TESIS

**PENGGABUNGAN METODE *U-CONTROL CHART* DAN METODE
AUTOMATIC CLUSTERING DIFFERENTIAL EVOLUTION UNTUK
PENENTUAN JUMLAH KLASSTER PADA METODE K-MEANS**

Oleh:

AHMAD ILHAM

P31.2015.01851

**Tesis Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar
Magister Komputer**

**PROGRAM MAGISTER TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS DIAN NUSWANTORO
SEMARANG**

2017

UNIVERSITAS DIAN NUSWANTORO

PENGESAHAN STATUS TESIS

JUDUL : PENGABUNGAN METODE KONTROL U-CHART DAN METODE
DIFFERENTIAL EVOLUTION CLUSTERING UNTUK PENENTUAN
JUMLAH KLASER PADA METODE K-MEANS

SAYA : AHMAD ILHAM

Mengijinkan Tesis Magister Komputer ini disimpan di Perpustakaan Universitas
Dian Nuswantoro dengan syarat-syarat kegunaan sebagai berikut:

1. Tesis adalah hak milik Universitas Dian Nuswantoro
2. Perpustakaan Universitas Dian Nuswantoro dibenarkan membuat Salinan untuk tujuan referensi saja
3. Perpustakaan juga dibenarkan membuat Salinan Tesis ini sebagai bahan pertukaran antar institusi pendidikan tinggi
4. Berikan tanda ☒ sesuai dengan kategori Tesis

☐ Sangat Rahasia

☐ Rahasia

☐ Biasa

Ahmad Ilham

Alamat tetap:
Perumahan Beringin Lestari Selatan
Blok F 138 RT2/ RW10
Kelurahan Gondorio, Ngalian Semarang

Tanggal: 13 Nopember 2017

Disahkan oleh:

Rendi Satria Wahono, Ph.D

Tanggal: 25 November 2017

UNIVERSITAS DIAN NUSWANTORO

PERNYATAAN PENULIS

JUDUL : PENGGABUNGAN METODE KONTROL U-CHART DAN
METODE DIFFERENTIAL EVOLUTION CLUSTERING
UNTUK PENENTUAN JUMLAH KLASSTER PADA METODE
K-MEANS

PENULIS : AHMAD ILHAM

NPM : P31.2015.01851

“Saya menyatakan dan bertanggung jawab dengan sebenarnya bahwa Tesis ini adalah hasil karya saya sendiri kecuali cuplikan dan ringkasan yang masing-masing telah saya jelaskan sumbernya. Jika pada waktu selanjutnya ada pihak lain yang mengklaim bahwa Tesis ini sebagai karyanya, yang disertai dengan bukti-bukti yang cukup, maka saya bersedia untuk dibatalkan gelar Magister Komputer saya beserta segala hak dan kewajiban yang telah melekat pada gelar tersebut.”

Semarang, 13 Nopember 2017

AHMAD ILHAM

Penulis

UNIVERSITAS DIAN NUSWANTORO

PERSETUJUAN TESIS

JUDUL : PENGGABUNGAN METODE KONTROL U-CHART DAN METODE
DIFFERENTIAL EVOLUTION CLUSTERING UNTUK PENENTUAN
JUMLAH KLASTER PADA METODE K-MEANS

NAMA : AHMAD ILHAM

NPM : P31.2015.01851

Tesis ini telah diperiksa dan disetujui,

Semarang, 1 Nopember 2017

Romi Satria Wahono, Ph.D
Pembimbing Utama

Catur Supriyanto, M.CS
Pembimbing Pembantu

UNIVERSITAS DIAN NUSWANTORO

PENGESAHAN TESIS

JUDUL : PENGGABUNGAN METODE KONTROL U-CHART DAN METODE
DIFFERENTIAL EVOLUTION CLUSTERING UNTUK PENENTUAN
JUMLAH KLASSTER PADA METODE K-MEANS

NAMA : AHMAD ILHAM

NPM : P31.2015.01851

Tesis ini telah diujikan dan dipertahankan di hadapan Dewan Penguji pada Sidang Tesis tanggal 03 Nopember 2017. Menurut pandangan kami, Tesis ini memadai dari segi kualitas untuk tujuan penganugrahan gelar Magister Komputer (M.Kom.)

Semarang, 13 Nopember 2017

Dewan Penguji:

A Zaenul Fanani, S.Si, M.Kom
Anggota

Aris Marjuni, S.Si, M.Kom
Anggota

Abdul Syukur
Dekan

Dr. Pujiono S.Si, M.Kom
Ketua Penguji

ABSTRAK

Penentuan jumlah kluster k-Means adalah masalah utama yang paling populer di kalangan peneliti data mining karena sulitnya menentukan informasi dari data secara apriori akibatnya dimungkinkan hasil kluster tidak optimal dan cepat terjebak ke dalam minimum lokal. Metode pengklasteran otomatis dengan pendekatan *evolutionary computation* (EC) dapat menyelesaikan masalah k-Means. Metode *automatic clustering differential evolution* (ACDE) adalah salah satu metode pendekatan EC yang terkenal karena dapat menangani data berdimensi tinggi dan meningkatkan kinerja pengklasteran k-Means dengan nilai validitas kluster yang rendah. Namun, proses penentuan ambang batas aktivasi k pada ACDE masih bergantung pada pertimbangan pengguna sehingga proses penentuan jumlah kluster k-Means belum efisien. Pada penelitian ini, masalah ACDE akan diperbaiki menggunakan metode *u-control chart* (UCC) yang terbukti efisien digunakan untuk mengatasi masalah penentuan jumlah kluster k-Means secara otomatis. Metode yang diusulkan dievaluasi menggunakan kumpulan data terkini seperti data sintetik dan data real (iris, glass, wine, vowel, ruspini) dari repositori UCI serta menggunakan *davies bouldin index* (DBI) dan *cosine similarity measure* (CS) sebagai metode evaluasinya. Hasil dari penelitian ini menunjukkan bahwa metode UCC berhasil meningkatkan metode k-Means dengan nilai objektif DBI dan CS terendah masing-masing sebesar 0.470 dan 0.577. Nilai objektif DBI dan CS terendah adalah metode terbaik. Metode yang diusulkan memiliki kinerja pengklasteran lebih unggul setelah dibandingkan dengan metode terkini lainnya seperti metode *genetic clustering for unknown k* (GCUK), *dynamic clustering pso* (DCPSO) dan *automatic clustering approach based on differential evolution algorithm combining with k-Means for crisp clustering* (ACDE) untuk hampir seluruh evaluasi fungsi objektif DBI dan CS. Dapat disimpulkan bahwa, metode UCC mampu memperbaiki kelemahan metode ACDE pada penentuan jumlah kluster k-Means dengan menentukan ambang batas aktivasi k secara otomatis.

Kata kunci: k-means, *differential evolution*, *u-control chart*, ambang batas aktivasi k , penentuan jumlah kluster, pengklasteran otomatis

UCAPAN TERIMA KASIH

Puji dan syukur penulis panjatkan kepada Tuhan YME yang telah melimpahkan rahmat dan karuniaNya sehingga pada akhirnya penulis dapat menyelesaikan tesis yang berjudul “Penggabungan Metode *u-Control Chart* dan Metode *Automatic Clustering Differential Evolution* untuk Penentuan Jumlah Klaster pada Metode k-Means”. Tesis ini diajukan dalam rangka menempuh ujian akhir untuk memperoleh gelar Magister Komputer (M.Kom) pada Program Pasca Sarjana (S2) Program Studi Teknik Informatika di Universitas Dian Nuswantoro.

Penulis menyadari bahwa tanpa dukungan dan dorongan dari semua pihak, maka penulisan tesis ini tidak dapat selesai dengan baik, oleh karena itu pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Ibu Hj. Syuhada Yusuf dan Bapak H. Ilham Muis selaku orang tua yang selalu mendoakan, mendukung, dan membimbing penulis sehingga tesis ini dapat terselesaikan dengan baik.
2. Istri saya Lalelatul Khikmah dan kedua anak saya Muhammad Zakaria Al Fath dan Maryam Syifa Al Wahda yang telah mendoakan dan mendukung penulis hingga terselesaikan penyusunan tesis ini.
3. Bapak Prof. Dr. Ir. Edi Noersasongko, M.Kom selaku Rektor Universitas Dian Nuswantoro.
4. Bapak Dr. Abdul Syukur selaku Dekan Fakultas Ilmu Komputer Universitas Dian Nuswantoro.
5. Bapak Romi Satria Wahono, Ph.D sebagai pembimbing utama yang selalu memberikan ilmu, membimbing dan mendukung penulis dalam menyelesaikan tesis ini.
6. Bapak Catur Supriyanto, M.CS selaku pembimbing kedua tesis ini.
7. Teman-teman dalam Intelligent Systems Research Group dibawah asuhan oleh Romi Satria Wahono, Ph.D (Bapak Adi Wijaya, Bapak Deden Istiawan, Bapak Joko Suntoro dan kawan-kawan) yang telah banyak membantu dalam diskusi.
8. Teman-teman MTI angkatan XXVIII.

9. Serta semua pihak yang telah banyak membantu dan terlibat dalam pembuatan tesis ini yang tidak mungkin penulis sebutkan satu persatu.

Penulis menyadari bahwa tesis ini masih jauh dari sempurna, oleh karena itu kritik dan saran yang bersifat membangun sangat penulis harapkan guna kesempurnaan penelitian yang akan datang.

Semarang, 13 November 2017

Ahmad Ilham

DAFTAR ISI

HALAMAN UTAMA	i
PENGESAHAN STATUS TESIS.....	ii
PERNYATAAN PENULIS.....	iii
PERSETUJUAN TESIS	iv
PENGESAHAN TESIS.....	v
ABSTRACT	vi
ABSTRAK.....	vii
UCAPAN TERIMA KASIH.....	viii
DAFTAR ISI	x
DAFTAR GAMBAR.....	xiv
DAFTAR TABEL	xvi
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Identifikasi Masalah.....	6
1.3 Rumusan Masalah.....	6
1.4 Tujuan Penelitian.....	6
1.5 Manfaat Penelitian.....	6
1.6 Sistematika Penulisan	7
BAB 2 TINJAUAN PUSTAKA	8
2.1 Pendahuluan	8
2.2 Metode <i>Systematic Literature Review</i> (SLR).....	8
2.2.1 Pertanyaan Penelitian.....	11
2.2.2 Strategi Pencarian	12
2.2.3 Seleksi Studi Utama.....	13
2.2.4 Ekstraksi Data.....	15

2.2.5	Nilai Kualitas Penelitian dan Sintetis Data	15
2.3	Publikasi Jurnal Paling Signifikan dalam Topik Penentuan Jumlah Klaster pada Metode k-Means.....	15
2.4	Peneliti Paling Aktif dalam Topik Penentuan Jumlah Klaster pada Metode k-Means	18
2.5	Dataset yang Digunakan dalam Topik Penentuan Jumlah Klaster pada Metode k-Means.....	18
2.6	Strategi Pendekatan Metode dalam Topik Penentuan Jumlah Klaster pada Metode k-Means.....	21
2.6.1	Metode <i>Traditional-Based</i> (TB).....	21
2.6.2	Metode <i>Merge-Split Based</i> (MS).....	22
2.6.3	Metode <i>Evolutionary Computation Based</i> (EC).....	22
2.7	Metode-metode yang Pernah Diusulkan dalam Topik Penentuan Jumlah Klaster pada Metode k-Means.....	23
2.7.1	Metode MacQueen.....	23
2.7.2	Metode Sarkar et al.	24
2.7.3	Metode Bandyopadhyay dan Maulik.....	27
2.7.4	Metode Omran et al.....	29
2.7.5	Metode Kuo et al.....	30
2.7.6	Metode Kuo dan Sulvia.....	33
2.7.7	Metode Xiao et al.....	35
2.7.8	Metode Das et al.	37
2.7.9	Metode Kuo et al.....	39
2.8	Metode-metode yang Paling Banyak Digunakan dalam Topik Penentuan Jumlah Klaster pada Metode k-Means.....	41
2.8.1	Metode Bandyopadhyay dan Maulik	42
2.8.2	Metode Omran et al.....	44

2.8.3	Metode Das et al.	45
2.9	Daftar Referensi <i>Systematic Literature Review</i> dan Peta Pikiran Lengkap.....	48
BAB 3	METODE PENELITIAN	51
3.1	Perancangan Penelitian	51
3.2	Analisis Masalah dan Tinjauan Pustaka	51
3.3	Pengumpulan Dataset.....	52
3.4	Metode yang Diusulkan dalam Topik Penentuan Jumlah Klaster pada Metode k-Means.....	53
3.5	Kerangka Pemikiran	67
3.6	Eksperimen dan Pengujian Metode	68
3.7	Evaluasi Hasil.....	69
3.7.1	Evaluasi Klaster	69
3.7.2	Uji Hasil Evaluasi Beberapa Metode Menggunakan Uji Statistik	71
BAB 4	HASIL DAN PEMBAHASAN.....	73
4.1	Hasil.....	73
4.1.1	Hasil Eksperimen Metode MacQueen [23]	74
4.1.2	Hasil Eksperimen Metode Bandiopadhyay dan Maulik [44] ...	74
4.1.3	Hasil Experimen Metode Omran et al [46]	75
4.1.4	Hasil Eksperimen Metode Das et al [42]	76
4.1.5	Hasil Eksperimen Metode yang Diusulkan.....	77
4.1.6	Hasil Eksperimen Perbandingan Kinerja Metode.....	77
4.1.7	Hasil Pengukuran Kinerja Metode dari Fungsi Objektif DBI dan CS.....	78
4.2	Pembahasan.....	80

4.2.1	Perbandingan Kinerja Metode Penentuan Jumlah Klaster pada Metode k-Means	80
4.2.2	Perbandingan Metode Pengklasteran untuk Hasil Pencarian Jumlah Klaster pada Dataset Berdasarkan Fungsi Objektif DBI dan CS Secara Otomatis	87
BAB 5	KESIMPULAN DAN SARAN.....	90
5.1	Kesimpulan.....	90
5.2	Saran	92
	DAFTAR PUSTAKA.....	94
	LAMPIRAN.....	105

DAFTAR GAMBAR

Gambar 2.1 : Tahapan SLR.....	10
Gambar 2.2 : Peta Pikiran Pertanyaan Penelitian.....	12
Gambar 2.3 : Penelusuran dan Seleksi Studi Utama	14
Gambar 2.4 : Jumlah Publikasi Tahun 2007-2016	16
Gambar 2.5 : Jumlah Publikasi Penelitian	16
Gambar 2.6 : Peneliti yang Aktif dan Jumlah Publikasi.....	18
Gambar 2.7 : Distribusi Dataset Privat dan Publik.....	19
Gambar 2.8 : Distribusi Dataset	20
Gambar 2.9 : Dataset yang Sering Digunakan	20
Gambar 2.10 : Peta Pikiran Klasifikasi Metode <i>Traditional-Based</i>	21
Gambar 2.11 : Peta Pikiran Klasifikasi Metode <i>Merge-Split Based</i>	22
Gambar 2.12 : Peta Pikiran Klasifikasi Metode <i>Evolutionary Computation</i>	23
Gambar 2.13 : Metode MacQueen (Disusun dari [23]).....	24
Gambar 2.14 : Metode Sarkar et al (Disusun dari [60])	26
Gambar 2.15 : Metode Bandyopadhyay dan Maulik (Disusun dari [44])	28
Gambar 2.16 : Metode Omran et al (Disusun dari [46]).....	30
Gambar 2.17 : Metode Kuo et al. (Disusun dari [61]).....	32
Gambar 2.18 : Metode Kuo dan Sulvia (Disusun dari [47])	34
Gambar 2.19 : Metode Xiao et al (Disusun dari [62])	36
Gambar 2.20 : Metode Das et al. (Disusun dari [42]).....	38
Gambar 2.21 : Metode Kuo et al. (Disusun dari [51]).....	40
Gambar 2.22 : Metode Bandyopadhyay dan Maulik (Disusun dari [44])	43
Gambar 2.23 : Metode Omran et al. (Diambil dari [46]).....	45
Gambar 2.24 : Metode Das et al. (Diambil dari [42])	47
Gambar 2.25 : Peta Pikiran Lengkap dari SLR untuk Penentuan Jumlah Klaster pada Metode k-Means	50
Gambar 3.1 : Tahapan Penelitian, Aktifitas dan Relasi Bab.....	51
Gambar 3.2 : Metode Penentuan Jumlah Klaster pada Metode k-Means yang Diusulkan.....	54

Gambar 3.3 : Kerangka Pemikiran Penelitian	68
Gambar 4.1 : Diagram Perbandingan Kinerja Metode Berdasarkan Fungsi Objektif DBI pada Setiap Metode untuk Masing-masing Dataset	80
Gambar 4.2 : Diagram Perbandingan Kinerja Metode Berdasarkan Fungsi Objektif CS.....	81
Gambar 4.3 : Perbandingan Kinerja Metode Berdasarkan Fungsi Objektif DBI pada Setiap Metode untuk Semua Dataset.....	83
Gambar 4.4 : Perbandingan Kinerja Metode Berdasarkan Fungsi Objektif CS Pada Setiap Metode untuk Semua Dataset.....	87
Gambar 4.5 : Perbandingan Hasil Metode Pengklasteran yang Optimal dan Tidak Optimal dalam Penentuan Jumlah Klaster berdasarkan Fungsi Objektif DBI pada Setiap Metode untuk Masing-masing Dataset	88
Gambar 4.6 : Perbandingan Hasil Pengklasteran yang Optimal dan Tidak Optimal dalam Penentuan Jumlah Klaster berdasarkan Fungsi Objektif CS Pada Setiap Metode untuk Masing-masing Dataset	89

DAFTAR TABEL

Tabel 2.1 : Kriteria PICOC	11
Tabel 2.2 : Pertanyaan Penelitian (RQ)	11
Tabel 2.3 : Kriteria Penelitian yang Diambil dan Tidak Diambil	13
Tabel 2.4 : Ekstraksi Data untuk Pertanyaan Penelitian (RQ)	15
Tabel 2.5 : Nilai <i>Scimago Journal Rank</i> (SJR) dari Publikasi Jurnal.....	17
Tabel 2.6 : Nilai <i>Scimago Journal Rank</i> (SJR) dari Publikasi Prosiding	17
Tabel 2.7 : Rangkuman Metode-metode yang Sering Digunakan dalam Topik Penentuan Jumlah Klaster pada Metode k-Means	41
Tabel 2.8: Daftar Referensi <i>state-of-the-art</i> dalam Topik Penentuan Jumlah Klaster pada Metode k-Means	48
Tabel 3.1 : Identifikasi Datasest	52
Tabel 3.2 : Perbandingan Metode Penentuan Jumlah Klaster pada Metode k-Means	53
Tabel 3.3 : Data Iris	55
Tabel 3.4 : Setting Parameter Metode Usulan.....	56
Tabel 3.5 : Hasil Perhitungan Metode UCC	58
Tabel 3.6 : Contoh Hasil Posisi Klaster Aktif.....	59
Tabel 3.7 : Contoh Hasil Perhitungan Fungsi Objektif DBI dari Metode Usulan	60
Tabel 3.8 : Source Code Tahap Mutasi Pada Metode Usulan.....	61
Tabel 3.9 : Hasil Update Posisi N-Populasi.....	61
Tabel 3.10 : Contoh Sebagian Hasil Perhitungan Mutasi pada Metode Usulan	62
Tabel 3.11 : Contoh Hasil <i>Crossover</i> pada Metode Usulan.....	62
Tabel 3.12 : Pusat Awal Klaster	62
Tabel 3.13 : Pusat Klaster Iterasi Pertama	63
Tabel 3.14 : Iterasi Pertama.....	64
Tabel 3.15 : Iterasi ke-2	65
Tabel 3.16 : Pusat Klaster Iterasi ke-2.....	65
Tabel 3.17 : Iterasi ke-3	66
Tabel 3.18 : Pusat Klaster Iterasi ke-3	66
Tabel 3.19 : Iterasi ke-11	67
Tabel 3.20 : Spesifikasi Komputer yang Digunakan	69
Tabel 4.1 : Rangkuman Dataset.....	73

Tabel 4.2 : Hasil Evaluasi DBI dari Metode MacQueen untuk Semua Dataset.....	74
Tabel 4.3 : Hasil Evaluasi Fungsi Objektif DBI dari Metode GCUK untuk Semua Dataset	74
Tabel 4.4 : Hasil Evaluasi Fungsi Objektif CS dari Metode GCUK untuk Semua Dataset	75
Tabel 4.5 : Hasil Evaluasi Fungsi Objektif DBI dari Metode DCPSO untuk Semua Dataset	75
Tabel 4.6 : Hasil Evaluasi Fungsi Objektif CS dari Metode DCPSO untuk Semua Dataset	75
Tabel 4.7 : Hasil Evaluasi Fungsi Objektif DBI dari Metode ACDE untuk Semua Dataset	76
Tabel 4.8 : Hasil Evaluasi Fungsi Objektif CS dari Metode ACDE untuk Semua Dataset	76
Tabel 4.9 : Hasil Evaluasi Fungsi Objektif DBI dari Metode Usulan untuk semua Dataset	77
Tabel 4.10 : Hasil Evaluasi Fungsi Objektif CS dari Metode Usulan untuk Semua Dataset	77
Tabel 4.11 : Peringkat Evaluasi Metode Berdasarkan Fungsi Objektif DBI dan CS pada Setiap Metode untuk Masing-masing Dataset	78
Tabel 4.12 : Rangkuman Hasil Evaluasi Metode Berdasarkan Fungsi Objektif DBI dan CS untuk Setiap Dataset.....	79
Tabel 4.13 : Uji Friedman dari Fungsi Objektif DBI	82
Tabel 4.14 : Perbandingan Kinerja Metode Berdasarkan Fungsi Objektif DBI dan Uji Friedman pada Semua Metode untuk Semua Dataset	82
Tabel 4.15 : Hasil Uji Friedman dari Fungsi Objektif CS	84
Tabel 4.16 : Perbandingan Evaluasi Metode Berdasarkan Fungsi Objektif CS dan Uji Friedman pada Setiap Metode untuk Masing-masing Dataset	84
Tabel 4.17 : Perbandingan berpasangan Uji <i>Nemenyi post hoc</i>	85
Tabel 4.18 : Nilai <i>P</i> -value Uji <i>Nemenyi post hoc</i> dari CS	86
Tabel 4.19 : Perbedaan Signifikan Hasil CS dari Uji <i>Nemenyi post hoc</i>	86

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Metode klastering merupakan sebuah teknik *unsupervised learning* [1], bertujuan mengelompokkan data ke dalam beberapa kelompok atau klaster di mana tidak memiliki target label dalam prosesnya [2]. Metode klastering adalah sebuah proses membagi data menjadi beberapa klaster, dalam satu klaster memiliki tingkat kemiripan yang maksimum dan data antar klaster memiliki tingkat kemiripan yang minimum [3]. Kemiripan antar data dapat dilakukan dengan menghitung jarak antar data [4], semakin kecil jarak antar data maka semakin tinggi kemiripan antar data tersebut dan sebaliknya [1]. Metode ini telah banyak digunakan di berbagai aplikasi dunia nyata seperti segmentasi citra [5], dan analisis DNA microarray [6] dan lain sebagainya.

Menurut Han dan Kamber [4], secara umum metode klastering dapat diklasifikasikan menjadi empat kategori dasar yaitu metode partisi, metode hirarki, metode berbasis kepadatan, dan metode berbasis *grid*. Metode klastering yang paling populer adalah metode hirarki dan metode partisi [7]. Pada penelitian ini berfokus pada metode partisi. Metode partisi memisahkan data ke dalam kelompok sebanyak k di mana setiap partisi mewakili klaster [8][9]. Kelebihan dari metode partisi adalah dapat memanipulasi dataset dalam jumlah yang besar [10] dan membutuhkan waktu komputasi yang sedikit [10] dari pada metode hirarki [11][12]. Metode partisi dikelompokkan menjadi dua bagian utama [13][14], *hard partition* dan *soft partition*. *Hard partition* mengelompokkan masing-masing objek harus tepat dalam satu klaster dan tidak tumpang tindih [15]. Sedangkan metode *soft partition*, setiap objek memiliki tingkat nilai keanggotaan dalam interval $[0, 1]$ untuk setiap klaster [10].

Metode k-Means merupakan salah satu metode *hard partition* yang paling banyak digunakan [16][17]. Metode k-Means memiliki beberapa kelebihan, di antaranya, sederhana untuk diimplementasikan [18], dapat mengelompokkan dataset yang besar [9], kuat diparalel *processing* [11] dan metode ini cukup efisien dalam hal kompleksitas waktu [4]. Penelitian tentang k-Means dapat ditelusuri kembali ke

pertengahan abad yang lalu di berbagai disiplin ilmu yang berbeda [19], terutama oleh Lloyd (1957) [20], Forgey (1965) [21], Friedman dan Robin (1967) [22] dan MacQueen (1967) [23].

k-Means dalam mengelompokkan data dimulai dengan memilih jumlah k klaster sesuai kelompok yang akan dibentuk, kemudian membangkitkan k pusat awal klaster secara acak, selanjutnya setiap objek akan dikelompokkan berdasarkan jarak terdekat dengan pusat klaster. Menurut beberapa peneliti seperti Jain [16], M. C. Naldi et al [18], dan Bagirov et al [24], hasil pengklasteran dari k-Means bergantung pada penentuan jumlah k klaster sebagai masukan, jika penentuan jumlah klaster jauh dari solusi akhir maka kemungkinan hasil pengklasteran tidak tepat. Sedangkan, untuk mendapatkan jumlah k sebagai input k-Means sulit dilakukan karena diperlukan informasi dari data secara apriori [25][26][27][28]. Kondisi ini dikenal sebagai solusi optimum lokal [25]. Dalam praktiknya, masalah optimum lokal dapat diselesaikan dengan menjalankan metode beberapa kali dengan jumlah k yang berbeda kemudian memilih hasil yang terbaik. Penentuan jumlah klaster sangat penting bagi metode k-Means [3][16][29][30][31].

Menurut Hancer dan Karaboga [32] terdapat tiga pendekatan utama dalam mengatasi penentuan jumlah klaster pada metode k-Means, yaitu, *traditional-based* (TB), *merge-split* (MS) dan *evolutionary computation* (EC). Pendekatan *traditional-based* bertujuan mendapatkan jumlah klaster yang optimal dari grafik evaluasi yang ditentukan oleh deteksi *knee-point* [32]. Solusi jumlah klaster mengacu pada grafik evaluasi *knee-point* di mana sumbu x adalah jumlah klaster dan sumbu y adalah nilai fungsi evaluasi yang sesuai [33]. Keuntungan dari pendekatan ini adalah dapat menentukan jumlah klaster yang global optimum [33]. Namun, dalam mendeteksi *knee-point* atau titik-titik jumlah k pada grafik evaluasi tidak mudah ditemukan karena tidak ada ketentuan khusus yang baik pada pemilihan solusi akhir *knee-based* [27][28][34]. Selain itu *knee-based* membutuhkan waktu komputasi yang lebih tinggi, karena metode tersebut harus dijalankan berkali-kali [35].

Berbeda dengan pendekatan *merge-split*, metode *merge-split* adalah metode berbasis *wrapper* [32] yang digunakan untuk mengelompokkan data tanpa memerlukan informasi apriori [36] dan populer digunakan pada data gambar [37]. Metode *merge-split* dipopulerkan oleh Chaudhuri dan Murthy [36]. Metode ini

bekerja dengan cara membagi data ke dalam beberapa kelompok berdasarkan jarak terjauh kemudian klaster yang terbentuk dipecah menjadi beberapa klaster kecil sampai dengan batas (*threshold*) yang ditentukan pengguna untuk split. Kemudian dilakukan proses *merge* atau penggabungan antar klaster yang berdekatan. Proses selanjutnya adalah partisi, menentukan pixel terhadap tiap-tiap pusat klaster untuk mendapatkan pusat klaster baru. Pendekatan *merge-split* bekerja dengan baik mengidentifikasi jumlah klaster optimal [32] walaupun informasi data tidak ditentukan secara apriori [38]. Namun, kinerja pada metode pendekatan *merge-split* masih bergantung pada ambang batas parameter k sebagai representasi jumlah klaster dari pengguna [37][39][32].

Pendekatan EC merupakan sebuah teknik pencarian solusi pada permasalahan optimasi berdasarkan prinsip evolusi biologis, seperti seleksi alam dan keturunan genetika yang terinspirasi oleh evolusi Darwin [40]. Teknik EC telah banyak diaplikasikan untuk memecahkan masalah pengklasteran [32]. Pendekatan EC bekerja dengan cara menginisialisasi jumlah populasi k klaster (kromosom atau partikel) secara acak. Kromosom yang terpilih akan dibangkitkan kemudian dievaluasi menggunakan fungsi kesesuaian atau fungsi objektif (*function objective*). Individual terbaik akan terpilih, selanjutnya akan direproduksi, di mana alat reproduksi ini bertujuan untuk mendapatkan keturunan terbaik (*offspring*) atau solusi kandidat baru yang mewarisi sebagian sifat dari induknya. Proses pencarian evolusioner dilakukan secara iteratif sampai kriteria tertentu terpenuhi atau solusi optimal telah tercapai. Pendekatan EC memiliki kelebihan yaitu efektif untuk mengatasi masalah *NP-hard partition* [41] seperti tidak membutuhkan definisi jumlah klaster [26], dapat mengelompokkan data secara otomatis [42] dan mampu mengatasi masalah optimum lokal [41].

Berdasarkan landasan di atas, pendekatan EC dapat memecahkan masalah *traditional-based* dan *merge-split based*, yaitu waktu komputasi yang lebih tinggi dan inisialisasi jumlah k klaster [32] yang mana ini tepat untuk karakteristik dari metode pendekatan EC sebagai metode pencarian berbasis heuristik. Menurut [41] pendekatan *evolutionary computation* merupakan cara efektif untuk memberikan sejumlah solusi klaster yang hampir optimal secara otomatis.

Seperti yang telah dijelaskan sebelumnya bahwa penentuan jumlah kluster dapat mempengaruhi hasil akhir pengklasteran maka metode pengklasteran data otomatis merupakan solusi yang menjanjikan dalam masalah ini. Pengklasteran data otomatis adalah salah satu teknik yang dapat membantu pengguna menentukan jumlah kluster optimal [43].

Telah banyak penelitian [32][8][42] yang dilakukan untuk penentuan jumlah kluster berbasis pendekatan EC baik menggunakan k-Means sebagai metode dasar maupun metode heuristik lainnya yang berdiri sendiri dengan bantuan fungsi objektif dalam memandu pencarian evolusioner. Bandyopadhyay dan Maulik [44] telah mengusulkan metode *genetic clustering for unknown k* (GCUK) untuk melakukan pengklasteran data secara otomatis. Metode ini dimulai dengan mengkodekan sejumlah kluster berdasarkan pada *prototype don't-care*. Kesesuaian kromosom dihitung menggunakan DBI untuk memandu pencarian evolusioner [44], di mana nilai fungsi objektif terkecil adalah hasil partisi yang valid [40]. Metode GCUK efisien mengelompokkan data tanpa memerlukan informasi label pada data dengan cara otomatis [44]. Namun metode ini hanya efektif pada empat dataset sintetis dan dua dataset real [45].

Omran et al [46] telah mengusulkan metode *dynamic clustering pso* (DCPSO), merupakan penggabungan dari PSO dan k-Means dengan menggunakan pendekatan PSO biner dalam menemukan jumlah pusat kluster dalam data [47][13]. PSO melakukan pengklasteran dan k-Means melakukan penyempurnaan kluster [48]. Keuntungan DCPSO, selain dapat menghasilkan jumlah kluster yang optimal juga efektif memecahkan masalah optimasi non-linear pada data skala besar [49]. Akan tetapi, masih terbatas pada pengujian data gambar asli dan beberapa data gambar sintetis [48].

Swagatam Das et al [42] telah mengusulkan metode *automatic clustering differential evolution* (ACDE), di mana metode ini dapat memecahkan masalah GCUK dan DCPSO, yaitu pengujian dataset masih terbatas pada gambar, yang mana ini tepat untuk karakteristik ACDE. Metode dasar dari ACDE adalah *differential evolution* (DE). Menurut [50], metode DE tidak menemukan kesulitan pada data berdimensi tinggi dan data polynomial dan pada uji fungsi lainnya.

Keberhasilan ACDE dilakukan dengan membandingkan DCPSO dan GCUK pada dataset real, gambar dan data sintetis [42][45][32].

ACDE kemudian dikembangkan oleh [51] dengan cara melakukan kombinasi metode ACDE dan k-Means diberi nama *automatic clustering approach based on differential evolution algorithm combining with k-Means for crisp clustering* (ACDE) yang bertujuan untuk meningkatkan kinerja pengklasteran pada metode k-Means. Metode ACDE memiliki kemampuan untuk menemukan jumlah kluster secara otomatis dan mampu menyeimbangkan proses evolusi metode DE sehingga dapat mencapai partisi yang lebih baik dibandingkan DE klasik. Namun metode ACDE masih bergantung pada pertimbangan pengguna untuk menentukan ambang batas aktivasi k sehingga mempengaruhi kinerja metode ACDE [52].

Metode *u-control chart* (UCC) adalah sebuah metode dari *statistical process control* (SPC) yang terbukti efektif digunakan untuk memecahkan masalah pengendalian kontrol atribut [53]. Metode UCC bekerja dengan cara menilai rata-rata suatu data yang akan diukur dengan menggunakan batas atas (*upper value*) dan batas bawah (*under value*). Suatu produk dikatakan memiliki kualitas baik apabila nilai rata-rata berada pada ambang batas atau nilai rata-rata berada pada batas atas dan batas bawah. Dengan asumsi ini maka suatu data dikategorikan baik jika masih dalam ambang batas UCC. Metode ini telah berhasil diaplikasikan pada metode genetika [53], di mana penelitian tersebut bertujuan menentukan ukuran sampel batas atas dan batas bawah data pada bagan kontrol atribut dalam proses pembuatan piston mesin [53].

Metode yang akan diusulkan pada penelitian ini adalah menggunakan metode UCC untuk mengatasi penentuan ambang batas aktivasi k pada metode ACDE. Metode UCC dimulai dengan mencari nilai rata-rata atau *center line* pada semua atribut data. Hasil dari nilai rata-rata tersebut kemudian dihitung batas atas dan batas bawah-nya. Tahap selanjutnya ACDE akan digunakan untuk menentukan jumlah kluster pada metode k-Means sehingga hasil kluster tidak lagi terjebak ke dalam minimum lokal dan dapat meningkatkan kinerja pengklasteran dengan nilai validitas kluster terendah. Untuk mengevaluasi hasil penelitian akan digunakan DBI dan CS sebagai indikator kinerja metode.

1.2 Identifikasi Masalah

Berdasarkan latar belakang masalah yang diuraikan pada sub bab 1.1, maka identifikasi masalah dalam penelitian ini adalah metode *automatic clustering differential evolution* (ACDE) dapat mengatasi penentuan jumlah klaster pada metode k-Means dengan menggunakan data yang belum diketahui label atau kelasnya, akan tetapi metode ACDE masih bergantung pada penentuan ambang batas aktivasi k pada keputusan pengguna, sehingga mengurangi kinerja metode k-Means.

1.3 Rumusan Masalah

Berdasarkan latar belakang masalah dan identifikasi masalah penelitian yang disajikan pada sub bab 1.1 dan 1.2, maka rumusan masalah dalam penelitian ini adalah bagaimana peningkatan kinerja metode k-Means apabila metode *u-control chart* (UCC) digunakan untuk penentuan ambang batas aktivasi k pada metode *automatic clustering differential evolution* (ACDE) dalam penentuan jumlah klaster k-Means?

1.4 Tujuan Penelitian

Tujuan penelitian ini adalah untuk menerapkan metode *u-control chart* (UCC) untuk penentuan ambang batas aktivasi k pada metode *automatic clustering differential evolution* (ACDE) dalam penentuan jumlah klaster pada metode k-Means sehingga kinerja metode k-Means lebih meningkat.

1.5 Manfaat Penelitian

Hasil penelitian ini diharapkan dapat digunakan untuk melakukan pengklasteran data secara otomatis dengan peningkatan kualitas klaster yang lebih baik. Hal tersebut telah dikonfirmasi dalam penelitian ini di mana metode yang diusulkan memiliki nilai objektif *davies bouldin index* (DBI) dan *cosine similarity measure* (CS) terendah yang lebih baik dari pada metode lain yang dijadikan pembanding (GCUK, DCPSO dan ACDE) dalam penelitian ini. Selain dari itu penelitian ini diharapkan dapat dijadikan referensi dalam pengembangan teori metode penentuan jumlah klaster k-Means dengan pendekatan EC pada analisis pengklasteran otomatis.

1.6 Sistematika Penulisan

Pada penelitian ini akan dibagi menjadi lima bab dan di setiap bab akan dibagi lagi menjadi beberapa sub bab sesuai topik yang dibahas. Uraian singkat setiap bab akan dijelaskan.

Bab 1 pendahuluan. Pada bab ini akan dibahas mengenai latar belakang penelitian, identifikasi masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

Bab 2 tinjauan pustaka. Pada bab ini membahas tentang tinjauan studi serta kerangka pemikiran. Digunakan *systematic literature review* (SLR) tentang penentuan jumlah klaster pada metode k-Means. Dalam bab ini juga diuraikan tentang penelitian terkait dengan topik penentuan jumlah klaster.

Bab 3 metode penelitian. Pada bab ini menjelaskan tentang metode penelitian yang digunakan, metode tersebut terdiri dari pengumpulan data, pengolahan data, eksperimen dan pengujian terhadap metode yang diusulkan.

Bab 4 hasil dan pembahasan. Pada bab ini berisi pembahasan eksperimen yang dilakukan serta evaluasi dan validasi klaster. Hasil eksperimen yang berisi data disajikan dalam bentuk tabel-tabel dan perbandingan hasil validitas klaster dari model yang digunakan.

Bab 5 penutup. Pada bab ini akan berisi kesimpulan dari hasil penelitian, dan saran untuk penelitian lebih lanjut.

BAB 2

TINJAUAN PUSTAKA

2.1 Pendahuluan

Sebelum melakukan penelitian lebih lanjut, diperlukan kajian terhadap penelitian terdahulu untuk mengetahui metode yang sudah pernah dilakukan. Bab ini akan memberikan tinjauan pustaka pada bidang penentuan jumlah klaster pada metode k-Means. Tinjauan pustaka ini bertujuan untuk mengidentifikasi dan menganalisis metode-metode yang telah digunakan untuk penentuan jumlah klaster pada metode k-Means. Ada sebanyak 39 *paper* yang dipublikasikan antara 1 Januari 2007 sampai 31 Desember 2016 akan diinvestigasi. Telah banyak metode yang digunakan untuk penentuan jumlah klaster pada metode k-Means diterbitkan cukup beragam dan kompleks, sehingga diperlukan gambaran yang menyeluruh mengenai metode *state-of-the-art* dari penelitian tersebut. Studi tinjauan pustaka yang sistematis ini memberikan gambaran umum tentang apa yang telah diterbitkan.

2.2 Metode *Systematic Literature Review* (SLR)

Metode tinjauan pustaka merupakan evaluasi kritis dan mendalam dari penelitian sebelumnya yang dilakukan dengan meringkaskan dan sinopsis dari bidang penelitian tertentu, memungkinkan bagi siapa pun dapat menetapkan alasan dalam mengejar suatu penelitian tertentu [54][55]. Menurut Wahono [55], tinjauan pustaka yang baik mengevaluasi kualitas dan temuan penelitian sebelumnya.

Metode tinjauan pustaka dalam penelitian ini menggunakan teknik *systematic literature review* (SLR). SLR merupakan sebuah proses untuk mengidentifikasi, menilai, dan menginterpretasikan semua penelitian yang tersedia dengan tujuan memberikan jawaban atas pertanyaan penelitian (*Research Question* (RQ)) yang spesifik [56]. SLR dalam penelitian ini bertujuan meninjau kembali mengenai bidang penentuan jumlah klaster pada metode k-Means. Metode tinjauan pustaka, sumber literatur, gaya dan perumusan pertanyaan pada bab ini terinspirasi oleh Wahono [55].

Seperti pada Gambar 2.1, secara umum menunjukkan penelitian ini dilakukan dalam tiga tahapan SLR: 1) tahap perencanaan, 2) tahap pelaksanaan dan 3) tahap pelaporan. Tahap awal (Tahap 1) adalah menentukan kebutuhan dari dilakukannya

tinjauan pustaka, secara sistematis dilakukan dengan tiga langkah yaitu identifikasi kebutuhan SLR, perumusan pertanyaan penelitian dan mengembangkan ulasan protokol agar terfokus dan konsisten sesuai dengan bidang penelitian yang akan ditinjau. Tahap berikutnya (Tahap 2) adalah merencanakan tahapan tinjauan pustaka dengan cara pencarian studi utama, pemilihan studi utama, ekstraksi data dari studi utama, menilai kualitas studi utama dan mensintesis data sehingga akan diperoleh jenis *paper* yang berkualitas tinggi dan siap dijadikan rujukan utama dalam penelitian. Tahap terakhir (Tahap 3) adalah melaporkan hasil tinjauan pustaka. Tahapan tinjauan telah dijabarkan pada sub bab 2.2.1, 2.2.2, 2.2.3, 2.2.4 dan 2.2.5.

Gambar 2.1 : Tahapan SLR

2.2.1 Pertanyaan Penelitian

Pertanyaan penelitian atau *Research Question* (RQ) ini dilakukan agar tinjauan terhadap penelitian terkini tetap fokus. Pada tahap ini akan menggunakan desain kriteria dari [56], yaitu *population*, *intervention*, *comparison*, *outcomes*, dan *context* (PICOC). Tabel 2.1 menunjukkan desain PICOC dari pertanyaan penelitian.

Tabel 2.1 : Kriteria PICOC

Population	k-Means Klustering
Intervention	Metode penentuan jumlah klaster
Comparison	-
Outcomes	Kinerja k-Means
Context	Penelitian di industri dan universitas dengan menggunakan dataset kecil dan besar

Pertanyaan penelitian (RQ) pada tinjauan pustaka dapat dilihat pada Tabel 2.2. Rumusan masalah ini hanya ditujukan untuk tinjauan pustaka, tidak untuk studi utama. Peta pikiran pertanyaan penelitian secara visual dapat dilihat pada Gambar 2.2.

Tabel 2.2 : Pertanyaan Penelitian (RQ)

ID	Pertanyaan Penelitian	Motivasi
RQ1	Jurnal manakah yang paling banyak memublikasikan untuk topik penentuan jumlah klaster pada metode k-Means?	Mengidentifikasi jurnal yang paling banyak memublikasikan untuk topik penentuan jumlah klaster pada metode k-Means
RQ2	Siapakah peneliti yang paling aktif melakukan publikasi untuk topik penentuan jumlah klaster pada metode k-Means?	Mengidentifikasi peneliti yang paling aktif melakukan publikasi untuk topik penentuan jumlah klaster pada metode k-Means
RQ3	Dataset manakah yang paling sering digunakan untuk topik penentuan jumlah klaster pada metode k-Means?	Mengidentifikasi dataset yang paling sering digunakan untuk topik penentuan jumlah klaster pada metode k-Means
RQ4	Strategi pendekatan metode apa yang paling sering digunakan untuk	Mengidentifikasi strategi pendekatan metode yang paling sering digunakan

ID	Pertanyaan Penelitian	Motivasi
	topik penentuan jumlah klaster pada metode k-Means	untuk topik penentuan jumlah klaster pada metode k-Means
RQ5	Metode-metode manakah yang pernah diusulkan untuk topik penentuan jumlah klaster pada metode k-Means?	Mengidentifikasi metode-metode yang pernah diusulkan peneliti sebelumnya dalam topik penentuan jumlah klaster pada metode k-Means
RQ6	Metode-metode manakah yang sering digunakan untuk topik penentuan jumlah klaster pada metode k-Means?	Mengidentifikasi metode-metode yang sering digunakan dalam topik penentuan jumlah klaster pada metode k-Means

RQ1 dan RQ2 akan digunakan untuk mendukung konteks penelitian, sedangkan RQ3, RQ4, RQ5 dan RQ6 akan digunakan untuk mendukung studi utama.

Gambar 2.2 : Peta Pikiran Pertanyaan Penelitian

2.2.2 Strategi Pencarian

Pada tahap berikutnya (Langkah 4) SLR dalam mencari penelitian terkait terdiri dari beberapa proses yaitu memilih sumber pencarian, menentukan kata kunci pencarian, melakukan pencarian awal, mengevaluasi dan menyusun ulang kata kunci pencarian, dan mengelola hasil pencarian dari sumber pencarian berdasarkan kata kunci pencarian. Sebelum memulai pencarian, menentukan kata kunci harus terlebih dahulu dilakukan untuk memperbesar kemungkinan menemukan penelitian terkait yang sesuai.

Pada penelitian ini menggunakan sumber pencarian dari:

1. ScienceDirect ([sciencedirect.com](https://www.sciencedirect.com))
2. IEEE eXplore (ieeexplore.ieee.org)

Kata kunci pencarian akan disusun berdasarkan langkah langkah berikut:

1. Identifikasi kata kunci dari PICOC, terutama dari *population* dan *intervention*
2. Identifikasi kata kunci dari rumusan masalah
3. Identifikasi kata kunci dari judul, abstraksi dan kata kunci yang relevan
4. Identifikasi kata kunci dari sinonimnya
5. Mengkonstruksi kata kunci yang kompleks, yang terdiri dari beberapa kata kunci dengan menggunakan *boolean*, *AND* dan *OR*.

Kata kunci pencarian yang digunakan adalah:

(determ) AND (number* OR count* OR initial*) AND (k OR point) AND (cluster* OR seed* OR segmentation) AND (k-Means)*

Pada saat penyesuaian kata kunci pencarian dibentuk, kata kunci pencarian awal akan terus disimpan hingga hasil pencarian dapat secara signifikan menaikkan kecocokan pencarian. Pencarian dibatasi pada tahun 2007-2016. Publikasi yang diambil adalah publikasi berupa jurnal dan prosiding konferensi. Selain itu, pencarian juga dibatasi pada publikasi yang menggunakan bahasa inggris.

2.2.3 Seleksi Studi Utama

Gambar 2.3 menunjukkan penelusuran dan seleksi studi utama pada setiap tahapan. Seleksi studi utama menggunakan dua kriteria utama yaitu kriteria penelitian yang akan diambil dan tidak diambil di mana bertujuan sebagai penilaian terhadap penelitian. Kriteria-kriteria tersebut dapat dilihat pada Tabel 2.3.

Tabel 2.3 : Kriteria Penelitian yang Diambil dan Tidak Diambil

Kriteria penelitian yang diambil	Penelitian yang membahas mengenai penentuan jumlah klaster pada metode k-Means
	Untuk penelitian yang mempunyai dua tipe publikasi jurnal dan prosiding. Maka akan diambil dua tipe publikasi yang jurnal dan prosiding konferensi
	Untuk penelitian yang duplikasi, akan diambil data terlengkap dan terbaru
Kriteria penelitian yang tidak diambil	Penelitian yang membahas pada bidang penentuan jumlah klaster pada metode k-Means, tetapi tidak mengusulkan metode baru untuk mengatasi masalah penentuan jumlah klaster metode k-Means
	Penelitian tidak ditulis dalam Bahasa Inggris.

Gambar 2.3 : Penelusuran dan Seleksi Studi Utama

Software Mendeley (<http://mendeley.com>) digunakan untuk menyimpan dan mengorganisasi hasil pencarian. Penelitian yang tidak menampilkan hasil penelitian tidak akan digunakan.

Setelah melalui tahap seleksi dan menilai kualitas *paper*, maka ada 39 *paper* yang akan ditinjau. Tahap seleksi sudah memperhitungkan kriteria-kriteria yang terdapat pada Tabel 2.3. Selain itu penelitian yang sama juga telah dihilangkan.

2.2.4 Ekstraksi Data

Ekstraksi data (Tahap 6) dirancang untuk mengumpulkan data dari studi utama yang dibutuhkan untuk menjawab Pertanyaan Penelitian (RQ). Terdapat lima informasi yang digunakan untuk menjawab RQ, dapat dilihat pada Tabel 2.4.

Tabel 2.4 : Ekstraksi Data untuk Pertanyaan Penelitian (RQ)

Ekstraksi Data	Pertanyaan Penelitian
Peneliti dan tahun publikasi	RQ1, RQ2
Dataset yang digunakan	RQ3
Strategi pendekatan metode yang diusulkan	RQ4
Metode yang pernah diusulkan	RQ5
Metode perbaikan yang diusulkan	RQ6

2.2.5 Nilai Kualitas Penelitian dan Sintetis Data

Penilaian kualitas penelitian (Tahap 7) dapat digunakan untuk membantu menginterpretasikan kualitas dari temuan dan untuk menentukan kekuatan kesimpulan yang diuraikan. Sedangkan tujuan dari sintetis data adalah untuk mengumpulkan bukti dari publikasi yang sudah didapatkan untuk menjawab pertanyaan penelitian. Sintetis data (Tahap 8) yang digunakan pada penelitian ini, secara umum akan berupa sintetis narasi. Beberapa tabel dan alat visual (Microsoft Office Visio [57] dan XMind [58]) akan digunakan untuk menunjang penjelasan pada penelitian ini.

2.3 Publikasi Jurnal Paling Signifikan dalam Topik Penentuan Jumlah Klaster pada Metode k-Means

Pada tinjauan pustaka ini akan menganalisa sebanyak 39 *paper* yang telah dipublikasi pada jurnal bereputasi ilmiah seperti Elsevier dan IEEE di mana diperoleh melalui proses penelusuran dan seleksi studi utama seperti yang ditampilkan pada Gambar 2.3, mengenai penentuan jumlah klaster pada metode k-Means. Hal ini dilakukan sebagai upaya untuk menjawab pertanyaan penelitian (RQ1). Seperti yang telah dijelaskan sebelumnya, tinjauan pustaka ini akan dibatasi pada jurnal yang telah dipublikasikan dari tahun 2007 sampai 2016. Rentang waktu tersebut untuk melihat apakah penelitian dalam topik penentuan jumlah klaster pada metode k-Means masih relevan. Pada Gambar 2.4 dapat dilihat bahwa tren penelitian dari tahun 2007 sampai

2016 menunjukkan topik penentuan jumlah kluster pada k-Means masih sangat relevan sampai saat ini.

Gambar 2.4 : Jumlah Publikasi Tahun 2007-2016

Gambar 2.5 : Jumlah Publikasi Penelitian

Seperti yang dapat dilihat pada Gambar 2.5 dan Tabel 2.6 menunjukkan jumlah jurnal dan prosiding dalam topik penentuan jumlah klaster pada metode k-Means telah dipublikasikan dari tahun 2007 sampai 2016. Sebagai catatan, jurnal dan prosiding yang dimaksud adalah yang memublikasikan *paper* terpilih. Nilai *scimago journal rank* (SJR) dan kategori *Q rank* (Q1-Q6) dipilih dari jurnal dan prosiding. Publikasi jurnal telah diurutkan berdasarkan nilai SJR-nya.

Tabel 2.5 : Nilai *Scimago Journal Rank* (SJR) dari Publikasi Jurnal

No	Publikasi Jurnal	SJR	Q Rank
1	IEEE Transactions on Geoscience and Remote Sensing	2.563	Q1
2	Information Sciences	2.284	Q1
3	IEEE Transactions on Evolutionary Computation	2.26	Q1
4	Pattern Recognition	2.05	Q1
5	IEEE Transactions on Knowledge and Data Engineering	2.02	Q1
6	Expert Systems with Applications	1.84	Q1
7	Fuzzy Sets and Systems	1.71	Q1
8	Knowledge-Based Systems	1.6	Q1
9	Applied Soft Computing Journal	1.579	Q1
10	Computational Statistics & Data Analysis	1.35	Q1
11	Applied Mathematics and Computation	1.074	Q1
12	Engineering Applications of Artificial Intelligence	1.074	Q1
13	Data and Knowledge Engineering	1.04	Q2
14	Pattern Recognition Letters	0.976	Q2
15	Neurocomputing	0.942	Q2
16	Theoretical Computer Science	0.81	Q2
17	Computer Network	0.585	Q2
18	Progress in Natural Science: Materials International	0.221	Q2

Tabel 2.6 : Nilai *Scimago Journal Rank* (SJR) dari Publikasi Prosiding

No	Publikasi Jurnal	SJR	Rank
19	IEEE Congress on Evolutionary Computation, CEC 2009	0.211	Prosiding
20	SHUSER 2012 - 2012 IEEE Symposium on Humanities, Science and Engineering Research	0.115	Prosiding
21	IEEE Region 10 Annual International Conference, Proceedings/TENCON	0.109	Prosiding

No	Publikasi Jurnal	SJR	Rank
22	IEEE Congress on Evolutionary Computation (CEC)	0	Prosiding
23	International Conference on Informatics, Electronics and Vision (ICIEV)	0	Prosiding

2.4 Peneliti Paling Aktif dalam Topik Penentuan Jumlah Kluster pada Metode k-Means

Dari studi utama yang diperoleh, peneliti yang paling aktif dan berkontribusi pada penelitian tentang penentuan jumlah kluster pada metode k-Means akan diinvestigasi dan diidentifikasi sebagai upaya menjawab pertanyaan penelitian (RQ2). Gambar 2.6 menunjukkan peneliti yang paling aktif di bidang ini. Perlu dicatat bahwa Sriparna Saha, Swagatam Das, Celal Ozturk, Sanghamitra Bandyopadhyay, Ruochen Liu dan R. J Kuo adalah peneliti paling aktif dalam bidang penentuan jumlah kluster pada metode k-Means.

Gambar 2.6 : Peneliti yang Aktif dan Jumlah Publikasi

2.5 Dataset yang Digunakan dalam Topik Penentuan Jumlah Kluster pada Metode k-Means

Distribusi dataset yang digunakan selama bertahun-tahun disajikan untuk menunjukkan bagaimana minat terhadap tipe dataset telah berubah dari waktu ke waktu sebagai upaya menjawab pertanyaan penelitian (RQ3), seperti yang ditunjukkan pada Gambar 2.7. Pada Gambar 2.8 menunjukkan hanya 10% penelitian

menggunakan dataset privat. Ini berarti bahwa hanya akan ada beberapa hasil penelitian yang dapat dibandingkan dan dapat diulangi, sehingga tidak mungkin membandingkan hasil penelitian tersebut dengan hasil model yang diusulkan karena dataset mereka tidak didistribusikan sebagai publik. Penggunaan dataset publik membuat penelitian dapat dilakukan berulang, dapat disanggah dan dapat diverifikasi [59]. Distribusi dataset yang sering digunakan selama bertahun-tahun, dan per sumber, disajikan pada Gambar 2.8. Sejak tahun 2005, hasil penelitian yang telah banyak dilakukan oleh peneliti sebelumnya menunjukkan bahwa dataset publik lebih banyak digunakan untuk penentuan jumlah klaster pada metode k-Means. Hampir semua dataset yang diidentifikasi bersifat publik berasal dari repositori *University of California Irvine* (UCI). Ini berarti, ada peningkatan kesadaran di antara para peneliti mengenai penggunaan dataset publik.

Gambar 2.7 : Distribusi Dataset Privat dan Publik

Gambar 2.8 menunjukkan sebagian kecil penelitian menggunakan dataset privat (10%) sejak tahun 2007. Dengan demikian, tidak mungkin membandingkan hasil penelitian tersebut dengan hasil model yang diusulkan, karena dataset mereka tidak dapat digunakan. Ini menjelaskan mengapa penelitian ini perlu menggunakan dataset publik agar dapat diverifikasi dan diulang serta hasilnya dapat divalidasi.

Gambar 2.8 : Distribusi Dataset

Gambar 2.9 menunjukkan dataset publik yang sering digunakan dalam topik penentuan jumlah kluster pada metode k-Means. Dataset tersebut dipublikasikan dan dapat diakses secara gratis di repositori UCI. Dataset yang sering digunakan adalah *artificial dataset*, *iris*, *wisconsin breast cancer*, *wine*, *glass*, *new theroid*, *lung cancer*, *liver disorder*, *vowel* dan *dermatology*.

Gambar 2.9 : Dataset yang Sering Digunakan

2.6 Strategi Pendekatan Metode dalam Topik Penentuan Jumlah Kluster pada Metode k-Means

Terdapat tiga strategi pendekatan utama dalam menangani masalah penentuan jumlah kluster pada metode k-Means yaitu pendekatan metode *traditional-based*, metode *merge-split based* dan metode *evolutionary computation based*. Ketiga strategi pendekatan tersebut telah dilakukan oleh peneliti sebelumnya yang akan dijelaskan secara ringkas pada sub bab 2.6.1, 2.6.2 dan 2.6.3.

2.6.1 Metode *Traditional-Based* (TB)

Metode TB dapat digunakan untuk menentukan jumlah kluster pada sebagian besar metode pengklasteran termasuk metode k-Means [32]. Pada bab 1, telah dijelaskan secara ringkas tentang tujuan, kelebihan dan kelemahan serta cara metode TB dalam menemukan jumlah kluster yang optimal. Berikut ini adalah peta pikiran klasifikasi metode TB, dapat dilihat pada Gambar 2.10. Pendekatan metode TB secara ringkas diklasifikasikan menjadi empat kategori utama: *knee-based*, *resampling-based*, *static-dynamic based* dan *observation area*. Metode yang paling banyak digunakan adalah *knee-based* dan *resampling based* [32]. Informasi detail dan komprehensif dapat merujuk pada Hancer dan Karaboga [32].

Gambar 2.10 : Peta Pikiran Klasifikasi Metode *Traditional-Based*

2.6.2 Metode *Merge-Split Based* (MS)

Pendekatan metode MS pertama kali dipopulerkan oleh Chaudhuri et al [36]. Pendekatan metode MS adalah berbasis *wrapper* [32] dan dapat digunakan untuk menentukan jumlah kluster pada sebagian besar metode pengklasteran termasuk metode k-Means [32]. Pada bab 1, juga telah dijelaskan secara ringkas tentang tujuan, kelebihan dan kelemahan serta cara metode MS dalam menemukan jumlah kluster yang optimal. Berikut ini adalah peta pikiran klasifikasi metode MS, dapat dilihat pada Gambar 2.11. Pendekatan metode MS diklasifikasikan menjadi dua kategori utama: *model selection* dan *homogeneity*. Informasi detail dan komprehensif dapat merujuk pada Hancer dan Karaboga [32].

Gambar 2.11 : Peta Pikiran Klasifikasi Metode *Merge-Split Based*

2.6.3 Metode *Evolutionary Computation Based* (EC)

Pendekatan metode EC adalah sebuah metode pencarian jumlah kluster yang paling banyak diminati oleh para peneliti di bidang pengklasteran karena memiliki kelebihan yang tidak dimiliki oleh *traditional-based* (TB) dan *merge-split based* (MS) yaitu waktu komputasi yang tinggi dan inisialisasi jumlah kluster. Pada bab 1, paragraf ketujuh telah dijelaskan secara ringkas mulai dari tujuan, kelebihan dan kelemahan serta cara metode EC dalam menemukan jumlah kluster yang optimal. Berikut ini adalah peta pikiran klasifikasi metode EC, dapat dilihat pada Gambar 2.12. Informasi detail dan komprehensif dapat merujuk pada Hancer dan Karaboga [32]. Metode EC merupakan pendekatan yang paling banyak digunakan untuk penentuan jumlah kluster pada sebagian besar masalah metode pengklasteran termasuk metode k-Means.

klaster yang baru merupakan rata-rata dari setiap klaster. Proses ini diulangi sampai tidak ada perubahan anggota klaster [23]. Namun masih terbatas pada penentuan jumlah klaster dan seringkali terjebak pada kondisi terminasi yang lokal optimum [1]. Bagan alur dapat dilihat pada Gambar 2.13.

Gambar 2.13 : Metode MacQueen (Disusun dari [23])

2.7.2 Metode Sarkar et al.

Metode *modified k-Means* (MKM) berbasis *evolutionary programming clustering* (EP-clustering) telah diusulkan oleh Sarkar et al [60]. Metode ini bertujuan untuk menentukan jumlah klaster pada metode k-Means. Metode ini

berbasis pencarian stokastik yang menggunakan fungsi objektif dalam memandu pencarian evolusioner, fungsi objektif yang digunakan adalah DBI. Awalnya, EP-*clustering* perlu menginisialisasi kromosom untuk pembangkitan populasi. Populasi dibangkitkan dengan cara acak (*uniform distribution*). Dengan demikian, serangkaian pusat klaster dibentuk, di mana masing-masing kelompok memiliki sejumlah pusat klaster. Jumlah pusat klaster di setiap set ditentukan secara acak. Set ini disebut orang tua. Seluruh kumpulan data kemudian dikelompokkan berdasarkan kumpulan pusat klaster induk dengan metode k-Means yang dimodifikasi. Kemudian menghitung nilai objektif untuk setiap set induk. Kemudian dimutasi menggunakan mutasi parametrik di mana setiap orang tua diperbolehkan untuk menciptakan satu keturunan (*offspring*), sehingga (pada fase seleksi) terbentuk kumpulan anak-anak dari pusat klaster. Setelah membuat *offspring*, seluruh kumpulan data dikelompokkan lagi oleh MKM berdasarkan himpunan anak pusat klaster. Kemudian nilai objektif setiap set himpunan diukur. Sebagai hasilnya, kita memperoleh dua kumpulan kelompok yang terdiri dari orang tua dan anak. Selanjutnya, melakukan kompetisi (fase mutasi), nilai objektif dari semua himpunan (orang tua dan juga keturunan atau anak) dibandingkan untuk mendapatkan sejumlah solusi global optimal. Proses ini diulangi hingga terminasi iterasi terpenuhi. Namun metode ini masih bergantung pada metode jarak *euclidean* antara data masukan dan pusat klaster [60], sehingga pendekatan pengklasteran ini hanya dapat diterapkan jika kedekatan jarak antar data menggunakan pendekatan *euclidean*. Bagan alur dapat dilihat pada Gambar 2.14

Gambar 2.14 : Metode Sarkar et al (Disusun dari [60])

2.7.3 Metode Bandyopadhyai dan Maulik

Metode *genetic clustering for unknown K* (GCUK) telah diusulkan oleh [44] menggunakan skema pengkodean dan DBI untuk merepresentasikan dan mencari jumlah klaster yang optimal. Metode GCUK bekerja dengan cara mereprestasikan sejumlah k dengan *string representation* $[K_{min}, K_{max}]$ di mana masing-masing gen mewakili pusat aktual klaster dan ditentukan secara acak. Kemudian populasi dibangkitkan pada ambang batas $[K_{min}, K_{max}]$ dengan cara random dari data set. Kromosom diperbaharui lalu hitung kesesuaian kromosom dengan fungsi objektif menggunakan DBI. Perbaharui posisi kromosom terbaru. Kemudian ketahap *crossover* dan mutasi untuk mendapatkan seleksi kromosom (pusat klaster) terbaik. Sampai kriteria terminasi terpenuhi akan kembali ketahap pembangkitan populasi dan selesai. Bagan alur dapat dilihat pada Gambar 2.15.

Gambar 2.15 : Metode Bandyopadhyay dan Maulik (Disusun dari [44])

2.7.4 Metode Omran et al.

Omran et al [46] telah mengusulkan metode pengklasteran PSO, diberi nama *dynamic clustering pso* (DCPSO) berdasarkan skema pengkodean aktivasi kontinyu dan *validity index* (VI). Metode ini memilih pusat kluster secara acak dari data dan masing-masing partikel diinisialisasi secara acak dengan [0, 1]. Lalu, hitung nilai objektif tiap partikel menggunakan VI kemudian update posisi untuk membandingkan posisi partikel dalam mendapatkan posisi partikel terbaik, diulangi sampai jumlah iterasi terpenuhi. Selanjutnya, menerapkan k-Means untuk memperbaiki posisi pusat kluster S_{best} . Setelah S_{best} diperbarui, prosedur diulang sampai kriteria terminasi terpenuhi. Dibandingkan dengan SOM dan SNOB, DCPSO mencapai hasil yang baik [32]. Bagan alur dapat dilihat pada Gambar 2.16.

Gambar 2.16 : Metode Omran et al (Disusun dari [46])

2.7.5 Metode Kuo et al.

Kuo et al pada tahun 2012 [61] telah mengusulkan metode *dynamic clustering approach based on particle swarm optimization and genetic algorithm* (DCPG). Metode ini pengembangan dari metode DCPSO [46]. Metode DCPG

bekerja berdasarkan skema pengkodean berbasis aktivasi biner dan jarak intra kluster dengan cara dimulai dari sekelompok pusat kluster M dipilih secara acak dari himpunan titik data Z . Sekelompok partikel S , kemudian di inisialisasi secara acak. Panjang partikel akan sama dengan jumlah pusat kluster M . Dalam metode DCPG, metode PSO dan GA digunakan untuk menemukan pusat kluster yang lebih baik dari sejumlah iterasi tertentu secara bersamaan, individu terbaik yang ditemukan oleh PSO dan GA disimpan untuk generasi berikutnya. Metode PSO memiliki memori sementara yang berfungsi melakukan koreksi berdasarkan solusi optimal saat ini dengan *velocity* konvergen yang cepat. Namun bila partikel optimal adalah solusi yang optimum lokal, solusi yang optimal global tidak dapat ditemukan. Metode GA digunakan untuk mengatasi kelemahan metode PSO. Jadi, *parent* yang dihasilkan melalui PSO kemudian akan melewati operator *crossover* dan mutasi di GA sehingga menghasilkan induk iteratif dari seleksi elit kromosom. Kemudian menerapkan k-Means untuk memperbaiki posisi kluster, posisi kluster M diatur ke M_a di mana M_a adalah kumpulan pusat kluster yang dipilih secara acak dari Z . Proses diulangi sampai iterasi yang ditentukan terpenuhi, maka prosesnya berhenti dan M_a akan menjadi solusi akhir. Hasil penelitian [61] menunjukkan bahwa DCPG lebih baik dari pada ACMPSO dan DCPSO. Bagan alur dapat dilihat pada Gambar 2.17.

Gambar 2.17 : Metode Kuo et al. (Disusun dari [61])

2.7.6 Metode Kuo dan Sulvia

Metode ACPSO telah diusulkan pada tahun 2013 oleh [47]. Metode ini berbasis PSO berdasarkan pengkodean aktivasi kontinyu dan *validity index* (VI). Metode ini bekerja dengan cara dataset dinormalisasi lalu populasi partikel dibangkitkan secara *random*. Selanjutnya, hitung nilai objektif tiap partikel kemudian evaluasi jumlah klaster aktif dan ukuran tiap partikel. Kemudian update/perbaiki posisi pusat klaster dengan menerapkan k-Means. Lalu, pembangkitan kecepatan atau *velocity* tiap partikel, iterasi berjalan untuk memilih partikel terbaik, lalu update posisi partikel. Hitung nilai objektif tiap partikel yang telah diupdate. Terapkan kembali k-Means untuk memperbaiki posisi klaster yang diakibatkan dari efek konvergensi prematur dari kondisi awal. Update P_{best} dan g_{best} . Kondisi diulang sampai iterasi terminasi dipenuhi. Pendekatan yang diusulkan dibandingkan dengan DCPSO, DCPG dan DCGA pada dataset benchmark *iris*, *wine*, *glass* dan *aggregation*. Hasil penelitian menunjukkan bahwa pendekatan yang diusulkan dan DCPG lebih baik dari yang lain. Bagan alur dapat dilihat pada Gambar 2.18.

Gambar 2.18 : Metode Kuo dan Sulvia (Disusun dari [47])

2.7.7 Metode Xiao et al.

Xiao et al [62] telah mengusulkan metode *the quantum-inspired genetic algorithm for k-means clustering* (KMQGA). Metode ini menggunakan representasi Q-bit dan konsep komputasi kuantum. Empat operasi utama dan operasi akses (seleksi GA, crossover GA, mutasi GA, operasi rotasi kuantum dan operasi bencana kuantum) dilakukan untuk mencari solusi partisi optimal untuk dataset tertentu. Dan panjang string Q-bit dapat diubah dalam operasi crossover. Karena perubahan ini, partisi yang dilambangkan dengan kromosom disesuaikan. Dengan demikian, ruang pencarian metode cukup besar untuk mendapatkan solusi optimal setelah beberapa iterasi evolusi. Bagan alur dapat dilihat pada Gambar 2.19.

2.7.8 Metode Das et al.

Das et al [42] mengusulkan pendekatan *automatic clustering diferensial evolution* (ACDE), di mana parameter tingkat F dan *crossover rate* ditentukan secara adaptif. Skema representasi ACDE dipilih sebagai pengkodean berbasis aktivasi kontinyu, dan fungsi objektif dipilih sebagai fungsi objektif menggunakan DBI dan CS secara individual. Keberhasilan metode ACDE dilakukan dengan membandingkannya dengan DCPSO, DE klasik dan GCUK pada kumpulan gambar dan kumpulan data benchmark. Das et al. [167] mengemukakan bahwa sebagian besar metode yang ada yang menggunakan jarak *euclidean* dapat mencapai hasil yang cukup dalam dataset yang hampir bersifat *hyper-spectral* dan linear, namun kinerjanya dapat memburuk dalam klaster yang tumpang tindih. Untuk mengatasi masalah ini, Das et al. [168, 169] mengusulkan fungsi tujuan berbasis kernel, yang merupakan reformulasi dari indeks XB menggunakan metrik jarak yang dikomunikasikan. Fungsi berbasis kernel ini diminimalkan dengan varian DE berdasarkan strategi mutasi lingkungan (DEGL) dengan pengkodean berbasis aktivasi kontinyu. Das et al. [170] juga merumuskan kembali indeks CS menggunakan metrik jarak yang dikomunikasikan. Indeks CS yang terkomputerisasi ini juga digunakan pada [171]. Bagan alur dapat dilihat pada Gambar 2.20.

Gambar 2.20 : Metode Das et al. (Disusun dari [42])

2.7.9 Metode Kuo et al.

Metode *automatic clustering approach based on differential evolution combining with k-Means for scrip clustering* (ACDE-k-means) telah diusulkan oleh Kuo et al [51] dimana kombinasi algoritma DE dan algoritma k-Means dapat menyeimbangkan proses eksplorasi dan eksploitasi pencarian. Hasil pengklasteran dievaluasi menggunakan *validity index* (VI) dan *cosine similarity measure* (CS) untuk mmenentukan jumlah cluster dan menemukan partisi terbaik. Bagan alur dapat dilihat pada Gambar 2.21.

Gambar 2.21 : Metode Kuo et al. (Disusun dari [51])

2.8 Metode-metode yang Paling Banyak Digunakan dalam Topik Penentuan Jumlah Klaster pada Metode k-Means

Ada tiga metode yang sering digunakan oleh banyak peneliti sebelumnya untuk penentuan jumlah klaster pada metode k-Means adalah: 1) metode *genetic clustering for unknown K* (GCUK) yang telah diusulkan oleh Bandyopadhyay et al [44], 2) metode *dynamic clustering pso* (DCPSO) yang telah diusulkan oleh Omran et al [46], dan 3) metode *automatic clustering differential evolution* (ACDE) yang telah diusulkan oleh Das et al [42]. Tabel 2.7 menunjukkan rangkuman metode yang paling banyak dikutip dan berpengaruh dalam topik penentuan jumlah klaster pada metode k-Means.

Tabel 2.7 : Rangkuman Metode-metode yang Sering Digunakan dalam Topik Penentuan Jumlah Klaster pada Metode k-Means

Studi utama	Dataset	Metode penentuan jumlah klaster	Metode Penentuan pusat klaster	Hasil
Bandyopadhyay dan Maulik [44]	Publik	GCUK	Tidak ada	Metode GCUK juga dapat mendeteksi secara otomatis pengklasteran data yang sesuai. Untuk data gambar berdimensi tinggi, metode ini menghasilkan jumlah klaster optimal.
Omran et al [46]	Publik	DCPSO	Acak	Metode DCPSO dapat mengklasterkan data tanpa mewajibkan pengguna untuk menentukan jumlah klaster sebelumnya. Metode ini dapat bekerja dengan pengukuran indeks klaster apapun dan menghasilkan jumlah klaster yang optimal.
Das et al [42]	Publik	ACDE	Tidak Ada	Metode yang diusulkan dapat secara otomatis menemukan jumlah klaster di dalam data bahkan untuk pengklasteran data berdimensi tinggi. Metode ACDE menghasilkan jumlah klaster yang optimal.

2.8.1 Metode Bandyopadhyay dan Maulik

Bandyopadhyay dan Maulik [44] telah menerbitkan sebuah studi sebuah penggabungan metode genetik (GA) dan k-Means untuk pengklasteran data yang belum diketahui informasi data secara apriori. Metode ini merupakan tindak lanjut dari metode MacQueen [23] pada penentuan jumlah kluster pada metode k-Means. Mereka menggunakan dataset dari repositori UCI [63], yang berisi 6 dataset yang berbeda (4 dataset sintetik dan 2 dataset real). Penelitian mereka menunjukkan GCUK dapat secara otomatis mengembangkan nilai k yang tepat serta menyediakan jumlah pengelompokan yang sesuai. Kromosom mengkodekan pusat sejumlah kluster, yang nilainya dapat bervariasi. Versi modifikasi dari operator *crossover* dan mutasi digunakan. Indeks validitas kluster seperti *davies bouldin index* (DBI) digunakan untuk menghitung kebugaran kromosom. Efektivitas teknik pengelompokan ditunjukkan dari beberapa uji data real dan data sintetik yang digunakan. Kemudian uji metode pada data gambar SPOT image menunjukkan bahwa metode tersebut mampu secara otomatis mengidentifikasi beberapa jenis tutupan lahan bahkan ketika ukuran kumpulan data secara signifikan besar (gambar SPOT memiliki 262.144 piksel atau titik data dalam ruang 2-D). Gambar 2.22 menunjukkan kerangka metode GCUK dari metode Bandyopadhyay dan Maulik.

Gambar 2.22 : Metode Bandyopadhyay dan Maulik (Disusun dari [44])

2.8.2 Metode Omran et al.

Omran et al. [46] juga melakukan tindak lanjut terhadap hasil penelitian MacQueen [23] mengenai penentuan jumlah klaster pada metode k-Means. Das et al mengusulkan *automatic clustering differential evolution* (ACDE) untuk mengatasi penentuan jumlah klaster pada metode k-Means. Metode DCPSO diuji pada lima dataset yang bersumber dari repositori UCI di antaranya *iris*, *wine*, *breast cancer*, *vowel*, dan *glass*. Metode DCPSO dapat bekerja dengan indeks validitas apa pun. Selain itu, pendekatan yang diusulkan dapat digunakan dengan metode genetika (GA), dan telah dibandingkan dengan teknik pengelompokan lain seperti DCGA dan DCRS. Dari percobaan ini dapat disimpulkan bahwa DCPSO menggunakan indeks validitas (VI) yang diusulkan oleh Turi [64] telah mengungguli pendekatan lain seperti DCGA dan DCRS. Secara umum, DCPSO berhasil menemukan jumlah klaster optimal pada gambar yang diuji (*lenna*, *mandrill*, *jet*, *peppers*, *MRI*, *tahoe*). Gambar 2.23 menunjukkan kerangka metode DCPSO dari metode Omran et. al.

Gambar 2.23 : Metode Omran et al. (Diambil dari [46])

2.8.3 Metode Das et al.

Das et al. [42] juga melakukan tindak lanjut dari metode k-Means mengenai penentuan jumlah kluster. Das et al., mengusulkan *automatic clustering differential*

evolution (ACDE) untuk mengatasi penentuan jumlah klaster pada metode k-Means. Metode ACDE diuji pada lima dataset yang bersumber dari repositori UCI di antaranya *iris*, *wine*, *breast cancer*, *vowel* dan *glass*. Metode ACDE dapat bekerja dengan indeks validitas apa pun mirip dengan metode DCPSO. Percobaan dilakukan dengan membandingkan metode ACDE dengan metode GCUK dan DCPSO menggunakan fungsi objektif *davies bouldin index* (DBI) dan *cosine similarity measure* (CS). Metode ACDE yang telah diusulkan dapat mengungguli dua metode *state-of-the-art* lainnya seperti GCUK dan DCPSO. Gambar 2.24, menunjukkan metode yang diusulkan oleh Das et al [42].

Gambar 2.24 : Metode Das et al. (Diambil dari [42])

2.9 Daftar Referensi *Systematic Literature Review* dan Peta Pikiran Lengkap

Daftar studi utama disajikan pada Tabel 2.8, daftar ini terdiri dari 5 atribut (tahun, studi utama, publikasi, dataset, dan metode) dan 39 studi utama (dari Januari 2010 hingga Desember 2016) berdasarkan tahun publikasi. Gambar 2.25 menunjukkan peta pikiran lengkap yang menyajikan hasil dari *systematic literature review* dalam topik penentuan jumlah kluster pada metode k-Means.

Tabel 2.8: Daftar Referensi *state-of-the-art* dalam Topik Penentuan Jumlah Kluster pada Metode k-Means

Tahun	Studi Utama	Publikasi	Dataset	Metode Usulan
2007	Amir Ahmad [31]	Data and Knowledge Engineering	Publik	ALGO_DISTANCE
	E. R. Hruschka [65]	Information Sciences	Publik	EAC
	Julia Handl [66]	IEEE Transactions on Evolutionary Computation	Publik	MOCK
2008	A.R.M. Faruquzzaman [39]	IEEE Region 10 Annual International Conference, Proceedings/TENCON	Publik	OSSM
	Swagatam Das [42]	IEEE Transactions On Systems, Man, and Cybernetics—Part A: Systems and Human	Publik	ACDE
	Bandyopadhyay et al [67]	IEEE Transactions on Knowledge and Data Engineering	Publik	VGAPS-clustering
2009	M. M Hasan [37]	IEEE Region 10 Annual International Conference, Proceedings/TENCON	Privat	OSP
	Saha & Bandyopadhyay [68]	Information Sciences	Publik	Fuzzy-VGAPS
	Das et al [69]	IEEE Congress on Evolutionary Computation, CEC 2009	Publik	ACBEA
	Paoli et al [70]	IEEE Transactions on Geoscience and Remote Sensing	Publik	MOPSO
	Ma et al [71]	Progress in Natural Science: Materials International	Publik	IDCMC
2010	Xiao et al [72]	Expert Systems with Applications	Publik	KMKGA
	Saha & Bandyopadhyay [73]	Pattern Recognition	Pubik	VAMOSA
2011	Bai et al [25]	Knowledge-Based Systems	Publik	Potential Exemplar
	Liu et al [74]	Applied Mathematics and Computation	Publik	AGCUK
	Horta et al [75]	Theoretical Computer Science	Publik	F-EARFC
2012	Liang et al [76]	Pattern Recognition	Publik	Modified K-Prototypes
	Fang & Wang [77]	Computational Statistics & Data Analysis	Publik	Bootstrap
	Cura [78]	Expert Systems with Applications	Publik	K-NM-PSO
	Kuo et al [61]	Information Sciences	Publik	DCPG

Tahun	Studi Utama	Publikasi	Dataset	Metode Usulan
	Nanda et al [79]	SHUSER 2012 - 2012 IEEE Symposium on Humanities, Science and Engineering Research	Publik	MOCLONAL
2013	Gou et al [80]	Neurocomputing	Publik	ME-ICQC
	Xia et al [81]	Pattern Recognition	Publik	MOEASSC
	Nanda & Panda [82]	Engineering Applications of Artificial Intelligence	Publik	MOIMPSO
	Saha & Bandyopadhyay [83]	Applied Soft Computing Journal	Publik	GenClustMOO
2014	Zhang et al [84]	Fuzzy Sets and Systems	Publik	WGLI
	Rahman & Islam [26]	Knowledge-Based Systems	Publik	GenClust
2015	Kolesnikov et al [85]	Pattern Recognition	Publik	RD, PCF
	Lughofer & Mouchaweh [86]	Information Sciences	Publik	eVQ-AMS
	Saleem et al [87]	Computer Networks	Publik	SMART
	Ozturk et al [13]	Applied Soft Computing Journal	Publik	DisABC
	Ozturk et al [88]	Information Sciences	Publik	IDisABC
	Liu et al [89]	Applied Soft Computing Journal	Publik	DLSIAC
	Peng et al [90]	Pattern Recognition Letters	Publik	Membran System
2016	Mur et al [91]	Expert System with Application	Privat	GSWA, GSWB, Nyström WB
	Kar & Maity [92]	International Conference on Informatics, Electronics and Vision (ICIEV)	Private	DE based optimal clustering
	Kuo & Zulvia [93]	IEEE Congress on Evolutionary Computation (CEC)	Publik	GE-based k-Means
	Lezama et al [94]	International Conference on Informatics, Electronics and Vision (ICIEV)	Private	ADE-LPC
	Das R & Saha [95]	2016 IEEE Congress on Evolutionary Computation (CEC)	Publik	Fuzzy-VMODEPS

Gambar 2.25 : Peta Pikiran Lengkap dari SLR untuk Penentuan Jumlah Kluster pada Metode k-Means

BAB 3

METODE PENELITIAN

3.1 Perancangan Penelitian

Pada penelitian ini menggunakan metode penelitian eksperimen. Metode penelitian eksperimen adalah uji coba yang dilakukan oleh peneliti sendiri untuk melakukan investigasi hubungan sebab akibat. Tahapan pada penelitian ini dapat dilihat pada Gambar 3.1.

Gambar 3.1 : Tahapan Penelitian, Aktifitas dan Relasi Bab

3.2 Analisis Masalah dan Tinjauan Pustaka

Pada Gambar 3.1 dapat dilihat bahwa pada penelitian ini akan mencoba untuk meninjau beberapa publikasi penelitian sebagai langkah awal. Metode tinjauan pustaka yang digunakan adalah metode *systematic literature review* (SLR) seperti yang diusulkan oleh [56]. SLR adalah sebuah proses identifikasi, menilai, dan menginterpretasikan semua penelitian dengan tujuan untuk menjawab pertanyaan

penelitian tertentu. Metode-metode yang telah diusulkan untuk topik penentuan jumlah kluster pada metode k-Means diidentifikasi berdasarkan teknik SLR. Hasil dari SLR dalam topik penentuan jumlah kluster pada k-Means telah dijelaskan di bab 2.

Hasil dari SLR pada bab 2 adalah merangkum metode-metode yang pernah diusulkan oleh para peneliti untuk topik penentuan jumlah kluster pada metode k-Means. Dari rangkuman tersebut dapat diidentifikasi kelebihan dan kekurangan metode yang pernah diusulkan dan pada akhirnya dijadikan landasan permasalahan sehingga menjadi dasar untuk membuat metode usulan yang akan digunakan pada penelitian ini.

3.3 Pengumpulan Dataset

Penelitian ini menggunakan dataset publik dari repositori UCI yang dapat diunduh melalui situs <https://archive.ics.uci.edu/>. Karakteristik masing-masing dataset yang digunakan pada penelitian ini menampilkan kelas, atribut, jumlah *record* dan *missing value* seperti yang ditunjukkan pada Tabel 3.1. Pada penelitian ini dataset yang digunakan bersumber dari repositori UCI di antara dataset yang digunakan adalah dataset *synthetic*, *iris*, *wine*, *glass*, *vowel* ditambah satu dataset *ruspini* yang merupakan dataset yang populer digunakan untuk mengilustrasikan dan mengevaluasi teknik pengklasteran [2].

Tabel 3.1 : Identifikasi Datasest

Dataset	Kelas	Atribut	Jumlah <i>Record</i>	<i>Missing Value</i>
<i>Synthetic</i>	3	2	300	0
<i>Iris</i>	3	4	150	0
<i>Wine</i>	3	13	178	0
<i>Glass</i>	6	9	214	0
<i>Vowel</i>	11	13	990	0
<i>Ruspini</i>	4	2	75	0

3.4 Metode yang Diusulkan dalam Topik Penentuan Jumlah Klaster pada Metode k-Means

Metode yang diusulkan disajikan pada Gambar 3.2. Perbedaan utama antara metode yang diusulkan dan metode lainnya adalah pada blok penentuan jumlah klaster pada metode k-Means. Skema pembelajaran yang diusulkan terdiri dari 1) metode pencarian, dan 2) fungsi objektif. Bandyopadhyay dan Maulik menggunakan GCUK sebagai metode penentuan jumlah klaster dan DBI sebagai fungsi objektif. Omran et al. menggunakan DCPSO sebagai metode penentuan jumlah klaster dan VI sebagai fungsi objektif. Das et al. menggunakan ACDE sebagai metode penentuan jumlah klaster dan DBI dan CS sebagai fungsi objektif.

Metode yang diusulkan adalah metode UCC digabungkan dengan metode ACDE untuk penentuan jumlah klaster pada metode k-Means. Fungsi objektif yang digunakan adalah validasi klaster internal menggunakan DBI dan CS untuk memandu perhitungan pencarian evolusi populasi. Gambar 3.2 menunjukkan perbandingan metode yang diusulkan dengan metode yang pernah diusulkan peneliti lain untuk penentuan jumlah klaster pada metode k-Means.

Tabel 3.2 : Perbandingan Metode Penentuan Jumlah Klaster pada Metode k-Means

Peneliti	Metode	Penentuan jumlah k klaster		Penentuan Pusat Klaster
		<i>Strategi</i>	<i>Fungsi Objektif</i>	
Bandyopadhyay dan Maulik	GCUK	Otomatis	DBI	Tidak Ada
Omran et al.	DCPSO	Otomatis	VI	Ada
Swagatam Das	ACDE	Otomatis	DBI dan CS	Tidak Ada
Metode Usulan	UCC+ACDE+k-Means	Otomatis	DBI dan CS	Ada

Gambar 3.2 : Metode Penentuan Jumlah Kluster pada Metode k-Means yang Diusulkan

Metode k-Means dalam mengelompokkan sebuah data dikenal dua bagian proses yang sangat penting, yaitu: 1) penentuan jumlah klaster dan (2) penentuan pusat awal klaster. Masing-masing bagian memiliki peranan penting untuk meningkatkan kinerja pengklasteran pada metode k-Means. Dalam penelitian ini akan fokus mengatasi masalah dalam penentuan jumlah klaster pada metode k-Means di mana pendekatan metode *evolutioner computation* (EC) akan diterapkan untuk penentuan jumlah klaster dengan cara otomatis.

Pada penelitian ini akan diusulkan penggabungan metode *u-control chart* (UCC) dan *automatic clustering differential evolution* (ACDE) untuk penentuan jumlah klaster pada metode k-Means (UCC+ACDE+k-Means). Metode UCC digunakan untuk penentuan ambang batas aktivasi k pada metode ACDE yang digunakan sebagai sampling populasi awal di mana k yang akan dicari kira-kira akan berada. Kemudian ACDE menggunakan fungsi objektif untuk mengontrol pencarian k klaster dengan bantuan operator produksi (mutasi, *crossover* dan seleksi) dan membimbing pencarian k klaster secara evolusioner menggunakan *davies bouldin index* (DBI) dan *cosine similarity measure* (CS).

Tahapan metode yang diusulkan untuk menentukan jumlah klaster pada metode k-Means adalah sebagai berikut (*pseudocode* dapat dilihat pada Lampiran 5):

Step 1. Menyiapkan dataset:

Pada contoh ini akan digunakan dataset *iris*. Jumlah kelas pada dataset *iris* adalah tiga yaitu kelas 1 (*iris sentosa*), kelas 2 (*iris versicolour*) dan kelas 3 (*iris virginia*). Pada perhitungan ini label/ kelas data tidak akan digunakan atau atribut label/ kelas dihapus untuk menguji performa metode yang diusulkan dalam mencari jumlah klaster optimal pada data *iris*. Tabel 3.3 menunjukkan sebagian dataset *iris*.

Tabel 3.3 : Data *Iris*

<i>id</i>	<i>Sepal Length</i> (cm)	<i>Sepal Width</i> (cm)	<i>Petal Length</i> (cm)	<i>Petal Width</i> (cm)	Kelas
1	5.1	3.5	1.4	0.2	<i>Iris Sentosa</i>
2	4.9	3	1.4	0.2	<i>Iris Sentosa</i>
3	4.7	3.2	1.3	0.2	<i>Iris Sentosa</i>
4	4.6	3.1	1.5	0.2	<i>Iris Sentosa</i>

<i>id</i>	<i>Sepal Length (cm)</i>	<i>Sepal Width (cm)</i>	<i>Petal Length (cm)</i>	<i>Petal Width (cm)</i>	Kelas
5	5	3.6	1.4	0.2	<i>Iris Sentosa</i>
...					
51	7	3.2	4.7	1.4	<i>Iris Versicolor</i>
52	6.4	3.2	4.5	1.5	<i>Iris Versicolor</i>
53	6.9	3.1	4.9	1.5	<i>Iris Versicolor</i>
54	5.5	2.3	4	1.3	<i>Iris Versicolor</i>
55	6.5	2.8	4.6	1.5	<i>Iris Versicolor</i>
...					
101	6.3	3.3	6	2.5	<i>Iris Virginica</i>
102	5.8	2.7	5.1	1.9	<i>Iris Virginica</i>
103	7.1	3	5.9	2.1	<i>Iris Virginica</i>
104	6.3	2.9	5.6	1.8	<i>Iris Virginica</i>
105	6.5	3	5.8	2.2	<i>Iris Virginica</i>
...	...	...	...	...	...

Step 2. Inisialisasi Kromosom

a) Setup Parameter

Sebelum melakukan inisialisasi kromosom, terlebih dahulu dilakukan setting parameter, seperti yang ditunjukkan pada Tabel 3.4.

Tabel 3.4 : Setting Parameter Metode Usulan

Nama Parameter	Nilai
Maksimal iterasi	200
Besarnya populasi	50
Probabilitas <i>Crossover</i>	0.2
Banyaknya klaster awal	10

b) Tentukan ambang batas aktivasi k

Untuk menentukan batas atas dan batas bawah variabel akan digunakan metode UCC pada metode ACDE dengan mengikuti persamaan 3.1, 3.2 dan 3.3. Tabel 3.5 menunjukkan hasil perhitungan dari *u-control chart* (UCC) pada data *iris*. Berikut ini adalah keterangan dari tahap-tahap untuk menentukan ambang batas aktivasi k , adalah sebagai berikut:

1) Hitung $\bar{u}$ -bar dengan mengikuti persamaan 3.1

Mencari nilai rata-rata data/ *center line*

$$\bar{u} = \frac{\sum x_i}{\sum n_i} \quad (3.1)$$

Di mana:

$\bar{u}$ adalah nilai rata-rata semua data

x_i adalah *record* data

n_i adalah banyaknya *record* data

$$\begin{aligned} \bar{u}_{x1} &= \frac{51 + 49 + 79 + \dots + 65 + 62 + 59}{150} = 58.4333 \\ \bar{u}_{x2} &= \frac{35 + 30 + 32 + \dots + 30 + 34 + 30}{150} = 30.5733 \\ \bar{u}_{x3} &= \frac{13 + 15 + 14 + \dots + 52 + 54 + 51}{150} = 37.5800 \\ \bar{u}_{x4} &= \frac{51 + 49 + 79 + \dots + 65 + 62 + 59}{150} = 11.9933 \\ \bar{u}_{xi} &= \frac{58.4333 + 30.5733 + 37.5800 + 11.9933}{4} = 34.645 \end{aligned}$$

2) Kemudian, hitung simpangan baku K megikuti persamaan 3.2:

$$K = \sqrt{\frac{(x_i - \bar{x})^2}{n - 1}} \quad (3.2)$$

Dimana K adalah standar deviasi

$$\begin{aligned} K_1 &= \sqrt{\frac{(51 - 58.4333)^2 + \dots + (59 - 58.4333)^2}{(150 - 1)}} = 8.2807 \\ K_2 &= \sqrt{\frac{(35 - 30.5733)^2 + \dots + (30 - 30.5733)^2}{(150 - 1)}} = 4.3587 \\ K_3 &= \sqrt{\frac{(13 - 37.5800)^2 + \dots + (51 - 37.5800)^2}{(150 - 1)}} = 17.6530 \\ K_4 &= \sqrt{\frac{(51 - 11.9933)^2 + \dots + (59 - 11.9933)^2}{(150 - 1)}} = 7.6224 \\ K &= \frac{8.2807 + 4.3587 + 17.653 + 7.6224}{4} = 5.7130 \end{aligned}$$

3) Hitung nilai batas bawah menggunakan persamaan 3.3:

$$bb = \bar{u} - K \sqrt{\frac{\bar{u}}{n_i}} \quad (3.3)$$

$$BB = 34.6450 - \left[(5.7130) \sqrt{\frac{34.645}{150}} \right] = 31.90$$

4) Hitung batas atas atribut menggunakan persamaan 3.4 adalah sebagai berikut:

$$bb = \bar{u} + K \sqrt{\frac{\bar{u}}{n_i}} \quad (3.4)$$

Tabel 3.5 : Hasil Perhitungan Metode UCC

	Id	Atr 1	Atr 2	Atr 3	Atr 4	
	1	51	35	14	2	
	2	49	30	14	2	
	3	47	32	13	2	
	4	46	31	15	2	
	5	50	36	14	2	
	...					
	26	50	30	16	2	
	27	50	34	16	4	
	28	52	35	15	2	
	29	52	34	14	2	
	30	47	32	16	2	
	...	...	...	...	...	
	146	67	30	52	23	
	147	63	25	50	19	
	148	65	30	52	20	
	149	62	34	54	23	
	150	59	30	51	18	
	u-bar	58.4333	30.5733	37.5800	11.9933	34.6450
	K	8.2807	4.3587	17.6530	7.6224	5.7130
	Ni	150				
Control	BB	31.90				
	BA	37.39				

$$BA = 34.6450 + \left[(5.7130) \sqrt{\frac{34.645}{150}} \right] = 37.39$$

- 5) Simpan hasil ambang batas aktivasi k yang mana akan dijadikan input pada metode ACDE untuk membatasi populasi yang bertujuan agar k yang mungkin dicari tidak keluar dari batas yang telah ditetapkan

c) Bangkitkan populasi

Pada tahap ini populasi atau kromosom akan dibangkitkan secara acak berdasarkan penentuan ambang batas aktivasi k yang telah ditentukan sebelumnya oleh metode UCC berdasarkan fungsi fitness DBI

Step 3. Temukan pusat kluster aktif

Pada tahap ini akan dilakukan definisi kluster aktif dari kromosom aktif yang telah dibangkitkan dengan mengikuti persamaan 2.2. Tabel 3.6 menunjukkan contoh hasil pada iterasi pertama.

Tabel 3.6 : Contoh Hasil Posisi Kluster Aktif

Kromosom acak				Kluster Aktif
79.00	20.00	69.00	3.00	37.39
43.00	34.57	10.00	3.00	31.90
43.00	20.00	10.00	1.00	31.90
62.41	41.51	69.00	3.00	37.39
70.87	38.10	10.00	1.00	37.39
43.00	44.00	10.00	1.00	35.27
79.00	20.00	69.00	1.35	31.90
79.00	27.34	69.00	3.00	34.28
79.00	44.00	69.00	1.00	37.39
79.00	31.29	69.00	1.00	31.90

Pada Tabel 3.6 atribut “Kromosom acak” merupakan posisi kromosom yang dipilih secara acak dari variabel min, variabel max dan variabel size diacak menggunakan fungsi `unifrnd` dalam matlab. Fungsi `unifrnd` berfungsi mengembalikan sebuah array dari bilangan acak yang dihasilkan

dari distribusi seragam kontinyu dari titik batas atas dan batas bawah yang ditentukan oleh VarMin dan VarMax pada dataset asli.

Step 4. Perulangan iterasi $t = 1$

Pada tahap perulangan ini akan melewati empat langkah yaitu pertama menghitung nilai fungsi objektif setiap kromosom aktif, kedua proses mutasi, ketiga proses kawin silang dan keempat seleksi pusat klaster, berikut ini adalah pembahasan detailnya:

Step 5. Hitung nilai objektif

Untuk masing-masing kromosom hitung vektor target pada nilai menggunakan fungsi objektif DBI dan CS dalam kasus ini dijelaskan menggunakan DBI, nilai terkecil adalah nilai objektif terbaik. Tabel 3.7 menunjukkan sebagian hasil dari nilai objektif pada setiap kromosom di mana jumlah kromosom yang dipilih sebelumnya adalah 50.

Tabel 3.7 : Contoh Hasil Perhitungan Fungsi Objektif DBI dari Metode Usulan

Kromosom ke-i	Kromosom acak				Klaster Aktif	Nilai Fitnes
1	43.00	20.00	10.00	25.00	31.90	1.68
	43.00	44.00	69.00	25.00	31.90	
	79.00	27.99	69.00	1.00	31.90	
	43.00	44.00	10.00	25.00	31.90	
	79.00	44.00	10.00	1.00	31.90	
	79.00	20.00	69.00	1.00	31.90	
	79.00	44.00	23.72	25.00	31.90	
	79.00	20.00	69.00	25.00	31.90	
	43.00	20.00	69.00	25.00	31.90	
	79.00	44.00	69.00	25.00	31.90	
...	...	...	...	...	...	...
50	43.00	20.00	10.00	1.00	3.01	0.15
	43.00	20.00	69.00	25.00	3.92	
	79.00	20.00	69.00	1.00	3.92	
	79.00	44.00	10.00	1.00	3.92	
	79.00	20.00	69.00	25.00	3.01	
	43.00	20.00	69.00	16.48	3.01	

Step 6. Mutasi

Pada tahap ini, proses mutasi akan dilakukan dengan cara mengkombinasi populasi awal yang dibangkitkan dengan posisi populasi yang telah di update untuk menghasilkan populasi baru dengan ukuran N vektor percobaan. Pada Tabel 3.8, menunjukkan ilustrasi bagaimana populasi mutan terbentuk dalam populasi data. Di mana, baris pertama menunjukkan batas atas dan batas bawah yang dijadikan acuan untuk membangkitkan populasi baru secara acak dari 50 populasi yang telah ditentukan sebelumnya.

Tabel 3.8 : Source Code Tahap Mutasi Pada Metode Usulan

Metode: ACDE, Mutasi	
Step	Deskripsi
1	beta=unifrnd (beta_min,beta_max,VarSize);
2	y=pop(a).Position+beta.*(pop(b).Position-
3	pop(c).Position);
4	y=max(y,VarMin);
5	y=min(y,VarMax)

Kemudian *update* posisi populasi diikuti oleh pusat kluster aktif berdasarkan nilai batas atas dan batas bawah. Pada Tabel 3.9 menunjukkan update posisi N -populasi terbaru dari hasil objektif terbaik yang diperoleh dari proses mutasi (Tabel 3.8).

Selanjutnya, mengombinasikan antara N -populasi terbaik (dari Tabel 3.8) dan posisi update terbaru N -populasi (pada Tabel 3.9) menggunakan persamaan 3.5 sehingga hasilnya seperti yang ditunjukkan pada Tabel 3.10.

Tabel 3.9 : Hasil Update Posisi N -Populasi

N-Populasi				Pusat Kluster Aktif
43.00	44.00	69.00	25.00	37.39
79.00	20.00	69.00	1.00	31.90
79.00	26.86	69.00	25.00	31.90
79.00	44.00	69.00	25.00	37.39
43.00	20.00	69.00	1.00	31.90
...	...	...	...	...
79.00	20.00	69.00	25.00	31.90

Tabel 3.10 : Contoh Sebagian Hasil Perhitungan Mutasi pada Metode Usulan

N-Populasi (Mutasi)				
0.89	0.75	-0.61	0.01	0.15
0.16	-0.78	-1.65	-0.32	0.16
-0.66	0.70	-1.47	-0.25	0.20
-0.07	0.80	-0.92	0.74	-0.05
0.56	0.30	0.81	0.40	0.12
...	...	...	...	...
-0.70	-0.29	-0.46	0.61	0.05

Step 7. Crossover

Selanjutnya ketahap kawin silang atau *crossover*. Pada tahap ini dilakukan penyilangan setiap vektor, $y_{(i,j)}$ dengan vektor mutan $x_{(i,j)}$ untuk membentuk vektor hasil persilangan $z_{(i,j)}$ dengan persamaan 3.6.

Tabel 3.11 menunjukkan contoh hasil dari proses penyilangan vektor atau proses *crossover*. Pada baris pertama $y_{i,j,g}$ adalah klaster aktif dari nilai objektif terbaik pada 50 populasi di iterasi pertama, sedangkan pada baris kedua $x_{i,j,g}$ merupakan klaster aktif dari vektor mutan atau biasa disebut induk sel.

Tabel 3.11 : Contoh Hasil *Crossover* pada Metode Usulan

Pusat	$y_{(i,j,g)}$	31.9	31.9	31.9	37.39	37.39	31.9	31.9	37.39	31.9	31.9
Klaster	$x_{(i,j,g)}$	31.9	31.9	31.9	37.39	37.39	31.9	31.9	37.39	31.9	37.39
Aktif	$z_{(i,j,g)}$	31.9	31.9	31.9	37.39	37.39	31.9	31.9	37.39	31.9	0

Tabel 3.12 : Pusat Awal Klaster

Pusat Klaster	Atrb-1	Atrb-2	Atrb-3	Atrb-4
C1	5.0	3.2	1.5	0.2
C2	5.9	3.0	4.3	1.4
C3	6.8	3.0	5.6	2.0

Step 8. Seleksi

Cek nilai $f(z_{i,j})$ bandingkan dengan nilai $f(y_{i,j})$. Jik $z_{(i,j)} < y_{(i,j)}$ maka $y_{i,j} = z_{(i,j)}$, jika tidak maka $y_{i,j} = y_{i,j}$

Step 9. Kembali Ke-Langkah 5

Jika jumlah iterasi sudah terpenuhi, dilanjutkan ke-langkah 10

Step 10. Bangkitkan data berdasarkan jumlah kluster optimal

Setelah mendapatkan jumlah kluster optimal. Selanjutnya dari jumlah kluster optimal akan digunakan untuk memilih pusat awal kluster secara acak dalam data asli untuk dibangkitkan. Pada Tabel 3.13, menunjukkan pusat awal kluster yang telah dipilih secara acak dari data asli Iris.

Tabel 3.13 : Pusat Kluster Iterasi Pertama

Pusat kluster	Atrb-1	Atrb-2	Atrb-3	Atrb-4
C1	4.9520	3.2120	1.4640	0.2440
C2	5.8870	2.9278	4.2963	1.3593
C3	6.7609	3.0304	5.6217	2.0478

Step 11. Hitung jarak tiap data dengan pusat kluster menggunakan fungsi jarak Euclidean***Iterasi Ke-1***

Hitung jarak setiap data ke pusat kluster terdekat. Pusat kluster terdekat akan menjadi kluster yang diikuti oleh data tersebut. Tabel 3.14 menunjukkan hasil iterasi pertama.

$$d(x_1, c_1) = \sqrt{\sum_{i=1}^n (x_{1i} - c_{1g})^2} = \sqrt{(5.1 - 5.0)^2 + (3.5 - 3.2)^2 + (1.4 - 1.5)^2 + (0.2 - 0.2)^2} = 0.37$$

$$d(x_1, c_2) = \sqrt{\sum_{i=1}^n (x_{1i} - c_{2g})^2} = \sqrt{(5.1 - 5.9)^2 + (3.5 - 3.0)^2 + (1.4 - 4.3)^2 + (0.2 - 1.4)^2} = 3.28$$

$$d(x_1, c_3) = \sqrt{\sum_{i=1}^n (x_{1i} - c_{3g})^2} = \sqrt{(5.1 - 6.8)^2 + (3.5 - 3.0)^2 + (1.4 - 5.6)^2 + (0.2 - 2.0)^2} = 4.93$$

Tabel 3.14 : Iterasi Pertama

No	Data ke-i	Jarak ke pusat klaster			Jarak Terdekat	Klaster yang Diikuti
		C1	C2	C3		
1	id-1	0.37	3.28	4.93	0.37	1
2	id-2	0.19	3.29	4.98	0.19	1
3	id-3	0.31	3.45	5.15	0.31	1
4	id-45	1.88	1.76	3.35	1.76	2
5	id-47	3.50	1.21	2.15	1.21	2
6	id-49	3.35	0.97	2.04	0.97	2
7	id-127	4.24	1.07	0.81	0.81	3
8	id-128	4.16	0.97	0.86	0.86	3
9	id-134	5.25	2.08	0.78	0.78	3
10	id-135	4.41	1.36	1.02	1.02	3
...	...	...	...	...	...	...
150	id-136	4.82	2.01	1.11	1.11	3

Selanjutnya dihitung pusat klaster yang baru untuk setiap klaster berdasarkan data yang bergabung pada setiap pusat klasternya. Untuk klaster C1

$$\text{Pusat klaster } C1 = \frac{5.1 + 4.9 + 4.7 + \dots + 5.4 + 5.2 + 5.5}{50} = 4.95$$

$$\text{Pusat klaster } C2 = \frac{5.1 + 5.1 + 5.3 + \dots + 6.4 + 6.6 + 6.5}{54} = 5.89$$

$$\text{Pusat klaster } C3 = \frac{6.2 + 6.1 + 6.3 + \dots + 7.7 + 7.9 + 7.7}{46} = 6.76$$

Iterasi ke-2

Untuk menghitung jarak terdekat data dengan pusat klaster yang baru sama dengan perhitungan jarak pada iterasi sebelumnya yaitu menggunakan menggunakan jarak *Euclidean*. Tabel 3.15 dan Tabel 3.16 menunjukan perhitungan jarak pada iterasi ke-2 dan hasil pusat klaster dari iterasi ke-2.

$$\text{Pusat klaster } C1 = \frac{5.1 + 4.9 + 4.7 + \dots + 5.4 + 5.4 + 5.7}{51} = 5.01$$

$$\text{Pusat klaster } C2 = \frac{5.4 + 5.1 + 5.4 + \dots + 6.4 + 6.7 + 5.9}{62} = 5.98$$

$$\text{Pusat klaster } C3 = \frac{5.7 + 6.5 + 6.7 + \dots + 7.7 + 7.9 + 7.7}{37} = 6.78$$

Tabel 3.15 : Iterasi ke-2

No	Data ke-i	Jarak ke pusat klaster			Jarak Terdekat	Klaster yang diikuti
		C1	C2	C3		
1	id-1	0.71	3.65	5.05	0.71	1
2	id-2	0.51	3.69	5.11	0.51	1
3	id-3	0.72	3.84	5.28	0.72	1
4	id-17	3.20	1.04	2.08	1.04	2
5	id-20	3.01	1.18	2.35	1.18	2
6	id-21	1.86	1.45	2.89	1.45	2
7	id-97	4.72	1.71	1.31	1.31	3
8	id-105	4.09	1.06	0.84	0.84	3
9	id-109	4.31	1.34	0.81	0.81	3
10	id-111	4.19	1.01	0.52	0.52	3
...	...	...	...	...	...	...
150	id-136	4.42	1.73	1.07	1.07	3

Tabel 3.16 menunjukkan masih terdapat data yang berpindah dari satu klaster ke klaster lainnya. Sehingga perlu dilakukan iterasi selanjutnya hingga tidak ada data yang berpindah ke klaster lainnya. Pada kondisi ini perlu dilakukan iterasi selanjutnya untuk menghitung kembali pusat klaster dari masing-masing klaster yang terbentuk.

Tabel 3.16 : Pusat Klaster Iterasi ke-2

Pusat klaster	Atrb-1	Atrb-2	Atrb-3	Atrb-4
C1	5.01	3.03	1.49	0.26
C2	5.98	3.09	4.43	1.44
C3	6.78	3.04	5.75	2.08

Iterasi ke-3

Untuk menghitung jarak terdekat data dengan pusat klaster yang baru sama dengan perhitungan jarak pada iterasi sebelumnya dengan jarak *euclidean* ditunjukkan pada Tabel 3.17 dan Tabel 3.18 menunjukkan pusat klaster pada iterasi ketiga.

Tabel 3.17 : Iterasi ke-3

No	Data ke-i	Jarak ke pusat klaster			Jarak Terdekat	Klaster yang diikuti
		C1	C2	C3		
1	id-1	0.49	3.42	5.05	0.49	1
2	id-2	0.16	3.45	4.74	0.16	1
3	id-3	0.41	3.61	4.92	0.41	1
4	id-20	1.89	1.86	3.43	1.86	2
5	id-21	3.45	0.71	2.14	0.71	2
6	id-22	3.32	1.07	2.55	1.07	2
7	id-100	4.45	1.28	1.11	1.11	3
8	id-109	4.14	1.10	1.01	1.01	3
9	id-113	4.51	1.37	0.72	0.72	3
10	id-124	5.21	1.96	0.94	0.94	3
...	...	...	...	...	...	...
150	id-136	4.76	1.88	1.24	1.24	3

Tabel 3.18 : Pusat Klaster Iterasi ke-3

Pusat klaster	Atrb-1	Atrb-2	Atrb-3	Atrb-4
C1	4.99	3.06	1.46	0.24
C2	5.95	3.07	4.37	1.41
C3	6.76	3.03	5.71	2.07

Seperti yang ditunjukkan pada Tabel 3.18 masih terdapat data yang berpindah dari satu klaster ke klaster lainnya. Sehingga perlu dilakukan iterasi selanjutnya hingga tidak ada data yang berpindah ke klaster lainnya. Pada iterasi keempat, kelima, keenam sampai iterasi ke-10 perlu dilakukan iterasi selanjutnya untuk menghitung kembali pusat klaster dari masing-masing klaster yang terbentuk.

$$\text{Pusat klaster } C1 = \frac{5.1 + 4.9 + 4.7 + \dots + 5.4 + 5.4 + 5.4}{50} = 4.99$$

$$\text{Pusat klaster } C2 = \frac{5.1 + 5.4 + 5.1 + \dots + 5.8 + 6.4 + 6.5}{60} = 5.95$$

$$\text{Pusat klaster } C3 = \frac{5.7 + 6.7 + 6.8 + \dots + 7.7 + 7.9 + 7.7}{46} = 6.76$$

Iterasi ke-11

Untuk menghitung jarak data terdekat dengan pusat klaster yang baru sama dengan perhitungan jarak pada iterasi sebelumnya dengan jarak *Euclidean*, dapat dilihat pada Tabel 3.19. Pada Tabel 3.19 menunjukkan bahwa iterasi keempat sudah tidak terdapat data yang berpindah, sehingga proses iterasi berhenti.

Tabel 3.19 : Iterasi ke-11

No	Data ke-i	Jarak ke pusat klaster			Jarak Terdekat	Klaster yang diikuti
		<i>C1</i>	<i>C2</i>	<i>C3</i>		
1	id-1	0.33	3.34	5.02	0.33	1
2	id-2	0.23	3.35	5.07	0.23	1
3	id-3	0.30	3.51	5.24	0.30	1
4	id-20	1.86	1.83	3.43	1.83	2
5	id-21	3.49	1.25	2.24	1.25	2
6	id-22	3.34	1.01	2.13	1.01	2
7	id-100	4.27	1.03	0.80	0.80	3
8	id-109	4.28	1.04	0.85	0.85	3
9	id-113	5.25	2.07	0.82	0.82	3
10	id-124	4.47	1.35	0.81	0.81	3
...	...	...	...	...	...	...
150	id-136	4.82	1.98	1.09	1.09	3

Step 12. Selesai

Jika data sudah tidak berpindah klaster, maka perhitungan metode yang diusulkan berhenti.

3.5 Kerangka Pemikiran

Kerangka pemikiran pada penelitian ini terdiri dari beberapa tahap seperti yang ditunjukkan pada Gambar 3.3. Masalah dalam penelitian ini adalah ACDE dapat digunakan untuk menentukan jumlah klaster pada metode k-Means, akan tetapi memiliki kelemahan pada penentuan ambang batas aktivasi k di mana mempengaruhi kinerja ACDE dalam penentuan jumlah klaster pada metode k-Means. Metode *u-control chart* (UCC) digunakan untuk memperbaiki kelemahan metode ACDE pada

penentuan ambang batas aktivasi k yang masih menggunakan cara manual yang bertujuan membatasi ruang pencarian k di mana kira-kira nilai variabel yang dicari dapat ditemukan. Jumlah kluster sebanyak k yang dihasilkan oleh metode ACDE akan menjadi jumlah k input pada metode k-Means, tujuannya adalah untuk meningkatkan kinerja pengklasteran pada metode k-Means. Selanjutnya, kinerja metode akan dievaluasi menggunakan DBI dan CS sebagai validitas internal kluster. Nilai rata-rata validitas internal terendah atau mendekati nol menunjukkan kinerja metode terbaik.

Gambar 3.3 : Kerangka Pemikiran Penelitian

3.6 Eksperimen dan Pengujian Metode

Tahap eksperimen pada penelitian ini adalah sebagai berikut:

1. Menyiapkan dataset yang akan digunakan
2. Perhitungan dan eksperimen dengan metode MacQueen
3. Mengukur performa dari metode Bandyopadhyay dan Maulik
4. Mengukur performa dari metode Omran et al.
5. Mengukur performa dari metode Das et al.
6. Membandingkan hasil klastering dengan DBI dan CS sebagai evaluasi validitas internal kluster.

Dalam penelitian ini akan menggunakan program bantu software berupa Matlab 2016, Microsofot Excel 2016, Rapidminer Studio 7.5, IBM SPSS statistic 20.0.0. Spesifikasi komputer yang digunakan pada penelitian ini dapat dilihat pada Tabel 3.20.

Tabel 3.20 : Spesifikasi Komputer yang Digunakan

Processor	Intel N2840 2.16GHz
Memori	8 GB
Hardisk	500GB HDD
Sistem Operasi	Windows 10 Home 64bit
Aplikasi	Matlab 2016, Microsoft Excel 2016, Rapidminer 7.5, IBM SPSS Statistic 20.0.0

3.7 Evaluasi Hasil

Untuk membantu pencarian jumlah klaster pada metode yang diusulkan akan menggunakan validitas internal DBI dan CS sebagai fungsi objektif yang bertujuan memandu pencarian evolusioner, bagian ini dijelaskan pada sub bab 3.7.1 dan 3.7.1.2. Tahap selanjutnya menggunakan uji Friedman untuk uji beda metode terhadap semua dataset [96]. Uji Friedman akan menunjukkan peringkat metode terbaik di mana peringkat 1 menunjukkan metode terbaik pertama, peringkat 2 menunjukkan metode terbaik kedua dan seterusnya [96].

3.7.1 Evaluasi Klaster

3.7.1.1 *Davies Bouldin Index (DBI)*

DBI banyak digunakan untuk validitas internal klaster [42]. Metrik DBI pertama kali diperkenalkan oleh Davies dan Bouldin pada tahun 1979 [97]. Validitas internal yang dilakukan adalah seberapa baik klastering sudah dilakukan dengan menghitung kuantitas dan atribut turunan dari dataset. *Sum of square within cluster* (SSW) adalah metrik kohesi dalam sebuah klaster ke-*i* diformulasikan menggunakan persamaan (3.8):

$$SSW_i = \frac{1}{m_i} \sum_{j=1} d(x_j, c_i) \quad (3.8)$$

Di mana m_i adalah jumlah data yang berada dalam kluster ke- i . Sedangkan c_i adalah pusat kluster ke- i . Sementara metrik untuk separasi antara dua kluster, misalnya, kluster i dan j . *Sum of square between cluster* (SSB) digunakan untuk mengukur jarak antara pusat kluster c_i dan c_j seperti pada persamaan (3.9).

$$SSB_{i,j} = d(x_j, c_i) \quad (3.9)$$

$R_{i,j}$ didefinisikan sebagai perbandingan ukuran rasio yakni seberapa baik nilai perbandingan antara kluster ke- i dan kluster ke- j . Nilainya didapatkan dari komponen kohesi dan sparasi. Kluster yang baik adalah yang mempunyai kohesi yang sekecil mungkin dan sparasi yang sebesar mungkin [97].

$$R_{ij} = \frac{SSW_i + SSW_j}{SSB_{i,j}} \quad (3.10)$$

Nilai DBI didapatkan dari persamaan (3.11).

$$DBI = \frac{1}{K} \sum_{i=1}^K \max(R_{i,j}) \quad (3.11)$$

K adalah jumlah kluster yang ditentukan. DBI mengukur rata-rata kesamaan antara setiap kluster dan satu kluster lagi yang paling mirip. Nilai DBI terendah menunjukkan bahwa kluster yang terbentuk adalah kompak dan dapat terpisah pada kluster yang tepat.

3.7.1.2 Cosine Similarity Measure (CS)

CS pertama kali diperkenalkan oleh Chou et al [98] digunakan untuk mengevaluasi atau validitas hasil pengklasteran. CS digunakan untuk memperoleh jumlah kluster akhir dan melakukan partisi data secara otomatis. Sebelum menerapkan CS pusat kluster dari sebuah kluster dihitung dengan rata-rata vektor data yang dimiliki oleh kluster tersebut dengan persamaan (3.12).

$$\vec{m}_i = \frac{1}{N_i} \sum_{x_i \in C_i} \vec{x}_j \quad (3.12)$$

Metrik jarak antara dua data, point X_i dan X_j dinyatakan dengan $d(\vec{X}_i, \vec{X}_j)$ maka CS didefinisikan seperti pada persamaan (3.13).

$$\begin{aligned}
CS(K) &= \frac{\frac{1}{K} \sum_{i=1}^K \frac{1}{N_i} \sum_{\vec{X}_i \in C_i} \max_{\vec{X}_q \in C_i} \{d(\vec{X}_i, \vec{X}_q)\}}{\frac{1}{K} \sum_{i=1}^K \left[\min_{j \in K, j \neq i} \{d(\vec{m}_i, \vec{m}_j)\} \right]} \\
&= \frac{\sum_{i=1}^K \left[\frac{1}{N_i} \sum_{\vec{X}_i \in C_i} \max_{\vec{X}_q \in C_i} \{d(\vec{X}_i, \vec{X}_q)\} \right]}{\sum_{i=1}^K \left[\min_{j \in K, j \neq i} \{d(\vec{m}_i, \vec{m}_j)\} \right]}
\end{aligned} \tag{3.13}$$

Ukuran CS merupakan fungsi dari jumlah rasio penyebaran kluster sampai ke pemisahan kluster hingga mencapai batas minimum. Menurut Chou et al [98] ukuran CS cukup efisien dalam mengatasi kelompok dengan ukuran yang berbeda dari pada ukuran validitas lainnya seperti Dunn Index dan DBI. Namun waktu komputasi yang tinggi seiring dengan peningkatan K dan n .

3.7.2 Uji Hasil Evaluasi Beberapa Metode Menggunakan Uji Statistik

Beberapa tahun terakhir penggunaan uji statistik non parametrik “uji Friedman” menjadi populer dikalangan peneliti *evolutioner computation* berbasis klustering [8], [73], [83], [99], di mana digunakan untuk perbandingan metode yang tidak membutuhkan asumsi distribusi normal dan varians populasi yang tidak diketahui, hal ini sesuai yang direkomendasikan oleh Demsar [96]. Berdasarkan kepopuleran dan rekomendasi tersebut, uji Friedman akan digunakan dalam penelitian ini untuk membandingkan hasil evaluasi DBI dan CS dari semua metode. Uji Friedman adalah uji statistik non parametrik yang dikembangkan pertama kali oleh Milton Friedman [100]. Prosesnya melibatkan peringkat setiap metode di setiap baris, kemudian mempertimbangkan nilai peringkat berdasarkan kolom. Uji Friedman didasarkan pada *mean of rank (R)*. Uji Friedman dihitung dengan persamaan (3.14).

$$X_F^2 = \frac{12D}{C(C+1)} \left[\sum_j^D R_j^2 - \frac{C(C+1)}{4} \right] \tag{3.14}$$

Dimana :

- r_i^j adalah peringkat ke- j setiap metode C pada dataset D ke i .

- $R_j = \frac{1}{D} \sum_{i=1}^D R_i^j$ adalah uji Friedman membandingkan rata-rata peringkat setiap metode.
- X_F^2 adalah distribusi setiap metode

Jika hipotesis nol ditolak, dilanjutkan dengan uji *Nemenyi post hoc*. Uji *Nemenyi* digunakan saat semua metode pengklasteran dibandingkan satu sama lain. Kinerja dua metode pengklasteran signifikan berbeda jika rata-rata peringkat yang sesuai berbeda setidaknya pada perbedaan kritis, di hitung dengan persamaan (3.15), di mana nilai kritis q_α berdasarkan *range statistic*.

$$CD = q_\alpha \sqrt{\frac{C(C+1)}{D}} \quad (3.15)$$

BAB 4

HASIL DAN PEMBAHASAN

4.1 Hasil

Pada tahap ini akan dilaporkan hasil penelitian dari metode yang telah diusulkan. Metode yang diusulkan adalah penggabungan metode *u-control chart* (UCC) dan *automatic clustering differential evolution* (ACDE) untuk penentuan jumlah klaster pada metode k-Means. Selanjutnya fungsi objektif menggunakan *davies bouldin index* (DBI) dan *cosine similarity measure* (CS) digunakan untuk pencarian jumlah klaster secara evolusioner dan mengevaluasi kinerja metode. Nilai objektif DBI dan CS terendah adalah metode yang menunjukkan metode kinerja terbaik [97][98]. Enam dataset digunakan untuk menguji metode yang diusulkan dengan metode pembanding (GCUK, DCPSO dan ACDE) yang bertujuan untuk mengamati perilaku metode sehubungan dengan dataset dengan karakteristik yang berbeda-beda.

Pada Tabel 4.1 menunjukkan dataset yang digunakan di mana menampilkan jumlah *record*, jumlah kelas, dan jumlah atribut pada dataset [101]. Jumlah klaster yang optimal didasarkan pada jumlah kelas pada setiap dataset, apabila jumlah klaster yang ditemukan sama dengan jumlah kelas data, maka dapat disimpulkan jumlah klaster tersebut optimal.

Tabel 4.1 : Rangkuman Dataset

Dataset	Jumlah <i>Record</i>	Atribut	Kelas
<i>Synthetic</i>	300	2	3
<i>Iris</i>	150	4	3
<i>Glass</i>	214	9	6
<i>Wine</i>	178	13	3
<i>Vowel</i>	990	13	11
<i>Ruspini</i>	75	2	4

4.1.1 Hasil Eksperimen Metode MacQueen [23]

Pada Tabel 4.2 dapat dilihat hasil rangkuman pengukuran metode MacQueen untuk semua dataset.

Tabel 4.2 : Hasil Evaluasi DBI dari Metode MacQueen untuk Semua Dataset

Dataset	Kelas (k)	Metode MacQueen
<i>Synthetic</i>	3	0.532
<i>Iris</i>	3	0.666
<i>Glass</i>	6	0.945
<i>Wine</i>	3	0.551
<i>Vowel</i>	11	1.106
<i>Ruspini</i>	4	0.357

4.1.2 Hasil Eksperimen Metode Bandiopadhyay dan Maulik [44]

Pada Tabel 4.3 dan Tabel 4.4 dapat dilihat hasil evaluasi fungsi objektif DBI dan CS dari metode *genetic clustering with an unknown k* (GCUK) [44] untuk semua dataset. Seperti yang ditunjukkan pada Tabel 4.3 metode GCUK dapat menemukan jumlah kluster yang sesuai dengan jumlah kelas data dan memiliki nilai DBI paling rendah namun hanya pada satu dataset yaitu *wine*. Sedangkan pada Tabel 4.4 metode GCUK dapat menemukan jumlah kluster yang sesuai dengan jumlah kelas data dan memiliki nilai CS paling rendah namun hanya pada tiga dataset di antaranya *iris*, *wine* dan *ruspini*.

Tabel 4.3 : Hasil Evaluasi Fungsi Objektif DBI dari Metode GCUK untuk Semua Dataset

Dataset	Jumlah Kelas	Jumlah Kelas Optimal	DBI
<i>Synthetic</i>	3	2	0.5289
<i>Iris</i>	3	4	0.0382
<i>Glass</i>	6	4	1.5239
<i>Wine</i>	3	3	0.5521
<i>Vowel</i>	11	4	0.1013
<i>Ruspini</i>	4	6	0.5289

Tabel 4.4 : Hasil Evaluasi Fungsi Objektif CS dari Metode GCUK untuk Semua Dataset

Dataset	Jumlah Kelas	Jumlah Kelas Optimal	CS
<i>Synthetic</i>	3	8	0.7338
<i>Iris</i>	3	3	0.0449
<i>Glass</i>	6	3	0.0650
<i>Wine</i>	3	3	1.3933
<i>Vowel</i>	11	2	0.7818
<i>Ruspini</i>	4	4	2.7881

4.1.3 Hasil Experimen Metode Omran et al [46]

Pada Tabel 4.5 dan Tabel 4.6 dapat dilihat hasil evaluasi fungsi objektif DBI dan CS dari metode *dynamic clustering with PSO* (DCPSO) [46] untuk semua dataset. Seperti yang ditunjukkan pada Tabel 4.5 metode DCPSO dapat menemukan jumlah klaster yang sesuai dengan jumlah kelas data dan memiliki nilai DBI paling rendah namun hanya pada satu dataset yaitu *glass*. Sedangkan, pada Tabel 4.6 metode DCPSO dapat menemukan jumlah klaster yang sesuai dengan jumlah kelas data dan memiliki nilai CS paling rendah namun hanya pada dua dataset di antaranya *iris* dan *glass*.

Tabel 4.5 : Hasil Evaluasi Fungsi Objektif DBI dari Metode DCPSO untuk Semua Dataset

Dataset	Jumlah Kelas	Jumlah Kelas Optimal	DBI
<i>Synthetic</i>	3	2	0.6305
<i>Iris</i>	3	4	0.0486
<i>Glass</i>	6	6	0.5949
<i>Wine</i>	3	7	0.483
<i>Vowel</i>	11	8	0.9561
<i>Ruspini</i>	4	3	0.1762

Tabel 4.6 : Hasil Evaluasi Fungsi Objektif CS dari Metode DCPSO untuk Semua Dataset

Dataset	Jumlah Kelas	Jumlah Kelas Optimal	CS
<i>Synthetic</i>	3	4	0.6305
<i>Iris</i>	3	3	0.0486
<i>Glass</i>	6	6	0.5949

Dataset	Jumlah Kelas	Jumlah Kelas Optimal	CS
<i>Wine</i>	3	9	0.483
<i>Vowel</i>	11	6	0.9561
<i>Ruspini</i>	4	3	0.1762

4.1.4 Hasil Eksperimen Metode Das et al [42]

Pada Tabel 4.7 dan Tabel 4.8 dapat dilihat hasil evaluasi fungsi objektif DBI dan CS dari metode *automatic clustering differential evolution* (ACDE) [42] untuk semua dataset. Seperti yang ditunjukkan pada Tabel 4.7 metode ACDE dapat menemukan jumlah klaster yang sesuai dengan jumlah kelas data dan memiliki nilai DBI paling rendah pada empat dataset yaitu *synthetic*, *iris*, *wine* dan *ruspini*. Sedangkan, pada Tabel 4.8 metode GCUK dapat menemukan jumlah klaster yang sesuai dengan jumlah kelas data dan memiliki nilai CS paling rendah hampir semua dataset yaitu *synthetic*, *iris*, *glass*, *wine* dan *ruspini*.

Tabel 4.7 : Hasil Evaluasi Fungsi Objektif DBI dari Metode ACDE untuk Semua Dataset

Dataset	Jumlah Kelas	Jumlah Kelas Optimal	DBI
<i>Synthetic</i>	3	3	0.5642
<i>Iris</i>	3	3	0.039
<i>Glass</i>	6	3	0.6568
<i>Wine</i>	3	3	0.3956
<i>Vowel</i>	11	5	1.2728
<i>Ruspini</i>	4	4	0.1042

Tabel 4.8 : Hasil Evaluasi Fungsi Objektif CS dari Metode ACDE untuk Semua Dataset

Dataset	Jumlah Kelas	Jumlah Kelas Optimal	CS
<i>Synthetic</i>	3	3	0.6695
<i>Iris</i>	3	3	0.0425
<i>Glass</i>	6	6	0.0480
<i>Wine</i>	3	3	0.0071
<i>Vowel</i>	11	2	0.7818
<i>Ruspini</i>	4	4	2.5039

4.1.5 Hasil Eksperimen Metode yang Diusulkan

Pada Tabel 4.9 dan Tabel 4.10 dapat dilihat hasil evaluasi fungsi objektif DBI dan CS dari metode yang diusulkan dalam penelitian ini untuk semua dataset. Seperti yang ditunjukkan pada Tabel 4.9 metode yang diusulkan dapat menemukan jumlah klaster yang sesuai dengan jumlah kelas data dan memiliki nilai DBI paling rendah hampir pada semua dataset yaitu *synthetic*, *iris*, *glass*, *wine* dan *ruspini*. Sedangkan, pada Tabel 4.10 metode yang diusulkan dapat menemukan jumlah klaster yang sesuai dengan jumlah kelas data dan memiliki nilai CS paling rendah hampir pada semua dataset yaitu *synthetic*, *iris*, *glass*, *wine* dan *ruspini*.

Tabel 4.9 : Hasil Evaluasi Fungsi Objektif DBI dari Metode Usulan untuk semua Dataset

Dataset	Jumlah Kelas	Jumlah Kelas Optimal	DBI
<i>Synthetic</i>	3	3	0.4830
<i>Iris</i>	3	3	0.5090
<i>Glass</i>	6	3	0.4989
<i>Wine</i>	3	3	0.2083
<i>Vowel</i>	11	4	1.0324
<i>Ruspini</i>	4	4	0.0913

Tabel 4.10 : Hasil Evaluasi Fungsi Objektif CS dari Metode Usulan untuk Semua Dataset

Dataset	Jumlah Kelas	Jumlah Kelas Optimal	CS
<i>Synthetic</i>	3	3	0.5495
<i>Iris</i>	3	3	0.3043
<i>Glass</i>	6	6	0.0358
<i>Wine</i>	3	3	0.2677
<i>Vowel</i>	11	3	0.9209
<i>Ruspini</i>	4	4	1.3862

4.1.6 Hasil Eksperimen Perbandingan Kinerja Metode

Perhitungan metode dilakukan dengan menguji enam dataset publik antara lain: *synthetic*, *iris*, *glass*, *wine*, *vowel* dan *ruspini*. Enam dataset tersebut akan diuji dengan menggunakan beberapa metode, di antaranya:

1. Metode Usulan yang diusulkan dalam penelitian ini;

2. Metode k-Means yang telah diusulkan oleh MacQueen [23];
3. Metode DCPSO yang telah diusulkan oleh Omran at al. [46];
4. Metode GCUK yang telah diusulkan oleh Bandyopadhyay dan Maulik [44]
5. Metode ACDE yang telah diusulkan oleh Das et al. [42].

Hasil perhitungan evaluasi dari fungsi objektif DBI dan CS untuk penentuan jumlah kluster pada k-Means dirangkum pada sub bab 4.1.7. Tujuannya adalah untuk melihat secara detail pengaruh dari karakteristik dataset yang digunakan terhadap metode yang diusulkan dan metode pembandingan lainnya.

4.1.7 Hasil Pengukuran Kinerja Metode dari Fungsi Objektif DBI dan CS

Pada Tabel 4.11 dapat dilihat rata-rata *rank* terbaik, metode yang diusulkan menduduki posisi *rank* pertama, selanjutnya disusul oleh metode ACDE, GCUK dan DCPSO. Nilai rata-rata *rank* terkecil adalah metode terbaik.

Pada Tabel 4.12, menunjukkan hasil rangkuman eksperimen yang telah dilakukan berdasarkan fungsi objektif DBI dan CS pada setiap metode untuk masing-masing dataset. Rangkuman metode tersebut bertujuan mengidentifikasi metode terbaik dengan melihat nilai rata-rata fungsi objektif DBI dan CS terendah adalah metode yang baik.

Tabel 4.11 : Peringkat Evaluasi Metode Berdasarkan Fungsi Objektif DBI dan CS pada Setiap Metode untuk Masing-masing Dataset

Dataset	DBI				CS			
	<i>DCPSO</i>	<i>GCUK</i>	<i>ACDE</i>	<i>Metode Usulan</i>	<i>DCPSO</i>	<i>GCUK</i>	<i>ACDE</i>	<i>Metode Usulan</i>
<i>Synthetic</i>	4	2	3	1	4	3	2	1
<i>Iris</i>	3	1	2	4	3	2	1	4
<i>Glass</i>	2	5	3	1	4	3	2	1
<i>Wine</i>	3	4	2	1	3	4	1	2
<i>Vowel</i>	1	3	4	2	1	3	4	2
<i>Ruspini</i>	4	2	3	1	4	3	2	1
<i>Average Rank</i>	2.83	2.83	2.83	1.67	3.17	3	2	1.83

Tabel 4.12 : Rangkuman Hasil Evaluasi Metode Berdasarkan Fungsi Objektif DBI dan CS
untuk Setiap Dataset

Dataset	Metode	Fungsi Objektif			
		DBI		CS	
		<i>k</i>	Objektif	<i>k</i>	Objektif
<i>Synthetic</i>	<i>GCUK</i>	2	0.5289	8	0.7338
	<i>DCPSO</i>	2	0.6305	4	0.8847
	<i>ACDE</i>	3	0.5642	3	0.6695
	<i>Metode yang diusulkan</i>	3	0.483	3	0.5495
<i>Iris</i>	<i>GCUK</i>	4	0.0382	3	0.0447
	<i>DCPSO</i>	6	0.0486	3	0.0653
	<i>ACDE</i>	3	0.039	3	0.0425
	<i>Metode yang diusulkan</i>	3	0.509	3	0.3043
<i>Glass</i>	<i>GCUK</i>	4	1.5239	3	0.0650
	<i>DCPSO</i>	6	0.5949	6	0.3456
	<i>ACDE</i>	3	0.6568	6	0.0480
	<i>Metode yang diusulkan</i>	6	0.49888	6	0.0358
<i>Wine</i>	<i>GCUK</i>	3	0.5521	3	1.3933
	<i>DCPSO</i>	7	0.483	9	0.5863
	<i>ACDE</i>	3	0.3956	3	0.0071
	<i>Metode yang diusulkan</i>	3	0.2083	3	0.2677
<i>Vowel</i>	<i>GCUK</i>	13	1.0638	13	1.1739
	<i>DCPSO</i>	8	0.9561	6	0.9218
	<i>ACDE</i>	5	1.2728	2	0.7818
	<i>Metode yang diusulkan</i>	4	1.0324	3	0.9209
<i>Ruspini</i>	<i>GCUK</i>	4	0.1013	4	2.7881
	<i>DCPSO</i>	3	0.1762	3	2.8328
	<i>ACDE</i>	4	0.1042	4	2.5039
	<i>Metode yang diusulkan</i>	4	0.0913	4	1.3862

4.2 Pembahasan

4.2.1 Perbandingan Kinerja Metode Penentuan Jumlah Klaster pada Metode k-Means

Pada bagian ini akan dibahas perbandingan kinerja metode antara metode yang diusulkan dan metode pembanding yang digunakan dalam penelitian ini. Perbandingan kinerja metode berdasarkan fungsi objektif menggunakan *davies bouldin index* (DBI) dan *cosine similarity measure* (CS). Metode yang diusulkan akan dibandingkan dengan metode MacQueen [23], metode Omran at al [46], metode Bandyopadhyay dan Maulik [44] dan metode Das et al [42]. Gambar 4.1 dan Gambar 4.2 menunjukkan perbandingan nilai objektif DBI dan CS pada setiap metode untuk masing-masing dataset.

Pada Gambar 4.1 dapat dilihat diagram perbandingan kinerja pada semua metode berdasarkan nilai objektif DBI untuk masing-masing dataset. Diagram perbandingan kinerja tersebut menunjukkan bahwa kinerja metode yang diusulkan lebih unggul dibandingkan metode pembanding lainnya di hampir semua dataset, di mana metode yang diusulkan memiliki nilai objektif DBI paling rendah dari metode pembanding lainnya. Metode ACDE menempati posisi unggul kedua hanya di beberapa dataset seperti *iris*, *wine* dan *ruspini*. Selanjutnya metode unggul ketiga adalah metode DCPSO, unggul kedua GCUK dan unggul yang terakhir adalah k-Means.

Gambar 4.1 : Diagram Perbandingan Kinerja Metode Berdasarkan Fungsi Objektif DBI pada Setiap Metode untuk Masing-masing Dataset

Berbeda pada diagram perbandingan kinerja metode berdasarkan fungsi objektif CS yang disajikan pada Gambar 4.2 di mana menunjukkan metode ACDE menempati posisi unggul terbaik pertama dengan nilai CS terendah, posisi unggul kedua diperoleh metode yang diusulkan, selanjutnya posisi terbaik ketiga, keempat dan kelima adalah DCPSO, GCUK dan k-Means.

Gambar 4.2 : Diagram Perbandingan Kinerja Metode Berdasarkan Fungsi Objektif CS

Setelah mengevaluasi kinerja masing-masing metode untuk setiap dataset, tahap selanjutnya adalah dengan menguji perbedaan antara metode dengan menggunakan uji statistik non parametrik. Uji statistik non-parameterik telah banyak direkomendasikan untuk digunakan di bidang penelitian evolusioner dalam pengklasteran [8][73]. Berdasarkan rekomendasi ini, dalam kerangka penelitian ini, pengujian Friedman digunakan untuk membandingkan nilai DBI dan CS dari semua metode. Uji Friedman didasarkan pada kinerja *mean of rank (R)* metode pengklasteran pada masing-masing dataset.

4.2.1.1 Perbandingan Kinerja Metode Berdasarkan Fungsi Objektif DBI pada Uji Friedman

Uji Friedman digunakan dalam penelitian ini untuk melakukan uji beda metode di mana berguna untuk mengetahui perbedaan yang signifikan antara metode yang diusulkan dan metode pembanding lainnya. Tabel 4.13 menunjukkan hasil uji Friedman, diperoleh *P-value* sebesar 0,107, nilai ini lebih besar dari tingkat

signifikansi $\alpha = 0,05$, sehingga dapat disimpulkan bahwa tidak ada perbedaan signifikan antar metode.

Tabel 4.13 : Uji Friedman dari Fungsi Objektif DBI

Q (<i>Observed value</i>)	7.600
Q (<i>Critical value</i>)	9.488
DF	4
<i>p-value (Two-tailed)</i>	0.107
<i>alpha</i>	0.05

Untuk mengetahui peringkat terbaik antar metode digunakan *mean of rank (R)* berdasarkan *Two- tailed test* atau prosedur *Nemenyi post hoc*. Pada Tabel 4.14 ada dua kolom evaluasi, kolom *M* dan *R*. Kolom *M* adalah nilai rata-rata objektif DBI dari masing-masing metode untuk semua dataset sedangkan kolom *R* menunjukkan *mean of rank* berdasarkan prosedur *Nemenyi post hoc* dari setiap metode untuk semua dataset. *Mean of rank (R)* diperoleh dari perbandingan antar metode, peringkat terbaik adalah dengan nilai *mean of rank* terkecil [102] di mana nilai objektif DBI terendah adalah metode terbaik [97].

Tabel 4.14 : Perbandingan Kinerja Metode Berdasarkan Fungsi Objektif DBI dan Uji Friedman pada Semua Metode untuk Semua Dataset

	<i>Sintetik</i>	<i>Iris</i>	<i>Glass</i>	<i>Wine</i>	<i>Vowel</i>	<i>Ruspini</i>	<i>M</i>	<i>R</i>
<i>k-Means</i>	0.532	0.666	0.945	0.551	1.106	0.357	0.693	4.167
<i>GCUK</i>	0.529	0.038	1.524	0.552	1.064	0.101	0.635	3.000
<i>DCPSO</i>	0.631	0.049	0.595	0.483	0.956	0.176	0.482	3.000
<i>ACDE</i>	0.564	0.039	0.657	0.396	1.273	0.104	0.505	3.167
<i>Metode Usulan</i>	0.483	0.509	0.499	0.208	1.032	0.091	0.470	1.667

Gambar 4.3 : Perbandingan Kinerja Metode Berdasarkan Fungsi Objektif DBI pada Setiap Metode untuk Semua Dataset

Metode penentuan jumlah kluster terbaik untuk setiap dataset disorot dengan cetak tebal. Seperti yang ditunjukkan pada Tabel 4.14 bahwa metode yang diusulkan memiliki nilai rata-rata objektif DBI paling rendah yaitu $M = 0.470$ dari semua dataset untuk setiap metode dan uji Friedman dalam hal ini pada nilai *mean of rank* (R) menduduki peringkat metode terbaik pertama sebesar $R = 1.667$, metode DCPSO terbaik kedua, GCUK terbaik ketiga, ACDE dan k-Means terbaik kelima. Hasilnya adalah tidak ada perbedaan yang signifikan ketika metode yang diusulkan dibandingkan dengan metode *state-of-the-art* seperti k-Means, DCPSO, GCUK dan ACDE karena $P\text{-value} > 0.05$. Diagram perbandingan kinerja fungsi objektif DBI (M) dari semua metode untuk semua dataset ditunjukkan pada Gambar 4.3.

4.2.1.2 Perbandingan Kinerja Metode Berdasarkan Fungsi Objektif CS pada Uji Friedman dan *Nemenyi post hoc*

Uji Friedman digunakan dalam penelitian ini untuk melakukan uji beda metode di mana berguna untuk mengetahui perbedaan yang signifikan antara metode yang diusulkan dan metode pembanding lainnya. Tabel 4.15 menunjukkan hasil uji Friedman, diperoleh $P\text{-value}$ sebesar 0.001, nilai ini lebih kecil dari tingkat signifikansi $\alpha = 0.05$, sehingga dapat disimpulkan bahwa ada perbedaan signifikan antar metode.

Tabel 4.15 : Hasil Uji Friedman dari Fungsi Objektif CS

<i>Q (Observed value)</i>	18.933
<i>Q (Critical value)</i>	9.488
DF	4
<i>p-value (Two-tailed)</i>	0.001
<i>alpha</i>	0.05

Untuk mengetahui peringkat terbaik antar metode digunakan *mean of rank* (*R*) berdasarkan *Two- tailed test* atau prosedur *Nemenyi post hoc*. Pada Tabel 4.16 ada dua kolom evaluasi, kolom *M* dan *R*. Kolom *M* pada adalah nilai rata-rata objektif CS dari masing-masing metode untuk semua dataset sedangkan kolom *R* menunjukkan *mean of rank* berdasarkan prosedur *Nemenyi post hoc* dari setiap metode untuk semua dataset. *Mean of rank* (*R*) diperoleh dari perbandingan antar metode, peringkat terbaik adalah dengan nilai *mean of rank* terkecil [102] di mana nilai objektif CS terendah adalah metode terbaik [97].

Metode penentuan jumlah kluster terbaik untuk setiap dataset disorot dengan cetak tebal. Seperti yang ditunjukkan pada Tabel 4.16 bahwa metode yang diusulkan memiliki nilai rata-rata objektif CS paling rendah yaitu *M* = **0.577** dari semua dataset untuk setiap metode. Namun, berbeda pada hasil uji Friedman dalam hal ini *mean of rank* (*R*), metode ACDE memiliki nilai *R* = **1.500** terbaik unggul dari metode metode pembandingan lainnya.

Tabel 4.16 : Perbandingan Evaluasi Metode Berdasarkan Fungsi Objektif CS dan Uji Friedman pada Setiap Metode untuk Masing-masing Dataset

	<i>Sintetik</i>	<i>Iris</i>	<i>Glass</i>	<i>Wine</i>	<i>Vowel</i>	<i>Ruspini</i>	<i>M</i>	<i>R</i>
<i>k-Means</i>	1.978	0.526	2.415	1.682	5.728	3.087	2.569	5.000
<i>GCUK</i>	0.734	0.045	0.065	1.393	1.174	2.788	1.033	3.167
<i>DCPSO</i>	0.885	0.065	0.346	0.586	0.922	2.833	0.939	3.500
<i>ACDE</i>	0.669	0.042	0.048	0.007	0.782	2.504	0.675	1.500
<i>Metode Usulan</i>	0.549	0.304	0.036	0.268	0.921	1.386	0.577	1.833

Karena ada perbedaan yang signifikan antara metode yang diusulkan dengan metode pembandingan lainnya, maka analisis akan dilanjutkan dengan menggunakan

pengujian perbandingan berpasangan atau uji *Nemenyi post hoc*. Perbandingan uji *Nemenyi post hoc* dilakukan untuk mengecek metode yang berbeda secara signifikan di mana uji ini menghitung semua perbandingan berpasangan antara metode pengklasteran, apabila nilai hasil perbandingan berpasangan lebih besar dari nilai *Critical Different* (CD) maka ada perbedaan antara dua metode yang signifikan. Tabel 4.17 menunjukkan metode yang diusulkan dan metode ACDE memperoleh nilai **3.167** dan **3.500** yang mana lebih besar dari **CD = 2.490**, oleh karena itu maka dapat disimpulkan ada perbedaan yang signifikan antara dua metode tersebut. Menghitung CD dapat menggunakan persamaan (4.1).

$$Critical\ Different = q_{\alpha, \infty, K} \sqrt{\frac{K(K+1)}{12D}} \quad (4.1)$$

di mana nilai *Critical Difference* (CD) pada *Nemenyi post hoc* dipengaruhi nilai *student size range statistic* $q_{\alpha} = 3,314$, jumlah metode ($K = 5$) dan jumlah dataset ($D = 6$).

$$Critical\ different = 3.314 \sqrt{\frac{5(5+1)}{12*6}}$$

$$Critical\ Different = \mathbf{2.4902}$$

Tabel 4.17 : Perbandingan berpasangan Uji *Nemenyi post hoc*

	<i>k-Means</i>	<i>DCPSO</i>	<i>GCUK</i>	<i>ACDE</i>	<i>Metode Usulan</i>
<i>k-Means</i>	0	1.500	1.833	3.500	3.167
<i>DCPSO</i>	-1.500	0	0.333	2.000	1.667
<i>GCUK</i>	-1.833	-0.333	0	1.667	1.333
<i>ACDE</i>	-3.500	-2.000	-1.667	0	-0.333
<i>Metode Usulan</i>	-3.167	-1.667	-1.333	0.333	0
Critical different: 2.4902					

Selanjutnya, untuk mengetahui metode mana yang memiliki perbedaan metode yang signifikan, maka dilanjutkan dengan membaca hasil *P-value* pada uji *Nemenyi post hoc*. Apabila *P-value* < 0.05 maka performa metode tersebut berbeda secara signifikan sebagai mana di tunjukkan pada Tabel 4.18. Perbedaan metode

yang signifikan yang dihasilkan oleh uji *Nemenyi post hoc* ditunjukkan pada Tabel 4.19.

Tabel 4.18 : Nilai *P-value* Uji *Nemenyi post hoc* dari CS

	<i>k-Means</i>	<i>DCPSO</i>	<i>GCUK</i>	<i>ACDE</i>	<i>Metode Usulan</i>
<i>k-Means</i>	1	0.470	0.262	0.005	0.001
<i>DCPSO</i>	0.470	1	0.996	0.183	0.359
<i>GCUK</i>	0.262	0.996	1	0.359	0.588
<i>ACDE</i>	0.005	0.183	0.359	1	0.996
<i>Metode Usulan</i>	0.001	0.359	0.588	0.996	1

Tabel 4.19 : Perbedaan Signifikan Hasil CS dari Uji *Nemenyi post hoc*

	<i>k-Means</i>	<i>DCPSO</i>	<i>GCUK</i>	<i>ACDE</i>	<i>Metode Usulan</i>
<i>k-Means</i>	No	No	No	Yes	Yes
<i>DCPSO</i>	No	No	No	No	No
<i>GCUK</i>	No	No	No	No	No
<i>ACDE</i>	Yes	No	No	No	No
<i>Metode Usulan</i>	Yes	No	No	No	No

Seperti yang ditunjukkan pada Tabel 4.18, hasil *P-value* $< 0,05$ disorot dengan cetak tebal, yang berarti ada perbedaan yang signifikan secara statistik antara metode pengklasteran yaitu antara metode ACDE dan k-Means standar terhadap metode yang diusulkan. Berdasarkan hasil tersebut dapat disimpulkan bahwa penggabungan metode *u-control chart* (UCC) dan metode *automatic clustering based on differential evolution* (ACDE) mampu meningkatkan kinerja pengklasteran pada metode k-Means dengan mendapatkan nilai *P-value* sebesar $0.001 < \alpha = 0.005$. Selain itu metode yang diusulkan juga lebih unggul dari metode pembandingan lainnya seperti GCUK dan DCPSO. Akan tetapi, metode GCUK dan DCPSO tidak dapat diklaim bahwa metode yang diusulkan memiliki performa lebih unggul karena performa kedua metode tersebut masih dapat ditingkatkan dengan cara mengatur parameter yang sesuai, hal ini telah dikonfirmasi oleh Das et al [42] bahwa kinerja metode GCUK dan DCPSO masih dapat ditingkatkan kinerjanya dengan mengatur parameter yang sesuai dan bijak. Diagram perbandingan kinerja metode berdasarkan

fungsi objektif CS dari semua metode untuk semua dataset ditunjukkan pada Gambar 4.4.

Gambar 4.4 : Perbandingan Kinerja Metode Berdasarkan Fungsi Objektif CS Pada Setiap Metode untuk Semua Dataset

4.2.2 Perbandingan Metode Pengklasteran untuk Hasil Pencarian Jumlah Klaster pada Dataset Berdasarkan Fungsi Objektif DBI dan CS Secara Otomatis

Setelah menguji metode yang diusulkan dengan metode pembanding lainnya yang sebelumnya dijelaskan pada sub bab 4.2.1.1 dan 4.2.1.2. Pada tahap ini menunjukkan hasil pencarian dari metode yang diusulkan dan metode pembanding lainnya secara otomatis menentukan jumlah kluster yang optimal pada setiap dataset. Hasil pencarian dari setiap metode berdasarkan fungsi objektif menggunakan DBI dan CS di mana fungsi tersebut digunakan sebagai dasar pencarian kluster yang terbimbing secara evolusioner dan sebagai evaluasi kinerja metode pengklasteran.

Pada Gambar 4.5 menunjukkan metode GCUK berhasil mendapatkan jumlah kluster yang optimal di dua dataset (*wine*, *ruspini*) dan tidak optimal di empat dataset (*synthetic*, *iris*, *glass* dan *vowel*). Metode DCPSO berhasil mendapatkan jumlah kluster yang optimal hanya satu dataset (*glass*) dan tidak optimal hampir pada semua dataset (*synthetic*, *iris*, *wine*, *vowel* dan *ruspini*). Metode ACDE berhasil mendapatkan jumlah kluster yang optimal pada empat dataset (*synthetic*, *iris*, *wine*, *ruspini*) dan tidak optimal di dua dataset (*glass* dan *vowel*). Metode yang diusulkan

berhasil mendapatkan jumlah kluster yang optimal hampir semua dataset (*synthetic*, *iris*, *glass*, *wine*, *ruspini*) dan tidak optimal hanya satu dataset (*vowel*). Berdasarkan hasil eksperimen dari semua metode dalam menentukan jumlah kluster berdasarkan fungsi objektif DBI pada masing-masing dataset disimpulkan bahwa metode yang diusulkan menunjukkan keunggulan dalam ketepatan menentukan jumlah kluster yang optimal hampir pada semua dataset apabila dibandingkan dengan metode pembanding lainnya.

Gambar 4.5 : Perbandingan Hasil Metode Pengklasteran yang Optimal dan Tidak Optimal dalam Penentuan Jumlah Kluster berdasarkan Fungsi Objektif DBI pada Setiap Metode untuk Masing-masing Dataset

Pada Gambar 4.6 menunjukkan metode GCUK berhasil mendapatkan jumlah kluster yang optimal pada tiga dataset (*iris*, *wine*, *ruspini*) dan tidak optimal pada tiga dataset (*synthetic*, *glass* dan *vowel*). Metode DCPSO juga berhasil mendapatkan jumlah kluster yang optimal pada tiga dataset (*iris*, *glass*, *ruspini*) dan tidak optimal pada tiga dataset (*synthetic*, *wine* dan *vowel*). Kemudian metode ACDE dan metode yang diusulkan sama-sama berhasil mendapatkan jumlah kluster yang optimal hampir semua dataset (*synthetic*, *iris*, *glass*, *wine*, *rispini*) dan tidak optimal hanya pada satu dataset (*vowel*). Berdasarkan hasil eksperimen dari semua metode dalam menentukan jumlah kluster berdasarkan fungsi objektif CS pada masing-masing dataset disimpulkan bahwa metode yang diusulkan berhasil menunjukkan keunggulan yang sama dengan metode ACDE dalam ketepatan menentukan jumlah

klaster yang optimal hampir semua dataset apabila dibandingkan dengan metode pembandingan lainnya.

Gambar 4.6 : Perbandingan Hasil Pengklasteran yang Optimal dan Tidak Optimal dalam Penentuan Jumlah Klaster berdasarkan Fungsi Objektif CS Pada Setiap Metode untuk Masing-masing Dataset

Metode DCPSO sebenarnya memiliki kemampuan dalam menentukan jumlah klaster yang optimal, ini terjadi karena metode PSO memiliki struktur operator yang sederhana sehingga untuk mengendalikan solusi optimal selalu berdasarkan informasi *personal best* (*Pbest*) dan *global best* (*Gbest*) akibatnya metode DCPSO cepat mengalami konvergen di mana menimbulkan efek pada mahalanya komputasi ketika diterapkan [103]. Hal ini juga telah ditegaskan oleh beberapa penelitian yang menunjukkan bahwa DCPSO dapat memiliki kinerja yang baik untuk penentuan jumlah klaster pada metode k-Means ketika memilih parameter yang sesuai [61][47]. Namun dalam penelitian ini DCPSO kurang baik apabila dibandingkan dengan metode yang diusulkan karena memerlukan penyesuaian parameter sehingga mempengaruhi kinerja pengklasteran.

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Metode k-Means adalah metode pengklasteran yang sangat populer di berbagai bidang seperti segmentasi citra [5], pengenalan pola [74], bioinformatika [3] dan sebagainya karena dapat memanipulasi dataset yang besar [10]. Akan tetapi, hasil pengklasteran dari metode k-Means sangat bergantung pada penentuan jumlah kluster sebagai masukan, jika penentuan jumlah k kluster jauh dari solusi akhir maka kemungkinan hasil pengklasteran tidak tepat. Sedangkan, untuk mendapatkan jumlah kluster sebagai input pada metode k-Means sulit dilakukan karena diperlukan informasi dari data secara apriori [25][26][27][28]. Kondisi ini dikenal sebagai solusi optimum lokal [25]. Dalam praktiknya, masalah optimum lokal dapat diselesaikan dengan menjalankan metode beberapa kali dengan jumlah k yang berbeda kemudian memilih hasil yang terbaik. Penentuan jumlah kluster sangat penting bagi metode k-Means [3][16][29][30][31].

Sejak metode *evolutionary computation* (EC) digunakan untuk menyelesaikan permasalahan pengklasteran *hard partition* [32]. Maka dalam penelitian ini metode *automatic clustering based on differential evolution* (ACDE) sebagai salah satu metode terpopuler di bidang EC digunakan untuk memperbaiki kelemahan k-Means untuk penentuan jumlah kluster. Metode ACDE telah dikembangkan pertama kali oleh Das et al [42], diusulkan untuk meningkatkan kinerja pengklasteran pada penentuan jumlah kluster yang optimal secara otomatis. Namun dalam prosesnya masih bergantung pada penentuan ambang batas aktivasi k akhirnya kurang efisien dalam mencari k kluster yang tepat di mana kira-kira akan ditemukan sehingga menurunkan kinerja pengklasteran otomatis pada ACDE.

Sejak metode *statistical process control* (SPC) terbukti efektif digunakan untuk memecahkan masalah pengendalian kontrol atribut pada permasalahan metode-metode EC seperti *genetic algorithm* (GA) yang mana dalam penelitian Ihsan Kaya [53], berhasil menggabungkan metode *u-control chart* (UCC) dengan GA (UCC+GA) bertujuan untuk tahap produksi piston dengan menggunakan nilai

diagram kontrol yang diperoleh untuk memperoleh ukuran piston yang sesuai dengan mesin. Maka dalam penelitian ini UCC diusulkan untuk memperbaiki kelemahan ACDE dalam penentuan ambang batas aktivasi k untuk penentuan jumlah klaster pada metode k-Means.

Dalam penelitian ini metode yang diusulkan digunakan untuk penentuan jumlah klaster pada metode k-Means. Pengujian dilakukan dengan menggunakan enam dataset (*synthetic*, *iris*, *glass*, *wine*, *vowel*, dan *ruspini*) dari repositori UCI. Metode yang diusulkan tersebut kemudian dibandingkan dengan penelitian terkait sebelumnya yaitu k-Means, GCUK, DCPSO, dan ACDE. Kelima metode tersebut diukur evaluasinya menggunakan DBI dan CS, nilai evaluasi terkecil dari kedua evaluasi tersebut yang mendekati nilai nol adalah metode terbaik [42]. Kemudian dilakukan uji beda pada kelima metode tersebut dengan uji Friedman.

Pada pengukuran fungsi objektif DBI, metode yang diusulkan memiliki kinerja terbaik dengan nilai rata-rata terendah yaitu $M = 0.470$ dari semua dataset untuk setiap metode. Kemudian hasil dari pengukuran fungsi objektif DBI digunakan untuk uji beda metode menggunakan uji Friedman, hasilnya, tidak ada perbedaan yang signifikan ketika metode yang diusulkan dibandingkan dengan metode GCUK, DCPSO, dan ACDE karena diperoleh $P\text{-value} > \alpha = 0.05$. Uji beda dilanjutkan dengan melihat *mean of rank* (R) sebagai pemeringkatan setiap metode berdasarkan prosedur *Nemenyi* yang mana diperoleh nilai $R = 1.667$ pada metode yang diusulkan berada pada peringkat ke-1, metode DCPSO peringkat ke-2, metode GCUK peringkat ke-3, metode ACDE peringkat ke-4, dan metode k-Means adalah peringkat ke-5.

Pada pengukuran fungsi objektif CS, metode yang diusulkan memiliki kinerja terbaik dengan nilai rata-rata terendah $M = 0.577$, disusul oleh metode ACDE, DCPSO, GCUK, dan k-Means dengan masing-masing nilai rata-rata 0.675, 0.939, 1.033, dan 2.569. Kemudian hasil dari pengukuran fungsi objektif CS digunakan untuk uji beda menggunakan uji Friedman, hasilnya, ada perbedaan signifikan di antara metode yang diuji karena diperoleh $P\text{-value} < \alpha = 0.05$. Uji beda dilanjutkan dengan melihat *mean of rank* (R) sebagai pemeringkatan beberapa perbandingan berpasangan berdasarkan prosedur *Nemenyi*. Diperoleh nilai terbaik dari metode ACDE sebesar $R = 1.500$ di mana metode ini berada pada peringkat ke-1, menyusul

metode yang diusulkan pada peringkat ke-2, metode GCUK pada peringkat ke-3, metode DCPSO pada peringkat ke-4, dan metode k-Means pada peringkat ke-5 dengan masing-masing nilai R adalah 1.833, 3.167, 3.500, dan 5.000. Selanjutnya pada hasil perbedaan berpasangan diketahui ada dua metode berpasangan yang berbeda secara signifikan apabila nilai perbedaan berpasangan lebih besar dari nilai *critical different* (CD) = **2.490**, adalah antara: 1) metode yang diusulkan dan metode k-Means, dan 2) metode ACDE dan metode k-Means, masing-masing nilainya adalah (-3.167 dan 3.167) dan (-3.500 dan 3.500).

Hasil penelitian menunjukkan bahwa metode yang diusulkan yaitu dengan penggabungan metode *u-control chart* (UCC) dan *automatic clustering based on differential evolution* (ACDE) yang memiliki masalah pada penentuan jumlah kluster pada k-Means dipilih secara acak. Kontribusi ini sebagai jawaban dari masalah dan tujuan pada penelitian ini.

5.2 Saran

Penelitian ini telah memberikan kontribusi untuk penentuan jumlah kluster yang optimal pada metode k-Means. Namun ada beberapa metode yang dapat diuji coba lagi pada penelitian selanjutnya agar dapat menghasilkan kinerja yang lebih baik adalah sebagai berikut:

1. Dalam penelitian ini metode UCC digunakan untuk penentuan ambang batas aktivasi k sebagai langkah awal pembangkitan populasi untuk mencari nilai variabel pada metode ACDE dalam menentukan jumlah kluster yang optimal pada k-Means. Penelitian selanjutnya mungkin dapat ditambahkan dengan metode peta kontrol lainnya dari *statistical proses control* (SPC) seperti *p-control chart* (PCC) dan *c-control chart* (CCC) [105]. Menurut, Ihsan Kaya [53] metode SPC dapat secara efektif mendeteksi perubahan data dalam proses yang dapat mempengaruhi kualitas hasil.
2. Penentuan jumlah kluster pada metode k-Means masih menggunakan cara manual sehingga kurang efektif dan efisien. Mengatasi masalah penentuan jumlah kluster dapat meningkatkan kualitas pengklasteran k-Means [26]. Penentuan jumlah kluster pada metode k-Means paling banyak dipecahkan dengan pendekatan metode *evolutioner computation* (EC) [32]. Beberapa teknik pencarian dengan pendekatan metode EC yang dapat digunakan adalah

termasuk metode *artificial bee colony* (ABC) yang telah diusulkan oleh Wang et al [106] dan *quantum binary* (QB) yang pernah diusulkan oleh Chung et al [107].

3. Pada penelitian ini terdapat dataset yang memiliki jumlah atribut yang banyak. Penelitian selanjutnya dapat dikaitkan dengan metode seleksi atribut. Menurut beberapa peneliti seperti Breaban et al [108] dan Tsai et al [109] metode seleksi atribut dapat meningkatkan kinerja metode pengklasteran dengan cara menghapus atribut yang tidak relevan karena tidak semua atribut informatif, sehingga metode seleksi atribut sangat penting digunakan pada penelitian selanjutnya.

DAFTAR PUSTAKA

- [1] C. C. Aggarwal and C. K. Reddy, *Data Clustering: Algorithms and Applications*. CRC Press Taylor & Francis Group, 2014.
- [2] H. He and Y. Tan, “A two-stage genetic algorithm for automatic clustering,” *Neurocomputing*, vol. 81, pp. 49–59, 2012.
- [3] W. Xiang, N. Zhu, S. Ma, X. Meng, and M. An, “A dynamic shuffled differential evolution algorithm for data clustering,” *Neurocomputing*, 2015.
- [4] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*. Elsevier Inc., 2012.
- [5] G. B. Coleman and H. C. Andrews, “Image-understanding systems,” *Proc. IEEE*, vol. 67, no. 5, 1979.
- [6] R. Suzuki and H. Shimodaira, “Pvclust: An R package for assessing the uncertainty in hierarchical clustering,” *Bioinformatics*, vol. 22, no. 12, pp. 1540–1542, 2006.
- [7] F. D. A. T. De Carvalho, Y. Lechevallier, and F. M. De Melo, “Partitioning hard clustering algorithms based on multiple dissimilarity matrices,” *Pattern Recognit.*, vol. 45, no. 1, pp. 447–464, 2012.
- [8] S. J. Nanda and G. Panda, “A survey on nature inspired metaheuristic algorithms for partitional clustering,” *Swarm Evol. Comput.*, vol. 16, pp. 1–18, 2014.
- [9] A. Fahad *et al.*, “A survey of clustering algorithms for big data: Taxonomy and empirical analysis,” *IEEE Trans. Emerg. Top. Comput.*, vol. 2, no. 3, pp. 267–279, 2014.
- [10] B. A. Pimentel and R. M. C. R. De Souza, “Neurocomputing Multivariate Fuzzy C-Means algorithms with weighting,” *Neurocomputing*, pp. 1–20, 2015.
- [11] R. Xu and D. Wunsch, “Survey of Clustering Algorithms,” *IEEE Trans.*

NEURAL Netw., vol. 16, no. 3, pp. 645–678, 2005.

- [12] A. K. Jain, M. . Murty, and P. J. Flynn, “Data Clustering : A Review,” *ACM Comput. Surv.*, vol. 31, no. 3, p. 60, 2000.
- [13] C. Ozturk, E. Hancer, and D. Karaboga, “Dynamic Clustering with Improved Binary Artificial Bee Colony Algorithm Author:,” *Appl. Soft Comput. J.*, 2015.
- [14] H. Frigui, R. Krishnapuram, and S. Member, “With Applications in Computer Vision,” vol. 21, no. 5, pp. 450–465, 1999.
- [15] S. Das, A. Abraham, and A. Konar, “Automatic kernel clustering with a Multi-Elitist Particle Swarm Optimization Algorithm,” *Pattern Recognit. Lett.*, vol. 29, pp. 688–699, 2008.
- [16] A. K. Jain, “Data clustering : 50 years beyond K-means,” *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, 2010.
- [17] C. C. Aggarwal and C. K. Reddy, *DATA CLUSTERING Algorithms and Applications*. CRC Press Taylor & Francis Group, 2014.
- [18] M. C. Naldi, R. J. G. B. Campello, E. R. Hruschka, and A. C. P. L. F. Carvalho, “Efficiency issues of evolutionary k -means,” *Appl. Soft Comput. J.*, vol. 11, no. 2, pp. 1938–1952, 2011.
- [19] J. Wu, *Advances in K-means Clustering: A Data Mining Thinking*. 2012.
- [20] S. P. Lloyd, “Least Squares Quantization in PCM,” *IEEE Trans. Inf. THEORY*, vol. I, no. 2, pp. 129–137, 1982.
- [21] E. W. Forgy, “Cluster analysis of multivariate data: efficiency versus interpretability of classifications,” *Biometrics*, vol. 21, no. 3, pp. 768–769, 1965.
- [22] H. P. Friedman and J. Rubin, “On Some Invariant Criteria for Grouping Data,” *J. Am. Stat. Assoc.*, vol. 62, no. 320, 1967.
- [23] J. MAcQueen, “Some methods for classification and analysis of multivariate observations,” *Proc. Fifth Berkeley Symp. Math. Stat. Probab*, vol. 1, no. 23,

pp. 281–297, 1967.

- [24] A. M. Bagirov, “Modified global k -means algorithm for minimum sum-of-squares clustering problems,” *Pattern Recognit.*, vol. 41, pp. 3192–3199, 2008.
- [25] L. Bai, J. Liang, and C. Dang, “An initialization method to simultaneously find initial cluster centers and the number of clusters for clustering categorical data,” *Knowledge-Based Syst.*, vol. 24, no. 6, pp. 785–795, 2011.
- [26] M. A. Rahman and M. Z. Islam, “A hybrid clustering technique combining a novel genetic algorithm with K-Means,” *Knowledge-Based Syst.*, vol. 71, no. August, pp. 345–365, 2014.
- [27] S. Salvador and P. Chan, “Determining the Number of Clusters / Segments in Hierarchical Clustering / Segmentation Algorithms,” *Proc. - Int. Conf. Tools with Artif. Intell. ICTAI*, no. Ictai, p. 20, 2003.
- [28] Z. Yaqian, Q. H. Chai, and G. W. Boon, “Curvature-based method for determining the number of clusters,” *Inf. Sci. (Ny)*, 2017.
- [29] S. S. Khan and A. Ahmad, “Cluster center initialization algorithm for K -means clustering,” *Pattern Recognit. Lett.*, vol. 25, pp. 1293–1302, 2004.
- [30] F. Khan, “An initial seed selection algorithm for k-means clustering of georeferenced data to improve replicability of cluster assignments for mapping application,” *Appl. Soft Comput. J.*, vol. 12, no. 11, pp. 3698–3700, 2012.
- [31] A. Ahmad, “A k -mean clustering algorithm for mixed numeric and categorical data,” *Data Knowl. Eng.*, vol. 63, pp. 503–527, 2007.
- [32] E. Hancer and D. Karaboga, “A comprehensive survey of traditional, merge-split and evolutionary approaches proposed for determination of cluster number,” *Swarm Evol. Comput.*, vol. 32, pp. 49–67, 2016.
- [33] S. Salvador and P. Chan, “Learning States and Rules for Detecting Anomalies in Time Series,” *Appl. Intell.*, pp. 241–255, 2005.
- [34] Q. Zhao, “Cluster Validity in Clustering Methods,” University of Eastern

Finland, 2012.

- [35] A. Mukhopadhyay *et al.*, “Survey of Multiobjective Evolutionary Algorithms for Data Mining : Part II,” *IEEE Trans. Evol. Comput.*, vol. 18, no. 1, pp. 20–35, 2014.
- [36] D. Chaudhuri, B. . Chaudhuri, and C. . Murthy, “A new split-and-merge clustering technique,” *Pattern Recognit. Lett.*, vol. 13, no. June, pp. 399–409, 1992.
- [37] M. M. Hasan and M. Ameer, “Object Segmentation Using Block Based Patterns,” *IEEE Reg. 10 Annu. Int. Conf. Proceedings/TENCON*, pp. 1–6, 2009.
- [38] V. Manohar, S. N. Vitaladevuni, H. Cao, R. Prasad, and P. Natarajan, “Graph Clustering-based Ensemble Method for Handwritten Text Line Segmentation,” *Int. Conf. Doc. Anal. Recognit.*, pp. 574–578, 2011.
- [39] A. R. M. Faruquzzaman, N. Rabbani, J. Arafae, Z. Karimi, and M. A. Ali, “Object Segmentation Based on Split and Merge Algorithm,” *IEEE Reg. 10 Annu. Int. Conf. Proceedings/TENCON*, no. c, 2008.
- [40] S. Das, A. Abraham, and A. Konar, *Metaheuristic Clustering*. Springer Berlin Heidelberg, 2009.
- [41] E. R. Hruschka, R. J. G. B. Campello, A. A. Freitas, C. Ponce, and L. F. De Carvalho, “A Survey of Evolutionary Algorithms for Clustering,” *IEEE Trans. Syst. MAN, Cybern. C Appl. Rev.*, vol. 39, no. 2, pp. 133–155, 2009.
- [42] S. Das, A. Abraham, and A. Konar, “Automatic Clustering Using an Improved Differential Evolution Algorithm,” *IEEE Trans. Syst. Man, Cybern. - Part A Syst. Humans*, vol. 38, no. 1, pp. 218–237, 2008.
- [43] R. . Kuo, Y. . Huang, C. . Lin, Y.-H. Wu, and F. . Zulvia, “Automatic Kernel Clustering with Bee Colony Optimization Algorithm,” *Inf. Sci. (Ny)*, 2014.
- [44] S. Bandyopadhyay and U. Maulik, “Genetic clustering for automatic evolution of clusters and application to image classification,” *Pattern Recognit.*, vol. 35,

pp. 1197–1208, 2002.

- [45] A. J. Garcia and W. G. Flores, “Automatic Clustering Using Nature-Inspired Metaheuristics: A Survey,” *Appl. Soft Comput.*, 2015.
- [46] M. G. H. Omran, A. P. Engelbrecht, and A. Salman, “Dynamic clustering using particle swarm optimization with application in image segmentation,” *Pattern Anal. Appl.*, pp. 332–344, 2006.
- [47] R. J. Kuo and F. E. Zulvia, “Automatic Clustering Using an Improved Particle Swarm Optimization,” *J. Ind. Intell. Inf.*, vol. 1, no. 1, pp. 46–51, 2013.
- [48] G. Dobbie, Y. Sing, P. Riddle, and S. Ur, “Research on particle swarm optimization based clustering: A systematic review of literature and techniques,” *Swarm Evol. Comput.*, vol. 17, pp. 1–13, 2014.
- [49] G. Kumarl, “Satellite Image Clustering and Optimization using,” *IEEE Int. Conf. Power Electron. Intell. Control Energy Syst.*, pp. 1–4, 2016.
- [50] R. Storn and K. Price, “Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces,” *J. Glob. Optim.*, vol. Volume 11, no. Issue 4, pp. 341–359, 1997.
- [51] R. . Kuo, Suryani Erma, and A. Yasid, “Automatic Clustering Combining Differential Evolution Algorithm and k-Means Algorithm,” *Proc. Inst. Ind. Eng. Asian Conf. 2013*, pp. 1207–1215, 2013.
- [52] M. Weber, F. Neri, and V. Tirronen, “A study on scale factor in distributed differential evolution,” *Inf. Sci. (Ny)*, vol. 181, no. 12, pp. 2488–2511, 2011.
- [53] I. Kaya, “A genetic algorithm approach to determine the sample size for attribute control charts,” *Inf. Sci. (Ny)*, vol. 179, no. 10, pp. 1552–1566, 2009.
- [54] Shuttleworth, “What is a Literature Review?,” 2009. [Online]. Available: <https://explorable.com/what-is-a-literature-review>.
- [55] R. S. Wahono, “A Systematic Literature Review of Software Defect Prediction : Research Trends , Datasets , Methods and Frameworks,” *J. Softw. Eng.*, vol. 1, no. 1, pp. 1–16, 2015.

- [56] B. Kitchenham and S. Charters, *Guidelines for performing Systematic Literature Reviews in Software Engineering*. 2007.
- [57] Microsoft Visio, “Microsoft Office Visio,” *Microsoft*, 2016. [Online]. Available: <http://office.microsoft.com/visio>.
- [58] Xm. Ltd, “XMind,” 2008. [Online]. Available: <https://en.wikipedia.org/wiki/XMind>.
- [59] C. Catal and B. Diri, “A systematic review of software fault prediction studies,” *Expert Syst. Appl.*, vol. 36, no. 4, pp. 7346–7354, 2009.
- [60] M. Sarkar, B. Yegnanarayana, and D. Khemani, “A clustering algorithm using an evolutionary programming-based approach,” *Pattern Recognit. Lett.*, vol. 18, pp. 975–986, 1997.
- [61] R. J. Kuo, Y. J. Syu, Z. Chen, and F. C. Tien, “Integration of particle swarm optimization and genetic algorithm for dynamic clustering,” *Inf. Sci. (Ny)*, vol. 195, pp. 124–140, 2012.
- [62] J. Xiao, Y. Yan, J. Zhang, and Y. Tang, “A quantum-inspired genetic algorithm for k-means clustering,” *Expert Syst. Appl.*, vol. 37, no. 7, pp. 4966–4973, 2010.
- [63] David Aha, “UCI Machine Learning Repository,” *UCI archive*, 2017. .
- [64] R. H. Turi and B. Comp, “Clustering-based Color Image Segmentation,” Monash University, Australia, 2001.
- [65] E. R. Hruschka, R. J. G. B. Campello, and L. N. De Castro, “Evolving clusters in gene-expression data,” *Inf. Sci. (Ny)*, vol. 176, pp. 1898–1927, 2006.
- [66] J. Handl and J. Knowles, “An Evolutionary Approach to Multiobjective Clustering,” *IEEE Trans. Evol. Comput.*, vol. 11, no. 1, pp. 56–76, 2007.
- [67] S. Bandyopadhyay, S. Member, S. Saha, and S. Member, “A Point Symmetry-Based Clustering Technique for Automatic Evolution of Clusters,” *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 11, pp. 1441–1457, 2008.
- [68] S. Saha and S. Bandyopadhyay, “A new point symmetry based fuzzy genetic

- clustering technique for automatic evolution of clusters,” *Inf. Sci. (Ny)*, vol. 179, no. 19, pp. 3230–3246, 2009.
- [69] S. Das, A. Chowdhury, and A. Abraham, “A Bacterial Evolutionary Algorithm for Automatic Data Clustering,” *IEEE Congr. Evol. Comput. CEC 2009*, 2009.
 - [70] A. Paoli, F. Melgani, S. Member, E. Pasolli, and S. Member, “Clustering of Hyperspectral Images Based on Multiobjective Particle Swarm Optimization,” *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 12, pp. 4175–4188, 2009.
 - [71] W. Ma, L. Jiao, and M. Gong, “Immunodominance and clonal selection inspired multiobjective clustering,” *Prog. Nat. Sci.*, vol. 19, no. 6, pp. 751–758, 2009.
 - [72] J. Xiao, Y. Yan, J. Zhang, and Y. Tang, “A quantum-inspired genetic algorithm for k -means clustering,” *Expert Syst. Appl.*, vol. 37, no. 7, pp. 4966–4973, 2010.
 - [73] S. Saha and S. Bandyopadhyay, “A symmetry based multiobjective clustering technique for automatic evolution of clusters,” *Pattern Recognit.*, vol. 43, no. 3, pp. 738–751, 2010.
 - [74] Y. Liu, X. Wu, and Y. Shen, “Automatic clustering using genetic algorithms,” *Appl. Math. Comput.*, vol. 218, no. 4, pp. 1267–1279, 2011.
 - [75] D. Horta, I. C. De Andrade, and R. J. G. B. Campello, “Evolutionary fuzzy clustering of relational data,” *Theor. Comput. Sci.*, vol. 412, no. 42, pp. 5854–5870, 2011.
 - [76] J. Liang, X. Zhao, D. Li, F. Cao, and C. Dang, “Determining the number of clusters using information entropy for mixed data,” *Pattern Recognit.*, vol. 45, no. 6, pp. 2251–2265, 2012.
 - [77] Y. Fang and J. Wang, “Selection of the number of clusters via the bootstrap method,” *Comput. Stat. Data Anal.*, vol. 56, no. 3, pp. 468–477, 2012.
 - [78] T. Cura, “A particle swarm optimization approach to clustering,” *Expert Syst.*

Appl., vol. 39, no. 1, pp. 1582–1588, 2012.

- [79] S. J. Nanda, G. Student, M. Ieee, G. Panda, and S. M. Ieee, “Automatic Clustering using MOCLONAL for Classifying Actions of 3D Human Models,” *SHUSER 2012 - 2012 IEEE Symp. Humanit. Sci. Eng. Res.*, pp. 945–950, 2012.
- [80] S. Gou, X. Zhuang, Y. Li, C. Xu, and L. C. Jiao, “Neurocomputing Multi-elitist immune clonal quantum clustering algorithm,” *Neurocomputing*, vol. 101, pp. 275–289, 2013.
- [81] H. Xia, J. Zhuang, and D. Yu, “Novel soft subspace clustering with multi-objective evolutionary approach for high-dimensional data,” *Pattern Recognit.*, vol. 46, no. 9, pp. 2562–2575, 2013.
- [82] S. Jagannath and G. Panda, “Engineering Applications of Artificial Intelligence Automatic clustering algorithm based on multi-objective Immunized PSO to classify actions of 3D human models,” *Eng. Appl. Artif. Intell.*, vol. 26, no. 5–6, pp. 1429–1441, 2013.
- [83] S. Saha and S. Bandyopadhyay, “A generalized automatic clustering algorithm in a multiobjective framework,” *Appl. Soft Comput. J.*, vol. 13, no. 1, pp. 89–108, 2013.
- [84] D. Zhang, M. Ji, J. Yang, Y. Zhang, and F. Xie, “A novel cluster validity index for fuzzy clustering based on bipartite modularity,” *Fuzzy Sets Syst.*, vol. 253, no. c, pp. 122–137, 2014.
- [85] A. Kolesnikov, E. Trichina, and T. Kauranne, “Estimating the number of clusters in a numerical data set via quantization error modeling,” *Pattern Recognit.*, pp. 1–12, 2014.
- [86] E. Lughofer and M. Sayed-mouchaweh, “Autonomous data stream clustering implementing split-and-merge concepts – Towards a plug-and-play approach,” *Inf. Sci. (Ny)*, vol. 304, pp. 54–79, 2015.
- [87] Y. Saleem, K. A. Yau, H. Mohamad, N. Ramli, and M. Husain, “SMART : A SpectruM-Aware clusteR-based rouTing scheme for distributed cognitive

radio networks,” *Comput. Networks*, 2015.

- [88] C. Ozturk, E. Hancer, and D. Karaboga, “A Novel Binary Artificial Bee Colony Algorithm Based On Genetic Operators,” *Inf. Sci. (Ny)*, no. October, 2015.
- [89] R. Liu, B. Zhu, R. Bian, Y. Ma, and L. Jiao, “Dynamic local search based immune automatic clustering algorithm and its applications,” *Appl. Soft Comput. J.*, vol. 27, pp. 250–268, 2015.
- [90] H. Peng, J. Wang, P. Shi, J. Wang, and P. Shi, “An automatic clustering algorithm inspired by membrane computing,” *Pattern Recognit. Lett.*, 2015.
- [91] A. Mur, R. Dormido, N. Duro, S. Dormido-canto, and J. Vega, “Determination of the optimal number of clusters using a spectral clustering optimization,” *Expert Syst. Appl.*, vol. 65, pp. 304–314, 2016.
- [92] S. S. Kar and S. P. Maity, “Differential Evolution based Optimal Clustering for Retinal Blood Vessel Extraction,” *Int. Conf. Informatics, Electron. Vis.*, pp. 675–680, 2016.
- [93] R. J. Kuo and F. . Zulvia, “Cluster Analysis using A Gradient Evolution-based K -means Algorithm,” *IEEE Congr. Evol. Comput.*, pp. 5138–5145, 2016.
- [94] F. Lezama and A. Y. Rodr, “Load Pattern Clustering Using Differential Evolution with Pareto Tournament,” *Int. Conf. Informatics, Electron. Vis.*, pp. 241–248, 2016.
- [95] R. Das and S. Saha, “Gene Expression Data Classification using Automatic Differential Evolution Based Algorithm,” *2016 IEEE Congr. Evol. Comput.*, pp. 3124–3130, 2016.
- [96] J. Demsar, “Statistical Comparisons of Classifiers over Multiple Data Sets,” *J. of Machine Learn. Res.*, vol. 7, pp. 1–30, 2006.
- [97] D. L. Davies and D. W. Bouldin, “A Cluster Separation Measure,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 2, pp. 224–227, 1979.
- [98] C. H. Chou, M. C. Su, and E. Lai, “A new cluster validity measure and its

- application to image compression,” *Pattern Anal. Appl.*, vol. 7, pp. 205–220, 2004.
- [99] M. Ramadas, A. Abraham, and S. Kumar, “FSDE-Forced Strategy Differential Evolution used for data clustering,” *J. King Saud Univ. - Comput. Inf. Sci.*, 2016.
- [100] S. Bandyopadhyay and U. Maulik, “Nonparametric Genetic Clustering: Comparison of Validity Indices,” *IEEE Trans. Syst. MAN, Cybern. C Appl. Rev.*, vol. 31, no. 1, pp. 62–66, 2001.
- [101] D. Aha *et al.*, “UCI Repository of Machine Learning Database,” 1987.
- [102] I. Brown and C. Mues, “An experimental comparison of classification algorithms for imbalanced credit scoring data sets,” *Expert Syst. Appl.*, vol. 39, no. 3, pp. 3446–3453, 2012.
- [103] Q. Cui *et al.*, “Globally-optimal prediction-based adaptive mutation particle swarm optimization,” *Inf. Sci. (Ny)*, vol. 418–419, pp. 186–217, 2017.
- [104] K. Fukunaga, *Statistical Pattern Statistical Pattern Recognition*. Academic Press, 1990.
- [105] D. C. Montgomery, *Introduction to Statistical Quality Control*, Seventh Ed. Wiley, 2015.
- [106] Z. Su, P. Wang, J. Shen, Y. Li, Y. Zhang, and E. Hu, “Automatic fuzzy partitioning approach using Variable string length Artificial Bee Colony (VABC) algorithm &,” *Appl. Soft Comput. J.*, vol. 12, no. 11, pp. 3421–3441, 2012.
- [107] C. Y. Chung, S. Member, H. Yu, and K. P. Wong, “An Advanced Quantum-Inspired Evolutionary Algorithm for Unit Commitment,” *IEEE Trans. POWER Syst.*, vol. 26, no. 2, pp. 847–854, 2011.
- [108] M. Breaban and H. Luchian, “A unifying criterion for unsupervised clustering and feature selection,” *Pattern Recognit.*, vol. 44, no. 4, pp. 854–865, 2011.
- [109] C. Tsai, W. Eberle, and C. Chu, “Knowledge-Based Systems Genetic

algorithms in feature and instance selection,” *Knowledge-Based Syst.*, vol. 39, pp. 240–247, 2013.

LAMPIRAN

Lampiran 1 : Alur Eksperimen Pseudo Code Metode k-Means [23]

Step	Deskripsi
1	load iris
2	data = X(:,1:4);
3	% Tentukan jumlah klaster
4	k = 3;
5	% Bangkitkan Pusat Klaster secara acak
6	[m,C,sumd,D] = kmeans(data,k);
7	display('CENTROID');
8	C
9	% Tampilkan jarak tiap klaster pada pusat klaster terdekat
10	display('IDX JARAK KE C1 JARAK KE C2 JARAK KE C3');
11	[m D.^0.5]
12	display('EVALUASI KLASSTER DENGAN DBI');
13	% Evaluasi klaster
14	eva = evalclusters(data, C, 'DaviesBouldin');

Lampiran 2 : Alur Eksperimen Pseudo Code Metode GCUK

Metode: GCUK	
Step	Deskripsi
1	%Menyiapkan dataset
2	data = load('iris');
3	X = data.X/100;
4	X(:,5)=[];
5	%Tentukan jumlah k
6	k = 10;
7	CostFunction=@(s) ClusteringCost(s, X); % Cost Function
8	VarSize=[k size(X,2)+1]; % Decision Variables Matrix Size
9	nVar=prod(VarSize); % Number of Decision Variables
10	VarMin= repmat([min(X) 0],k,1);
11	VarMax= repmat([max(X) 1],k,1);
12	%% GCUK Parameters
13	MaxIt=200; % Maximum Number of Iterations
14	nPop=50; % Population Size
15	pc=0.8; % Crossover Percentage
16	nc=2*round(pc*nPop/2); % Number of Offsprings (Parnets)
17	pm=0.3; % Mutation Percentage
18	nm=round(pm*nPop); % Number of Mutants
19	gamma=0.2;
20	mu=0.02; % Mutation Rate
21	beta=8; % Selection Pressure
22	%% Initialization
23	empty_individual.Position=[];
24	empty_individual.Cost=[];
25	empty_individual.Out=[];
26	pop=repmat(empty_individual,nPop,1);
27	for i=1:nPop
28	% Initialize Position
29	pop(i).Position=unifrnd(VarMin,VarMax,VarSize);
30	% Evaluation
31	[pop(i).Cost,
32	pop(i).Out]=CostFunction(pop(i).Position);
33	end

Metode: GCUK

Step	Deskripsi
34	% Sort Population
35	Costs=[pop.Cost];
36	[Costs, SortOrder]=sort(Costs);
37	pop=pop(SortOrder);
38	% Store Best Solution
39	BestSol=pop(1);
40	% Array to Hold Best Cost Values
41	BestCost=zeros(MaxIt,1);
42	% Store Cost
43	WorstCost=pop(end).Cost;
44	%% Main Loop
45	for it=1:MaxIt
46	P=exp(-beta*Costs/WorstCost);
47	P=P/sum(P);
48	% Crossover
49	popc= repmat(empty_individual,nc/2,2);
50	for k=1:nc/2
51	% Select Parents Indices
52	i1=RouletteWheelSelection(P);
53	i2=RouletteWheelSelection(P);
54	% Select Parents
55	p1=pop(i1);
56	p2=pop(i2);
57	% Apply Crossover
58	[popc(k,1).Position, popc(k,2).Position]=...
59	Crossover(p1.Position,p2.Position,gamma,VarMin,VarMax);
60	% Evaluate Offsprings
61	[popc(k,1).Cost,
62	popc(k,1).Out]=CostFunction(popc(k,1).Position);
63	[popc(k,2).Cost,
64	popc(k,2).Out]=CostFunction(popc(k,2).Position);
65	end
66	popc=popc(:);
67	% Mutation
68	popm= repmat(empty_individual,nm,1);
69	for k=1:nm
70	% Select Parent
71	i=randi([1 nPop]);
72	p=pop(i);
73	% Apply Mutation
74	popm(k).Position=Mutate(p.Position,mu,VarMin,VarMax);
75	% Evaluate Mutant
76	[popm(k).Cost,
77	popm(k).Out]=CostFunction(popm(k).Position);
78	end
79	% Create Merged Population
80	pop=[pop
81	popc
82	popm]; %#ok
83	% Sort Population
84	Costs=[pop.Cost];
85	[Costs, SortOrder]=sort(Costs);
86	pop=pop(SortOrder);
87	% Update Worst Cost
88	WorstCost=max(WorstCost,pop(end).Cost);
89	% Truncation
90	pop=pop(1:nPop);

Metode: GCUK	
Step	Deskripsi
91	Costs=Costs(1:nPop);
92	% Store Best Solution Ever Found
93	BestSol=pop(1);
94	% Store Best Cost Ever Found
95	BestCost(it)=BestSol.Cost;
96	% Show Iteration Information
97	disp(['Iteration ' num2str(it) ': Best Cost = ']
98	num2str(BestCost(it))]);
99	end

Lampiran 3 : Alur Eksperimen Metode DCPSO

Metode: DCPSO	
Step	Deskripsi
1	%% Problem Definition
2	data = load('iris');
3	X = data.X/100;
4	X(:,5)=[];
5	k = 10;
6	CostFunction=@(s) ClusteringCost(s, X); % Cost Function
7	VarSize=[k size(X,2)+1]; % Decision Variables Matrix Size
8	nVar=prod(VarSize); % Number of Decision Variables
9	VarMin= repmat([min(X) 0],k,1); % Lower Bound of Variables
10	VarMax= repmat([max(X) 1],k,1); % Upper Bound of Variables
11	%% PSO Parameters
12	MaxIt=200; % Maximum Number of Iterations
13	nPop=50; % Population Size (Swarm Size)
14	w=1; % Inertia Weight
15	wdamp=0.99; % Inertia Weight Damping Ratio
16	c1=1.5; % Personal Learning Coefficient
17	c2=2.0; % Global Learning Coefficient
18	% Velocity Limits
19	VelMax=0.1*(VarMax-VarMin);
20	VelMin=-VelMax;
21	%% Initialization
22	empty_particle.Position=[];
23	empty_particle.Cost=[];
24	empty_particle.Out=[];
25	empty_particle.Velocity=[];
26	empty_particle.Best.Position=[];
27	empty_particle.Best.Cost=[];
28	empty_particle.Best.Out=[];
29	particle=repmat(empty_particle,nPop,1);
30	BestSol.Cost=inf;
31	for i=1:nPop
32	% Initialize Position
33	particle(i).Position=unifrnd(VarMin,VarMax,VarSize);
34	% Initialize Velocity
35	particle(i).Velocity=zeros(VarSize);
36	% Evaluation
37	[part,
38	particle(i).Out]=CostFunction(particle(i).Position)
39	% Update Personal Best
40	particle(i).Best.Position=particle(i).Position;
41	particle(i).Best.Cost=particle(i).Cost;
42	particle(i).Best.Out=particle(i).Out;

Metode: DCPSO

Step	Deskripsi
43	% Update Global Best
44	if particle(i).Best.Cost<BestSol.Cost
45	BestSol=particle(i).Best;
46	end
47	end
48	BestCost=zeros(MaxIt,1);
49	%% PSO Main Loop
50	for it=1:MaxIt
51	for i=1:nPop
52	% Update Velocity
53	particle(i).Velocity = w*particle(i).Velocity ...
54	+c1*rand(VarSize).*(particle(i).Best.Position-
55	particle(i).Position) ...
56	+c2*rand(VarSize).*(BestSol.Position-
57	particle(i).Position);
58	% Apply Velocity Limits
59	particle(i).Velocity =
60	max(particle(i).Velocity, VelMin);
61	particle(i).Velocity =
62	min(particle(i).Velocity, VelMax);
63	% Update Position
64	particle(i).Position = particle(i).Position +
65	particle(i).Velocity;
66	% Velocity Mirror Effect
67	IsOutside=(particle(i).Position<VarMin
68	particle(i).Position>VarMax);
69	particle(i).Velocity(IsOutside)=-
70	particle(i).Velocity(IsOutside);
71	% Apply Position Limits
72	particle(i).Position =
73	max(particle(i).Position, VarMin);
74	particle(i).Position =
75	min(particle(i).Position, VarMax);
76	% Evaluation
77	[particle(i).Cost, particle(i).Out] =
78	CostFunction(particle(i).Position);
79	% Update Personal Best
80	if particle(i).Cost<particle(i).Best.Cost
81	particle(i).Best.Position=particle(i).Position;
82	particle(i).Best.Cost=particle(i).Cost;
83	particle(i).Best.Out=particle(i).Out;
84	% Update Global Best
85	if particle(i).Best.Cost<BestSol.Cost
86	BestSol=particle(i).Best;
87	end
88	end
89	end
90	
91	BestCost(it)=BestSol.Cost;
92	
93	disp(['Iteration ' num2str(it) ': Best Cost = '
94	num2str(BestCost(it))]);
95	% Plot Solution
96	figure(1);
97	PlotSolution(X, BestSol);
98	pause(0.01);
99	

Metode: DCPSO		
Step		Deskripsi
100	w=w*wdamp;	
	end	

Lampiran 4 : Alur Eksperimen Metode ACDE

Metode: ACDE		
Step		Deskripsi
1	%% DE Parameters	
2	MaxIt=200;	% Nilai maximum iterasi
3	nPop= 50;	% Besarnya populasi
4	beta_min= 0.2;	% Faktor Skala Batas Atas
5	beta_max= 0.8;	% Faktor Skala Batas Bawah
6	pCR=0.2;	% Crossover Probability
7	%% Problem Definition	
8	data = load('iris');	
9	X = data.X;	
10	X(:,5)=[];	
11	k = 10;	
12	VarSize=[k size(X,2)+1];	% Decision Variables Matrix Size
13	nVar=prod(VarSize);	% Number of Decision Variables
14	VarMin= repmat([min(X) 0],k,1);	% Lower Bound of
15	Variables	
16	VarMax= repmat([max(X) 1],k,1);	% Upper Bound of
17	Variables (), (0.60)	
18	CostFunction=@(s) ClusteringCost(s, X);	% Cost Function
19	%% Initialization	
20	empty_individual.Position=[];	
21	empty_individual.Cost=[];	
22	empty_individual.Out=[];	
23	BestSol.Cost=inf;	
24	pop=repmat(empty_individual,nPop,1);	
25	for i=1:nPop	
26	pop(i).Position=unifrnd(VarMin,VarMax,VarSize);	
27	[pop(i).Cost, pop(i).Out]=CostFunction(pop(i).Position);	
28	if pop(i).Cost<BestSol.Cost	
29	BestSol=pop(i);	
30	end	
31	end	
32	BestCost=zeros(MaxIt,1);	
33	%% DE Main Loop	
34	for it=1:MaxIt	
35	for i=1:nPop	
36	x=pop(i).Position;	
37	A=randperm(nPop);	
38	A(A==i)=[];	
39	a=A(1);	
40	b=A(2);	
41	c=A(3);	
42	% Mutation	
43	%beta=unifrnd(beta_min,beta_max);	
44	beta=unifrnd(beta_min,beta_max,VarSize);	
45	y=pop(a).Position+beta.*(pop(b).Position-	
46	pop(c).Position);	
47	y=max(y,VarMin);	
48	y=min(y,VarMax);	
49	% Crossover	

Metode: ACDE

```

50         z=zeros(size(x));
51         j0=randi([1 numel(x)]);
52         for j=1:numel(x)
53             if j==j0 || rand<=pCR
54                 z(j)=y(j);
55             else
56                 z(j)=x(j);
57             end
58         end
59         NewSol.Position=z;
60         [NewSol.Cost,
61 NewSol.Out]=CostFunction(NewSol.Position);
62         if NewSol.Cost<pop(i).Cost
63             pop(i)=NewSol;
64             if pop(i).Cost<BestSol.Cost
65                 BestSol=pop(i);
66             end
67         end
68     end
69     % Update Best Cost
70     BestCost(it)=BestSol.Cost;
71     % Show Iteration Information
72     disp(['Iteration ' num2str(it) ': Best Cost = '
73 num2str(BestCost(it))]);
74     % Plot Solution
75     figure(1);
76     PlotSolution(X, BestSol);
77     pause(0.01);
78 end

```

Lampiran 5 : Alur Psedudocode Metode Usulan

Metode: Metode Usulan (Metode UCC+ACDE)

Step	Deskripsi
1	%% DE Parameters
2	MaxIt=200; % Nilai maximum iterasi
3	nPop= 50; % Besarnya populasi
4	%F = 0.75;
5	beta_min= 3.01; % Lower Bound of Scaling Factor
6	beta_max= 3.92; % Upper Bound of Scaling Factor
7	pCR=0.2; % Crossover Probability
8	
9	%% Defenisikan masalah
10	data = load('iris');
11	X = data.X;
12	X(:,5)=[];
13	k = 10;
14	VarSize=[k size(X,2)+1]; % Decision Variables Matrix Size
15	nVar=prod(VarSize); % Number of Decision Variables
16	VarMin= repmat([min(X) 0],k,1); % Lower Bound of
17	Variables
18	VarMax= repmat([max(X) 1],k,1); % Upper Bound of
19	Variables
20	CostFunction=@(s) ClusteringCost(s, X); % Cost Function
21	
22	%% Inisial populasi
23	empty_individual.Position=[];

Metode: Metode Usulan (Metode UCC+ACDE)	
Step	Deskripsi
24	empty_individual.Cost=[];
25	empty_individual.Out=[];
26	BestSol.Cost=inf;
27	pop= repmat(empty_individual,nPop,1);
28	for i=1:nPop
29	pop(i).Position=unifrnd(VarMin,VarMax,VarSize);
30	[pop(i).Cost,
31	pop(i).Out]=CostFunction(pop(i).Position);
32	if pop(i).Cost<BestSol.Cost
33	BestSol=pop(i);
34	end
35	end
36	BestCost=zeros(MaxIt,1);
37	
38	%% DE Main Loop
39	for it=1:MaxIt
40	for i=1:nPop
41	x=pop(i).Position;
42	A=randperm(nPop);
43	A(A==i)=[];
44	a=A(1);
45	b=A(2);b
46	c=A(3);
47	
48	% Mutation
49	%beta=unifrnd(beta_min,beta_max);
50	beta=unifrnd (beta_min,beta_max,VarSize);
51	y=pop(a).Position+beta.*(pop(b).Position-
52	pop(c).Position);
53	y=max(y,VarMin);
54	y=min(y,VarMax);
55	
56	% Crossover
57	z=zeros(size(x));
58	j0=randi([1 numel(x)]);
59	for j=1: numel(x)
60	if j==j0 rand<=pCR
61	z(j)=y(j);
62	else
63	z(j)=x(j);
64	end
65	end
66	NewSol.Position=z;
67	[NewSol.Cost,
68	NewSol.Out]=CostFunction(NewSol.Position);
69	if NewSol.Cost<pop(i).Cost
70	pop(i)=NewSol;
71	if pop(i).Cost<BestSol.Cost
72	BestSol=pop(i);
73	end
74	end
75	end
76	end
77	
78	% Update Best Cost
79	BestCost(it)=BestSol.Cost;
80	

Metode: Metode Usulan (Metode UCC+ACDE)	
Step	Deskripsi
81	% Show Iteration Information
82	disp(['Iteration ' num2str(it) ': Best Cost = ']
83	num2str(BestCost(it))]);
84	
85	% Plot Solution
86	figure(1);
87	PlotSolution(X, BestSol);
88	pause(0.01);
89	end
90	