

**PREDIKSI *REUSABILITY* PADA *ENTROPY* PERANGKAT
LUNAK *ECLIPSE EQUINOX* MENGGUNAKAN
PENDEKATAN *FUZZY INFERENCE*
SYSTEM MAMDANI**



TESIS

Diajukan sebagai salah satu syarat untuk memperoleh gelar

Magister Ilmu Komputer (M.Kom)

AHMAD FAUZI
14002161

PROGRAM PASCASARJANA MAGISTER ILMU KOMPUTER
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
NUSA MANDIRI
JAKARTA
2019

PERSETUJUAN DAN PENGESAHAN TESIS

Tesis ini diajukan oleh:

Nama : Ahmad Fauzi
NIM : 14002161
Program Studi : Ilmu Komputer
Jenjang : Program STRATA-2
Judul Tesis : Prediksi Reusability Pada Entropy Perangkat Lunak
Eclipse Equinox Menggunakan Pendekatan Fuzzy
Inference System Mamdani

Telah dipertahankan pada periode 2019-1 dihadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh MAGISTER ILMU KOMPUTER (M.Kom) pada Program STRATA-2 Program Studi Ilmu Komputer di Sekolah Tinggi Manajemen Informatika dan Komputer Nusa Mandiri (STMIK Nusa Mandiri).

Jakarta, 27 Agustus 2019

PEMBIMBING TESIS

Dosen Pembimbing : Dr. Dwiza Riana, S.Si, MM, M.Kom



DEWAN PENGUJI

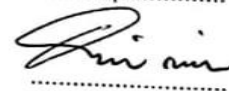
Penguji I : Dr. Windu Gata, M.Kom



Penguji II : Dr. Sularso Budilaksono, M.Kom



Penguji III /
Dosen
Pembimbing : Dr. Dwiza Riana, S.Si, MM, M.Kom



KATA PENGANTAR

Puji dan syukur atas kehadiran Allah, SWT, Rabb semesta alam yang telah memberi nikmat iman, Islam, dan kesehatan sehingga mempermudah langkah-langkah hamba beserta keluarga dan para sahabatnya yang selalu istiqomah hingga akhir solawat dan salam semoga terlimpahkan kepada Nabi Muhamad saw. zaman. Adapun judul tesis yang diambil yaitu “Prediksi *Reuasbility* Pada *Entropy* Perangkat Lunak *Eclipse Equinox* Menggunakan Pendekatan *Fuzzy Inference System Mamdani*”.

Tujuan penulisan tesis ini dibuat sebagai salah satu syarat untuk mendapatkan gelar Magister Ilmu Komputer (M. Kom) pada Program Studi Magister Ilmu Komputer Sekolah Tinggi Manajemen Informatika dan Komputer Nusa Mandiri Jakarta (STMIK Nusa Mandiri Jakarta).

Tesis ini diambil berdasarkan hasil penelitian mengenai kualitas dan *reusability* perangkat lunak. Selain itu, penulis juga melakukan studi *literatur* dari berbagai sumber *referensi*, baik dalam bentuk jurnal ilmiah, buku-buku *literatur*, dan lain sebagainya yang sesuai dengan judul tesis yang diambil.

Pada kesempatan ini, penulis ingin menyampaikan ucapan terima kasih kepada pihak-pihak yang telah membantu dalam penyusunan tesis ini, diantaranya:

1. Ibu Dr. Dwiza Riana, S.SI, MM, M.KOM selaku pembimbing tesis yang telah menyediakan waktu, pikiran, dan tenaga untuk membimbing penulis dalam menyelesaikan tesis ini.
2. Orang tua yang telah memberikan dukungan material dan moral kepada penulis.
3. Teman-teman seperjuangan selama menjalani perkuliahan program Studi MIIK STMIK Nusa Mandiri Jakarta.

Serta semua pihak yang tidak bisa disebutkan satu-persatu yang mendukung dalam penulisan tesis ini. Penulis menyadari bahwa masih banyak kekurangan dalam penulisan naskah tesis ini. Akhir kata semoga tesis ini dapat bermanfaat bagi penulis pada khususnya dan bagi pembaca pada umumnya.

Jakarta, 23 Agustus 2019

Ahmad Fauzi

Penulis

ABSTRAK

Nama : Ahmad Fauzi

NIM :14002161

Program Studi : Magister Ilmu Komputer

Jenjang : Strata Dua (S2)

Konsentrasi : Software Engineering

Judul : Software Engineering Judul Tesis :“ Prediksi *Reusability* Pada *Entropy* Perangkat Lunak *Eclipse Equinox* Menggunakan Pendekatan *Fuzzy Inference System* Mamdani

Evaluasi dan penilaian kualitas perangkat lunak penting dalam pengembangan sistem perangkat lunak. Ada banyak model kualitas perangkat lunak, diantaranya ISO/IEC 9126-1 yang merupakan hasil kolaborasi dari model kualitas perangkat lunak yang telah ada sebelumnya. Penilaian karakteristik kualitas sebuah perangkat lunak yang telah digambarkan dalam sebuah model kualitas perangkat lunak berdasarkan indikator properti desain *Chidamber Kemerer (CK) entropy metrics* dari terdiri dari 6 properti yaitu WMC, DIT, NOC, CBO, RFC dan LCOM. *Metrics* mengukur sifat tertentu sistem perangkat lunak dengan memetakannya ke dalam angka menurut aturan pengukuran yang objektif dan terdefinisi dengan baik. Melakukan penilaian kualitas perangkat lunak bermanfaat untuk membantu proses penjaminan kualitas dari perangkat lunak berorientasi objek yang akan dikembangkan . Untuk dapat membantu melakukan perhitungan kualitas perangkat lunak berdasar *metrics* pengujian penilaian kualitas dan *reusability* perangkat lunak terhadap proyek *eclipse equinox* dengan menggunakan *fuzzy inference system* mamdani, dan membagi kedalam 4 klasifikasi *bug* hasil pengujian yang didapat dari *Defuzzifikasi* sebesar 8,5 dari nilai angka tersebut masi tergolong sangat rendah dan bisa diklasifikasikan ke dalam *trivial bugs* dengan *reusability* nilai 0-30, sehingga bisa diartikan hasil dari pengujian memiliki kualitas dan *reusability* yang baik terhadap pengujian perangkat lunak

Kata kunci : *fuzzy inference system* mamdani, kualitas perangkat lunak, *chidamber kemerer metrics*

ABSTRACT

Nama : Ahmad Fauzi

NIM :14002161

Program Studi : Magister Ilmu Komputer

Jenjang : Strata Dua (S2)

Konsentrasi : Software Engineering

Judul : Software Engineering Judul Tesis :“ Prediksi Reuasbility Pada Entropy Perangkat Lunak Eclipse Equinox Menggunakan Pendekatan Fuzzy Inference System mamdani

Evaluation and assessment of software quality is important in developing software systems. There are many software quality models, including ISO / IEC 9126-1 which is the result of collaboration from pre-existing software quality models. An assessment of the quality characteristics of a software that has been described in a software quality model based on the Chidamber Kemerer (CK) entropy metrics design property indicators consists of 6 properties namely WMC, DIT, NOC, CBO, RFC and LCOM. Metrics measure the specific nature of software systems by mapping them into numbers according to objective and well-defined measurement rules. Conducting software quality assessment is useful to help the quality assurance process of object-oriented software that will be developed. To be able to help calculate software quality based on metrics testing quality assessment and software reusability of the eclipse equinox project by using the mamdani fuzzy inference system, and dividing into 4 bug classification test results obtained from Defuzzification of 8.5 from the value of those numbers are classified as a very low and can be classified into trivial bugs with rengs nilia 0-30, so it can be interpreted the results of the test have good quality and reusability of software testing.

Keywords: fuzzy inference system mamdani, software quality, chidamber kemerer metrics.

DAFTAR ISI

	Halaman
HALAMAN SAMPUL.....	i
HALAMAN JUDUL.....	ii
HALAMAN PERNYATAAN ORISINALITAS.....	iii
HALAMAN PENGESAHAN.....	iv
HALAMAN KONSULTASI BIMBINGAN.....	v
KATA PENGANTAR.....	vi
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS.....	viii
ABSTRAK.....	ix
ABSTRACT.....	x
DAFTAR ISI.....	xi
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR.....	xiv
BAB 1. PENDAHULUAN.....	1
1.1. Latar Belakang Penulisan.....	1
1.2. Identifikasi Masalah.....	2
1.3. Tujuan Penelitian.....	3
1.4. Ruang Lingkup Penelitian.....	3
1.5. Hipotesis.....	4
1.6. Sistematika Penulisan.....	4
BAB 2. LANDASAN TEORI.....	6
2.1. Tinjauan Pustaka.....	6
2.2. Tinjauan Studi.....	18
2.3. Tinjauan Objek Penelitian.....	21
BAB 3. METODE PENELITIAN.....	34
3.1. Perancangan Penelitian.....	24
3.2. Metode pengumpulan Sampel.....	25
3.3. Metode Pengumpulan Data.....	26
3.4. Teknik Analisis Data.....	26
3.5. Logika <i>Fuzzy Inference System</i> Mamdani.....	26
3.6. <i>Fuzzy Inference System</i>	26
BAB 4. HASIL PENELITIAN DAN PEMBAHASAN.....	37
4.1. Analisis Data dan Penerapan Logika Logika <i>Fuzzy Inference</i> <i>System Mamdani</i>	37
4.2. Mendekomposisikan Variabel Kedalam Himpunan Fuzzy ...	38
4.3. Pengujian	51
BAB 5. PENUTUP.....	57
5.1. Kesimpulan.....	58
5.2. Saran.....	59
DAFTAR PUSTAKA.....	60

DAFTAR TABEL

	Halaman
Tabel 2.1. Properti Kualitas Perangkat Lunak <i>dan</i> Parameter <i>Metrics</i>	12
Tabel 2.2. Hubungan dan pengaruh parameter metrics	13
Tabel 2.3. Penelitian Terkait.....	23
Tabel 3.1. <i>Threshold</i> CK Versi	27
Tabel 3.2. <i>Threshold</i> CK Penulis.....	27
Tabel 3.3. Hubungan Paramater CK <i>metrics</i> dengan Faktor Kualitas Perangkat lunak	27
Tabel 3.4. <i>Weight Methods per Class</i>	29
Tabel 3.5. <i>Coupling Between Classes</i>	30
Tabel 3.6. <i>Depth of Inheritance</i>	30
Tabel 3.7. <i>Number of Children</i>	30
Tabel 3.8. <i>Lack of Cohesion of Methods</i>	30
Tabel 3.9. <i>Response sets For a Class</i>	30
Tabel 4.1. <i>Threshold</i> CK <i>Eclipse Equinox</i>	36
Tabel 4.2. <i>Threshold</i> CK Penulis.....	37
Tabel 4.3. Aturan <i>Fuzzy</i> Penentuan Tingkat <i>Bug</i>	44

DAFTAR GAMBAR

	Halaman
Gambar 2.1. McCall's <i>Perangkat lunak Quality Factors</i>	10
Gambar 2.2. Fungsi keanggotaan <i>Linear Naik</i>	13
Gambar 2.3. Fungsi keanggotaan <i>Linear Turun</i>	14
Gambar 2.4. Fungsi keanggotaan <i>segitiga</i>	14
Gambar 2.5. Fungsi keanggotaan <i>trapesium</i>	15
Gambar 2.6. Fungsi keanggotaan <i>trapsium</i>	16
Gambar 3.1. Kerangka Pemikiran Penentuan.....	24
Gambar 3.2. <i>Weight Methods per Class</i>	31
Gambar 3.3. <i>Coupling Between Classes</i>	31
Gambar 3.4. <i>Depth of Inheritance</i>	31
Gambar 3.5. <i>Number of Children</i>	33
Gambar 3.6. <i>Lack of Cohesion of Methods</i>	33
Gambar 3.7. <i>Response sets For a Class</i>	33
Gambar 3.8. <i>Response Output</i>	34
Gambar 4.1. Grafik fungsi keanggotaan <i>Weighted Methods for Class</i>	38
Gambar 4.2. Grafik fungsi <i>Coupling between Objects</i>	39
Gambar 4.3. Grafik fungsi <i>Number of Children</i>	40
Gambar 4.4. Grafik fungsi <i>Depth of Inheritance Tree</i>	42
Gambar 4.5. Grafik fungsi <i>Lack of Cohesion of Methods</i>	43
Gambar 4.6. <i>Fuzzy Inference System Mamdani Editor</i>	46
Gambar 4.7. Himpunan <i>Fuzzy Input Indikator Weight Methods per Class</i>	46
Gambar 4.8. Himpunan <i>Fuzzy Input Indikator Coupling Between Classes</i>	47
Gambar 4.9. Himpunan <i>Fuzzy Input Indikator Depth of Inheritance</i>	47
Gambar 4.10. Himpunan <i>Fuzzy Input Indikator Number of Children</i>	48
Gambar 4.11. Himpunan <i>Fuzzy Input Indikator Lack of Cohesion of Methods</i>	48
Gambar 4.12. Himpunan <i>Fuzzy Input Indikator Response sets For a Class</i>	49
Gambar 4.13. <i>Rule Integritas</i>	49
Gambar 4.14. <i>Fuzzy Inference System Model Mamdani Editor Integritas</i>	52
Gambar 4.15. Himpunan <i>Fuzzy Ouput</i>	53
Gambar 4.12. Halaman utama Aplikasi.....	54
Gambar 4.13. Halaman <i>Upload Dataset</i>	55
Gambar 4.14. Halaman <i>Isi Dataset</i>	55
Gambar 4.15. Himpunan Hasil Perhitungan.....	56
Gambar 4.15. Himpunan Hasil Klasifikasi.....	56

BAB I

PENDAHULUAN

1.1. Latar Belakang Penelitian

Industri perangkat lunak bergerak menuju *reuse* skala besar, menghasilkan penghematan waktu dan uang. Untuk mengembangkan sistem baru dari awal sangat mahal dan membutuhkan waktu yang cukup panjang. Secara umum diasumsikan bahwa penggunaan kembali perangkat lunak yang ada akan meningkatkan keadaan aplikasi perangkat lunak menjadi baru. Konsep ini hampir secara universal diterima karena fakta jelas bahwa produk sebelumnya telah berjalan.

Mengidentifikasi kualitas komponen perangkat lunak dapat menggunakan *metrics* yang secara tepat memprediksi dari komponen perangkat lunak yang masih dapat digunakan dan dikembangkan dengan baik (Shri et al. 2010) . *Usabilitas metrics* ini jika diidentifikasi dalam tahap desain dan bahkan dalam tahap *coding* dapat membantu untuk mengurangi dan meningkatkan kualitas penggunaan kembali komponen *software* (Halim, Xandra, and Sim 2015).

Set data *Chidamber Kemerer (CK) metrics entropy* dari *eclipse equinox* terdiri dari 6 properti yaitu WMC, DIT, NOC, CBO, RFC dan LCOM, yang terdapat pada situs <http://bug.inf.usi.ch> akan digunakan untuk memperoleh analisis struktur komponen *software* berbasis *object oriented*, *Metrics entropy eclipse equinox* adalah implementasi spesifikasi kerangka kerja *open service gateway initiative*, dari satu set paket yang mengimplementasikan berbagai layanan *open service gateway initiative* (OSGI), dan infrastruktur lainnya untuk menjalankan sistem berbasis OSGI, Lebih umum, tujuan dari proyek *eclipse equinox* adalah menjadi komunitas OSGI kelas satu dan menumbuhkan visi

eclipse equinox sebagai lanskap kumpulan proyek, bertanggung jawab untuk mengembangkan dan memberikan implementasi kerangka kerja *open service gateway initiative* yang digunakan untuk semua *eclipse*. Sebagai tambahan proyek *open source*, kualitas *software* diinterpretasikan dalam parameter-parameter yang sesuai dengan CK *metrics* yang digunakan.,sehingga dapat ditentukan aplikasi yang telah berjalan dapat digunakan kembali sebagai proses *development* dengan yang lebih besar dengan tingkat kesalahan yang lebih kecil.*Entropy metrics eclipse equinox* yang digunakan pada penelitian ini mengevaluasi bagian dari *sub system* yang terdapat pada *eclipse*.

Penelitian sebelumnya yang memiliki data set yang sama dilakukan oleh M. Rizwan Jameel Qureshi dan Waseem 2012, yang mengevaluasi CK *metrics* untuk mengurangi cact desain perangkat lunak,dengan menggunakan analisis regresi.penelitian berikutnya,dengan *dataset* CK dilakukan oleh Roberto dilakukan pada 2013,penelitian tersebut fokus pada menganalisis setiap rilis perangkat lunak, namun tidak menganalisis *sub system* dari perangkat lunak

Penelitian lain oleh Arwin Halim mengukur tingkat *reusability* dan *effectivity*,tetapi hanya menggunakan 4 *metrics* saja.yaitu :(WMC, LCOM, CBO, DIT dan NOC) dan tidak meneliti variable (LOCM dan RFC). berdasarkan penelitian-penelitian sebelumnya maka perlu dilakukan penelitian lanjutan menggunakan metode selain analisa regresi.dan tidak hanya menganalisa di proyek utama, menganalisis pada *sub system*, serta menggunakan lengkap enam variable *independent* CK *metrics*. sehingga pada penelitian ini mengusulkan prediksi *reusability* pada *entropy eclipse equinox* menggunakan pendekatan *fuzzy* yang menganalisa secara lengkap hingga *sub system* proyek dan menggunakan enam variable (WMC, LCOM, CBO, DIT,NOC, LOCM dan RFC). Penulis akan melakukan penelitian yang menerapkan pendekatan *fuzzy inference system* mamdani dengan memanfaatkan *output* data unsur pengamatan

dari set data *entropy CK metrics* yang telah tersedia untuk memprakirakan kualitas perangkat lunak.

Ada tiga tahapan utama yang diharapkan dapat mengetahui tingkat *reusability* dari *sub system* pada proyek *eclipse equinox* yang sedang menjadi bahan penelitian ini. Dengan membagi 4 klasifikasi bug, sehingga bisa dikategorikan hasil yang didapatkan.

1.2. Identifikasi Masalah

1. Parameter property desain dan *indicator* apa saja yang diperlukan pada prediksi *reusability* pada *entropy* perangkat lunak *eclipse equinox*?
2. Bagaimana menentukan *indicator* penilaian dari set data yang ada?
3. Bagaimana mendekomposisikan *variable* kedalam himpunan *Fuzzy Inference System Mamdani*?
4. Apa yang dihasilkan dari proses *inference* dan *defuzzyfikasi*?
5. Apakah *Fuzzy Inference System Mamdani* dapat mengantisipasi kesalahan desain perangkat lunak?
6. Bagaimana klasifikasi yang dihasilkan?

1.3. Tujuan Penelitian

2. Parameter desain yang digunakan dalam prediksi *reusability* yaitu : *Weighted Methods Per Class (WMC)*, *Depth of Inheritance Tree (DIT)*, *Number of Children (NOC)*, *Coupling Between Object Classes (CBO)*, *Response for a Class (RFC)*, *Lack of Cohesion Method (LCOM)*. Enam parameter *Chidamber dan Kemerer* ini yang digunakan untuk mengukur tingkat *reusability* perangkat lunak berbasis *object oriented*.
3. Terlebih dahulu lakukan penentuan ambang batas nilai *CK metrics* dari set data. Untuk menunjukkan nilai ambang batas yang akan digunakan sebagai acuan penilaian.

4. mendekomposisikan variable dari enam parameter untuk dijadikan sebagai bahan *input dari system fuzzy logic* yang tergantung pada empat parameter skalar a, b, c dan d.
5. Aturan logika pengambilan keputusan (*Fuzzy inference*) mengamplifikasikan aturan-aturan *fuzzy* pada masukkan *fuzzy*, Prinsip logika *fuzzy* digunakan untuk mengkombinasikan aturan-aturan (*IF THEN*) dalam basis aturan suatu pemetaan dari suatu himpunan *fuzzy*. *output* yang dihasilkan merupakan suatu bilangan pada domain himpunan *fuzzy*. Jika diberikan suatu himpunan *fuzzy* dalam range tertentu, maka dapat diambil suatu nilai *crisp* tertentu sebagai output. *Defuzifikasi* pada komposisi aturan Mamdani.
6. Kesalahan desain bisa diantisipasi dalam proses perancangan perangkat lunak mulai dari proses development hingga implementasi, namun *Fuzzy Inference System* Mamdani dapat di gunakan untuk mengukur tingkat kualitas perangkat lunak untuk di gunakan kembali dalam proses pengembangan perangkat lunak yang lebih besar.
7. *Fuzzy Inference System* Mamdani sangat baik digunakan dalam prediksi kualitas perangkat lunak dari 6 CK *metrics*, dari pengujian ini akan dihasilkan empat klasifikasi bug yaitu: *trivial bugs, normal bugs, critical bugs, high priority bugs*

Tujuan di lakukan penelitian terhadap *entropy Chidamber dan Kemerer metrics* pada bagian dari *sub system project eclipse equinox* untuk mengetahui tingkat properti kualitas desain perangkat lunak seperti *efficiency, understandability, reusability, testability, maintainability* yang dihasilkan oleh organisasi pengembang open source, akan dibahas dengan melihat hubungan antara parameter-parameter CK dan properti kualitas desain perangkat lunak dengan menggunakan *Fuzzy Inference System* Mamdani.

1.4. Manfaat Penelitian

Penelitian ini diharapkan memberikan nilai manfaat diantaranya adalah sebagai berikut: Bagi Sivitas Akademik program pasca sarjana ilmu *computer science*:

1. Metode yang penulis pakai dapat dipergunakan untuk menganalisis dataset atau uji perangkat lunak dari software yang akan dikembangkan.
2. Dapat menjadi referensi dalam pembelajaran untuk mengukur kualitas disain perangkat lunak yang lainnya, baik itu aplikasi desktop, aplikasi *web mobile application*.
3. Menerapkan ilmu-ilmu yang diperoleh selama kuliah.
4. Membandingkan teori, melakukan studi *literature* terhadap jurnal yang berkaitan dan melakukan studi secara langsung

1.5. Ruang Lingkup Penelitian

Pada tahap ini penulis membatasi permasalahan penelitian, melakukan uji data *entropi chidamber dan kemerer metrics object oriented* dengan menggunakan *fuzzy inference system* mamdani sebagai metode untuk mengukur tingkat kualitas disain perangkat seperti *efficiency, understandability, reusability, testability, maintainability*

1.6. Sistematika Penulisan

Sistematika Penulisan laporan ini adalah sebagai berikut :

BAB I : PENDAHULUAN

Bab I berisi tentang latar belakang masalah, identifikasi masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan penelitian serta sistematika penulisan.

BAB II : LANDASAN TEORI

Bab ini merupakan penjelasan *entropy Chidamber Kemerer metrics* properti kualitas disain software seperti *efficiency, understandability, reusability, testability, maintainability* akan dibahas dengan melihat hubungan antara parameter-parameter *chidamber kemerer* dan properti kualitas disain perangkat lunak dan penjabaran mengenai *Fuzzy Inference System* Mamdani sebagai metode yang digunakan.

BAB III : METODOLOGI PENELITIAN

Bab ini menjelaskan tentang metode pengumpulan data, pengolahan awal data, metode yang diusulkan, eksperiment dan pengujian data.

BAB IV : ANALISIS PENELITIAN

Bab ini berisi hasil penelitian terkait dataset parameter *Chidamber Kemerer metrics*, parameter properti kualitas disain perangkat lunak. Dalam bab ini juga dibuat diagram alur dari beberapa tahapan pengukuran kualitas disain dari perangkat lunak.

BAB V : PENUTUP

Bab ini berisi kesimpulan dari penelitian yang telah dilakukan dan juga berisi saran-saran untuk pengembangan penelitian ini lebih lanjut.

BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

Metrics for object oriented software engineering CK merupakan salah satu *metrics* yang dipergunakan untuk mengukur kualitas desain sebuah perangkat lunak berdasarkan enam parameter dengan melihat pada perspektif *object oriented design* (Borys 2014). Pengukuran didefinisikan sebagai proses dimana nilai-nilai menjadi atribut dari entitas sehingga menggambarkan perangkat lunak sesuai dengan aturan yang jelas.

Kualitas desain sebuah perangkat lunak sangat penting untuk dapat selalu diukur, dengan demikian setiap pengembang akan dapat mengetahui sejauh mana kualitas perangkat lunak dapat berjalan dengan baik. Sehingga apabila ditemukan sebuah permasalahan akan cepat ditangani. Indikator penilaian dari pengukuran kualitas sebuah perangkat lunak dinamakan *metrics*. *Metrics* digunakan untuk mengukur kualitas desain sebuah perangkat lunak yang terdiri dari enam properti desain. *Weighted methods per class* (WMC), *depth of inheritance tree* (DIT), *number of children* (NOC), *coupling between object classes* (COB), *response for a class* (RFC), *lack of cohesion method* (LOCM) *The IEEE Standard* (Glossary, 1990).

2.1.1. Eclipse Equinox

Eclipse equinox adalah implementasi spesifikasi kerangka kerja OSGi (*Open source gateway initiative*), yang dijelaskan pada halaman utama web <https://projects.eclipse.org/projects/eclipse.equinox> yang mengimplementasikan berbagai layanan OSGi opsional dan infrastruktur lainnya untuk menjalankan sistem berbasis OSGi. Lebih umum tujuan dari proyek *equinox* adalah menjadi komunitas OSGi kelas satu dan menumbuhkan visi *eclipse* sebagai lanskap kumpulan. Sebagai bagian dari ini, bertanggung jawab untuk mengembangkan dan memberikan implementasi kerangka kerja OSGi yang utama digunakan untuk semua *eclipse*. Sebagai tambahan proyek *open source* untuk:

1. Implementasi semua aspek spesifikasi OSGi (termasuk CPEG dan EEG).
2. Investigasi dan penelitian terkait dengan versi spesifikasi OSGi dimasa depan dan masalah *runtime* terkait.
3. Pengembangan infrastruktur *non-standart* dianggap penting untuk menjalankan dan mengelola sistem berbasis OSGi.
4. Implementasi layanan kerangka kerja utama dan ekstensi yang diperlukan untuk menjalankan *eclipse* (misalnya, *eclipse adapter*, *ekstensi Registry*) dan dianggap secara umum bermanfaat bagi orang yang menggunakan OSGi.

Sebagai bagian dari proyek *eclipse*, kode *equinox* OSGi dikelola oleh *eclipse* PMC. *Equinox*, sebagai basis untuk semua *eclipse*, dikirimkan dengan semua rilis utama. Berbagai modul yang lain dikembangkan disini dan dapat dikirimkan secara independen pada jadwal yang berbeda dan tertera pada halaman utama.

2.1.2. Bug Perangkat lunak

Faults merupakan suatu cacat dalam perangkat lunak yang bisa mengakibatkan suatu perangkat lunak berjalan tidak sesuai dengan apa yang diharapkan atau dengan kata lain perangkat lunak tersebut rusak. Penting melakukan analisa penyebab perangkat lunak *error*, (Ambros, Lanza, dan Robbes 2010) hal ini dilakukan untuk melakukan pencegahan. kesalahan dari perangkat lunak karena kesalahan “ koding “*error*”, “kesalahan prosedur”, “kesalahan dokumentasi” atau “kesalahan data perangkat lunak ”. Sebagian besar berasal dari *human error*, baik dari sistem analis, *programmer*, *tester*, dokumentasi dan sering kali dari *client*. Terdapat sembilan penyebab *error* pada perangkat lunak:

1. Kesalahan dalam pendefinisian kebutuhan
2. Kesalahan komunikasi antara *client* dan *developer*
3. Penyimpangan kebutuhan perangkat lunak secara sengaja
4. Kesalahan *design* logika
5. Kesalahan kode program
6. Ketidaksesuaian antara dokumentasi dengan kode program
7. Kekurangan dari proses pengujian

8. Kesalahan prosedur
9. Kesalahan dokumentasi

2.1.3. *Metrics*

Mengeksplorasi hubungan antara beberapa *metrics* desain *object oriented* (termasuk *metrics* dari *suite* CK) dan sejauh mana perubahan kode, yang digunakan sebagai ukuran pengganti untuk upaya pemeliharaan. Demikian pula berdasarkan investigasi beberapa pengukuran *metrics* (termasuk CBO) dan *metrics* NOC dari *suite* CK. Studi ini menggunakan enam *metrics* desain CK dalam analisis *metrics* CK untuk kompleksitas desain *object oriented* implikasi cacat perangkat lunak.

1. *Weighted Methods Per class (WMC)*

WMC berhubungan secara langsung terhadap kompleksitas dari sebuah objek, karena metode adalah properti dari sebuah objek dan kompleksitas sebuah objek ditentukan oleh properti.

$$WMC = \sum_{i=1}^n C_i \quad C_i = \text{Static Kompleksitas} \quad (1)$$

Jumlah metode dan kompleksitas dari metode merupakan salah satu indikator berapa lama waktu dan usaha yang dibutuhkan untuk membangun dan memelihara objek. Objek dengan jumlah metode yang lebih banyak dapat membatasi kemungkinan objek tersebut untuk digunakan kembali (*reuse*). Nilai WMC yang tinggi mempunyai kecenderungan kegagalan perangkat lunak (Sandhu dan Singh 2008). Menurut www.aivosto.com toleransi nilai WMC adalah antara 20-50 dalam sebuah *class* atau maksimum 10% dari jumlah metode dalam sebuah *class*. (Goel 2011).

Menurut (Sandhu dan Singh 2008), dengan sampel 30 proyek C++ menunjukkan semakin tinggi nilai WMC akan meningkatkan *bugs* dan menurunkan kualitas. Namun penelitian tersebut juga tidak menyebutkan berapa *optimum range* yang dapat digunakan. Pada penelitian yang sama mengatakan bahwa nilai yang tinggi untuk proyek VB terlihat diperbolehkan.

2. ***Depth of Inheritance Tree (DIT)***

Jalur pewarisan maksimum dari kelas ke kelas *root*. Ini menilai seberapa dalam kelas dalam struktur hierarki (*Assistant Professor, Department of MCA, SHIMT, sitapur*) kedalaman dari sebuah *class* dalam *inheritance* hirarki dimana angka maksimum dimulai dari *class* atas dan menurun pada *class root* pada sebuah hirarki. Sehingga hirarki yang dalam akan menyebabkan tingginya kompleksitas dari desain karena makin banyak metode dan *class* yang terlibat. Rekomendasi www.aivosto.com nilai DIT adalah 5 atau kurang (Halim, Xandra, and Sim 2015). Dokumentasi *Visual Studio.Net* menyebutkan nilai $DIT \leq 5$. Beberapa sumber lain mengijinkan nilai DIT sampai dengan 8.

3. ***Number of Children (NOC)***

NOC adalah jumlah *subclass* dalam sebuah *class hierarchy*. NOC merupakan indikator besarnya pengaruh sebuah *class* terhadap desain sistem secara keseluruhan. Semakin besar nilai NOC, semakin besar pula potensi ketidakcocokan *subclass* dengan abstraksi pada *parent class*. (Shyam R. Chidamber dan Chris F. Kemerer) hal ini dapat mengakibatkan kesalahan penggunaan *subclass* seperti ketika menggunakan salah satu *subclass* dari sekian banyak *subclass* yang lain dalam salah satu program.

4. ***Coupling Between Object Classes (CBO)***

CBO menghitung *class* yang berhubungan dengan *class* yang lain. Hal ini dihitung terhadap *non-inheritance class*. Sehingga semakin sedikit *class* yang berhubungan maka mengindikasikan *class* yang baik. Karena meningkatkan *modularity* dan *reuse* (Halim, Xandra, and Sim 2015). Semakin banyak *class* yang tidak tergantung satu sama lain akan lebih baik untuk digunakan dalam aplikasi yang lain. Hal ini sangat konsisten dengan tingkat *interdependency* antara modul. Berapa nilai maksimum yang diijinkan? $CBO > 14$ terlalu tinggi.

5. ***Response for a class (RFC)***

RFC adalah jumlah semua metode yang dipanggil sebagai respon terhadap diluar objek dari sebuah *class*. RFC juga mengukur komunikasi antara objek (Halim, Xandra, dan Sim 2015). Hal ini berlaku terhadap semua metode yang

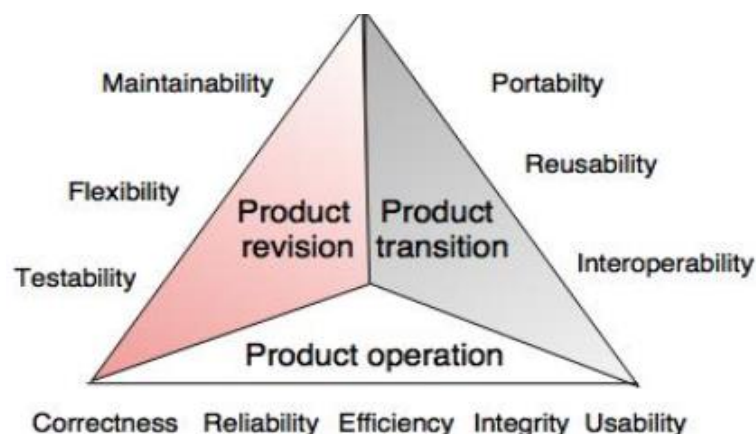
diakses dalam *class* hirarki. Sehingga semakin banyak metode yang digunakan untuk merespon objek dari luar semakin kompleks dan meningkatkan waktu tes.

6. *Lack of Cohesion Method (LCOM)*

LCOM mengukur ketidaksamaan metode dalam sebuah *class* dari *instance* variabel atau atribut. Semakin kecil nilai LCOM, semakin besar keandalan suatu perangkat lunak. Nilai LCOM yang tinggi umumnya menunjukkan kelas *kohesif* yang buruk. Sehingga lebih sederhana dan memiliki sifat *reusability* yang tinggi. Semua parameter MOOSE CK tersebut digunakan sebagai tolak ukur dalam. Parameter-parameter tersebut akan digabungkan dengan parameter properti kualitas, mengenai properti kualitas lebih lengkapnya akan dijelaskan pada sub bab selanjutnya.

2.1.4. Properti Kualitas Desain Perangkat lunak

Beberapa penelitian telah menghasilkan model kualitas perangkat lunak dan perspektif yang berbeda-beda. Salah satunya penelitian yang dilakukan oleh McCall dengan melihat dari tiga perspektif yaitu *product revision*, *product transition* dan *product operations* (Suresh, Pati, dan Rath 2012). (Axelius et al. 2019). Mengembangkan dari masing-masing perspektif menjadi beberapa faktor kualitas perangkat lunak yaitu *maintainability*, *flexibility*, *testability*, *portability*, *reusability*, *interoperability*, *correctness*, *reliability*, *efficiency*, *integrity* and *usability*. Sehingga total memiliki 11 faktor kualitas perangkat lunak.



Sumber: Ambros, Lanza and Robbes, (2010)

Gambar 2.1. McCall's Perangkat lunak Quality Factors

2.1.5. Hubungan Properti Kualitas Desain Perangkat lunak dan Parameter

Tujuh faktor kualitas *portability*, *reliability*, *efficiency*, *usability*, *testability*, *understandability* dan *flexibility*. ISO juga mengeluarkan model kualitas perangkat lunak yang dituangkan dalam ISO1926-1 dan memiliki enam karakteristik kualitas perangkat lunak *functionality*, *reliability*, *usability*, *efficiency*, *maintainability*, *portability* yang masing-masing dipecah dalam beberapa sub karakteristik.

Kualitas yang dapat dievaluasi untuk mengukur kualitas kode dan desain yaitu: *efficiency*, *complexity*, *understandability*, *reusability* dan *maintainability* dan *testability*. Masing-masing parameter dalam *metrics* memiliki kontribusi yang berbeda pada properti kualitas desain perangkat lunak yang dapat dilihat pada Tabel 2.3.

Tabel 2.1. Properti Kualitas Perangkat Lunak dan Parameter Metrics

Properti Kualitas Perangkat lunak	Metrics
<i>Efficiency</i>	LOCM,COB,DIT,NOC
<i>Complexity</i>	CC (Traditional Metrics)
<i>Understandability</i>	WMC,RFC,DIT
<i>Reusability</i>	WMC,LCOM,COB,DIT,NOC
<i>Maintainability/Testability</i>	WMC,RFC,DIT,NOC

Sumber : (Sumber: Ambros, Lanza dan Robbes, 2010)

Pengaruh parameter CK *metrics* pada properti kualitas desain perangkat lunak. Menurut (Mago, Kaur, dan Saurabh 2012), dan (Hyatt, Rosenberg, dan Ph 1997) menjelaskan hubungan dan pengaruh antara parameter-parameter *object orientid metrics* serta properti kualitas desain perangkat lunak menunjukkan bahwa nilai parameter CK berbanding terbalik dengan kualitas desain perangkat lunak seperti pada Tabel 2.1. dapat dilihat di bawah ini.

Tabel 2.2. Hubungan dan Pengaruh Parameter *Metrics*

Metric Develop	Objective	Testing Efforts	Understand Ability	Maintain-ability	Effort	Reuse
Complexity	↓	↓	↑	↑		
Size (LOC)	↓	↓	↑	↑		
Comment %	↑	↓		↑	↓	
WMC	↓			↑	↓	↑
RFC	↓	↓	↑			↑
LOCM	↓		↑	↑	↓	↑
COB	↓	↓	↑	↑		↑

Sumber : (Sumber: Ambros, Lanza dan Robbes, 2010)

ISO 9126 adalah standar terhadap kualitas perangkat lunak yang diakui secara internasional. ISO 9126 mendefinisikan kualitas produk perangkat lunak model karakteristik mutu, dan metrik terkait yang digunakan untuk mengevaluasi dan menetapkan kualitas sebuah produk perangkat lunak. Selain itu standar ISO juga harus dipenuhi dari sisi manajemen. Jika manajemennya tidak memenuhi standar ISO maka tidak dapat diberikan sertifikat standar ISO (Axelius et al. 2019). Faktor kualitas menurut ISO 9126 meliputi enam karakteristik kualitas sebagai berikut:

1. *Functionality* (fungsionalitas). Kemampuan perangkat lunak untuk menyediakan fungsi sesuai kebutuhan *user* dan memuaskan *user*.
 - a. *Suitability* adalah kemampuan perangkat lunak untuk menyediakan serangkaian fungsi yang sesuai dengan tugas-tugas tertentu dan tujuan pengguna.
 - b. *Accuracy* adalah kemampuan perangkat lunak dalam memberikan hasil yang sesuai dengan kebutuhan.

- c. *Security* adalah kemampuan perangkat lunak untuk mencegah akses yang tidak diinginkan, menghadapi penyusup (*hacker*) maupun otorisasi dalam modifikasi data.
 - d. *Interoperability* adalah kemampuan perangkat lunak untuk berinteraksi dengan satu atau lebih sistem tertentu.
 - e. *Compliance* adalah kemampuan perangkat lunak dalam memenuhi standar dan kebutuhan sesuai peraturan yang berlaku.
- 2. *Reliability* adalah kemampuan perangkat lunak untuk mempertahankan tingkat kinerja tertentu atau *performance* dari perangkat lunak (akurasi, konsistensi, kesederhanaan, toleransi kesalahan).
 - a. *Maturity* adalah kemampuan perangkat lunak untuk menghindari kegagalan sebagai akibat dari kesalahan dalam perangkat lunak.
 - b. *Fault tolerance* adalah perangkat lunak mempertahankan kinerjanya jika terjadi kesalahan perangkat lunak.
 - c. *Recoverability* adalah kemampuan perangkat lunak membangun kembali tingkat kinerja ketika terjadi kegagalan sistem, termasuk data dan koneksi jaringan.
- 3. *Usability* adalah kemampuan perangkat lunak untuk dipahami, dipelajari, digunakan, dan menarik bagi pengguna.
 - a. *Understandibility* adalah kemampuan perangkat lunak dalam kemudahan untuk dipahami.
 - b. *Learnability* adalah kemampuan perangkat lunak dalam kemudahan untuk dipelajari.
 - c. *Operability* adalah kemampuan perangkat lunak dalam kemudahan untuk dioperasikan.
 - d. *Attractiveness* adalah kemampuan perangkat lunak dalam menarik pengguna.
- 4. *Efficiency* adalah kemampuan perangkat lunak untuk memberikan kinerja yang sesuai dan relatif terhadap jumlah sumber daya yang digunakan pada saat keadaan tersebut (ex: efisiensi penyimpanan).

- a. *Time behaviour* kemampuan perangkat lunak dalam memberikan respon dan waktu pengolahan yang sesuai saat melakukan fungsinya.
 - b. *Resource behavior* adalah kemampuan perangkat lunak dalam menggunakan sumber daya yang dimilikinya ketika melakukan fungsi yang ditentukan.
 - c. *Analyzability* adalah kemampuan perangkat lunak dalam mendiagnosis kekurangan atau penyebab kegagalan.
- 5. *Maintainability* (pemeliharaan). Kemampuan perangkat lunak untuk dimodifikasi. *Modifikasi* meliputi koreksi, perbaikan atau adaptasi terhadap perubahan lingkungan, persyaratan, dan spesifikasi fungsional (ex: konsistensi).
 - a. *Changeability* adalah kemampuan perangkat lunak untuk dimodifikasi tertentu.
 - b. *Stability* adalah kemampuan perangkat lunak untuk meminimalkan efek tak terduga dari modifikasi perangkat lunak.
 - c. *Testability* adalah kemampuan perangkat lunak untuk dimodifikasi dan divalidasi perangkat lunak lain.
- 6. *Portability* (Portabilitas). Kemampuan perangkat lunak untuk ditransfer dari satu lingkungan ke lingkungan lain atau kemampuan perangkat lunak beradaptasi saat digunakan di area tertentu.
 - a. *Adaptability* kemampuan perangkat lunak untuk diadaptasikan pada lingkungan yang berbeda-beda.
 - b. *Instalability* kemampuan perangkat lunak untuk diinstal dalam lingkungan yang berbeda-beda.
 - c. *Coexistence* kemampuan perangkat lunak untuk berdampingan dengan perangkat lunak lainnya dalam satu lingkungan dengan berbagi sumber daya.
 - d. *Replaceability* kemampuan perangkat lunak untuk digunakan sebagai pengganti perangkat lunak lainnya.

2.1.6. Logika Fuzzy

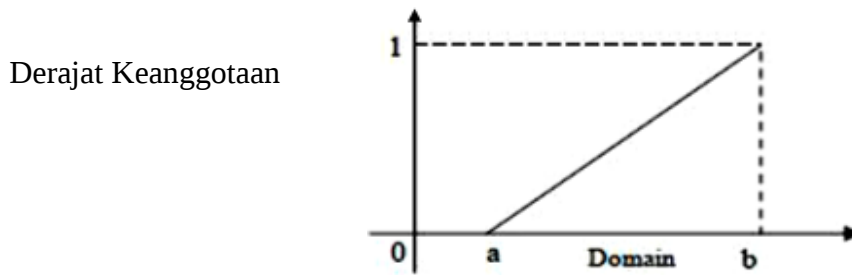
Fuzzy Logic adalah suatu cabang ilmu *Artificial Intelligence*, yaitu suatu pengetahuan yang membuat komputer dapat meniru kecerdasan manusia sehingga diharapkan komputer dapat melakukan hal-hal yang apabila dikerjakan manusia memerlukan kecerdasan. Logika *fuzzy* merupakan pengembangan dari logika *primitif* yang hanya mengenal keadaan, yaitu “ya” atau “tidak”. Dengan adanya logika *fuzzy*, dapat mengenal peubah linguistik seperti “lebih besar”, “besar”, “sangat besar”, dan sebagainya. Dengan demikian, aplikasi logika *fuzzy* akan menyebabkan sistem lebih adaptif (Jang 1995). Dalam membangun sebuah sistem *fuzzy* dikenal beberapa metode penalaran yaitu metode tsukomoto, metode Mamdani dan metode sugeno. Untuk perancangan sistem pendukung keputusan digunakan metode Mamdani.

Logika *fuzzy* merupakan sebuah logika yang memiliki derajat keanggotaan diantara 0 dan 1 dimana berbeda dengan logika klasik *boolean* yang memiliki nilai 0 dan 1 saja. Dengan demikian, variabel dalam logika *fuzzy* dideskripsikan dalam bentuk himpunan *fuzzy* (Kumar 2019), diantaranya dalam bentuk *linear* turun pada Gambar 2.1, *linear* naik Gambar 2.2, segitiga Gambar 2.3, dan trapesium Gambar 2.4. menunjukkan salah satu bentuk himpunan *fuzzy*. *Membership Function* (MF) besarnya derajat keanggotaan untuk setiap nilai pada variabel.

1. *Linear* Naik

Kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol [0] bergerak kekanan menuju kenilai domain yang memiliki derajat keanggotaan lebih tinggi yang disebut dengan representasi fungsi *linear* naik. Representasi fungsi keanggotaan untuk *linear* naik adalah sebagai berikut :

$$Linear \text{ Naik } (x,a,b) \left\{ \begin{array}{ll} a; & ; x \leq a \text{ atau } x \geq c \\ (x - a) / (b - a) & ; a \leq x \leq b \\ (b - a) / (c - b) & ; b \leq x \leq c \end{array} \right. \quad (2)$$



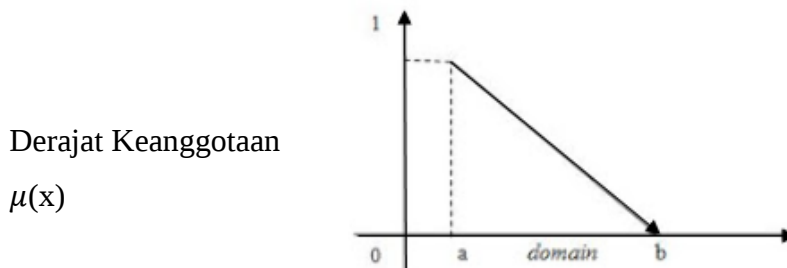
Sumber: (Saelan 2009)

Gambar 2.2. Fungsi keanggotaan *Linear Naik*

2. *Linear Turun*

Fungsi *Linear* turun merupakan kebalikan dari fungsi *linear* naik. Garis lurus dimulai dari nilai domain dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian bergerak menurun ke nilai domain yang memiliki derajat keanggotaan lebih rendah. Representasi fungsi keanggotaan untuk *linear* turun adalah sebagai berikut:

$$\text{Linear Turun } (x,a,b) \begin{cases} (x - a) / (b - a) & ; a \leq x \leq b \\ 0 & ; x \geq b \end{cases} \quad (3)$$



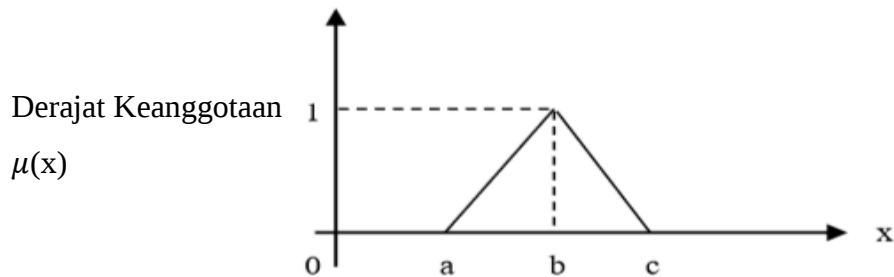
Sumber: (K, Suyatno, and Rahayu 2011)

Gambar 2.3. Fungsi keanggotaan *Linear Turun*

3. Segitiga

Untuk menentukan derajat keanggotaan dari himpunan *fuzzy* yang dirancang, maka diperlukan fungsi dari himpunan tersebut. Fungsi ini dibangun berdasarkan persamaan garis yang dibentuk oleh himpunan *fuzzy* tersebut. Contoh fungsi dari himpunan segitiga dimana $f(x,a,b,c)$ adalah derajat keanggotaan, (Kumar 2019) adalah nilai dari variabel, berturut-turut adalah nilai awal, tengah dan akhir dari variabel.

$$\text{Segitiga } (x,a,b,c) \begin{cases} 0; & ; x \leq a \text{ atau } x \geq c \\ (x - a) / (b - a) & ; a \leq x \leq b \\ (b - a) / (c - b) & ; b \leq x \leq c \end{cases} \quad (3)$$



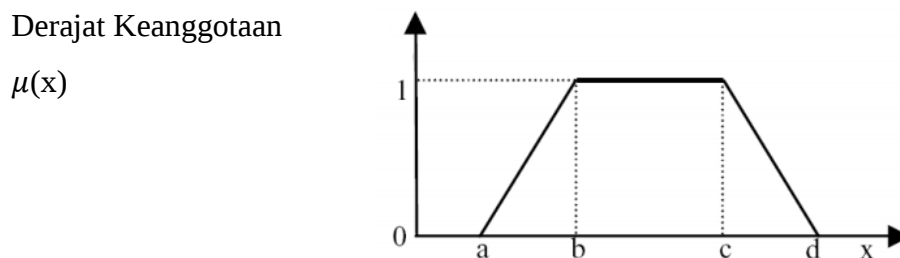
Sumber: (Wahab et al. 2017)

Gambar 2.4. Fungsi keanggotaan segitiga

4. Kurva trapesium

Kurva trapesium pada dasarnya menyerupai bentuk segitiga, hanya saja ada beberapa titik yang memiliki nilai keanggotaan satu. Representasi fungsi keanggotaan untuk kurva trapesium adalah sebagai parameter $\{a, b, c, d\}$ (dengan $a < b < c < d$) menentukan koordinat x pada empat sudut dari fungsi keanggotaan trapesium.

$$\text{Trapesium } (x,a,b,c,d) \begin{cases} a; & ; x \geq a \\ (x - a) / (b - a) & ; a \leq x \leq b \\ 1; & ; b \leq x \leq c \\ (d - x) / (d - c) & ; c \leq x \leq d \\ 0; & ; x \geq d \end{cases} \quad (4)$$



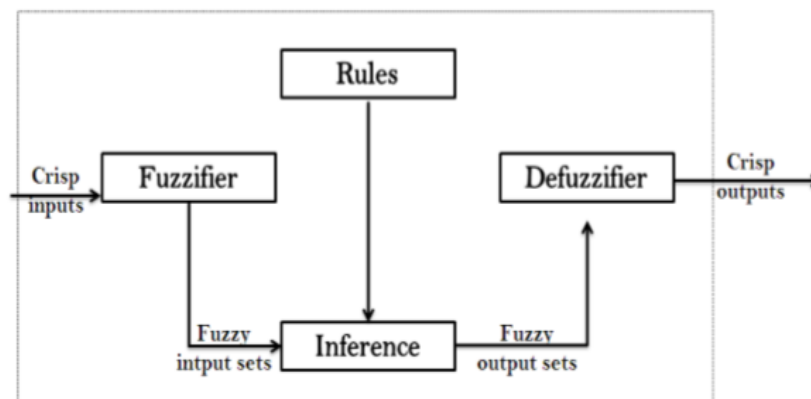
Sumber: (K, Suyatno, and Rahayu 2011)

Gambar 2.5. Fungsi keanggotaan trapesium

2.1.7. Sistem Inferensi Fuzzy Mamdani

Sistem Inferensi Fuzzy Mamdani (FIS) tampaknya menjadi pendekatan yang berpotensi cocok. FIS mampu mengurangi ketidakpastian ketika memecahkan masalah yang kompleks, dan keuntungan dari FIS dirangkum oleh (Haryatmi and Mashuri 2016) sebagai berikut:

1. FIS memungkinkan ekspresi eksplisit pengetahuan sistem melalui *fuzzy* “if – then” rules.
2. FIS berurusan dengan ketidakpastian subjektif (ketidak jelasan, ketidak jelasan, dan ketidak tepatan) yang melekat dalam cara pendekatan para ahli masalah mereka.
3. FIS dapat menggabungkan data numerik dan kategorikal.
4. FIS memberikan dasar matematika yang kuat.



Sumber : (K, Suyatno, and Rahayu 2011)

Gambar 2.6. Fungsi keanggotaan trapesium

Metode Mamdani sering juga dikenal dengan nama metode *Max-Min* atau *Max-Product*. Metode ini diperkenalkan oleh (ebrahim Mamdani pada tahun 1975). Untuk memperoleh *output* diperlukan empat tahap yaitu (K, Suyatno, and Rahayu 2011).

1. Pembentukan himpunan *fuzzy* metode Mamdani, baik variabel *input* maupun *variable output* dibagi menjadi satu atau lebih himpunan *fuzzy*, himpunan *fuzzy* diambil dari fungsi keanggotaan dinyatakan sebagai fungsi matematis tertentu. Derajat keanggotaan dari masing-masing elemen semesta pembicaraan memerlukan perhitungan. Fungsi matematis yang biasa

digunakan yaitu fungsi trapesium (Utara and Utara 2018). Fungsi keanggotaan trapesium mempunyai bentuk seperti pada Gambar 2.6. Dan dispesifikasikan oleh empat parameter $\{a, b, c, d\}$ seperti persamaan 1 Parameter $\{a, b, c, d\}$ (dengan $a < b < c < d$) menentukan kordinat x pada empat sudut dari fungsi keanggotaan trapesium.

2. Aplikasi fungsi implikasi (aturan) Logika pengambilan keputusan (*fuzzy inference*) mengamplikasikan aturan-aturan *fuzzy* pada masukkan *fuzzy*, kemudian mengevaluasi setiap aturan. Prinsip logika *fuzzy* digunakan untuk mengkombinasikan aturan-aturan jika-maka (*IF THEN*) yang terdapat dalam basis aturan suatu pemetaan dari suatu himpunan *fuzzy input* himpunan *fuzzy output*. Logika pengambilan keputusan merupakan langkah kedua dalam pemrosesan logika *fuzzy* (Wahab et al. 2017).
3. Komponen aturan langkah pertama pengambilan keputusan metode Mamdani adalah melakukan proses *fuzifikasi* Langkah kedua adalah melakukan proses terhadap kedua data *fuzzy* tersebut dengan operator *and* yang akan mengambil nilai paling minimal dari dua data tersebut. Langkah ketiga dengan impilasi *min* akan memotong fungsi keanggotaan keluaran setelah melalui operator *and* sehingga didapat daerah *fuzzy*. Ketiga proses tersebut juga diterapkan pada aturan-aturan *fuzzy* berikutnya. (K, Suyatno, and Rahayu 2011). Setelah aturan *fuzzy* dieksekusi, dilakukan proses komposisi dengan metode *max* yaitu solusi himpunan *fuzzy* diperoleh dengan cara mengambil nilai aturan maksimum, kemudian menggunakannya untuk memodifikasi daerah *fuzzy*, dan mengaplikasikan ke *output* dengan menggunakan operator *or* (*union*). Jika proposisi telah dievaluasi, maka *output* akan berisi suatu himpunan *fuzzy* yang merefleksikan konstribusi dari tiap-tiap posisi. Setelah proses implikasi dan komposisi telah dilakukan makan proses selanjutnya adalah proses *defuzzyfikasi*.
4. Penegasan (*defuzzyfikasi*). *Input* dari proses *defuzzyfikasi* adalah suatu himpunan *fuzzy* yang diperoleh dari komposisi aturan-aturan *fuzzy*, sedangkan *output* yang dihasilkan merupakan suatu bilangan pada domain himpunan *fuzzy* tersebut (Suryawati 2013). Jika diberikan suatu himpunan

fuzzy dalam *ranges* tertentu, maka dapat diambil suatu nilai *crisp* tertentu sebagai *output*. *Defuzzyfikasi* pada komposisi aturan Mamdani. Diantaranya yaitu metode COA, bisektor, MOM, LOM, dan SOM

2.2. Tinjauan Studi

Studi literatur mengenai pembahasan parameter *entropy* CK *metrics* pada properti kualitas desain perangkat lunak dan kerangka kerja sebuah perangkat lunak, sebagai bagian dari evaluasi kualitas desain yaitu:

1. Penelitian yang pernah dilakukan oleh (Turnu, Marchesi, and Tonelli 2012) entropi beberapa metrik CK untuk menilai kualitas perangkat lunak berorientasi objek. Menganalisis berbagai rilis dari sistem perangkat lunak *eclipse* dan *netbeans* mempresentasikan analisis empiris dari entropi dalam berbagai versi dari dua sistem perangkat lunak besar. Dengan demikian, penyelidikan penuh atas kemungkinan menggunakan metrik untuk menggambarkan kompleksitas sistem perangkat lunak secara keseluruhan yang biasa digunakan dalam rekayasa perangkat lunak sebagai sistem patokan untuk analisis empiris terhadap kualitas perangkat lunak, hasil dari penelitian menggambarkan kualitas perangkat lunak dalam hal degradasi kode dan mengurangi rawatan selama waktu pembangunan. Singkatnya, hasil menunjukkan bahwa entropi ini bisa sangat aktif dalam mewakili degradasi perangkat lunak.
2. Penelitian selanjutnya dilakukan oleh (Qureshi 2015). Evaluasi metrik desain untuk mengurangi jumlah cacat dalam pengembangan perangkat lunak. Mengevaluasi metrik CK untuk mengusulkan CK ditingkatkan untuk mengurangi cacat pada tahap desain perangkat lunak dan membahas mengenai efek yang signifikan dari setiap metrik desain CK ada jumlah cacat per modul atau tidak. Hasil yang diberikan dari penelitian yaitu analisis regresi menunjukkan bahwa semua variabel independen CBO, DIT, NOC, WMC, dan RFC. Memiliki efek signifikan pada jumlah cacat kecuali LCOM. LCOM adalah satu-satunya atribut yang memiliki efek signifikan pada jumlah cacat.

3. Penelitian yang dilakukan oleh (Bianchi et al. 2001), mengevaluasi degradasi perangkat lunak melalui entropi, tujuan penelitian ini memvalidasi sebagian kelas entropi metrik direplikasi pada rilis berturut-turut dari serangkaian sistem perangkat lunak. Validitas ditunjukkan melalui pengukuran langsung kualitas perangkat lunak seperti jumlah cacat yang terdeteksi, upaya pemeliharaan, dan jumlah cacat yang terdapat pada perangkat lunak. Hasil dari penelitian ini yaitu menyimpulkan bahwa metrik entropi, pada tingkat granularitas tertentu, konsisten dengan ukuran langsung degradasi. Ini berarti bahwa jika setelah perubahan entropi tumbuh maka kita dapat mengatakan bahwa degradasi perangkat lunak (diukur secara langsung) Akan meningkat sesuai dengan keadaan yang sesungguhnya.
4. Penelitian selanjutnya (Halim, Xandra, and Sim 2015), mengukur tingkat *reusability* dan *efficiency* dari kode program dengan Pendekatan *fuzzy logic*, penelitian ini membahas faktor kualitas *eksternal* berupa *reusability* dan *efficiency* diukur melalui pendekatan *fuzzy logic* Mamdani. Masukan dari *fuzzy logic* diperoleh dari hasil perhitungan CK *metrics* dari kode program berorientasi objek. Dipengaruhi oleh metrik WMC, LCOM, CBO, DIT dan NOC. Faktor kualitas eksternal *efficiency* dipengaruhi oleh metrik LCOM, CBO, DIT dan NOC. Hasil dari penelitian ini yaitu aplikasi ini dapat memberikan informasi yang jelas mengenai kualitas dari *class* pada perangkat lunak berorientasi objek. Hasilnya berupa nilai kuantitatif dan kualitatif, sehingga dapat dengan mudah digunakan untuk melakukan perbandingan kualitas antara dua kode program yang memiliki fungsi yang sama dan memberikan panduan untuk pengembangan perangkat lunak versi berikutnya.
5. Penelitian selanjutnya dilakukan oleh (Babich, Clarke, and Power 2011), menggunakan teknik abstraksi kelas untuk memprediksi kesalahan pada kelas OO, penelitian ini membahas mengusulkan metrik *inovatif suite* didasarkan pada teknik abstraksi kelas yang menggunakan taksonomi untuk kelas OO (CAT) untuk mengukur sifat kuantitatif kelas OO dalam sistem perangkat lunak menggunakan kombinasi dari karakteristik kelas, dan

melakukan analisis statistik metrik kelas CK banyak digunakan dan metrik CAT menggunakan enam versi dari industri-kekuatan sistem *open source eclipse*. Hasil studi kasus ini menunjukkan bahwa kedua CK dan CAT *metrics* efektif dalam mengembangkan model statistik yang berkualitas untuk beberapa kesalahan dalam memprediksi, rilis berurutan dari sistem perangkat lunak *object oriented* ditulis di java.

Tabel 2.3. Penelitian Terkait

Penelitian	Masalah	Metode	Hasil
Bianchi et al. (2001)	Memvalidasi sebagian kelas entropi metrik direplikasi pada rilis berturut-turut	Model MDD pada lima tingkat granularitas	Menyimpulkan bahwa metrik entropi, pada tingkat granularitas tertentu, konsisten dengan ukuran langsung degradasi
Halim, Xandra, dan Sim (2015)	Mengukur Tingkat <i>Reusability</i> dan <i>Efficiency</i> dari kode program	Pendekatan <i>Fuzzy Logic</i>	Menghasilkan berupa aplikasi yang dapat mengukur perbandingan kualitas antara dua kode program
Babich, Clarke, dan Power (2011)	Mengukur sifat sistem perangkat lunak	Regresi logistik biner univariat (UBLR).	Menyimpulkan bahwa kedua CK metrik <i>efektif</i> dalam model statistik yang berkualitas untuk memprediksi kesalahan perangkat lunak
Babich, Clarke, dan Power (2011)	Mengevaluasi metrik CK pada tahap desain perangkat lunak	Analisis regresi, dengan penentuan nilai max,min	Menghasilkan variabel independen CBO, DIT, NOC, WMC, RFC, berpengaruh signifikan terhadap perangkat lunak

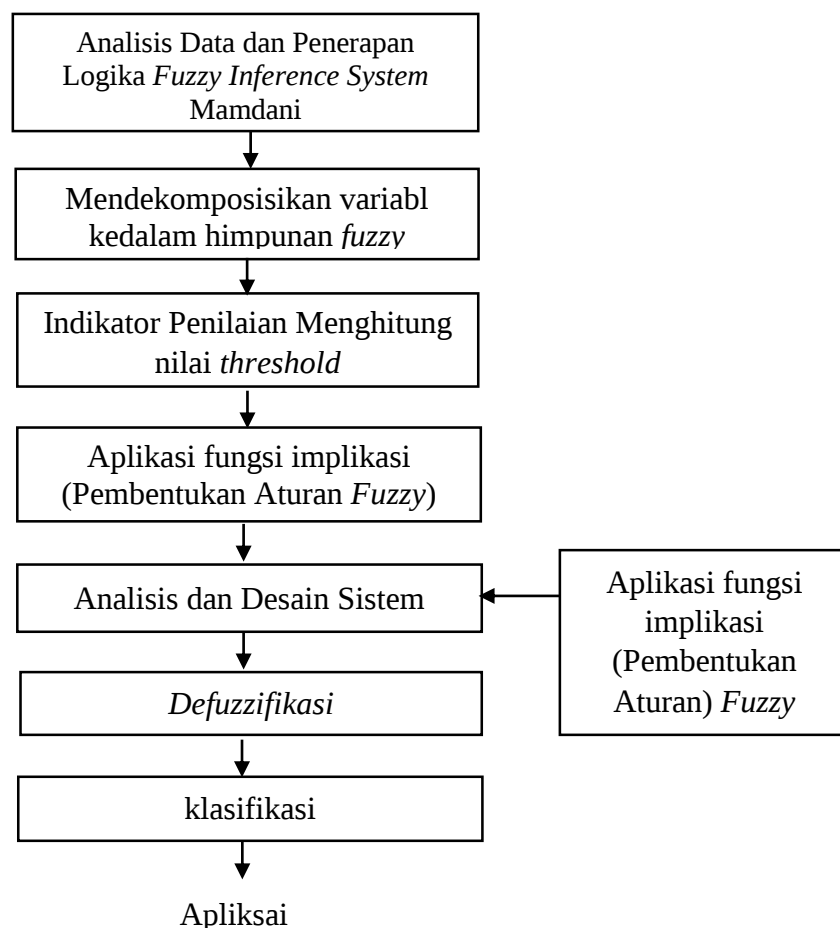
Penelitian	Masalah	Metode	Hasil
Turnu, Marchesi, dan Tonelli (2012)	Menganalisis berbagai <i>rilis</i> dari sistem perangkat lunak, mempresentasikan analisis regresi dari dari dua sistem perangkat lunak <i>eclipse</i> dan <i>Netbeans</i> .	Analisis regresi	Mengurangi rawatan selama waktu pembangunan. hasil menunjukkan bahwa entropi bisa sangat aktif dalsam mewakili degradasi perangkat lunak

BAB III

METODE PENELITIAN

3.1. Metode Penelitian

Untuk mendapatkan informasi mengenai kualitas perangkat lunak penulis melakukan pengujian data *metrics* dari kode program berorientasi objek dengan melakukan pendekatan *fuzzy inference system* Mamdani, sehingga dapat diketahui tingkat kualitas dari sebuah perangkat lunak, adapun set data yang digunakan adalah *entropy eclipse equinox* sebagai sumber aplikasi *open source*. kerangka penelitian digambarkan dalam Gambar 3.1. Sebagai dasar penelitian, yang dilakukan.



Gambar 3.1. Kerangka Pemikiran Penentuan

1. *Dataset* yang digunakan *eclipse equinox*
2. Parameter properti desain, sebagai properti yang akan dijadikan bahan penelitian yang dapat mewakili pengukuran tingkat kualitas dari sebuah perangkat lunak.
3. Indikator Penilaian Merupakan indikator yang diperlukan untuk menilai enam *entropy CK metrics* dan penentuan nilai *max, min median*
4. Mendekomposisikan enam variabel ke dalam himpunan *fuzzy* untuk uji kualitas perangkat lunak diperlukan pengujian data untuk menentukan *reusability* dan kualitas dari enam properti desain sebagai penentu perangkat lunak dapat digunakan kembali dan dikembangkan dengan sistem yang lebih kompleks,
5. Penialain *fuzzy* menentukan nilai *reusability* dan kualitas dari set data *entropy* dan berdasarkan *fuzzy inference system* Mamdani kemudian diaplikasikan menggunakan *Toolbox* MATLAB R2011b.
6. Membentuk aturan *fuzzy*, sehingga terbentuk aturan yang akan digunakan.
7. *Defuzzifikasi* mendapatkan hasil dari pengujian data
8. Klasifikasi hasil uji coba sistem pendukung keputusan mengukur *bug* yang terdapat pada set data *entropy metrics* yang dibagi menjadi empat kategori dengan *fuzzy inference system*. *Trivial Bugs*, *Normal bugs*, *Critical Bugs*, *High Priority Bugs*, dari empat kategori tersebut akan didapatkan hasil apakah memiliki tingkat *reusability* dan kualitas yang baik atau sebaliknya hasil dari proses ini diharapkan memberikan kontribusi yang baik dalam proses pengembangan dan pembangunan perangkat lunak.

3.2. Metode Pemilihan Sampel

Penulis mengambil sampel set data dari situs *website repository* <http://bug.inf.usi.ch> pengujian aplikasi berbasis java, set data yang digunakan adalah *entropy eclipse equinox* sebagai implementasi spesifikasi kerangka kerja OSGi, satu paket yang mengimplementasikan berbagai layanan OSGi opsional dan infrastruktur lainnya untuk menjalankan sistem berbasis OSGi. penulis mengambil 324 record data dari satu *entropy CK metrics*.

3.3. Metode Pengumpulan Data

Untuk mengumpulkan data serta informasi yang diperlukan dalam penelitian menggunakan metode sebagai berikut :

1. Data primer

Diperoleh melalui *repository* <http://bug.inf.usi.ch>, sebagai set data yang digunakan.

2. Data sekunder

Diperoleh melalui beberapa referensi seperti buku, *literatur*, internet dan jurnal sehingga didapatkan informasi yang akurat yang berkaitan dengan penelitian yang terkait.

3.4. Teknik Analisis Data

Teknik analisis data merupakan cara menganalisa data penelitian untuk menjawab permasalahan yang ada dan menguji hipotesa. Martin Fowler (2004:28). Pada penelitian ini proses penilaian kualitas rekayasa perangkat lunak dari data *entropy eclipse equinox*, terdapat enam paramater pengukur properti disain yaitu: WMC (*Weight Methods per Class*), RFC (*Response sets For a Class*), LCOM (*Lack of Cohesion of Methods*), CBO (*Coupling Between Classes*), DIT (*Dept of Inheritance*) dan NOC (*Number Of Children*). Sebagai proses analisis kualitas *metrics*, pengujian dilakukan dengan menggunakan pendekatan *fuzzy inference system* Mamdani.

1. Instrumen pengujian

Instrumen pengujian berupa set data yang akan digunakan untuk melakukan pengukuran dengan tujuan menghasilkan data kuantitatif yang akurat. Dalam penelitian ini pengujian akan dilakukan terhadap *Entropy CK metrics*, memiliki enam paramater pengukuran properti desain, yaitu: WMC, LCOM, CBO, DIT, RFC, dan NOC. Ada beberapa referensi yang digunakan pada penelitian dalam memilih nilai *threshold CK metrics*. Dapat dilihat pada Tabel 3.1. Menunjukkan nilai ambang batas yang digunakan oleh (Goyal, Chandra, dan Singh 2013). Sementara nilai ambang batas penulis gunakan Dapat dilihat pada Tabel 3.2. Diambil dari set data yang ada.

Tabel 3.1. Threshold CK Versi
(Goyal, Chandra, and Singh 2013)

CK Metrics	Threshold
WMC	0 - 15
RFC	0- 35
LOCM	0 - 1
COB	0- 8
DIT	0 - 6
NOC	0 - 6

Tabel 3.2. Threshold CK Penulis

CK Metrics	Threshold
WMC	0 – 16.45
RFC	0 - 18.07
LCOM	0 - 6.72
CBO	0 - 13
DIT	0 - 0. 43
NOC	0 - 1.63

Tabel 3.3. Hubungan Paramater CK metrics dengan Faktor Kualitas Perangkat lunak

CK Metrics	Faktor Kualitas
WMC	<i>Maintainability, Understandability, Usability, Reusability</i>
RFC	<i>Understandability, Usability, Testability</i>
LOCM	<i>Reusability, Efficiency</i>
COB	<i>Reusability, Efficiency</i>
DIT	<i>Reusability, Efficiency, Understandability, Testability</i>
NOC	<i>Reusability, Efficiency, Testtability</i>

Hubungan paramater CK metrics dengan faktor kualitas perangkat lunak yang akan diukur dijabarkan menjadi indikator. Kemudian indikator tersebut

dijadikan sebagai titik tolak untuk menyusun instrumen-instrumen berupa pernyataan.

3.2. Logika *Fuzzy Inference System* Mamdani

Dalam Teknik pendekatan *fuzzy inference system* Mamdani meliputi beberapa tahapan, dua tahapan pertama merupakan tahap pengumpulan data dan dua tahapan selanjutnya merupakan tahap analisis data (Utara dan Utara 2018). Adapun tahap pengumpulan data terdiri dari model *reusability* masukkan dan fungsi keanggotaan serta tahapan untuk analisis data yang terdiri dari aplikasi fungsi implikasi, komposisi aturan dan *defuzzyfikasi*.

Pada proses menganalisis data, peneliti menggunakan bantuan perangkat lunak MATLAB 2018b, sebagai sistem pengukuran metrik pada entropi perangkat lunak, tahapan ini menitik beratkan pada analisis data yang menghasilkan empat kategori *bug*, sehingga akan diketahui kualitas dan *reusability* perangkat lunak.

3.3. *Fuzzy Inference System*

1. Model *Reusability*

Fuzzy Inference System adalah pembuatan formulasi proses pemetaan *input* ke bagian *output* dengan menggunakan logika *fuzzy*. Tahapan utamanya meliputi pendefinisian fungsi keanggotaan *input* dan *output* serta pembuatan basis pengetahuan dengan menggunakan aturan (*rules*) *fuzzy if-then-else*. Berbagai kriteria inilah yang disebut sebagai variabel masukkan *fuzzy* (Bhatia dan Mann 2015). Variabel masukkan dalam penentuan *reusability* dan kualitas perangkat lunak

2. Pendefinisian Fungsi Keanggotaan *Input* dan *Output*

Variabel *fuzzy* dimodelkan adalah enam paramater *metrics* sebagai enam variabel *input* sistem dan paramater *reusability* dan variabel *output* sistem. Setiap paramater *metrics* memiliki tiga himpunan *fuzzy*, yaitu *low*, *medium* dan *high* (Borys 2014). Begitu pula dengan paramater *reusability* memiliki empat himpunan yaitu *trivial bugs*, *normal bugs*, *critical bugs*, *high priority bugs*. Masing-masing himpunan dipetakan terhadap derajat keanggotaan dari 0 sampai dengan 1. Tipe

dari fungsi keanggotaan untuk enam variabel *input metrics* adalah fungsi trapesium dengan sintaks seperti pada persamaan 1 (Mago n.d.).

Kurva trapesium (Mago n.d.). Tersebut adalah fungsi dari sebuah vektor, x yang tergantung pada empat paramater sakalar a , b , c dan d , sebagaimana diperlihatkan oleh persamaan (2), (3), (4) dan (5). Nilai a , b , c dan d disesuaikan dengan batasan himpunan keanggotaan yang dimiliki oleh setiap paramater *input metrics*. Kurva trapesium pada dasarnya menyerupai bentuk segitiga, hanya saja ada beberapa titik yang memiliki nilai keanggotaan satu. Representasi fungsi keanggotaan untuk kurva trapesium adalah sebagai parameter $\{a, b, c, d\}$ (dengan $a < b < c < d$) menentukan koordinat x pada empat sudut dari fungsi keanggotaan trapesium.

$$\text{Trapesium } (x, a, b, c, d) \left\{ \begin{array}{ll} a; & ; \quad x \geq a \\ (x - a) / (b - a) & ; \quad a \leq x \leq b \\ 1; & ; \quad b \leq x \leq c \\ (d - x) / (d - c) & ; \quad c \leq x \leq d \\ 0; & ; \quad x \geq d \end{array} \right. \quad (3.1)$$

Untuk melengkapi fungsi trapesium tersebut, segera ditentukan nilai paramater a , b , c dan d untuk setiap variabel *input metrics*, yang diperlihatkan oleh Tabel. 3.4. Tabel. 3.5. Tabel. 3.6, Tabel. 3.7. Tabel. 3.8. Tabel. 3.9, Sementara itu bentuk kurva keanggotaan untuk setiap variabel *input CK metrics* diperlihatkan Gambar 3.1. Gambar 3.2. Gambar 3.3. Gambar 3.4. dan Gambar 3.5.

Tabel 3.4. Weight Methods per Class

WMC	a	b	c	d
<i>low</i>	0	2	4.6	7
<i>medium</i>	4.6	7	9.6	12
<i>high</i>	9.6	12	14.5	16.5

Tabel 3.5. Coupling Between Classes

COB	a	b	c	d
<i>low</i>	0.1	2.3	4.5	6.5
<i>medium</i>	4.5	6.5	9	11
<i>high</i>	9	11	13.5	15

Tabel 3.6. Depth of Inheritance

DIT	a	b	c	d
<i>low</i>	0	0.7	1.8	2.5
<i>medium</i>	1.8	2.5	3.8	4.5
<i>high</i>	3.8	4.5	5.8	6.9

Tabel 3.7. Number of Children

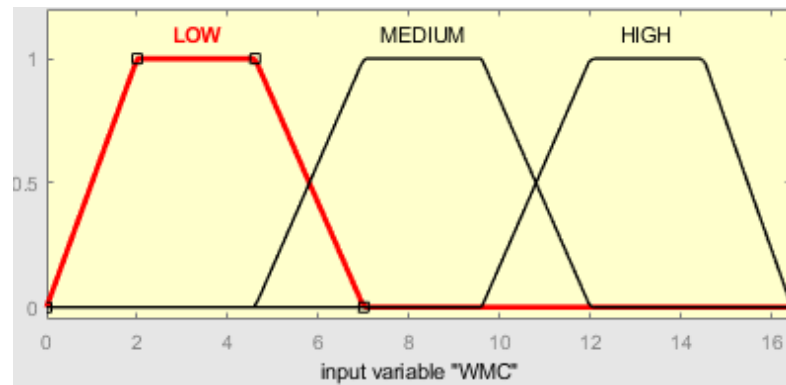
NOC	a	b	c	d
<i>low</i>	0	0.25	0.56	0.8
<i>medium</i>	0.56	0.8	1.16	1.39
<i>high</i>	1.16	1.4	1.75	2

Tabel 3.8. Lack of Cohesion of Methods

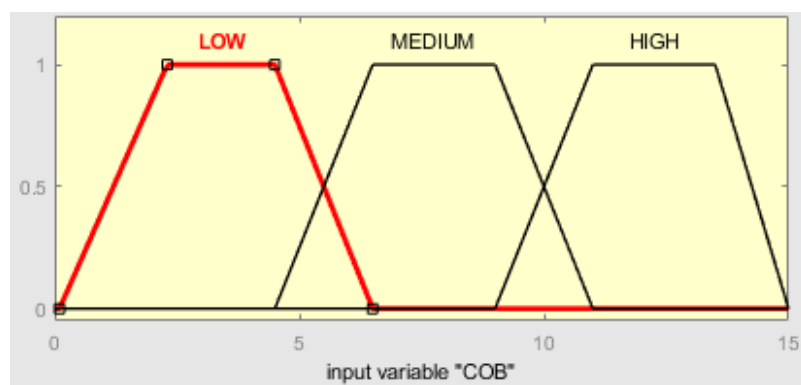
LOCM	a	b	c	d
<i>low</i>	0	2	4.8	7
<i>medium</i>	4.8	7	9.8	12
<i>high</i>	9.8	12	14.5	16.45

Tabel 3.9. Response sets For a Class

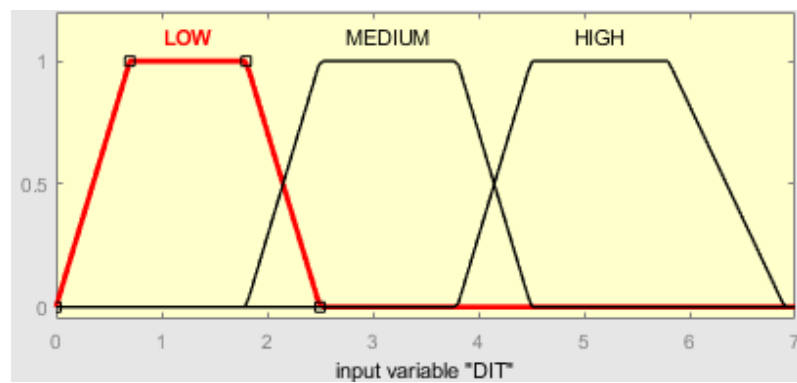
RFC	a	b	c	d
<i>low</i>	0	2.3	5.2	7.8
<i>medium</i>	5.2	7.8	11	13
<i>high</i>	11	13	16	18.07



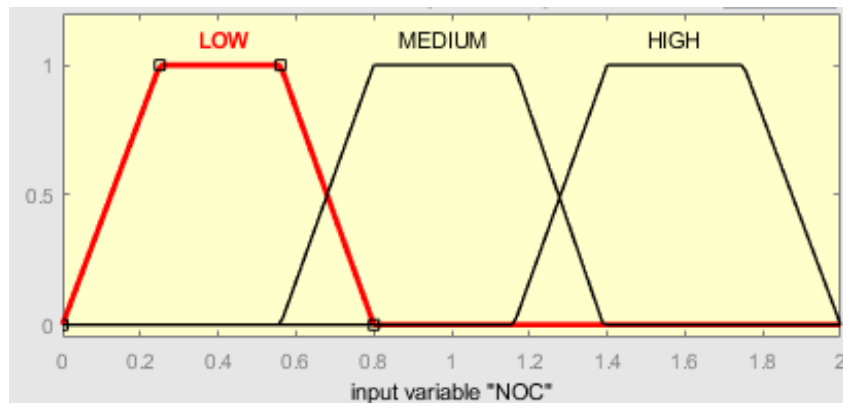
Gambar 3.2. Weight Methods per Class



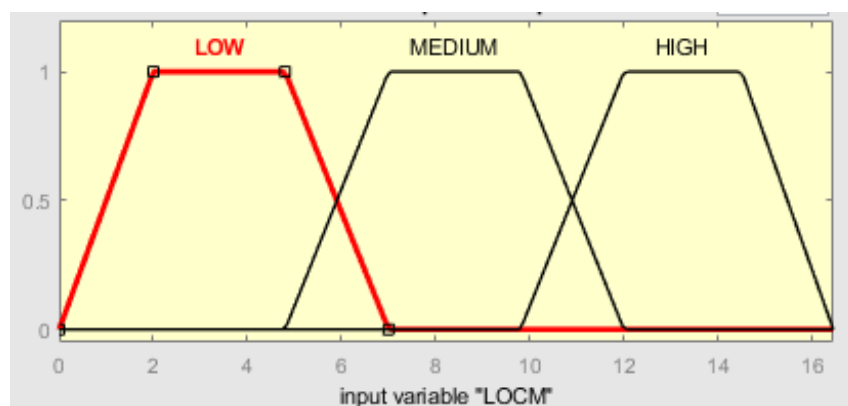
Gambar 3.3. Coupling Between Classes



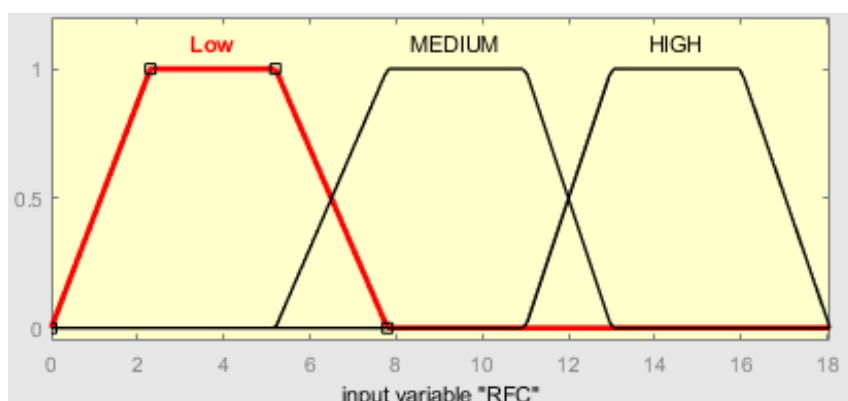
Gambar 3.4. Depth of Inheritance



Gambar 3.5. Number of Children



Gambar 3.6. Lack of Cohesion of Methods



Gambar 3.7. Response sets For a Class

3. Aplikasi Fungsi Implikasi

Setelah pembentukan himpunan *fuzzy*, maka dilakukan pembentukan aturan *fuzzy* atau *rule*. Aturan-aturan dibentuk untuk menyatakan relasi antara *input* dan *output*. Tiap aturan merupakan suatu implikasi. Operator yang digunakan untuk

menghubungkan antara beberapa *input* adalah operator *AND* dan yang memetakan antara *input-output* adalah *IF-THEN*.

Aturan *fuzzy* dalam bentuk *if then* menggunakan model Mamdani dapat ditulis sebagai berikut :

$$IF (x_1 \text{ is } A_1) AND (x_2 \text{ is } A_2) \dots \dots \dots AND (x_n \text{ is } A_n) \dots \dots \dots (3.2)$$

Keterangan :

- x_1 is A_1 disebut *anteseden*
- y is B disebut *konsekuen*.
- x dan y adalah skalar
- A dan B adalah variabel linguistik

Setelah aturan dibentuk, maka dilakukan aplikasi fungsi *implikasi*. Fungsi *implikasi* yang digunakan adalah *MIN*. Fungsi *implikasi MIN* berarti tingkat keanggotaan yang didapat sebagai konsekuen dari proses ini adalah nilai minimum dari variabel *input* untuk mendapatkan daerah *fuzzy* pada variabel hasil.

4. Komposisi Aturan

Pada saat menghitung komposisi aturan fungsi *implikasi* menggunakan fungsi *MAX* yaitu dengan cara mengambil nilai maksimum dari *output* aturan kemudian menggabungkan daerah *fuzzy* masing-masing aturan dengan operator *AND*. Secara umum dapat dituliskan:

$$\mu_{sf}[x_i] \leftarrow \max(\mu_{sf}[x_i], \mu_{kf}[x_i]) \dots \dots \dots (3.3)$$

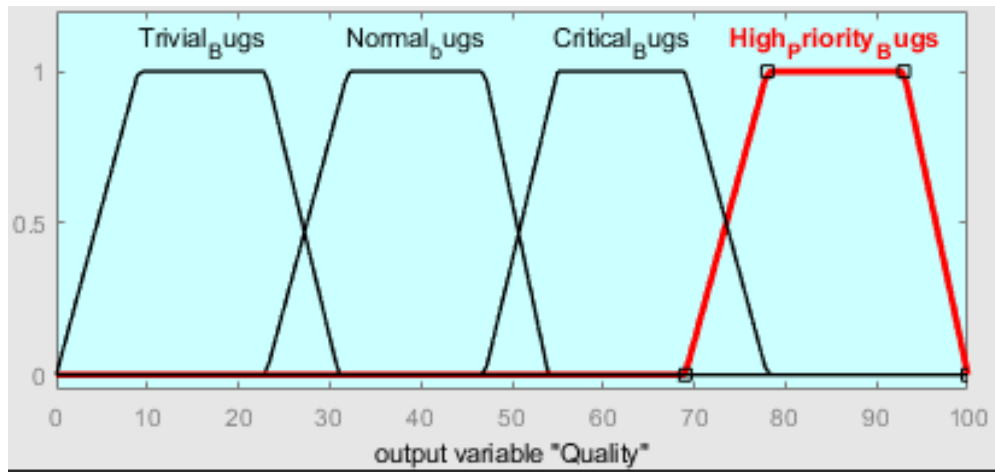
Keterangan :

$\mu_{sf}[x_i]$ = nilai fungsi keanggotaan solusi *fuzzy* sampai aturan ke- i ;

$\mu_{kf}[x_i]$ = nilai fungsi keanggotaan konsekuen *fuzzy* aturan ke- i ;

Keenam paramater *input CK metrics* memiliki kecenderungan semakin memiliki nilai rendah (*low*) maka semakin baik kualitas. Semakin tinggi nilai *input* maka semakin buruk kualitas. Sehingga dalam aturan ini diberikan pembobotan tertinggi adalah saat kondisi *low* dan terendah saat kondisi *high*. Urutan pembobotan adalah *low* simpulan yang bersesuaian. Terdapat enam *input metrics*, di mana setiap *input* memiliki empat kemungkinan kondisi yaitu *Trivial Bugs*, *Normal bugs*, *Critical Bugs*, *High Priority Bugs*. Contoh aturan pertama adalah. *If (WMC is LOW) and (COB is LOW) and (DIT is LOW) and (NOC is LOW) and (LOCM is LOW) and*

(RFC is Low) then (Quality is Trivial_Bugs) berikut contoh terdapat pada gambar 3.7 response sets For a Clas, di bawah ini.



Gambar 3.8. Response Output

3. Proses Defuzzyfikasi

Proses *defuzzyfikasi* adalah mengubah *fuzzy output* menjadi nilai tegas berdasarkan fungsi keanggotaan yang telah ditentukan. simulasi dan analisa melalui perangkat lunak Matlab, maka aturan yang telah ditetapkan, selanjutnya melewati tahapan *defuzzyfikasi* dengan teknik *centroid*. Hasil *defuzzyfikasi* menampilkan kombinasi *input output* dalam kondisi awal. Nilai-nilai *input* yang ditampilkan mengambil nilai tengahnya masing-masing dan nilai *output* nya menunjukkan kualitas dengan kategori yang sesuai aturan dalam menghitung *defuzzyfikasi* digunakan metode *centroid method* dengan rumus sebagai berikut:

$$j^* = \frac{\sum_{j=1}^n Z_j \mu(Z_j)}{\sum_{j=1}^n \mu(Z_j)} \quad (3.4)$$

Keterangan

- $\pi(z_j)$: Minimum derajat keanggotaan di fungsi implikasi
- z_j : Nilai maksimum dari *output* aturan
- Z^* : Hasil nilai *defuzzyfikasi* atau hasil akhir

- a. Metode *centre of area* (COA) pada metode ini, solusi *crisp* diperoleh dengan cara mengambil titik pusat daerah *fuzzy*, secara umum dirumuskan pada persamaan 2 untuk variabel kontinu dan persamaan 3 untuk variabel diskrit. Secara khusus misalkan y_p adalah pusat dari himpunan *fuzzy* ke- p dan h_p adalah tingginya, *center average defuzzier* menentukan y^* sebagai.

$$y^* = \frac{\sum_{p=1}^M y_p h_p}{\sum_{p=1}^M h_p} \quad (3.5)$$

- b. Metode *Bisektor*. Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil nilai pada domain *fuzzy* yang memiliki nilai keanggotaan sebagian dari jumlah total nilai keanggotaan pada daerah *fuzzy*. Secara umum dituliskan pada rumus 3.6. Dibawah ini :

$$\int_{\alpha}^{z^{BOA}} \mu(z) dz = \int_{z^{BOA}}^{\beta} \mu(z) dz \quad (3.6)$$

Dimana:

$$\alpha = \min \{z \mid z \in Z\}$$

$$\beta = \max \{z \mid z \in Z\}$$

- c. Metode Mom (*Mean Of Maximum*) pada solusi ini, solusi *crisp* diperoleh dengan cara mengambil nilai rata-rata domain yang memiliki nilai keanggotaan maksimum.
- d. Metode Lom (*Largest Of Maximum*) pada metode ini, solusi *crisp* diperoleh dengan cara mengambil nilai terbesar dari domain yang memiliki nilai keanggotaan maksimum.
- e. Metode Som (*Smallest Of Maximum*) pada metode ini, solusi *crisp* diperoleh dengan cara mengambil nilai terkecil dari domain yang memiliki nilai keanggotaan maksimum.

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1. Analisis Data dan Penerapan Logika *Fuzzy Inference System* Mamdani

Tahap analisa data menggunakan logika *fuzzy inference system* model mamdani secara detail. Berikut set data yang akan dijadikan bahan penelitian yang terdapat pada table 4.1. Data ini diambil dari repository <http://bug.inf.usi.ch> dari proyek *eclipse equinox* bagian dari *sub system* terapat 324 record *cllas* data.

Tabel 4.1. Set Data CK Eclipse Equinox

Class Name	cbo	dit	lcom	noc	rfc	wmc
ext::framework::a::importer::Activator	0	0	0	0	0	0
org::eclipse::osgi::framework::internal::reliablefile::ReliableFile	3.82 339	0	0.43 097	0	1.01 202	0.49 773
org::osgi::framework::ServiceEvent	0.51 822	0	0	0	0	0
org::eclipse::osgi::framework::internal::core::BundleProtectionDomainImpl	0	0	0.48 097	0	0.53 315	0.52 122
substitutes::z::Fz	0	0	0	0	0	0
circularity::test::Activator	0	0	0	0	0	0
org::eclipse::osgi::framework::internal::core::ConditionalPermissionSet	1.17 078	0	0.85 533	0	2.19 094	2.65 919
org::eclipse::osgi::internal::module::BundleConstraint	1.06 96	0.4 306	0.85 807	0	1.00 411	1.04 214
org::eclipse::osgi::internal::resolver::SystemState	0.39 853	0	1.33 692	0	1.75 667	1.78 779

4.1.1. Pembentukan Himpunan *Fuzzy* (*Fuzzyfikasi*)

Kriteria dan indikator yang peneliti tulis di dalam bab tiga akan dianalisis dan dijadikan variabel dalam melakukan proses penentuan kualitas perangkat lunak yang dinilai dari enam *metrics object oriented*. Indikator yang digunakan terdiri dari:

Berdasarkan sampel penilaian kualitas perangkat lunak dari enam properti desain (WMC, LCOM, CBO, DIT, RFC dan NOC) akan menitik beratkan pada penentuan kualitas dan *reusability* dengan penentuan tingkat *bug* pada set data *entropy eclipse equinox* data yang dijadikan bahan penelitian yaitu : *Trivial Bugs*,

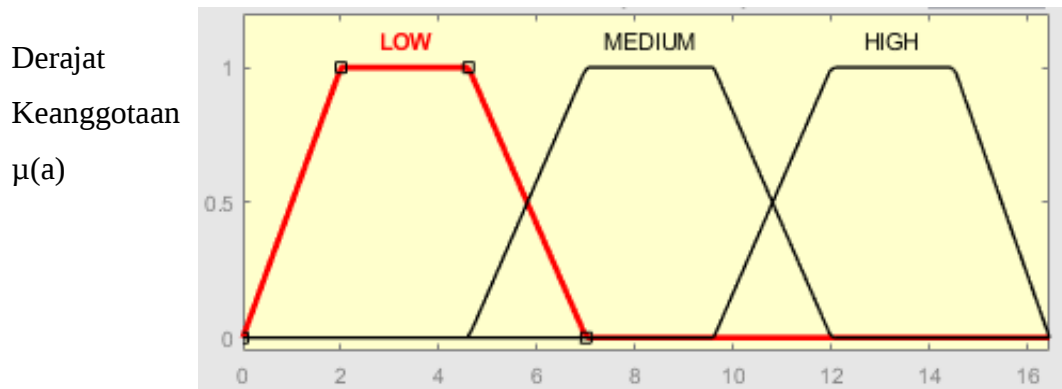
Normal Bugs, Critical Bugs, High Priority Bugs. Instrumen pengujian berupa set data yang akan digunakan untuk melakukan pengukuran dengan tujuan menghasilkan data kuantitatif yang akurat. Dalam penelitian ini pengujian akan dilakukan terhadap *Entropy CK metrics*, memiliki enam paramater pengukur properti disain, *WMC (Weight Methods per Class)*, *RFC (Response sets For a Class)*, *LCOM (Lack of Cohesion of Methods)*, *CBO (Coupling Between Classes)*, *DIT (Dept of Inheritance)* dan *NOC (Number Of Children)*.sesuai dengan *threshold CK metrics*. Pada Tabel 4.1. dapat dilihat tentang nilai *Threshold*.

Tabel 4.2. Threshold CK Penulis

CK Metrics	Threshold
WMC	0 – 16.45
RFC	0 - 18.07
LCOM	0 - 6.72
CBO	0 - 13
DIT	0 - 0.43
NOC	0 - 1.63

1. Indikator WMC - *Weighted Methods for Class*

Pada indikator *Weighted Methods for Class* didefinisikan tiga himpunan fuzzy, yaitu *Low*, *Medium*, dan *High*. Himpunan fuzzy *Low* memiliki domain $[0, 7]$ dimana derajat keanggotaan *Low* tertinggi ($=1$) terletak pada angka 0-2. Himpunan fuzzy *Medium* memiliki domain $[4.6, 12]$ dimana derajat keanggotaan ($=1$) terletak pada nilai 7. Himpunan fuzzy *High* akan memiliki domain $[9.6, 16.45]$ dimana derajat keanggotaan *High* tertinggi ($=1$) terletak pada angka ≥ 12 . Untuk mempersentasikan indikator kualitas perangkat lunak digunakan bentuk trapesium representasi dengan grafik dapat dilihat pada Gambar 4.2.



Sumber : Hasil peneltiana (2019)

Gambar 4.1. Grafik fungsi keanggotaan *Weighted Methods for Class*

Ekspresi untuk fungsi keanggotaan *fuzzy weighted methods for class* sebagai berikut :

$$\mu_{\text{Low}}[x] = \begin{cases} 0; & ; \quad x \geq 7 \\ (x - 0) / (2 - 0) & ; \quad 0 \leq x \leq 2 \\ 1; & ; \quad 2 \leq x \leq 4.6 \\ (7 - x) / (7 - 4.6) & ; \quad 4.6 \leq x \leq 7 \\ 0; & ; \quad x \geq 7 \end{cases}$$

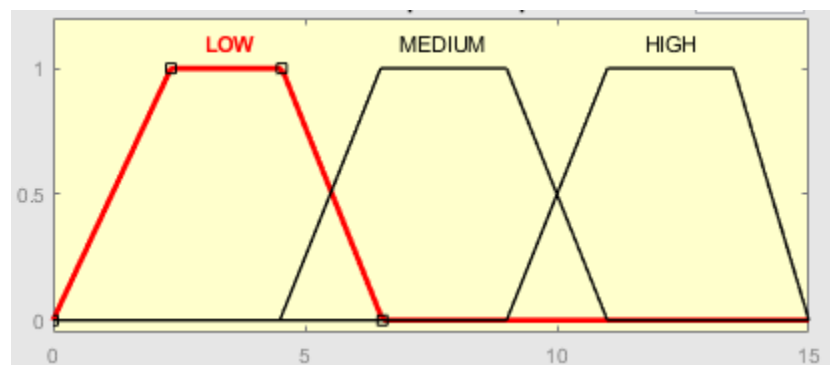
$$\mu_{\text{Medium}}[x] = \begin{cases} 0; & ; \quad x \geq 4.6 \\ (4.6 - x) / (4.6 - 7) & ; \quad 4.6 \leq x \leq 7 \\ 1; & ; \quad 7 \leq x \leq 9.6 \\ (12 - x) / (12 - 9.6) & ; \quad 9.6 \leq x \leq 12 \\ 0; & ; \quad x \geq 12 \end{cases}$$

$$\mu_{\text{High}}[x] = \begin{cases} 0; & ; \quad x \geq 9.6 \\ (9.6 - x) / (12 - 9.6) & ; \quad 9.6 \leq x \leq 12 \\ 1; & ; \quad 12 \leq x \leq 14.5 \\ (16.45 - x) / (16.45 - 14.45) & ; \quad 14.5 \leq x \leq 16.45 \\ 0; & ; \quad x \geq 16.45 \end{cases}$$

2. Indikator CBO - *Coupling between Objects*

Pada indikator *Coupling between Objects* dalam ucapan dan tindakan didefinisikan tiga himpunan *fuzzy*, yaitu *Low*, *Medium*, dan *High*. Himpunan *fuzzy Low* memiliki domain $[0, 6.5]$ dimana derajat keanggotaan *Low* tertinggi ($=1$) terletak pada angka 0-2.32 Himpunan *fuzzy Medium* memiliki domain $[4.5, 11]$ dimana derajat keanggotaan ($=1$) terletak pada nilai 6.5. Himpunan *fuzzy High* akan memiliki domain $[9,15]$ dimana derajat keanggotaan *High* tertinggi ($=1$) terletak pada angka ≥ 11 . Untuk mempersentasikan indikator kualitas komponen perangkat lunak digunakan bentuk trapesium representasi dengan grafik dapat dilihat pada Gambar 4.2.

Derajat
Keanggotaan
 $\mu(a)$



Sumber : Hasil peneltiana (2019)

Gambar 4.2. Grafik fungsi *Coupling between Objects*

Ekspresi untuk fungsi keanggotaan *fuzzy coupling between objects* sebagai berikut :

$$\mu_{\text{Low}}[x] = \begin{cases} 0; & ; \quad x \geq 0 \\ (x - 0) / (2.34 - 0) & ; \quad 0 \leq x \leq 2.34 \\ 1; & ; \quad 2.34 \leq x \leq 4.54 \\ (6.5 - x) / (6.5 - 4.54) & ; \quad 4.54 \leq x \leq 6.5 \\ 0; & ; \quad x \geq 6.5 \end{cases}$$

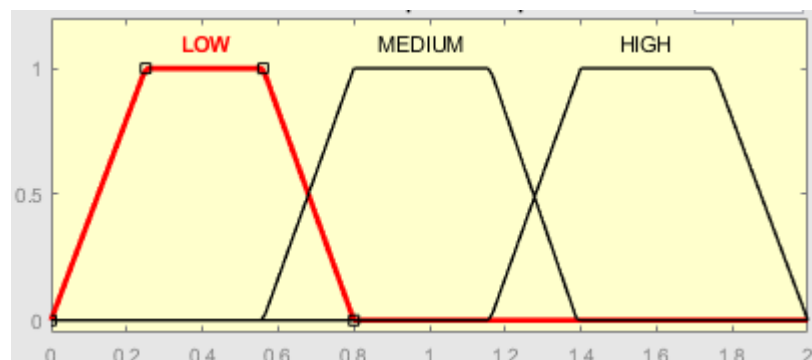
$$\mu_{\text{Medium}}[x] = \begin{cases} 0; & x \geq 4.5 \\ (x - 4.5) / (6.5 - 4.5) & ; 4.5 \leq x \leq 6.5 \\ 1; & ; 6.5 \leq x \leq 9 \\ (11 - x) / (11 - 9) & ; 9 \leq x \leq 11 \\ 0; & ; x \geq 11 \end{cases}$$

$$\mu_{\text{High}}[x] = \begin{cases} 0 & ; x \geq 9 \\ (x - 9) / (11 - 9) & ; 9 \leq x \leq 11 \\ 1 & ; 11 \leq x \leq 13.5 \\ (15 - x) / (15 - 13.5) & ; 13.5 \leq x \leq 15 \\ 0 & ; x \geq 15 \end{cases}$$

3. Indikator Number of Children

Pada indikator *coupling between objects* dalam ucapan dan tindakan didefinisikan tiga himpunan *fuzzy*, yaitu *Low*, *Medium*, dan *High*. Himpunan *fuzzy Low* memiliki domain [0, 8] dimana derajat keanggotaan *Low* tertinggi (=1) terletak pada angka 0 - 0.2. Himpunan *fuzzy Medium* memiliki domain [0.6, 1.4] dimana derajat keanggotaan (=1) terletak pada nilai 0.8. Himpunan *fuzzy High* akan memiliki domain [1.2, 2] dimana derajat keanggotaan *High* tertinggi (=1) terletak pada angka ≥ 1.4 . Untuk mempersentasikan indikator kualitas perangkat lunak digunakan bentuk trapesium representasi dengan grafik dapat dilihat pada Gambar 4.3.

Derajat
Keanggotaan
 $\mu(a)$



Sumber : Hasil penelitian (2019)

Gambar 4.3. Grafik fungsi *Number of Children*

Ekspresi untuk fungsi keanggotaan *fuzzy Number of Children* sebagai berikut :

$$\mu_{\text{Low}} [x] = \begin{cases} 0; & ; \quad x \geq 0 \\ (x - 0) / (2.34 - 0) & ; \quad 0 \leq x \leq 0.25 \\ 1; & ; \quad 0.25 \leq x \leq 0.56 \\ (0.8 - x) / (0.8 - 0.56) & ; \quad 0.56 \leq x \leq 0.8 \\ 0; & ; \quad x \geq 0.8 \end{cases}$$

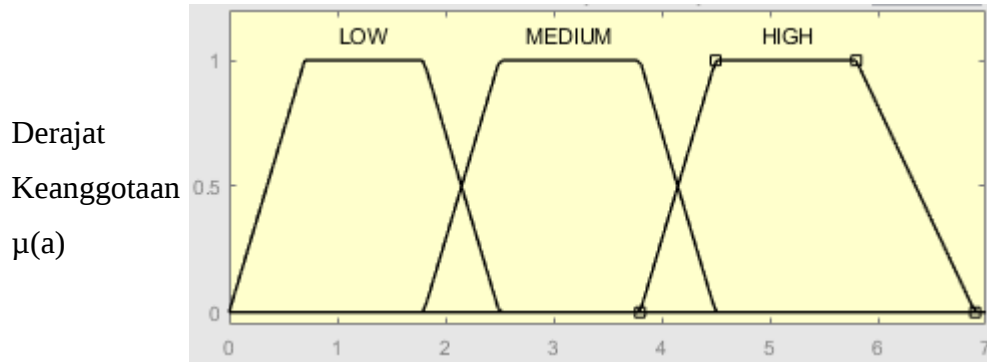
$$\mu_{\text{Medium}} [x] = \begin{cases} 0; & ; \quad x \geq 0.56 \\ (x - 4.5) / (6.5 - 4.5) & ; \quad 0.58 \leq x \leq 0.8 \\ 1; & ; \quad 0.8 \leq x \leq 1.16 \\ (1.39 - x) / (1.39 - 1.16) & ; \quad 1.16 \leq x \leq 1.39 \\ 0; & ; \quad x \geq 1.39 \end{cases}$$

$$\mu_{\text{High}} [x] = \begin{cases} 0 & ; \quad x \geq 1.16 \\ (x - 1.16) / (1.4 - 1.16) & ; \quad 1.16 \leq x \leq 1.4 \\ 1 & ; \quad 1.4 \leq x \leq 1.75 \\ (2 - x) / (2 - 1.75) & ; \quad 1.75 \leq x \leq 2 \\ 0 & ; \quad x \geq 2 \end{cases}$$

4. Indikator

Depth of Inheritance Tree

Pada indikator *depth of inheritance tree* dalam ucapan dan tindakan didefinisikan tiga himpunan *fuzzy*, yaitu *Low*, *Medium*, dan *High*. Himpunan *fuzzy Low* memiliki domain [0, 2.5] dimana derajat keanggotaan *Low* tertinggi (=1) terletak pada angka 0 - 0.7. Himpunan *fuzzy Medium* memiliki domain [1.8, 4.5] dimana derajat keanggotaan (=1) terletak pada nilai 2.5. Himpunan *fuzzy High* akan memiliki domain [3.8, 6.9] dimana derajat keanggotaan *High* tertinggi (=1) terletak pada angka ≥ 4.5 . Untuk mempersentasikan indikator kualitas perangkat lunak digunakan bentuk trapesium representasi dengan grafik dapat dilihat pada Gambar 4.4.



Sumber : Hasil penelitian (2019)

Gambar 4.4. Grafik fungsi *Depth of Inheritance Tree*

Ekspresi untuk fungsi keanggotaan *fuzzy depth of inheritance tree* sebagai berikut :

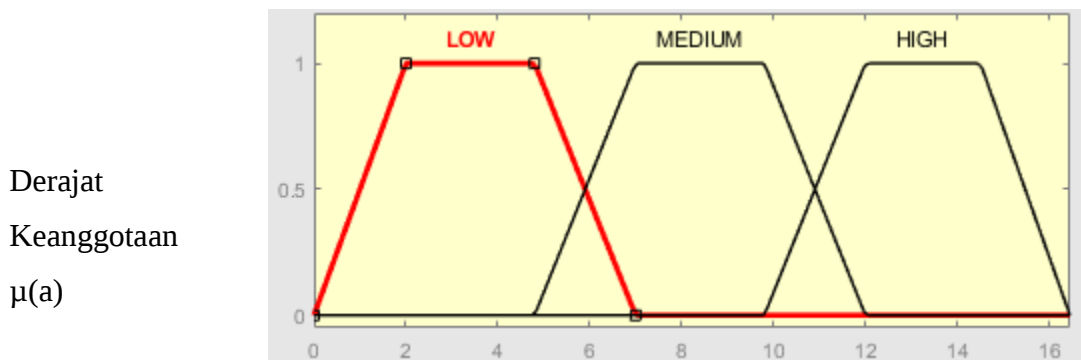
$$\mu_{\text{Low}}[x] = \begin{cases} 0; & ; \quad x \geq 0 \\ (x - 0) / (0.7 - 0) & ; \quad 0 \leq x \leq 0.7 \\ 1; & ; \quad 0.7 \leq x \leq 1.8 \\ (2.5 - x) / (2.5 - 1.8) & ; \quad 1.8 \leq x \leq 2.5 \\ 0; & ; \quad x \geq 2.5 \end{cases}$$

$$\mu_{\text{Medium}}[x] = \begin{cases} 0; & ; \quad x \geq 1.8 \\ (x - 1.8) / (2.5 - 1.8) & ; \quad 1.8 \leq x \leq 2.5 \\ 1; & ; \quad 2.5 \leq x \leq 3.8 \\ (4.5 - x) / (4.5 - 3.8) & ; \quad 3.8 \leq x \leq 4.5 \\ 0; & ; \quad x \geq 4.5 \end{cases}$$

$$\mu_{\text{High}}[x] = \begin{cases} 0 & ; \quad x \geq 3.8 \\ (x - 3.8) / (4.5 - 3.8) & ; \quad 3.8 \leq x \leq 4.5 \\ 1 & ; \quad 4.5 \leq x \leq 5.8 \\ (6.9 - x) / (6.9 - 5.8) & ; \quad 5.8 \leq x \leq 6.9 \\ 0 & ; \quad x \geq 6.9 \end{cases}$$

5. Indikator *Lack of Cohesion of Methods*

Pada indikator *lack of cohesion of methods tree* dalam ucapan dan tindakan didefinisikan tiga himpunan *fuzzy*, yaitu *Low*, *Medium*, dan *High*. Himpunan *fuzzy Low* memiliki domain $[0, 7]$ dimana derajat keanggotaan *Low* tertinggi ($=1$) terletak pada angka 0-2. Himpunan *fuzzy Medium* memiliki domain $[4.8, 12]$ dimana derajat keanggotaan ($=1$) terletak pada nilai 7. Himpunan *fuzzy High* akan memiliki domain $[9.8, 16.45]$ dimana derajat keanggotaan *High* tertinggi ($=1$) terletak pada angka ≥ 12 . Untuk mempersentasikan indikator kualitas perangkat lunak digunakan bentuk trapesium representasi dengan grafik dapat dilihat pada Gambar 4.5.



Sumber : Hasil penelitian (2019)

Gambar 4.5. Grafik fungsi *Lack of Cohesion of Methods*

Ekspresi untuk fungsi keanggotaan *fuzzy Lack of Cohesion of Methods* sebagai berikut :

$$\mu_{\text{Low}}[x] = \begin{cases} 0; & ; x \geq 7 \\ (x - 0) / (2 - 0) & ; 0 \leq x \leq 2 \\ 1; & ; 2 \leq x \leq 4.8 \\ (7 - x) / (7 - 4.8) & ; 4.8 \leq x \leq 7 \\ 0; & ; x \geq 7 \end{cases}$$

$$\mu_{\text{Medium}}[x] = \begin{cases} 0; & ; x \geq 12 \\ (x - 4.8) / (7 - 4.8) & ; 4.8 \leq x \leq 7 \\ 1; & ; 7 \leq x \leq 9.8 \\ (12 - x) / (12 - 9.8) & ; 9.8 \leq x \leq 12 \\ 0; & ; x \geq 12 \end{cases}$$

$$\mu_{High} [x] = \begin{cases} 0 & ; x \geq 9.8 \\ (x - 9.8) / (12 - 9.8) & ; 9.8 \leq x \leq 12 \\ 1 & ; 12 \leq x \leq 14.5 \\ (16.45 - x) / (16.45 - 14.5) & ; 14.5 \leq x \leq 16.45 \\ 0 & ; x \geq 16.45 \end{cases}$$

4.1.2. Aplikasi fungsi implikasi

Setelah pembentukan himpunan *fuzzy*, maka dilakukan pembentukan aturan-aturan *fuzzy*. Tiap aturan merupakan suatu *implikasi*. Pada penelitian ini terdapat beberapa *rule*. Untuk menentukan aturan sebagai berikut:

$$\text{Rule} = \text{Banyak Keanggotaan}^{\text{banyak variabel / indikator}}$$

Semua *rule* yang terbentuk diatas telah terlampir dalam lampiran. Khusus untuk penentuan kualitas perangkat lunak yang memiliki 2188 *rule*, Berikut aturan *fuzzy* untuk penentuan kualitas dan *reusability* perangkat lunak, sebagai berikut pada Tabel 4.3.

Tabel 4.3. Aturan Fuzzy Penentuan Tingkat Bug

No	WMC	COB	DIT	NOC	LCOM	RFC	Hasil
1	Low	Low	Low	Low	Low	Low	trivial bugs
2	Medium	Low	Low	Low	Low	Low	trivial bugs
3	High	Low	Low	Low	Low	Low	trivial bugs
4	High	Medium	High	Low	Low	Low	normal bugs
5	Low	High	High	Low	Low	Low	trivial bugs
6	Medium	High	High	Low	Low	Low	normal bugs
8	Low	High	Medium	High	Medium	Low	critical bugs
9	Medium	High	Medium	High	Medium	Low	critical bugs
10	High	High	Medium	High	Medium	Low	critical bugs
11	Low	High	High	High	High	Low	critical bugs
12	Medium	High	High	High	High	Low	high priority bugs
13	High	High	High	High	High	Low	high priority bugs

Sumber : Hasil pengolahan data (2019)

Selanjutnya untuk aturan *fuzzy* penentuan tingkat kualitas dan *reusability* akan dikalsifikasikan kedalam empat tingkatan *bug* dari enam *metrics* yang diujikan dengan melihat Tabel 4.1. Di atas dapat diuraikan aturan *fuzzy*.

Rules 1:

1. *If (WMC is LOW) and (COB is LOW) and (DIT is LOW) and (NOC is LOW) and (LOCM is LOW) and (RFC is Low) then (Quality is trivial bugs)*

Rules 2:

2. *If (WMC is MEDIUM) and (COB is LOW) and (DIT is LOW) and (NOC is LOW) and (LOCM is LOW) and (RFC is Low) then (Quality is trivial bugs).*

Rules 3:

3. *If (WMC is HIGH) and (COB is LOW) and (DIT is LOW) and (NOC is LOW) and (LOCM is LOW) and (RFC is Low) then (Quality is trivial bugs).*

Rules 4:

4. *If (WMC is LOW) and (COB is MEDIUM) and (DIT is LOW) and (NOC is LOW) and (LOCM is LOW) and (RFC is Low) then (Quality is trivial bugs).*

Rules 5:

5. *If (WMC is MEDIUM) and (COB is MEDIUM) and (DIT is LOW) and (NOC is LOW) and (LOCM is LOW) and (RFC is Low) then (Quality is trivial bugs).*

Rules 6:

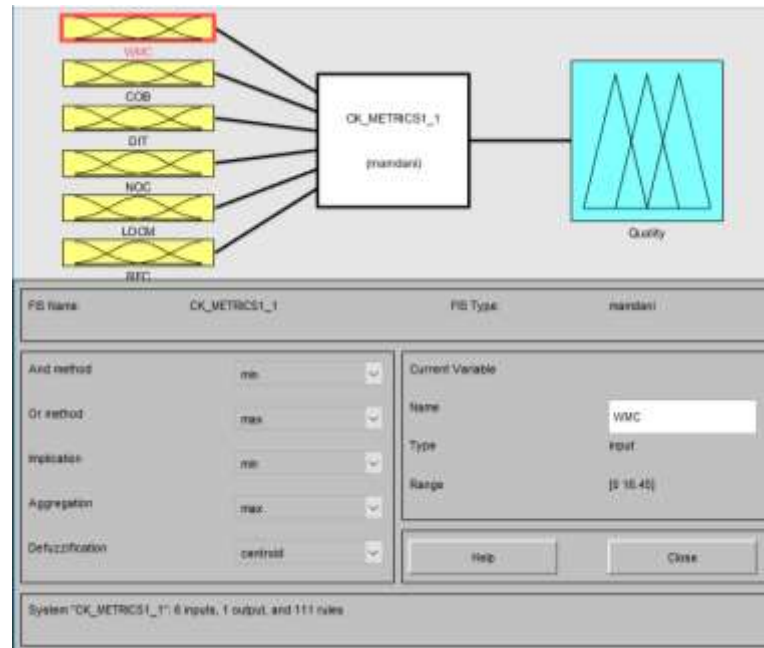
6. *If (WMC is HIGH) and (COB is MEDIUM) and (DIT is LOW) and (NOC is LOW) and (LOCM is LOW) and (RFC is Low) then (Quality is trivial bugs).*

Setelah aturan dibentuk maka dilakukan aplikasi fungsi implikasi. Fungsi implikasi yang dilakukan adalah *MIN* yang berarti tingkat keanggotaan yang didapat dari proses ini adalah nilai minimal dari indikator atau variabel *input* untuk mendapatkan daerah *fuzzy* pada indikator hasil penentuan kualitas perangkat lunak. Berdasarkan sampel penentuan properti desain perangkat lunak, maka fungsi implikasi sampel berikut:

4.1.3. Analisis dan Desain Sistem

Kriteria yang akan dianalisis dijadikan variabel *fuzzy* dari enam *metrics* dengan bantuan *Toolbox* MATLAB R2011b pada Gambar 4.6.

1. Sistem operasi terdapat 3 *input* dan 1 *output* :

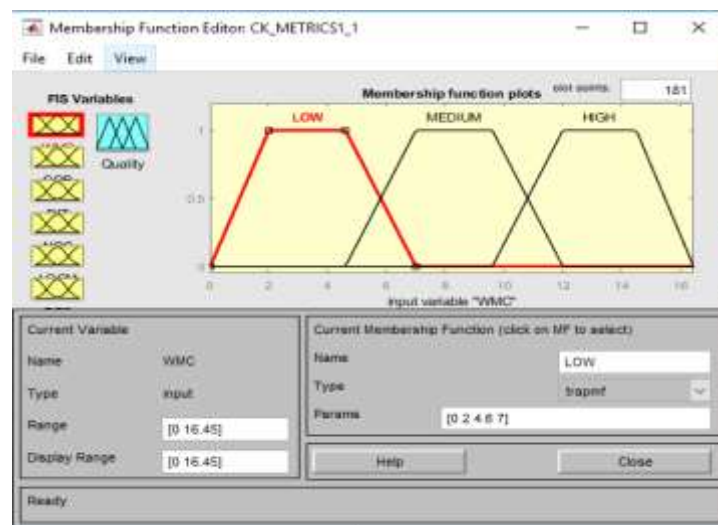


Sumber : Hasil pengolahan data (2019)

Gambar 4.6. Fuzzy Inference System Mamdani Editor

Himpunan *fuzzy input* terdiri dari enam inputan dan satu hasil *output*.

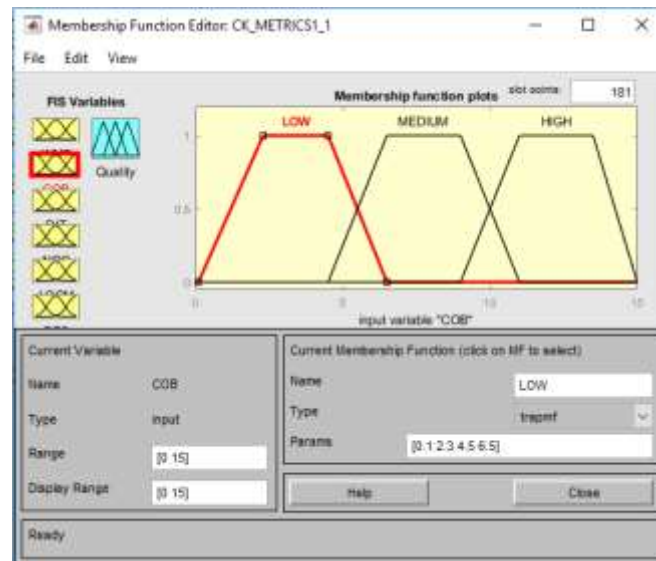
a. *Weight Methods per Class*



Sumber : Hasil pengolahan data (2019)

Gambar 4.7. Himpunan Fuzzy Input Indikator Weight Methods per Class

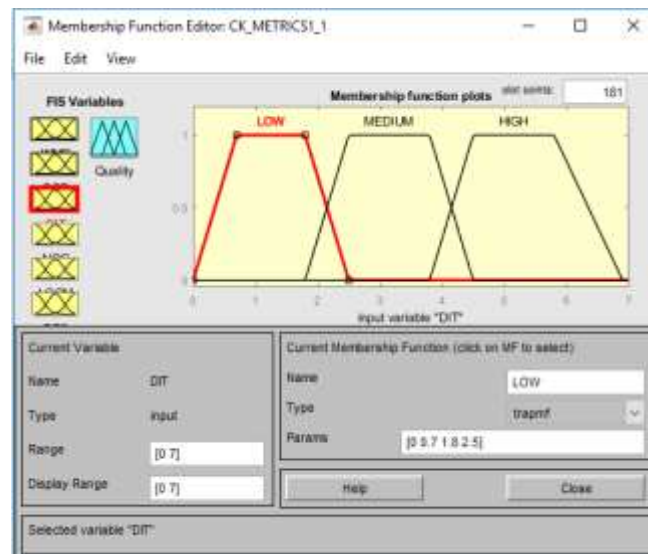
b. *Coupling Between Classes*



Sumber : Hasil pengolahan data (2019)

Gambar 4.8. Himpunan Fuzzy Input Indikator Coupling Between Classes

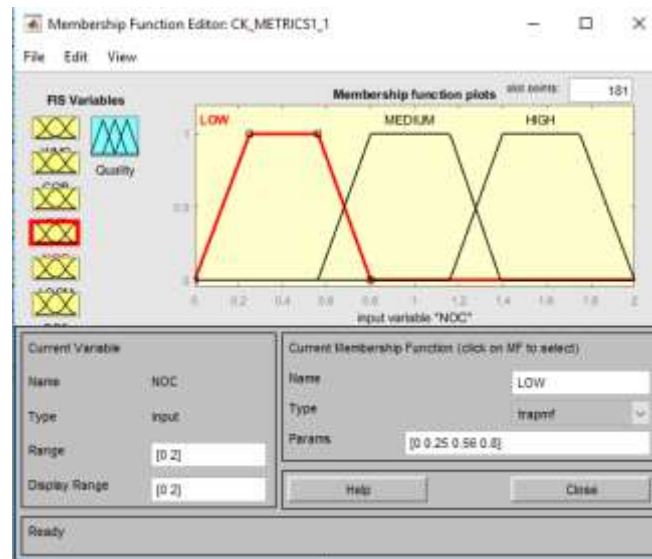
c. *Depth of Inheritance*



Sumber : Hasil pengolahan data (2019)

Gambar 4.9. Himpunan Fuzzy Input Indikator Depth of Inheritance

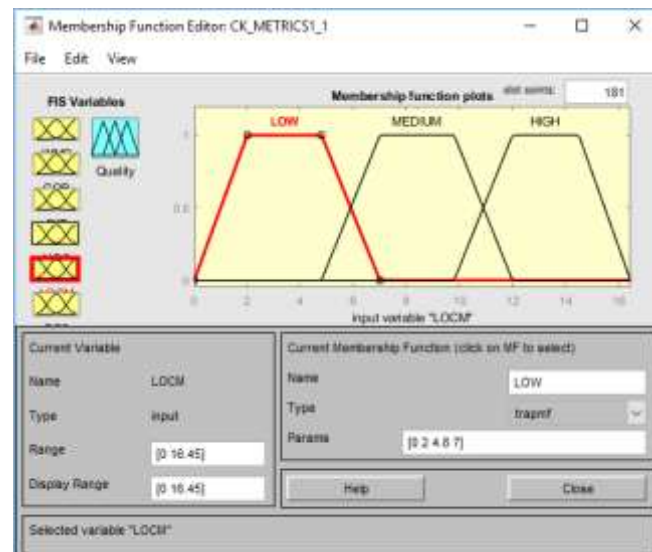
d. *Number of Children*



Sumber : Hasil pengolahan data (2019)

Gambar 4.10. Himpunan Fuzzy Input Indikator Number of Children

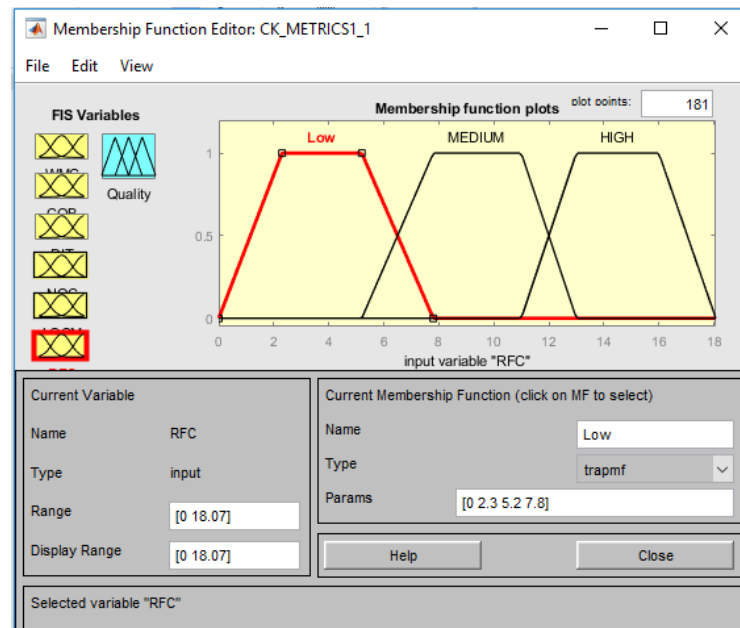
e. *Lack of Cohesion of Methods*



Sumber : Hasil pengolahan data (2019)

Gambar 4.11. Himpunan Fuzzy Input Indikator Lack of Cohesion of Methods

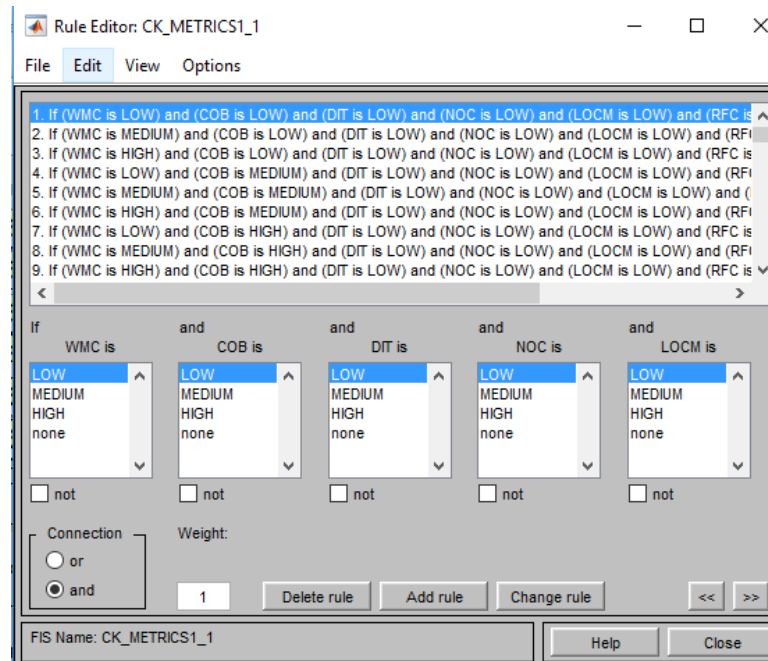
f. *Response sets For a Class*



Sumber : Hasil pengolahan data (2019)

Gambar 4.12. Himpunan Fuzzy Input Indikator Response sets For a Class

g. *Rule Integritas*



Sumber : Hasil pengolahan data (2019)

Gambar 4.13. Rule Integritas

4.1.4. Komparasi Aturan

Komposisi aturan fungsi *implikasi* menggunakan fungsi *MAX* yaitu dengan cara mengambil nilai maksimum dari *output* aturan. Jika semua proposisi telah dievaluasi, maka *output* akan berisi suatu himpunan *fuzzy* yang merefleksikan kontribusi dari tiap-tiap proposisi. Komposisi aturan untuk sampel sebelumnya adalah:

Variabel *output*

$$\begin{aligned}\text{Derajat kebenaran himpunan baik} &= \text{Max} (\alpha_{325}; \alpha_{325}) \\ &= \text{Max} (0,1645; 0) \\ &= 0,1645\end{aligned}$$

Daerah hasil inferensi tertinggi adalah 0,1645 dan terendah 0

4.1.5. Proses Defuzzyfikasi

Metode *defuzzyfikasi* pada kasus ini dilakukan dengan menggunakan metode COA (*centre of area*), Bisektor, Mom (*Mean Of Maximum*), Lom (*Largest Of Maximum*), dan Som (*Smallest Of Maximum*), untuk menentukan nilai *crisp x*, didapat dari fungsi keanggotaan yang terbentuk dari proses komposisi semua *output*. Berikut akan dijabarkan hasil dari metode *defuzzyfikasi* tersebut :

1. Metode COA (*Centre Of Area*)

$$\begin{aligned}COA &= \frac{(4.6 \times 0) + (7 \times 1) + (10 \times 1) + (12 \times 0)}{0 + 1 + 1 + 0} \\ &= \frac{0 + 7 + 10 + 0}{2} \\ &= 8,5\end{aligned}$$

3. Metode Bisektor membagi area dari derajat keanggotaan yang diperoleh kemudian mengambil nilai z lebih besar dari hasil pembagian di atas

$$\text{Sum } \mu(z) = 0 + 1 + 1 + 0 = 2$$

$$\text{Sum } \mu(z) / 2 = 0 + 1 + 1 + 0 / 2 = 1$$

Ambil nilai z dari yang hasil penjumlahan

$$\mu(z) \geq 1 = 7$$

4. Metode Mom (*Mean of Maximum*). Mengambil nilai z rata-rata dari nilai derajat keanggotaan ($\mu(z)$) yang maksimal.

$$= \frac{7 + 10}{2} + \frac{17}{2}$$

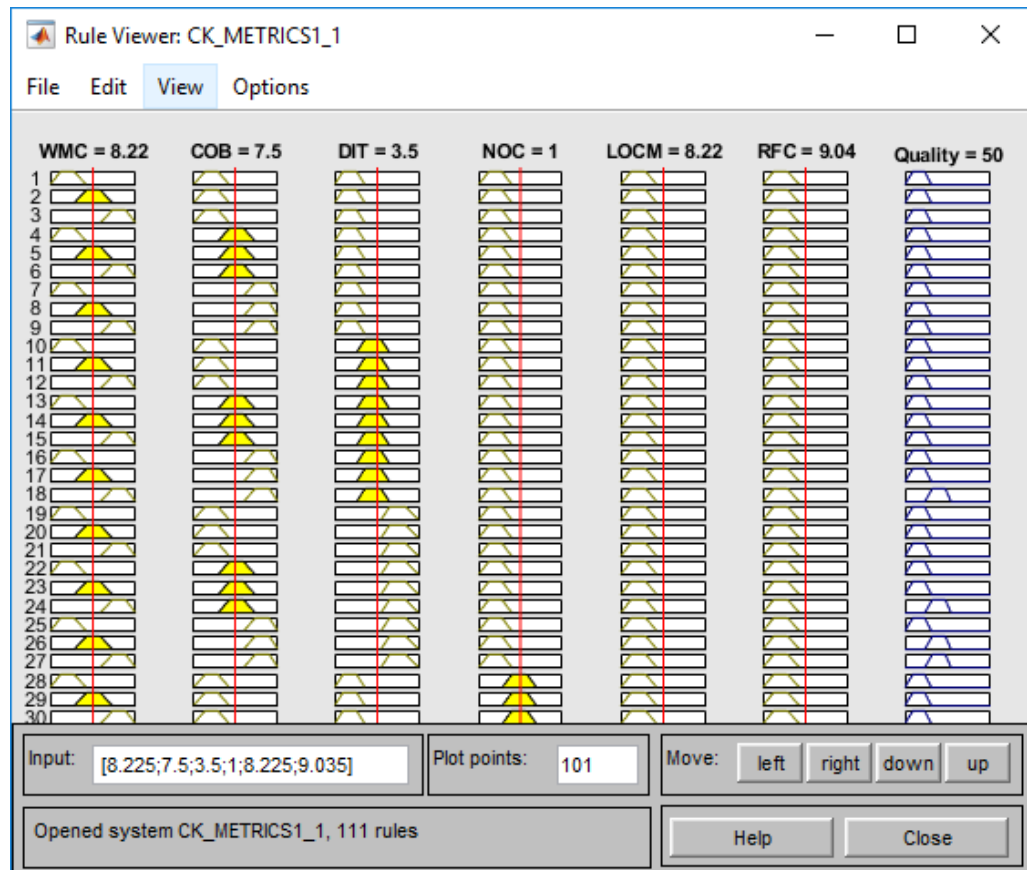
$$= 8.2$$

5. Metode Lom (*Largest Of Maximum*). Mengambil nilai z terbesar dari nilai derajat keanggotaan ($\mu(z)$) yang maksimal Dari nilai 4.6,7,10,12 yang terbesar adalah 12
6. Metode Som (*Smallest Of Maximum*). Mengambil nilai z terkecil dari nilai derajat keanggotaan ($\mu(z)$) yang maksimal dari nilai 4.6,7,10,12 yang terkecil adalah = 4.6

Jadi dapat disimpulkan dengan data-data yang ada dan setelah dianalisis data yang dijadikan sampel dikategorikan memberikan penilaian berikut dapat dicek menggunakan MATLAB R2018b sebagai *tools* yang digunakan pada penelitian ini.

4.6. PENGUJIAN

Pada tahap ini dilakukan pengujian untuk memperlihatkan perbandingan hasil *defuzzyfikasi* sesuai dengan beberapa metode yaitu metode COA (*center of area*), bisektor, MOM (*mean of maximum*), LOM (*largest of maximum*), dan SOM (*smallest of maximum*), berikut pengujian metode *deffuzzyfykasi* dengan MATLAB 2018b.

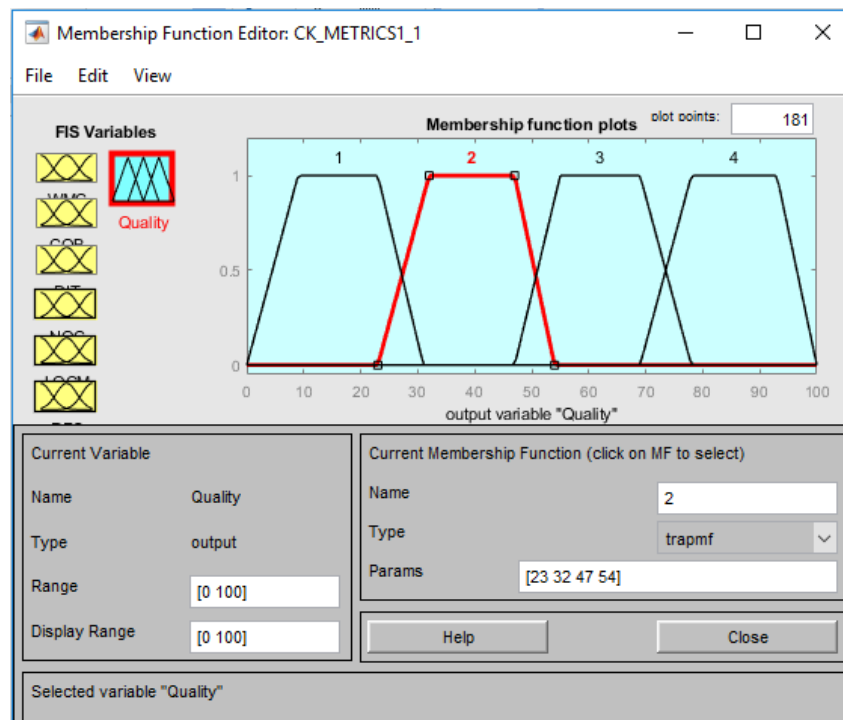


Sumber : Hasil pengolahan data (2019)

Gambar 4.14. Fuzzy Inference System Model Mamdani Editor Integritas

Pada Gambar 4.16 yang berwarna kuning menandakan komposisi variabel masukkan yang nilainya bisa berubah secara manual dengan memasukkan nilai pada *field* masukkan yang berada di pojok kiri bawah. Hasil untuk keluaran dapat langsung ditampilkan berada di kotak kanan atas. Hasil *deffuzzyfykasi* diatas akan diterapkan pada data pengujian. Berdasarkan sampel penilaian kualitas perangkat lunak dari enam properti desain (WMC, LCOM, CBO, DIT, RFC dan NOC) akan menitik beratkan pada penentuan kualitas dan *reusability* dengan penentuan tingkat *bug* pada set data *entropy eclipse equinox* data yang dijadikan bahan penelitian yaitu: *Trivial Bugs, Normal bugs, Critical Bugs, High Priority Bugs*.

h. Skor Hasil Pengujian



Sumber : Hasil pengolahan data (2019)

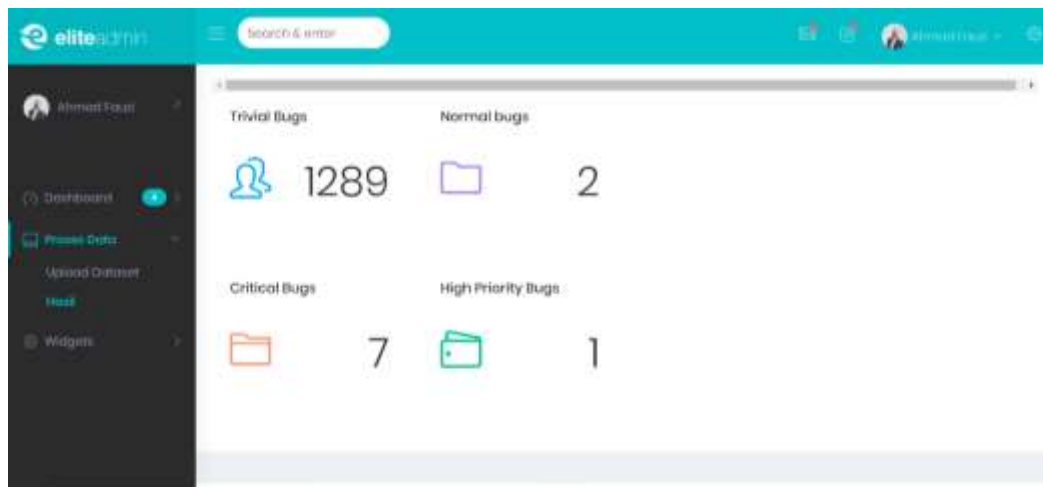
Gambar 4.15. Himpunan Fuzzy Ouput

WMC	CBQ	DIT	NDC	LOCM	BFC
20	23	20	20	20	16
104214	10696	0.430599	0	0.858065	1.00411
145478	142699	0.430599	0	0.6838	2.04012
225641	120076	0.430599	0	0	2.38788

Sumber : Hasil pengolahan data (2019)

Gambar 4.16. Halaman Isi Dataset

Pada Gambar 4.16. menunjukan hasil upload dataset yang akan dihitung sebagai data testing yang akan di gunakan sebagai bahan uji coba perhitungan *fuzzy logic*.



Sumber : Hasil pengolahan data (2019)

Gambar 4.16. Halaman Isi Dataset

Pada Gambar 4.17. menunjukan hasil pengklasifikasian dari data testing yang di ujikan. Berdasarkan hasil pengujian diatas di dapatkan *Trivial Bugs*, *Normal bugs*, *Critical Bugs*, *High Priority Bugs*.

BAB V

PENUTUP

5.1. Kesimpulan

Dari hasil penelitian yang dilakukan, maka dapat diambil kesimpulan sebagai berikut:

1. *Fuzzy Inference System* Mamdani. sangat cocok digunakan untuk mengukur sebuah kelas dari enam parameter desain perangkat lunak. Kesalahan desain bisa diantisipasi dalam proses perancangan perangkat lunak mulai dari proses *development* hingga *implementasi, fuzzy inference system*. Mamdani dapat digunakan untuk mengukur tingkat kualitas dan *reusability* perangkat lunak.
2. Dalam perhitungan prediksi kualitas dan *reusability* dengan menggunakan pendekatan logika *fuzzy inference system*. Mamdani tingkat kualitas yang baik memiliki angka yang relatif kecil, jika hasil perhitungan dari enam *metrics* tergolong tinggi maka tingkat *reusability* sangat rendah, hasil yang didapatkan akan dikategorikan ke dalam 4 klasifikasi *bug* yaitu : *trivial bugs, normal bugs, critical bugs, high priority bugs*. Dari set data yang diujikan memiliki tingkat *reusability* sangat baik dengan dan masuk ke dalam kategori *trivial bugs*.
3. Metode logika *fuzzy* dapat diterapkan pada analisa perbandingan pada sistem logika *fuzzy* dengan menggunakan beberapa metode *defuzzyfikasi*, yaitu metode COA (*center of area*), bisektor MOM (*mean of maximum*), LOM (*largest of maximum*), dan SOM (*smallest of maximum*) pada perediksi kualitas dan *reusability* perangkat lunak.

5.2. Saran

Dari hasil penelitian yang telah dilakukan, maka peneliti memberikan saran sebagai berikut :

1. Dalam penelitian ini perlu dilakukan testing lanjutan mengenai pengujian semua data proyek dari enam *metrics* yang terdapat pada subsistem *eclipse*

equinox dan sistem utama *eclipse*, sehingga diketahui secara utuh bahwa proyek subsistem dan sistem utama dapat digunakan kembali atau tidak.

2. Diperlukan *tools* yang dapat menguji enam parameter desain, dengan cara yang mudah dimengerti dan dipahami oleh semua *development* perangkat lunak.
3. Untuk studi penelitian selanjutnya, pemodelan ini dapat dikembangkan lebih lanjut sehingga memudahkan pengguna dan menyempurnakan penelitian ini dalam memprediksi dan perbandingan kualitas perangkat lunak.

DAFTAR REFERENSI

- Ambros, Marco D, Michele Lanza, and Romain Robbes. 2010. "An Extensive Comparison of Bug Prediction Approaches." : 31–41.
- Axelius, Victor, Kristianto Siren, Nanang Yudi Setiawan, and Retno Indah Rokhmawati. 2019. "Evaluasi Kualitas Perangkat lunak Menggunakan ISO / IEC 9126-4 Quality In Use (Studi Kasus : FILKOM Apps)." 3(2): 1625–32.
- Babich, Djuradj, Peter J Clarke, and James F Power. 2011. "Using a Class Abstraction Technique to Predict Faults in OO Classes : A Case Study through Six Releases of the Eclipse JDT Using a Class Abstraction Technique to Predict Faults in OO Classes A Case Study through Six Releases of the Eclipse JDT." (May 2014).
- Bhatia, Pradeep Kumar, and Rajbeer Mann. 2015. "An Approach to Measure Perangkat lunak Reusability of OO Design An Approach to Measure Perangkat lunak Reusability of OO Design." (January 2008).
- Bianchi, Alessandro et al. 2001. "Evaluating Perangkat lunak Degradation through Entropy." (May 2014).
- Borys, Magdalena. 2014. "Metrics of Object Oriented Perangkat lunak ." (May).
- Goel, Brij Mohan. 2011. "Analysis of Perangkat lunak Measurement the CK Metrics for Object-Oriented Design Complexity." (Acct): 978–81.
- Goyal, Rinkaj, Pravin Chandra, and Yogesh Singh. 2013. "Identifying Influential Metrics in the Combined Metrics Approach of Fault Prediction."
- Halim, Arwin, Alex Xandra, and Albert Sim. 2015. "Mengukur Tingkat Reusability Dan Efficiency Dari Kode Program Dengan Pendekatan Fuzzy Logic." 4: 109–14.
- Haryatmi, Emy, and Muhammad Yuda Mashuri. 2016. "Penerapan Fuzzy Logic Inference System Metode Mamdani Sebagai Penunjang Diagnosis Kanker Paru." (April).

- Hyatt, Lawrence E, Liida H Rosenberg, and D Ph. 1997. "PII: Soo94-5765(97)00148-3." 40(2): 223–33.
- Jang, Jyh-shing Roger. 1995. "Neuro-Fuzzy Modeling." 83(3).
- K, Awang Harsa, Addy Suyatno, and Juwita Rahayu. 2011. "Simulasi Pendaratan Pesawat Terbang Jenis Cassa 212 Menggunakan Logika Fuzzy Landing Airplane Simulation for Cassa 212 Using Fuzzy Logic." 2.
- Kumar, Anil. 2019. "Analysis of Object-Oriented System Quality Model Using Soft Computing Techniques." 5(2): 214–24.
- Mago, Jagmohan. "Analysis of Quality of the Design of the Object Oriented Perangkat lunak Using Fuzzy Logic." : 21–25.
- Mago, Jagmohan, Sarabjit Kaur, and Kumar Saurabh. 2012. "Fuzzy Model to Analyze and Interpret Object Oriented Perangkat lunak Design."
- Qureshi, M Rizwan Jameel. 2015. "Evaluation of the Design Metric to Reduce the Number of Defects in Perangkat lunak Development." (April 2012).
- Sandhu, Parvinder S, and Hardeep Singh. 2008. "A Reusability Evaluation Model for OO-Based Perangkat lunak Components." 2(3): 912–17.
- September, Approved. 1990. "IEEE Standard Glossary of Perangkat lunak Engineering Terminology." 121990.
- Shri, Anju, Parvinder S Sandhu, Vikas Gupta, and Sanyam Anand. 2010. "Prediction of Reusability of Object Oriented Perangkat lunak Systems Using Clustering Approach." 4(7): 1174–77.
- Suresh, Yeresime, Jayadeep Pati, and Santanu Ku Rath. 2012. "Effectiveness of Perangkat lunak Metrics for Object-Oriented System Effectiveness of Perangkat lunak Metrics for Object-Oriented System." (December).
- Suryawati, Endang. 2013. "Fuzzy Model for Predicting Reusability Factor of a Perangkat lunak Model Fuzzy Untuk Memprediksi Faktor Reusability Sebuah Perangkat lunak ." 7(1): 7–9.
- Turnu, Ivana, Michele Marchesi, and Roberto Tonelli. 2012. "Entropy of the

Degree Distribution and Object-Oriented Perangkat lunak Quality.” (June).

Utara, Universitas Sumatera, and Universitas Sumatera Utara. 2018. *Analisis Kinerja Fuzzy AHP Dalam Perangkingan*.

Wahab, Faisal et al. 2017. “Desain Dan Purwarupa Fuzzy Logic Control Untuk Pengendalian Suhu Ruangan.” 2(1): 1–8.