

**ALGORITMA FUZZY C-MEANS DAN CLASSIFICATION
AND REGRESSION TREES UNTUK PENGELOMPOKAN
DAN PENGKLASIFIKASI KOLEKTIBILITAS
CALON DEBITUR**

SKRIPSI

Sebagai salah satu syarat untuk memperoleh gelar
Sarjana Statistika

Oleh:
PUTRI FIDYA FARRADILLA
155090500111020



**PROGRAM STUDI SARJANA STATISTIKA
JURUSAN STATISTIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2019**

LEMBAR PENGESAHAN SKRIPSI

ALGORITMA FUZZY C-MEANS DAN CLASSIFICATION AND REGRESSION TREES UNTUK PENGELOMPOKAN DAN PENGKLASIFIKASI KOLEKTIBILITAS CALON DEBITUR

oleh:
PUTRI FIDYAFARRADILLA
155090500111020

**Setelah dipertahankan di depan Majelis Penguji
pada tanggal 18 Juli 2019
dan dinyatakan memenuhi syarat untuk memperoleh gelar
Sarjana Statistika**

Dosen Pembimbing

**Dr. Dra. Ani Budi Astuti, MSi.
NIP. 196802091992032001**

**Mengetahui,
Ketua Jurusan Statistika
Fakultas MIPA
Universitas Brawijaya**

**Rahma Fitriani, S.Si., M.Sc., Ph.D
NIP. 197603281999032001**

LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Putri Fidya Farradilla
NIM : 155090500111020
Jurusan : Statistika
Judul Skripsi : *Algoritma Fuzzy C-Means Dan Classification And Regression Trees Untuk Pengelompokan Dan Pengklasifikasian Kolektibilitas Calon Debitur*

ALGORITMA FUZZY C-MEANS DAN CLASSIFICATION AND REGRESSION TREES UNTUK PENGELOMPOKAN DAN PENGKLASIFIKASIAN KOLEKTIBILITAS CALON DEBITUR

Dengan ini menyatakan bahwa:

1. Isi dari Skripsi yang saya buat adalah benar-benar karya sendiri dan tidak menjiplak karya orang lain, selain nama-nama yang bermaksud di isi dan tertulis di daftar pustaka dalam Skripsi ini.
2. Apabila dikemudian hari ternyata Skripsi yang saya tulis terbukti hasil jiplakan, maka saya akan bersedia menanggung segala risiko yang saya terima.

Demikian pernyataan ini dibuat dengan segala kesadaran.

Malang, 18 Juli 2019
Yang menyatakan,

Putri Fidya Farradilla
NIM. 155090500111020

**ALGORITMA FUZZY C-MEANS DAN CLASSIFICATION AND
REGRESSION TREES UNTUK PENGELOMPOKAN DAN
PENGKLASIFIKASIAN KOLEKTIBILITAS
CALON DEBITUR**

ABSTRAK

Fuzzy C-means (FCM) adalah salah satu metode pengelompokan dalam metode pengelompokan non hierarki. *Classification and Regression Trees (CART)* merupakan salah satu pendekatan yang praktis dalam mengklasifikasikan objek penelitian ke dalam pohon klasifikasi. Tujuan dari penelitian ini adalah menentukan pengelompokan calon debitur di 18 kota di Jawa Timur berdasarkan prinsip penyekoran kredit 5C yang optimal dengan menggunakan algoritma *FCM*. Kemudian, calon debitur pada masing-masing *cluster* yang telah terbentuk diklasifikasikan ke dalam kelas kredit lancar atau kelas kredit macet menggunakan pohon klasifikasi *Classification and Regression Trees (CART)*. Hasil penelitian menunjukkan bahwa pengelompokan yang optimal dengan algoritma *FCM* menggunakan parameter *fuzzification* sebesar 2 dan membentuk empat *cluster* calon debitur. *Cluster* satu merupakan *cluster* “kemampuan baik”, *cluster* dua merupakan *cluster* “kemampuan dan modal baik”, *cluster* tiga merupakan *cluster* “watak dan kondisi perekonomian baik” dan *cluster* empat merupakan *cluster* “unggulan” Pohon klasifikasi *CART* pada *cluster* satu merupakan pohon dengan dua simpul terminal, pada *cluster* dua merupakan pohon dengan dua simpul terminal, pada *cluster* tiga merupakan pohon dengan tiga simpul terminal dan pada *cluster* empat merupakan pohon dengan tiga simpul terminal. Pohon klasifikasi disetiap *cluster* memiliki nilai ketepatan hasil klasifikasi keseluruhan (*Hit Ratio*) lebih dari 50%, sehingga pohon klasifikasi yang terbentuk sudah cukup baik dan tepat dalam mengklasifikasikan calon debitur

Kata Kunci : Analisis *Cluster*, *Classification and Regression Trees*, *Fuzzy C-means*, Prinsip Penyekoran Kredit 5C

FUZZY C-MEANS ALGORITHM AND CLASSIFICATION AND REGRESSION TREES FOR CLUSTERING AND DEBTOR COLLECTIBILITY CLASSIFICATION

ABSTRACT

Fuzzy C-means (FCM) is one method of grouping in a non-hierarchical grouping method. Classification and Regression Trees (CART) is a practical approach in classifying research objects into classification trees. The purpose of this study is to determine the grouping of prospective debtors in 18 cities in East Java based on the principle of optimal 5C credit scoring using the FCM algorithm. Then, the prospective debtor in each cluster that has been formed is classified into the current credit class or bad credit class using the Classification and Regression Trees (CART) classification tree. The results showed that the optimal grouping with the FCM algorithm used fuzzification parameters of 2 and formed four clusters of prospective debtors. Cluster one is a cluster of "good abilities", cluster two is a cluster of "good capabilities and capital", cluster three is a cluster of "good character and economic condition" and cluster four is a cluster of "superior" CART classification tree in cluster one is a tree with two vertices terminal, in cluster two is a tree with two terminal nodes, in cluster three is a tree with three terminal nodes and in cluster four is a tree with three terminal nodes. The classification tree in each cluster has a value of the accuracy of the overall classification (Hit Ratio) of more than 50%, so that the classification tree that is formed is quite good and appropriate in classifying prospective debtors.

Keywords : Cluster Analysis, Classification and Regression Trees, Collectability of Prospective Debtors, Fuzzy C-means

KATA PENGANTAR

Syukur Alhamdulillah penulis ucapkan kepada Allah SWT atas segala rahmat, hidayah, dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Algoritma *Fuzzy C-Means* Dan *Classification and Regression Trees* Untuk Pengelompokan Dan Pengklasifikasian Kolektibilitas Calon Debitur”. Penelitian ini disusun untuk memenuhi salah satu syarat dalam memperoleh gelar Sarjana Statistika.

Dalam penyusunan penelitian ini tidak dapat diselesaikan dengan baik tanpa dukungan, motivasi, kerjasama maupun bimbingan dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan terima kasih kepada:

1. Ibu Dr. Dra. Ani Budi Astuti, M.Si. selaku dosen pembimbing skripsi atas waktu dan bimbingan yang telah diberikan selama proses penulisan penelitian ini dengan sabar dan penuh perhatian.
2. Bapak Samingun Handoyo, S.Si., M.Cs. selaku dosen penguji I yang telah memberikan saran dan bimbingan dalam penelitian ini.
3. Ibu Nurjannah, S.Si., M.Phil., Ph.D. selaku dosen penguji II yang telah memberikan saran dan bimbingan dalam penelitian ini.
4. Bapak Achmad Efendi, S.Si., M.Sc., Ph.D. selaku ketua Program Studi Sarjana Statistika FMIPA Universitas Brawijaya
5. Ibu Rahma Fitriani, S.Si., M.Sc., Ph.D. selaku ketua Jurusan Statistika FMIPA Universitas Brawijaya.
6. Mamah Sri Dyah Agusnita, adik Shadeka Putratama Fidya, dan seluruh keluarga besar atas kasih sayang, do'a, dan dukungannya yang selalu menyertai penulis.
7. Ibu dan Bapak Dosen KKU.PSBM atas bimbingan dan dukungannya
8. Teman KKU.PSBM untuk dukungan selama ini.
9. Keluarga KKU.PSBM khususnya Batch 4, yaitu Briga, Iqbal, Isam, Hanny, Trias, Chacha, Putri, Bella, Armita, Ghea dan Lyshe atas dukungan, tawa dan canda yang selalu menemani penulis.
10. Teman-teman HIMAHENA, yaitu Dirga, Oki, Hanny, Trias, Putri, Chacha, Isam, Zainal, Derin, Mesya, dan Sasgia yang selalu menemani dan memberikan semangat dan dukungan sejak Semester satu perkuliahan.

11. Teman-teman semasa putih abu-abu, yaitu Aura, Chika, Nanda, Mandan dan Yola yang selalu memberikan dukungan dan semangat yang tiada hentinya.
12. Hadi Tulus Wibowo, S.Si. yang selalu memberikan dukungan, semangat dan selalu menemani penulis dalam menyelesaikan penulisan peneitian ini.
13. Semua pihak yang telah membantu penyelesaian penelitian ini.

Penyusunan skripsi ini masih banyak kesalahan. Untuk itu, penulis mengharapkan kritik dan saran yang bersifat membangun demi perbaikan dan penyempurnaan penelitian ini. Semoga skripsi ini dapat bermanfaat bagi semua pihak dan penulis pada khususnya.

Malang, 18 Juli 2019

Penulis

DAFTAR ISI

	Hal
LEMBAR PENGESAHAN SKRIPSI	iii
LEMBAR PERNYATAAN	v
ABSTRAK	vii
ABSTRACT	ix
KATAPENGANTAR	xi
DAFTAR ISI	xv
DAFTAR TABEL	xxi
DAFTAR GAMBAR	xxv
DAFTAR LAMPIRAN	xxvii
BAB I. PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan Penelitian	3
1.4. Manfaat Penelitian	3
1.5. Batasan Masalah	3
BAB II. TINJAUAN PUSTAKA	5
2.1. Standarisasi Data	5
2.2. Analisis <i>Cluster</i>	5
2.3. Konsep Jarak	6
2.3.1. Jarak <i>Euclidean</i>	6
2.4. Analisis <i>Cluster</i> Nonhierarki	7
2.5. <i>Fuzzy Clustering</i>	7
2.5.1. <i>Fuzzy C-means (FCM)</i>	8
2.5.2. Algoritma <i>Fuzzy C-means</i>	10
2.6. Indeks Validitas <i>Cluster</i>	12
2.7. <i>Decision Tree</i>	12
2.8. <i>Classification and Regression Trees (CART)</i>	14
2.8.1. Pembentukan Pohon Klasifikasi	14
2.8.2. Pemangkasan Pohon Klasifikasi	16
2.9. Kolektibilitas	17
2.9.1. Penggolongan Kolektibilitas	17
2.10. Variabel Penelitian	17
2.10.1. <i>Character</i> (Karakter)	17

	Hal.
2.10.2. <i>Capacity</i> (Kemampuan)	18
2.10.3. <i>Capital</i> (Modal)	18
2.10.4. <i>Collateral</i> (Jaminan)	19
2.10.5. <i>Condition of Economy</i> (Kondisi Ekonomi)	19
BAB III. METODE PENELITIAN	21
3.1. Sumber Data.....	21
3.2. Variabel Penelitian	22
3.3. Penentuan Parameter dalam Algoritma <i>Fuzzy C-means</i> (<i>FCM</i>).....	23
3.4. Penggolongan Kolektibilitas	24
3.5. Metode Penelitian.....	24
3.6. Diagram Alir	27
BAB IV. HASIL DAN PEMBAHASAN	31
4.1. Analisis Deskriptif	31
4.2. Hasil Algoritma <i>Fuzzy C-means</i>	34
4.2.1. Hasil Pengelompokan Kombinasi Ke-1	34
4.2.2. Hasil Pengelompokan Kombinasi Ke-2	35
4.2.3. Hasil Pengelompokan Kombinasi Ke-3	37
4.2.4. Hasil Pengelompokan Kombinasi Ke-4	38
4.2.5. Hasil Pengelompokan Kombinasi Ke-5	40
4.2.6. Hasil Pengelompokan Kombinasi Ke-6	41
4.2.7. Hasil Pengelompokan Kombinasi Ke-7	42
4.2.8. Hasil Pengelompokan Kombinasi Ke-8	44
4.2.9. Hasil Pengelompokan Kombinasi Ke-9	45
4.3. Pengelompokan Paling Optimal	46
4.4. Interpretasi Hasil Pengelompokan Kombinasi ke-6...47	47
4.5. Analisis <i>CART</i>	50
4.5.1. Pohon Klasifikasi <i>CART</i> pada <i>Cluster</i> Satu	51
4.5.2. Pohon Klasifikasi <i>CART</i> pada <i>Cluster</i> Dua.....	54
4.5.3. Pohon Klasifikasi <i>CART</i> pada <i>Cluster</i> Tiga	57
4.5.4. Pohon Klasifikasi <i>CART</i> pada <i>Cluster</i> Empat ..	60
BAB V. PENUTUP.....	65
5.1. Kesimpulan.....	65
5.2. Saran.....	65

	Hal.
DAFTAR PUSTAKA	67
LAMPIRAN.....	71

DAFTAR TABEL

	Hal.
Tabel 3.1. Struktur Data Penelitian	21
Tabel 3.2. Variabel Penelitian	22
Tabel 3.3. Parameter Dalam Algoritma <i>FCM</i>	23
Tabel 3.4. Kombinasi Parameter Dalam Algoritma <i>FCM</i>	23
Tabel 4.1. Nilai Rata-Rata Tertinggi dan Terendah Tiap Variabel ..	32
Tabel 4.2. Nilai Parameter Pada Pengelompokan Kombinasi Ke-1.34	
Tabel 4.3. Anggota Tiap <i>Cluster</i> Pada Pengelompokan Kombinasi Ke-1.....	35
Tabel 4.4. Nilai Variabel 5C Tiap <i>Cluster</i>	35
Tabel 4.5. Nilai Parameter pada Pengelompokan Kombinasi Ke-2.....	35
Tabel 4.6. Anggota Tiap <i>Cluster</i> Pada Pengelompokan Kombinasi Ke-2.....	36
Tabel 4.7. Nilai Rata-rata Tiap <i>Cluster</i>	36
Tabel 4.8. Nilai Parameter pada Pengelompokan Kombinasi Ke-3.....	37
Tabel 4.9. Anggota Tiap <i>Cluster</i> Pada Pengelompokan Kombinasi Ke-3.....	37
Tabel 4.10. Nilai Rata-rata Tiap <i>Cluster</i>	37
Tabel 4.11. Nilai Parameter pada Pengelompokan Kombinasi Ke-4.....	38
Tabel 4.12. Anggota Tiap <i>Cluster</i> Pada Pengelompokan Kombinasi Ke-4.....	39
Tabel 4.13. Nilai Rata-rata Tiap <i>Cluster</i>	39
Tabel 4.14. Nilai Parameter pada Pengelompokan Kombinasi ke-5.....	40
Tabel 4.15. Anggota Tiap <i>Cluster</i> Pada Pengelompokan Kombinasi Ke-5.....	40
Tabel 4.16. Nilai Rata-rata Tiap <i>Cluster</i>	40
Tabel 4.17. Nilai Parameter pada Pengelompokan Kombinasi Ke-6.....	41
Tabel 4.18. Anggota Tiap <i>Cluster</i> Pada Pengelompokan Kombinasi Ke-6.....	41
Tabel 4.19. Nilai Rata-rata Tiap <i>Cluster</i>	42

Tabel 4.20. Nilai Parameter pada Pengelompokan Kombinasi Ke-7.....	42
Tabel 4.21. Anggota Tiap <i>Cluster</i> Pada Pengelompokan Kombinasi Ke-7.....	43
Tabel 4.22. Nilai Rata-rata Tiap <i>Cluster</i>	43
Tabel 4.23. Nilai Parameter pada Pengelompokan Kombinasi Ke-8.....	44
Tabel 4.24. Anggota Tiap <i>Cluster</i> Pada Pengelompokan Kombinasi Ke-8.....	44
Tabel 4.25. Nilai Rata-rata Tiap <i>Cluster</i>	44
Tabel 4.26. Nilai Parameter pada Pengelompokan Kombinasi Ke-9.....	45
Tabel 4.27. Anggota Tiap <i>Cluster</i> Pada Pengelompokan Kombinasi Ke-9.....	45
Tabel 4.28. Nilai Rata-rata Tiap <i>Cluster</i>	46
Tabel 4.29. Nilai Indeks Xie-Beni Masing-masing Pengelompokan	46
Tabel 4.30. Anggota Tiap <i>Cluster</i>	48
Tabel 4.31. Hasil Perhitungan CV pada Pohon Klasifikasi	52
Tabel 4.32. Tabel Konfusi <i>CART</i> pada <i>Cluster</i> Satu	53
Tabel 4.33. Hasil Perhitungan CV pada Pohon Klasifikasi	55
Tabel 4.34. Tabel Konfusi <i>CART</i> pada <i>Cluster</i> Dua	56
Tabel 4.35. Hasil Perhitungan CV pada Pohon Klasifikasi	58
Tabel 4.36. Tabel Konfusi <i>CART</i> pada <i>Cluster</i> Tiga	59
Tabel 4.37. Hasil Perhitungan CV pada Pohon Klasifikasi	61
Tabel 4.38. Tabel Konfusi <i>CART</i> pada <i>Cluster</i> Tiga	62

DAFTAR GAMBAR

	Hal.
Gambar 2.1. Pengelompokan Non <i>fuzzy</i>	8
Gambar 2.2. Pengelompokan <i>Fuzzy</i>	8
Gambar 2.3. Contoh <i>Decision Tree</i>	13
Gambar 3.1. Diargam Alir Penelitian Algoritma <i>Fuzzy C-means</i>	27
Gambar 3.2. Diagram Alir Algoritma <i>CART</i>	29
Gambar 4.1. Analisis Deskriptif Variabel 5C.....	31
Gambar 4.2. Rata-rata Variabel 5C Pada Setiap <i>Cluster</i>	48
Gambar 4.3. Pohon Klasifikasi <i>CART</i> Optimal Pada <i>Cluster</i> Satu	51
Gambar 4.4. Plot <i>Complexity Parameter</i>	52
Gambar 4.5. Pohon Klasifikasi <i>CART</i> Optimal Pada <i>Cluster</i> Dua	54
Gambar 4.6. Plot <i>Complexity Parameter</i>	55
Gambar 4.7. Pohon Klasifikasi <i>CART</i> Optimal Pada <i>Cluster</i> Tiga.....	57
Gambar 4.8. Plot <i>Complexity Parameter</i>	58
Gambar 4.9. Pohon Klasifikasi <i>CART</i> Optimal Pada <i>Cluster</i> Empat.....	60
Gambar 4.10. Plot <i>Complexity Parameter</i>	61

DAFTAR LAMPIRAN

Hal.

Lampiran 1. <i>Source code</i> untuk Algoritma <i>Fuzzy C-means</i>	71
Lampiran 2. <i>Source Code</i> untuk Algoritma <i>Fuzzy C-means</i> (<i>Package R</i>).....	75
Lampiran 3. <i>Source Code</i> untuk <i>Classification and Regression</i> <i>Trees (Package Rstudio)</i>	75
Lampiran 4. <i>Output</i> Algoritma <i>Fuzzy C-means</i> Pada Pengelompokan Kombinasi 1.....	76
Lampiran 5. <i>Output</i> Algoritma <i>Fuzzy C-means</i> Pada Pengelompokan Kombinasi 2.....	80
Lampiran 6. <i>Output</i> Algoritma <i>Fuzzy C-means</i> Pada Pengelompokan Kombinasi 3.....	83
Lampiran 7. <i>Output</i> Algoritma <i>Fuzzy C-means</i> Pada Pengelompokan Kombinasi 4.....	86
Lampiran 8. <i>Output</i> Algoritma <i>Fuzzy C-means</i> Pada Pengelompokan Kombinasi 5.....	89
Lampiran 9. <i>Output</i> Algoritma <i>Fuzzy C-means</i> Pada Pengelompokan Kombinasi 6.....	92
Lampiran 10. <i>Output</i> Algoritma <i>Fuzzy C-means</i> Pada Pengelompokan Kombinasi 7	95
Lampiran 11. <i>Output</i> Algoritma <i>Fuzzy C-means</i> Pada Pengelompokan Kombinasi 8	98
Lampiran 12. <i>Output</i> Algoritma <i>Fuzzy C-means</i> Pada Pengelompokan Kombinasi 9	101
Lampiran 13. <i>Output</i> Algortima <i>CART</i> Pada <i>Cluster 1</i>	104
Lampiran 14. <i>Output</i> Algortima <i>CART</i> Pada <i>Cluster 2</i>	106
Lampiran 15. <i>Output</i> Algortima <i>CART</i> Pada <i>Cluster 3</i>	108
Lampiran 16. <i>Output</i> Algortima <i>CART</i> Pada Cluster 4.....	112

BAB I

PENDAHULUAN

1.1. Latar Belakang

Analisis *cluster* merupakan proses pengelompokan sekumpulan objek penelitian ke dalam beberapa *cluster* (Johnson dan Wichern, 2007). Salah satu metode dalam pengelompokan adalah metode non hierarki. Pengelompokan dengan metode non hierarki digunakan ketika jumlah *cluster* telah diketahui, sehingga dapat dikatakan jumlah *cluster* pada metode ini ditentukan terlebih dahulu oleh peneliti (Handoyo dan Prasojo, 2017).

Menurut Ross (2010), salah satu metode pengelompokan dalam metode non hierarki adalah *Fuzzy C-means (FCM)*. Pengelompokan tiap-tiap objek dengan *FCM* ditentukan oleh derajat keanggotaan yang bernilai 0 sampai 1, sehingga tiap objek memiliki kemungkinan berada dalam satu *cluster* atau lebih (Handoyo dan Prasojo, 2017). Besar derajat keanggotaan dipengaruhi oleh koefisien parameter *fuzzification* (w). Bezdek (1981), mengusulkan untuk mendapatkan pengelompokan yang optimal dalam satu algoritma *FCM* dapat menggunakan nilai w yang lebih dari satu. Penentuan pengelompokan yang optimal berdasarkan penggunaan nilai w yang berbeda, dapat dilakukan uji validitas menggunakan indeks Xie-Beni (Balasko dkk., 2007).

Classification and Regression Trees (CART) merupakan salah satu pendekatan yang praktis dalam permasalahan klasifikasi. Teknik *CART* dalam membuat sebuah pohon klasifikasi dikenal dengan istilah *Binary Recursive Partitioning*. Proses tersebut dikatakan *binary* karena setiap simpul data (*node*) akan selalu mengalami pemisahan kedalam *child node*. Proses pemisahan akan terus dilakukan sampai terbentuk *node* terminal.

Pada penelitian Amalia (2016), *Fuzzy C-means* diterapkan untuk pengelompokan rasio keuangan CAMLES pada tahun 2014. Setelah melakukan pengelompokan, langkah selanjutnya adalah mencari banyaknya *cluster* yang optimal menggunakan indeks Xie-Beni. Pada penelitian Kartiko dan Tanjung (2017), *Classification and Regression Trees (CART)* diterapkan untuk pengklasifikasian pembayaran kredit nasabah bank. Hasil pengklasifikasian berupa faktor-faktor penciri yang memengaruhi pembayaran kredit oleh nasabah.

Berdasarkan uraian di atas dan merujuk penelitian sebelumnya, peneliti ingin menerapkan Algoritma *Fuzzy C-means* dan Algoritma *CART* pada data calon debitur 18 kota di Jawa Timur yang akan memperoleh fasilitas kredit. Misi bank memberikan kredit adalah membantu usaha debitur yang memerlukan dana (Fitria dan Sari, 2012). Namun, bank harus memberikan perhatian khusus dalam pemberian kredit terhadap calon debitur. Keputusan pemberian kredit memiliki risiko yang tinggi atas ketidakmampuan debitur dalam membayar kredit pada saat jatuh tempo. Salah satu bentuk pertimbangan yang dapat dilakukan pihak bank adalah dengan menggunakan prinsip 5C, yaitu *Character, Capacity, Capital, Collateral* dan *Condition of Economy*. Algoritma *Fuzzy C-means* diterapkan untuk pengelompokan calon debitur nasabah bank berdasarkan prinsip penyekoran kredit 5C. Hasil pengelompokan akan diuji validitas menggunakan indeks Xie-Beni untuk menentukan pengelompokan yang paling optimal. Kemudian, calon debitur pada masing-masing *cluster* yang telah terbentuk diklasifikasikan ke dalam kelas kredit lancar atau kelas kredit macet menggunakan pohon klasifikasi *Classification and Regression Trees (CART)*. Hasil penelitian ini diharapkan dapat memberikan informasi tambahan untuk pihak bank dan dapat dijadikan bahan pertimbangan pemberian kredit dan pemberlakuan kebijakan-kebijakan bank.

1.2. Rumusan Masalah

Berdasarkan uraian dari latar belakang, maka rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana hasil pengelompokan calon debitur bank berdasarkan prinsip penyekoran kredit 5C yang optimal berdasarkan penggunaan koefisien parameter *fuzzification* (w) yang berbeda pada Algoritma *Fuzzy C-means* dengan indeks Xie-Beni?
2. Bagaimana pohon klasifikasi calon debitur pada setiap *cluster* yang telah terbentuk dengan menggunakan *Classification and Regression Trees (CART)*

1.3. Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut:

1. Mengetahui hasil pengelompokan calon debitir bank berdasarkan prinsip penyekoran kredit *5C* yang optimal berdasarkan penggunaan koefisien parameter *fuzzification* (*w*) yang berbeda pada algoritma *Fuzzy C-means* dengan indeks Xie-Beni.
2. Mengetahui pohon klasifikasi calon debitir bank di setiap *cluster* yang telah terbentuk dengan menggunakan *Classification and Regression Trees*.

1.4. Manfaat Penelitian

Manfaat yang akan diperoleh dari penelitian ini adalah sebagai berikut:

1. Mengetahui penerapan Algoritma *Fuzzy C-means* dalam pengelompokan karakteristik calon debitir dengan prinsip penyekoran kredit *5C*.
2. Mengetahui hasil pengelompokan yang optimal berdasarkan penggunaan koefisien parameter *fuzzification* (*w*) yang berbeda pada pengelompokan karakteristik calon debitir
3. Mengetahui penerapan Algoritma *Classification and Regression Trees* dalam membentuk pohon klasifikasi calon debitir pada setiap *cluster* yang telah terbentuk.
4. Mengetahui hasil klasifikasi calon debitir berupa pohon klasifikasi calon debitir pada setiap *cluster* dengan menggunakan *Classification and Regression Trees*.
5. Menyajikan informasi pendukung kepada pihak bank dalam menentukan kebijakan pemberian kredit kepada calon debitir dari hasil pengelompokan calon debitir dengan algoritma FCM.

1.5. Batasan Masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Ukuran kedekatan jarak yang digunakan adalah jarak *euclid*.
2. Parameter *Fuzzification* (*w*) yang digunakan dalam algoritma *Fuzzy C-means* adalah 1,5; 2 dan 2,5.
3. Kriteria penghentian atau kesalahan terkecil yang diharapkan (ξ) adalah 0,001.

4. Maksimum iterasi hingga mencapai fungsi objektif yang konvergen adalah 100.
5. Uji Validitas menggunakan indeks Xie-Beni.

BAB II

TINJAUAN PUSTAKA

2.1. Standarisasi Data

Standarisasi data merupakan proses transformasi data menjadi *standardize* (normal baku) dengan rata-rata sama dengan 0 dan varian sama dengan 1 (Solimun, 2010). Menurut Mohammad dan Usman (2013), standarisasi data merupakan proses menstandarkan data mentah dengan mengubahnya menjadi rentang homogen (-3,5 sampai +3,5) menggunakan transformasi linier. Rumus transformasi *standardize* dapat dilihat seperti pada persamaan (2.1)

$$Z(x_{ij}) = \frac{x_{ij} - \bar{x}_j}{s_j} \quad (2.1)$$

Keterangan:

m : banyak variabel, $j = 1, 2, \dots, m$

n : banyak objek pengamatan, $i = 1, 2, \dots, n$

x_{ij} : objek data ke- i variabel ke- j

\bar{x}_j : rata-rata variabel ke- j

s_j : standar deviasi variabel ke- j

2.2. Analisis *Cluster*

Analisis *cluster* merupakan suatu analisis multivariat yang digunakan untuk mengelompokan objek pengamatan menjadi beberapa kelompok (*cluster*) berdasarkan kesamaan atau kemiripan karakteristik antar objek-objek yang diteliti (Johnson dan Wichern, 2007). Objek-objek dalam suatu *cluster* harus semirip mungkin dengan satu sama lain dan berbeda dengan objek-objek dalam *cluster* lain.

Pada dasarnya terdapat dua metode dalam pengelompokan, yaitu metode hierarki dan non hierarki. Prinsip utama metode hierarki adalah membentuk diagram pohon (dendogram) yang menggambarkan *cluster* berdasarkan nilai jarak, sehingga jumlah *cluster* tidak ditentukan pada awal proses analisis. Pada sisi lain, pengelompokan dengan metode non hierarki digunakan ketika jumlah *cluster* telah diketahui, sehingga dapat

dikatakan jumlah *cluster* pada metode ini ditentukan terlebih dahulu oleh peneliti (Handoyo dan Prasojo, 2017).

Menurut Supranto (2004), hasil pengelompokan yang ideal adalah setiap objek hanya masuk atau menjadi anggota dari salah satu *cluster*, tidak mungkin menjadi anggota dari dua *cluster* atau lebih (tidak terjadi tumpang tindih atau *overlapping*). Namun, dengan menggunakan algoritma *fuzzy* suatu objek pengamatan tidak mutlak menjadi anggota satu kelompok, tetapi bisa menjadi anggota beberapa kelompok dengan ukuran tingkat keanggotaan yang berbeda-beda (Handoyo dan Prasojo, 2017).

2.3. Konsep Jarak

Konsep dasar dari analisis *cluster* adalah mengelompokan objek berdasarkan karakteristik yang dimiliki. Menurut Han dan Kamber (2001), kesamaan dan ketidaksamaan antara dua objek dapat diukur dengan menggunakan fungsi jarak. Ukuran jarak menurut Han dan Kamber (2001) harus memenuhi syarat-syarat sebagai berikut.

- 1) $d(i, j) \geq 0$, jarak adalah sebuah angka non negatif. Tidak ada jarak yang memiliki nilai negatif.
- 2) $d(i, i) = 0$, jarak dari objek terhadap dirinya sendiri adalah nol.
- 3) $d(i, j) = d(j, i)$, jarak adalah sebuah fungsi simetrik. Jarak dari i ke j adalah sama dengan jarak dari j ke i .
- 4) $d(i, j) \leq d(i, h) + d(h, j)$, jika diatur dari objek i ke objek j dalam ruang sama, maka tidak lebih dari pembuatan cara lain pada objek h yang lain (*triangular inequality*).

Terdapat beberapa macam ukuran jarak yang dapat digunakan dalam analisis *cluster*. Salah satunya adalah jarak *euclidean*.

2.3.1. Jarak *Euclidean*

Salah satu ukuran kedekatan yang dapat digunakan adalah jarak *euclid*. Menurut Cox (2005), jarak *euclid* merupakan tipe pengukuran jarak yang paling umum dalam analisis *cluster*. Jarak *euclid* merupakan jarak geometris antar dua objek data. Semakin kecil nilai jarak, semakin dekat pula jarak kedua objek tersebut. Jarak *euclid* dapat dilihat seperti pada persamaan (2.2).

$$d_{ik} = \sqrt{\sum_{j=1}^m (x_{ij} - x_{kj})^2} \quad (2.2)$$

Keterangan:

m : banyak variabel, $j = 1, 2, \dots, m$

n : banyak objek pengamatan, $i, k = 1, 2, \dots, n$

d_{ik} : jarak *euclid* objek data ke- i dan objek data ke- k

x_{ij} : objek data ke- i variabel ke- j

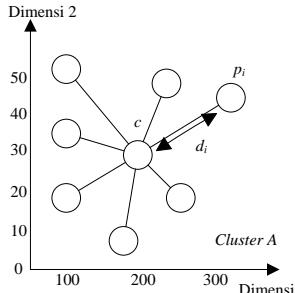
x_{kj} : objek data ke- k variabel ke- j

2.4. Analisis *Cluster* Nonhierarki

Analisis *cluster* nonhierarki adalah teknik pengelompokan objek penelitian ke dalam k *cluster*. Dengan demikian, jumlah *cluster* pada metode ini ditentukan terlebih dahulu oleh peneliti (Johnson dan Wichern, 2002). Terdapat dua sifat dasar metode pengelompokan nonhierarki yaitu metode *hard clustering* dan *fuzzy clustering*. Metode *hard clustering* adalah algoritma *k-means*, sedangkan metode *fuzzy clustering* adalah algoritma *fuzzy c-means* (Balasko dkk., 2008). Algoritma *k-means* bertujuan untuk mengelompokan objek pengamatan ke dalam suatu *cluster* tanpa ada *overlapping* (tumpang tindih), sedangkan algoritma *fuzzy c-means* dapat menangani adanya *overlapping* atau suatu objek penelitian berada dalam satu kelompok atau lebih.

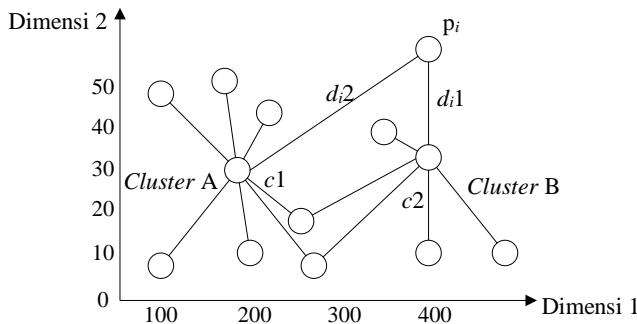
2.5. *Fuzzy Clustering*

Pengelompokan bisa diselesaikan secara *fuzzy* atau non *fuzzy*. Pada pengelompokan non *fuzzy* hasil matriks transformasinya berupa nilai derajat keanggotaan 0 dan 1. Gambar 2.1 menunjukkan bahwa pada pengelompokan non*fuzzy* suatu objek p_i hanya berada dalam satu *cluster* (*cluster A*) dengan jarak antara objek p_i dengan objek pusat c sebesar d_i .



Gambar 2.1. Pengelompokan Non fuzzy

Pada sisi lain, pengelompokan *fuzzy* hasil transformasinya berupa nilai derajat keanggotaan antara 0 dan 1. Gambar 2.2 menunjukkan bahwa pada pengelompokan *fuzzy* suatu objek p_i berada dalam dua *cluster* yaitu *cluster A* dan *cluster B* dengan jarak sebesar d_{i1} terhadap pusat (c_1) *cluster B* dan d_{i2} terhadap (c_2) *cluster A*. Hal ini merupakan keistimewaan pengelompokan menggunakan pengelompokan *fuzzy*.



Gambar 2.2. Pengelompokan Fuzzy

2.5.1. Fuzzy C-means (FCM)

Fuzzy C-means (FCM) merupakan suatu teknik pengelompokan data dimana keberadaaan setiap data dalam suatu *cluster* ditentukan oleh nilai keanggotaan. Menurut Kusumadewi (2002), pengelompokan dapat dikatakan *fuzzy* apabila tiap-tiap objek dihubungkan dengan menggunakan derajat keanggotaan. Derajat keanggotaan adalah ukuran seberapa kuat suatu objek menjadi bagian dalam *cluster* (Cox, 2005).

Menurut Handoyo dan Prasojo (2017), Konsep dasar *FCM* adalah menentukan pusat *cluster* yang akan menandai lokasi rata-rata untuk tiap-tiap *cluster*. Pada kondisi awal, pusat *cluster* masih belum akurat karena objek data memiliki derajat keanggotaan yang acak untuk tiap-tiap *cluster*. Kemudian pusat kelompok dan derajat keanggotaan objek data diiterasi sampai konvergen. Perulangan tersebut didasarkan pada meminimasi fungsi objektif. Dengan demikian, akan didapatkan pusat *cluster* yang bergerak ke tempat yang tepat.

Fungsi objektif pada iterasi ke-*t* yang digunakan pada *FCM* dapat dilihat pada persamaan (2.3).

$$P_t = \sum_{k=1}^n \sum_{i=1}^c \left((\mathbf{U})^w D(x_k, v_i)^2 \right) \quad (2.3)$$

dimana:

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nm} \end{bmatrix} \quad (2.4)$$

$$\mathbf{V} = \begin{bmatrix} v_{11} & v_{12} & \dots & v_{1m} \\ v_{21} & v_{22} & \dots & v_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ v_{c1} & v_{c2} & \dots & v_{cm} \end{bmatrix} \quad (2.5)$$

$$\mathbf{U} = \begin{bmatrix} \mu_{11} & \mu_{12} & \dots & \mu_{1n} \\ \mu_{21} & \mu_{22} & \dots & \mu_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{c1} & \mu_{c2} & \dots & \mu_{cn} \end{bmatrix} \quad (2.6)$$

Keterangan:

P_t : fungsi objektif *FCM* pada iterasi ke-*t*

c : banyak *cluster* yang digunakan

n	: banyak data yang akan dikelompokan
w	: parameter <i>fuzzification</i>
m	: banyak variabel
x_{kj}	: nilai objek ke- k pada variabel ke- j
v_{ij}	: nilai pusat <i>cluster</i> ke- i pada variabel ke- j
μ_{ik}	: nilai keanggotaan objek ke- k pada <i>cluster</i> ke- i
$D(x_k, v_i)$: jarak antara objek (x) ke- k dengan pusat <i>cluster</i> (v) ke- i

2.5.2. Algoritma *Fuzzy C-means*

Langkah-langkah algoritma *fuzzy C-mens* menurut Handoyo dan Prasojo (2017) adalah sebagai berikut:

- 1) Menginput data yang akan dikelompokan ke dalam matriks \mathbf{X} yang berukuran $n \times m$.
- 2) Menentukan jumlah *cluster* (c), parameter *fuzzification* (w), maksimum iterasi yang diinginkan, kriteria penghentian perulangan (ξ).
- 3) Membangkitkan bilangan acak μ_{0ik} sebagai elemen-elemen matriks partisi awal \mathbf{U} . Kemudian, menghitung jumlah setiap kolom pada matriks \mathbf{U} dengan menggunakan rumus (2.7).

$$Q_k = \sum_{i=1}^c \mu_{0ik} \quad (2.7)$$

- 4) Menghitung matriks partisi awal \mathbf{U} dengan menggunakan rumus (2.8).

$$\mu_{ik} = \frac{\mu_{0ik}}{Q_k} \quad (2.8)$$

- 5) Menghitung pusat *cluster* (v) ke- i berdasarkan rumus (2.9).

$$v_i = \frac{\sum_{k=1}^n (\mu_{ik})^w x_k}{\sum_{k=1}^n (\mu_{ik})^w} \quad (2.9)$$

Keterangan:

v_i : pusat *cluster* ke- i

μ_{ik} : nilai keanggotaan objek ke- k pada *cluster* ke- i

x_k : nilai objek ke- k

- 6) Menghitung fungsi objektif pada iterasi ke- t dengan menggunakan rumus pada persamaan (2.3). Fungsi objektif dikatakan konvergen apabila memenuhi persamaan (2.10).

$$|P_t - P_{t-1}| \leq \xi \quad (2.10)$$

Keterangan:

P_t : fungsi objektif *FCM* pada iterasi ke- t

P_{t-1} : fungsi objektif *FCM* pada iterasi ke- $(t-1)$

t : langkah iterasi

ξ : kesalahan terkecil yang diharapkan

- 7) Jika fungsi objektif belum konvergen, maka perlu merubah matriks partisi dengan menghitung derajat keanggotaan yang baru menggunakan rumus pada persamaan (2.11).

$$\mu_{ik} = \frac{\left[\sum_{j=1}^m (x_{kj} - v_{ij})^2 \right]^{-\frac{1}{w-1}}}{\sum_{i=1}^c \left[\sum_{j=1}^m (x_{kj} - v_{ij})^2 \right]^{-\frac{1}{w-1}}} \quad (2.11)$$

Keterangan:

μ_{ik} : nilai keanggotaan objek ke- k pada *cluster* ke- i

x_{kj} : nilai objek ke- k pada variabel ke- j

v_{ij} : pusat *cluster* ke- i pada variabel ke- j

Dengan syarat $\sum_{i=1}^c \mu_{ik} = 1$ yang berarti bahwa jumlah derajat

keanggotaan objek data pada setiap *cluster* sama dengan 1.

- 8) Jika nilai fungsi objektif telah konvergen sebagaimana memenuhi syarat pada persamaan (2.10), maka pengelompokan setiap objek ke dalam suatu *cluster* ditentukan berdasarkan nilai derajat keanggotaan terbesar.

2.6. Indeks Validitas *Cluster*

Indeks validitas adalah suatu ukuran yang digunakan untuk menentukan kelompok yang optimal. Suatu kelompok dikatakan optimal apabila memiliki keragaman yang kecil. Hal tersebut menunjukkan bahwa objek yang terdapat dalam suatu *cluster* lebih bersifat homogen (Balasko dkk., 2007).

Salah satu indeks validitas yang dapat digunakan adalah indeks Xie-Beni (Balasko dkk., 2008). Indeks Xie-Beni ditunjukkan pada persamaan (2.12).

$$XB = \frac{\sum_{i=1}^c \sum_{k=1}^n (\mu_{ik})^w (x_k - v_i)^2}{n \left(\min(x_k - v_i)^2 \right)} \quad (2.12)$$

Keterangan:

n : banyak data yang akan dikelompokan

μ_{ik} : nilai keanggotaan objek ke- k pada *cluster* ke- i

x_k : nilai objek ke- k pada variabel ke- j

v_i : pusat *cluster* ke- i pada variabel ke- j

Terdapat parameter *fuzzification* (w) dalam algoritma *FCM* yang dapat mengubah besaran pengaruh derajat keanggotaan. Menurut Bezdek (1981), nilai w yang digunakan bisa lebih dari 1. Persamaan indeks Xie-Beni tidak menetapkan nilai w yang digunakan. Oleh karena itu, indeks Xie-Beni dapat digunakan untuk menentukan nilai w yang mengoptimalkan suatu *cluster*. Pengelompokan yang optimal adalah yang memiliki nilai indeks Xie-Beni terkecil.

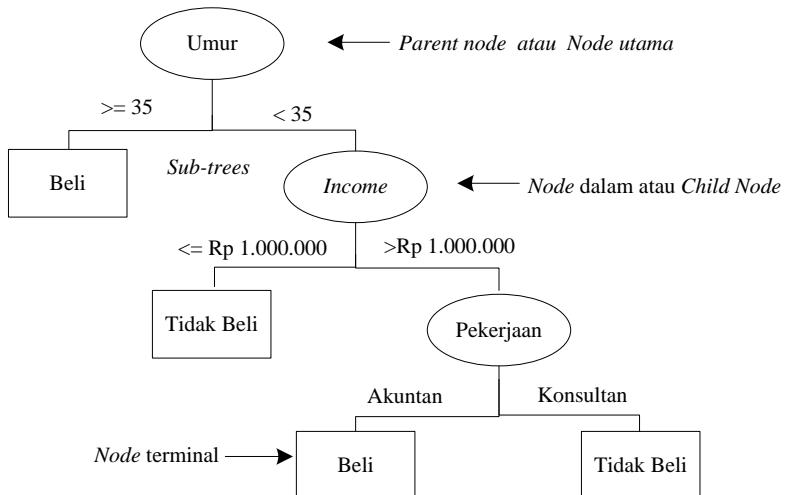
2.7. Decision Tree

Decision tree merupakan suatu pendekatan yang praktis dalam mengklasifikasikan objek ke dalam kelompok berdasarkan kriteria tertentu. Konsep dasar *decision tree* pada dasarnya adalah mengubah data menjadi pohon keputusan dan aturan hirarki (aturan-aturan keputusan). Algoritma dari *decision tree* secara otomatis akan menentukan peubah yang terpenting berdasarkan kemampuan peubah dalam menyortir data menjadi *output* kategori atau kelas yang benar. Menurut Santosa dan Umam (2018) pada umumnya, kasus-kasus yang

cocok ditangani dengan *decision tree* diantaranya adalah sebagai berikut:

- 1) Objek atau data dinyatakan dengan pasangan variabel dan nilainya. Variabel tersebut menyatakan kriteria dalam pembentukan pohon.
- 2) *Output* hasil klasifikasi bernilai diskrit. Biasanya berupa ya atau tidak. Contoh untuk penelitian ini adalah lancar membayar kredit atau tidak lancar membayar kredit.
- 3) Objek yang tidak memiliki variabel lengkap masih dapat diklasifikasikan dengan *decision tree* dengan baik.

Contoh model *decision tree* dapat dilihat pada Gambar 2.3. sebagai berikut.



Gambar 2.3. Contoh Decision Tree

Pada *decision tree* setiap variabel dinyatakan pada tiap simpul (*node*). Simpul utama dari suatu pohon klasifikasi disebut dengan *parent node*. Cabang-cabang kecil dari pohon yang dipisahkan oleh pemilah disebut dengan *sub-trees*. Simpul yang berhubungan langsung dengan *parent node* dan merupakan hasil pemilahan dari *parent node* disebut dengan *child node*. Simpul akhir dari suatu pohon disebut dengan *Node terminal*.

Salah satu metode atau algoritma *decision tree* adalah *Classification and Regression Trees* (*CART*). Metode *CART* dikembangkan oleh Leo Breiman, Jerome H. Friedman, Richard A.

Olshen, dan Charles J. Stone ini merupakan teknik klasifikasi dengan menggunakan algoritma penyekatan rekursif secara biner (*Binary Recursive Partitioning*). Metode *CART* akan dijelaskan lebih lanjut pada subbab berikutnya.

2.8. Classification and Regression Trees (CART)

Classification and Regression Trees (CART) merupakan salah satu jenis *decision tree*. Tujuan utama *CART* adalah untuk mendapatkan suatu kelompok data yang akurat sebagai penciri dari suatu pengklasifikasian. *CART* akan menghasilkan pohon klasifikasi jika variabel respon mempunyai skala kategorik dan akan menghasilkan pohon regresi jika variabel respon berupa data kontinu.

Teknik atau proses kerja dari *CART* dalam membuat sebuah pohon klasifikasi dikenal dengan istilah *Binary Recursive Partitioning*. Proses tersebut dikatakan *binary* karena setiap simpul data (*node*) akan selalu mengalami pemisahan kedalam *child node*. Sedangkan, maksut dari *recursive* adalah proses pemisahan tersebut diulang kembali pada setiap *child node* sebagai hasil pemisahan terdahulu, kemudian *child node* tersebut menjadi *parent node*. Proses pemisahan akan terus dilakukan sampai tidak ada lagi kesempatan untuk melakukan pemisahan berikutnya. Istilah *partitioning* berarti *testing sample* yang dimiliki dipisah kedalam bagian-bagian yang lebih kecil (Lewis, 2000).

Algoritma *CART* melalui tiga tahapan, yaitu pembentukan pohon klasifikasi, pemangkasan pohon klasifikasi dan penentuan pohon klasifikasi optimum. Langkah-langkah algoritma *CART* menurut Sumartini dan Purnami (2015) akan dijelaskan pada subbab selanjutnya.

2.8.1. Pembentukan Pohon Klasifikasi

Tahap pembentukan pohon klasifikasi diawali dengan menentukan variabel dan *threshold* untuk dijadikan pemilah tiap simpul. Tahapan pembentukan pohon klasifikasi terdiri dari:

1) Pemilihan Pemilah (*Classifier*)

Menurut Breiman dkk. (1993), setiap pemilah hanya bergantung pada nilai satu variabel prediktor. Variabel yang terpilih adalah yang menghasilkan simpul yang paling “*purest*” (paling bersih). Suatu cabang dikatakan *purest* apabila cabang tersebut hanya berasal dari satu kelas. Ukuran *purity* dinyatakan dengan tingkat *impurity*. Salah satu kriteria *impurity* adalah indeks Gini. Indeks Gini dapat digunakan

sebagai fungsi yang mengukur tingkat keragaman variabel dependen dalam metode *CART*. Fungsi Indeks Gini ditunjukkan pada persamaan (2.13)

$$i(t) = 1 - \sum_j p^2(j | t) \quad (2.13)$$

Keterangan:

$i(t)$: fungsi Indeks Gini pada simpul- t

$p(j|t)$: peluang unit pengamatan dalam kelas ke- j dari simpul- t

Pemilahan yang terpilih akan membentuk suatu himpunan kelas yang disebut simpul. Simpul tersebut akan melakukan pemilahan secara rekursif sampai diperoleh *terminal nodes*. Tahapan berikutnya adalah menentukan kriteria *Goodness of Split* untuk mengevaluasi pemilah dari pemilah s pada simpul t dengan menggunakan persaaan (2.14)

$$\phi(s,t) = \Delta i(s,t) = i(t) - p_L i(t_L) - p_R i(t_R) \quad (2.14)$$

Keterangan:

$\phi(s,t)$: kriteria nilai kebaikan pemilah- s pada simpul- t

$i(t)$: indeks Gini pada simpul- t

$i(t_L)$: indeks Gini untuk pada simpul anak kiri

$i(t_R)$: indeks Gini untuk pada simpul anak kanan

p_L : peluang pengamatan menuju simpul kiri

p_R : peluang pengamatan menuju simpul kanan

Pemilah yang terbaik adalah yang menghasilkan nilai $\phi(s,t)$ tertinggi karena mampu mereduksi heterogenitas yang lebih tinggi.

2) Penandaan Label Kelas

Penentuan label kelas pada simpul terminal berdasarkan proporsi kelas terbesar. Proporsi kelas terbesar menunjukkan bahwa kelas dominan. Fungsi peluang label kelas dapat dilihat seperti pada persamaan (2.15)

$$p(j_0, t) = \max_j p(j | t) = \max \frac{N_j(t)}{N(t)} \quad (2.15)$$

Keterangan:

$p(j|t)$: peluang unit pengamatan dalam kelas- j dari simpul- t

$N_j(t)$: banyak pengamatan kelas- j pada simpul- t

$N(t)$: banyak pengamatan pada simpul t

2.8.2. Pemangkasan Pohon Klasifikasi

Pohon yang dibentuk dengan aturan pemilah dan kriteria *Goodness of Split* berukuran sangat besar karena penghentian pohon berdasarkan banyaknya amatan pada simpul terminal atau besarnya tingkat kehomogenan. Ukuran pohon yang besar dapat dapat memunculkan adanya *overfitting*, akan tetapi jika pengamatan pohon dibatasi dengan ketepatan batas tertentu, maka dapat terjadi kasus *underfitting*. Ukuran pohon yang layak dapat dilakukan dengan pemangkasan pohon dengan ukuran *cost complexity minimum* pada persamaan (2.16)

$$R_\alpha(T) = R(T) + \alpha |\tilde{T}| \quad (2.16)$$

Pohon yang optimum akan membentuk kriteria seperti pada persamaan (2.17)

$$R_\alpha(T(\alpha)) = \min R_\alpha(T) \quad (2.17)$$

Keterangan:

$R_\alpha(T)$: biaya kompleksitas (*cost complexity*)

$R(T)$: kesalahan klasifikasi pohon pada pohon bagian T

α : parameter biaya kompleksitas (*complexity parameter*)

$|\tilde{T}|$: himpunan simpul terminal pada T

2.9. Kolektibilitas

Menurut Sinungan (2007), kolektibilitas dapat diartikan sebagai keadaan pembayaran kembali pokok, angsuran pokok atau bunga kredit oleh nasabah serta tingkat kemungkinan diterima kembali dana yang ditanamkan dalam surat berharga atau penanaman lainnya. Sedangkan tingkat kolektibilitas dapat dibedakan menjadi empat tingkat, yaitu apakah lancar, kurang lancar, diragukan, atau macet. Pembedaan tersebut dilakukan untuk mengantisipasi terjadinya suatu kerugian yang diakibatkan oleh adanya kredit yang tidak terbayarkan atau kredit bermasalah.

2.9.1. Penggolongan Kolektibilitas

Kolektibilitas kredit berdasarkan ketentuan yang dibuat Bank Indonesia adalah sebagai berikut:

- 1) Kredit lancar adalah kredit yang tidak mengalami penundaan pengembalian pokok pinjaman dan pembayaran bunga.
- 2) Kredit dalam perhatian khusus, apabila memenuhi kriteria sebagai berikut:
 - a) Terdapat tunggakan angsuran pokok dan atau bunga yang belum melampaui 90 hari.
 - b) Mutasi rekening relatif aktif.
 - c) Didukung oleh pelayanan baru.
- 3) Kredit kurang lancar adalah kredit yang pengembalian pokok pinjaman dan pembayaran bunganya telah mengalami penundaan selama 3 bulan dari waktu yang diperjanjikan.
- 4) Kredit diragukan adalah kredit yang pengembalian pokok pinjaman dan pembayaran bunganya telah mengalami penundaan selama 6 bulan atau dua kali dari jadwal yang telah diperjanjikan.
- 5) Kredit macet adalah kredit yang pengembalian pokok dan pembayaran bunganya telah mengalami penundaan lebih dari 1 tahun sejak jatuh tempo memuat jadwal yang telah diperjanjikan.

Adapun status kolektibilitas 1 sampai kolektibilitas 2 tergolong *Performing Loan (PL)*, sedangkan kolektibilitas 3 sampai kolektibilitas 5 tergolong *Nonperforming Loan (NPL)*.

2.10. Variabel Penelitian

2.10.1. Character (Karakter)

Prinsip pertama yang harus dipertimbangkan pihak bank dalam memberikan kredit kepada calon debitur adalah *character*. Menurut

Saraswati (2012), *character* merupakan informasi mengenai watak calon debitur. Bank perlu menjadikan prinsip *character* sebagai suatu hal yang penting karena *character* merupakan faktor kunci melihat watak atau sifat calon debitur dalam memenuhi kewajibannya, yaitu mampu menyelesaikan hutang. Menurut Fitria dan Sari (2012), *character* calon debitur dapat diperoleh melalui upaya meneniliti riwayat hidup calon debitur, meneliti reputasi calon debitur di lingkungan usahanya, meninjau status kolektibilitas, dan mencari informasi mengenai gaya hidup calon debitur.

2.10.2. Capacity (Kemampuan)

Capacity adalah kemampuan calon debitur dalam menjalankan usahanya guna memperoleh laba yang diharapkan. Prinsip ini berfungsi untuk mengukur kemampuan calon debitur dalam membayar dan mengembalikan hutang berserta dengan biaya bunganya secara tepat waktu. Pengukuran *capacity* calon debitur dapat dilakukan melalui pendekatan sebagai berikut (Fitri dan Sari, 2012).

- 1) Pendekatan historis, yaitu menilai kemampuan calon debitur dalam mengembangkan usahanya.
- 2) Pendekatan finansial, yaitu menilai latar belakang pendidikan calon debitur.
- 3) Pendekatan yuridis, yaitu menilai kapasitsa calon debitur untuk mewakili badan usaha miliknya yang digunakan untuk mengadakan perjanjian kredit dengan bank.
- 4) Pendekatan managerial, yaitu menilai kemampuan dan keterampilan calon debitur dalam melaksanakan fungsi-fungsi manajemen dalam memimpin perusahaan.
- 5) Pendekatan teknis, yaitu menilai kemampuan dan keterampilan calon debitur dalam mengelola tenaga kerja, sumber bahan baku, administrasi keuangan, dan faktor-faktor produksi lain.

2.10.3. Capital (Modal)

Capital merupakan sejumlah dana atau modal sendiri yang dimiliki oleh calon debitur. *Capital* merupakan prinsip yang diperlukan oleh bank sebagai indikator kesungguhan dan tanggung jawab calon debitur. Prinsip ini bertujuan untuk mengetahui kemampuan calon debitur dalam memenuhi keperluan biaya dan menanggung risiko kegagalan usaha (Saraswati, 2012).

Capital dapat dimanifestasikan dalam bentuk kewajiban calon debitur dalam menyediakan pembiayaan sendiri dalam praktik usahanya yang jumlahnya lebih besar daripada kredit yang diajukan. Bentuk pembiayaan tidak harus berupa uang tunai, tetapi dapat berupa barang modal seperti tanah, bangunan, mesin, dan sebagainya. Dalam menganalisis *capital*, pihak bank dapat melihat modal usaha yang dimiliki calon debitur sebelum kredit diberikan. Hal ini dapat dilihat dari laporan keuangan atau proposal yang dibuat oleh calon debitur.

2.10.4. *Collateral* (Jaminan)

Collateral merupakan barang-barang yang diserahkan oleh calon debitur kepada pihak bank sebagai jaminan atau agunan terhadap kredit yang diterimanya. Prinsip ini bertujuan untuk menghindari terjadinya pemalsuan bukti kepemilikan. *Collateral* dapat berupa nonmaterial seperti jaminan pribadi, *letter of guarantee*, *letter of comfort*, dan lain sebagainya yang nilainya tidak kurang dari jumlah kredit yang diberikan kepada calon debitur.

2.10.5. *Condition of Economy* (Kondisi Ekonomi)

Condition of economy menggambarkan situasi dan kondisi ekonomi, sosial, politik, budaya yang memengaruhi perekonomian dan kelancaran kegiatan usaha calon debitur. Prinsip *condition of economy* meneliti hal-hal seperti keadaan konjungtur, peraturan pemerintah, situasi politik, dan perekonomian politik guna mendapat gambaran mengenai kondisi ekonomi saat ini.

BAB III

METODE PENELITIAN

3.1. Sumber Data

Data yang digunakan dalam penelitian ini adalah data sekunder. Data pada penelitian ini adalah data calon debitur di 18 kota di Jawa Timur yang dimiliki pihak bank hingga tahun 2018. 18 kota tersebut adalah Banyuwangi, Batu, Blitar, Gresik, Jember, Jombang, Kediri, Lamongan, Madiun, Magetan, Malang, Mojokerto, Nganjuk, Ngawi, Pacitan, Pasuruan, Ponorogo dan Surabaya. Struktur data pada penelitian ini dapat dilihat pada Tabel 3.1.

Tabel 3.1. Struktur Data Penelitian

Kota	Nasabah	Variabel			
		X₁	X₂	...	X_m
Banyuwangi	1	X ₁₁	X ₁₂	...	X _{1m}
	2	X ₂₁	X ₂₂	...	X _{2m}
	⋮	⋮	⋮	⋮	⋮
	n	X _{n1}	X _{n2}	...	X _{nm}
Batu	1	X ₁₁	X ₁₂	...	X _{1m}
	2	X ₂₁	X ₂₂	...	X _{2m}
	⋮	⋮	⋮	⋮	⋮
	n	X _{n1}	X _{n2}	...	X _{nm}
⋮	⋮	⋮	⋮	⋮	⋮
Surabaya	1	X ₁₁	X ₁₂	...	X _{1m}
	2	X ₂₁	X ₂₂	...	X _{2m}
	⋮	⋮	⋮	⋮	⋮
	n	X _{n1}	X _{n2}	...	X _{nm}

Keterangan:

n : banyak objek pengamatan

m : banyak variabel

3.2. Variabel Penelitian

Variabel penelitian yang digunakan dalam penelitian ini diambil dari variabel pada prinsip penyekoran kredit (*credit score model*), yaitu variabel 5C (*Character, Capacity, Capital, Collateral* dan *Condition of Economy*). Variabel 5C tersebut menjadi bahan pertimbangan bank dalam memberikan kredit kepada calon debitur. Definisi dari masing-masing variabel dijelaskan dalam subbab 2.9. Variabel penelitian dapat dilihat pada Tabel 3.2.

Tabel 3.2. Variabel Penelitian

Variabel Penelitian	Indikator
<i>Character</i> (X1)	Pendidikan (X _{1.1})
	Usia (X _{1.2})
	Status Kolektibilitas SID (X _{1.3})
<i>Capacity</i> (X2)	<i>Joint Income</i> (X _{2.1})
	Bentuk Badan Usaha (X _{2.2})
	Jangka Waktu Kredit (X _{2.3})
	Rasio Penghasilan Angsuran (X _{2.4})
	Pekerjaan (X _{2.5})
	Pengalaman Kerja (X _{2.6})
	Jumlah Tanggungan Keluarga (X _{2.7})
<i>Capital</i> (X3)	<i>Loan to Value</i> (X _{3.1})
<i>Collateral</i> (X4)	Dokumen Jaminan (X _{4.1})
	Lama Tempat Tinggal (X _{4.2})
<i>Condition of Economy</i> (X5)	Indikator Kepemilikan Tabungan (X _{5.1})

Masing-masing indikator pada tiap variabel memiliki banyak item yang berbeda, sehingga perlu transformasi standarisasi data untuk mendapatkan nilai suatu variabel. Proses transformasi standarisasi data dapat menggunakan rumus pada persamaan (2.1). Nilai suatu variabel didapatkan dengan cara menghitung rata-rata indikator yang sudah di standarisasi.

3.3. Penentuan Parameter dalam Algoritma *Fuzzy C-means (FCM)*

Terdapat beberapa parameter yang perlu ditentukan terlebih dahulu oleh peneliti. Parameter yang dimaksud adalah jumlah *cluster* (c), kriteria penghentian (ξ), parameter *fuzzification* (w) dan maksimum iterasi yang diinginkan. Parameter yang digunakan dalam penelitian ini dapat dilihat pada Tabel 3.3.

Tabel 3.3. Parameter Dalam Algoritma *FCM*

Parameter	Nilai
Jumlah <i>cluster</i> (c)	2; 3 dan 4
Kriteria Penghentian (ξ)	0,001
Maksimum iterasi	100
Parameter <i>fuzzification</i> (w)	1,5; 2 dan 2,5

Berdasarkan Tabel 3.3, penelitian ini menggunakan sembilan kombinasi nilai parameter dalam algoritma *FCM*. Sembilan kombinasi tersebut berasal dari kombinasi penggunaan masing-masing parameter dalam satu algoritma *FCM* seperti pada Tabel 3.4.

Tabel 3.4. Kombinasi Parameter Dalam Algoritma *FCM*

Kombinasi Pengelompokan	w	c	Kriteria Penghentian	Maks. Iterasi
1	1,5	2	0,001	100
2		3		
3		4		
4	2	2	0,001	100
5		3		
6		4		

Tabel 3.4. Lanjutan

Kombinasi Pengelompokan	<i>w</i>	<i>c</i>	Kriteria Penghentian	Maks. Iterasi
7	2,5	2	0,001	100
8		3		
9		4		

Jumlah *cluster* 2; 3 dan 4 yang digunakan dalam penelitian ini dimaksud untuk menemukan *cluster* yang terbaik. Bezdek (1981) menyatakan bahwa parameter *fuzzification* dapat bernilai lebih dari satu sehingga parameter *fuzzification* yang digunakan dalam penelitian ini adalah sebesar 1,5; 2 dan 2,5. Interval tersebut digunakan dengan maksud untuk menentukan parameter *fuzzification* yang paling optimal. Kriteria penghentian yang digunakan adalah sebesar 0,001 yang didasarkan dari hasil penelitian Alata dkk (2013). Maksimum iterasi yang digunakan adalah sebesar 100 yang didasarkan dari hasil penelitian Fan dkk (2009).

Pengelompokan dengan kombinasi parameter yang optimal akan ditentukan dengan indeks Xie-Beni yang terkecil. Indeks Xie-Beni dapat dilihat pada persamaan (2.24).

3.4. Penggolongan Kolektibilitas

Penggolongan kolektibilitas telah dijelaskan pada subbab 2.8.1. Calon debitur dalam penelitian ini akan diklasifikasikan dengan menggunakan metode *Classification and Regression Trees* ke dalam dua kelas, yaitu kelas *Performing Loan* (kredit lancar) dan *Nonperforming Loan* (kredit macet). Notasi untuk kelas kredit lancar adalah 1 dan untuk kelas kredit macet adalah 2.

3.5. Metode Penelitian

Software yang digunakan untuk menjalankan algoritma dalam penelitian ini adalah software *R* dan *Rstudio*. Algoritma pertama yang dilakukan adalah Algoritma *Fuzzy C-means*. Langkah-langkah yang digunakan dalam Algoritma *FCM* adalah sebagai berikut:

1. Menginput data.
2. Menginput banyak *cluster* = 2; 3 dan 4.
3. Menentukan derajat *fuzzy w*. Dalam penelitian ini, derajat *fuzzy* yang digunakan adalah $w = 1,5; w = 2$ dan $w = 2,5$.

4. Membangkitkan bilangan acak μ_{0ik} sebagai elemen-elemen matriks partisi awal \mathbf{U} . Kemudian, menghitung jumlah setiap kolom pada matriks \mathbf{U} dengan menggunakan rumus pada persamaan (2.19).
5. Menghitung matriks partisi awal \mathbf{U} dengan menggunakan rumus pada persamaan (2.20).
6. Menghitung pusat *cluster* ke- i (v_i) berdasarkan rumus pada persamaan (2.21).
7. Menghitung fungsi objektif pada iterasi ke- t dengan menggunakan rumus pada persamaan (2.15).
8. Menghitung perubahan matriks partisi dengan derajat keanggotaan baru menggunakan rumus pada persamaan (2.23).
9. Memeriksa kondisi fungsi objektif yang harus memenuhi kriteria seperti pada persamaan (2.22).
10. Melakukan pengelompokan objek berdasarkan nilai derajat keanggotaan terbesar berdasarkan persamaan (2.15).
11. Menentukan koefisien parameter *fuzzification* (w) yang mengoptimalkan *cluster* dengan indeks Xie-Beni pada persamaan (2.23).
12. Mendapatkan *cluster* optimal yang memiliki nilai indeks Xie-Beni terkecil
13. Menginterpretasi hasil *cluster*.
14. Selesai.

Hasil dari Algoritma *FCM* berupa sejumlah c -*cluster* kota-kota calon debitur bank yang dikelompokkan berdasarkan prinsip penyekoran kredit 5C. Kemudian, calon debitur pada tiap *cluster* diklasifikasikan ke dalam kelas kredit lancar atau kelas kredit macet. Metode klasifikasi yang digunakan adalah *Classification and Regression Trees*.

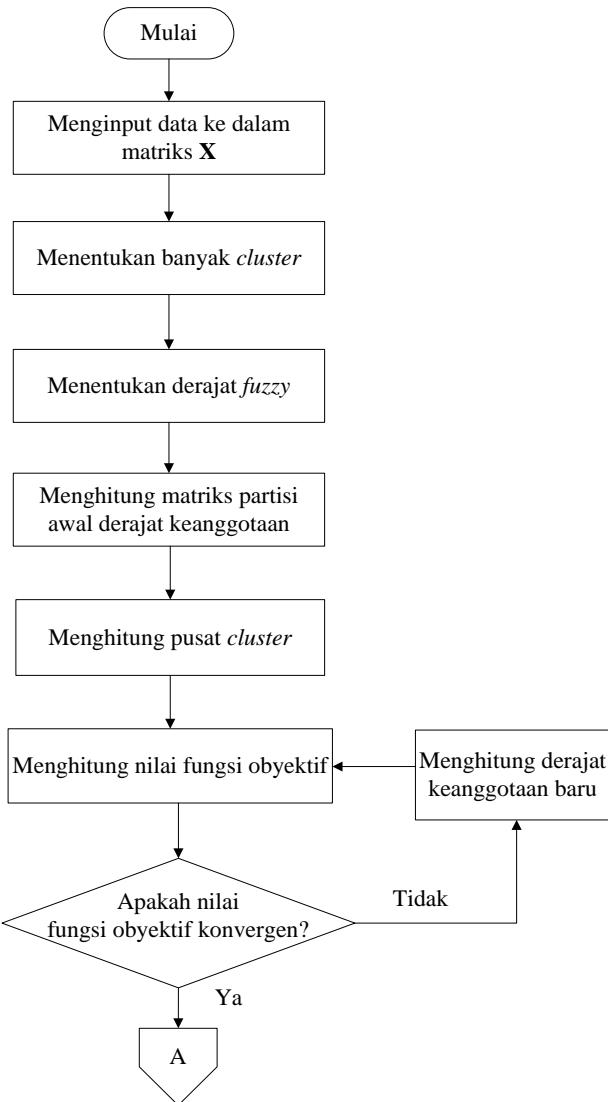
Langkah-langkah yang digunakan dalam Algoritma *Classification and Regression Trees* (*CART*) pada data tiap *cluster* yang terbentuk adalah sebagai berikut:

1. Membagi data menjadi data *training* (data latih) dan data *testing* (data uji). Data dibagi dengan perbandingan 75%:15%.
2. Menentukan variabel sebagai pemilah terbaik dengan menggunakan kriteria *impurity* Indeks Gini pada persamaan (2.13).

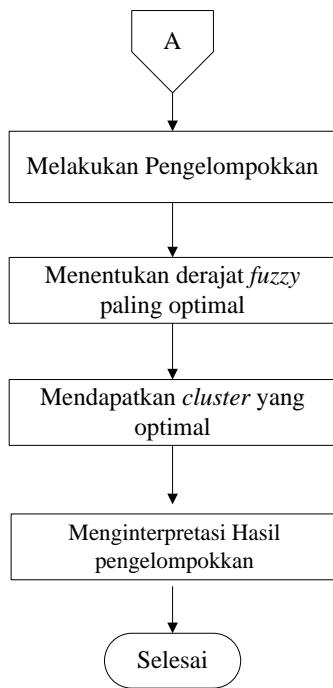
3. Menentukan pemilah terbaik yang dapat dihitung menggunakan *Godness of Split* melalui persamaan (2.14).
4. Pemberian label kelas menggunakan persamaan (2.15).
5. Pembentukan pohon klasifikasi maksimal.
6. Pemangkasan pohon klasifikasi maksimal dengan menggunakan persamaan (2.16).
7. Menentukan pohon klasifikasi optimal dengan menggunakan persamaan (2.17).
8. Interpretasi pohon klasifikasi *CART*.
9. Selesai.

3.6. Diagram Alir

Skema langkah-langkah yang digunakan dalam Algoritma Fuzzy *C-means* dapat dilihat pada Gambar 3.1.

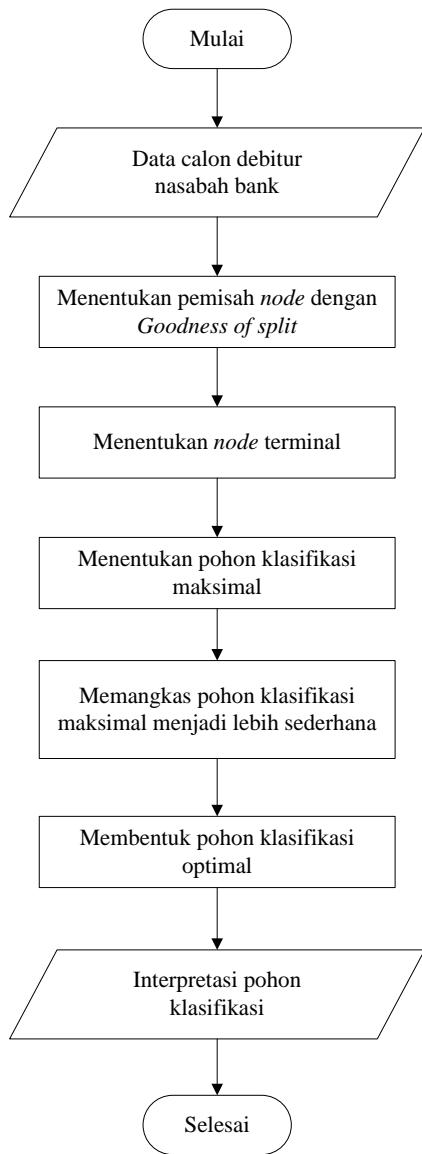


Gambar 3.1. Diagram Alir Penelitian Algoritma Fuzzy *C-means*



Gambar 3.1. (Lanjutan)

Skema langkah-langkah yang digunakan dalam Algoritma *Classification and Regression Trees* (*CART*) dapat dilihat pada Gambar 3.2.

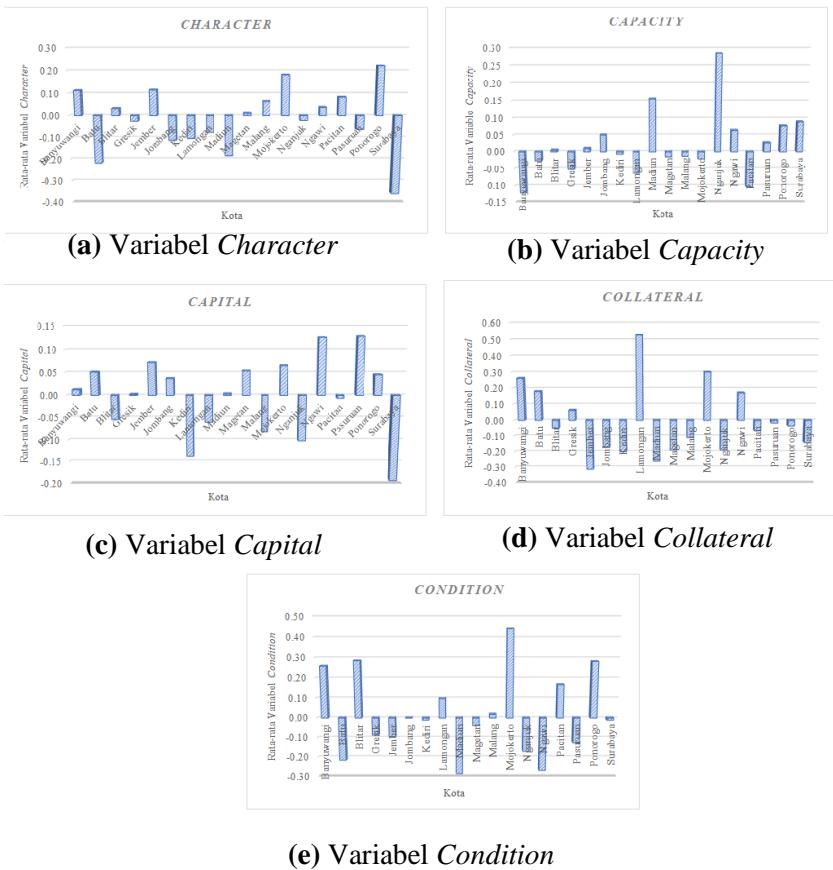


Gambar 3.2. Diagram Alir Algoritma *CART*

BAB IV

HASIL DAN PEMBAHASAN

4.1. Analisis Deskriptif



Gambar 4.1. Analisis Deskriptif Variabel 5C

Analisis deskriptif pada penelitian ini dilakukan pada data calon debitur bank di 18 daerah di Jawa Timur yang sudah distandarisasi, sehingga rata-rata tiap variabel adalah 0 dan memiliki varian sebesar 1. Rentang masing-masing variabel adalah -3,5 sampai +3,5. Jika variabel bernilai positif, maka nilai variabel tersebut berada di atas rata-rata,

sebaliknya jika variabel bernilai negatif, maka nilai variabel tersebut berada di bawah rata-rata. Analisis deskriptif berfungsi untuk mengetahui gambaran umum mengenai calon debitur pada 18 kota di Jawa Timur. Variabel penelitian yang digunakan adalah variabel *5C* (*Character, Capacity, Capital, Collateral* dan *Condition of Economy*) pada prinsip penyekoran kredit yang digunakan pihak bank dalam pemberian kredit kepada calon debitur.

Analisis deskriptif mengenai variabel *5C* (*Character, Capacity, Capital, Collateral* dan *Condition of Economy*) calon debitur bank di 18 kota di Jawa Timur tersaji pada Gambar 4.1. Kota dengan nilai rata-rata variabel tertinggi dan terendah pada Gambar 4.1. diringkas dalam Tabel 4.1.

Tabel 4.1. Nilai Rata-Rata Tertinggi dan Terendah Tiap Variabel

Variabel	Kesimpulan
<i>Character</i>	<ol style="list-style-type: none"> Nilai rata-rata tertinggi terdapat pada Kota Ponorogo. Nilai rata-rata terendah terdapat pada Kota Surabaya.
<i>Capacity</i>	<ol style="list-style-type: none"> Nilai rata-rata tertinggi terdapat pada Kota Nganjuk. Nilai rata-rata terendah terdapat pada Kota Banyuwangi.
<i>Capital</i>	<ol style="list-style-type: none"> Nilai rata-rata tertinggi terdapat pada Kota Pasuruan. Nilai rata-rata terendah terdapat pada Kota Surabaya.
<i>Collateral</i>	<ol style="list-style-type: none"> Nilai rata-rata tertinggi terdapat pada Kota Lamongan. Nilai rata-rata terendah terdapat pada Kota Jember.
<i>Condition of Economy</i>	<ol style="list-style-type: none"> Nilai rata-rata tertinggi terdapat pada Kota Mojokerto. Nilai rata-rata terendah terdapat pada Kota Madiun.

Analisis deskriptif variabel *Character* di 18 kota di Jawa Timur pada Gambar 4.1. (a) menunjukkan bahwa terdapat sembilan kota, yaitu

Batu, Gresik, Jombang, Kediri, Lamongan, Madiun, Nganjuk, Pasuruan dan Surabaya, yang memiliki nilai variabel *Character* di bawah rata-rata (bernilai negatif). Sembilan kota lainnya, yaitu Banyuwangi, Blitar, Jember, Magetan, Malang, Mojokerto, Ngawi, Pacitan dan Ponorogo, yang memiliki nilai variabel *Character* di atas rata-rata (bernilai positif).

Analisis deskriptif variabel *Capacity* di 18 kota di Jawa Timur pada Gambar 4.1. (b) menunjukkan bahwa terdapat sembilan kota, yaitu Banyuwangi, Batu, Gresik, Kediri, Lamongan, Magetan, Malang, Mojokerto, dan Pacitan, yang memiliki nilai variabel *Capacity* di bawah rata-rata (bernilai negatif). Sembilan kota lainnya, yaitu Blitar, Jember, Jombang, Madiun, Nganjuk, Ngawi, Pasuruan, Ponorogo dan Surabaya, memiliki nilai variabel *Capacity* di atas rata-rata (bernilai positif).

Analisis deskriptif variabel *Capital* di 18 di Jawa Timur pada Gambar 4.1. (c) menunjukkan bahwa terdapat tujuh kota, yaitu Blitar, Kediri, Lamongan, Malang, Nganjuk, Pacitan dan Surabaya, yang memiliki nilai variabel *Capital* di bawah rata-rata (bernilai negatif). Sedangkan, 11 kota lainnya, yaitu Banyuwangi, Batu, Gresik, Jember, Jombang, Madiun, Magetan, Mojokerto, Ngawi, Pasuruan, dan Ponorogo memiliki nilai variabel *Capital* di atas rata-rata (bernilai positif)

Analisis deskriptif variabel *Collateral* di 18 kota di Jawa Timur pada Gambar 4.1. (d) menunjukkan bahwa terdapat 12 kota, yaitu Blitar, Jember, Jombang, Kediri, Madiun, Magetan, Malang, Nganjuk, Pacitan, Pasuruan, Ponorogo dan Surabaya, yang memiliki nilai variabel *Collateral* di bawah rata-rata (bernilai negatif). Enam kota lainnya, yaitu Banyuwangi, Batu, Gresik, Lamongan, Mojokerto dan Ngawi memiliki nilai variabel *Collateral* di atas rata-rata (bernilai positif).

Analisis deskriptif variabel *Condition of Economy* di 18 kota di Jawa Timur pada Gambar 4.1. (e) menunjukkan bahwa terdapat 12 kota, yaitu Batu, Gresik, Jember, Jombang, Kediri, Madiun, Magetan, Malang, Nganjuk, Ngawi, Pasuruan dan Surabaya, yang memiliki nilai variabel *Condition of Economy* di bawah rata-rata (bernilai negatif). Enam kota lainnya, yaitu Banyuwangi, Blitar, Lamongan, Mojokerto, Pacitan dan Ponorogo memiliki nilai variabel *Condition of Economy* di atas rata-rata (bernilai positif).

4.2. Hasil Algoritma *Fuzzy C-means*

Fuzzy C-means (FCM) pada penelitian ini diterapkan untuk pengelompokan calon debitur di 18 kota di Jawa Timur berdasarkan prinsip penyekoran kredit 5C (*Character, Capacity, Capital, Collateral* dan *Condition of Economy*). Algoritma *FCM* dalam penelitian ini menggunakan kombinasi nilai parameter yang telah dijelaskan pada subbab 3.3. Jumlah *cluster* yang digunakan adalah sebesar 2; 3 dan 4. Parameter *fuzzification* yang digunakan dalam penelitian ini adalah sebesar 1,5; 2 dan 2,5. Interval tersebut digunakan dengan maksud untuk menentukan parameter *fuzzification* yang paling optimal. Kriteria penghentian yang digunakan adalah sebesar 0,001. Maksimum iterasi yang digunakan adalah sebesar 100. Oleh karena itu, terdapat sembilan kombinasi pengelompokan calon debitur berdasarkan penggunaan nilai parameter algoritma *FCM* yang berbeda. Algoritma *FCM* dijalankan dengan bantuan *Software R 3.5.1*. Hasil pengelompokan berdasarkan sembilan kombinasi nilai parameter yang digunakan dalam algoritma *FCM* akan dibahas dalam subbab selanjutnya.

4.2.1. Hasil Pengelompokan Kombinasi Ke-1

Nilai-nilai parameter yang digunakan dalam algoritma *FCM* pada pengelompokan kombinasi ke-1 dapat dilihat pada Tabel 4.2.

Tabel 4.2. Nilai Parameter Pada Pengelompokan Kombinasi Ke-1

Parameter	Nilai
Jumlah <i>cluster</i> (c)	2
Kriteria Penghentian (ξ)	0,001
Maksimum iterasi	100
Parameter <i>fuzzification</i> (w)	1,5

Hasil analisis algoritma *FCM* dengan bantuan *Software R 3.5.1*. dapat dilihat pada Lampiran 3. Anggota tiap *cluster* dapat dilihat pada Tabel 4.3.

Tabel 4.3. Anggota Tiap *Cluster* Pada Pengelompokan Kombinasi Ke-1

<i>Cluster</i>	Anggota <i>Cluster</i>	Banyak Anggota
1	Banyuwangi, Blitar, Lamongan, Mojokerto Pacitan, dan Ponorogo	$n = 6$
2	Batu, Gresik, Jember, Jombang, Kediri, Madiun, Magetan, Malang, Nganjuk, Ngawi Pasuruan dan Surabaya	$n = 12$

Setiap *cluster* memiliki karakteristik calon debitur yang berbeda. Karakteristik calon debitur dari setiap *cluster* dapat dilihat melalui nilai masing-masing variabel. Nilai variabel 5C disetiap *cluster* dapat dilihat pada Tabel 4.4.

Tabel 4.4. Nilai Variabel 5C Tiap *Cluster*

<i>Cluster</i>	<i>Character</i>	<i>Capacity</i>	<i>Capital</i>	<i>Collateral</i>	<i>Condition</i>
1	0,0918	-0,0378	0,0000	0,1564	0,2549
2	-0,0719	0,0476	-0,0037	-0,0974	-0,1070

Berdasarkan Tabel 4.4. dapat disimpulkan bahwa *cluster* 1 merupakan *cluster* dengan calon debitur yang memiliki nilai variabel *Character*, *Capital*, *Collateral* dan *Condition* di atas rata-rata, sedangkan *cluster* 2 merupakan *cluster* yang memiliki nilai variabel *Capacity* di atas rata-rata. Dengan demikian, pihak bank dapat membentuk kebijakan-kebijakan yang menyesuaikan kondisi tiap *cluster* untuk mengurangi resiko kegagalan pembayaran kredit oleh calon debitur.

4.2.2. Hasil Pengelompokan Kombinasi Ke-2

Nilai-nilai parameter yang digunakan dalam algoritma *FCM* pada pengelompokan kombinasi ke-2 dapat dilihat pada Tabel 4.5.

Tabel 4.5. Nilai Parameter pada Pengelompokan Kombinasi Ke-2

Parameter	Nilai
Jumlah <i>cluster</i> (c)	3
Kriteria Penghentian (ξ)	0,001

Tabel 4.5. Lanjutan

Parameter	Nilai
Maksimum iterasi	100
Parameter <i>fuzzification</i> (<i>w</i>)	1,5

Hasil analisis algoritma *FCM* dengan bantuan *Software R 3.5.1*. dapat dilihat pada Lampiran 4. Anggota tiap *cluster* dapat dilihat pada Tabel 4.6.

Tabel 4.6. Anggota Tiap *Cluster* Pada Pengelompokan Kombinasi Ke- 2

Cluster	Anggota Cluster	Banyak Anggota
1	Banyuwangi, Blitar, Lamongan, Mojokerto Pacitan dan Ponorogo	$n = 6$
2	Batu, Jember, Magetan, Ngawi dan Pasuruan	$n = 5$
3	Gresik, Jombang, Kediri, Madiun, Malang Nganjuk dan Surabaya	$n = 7$

Setiap *cluster* memiliki karakteristik calon debitir yang berbeda. Karakteristik calon debitir dari setiap *cluster* dapat dilihat melalui nilai masing-masing variabel. Nilai variabel 5C disetiap *cluster* dapat dilihat pada Tabel 4.7.

Tabel 4.7. Nilai Rata-rata Tiap *Cluster*

Cluster	Character	Capacity	Capital	Collateral	Condition
1	0,0918	-0,0378	0,0000	0,1564	0,2549
2	-0,0243	0,0122	0,0857	-0,0329	-0,1482
3	-0,1059	0,0729	-0,0675	-0,1434	-0,0776

Berdasarkan Tabel 4.7. dapat disimpulkan bahwa *cluster* 1 merupakan *cluster* dengan calon debitir yang memiliki nilai variabel *Character*, *Capital*, *Collateral* dan *Condition* di atas rata-rata. *Cluster* 2 merupakan *cluster* dengan calon debitir yang memiliki nilai variabel *Capacity* dan *Capital* di atas rata-rata, sedangkan *cluster* 3 merupakan *cluster* dengan nilai variabel *Capacity* di atas rata-rata. Dengan demikian, pihak bank dapat membentuk kebijakan-kebijakan yang

menyesuaikan kondisi tiap *cluster* untuk mengurangi resiko kegagalan pembayaran kredit oleh calon debitur.

4.2.3. Hasil Pengelompokan Kombinasi Ke-3

Nilai-nilai parameter yang digunakan dalam algoritma *FCM* pada pengelompokan kombinasi ke-3 dapat dilihat pada Tabel 4.8.

Tabel 4.8. Nilai Parameter pada Pengelompokan Kombinasi Ke-3

Parameter	Nilai
Jumlah <i>cluster</i> (c)	4
Kriteria Penghentian (ξ)	0,001
Maksimum iterasi	100
Parameter <i>fuzzification</i> (w)	1,5

Hasil analisis algoritma *FCM* dengan bantuan *Software R 3.5.1*. dapat dilihat pada Lampiran 5. Anggota tiap *cluster* dapat dilihat pada Tabel 4.9.

Tabel 4.9. Anggota Tiap *Cluster* Pada Pengelompokan Kombinasi Ke-3

Cluster	Anggota <i>Cluster</i>	Banyak Anggota
1	Pacitan, Batu, Gresik, Ngawi, Blitar, Malang, Lamongan dan Jember	$n = 8$
2	Kediri, Ponorogo, Mojokerto dan Pasuruan	$n = 4$
3	Nganjuk, Jombang dan Surbaya	$n = 3$
4	Banyuwangi, Madiun dan Magetan	$n = 3$

Setiap *cluster* memiliki karakteristik calon debitur yang berbeda. Karakteristik calon debitur dari setiap *cluster* dapat dilihat melalui nilai masing-masing variabel. Nilai variabel 5C disetiap *cluster* dapat dilihat dalam Tabel 4.10.

Tabel 4.10. Nilai Rata-rata Tiap *Cluster*

Cluster	Character	Capacity	Capital	Collateral	Condition
1	0,0725	-0,0683	0,0054	0,3630	0,2656
2	-0,0243	0,0122	0,0857	-0,0329	-0,1482

Tabel 4.10. Lanjutan

<i>Cluster</i>	<i>Character</i>	<i>Capacity</i>	<i>Capital</i>	<i>Collateral</i>	<i>Condition</i>
3	0,0170	0,0304	-0,0373	-0,0943	0,0605
4	-0,2720	0,1225	-0,0950	-0,1999	-0,1475

Berdasarkan Tabel 4.10, dapat disimpulkan bahwa *cluster* 1 merupakan *cluster* dengan calon debitur yang memiliki nilai variabel *Character*, *Capital*, *Collateral* dan *Condition* di atas rata-rata. *Cluster* 2 merupakan *cluster* dengan calon debitur yang memiliki nilai variabel *Capacity* dan *Capital* di atas rata-rata. *Cluster* 3 merupakan *cluster* dengan calon debitur yang memiliki nilai variabel *Capacity* dan *Condition of Economy* di atas rata-rata, sedangkan *cluster* 4 merupakan *cluster* yang memiliki nilai variabel *Capacity* di atas rata-rata. Dengan demikian, pihak bank dapat membentuk kebijakan-kebijakan yang menyesuaikan kondisi tiap *cluster* untuk mengurangi resiko kegagalan pembayaran kredit oleh calon debitur.

4.2.4. Hasil Pengelompokan Kombinasi Ke-4

Nilai-nilai parameter yang digunakan dalam algoritma *FCM* pada pengelompokan kombinasi ke-4 dapat dilihat pada Tabel 4.11.

Tabel 4.11. Nilai Parameter pada Pengelompokan Kombinasi Ke-4

Parameter	Nilai
Jumlah <i>cluster</i> (<i>c</i>)	2
Kriteria Penghentian (ξ)	0,001
Maksimum iterasi	100
Parameter <i>fuzzification</i> (<i>w</i>)	2

Hasil analisis algoritma *FCM* dengan bantuan *Software R 3.5.1*. dapat dilihat pada Lampiran 6. Anggota tiap *cluster* dapat dilihat pada Tabel 4.12.

Tabel 4.12. Anggota Tiap *Cluster* Pada Pengelompokan Kombinasi Ke-4

Cluster	Anggota Cluster	Banyak Anggota
1	Batu, Gresik, Jember, Jombang, Madiun, Magetan, Malang, Nganjuk, Ngawi, Pasuruan dan Surabaya	$n = 11$
2	Banyuwangi, Blitar, Kediri, Lamongan, Mojokerto, Pacitan dan Ponorogo	$n = 7$

Setiap *cluster* memiliki karakteristik calon debitur yang berbeda. Karakteristik calon debitur dari setiap *cluster* dapat dilihat melalui nilai masing-masing variabel. Nilai variabel 5C disetiap *cluster* disajikan dalam Tabel 4.13.

Tabel 4.13. Nilai Rata-rata Tiap *Cluster*

Cluster	Character	Capacity	Capital	Collateral	Condition
1	-0,0690	0,0525	0,0086	-0,0882	-0,1158
2	0,0639	-0,0333	-0,0197	0,1057	0,2169

Berdasarkan Tabel 4.13, dapat disimpulkan bahwa *cluster* 1 merupakan *cluster* dengan calon debitur yang memiliki nilai variabel *Capacity* dan *Capital* di atas rata-rata, sedangkan *cluster* 2 merupakan *cluster* yang memiliki nilai variabel *Character* dan *Condition of Economy* di atas rata-rata. Dengan demikian, pihak bank dapat membentuk kebijakan-kebijakan yang menyesuaikan kondisi tiap *cluster* untuk mengurangi resiko kegagalan pembayaran kredit oleh calon debitur.

4.2.5. Hasil Pengelompokan Kombinasi Ke-5

Nilai-nilai parameter yang digunakan dalam algoritma *FCM* pada pengelompokan kombinasi ke-5 dapat dilihat pada Tabel 4.14.

Tabel 4.14. Nilai Parameter pada Pengelompokan Kombinasi ke-5

Parameter	Nilai
Jumlah <i>cluster</i> (c)	3
Kriteria Penghentian (ξ)	0,001
Maksimum iterasi	100
Parameter <i>fuzzification</i> (w)	2

Hasil analisis algoritma *FCM* dengan bantuan *Software R 3.5.1*. dapat dilihat pada Lampiran 7. Anggota tiap *cluster* dapat dilihat pada Tabel 4.15.

Tabel 4.15. Anggota Tiap *Cluster* Pada Pengelompokan Kombinasi Ke-5

Cluster	Anggota <i>Cluster</i>	Banyak Anggota
1	Batu, Jember, Magetan, Ngawi dan Pasuruan	$n = 5$
2	Gresik, Jombang, Kediri, Madiun, Malang, Nganjuk dan Surabaya	$n = 7$
3	Banyuwangi, Blitar, Lamongan, Mojokerto, Pacitan dan Ponorogo	$n = 7$

Setiap *cluster* memiliki karakteristik calon debitur yang berbeda. Karakteristik calon debitur dari setiap *cluster* dapat dilihat melalui nilai masing-masing variabel. Nilai variabel 5C disetiap *cluster* disajikan dalam Tabel 4.16.

Tabel 4.16. Nilai Rata-rata Tiap *Cluster*

Cluster	Character	Capacity	Capital	Collateral	Condition
1	-0,0243	0,0122	0,0857	-0,0329	-0,1482
2	-0,1059	0,0729	-0,0675	-0,1434	-0,0776
3	0,2248	0,7963	0,1622	0,9002	-1,0204

Berdasarkan Tabel 4.16, dapat disimpulkan bahwa *cluster* 1 merupakan *cluster* dengan calon debitur yang memiliki nilai variabel *Capacity* dan *Capital* di atas rata-rata. *Cluster* 2 merupakan *cluster* dengan calon debitur yang memiliki nilai variabel *Capacity* di atas rata-rata, sedangkan *cluster* 3 merupakan *cluster* dengan nilai variabel *Capacity*, *Capital* dan *Collateral* di atas rata-rata. Dengan demikian, pihak bank dapat membentuk kebijakan-kebijakan yang menyesuaikan kondisi tiap *cluster* untuk mengurangi resiko kegagalan pembayaran kredit oleh calon debitur.

4.2.6. Hasil Pengelompokan Kombinasi Ke-6

Nilai-nilai parameter yang digunakan dalam algoritma *FCM* pada pengelompokan kombinasi ke-6 dapat dilihat pada Tabel 4.17.

Tabel 4.17. Nilai Parameter pada Pengelompokan Kombinasi Ke-6

Parameter	Nilai
Jumlah <i>cluster</i> (c)	4
Kriteria Penghentian (ξ)	0,001
Maksimum iterasi	100
Parameter <i>fuzzification</i> (w)	2

Hasil analisis algoritma *FCM* dengan bantuan *Software R 3.5.1*. dapat dilihat pada Lampiran 8. Anggota tiap *cluster* dapat dilihat pada Tabel 4.18.

Tabel 4.18. Anggota Tiap *Cluster* Pada Pengelompokan Kombinasi Ke-6

<i>Cluster</i>	Anggota <i>Cluster</i>	Banyak Anggota
1	Madiun dan Surabaya	$n = 2$
2	Blitar, Gresik, Jombang, Kediri, Malang, Nganjuk, Pacitan dan Ponorogo	$n = 8$
3	Batu, Jember, Magetan, Ngawi dan Pasuruan	$n = 5$
4	Banyuwangi, Lamongan dan Mojokerto	$n = 3$

Setiap *cluster* memiliki karakteristik calon debitur yang berbeda. Karakteristik calon debitur dari setiap *cluster* dapat dilihat melalui nilai

masing-masing variabel. Nilai variabel $5C$ disetiap *cluster* disajikan dalam Tabel 4.19.

Tabel 4.19. Nilai Rata-rata Tiap *Cluster*

Cluster	Character	Capacity	Capital	Collateral	Condition
1	-0,2720	0,1225	-0,0950	-0,1999	-0,1475
2	0,0170	0,0304	-0,0373	-0,0943	0,0605
3	-0,0243	0,0122	0,0857	-0,0329	-0,1482
4	0,0725	-0,0683	0,0054	0,3630	0,2656

Berdasarkan Tabel 4.19, dapat disimpulkan bahwa *cluster* 1 merupakan *cluster* dengan calon debitur yang memiliki nilai variabel *Capacity* di atas rata-rata. *Cluster* 2 merupakan *cluster* dengan calon debitur yang memiliki nilai variabel *Character*, *Capacity* dan *Condition of Economy* di atas rata-rata. *Cluster* 3 merupakan *cluster* dengan calon debitur yang memiliki nilai variabel *Capacity* dan *Capital* di atas rata-rata, sedangkan *cluster* 4 merupakan *cluster* yang memiliki nilai variabel *Character*, *Capital*, *Collateral* dan *Condition of Economy* di atas rata-rata. Dengan demikian, pihak bank dapat membentuk kebijakan-kebijakan yang menyesuaikan kondisi tiap *cluster* untuk mengurangi resiko kegagalan pembayaran kredit oleh calon debitur.

4.2.7. Hasil Pengelompokan Kombinasi Ke-7

Nilai-nilai parameter yang digunakan dalam algoritma *FCM* pada pengelompokan kombinasi ke-7 dapat dilihat pada Tabel 4.20.

Tabel 4.20. Nilai Parameter pada Pengelompokan Kombinasi Ke-7

Parameter	Nilai
Jumlah <i>cluster</i> (c)	2
Kriteria Penghentian (ξ)	0,001
Maksimum iterasi	100
Parameter fuzzification (w)	2,5

Hasil analisis algoritma *FCM* dengan bantuan *Software R 3.5.1*. dapat dilihat pada Lampiran 9. Anggota tiap *cluster* dapat dilihat pada Tabel 4.21.

Tabel 4.21. Anggota Tiap *Cluster* Pada Pengelompokan Kombinasi Ke-7

<i>Cluster</i>	Anggota <i>Cluster</i>	Banyak Anggota
1	Batu, Gresik, Jember, Madiun, Magetan, Malang, Nganjuk, Ngawi, Pasuruan dan Surabaya	$n = 10$
2	Banyuwangi, Blitar, Jombang, Kediri, Lamongan, Mojokerto, Pacitan dan Ponorogo	$n = 8$

Setiap *cluster* memiliki karakteristik calon debitur yang berbeda. Karakteristik calon debitur dari setiap *cluster* dapat dilihat melalui nilai masing-masing variabel. Nilai variabel 5C disetiap *cluster* disajikan dalam Tabel 4.22.

Tabel 4.22. Nilai Rata-rata Tiap *Cluster*

<i>Cluster</i>	<i>Character</i>	<i>Capacity</i>	<i>Capital</i>	<i>Collateral</i>	<i>Condition</i>
1	-0,0647	0,0527	0,0058	-0,0798	-0,1273
2	0,0419	-0,0228	-0,0127	0,0710	0,1897

Berdasarkan Tabel 4.22, dapat disimpulkan bahwa *cluster* 1 merupakan *cluster* dengan calon debitur yang memiliki nilai variabel *Capacity* dan *Capital* di atas rata-rata, sedangkan *cluster* 2 merupakan *cluster* yang memiliki nilai variabel *Character*, *Collateral* dan *Condition of Economy* di atas rata-rata. Dengan demikian, pihak bank dapat membentuk kebijakan-kebijakan yang menyesuaikan kondisi tiap *cluster* untuk mengurangi resiko kegagalan pembayaran kredit oleh calon debitur.

4.2.8. Hasil Pengelompokan Kombinasi Ke-8

Nilai-nilai parameter yang digunakan dalam algoritma *FCM* pada pengelompokan kombinasi ke-8 dapat dilihat pada Tabel 4.23.

Tabel 4.23. Nilai Parameter pada Pengelompokan Kombinasi Ke-8

Parameter	Nilai
Jumlah <i>cluster</i> (c)	3
Kriteria Penghentian (ξ)	0,001
Maksimum iterasi	100
Parameter <i>fuzzification</i> (w)	2,5

Hasil analisis algoritma *FCM* dengan bantuan *Software R 3.5.1*. dapat dilihat pada Lampiran 10. Anggota tiap *cluster* dapat dilihat pada Tabel 4.24.

Tabel 4.24. Anggota Tiap *Cluster* Pada Pengelompokan Kombinasi Ke-8

Cluster	Anggota Cluster	Banyak Anggota
1	Banyuwangi, Blitar, Lamongan, Mojokerto, Pacitan dan Ponorogo	$n = 6$
2	Batu, Jember, Magetan, Ngawi dan Pasuruan	$n = 5$
3	Gresik, Jombang, Kediri, Madiun, Malang, Nganjuk dan Surabaya	$n = 7$

Setiap *cluster* memiliki karakteristik calon debitur yang berbeda. Karakteristik calon debitur dari setiap *cluster* dapat dilihat melalui nilai masing-masing variabel. Nilai variabel 5C disetiap *cluster* disajikan dalam Tabel 4.25.

Tabel 4.25. Nilai Rata-rata Tiap *Cluster*

Cluster	Character	Capacity	Capital	Collateral	Condition
1	0,0918	-0,0378	0,0000	0,1564	0,2549
2	-0,0243	0,0122	0,0857	-0,0329	-0,1482
3	-0,1059	0,0729	-0,0675	-0,1434	-0,0776

Berdasarkan Tabel 4.25, dapat disimpulkan bahwa *cluster* 1 merupakan *cluster* dengan calon debitur yang memiliki nilai variabel *Character*, *Capital* dan *Collateral* di atas rata-rata. *Cluster* 2 merupakan *cluster* dengan calon debitur yang memiliki nilai variabel *Capacity* dan *Capital* di atas rata-rata, sedangkan *cluster* 3 merupakan *cluster* dengan nilai variabel *Capacity* di atas rata-rata. Dengan demikian, pihak bank dapat membentuk kebijakan-kebijakan yang menyesuaikan kondisi tiap *cluster* untuk mengurangi resiko kegagalan pembayaran kredit oleh calon debitur.

4.2.9. Hasil Pengelompokan Kombinasi Ke-9

Nilai-nilai parameter yang digunakan dalam algoritma *FCM* pada pengelompokan kombinasi ke-9 dapat dilihat pada Tabel 4.26.

Tabel 4.26. Nilai Parameter pada Pengelompokan Kombinasi Ke-9

Parameter	Nilai
Jumlah <i>cluster</i> (c)	4
Kriteria Penghentian (ξ)	0,001
Maksimum iterasi	100
Parameter <i>fuzzification</i> (w)	2,5

Hasil analisis algoritma *FCM* dengan bantuan *Software R 3.5.1*. dapat dilihat pada Lampiran 11. Anggota tiap *cluster* dapat dilihat pada Tabel 4.27.

Tabel 4.27. Anggota Tiap *Cluster* Pada Pengelompokan Kombinasi Ke-9

<i>Cluster</i>	Anggota <i>Cluster</i>	Banyak Anggota
1	Jember, Magetan, Ngawi dan Pasuruan	$n = 4$
2	Jombang, Kediri, Madiun, Malang dan Surabaya	$n = 5$
3	Banyuwangi, Blitar, Lamongan, Mojokerto dan Pacitan	$n = 5$
4	Batu, Gresik, Nganjuk dan Ponorogo	$n = 4$

Setiap *cluster* memiliki karakteristik calon debitur yang berbeda. Karakteristik calon debitur dari setiap *cluster* dapat dilihat melalui nilai

masing-masing variabel. Nilai variabel 5C disetiap *cluster* disajikan dalam Tabel 4.28

Tabel 4.28. Nilai Rata-rata Tiap *Cluster*

Cluster	Character	Capacity	Capital	Collateral	Condition
1	0,4746	0,2906	0,2199	0,6137	0,7920
2	-0,1391	0,0553	-0,0746	-0,1758	-0,0576
3	0,0661	-0,0605	-0,0089	0,1948	0,2496
4	-0,0106	0,0710	-0,0011	0,0048	-0,0473

Berdasarkan Tabel 4.28, dapat disimpulkan bahwa *cluster* 1 merupakan *cluster* dengan calon debitur yang memiliki nilai seluruh variabel di atas rata-rata. *Cluster* 2 merupakan *cluster* dengan calon debitur yang memiliki nilai variabel *Capacity* di atas rata-rata. *Cluster* 3 merupakan *cluster* dengan calon debitur yang memiliki nilai variabel *Character* dan *Collateral* di atas rata-rata, sedangkan *cluster* 4 merupakan *cluster* yang memiliki nilai variabel *Capacity* dan *Collateral* di atas rata-rata. Dengan demikian, pihak bank dapat membentuk kebijakan-kebijakan yang menyesuaikan kondisi tiap *cluster* untuk mengurangi resiko kegagalan pembayaran kredit oleh calon debitur.

4.3. Pengelompokan Paling Optimal

Dari sembilan pengelompokan yang telah dilakukan, hanya satu pengelompokan optimal yang terpilih. Pengelompokan yang optimal dapat ditentukan dengan melihat nilai indeks Xie-Beni pada masing-masing pengelompokan. Nilai Indeks Xie-Beni dapat dihitung dengan menggunakan persamaan (2.12). Pengelompokan yang optimal adalah yang memiliki nilai indeks Xie-Beni terkecil. Nilai indeks Xie-Beni masing-masing pengelompokan dapat dilihat pada Tabel 4.29.

Tabel 4.29. Nilai Indeks Xie-Beni Masing-masing Pengelompokan

Pengelompokan	Nilai Indeks Xie-Beni
Kombinasi 1	0,5906
Kombinasi 2	0,7532
Kombinasi 3	0,4748
Kombinasi 4	0,8584

Tabel 4.29. Lanjutan

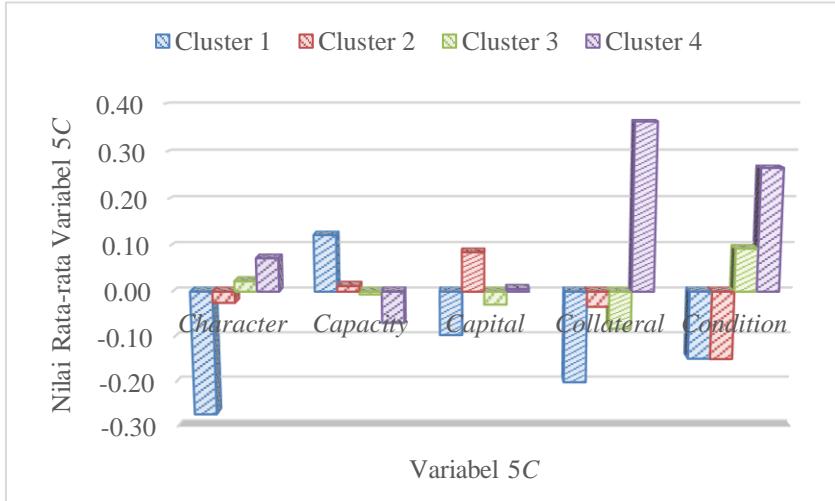
Pengelompokan	Nilai Indeks Xie-Beni
Kombinasi 5	0,8091
Kombinasi 6	0,3919
Kombinasi 7	2,2854
Kombinasi 8	1,1093
Kombinasi 9	0,8828

Berdasarkan Tabel 4.29 di atas, dapat dilihat bahwa pengelompokan kombinasi ke-6 memiliki nilai Indeks Xie-Beni terkecil. Dengan demikian, pengelompokan paling optimal adalah pengelompokan dengan penggunaan nilai parameter yang terdapat pada kombinasi ke-6. Nilai-nilai parameter yang digunakan pada kombinasi ke-6 dapat dilihat pada Tabel 4.17 Anggota dari masing-masing *cluster* pada pengelompokan kombinasi ke-6 dapat dilihat pada Tabel 4.18.

4.4. Interpretasi Hasil Pengelompokan Kombinasi ke-6

Nilai parameter pada pengelompokan kombinasi ke-6 adalah jumlah *cluster* (c) = 4; kriteria penghentian = 0,001; maksimum iterasi = 100 dan *fuzzification* (w) = 2.

Tentunya, dari keempat *cluster* yang terbentuk memiliki karakteristik yang berbeda. Karakteristik tiap *cluster* dapat dilihat melalui nilai variabel 5C (*Character, Capacity, Capital, Collateral* dan *Condition of Economy*) calon debitur pada 18 daerah di Jawa Timur. Secara grafis, nilai variabel 5C pada masing-masing *cluster* disajikan pada Gambar 4.6



Gambar 4.2. Rata-rata Variabel 5C Pada Setiap *Cluster*

Anggota masing-masing *cluster* dari hasil pengelompokan kombinasi ke-6 disajikan pada Tabel 4.30

Tabel 4.30. Anggota Tiap *Cluster*

Cluster	Anggota <i>Cluster</i>	Banyak Anggota
1	Madiun dan Surabaya	$n = 2$
2	Blitar, Gresik, Jombang, Kediri, Malang, Nganjuk, Pacitan dan Ponorogo	$n = 8$
3	Batu, Jember, Magetan, Ngawi dan Pasuruan	$n = 5$
4	Banyuwangi, Lamongan dan Mojokerto	$n = 3$

Berdasarkan Gambar 4.2 dapat diketahui bahwa *cluster* satu memiliki nilai variabel *Capacity* di atas rata-rata. Namun, memiliki nilai variabel *Character*, *Capital*, *Collateral* dan *Condition of Economy* di bawah rata-rata. Nilai variabel *Capacity* yang di atas rata-rata menunjukkan bahwa calon debitur pada *cluster* satu memiliki kemampuan mengelola usaha dengan baik. Jika kemampuan calon debitur dalam mengelola usaha baik, maka calon debitur tersebut akan mudah mendapatkan laba, sehingga calon debitur akan mudah

memenuhi tanggung jawabnya dalam membayar kredit. Oleh karena itu, *cluster* satu dinamakan “*cluster* kemampuan baik”. Berdasarkan karakteristik *cluster* satu, pihak bank dapat menentukan kebijakan yang sesuai sebagai bahan pertimbangan dalam pemberian kredit kepada calon debitur.

Cluster dua memiliki nilai variabel *Capacity* dan *Capital* di atas rata-rata. Namun, memiliki nilai variabel *Character*, *Collateral* dan *Condition of Economy* di bawah rata-rata. Nilai variabel *Capacity* dan *Capital* yang di atas rata-rata menunjukkan bahwa calon debitur pada *cluster* dua memiliki kemampuan mengelola usaha dan modal usaha yang cukup baik. Oleh karena itu, *cluster* dua dinamakan “*cluster* kemampuan dan modal baik”. Berdasarkan karakteristik *cluster* dua, pihak bank dapat menentukan kebijakan yang sesuai sebagai bahan pertimbangan dalam pemberian kredit kepada calon debitur.

Cluster tiga memiliki nilai variabel *Character* dan *Condition of Economy* di atas rata-rata. Namun, memiliki nilai variabel *Capacity*, *Capital* dan *Collateral* di bawah rata-rata. Nilai variabel *Character* yang di atas rata-rata menunjukkan bahwa calon debitur pada *cluster* tiga memiliki itikad yang baik dalam memenuhi tanggung jawabnya dalam membayar kredit. Nilai variabel *Condition of Economy* yang di atas rata-rata menunjukkan bahwa kota-kota yang berada dalam *cluster* tiga merupakan kota dengan kondisi perekonomian yang baik, sehingga dapat memberikan pengaruh yang baik untuk calon debitur dalam membayar kredit. Oleh karena itu, *cluster* tiga dinamakan “*cluster* watak dan kondisi perekonomian baik”. Berdasarkan karakteristik *cluster* tiga, pihak bank dapat menentukan kebijakan yang sesuai sebagai bahan pertimbangan dalam pemberian kredit kepada calon debitur.

Cluster empat memiliki nilai variabel *Character*, *Capital*, *Collateral* dan *Condition of Economy* di atas rata-rata. Namun, memiliki nilai variabel *Capacity* di bawah rata-rata. *Cluster* empat merupakan *cluster* yang paling unggul dibandingkan dengan ketiga *cluster* lainnya karena memiliki empat variabel yang di atas rata-rata. Nilai *Character* yang di atas rata-rata menunjukkan bahwa calon debitur pada *cluster* empat memiliki itikad yang baik dalam membayar kredit. Nilai *Capital* yang di atas rata-rata menunjukkan bahwa calon debitur pada *cluster* empat memiliki kemampuan mengelola usaha yang baik. Nilai *Collateral* yang di atas rata-rata menunjukkan bahwa jaminan yang

diberikan oleh calon debitir memiliki kondisi objek yang baik, memiliki kemudahan pengalihan kepemilikan dan memiliki tingkat harga yang jelas dan prospek pemasaran yang baik. Nilai *Condition of Economy* yang di atas rata-rata menunjukan bahwa kota-kota yang berada dalam *cluster* empat memiliki kondisi ekonomi yang baik. Oleh karena itu, *cluster* empat dinamakan “*cluster* unggulan”. Berdasarkan karakteristik *cluster* empat, pihak bank dapat menentukan kebijakan yang sesuai sebagai bahan pertimbangan dalam pemberian kredit kepada calon debitir.

Hasil pengelompokan dengan algoritma *Fuzzy C-means* dapat digunakan oleh pihak bank sebagai bahan pertimbangan dalam pemberian kredit kepada calon debitir. Misalnya, seorang calon debitir bernama Putri berasal dari Kota Malang mengajukan permohonan kredit. Kota Malang berada dalam “*cluster* kemampuan dan modal baik” (*cluster* dua). Dengan demikian, informasi karakter *cluster* dua dapat digunakan sebagai informasi tambahan oleh pihak bank dalam mempertimbangkan permohonan kredit Putri.

Saran yang dapat diberikan dari penelitian ini kepada pihak bank adalah sebaiknya pihak bank lebih mengalokasikan dana kredit kepada calon debitir dalam “*cluster* unggulan” karena calon debitir dalam *cluster* tersebut memiliki nilai variabel *Character*, *Capital*, *Collateral* dan *Condition of Economy* yang baik, sehingga dapat mendukung calon debitir memenuhi tanggung jawabnya dalam membayar kredit. Jika debitir lancar dalam membayar kredit, maka akan mempercepat proses perputaran uang yang artinya pihak bank dapat segera memberikan pinjaman kepada calon debitir lain dan mendapatkan keuntungan dari bunga pelunasan kredit tersebut.

Setelah mendapatkan hasil pengelompokan melalui Algoritma *Fuzzy C-means*, selanjutnya adalah membentuk pohon klasifikasi masing-masing *cluster*. Calon debitir pada tiap *cluster* (*cluster* satu, *cluster* dua, *cluster* tiga dan *cluster* empat) akan diklasifikasikan ke dalam kelas kredit lancar atau macet dengan menggunakan Algoritma *CART* yang akan dijelaskan dalam subbab berikutnya.

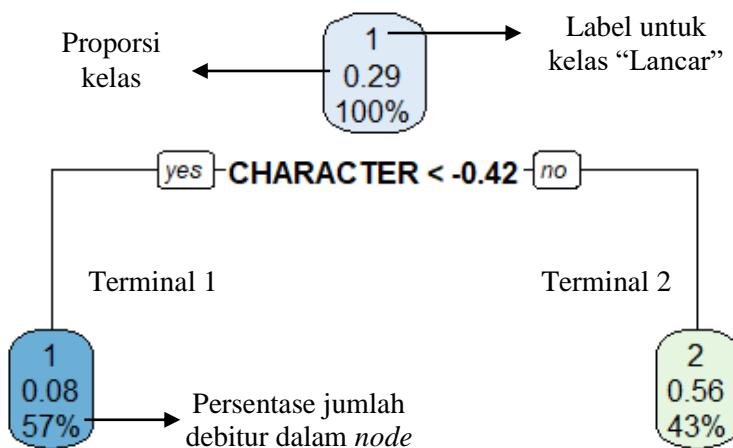
4.5. Analisis *CART*

Pada algoritma *Classification and Regression Trees* (*CART*), data calon debitir pada setiap *cluster* perlu dibagi dua menjadi data *training* (data latih) dan data *testing* (data uji) untuk mendapatkan pohon klasifikasi yang optimal. Pembagian data *training* dan data *testing*

dilakukan secara acak dengan perbandingan 75% untuk data *training* dan 15% untuk data *testing*. Analisis *CART* pada penelitian ini digunakan untuk membentuk pohon klasifikasi calon debitur pada masing-masing *cluster*. Hasil pengklasifikasian calon debitur dengan menggunakan analisis *CART* akan dijelaskan dalam subbab berikutnya.

4.5.1. Pohon Klasifikasi *CART* pada *Cluster* Satu

Pohon klasifikasi *CART* pada *cluster* satu beserta keterangannya disajikan pada Gambar 4.3. Terdapat tiga angka di setiap kotak simpul. Angka paling atas menunjukkan prediksi kelas klasifikasi. Kelas lancar dinotasikan “1”, sedangkan kelas macet dinotasikan “2”. Angka di tengah menunjukkan proporsi kelas dalam *node*. Angka paling bawah menunjukkan persentase jumlah debitur yang diklasifikasikan dalam *node*.



Gambar 4.3. Pohon Klasifikasi *CART* Optimal Pada *Cluster* Satu

Hasil analisis *CART* pada *cluster* satu membentuk pohon klasifikasi dengan dua simpul terminal yang dinotasikan dengan “Terminal 1” dan “Terminal 2”. Pembentukan pohon klasifikasi berdasarkan ukuran *cost complexity minimum* pada persamaan (2.16).

Hasil perhitungan *cost complexity minimum* melalui paket *rpart* pada software *R* disajikan dalam Tabel 4.31

Tabel. 4.31. Hasil Perhitungan *CV* pada Pohon Klasifikasi

<i>cp</i>	<i>nsplit</i>	<i>R(T_t)</i>	<i>Re(T_t)</i>	<i>CV(T_t)</i>	<i>SE(T_t)</i>
0,17	0	0,28571	1,0	1,0	0,34503
0,00	1	0,22856	0,8	1,0	0,34503

Keterangan:

cp (α) : *complexity parameter*

R(T_t) : *resubstitution estimate* pada subpohon *T_t*

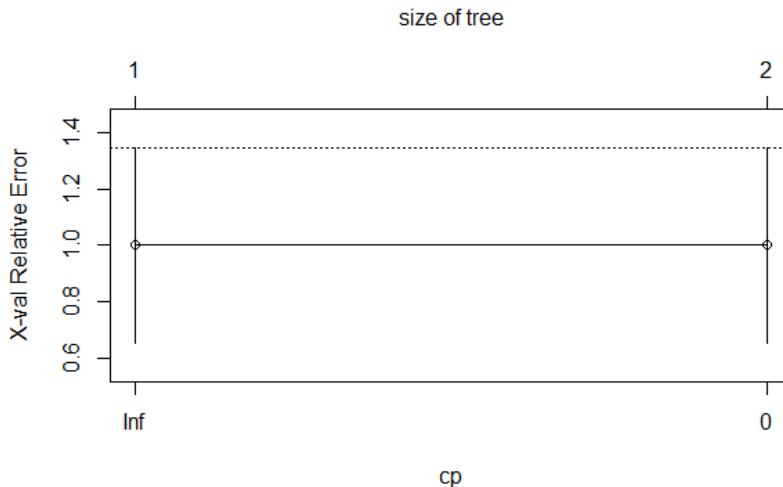
Re(T_t) : *relative error* pada sub pohon *T_t*

nsplit : banyaknya pemilah

CV(T_t) : rata-rata *relative error* pada sub pohon *T_t*

SE(T_t) : standar *error* pada sub pohon *T_t*

Nilai-nilai pada table 4.31 disajikan ke dalam Gambar 4.4.



Gambar 4.4. Plot *Complexity Parameter*

X-val Relative Error pada Gambar 4.4. menunjukkan nilai *cross validation relative error* pada tiap ukuran pohon yang terbentuk (*size of tree*). Garis putus-putus pada Gambar 4.4. menunjukkan nilai *one*

standard error rule, yaitu nilai minimum $CV(T_i)$ ditambah dengan $SE(T_i)$. Pohon klasifikasi yang optimal merupakan pohon yang memiliki nilai *one standard erro rule* yang paling mendekati garis putus-putus dan terlihat pada Gambar 4.4. pohon klasifikasi optimal merupakan pohon klasifikasi yang terdiri dari dua pemilahan simpul terminal seperti pada Gambar 4.3.

Pohon klasifikasi pada Gambar 4.3. menunjukkan bahwa variabel *Character* (X_1) merupakan variabel pemilah utama dan paling menentukan klasifikasi calon debitur pada *cluster* satu. Karakteristik masing-masing simpul terminal dijelaskan sebagai berikut:

1. Terminal 1 merupakan simpul terminal dengan faktor calon debitur yang memiliki nilai *Character* kurang dari -0,4192. Pada simpul ini terdapat 57,1% calon debitur yang tepat diklasifikasikan ke dalam kelas 1, yaitu kelas kredit lancar.
2. Terminal 2 merupakan simpul dengan faktor calon debitur yang memiliki nilai *Character* lebih dari sama dengan 6,5. Pada simpul ini terdapat 42,9% calon debitur yang tepat diklasifikasikan ke dalam kelas 2, yaitu kelas kredit macet.

Setelah mendapatkan pohon klasifikasi, langkah selanjutnya adalah menghitung ketepatan hasil klasifikasi untuk mengetahui ketepatan calon debitur berada dalam suatu kelas. Tabel konfusi yang diperoleh dari pohon klasifikasi pada data *training* dapat dilihat pada Tabel 4.32.

Tabel 4.32. Tabel Konfusi *CART* pada *Cluster* Satu

Keanggotaan	Prediksi Keanggotaan	
	Lancar	Macet
Lancar	3	1
Macet	2	1

$$\text{Hit ratio (100\%)} = \frac{3+1}{3+2+1+1} \times 100\% = 57,14\%$$

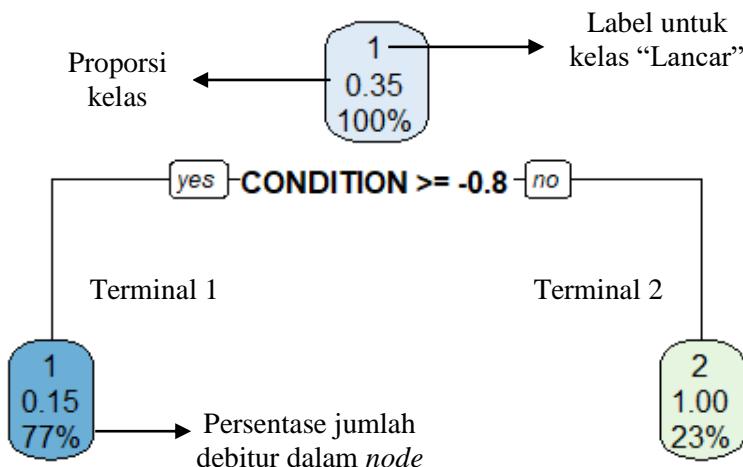
$$\text{APER(100\%)} = 100\% - 57,14\% = 42,86\%$$

Didapatkan ketepatan hasil klasifikasi keseluruhan (*Hit ratio*) sebesar 57,14%, artinya pohon klasifikasi *CART* mampu

mengklasifikasikan kelas pada data *testing* dengan baik dan tepat sebesar 57%. Selain itu, didapatkan nilai total tingkat kesalahan klasifikasi (*APER*) sebesar 42,86%, artinya pohon klasifikasi *CART* salah mengklasifikasikan kelas pada data *testing* sebesar 43%.

4.5.2. Pohon Klasifikasi *CART* pada Cluster Dua

Pohon klasifikasi *CART* pada *cluster* dua beserta keterangannya disajikan pada Gambar 4.5. Terdapat tiga angka di setiap kotak simpul. Angka paling atas menunjukkan prediksi kelas klasifikasi. Kelas lancar dinotasikan “1”, sedangkan kelas macet dinotasikan “2”. Angka di tengah menunjukkan proporsi kelas dalam *node*. Angka paling bawah menunjukkan persentase jumlah debitur yang diklasifikasikan dalam *node*.



Gambar 4.5. Pohon Klasifikasi *CART* Optimal Pada *Cluster* Dua

Hasil analisis *CART* pada *cluster* dua membentuk pohon klasifikasi dengan dua simpul terminal yang dinotasikan dengan “Terminal 1” dan “Terminal 2”. Pembentukan pohon klasifikasi berdasarkan ukuran *cost complexity minimum* pada persamaan (2.16). Hasil perhitungan *cost complexity minimum* melalui paket *rpart* pada *software R* disajikan dalam Tabel 4.33.

Tabel. 4.33. Hasil Perhitungan CV pada Pohon Klasifikasi

cp	$nsplit$	$R(T_t)$	$Re(T_t)$	$CV(T_t)$	$SE(T_t)$
0,67	0	0,34615	1,0	1,0	0,19059
0,00	1	0,10384	0,3	0,5	0,14212

Keterangan:

cp (α) : complexity parameter

$R(T_t)$: resubstitution estimate pada subpohon T_t

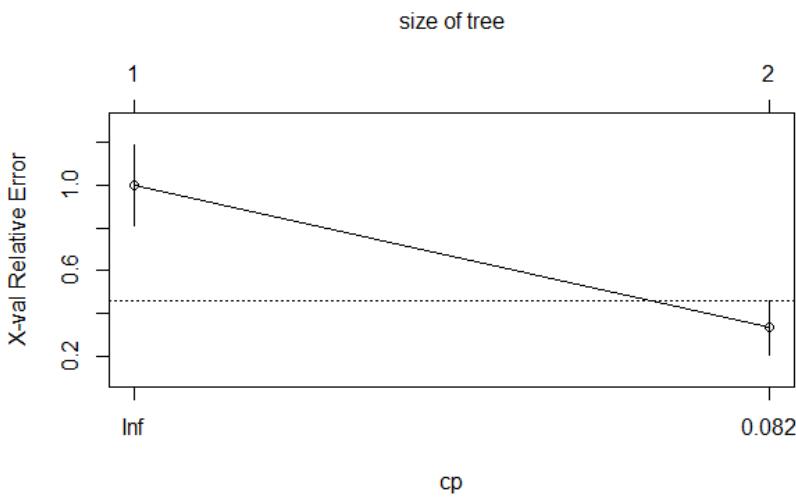
$Re(T_t)$: relative error pada sub pohon T_t

$nsplit$: banyaknya pemilah

$CV(T_t)$: rata-rata relative error pada sub pohon T_t

$SE(T_t)$: standar error pada sub pohon T_t

Nilai-nilai pada table 4.32 disajikan ke dalam Gambar 4.6.



Gambar 4.6. Plot Complexity Parameter

$X\text{-val Relative Error}$ pada Gambar 4.6. menunjukkan nilai *cross validation relative error* pada tiap ukuran pohon yang terbentuk (*size of tree*). Garis putus-putus pada Gambar 4.6. menunjukkan nilai *one standard error rule*, yaitu nilai minimum $CV(T_t)$ ditambah dengan $SE(T_t)$. Pohon klasifikasi yang optimal merupakan pohon yang memiliki nilai *one standard error rule* yang paling mendekati garis putus-putus

dan terlihat pada Gambar 4.6. pohon klasifikasi optimal merupakan pohon klasifikasi yang terdiri dari dua pemilahan simpul terminal seperti pada Gambar 4.5

Pohon klasifikasi pada Gambar 4.5. menunjukkan bahwa variabel *Condition of Economy* (X_5) merupakan variabel pemilah utama dan paling menentukan klasifikasi calon debitur pada *cluster* dua. Karakteristik masing-masing simpul terminal dijelaskan sebagai berikut:

1. Terminal 1 merupakan simpul terminal dengan faktor calon debitur yang memiliki nilai *Condition of Economy* lebih dari sama dengan -0,8042. Pada simpul ini terdapat 76,9% calon debitur yang tepat diklasifikasikan ke dalam kelas 1, yaitu kelas kredit lancar.
2. Terminal 2 merupakan simpul dengan faktor calon debitur yang memiliki nilai *Condition of Economy* kurang dari -0,8042. Pada simpul ini terdapat 23,1% calon debitur yang tepat diklasifikasikan ke dalam kelas 2, yaitu kelas kredit macet.

Setelah mendapatkan pohon klasifikasi, langkah selanjutnya adalah menghitung ketepatan hasil klasifikasi untuk mengetahui ketepatan calon debitur berada dalam suatu kelas. Tabel konfusi yang diperoleh dari pohon klasifikasi pada data *training* dapat dilihat pada Tabel 4.34.

Tabel 4.34. Tabel Konfusi *CART* pada *Cluster* Dua

Prediksi Keanggotaan	Aktual Keanggotaan	
	Lancar	Macet
Lancar	16	2
Macet	0	0

$$\text{Hit ratio (100\%)} = \frac{16+0}{16+2} \times 100\% = 88,88\%$$

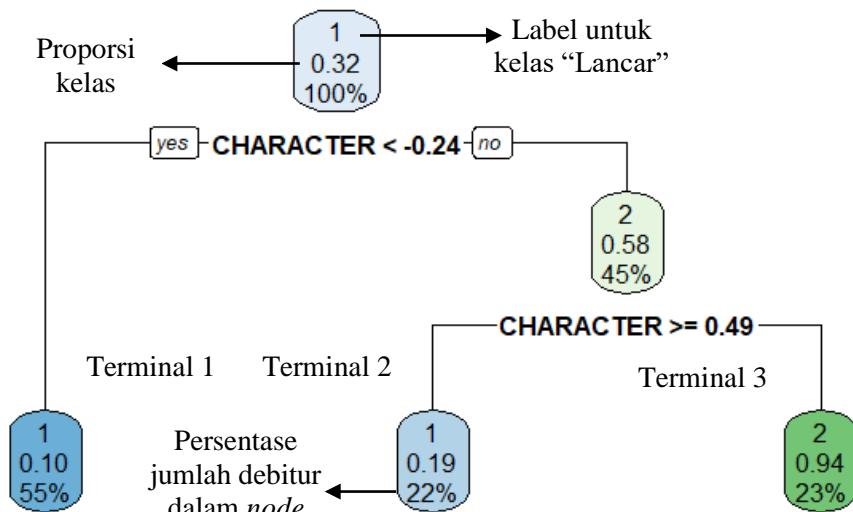
$$\text{APER(100\%)} = 100\% - 88,88\% = 11,12\%$$

Didapatkan ketepatan hasil klasifikasi keseluruhan (*Hit ratio*) sebesar 88,88%, artinya pohon klasifikasi *CART* mampu mengklasifikasikan kelas pada data *testing* dengan baik dan tepat sebesar 88,9%. Selain itu, didapatkan nilai total tingkat kesalahan

klasifikasi (*APER*) sebesar 11,12%, artinya pohon klasifikasi *CART* salah mengklasifikasikan kelas pada data *testing* sebesar 11,1%.

4.5.3. Pohon Klasifikasi *CART* pada Cluster Tiga

Pohon klasifikasi *CART* pada *cluster* tiga beserta keterangannya disajikan pada Gambar 4.7. Terdapat tiga angka di setiap kotak simpul. Angka paling atas menunjukkan prediksi kelas klasifikasi. Kelas lancar dinotasikan “1”, sedangkan kelas macet dinotasikan “2”. Angka di tengah menunjukkan proporsi kelas dalam *node*. Angka paling bawah menunjukkan persentase jumlah debitir yang diklasifikasikan dalam *node*.



Gambar 4.7. Pohon Klasifikasi *CART* Optimal Pada Cluster Tiga

Hasil analisis *CART* pada *cluster* tiga membentuk pohon klasifikasi dengan tiga simpul terminal yang dinotasikan dengan “Terminal 1”, “Terminal 2” dan Terminal “3”. Pembentukan pohon klasifikasi berdasarkan ukuran *cost complexity minimum* pada persamaan (2.16). Hasil perhitungan *cost complexity minimum* melalui paket *rpart* pada *software R* disajikan dalam Tabel 4.35.

Tabel. 4.35. Hasil Perhitungan CV pada Pohon Klasifikasi

cp	$nsplit$	$R(T_t)$	$Re(T_t)$	$CV(T_t)$	$SE(T_t)$
0,33	0	0,31507	1,0	1,0	0,17257
0,00	2	0,34783	0,56522	0,5	0,14212

Keterangan:

cp (α) : complexity parameter

$R(T_t)$: resubstitution estimate pada subpohon T_t

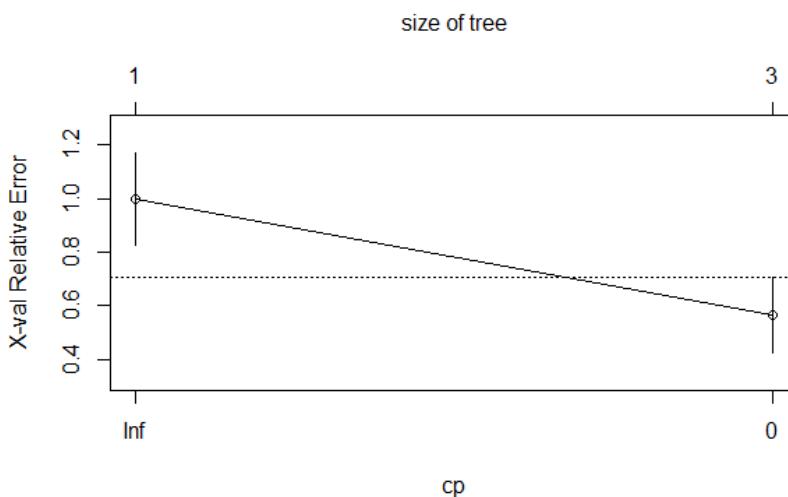
$Re(T_t)$: relative error pada sub pohon T_t

$nsplit$: banyaknya pemilah

$CV(T_t)$: rata-rata relative error pada sub pohon T_t

$SE(T_t)$: standar error pada sub pohon T_t

Nilai-nilai pada table 4.35 disajikan ke dalam Gambar 4.8.



Gambar 4.8. Plot Complexity Parameter

X-val Relative Error pada Gambar 4.8. menunjukkan nilai *cross validation relative error* pada tiap ukuran pohon yang terbentuk (*size of tree*). Garis putus-putus pada Gambar 4.8. menunjukkan nilai *one standard error rule*, yaitu nilai minimum $CV(T_t)$ ditambah dengan $SE(T_t)$. Pohon klasifikasi yang optimal merupakan pohon yang memiliki nilai *one standard error rule* yang paling mendekati garis putus-putus

dan terlihat pada Gambar 4.8. pohon klasifikasi optimal merupakan pohon klasifikasi yang terdiri dari tiga pemilahan simpul terminal seperti pada Gambar 4.7.

Pohon klasifikasi pada Gambar 4.7. menunjukkan bahwa variabel *Character* (X_1) merupakan variabel pemilah utama dan paling menentukan klasifikasi calon debitur pada *cluster* tiga. Karakteristik masing-masing simpul terminal dijelaskan sebagai berikut:

1. Terminal 1 merupakan simpul terminal dengan faktor calon debitur yang memiliki nilai *Character* kurang dari -0,2351. Pada simpul ini terdapat 54,8% calon debitur yang tepat diklasifikasikan ke dalam kelas 1, yaitu kelas kredit lancar.
2. Terminal 2 merupakan simpul terminal dengan faktor calon debitur yang memiliki nilai *Character* lebih dari sama dengan 0,4876. Pada simpul ini terdapat 21,9% calon debitur yang tepat diklasifikasikan ke dalam kelas 1, yaitu kelas kredit lancar.
3. Terminal 3 merupakan simpul terminal dengan faktor calon debitur yang memiliki nilai *Character* kurang dari 0,4876. Pada simpul ini terdapat 23,3% calon debitur yang tepat diklasifikasikan ke dalam kelas 2, yaitu kelas kredit macet.

Setelah mendapatkan pohon klasifikasi, langkah selanjutnya adalah menghitung ketepatan hasil klasifikasi untuk mengetahui ketepatan calon debitur berada dalam suatu kelas. Matriks konfusi yang diperoleh dari pohon klasifikasi pada data *training* dapat dilihat pada Tabel 4.36.

Tabel 4.36. Tabel Konfusi *CART* pada *Cluster* Tiga

Prediksi Keanggotaan	Aktual Keanggotaan	
	Lancar	Macet
Lancar	20	1
Macet	0	4

$$\text{Hit ratio (100\%)} = \frac{20+4}{20+1+4} \times 100\% = 96\%$$

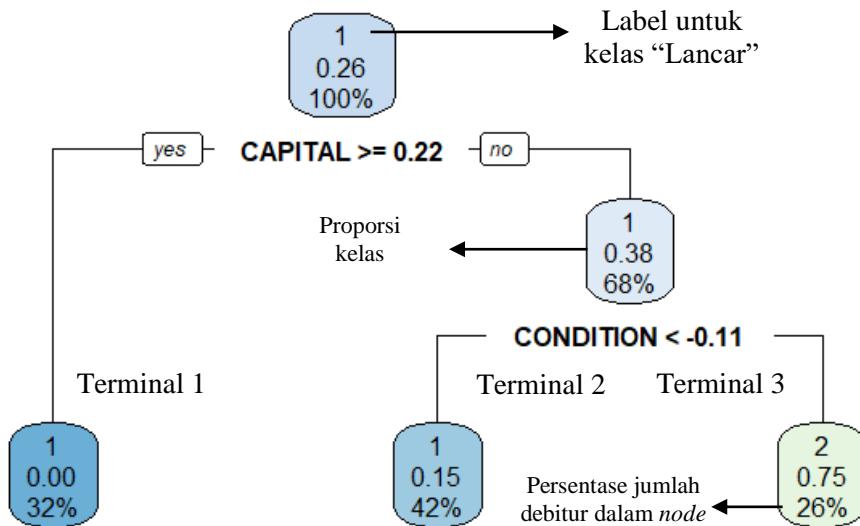
$$APER(100\%) = 100\% - 96\% = 0,04\%$$

Didapatkan ketepatan hasil klasifikasi keseluruhan (*Hit ratio*) sebesar 96%, artinya pohon klasifikasi *CART* mampu

mengklasifikasikan kelas pada data *testing* dengan baik dan tepat sebesar 96%. Selain itu, didapatkan nilai total tingkat kesalahan klasifikasi (*APER*) sebesar 0,04%, artinya pohon klasifikasi *CART* salah mengklasifikasikan kelas pada data *testing* sebesar 0,04%.

4.5.4. Pohon Klasifikasi *CART* pada Cluster Empat

Pohon klasifikasi *CART* pada *cluster* empat beserta keterangannya disajikan pada Gambar 4.9. Terdapat tiga angka di setiap kotak simpul. Angka paling atas menunjukkan prediksi kelas klasifikasi. Kelas lancar dinotasikan “1”, sedangkan kelas macet dinotasikan “2”. Angka di tengah menunjukkan proporsi kelas dalam *node*. Angka paling bawah menunjukkan persentase jumlah debitir yang diklasifikasikan dalam *node*.



Gambar 4.9. Pohon Klasifikasi *CART* Optimal Pada *Cluster* Empat

Hasil analisis *CART* pada *cluster* empat membentuk pohon klasifikasi dengan tiga simpul terminal yang dinotasikan dengan “Terminal 1”, “Terminal 2” dan Terminal “3”. Pembentukan pohon klasifikasi berdasarkan ukuran *cost complexity minimum* pada persamaan (2.16). Hasil perhitungan *cost complexity minimum* melalui paket *rpart* pada *software R* disajikan dalam Tabel 4.37.

Tabel. 4.37. Hasil Perhitungan CV pada Pohon Klasifikasi

cp	$nsplit$	$R(T_t)$	$Re(T_t)$	$CV(T_t)$	$SE(T_t)$
0,25	0	0,25806	1,0	1,0	0,30454
0,00	2	0,12903	0,5	1,25	0,32534

Keterangan:

cp (α) : complexity parameter

$R(T_t)$: resubstitution estimate pada subpohon T_t

$Re(T_t)$: relative error pada sub pohon T_t

$nsplit$: banyaknya pemilah

$CV(T_t)$: rata-rata relative error pada sub pohon T_t

$SE(T_t)$: standar error pada sub pohon T_t

Nilai-nilai pada table 4.37 disajikan ke dalam Gambar 4.10.



Gambar 4.10. Plot Complexity Parameter

X-val Relative Error pada Gambar 4.10. menunjukkan nilai *cross validation relative error* pada tiap ukuran pohon yang terbentuk (*size of tree*). Garis putus-putus pada Gambar 4.10. menunjukkan nilai *one standard error rule*, yaitu nilai minimum $CV(T_t)$ ditambah dengan $SE(T_t)$. Pohon klasifikasi yang optimal merupakan pohon yang memiliki nilai *one standard error rule* yang paling mendekati garis putus-putus

dan terlihat pada Gambar 4.10. pohon klasifikasi optimal merupakan pohon klasifikasi yang terdiri dari tiga pemilahan simpul terminal seperti pada Gambar 4.9.

Pohon klasifikasi pada Gambar 4.9. menunjukkan bahwa variabel *Capital* (X_3) merupakan variabel pemilah utama dan paling menentukan klasifikasi calon debitur pada *cluster* empat. Selain itu, variabel *Condition of Economy* juga turut menentukan klasifikasi calon debitur pada *cluster* empat. Pembentukan pohon klasifikasi Karakteristik masing-masing simpul terminal dijelaskan sebagai berikut:

1. Terminal 1 merupakan simpul terminal dengan faktor calon debitur yang memiki nilai *Capital* lebih besar dari sama dengan 0,2224. Pada simpul ini terdapat 32,3% calon debitur yang tepat diklasifikasikan ke dalam kelas 1, yaitu kelas kredit lancar.
2. Terminal 2 merupakan simpul terminal dengan faktor calon debitur yang memiki nilai *Condition of Economy* kurang dari lebih dari sama dengan -0,1123. Pada simpul ini terdapat 41,9% calon debitur yang tepat diklasifikasikan ke dalam kelas 1, yaitu kelas kredit lancar.
3. Terminal 3 merupakan simpul terminal dengan faktor calon debitur yang memiki nilai *Condition of Economy* lebih dari sama dengan -0,1125. Pada simpul ini terdapat 25,8% calon debitur yang tepat diklasifikasikan ke dalam kelas 2, yaitu kelas kredit macet.

Setelah mendapatkan pohon klasifikasi, langkah selanjutnya adalah menghitung ketepatan hasil klasifikasi untuk mengetahui ketepatan calon debitur berada dalam suatu kelas. Tabel konfusi yang diperoleh dari pohon klasifikasi pada data *training* dapat dilihat pada Tabel 4.38.

Tabel 4.38. Tabel Konfusi CART pada *Cluster* Tiga

Prediksi Keanggotaan	Aktual Keanggotaan	
	Lancar	Macet
Lancar	5	2
Macet	2	2

$$\text{Hit ratio (100\%)} = \frac{5+2}{5+2+2+2} \times 100\% = 90,9\%$$

$$\text{APER(100\%)} = 100\% - 90,9\% = 9,1\%$$

Didapatkan ketepatan hasil klasifikasi keseluruhan (*Hit ratio*) sebesar 90,9%, artinya pohon klasifikasi *CART* mampu mengklasifikasikan kelas pada data *testing* dengan baik dan tepat sebesar 91%. Selain itu, didapatkan nilai total tingkat kesalahan klasifikasi (*APER*) sebesar 9,1%, artinya pohon klasifikasi *CART* salah mengklasifikasikan kelas pada data *testing* sebesar 9%.

BAB V **PENUTUP**

5.1. Kesimpulan

Berikut ini kesimpulan yang dapat diambil dari hasil dan pembahasan pada bab sebelumnya:

1. *Cluster* optimal yang terbentuk dari data calon debitur pada 18 kota di Jawa Timur berdasarkan prinsip 5C melalui algoritma *FCM* adalah sebanyak empat *cluster* dengan nilai index Xie-Beni sebesar 0,3919 dan parameter *fuzzification* sebesar 2. Setiap *cluster* memiliki karakteristik yang berbeda. *Cluster* satu merupakan *cluster* “kemampuan baik”, *cluster* dua merupakan *cluster* “kemampuan dan modal baik”, *cluster* tiga merupakan *cluster* “watak dan kondisi perekonomian baik” dan *cluster* empat merupakan *cluster* “unggulan”
2. Pohon klasifikasi yang terbentuk melalui algoritma *CART* pada *cluster* pertama merupakan pohon dengan dua simpul terminal. Sebanyak 57,1% calon debitur tepat diklasifikasikan ke dalam kelas kredit lancar dan 42,9% diklasifikasikan kedalam kredit macet. Pohon klasifikasi pada *cluster* dua merupakan pohon dengan dua simpul terminal. Sebanyak 76,9% calon debitur tepat diklasifikasikan ke dalam kelas kredit lancar dan 23,1% diklasifikasikan ke dalam kelas macet. Pohon klasifikasi pada *cluster* tiga merupakan pohon dengan tiga simpul terminal. Sebanyak 76,7% calon debitur yang tepat diklasifikasikan ke dalam kelas kredit lancar, sedangkan 23,3% diklasifikasikan ke dalam kelas macet. Pohon klasifikasi pada *cluster* empat merupakan pohon dengan tiga simpul terminal. Sebanyak 74,2% calon debitur yang tepat diklasifikasikan ke dalam kelas kredit lancar, sedangkan 25,8% diklasifikasikan ke dalam kelas macet. Pohon klasifikasi disetiap *cluster* memiliki nilai ketepatan hasil klasifikasi keseluruhan (*Hit Ratio*) lebih dari 50%, sehingga pohon klasifikasi yang terbentuk sudah cukup baik dan tepat dalam mengklasifikasikan calon debitur.

5.2. Saran

Beberapa saran yang dapat diberikan berdasarkan hasil penelitian ini adalah sebagai berikut:

1. Algoritma *Fuzzy C-means* dapat diterapkan oleh pihak bank untuk mengelompokan data calon debitur berdasarkan prinsip 5C. Hasil

pengelompokan dapat digunakan sebagai bahan pertimbangan pemberian kredit, pengalokasian dana kredit dan pemberlakuan kebijakan-kebijakan bank.

2. Algoritma *CART* dapat diterapkan oleh pihak bank untuk mengklasifikasikan calon debitur ke dalam kelas kredit lancar atau kelas kredit macet. Hasil klasifikasi dapat digunakan untuk memprediksi potensi seorang calon debitur menyelesaikan kreditnya dengan lancar atau macet (menunggak).
3. Sebaiknya pihak bank lebih mengalokasikan dana kredit kepada calon debitur dalam “*cluster unggulan*” karena calon debitur dalam *cluster* tersebut memiliki nilai variabel *Character*, *Capital*, *Collateral* dan *Condition of Economy* yang baik, sehingga dapat mendukung calon debitur memenuhi tanggung jawabnya dalam membayar kredit. Jika debitur lancar dalam membayar kredit, maka akan mempercepat proses perputaran uang yang artinya pihak bank dapat segera memberikan pinjaman kepada calon debitur lain dan mendapatkan keuntungan dari bunga pelunasan kredit tersebut.
4. Pengelompokan yang dilakukan pada penelitian ini berdasarkan variabel *5C* yang dimiliki calon debitur. Hanya terdapat satu variabel pada *5C* yang menunjukkan karakteristik kota, yaitu variabel *Condition of Economy*. Saran untuk penelitian selanjutnya adalah apabila tujuan penelitian ingin mengelompokan kota sebaiknya menggunakan variabel yang lebih banyak dan mencerminkan karakteristik dari suatu kota, seperti indeks pendapatan perkapita, jumlah penduduk dan tetap menyertakan variabel *5C*.

DAFTAR PUSTAKA

- Amalia, N.A., Widodo, D.A. dan Oktaviana, P.P. 2016. "Analisis Clustering Perusahaan Sub Sektor Perbankan Berdasarkan Rasio Keuangan CAMELS Tahun 2014 Menggunakan Metode *Fuzzy C-Means* dan *Fuzzy Gustafon Kessel*". *Jurnal SAINS dan Seni ITS*. Vol. 5, No. 2, pp. 321 – 326.
- Balasko, B., Abonyi, J. dan Feil, B. 2007. *Fuzzy Clustering and Data Analysis Toolbox For Use with Matlab*. University of Veszprem: Veszprem
- Bezdek, J. 1981. *Pattern Recognition with Fuzzy Objective Function Algorithm*. New York: Plenum Press
- Breiman, L., Friedman, J.H., Olshen, R.A., dan Stone, C. J. 1993. *Classification and Regression Trees*. CA: Wadsworth
- Cox, Earl. 2005. *Fuzzy Modelling and Genetic Algorithms for Data Mining and Exploration*. San Fransisco: Morgan Kaufmann Publishers.
- Fernandes, A.A.R. dan Soehono, L.A. 2006. Kajian Analisis *Cluster* Pada Data Berskala Campuran. *Penelitian DPP*. Malang. Universitas Brawijaya
- Fitria, N. dan Sari, L. 2012. "Analisis Kebijakan Pemberian Kredit Dan Pengaruh Nonperforming Loan Terhadap *Loan To Deposit Ratio* Pada PT. Bank Rakyat Indonesia (PERSERO), TBK. Cabang Rantau, Aceh Tamiang. Periode 2007 – 2011". *Jurnal Ekonomi dan Keuangan*. Vol. 1, No. 1, pp. 88 – 100.
- Gifi, A. 1990. *Nonlinear Multivariate Analysis*. England: John Wiley & Sons Ltd.
- Han, J. dan Kamber, M. 2001. *Data Mining Concept and Technique*. California: Morgan Kaufmann.
- Handoyo, S. dan Prasojo, A. P. S. 2017. Sistem *Fuzzy* Terapan dengan Software R. Malang: UB Press

- Johnson, R.A. dan Winchern D.W. 2002. *Applied Multivariate Statistical Analisys 5th Edition*. London : Prentice-Hall
- Johnson, R.A. dan Winchern D.W. 2007. *Applied Multivariate Statistical Analisys 6th Edition*. London : Prentice-Hall
- Kusumadewi, S. 2002. *Analisis & Desain Sistem Fuzzy Menggunakan Tool Box Matlab*. Yogyakarta. Graha Ilmu.
- Megawati, N., Mukid, M.A. dan Rahmawati, R. 2013. “Segmentasi Pasar Pada Pusat Pembelanjaan Menggunakan *Fuzzy C-Means* (Studi Kasus: Rita Pasaraya Cilacap)”. *Jurnal Gaussian*. Vol. 2, No. 4, pp. 343 – 350.
- Ross T.J. 2010. *Fuzzy Logic with Engineering Applications. Second Edition*. New York: John Wiley and Sons
- Santosa, B. dan Umam, A. 2018. Data Mining dan *Big Data Anaytics*, 2nd ed. Yogyakarta: Penebar Media Pustaka.
- Saraswati, R.A. 2012. “Peranan Analisis Laporan Keuangan, Penilaian Prinsip 5C Calon Debitur dan Pegawai Kredit Terhadap Efektivitas Pemberian Kredit Pada PD BPR Bank Pasar Kabupaten Temanggung”. *Jurnal Nominal*. Vol. 1, No. 1.
- Sinungan. 2007. Manajemen Lembaga Keuangan. Jakarta: LPFEUI
- Solimun. 2010. *Analisis Multivariat Pemodelan Struktural Metode Partial Least Square –PLS*. Malang: CV. Citra Malang.
- Solimun., Fernandes, A.A.R. dan Nurjannah. 2017. *Metode Statistika Multivariat. Pemodelan Persamaan Struktural (SEM) Pendekatan WarpPLS*. Malang: UB Press.
- Supranto, J. 2004. *Analisis Multivariat Arti dan Interpretasi*. Jakarta: PT. Rineka Cipta
- Tamonob, A.M., Saefuddin, A. dan Wigena, A.H. 2015. “Nonliniear Principal Component Analysis and Principal Component Analysis With Successive Interval In K-Means Cluster

Analysis”. *Indonesian Journal of Statistic*. Vol. 20, No. 2, pp. 68 – 77.

Tanjung, R. H. dan Kartiko, K. 2017. Penerapan Metode CART (Classification and Regression Trees) untuk Menentukan Faktor-Faktor yang Mempengaruhi Pembayaran Kredit oleh Nasabah (Studi Kasus Bank BRI Unit Aek Tarum–Sumatera Utara). *Jurnal Statistika Industri dan Komputasi*, 2(02), 78-83.

LAMPIRAN

Lampiran 1. *Source code* untuk Algoritma *Fuzzy C-means*

```
initpar <- function (c,n) {  
#bangkitkan bilangan random uniform  
  set.seed(1)  
  U<-matrix(runif(n*c),nrow=c)  
  col.sum<-colSums(U)  
#normalisasi elemen kolom U  
  for(h in 1:c) {  
    for(i in 1:n){  
      U[h,i] = U[h,i]/col.sum[i]  
    }  
  }  
  return(U)  
}  
#Fungsi initpar U  
U<-initpar (2,18)  
round(t(head(t(U))),4) #U untuk 6 pengamatan pertama  
colSums(U)  
#MENGHITUNG PUSAT CLUSTER  
compcenter<-function(x,U,w){  
  #menghitung pusat cluster  
  mf<-U^w  
  dum<-matrix(1,dim(x)[2],1)  
  V<-mf%*%x/(t(dum%*%colSums(t(mf))))  
  return(V)  
}  
#Fungsi menghitung pusat cluster  
V<-compcenter(x,U,1.5)  
V  
#MENGHITUNG FUNGSI OBJEKTIF  
fobj<-function(D,U,w){  
  obj<-sum((D^2)*(U^w))  
  return (obj)  
}  
#Jarak Euclid (D)  
euclideanDist<-function(x,V){  
  #identifikasi struktur input  
  c<-dim(V)[1]
```

Lampiran 1. (Lanjutan)

```
n<-dim(x)[1]
D<-matrix(0,c,n)
#menghitung jarak euclidean
for(h in 1:c) {
  D[h,]<=sqrt(colSums(t((xmatrix(1,n,1)%*%V[h,])^2)))
}
return(D)
}
#Memanggil hasil euclidean dan fungsi objektif
dist<-euclideandist(x,V)
dist
objval<-fobj(dist,U,1.5)
objval
onestepfkm<-function(x,U,w) {
  c<-nrow(U) #banyaknya cluster
  V<-compccenter(x,U,w) #memanggil fungsi compcenter
  D<-euclideandist(x,V)#memanggil fungsi euclidean
  obj<-fobj(dist,U,w)#memanggil nilai fungsi
  #objektif fcm
  U_new<-compU(D,w) #memanggil fungsi compU
  output<-list(U_new=U_new,center=V,f.obj=obj)
#output
  return(output)
}
U<-initpar(2,18)
U
step1<-onestepfkm(x,U,1.5)
step1
fkmclust<-function(x,n.clust,w,max.it,tol){
  #mencetak keterangan FCM
  cat('Fuzzy C-Means Clustering (FCM); n.clust :
  ',n.clust,'\n')
  #Identifikasi struktur input dan inisialisasi
  n<-nrow(x)
  U<-initpar(n.clust,n) #memanggil fungsi initpar
  member<-NULL
  crit.term<-1
  it<-0
  stops<-FALSE
```

Lampiran 1. (Lanjutan)

```
prevobj<-0
rec.term<-NULL
#proses iterasi sampai mencapai konvergen
while (stops==FALSE){
    it<- it+1 #penghitung iterasi
    temp<-onestepfkm(x,U,w) #tahapan algoritma
    U_new<-temp$U_new #matriks partisi
    V<-temp$center #pusat cluster
    obj<-temp$f.obj #nilai objektif c-means
    crit.term<-abs(obj-prevobj)      #kriteria
pemberhentian
    rec.term<-c(rec.term,obj)
    stops<-crit.term< tol|it==max.it    #cek
kondisi berenti
    U<-U_new
    prevobj<-obj
    cat('iterasi:',it,;fn.obj:',obj,'\\n')
    }
    #pengaturan output
    colnames(V)<-NULL
    idksort<-order(V[,1])
    V<-V[idksort,]
    U_new<-U_new [idksort,]
    for(i in 1:n) {
        mem<-which.max(U_new[,i])
        member<-c(member,mem)
    }
    output<-
list(U=U_new,membership=member,center=V,
     objective.fn=obj,iteration=it,rec=rec.term)
    return(output)
}
#Panggil fungsi FCM
fkm<-fkmcclust(x,c,w,100,.001)
fkm
```

Lampiran 2. *Source code untuk Algoritma Fuzzy C-means (package Software R)*

```
#Memanggil Data
data=read.csv("R://(Letak documen.csv",sep=",",header=T)

#Membentuk data ke dalam matriks
x = as.matrix(data)

#Fungsi Fuzzy C-Means
Require (fclust)
clust=FKM(X=x,      k=    (jumlah cluster),    m=(parameter
fuzzification),   RS=1,   stand=0,   index='XB',   conv=0.001,
maxit=100,)
```

Lampiran 3. *Source Code* untuk *Classification and Regression Trees*
(*Package Rstudio*)

```
data=read.csv("R://Cluster1-  
asemhas.csv",sep=",",header=T)  
str(data)  
  
table(data$Kol)  
  
#Data Testing dan Training  
set.seed(1)  
train=sample(1:nrow(data), 0.75*nrow(data))  
  
#FUNGSI CART  
require(rpart)  
require(rpart.plot)  
data$Kol<-as.factor(data$Kol)  
dtm<-rpart(Kol~.,data=data[train,] , method="class",  
cp=0)  
dtm  
  
## menampilkan table cp  
printcp(dtm)  
cek<-data[train,]  
  
#Memunculkan Plot  
rpart.plot(dtm)  
plotcp(dtm)  
  
#melihat kebaikan model  
dataPred=predict(dtm, data[-train,], type='class')  
tabel=table(dataPred, data[-train,]$Kol)  
tabel
```

Lampiran 4. *Output Algoritma Fuzzy C-means Pada Pengelompokan Kombinasi 1*

```
> clust=FKM(X=x, k=2, m=1.5, RS=1, stand=0, index='XB', c  
onv=0.001, maxit=100,)  
trace: FKM(X = x, k = 2, m = 1.5, RS = 1, stand = 0, inde  
x = "XB",  
        conv = 0.001, maxit = 100, )  
> clust
```

Fuzzy clustering object of class 'fclust'

Number of objects:
18

Number of clusters:
2

Clustering index values:
XB k=2
0.5906148

Closest hard clustering partition:
Obj 1 Obj 2 Obj 3 Obj 4 Obj 5 Obj 6 Obj 7 Obj 8
Obj 9 Obj 10 Obj 11 Obj 12 Obj 13 Obj 14 Obj 15 Obj 16 Obj
j 17
1 2 1 2 2 2 2 2 2 1 2 1
2 2 2 1 2 2 2 2 1 2 1
Obj 18
2

Membership degree matrix (rounded):
Clus 1 Clus 2

Obj 1	0.98	0.02
Obj 2	0.17	0.83
Obj 3	0.97	0.03
Obj 4	0.16	0.84
Obj 5	0.08	0.92
Obj 6	0.20	0.80
Obj 7	0.37	0.63

Obj 8 0.75 0.25

Lampiran 4. Lanjutan

Obj 9	0.08	0.92
Obj 10	0.12	0.88
Obj 11	0.06	0.94
Obj 12	0.95	0.05
Obj 13	0.10	0.90
Obj 14	0.16	0.84
Obj 15	0.92	0.08
Obj 16	0.19	0.81
Obj 17	0.65	0.35
Obj 18	0.18	0.82

Available components:

```
[1] "U"           "H"           "F"           "clus"        "med
oid"      "value"       "criterion"   "iter"        "k"
[10] "m"          "ent"         "b"          "vp"          "del
ta"      "stand"       "Xca"        "X"          "D"
[19] "call"
```

```
> View(clust)
> clust[["U"]]
      Clus 1     Clus 2
Obj 1 0.97962307 0.02037693
Obj 2 0.16756847 0.83243153
Obj 3 0.96758047 0.03241953
Obj 4 0.16099439 0.83900561
Obj 5 0.07606099 0.92393901
Obj 6 0.19859001 0.80140999
Obj 7 0.36883485 0.63116515
Obj 8 0.75057378 0.24942622
Obj 9 0.07541478 0.92458522
Obj 10 0.12441868 0.87558132
Obj 11 0.05661768 0.94338232
Obj 12 0.95188781 0.04811219
Obj 13 0.09908397 0.90091603
Obj 14 0.15678897 0.84321103
Obj 15 0.91585616 0.08414384
Obj 16 0.18863387 0.81136613
Obj 17 0.65452295 0.34547705
```

Obj 18 0.17904468 0.82095532

Lampiran 4. Lanjutan

```
> clust[["H"]]
      X1..Character. X2..Capacity. X3..Capital. X4..Coll
  ateral. X5..Condition.
Clus 1      0.07620524   -0.05644786    0.1191174     -0.0
8511179      0.21415276
Clus 2      -0.08607654    0.07343718   -0.0905757     0.0
3198806      -0.09815071
> clust[["iter"]]
Start 1
      18
> clust[["criterion"]]
      XB k=2
0.5906148
```

Lampiran 5. *Output Algoritma Fuzzy C-means Pada Pengelompokan Kombinasi 2*

```
> clust2=FKM(X=x, k=3, m=1.5, RS=1, stand=0, index='XB',
  conv=0.001, maxit=100, )
trace: FKM(X = x, k = 3, m = 1.5, RS = 1, stand = 0, inde
x = "XB",
      conv = 0.001, maxit = 100, )
> clust2
```

Fuzzy clustering object of class 'fclust'

Number of objects:
18

Number of clusters:
3

Clustering index values:
XB k=3
0.7532068

Closest hard clustering partition:

Obj 1	Obj 2	Obj 3	Obj 4	Obj 5	Obj 6	Obj 7	Obj 8
Obj 9	Obj 10	Obj 11	Obj 12	Obj 13	Obj 14	Obj 15	Obj 16
Obj 17							
1	2	1	3	2	3	3	1
3	2	3	1	3	2	1	2
Obj 18							
3							

Membership degree matrix (rounded):

	Clus 1	Clus 2	Clus 3
Obj 1	0.98	0.01	0.01
Obj 2	0.08	0.66	0.26
Obj 3	0.87	0.03	0.10
Obj 4	0.08	0.24	0.68
Obj 5	0.02	0.87	0.11
Obj 6	0.02	0.03	0.94
Obj 7	0.11	0.11	0.78

Lampiran 5. Lanjutan

Obj 8	0.59	0.16	0.25
Obj 9	0.04	0.42	0.53
Obj 10	0.02	0.93	0.05
Obj 11	0.00	0.00	0.99
Obj 12	0.95	0.03	0.02
Obj 13	0.05	0.32	0.63
Obj 14	0.03	0.90	0.06
Obj 15	0.81	0.07	0.12
Obj 16	0.03	0.91	0.06
Obj 17	0.44	0.19	0.37
Obj 18	0.05	0.11	0.84

Available components:

```
[1] "U"           "H"           "F"           "clus"        "med
oid"      "value"       "criterion"   "iter"        "k"
[10] "m"          "ent"         "b"           "vp"          "del
ta"        "stand"       "Xca"         "X"           "D"
[19] "call"
```

```
> View(clust2)
> clust2[["U"]]
    Clus 1      Clus 2      Clus 3
Obj 1  0.984688024 0.007580361 0.007731615
Obj 2  0.081114104 0.661034435 0.257851461
Obj 3  0.874108777 0.027491483 0.098399740
Obj 4  0.080189415 0.241343800 0.678466785
Obj 5  0.021807005 0.871974157 0.106218837
Obj 6  0.022649541 0.034214314 0.943136145
Obj 7  0.110383722 0.111938931 0.777677346
Obj 8  0.591551907 0.155366508 0.253081585
Obj 9  0.044770697 0.424483050 0.530746253
Obj 10 0.019815134 0.929979062 0.050205804
Obj 11 0.001411369 0.004246686 0.994341945
Obj 12 0.950832822 0.026042328 0.023124850
Obj 13 0.054320282 0.320389629 0.625290089
Obj 14 0.031176761 0.904773849 0.064049390
Obj 15 0.811880042 0.066596405 0.121523554
Obj 16 0.033157737 0.908776792 0.058065471
```

Lampiran 5. Lanjutan

```
Obj 17 0.438893452 0.191046797 0.370059751
Obj 18 0.052244484 0.109266570 0.838488946
> clust2[["H"]]
      X1..Character. X2..Capacity. X3..Capital. X4..Coll
      ateral. X5..Condition.
Clus 1     0.10869876   -0.07802886    0.15341412     -0.0
8527651     0.25928470
Clus 2     -0.01097153    0.02694926   -0.05725675      0.1
7631332     -0.13495346
Clus 3     -0.12845707    0.08679359   -0.10974555     -0.1
1628595     -0.03403153
> clust2[["value"]]
Start 1
1.571867
> clust2[["criterion"]]
XB k=3
0.7532068
> clust2[["iter"]]
Start 1
```

37

Lampiran 6. *Output Algoritma Fuzzy C-means Pada Pengelompokan Kombinasi 3*

```
> clust3=FKM(X=x, k=4, m=1.5, RS=1, stand=0, index='XB',
  conv=0.001, maxit=100, )
trace: FKM(X = x, k = 4, m = 1.5, RS = 1, stand = 0, inde
x = "XB",
      conv = 0.001, maxit = 100, )
> clust3
```

Fuzzy clustering object of class 'fclust'

Number of objects:

18

Number of clusters:

4

Clustering index values:

XB k=4

0.4748212

Closest hard clustering partition:

Obj 1	Obj 2	Obj 3	Obj 4	Obj 5	Obj 6	Obj 7	Obj 8
Obj 9	Obj 10	Obj 11	Obj 12	Obj 13	Obj 14	Obj 15	Obj 16
Obj 17							
4	1	2	3	3	2	3	3
Obj 18							
4							

Membership degree matrix (rounded):

	Clus 1	Clus 2	Clus 3	Clus 4
Obj 1	0.98	0.00	0.01	0.00
Obj 2	0.08	0.54	0.17	0.21
Obj 3	0.15	0.03	0.81	0.01
Obj 4	0.03	0.18	0.76	0.02
Obj 5	0.01	0.87	0.07	0.04
Obj 6	0.00	0.01	0.99	0.00
Obj 7	0.02	0.04	0.93	0.02

Lampiran 6. Lanjutan

```
Obj 8      0.65   0.09   0.18   0.08
Obj 9      0.00   0.02   0.01   0.96
Obj 10     0.01   0.95   0.03   0.01
Obj 11     0.01   0.06   0.85   0.08
Obj 12     0.97   0.01   0.01   0.00
Obj 13     0.04   0.35   0.43   0.18
Obj 14     0.03   0.88   0.05   0.04
Obj 15     0.20   0.07   0.72   0.01
Obj 16     0.01   0.92   0.06   0.00
Obj 17     0.20   0.18   0.54   0.07
Obj 18     0.00   0.01   0.03   0.95
```

Available components:

```
[1] "U"          "H"          "F"          "clus"        "med
oid"         "value"       "criterion"   "iter"        "k"
[10] "m"         "ent"        "b"          "vp"          "del
ta"         "stand"       "Xca"        "X"          "D"
[19] "call"
```

```
> View(clust3)
> clust3[["U"]]
    Clus 1     Clus 2     Clus 3     Clus 4
Obj 1  0.982608575 0.004538438 0.01197601 0.000876981
Obj 2  0.083004413 0.535883552 0.16720693 0.213905106
Obj 3  0.147834175 0.033175773 0.80781158 0.011178467
Obj 4  0.034354183 0.179573952 0.76249780 0.023574066
Obj 5  0.011448272 0.870697439 0.07442041 0.043433879
Obj 6  0.001978654 0.007416954 0.98703947 0.003564918
Obj 7  0.015576344 0.038110700 0.92807213 0.018240828
Obj 8  0.651265811 0.092675291 0.17640451 0.079654382
Obj 9  0.002452128 0.023938695 0.01263078 0.960978399
Obj 10 0.005883197 0.953760938 0.03361925 0.006736611
Obj 11 0.011519110 0.057655627 0.84763596 0.083189306
Obj 12 0.974009148 0.008876434 0.01496135 0.002153071
Obj 13 0.036911963 0.352770159 0.43413485 0.176183032
Obj 14 0.030768690 0.878864678 0.05189567 0.038470962
Obj 15 0.195005608 0.071674489 0.71886074 0.014459166
Obj 16 0.010367268 0.924186336 0.06078321 0.004663184
```

Lampiran 6. Lanjutan

```
Obj 17 0.204158737 0.181834532 0.54244665 0.071560080
Obj 18 0.004800723 0.011650816 0.03006442 0.953484041
> clust3[["H"]]
      X1..Character. X2..Capacity. X3..Capital. X4..Coll
      ateral. X5..Condition.
Clus 1    0.107624788 -0.089998908   0.29535401    -0.0
5134552    0.2815348
Clus 2    0.031769118   0.017470368  -0.06031933    0.1
8941534    -0.1257212
Clus 3    0.007819685   0.008604484  -0.08759823  -0.1
4037491    0.0546257
Clus 4    -0.418074868   0.207801218  -0.17027568  -0.0
2682043    -0.1428602
> clust3[["value"]]
Start 1
1.21948
> clust3[["criterion"]]
      XB k=4
0.4748212
> clust3[["iter"]]
Start 1
31
```

Lampiran 7. *Output Algoritma Fuzzy C-means Pada Pengelompokan Kombinasi 4*

```
> clust4=FKM(X=x, k=2, m=2, RS=1, stand=0, index='XB', co  
nv=0.001, maxit=100,)  
trace: FKM(X = x, k = 2, m = 2, RS = 1, stand = 0, index  
= "XB",  
        conv = 0.001, maxit = 100, )  
> clust4
```

Fuzzy clustering object of class 'fclust'

Number of objects:
18

Number of clusters:
2

Clustering index values:
XB k=2
0.8584297

Closest hard clustering partition:
Obj 1 Obj 2 Obj 3 Obj 4 Obj 5 Obj 6 Obj 7 Obj 8
Obj 9 Obj 10 Obj 11 Obj 12 Obj 13 Obj 14 Obj 15 Obj 16 Obj
j 17
2 1 2 1 1 1 1 2 2 1 2 1 2
1 1 1 2 1 1 1 2 1 2 1 2
Obj 18
1

Membership degree matrix (rounded):

	Clus 1	Clus 2
Obj 1	0.22	0.78
Obj 2	0.65	0.35
Obj 3	0.16	0.84
Obj 4	0.58	0.42
Obj 5	0.71	0.29
Obj 6	0.54	0.46
Obj 7	0.47	0.53

Lampiran 7. Lanjutan

Obj 8	0.40	0.60
Obj 9	0.73	0.27
Obj 10	0.65	0.35
Obj 11	0.72	0.28
Obj 12	0.27	0.73
Obj 13	0.67	0.33
Obj 14	0.66	0.34
Obj 15	0.23	0.77
Obj 16	0.58	0.42
Obj 17	0.40	0.60
Obj 18	0.64	0.36

Available components:

```
[1] "U"           "H"           "F"           "clus"        "med
oid"      "value"       "criterion"   "iter"        "k"
[10] "m"          "ent"         "b"           "vp"          "del
ta"        "stand"       "Xca"         "X"           "D"
[19] "call"
```

```
> View(clust4)
> clust4[["U"]]
    Clus 1    Clus 2
Obj 1 0.2190080 0.7809920
Obj 2 0.6538347 0.3461653
Obj 3 0.1610202 0.8389798
Obj 4 0.5779704 0.4220296
Obj 5 0.7069642 0.2930358
Obj 6 0.5392815 0.4607185
Obj 7 0.4702373 0.5297627
Obj 8 0.4045525 0.5954475
Obj 9 0.7256360 0.2743640
Obj 10 0.6487118 0.3512882
Obj 11 0.7220354 0.2779646
Obj 12 0.2744688 0.7255312
Obj 13 0.6739487 0.3260513
Obj 14 0.6572740 0.3427260
Obj 15 0.2332669 0.7667331
Obj 16 0.5786711 0.4213289
Obj 17 0.4007310 0.5992690
```

Lampiran 7. Lanjutan

Obj 18 0.6359819 0.3640181

```
> clust4[["H"]]
      X1..Character. X2..Capacity. X3..Capital. X4..Coll
      ateral. X5..Condition.
Clus 1    -0.08988667    0.07126703   -0.07160733      0.0
3387923    -0.08176081
Clus 2    0.04723756   -0.03008863    0.05377173     -0.0
6836673    0.13906798
> clust4[["value"]]
Start 1
1.60722
> clust4[["criterion"]]
  XB k=2
0.8584297
> clust4[["iter"]]
Start 1
22
```

Lampiran 8. *Output Algoritma Fuzzy C-means Pada Pengelompokan Kombinasi 5*

```
> clust5=FKM(X=x, k=3, m=2, RS=1, stand=0, index='XB', co  
nv=0.001, maxit=100,)  
trace: FKM(X = x, k = 3, m = 2, RS = 1, stand = 0, index  
= "XB",  
        conv = 0.001, maxit = 100, )  
> clust5
```

Fuzzy clustering object of class 'fclust'

Number of objects:
18

Number of clusters:
3

Clustering index values:
XB k=3
0.8090936

Closest hard clustering partition:
Obj 1 Obj 2 Obj 3 Obj 4 Obj 5 Obj 6 Obj 7 Obj 8
Obj 9 Obj 10 Obj 11 Obj 12 Obj 13 Obj 14 Obj 15 Obj 16 Ob
j 17
2 3 1 2 3 2 2 1 2 3 2 1 3
Obj 18
2

Membership degree matrix (rounded):

	Clus 1	Clus 2	Clus 3
Obj 1	0.08	0.08	0.83
Obj 2	0.46	0.35	0.19
Obj 3	0.14	0.24	0.62
Obj 4	0.33	0.50	0.16
Obj 5	0.64	0.25	0.11
Obj 6	0.19	0.69	0.12
Obj 7	0.24	0.55	0.21

Lampiran 8. Lanjutan

```
Obj 8    0.24   0.31   0.45  
Obj 9    0.42   0.44   0.14  
Obj 10   0.70   0.19   0.11  
Obj 11   0.06   0.92   0.03  
Obj 12   0.13   0.13   0.74  
Obj 13   0.41   0.44   0.15  
Obj 14   0.61   0.23   0.16  
Obj 15   0.19   0.24   0.57  
Obj 16   0.67   0.19   0.14  
Obj 17   0.28   0.35   0.37  
Obj 18   0.26   0.58   0.16
```

Available components:

```
[1] "U"        "H"        "F"        "clus"      "med  
oid"       "value"     "criterion" "iter"      "k"  
[10] "m"       "ent"      "b"        "vp"        "del  
ta"        "stand"    "Xca"      "X"         "D"  
[19] "call"
```

```
> View(clust5)  
> clust5[["U"]]  
          Clus 1     Clus 2     Clus 3  
Obj 1  0.08126336 0.08382539 0.8349112  
Obj 2  0.46451625 0.34864333 0.1868404  
Obj 3  0.13780682 0.24128777 0.6209054  
Obj 4  0.33340437 0.50461084 0.1619848  
Obj 5  0.64063335 0.24756401 0.1118026  
Obj 6  0.18839138 0.68922291 0.1223857  
Obj 7  0.24381583 0.54834042 0.2078438  
Obj 8  0.24009463 0.31072860 0.4491768  
Obj 9  0.42103407 0.44025406 0.1387119  
Obj 10 0.69914941 0.18688429 0.1139663  
Obj 11 0.05798813 0.91545667 0.0265552  
Obj 12 0.13326472 0.12973155 0.7370037  
Obj 13 0.40800361 0.44258301 0.1494134  
Obj 14 0.61079570 0.23060627 0.1585980  
Obj 15 0.18609687 0.24153651 0.5723666  
Obj 16 0.66533424 0.19399131 0.1406744
```

Lampiran 8. Lanjutan

```
Obj 17 0.28043118 0.34710070 0.3724681
Obj 18 0.25833924 0.57933502 0.1623257
> clust5[["H"]]
      X1..Character. X2..Capacity. X3..Capital. X4..Coll
      ateral. X5..Condition.
Clus 1    -0.01199504    0.04077448   -0.06895538     0.1
3295978    -0.10821948
Clus 2    -0.12052157    0.06492756   -0.09156011   -0.0
9234045    -0.02319875
Clus 3    0.11023898   -0.07408635    0.15188403   -0.0
7018583    0.23921224
> clust5[["value"]]
  Start 1
1.031998
> clust5[["criterion"]]
  XB k=3
0.8090936
> clust5[["iter"]]
Start 1
  51
```

Lampiran 9. *Output Algoritma Fuzzy C-means Pada Pengelompokan Kombinasi 6*

```
> clust6=FKM(X=x, k=4, m=2, RS=1, stand=0, index='XB', co  
nv=0.001, maxit=100, )  
trace: FKM(X = x, k = 4, m = 2, RS = 1, stand = 0, index  
= "XB",  
        conv = 0.001, maxit = 100, )  
> clust6
```

Fuzzy clustering object of class 'fclust'

Number of objects:

18

Number of clusters:

4

Clustering index values:

XB k=4

0.3918585

Closest hard clustering partition:

Obj 1	Obj 2	Obj 3	Obj 4	Obj 5	Obj 6	Obj 7	Obj 8	
Obj 9	Obj 10	Obj 11	Obj 12	Obj 13	Obj 14	Obj 15	Obj 16	Obj 17
1	3	2	4	2	3	2	2	4
Obj 18								2
	1							

Membership degree matrix (rounded):

	Clus 1	Clus 2	Clus 3	Clus 4
Obj 1	0.02	0.06	0.04	0.88
Obj 2	0.31	0.24	0.32	0.14
Obj 3	0.09	0.43	0.14	0.33
Obj 4	0.11	0.57	0.22	0.10
Obj 5	0.14	0.17	0.63	0.07
Obj 6	0.06	0.83	0.08	0.04
Obj 7	0.11	0.64	0.15	0.09

Lampiran 9. Lanjutan

Obj 8	0.20	0.25	0.18	0.37
Obj 9	0.66	0.13	0.16	0.05
Obj 10	0.05	0.10	0.80	0.05
Obj 11	0.23	0.59	0.13	0.05
Obj 12	0.04	0.09	0.08	0.80
Obj 13	0.24	0.35	0.31	0.10
Obj 14	0.18	0.19	0.50	0.13
Obj 15	0.09	0.40	0.18	0.34
Obj 16	0.06	0.19	0.67	0.08
Obj 17	0.16	0.34	0.24	0.26
Obj 18	0.71	0.15	0.09	0.05

Available components:

```
[1] "U"          "H"          "F"          "clus"       "med
oid"        "value"      "criterion"   "iter"       "k"
[10] "m"         "ent"        "b"          "vp"        "del
ta"         "stand"      "Xca"        "X"         "D"
[19] "call"
```

```
> View(clust6)
> clust6[["U"]]
      Clus 1     Clus 2     Clus 3     Clus 4
Obj 1 0.02012198 0.05525683 0.03991346 0.88470773
Obj 2 0.30521102 0.23719870 0.31688564 0.14070464
Obj 3 0.08953004 0.43488422 0.14252311 0.33306262
Obj 4 0.10960819 0.57104022 0.22057557 0.09877602
Obj 5 0.13612208 0.16577780 0.63084496 0.06725516
Obj 6 0.05866774 0.82818218 0.07601647 0.03713360
Obj 7 0.11450993 0.64156928 0.14992523 0.09399556
Obj 8 0.19995439 0.25245625 0.17565251 0.37193685
Obj 9 0.66180617 0.13087428 0.15536599 0.05195356
Obj 10 0.05353267 0.10271438 0.79641798 0.04733497
Obj 11 0.22758978 0.58946528 0.12881839 0.05412655
Obj 12 0.04188268 0.08517557 0.07517727 0.79776447
Obj 13 0.23572314 0.35240877 0.31197975 0.09988834
Obj 14 0.17719779 0.18696431 0.50485875 0.13097915
Obj 15 0.08563313 0.39968997 0.17778266 0.33689423
Obj 16 0.06021451 0.18518027 0.67455825 0.08004697
```

Lampiran 9. Lanjutan

```
Obj 17 0.16276274 0.33801554 0.24324888 0.25597284
Obj 18 0.70885777 0.15017475 0.08711187 0.05385562

> clust6[["H"]]
      X1..Character. X2..Capacity. X3..Capital. X4..Coll
aateral. X5..Condition.
Clus 1    -0.35889949    0.17139314   -0.12620047    -0.0
2986350    -0.11691217
Clus 2    -0.01299162    0.01712261   -0.08688895    -0.1
3182778    0.01524617
Clus 3    0.04740391    0.01979678   -0.09388866    0.1
6921473    -0.09090697
Clus 4    0.15227781   -0.09259655    0.23413773    -0.0
4316579    0.28437475

> clust6[["value"]]
Start 1
0.7448364
> clust6[["criterion"]]
XB k=4
0.3918585
> clust6[["iter"]]
Start 1
100
```

Lampiran 10. *Output Algoritma Fuzzy C-means Pada Pengelompokan Kombinasi 7*

```
> clust7=FKM(X=x, k=2, m=2.5, RS=1, stand=0, index='XB',
  conv=0.001, maxit=100, )
trace: FKM(X = x, k = 2, m = 2.5, RS = 1, stand = 0, inde
x = "XB",
  conv = 0.001, maxit = 100, )
> clust7
```

Fuzzy clustering object of class 'fclust'

Number of objects:

18

Number of clusters:

2

Clustering index values:

XB k=2

2.285371

Closest hard clustering partition:

Obj 1	Obj 2	Obj 3	Obj 4	Obj 5	Obj 6	Obj 7	Obj 8
Obj 9	Obj 10	Obj 11	Obj 12	Obj 13	Obj 14	Obj 15	Obj 16
Obj 17							
1	2	1	2	1	1	2	2
Obj 18							
1							

Membership degree matrix (rounded):

	Clus 1	Clus 2
Obj 1	0.40	0.60
Obj 2	0.56	0.44
Obj 3	0.35	0.65
Obj 4	0.52	0.48
Obj 5	0.58	0.42
Obj 6	0.49	0.51
Obj 7	0.47	0.53

Lampiran 10. Lanjutan

```
Obj 8    0.47  0.53
Obj 9    0.58  0.42
Obj 10   0.55  0.45
Obj 11   0.59  0.41
Obj 12   0.42  0.58
Obj 13   0.56  0.44
Obj 14   0.56  0.44
Obj 15   0.39  0.61
Obj 16   0.53  0.47
Obj 17   0.46  0.54
Obj 18   0.55  0.45
```

Available components:

```
[1] "U"        "H"        "F"        "clus"      "med
oid"      "value"     "criterion" "iter"      "k"
[10] "m"        "ent"      "b"        "vp"        "del
ta"       "stand"     "Xca"      "X"        "D"
[19] "call"
```

```
> View(clust7)
> clust7[["U"]]
      Clus 1   Clus 2
Obj 1  0.3971072 0.6028928
Obj 2  0.5623850 0.4376150
Obj 3  0.3535804 0.6464196
Obj 4  0.5192769 0.4807231
Obj 5  0.5760760 0.4239240
Obj 6  0.4933947 0.5066053
Obj 7  0.4739386 0.5260614
Obj 8  0.4670220 0.5329780
Obj 9  0.5840265 0.4159735
Obj 10 0.5534024 0.4465976
Obj 11 0.5904194 0.4095806
Obj 12 0.4201778 0.5798222
Obj 13 0.5602064 0.4397936
Obj 14 0.5628741 0.4371259
Obj 15 0.3910343 0.6089657
Obj 16 0.5254447 0.4745553
```

Lampiran 10. Lanjutan

```
Obj 17 0.4556965 0.5443035
Obj 18 0.5477317 0.4522683
> clust7[["H"]]
  X1..Character. X2..Capacity. X3..Capital. X4..Coll
  ateral. X5..Condition.
Clus 1 -0.065044314 0.0511880590 -0.04109577 0.0
1496431 -0.04034295
Clus 2 0.008825759 -0.0003668905 0.01582092 -0.0
4414648 0.07472198
> clust7[["value"]]
Start 1
1.155457
> clust7[["criterion"]]
  XB k=2
2.285371
> clust7[["iter"]]
Start 1
42
```

Lampiran 11. *Output* Algoritma *Fuzzy C-means* Pada Pengelompokan Kombinasi 8

```
> clust8=FKM(X=x, k=3, m=2.5, RS=1, stand=0, index='XB',
  conv=0.001, maxit=100, )
trace: FKM(X = x, k = 3, m = 2.5, RS = 1, stand = 0, inde
x = "XB",
      conv = 0.001, maxit = 100, )
> clust8
```

Fuzzy clustering object of class 'fclust'

Number of objects:
18

Number of clusters:
3

Clustering index values:
XB k=3
1.109287

Closest hard clustering partition:

Obj 1	Obj 2	Obj 3	Obj 4	Obj 5	Obj 6	Obj 7	Obj 8
Obj 9	Obj 10	Obj 11	Obj 12	Obj 13	Obj 14	Obj 15	Obj 16
j 17							
1	2	1	3	2	3	3	1
3	2	3	1	3	2	1	2
Obj 18							
3							

Membership degree matrix (rounded):

	Clus 1	Clus 2	Clus 3
Obj 1	0.59	0.21	0.20
Obj 2	0.24	0.41	0.35
Obj 3	0.58	0.18	0.23
Obj 4	0.22	0.37	0.40
Obj 5	0.21	0.47	0.33
Obj 6	0.23	0.28	0.49
Obj 7	0.29	0.29	0.41

Lampiran 11. Lanjutan

Obj 8	0.39	0.29	0.32
Obj 9	0.21	0.39	0.40
Obj 10	0.22	0.49	0.29
Obj 11	0.11	0.19	0.70
Obj 12	0.53	0.24	0.23
Obj 13	0.22	0.39	0.39
Obj 14	0.23	0.47	0.30
Obj 15	0.51	0.23	0.26
Obj 16	0.23	0.51	0.26
Obj 17	0.38	0.30	0.33
Obj 18	0.24	0.31	0.45

Available components:

```
[1] "U"           "H"           "F"           "clus"        "med
oid"      "value"       "criterion"   "iter"        "k"
[10] "m"          "ent"         "b"           "vp"          "del
ta"       "stand"       "Xca"         "X"           "D"
[19] "call"
```

```
> View(clust8)
> clust8[["U"]]
    Clus 1     Clus 2     Clus 3
Obj 1 0.5887019 0.2078523 0.2034459
Obj 2 0.2387655 0.4102822 0.3509523
Obj 3 0.5826731 0.1833777 0.2339492
Obj 4 0.2240188 0.3726173 0.4033639
Obj 5 0.2077037 0.4672281 0.3250681
Obj 6 0.2320838 0.2800237 0.4878925
Obj 7 0.2915725 0.2934284 0.4149990
Obj 8 0.3886752 0.2907584 0.3205664
Obj 9 0.2105692 0.3860062 0.4034246
Obj 10 0.2202511 0.4938433 0.2859057
Obj 11 0.1139469 0.1863269 0.6997261
Obj 12 0.5268663 0.2407761 0.2323576
Obj 13 0.2234499 0.3856851 0.3908650
Obj 14 0.2317913 0.4709272 0.2972814
Obj 15 0.5114986 0.2332947 0.2552067
Obj 16 0.2306970 0.5055277 0.2637752
```

Lampiran 11. Lanjutan

```
Obj 17 0.3755646 0.2990980 0.3253374
Obj 18 0.2367924 0.3103167 0.4528909

> clust8[["H"]]
      X1..Character. X2..Capacity. X3..Capital. X4..Coll
aateral. X5..Condition.
Clus 1      0.07840292   -0.05126694    0.08944156     -0.0
7081195      0.19026003
Clus 2      -0.02719622    0.04291608   -0.04684532     0.0
8225069      -0.08220609
Clus 3      -0.10993762    0.05975969   -0.07509331     -0.0
5868981      -0.02372553

> clust8[["value"]]
Start 1
0.6232148
> clust8[["criterion"]]
XB k=3
1.109287
> clust8[["iter"]]
Start 1
99
```

Lampiran 12. *Output Algoritma Fuzzy C-means Pada Pengelompokan Kombinasi 9*

```
> clust9=FKM(X=x, k=4, m=2.5, RS=1, stand=0, index='XB',
  conv=0.001, maxit=100, )
trace: FKM(X = x, k = 4, m = 2.5, RS = 1, stand = 0, inde
x = "XB",
  conv = 0.001, maxit = 100, )
> clust9
```

Fuzzy clustering object of class 'fclust'

Number of objects:

18

Number of clusters:

4

Clustering index values:

XB k=4

0.8827794

Closest hard clustering partition:

Obj 1	Obj 2	Obj 3	Obj 4	Obj 5	Obj 6	Obj 7	Obj 8
Obj 9	Obj 10	Obj 11	Obj 12	Obj 13	Obj 14	Obj 15	Obj 16
j 17							
2	3	4	3	4	1	2	2
Obj 18	1	2	3	4	1	3	1
	2						3
							4

Membership degree matrix (rounded):

	Clus 1	Clus 2	Clus 3	Clus 4
Obj 1	0.08	0.09	0.72	0.11
Obj 2	0.27	0.27	0.16	0.30
Obj 3	0.18	0.25	0.32	0.25
Obj 4	0.14	0.18	0.08	0.59
Obj 5	0.49	0.20	0.11	0.20
Obj 6	0.19	0.37	0.11	0.33
Obj 7	0.21	0.34	0.15	0.30

Lampiran 12. Lanjutan

Obj 8	0.19	0.25	0.30	0.26
Obj 9	0.29	0.31	0.13	0.26
Obj 10	0.56	0.15	0.10	0.19
Obj 11	0.05	0.82	0.03	0.10
Obj 12	0.14	0.13	0.57	0.16
Obj 13	0.29	0.27	0.13	0.31
Obj 14	0.35	0.21	0.16	0.28
Obj 15	0.19	0.22	0.33	0.25
Obj 16	0.40	0.18	0.13	0.29
Obj 17	0.24	0.26	0.23	0.27
Obj 18	0.20	0.39	0.14	0.26

Available components:

```
[1] "U"          "H"          "F"          "clus"        "med
oid"      "value"      "criterion"   "iter"        "k"
[10] "m"         "ent"        "b"          "vp"          "del
ta"       "stand"      "Xca"        "X"          "D"
[19] "call"
```

```
> View(clust9)
> clust9[["U"]]
      Clus 1     Clus 2     Clus 3     Clus 4
Obj 1  0.08345603 0.08568803 0.72498644 0.1058695
Obj 2  0.26919812 0.27000832 0.16034184 0.3004517
Obj 3  0.17602810 0.25064093 0.32088832 0.2524426
Obj 4  0.13842999 0.18496905 0.08341603 0.5931849
Obj 5  0.49197221 0.19711130 0.10632903 0.2045875
Obj 6  0.18591939 0.36985782 0.11037938 0.3338434
Obj 7  0.20643872 0.34166380 0.15217134 0.2997261
Obj 8  0.18741983 0.24554785 0.30415664 0.2628757
Obj 9  0.29145020 0.31401345 0.13004209 0.2644943
Obj 10 0.55769534 0.15363867 0.10223348 0.1864325
Obj 11 0.05215736 0.81934497 0.02614359 0.1023541
Obj 12 0.13621944 0.13486839 0.57375342 0.1551588
Obj 13 0.29101431 0.27101259 0.13007228 0.3079008
Obj 14 0.35016591 0.21075600 0.16025423 0.2788239
Obj 15 0.19429514 0.22165920 0.33350862 0.2505370
Obj 16 0.39895677 0.17636994 0.13494068 0.2897326
```

Lampiran 12. Lanjutan

```
Obj 17 0.23681419 0.25618675 0.23405611 0.2729430
Obj 18 0.20494940 0.39296295 0.13754840 0.2645393
> clust9[["H"]]
      X1..Character. X2..Capacity. X3..Capital. X4..Coll
      ateral. X5..Condition.
Clus 1     0.01159861    0.04191877 -0.118213630      0.1
4855963   -0.079509732
Clus 2     -0.15128952    0.05083722 -0.093156026     -0.0
8533276   -0.002367073
Clus 3     0.15612587    -0.09844098  0.213758629     -0.0
5230024   0.264413432
Clus 4     -0.03235688   0.05022206 -0.005763641     -0.0
4902831   -0.047105758
> clust9[["value"]]
Start 1
0.398877
> clust9[["criterion"]]
      XB k=4
0.8827794
> clust9[["iter"]]
Start 1
94
```

Lampiran 13. Output Algortima CART Pada Cluster 1

```
> data=read.csv("R://cluster1-asemhas.csv",sep=",",header=T)
> str(data)
'data.frame': 28 obs. of 6 variables:
 $ Kol      : int  1 1 1 1 1 1 1 1 1 ...
 $ CHARACTER : num -0.42 -0.42 -0.42 -0.42 -0.238 ...
 ...
 $ CAPACITY  : num  0.0522 0.2661 0.2302 -0.1313 0.2535 ...
 $ CAPITAL   : num  0.283 -0.154 0.283 0.283 0.283 ...
 ...
 $ COLLATERAL: num  0.752 -0.353 0.752 -0.353 -0.353 ...
 ...
 $ CONDITION  : num -0.588 -0.588 -0.588 -0.588 -0.588 ...
> table(data$Kol)

 1 2
20 8

> #Data Testing dan Training
> set.seed(1)
> train=sample(1:nrow(data), 0.75*nrow(data))
> #FUNGSI CART
> require(rpart)
> require(rpart.plot)
> data$Kol<-as.factor(data$Kol)
> dtm<-rpart(Kol~.,data=data[train,] , method="class")
> dtm
n= 21

node), split, n, loss, yval, (yprob)
      * denotes terminal node

1) root 21 6 1 (0.71428571 0.28571429)
   2) CHARACTER< -0.4192 12 1 1 (0.91666667 0.08333333
3) *
   3) CHARACTER>=-0.4192 9 4 2 (0.44444444 0.55555556
) *
> #MEMUNCULKAN PLOT
> plotcp(dtm)
```

Lampiran 13. Lanjutan

```
> rpart.plot(dtm)
> printcp(dtm)

Classification tree:
rpart(formula = Kol ~ ., data = data[train, ], method
d = "class",
    cp = 0)

Variables actually used in tree construction:
[1] CHARACTER

Root node error: 6/21 = 0.28571

n= 21

      CP nsplit rel_error xerror      xstd
1 0.16667      0   1.00000   1 0.34503
2 0.00000      1   0.83333   1 0.34503

> #melihat kebaikan model
> dataPred=predict(dtm, data[-train,], type='class')
> tabel=table(dataPred, data[-train,]$Kol)
> tabel

dataPred 1 2
      1 3 1
      2 2 1
```

Lampiran 14. *Output* Algortima CART Pada Cluster 2

```
> data=read.csv("R://Cluster2-asemhas.csv",sep=",",header=T)
> str(data)
'data.frame':    70 obs. of  6 variables:
 $ Kol        : int  1 1 1 1 1 1 1 1 1 ...
 $ CHARACTER : num  -0.42 -0.42 -0.238 -0.42 0.369 ...
 $ CAPACITY   : num  0.0522 0.0522 0.4101 0.2535 -0.4559 ...
 ..
$ CAPITAL    : num  -0.154 -0.154 0.283 -0.154 0.299 ...
$ COLLATERAL: num  -0.353 -0.353 -0.353 -0.353 0.752 ...
$ CONDITION  : num  -0.588 -0.588 -0.588 -0.588 1.021 ...
> table(data$Kol)

 1  2
50 20

> #Data Testing dan Training
> set.seed(1)
> train=sample(1:nrow(data), 0.75*nrow(data))
> #FUNGSI CART
> require(rpart)
> require(rpart.plot)
> data$Kol<-as.factor(data$Kol)
> dtm<-rpart(Kol~.,data=data[train,] , method="class")
> dtm
n= 52

node), split, n, loss, yval, (yprob)
  * denotes terminal node

1) root 52 18 1 (0.6538462 0.3461538)
  2) CONDITION>=-0.8042 40  6 1 (0.8500000 0.1500000) *
  3) CONDITION< -0.8042 12  0 2 (0.0000000 1.0000000) *
> ## menampilkan table cp
> printcp(dtm)

Classification tree:
rpart(formula = Kol ~ ., data = data[train, ], method = "class")
```

Lampiran 14. Lanjutan

```
Variables actually used in tree construction:  
[1] CONDITION
```

```
Root node error: 18/52 = 0.34615
```

```
n= 52
```

```
          CP nsplit rel error  xerror     xstd  
1 0.66667      0    1.00000 1.00000 0.19059  
2 0.01000      1    0.33333 0.33333 0.12799  
> cek<-data[train,]  
> #MEMUNCULKAN PLOT  
> rpart.plot(dtm)  
> plotcp(dtm)  
> #melihat kebaikan model  
> dataPred=predict(dtm, data[-train,], type='class')  
> tabel=table(dataPred, data[-train,]$Kol)  
> tabel
```



```
dataPred  1  2  
       1 16  2  
       2  0  0
```

Lampiran 15. *Output* Algortima CART Pada Cluster 3

```
> data=read.csv("R://Cluster3-asemhas.csv",sep=",",header=T)
> str(data)
'data.frame':   98 obs. of  6 variables:
 $ Kol        : int  1 1 1 1 1 1 1 1 1 ...
 $ CHARACTER : num  1.578 1.268 -0.42 0.366 -0.42 ...
 $ CAPACITY  : num  -0.5361 -0.3937 0.0522 -0.2305 0.1 ...
 .
$ CAPITAL    : num  0.117 0.461 -0.592 0.461 -0.154 ...
$ COLLATERAL: num  -0.9219 0.7519 -0.3528 0.0824 0.7519
...
$ CONDITION  : num  2.678 1.986 -0.588 1.986 -0.588 ...
> table(data$Kol)

 1  2
70 28

> #Data Testing dan Training
> set.seed(1)
> train=sample(1:nrow(data), 0.75*nrow(data))
> #FUNGSI CART
> require(rpart)
Loading required package: rpart
> require(rpart.plot)
Loading required package: rpart.plot
> data$Kol<-as.factor(data$Kol)
> dtm<-rpart(Kol~,data=data[train,] , method="class")
> dtm
n= 73

node), split, n, loss, yval, (yprob)
      * denotes terminal node

1) root 73 23 1 (0.68493151 0.31506849)
  2) CHARACTER< -0.2351 40  4 1 (0.90000000 0.10000000) *
  3) CHARACTER>=-0.2351 33 14 2 (0.42424242 0.57575758)
    6) CHARACTER>=0.48775 16  3 1 (0.81250000 0.18750000)
*
  7) CHARACTER< 0.48775 17  1 2 (0.05882353 0.94117647)
*
```

Lampiran 15. Lanjutan

```
> rpart.plot(dtm)
Warning message:
Bad 'data' field in model 'call' (expected a data.frame or
a matrix).
To silence this warning:
  Call rpart.plot with roundint=FALSE,
  or rebuild the rpart model with model=TRUE.
> plotcp(dtm)
> dtm<-rpart(Kol~.,data=data[train,] , method="class", cp
=0)
> dtm
n= 73

node), split, n, loss, yval, (yprob)
      * denotes terminal node

1) root 73 23 1 (0.68493151 0.31506849)
  2) CHARACTER< -0.2351 40  4 1 (0.90000000 0.10000000) *
  3) CHARACTER>=-0.2351 33 14 2 (0.42424242 0.57575758)
    6) CHARACTER>=0.48775 16  3 1 (0.81250000 0.18750000)
*
  7) CHARACTER< 0.48775 17  1 2 (0.05882353 0.94117647)
*
> rpart.plot(dtm)
> plotcp(dtm)
> #melihat kebaikan model
> dataPred=predict(dtm, data[-train,], type='class')
> tabel=table(dataPred, data[-train,]$Kol)
> tabel
```

```
dataPred  1  2
      1 20  1
      2  0  4
```

Lampiran 15. Lanjutan

```
> printcp(dtm)
```

```
Classification tree:  
rpart(formula = Kol ~ ., data = data[train, ], method = "  
class",  
      cp = 0)
```

```
Variables actually used in tree construction:  
[1] CHARACTER
```

```
Root node error: 23/73 = 0.31507
```

```
n= 73
```

	CP	nsplit	rel error	xerror	xstd
1	0.32609	0	1.00000	1.00000	0.17257
2	0.00000	2	0.34783	0.56522	0.14212

Lampiran 16. *Output* Algortima CART Pada Cluster 4

```
> data=read.csv("R://Cluster4-asemhas.csv",sep=",",header=T)
> str(data)
'data.frame':   42 obs. of  6 variables:
 $ Kol      : int  1 1 1 1 1 1 1 1 1 ...
 $ CHARACTER : num  0.369 0.596 -0.42 1.356 -0.42 ...
 $ CAPACITY : num  -0.2767 -0.6338 -0.1098 -0.2033 0.052
2 ...
 $ CAPITAL  : num  0.117 0.117 -0.154 0.299 0.283 ...
 $ COLLATERAL: num  -0.922 0.752 -0.353 -0.922 0.752 ...
 $ CONDITION : num  1.021 1.021 -0.588 1.021 -0.588 ...
> table(data$Kol)

 1  2
30 12

> set.seed(1)
> train=sample(1:nrow(data), 0.75*nrow(data))
> require(rpart)
> require(rpart.plot)
> dtm<-rpart(Kol~ CHARACTER+CAPACITY+CAPITAL+COLLATERAL+
ONDITION,data=data[train,] , method="class")
> plot=rpart.plot(dtm, type=4, digits=5, fallen.leaves=FA
LSE, shadow.col="gray", box.palette="BuGn", roundint = FA
LSE)
> plot
> dataPred=predict(dtm, data[-train,], type='class')
> tabel=table(dataPred, data[-train,]$Kol)
> tabel

dataPred 1 2
      1 5 2
      2 2 2
```

Lampiran 16. Output Algortima CART Pada Cluster 3

```
> summary(dt)  
Call:  
rpart(formula = Kol ~ CHARACTER + CAPACITY + CAPITAL + CO  
LLATERAL +  
    CONDITION, data = data[train, ], method = "class")  
n= 31  
  
          CP nsplit rel error xerror      xstd  
1 0.25      0     1.00  1.00 0.3045356  
2 0.01      2     0.50  1.25 0.3253410  
  
Variable importance  
CONDITION      CAPITAL   CHARACTER COLLATERAL  
        28         25         21         15  
CAPACITY  
        12  
  
Node number 1: 31 observations, complexity param=0.25  
predicted class=1 expected loss=0.2580645 P(node) =1  
  class counts: 23     8  
  probabilities: 0.742 0.258  
left son=2 (10 obs) right son=3 (21 obs)  
Primary splits:  
  CAPITAL < 0.2224 to the right, improve=1.96620  
60, (0 missing)  
  CONDITION < -0.11225 to the left, improve=1.85387  
40, (0 missing)  
  CAPACITY < -0.16145 to the right, improve=1.31642  
20, (0 missing)  
  CHARACTER < -0.19035 to the left, improve=0.98500  
28, (0 missing)  
  COLLATERAL < -0.1813 to the left, improve=0.11028  
40, (0 missing)  
Surrogate splits:  
  CONDITION < 1.8667 to the right, agree=0.774, adj  
=0.3, (0 split)  
  CHARACTER < 0.8817 to the right, agree=0.742, adj  
=0.2, (0 split)
```

Lampiran 16. Lanjutan

```
CAPACITY < -0.7608 to the left, agree=0.742,
adj=0.2, (0 split)

Node number 2: 10 observations
predicted class=1 expected loss=0 P(node) =0.3225806
  class counts: 10 0
  probabilities: 1.000 0.000

Node number 3: 21 observations, complexity param=0.25
predicted class=1 expected loss=0.3809524 P(node) =0.
6774194
  class counts: 13 8
  probabilities: 0.619 0.381
  left son=6 (13 obs) right son=7 (8 obs)
  Primary splits:
    CONDITION < -0.11225 to the left, improve=3.52014
70, (0 missing)
    CHARACTER < -0.19035 to the left, improve=2.33333
30, (0 missing)
    CAPACITY < -0.1567 to the right, improve=1.53937
70, (0 missing)
    COLLATERAL < -0.1813 to the left, improve=0.76190
48, (0 missing)
    CAPITAL < -0.188 to the right, improve=0.36630
04, (0 missing)
  Surrogate splits:
    CHARACTER < -0.10655 to the left, agree=0.905, ad
j=0.750, (0 split)
    COLLATERAL < -0.4157 to the right, agree=0.857, ad
j=0.625, (0 split)
    CAPITAL < -0.13615 to the left, agree=0.810, ad
j=0.500, (0 split)
    CAPACITY < -0.11895 to the right, agree=0.762, ad
j=0.375, (0 split)

Node number 6: 13 observations
predicted class=1 expected loss=0.1538462 P(node) =0.
4193548
  class counts: 11 2
```

Lampiran 16. Lanjutan

```
probabilities: 0.846 0.154

Node number 7: 8 observations
  predicted class=2  expected loss=0.25  P(node) =0.25806
45
  class counts:     2      6
  probabilities: 0.250 0.750
> printcp(dtm)

Classification tree:
rpart(formula = Kol ~ ., data = data[train, ], method = "
class",
  cp = 0)

Variables actually used in tree construction:
[1] CAPITAL    CONDITION

Root node error: 8/31 = 0.25806

n= 31

      CP nsplit rel error xerror      xstd
1 0.25      0     1.00  1.00 0.30454
2 0.00      2     0.50  1.25 0.32534
```