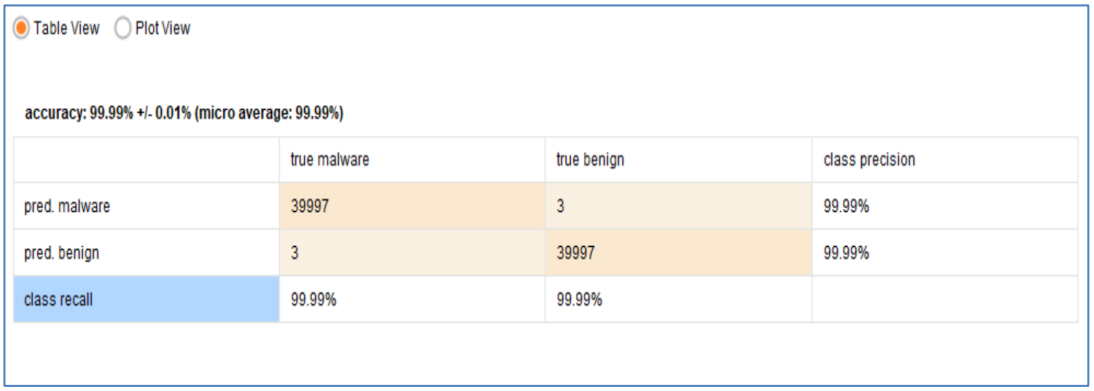


Table View Plot View

accuracy: 99.99% +/- 0.01% (micro average: 99.99%)

	true malware	true benign	class precision
pred. malware	39997	3	99.99%
pred. benign	3	39997	99.99%
class recall	99.99%	99.99%	

Gambar 4.11 Confusion Matrix C4.5 Tanpa Diskritisasi

Berdasarkan gambar 4.9 di atas menunjukan bahwa dari 80.000 data training, terdapat 39.997 malware yang sesuai dengan prediksi dan 3 malware yang tidak sesuai prediksi dengan nilai class precision sebesar 99.99%. Sedangkan dari non malware (benign) juga terdapat 3 benign yang tidak sesuai dengan prediksi dengan nilai precision yang sama yaitu 99.99%. Sehingga tingkat akurasi dengan menggunakan algoritma decision tree C4.5 tanpa diskritisasi adalah 99,99%. Dalam menentukan persentase akurasi dari data yang sudah diolah, maka formula yang digunakan adalah sebagai berikut:

Tabel 4.14 Confusion Matrix C4.5 Tanpa Diskritisasi

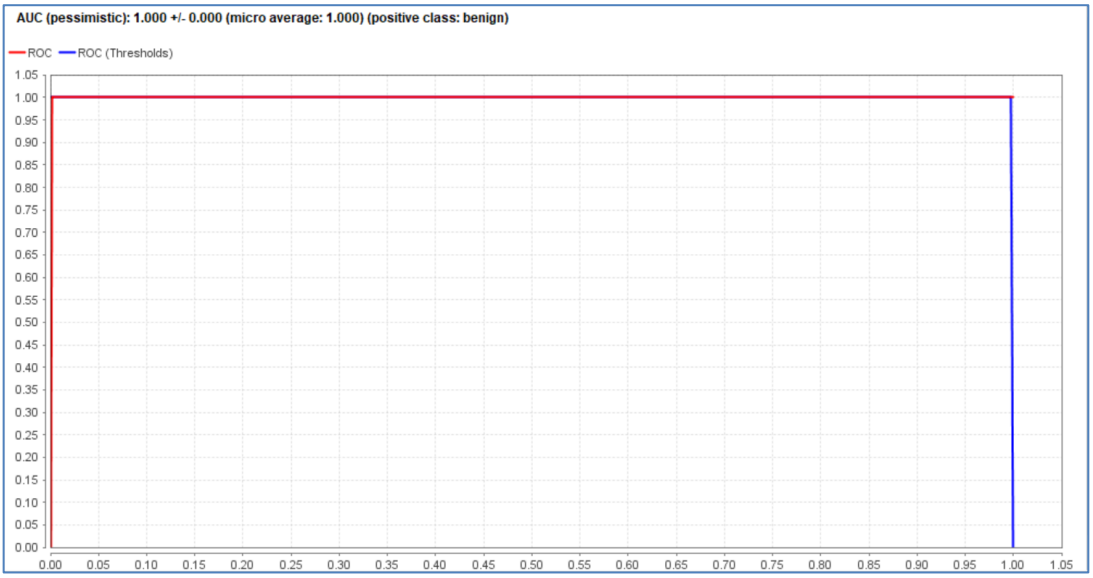
NILAI PREDIKSI	NILAI SEBENARNYA		
	Class	True (Malware)	False (Benign)
	Malware	39.997 (TP)	3 (FP)
	Benign	3 (FN)	39.997 (TN)

Precision $= \frac{TN}{TN+FN} = \frac{39.997}{39.997+3} = 0.9999 = 99.99\%$

Recall $= \frac{TN}{TN+FP} = \frac{39.997}{39.997+3} = 0.9999 = 99.99\%$

Accuracy $= \frac{TP+TN}{TP+TN+FP+FN} = \frac{39.997+39.997}{39.997+39.997+3+3} = 0.9999 = 99.99\%$

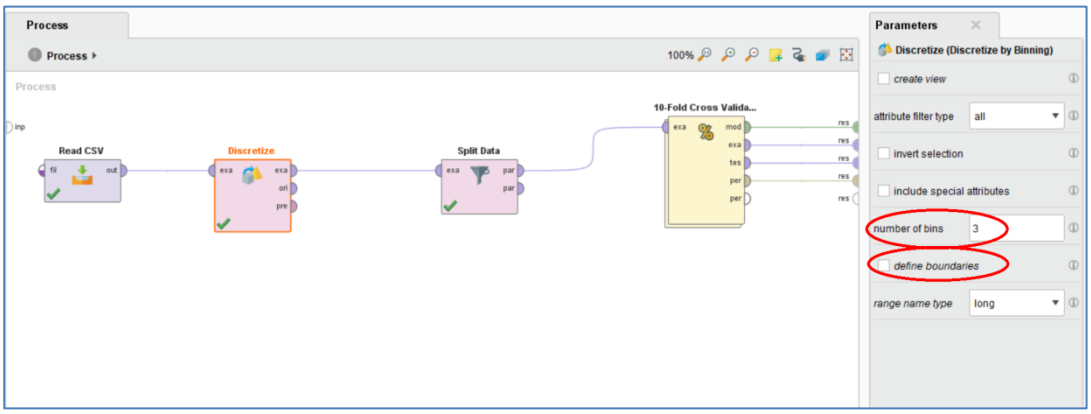
Setelah dilakukan seluruh tahapan evaluasi untuk confusion matrix maka selanjutnya dilakukan analisa evaluasi pembanding yakni terhadap pengukuran *Receiver Operating Characteristic (ROC)* sebagai berikut dapat dilihat pada gambar 4.12 dibawah ini :



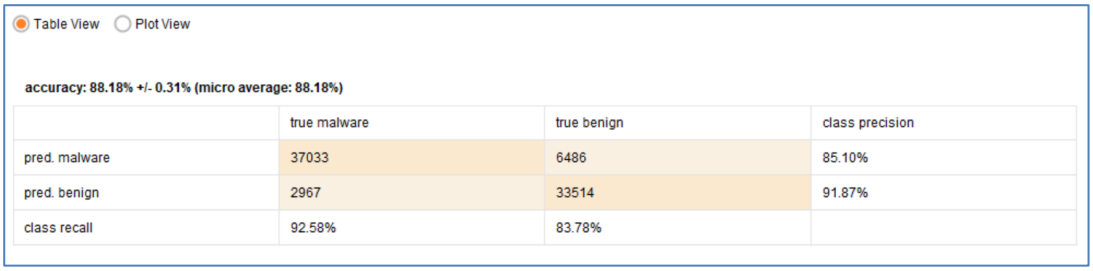
Gambar 4.12 Pengukuran *AUC Pesimistic* C4.5 Tanpa Diskritisasi

Nilai yang diperoleh dari AUC pessimistic adalah 1.00 untuk class predictive non malware(benign). Secara keseluruhan model yang dihasilkan dengan metode C4.5. tanpa diskritisasi terlihat pada nilai accuracy, peceision dan recall. Untuk klasifikasi nilai AUC diperoleh dari pengolahan ROC seperti terlihat pada gambar 4.12 (Pengukuran UAC Optimistic) dengan tingkat diagnosa klasifikasi sangat baik.

b. Dengan Diskritisasi 3 Variabel (no define boundaries)



Gambar 4.13 Proses C4.5 Diskritisasi 3 bin *no define boundaries*



Gambar 4.14 Confusion Matrix C4.5 Diskritisasi 3 Bin No define Boundaries

Berdasarkan gambar 4.12 di atas menunjukan bahwa dari 80.000 data training, terdapat 37.033 malware yang sesuai dengan prediksi dan 6.486 malware yang tidak sesuai prediksi dengan nilai class precision sebesar 85.10%. Sedangkan dari non malware (benign) terdapat 2.967 benign yang tidak sesuai dengan prediksi dengan nilai precision yang sama yaitu 91.87%. Sehingga tingkat akurasi dengan menggunakan algoritma decision tree C4.5 dengan diskritisasi 3 variabel tanpa batasan min max (-1,1) adalah 88,18%. Dalam menentukan persentase akurasi dari data yang sudah diolah, maka formula yang digunakan adalah sebagai berikut:

Tabel 4.15 Confusion Matrix C4.5 Diskritisasi 3 Bin No define Boundaries

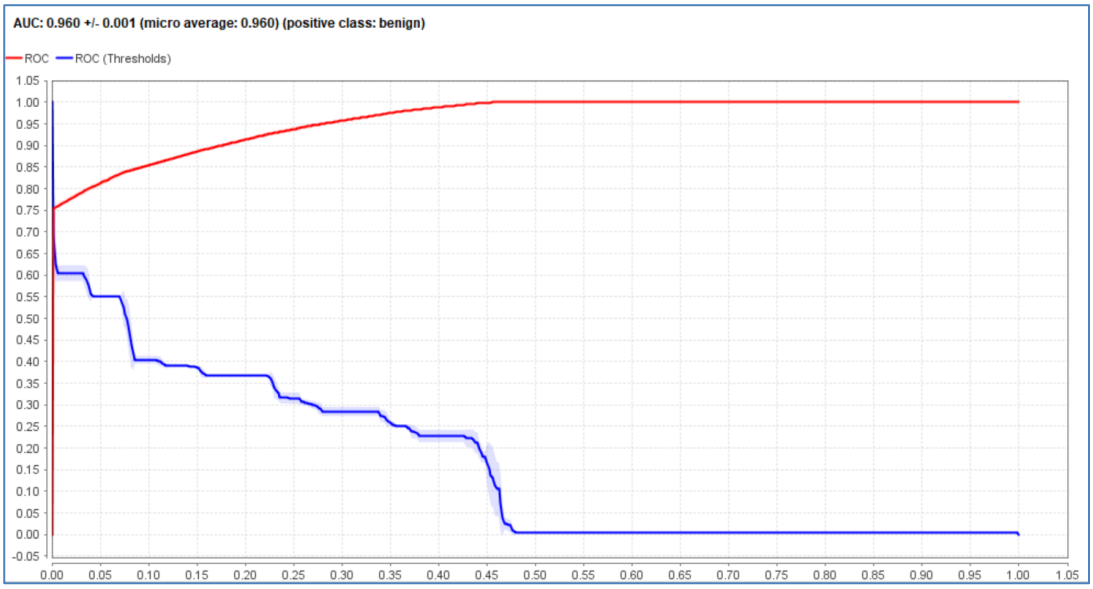
NILAI PREDIKSI	**NILAI SEBENARNYA**		
	Class	True (Malware)	False (Benign)
	Malware	37.033 (TP)	6.486 (FP)
	Benign	2.967 (FN)	33.514 (TN)

$$\text{Precision} = \frac{TN}{TN+FN} = \frac{33.514}{33.514 + 2.967} = 0.9187 = 91.87\%$$

$$\text{Recall} = \frac{TN}{TN+FP} = \frac{33.514}{35.514 + 6.486} = 0.8378 = 83.78\%$$

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} = \frac{37.033+33.514}{37.033 +33.514+6.486+2.967} = 0.8818 = 88.18\%$$

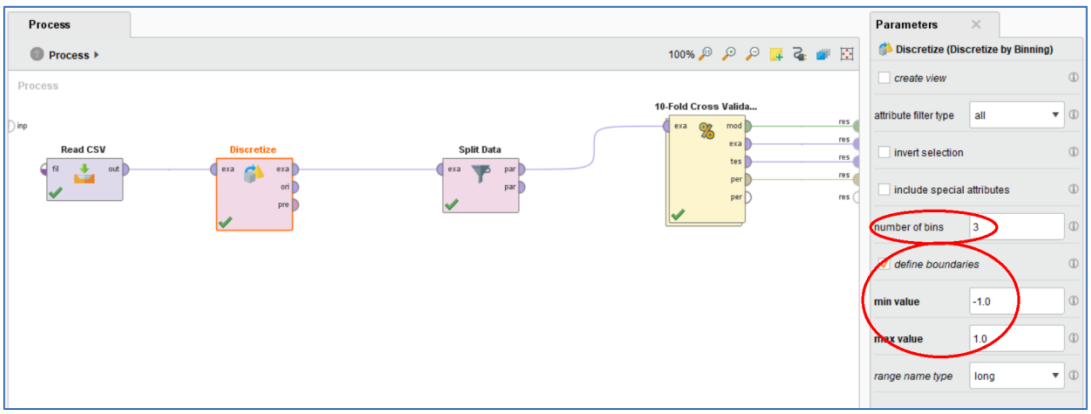
Setelah dilakukan seluruh tahapan evaluasi untuk confusion matrix maka selanjutnya dilakukan analisa evaluasi pembanding yakni terhadap pengukuran *Receiver Operating Characteristic (ROC)* sebagai berikut dapat dilihat pada gambar 4.13 dibawah ini :



Gambar 4.15 Pengukuran *AUC* C4.5 Diskritisasi 3 Bin No define Boundaries

Nilai yang diperoleh dari AUC adalah 0.960 untuk class predictive non malware(benign). Secara keseluruhan model yang dihasilkan dengan metode C4.5 dengan diskritisasi 3 variabel tanpa batasan min max(-1,1) terlihat pada nilai accuracy, precision dan recall. Untuk klasifikasi nilai AUC tersebut diperoleh dari pengolahan ROC seperti terlihat pada gambar 4.15 (Pengukuran UAC) dengan tingkat diagnosa klasifikasi sangat baik.

c. Dengan Diskritisasi 3 Variabel (define boundaries)



Gambar 4.16 Proses C4.5 Diskritisasi 3 bin *define boundaries*

Pada gambar di atas proses diskritisasi dengan 3 variabel dengan menetapkan batasan nilai minimal = -1 dan maksimal=1 menghasilkan confusion matrix seperti di bawah ini

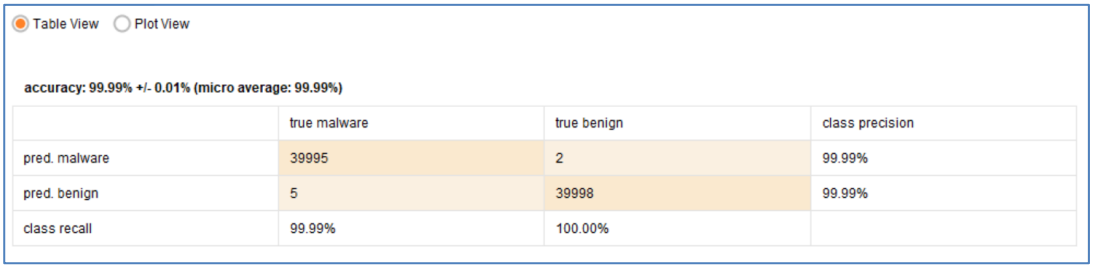


Table View Plot View

accuracy: 99.99% +/- 0.01% (micro average: 99.99%)

	true malware	true benign	class precision
pred. malware	39995	2	99.99%
pred. benign	5	39998	99.99%
class recall	99.99%	100.00%	

Gambar 4.17 Confusion Matrix C4.5 Diskritisasi 3 Bin define Boundaries

Berdasarkan gambar 4.18 di atas menunjukan bahwa dari 80.000 data training, terdapat 39.995 malware yang sesuai dengan prediksi dan 2 malware yang tidak sesuai prediksi dengan nilai class precision sebesar 99.99%. Sedangkan dari non malware (benign) terdapat 5 benign yang tidak sesuai dengan prediksi dengan nilai precision yang sama yaitu 99.99%. Sehingga tingkat akurasi dengan menggunakan algoritma decision tree C4.5 diskritisasi 3 variabel dengan batasan min max(-1,1) adalah 99,99%. Dalam menentukan persentase akurasi dari data yang sudah diolah, maka formula yang digunakan adalah sebagai berikut:

Tabel 4.16 Confusion Matrix C4.5 Diskritisasi 3 Bin define Boundaries

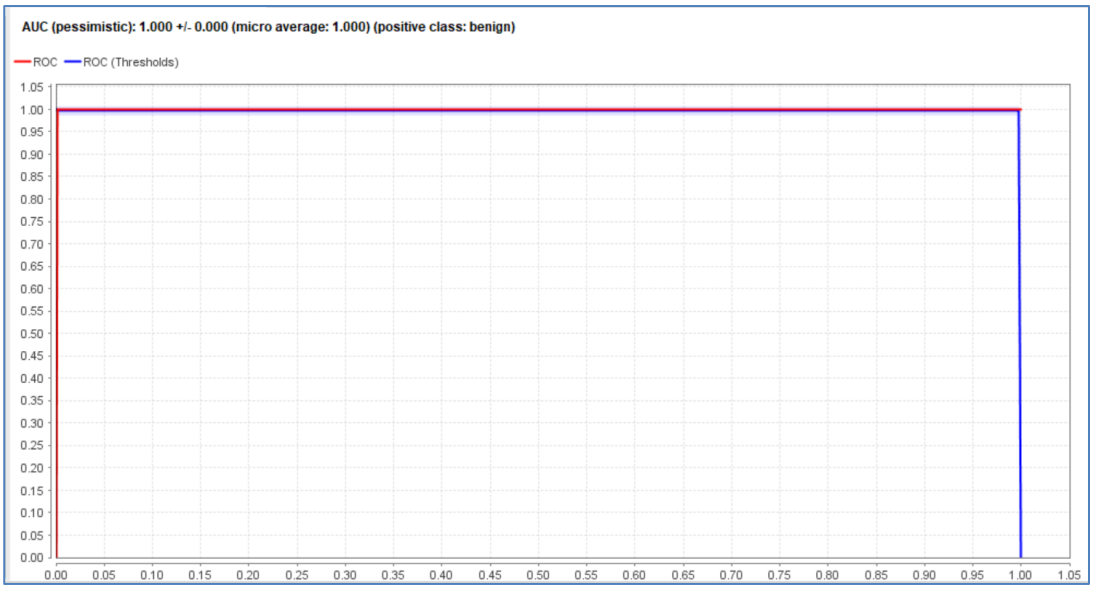
NILAI PREDIKSI	**NILAI SEBENARNYA**		
	Class	True (Malware)	False (Benign)
	Malware	39.995 (TP)	2 (FP)
	Benign	5 (FN)	39.998 (TN)

$$\text{Precision} = \frac{TN}{TN+FN} = \frac{39.998}{39.998+5} = 0.9999 = 99.99\%$$

$$\text{Recall} = \frac{TN}{TN+FP} = \frac{39.998}{39.998+2} = 1.0000 = 100.00\%$$

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} = \frac{39.995+39.998}{39.995+39.998+2+5} = 0.9999 = 99.99\%$$

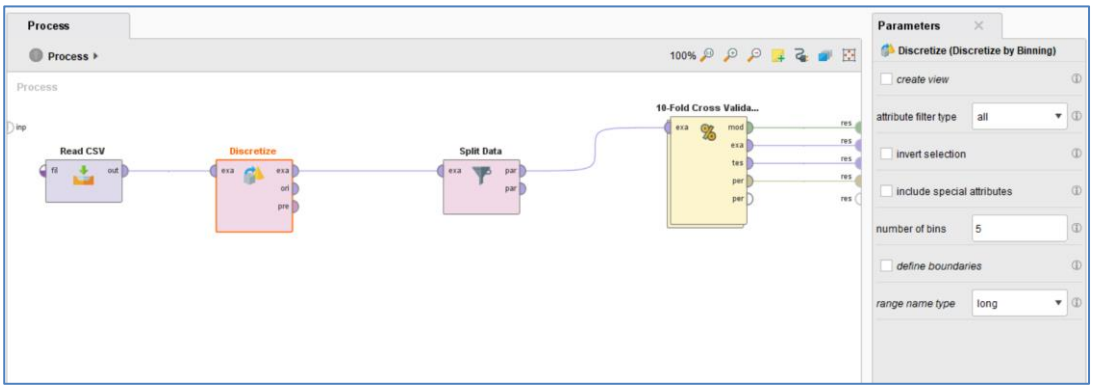
Setelah dilakukan seluruh tahapan evaluasi untuk confusion matrix maka selanjutnya dilakukan analisa evaluasi pembanding yakni terhadap pengukuran *Receiver Operating Characteristic (ROC)* sebagai berikut dapat dilihat pada gambar ……



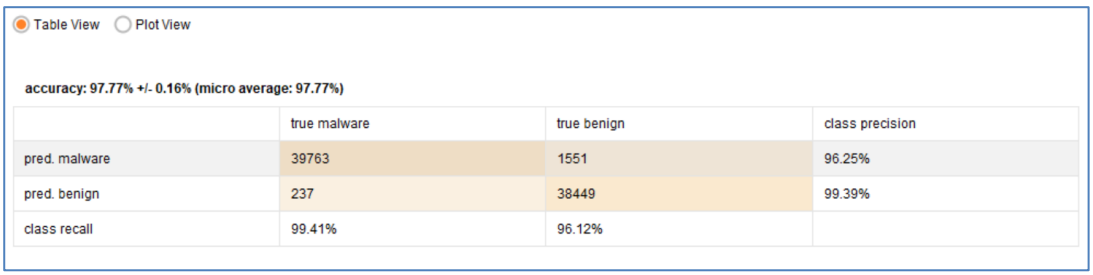
Gambar 4.18 Pengukuran *AUC Pessimistic* C4.5 Diskritisasi 3 Bin define Boundaries

Nilai yang diperoleh dari AUC adalah 1.000 untuk class predictive non malware(benign). Secara keseluruhan model yang dihasilkan dengan metode C4.5 diskritisasi 3 variabel dengan batasan min max(-1,1) terlihat pada nilai accuracy, precision dan recall. Untuk klasifikasi nilai AUC tersebut diperoleh dari pengolahan ROC seperti terlihat pada gambar 4.18 (Pengukuran UAC Pessimistic) dengan tingkat diagnosa klasifikasi sangat baik.

d. Dengan Diskritisasi 5 Variabel (no define boundaries)



Gambar 4.19 Proses C4.5 Diskritisasi 5 bin *no define boundaries*



	true malware	true benign	class precision
pred. malware	39763	1551	96.25%
pred. benign	237	38449	99.39%
class recall	99.41%	96.12%	

Gambar 4.20 Confusion Matrix C4.5 Diskritisasi 5 Bin No define Boundaries

Berdasarkan gambar 4.18 di atas menunjukan bahwa dari 80.000 data training, terdapat 39.763 malware yang sesuai dengan prediksi dan 1.551 malware yang tidak sesuai prediksi dengan nilai class precision sebesar 96.25%. Sedangkan dari non malware (benign) terdapat 237 benign yang tidak sesuai dengan prediksi dengan nilai precision yang sama yaitu 99.39%. Sehingga tingkat akurasi dengan menggunakan algoritma decision tree C4.5 diskritisasi 5 variabel tanpa batasan min max(-1,1) adalah 97,77%. Dalam menentukan persentase akurasi dari data yang sudah diolah, maka formula yang digunakan adalah sebagai berikut:

Tabel 4.17 Confusion Matrix C4.5 Diskritisasi
5 Bin No define Boundaries

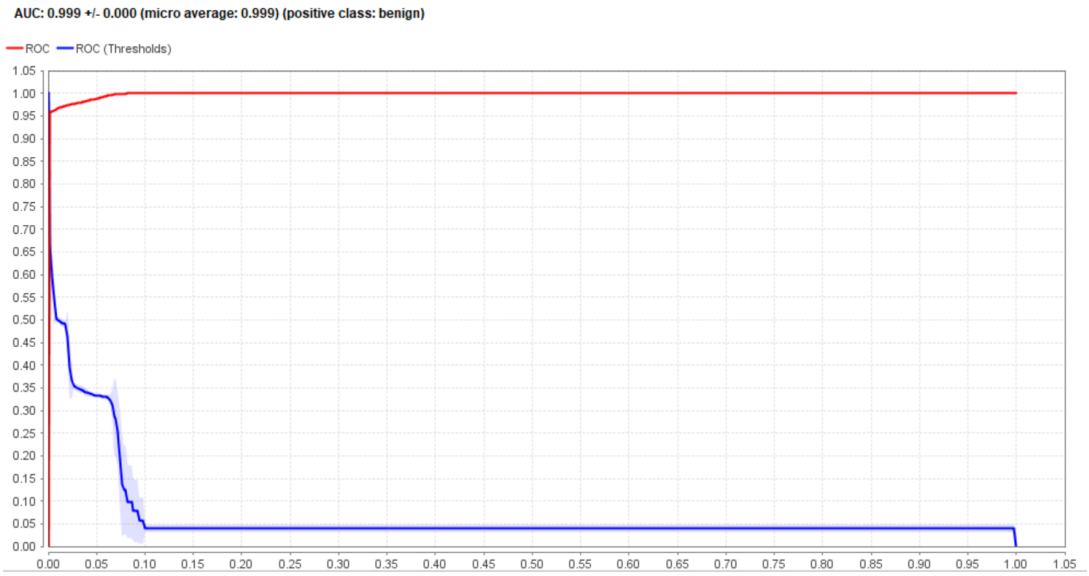
NILAI PREDIKSI	**NILAI SEBENARNYA**		
	Class	True (Malware)	False (Benign)
	Malware	39.763 (TP)	1.551 (FP)
	Benign	237 (FN)	38.449 (TN)

Precision Benign = $\frac{TN}{TN+FN} = \frac{38.449}{38.449 + 237}$ = 0.9939 = 99.39%

Recall Benign = $\frac{TN}{TN+FP} = \frac{38.449}{38.449 + 1.551}$ = 0.9612 = 96.12%

Accuracy = $\frac{TP+TN}{TP+TN+FP+FN} = \frac{39.763+38.449}{39.449 +38.449+1.551+237}$ = 0.9777 = 97.77%

Setelah dilakukan seluruh tahapan evaluasi untuk confusion matrix maka selanjutnya dilakukan analisa evaluasi pembanding yakni terhadap pengukuran *Receiver Operating Characteristic (ROC)* sebagai berikut dapat dilihat pada gambar 4.21 di bawah ini :

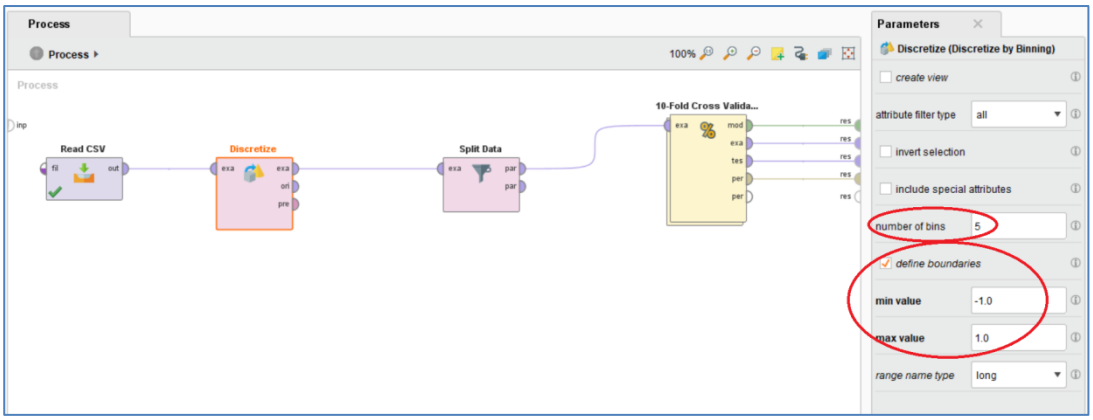


Gambar 4.21 Pengukuran *AUC* C4.5 Diskritisasi 5 Bin No define Boundaries

Nilai yang diperoleh dari AUC adalah 0.999 untuk class predictive non malware(benign). Secara keseluruhan model yang dihasilkan dengan metode C4.5 diskritisasi 5 variabel tanpa batasan min max(-1,1) terlihat pada nilai accuracy, precision dan recall. Untuk klasifikasi nilai AUC tersebut diperoleh dari

pengolahan ROC seperti terlihat pada gambar 4.21 (Pengukuran UAC) dengan tingkat diagnosa klasifikasi sangat baik.

e. Dengan Diskritisasi 5 Variabel (define boundaries)



Gambar 4.22 Proses C4.5 Diskritisasi 5 bin *define boundaries*

Pada gambar di atas proses diskritisasi dengan 5 variabel dengan menetapkan batasan nilai minimal = -1 dan maksimal=1 menghasilkan confusion matrix seperti di bawah ini

Table View Plot View

accuracy: 100.00% +/- 0.00% (micro average: 100.00%)

	true malware	true benign	class precision
pred. malware	40000	0	100.00%
pred. benign	0	40000	100.00%
class recall	100.00%	100.00%	

Gambar 4.23 Confusion Matrix C4.5 Diskritisasi 5 Bin define Boundaries

Berdasarkan gambar 4.21 di atas menunjukan bahwa dari 80.000 data training, terdapat 40.000 seluruh malware yang sesuai dengan prediksi sehingga nilai class precision sebesar 100%. Begitu juga dari non malware (benign) seluruhnya sesuai dengan prediksi dengan nilai precision yang sama yaitu 100%. Sehingga tingkat akurasi dengan menggunakan algoritma decision tree C4.5 diskritisasi 5 variabel dengan batasan min max (-1,1) adalah 100%. Dalam menentukan persentase akurasi dari data yang sudah diolah, maka formula yang digunakan adalah sebagai berikut:

Tabel 4.18 Confusion Matrix C4.5 Diskritisasi
5 Bin define Boundaries

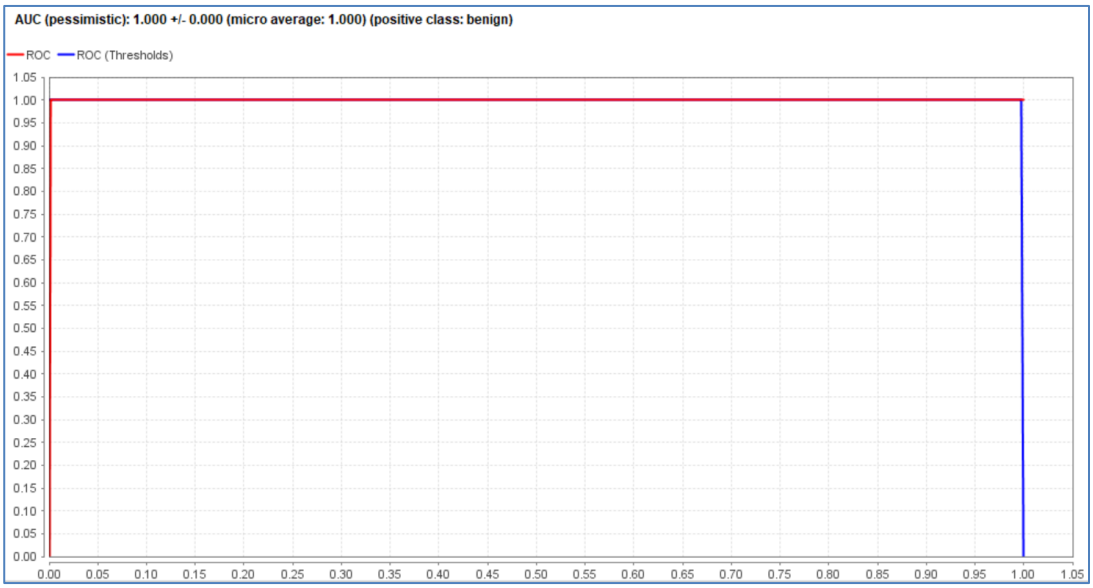
NILAI PREDIKSI	**NILAI SEBENARNYA**		
	Class	True (Malware)	False (Benign)
	Malware	40.000 (TP)	0 (FP)
	Benign	0 (FN)	40.000 (TN)

Precision Benign = $\frac{TN}{TN+FN} = \frac{40.000}{40.000+0}$ = 1.0000 = 100%

Recall Benign = $\frac{TN}{TN+FP} = \frac{40.000}{40.000+0}$ = 1.0000 = 100%

Accuracy = $\frac{TP+TN}{TP+TN+FP+FN} = \frac{40.000+40.000}{40.000+40.000+0+0}$ = 1.0000 = 100%

Setelah dilakukan seluruh tahapan evaluasi untuk confusion matrix maka selanjutnya dilakukan analisa evaluasi pembanding yakni terhadap pengukuran *Receiver Operating Characteristic (ROC)* sebagai berikut dapat dilihat pada gambar 4.24 seperti di bawah ini :



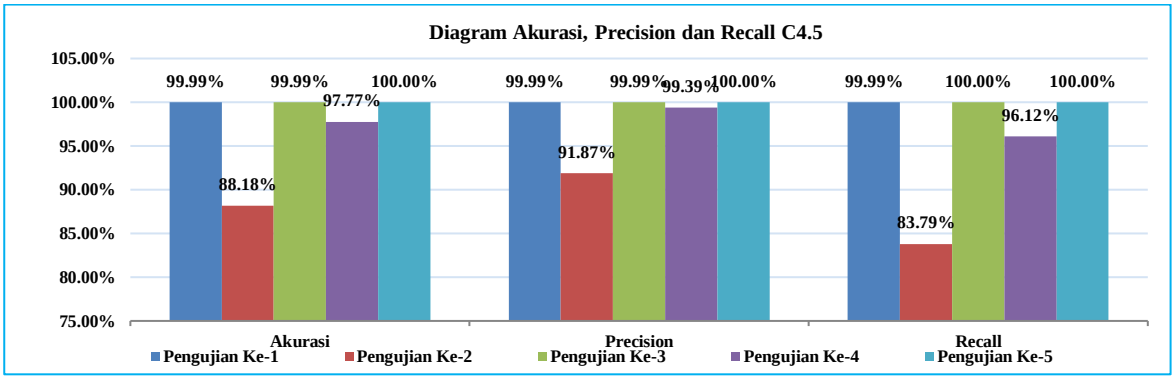
Gambar 4.24 Pengukuran *AUC Pessimistic* C4.5 Diskritisasi 5 Bin define Boundaries

Nilai yang diperoleh dari AUC adalah 1.000 untuk class predictive non malware(benign). Secara keseluruhan model yang dihasilkan dengan metode C4.5 diskritisasi 5 variabel dengan batasan min max(-1,1) terlihat pada nilai accuracy, precision dan recall. Untuk klasifikasi nilai AUC tersebut diperoleh dari pengolahan ROC seperti terlihat pada gambar 4.23 (Pengukuran UAC Pessimistic) dengan tingkat diagnosa klasifikasi sangat baik.

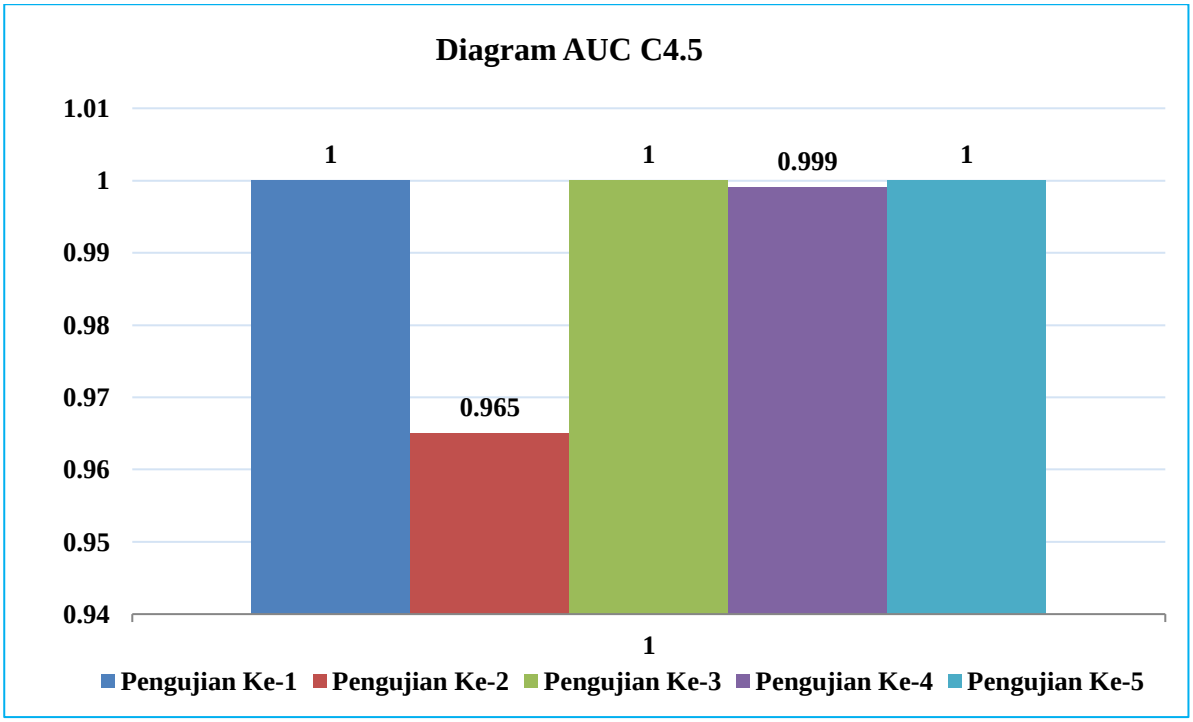
Hasil lengkap ke-5 pengujian algoritma C4.5 di atas (pengujian sesuai Tabel 4.13) di tampilkan ke dalam tabel dan diagram seperti di bawah ini :

Tabel 4.19 Hasil Pengujian Algoritma C4.5 Kriteria Gini Index

Hasil	Akurasi	Precision	Recall	AUC
Pengujian Ke-1	99.99%	99.99%	99.99%	1
Pengujian Ke-2	88.18%	91.87%	83.79%	0.97
Pengujian Ke-3	99.99%	99.99%	100.00%	1
Pengujian Ke-4	97.77%	99.39%	96.12%	1
Pengujian Ke-5	100.00%	100.00%	100.00%	1

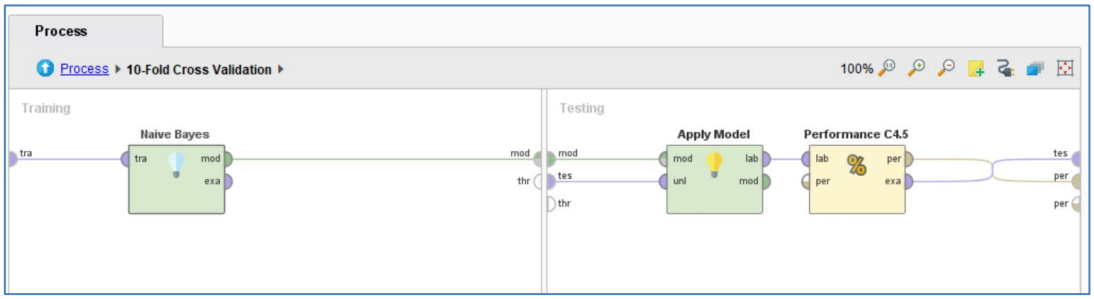


Gambar 4.25 Diagram Akurasi, Precision dan Recall C4.5 Gini Index



Gambar 4.26 Diagram AUC C4.5 Gini Index

4.4.2 Pengujian Model Naïve Bayes



Gambar 4.27 Struktur Naïve Bayes

a. Tanpa Diskritisasi

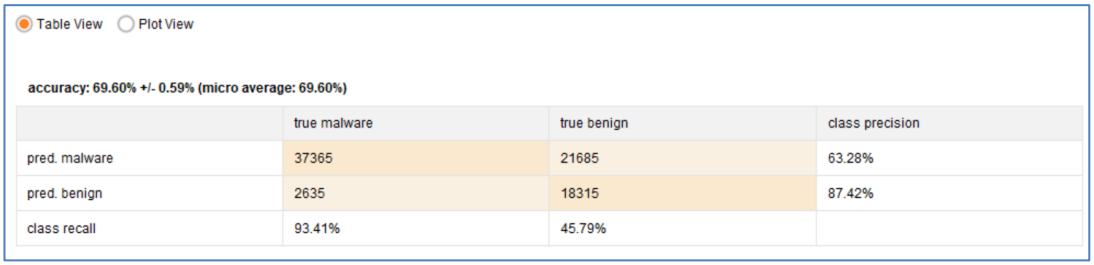


Table View Plot View

accuracy: 69.60% +/- 0.59% (micro average: 69.60%)

	true malware	true benign	class precision
pred. malware	37365	21685	63.28%
pred. benign	2635	18315	87.42%
class recall	93.41%	45.79%	

Gambar 4.28 Confusion Matrix Naïve Bayes Tanpa Diskritisasi

Berdasarkan gambar 4.25 di atas menunjukan bahwa dari 80.000 data training, terdapat 37.365 malware yang sesuai dengan prediksi dan 21.685 malware yang tidak sesuai prediksi dengan nilai class precision sebesar 63.28%. Sedangkan dari non malware (benign) juga terdapat 2.635 benign yang tidak sesuai dengan prediksi dengan nilai precision yang sama yaitu 87.42%. Sehingga tingkat akurasi dengan menggunakan algoritma Naïve Bayes tanpa diskritisasi adalah 69,60%. Dalam menentukan persentase akurasi dari data yang sudah diolah, maka formula yang digunakan adalah sebagai berikut:

Tabel 4.20 Confusion Matrix Naïve Bayes Tanpa Diskritisasi

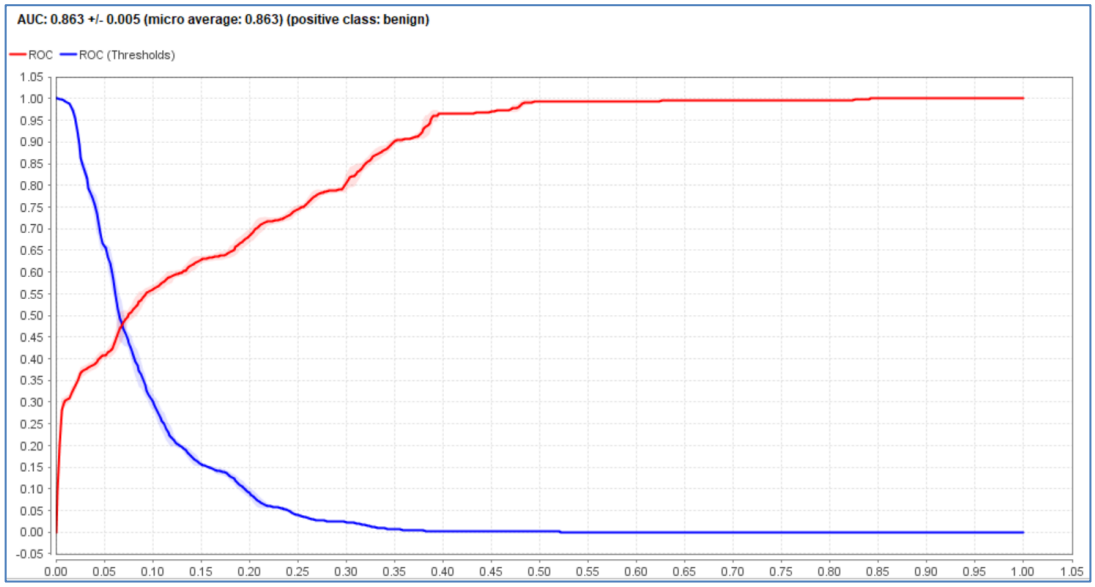
NILAI PREDIKSI	NILAI SEBENARNYA		
	Class	True (Malware)	False (Benign)
	Malware	37.365 (TP)	21.685 (FP)
	Benign	2.635 (FN)	18.315 (TN)

Precision = $\frac{TN}{TN+FN}$ = $\frac{18.315}{18.315 + 2.635}$ = 0.8742 = 87.42%

Recall Benign = $\frac{TN}{TN+FP}$ = $\frac{18.315}{18.315 +21.685}$ = 0.4579 = 45.79%

Accuracy = $\frac{TP+TN}{TP+TN+FP+ FN}$ = $\frac{37.365+18.315}{37.365 + 18.315+21.685+2.635}$ = 0.6960 = 69.60%

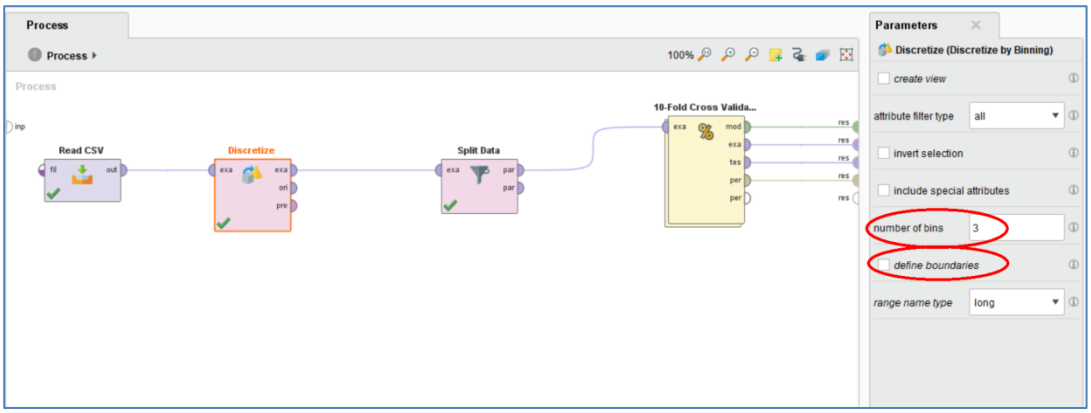
Setelah dilakukan seluruh tahapan evaluasi untuk confusion matrix maka selanjutnya dilakukan analisa evaluasi pembanding yakni terhadap pengukuran *Receiver Operating Characteristic (ROC)* sebagai berikut dapat dilihat pada gambar 4.29 seperti di bawah ini :



Gambar 4.29 Pengukuran *AUC* NB Tanpa Diskritisasi

Nilai yang diperoleh dari AUC pessimistic adalah 0.863 untuk class predictive non malware(benign). Secara keseluruhan model yang dihasilkan dengan metode naïve bayes tanpa diskritisasi terlihat pada nilai accuracy, precision dan recall. Untuk klasifikasi nilai AUC diperoleh dari pengolahan ROC seperti terlihat pada gambar 4.29 (Pengukuran UAC) dengan tingkat diagnosa klasifikasi baik.

b. Dengan Diskritisasi 3 Variabel (no define boundaries)



Gambar 4.30 Proses NB Diskritisasi 3 bin *no define boundaries*

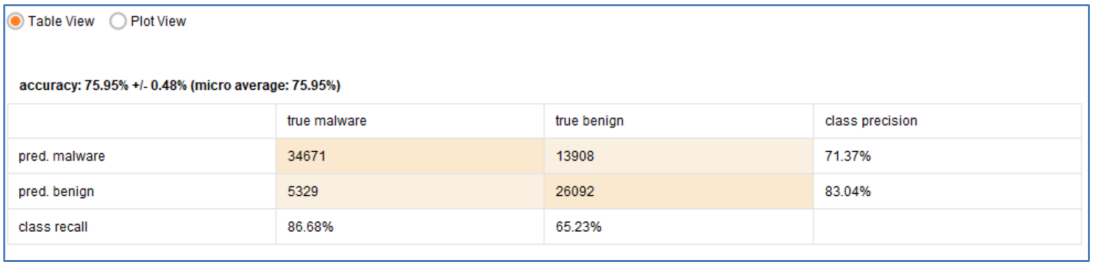


Table View Plot View

accuracy: 75.95% +/- 0.48% (micro average: 75.95%)

	true malware	true benign	class precision
pred. malware	34671	13908	71.37%
pred. benign	5329	26092	83.04%
class recall	86.68%	65.23%	

Gambar 4.31 Confusion Matrix NB Diskritisasi 3 Bin No define Boundaries

Berdasarkan gambar 4.28 di atas menunjukan bahwa dari 80.000 data training, terdapat 34.671 malware yang sesuai dengan prediksi dan 13.908 malware yang tidak sesuai prediksi dengan nilai class precision sebesar 71.37%. Sedangkan dari non malware (benign) terdapat 5.329 benign yang tidak sesuai dengan prediksi dengan nilai precision yang sama yaitu 83.04%. Sehingga tingkat akurasi dengan menggunakan algoritma Naïve Bayes diskritisasi 3 variabel tanpa batasan min max(-1,1) dengan adalah 75,95%. Dalam menentukan persentase akurasi dari data yang sudah diolah, maka formula yang digunakan adalah sebagai berikut:

Tabel 4.21 Confusion Matrix NB Diskritisasi
3 Bin No define Boundaries

NILAI PREDIKSI	**NILAI SEBENARNYA**		
	Class	True (Malware)	False (Benign)
	Malware	34.671 (TP)	13.908 (FP)
	Benign	5.329 (FN)	26.092 (TN)

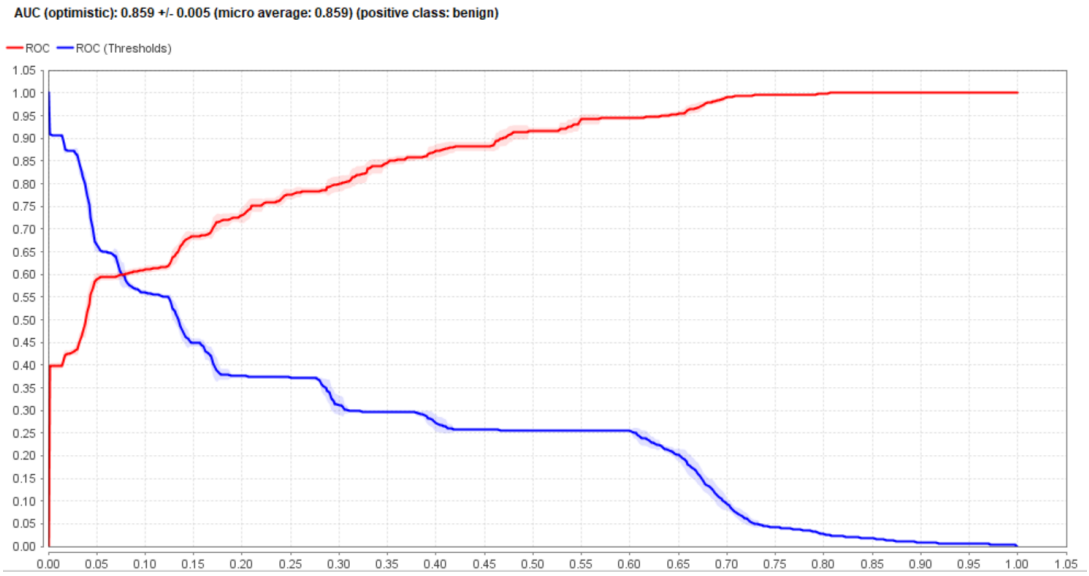
$$\text{Precision} = \frac{TN}{TN+FN} = \frac{26.092}{26.092 + 5.329} = 0.8304 = 83.04\%$$

$$\text{Recall} = \frac{TN}{TN+FP} = \frac{26.092}{26.092 + 13.908} = 0.6523 = 65.23\%$$

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} = \frac{34.671+26.092}{34.671+26.092+13.908+5.329} = 0.7595 = 75.95\%$$

Setelah dilakukan seluruh tahapan evaluasi untuk confusion matrix maka selanjutnya dilakukan analisa evaluasi pembanding yakni terhadap pengukuran

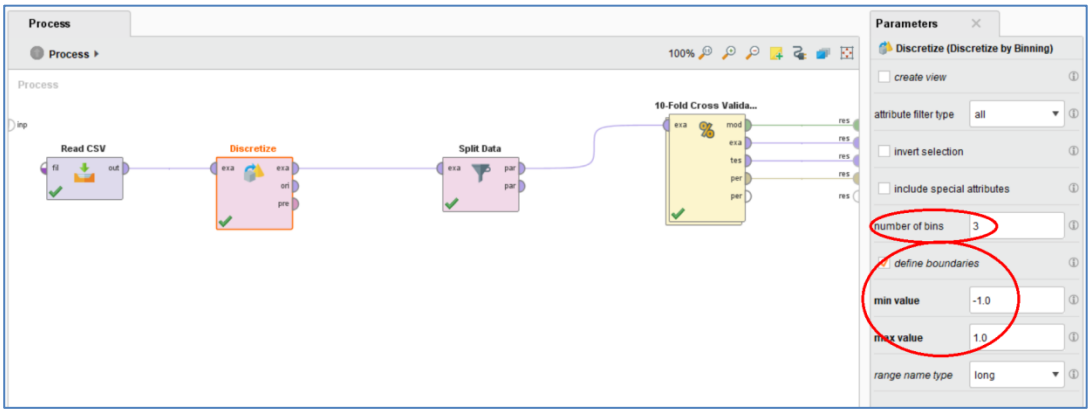
Receiver Operating Characteristic (ROC) sebagai berikut dapat dilihat pada gambar 4.32 seperti di bawah ini :



Gambar 4.32 Pengukuran *AUC optimistic* NB Diskritisasi 3 Bin No define Boundaries

Nilai yang diperoleh dari AUC optimistic adalah 0.859 untuk class predictive non malware(benign). Secara keseluruhan model yang dihasilkan dengan metode naïve bayes diskritisasi 3 variabel tanpa batasan min max(-1,1) terlihat pada nilai accuracy, precision dan recall. Untuk klasifikasi nilai AUC tersebut diperoleh dari pengolahan ROC seperti terlihat pada gambar 4.29 (Pengukuran UAC optimistic) dengan tingkat diagnosa klasifikasi baik.

c. Dengan Diskritisasi 3 Variabel (define boundaries)



Gambar 4.33 Proses NB Diskritisasi 3 bin *define boundaries*

Pada gambar di atas proses diskritisasi dengan 3 variabel dengan menetapkan batasan nilai minimal = -1 dan maksimal=1 menghasilkan confusion matrix seperti di bawah ini

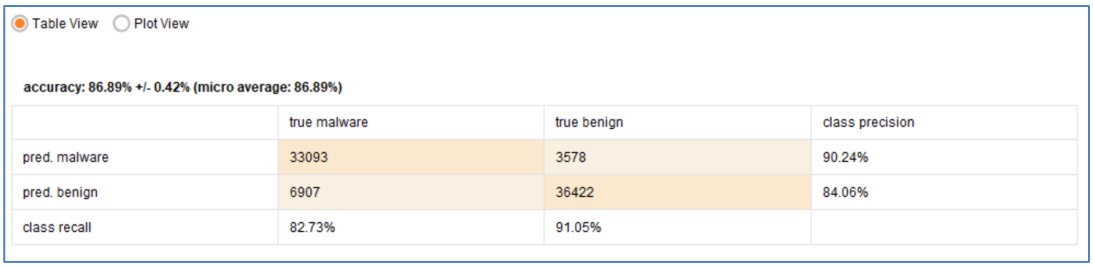


Table View Plot View

accuracy: 86.89% +/- 0.42% (micro average: 86.89%)

	true malware	true benign	class precision
pred. malware	33093	3578	90.24%
pred. benign	6907	36422	84.06%
class recall	82.73%	91.05%	

Gambar 4.34 Confusion Matrix NB Diskritisasi 3 Bin define Boundaries

Berdasarkan gambar 4.34 di atas menunjukan bahwa dari 80.000 data training, terdapat 33.093 malware yang sesuai dengan prediksi dan 3.578 malware yang tidak sesuai prediksi dengan nilai class precision sebesar 90.24%. Sedangkan dari non malware (benign) terdapat 6.907 benign yang tidak sesuai dengan prediksi dengan nilai precision yang sama yaitu 84.06%. Sehingga tingkat akurasi dengan menggunakan algoritma naïve bayes diskritisasi 3 variabel dengan batasa min max (-1,1) adalah 86,89%. Dalam menentukan persentase akurasi dari data yang sudah diolah, maka formula yang digunakan adalah sebagai berikut:

Tabel 4.22 Confusion Matrix NB Diskritisasi
3 Bin define Boundaries

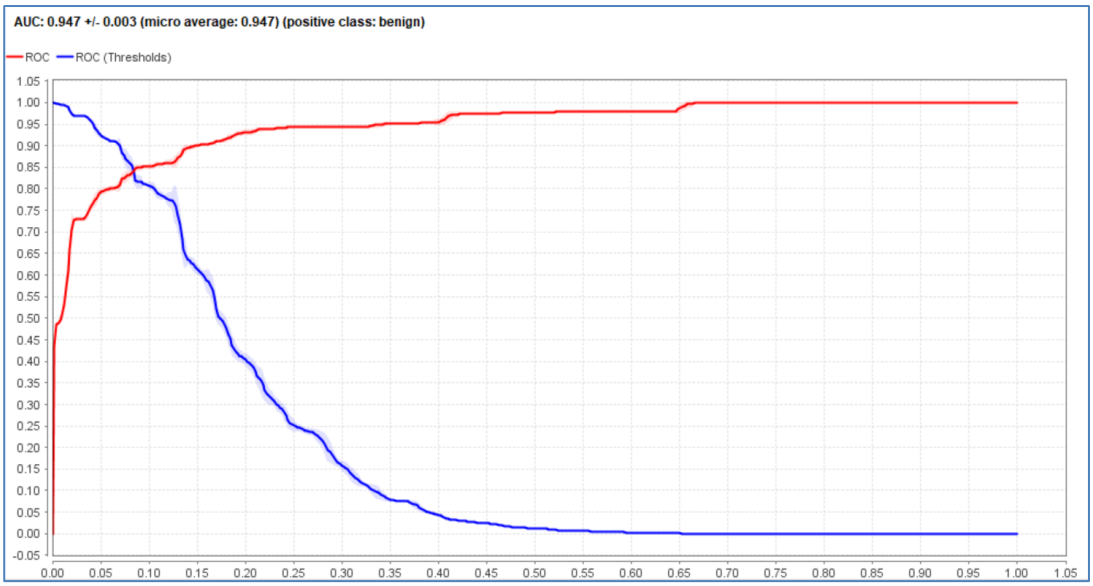
NILAI PREDIKSI	**NILAI SEBENARNYA**		
	Class	True (Malware)	False (Benign)
	Malware	33.093 (TP)	3.578 (FP)
	Benign	6.907 (FN)	36.422 (TN)

$$\text{Precision} = \frac{TN}{TN+FN} = \frac{36.422}{36.422 + 6.907} = 0.8406 = 84.06\%$$

$$\text{Recall} = \frac{TN}{TN+FP} = \frac{36.422}{36.422 + 3.578} = 0.9105 = 91.05\%$$

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} = \frac{33.093+36.422}{33.093 +36.422+3.578+6.907} = 0.8689 = 86.89\%$$

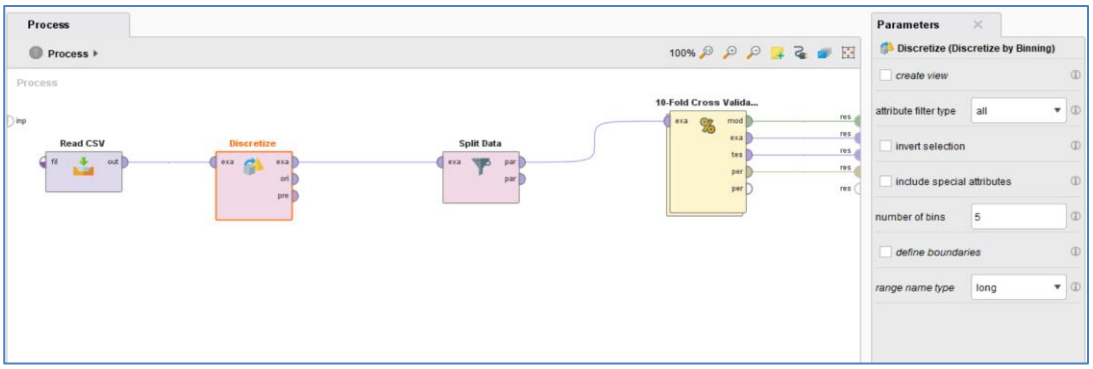
Setelah dilakukan seluruh tahapan evaluasi untuk confusion matrix maka selanjutnya dilakukan analisa evaluasi pembanding yakni terhadap pengukuran *Receiver Operating Characteristic (ROC)* sebagai berikut dapat dilihat pada gambar 4.35 seperti dibawah ini :



Gambar 4.35 Pengukuran *AUC* NB Diskritisasi 3 Bin define Boundaries

Nilai yang diperoleh dari AUC adalah 0.947 untuk class predictive non malware(benign). Secara keseluruhan model yang dihasilkan dengan metode naïve bayes diskritisasi 3 variabel dengan batasan min max(-1,1) terlihat pada nilai accuracy, precision dan recall. Untuk klasifikasi nilai AUC tersebut diperoleh dari pengolahan ROC seperti terlihat pada gambar 4.35 (Pengukuran UAC) dengan tingkat diagnosa klasifikasi sangat baik.

d. Dengan Diskritisasi 5 Variabel (no define boundaries)



Gambar 4.36 Proses NB Diskritisasi 5 bin *no define boundaries*

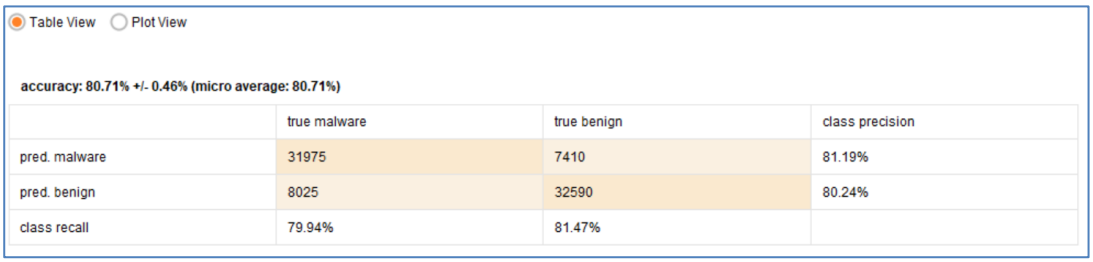


Table View Plot View

accuracy: 80.71% +/- 0.46% (micro average: 80.71%)

	true malware	true benign	class precision
pred. malware	31975	7410	81.19%
pred. benign	8025	32590	80.24%
class recall	79.94%	81.47%	

Gambar 4.37 Confusion Matrix NB Diskritisasi 5 Bin No define Boundaries

Berdasarkan gambar 4.34 di atas menunjukan bahwa dari 80.000 data training, terdapat 31.975 malware yang sesuai dengan prediksi dan 7.410 malware yang tidak sesuai prediksi dengan nilai class precision sebesar 71.37%. Sedangkan dari non malware (benign) terdapat 8.025 benign yang tidak sesuai dengan prediksi dengan nilai precision yang sama yaitu 81.19%. Sehingga tingkat akurasi dengan menggunakan algoritma Naïve Bayes dengan diskritisasi 5 variabel tanpa batasan min max(-1,1) adalah 80,71%. Dalam menentukan persentase akurasi dari data yang sudah diolah, maka formula yang digunakan adalah sebagai berikut:

Tabel 4.23 Confusion Matrix NB Diskritisasi
5 Bin No define Boundaries

NILAI PREDIKSI	**NILAI SEBENARNYA**		
	Class	True (Malware)	False (Benign)
	Malware	31.975 (TP)	7.410 (FP)
	Benign	8.025 (FN)	32.590 (TN)

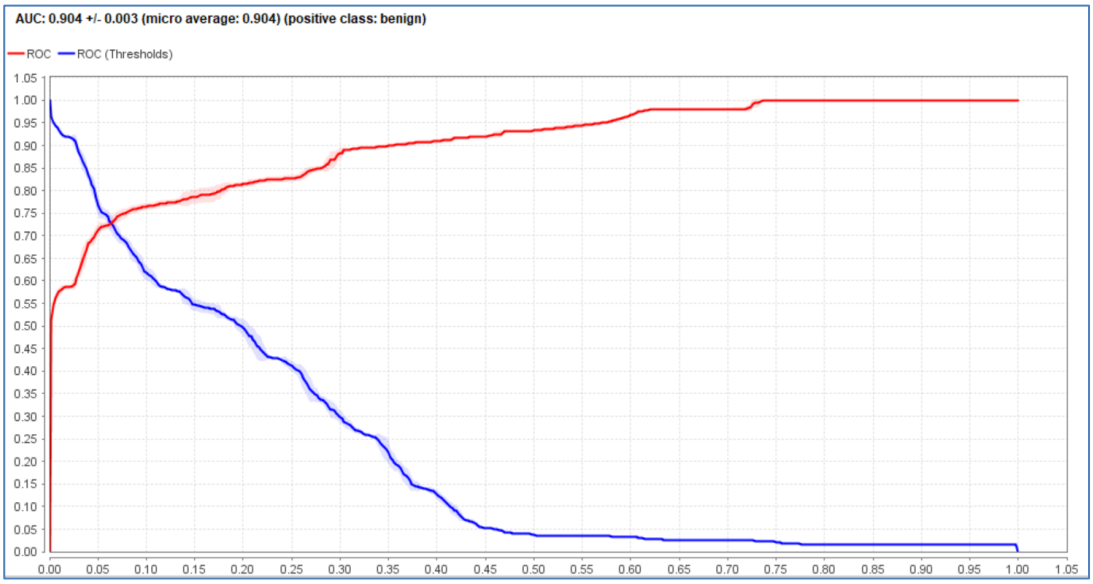
$$\text{Precision} = \frac{TN}{TN+FN} = \frac{32.590}{32.590 + 8.025} = 0.8024 = 80.24\%$$

$$\text{Recall} = \frac{TN}{TN+FP} = \frac{32.590}{32.590 + 7.410} = 0.8147 = 81.47\%$$

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} = \frac{31.975+32.590}{31.975+32.590+7.410+8.025} = 0.8071 = 80.71\%$$

Setelah dilakukan seluruh tahapan evaluasi untuk confusion matrix maka selanjutnya dilakukan analisa evaluasi pembanding yakni terhadap pengukuran

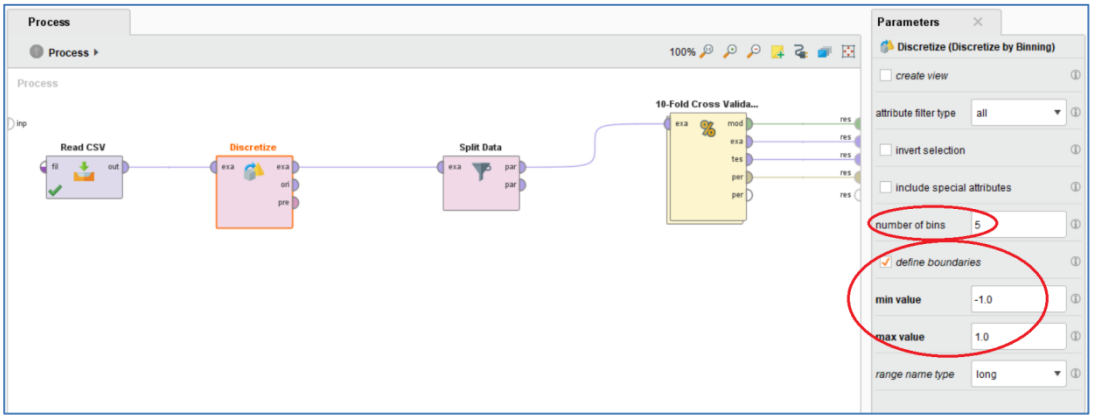
Receiver Operating Characteristic (ROC) sebagai berikut dapat dilihat pada gambar 4.38 seperti gambar di bwah ini :



Gambar 4.38 Pengukuran *AUC* Diskritisasi 5 Bin No define Boundaries

Nilai yang diperoleh dari AUC adalah 0.904 untuk class predictive non malware(benign). Secara keseluruhan model yang dihasilkan dengan metode naïve bayes diskritisasi 5 variabel tanpa batasan min max (-1,1) terlihat pada nilai accuracy, precision dan recall. Untuk klasifikasi nilai AUC tersebut diperoleh dari pengolahan ROC seperti terlihat pada gambar 4.38 (Pengukuran) dengan tingkat diagnosa klasifikasi sangat baik.

e. Dengan Diskritisasi 5 Variabel (define boundaries)



Gambar Proses 4.39 NB Diskritisasi 5 bin *define boundaries*

Pada gambar di atas proses diskritisasi dengan 5 variabel dengan menetapkan batasan nilai minimal = -1 dan maksimal=1 menghasilkan confusion matrix seperti di bawah ini

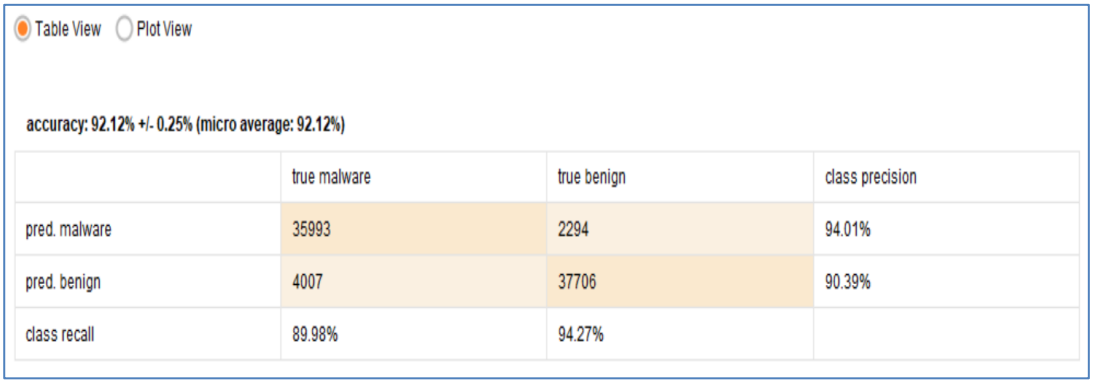


Table View Plot View

accuracy: 92.12% +/- 0.25% (micro average: 92.12%)

	true malware	true benign	class precision
pred. malware	35993	2294	94.01%
pred. benign	4007	37706	90.39%
class recall	89.98%	94.27%	

Gambar 4.40 Confusion Matrix NB Diskritisasi 5 Bin define Boundaries

Berdasarkan gambar 4.36 di atas menunjukan bahwa dari 80.000 data training, terdapat 35.993 malware yang sesuai dengan prediksi dan 2.294 non malware(benign) tidak sesuai prediksi sehingga nilai class precision sebesar 94.01%. Untuknon malware (benign) 37.706 benign sesuai dengan prediksi dengan nilai precision yang sama yaitu 90.39%. Sehingga tingkat akurasi dengan menggunakan algoritma naïve bayes diskritisasi 5 variabel dengan batasan min max (-1,1) adalah 92.12%. Dalam menentukan persentase akurasi dari data yang sudah diolah, maka formula yang digunakan adalah sebagai berikut:

Tabel 4.24 Confusion Matrix NB Diskritisasi
5 Bin define Boundaries

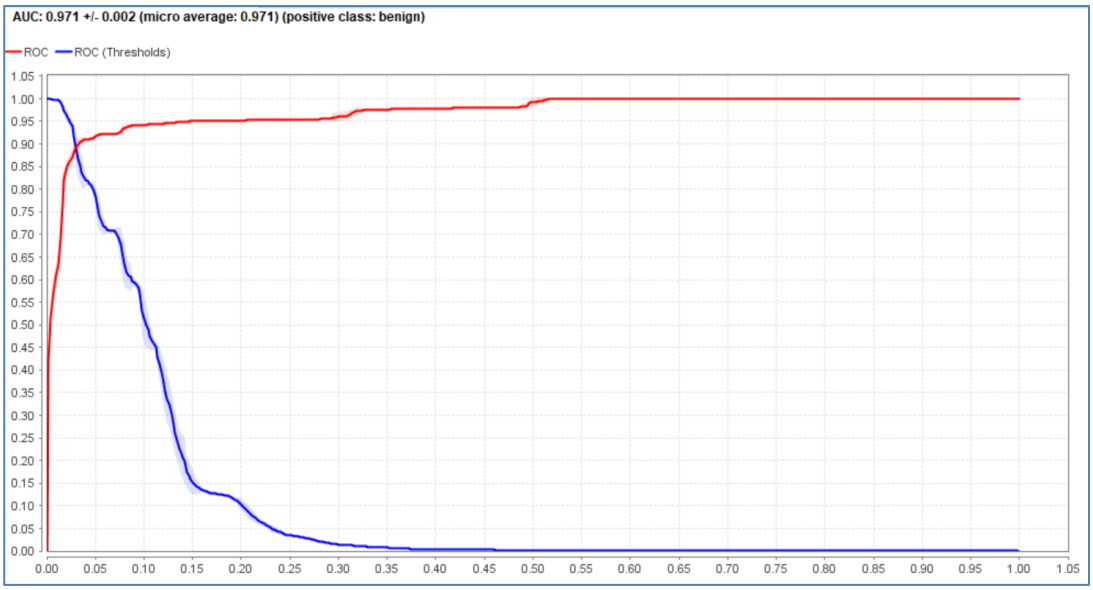
NILAI PREDIKSI	**NILAI SEBENARNYA**		
	Class	True (Malware)	False (Benign)
	Malware	35.993 (TP)	2.294 (FP)
	Benign	4.007 (FN)	37.706 (TN)

$$\text{Precision} = \frac{TN}{TN+FN} = \frac{37.706}{37.706 + 4.007} = 0.9039 = 90.39\%$$

$$\text{Recall} = \frac{TN}{TN+FP} = \frac{37.706}{37.706 + 2.294} = 0.9427 = 94.27\%$$

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} = \frac{35.993+37.706}{35.993 +37.706+2.294+4.007} = 0.9212 = 92.12\%$$

Setelah dilakukan seluruh tahapan evaluasi untuk confusion matrix maka selanjutnya dilakukan analisa evaluasi pembanding yakni terhadap pengukuran *Receiver Operating Characteristic (ROC)* sebagai berikut dapat dilihat pada gambar 4.41 seperti dibawah ini :



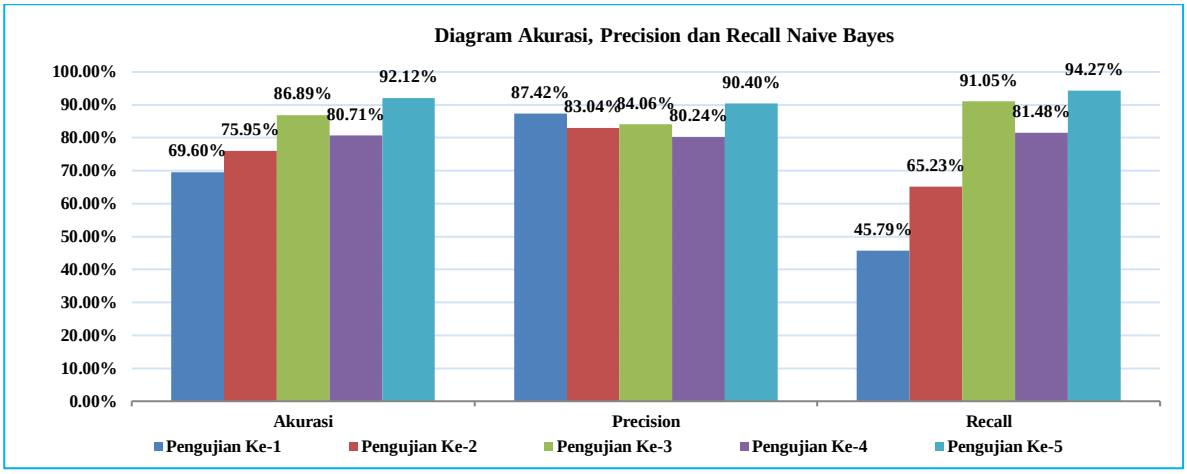
Gambar 4.41 Pengukuran *AUC NB* Diskritisasi 5 Bin define Boundaries

Nilai yang diperoleh dari AUC adalah 0.971 untuk class predictive non malware(benign). Secara keseluruhan model yang dihasilkan dengan metode naïve bayes diskritisasi 5 variabel dengan batasan min max(-1,1) terlihat pada nilai accuracy, precision dan recall. Untuk klasifikasi nilai AUC tersebut diperoleh dari pengolahan ROC seperti terlihat pada gambar 4.41 (Pengukuran UAC Pessimistic) dengan tingkat diagnosa klasifikasi sangat baik.

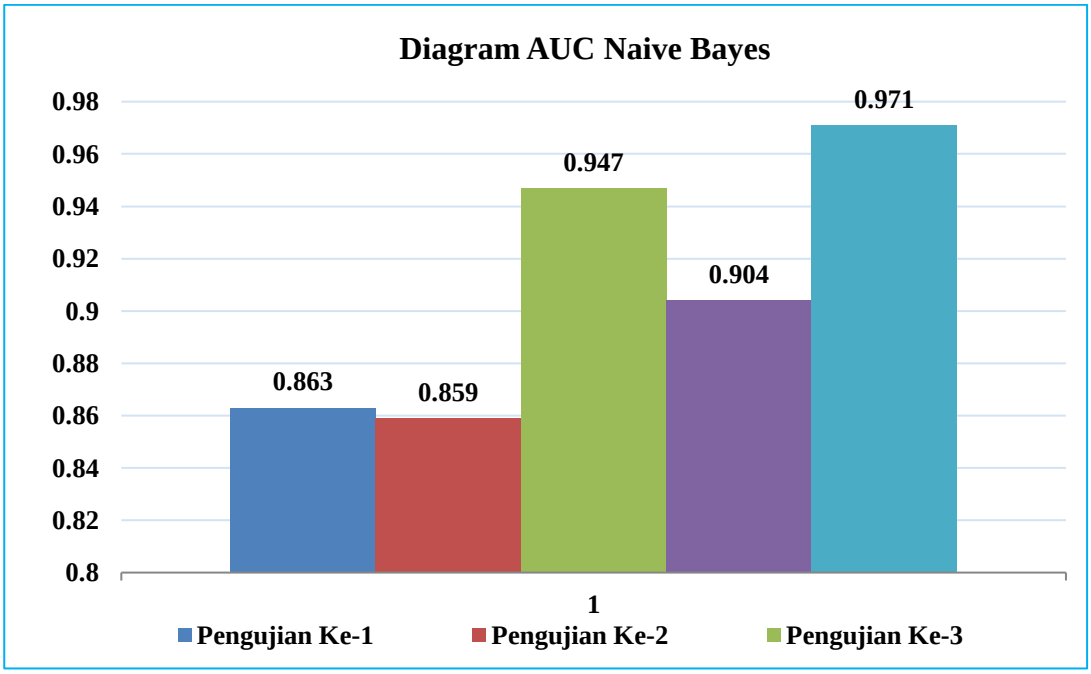
Hasil lengkap ke-5 pengujian algoritma Naïve Bayes di atas (pengujian sesuai Tabel) di tampilkan ke dalam tabel dan diagram seperti di bawah ini :

Tabel 4.25 Hasil Pengujian Algoritma Naïve Bayes

Hasil	Akurasi	Precision	Recall	AUC
Pengujian Ke-1	69.60%	87.42%	45.79%	0.863
Pengujian Ke-2	75.95%	83.04%	65.23%	0.859
Pengujian Ke-3	86.89%	84.06%	91.05%	0.947
Pengujian Ke-4	80.71%	80.24%	81.48%	0.904
Pengujian Ke-5	92.12%	90.40%	94.27%	0.971



Gambar 4.42 Diagram Akurasi, Precision dan Recall Naïve Bayes

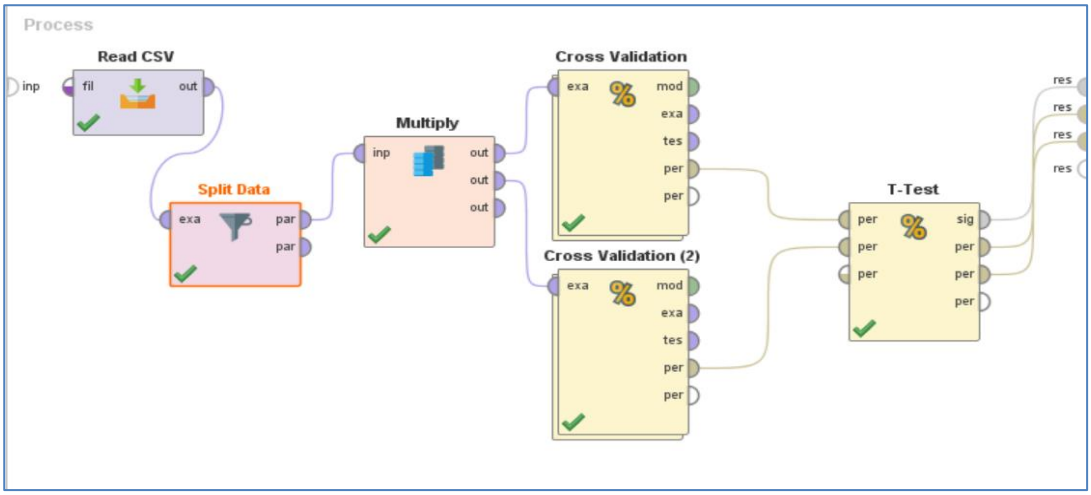


Gambar 4.43 Diagram AUC Naïve Bayes

4.5 Analisa Komparasi

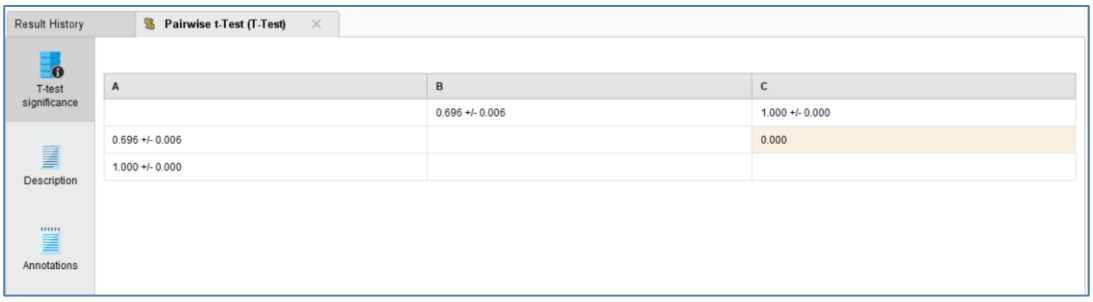
Percobaan dilakukan untuk mengetahui tingkat akurasi dari algoritma Decision Tree C4.5 dan algoritma Naïve Bayes dengan diskritisasi variabel serta menambahkan metode define boundaries min max(-1,1) yang dilakukan pada data set malware sebanyak 100.000 record. Percobaan penelitian ini menggunakan Rapidminer Studio 9.3.0 Validasinya menggunakan 10 fold cross-validation,

sedangkan pengukuran performanya menggunakan confussion matrix. Selanjutnya ditambahkan diskritisasi variabel metode define boundaries sebagai pembanding. Selanjutnya dilakukan pengujian pembandingan antara masing-masing variabel yang di dapat dengan menggunakan pengujian t-test pada rapidminer seperti gambar di bawah ini :



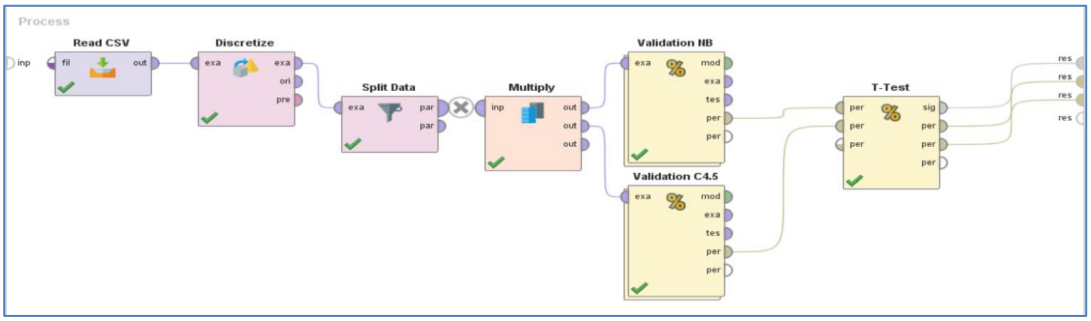
Gambar 4.44 T-test C4.5 dan NB tanpa diskritisasi

Hasil dari t-test yang telah dilakukan tersaji dalam gambar seperti di bawah ini :

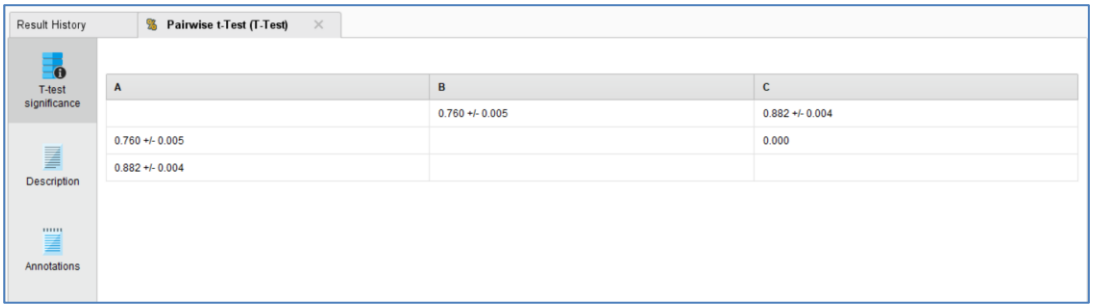


Gambar 4.45 Hasil T-test dan NB tanpa diskritisasi

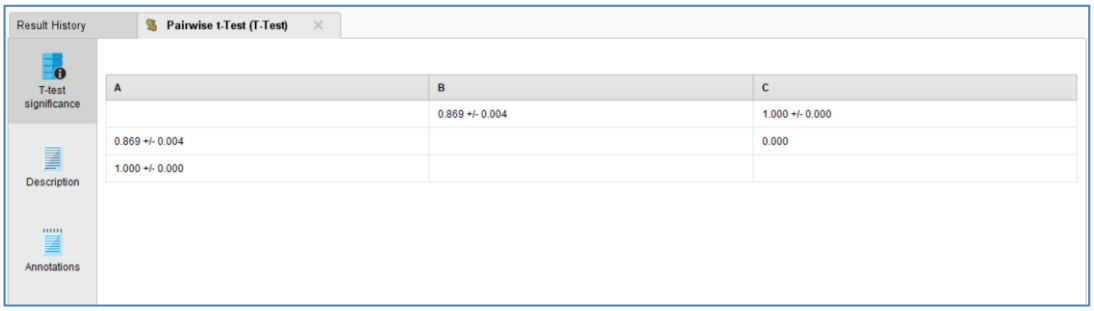
Jika menggunakan diskritisasi seperti tampilan gambar 4.42 di bawah ini :



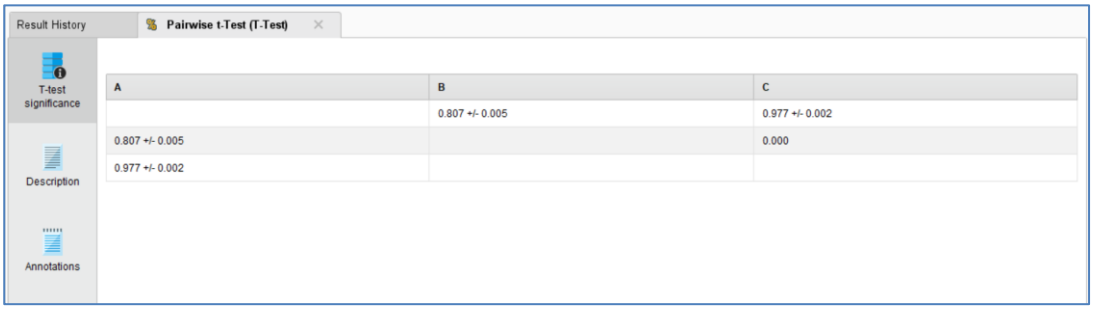
Gambar 4.46 T-test C4.5 dan NB dengan diskritisasi



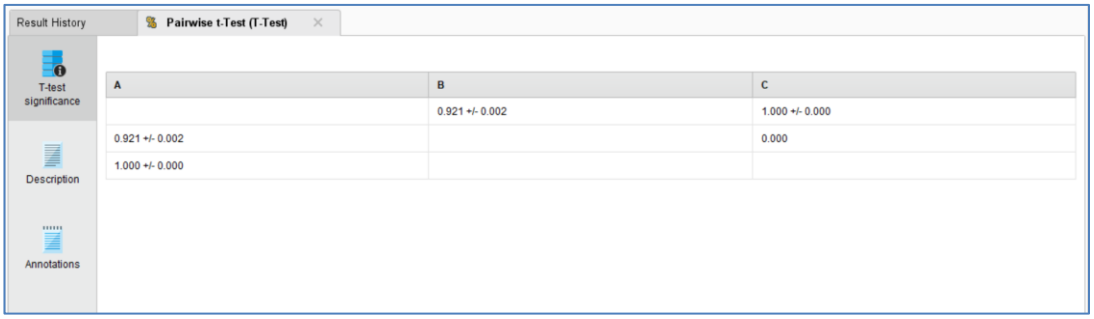
Gambar 4.47 Hasil T-test C4.5 dan NB dengan diskritisasi 3 Variabel Tanpa Parameter Batasan -1 dan 1



Gambar 4.48 Hasil T-test C4.5 dan NB dengan diskritisasi 3 Variabel dengan Parameter Batasan -1 dan 1



Gambar 4.49 Hasil T-test C4.5 dan NB dengan diskritisasi 5 Variabel Tanpa Parameter Batasan -1 dan 1



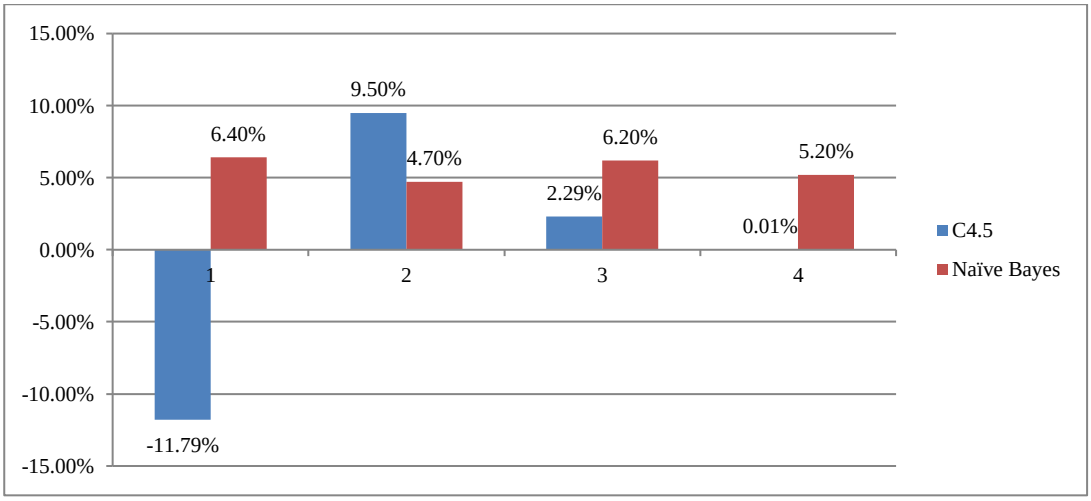
Gambar 4.50 Hasil T-test C4.5 dan NB dengan diskritisasi 5 Variabel dengan Parameter Batasan -1 dan 1

Hasil lengkap pengujian dengan T-test algoritma C4.5 dan Naïve bayes seperti tabel 4.26 di bawah ini :

Tabel 4.26 Hasil Pengujian T-Test C4.5 dan Naïve Bayes

No	Metode	Akurasi	
		C4.5	Naïve Bayes
1	Tanpa diskritisasi	99.99%	69.60%
2	Diskritisasi 3 Variabel Tanpa Parameter Batasan -1 dan 1	88.20%	76.00%
3	Diskritisasi 3 Variabel dengan Parameter Batasan -1 dan 1	99.99%	86.90%
4	Diskritisasi 5 Variabel Tanpa Parameter Batasan -1 dan 1	97.70%	80.70%
5	Diskritisasi 5 Variabel Dengan Parameter Batasan -1 dan 1	100%	92.10%

Berdasarkan hasil percobaan, menunjukkan bahwa penggunaan diskritisasi variabel dapat meningkatkan akurasi algoritma naïve bayes sebesar 6.40% dari 69.60% menjadi 76.00% dengan diskritisasi 3 variabel tanpa input parameter batasan -1 dan 1, meningkat lagi sebesar 4.7% menjadi 80.70% dengan diskritisasi 5 variabel tanpa input parameter batasan -1 dan 1, akan meningkat kembali sebesar 6.2% dengan diskritisasi 3 variabel input parameter batasan -1 dan 1 menjadi 86.90% serta meningkat kembali sebesar 5.2% menjadi 92.1% dengan diskritisasi 5 variabel input parameter batasan -1 dan 1. Sedangkan untuk algoritma C4.5 kriteria gini index yang semula 99.99% tanpa diskritisasi terjadi penurunan sebesar 11,79% menjadi 88,20% dengan diskritisasi. Akan tetapi meningkat kembali sebesar 9.5% menjadi 97.70 dari 88.20% dengan diskritisasi 3 variabel tanpa input parameter batasan -1 dan 1, meningkat kembali sebesar 2.29% menjadi 99.99% dari 97.70% dengan diskritisasi 3 variabel dengan input batasan parameter -1 dan 1, meningkat kembali menjadi 100% dari 99.99% dengan diskritisasi 5 variabel dengan input batasan parameter -1 dan 1. Untuk diagram kenaikan tingkat akurasi pada algoritma C4.5 dan naïve bayes terlihat seperti di bawah ini :

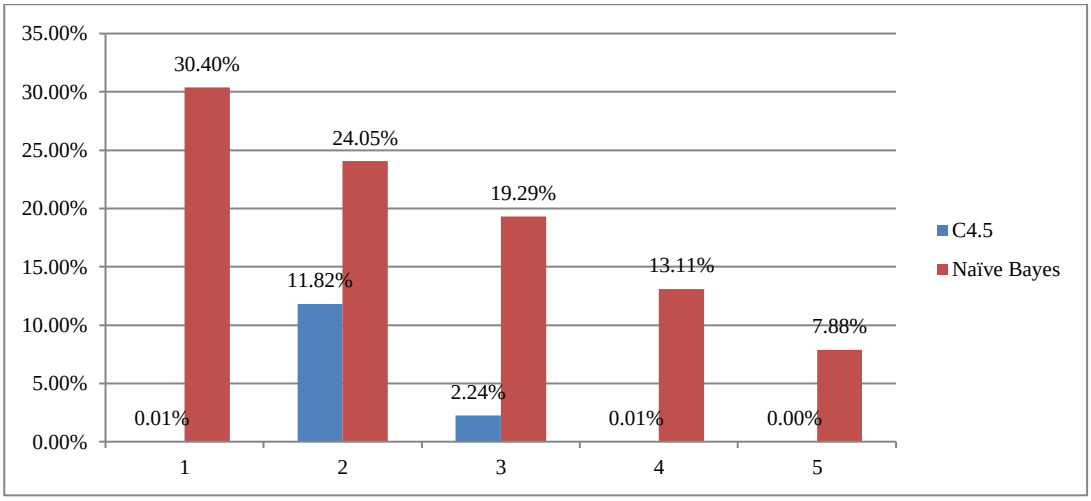


Gambar 4.51 Diagram Kenaikan Tingkat Akurasi Metode C4.5 dan Naïve Bayes

Pada diagram di atas terlihat bahwa penggunaan teknik diskritisasi pada algoritma naïve bayes mampu meningkatkan tingkat akurasi dari algoritma tersebut, sedangkan pada decision tree C4.5 tidak terjadi peningkatan performa pada tingkat akurasinya. Sedangkan hasil penghitungan tingkat kesalahan (incorrectly classified) pada algoritma C4.5 dengan naïve bayes adalah sebagai berikut :

Tabel 4.27 Tingkat Kesalahan Klasifikasi C4.5 dan Naïve Bayes

No	Metode	Kesalahan Klasifikasi	
		C4.5	Naïve Bayes
1	Tanpa diskritisasi	0.01%	30.40%
2	Diskritisasi 3 Variabel Tanpa Parameter Batasan -1 dan 1	11.82%	24.05%
3	Diskritisasi 3 Variabel dengan Parameter Batasan -1 dan 1	0.01%	13.11%
4	Diskritisasi 5 Variabel Tanpa Parameter Batasan -1 dan 1	2.24%	19.29%
5	Diskritisasi 5 Variabel Dengan Parameter Batasan -1 dan 1	0.00%	7.88%



Gambar 4.52 Diagram Tingkat Kesalahan Klasifikasi C4.5 dan Naïve Bayes

Terlihat pada tabel dan gambar di atas tingkat kesalahan klasifikasi C4.5 fluktuatif naik dan turun, sedangkan untuk naïve bayes cenderung menurun tingkat kesalahan klasifikasinya.