

**PENERAPAN GREEDY FORWARD SELECTION DAN BAGGING
PADA LOGISTIC REGRESSION UNTUK PREDIKSI CACAT
PERANGKAT LUNAK**



TESIS

**RAKHMAT PURNOMO
14000973**

**PROGRAM PASCASARJANA MAGISTER ILMU KOMPUTER
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
NUSA MANDIRI
JAKARTA
2015**

HALAMAN PENGESAHAN

Tesis ini diajukan oleh:

Nama : Rakhmat Purnomo
NIM : 14000973
Program Studi : Magister Ilmu Komputer
Jenjang : Strata Dua (S2)
Konsentrasi : *Management Information System*
Judul Tesis : "Penerapan *Greedy Forward Selection* dan *Bagging* pada *Logistic Regression* untuk Prediksi Cacat Perangkat Lunak"

Telah berhasil dipertahankan dihadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Magister Ilmu Komputer (M.Kom) pada Program Pascasarjana Magister Ilmu Komputer Sekolah Tinggi Manajemen Informatika dan Komputer Nusa Mandiri (STMIK Nusa Mandiri).

Jakarta, 1 Desember 2015
Pascasarjana Magister Ilmu Komputer
STMIK Nusa Mandiri
Direktur

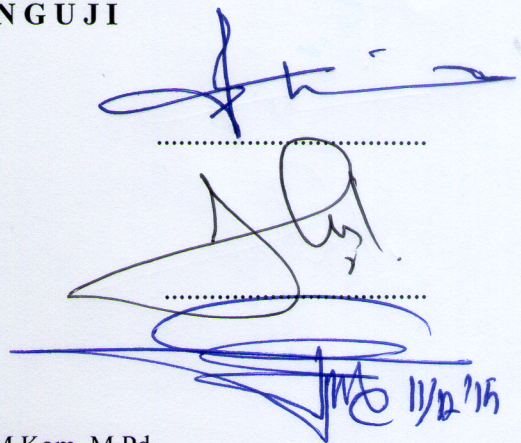
Prof. Dr. Ir. R. Eko Indrajit, M.Sc, MBA

DEWAN PENGUJI

Penguji I : Dr. Sfenrianto, M.Kom

Penguji II : Dr. Windu Gata, M.Kom

Pembimbing : Dr. Mochamad Wahyudi, MM, M.Kom, M.Pd





Lembar Konsultasi Bimbingan Mini Tesis

Pascasarjana Magister Ilmu Komputer STMIK Nusa Mandiri

- NIM : 14000973
- Nama Lengkap : Rakhmat Purnomo
- Dosen Pembimbing : Dr. Mochammad Wahyudi, MM, M.Kom, M.Pd
- Judul Tesis : Penerapan Greedy Forward Selection dan Bagging pada Logistic Regression untuk Prediksi Cacat Perangkat Lunak



NO	Tanggal Bimbingan	Materi Bimbingan	Paraf Doses Pembimbing
1	26 Okt 2015	Bimbingan perdana dan Pengajuan Bab I	
2	2 Nov 2015	Pengajuan Bab II	
3	9 Nov 2015	Pengajuan Bab III	
4	16 Nov 2015	Pengajuan Bab IV	
5	23 Nov 2015	Pengajuan Bab V	
6	30 Nov 2015	Periksa Keseluruhan dan ACC Ujian Sidang	

Catatan :

Total bimbingan yang harus dilakukan adalah 6 (enam) kali pertemuan.

- Bimbingan dimulai pada tanggal : 26 Oktober 2015
- Bimbingan diakhiri pada tanggal : 30 November 2015
- Jumlah pertemuan : 6 kali Bimbingan

Jakarta, 30 November 2015

Dosen Pembimbing

Dr. Mochammad Wahyudi, MM, M.Kom, M.Pd

SURAT PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini :

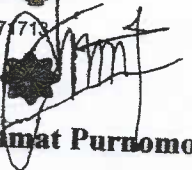
Nama : Rakhmat Purnomo
NIM : 14000973
Program Studi : Magister Ilmu Komputer
Jenjang : Strata Dua (S2)
Konsentrasi : *Management Information System*

Dengan ini menyatakan bahwa tesis yang telah saya buat dengan judul : "Penerapan Greedy Forward Selection dan Bagging pada Logistic Regression untuk Prediksi Cacat Perangkat Lunak" adalah hasil karya sendiri, dan semua sumber baik yang kutipan maupun yang dirujuk telah saya nyatakan dengan benar. Tesis ini belum pernah diterbitkan atau dipublikasikan dimanapun dan dalam bentuk apapun.

Demikian surat pernyataan ini saya buat dengan sebenar-benarnya. Apabila dikemudian hari ternyata saya memberikan keterangan palsu dan atau ada pihak lain yang mengklaim bahwa tesis yang telah saya buat adalah hasil karya milik seseorang atau badan tertentu, saya bersedia diproses baik secara pidana maupun perdata dan kelulusan saya dari Program Pascasarjana Magister Ilmu Komputer Sekolah Tinggi Manajemen Informatika dan Komputer Nusa Mandiri dicabut/dibatalkan.

Jakarta, 11 Desember 2015

Yang Menyatakan,



Rakhmat Purnomo

**SURAT PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH
UNTUK KEPENTINGAN AKADEMIS**

Nama : Rakhmat Purnomo
NIM : 14000973
Program Studi : Magister Ilmu Komputer
Jenjang : Strata (S2)
Konsentrasi : *Management Information System*
Jenis Karya : Tesis

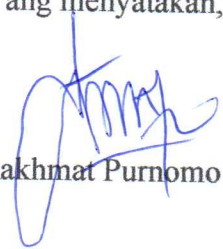
Demi pengembangan ilmu pengetahuan, dengan ini menyetujui untuk memberikan izin kepada pihak Program Pascasarjana Magister Ilmu Komputer Sekolah Tinggi Manajemen Informatika dan Komputer Nusa Mandiri (STMIK Nusa Mandiri) Hak Bebas Royalti Non-Eksklusif (Non-exclusive Royalti-Free Right) atas karya ilmiah kami yang berjudul: "Penerapan Greedy forward selection dan Bagging pada Prediksi Cacat Software Menggunakan Logistic Regression" beserta perangkat yang diperlukan (apabila ada).

Dengan Hak Bebas Royalti Non-Eksklusif ini pihak STMIK Nusa Mandiri berhak menyimpan, mengalih-media atau bentuk-kan, mengelolaannya dalam pangkalan data (database), mendistribusikannya dan menampilkan atau mempublikasikannya di internet atau media lain untuk kepentingan akademis tanpa perlu meminta izin dari kami selama tetap mencantumkan nama kami sebagai penulis/pencipta karya ilmiah tersebut.

Saya bersedia untuk menanggung secara pribadi, tanpa melibatkan pihak STMIK Nusa Mandiri, segala bentuk tuntutan hukum yang timbul atas pelanggaran Hak Cipta dalam karya ilmiah saya ini.

Demikian pernyataan ini saya buat dengan sebenarnya.

Jakarta, 1 Desember 2015
Yang menyatakan,


Rakhmat Purnomo

KATA PENGANTAR

Puji syukur Alhamdulillah, penulis panjatkan kehadiran Allah SWT, yang telah melimpahkan rahmat dan karunia-Nya sehingga pada akhirnya penulis dapat menyelesaikan tesis ini dengan baik dimana tesis ini penulis buat dengan judul sebagai berikut: “Penerapan Bagging dan Greedy forward selection pada Prediksi Cacat Software Menggunakan Logistic Regression”.

Tujuan penulisan skripsi ini dibuat untuk memenuhi salah satu syarat kelulusan Program Pascasarjana Magister Ilmu Komputer Sekolah Tinggi Manajemen Informatika Nusa Mandiri (STMIK Nusa Mandiri). Tesis ini dibuat dari hasil penelitian pada prediksi cacat software dengan menggunakan model usulan seleksi fitur dengan greedy forward selection dan bagging untuk penanganan ketidakseimbangan kelas dan redundant data. Penulisan tesis ini juga mencari dan menganalisa berdasarkan hasil penelitian, observasi dan beberapa sumber literatur yang berhubungan dalam penelitian ini.

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

1. Bapak Dr. Mochamad Wahyudi sebagai pembimbing yang selalu memberikan ilmu, membimbing dan mendukung untuk penulis.
2. Orangtua dan keluarga yang selalu memberikan nasihat baik kepada penulis.
3. Teman-teman seperjuangan yang selalu memberikan dukungan dan semangat.

Serta semua pihak yang terlalu banyak untuk disebut satu persatu sehingga tesis ini dapat diselesaikan oleh penulis. Penulis menyadari bahwa penulisan tesis ini masih belum sempurna, untuk itu penulis mohon kritik dan saran yang bersifat membangun demi kesempurnaan penelitian dimasa yang akan datang. Akhir kata penulis ucapkan semoga tesis ini dapat bermanfaat untuk semuanya.

Jakarta, 1 Desember 2015

Penulis

Rakhmat Purnomo

ABSTRAK

Nama : Rakhmat Purnomo
NIM : 14000973
Program Studi : Magister Ilmu Komputer
Jenjang : Strata (S2)
Konsentrasi : *Management Information System*
Judul Tesis : Penerapan *Greedy Forward Selection* Dan *Bagging* Pada Logistic Regression Untuk Prediksi Cacat Perangkat Lunak

Cacat perangkat lunak merupakan kesalahan atau kegagalan pada perangkat lunak. Pemeriksaan cacat perangkat lunak secara manual hanya dapat menghasilkan 60% dari total cacat yang ada. Metode prediksi cacat menggunakan probabilitas dapat menemukan sampai 71% lebih baik dari metode yang digunakan oleh industri. Salah satu metode terbaik untuk prediksi cacat perangkat lunak adalah Logistic Regression. Logistic Regression merupakan pengklasifikasi linier yang telah terbukti menghasilkan klasifikasi yang powerful dengan statistik probabilitas dan menangani masalah klasifikasi multi kelas. Kelemahan utama algoritma Logistic Regression adalah ketidakseimbangan kelas pada dataset berdimensi tinggi. Dataset software metric yang digunakan adalah dataset NASA MDP. Dataset ini umumnya bersifat tidak seimbang dan mengalami masalah pada redundant data. Penelitian ini mengusulkan metode greedy forward selection untuk mengatasi masalah redundant data dan metode bagging untuk mengatasi ketidakseimbangan kelas. Algoritma yang digunakan adalah Logistic Regression. Hasil percobaan pada penelitian ini mendapatkan nilai akurasi tertinggi sebesar 0,990 pada dataset PC2, naik 0,19% dibandingkan dengan metode logistic regression tanpa GFS dan bagging. Sedangkan nilai AUC tertinggi sebesar 0,995 pada PC2, naik 7,94% dibandingkan dengan metode logistic regression tanpa GFS dan bagging.

Kata Kunci: Greedy Forward Selection, Bagging, Logistic Regression, dataset NASA MDP, Prediksi Cacat Perangkat Lunak.

ABSTRACT

Name : Rakhmat Purnomo
NIM : 14000973
Study of Program : Magister Ilmu Komputer
Level : Strata (S2)
Concentration : Management Information System
Title : Implementation Greedy Forward Selection and Bagging on
Logistic Regression for Software Defect Prediction

Software defects are errors or failures in software. Software defect detection manually can only produce 60% of the total existing defects. Defect prediction method using probability to find up to 71% better than the method used by the industry. One of the best methods for prediction of software defects is Logistic Regression. Logistic Regression is a linear classifier that has been shown to produce a powerful classification with statistical probabilities and handle multi-class classification problem. The main weakness of Logistic Regression algorithm is a class imbalance in high-dimensional datasets. Dataset software metric used is NASA dataset MDP. The dataset is generally uLRalanced and experiencing problems with redundant data. This paper proposed a greedy forward selection method to solving the problem of redundant data and bagging technic to solving the class imbalance. The algorithm used is the Logistic Regression. Results of the experiments in this study scored the highest accuracy in the dataset PC2 at 0,990, up 0.19% compared with logistic regression method without GFS and bagging. While the highest AUC value of 0.995 at PC2, an increase of 7.94% compared to logistic regression method without GFS and bagging.

Keywords: *Greedy Forward Selection, Bagging, Logistic Regression, dataset NASA MDP, Software Defect Prediction.*

DAFTAR ISI

SURAT PERNYATAAN ORISINALITAS	i
HALAMAN PENGESAHAN	ii
LEMBAR KONSULTASI BIMBINGAN TESIS	iii
KATA PENGANTAR	iv
SURAT PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS.....	v
ABSTRAK.....	vi
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR	xiv
DAFTAR LAMPIRAN.....	xv
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang Penelitian	1
1.2. Identifikasi Masalah	4
1.3. Rumusan Masalah	4
1.4. Tujuan Penelitian.....	4
1.5. Manfaat Penelitian.....	5
1.6. Ruang Lingkup	5
1.7. Sistematika Penulisan.....	5
BAB II LANDASAN TEORI	7
2.1. Tinjauan Studi	7
2.1.1. Model Penelitian Komarek dan Moore (2005)	7
2.1.2. Model Penelitian Lin, Weng dan Keerthi (2008).....	8
2.1.3. Model Maalouf dan Trafalis (2011).....	9

2.1.4.	Model Penelitian Wahono, Suryana, dan Ahmad (2014)	10
2.1.5.	Rangkuman Penelitian Terkait	11
2.2.	Tinjauan Pustaka	13
2.2.1.	Prediksi Cacat Software	13
2.2.2.	Logistic Regresion	15
2.2.1.	Dataset NASA	16
2.2.2.	Teknik Validasi dan Evaluasi	18
2.2.3.	Algoritma Greedy	21
2.2.4.	Greedy Forward Selection	21
2.2.5.	Bagging	23
2.3.	Feature Selection	23
2.4.	Information Gain	24
2.5.	Teknik Ensemble	24
2.6.	Class Imbalance	24
2.7.	Confusion Matrix	25
2.8.	Kerangka Pemikiran Penelitian	25
BAB III METODE PENELITIAN		27
3.1.	Perancangan Penelitian	27
3.2.	Pengumpulan Data	28
3.3.	Pengolahan Data Awal	31
3.4.	Metode Yang Diusulkan	31
3.5.	Eksperimen dan Pengujian Metode	32
3.6.	Evaluasi dan Validasi Hasil	33
BAB IV HASIL DAN PEMBAHASAN		35
4.1.	Hasil	35
4.1.1.	Hasil Pengukuran Logistic Regression	35

4.1.2.	Hasil Pengukuran Logistic Regression dan Greedy Forward Selection	36
4.1.3.	Hasil Pengukuran Logistic Regression dan Bagging	36
4.1.4.	Hasil Pengukuran Logistic Regression, Greedy Forward Selection, dan Bagging	37
4.1.5.	Hasil Pengukuran Akurasi	37
4.1.6.	Hasil Pengukuran AUC.....	38
4.1.7.	Hasil Perbandingan AUC Model Prediksi Cacat Perangkat Lunak	40
4.2.	Pembahasan	41
4.2.1.	Perbandingan Kinerja Logistic Regression dengan Model yang diusulkan	41
4.2.2.	Perbandingan Logistic regression dengan Seleksi Fitur yang Lain	79
BAB V PENUTUP.....		82
5.1.	Kesimpulan.....	82
5.2.	Saran.....	83
DAFTAR REFERENSI		84
LAMPIRAN – LAMPIRAN		lxxxvii
DAFTAR RIWAYAT HIDUP		cviii

DAFTAR TABEL

Tabel 2.1 Contoh Umum Cacat Software	13
Tabel 2.2 Contoh Nyata Cacat Software.....	14
Tabel 2.3 Diskripsi Dataset NASA MDP	17
Tabel 2.4 Spesifikasi Dataset NASA MDP	17
Tabel 2.5 Spesifikasi Dataset Pembersihan Noise Kesatu	18
Tabel 2.6 Spesifikasi Dataset Pembersihan Noise kedua.....	18
Tabel 2.7 Confusion Matrix	25
Tabel 3.1 Dataset NASA MDP dan Atributnya.....	29
Tabel 3. 2 Deskripsi Atribut pada Dataset NASA MDP.....	29
Tabel 3.3 Nilai AUC, Keterangan dan Simbol	33
Tabel 4.1 Hasil Pengukuran LR.....	35
Tabel 4.2 Hasil Pengukuran LR + GFS	36
Tabel 4.3 Hasil Pengukuran LR + BG	37
Tabel 4.4 Hasil Pengukuran LR + GFS + Bagging.....	37
Tabel 4.5 Hasil Pengukuran Akurasi LR, LR + GFS, LR + BG, dan LR + GFS + BG	38
Tabel 4.6 Hasil Pengukuran AUC LR, LR + GFS, LR + BG, dan LR + GFS + BG	39
Tabel 4.7 Perbandingan AUC Model Prediksi Cacat Perangkat Lunak	40
Tabel 4.8 Perbandingan Akurasi LR dan LR+BG	41
Tabel 4.9 Hasil Uji Beda Statistik Akurasi LR dan LR+BG	42
Tabel 4.10 Perbandingan Sensitifitas LR dan LR+BG	43
Tabel 4.11 Hasil Uji beda Statistik Sensitifitas LR dan LR+BG	43
Tabel 4.12 Perbandingan F-measure LR dan LR+BG	44
Tabel 4.13 Hasil Uji beda Statistik F-measure LR dan LR+BG.....	44
Tabel 4.14 Perbandingan G-mean LR dan LR+BG	45
Tabel 4.15 Hasil Uji beda Statistik G-mean LR dan LR+BG.....	46
Tabel 4.16 Perbandingan AUC LR dan LR+BG	47

Tabel 4.17 Hasil Uji beda Statistik AUC LR dan LR+BG	47
Tabel 4.18 Perbandingan Akurasi LR dan LR+GFS	48
Tabel 4. 19 Hasil Uji beda Statistik Akurasi LR dan LR+GFS	49
Tabel 4.20 Perbandingan Sensitifitas LR dan LR+GFS	50
Tabel 4.21 Hasil Uji Beda Statistik Sensitifitas LR dan LR+GFS	50
Tabel 4.22 Perbandingan F-measure LR dan LR+GFS	51
Tabel 4.23 Hasil Uji Beda Statistik F-measure LR dan LR+GFS	52
Tabel 4.24 Perbandingan G-mean LR dan LR+GFS	53
Tabel 4.25 Hasil Uji beda Statistik G-mean LR dan LR+GFS	53
Tabel 4.26 Perbandingan AUC LR dan LR+GFS.....	54
Tabel 4.27 Hasil Uji Beda Statistik AUC LR dan LR+GFS.....	54
Tabel 4. 28 Perbandingan Akurasi LR dan LR+GFS+BG.....	56
Tabel 4. 29 Hasil Uji Beda Statistik Akurasi LR dan LR+GFS+BG.....	56
Tabel 4. 30 Perbandingan Sensitifitas LR dan LR+GFS+BG	57
Tabel 4. 31 Hasil Uji Beda Statistik Sensitifitas LR dan LR+GFS+BG.....	58
Tabel 4. 32 Perbandingan F-measure LR dan LR+GFS+BG	59
Tabel 4. 33 Hasil Uji Beda Statistik F-measure LR dan LR+GFS+BG	59
Tabel 4. 34 Perbandingan G-mean LR dan LR+GFS+BG	60
Tabel 4. 35Tabel 4.57 Hasil Uji Beda Statistik G-mean LR dan LR+GFS+BG ..	61
Tabel 4. 36 Perbandingan AUC LR dan LR+GFS+BG.....	62
Tabel 4. 37 Hasil Uji Beda Statistik AUC LR dan LR+GFS+BG.....	62
Tabel 4. 38 Perbandingan Akurasi LR+BG dan LR+GFS+BG.....	63
Tabel 4. 39 Hasil Uji Beda Statistik LR+BG dan LR+GFS+BG	64
Tabel 4. 40 Perbandingan Sensitifitas LR+BG dan LR+GFS+BG.....	65
Tabel 4. 41 Hasil Uji beda Statistik Sensitifitas LR+BG dan LR+GFS+BG	66
Tabel 4. 42 Perbandingan F-measure LR+BG dan LR+GFS+BG	67
Tabel 4. 43 Hasil Uji Beda Statistik F-measure LR+BG dan LR+GFS+BG.....	67
Tabel 4. 44 Perbandingan G-mean LR+BG dan LR+GFS+BG	68
Tabel 4. 45 Hasil Uji Beda Statistik G-mean LR+BG dan LR+GFS+BG.....	69
Tabel 4. 46 Perbandingan AUC LR+BG dan LR+GFS+BG	70
Tabel 4. 47 Hasil Uji beda Statistik AUC LR+BG dan LR+GFS+BG.....	70
Tabel 4. 48 Perbandingan Akurasi LR+BG dan LR+GFS.....	72

Tabel 4. 49 Hasil Uji beda Statistik Akurasi LR+BG dan LR+GFS	72
Tabel 4. 50 Perbandingan Sensitifitas LR+BG dan LR+GFS	73
Tabel 4. 51 Hasil Uji Beda Statistik Sensitifitas LR+BG dan LR+GFS.....	74
Tabel 4. 52 Perbandingan F-measure LR+BG dan LR+GFS	75
Tabel 4. 53 Hasil Uji Beda Statistik F-measure LR+BG dan LR+GFS	75
Tabel 4. 54 Perbandingan G-mean LR+BG dan LR+GFS	76
Tabel 4. 55 Hasil Uji Beda Statistik G-mean LR+BG dan LR+GFS	76
Tabel 4. 56 Perbandingan AUC LR+BG dan LR+GFS.....	78
Tabel 4. 57 Hasil Uji Beda Statistik AUC LR+BG dan LR+GFS.....	78
Tabel 4. 58 Perbandingan AUC pada model prediksi cacat software.....	79
Tabel 4. 59 Uji Friedman	80
Tabel 4. 60 Perbandingan Multiple Pairwise Menggunakan Prosedur Nemenyi .	80

DAFTAR GAMBAR

Gambar 2.1 Model Penelitian Komarek & Moore (2005)	8
Gambar 2.2 Model Lin, Wen, & Keerthi (2008)	9
Gambar 2.3 Model Maalouf dan Trafalis (2011)	10
Gambar 2.4 Model Wahono, Suryana, dan Ahmad (2014)	11
Gambar 2.5 10-Cross Fold Validation	19
Gambar 2.6 Metode Heuristik (Greedy) untuk Seleksi Atribut Subset	22
Gambar 2.7 Kerangka Pemikiran	26
Gambar 3.1 Tahapan Penelitian	27
Gambar 3.2 Metode yang diusulkan	32
Gambar 3.3 Confusion Matrix	33
Gambar 4.1 Perbandingan Akurasi dari metode LR, LR+GFS, LR+BG, dan LR+GFS+BG	38
Gambar 4.2 Perbandingan AUC dari metode LR, LR+GFS, LR+BG, dan LR+GFS+BG	39
Gambar 4.3 Perbandingan Nilai AUC Model Prediksi Cacat Perangkat Lunak...	40

DAFTAR LAMPIRAN

Lampiran 1. Hasil Seleksi Fitur dengan Greedy Forward Selection.....	lxxxvii
Lampiran 2 Rule Logistic Regression dengan Greedy Forward Selection dan Bagging	xcii

BAB I

PENDAHULUAN

1.1. Latar Belakang Penelitian

Perusahaan atau organisasi memerlukan perangkat lunak untuk meningkatkan efisiensi dan efektifitas pada proses bisnisnya. Perangkat lunak yang digunakan umumnya dibuat dan disesuaikan dengan kebutuhan perusahaan. Pembuatan perangkat lunak yang berkualitas tinggi selama proses pengembangan dan perawatan memerlukan biaya yang sangat mahal (Wahono et al., 2014), Pemeriksaan cacat software secara manual hanya dapat menghasilkan 60% dari total cacat yang ada. Sehingga perlu dicarikan alternatif lain.

Menurut (Naik & Tripathy, 2008), cacat perangkat lunak merupakan kesalahan atau kegagalan pada perangkat lunak. Perangkat lunak disebut gagal jika tidak dapat menjalankan fungsi sebagaimana yang telah dipersyaratkan sebelumnya. Indikator perangkat lunak yang berkualitas tinggi adalah tidak ditemukan cacat selama pengembangan dan pengujian (McDonald et al., 2008). Cacat perangkat lunak dapat juga dihasilkan oleh pengembang perangkat lunak sebagai akibat dari pengkodean program yang salah selama program dibuat. Pada beberapa kasus, perangkat lunak yang berkualitas tinggi banyak dinilai dari pengguna, sehingga perlu diukur dari sudut pandang pengguna.

Pengujian perangkat lunak biasanya memerlukan waktu sekitar 50% dari jadwal keseluruhan waktu pengerjaan (Fakhrahmad & Sami, 2009). Hal ini menunjukkan pentingnya proses pengujian untuk menghasilkan perangkat lunak yang berkualitas tinggi.

Untuk meningkatkan kualitas dan menjamin sebuah perangkat lunak berkualitas tinggi diperlukan program untuk prediksi cacat software (Chang et al., 2011). Prediksi cacat perangkat lunak digunakan untuk memprediksi modul perangkat lunak yang rawan cacat dengan yang tidak rawan cacat sehingga dapat di rilis perangkat lunak yang berkualitas tinggi. Metode prediksi cacat menggunakan probabilitas dapat menemukan sampai 71% (Menzies et al., 2007), lebih baik dari metode yang digunakan oleh industri. Hasil tersebut membuktikan bahwa

menggunakan probabilitas merupakan metode terbaik untuk menemukan cacat perangkat lunak.

Menurut (Song et al., 2011), penelitian prediksi cacat perangkat lunak berfokus pada 1) memperkirakan jumlah cacat software, 2) menemukan hubungan antar cacat, dan 3) mengklasifikasikan kerawanan cacat dari modul perangkat lunak yang dikelompokkan menjadi rawan atau tidak rawan cacat. Klasifikasi merupakan pendekatan yang paling banyak digunakan untuk prediksi cacat perangkat lunak (Lessmann et al., 2008). Dengan dideteksinya sejak awal cacat perangkat lunak dapat membantu pengembang menghasilkan perangkat lunak yang berkualitas.

Menurut (Hall et al., 2011), metode algoritma terbaik untuk prediksi cacat perangkat lunak adalah Logistic Regression atau Logistic Regresion. Logistic Regression merupakan pengklasifikasi probabilitas sederhana dengan penggunaan yang mudah dan nyaman karena tidak memerlukan perkiraan parameter yang rumit. Logistic Regression juga efektif digunakan pada dataset yang besar dan dapat menyajikan hasil klasifikasi kepada pengguna dengan mudah tanpa harus memiliki pengetahuan teknologi klasifikasi terlebih dahulu. Tetapi Logistic Regression berasumsi pada setiap atribut dataset sama penting dan tidak berhubungan satu sama lain. Logistic regresion merupakan model klasifikasi statistik probabilitas. Keuntungan Logistic Regression (LR) adalah algoritma ini telah dipelajari secara ekstensif (Hosmer, 2000). Sedangkan kelemahan LR adalah rentan terhadap underfitting dan memiliki akurasi yang rendah (Harrington, 2012).

LR merupakan pengklasifikasi linier yang telah terbukti menghasilkan klasifikasi yang powerful dengan statistik probabilitas dan menangani masalah klasifikasi multi kelas. Masalah utama pada algoritma LR adalah ketidakseimbangan kelas pada dataset berdimensi tinggi (Lin et al., 2007). Padahal untuk melakukan klasifikasi diperlukan data berupa dataset.

Data yang digunakan untuk mendeteksi modul perangkat lunak apakah rawan cacat atau tidak disebut software matrices (Chiş, 2008). Menurut (Khoshgoftaar et al., 2010), penggunaan data mining sangat efektif untuk mengidentifikasi modul perangkat lunak dari potensi rawan cacat jika diterapkan pada software metrics yang dikumpulkan selama proses pengembangan perangkat lunak. Software metric

yang digunakan selama pengembangan perangkat lunak disimpan dalam bentuk dataset.

Dataset NASA merupakan data metrik perangkat lunak yang paling banyak digunakan dalam pengembangan model prediksi cacat perangkat lunak, karena 62 penelitian dari 208 penelitian telah menggunakan dataset NASA (Witten et al., 2011). Dataset NASA tersedia secara online, sehingga para peneliti dapat menganalisa secara empiris untuk mengevaluasi penelitian sebelumnya. Penggunaan dataset NASA merupakan pilihan terbaik karena mudah diperoleh dan kinerja dari model yang digunakan menjadi mudah untuk dibandingkan dengan penelitian sebelumnya.

Software metrics berupa dataset yang umum digunakan untuk prediksi perangkat lunak bersifat tidak seimbang, karena umumnya cacat perangkat lunak ditemukan dalam persentase yang kecil dari modul perangkat lunak (Seiffert et al., 2014). Ketidakseimbangan data pada dataset yang rawan cacat dan tidak rawan cacat akan mengakibatkan model prediksi cacat perangkat lunak tidak efektif karena hasil prediksi cenderung menghasilkan kelas mayoritas.

Selain masalah data tidak seimbang, prediksi cacat perangkat lunak mengalami masalah pada redundant data, korelasi, fitur yang tidak relevan, missing sample, dan masalah ini dapat membuat dataset yang tidak seimbang tersebut menjadi sulit untuk memprediksi cacat pada modul perangkat lunak (Laradji et al., 2015). Seleksi fitur dapat menangani masalah redundant data dan fitur yang tidak relevan.

Greedy forward selection merupakan salah satu metode seleksi fitur yang sangat efisien, sederhana dan tidak seperti teknik seleksi fitur yang membutuhkan waktu lama dalam prosesnya. Dengan metode greedy forward selection, fitur yang dipilih hanya yang berkontribusi sehingga dapat meningkatkan performa klasifikasi (Laradji et al., 2015). Metode Seleksi fitur ini yang akan digunakan untuk yang menyelesaikan masalah redundant data.

Untuk masalah data tidak seimbang, salah satu cara menanganinya adalah dengan berbagai teknik resampling dan sintesis data untuk memperbaiki kecondongan distribusi kelas data latih (Saifudin & Wahono, 2015). Penggunaan teknik boosting dan bagging dapat meningkatkan pengklasifikasi.

Pada penelitian ini digunakan seleksi fitur greedy forward selection (GFS) untuk menangani masalah redundant data dan teknik bagging untuk menangani ketidakseimbangan kelas. Pengklasifikasi yang digunakan adalah algoritma logistic regression, dan menggunakan dataset NASA (National Aeronautics and Space Administration) MDP (Metrics Data Program).

1.2. Identifikasi Masalah

Berdasarkan uraian latar belakang terdapat permasalahan dalam prediksi cacat software sebagai berikut:

1. Redundant data dan fitur yang tidak relevan dapat menurunkan kinerja dari model prediksi cacat software.
2. Imbalance class atau kelas yang tidak seimbang pada dataset software metrics dapat menurunkan kinerja dari model prediksi cacat software.

1.3. Rumusan Masalah

Rumusan masalah dalam penelitian ini adalah meningkatkan kinerja klasifikasi Logistic Regression dengan menggunakan dataset yang tidak seimbang. Berikut rumusan masalah dalam penelitian ini sebagai berikut:

1. Seberapa meningkat model logistik regresi dengan greedy forward selection dalam mempengaruhi kinerja dari model prediksi cacat software sehingga dapat menangani permasalahan redundant data dan fitur-fitur yang tidak relevan?
2. Seberapa meningkat model logistik regresi dengan greedy forward selection dan bagging dalam mempengaruhi kinerja model prediksi cacat software sehingga dapat menangani ketidakseimbangan kelas?

1.4. Tujuan Penelitian

Tujuan penelitian ini adalah sebagai berikut:

1. Untuk meningkatkan kinerja logistik regresi dengan greedy forward selection dalam model prediksi cacat software dan dapat menangani permasalahan redundant data dan fitur-fitur yang tidak berkontribusi.

2. Untuk menangani ketidakseimbangan kelas dan meningkatkan kinerja dengan menggunakan model logistik regresi dengan greedy forward selection dan bagging pada prediksi cacat software, sehingga dilakukan integrasi antara metode bagging dan greedy forward selection yang menggunakan model logistic regression.

1.5. Manfaat Penelitian

Manfaat pada penelitian ini untuk penelitian selanjutnya adalah dapat berguna sebagai bahan referensi dalam penelitian prediksi cacat software dan dapat membantu pembuat software dalam membuat software yang berkualitas.

1.6. Ruang Lingkup

Ruang lingkup dalam penelitian ini adalah membatasi permasalahan dalam penelitian tentang software defect. Penelitian ini menggunakan dataset NASA (National Aeronautics and Space Administration) MDP (Metrics Data Program) repository sebagai software metrics. Aplikasi yang mendukung dalam penelitian ini menggunakan Rapidminer dan bahasa JAVA dan IDE (Integrated Development Environment) Netbeans. Algoritma klasifikasi menggunakan logistik regresi, sedangkan seleksi fitur dalam penelitian ini menggunakan greedy forward selection dan metode ensemble menggunakan metode bagging.

1.7. Sistematika Penulisan

Sistematika penulisan dalam penelitian ini terdiri dari lima bab sebagai berikut:

BAB I PENDAHULUAN

Bab ini membahas tentang latar belakang penulisan, identifikasi masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, ruang lingkup penelitian dan sistematika penulisan.

BAB II LANDASAN TEORI DAN KERANGKA PEMIKIRAN

Bab ini membahas tentang tinjauan pustaka, tinjauan studi, tinjauan objek penelitian. Tinjauan pustaka mengenai tentang sumber-sumber yang terkait dengan penelitian, tinjauan studi mengenai rangkuman penelitian yang terbaru dan berkaitan. Kemudian untuk obyek penelitian tentang uraian obyek yang diteliti dengan menggunakan kerangka pemikiran.

BAB III METODOLOGI PENELITIAN

Bab ini membahas metode penelitian, metode pengumpulan data, metode pengolahan data awal, model yang diusulkan dan pengujian model.

BAB IV HASIL DAN PEMBAHASAN

Bab ini membahas tentang hasil penelitian dan pembahasan serta implikasi dari penelitian yang dilakukan.

BAB V PENUTUP

Bab ini membahas tentang kesimpulan dan saran untuk penelitian selanjutnya

BAB II

LANDASAN TEORI

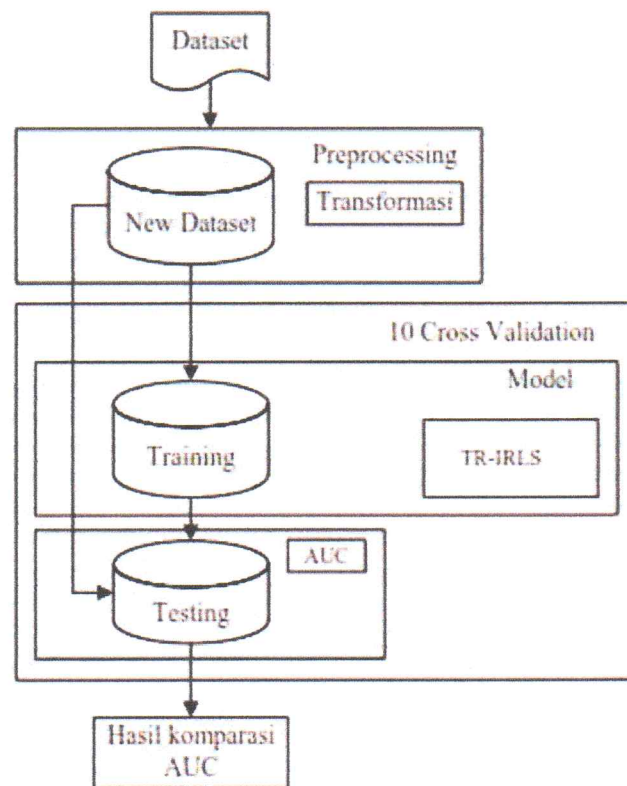
2.1. Tinjauan Studi

Tinjauan studi melingkupi penelitian terkait yang membahas ketidakseimbangan kelas dan redundant data pada LR. Kajian ini perlu dilakukan agar dapat mengetahui metode apa saja yang digunakan, data seperti apa yang diproses, dan metode apa yang dihasilkan.

2.1.1. Model Penelitian Komarek dan Moore (2005)

Penelitian yang dilakukan oleh Komarek dan Moore (Komarek & Moore, 2005), melakukan penelitian untuk meningkatkan akurasi prediksi model Logistic Regression dengan mengimplementasikan 3 metode yaitu: 1) Iteratively re-weighted least squares (IRLS), 2) Truncated Regularized IRLS (TR-IRLS), dan 3) Generic Likelihood Maximization. Logistic Regression merupakan algoritma klasifikasi dalam data mining yang memiliki performance tinggi. Dalam beberapa implementasi data mining, Logistic Regression dapat mengungguli algoritma lain seperti Logistic regression, Support Vector Mechine (SVM), dan K-Nearest Neighbor (KNN). Dalam penelitian ini menggunakan dataset yang dibagi dalam tiga kategori yaitu: 1) pendeteksian link (citeseer, imdb), 2) dataset penelitian (ds2, ds1, ds1.100, ds1.10) dan 3) klasifikasi teks (modapte.sub). Hasil penelitian menunjukkan matrik Area Under Curve (AUC) yaitu, TR-IRLS 0,94 citeseer, 0,98 imdb, 0.72 ds2, 0.94 ds1, 0,91 ds1.100 dan 0,84 ds1.10.

Model yang dilakukan (Komarek & Moore, 2005) dapat diilustrasikan pada Gambar 2.1.

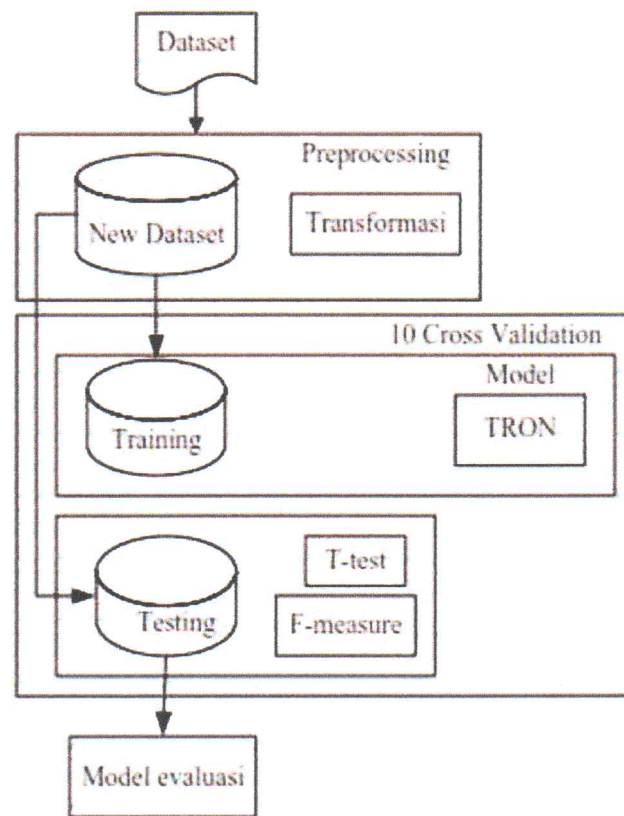


Gambar 2.1 Model Penelitian Komarek & Moore (2005)

2.1.2. Model Penelitian Lin, Weng dan Keerthi (2008)

Penelitian yang dilakukan oleh (Lin et al., 2007) melakukan penelitian dengan mengusulkan metode Trust Region Newton (TRON) pada LR. Penelitian ini menunjukkan hasil konvergensi yang lebih cepat dari pada metode quasi Newton. Metode ini telah diakui mampu menangani dataset berdimensi tinggi namun tidak sepenuhnya digunakan. Penelitian ini merupakan turunan penggunaan model Newton yang menggunakan dataset berdimensi tinggi yaitu a9a dengan 32561 instance, real-sim dengan 72309 instance, news20 dengan 19996 instance, yahoo-japan dengan 176203 instance, rev1 dengan 677399 instance dan yahoo-korea dengan 460554 instance. Penerapan metode trust Region Newton (TRON) memperlihatkan hasil komputasi 50% lebih cepat dibandingkan metode limited memory quasi Newton.

Model yang dilakukan (Lin et al., 2007) dapat dilihat pada Gambar 2.2.

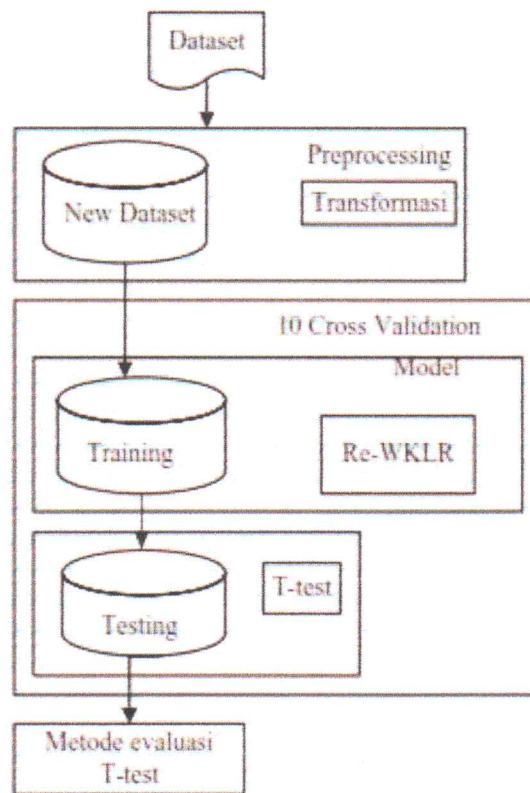


Gambar 2.2 Model Lin, Wen, & Keerthi 92008)

2.1.3. Model Maalouf dan Trafalis (2011)

Penelitian yang dilakukan oleh (Maalouf & Trafalis, 2011) melakukan penelitian dengan menerapkan model rare event re-wighted kernel logistic regression (RE-WKLR). Penelitian mampu menangani dataset yang seimbang dan peristiwa langka. Dataset yang digunakan adalah UCI Machine Learning dan kejadian nyata angin tornado. Performa dari metode RE-WKLR menunjukkan hasil klasifikasi yang lebih tinggi dari pada SVM. Secara keseluruhan hasil akurasi dari penelitian ini dengan menggunakan pengukuran komparasi paired t-test 0.017. Kesimpulan dari penelitian ini menunjukkan bahwa algoritma RE-WKLR sangat mudah diimplementasikan dan kuat dalam menangani data seimbang dan peristiwa langka.

Model yang dilakukan Maalouf dan Trafalis dapat dilihat pada Gambar 2.3.



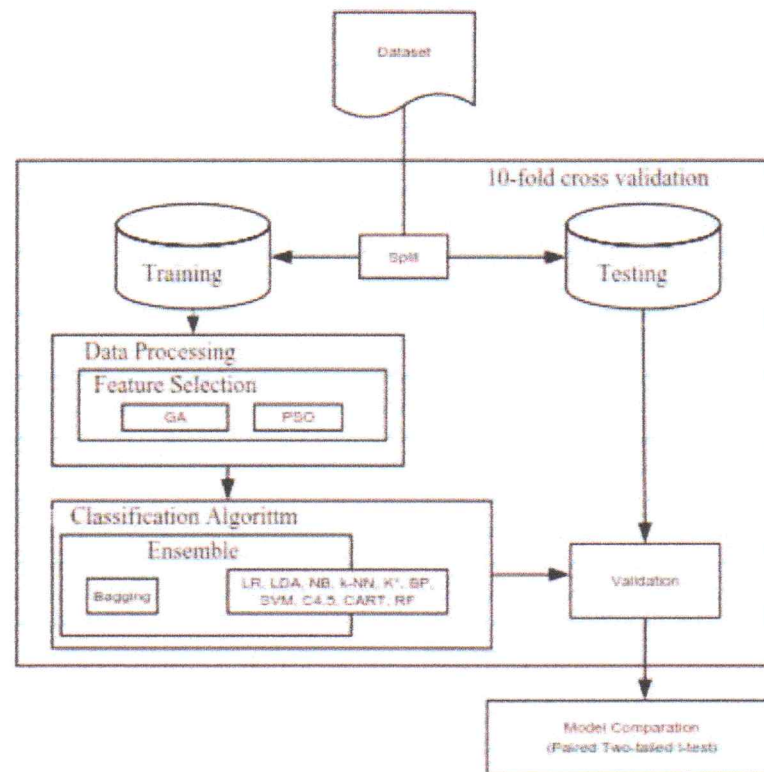
Gambar 2.3 Model Maalouf dan Trafalis (2011)

2.1.4. Model Penelitian Wahono, Suryana, dan Ahmad (2014)

Penelitian yang dilakukan oleh (Wahono et al., 2014) menerapkan optimasi metaheuristik untuk menemukan solusi optimal dalam seleksi fitur. Penelitian ini secara signifikan dapat mencari solusi berkualitas tinggi dengan jangka waktu yang wajar. Metode yang diusulkan dalam penelitian ini adalah optimasi metaheuristik (algoritma genetika dan particle swarm optimization (PSO)) dan teknik Bagging untuk meningkatkan kinerja prediksi cacat perangkat lunak. Bagging baik digunakan untuk model klasifikasi dan regresi. Bagging merupakan algoritma pembelajaran yang stabil pada dataset berdimensi tinggi dan atributnya masih mengandung noise. Dalam penelitian ini menggunakan 9 dataset NASA MDP dan 10 algoritma pengklasifikasi yang dikelompokkan dalam 5 tipe yaitu klasifikasi statistik tradisional (Logistic Regression (LR), Linear Discriminant Analysis (LDA), dan Logistic Regression (LR)), Nearest Neighbors (k-Nearest Neighbor (k-NN) dan K*), Neural Network (Back propagation (BP)), Support Vector Machine (SVM)), dan Decision Tree (C4.5, Classification and Regression Tree (CART), dan

Random Forest (RF)). Hasil penelitian ini menunjukkan bahwa metode yang diusulkan menunjukkan peningkatan kinerja model prediksi cacat perangkat lunak. Kesimpulan penelitian ini adalah tidak ada perbedaan signifikan dalam penggunaan optimasi Particle Swarm Optimizer (PSO) dan Genetic Algorithm (GA) saat digunakan pada seleksi fitur.

Model Wahono, Suryana, dan Ahmad dapat dilihat pada Gambar 2.4.



Gambar 2.4 Model Wahono, Suryana, dan Ahmad (2014)

2.1.5. Rangkuman Penelitian Terkait

Keempat penelitian terkait memiliki model prediksi yang berbeda pada penerapan algoritma Logistic Regression. Model Komarek dan Moore, menerapkan metode IRLS, TR-IRLS, dan Generic Likelihood Maximization dapat meningkatkan hasil prediksi di Logistic Regression (Komarek & Moore, 2005). Model TRON yang diterapkan oleh Lin, Weng, dan Keerthi mampu meningkatkan waktu komputasi yang lebih cepat dengan dataset skala besar (Lin et al., 2007). Model Maalouf dan Trafalis menerapkan metode RE-WKLR pada dataset yang skala menengah atau sedang (Maalouf & Trafalis, 2011). Model Wahono, Suryana,

dan Ahmad melakukan optimasi metaheuristik dan teknik bagging pada algoritma pengklasifikasi salah satunya adalah Logistic Regression (Wahono et al., 2014).

Tabel 2.1 menjelaskan secara singkat mengenai perbandingan penelitian terkait.

Peneliti	Tahun	Dataset	Metode	Akurasi
Komarek dan Moore	2005	- Citeseer - Imdb - ds2 - ds1 - ds1.100 - ds1.10 - modapte.sub	<i>Iteratively re-resampling least squares (IRLS)</i> <i>Truncated Regularized IRLS (TR-IRLS)</i> <i>Generic Likelihood Maximization</i>	AUC tertinggi 0.94 dan 0.72 terendah.
Lin, Weng, dan Keerthi	2008	- a9a - real-sim - news20 - yahoo-japan - rcv1 - yahoo-korea	<i>Trust Region Newton (TRON)</i>	Waktu TRON lebih cepat dibandingkan dengan LBFGS pada semua dataset
Maalouf dan Trafalis	2011	- UCI <i>Mechine Learning</i> - tornado	<i>rare event resampling kernel logistic regression (RE-WKLR)</i>	t-test 0.017
Wahono, Suryana, dan Ahmad	2015	NASA MDP	Optimasi metaheuristik (GA dan PSO) dan Bagging	AUC rata-rata meningkat 25,99% untuk GA dan 20,41% untuk PSO
Rakhmat Purnomo	2015	NASA MDP	Bagging, Greedy Forward Selection, Logistic Regresiion	Model Logistic Regresion dengan Seleksi Fitur Greedy Forward Selection dan Bagging dapat meningkatkan model prediksi caacat perangkat lunak dengan AUC 0,995 pada PC2

2.2. Tinjauan Pustaka

2.2.1. Prediksi Cacat Software

Cacat software atau perangkat lunak dapat didefinisikan sebagai cacat pada perangkat lunak seperti cacat pada dokumentasi, pada kode program, pada desain dan hal-hal lain yang menyebabkan kegagalan perangkat lunak. Cacat software dapat muncul pada berbagai tahap proses pengembangan (Pressman, 2015). Cacat software merupakan faktor penting yang mempengaruhi kualitas software tersebut. Kualitas software dapat ditingkatkan dengan mencegah munculnya cacat software melalui perbaikan modul yang mungkin menghasilkan cacat software pada proses pengembangan. Teknik pencegahan cacat perangkat lunak pertama kali diusulkan oleh IBM corporation dan dapat digunakan untuk mencegah munculnya cacat software di tahap lanjut pada proses pengembangan perangkat lunak.

Cacat software dapat menjadikan penyebab kegagalan software akibat aksi dari user yang menggunakan software pada saat tertentu. Bagi pengguna, cacat software adalah segala hal yang menyebabkan hasil dari software, yang diinginkan pengguna tidak terpenuhi. Contohnya pengguna ingin melakukan perhitungan yang tepat, namun hasil perhitungan yang dikeluarkan salah. Secara umum, kesalahan penulisan sintaks bukan merupakan cacat software, karena kesalahan sintaks dapat ditemukan menggunakan pengujian dasar.

Pada Tabel 2.2 menampilkan beberapa contoh umum cacat pada software. Harapan pengguna software tidak bisa dijalankan atau dieksekusi oleh software, maka masuk dalam kategori cacat software. Jika harapan pengguna dapat dieksekusi oleh software, maka cacat pada software bisa diartikan tidak ada

Tabel 2.1 Contoh Umum Cacat Software

Harapan Pengguna	Cacat software
Software memanggil metode API dan menjalankan sesuai dokumentasi dari software	Metode API gagal dijalankan karena perubahan API tidak teregister dipaket software
Dapat melakukan pengkopian file	File mengalami kerusakan selama proses copy

Software dapat membantu pengguna menyelesaikan tugas	Fungsi dari perangkat lunak hilang dan tidak mampu membantu menyelesaikan tugas pengguna
--	--

Sedangkan pada Tabel 2.3 menampilkan beberapa contoh spesifik dari cacat software. Pada contoh spesifik cacat software dapat dilihat bahwa, cacat software lebih spesifik kedalam sistem komputer.

Tabel 2.2 Contoh Nyata Cacat Software

Harapan Pengguna	Cacat software
Software mampu merespon aksi pengguna dengan cepat.	Software tidak mampu memberikan respon dengan cepat.
Software aman dari serangan hacker.	Hacker mampu mengeksploitasi software dan mengendalikannya.
Jika terjadi kesalahan yang fatal, software dapat menggunakan kode awal saat pengiriman, untuk mengurangi dampak dari kesalahan yang terjadi.	Software menggunakan kode awal, namun kode awal tidak bisa digunakan kembali.

Hasil cacat software yang terjadi karena beberapa jenis kesalahan antara lain (McDonald et al., 2008): 1) kesalahan manusia, ini biasa terjadi saat programmer mengetikkan kode program, karena beban kerja programmer yang terlalu berat sehingga terjadi kesalahan pengetikan kode program. 2) Kesalahan sistemik pada proses pengembangan, ini bisa terjadi karena ada kesalahan proses inputan atau keluaran yang digunakan pada proses lain. 3) Kesalahan dari penggunaan perangkat keras saat dilakukan pengembangan software. 4) Akibat salah pengertian kebutuhan pengguna dengan pengembang. Namun tidak semua kesalahan yang terjadi dapat menyebabkan cacat software, karena sebab-sebab terjadinya kesalahan dapat ditelusuri kesalahan yang menyebabkannya

2.2.2. Logistic Regresion

Logistic Regression (LR) dipresentasikan untuk prediksi dengan menggunakan lebih dari satu Linear Regression (Karsmakers et al., 2007). Sedangkan menurut (Witten et al., 2011), model LR merupakan probabilitas dari beberapa peristiwa, metode ini menggunakan fungsi linear untuk perhitungan prediksi pada beberapa variabel. LR menampilkan persamaan linear yang saling berhubungan antara beberapa variabel acak, dimana variabel yang tergantung adalah variabel yang berkelanjutan. Metode ini merupakan pengembangan metode Linear Regression yang menggunakan lebih dari satu variabel. Dalam beberapa kasus, variabel yang tergantung menunjuk kepada dua nilai atau kategori tidak dapat menggunakan Linear Regression, tetapi dapat melakukan pendekatan yang serupa yang dapat juga Multiple Linear LR.

Linier Regression sangat berguna untuk dataset linier dan aplikasi yang memadai (Komarek & Moore, 2005). Namun untuk dataset yang kontinue (0,1) atau dengan dataset bernilai biner (0,1) tidak mungkin tepat menggunakan linear regression. LR adalah model yang tepat digunakan untuk dataset bertipe seperti ini. LR menggunakan variabel yang sudah ditentukan atau variabel yang dikategorikan menjadi 2 variabel. Seperti pada prediksi sukses atau gagal, hidup atau mati, sakit atau tidak sakit, dan yang lainnya.

Notasi dalam LR dimana variabel $X \in \mathbb{R}^{N \times d}$ adalah data matrik dimana N adalah jumlah sampel, d adalah jumlah dari parameter atau attribut, dan variabel y yang merupakan variabel hasil bersifat biner. Untuk setiap baris dalam data sampel $X_i \in \mathbb{R}^d$ dimana X adalah vektor dari tiap baris data sampel, dimana $i = 1 \dots N$ dengan hasil dari $y_i = 1$ masuk dalam class positif dan sampel yang menghasilkan $y_i = 0$ adalah masuk dalam class negatif. Tujuan dari klasifikasi seluruh data sampel X_i adalah nilai positif atau negatif. Data sampel dapat menggunakan metode percobaan Bernouli dengan harapan hasil nilai $E(y_i)$ atau probabilitas p_i . Formula yang umum digunakan LR untuk model masing-masing data sampel X_i menganut pada fungsi dari (Hosmer, 2000) sebagai berikut :

$$E[y_i | x_i | \beta] = P_i = \frac{e^{(x_i \beta)}}{1 + e^{(x_i \beta)}} \dots\dots\dots (2.1)$$

Dimana β adalah parameter vektor dengan asumsi bahwa $x_{i0}=1$ sehingga β_0 adalah variabel konstan dengan nilai 1.

2.2.1. Dataset NASA

Dataset NASA yang telah ada untuk umum merupakan data metrik perangkat lunak yang sangat banyak digunakan untuk pengembangan model prediksi cacat perangkat lunak, karena 62 penelitian dari 208 penelitian telah menggunakan dataset NASA (Hall et al., 2011). Menurut (Wahono et al., 2014), sebanyak 64,79% penelitian prediksi cacat perangkat lunak menggunakan dataset publik. Hal ini dikarenakan mudah untuk didapatkan dan diuji kembali.

Dataset NASA MDP saat ini terdiri dari 13 dataset yang secara eksplisit ditujukan untuk penelitian metrik perangkat lunak (Gray et al., 2011). Metrik kode statis yang dicatat meliputi :

1. Ukuran LOC-count, merupakan jumlah baris kode program dan komentar.
2. Ukuran Helstead, merupakan perhitungan operan dan operator unik yang digunakan
3. Ukuran McCabe, merupakan kompleksitas dari cyclometric.

Pada dataset NASA yang masih asli, masih terdapat kegaduhan (noise) yaitu data yang tidak masuk akal atau tidak konsisten. Kegaduhan (noise) dapat disebut sebagai data yang salah (Liebchen & Shepperd, 2008). Untuk itu perlu dilakukan proses pembersihan data awal agar kegaduhan (noise) tidak ada. Prosedur pembersihan data dari kegaduhan (noise) adalah prosedur yang memakan waktu sangat padat dan merupakan suatu prosedur mutlak dalam proses pengambilan dataset (Witten et al., 2011). Diskripsi dari dataset NASA ditunjukkan pada Tabel 2.3. Spesifikasi dataset yang masih asli dapat dilihat pada Tabel 2.4. Spesifikasi dataset yang sudah dilakukan pembersihan kegaduhan (noise) pada Tabel 2.5 dan Tabel 2.6.

Tabel 2.3 Diskripsi Dataset NASA MDP

Dataset	Sistem	Bahasa	Total LOC
CM1	Instrumen pesawat ruang angkasa	C	20K
JM1	Sistem prediksi pendaratan <i>realtime</i>	C	315K
KC1	Manajemen penyimpanan data lapangan	C++	18K
KC3	Manajemen penyimpanan data lapangan	Java	18K
MC2	Sistem panduan video	C,C++	6K
PC1	Software penerbangan satelit yang mengorbit bumi	C	40K
PC2	Simulator dinamis untuk sistem kontrol perilaku	C	26K
PC3	Software penerbangan satelit yang mengorbit bumi	C	40K
PC4	Software penerbangan satelit yang mengorbit bumi	C	36K
PC5	Peningkatan keamanan sistem <i>upgrade</i> kokpit	C++	164K

Tabel 2.4 Spesifikasi Dataset NASA MDP

Dataset	Atribut	Modul	Cacat	Cacat (%)
CM1	41	523	48	9,18%
JM1	22	10878	2102	19,32%
KC1	22	2107	325	15,42%
KC3	41	458	43	9,39%
MC2	41	161	52	32,30%
PC1	41	1107	76	6,8%
PC2	41	5589	23	0,41%
PC3	41	1563	160	10,24%

PC4	41	1458	178	12,21%
PC5	40	17186	516	3,00%

Tabel 2.5 Spesifikasi Dataset Pembersihan Noise Kesatu

Dataset	Atribut	Modul	Cacat	Cacat (%)
CM1	38	344	42	12,21%
JM1	22	9591	1759	18,34%
KC1	22	2095	325	15,51%
KC3	40	200	36	18,00%
MC2	40	125	44	35,20%
PC1	38	759	61	8,30%
PC2	37	1586	16	1,07%
PC3	38	1125	140	12,56%
PC4	38	1399	178	12,91%
PC5	39	17001	503	2,96%

Tabel 2.6 Spesifikasi Dataset Pembersihan Noise kedua

Dataset	Atribut	Modul	Cacat	Cacat (%)
KC1	22	1162	294	25,30%
KC3	40	194	36	18,56%
MC2	40	124	44	35,48%
PC1	38	679	55	8,10%
PC2	37	722	16	2,22%
PC3	38	1053	130	12,35%
PC4	38	1270	176	13,86%
PC5	39	1694	458	27,04%

2.2.2. Teknik Validasi dan Evaluasi

Dalam pengujian dataset keseluruhan akurasi umumnya digunakan untuk mengevaluasi kinerja algoritma klasifikasi (Zhang & Wang, 2011). Namun untuk data yang tidak seimbang, kelas minoritas mendominasi akurasi yang mendalam, sehingga diperlukan alternatif matrik sebagai evaluasinya. Matrik evaluasi yang

tepat termasuk Area Under the ROC (Receiver Operating Characteristic) Curve (AUC), F-Measure, Geometric Mean (G-Mean), semua akurasi dan rata-rata akurasi untuk kelas minoritas. Validasi terhadap model yang diusulkan diperlukan beberapa pengujian menggunakan confusion matrix.

2.2.3.1. K-Fold Cross Validation

Pengujian yang dilakukan dalam penelitian ini adalah menggunakan metode cross validation. Cross validation adalah metode statistik yang digunakan untuk mengevaluasi dan membandingkan algoritma dengan membagi data menjadi dua segmen, segmen pertama digunakan sebagai data training dan segmen kedua sebagai data testing untuk memvalidasi model (Witten et al., 2011). Dalam cross validation segmen training dan segmen testing harus crossover sehingga setiap data memiliki kesempatan tervalidasi. Model cross validation yang terbagi dalam dua segment dapat dilihat pada Gambar 2.5

Split 1	Split 2	Split 3	Split 4	Split 5	Split 6	Split 7	Split 8	Split 9	Split 10
Training									Test
Training								Test	
Training							Test		
Training						Test			
Training					Test				
Training				Test					
Training			Test	Training					
Training		Test	Training						
	Test	Training							
Test	Training								

Gambar 2.5 10-Cross Fold Validation

2.2.3.2. Area Under the ROC (Receiver Operating Characteristic) Curve

Dalam penelitian ini diterapkan metode evaluasi menggunakan *Area Under Curve* (AUC) untuk mengukur hasil akurasi indikator dari performa model prediksi. Hasil akurasi dapat dilihat secara manual dengan dilakukan perbandingan klasifikasi menggunakan curva Receiver Operating Characteristic (ROC) dari hasil confusion matrix. ROC menghasilkan dua garis dengan bentuk true positives sebagai garis vertikal dan false positives sebagai garis horisontal (Vercellis, 2011). Kurva ROC adalah grafik antara sensitivitas (true positive rate), pada sumbu Y dengan 1-spesifitas pada sumbu X (false positive rate), curva ROC ini menggambarkan seakan-akan ada tarik-menawar antara sumbu Y dengan sumbu X

AUC dihitung berdasarkan rata-rata perkiraan bidang berbentuk trapesium untuk kurva yang dibuat oleh TPrate dan FPrate (Dubey, Zhou, Wang, Thompson, & Ye, 2014).

2.2.3.3. Uji Signifikan

Uji signifikansi dapat digunakan untuk membandingkan dua atau lebih model pengklasifikasi pada beberapa dataset. Evaluasi statistik hasil eksperimen dianggap sebagai bagian untuk validasi metode machine learning (Demšar, 2006). Tingkat signifikansi dapat diperoleh dengan menggunakan metode parametrik dan non-parametrik. Sebagian besar prosedur analisis statistik didasarkan pada asumsi bahwa beberapa bentuk model linier menggambarkan perilaku respon interval atau rasio variabel (Freund, J, & L, 2003). Ini berarti perilaku variabel respon diperkirakan oleh model linear dan kesimpulan yang dibuat mengacu pada parameter model. Karena fokus utama adalah pada parameter, metode statistik berdasarkan model linear sering disebut metode parametrik.

Uji t merupakan uji statistik untuk rata-rata dua populasi dari hasil eksperimen (Larose, 2005). Dua populasi sample uji t digunakan untuk memeriksa apakah dua sampel yang berbeda dan sering digunakan ketika varians dari dua distribusi normal tidak diketahui dan ketika percobaan menggunakan ukuran sampel yang kecil

2.2.3. Algoritma Greedy

Algoritma greedy merupakan algoritma yang menggunakan metaheuristic untuk menyelesaikan permasalahan dengan mengidentifikasi menggunakan lokal optimum dan mengarah pada global optimum dalam pendekatannya (Gorunescu, 2012).

Algoritma greedy merupakan metode yang paling populer untuk memecahkan persoalan optimasi (optimization problems) atau dapat disebut mencari solusi optimum. Algoritma greedy dapat diartikan sebagai rakus, tamak, loba dan lain-lain dengan menggunakan prinsip “take what you can get now”, algoritma ini membentuk solusi langkah per langkah (step by step). Pada setiap langkahnya terdapat banyak pilihan yang perlu dieksplorasi dan dari setiap langkah tersebut harus dibuat keputusan yang terbaik dalam menentukan pilihan. Dalam setiap langkah, algoritma ini membuat pilihan optimum lokal untuk dapat mengarah pada solusi optimum global.

Contoh pemecahan masalah penukaran uang menggunakan algoritma greedy:

Koin: 2, 3, 1, 4

Uang yang ditukar: 10

Solusi greedy: pada setiap langkah, memilih koin dengan nilai terbesar dari himpunan koin

Langkah 1: 2+4+4 (total 3 koin) Langkah 2: 2+2+3+3 (total 4 koin) Langkah 3: 2+1+1+4+2 (total 5 koin)

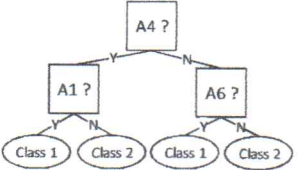
Langkah 3 merupakan solusi yang digunakan oleh greedy.

2.2.4. Greedy Forward Selection

Menurut (Gorunescu, 2012) algoritma greedy merupakan algoritma yang menggunakan metaheuristic untuk menyelesaikan permasalahan dengan mengidentifikasi lokal optimum dan pendekatannya mengarah pada global optimum. Algoritma greedy merupakan metode yang populer untuk memecahkan persoalan optimasi atau mencari solusi optimum.

Pada Gambar 2.1 Dijelaskan mengenai metode seleksi fitur dengan greedy untuk mencari atribut terbaik atau terburuk (Han et al., 2012).

Menurut (Han et al., 2012) metode seleksi atribut greedy terdapat beberapa pendekatan seperti : greedy forward selection, greedy backward elimination dan kombinasi forward selection dan backward elimination. Greedy forward selection merupakan salah satu pendekatan dalam algoritma greedy pada metode seleksi fitur.

Forward Selection	Backward Selection	Decision Tree Induction
Atribut awal : {A1, A2, A3, A4, A5, A6} Pengurangan atribut awal : {} $\Rightarrow$ {A1} $\Rightarrow$ {A1, A4} $\Rightarrow$ Atribut yang dikurangkan : {A1, A2, A6}	Atribut awal : {A1, A2, A3, A4, A5, A6} $\Rightarrow$ {A1 A3, A4, A5, A6} $\Rightarrow$ {A1, A4, A5, A6} $\Rightarrow$ Atribut yang dikurangkan : {A1, A4, A6}	Atribut awal {A1, A2, A3, A4, A5, A6}  Atribut yang dikurangkan : {A1, A3, A6}

Gambar 2.6 Metode Heuristik (Greedy) untuk Seleksi Atribut Subset

Algoritma greedy dengan seleksi subset atribut (Han, Kamber, & Pei, 2012) sebagai berikut untuk penjelasannya:

1. Stepwise forward selection

Prosedur dimulai dengan himpunan kosong dari atribut sebagai set yang dikurangi, atribut yang terbaik dari atribut asli ditentukan dan ditambahkan pada set yang kurang. Pada setiap iterasi berikutnya yang terbaik dari atribut asli yang tersisa ditambahkan ke set.

2. Stepwise backward elimination

Prosedur dimulai dengan full set atribut. Pada setiap langkah, teknik ini dapat menghilangkan atribut terburuk yang tersisa di dataset.

3. Kombinasi forward selection dan backward elimination

Stepwise forward selection dan backward elimination dapat dikombinasikan. Pada setiap langkah, prosedur memilih atribut terbaik dan menghilangkan yang terburuk dari atribut yang tersisa

Menurut (Laradji, Alshayeb, & Ghouti, 2015), metode greedy forward selection lebih baik dibandingkan metode feature selection yang lain dan greedy forward selection hanya memilih feature yang berkontribusi positif untuk meningkatkan kinerja klasifikasi

2.2.5. Bagging

Bagging merupakan kependekan dari bootstrap aggregating, yaitu metode belajar yang sederhana dan efektif. Menurut (Laradji et al., 2015), bagging merupakan salah satu teknik ensemble yang berhasil menangani dataset yang tidak seimbang meskipun tidak secara khusus dirancang untuk masalah tersebut. (Wahono dan Suryana, 2013) menjelaskan, teknik bagging merupakan salah satu teknik penggabungan yang pada klasifikasi memisahkan data training ke dalam beberapa data training baru dengan random sampling dan membangun model berbasis data training baru.

2.3. Feature Selection

Feature selection atau seleksi fitur dapat mengurangi dimensional pada data dan untuk meningkatkan kinerja dari mesin pembelajaran. Subset selection merupakan metode feature selection (seleksi fitur), subset selection adalah menemukan subset terbaik (fitur), subset yang terbaik mempunyai jumlah dimensi yang paling berkontribusi pada akurasi. Metode ini akan menghilangkan sisanya atau dimensi yang tidak penting menggunakan error function dan ini dapat menyelesaikan permasalahan pada regresi dan klasifikasi. Pemecahan masalah tersebut dapat diselesaikan dengan menggunakan metode heuristic untuk hasil yang masuk akal tapi bukan merupakan solusi yang optimal tapi masuk akal dalam waktu (Alpaydin, 2010).

Attribute subset selection adalah mengurangi ukuran dataset dengan menghilangkan atribut yang tidak relevan atau perulangan atribut (dimensi). Dalam machine learning, Attribute subset selection dapat disebut feature subset selection. Tujuan attribute subset selection adalah untuk menemukan satu set minimal atribut sehingga distribusi probabilitas yang dihasilkan dari kelas data sedekat mungkin dengan distribusi asli yang diperoleh dengan menggunakan semua atribut (Han, Kamber, & Pei, 2012).

Ada banyak metode yang digunakan dalam seleksi atribut atau fitur seperti: greedy forward selection, greedy backward elimination, forward selection, backward elimination, information gain dan lain-lain. Seleksi fitur dengan greedy forward selection dan greedy backward elimination akan dijelaskan pada sub bab

2.2.5. Seleksi fitur dengan forward selection adalah dimulai dengan tidak adanya atribut kemudian pada setiap langkah ditambahkan satu persatu yang memiliki kontribusi dan tidak menggunakan atribut yang tidak berkontribusi. Seleksi fitur dengan backward elimination adalah dengan seleksi mundur, maksudnya adalah semua atribut digunakan dan dihapus satu persatu hingga mendapatkan atribut yang menghasilkan kontribusi.

2.4. Information Gain

Atribut terbaik dan terburuk dapat ditentukan dengan menggunakan tes signifikansi statistik yang berasumsi bahwa atribut tidak saling berhubungan dengan satu sama lain (independen) (Han, Kamber, & Pei, 2012). Untuk langkah-langkah dalam menentukan evaluasi dalam pemilihan atribut dapat menggunakan information gain yang digunakan dalam pohon keputusan decision tree (Han, Kamber, & Pei, 2012).

Seleksi atribut menggunakan information gain adalah memilih gain tertinggi dan formula yang digunakan adalah sebagai berikut dimana langkah yang pertama adalah dengan mencari nilai entropy

2.5. Teknik Ensemble

Teknik ensemble merupakan teknik yang sukses untuk menangani dataset yang tidak seimbang meskipun tidak secara khusus dirancang untuk masalah data yang tidak seimbang (Laradji, Alshayeb, & Ghouti, 2015). Teknik bagging merupakan salah satu teknik ensemble dan teknik ini pada klasifikasi memisahkan data training ke dalam beberapa data training baru dengan random sampling dan membangun model berbasis data training baru (Wahono & Suryana, 2013).

2.6. Class Imbalance

Model prediksi cacat software dibentuk oleh kelas distribusi, dimana kelas distribusi dijelaskan dengan jumlah kasus pada masing-masing kelas di dataset training. Jika kasus satu kelas lebih besar dibanding kelas yang lain maka itu dapat disebut masalah class imbalance (kelas yang tidak seimbang), kelas yang lebih

banyak disebut dengan majority class (kelas mayor) dan kelas yang lebih sedikit disebut dengan minority class (kelas minor) (Arora, Tatarwal, & Saha, 2015).

2.7. Confusion Matrix

Confusion matrix merupakan alat untuk menganalisa seberapa baik kinerja dari pengklasifikasi dapat mengenali tupel dari kelas yang berbeda (Han, Kamber, & Pei, 2012). TP dan TN masuk dalam kategori benar, sedangkan FP dan FN masuk dalam kategori salah. Confusion matrix memberikan penilaian kinerja klasifikasi berdasarkan objek dengan benar atau salah (Gorunescu, 2011). Confusion matrix merupakan matrik 2 dimensi yang menggambarkan perbandingan antara hasil prediksi dengan kenyataan yang dapat ditunjukkan pada Tabel 2.7

Tabel 2.7 Confusion Matrix

Class		Actual	
		TRUE	FALSE
Prediction	TRUE	True Positive (TP)	False Negative (FN)
	FALSE	False Positive (FP)	True Negative (TN)

Keterangan:

TP = Nilai prediksi benardan nilai sebenarnya benar.

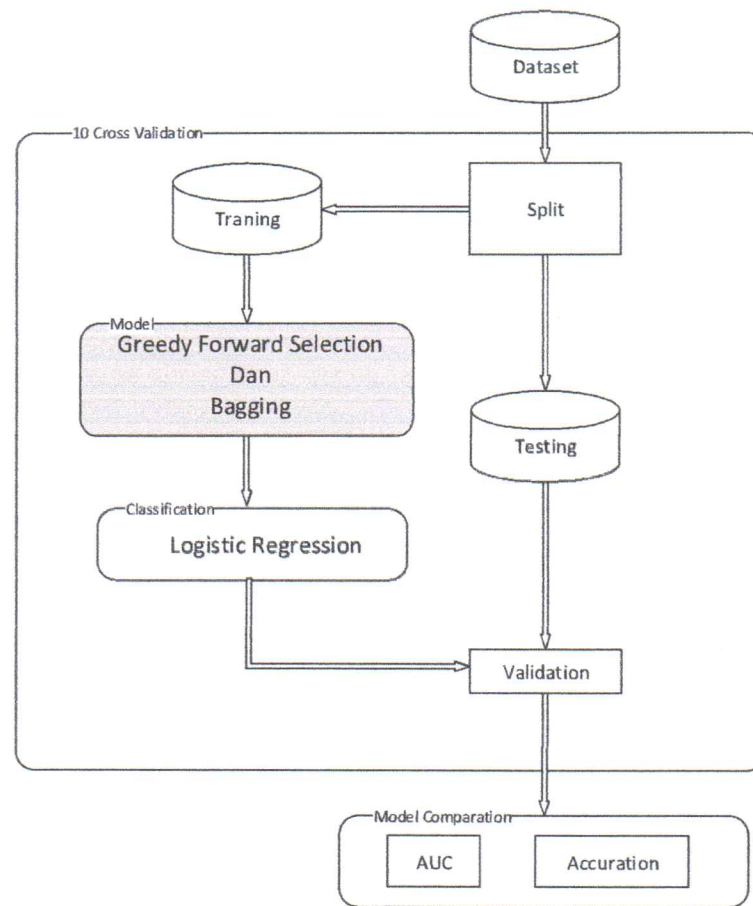
FN = Nilai prediksi salah dan nilai sebenarnya benar.

FP = Nilai prediksi benar dan nilai sebenarnya salah.

TN = Nilai prediksi salah dan nilai sebenarnya salah

2.8. Kerangka Pemikiran Penelitian

Gambar 2.7 merupakan kerangka pemikiran dalam penelitian ini yang terdiri dari beberapa bagian



Gambar 2.7 Kerangka Pemikiran

Gambar 2.7 terdapat bagian proposed method yang terdiri dari beberapa tahapan yaitu dataset NASA MDP yang akan diolah menggunakan seleksi fitur greedy forward selection, selanjutnya menggunakan metode ensemble bagging dimana metode bagging akan melakukan iterasi (factors) yang ditentukan dengan algoritma klasifikasi logistic regression. Validation dalam penelitian ini menggunakan 10 fold cross validation yaitu data dibagi menjadi 10 bagian, satu bagian menjadi data uji (testing) sedangkan bagian lain menjadi data latih (training). Proses validasi dilakukan berulang kali, mulai pada bagian pertama sampai pada bagian kesepuluh sehingga semua data dalam dataset diuji. Tujuan (objectives) dilakukan eksperimen ini adalah untuk menghasilkan kinerja dari model prediksi cacat software, kinerja tersebut dapat diukur (measurement) berdasarkan AUC, accuracy, sensitivity, f-measure, g-mean.

BAB III

METODE PENELITIAN

3.1. Perancangan Penelitian

Metode penelitian yang digunakan adalah pendekatan kuantitatif. Tujuan dari metode kuantitatif adalah untuk mengembangkan model, teori, dan hipotesis yang berkaitan dengan fenomena alam. Aspek kuantitatif adalah untuk menekankan pengukuran yang mendasar karena memberikan hubungan antara observasi dan formalisasi model, teori dan hipotesis (Berndtsson, Hanson, Olsson, & Lundell, 2008). Pada penelitian ini metode penelitian yang digunakan adalah eksperimen dan tahapan dalam penelitian ini dapat dilihat pada gambar 3.1.



Gambar 3.1 Tahapan Penelitian

Penelitian ini dirancang dengan tahapan sebagai berikut :

1. Pengumpulan data

Dalam tahap ini akan dijelaskan bagaimana data tersebut dikumpulkan dan darimana data tersebut diperoleh sehingga data tersebut dapat digunakan dalam eksperimen penelitian.

2. Pengolahan data awal

Pada tahap ini akan dijelaskan bagaimana pengolahan data untuk mendapatkan data yang sudah ditransformasikan agar sesuai dengan bentuk yang diinginkan.

3. Metode yang diusulkan

Setelah dilakukan tahap pengolahan data pada tahap sebelumnya, selanjutnya adalah menentukan model yang kemudian akan dilakukan pengujian terhadap data yang sudah ada

4. Eksperimen dan pengujian metode

Selanjutnya dalam tahapan ini akan dilakukan pengujian terhadap model yang diusulkan untuk mendapatkan hasil kinerja dari model yang diusulkan.

5. Evaluasi dan validasi hasil

Pada tahapan penelitian terakhir ini akan dilakukan evaluasi terhadap eksperimen dan pengujian pada model yang diusulkan sehingga dapat mengetahui hasil kinerja dalam penelitian ini.

3.2. Pengumpulan Data

Penelitian ini menggunakan dataset NASA MDP yang diperoleh dari PROMISE (Predictor Models in Software Engineering). Dataset ini mudah diperoleh dan tersedia untuk umum dan banyak digunakan oleh peneliti dalam bidang rekayasa perangkat lunak (Hall et al., 2011). Pada penelitian ini menggunakan dataset yang NASA MDP dengan 9 dataset cacat perangkat lunak dari NASA MDP (Wahono dan Suryana, 2013). Dataset NASA MDP dan atributnya ditunjukkan pada Tabel 3.1 (Wahono dan Suryana 2013).

Tabel 3.1 Dataset NASA MDP dan Atributnya

Code Attributes		NASA MDP Dataset									
		CMI	KC1	KC3	MC2	MW1	PC1	PC2	PC3	PC4	
LOC counts	LOC total	✓	✓	✓	✓	✓	✓	✓	✓	✓	
	LOC blank	✓	✓	✓	✓	✓	✓	✓	✓	✓	
	LOC code and comment	✓	✓	✓	✓	✓	✓	✓	✓	✓	
	LOC comments	✓	✓	✓	✓	✓	✓	✓	✓	✓	
	LOC executable	✓	✓	✓	✓	✓	✓	✓	✓	✓	
	number of lines	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Halstead	content	✓	✓	✓	✓	✓	✓	✓	✓	✓	
	difficulty	✓	✓	✓	✓	✓	✓	✓	✓	✓	
	effort	✓	✓	✓	✓	✓	✓	✓	✓	✓	
	error est	✓	✓	✓	✓	✓	✓	✓	✓	✓	
	length	✓	✓	✓	✓	✓	✓	✓	✓	✓	
	level	✓	✓	✓	✓	✓	✓	✓	✓	✓	
	prog time	✓	✓	✓	✓	✓	✓	✓	✓	✓	
	volume	✓	✓	✓	✓	✓	✓	✓	✓	✓	
	num operands	✓	✓	✓	✓	✓	✓	✓	✓	✓	
	num operators	✓	✓	✓	✓	✓	✓	✓	✓	✓	
	num unique operands	✓	✓	✓	✓	✓	✓	✓	✓	✓	
	num unique operators	✓	✓	✓	✓	✓	✓	✓	✓	✓	
	McCabe	cyclomatic complexity	✓	✓	✓	✓	✓	✓	✓	✓	✓
		cyclomatic density	✓		✓	✓	✓	✓	✓	✓	✓
design complexity		✓	✓	✓	✓	✓	✓	✓	✓	✓	
Misc.	essential complexity	✓	✓	✓	✓	✓	✓	✓	✓	✓	
	branch count	✓	✓	✓	✓	✓	✓	✓	✓	✓	
	call pairs	✓		✓	✓	✓	✓	✓	✓	✓	
	condition count	✓		✓	✓	✓	✓	✓	✓	✓	
	decision count	✓		✓	✓	✓	✓	✓	✓	✓	
	decision density	✓		✓	✓	✓	✓	✓	✓	✓	
	edge count	✓		✓	✓	✓	✓	✓	✓	✓	
	essential density	✓		✓	✓	✓	✓	✓	✓	✓	
	parameter count	✓		✓	✓	✓	✓	✓	✓	✓	
	maintenance severity	✓		✓	✓	✓	✓	✓	✓	✓	
	modified condition count	✓		✓	✓	✓	✓	✓	✓	✓	
	multiple condition count	✓		✓	✓	✓	✓	✓	✓	✓	
	global data complexity			✓	✓						
	global data density			✓	✓						
	normalized cyclo complx	✓		✓	✓	✓	✓	✓	✓	✓	
	percent comments	✓		✓	✓	✓	✓	✓	✓	✓	
	node count	✓		✓	✓	✓	✓	✓	✓	✓	
	Programming Language		C	C++	Java	C	C	C	C	C	C
	Number of Code Attributes		37	21	39	39	37	37	36	37	37
	Number of Modules		344	2096	200	127	264	759	1585	1125	1399
Number of fp Modules		42	325	36	44	27	61	16	140	178	
Percentage of fp Modules		12.21	15.51	18	34.65	10.23	8.04	1.01	12.44	12.72	

Tabel 3. 2 Deskripsi Atribut pada Dataset NASA MDP

Atribut Kode	Simbol	Deskripsi
LOC Count	LOC_total	Jumlah baris di modul
	LOC_blank	Jumlah baris kosong di modul
	LOC_code_and_comment	Jumlah baris yang mengandung kode dan komentar dalam sebuah modul
	LOC_comments	Jumlah baris komentar di modul
	LOC_executable	Jumlah baris kode yang dieksekusi di modul
	number_of_lines	Jumlah baris dalam sebuah modul
Halstead	content	Konten panjang halstead dari modul $\mu = \mu_1 + \mu_2$
	difficulty	Metrik kesulitan halstead dari modul $D = 1/L$
	effort	Metrik upaya halstead dari modul $E = V/L$
	error_est	Metrik estimasi kesalahan halstead di modul $B = E^{2/3}/1000$
	length	Metrik panjang halstead dari modul $N = N_1 + N_2$

	level	L	Metrik tingkat halstead di modul $L = (2^{\mu_1 + \mu_2}) / \mu_1 * N_2$
	prog_time	T	Metrik waktu program halstead di modul $T = E/18$
	volume	V	Metrik Volume Halstead dari modul $V = N * \log_2(\mu_1 + \mu_2)$
	num_operands	N1	Jumlah operan yang terkandung dalam modul
	num_operators	N2	Jumlah operator yang terkandung dalam modul
	num_unique_operands	μ_1	Jumlah operan yang unik yang terkandung dalam modul
	num_unique_operators	μ_2	Jumlah operator yang unik yang terkandung dalam modul
McCabe	cyclomatic_complexity	v(G)	Kompleksitas cyclomatic dari modul $v(G) = e - n + 2$
	cyclomatic_density		$v(G) / \text{NCSLOC}$
	design_complexity	iv(G)	Kompleksitas desain di modul
	essential_complexity	ev(G)	Kompleksitas penting dari modul
Misc.	branch_count		Metrik branch count
	call_pairs		Jumlah fungsi call dalam modul
	condition_count		Jumlah conditional dalam modul
	decision_count		Jumlah poin keputusan dalam modul
	decision_density		$\text{condition_count} / \text{decision_count}$
	edge_count		Jumlah edge yang ditemukan dalam modul yang diberikan dari satu modul ke yang lain
	essential_density		Essential density dihitung sebagai: $(\text{ev}(G) - 1) / (v(G) - 1)$
	parameter_count		Jumlah parameter untuk modul
	maintenance_severity		Maintenance severity dihitung sebagai: $\text{ev}(G) / v(G)$
	modified_condition_count		Pengaruh kondisi yang mempengaruhi hasil keputusan dengan hanya memvariasikan kondisi
	multiple_condition_count		Jumlah beberapa kondisi dalam sebuah modul
	global_data_complexity	gdv(G)	rasio kompleksitas cyclomatic pada struktur modul untuk menghitung parameter
	global_data_density		Global data density yang dihitung sebagai: $\text{gdv}(G) / v(G)$
	normalized_cyclo_complexity		$v(G) / \text{number_of_lines}$
	percent_comments		Persentase kode pada komentar
	node_count		Jumlah node yang ditemukan dalam modul

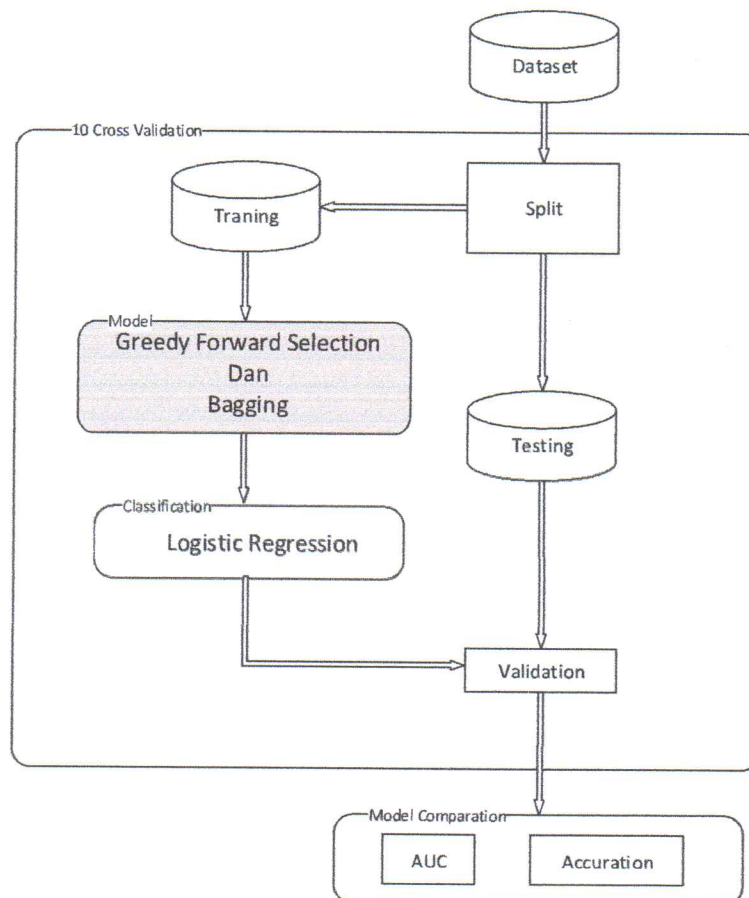
Pada Tabel 3.2 merupakan deskripsi dari masing-masing atribut yang terdapat pada dataset NASA MDP

3.3. Pengolahan Data Awal

Pada penelitian ini menggunakan 10 dataset NASA MDP yang sudah dilakukan pembersihan sehingga data sudah tidak mengandung noise. Dataset ini dapat langsung diolah dengan menggunakan model yang sudah ditentukan dan dalam penelitian ini akan menghasilkan nilai yang dapat menjadi pengukur seberapa baik model yang digunakan dalam penelitian ini..

3.4. Metode Yang Diusulkan

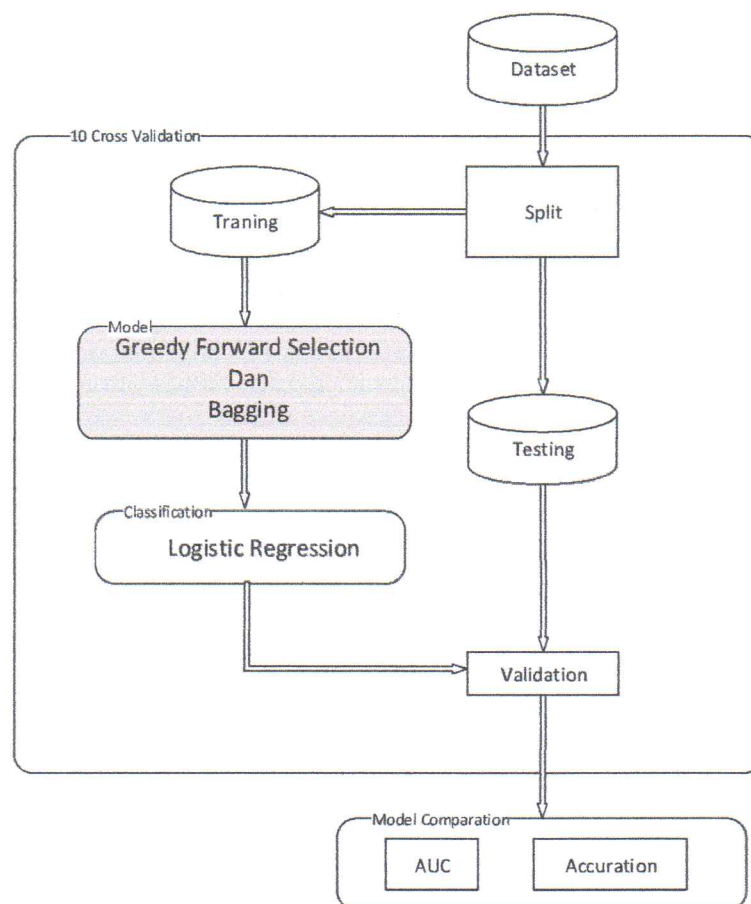
Metode yang diusulkan adalah penerapan Greedy Forward Selection (GFS) dan Bagging (BG) pada Logistic Regression (LR) pada prediksi cacat perangkat lunak. Gambar 3.1 menunjukkan pembagian dataset dengan metode 10-fold cross validation yaitu data latih dan data uji, kemudian data latih diproses menggunakan GFS. Kemudian dilakukan proses keseimbangan kelas dengan metode BG. Setelah itu dilakukan proses klasifikasi dengan LR. Hasil metode ini diukur dengan akurasi dan Area Under the ROC Curve (AUC).



3.5. Eksperimen dan Pengujian Metode

Tahapan eksperimen dan pengujian metode pada proses penelitian ini adalah :

1. Menyiapkan dataset untuk eksperimen.
2. Melakukan validasi dengan 10-fold cross validation.
3. Melakukan training dan testing terhadap model logistic regression dan mencatat hasilnya.
4. Mendesain arsitektur logistic regression dan bagging.
5. Melakukan training dan testing terhadap model logistic regression dan bagging dan mencatat hasilnya.
6. Mendesain arsitektur logistic regression dengan greedy forward dan bagging.
7. Melakukan training dan testing terhadap model logistic regression dengan greedy forward selection dan bagging, kemudian mencatat hasilnya.
8. Membandingkan nilai Akurasi dan AUC dari metode yang diusulkan, membuat grafik dan menentukan persentase kenaikannya.



Gambar 3.2 Metode yang diusulkan

3.6. Evaluasi dan Validasi Hasil

Evaluasi dan validasi metode yang diusulkan dimulai dari pembagian dataset dengan metode 10-fold cross validation yaitu membagi dataset menjadi dua bagian, bagian pertama digunakan sebagai data latih dan bagian kedua digunakan sebagai data uji untuk mem-validasi model (Witten et al., 2011). Setelah itu diterapkan tahapan evaluasi menggunakan Area Under Curve (AUC) untuk mengukur hasil akurasi indikator dari performa model prediksi. Hasil akurasi dapat dilihat dengan dilakukan perbandingan klasifikasi menggunakan curva Receiver Operating Characteristic (ROC) dari hasil confusion matrix. ROC menghasilkan dua garis dengan bentuk true positives sebagai garis vertikal dan false positive sebagai garis horisontal (Vercellis, 2009). Kurva ROC adalah grafik antara sensitivitas (true positive rate) pada sumbu Y dengan sumbu X. Pengukuran dengan confusion matrix dapat dilihat pada tabel 3.2.

Gambar 3.3 Confusion Matrix

Class		Actual	
		TRUE	FALSE
Prediction	TRUE	True Positive (TP)	False Negative (FN)
	FALSE	False Positive (FP)	True Negative (TN)

Menurut (Zhang dan Wang, 2011) untuk mengukur akurasi keseluruhan dan akurasi kelas minoritas pada data tidak seimbang menggunakan metrik yang tepat adalah AUC (Area Under the ROC Curve), F-Measure, G-Mean.

Dalam pengklasifikasi keakuratan dari tes diagnostik menggunakan Area under Curve (AUC) dapat dijelaskan melalui Tabel 3.3 (Gorunescu, 2012).

Tabel 3.3 Nilai AUC, Keterangan dan Simbol

Nilai AUC	Klasifikasi	Simbol
0,90 – 1,00	<i>Excellent classification</i>	
0,80 – 0,90	<i>Good classification</i>	
0,70 – 0,80	<i>Fair Classification</i>	

0,60 – 0,70	Poor classification	
< 0,60	Failure	

Menurut (Gorunescu, 2012), rumus-rumus yang digunakan untuk perhitungannya adalah

$$\begin{aligned}
 \text{Akurasi} &= \frac{TP + TN}{TP + TN + FP + FN} \\
 \text{Sensitivitas} = \text{Recall} = TP_{\text{rate}} &= \frac{TP}{TP + FN} \\
 \text{Spesifisitas} = TN_{\text{rate}} &= \frac{TN}{TN + FP} \\
 \text{FPrate} = \text{NPV} &= \frac{FP}{FP + TN} \\
 \text{Precision} = \text{PPV} &= \frac{TP}{TP + FP} \\
 \text{F-Measure} &= \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}} \\
 \text{G-Mean} &= \sqrt{\text{Sensitivitas} \times \text{Spesifisitas}}
 \end{aligned}$$

F-Measure adalah metrik evaluasi yang populer untuk masalah ketidakseimbangan. F-Measure mengkombinasikan recall/sensitivitas dan precision sehingga menghasilkan metrik yang efektif untuk pencarian kembali informasi dalam himpunan yang mengandung masalah ketidakseimbangan. Area Under the ROC (Receiver Operating Characteristic) Curve (AUROC atau AUC) adalah ukuran numerik untuk membedakan kinerja model dan menunjukkan seberapa sukses dan benar peringkat model mana yang lebih baik secara rata-rata (AtteLRerg, 2013). Menganalisa nilai AUC merupakan cara paling tepat untuk memilih metode mana yang terbaik.

BAB IV

HASIL DAN PEMBAHASAN

4.1. Hasil

Pada bagian ini akan dijelaskan mengenai hasil dari pengukuran kinerja pada model dan pengujian aplikasi untuk menunjukkan bahwa aplikasi yang dibuat telah sesuai dengan yang diharapkan. Untuk hasil yang ditampilkan menggunakan dataset NASA MDP dan akan dijelaskan cara pengolahan dataset tersebut

4.1.1. Hasil Pengukuran Logistic Regression

Eksperimen dilakukan menggunakan laptop HP Pavilion G series dengan prosesor Intel(R) Core (TM) i3-2310 CPU @ 2,10 GHz, memori RAM 4 GB, dan sistem operasi windows 7 ultimate 32 bit. Sedangkan perangkat lunak untuk eksperimen menggunakan rapid miner 6.

Eksperimen dimulai dengan menyiapkan dataset NASA MDP. Dataset yang digunakan yaitu yaitu CM1, KC1, KC3, MC2, MW1, PC1, PC2, PC3, PC3 beserta atributnya yaitu LOC counts, Halstead, McCabe, dan Misc.

Selanjutnya, dataset yang tersedia diterapkan pada algoritmya LR. Tabel 4.1 ditampilkan rekap pengukuran akurasi model LR. Informasi yang disajikan antara lain akurasi, sensitivity (recall), specificity, Positive Predictive Value (PPV) atau Precision, Negative Predictive Value (NPV) atau FPrate F-Measure, G-Mean dan AUC. Pada percobaan ini LR menghasilkan performa excellent pada Akurasi di PC1, PC2, dan PC4. Sedangkan pada AUC, performa excellent pada CM1, KC1, dan PC2.

Tabel 4.1 Hasil Pengukuran LR

Dataset	TP	TN	FP	FN	Accuracy	Recall	Specificity	PPV	NPV	F-Measure	G-Mean	AUC
CM1	299	2	3	40	0,875	0,882	0,400	0,990	0,070	0,550	0,594	0,906
KC1	1757	48	14	277	0,861	0,864	0,774	0,992	0,048	0,817	0,818	0,908
KC3	158	4	6	32	0,810	0,832	0,400	0,963	0,158	0,540	0,577	0,837
MC2	79	14	4	30	0,732	0,725	0,778	0,952	0,118	0,750	0,751	0,804
MW1	231	6	6	21	0,898	0,917	0,500	0,975	0,222	0,647	0,677	0,847
PC1	691	9	7	52	0,922	0,930	0,563	0,990	0,119	0,701	0,723	0,906
PC2	1566	0	3	16	0,988	0,990	0,000	0,998	0,158	0,000	0,000	0,916
PC3	973	11	12	129	0,875	0,883	0,478	0,988	0,085	0,620	0,650	0,899
PC4	79	1195	99	26	0,911	0,752	0,923	0,444	0,792	0,829	0,834	0,894
Rata - rata					0,875	0,864	0,535	0,921	0,197	0,606	0,625	0,880

4.1.2. Hasil Pengukuran Logistic Regression dan Greedy Forward Selection

Percobaan berikutnya dengan menerapkan Greedy Forward Selection (GFS) pada Logistic Regression (LR). Tabel 4.2 menunjukkan hasil percobaannya. Informasi yang disajikan antara lain akurasi, sensitivity (recall), specificity, Positive Predictive Value (PPV) atau Precision, Negative Predictive Value (NPV) atau FPrate F-Measure, G-Mean dan AUC. Pada percobaan ini menghasilkan performa excellent pada Akurasi di MW1, PC2, dan PC4. Sedangkan pada AUC, performa excellent pada CM1, KC1, MW1, PC1, PC, dan PC3.

Tabel 4.2 Hasil Pengukuran LR + GFS

Dataset	TP	TN	FP	FN	Accuracy	Recall	Specificity	PPV	NPV	F-Measure	G-Mean	AUC
CM1	301	2	1	40	0,881	0,883	0,667	0,997	0,024	0,760	0,767	0,929
KC1	1755	45	16	280	0,859	0,862	0,738	0,991	0,054	0,795	0,798	0,904
KC3	162	6	2	30	0,840	0,844	0,750	0,988	0,063	0,794	0,795	0,891
MC2	81	14	2	30	0,748	0,730	0,875	0,976	0,063	0,796	0,799	0,834
MW1	237	6	0	21	0,920	0,919	1,000	1,000	0,000	0,958	0,958	0,959
PC1	687	6	1	55	0,925	0,926	0,857	0,999	0,018	0,890	0,891	0,954
PC2	1569	0	0	16	0,990	0,990	0,000	1,000	0,000	0,000	0,000	0,995
PC3	984	5	1	135	0,879	0,879	0,833	0,999	0,007	0,856	0,856	0,936
PC4	73	1200	105	21	0,910	0,777	0,920	0,410	0,833	0,842	0,845	0,881
Rata - rata					0,875	0,875	0,875	0,875	0,875	0,875	0,875	0,920

4.1.3. Hasil Pengukuran Logistic Regression dan Bagging

Percobaan selanjutnya dengan menggunakan teknik Bagging (BG) pada Logistic Regression (LR). Tabel 4.3 menunjukkan hasil percobaannya. Informasi yang disajikan antara lain akurasi, sensitivity (recall), specificity, Positive Predictive Value (PPV) atau Precision, Negative Predictive Value (NPV) atau FPrate F-Measure, G-Mean dan AUC. Pada percobaan ini menghasilkan performa excellent pada Akurasi di MW1, PC1, PC2, dan PC4. Sedangkan pada AUC, performa excellent pada CM1, KC1, PC1, PC2, PC3, dan PC4.

Tabel 4.3 Hasil Pengukuran LR + BG

Dataset	TP	TN	FP	FN	Accuracy	Recall	Specificity	PPV	NPV	F-Measure	G-Mean	AUC
CM1	299	2	3	40	0,875	0,882	0,400	0,990	0,070	0,550	0,594	0,906
KC1	1756	47	15	278	0,860	0,863	0,758	0,992	0,051	0,807	0,809	0,906
KC3	159	4	5	32	0,815	0,832	0,444	0,970	0,135	0,579	0,608	0,849
MC2	78	15	5	29	0,732	0,729	0,750	0,940	0,147	0,739	0,739	0,791
MW1	232	7	5	20	0,905	0,921	0,583	0,979	0,200	0,714	0,733	0,860
PC1	693	9	5	52	0,925	0,930	0,643	0,993	0,068	0,760	0,773	0,921
PC2	1566	0	3	16	0,988	0,990	0,000	0,998	0,158	0,000	0,000	0,916
PC3	974	11	11	129	0,876	0,883	0,500	0,989	0,079	0,638	0,664	0,902
PC4	84	1192	94	29	0,912	0,743	0,927	0,472	0,764	0,825	0,830	0,912
Rata - rata					0,875	0,875	0,875	0,875	0,875	0,875	0,875	0,886

4.1.4. Hasil Pengukuran Logistic Regression, Greedy Forward Selection, dan Bagging

Percobaan berikutnya (propose method) dengan menggabungkan Greedy Forward Selection (GFS) dan Bagging (BG) pada Logistic Regression (LR). Tabel 4.4 menunjukkan hasil percobaannya. Informasi yang disajikan antara lain akurasi, sensitivity (recall), spesificity, Positive Predictive Value (PPV) atau Precision, Negative Predictive Value (NPV) atau FPrate F-Measure, G-Mean dan AUC. Pada percobaan ini menghasilkan performa excellent pada Akurasi di MW1, PC1, PC2, dan PC4. Sedangkan pada AUC, performa excellent pada CM1, KC1, MW1, PC1, PC2, dan PC3.

Tabel 4.4 Hasil Pengukuran LR + GFS + Bagging

Dataset	TP	TN	FP	FN	Accuracy	Recall	Specificity	PPV	NPV	F-Measure	G-Mean	AUC
CM1	302	2	0	40	0,884	0,883	1,000	1,000	0,000	0,938	0,940	0,942
KC1	1759	47	12	278	0,862	0,864	0,797	0,993	0,041	0,829	0,829	0,911
KC3	163	6	1	30	0,845	0,845	0,857	0,994	0,032	0,851	0,851	0,906
MC2	82	14	1	30	0,756	0,732	0,933	0,988	0,052	0,821	0,827	0,850
MW1	236	6	1	21	0,917	0,918	0,857	0,996	0,045	0,887	0,887	0,936
PC1	697	10	1	51	0,931	0,932	0,909	0,999	0,019	0,920	0,920	0,956
PC2	1569	0	0	16	0,990	0,990	0,000	1,000	0,000	0,000	0,000	0,995
PC3	984	5	1	135	0,879	0,879	0,833	0,999	0,007	0,856	0,856	0,936
PC4	69	1206	109	15	0,911	0,821	0,917	0,388	0,879	0,867	0,868	0,885
Rata - rata					0,875	0,875	0,875	0,875	0,875	0,875	0,875	0,924

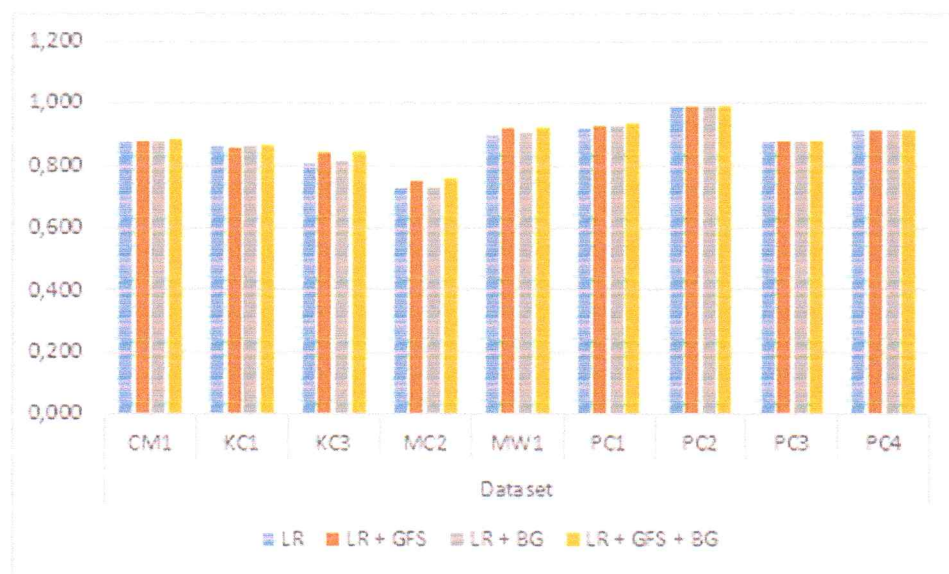
4.1.5. Hasil Pengukuran Akurasi

Terakhir, untuk melihat perbandingan hasil percobaan, Tabel 4.5 dan Gambar 4.1 menampilkan rekap pengukuran akurasi metode dan grafiknya metode Logistic Regression (LR), Logistic Regression (LR) dan Greedy Forward Selection (GFS), Logistic Regression (LR) dan Bagging (BG), serta Logistic Regression (LR),

Greedy Forward Selection (GFS) dan Bagging (BG) pada prediksi cacat perangkat lunak. Hasilnya metode LR+GFS+BG excellent pada dataset MW1, PC1, PC2, dan PC4. Hasil good pada dataset CM1, KC1, KC3, dan PC3. Hasil fair pada dataset MC2.

Tabel 4.5 Hasil Pengukuran Akurasi LR, LR + GFS, LR + BG, dan LR + GFS + BG

Metode	Dataset								
	CM1	KC1	KC3	MC2	MW1	PC1	PC2	PC3	PC4
LR	0,875	0,861	0,810	0,732	0,898	0,922	0,988	0,875	0,911
LR + GFS	0,881	0,859	0,840	0,748	0,920	0,925	0,990	0,879	0,910
LR + BG	0,875	0,860	0,815	0,732	0,905	0,925	0,988	0,876	0,912
LR + GFS + BG	0,884	0,862	0,845	0,756	0,917	0,931	0,990	0,879	0,911



Gambar 4.1 Perbandingan Akurasi dari metode LR, LR+GFS, LR+BG, dan LR+GFS+BG

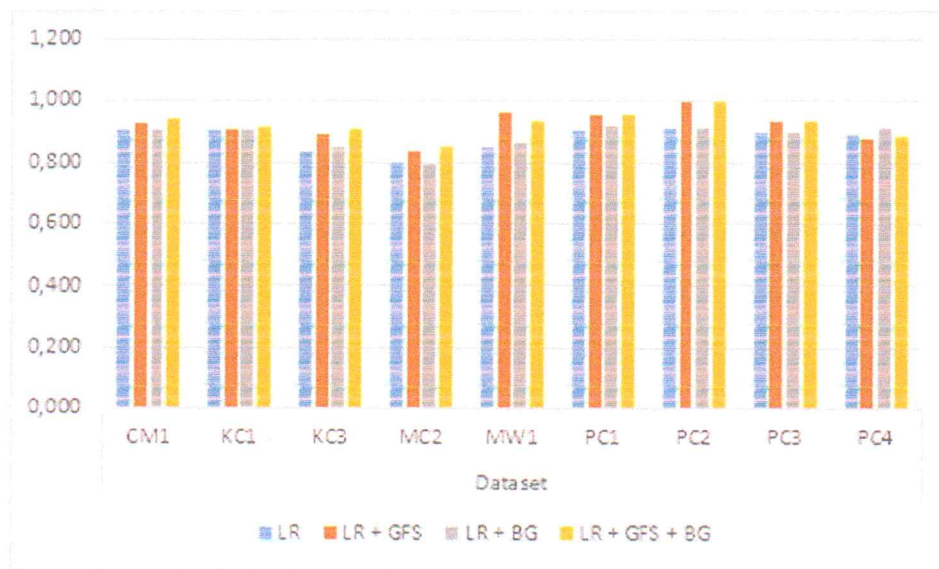
4.1.6. Hasil Pengukuran AUC

Sedangkan pada Tabel 4.6 dan Gambar 4.2 menunjukkan hasil perbandingan nilai Area Under Curve (AUC) dan grafiknya metode Logistic Regression (LR), Logistic Regression (LR) dan Greedy Forward Selection (GFS), Logistic Regression (LR) dan Bagging (BG), serta Logistic Regression (LR), Greedy Forward Selection (GFS) dan Bagging (BG) pada prediski cacat perangkat lunak.

Hasilnya metode LR+GFS+BG excellent pada CM1, KC1, KC3, MW1, PC1, PC2, dan PC3. Hasil good pada MC2, tetapi tidak terjadi peningkatan pada PC4.

Tabel 4.6 Hasil Pengukuran AUC LR, LR + GFS, LR + BG, dan LR + GFS + BG

Metode	Dataset								
	CM1	KC1	KC3	MC2	MW1	PC1	PC2	PC3	PC4
LR	0,906	0,908	0,837	0,804	0,847	0,906	0,916	0,899	0,894
LR + GFS	0,929	0,904	0,891	0,834	0,959	0,954	0,995	0,936	0,881
LR + BG	0,906	0,906	0,849	0,791	0,860	0,921	0,916	0,902	0,912
LR + GFS + BG	0,942	0,911	0,906	0,850	0,936	0,956	0,995	0,936	0,885



Gambar 4.2 Perbandingan AUC dari metode LR, LR+GFS, LR+BG, dan LR+GFS+BG

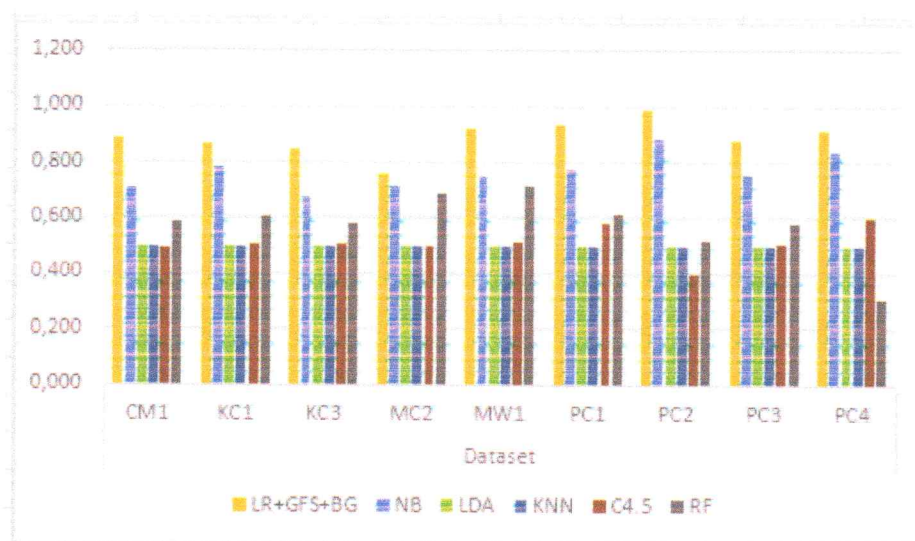
Berdasarkan hasil percobaan diatas maka dapat dilihat bahwa metode LR+GFS+BG mampu menangani ketidakseimbangan kelas dan redundant data pada logistic regresion dengan menghasilkan nilai akurasi dan AUC lebih tinggi dibandingkan dengan metode LR yang tidak menggunakan greedy forward selection dan bagging.

4.1.7. Hasil Perbandingan AUC Model Prediksi Cacat Perangkat Lunak

Kami juga membandingkan pengklasifikasi yang lainnya yaitu Logistic Regression (LR), Linear Discriminant Analys (LDA), K-Nearest Neighbour (KNN), C4.5, dan Random Forest (RF). Hasil rekap nilai AUC ditampilkan pada Tabel 4.7 dan Gambar 4.3.

Tabel 4.7 Perbandingan AUC Model Prediksi Cacat Perangkat Lunak

Model	Dataset								
	CM1	KC1	KC3	MC2	MW1	PC1	PC2	PC3	PC4
LR+GFS+BG	0,884	0,862	0,845	0,756	0,917	0,931	0,990	0,879	0,911
NB	0,708	0,786	0,677	0,712	0,752	0,775	0,885	0,756	0,840
LDA	0,500	0,500	0,500	0,500	0,500	0,500	0,500	0,500	0,500
KNN	0,500	0,500	0,500	0,500	0,500	0,500	0,500	0,500	0,500
C4.5	0,493	0,506	0,506	0,499	0,512	0,579	0,398	0,508	0,602
RF	0,583	0,610	0,578	0,686	0,717	0,616	0,521	0,581	0,305



Gambar 4.3 Perbandingan Nilai AUC Model Prediksi Cacat Perangkat Lunak

Pada Tabel 4.7 dan Gambar 4.3 terlihat, metode yang diusulkan (proposed method) LR+GFS+BG mendapatkan performa excellent pada PC1, PC2, dan PC4. Nilai AUC good pada MC1, KC1, KC3, dan PC3. Dan nilai AUC cukup pada MC2. Jika dibandingkan dengan pengklasifikasi yang lain metode LR+GFS+BG mengungguli secara signifikan.

4.2. Pembahasan

4.2.1. Perbandingan Kinerja Logistic Regression dengan Model yang diusulkan

Pada bagian ini akan dilakukan perbandingan kinerja Logistic Regression (LR), Logistic Regression dengan Bagging (LR+BG), Logistic Regression dengan Greedy forward selection (LR+GFS) dan Logistic Regression dengan Greedy forward selection dan Bagging (LG+GFS+BG).

4.2.1.1. Perbandingan Kinerja Logistic Regression dengan Logistic Regression dan Bagging

1. Akurasi Logistic Regression dan Logistic Regression dengan Baging

Dilakukan uji beda dengan menggunakan metode statistik untuk menguji hipotesa pada model Logistic Regression (LR) dengan Logistic Regression dan Bagging (LR+BG).

H0: Tidak ada perbedaan nilai rata-rata akurasi LR dengan LR+BG.

H1: Ada perbedaan antara nilai rata-rata akurasi LR dengan LR+BG.

Pada Tabel 4.8 dapat dilihat perbedaan nilai akurasi antara LR dengan LR+BG.

Tabel 4.8 Perbandingan Akurasi LR dan LR+BG

Dataset	LR	LR+BG
CM1	82.56	82.25%
JM1	81.31	81.32%
KC1	82.25	82.30%
KC3	79.00	79.00%
MC2	72.37	72.37%
PC1	88.54	88.54%
PC2	95.59	95.52%
PC3	33.53	31.91%
PC4	87.42	87.63%
PC5	96.63	96.63%

Setelah dibandingkan hasil nilai akurasi dari LR dan LR+BG, kemudian dilakukan analisa dengan menggunakan uji t-Test Paired Two Sample for Means dengan hasil yang dapat dilihat pada Tabel 4.9.

Tabel 4.9 Hasil Uji Beda Statistik Akurasi LR dan LR+BG

	LR	LR+BG
Mean	0.799198	0.7974
Variance	0.03199476	0.03368772
Observations	10	10
Pearson Correlation	0.999913985	
Hypothesized Mean Difference	0	
df	9	
t Stat	1.042599302	
P(T<=t) one-tail	0.162164641	
t Critical one-tail	1.833112933	
P(T<=t) two-tail	0.324329282	
t Critical two-tail	2.262157163	

Pada Tabel 4.9 dapat diketahui bahwa nilai rata-rata akurasi dari model LR lebih tinggi dibandingkan model LR+BG sebesar 0.799198. Dalam uji beda statistik nilai alpha ditentukan sebesar 0.05, jika nilai p lebih kecil dibanding alpha ($p < 0.05$) maka H_0 ditolak dan H_1 diterima sehingga disimpulkan ada perbedaan yang signifikan antara model yang dibandingkan, akan tetapi jika nilai p lebih besar dibanding alpha ($p > 0.05$) maka H_0 diterima dan H_1 ditolak sehingga disimpulkan tidak ada perbedaan yang signifikan antara model yang dibandingkan

Pada Tabel 4.9 dapat dilihat bahwa nilai $P(T \leq t)$ adalah 0.3 dan ini menunjukkan bahwa nilai p lebih besar dibandingkan nilai alpha ($0.3 > 0.05$) sehingga hipotesis H_0 diterima dan H_1 ditolak.

2. Sensitifitas logistik regression dan logistic regression dengan bagging

Dilakukan uji beda dengan menggunakan metode statistik untuk menguji hipotesa dengan menggunakan model Logistic regression (LR) dibandingkan dengan Logistic regression dan Bagging (LR+BG) yang dapat dilihat pada Tabel 4.10.

H_0 : Tidak ada perbedaan nilai rata-rata sensitifitas LR dengan LR+BG.

H_1 : Ada perbedaan antara nilai rata-rata sensitifitas LR dengan LR+BG.

Tabel 4.10 Perbandingan Sensitifitas LR dan LR+BG

Dataset	LR	LR+BG
CM1	33.33	30.95%
JM1	94.86	94.87%
KC1	37.54	37.54%
KC3	36.11	36.11%
MC2	35.83	36.36%
PC1	36.07	36.07%
PC2	18.75	18.75%
PC3	90.00	90.71%
PC4	94.10	94.27%
PC5	44.73	44.93%

Selanjutnya dilakukan analisa dengan menggunakan uji *t-Test Paired Two Sample for Means* dengan hasil yang dapat dilihat pada Tabel 4.11.

Tabel 4.11 Hasil Uji beda Statistik Sensitifitas LR dan LR+BG

	<i>LR</i>	<i>LR+BG</i>
Mean	0.52132	0.5205
Variance	0.083744454	0.08535145
Observations	10	10
Pearson Correlation	0.99961968	
Hypothesized Mean Difference	0	
df	9	
t Stat	0.283345795	
P(T<=t) one-tail	0.391660737	
t Critical one-tail	1.833112933	
P(T<=t) two-tail	0.783321474	
t Critical two-tail	2.262157163	

3. F-measure logistic regression dan logistic regression dengan bagging

Uji beda dilakukan pada f-measure dengan menggunakan metode statistik untuk menguji hipotesa pada model Logistic regression (LR) dengan Logistic regression dan Bagging (LR+BG).

H0: Tidak ada perbedaan nilai rata-rata f-measure LR dengan LR+BG.

H1: Ada perbedaan antara nilai rata-rata f-measure LR dengan LR+BG.

Perbandingan nilai f-measure antara LR dengan LR+BG dapat dilihat pada Tabel 4.12.

Tabel 4.12 Perbandingan F-measure LR dan LR+BG

Datas e t	LR	LR+BG
CM1	0.32	0.30
JM1	0.89	0.89
KC1	0.40	0.40
KC3	0.38	0.38
MC2	0.47	0.48
PC1	0.34	0.34
PC2	0.08	0.08
PC3	0.25	0.25
PC4	0.93	0.93
PC5	0.89	0.44

Selanjutnya hasil nilai f-measure dari LR dan LR+BG dilakukan analisa dengan menggunakan uji t-Test Paired Two Sample for Means dengan hasil yang ditunjukkan pada Tabel 4.13.

Tabel 4.13 Hasil Uji beda Statistik F-measure LR dan LR+BG

	<i>LR</i>	<i>LR+BG</i>
Mean	0.494303511	0.44818956
Variance	0.090029216	0.07212896
Observations	10	10
Pearson Correlation	0.884346875	
Hypothesized Mean Difference	0	
df	9	
t Stat	1.040798421	
P(T<=t) one-tail	0.162560153	
t Critical one-tail	1.833112933	
P(T<=t) two-tail	0.325120305	
t Critical two-tail	2.262157163	

Pada Tabel 4.13 dapat diketahui bahwa nilai rata-rata f-measure dari model LR lebih tinggi dibandingkan model LR+BG sebesar 0.494303511. Dalam uji beda statistik nilai alpha ditentukan sebesar 0.05, jika nilai p lebih kecil dibanding alpha ($p < 0.05$) maka H_0 ditolak dan H_1 diterima sehingga disimpulkan ada perbedaan yang signifikan antara model yang dibandingkan, akan tetapi jika nilai p lebih besar

dibanding alpha ($p > 0.05$) maka H_0 diterima dan H_1 ditolak sehingga disimpulkan tidak ada perbedaan yang signifikan antara model yang dibandingkan.

Pada Tabel 4.13 dapat dilihat bahwa nilai $P(T \leq t)$ adalah 0.3 dan ini menunjukkan bahwa nilai p lebih besar dibandingkan nilai alpha ($0.3 > 0.05$) sehingga hipotesis H_0 diterima dan H_1 ditolak. Hipotesis H_0 diterima berarti dapat disimpulkan bahwa tidak ada perbedaan signifikan antara model LR dan model LR+BG.

4. G-mean logistic regression dan logistic regression dengan bagging

Uji beda dilakukan pada g-mean dengan menggunakan metode statistik untuk menguji hipotesa model Logistic regression (LR) dan Logistic regression dengan Bagging (LR+BG).

H_0 : Tidak ada perbedaan nilai rata-rata g-mean LR dengan LR+BG.

H_1 : Ada perbedaan antara nilai rata-rata g-mean LR dengan LR+BG.

Perbandingan nilai f-mean antara LR dengan LR+BG dapat dilihat pada Tabel 4.14.

Tabel 4.14 Perbandingan G-mean LR dan LR+BG

Datas et	LR	LR+BG
CM1	0.55	0.53
JM1	0.95	0.95
KC1	0.58	0.58
KC3	0.57	0.57
MC2	0.57	0.58
PC1	0.58	0.58
PC2	0.43	0.42
PC3	0.48	0.46
PC4	0.94	0.94
PC5	0.66	0.66

Selanjutnya hasil nilai g-mean dari LR dan LR+BG dilakukan analisa dengan menggunakan uji t-Test Paired Two Sample for Means dengan hasil yang ditunjukkan pada Tabel 4.15.

Tabel 4.15 Hasil Uji beda Statistik G-mean LR dan LR+BG

	<i>LR</i>	<i>LR+BG</i>
Mean	0.630158868	0.62733422
Variance	0.031443819	0.03258219
Observations	10	10
Pearson Correlation	0.999083578	
Hypothesized Mean Difference	0	
df	9	
t Stat	1.076989224	
P(T<=t) one-tail	0.154752559	
t Critical one-tail	1.833112933	
P(T<=t) two-tail	0.309505118	
t Critical two-tail	2.262157163	

Pada Tabel 4.15 dapat diketahui bahwa nilai rata-rata g-mean dari model LR lebih tinggi dibandingkan model LR+BG sebesar 0.630158868. Dalam uji beda statistik nilai alpha ditentukan sebesar 0.05, jika nilai p lebih kecil dibanding alpha ($p < 0.05$) maka H_0 ditolak dan H_1 diterima sehingga disimpulkan ada perbedaan yang signifikan antara model yang dibandingkan, akan tetapi jika nilai p lebih besar dibanding alpha ($p > 0.05$) maka H_0 diterima dan H_1 ditolak sehingga disimpulkan tidak ada perbedaan yang signifikan antara model yang dibandingkan.

Pada Tabel 4.15 dapat dilihat bahwa nilai $P(T \leq t)$ adalah 0.3 dan ini menunjukkan bahwa nilai p lebih besar dibandingkan nilai alpha ($0.3 > 0.05$) sehingga hipotesis H_0 diterima dan H_1 ditolak. Hipotesis H_0 diterima berarti dapat disimpulkan bahwa tidak ada perbedaan signifikan antara model LR dan model LR+BG.

5. AUC LR dan LR+BG

Uji beda dilakukan pada AUC dengan menggunakan metode statistik untuk menguji hipotesa pada model Logistic regression (LR) dan Logistic regression dengan Bagging (LR+BG).

H_0 : Tidak ada perbedaan nilai rata-rata AUC LR dengan LR+BG.

H_1 : Ada perbedaan antara nilai rata-rata AUC LR dengan LR+BG.

Hasil perbandingan nilai AUC antara model LR dengan LR+BG dapat ditunjukkan pada Tabel 4.16.

Tabel 4.16 Perbandingan AUC LR dan LR+BG

Datase t	LR	LR+BG
CM1	0.708	0.717
JM1	0.683	0.685
KC1	0.786	0.789
KC3	0.677	0.673
MC2	0.712	0.721
PC1	0.775	0.794
PC2	0.885	0.877
PC3	0.756	0.756
PC4	0.84	0.838
PC5	0.94	0.942

Selanjutnya hasil nilai AUC dari LR dan LR+BG dilakukan analisa dengan menggunakan uji t-Test Paired Two Sample for Means dengan hasil yang ditunjukkan pada Tabel 4.17.

Tabel 4.17 Hasil Uji beda Statistik AUC LR dan LR+BG

	<i>LR</i>	<i>LR+BG</i>
Mean	0.7762	0.7792
Variance	0.007838178	0.00753195
Observations	10	10
Pearson Correlation	0.996337459	
Hypothesized Mean Difference	0	
df	9	
t Stat	-1.231606241	
P(T<=t) one-tail	0.12465284	
t Critical one-tail	1.833112933	
P(T<=t) two-tail	0.249305679	
t Critical two-tail	2.262157163	

Pada Tabel 4.17 dapat diketahui bahwa nilai rata-rata AUC dari model LR+BG lebih tinggi dibandingkan model LR sebesar 0.7792. Dalam uji beda statistik nilai alpha ditentukan sebesar 0.05, jika nilai p lebih kecil dibanding alpha ($p < 0.05$) maka H_0 ditolak dan H_1 diterima sehingga disimpulkan ada perbedaan yang signifikan antara model yang dibandingkan, akan tetapi jika nilai p lebih besar dibandingkan nilai alpha ($p > 0.05$) maka H_0 diterima dan H_1 ditolak sehingga disimpulkan tidak ada perbedaan yang signifikan antara model yang dibandingkan.

Pada Tabel 4.17 dapat dilihat bahwa nilai $P(T \leq t)$ adalah 0.2 dan ini menunjukkan bahwa nilai p lebih besar dibandingkan nilai alpha ($0.2 > 0.05$) sehingga hipotesis H_0 diterima dan H_1 ditolak. Hipotesis H_0 diterima berarti dapat disimpulkan bahwa tidak ada perbedaan signifikan antara model LR dan model LR+BG.

4.2.1.2. Perbandingan Kinerja Logistic Regression dengan Logistic Regression dan Greedy Forward Selection

1. Akurasi logistic regression dengan logistic regression dan greedy forward selection

Uji beda dilakukan dengan menggunakan metode statistik untuk menguji hipotesa untuk model Logistic regression (LR) dengan Logistic regression dan Greedy forward selection (LR+GFS).

H_0 : Tidak ada perbedaan nilai rata-rata akurasi LR dengan LR+GFS.

H_1 : Ada perbedaan antara nilai rata-rata akurasi LR dengan LR+GFS.

Pada Tabel 4.18 dapat dilihat perbedaan nilai akurasi antara LR dengan LR+GFS.

Tabel 4.18 Perbandingan Akurasi LR dan LR+GFS

Datas et	LR	LR+GFS
CM1	82.56	86.94%
JM1	81.31	81.48%
KC1	82.25	83.64%
KC3	79.00	82.00%
MC2	72.37	75.64%
PC1	88.54	91.84%
PC2	95.59	96.66%
PC3	33.53	84.00%
PC4	87.42	83.99%
PC5	96.63	97.21%

Setelah dibandingkan hasil nilai akurasi dari LR dan LR+GFS, kemudian dilakukan analisa dengan menggunakan uji t-Test Paired Two Sample for Means dengan hasil yang dapat dilihat pada Tabel 4.19.

Tabel 4. 19 Hasil Uji beda Statistik Akurasi LR dan LR+GFS

	<i>LR</i>	<i>LR+GFS</i>
Mean	0.799198	0.8634
Variance	0.03199476	0.00478336
Observations	10	10
Pearson Correlation	0.49847968	
Hypothesized Mean Difference	0	
df	9	
t Stat	-1.298542527	
P(T<=t) one-tail	0.113192617	
t Critical one-tail	1.833112933	
P(T<=t) two-tail	0.226385234	
t Critical two-tail	2.262157163	

Pada Tabel 4.19 dapat diketahui bahwa nilai rata-rata akurasi dari model LR+GFS lebih tinggi dibandingkan model LR sebesar 0.8634. Dalam uji beda statistik nilai alpha ditentukan sebesar 0.05, jika nilai p lebih kecil dibanding alpha ($p < 0.05$) maka H_0 ditolak dan H_1 diterima sehingga disimpulkan ada perbedaan yang signifikan antara model yang dibandingkan, akan tetapi jika nilai p lebih besar dibandingkan nilai alpha ($p > 0.05$) maka H_0 diterima dan H_1 ditolak sehingga disimpulkan tidak ada perbedaan yang signifikan antara model yang dibandingkan.

Pada Tabel 4.19 dapat dilihat bahwa nilai $P(T \leq t)$ adalah 0.2 dan ini menunjukkan bahwa nilai p lebih besar dibandingkan nilai alpha ($0.2 > 0.05$) sehingga hipotesis H_0 diterima dan H_1 ditolak. Hipotesis H_0 diterima berarti dapat disimpulkan bahwa tidak ada perbedaan signifikan antara model LR dan model LR+GFS.

2. Sensitifitas logistic regression dengan logistic regression dan greedy forward selection

Dilakukan uji beda dengan menggunakan metode statistik untuk menguji hipotesa untuk model Logistic regression (LR) dengan Logistic regression dan Greedy forward selection (LR+GFS).

H_0 : Tidak ada perbedaan nilai rata-rata akurasi LR dengan LR+GFS.

H_1 : Ada perbedaan antara nilai rata-rata akurasi LR dengan LR+GFS.

Pada Tabel 4.20 dapat dilihat perbedaan nilai sensitifitas antara LR dengan LR+GFS.

Tabel 4.20 Perbandingan Sensitifitas LR dan LR+GFS

Datase t	LR	LR+GFS
CM1	33.33	34.67%
JM1	94.86	95.25%
KC1	37.54	33.54%
KC3	36.11	33.33%
MC2	35.83	43.18%
PC1	36.07	36.07%
PC2	18.75	6.25%
PC3	90.00	37.86%
PC4	94.10	87.39%
PC5	44.73	33.80%

Selanjutnya hasil nilai sensitifitas dari LR dan LR+GFS dilakukan analisa dengan menggunakan uji t-Test Paired Two Sample for Means dengan hasil yang dapat dilihat pada Tabel 4.20.

Tabel 4.21 Hasil Uji Beda Statistik Sensitifitas LR dan LR+GFS

	<i>LR</i>	<i>LR+GFS</i>
Mean	0.52132	0.4413
Variance	0.083744454	0.0716638
Observations	10	10
Pearson Correlation	0.825233928	
Hypothesized	Mean	0
df		9
t Stat	1.523824931	
P(T<=t) one-tail	0.080943717	
t Critical one-tail	1.833112933	
P(T<=t) two-tail	0.161887434	
t Critical two-tail	2.262157163	

Pada Tabel 4.21 dapat diketahui bahwa nilai rata-rata sensitifitas dari model LR lebih tinggi dibandingkan model LR+GFS sebesar 0.52132. Dalam uji beda statistik nilai alpha ditentukan sebesar 0.05, jika nilai p lebih kecil dibanding alpha ($p < 0.05$) maka H_0 ditolak dan H_1 diterima sehingga disimpulkan ada perbedaan

yang signifikan antara model yang dibandingkan, akan tetapi jika nilai p lebih besar dibandingkan nilai α ($p > 0.05$) maka H_0 diterima dan H_1 ditolak sehingga disimpulkan tidak ada perbedaan yang signifikan antara model yang dibandingkan.

Pada Tabel 4.21 dapat dilihat bahwa nilai $P(T \leq t)$ adalah 0.1 dan ini menunjukkan bahwa nilai p lebih besar dibandingkan nilai α ($0.1 > 0.05$) sehingga hipotesis H_0 diterima dan H_1 ditolak. Hipotesis H_0 diterima berarti dapat disimpulkan bahwa tidak ada perbedaan signifikan antara model LR dan model LR+GFS.

3. F-measure logistic regression dengan logistic regression dan greedy forward selection

Uji beda dilakukan pada f-measure dengan menggunakan metode statistik untuk menguji hipotesa untuk model Logistic regression (LR) dengan Logistic regression dan Greedy forward selection (LR+GFS).

H_0 : Tidak ada perbedaan nilai rata-rata f-measure LR dengan LR+GFS.

H_1 : Ada perbedaan antara nilai rata-rata f-measure LR dengan LR+GFS.

Tabel 4.22 adalah perbandingan nilai f-measure antara LR dengan LR+GFS.

Tabel 4.22 Perbandingan F-measure LR dan LR+GFS

Datas e t	LR	LR+GFS
CM1	0.32	0.39
JM1	0.89	0.89
KC1	0.40	0.39
KC3	0.38	0.40
MC2	0.47	0.55
PC1	0.34	0.42
PC2	0.08	0.04
PC3	0.25	0.37
PC4	0.93	0.91
PC5	0.89	0.42

Selanjutnya hasil nilai f-measure dari LR dan LR+GFS dilakukan analisa dengan menggunakan uji t-Test Paired Two Sample for Means dengan hasil yang ditunjukkan pada Tabel 4.23.

Tabel 4.23 Hasil Uji Beda Statistik F-measure LR dan LR+GFS

		<i>LR</i>	<i>LR+GFS</i>
Mean		0.494303511	0.47694200
Variance		0.090029216	0.06622062
Observations		10	10
Pearson Correlation		0.831306335	
Hypothesized	Mean	0	
df		9	
t Stat		0.328835703	
P(T<=t) one-tail		0.374901231	
t Critical one-tail		1.833112933	
P(T<=t) two-tail		0.749802462	
t Critical two-tail		2.262157163	

Pada Tabel 4.23 dapat diketahui bahwa nilai rata-rata f-measure dari model LR lebih tinggi dibandingkan model LR+GFS sebesar 0.494303511. Dalam uji beda statistik nilai alpha ditentukan sebesar 0.05, jika nilai p lebih kecil dibanding alpha ($p < 0.05$) maka H_0 ditolak dan H_1 diterima sehingga disimpulkan ada perbedaan yang signifikan antara model yang dibandingkan, akan tetapi jika nilai p lebih besar dibandingkan nilai alpha ($p > 0.05$) maka H_0 diterima dan H_1 ditolak sehingga disimpulkan tidak ada perbedaan yang signifikan antara model yang dibandingkan.

Pada Tabel 4.23 dapat dilihat bahwa nilai $P(T \leq t)$ adalah 0.7 dan ini menunjukkan bahwa nilai p lebih besar dibandingkan nilai alpha ($0.7 > 0.05$) sehingga hipotesis H_0 diterima dan H_1 ditolak.

4. G-mean logistic regression dengan logistic regression dan greedy forward selection

Uji beda dilakukan pada g-mean dengan menggunakan metode statistik untuk menguji hipotesa pada Logistic regression (LR) dengan Logistic regression dan Greedy forward selection (LR+GFS).

H_0 : Tidak ada perbedaan nilai rata-rata g-mean LR dengan LR+GFS.

H_1 : Ada perbedaan antara nilai rata-rata g-mean LR dengan LR+GFS.

Tabel 4.24 adalah perbandingan nilai g-mean antara LR dengan LR+GFS dan Tabel 4.25 merupakan uji t-Test Paired Two Sample for Means.

Tabel 4.24 Perbandingan G-mean LR dan LR+GFS

Datas e t	LR	LR+GFS
CM1	0.55	0.57
JM1	0.95	0.44
KC1	0.58	0.56
KC3	0.57	0.56
MC2	0.57	0.63
PC1	0.58	0.59
PC2	0.43	0.25
PC3	0.48	0.59
PC4	0.94	0.73
PC5	0.66	0.58

Tabel 4.25 Hasil Uji beda Statistik G-mean LR dan LR+GFS

	LR	LR+GFS
Mean	0.630158868	0.54866767
Variance	0.031443819	0.01636942
Observations	10	10
Pearson Correlation	0.332211922	
Hypothesized	Mean	0
df		9
t Stat	1.424218339	
P(T<=t) one-tail	0.094059286	
t Critical one-tail	1.833112933	
P(T<=t) two-tail	0.188118572	
t Critical two-tail	2.262157163	

Pada Tabel 4.25 dapat diketahui bahwa nilai rata-rata g-mean dari model LR lebih tinggi dibandingkan model LR+GFS sebesar 0.630158868. Dalam uji beda statistik nilai alpha ditentukan sebesar 0.05, jika nilai p lebih kecil dibanding alpha ($p < 0.05$) maka H_0 ditolak dan H_1 diterima sehingga disimpulkan ada perbedaan yang signifikan antara model yang dibandingkan, akan tetapi jika nilai p lebih besar dibandingkan nilai alpha ($p > 0.05$) maka H_0 diterima dan H_1 ditolak sehingga disimpulkan tidak ada perbedaan yang signifikan antara model yang dibandingkan.

Pada Tabel 4.25 dapat dilihat bahwa nilai $P(T \leq t)$ adalah 0.1 dan ini menunjukkan bahwa nilai p lebih besar dibandingkan nilai alpha ($0.1 > 0.05$) sehingga hipotesis H_0 diterima dan H_1 ditolak. Hipotesis H_0 diterima berarti dapat

disimpulkan bahwa tidak ada perbedaan signifikan antara model LR dan model LR+GFS.

5. AUC logistic regression dengan logistic regression dan greedy forward selection

Uji beda dilakukan pada AUC dengan menggunakan metode statistik untuk menguji hipotesa pada Logistic regression (LR) dengan Logistic regression dan Greedy forward selection (LR+GFS).

H0: Tidak ada perbedaan nilai rata-rata AUC LR dengan LR+GFS.

H1: Ada perbedaan antara nilai rata-rata AUC LR dengan LR+GFS.

Tabel 4.26 merupakan hasil dari perbandingan nilai AUC antara LR dengan LR+GFS.

Tabel 4.26 Perbandingan AUC LR dan LR+GFS

Datas et	LR	LR+GFS
CM1	0.708	0.769
JM1	0.683	0.653
KC1	0.786	0.788
KC3	0.677	0.765
MC2	0.712	0.775
PC1	0.775	0.81
PC2	0.885	0.894
PC3	0.756	0.784
PC4	0.84	0.86
PC5	0.94	0.941

Selanjutnya hasil nilai AUC dari LR dan LR+GFS dilakukan analisa dengan menggunakan uji t-Test Paired Two Sample for Means dengan hasil yang ditunjukkan pada Tabel 4.27.

Tabel 4.27 Hasil Uji Beda Statistik AUC LR dan LR+GFS

	LR	LR+GFS
Mean	0.7762	0.8039
Variance	0.007838178	0.00634276
Observations	10	10
Pearson Correlation	0.917706055	
Hypothesized	Mean	0

df	9
t Stat	-2.487968197
P(T<=t) one-tail	0.017268476
t Critical one-tail	1.833112933
P(T<=t) two-tail	0.034536952
t Critical two-tail	2.262157163

Pada Tabel 4.27 dapat diketahui bahwa nilai rata-rata AUC dari model LR+GFS lebih tinggi dibandingkan model LR sebesar 0.8039. Dalam uji beda statistik nilai alpha ditentukan sebesar 0.05, jika nilai p lebih kecil dibanding alpha ($p < 0.05$) maka H_0 ditolak dan H_1 diterima sehingga disimpulkan ada perbedaan yang signifikan antara model yang dibandingkan, akan tetapi jika nilai p lebih besar dibandingkan nilai alpha ($p > 0.05$) maka H_0 diterima dan H_1 ditolak sehingga disimpulkan tidak ada perbedaan yang signifikan antara model yang dibandingkan.

Pada Tabel 4.27 dapat dilihat bahwa nilai $P(T \leq t)$ adalah 0.03 dan ini menunjukkan bahwa nilai p lebih kecil dibandingkan nilai alpha ($0.03 < 0.05$) sehingga hipotesis H_0 ditolak dan H_1 diterima. Hipotesis H_1 diterima berarti dapat disimpulkan bahwa ada perbedaan signifikan antara model LR dan model LR+GFS sehingga model LR+GFS membuat peningkatan ketika dibandingkan dengan model LR.

4.2.1.3. Perbandingan Kinerja Logistic Regression dan Logistic Regression dengan Greedy Forward Selection dan Bagging

1. Akurasi logistic regression dan logistic regression dengan greedy forward selection dan bagging

Dilakukan uji beda dengan menggunakan metode statistik untuk menguji hipotesa pada Logistic regression (LR) dan Logistic regression dengan Greedy forward selection dan bagging (LR+GFS+BG).

H_0 : Tidak ada perbedaan nilai rata-rata akurasi LR dengan LR+GFS+BG.

H_1 : Ada perbedaan antara nilai rata-rata akurasi LR dengan LR+GFS+BG.

Pada Tabel 4.28 dapat dilihat perbedaan nilai akurasi antara LR dengan LR+GFS+BG.

Tabel 4. 28 Perbandingan Akurasi LR dan LR+GFS+BG

Dataset	LR	LR+GFS+BG
CM1	82.56%	85.76%
JM1	81.31%	81.58%
KC1	82.25%	83.35%
KC3	79.00%	83.50%
MC2	72.37%	74.87%
PC1	88.54%	91.18%
PC2	95.59%	96.91%
PC3	33.53%	83.56%
PC4	87.42%	86.77%
PC5	96.63%	97.35%

Setelah dibandingkan hasil nilai akurasi dari LR dan LR+GFS+BG, kemudian dilakukan analisa dengan menggunakan uji t-Test Paired Two Sample for Means dengan hasil yang dapat dilihat pada Tabel 4.29.

Tabel 4. 29 Hasil Uji Beda Statistik Akurasi LR dan LR+GFS+BG

	LR	LR+GFS+BG
Mean	0.799198	0.79747
Variance	0.03199476	0.033687729
Observations	10	10
Pearson Correlation	0.999913985	
Hypothesized Mean Difference	0	
df	9	
t Stat	1.042599302	
P(T<=t) one-tail	0.162164641	
t Critical one-tail	1.833112933	
P(T<=t) two-tail	0.324329282	
t Critical two-tail	2.262157163	

Pada Tabel 4.29 dapat diketahui bahwa nilai rata-rata akurasi dari model LR+GFS+BG lebih tinggi dibandingkan model LR sebesar 0.86483. Dalam uji beda statistik nilai alpha ditentukan sebesar 0.05, jika nilai p lebih kecil dibanding alpha ($p < 0.05$) maka H_0 ditolak dan H_1 diterima sehingga disimpulkan ada perbedaan yang signifikan antara model yang dibandingkan, akan tetapi jika nilai p lebih besar dibandingkan nilai alpha ($p > 0.05$) maka H_0 diterima dan H_1 ditolak

sehingga disimpulkan tidak ada perbedaan yang signifikan antara model yang dibandingkan.

Pada Tabel 4.29 dapat dilihat bahwa nilai $P(T \leq t)$ adalah 0.2 dan ini menunjukkan bahwa nilai p lebih besar dibandingkan nilai α ($0.2 > 0.05$) sehingga hipotesis H_0 diterima dan H_1 ditolak. Hipotesis H_0 diterima berarti dapat disimpulkan bahwa tidak ada perbedaan signifikan antara model LR dan model LR+GFS+BG.

2. Sensitifitas logistic regression dan logistic regression dengan greedy forward selection dan bagging

Dilakukan uji beda dengan menggunakan metode statistik untuk menguji hipotesa pada Logistic regression (LR) dan Logistic regression dengan Greedy forward selection dan bagging (LR+GFS+BG).

H_0 : Tidak ada perbedaan nilai rata-rata sensitifitas LR dengan LR+GFS+BG.

H_1 : Ada perbedaan antara nilai rata-rata sensitifitas LR dengan LR+GFS+BG.

Pada Tabel 4.30 dapat dilihat perbedaan nilai sensitifitas antara LR dengan LR+GFS+BG.

Tabel 4. 30 Perbandingan Sensitifitas LR dan LR+GFS+BG

Datas e t	LR	LR+GFS+BG
CM1	33.33%	28.57%
JM1	94.86%	95.97%
KC1	37.54%	33.85%
KC3	36.11%	41.67%
MC2	35.83%	50%
PC1	36.07%	55.56%
PC2	18.75%	12.50%
PC3	90.00%	36.43%
PC4	94.10%	91.07%
PC5	44.73%	29.62%

Selanjutnya hasil nilai sensitifitas dari LR dan LR+GFS+BG dilakukan analisa dengan menggunakan uji t-Test Paired Two Sample for Means dengan hasil yang dapat dilihat pada Tabel 4.31.

Tabel 4. 31 Hasil Uji Beda Statistik Sensitifitas LR dan LR+GFS+BG

	LR	LR+GFS+BG
Mean	0.799198	0.79747
Variance	0.03199476	0.033687729
Observations	10	10
Pearson Correlation	0.999913985	
Hypothesized Mean Difference	0	
df	9	
t Stat	1.042599302	
P(T<=t) one-tail	0.162164641	
t Critical one-tail	1.833112933	
P(T<=t) two-tail	0.324329282	
t Critical two-tail	2.262157163	

Pada Tabel 4.31 dapat diketahui bahwa nilai rata-rata sensitifitas dari model LR lebih tinggi dibandingkan model LR+GFS+BG sebesar 0.52132. Dalam uji beda statistik nilai alpha ditentukan sebesar 0.05, jika nilai p lebih kecil dibanding alpha ($p < 0.05$) maka H_0 ditolak dan H_1 diterima sehingga disimpulkan ada perbedaan yang signifikan antara model yang dibandingkan, akan tetapi jika nilai p lebih besar dibandingkan nilai alpha ($p > 0.05$) maka H_0 diterima dan H_1 ditolak sehingga disimpulkan tidak ada perbedaan yang signifikan antara model yang dibandingkan.

Pada Tabel 4.31 dapat dilihat bahwa nilai $P(T \leq t)$ adalah 0.4 dan ini menunjukkan bahwa nilai p lebih besar dibandingkan nilai alpha ($0.4 > 0.05$) sehingga hipotesis H_0 diterima dan H_1 ditolak. Hipotesis H_0 diterima berarti dapat disimpulkan bahwa tidak ada perbedaan signifikan antara model LR dan model LR+GFS+BG.

3. F-measure logistic regression dan logistic regression dengan greedy forward selection dan bagging

Uji beda dilakukan pada f-measure dengan menggunakan metode Statistik untuk menguji hipotesa pada Logistic regression (LR) dan Logistic regression dengan Greedy forward selection dan bagging (LR+GFS+BG).

H_0 : Tidak ada perbedaan nilai rata-rata f-measure LR dengan LR+GFS+BG.

H_1 : Ada perbedaan antara nilai rata-rata f-measure LR dengan LR+GFS+BG.

Tabel 4.32 adalah perbandingan nilai f-measure antara LR dengan LR+GFS+BG.

Tabel 4. 32 Perbandingan F-measure LR dan LR+GFS+BG

Datas et	LR	LR+GFS+BG
CM1	0.32	0.33
JM1	0.89	0.90
KC1	0.40	0.39
KC3	0.38	0.48
MC2	0.47	0.58
PC1	0.34	0.45
PC2	0.08	0.20
PC3	0.25	0.36
PC4	0.93	0.92
PC5	0.89	0.40

Selanjutnya hasil nilai f-measure dari LR dan LR+GFS+BG dilakukan analisa dengan menggunakan uji t-Test Paired Two Sample for Means dengan hasil yang ditunjukkan pada Tabel 4.33.

Tabel 4. 33 Hasil Uji Beda Statistik F-measure LR dan LR+GFS+BG

	LR	LR+GFS+BG
Mean	0.799198	0.79747
Variance	0.03199476	0.033687729
Observations	10	10
Pearson Correlation	0.999913985	
Hypothesized Mean Difference	0	
df	9	
t Stat	1.042599302	
P(T<=t) one-tail	0.162164641	
t Critical one-tail	1.833112933	
P(T<=t) two-tail	0.324329282	
t Critical two-tail	2.262157163	

Pada Tabel 4.33 dapat diketahui bahwa nilai rata-rata f-measure dari model LR+GFS+BG lebih tinggi dibandingkan model LR sebesar 0.500064152. Dalam uji beda statistik nilai alpha ditentukan sebesar 0.05, jika nilai p lebih kecil

dibanding alpha ($p < 0.05$) maka H_0 ditolak dan H_1 diterima sehingga disimpulkan ada perbedaan yang signifikan antara model yang dibandingkan, akan tetapi jika nilai p lebih besar dibandingkan nilai alpha ($p > 0.05$) maka H_0 diterima dan H_1 ditolak sehingga disimpulkan tidak ada perbedaan yang signifikan antara model yang dibandingkan.

Pada Tabel 4.33 dapat dilihat bahwa nilai $P(T \leq t)$ adalah 0.9 dan ini menunjukkan bahwa nilai p lebih besar dibandingkan nilai alpha ($0.9 > 0.05$) sehingga hipotesis H_0 ditolak dan H_1 diterima. Hipotesis H_0 diterima berarti dapat disimpulkan bahwa tidak ada perbedaan signifikan antara model LR dan model LR+GFS+BG.

4. G-mean logistic regression dan logistic regression dengan greedy forward selection dan bagging

Uji beda dilakukan pada g-mean dengan menggunakan metode Statistik untuk menguji hipotesa pada Logistic regression (LR) dan Logistic regression dengan Greedy forward selection dan bagging (LR+GFS+BG).

H_0 : Tidak ada perbedaan nilai rata-rata g-mean LR dengan LR+GFS+BG.

H_1 : Ada perbedaan antara nilai rata-rata g-mean LR dengan LR+GFS+BG.

Tabel 4.34 adalah perbandingan nilai g-mean antara LR dengan LR+GFS+BG.

Tabel 4. 34 Perbandingan G-mean LR dan LR+GFS+BG

Datas e t	LR	LR+GFS+BG
CM1	0.55	0.52
JM1	0.95	0.42
KC1	0.58	0.56
KC3	0.57	0.62
MC2	0.57	0.66
PC1	0.58	0.71
PC2	0.43	0.35
PC3	0.48	0.57
PC4	0.94	0.72
PC5	0.66	0.54

Selanjutnya hasil nilai g-mean dari LR dan LR+GFS+BG dilakukan analisa dengan menggunakan uji t-Test Paired Two Sample for Means dengan hasil yang ditunjukkan pada Tabel 4.35.

Tabel 4. 35Tabel 4.57 Hasil Uji Beda Statistik G-mean LR dan LR+GFS+BG

	LR	LR+GFS+BG
Mean	0.799198	0.79747
Variance	0.03199476	0.033687729
Observations	10	10
Pearson Correlation	0.999913985	
Hypothesized Mean Difference	0	
df	9	
t Stat	1.042599302	
P(T<=t) one-tail	0.162164641	
t Critical one-tail	1.833112933	
P(T<=t) two-tail	0.324329282	
t Critical two-tail	2.262157163	

Pada Tabel 4.35 dapat diketahui bahwa nilai rata-rata g-mean dari model LR lebih tinggi dibandingkan model LR+GFS+BG sebesar 0.630158868. Dalam uji beda statistik nilai alpha ditentukan sebesar 0.05, jika nilai p lebih kecil dibanding alpha ($p < 0.05$) maka H_0 ditolak dan H_1 diterima sehingga disimpulkan ada perbedaan yang signifikan antara model yang dibandingkan, akan tetapi jika nilai p lebih besar dibandingkan nilai alpha ($p > 0.05$) maka H_0 diterima dan H_1 ditolak sehingga disimpulkan tidak ada perbedaan yang signifikan antara model yang dibandingkan.

Pada Tabel 4.35 dapat dilihat bahwa nilai $P(T \leq t)$ adalah 0.3 dan ini menunjukkan bahwa nilai p lebih besar dibandingkan nilai alpha ($0.3 > 0.05$) sehingga hipotesis H_0 diterima dan H_1 ditolak. Hipotesis H_0 diterima berarti dapat disimpulkan bahwa tidak ada perbedaan signifikan antara model LR dan model LR+GFS+BG.

5. AUC logistic regression dan logistic regression dengan greedy forward selection dan bagging

Uji beda dilakukan pada AUC dengan menggunakan metode statistik untuk menguji hipotesa pada Logistic regression (LR) dan Logistic regression dengan Greedy forward selection dan Bagging (LR+GFS+BG).

H0: Tidak ada perbedaan nilai rata-rata AUC LR dengan LR+GFS+BG.

H1: Ada perbedaan antara nilai rata-rata AUC LR dengan LR+GFS+BG.

Tabel 4.36 merupakan hasil dari perbandingan nilai AUC antara LR dengan LR+BG.

Tabel 4. 36 Perbandingan AUC LR dan LR+GFS+BG

Datase t	LR	LR+GFS+BG
CM1	0.708	0.771
JM1	0.683	0.688
KC1	0.786	0.788
KC3	0.677	0.763
MC2	0.712	0.781
PC1	0.775	0.817
PC2	0.885	0.918
PC3	0.756	0.778
PC4	0.84	0.866
PC5	0.94	0.923

Selanjutnya hasil nilai AUC dari LR dan LR+GFS+BG dilakukan analisa dengan menggunakan uji t-Test Paired Two Sample for Means dengan hasil yang ditunjukkan pada Tabel 4.37.

Tabel 4. 37 Hasil Uji Beda Statistik AUC LR dan LR+GFS+BG

	LR	LR+GFS+BG
Mean	0.799198	0.79747
Variance	0.03199476	0.033687729
Observations	10	10
Pearson Correlation	0.999913985	
Hypothesized Mean Difference	0	
df	9	
t Stat	1.042599302	
P(T<=t) one-tail	0.162164641	
t Critical one-tail	1.833112933	
P(T<=t) two-tail	0.324329282	
t Critical two-tail	2.262157163	

Pada Tabel 4.37 dapat diketahui bahwa nilai rata-rata AUC dari model LR+GFS+BG lebih tinggi dibandingkan model LR sebesar 0.8093. Dalam uji beda statistik nilai alpha ditentukan sebesar 0.05, jika nilai p lebih kecil dibanding alpha ($p < 0.05$) maka H_0 ditolak dan H_1 diterima sehingga disimpulkan ada perbedaan yang signifikan antara model yang dibandingkan, akan tetapi jika nilai p lebih besar dibandingkan nilai alpha ($p > 0.05$) maka H_0 diterima dan H_1 ditolak sehingga disimpulkan tidak ada perbedaan yang signifikan antara model yang dibandingkan.

Pada Tabel 4.37 dapat dilihat bahwa nilai $P(T \leq t)$ adalah 0.01 dan ini menunjukkan bahwa nilai p lebih kecil dibandingkan nilai alpha ($0.01 < 0.05$) sehingga hipotesis H_0 ditolak dan H_1 diterima. Hipotesis H_1 diterima berarti dapat disimpulkan bahwa ada perbedaan signifikan antara model LR dan model LR+GFS+BG sehingga model LR+GFS+BG membuat peningkatan ketika dibandingkan dengan model LR.

4.2.1.4. Perbandingan Kinerja Logistic Regression dengan Bagging dan Logistic Regression dengan Greedy Forward Selection dan Bagging

1. Akurasi logistic regression dengan bagging dan logistic regression dengan greedy forward selection dan bagging

Dilakukan uji beda dengan menggunakan metode statistik untuk menguji hipotesa pada model Logistic regression dengan Bagging (LR+BG) dan Logistic regression dengan Greedy forward selection dan Bagging (LR+GFS+BG).

H_0 : Tidak ada perbedaan nilai rata-rata akurasi LR+BG dengan LR+GFS+BG.

H_1 : Ada perbedaan antara nilai rata-rata akurasi LR+BG dengan LR+GFS+BG.

Pada Tabel 4.38 dapat dilihat perbedaan nilai akurasi antara LR+BG dengan LR+GFS+BG.

Tabel 4. 38 Perbandingan Akurasi LR+BG dan LR+GFS+BG

Dataset	LR+BG	LR+GFS+BG
CM1	82.25%	85.76%
JM1	81.32%	81.58%
KC1	82.30%	83.35%
KC3	79.00%	83.50%

MC2	72.37%	74.87%
PC1	88.54%	91.18%
PC2	95.52%	96.91%
PC3	31.91%	83.56%
PC4	87.63%	86.77%
PC5	96.63%	97.35%

Setelah dibandingkan hasil nilai akurasi dari LR+BG dan LR+GFS+BG, kemudian dilakukan analisa dengan menggunakan uji t-Test Paired Two Sample for Means dengan hasil yang dapat dilihat pada Tabel 4.39.

Tabel 4. 39 Hasil Uji Beda Statistik LR+BG dan LR+GFS+BG

	LR	LR+GFS+BG
Mean	0.799198	0.79747
Variance	0.03199476	0.033687729
Observations	10	10
Pearson Correlation	0.999913985	
Hypothesized Mean Difference	0	
df	9	
t Stat	1.042599302	
P(T<=t) one-tail	0.162164641	
t Critical one-tail	1.833112933	
P(T<=t) two-tail	0.324329282	
t Critical two-tail	2.262157163	

Pada Tabel 4.39 dapat diketahui bahwa nilai rata-rata akurasi dari model LR+GFS+BG lebih tinggi dibandingkan model LR+BG sebesar 0.86483. Dalam uji beda statistik nilai alpha ditentukan sebesar 0.05, jika nilai p lebih kecil dibanding alpha ($p < 0.05$) maka H_0 ditolak dan H_1 diterima sehingga disimpulkan ada perbedaan yang signifikan antara model yang dibandingkan, akan tetapi jika nilai p lebih besar dibanding alpha ($p > 0.05$) maka H_0 diterima dan H_1 ditolak sehingga disimpulkan tidak ada perbedaan yang signifikan antara model yang dibandingkan.

Pada Tabel 4.39 dapat dilihat bahwa nilai $P(T \leq t)$ adalah 0.2 dan ini menunjukkan bahwa nilai p lebih besar dibandingkan nilai alpha ($0.2 > 0.05$) sehingga hipotesis H_0 diterima H_1 ditolak. Hipotesis H_0 diterima berarti dapat

disimpulkan bahwa tidak ada perbedaan signifikan antara model LR+BG dan model LR+GFS+BG.

2. Sensitifitas logistic regression dengan bagging dan logistic regression dengan greedy forward selection dan bagging

Dilakukan uji beda dengan menggunakan metode statistik untuk menguji hipotesa pada model Logistic regression dengan Bagging (LR+BG) dan Logistic regression dengan Greedy forward selection dan Bagging (LR+GFS+BG).

H0: Tidak ada perbedaan nilai rata-rata sensitifitas LR+BG dengan LR+GFS+BG.

H1: Ada perbedaan antara nilai rata-rata sensitifitas LR+BG dengan LR+GFS+BG.

Pada Tabel 4.40 dapat dilihat perbedaan nilai sensitifitas antara LR+BG dengan LR+GFS+BG.

Tabel 4. 40 Perbandingan Sensitifitas LR+BG dan LR+GFS+BG

Dataset	LR+BG	LR+GFS+BG
CM1	30.95%	28.57%
JM1	94.87%	95.97%
KC1	37.54%	33.85%
KC3	36.11%	41.67%
MC2	36.36%	50%
PC1	36.07%	55.56%
PC2	18.75%	12.50%
PC3	90.71%	36.43%
PC4	94.27%	91.07%
PC5	44.93%	29.62%

Selanjutnya hasil nilai sensitifitas dari LR+BG dan LR+GFS+BG dilakukan analisa dengan menggunakan uji t-Test Paired Two Sample for Means dengan hasil yang dapat dilihat pada Tabel 4.63.

Tabel 4. 41 Hasil Uji beda Statistik Sensitifitas LR+BG dan LR+GFS+BG

	LR	LR+GFS+BG
Mean	0.799198	0.79747
Variance	0.03199476	0.033687729
Observations	10	10
Pearson Correlation	0.999913985	
Hypothesized Mean Difference	0	
df	9	
t Stat	1.042599302	
P(T<=t) one-tail	0.162164641	
t Critical one-tail	1.833112933	
P(T<=t) two-tail	0.324329282	
t Critical two-tail	2.262157163	

Pada Tabel 4.41 dapat diketahui bahwa nilai rata-rata sensitifitas dari model LR+BG lebih tinggi dibandingkan model LR+GFS+BG sebesar 0.52056. Dalam uji beda statistik nilai alpha ditentukan sebesar 0.05, jika nilai p lebih kecil dibanding alpha ($p < 0.05$) maka H_0 ditolak dan H_1 diterima sehingga disimpulkan ada perbedaan yang signifikan antara model yang dibandingkan, akan tetapi jika nilai p lebih besar dibanding alpha ($p > 0.05$) maka H_0 diterima dan H_1 ditolak sehingga disimpulkan tidak ada perbedaan yang signifikan antara model yang dibandingkan.

Pada Tabel 4.41 dapat dilihat bahwa nilai $P(T \leq t)$ adalah 0.4 dan ini menunjukkan bahwa nilai p lebih besar dibandingkan nilai alpha ($0.4 > 0.05$) sehingga hipotesis H_0 diterima H_1 ditolak. Hipotesis H_0 diterima berarti dapat disimpulkan bahwa tidak ada perbedaan signifikan antara model LR+BG dan model LR+GFS+BG.

3. F-measure logistic regression dengan bagging dan logistic regression dengan greedy forward selection dan bagging

Uji beda dilakukan pada f-measure dengan menggunakan metode statistik untuk menguji hipotesa pada model Logistic regression dengan Bagging (LR+BG) dan Logistic regression dengan Greedy forward selection dan Bagging (LR+GFS+BG).

H0: Tidak ada perbedaan nilai rata-rata f-measure LR+BG dengan LR+GFS+BG.

H1: Ada perbedaan antara nilai rata-rata f-measure LR+BG dengan LR+GFS+BG.

Hasil perbandingan nilai f-measure antara LR+BG dengan LR+GFS+BG dapat ditunjukkan pada Tabel 4.42.

Tabel 4. 42 Perbandingan F-measure LR+BG dan LR+GFS+BG

Datas et	LR+BG	LR+GFS+BG
CM1	0.30	0.33
JM1	0.89	0.90
KC1	0.40	0.39
KC3	0.38	0.48
MC2	0.48	0.58
PC1	0.34	0.45
PC2	0.08	0.20
PC3	0.25	0.36
PC4	0.93	0.92
PC5	0.44	0.40

Selanjutnya hasil nilai f-measure dari LR+BG dan LR+GFS+BG dilakukan analisa dengan menggunakan uji t-Test Paired Two Sample for Means dengan hasil yang ditunjukkan pada Tabel 4.43.

Tabel 4. 43 Hasil Uji Beda Statistik F-measure LR+BG dan LR+GFS+BG

	LR	LR+GFS+BG
Mean	0.799198	0.79747
Variance	0.03199476	0.033687729
Observations	10	10
Pearson Correlation	0.999913985	
Hypothesized Mean Difference	0	
df	9	
t Stat	1.042599302	
P(T<=t) one-tail	0.162164641	
t Critical one-tail	1.833112933	
P(T<=t) two-tail	0.324329282	
t Critical two-tail	2.262157163	

Pada Tabel 4.43 dapat diketahui bahwa nilai rata-rata f-measure dari model LR+GFS+BG lebih tinggi dibandingkan model LR+BG sebesar 0.500064152. Dalam uji beda statistik nilai alpha ditentukan sebesar 0.05, jika nilai p lebih kecil dibanding alpha ($p < 0.05$) maka H_0 ditolak dan H_1 diterima sehingga disimpulkan ada perbedaan yang signifikan antara model yang dibandingkan, akan tetapi jika nilai p lebih besar dibanding alpha ($p > 0.05$) maka H_0 diterima dan H_1 ditolak sehingga disimpulkan tidak ada perbedaan yang signifikan antara model yang dibandingkan.

Pada Tabel 4.43 dapat dilihat bahwa nilai $P(T \leq t)$ adalah 0.02 dan ini menunjukkan bahwa nilai p lebih kecil dibandingkan nilai alpha ($0.02 < 0.05$) sehingga hipotesis H_0 ditolak H_1 diterima. Hipotesis H_1 diterima berarti dapat disimpulkan bahwa ada perbedaan signifikan antara model LR+BG dan model LR+GFS+BG sehingga model LR+GFS+BG membuat peningkatan ketika dibandingkan dengan model LR+BG.

4. G-mean logistic regression dengan bagging dan logistic regression dengan greedy forward selection dan bagging

Uji beda dilakukan pada G-mean dengan menggunakan metode Statistik untuk menguji hipotesa pada model Logistic regression dengan Bagging (LR+BG) dan Logistic regression dengan Greedy forward selection dan Bagging (LR+GFS+BG).

H_0 : Tidak ada perbedaan nilai rata-rata g-mean LR+BG dengan LR+GFS+BG.

H_1 : Ada perbedaan antara nilai rata-rata g-mean LR+BG dengan LR+GFS+BG.

Tabel 4.44 adalah perbandingan nilai g-mean antara LR+BG dengan LR+GFS+BG.

Tabel 4. 44 Perbandingan G-mean LR+BG dan LR+GFS+BG

Datas et	LR+BG	LR+GFS+BG
CM1	0.53	0.52
JM1	0.95	0.42
KC1	0.58	0.56
KC3	0.57	0.62

MC2	0.58	0.66
PC1	0.58	0.71
PC2	0.42	0.35
PC3	0.46	0.57
PC4	0.94	0.72
PC5	0.66	0.54

Selanjutnya hasil nilai g-mean dari LR+BG dan LR+GFS+BG dilakukan analisa dengan menggunakan uji t-Test Paired Two Sample for Means dengan hasil yang ditunjukkan pada Tabel 4.45.

Tabel 4. 45 Hasil Uji Beda Statistik G-mean LR+BG dan LR+GFS+BG

	LR	LR+GFS+BG
Mean	0.799198	0.79747
Variance	0.03199476	0.033687729
Observations	10	10
Pearson Correlation	0.999913985	
Hypothesized Mean Difference	0	
df	9	
t Stat	1.042599302	
P(T<=t) one-tail	0.162164641	
t Critical one-tail	1.833112933	
P(T<=t) two-tail	0.324329282	
t Critical two-tail	2.262157163	

Pada Tabel 4.45 dapat diketahui bahwa nilai rata-rata sensitifitas dari model LR+BG lebih tinggi dibandingkan model LR+GFS+BG sebesar 0.627334228. Dalam uji beda statistik nilai alpha ditentukan sebesar 0.05, jika nilai p lebih kecil dibanding alpha ($p < 0.05$) maka H_0 ditolak dan H_1 diterima sehingga disimpulkan ada perbedaan yang signifikan antara model yang dibandingkan, akan tetapi jika nilai p lebih besar dibanding alpha ($p > 0.05$) maka H_0 diterima dan H_1 ditolak sehingga disimpulkan tidak ada perbedaan yang signifikan antara model yang dibandingkan.

Pada Tabel 4.45 dapat dilihat bahwa nilai $P(T \leq t)$ adalah 0.3 dan ini menunjukkan bahwa nilai p lebih besar dibandingkan nilai alpha ($0.3 > 0.05$) sehingga hipotesis H_0 diterima H_1 ditolak. Hipotesis H_0 diterima berarti dapat

disimpulkan bahwa tidak ada perbedaan signifikan antara model LR+BG dan model LR+GFS+BG.

5. AUC logistic regression dengan bagging dan logistic regression dengan greedy forward selection dan bagging.

Uji beda dilakukan pada AUC dengan menggunakan metode statistik untuk menguji hipotesa pada model Logistic regression dengan Bagging (LR+BG) dan Logistic regression dengan Greedy forward selection dan Bagging (LR+GFS+BG).

H0: Tidak ada perbedaan nilai rata-rata AUC LR+BG dengan LR+GFS+BG.

H1: Ada perbedaan antara nilai rata-rata AUC LR+BG dengan LR+GFS+BG.

Tabel 4.46 adalah perbandingan nilai AUC antara LR+BG dengan LR+GFS+BG.

Tabel 4. 46 Perbandingan AUC LR+BG dan LR+GFS+BG

Datas e t	LR+BG	LR+GFS+BG
CM1	0.717	0.771
JM1	0.685	0.688
KC1	0.789	0.788
KC3	0.673	0.763
MC2	0.721	0.781
PC1	0.794	0.817
PC2	0.877	0.918
PC3	0.756	0.778
PC4	0.838	0.866
PC5	0.942	0.923

Selanjutnya hasil nilai AUC dari LR+BG dan LR+GFS+BG dilakukan analisa dengan menggunakan uji t-Test Paired Two Sample for Means dengan hasil yang ditunjukkan pada Tabel 4.47.

Tabel 4. 47 Hasil Uji beda Statistik AUC LR+BG dan LR+GFS+BG

	LR	LR+GFS+BG
Mean	0.799198	0.79747
Variance	0.03199476	0.033687729
Observations	10	10
Pearson Correlation	0.999913985	

Hypothesized Mean Difference	0
df	9
t Stat	1.042599302
P(T<=t) one-tail	0.162164641
t Critical one-tail	1.833112933
P(T<=t) two-tail	0.324329282
t Critical two-tail	2.262157163

Pada Tabel 4.47 dapat diketahui bahwa nilai rata-rata AUC dari model LR+GFS+BG lebih tinggi dibandingkan model LR+BG sebesar 0.8093. Dalam uji beda statistik nilai alpha ditentukan sebesar 0.05, jika nilai p lebih kecil dibanding alpha ($p < 0.05$) maka H_0 ditolak dan H_1 diterima sehingga disimpulkan ada perbedaan yang signifikan antara model yang dibandingkan, akan tetapi jika nilai p lebih besar dibanding alpha ($p > 0.05$) maka H_0 diterima dan H_1 ditolak sehingga disimpulkan tidak ada perbedaan yang signifikan antara model yang dibandingkan.

Pada Tabel 4.47 dapat dilihat bahwa nilai $P(T \leq t)$ adalah 0.01 dan ini menunjukkan bahwa nilai p lebih kecil dibandingkan nilai alpha ($0.01 < 0.05$) sehingga hipotesis H_0 ditolak H_1 diterima. Hipotesis H_1 diterima berarti dapat disimpulkan bahwa ada perbedaan signifikan antara model LR+BG dan model LR+GFS+BG sehingga model LR+GFS+BG membuat peningkatan ketika dibandingkan dengan model LR+BG.

4.2.1.5. Perbandingan Kinerja Logistic Regression dengan Bagging dan Logistic Regression dengan Greedy forward selection

1. Akurasi logistic regression dengan bagging dan logistic regression dengan greedy forward selection

Dilakukan uji beda dengan menggunakan metode statistik untuk menguji hipotesa pada model Logistic regression dengan Bagging (LR+BG) dan Logistic regression dengan Greedy forward selection (LR+GFS).

H_0 : Tidak ada perbedaan nilai rata-rata akurasi LR+BG dengan LR+GFS.

H_1 : Ada perbedaan antara nilai rata-rata akurasi LR+BG dengan LR+GFS. Pada Tabel 4.48 dapat dilihat perbedaan nilai akurasi antara LR+BG dengan LR+GFS.

Tabel 4. 48 Perbandingan Akurasi LR+BG dan LR+GFS

Datase t	LR+BG	LR+GFS
CM1	82.25%	86.94%
JM1	81.32%	81.48%
KC1	82.30%	83.64%
KC3	79.00%	82.00%
MC2	72.37%	75.64%
PC1	88.54%	91.84%
PC2	95.52%	96.66%
PC3	31.91%	84.00%
PC4	87.63%	83.99%
PC5	96.63%	97.21%

Setelah dibandingkan hasil nilai akurasi dari LR+BG dan LR+GFS, kemudian dilakukan analisa dengan menggunakan uji t-Test Paired Two Sample for Means dengan hasil yang dapat dilihat pada Tabel 4.49.

Tabel 4. 49 Hasil Uji beda Statistik Akurasi LR+BG dan LR+GFS

	<i>LR+BG</i>	<i>LR+GFS</i>
Mean	0.79747	0.8634
Variance	0.03368772	0.00478336
Observations	10	10
Pearson Correlation	0.48772408	
Hypothesized Mean Difference	0	
df	9	
t Stat	-	
P(T<=t) one-tail	0.11447245	
t Critical one-tail	1.83311293	
P(T<=t) two-tail	0.2289449	
t Critical two-tail	2.26215716	

Pada Tabel 4.49 dapat diketahui bahwa nilai rata-rata akurasi dari model LR+GFS lebih tinggi dibandingkan model LR+BG sebesar 0.8634. Dalam uji beda statistik nilai alpha ditentukan sebesar 0.05, jika nilai p lebih kecil dibanding alpha ($p < 0.05$) maka H_0 ditolak dan H_1 diterima sehingga disimpulkan ada perbedaan yang signifikan antara model yang dibandingkan, akan tetapi jika nilai p lebih besar dibanding alpha ($p > 0.05$) maka H_0 diterima dan H_1 ditolak sehingga disimpulkan tidak ada perbedaan yang signifikan antara model yang dibandingkan.

Pada Tabel 4.49 dapat dilihat bahwa nilai $P(T \leq t)$ adalah 0.2 dan ini menunjukkan bahwa nilai p lebih besar dibandingkan nilai alpha ($0.2 > 0.05$) sehingga hipotesis H_0 diterima H_1 ditolak. Hipotesis H_0 diterima berarti dapat disimpulkan bahwa tidak ada perbedaan signifikan antara model LR+BG dan model LR+GFS.

2. Sensitifitas logistic regression dengan bagging dan logistic regression dengan greedy forward selection

Dilakukan uji beda dengan menggunakan metode Statistik untuk menguji hipotesa pada model Logistic regression dengan Bagging (LR+BG) dan Logistic regression dengan Greedy forward selection (LR+GFS).

H_0 : Tidak ada perbedaan nilai rata-rata sensitifitas LR+BG dengan LR+GFS.

H_1 : Ada perbedaan antara nilai rata-rata sensitifitas LR+BG dengan LR+GFS.

Pada Tabel 4.50 dapat dilihat perbedaan nilai sensitifitas antara LR+BG dengan LR+GFS.

Tabel 4. 50 Perbandingan Sensitifitas LR+BG dan LR+GFS

Datas et	LR+BG	LR+GFS
CM1	30.95%	34.67%
JM1	94.87%	95.25%
KC1	37.54%	33.54%
KC3	36.11%	33.33%
MC2	36.36%	43.18%
PC1	36.07%	36.07%
PC2	18.75%	6.25%
PC3	90.71%	37.86%
PC4	94.27%	87.39%
PC5	44.93%	33.80%

Selanjutnya hasil nilai sensitifitas dari LR+BG dan LR+GFS dilakukan analisa dengan menggunakan uji t-Test Paired Two Sample for Means dengan hasil yang dapat dilihat pada Tabel 4.51.

Tabel 4. 51 Hasil Uji Beda Statistik Sensitifitas LR+BG dan LR+GFS

	<i>LR+BG</i>	<i>LR+GFS</i>
Mean	0.52056	0.4413
Variance	0.085351452	0.0716638
Observations	10	10
Pearson Correlation	0.820748285	
Hypothesized Mean Difference	0	
df	9	
t Stat	1.480402445	
P(T<=t) one-tail	0.086449937	
t Critical one-tail	1.833112933	
P(T<=t) two-tail	0.17289987	
t Critical two-tail	2.262157163	

Pada Tabel 4.51 dapat diketahui bahwa nilai rata-rata akurasi dari model LR+BG lebih tinggi dibandingkan model LR+GFS sebesar 0.52056. Dalam uji beda statistik nilai alpha ditentukan sebesar 0.05, jika nilai p lebih kecil dibanding alpha ($p < 0.05$) maka H_0 ditolak dan H_1 diterima sehingga disimpulkan ada perbedaan yang signifikan antara model yang dibandingkan, akan tetapi jika nilai p lebih besar dibanding alpha ($p > 0.05$) maka H_0 diterima dan H_1 ditolak sehingga disimpulkan tidak ada perbedaan yang signifikan antara model yang dibandingkan.

Pada Tabel 4.51 dapat dilihat bahwa nilai $P(T \leq t)$ adalah 0.1 dan ini menunjukkan bahwa nilai p lebih besar dibandingkan nilai alpha ($0.1 > 0.05$) sehingga hipotesis H_0 diterima H_1 ditolak. Hipotesis H_0 diterima berarti dapat disimpulkan bahwa tidak ada perbedaan signifikan antara model LR+BG dan model LR+GFS.

3. F-measure logistic regression dengan bagging dan logistic regression dengan greedy forward selection

Uji beda dilakukan pada f-measure dengan menggunakan metode statistik untuk menguji hipotesa pada model Logistic regression dengan Bagging (LR+BG) dan Logistic regression dengan Greedy forward selection (LR+GFS).

H_0 : Tidak ada perbedaan nilai rata-rata f-measure LR+BG dengan LR+GFS.

H_1 : Ada perbedaan antara nilai rata-rata f-measure LR+BG dengan LR+GFS.

Tabel 4.52 adalah perbandingan nilai f-measure antara LR+BG dengan LR+GFS.

Tabel 4. 52 Perbandingan F-measure LR+BG dan LR+GFS

Datas et	LR+BG	LR+GFS
CM1	0.30	0.39
JM1	0.89	0.89
KC1	0.40	0.39
KC3	0.38	0.40
MC2	0.48	0.55
PC1	0.34	0.42
PC2	0.08	0.04
PC3	0.25	0.37
PC4	0.93	0.91
PC5	0.44	0.42

Selanjutnya hasil nilai f-measure dari LR+BG dan LR+GFS dilakukan analisa dengan menggunakan uji t-Test Paired Two Sample for Means dengan hasil yang ditunjukkan pada Tabel 4.53.

Tabel 4. 53 Hasil Uji Beda Statistik F-measure LR+BG dan LR+GFS

	<i>LR+BG</i>	<i>LR+GFS</i>
Mean	0.44818956	0.47694
Variance	0.07212896	0.06622062
Observations	10	10
Pearson Correlation	0.97664614	
Hypothesized Mean Difference	0	
df	9	
t Stat	-	
P(T<=t) one-tail	0.0754409	
t Critical one-tail	1.83311293	
P(T<=t) two-tail	0.15088196	
t Critical two-tail	2.26215716	

Pada Tabel 4.53 dapat diketahui bahwa nilai rata-rata f-measure dari model LR+GFS lebih tinggi dibandingkan model LR+BG sebesar 0.476942. Dalam uji beda statistik nilai alpha ditentukan sebesar 0.05, jika nilai p lebih kecil dibanding alpha ($p < 0.05$) maka H_0 ditolak dan H_1 diterima sehingga disimpulkan ada

perbedaan yang signifikan antara model yang dibandingkan, akan tetapi jika nilai p lebih besar dibanding α ($p > 0.05$) maka H_0 diterima dan H_1 ditolak sehingga disimpulkan tidak ada perbedaan yang signifikan antara model yang dibandingkan.

Pada Tabel 4.53 dapat dilihat bahwa nilai $P(T \leq t)$ adalah 0.1 dan ini menunjukkan bahwa nilai p lebih besar dibandingkan nilai α ($0.1 > 0.05$) sehingga hipotesis H_0 diterima H_1 ditolak.

4. G-mean logistic regression dengan bagging dan logistic regression dengan greedy forward selection

Uji beda dilakukan pada g-mean dengan menggunakan metode Statistik untuk menguji hipotesa pada model Logistic regression dengan Bagging (LR+BG) dan Logistic regression dengan Greedy forward selection (LR+GFS).

H_0 : Tidak ada perbedaan nilai rata-rata g-mean LR+BG dengan LR+GFS.

H_1 : Ada perbedaan antara nilai rata-rata g-mean LR+BG dengan LR+GFS.

Tabel 4.54 adalah perbandingan nilai g-mean antara LR+BG dengan LR+GFS dan Tabel 4.55 merupakan uji t-Test Paired Two Sample for Means.

Tabel 4. 54 Perbandingan G-mean LR+BG dan LR+GFS

Datas e t	LR+BG	LR+GFS
CM1	0.53	0.57
JM1	0.95	0.44
KC1	0.58	0.56
KC3	0.57	0.56
MC2	0.58	0.63
PC1	0.58	0.59
PC2	0.42	0.25
PC3	0.46	0.59
PC4	0.94	0.73
PC5	0.66	0.58

Tabel 4. 55 Hasil Uji Beda Statistik G-mean LR+BG dan LR+GFS

	LR+BG	LR+GFS
Mean	0.62733423	0.54866767
Variance	0.03258219	0.01636942
Observations	10	10
Pearson Correlation	0.32535509	

Hypothesized Mean Difference	0
df	9
t Stat	1.35063347
P(T<=t) one-tail	0.10489541
t Critical one-tail	1.83311293
P(T<=t) two-tail	0.20979084
t Critical two-tail	2.26215716

Pada Tabel 4.55 dapat diketahui bahwa nilai rata-rata g-mean dari model LR+BG lebih tinggi dibandingkan model LR+GFS sebesar 0.62733423. Dalam uji beda statistik nilai alpha ditentukan sebesar 0.05, jika nilai p lebih kecil dibanding alpha ($p < 0.05$) maka H_0 ditolak dan H_1 diterima sehingga disimpulkan ada perbedaan yang signifikan antara model yang dibandingkan, akan tetapi jika nilai p lebih besar dibandingkan nilai alpha ($p > 0.05$) maka H_0 diterima dan H_1 ditolak sehingga disimpulkan tidak ada perbedaan yang signifikan antara model yang dibandingkan.

Pada Tabel 4.55 dapat dilihat bahwa nilai $P(T \leq t)$ adalah 0.2 dan ini menunjukkan bahwa nilai p lebih besar dibandingkan nilai alpha ($0.2 > 0.05$) sehingga hipotesis H_0 diterima H_1 ditolak. Hipotesis H_0 diterima berarti dapat disimpulkan bahwa tidak ada perbedaan signifikan antara model LR+BG dan model LR+GFS.

5. AUC logistic regression dengan bagging dan logistic regression dengan greedy forward selection

Uji beda dilakukan pada AUC dengan menggunakan metode Statistik untuk menguji hipotesa pada model Logistic regression dengan Bagging (LR+BG) dan Logistic regression dengan Greedy forward selection (LR+GFS).

H_0 : Tidak ada perbedaan nilai rata-rata AUC LR+BG dengan LR+GFS.

H_1 : Ada perbedaan antara nilai rata-rata AUC LR+BG dengan LR+GFS.

Tabel 4.56 adalah perbandingan nilai AUC antara LR+BG dengan LR+GFS.

Tabel 4. 56 Perbandingan AUC LR+BG dan LR+GFS

Datase t	LR+BG	LR+GFS
CM1	0.717	0.769
JM1	0.685	0.653
KC1	0.789	0.788
KC3	0.673	0.765
MC2	0.721	0.775
PC1	0.794	0.81
PC2	0.877	0.894
PC3	0.756	0.784
PC4	0.838	0.86
PC5	0.942	0.941

Selanjutnya hasil nilai AUC dari LR+BG dan LR+GFS dilakukan analisa dengan menggunakan uji t-Test Paired Two Sample for Means dengan hasil yang ditunjukkan pada Tabel 4.57.

Tabel 4. 57 Hasil Uji Beda Statistik AUC LR+BG dan LR+GFS

	<i>LR+BG</i>	<i>LR+GFS</i>
Mean	0.7792	0.8039
Variance	0.00753195	0.00634276
Observations	10	10
Pearson Correlation	0.91670809	
Hypothesized Mean Difference	0	
df	9	
t Stat	-	
P(T<=t) one-tail	0.02539818	
t Critical one-tail	1.83311293	
P(T<=t) two-tail	0.05079637	
t Critical two-tail	2.26215716	

Pada Tabel 4.57 dapat diketahui bahwa nilai rata-rata AUC dari model LR+GFS lebih tinggi dibandingkan model LR+BG sebesar 0.8039. Dalam uji beda statistik nilai alpha ditentukan sebesar 0.05, jika nilai p lebih kecil atau sama dengan nilai alpha ($p \leq 0.05$) maka H_0 ditolak dan H_1 diterima sehingga disimpulkan ada perbedaan yang signifikan antara model yang dibandingkan, akan tetapi jika nilai p lebih besar dibanding alpha ($p > 0.05$) maka H_0 diterima dan H_1 ditolak sehingga disimpulkan tidak ada perbedaan yang signifikan antara model yang dibandingkan.

Pada Tabel 4.57 dapat dilihat bahwa nilai $P(T \leq t)$ adalah 0.05 dan ini menunjukkan bahwa nilai p sama dengan nilai alpha ($0.05 = 0.05$) sehingga hipotesis H_0 ditolak dan H_1 diterima. Hipotesis H_1 diterima berarti dapat disimpulkan bahwa ada perbedaan signifikan antara model LR+BG dan model LR+GFS sehingga model LR+GFS membuat peningkatan ketika dibandingkan dengan model LR+BG.

4.2.2. Perbandingan Logistic regression dengan Seleksi Fitur yang Lain

Eksperimen selanjutnya adalah membandingkan model yang diusulkan dengan model lain yang tujuannya adalah untuk mendapatkan model terbaik dengan kinerja yang paling baik. Model yang dibandingkan adalah model Logistic Regression (LR), model Logistic regression dan Bagging (LR+BG), model Logistic regression dan seleksi fitur Greedy Forward Selection (LR+GFS), Model Logistic regression dengan seleksi fitur Greedy Forward Selection dan Bagging (LR+GFS+BG), model Logistic regression dan seleksi fitur Genetic Feature Selection (LR+GAFS), model Logistic regression dan seleksi fitur Forward Selection (LR+FS) dan model Logistic regression dan Backward Elimination (LR+BE). Tabel 4.58 menunjukkan perbandingan hasil AUC pada dataset NASA MDP dengan menggunakan algoritma klasifikasi yang sama akan tetapi digabungkan dengan seleksi fitur yang lain.

Tabel 4. 58 Perbandingan AUC pada model prediksi cacat software

	CM1	JM1	KC1	KC3	MC2	PC1	PC2	PC3	PC4	PC5
LR	0.708	0.683	0.786	0.677	0.712	0.775	0.885	0.756	0.84	0.94
LR+BG	0.717	0.685	0.789	0.673	0.721	0.794	0.877	0.756	0.838	0.942
LR+GFS	0.769	0.653	0.788	0.765	0.775	0.81	0.894	0.784	0.86	0.941
LR+GFS+BG	0.771	0.688	0.788	0.763	0.781	0.817	0.918	0.778	0.866	0.923
LR+GAFS	0.757	0.635	0.79	0.719	0.758	0.79	0.75	0.792	0.857	0.952
LR+FS	0.601	0.612	0.799	0.749	0.707	0.742	0.824	0.583	0.812	0.886
LR+BE	0.717	0.659	0.768	0.734	0.275	0.799	0.908	0.808	0.885	0.938

Kemudian dilakukan uji friedman untuk mendapatkan model yang terbaik dengan menggunakan metode statistik.

H_0 : Tidak ada perbedaan nilai rata-rata AUC antara Logistic Regression dengan seleksi fitur yang lain.

H1: Ada perbedaan nilai rata-rata AUC antara Logistic Regression dengan seleksi fitur yang lain.

Uji friedman dilakukan dengan menggunakan aplikasi XLSTAT dan diketahui nilai alpha adalah 0.05 sehingga dapat disimpulkan jika $p\text{-value} < 0.05$ maka hipotesis H0 ditolak dan hipotesis H1 diterima. Akan tetapi, jika sebaliknya nilai $p\text{-value} > 0.05$ maka H0 diterima dan H1 ditolak. Setelah dilakukan perbandingan hasil AUC untuk seleksi fitur yang lain diketahui bahwa $p\text{-value} < \alpha$ atau $0.029 < 0.05$ sehingga hipotesis H0 ditolak dan hipotesis H1 diterima dan maksudnya adalah ada perbedaan antara nilai rata-rata AUC antara logistic regression dengan seleksi fitur yang lain. Hasil uji friedman ditunjukkan pada Tabel 4.59.

Tabel 4. 59 Uji Friedman

Q (Observed value)	18.39
Q (Critical value)	12.59
DF	6
p-value (Two-tailed)	0.00
alpha	0.05

Pada Tabel 4.59 diketahui bahwa terdapat perbedaan yang signifikan karena $p\text{-value} < \alpha$ dan hasilnya adalah hipotesis H0 ditolak dan hipotesis H1 diterima sehingga ada perbedaan yang signifikan antara Logistic Regression dengan seleksi fitur yang lain. Akan tetapi, belum dapat diketahui model mana yang mempunyai perbedaan signifikan sehingga dibutuhkan uji Nemenyi Post Hoc. Uji Post Hoc dilakukan untuk mengetahui model mana yang mempunyai perbedaan yang signifikan.

Tabel 4. 60 Perbandingan Multiple Pairwise Menggunakan Prosedur Nemenyi

Sample	Frequency	Sum of ranks	Mean of ranks
LR+FS	10	22.000	2.200
LR	10	29.500	2.950
LR+BG	10	37.000	3.700
LR+BE	10	41.500	4.150
LR+GAFS	10	42.000	4.200
LR+GFS	10	51.500	5.150
LR+GFS+BG	10	56.500	5.650

Terlihat bahwa model LR+GFS+BG merupakan model terbaik dalam penelitian ini dengan mempunyai rata rata AUC tertinggi dalam uji friedman dan model LR+GFS merupakan model terbaik kedua yang diikuti oleh model LR+GAFS, LR+BG, LR, LR+BE dan LR+FS. Gambar 4.3 mengenai rata- rata peringkat dimana model LR+GFS+BG merupakan model terbaik yang menunjukkan bahwa seleksi fitur greedy forward selection dan bagging mampu meningkatkan kinerja model prediksi cacat software dengan memperbaiki model LR dan dapat menangani permasalahan ketidakseimbangan kelas (imbalance class) dan fitur-fitur yang tidak relevan (irrelevant features).

BAB V

PENUTUP

5.1. Kesimpulan

Pengembangan software diperlukan agar menghasilkan software yang berkualitas. Kualitas software ditentukan dengan melakukan pemeriksaan dan pengujian, untuk menemukan cacat dalam software tersebut yang dapat menurunkan kualitas software. Terdapat banyak metode yang digunakan dalam pemeriksaan dan pengujian untuk melakukan prediksi kelompok cacat dan tidak cacat pada dataset cacat software. Dataset cacat software umumnya memiliki ketidaseimbangan kelas dan fitur-fitur yang tidak relevan yang dapat menyebabkan menurunnya kinerja dari algoritma pembelajaran. Salah satu algoritma pembelajaran yang digunakan dalam prediksi cacat software yaitu logistic regression, yang merupakan algoritma terbaik untuk prediksi cacat atau tidaknya sebuah software.

Penelitian dengan menerapkan metode greedy forward selection dan teknik bagging dapat menangani masalah ketidakseimbangan kelas dan redundant data pada dataset NASA MDP untuk prediksi cacat perangkat lunak dengan algoritma LR. Hasil percobaan pada penelitian ini mendapatkan nilai akurasi tertinggi sebesar 0,990 pada dataset PC2, naik 0,19% dibandingkan dengan metode LR tanpa GFS dan bagging. Sedangkan nilai AUC tertinggi sebesar 0,995 pada PC2, naik 7,94% dibandingkan dengan metode LR tanpa GFS dan bagging.

Tetapi jika dibandingkan dengan pengklasifikasi yang lain seperti Logistic Regression (LR), Linear Discriminant Analysis (LDA), K-Nearest Neighbour (KNN), C4.5, dan Random Forest (RF), metode yang diusulkan (LR+GFS+BG) naik 13,59% dari rata-rata hasil tertinggi pada 9 dataset.

5.2. Saran

Dalam penelitian ini terdapat beberapa cara untuk menangani ketidakseimbangan kelas dan redundant data (perulangan atribut). Beberapa hasil penelitian mungkin dapat menjadi kontribusi dalam penelitian selanjutnya pada penelitian lain.

Meskipun diketahui bahwa metode LR+GFS+BG memiliki akurasi dan nilai AUC yang tinggi, namun untuk penelitian selanjutnya hal-hal berikut dapat disarankan, antara lain, 1) menerapkan langsung pada industri pembuatan perangkat lunak, 2) menggunakan feature selection yang lain seperti Genetic algorithm, Particle Swarm Optimization, atau ant colony, 3) menggunakan algoritma lain seperti Adabost atau Boosting untuk mengetahui algoritma yang paling tepat untuk prediksi cacat perangkat lunak.

DAFTAR REFERENSI

- Alpaydin, E. (2010). *Introduction to machine learning*.
<https://doi.org/10.1007/s13398-014-0173-7.2>
- Chang, R., Mu, X., & Zhang, L. (2011). Software defect prediction using Non-Negative Matrix Factorization. *Journal of Software*, 6(11 SPEC. ISSUE), 2114–2120. <https://doi.org/10.4304/jsw.6.11.2114-2120>
- Chiş, M. (2008). Evolutionary Decision Trees and Software Metrics for Module Defects Identification. *Program*, 2(2), 25–29.
- Demšar, J. (2006). Statistical Comparisons of Classifiers over Multiple Data Sets. *The Journal of Machine Learning Research*, 7, 1–30. <https://doi.org/10.1016/j.jecp.2010.03.005>
- Fakhrahmad, S. M., & Sami, a. (2009). Effective Estimation of Modules' Metrics in Software Defect Prediction. *World Congress on Engineering, I*, 206–211.
- Gorunescu, F. (2012). Data Mining: Concepts, Models and Techniques. In *Springer: Vol. XXXIII* (Issue 2). Sport Management Association of Australia and New Zealand. <https://doi.org/10.1007/s13398-014-0173-7.2>
- Gray, D., Bowes, D., Davey, N., Christianson, B., Sun, Y., & Christianson, B. (2011). The misuse of the NASA Metrics Data Program data sets for automated software defect prediction. *IET Seminar Digest*, 2011(1), 96–103. <https://doi.org/10.1049/ic.2011.0012>
- Hall, T., Beecham, S., Bowes, D., Gray, D., & Counsell, S. (2011). A Systematic Review of Fault Prediction Performance in Software Engineering. *IEEE Transactions on Software Engineering*, 38(6), 1276–1304. <https://doi.org/10.1109/TSE.2011.103>
- Han, J., Kamber, M., & Pei, J. (2012). Data Mining Concepts and Technique. In *Data Mining* (third, Issue Tentang Data Mining). Morgan Kaufmann. <https://doi.org/10.1007/s13398-014-0173-7.2>
- Harrington, P. (2012). *Machine Learning in Action*. <https://doi.org/10.1007/978-0-387-77242-4>

- Hosmer, D. W. (2000). *Applied Logistic Regression*. A Wiley-Interscience.
- Karsmakers, P., Pelckmans, K., & Suykens, J. a. K. (2007). Multi-class kernel logistic regression: A fixed-size implementation. *IEEE International Conference on Neural Networks - Conference Proceedings*, 1(3), 1756–1761. <https://doi.org/10.1109/IJCNN.2007.4371223>
- Khoshgoftaar, T. M., Gao, K., & Seliya, N. (2010). Attribute Selection and Imbalanced Data: Problems in Software Defect Prediction. *2010 22nd IEEE International Conference on Tools with Artificial Intelligence*, 137–144. <https://doi.org/10.1109/ICTAI.2010.27>
- Komarek, P., & Moore, A. (2005). Making Logistic Regression A Core Data Mining Tool A Practical Investigation of Accuracy , Speed , and Simplicity. *Compute*, 1, 1–4.
- Laradji, I. H. I., Alshayeb, M., & Ghouti, L. (2015). Software defect prediction using ensemble learning on selected features. *Information and Software Technology*, 58, 388–402. <https://doi.org/10.1016/j.infsof.2014.07.005>
- Lessmann, S., Baesens, B., Mues, C., & Pietsch, S. (2008). Benchmarking Classification Models for Software Defect Prediction: A Proposed Framework and Novel Findings. *IEEE Transactions on Software Engineering*, 34(4), 485–496. <https://doi.org/10.1109/TSE.2008.35>
- Lin, C.-J., Weng, R. C., & Keerthi, S. S. (2007). *Trust region Newton method for large-scale logistic regression*. 561–568.
- Maalouf, M., & Trafalis, T. B. (2011). Robust weighted kernel logistic regression in imbalanced and rare events data. *Computational Statistics and Data Analysis*, 55(1), 168–183. <https://doi.org/10.1016/j.csda.2010.06.014>
- McDonald, M., Musson, R., & Smith, R. (2008). The Practival Guide to Defect Prevention. In W. Press (Ed.), *Control*. Microsoft.
- Menzies, T., Greenwald, J., & Frank, A. (2007). Data Mining Static Code Attribute to Learn Defect Predictors. *IEEE Transactions on Software Engineering*, 33(1), 2–14. <https://doi.org/10.1109/TSE.2007.10>

- Naik, K., & Tripathy, P. (2008). AND QUALITY Theory and Practice. In *Theory and Practice*.
- Pressman, R. S. (2015). *Rekayasa Perangkat Lunak*. Andi.
- Saifudin, A., & Wahono, R. S. (2015). Penerapan Teknik Ensemble untuk Menangani Ketidakseimbangan Kelas pada Prediksi Cacat Software. *Journal of Software Engineering*, 1(1).
- Seiffert, C., Khoshgoftaar, T. M., Van Hulse, J., & Folleco, A. (2014). An empirical study of the classification performance of learners on imbalanced and noisy software quality data. *Information Sciences*, 259, 571–595. <https://doi.org/10.1016/j.ins.2010.12.016>
- Song, Q., Jia, Z., Shepperd, M., Ying, S., & Liu, J. (2011). A general software defect-proneness prediction framework. *IEEE Transactions on Software Engineering*, 37(3), 356–370. <https://doi.org/10.1109/TSE.2010.90>
- Wahono, R. S., Herman, N. S., & Ahmad, S. (2014). Neural Network Parameter Optimization Based on Genetic Algorithm for Software Defect Prediction. *Advanced Science Letters*, 20(10), 1951–1955. <https://doi.org/10.1166/asl.2014.5641>
- Witten, I. H., Frank, E., & Hall, M. a. (2011). Data Mining. In *Data Mining* (Vol. 277, Issue Tentang Data Mining). [https://doi.org/10.1002/1521-3773\(20010316\)40:6<9823::AID-ANIE9823>3.3.CO;2-C](https://doi.org/10.1002/1521-3773(20010316)40:6<9823::AID-ANIE9823>3.3.CO;2-C)
- Zhang, H., & Wang, Z. (2011). A normal distribution-based over-sampling approach to imbalanced data classification. *Advanced Data Mining and Applications*, 83–96. https://doi.org/10.1007/978-3-642-25853-4_7

LAMPIRAN – LAMPIRAN

Lampiran 1. Hasil Seleksi Fitur dengan Greedy Forward Selection

Dataset CM1

NO	ATRIBUT	WEIGHT
1	LOC_BLANK	0
2	BRANCH_COUNT	0
3	CALL_PAIRS	0
4	LOC_CODE_AND_COMMENT	1
5	LOC_COMMENTS	1
6	CONDITION_COUNT	0
7	CYCLOMATIC_COMPLEXITY	1
8	CYCLOMATIC_DENSITY	0
9	DECISION_COUNT	0
10	DECISION_DENSITY	0
11	DESIGN_COMPLEXITY	1
12	DESIGN_DENSITY	1
13	EDGE_COUNT	0
14	ESSENTIAL_COMPLEXITY	0
15	ESSENTIAL_DENSITY	1
16	LOC_EXECUTABLE	0
17	PARAMETER_COUNT	0
18	HALSTEAD_CONTENT	1
19	HALSTEAD_DIFFICULTY	0
20	HALSTEAD_EFFORT	0
21	HALSTEAD_ERROR_EST	0
22	HALSTEAD_LENGTH	0
23	HALSTEAD_LEVEL	0
24	HALSTEAD_PROG_TIME	0
25	HALSTEAD_VOLUME	0
26	MAINTENANCE_SEVERITY	1
27	MODIFIED_CONDITION_COUNT	0
28	MULTIPLE_CONDITION_COUNT	0
29	NODE_COUNT	0
30	NORMALIZED_CYLOMATIC_COMPLEXITY	1
31	NUM_OPERANDS	0
32	NUM_OPERATORS	0
33	NUM_UNIQUE_OPERANDS	0
34	NUM_UNIQUE_OPERATORS	0
35	NUMBER_OF_LINES	0
36	PERCENT_COMMENTS	1
37	LOC_TOTAL	0

Dataset JM1

NO	ATRIBUT	WEIGHT
1	LOC_BLANK	1
2	BRANCH_COUNT	0
3	LOC_CODE_AND_COMMENT	1
4	LOC_COMMENTS	1
5	CYCLOMATIC_COMPLEXITY	0
6	DESIGN_COMPLEXITY	0
7	ESSENTIAL_COMPLEXITY	0
8	LOC_EXECUTABLE	0
9	HALSTEAD_CONTENT	1
10	HALSTEAD_DIFFICULTY	0
11	HALSTEAD_EFFORT	1
12	HALSTEAD_ERROR_EST	0
13	HALSTEAD_LENGTH	0
14	HALSTEAD_LEVEL	1
15	HALSTEAD_PROG_TIME	1
16	HALSTEAD_VOLUME	0
17	NUM_OPERANDS	0
18	NUM_OPERATORS	0
19	NUM_UNIQUE_OPERANDS	1
20	NUM_UNIQUE_OPERATORS	1
21	LOC_TOTAL	1

Dataset KC1

NO	ATRIBUT	WEIGHT
1	LOC_BLANK	1
2	BRANCH_COUNT	0
3	LOC_CODE_AND_COMMENT	1
4	LOC_COMMENTS	1
5	CYCLOMATIC_COMPLEXITY	0
6	DESIGN_COMPLEXITY	0
7	ESSENTIAL_COMPLEXITY	0
8	LOC_EXECUTABLE	0
9	HALSTEAD_CONTENT	1
10	HALSTEAD_DIFFICULTY	0
11	HALSTEAD_EFFORT	1
12	HALSTEAD_ERROR_EST	1
13	HALSTEAD_LENGTH	0
14	HALSTEAD_LEVEL	1
15	HALSTEAD_PROG_TIME	1
16	HALSTEAD_VOLUME	0
17	NUM_OPERANDS	1
18	NUM_OPERATORS	1
19	NUM_UNIQUE_OPERANDS	0
20	NUM_UNIQUE_OPERATORS	0
21	LOC_TOTAL	0

Dataset KC3

NO	ATRIBUT	WEIGHT
1	LOC_BLANK	0
2	BRANCH_COUNT	0
3	CALL_PAIRS	1
4	LOC_CODE_AND_COMMENT	1
5	LOC_COMMENTS	0
6	CONDITION_COUNT	0
7	CYCLOMATIC_COMPLEXITY	0
8	CYCLOMATIC_DENSITY	0
9	DECISION_COUNT	0
10	DECISION_DENSITY	0
11	DESIGN_COMPLEXITY	0
12	DESIGN_DENSITY	0
13	EDGE_COUNT	0
14	ESSENTIAL_COMPLEXITY	0
15	ESSENTIAL_DENSITY	0
16	LOC_EXECUTABLE	0
17	PARAMETER_COUNT	0
18	GLOBAL_DATA_COMPLEXITY	1
19	GLOBAL_DATA_DENSITY	0
20	HALSTEAD_CONTENT	0
21	HALSTEAD_DIFFICULTY	0
22	HALSTEAD_EFFORT	0
23	HALSTEAD_ERROR_EST	0
24	HALSTEAD_LENGTH	0
25	HALSTEAD_LEVEL	0
26	HALSTEAD_PROG_TIME	1
27	HALSTEAD_VOLUME	0
28	MAINTENANCE_SEVERITY	1
29	MODIFIED_CONDITION_COUNT	1
30	MULTIPLE_CONDITION_COUNT	1
31	NODE_COUNT	0
32	NORMALIZED_CYLOMATIC_COMPLEXITY	1
33	NUM_OPERANDS	0
34	NUM_OPERATORS	0
35	NUM_UNIQUE_OPERANDS	1
36	NUM_UNIQUE_OPERATORS	0
37	NUMBER_OF_LINES	0
38	PERCENT_COMMENTS	0
39	LOC_TOTAL	1

Dataset MC2

NO	ATRIBUT	WEIGHT
1	LOC_BLANK	0
2	BRANCH_COUNT	0
3	CALL_PAIRS	1
4	LOC_CODE_AND_COMMENT	0
5	LOC_COMMENTS	1
6	CONDITION_COUNT	0
7	CYCLOMATIC_COMPLEXITY	0
8	CYCLOMATIC_DENSITY	0
9	DECISION_COUNT	1
10	DECISION_DENSITY	0
11	DESIGN_COMPLEXITY	0
12	DESIGN_DENSITY	1
13	EDGE_COUNT	0
14	ESSENTIAL_COMPLEXITY	0
15	ESSENTIAL_DENSITY	1
16	LOC_EXECUTABLE	0
17	PARAMETER_COUNT	1
18	GLOBAL_DATA_COMPLEXITY	0
19	GLOBAL_DATA_DENSITY	0
20	HALSTEAD_CONTENT	0
21	HALSTEAD_DIFFICULTY	1
22	HALSTEAD_EFFORT	1
23	HALSTEAD_ERROR_EST	0
24	HALSTEAD_LENGTH	0
25	HALSTEAD_LEVEL	0
26	HALSTEAD_PROG_TIME	0
27	HALSTEAD_VOLUME	0
28	MAINTENANCE_SEVERITY	1
29	MODIFIED_CONDITION_COUNT	0
30	MULTIPLE_CONDITION_COUNT	0
31	NODE_COUNT	0
32	NORMALIZED_CYLOMATIC_COMPLEXITY	0
33	NUM_OPERANDS	0
34	NUM_OPERATORS	0
35	NUM_UNIQUE_OPERANDS	0
36	NUM_UNIQUE_OPERATORS	0
37	NUMBER_OF_LINES	0
38	PERCENT_COMMENTS	1
39	LOC_TOTAL	0

Dataset PC1

NO	ATRIBUT	WEIGHT
1	LOC_BLANK	0
2	BRANCH_COUNT	0
3	CALL_PAIRS	0
4	LOC_CODE_AND_COMMENT	1
5	LOC_COMMENTS	0
6	CONDITION_COUNT	0
7	CYCLOMATIC_COMPLEXITY	0
8	CYCLOMATIC_DENSITY	0
9	DECISION_COUNT	0
10	DECISION_DENSITY	0
11	DESIGN_COMPLEXITY	0
12	DESIGN_DENSITY	1
13	EDGE_COUNT	0
14	ESSENTIAL_COMPLEXITY	0
15	ESSENTIAL_DENSITY	1
16	LOC_EXECUTABLE	0
17	PARAMETER_COUNT	1
18	HALSTEAD_CONTENT	0
19	HALSTEAD_DIFFICULTY	1
20	HALSTEAD_EFFORT	1
21	HALSTEAD_ERROR_EST	0
22	HALSTEAD_LENGTH	0
23	HALSTEAD_LEVEL	0
24	HALSTEAD_PROG_TIME	1
25	HALSTEAD_VOLUME	0
26	MAINTENANCE_SEVERITY	1
27	MODIFIED_CONDITION_COUNT	0
28	MULTIPLE_CONDITION_COUNT	0
29	NODE_COUNT	0
30	NORMALIZED_CYLOMATIC_COMPLEXITY	1
31	NUM_OPERANDS	0
32	NUM_OPERATORS	0
33	NUM_UNIQUE_OPERANDS	0
34	NUM_UNIQUE_OPERATORS	0
35	NUMBER_OF_LINES	1
36	PERCENT_COMMENTS	0
37	LOC_TOTAL	0

Dataset PC4

NO	ATRIBUT	WEIGHT
1	LOC_BLANK	0
2	BRANCH_COUNT	0
3	CALL_PAIRS	1
4	LOC_CODE_AND_COMMENT	1
5	LOC_COMMENTS	1
6	CONDITION_COUNT	0
7	CYCLOMATIC_COMPLEXITY	1
8	CYCLOMATIC_DENSITY	0
9	DECISION_COUNT	0
10	DECISION_DENSITY	0
11	DESIGN_COMPLEXITY	1
12	DESIGN_DENSITY	0
13	EDGE_COUNT	0
14	ESSENTIAL_COMPLEXITY	0
15	ESSENTIAL_DENSITY	1
16	LOC_EXECUTABLE	0
17	PARAMETER_COUNT	1
18	HALSTEAD_CONTENT	0
19	HALSTEAD_DIFFICULTY	0
20	HALSTEAD_EFFORT	0
21	HALSTEAD_ERROR_EST	0
22	HALSTEAD_LENGTH	0
23	HALSTEAD_LEVEL	0
24	HALSTEAD_PROG_TIME	0
25	HALSTEAD_VOLUME	0
26	MAINTENANCE_SEVERITY	1
27	MODIFIED_CONDITION_COUNT	0
28	MULTIPLE_CONDITION_COUNT	0
29	NODE_COUNT	1
30	NORMALIZED_CYLOMATIC_COMPLEXITY	1
31	NUM_OPERANDS	0
32	NUM_OPERATORS	0
33	NUM_UNIQUE_OPERANDS	0
34	NUM_UNIQUE_OPERATORS	0
35	NUMBER_OF_LINES	0
36	PERCENT_COMMENTS	0
37	LOC_TOTAL	0

Dataset PC5

NO	ATRIBUT	WEIGHT
1	LOC_BLANK	0
2	BRANCH_COUNT	0
3	CALL_PAIRS	0
4	LOC_CODE_AND_COMMENT	0
5	LOC_COMMENTS	0
6	CONDITION_COUNT	0
7	CYCLOMATIC_COMPLEXITY	1
8	CYCLOMATIC_DENSITY	1
9	DECISION_COUNT	1
10	DESIGN_COMPLEXITY	0
11	DESIGN_DENSITY	0
12	EDGE_COUNT	0
13	ESSENTIAL_COMPLEXITY	0
14	ESSENTIAL_DENSITY	0
15	LOC_EXECUTABLE	0
16	PARAMETER_COUNT	1
17	GLOBAL_DATA_COMPLEXITY	0
18	GLOBAL_DATA_DENSITY	1
19	HALSTEAD_CONTENT	0
20	HALSTEAD_DIFFICULTY	0
21	HALSTEAD_EFFORT	0
22	HALSTEAD_ERROR_EST	0
23	HALSTEAD_LENGTH	0
24	HALSTEAD_LEVEL	1
25	HALSTEAD_PROG_TIME	1
26	HALSTEAD_VOLUME	1
27	MAINTENANCE_SEVERITY	0
28	MODIFIED_CONDITION_COUNT	0
29	MULTIPLE_CONDITION_COUNT	0
30	NODE_COUNT	0
31	NORMALIZED_CYLOMATIC_COMPLEXITY	1
32	NUM_OPERANDS	0
33	NUM_OPERATORS	0
34	NUM_UNIQUE_OPERANDS	0
35	NUM_UNIQUE_OPERATORS	0
36	NUMBER_OF_LINES	0
37	PERCENT_COMMENTS	0
38	LOC_TOTAL	1

Lampiran 2 Rule Logistic Regression dengan Greedy Forward Selection dan Bagging

```
//model1
hasil1=(1/(Math.sqrt(2*Math.PI*6.87790228708995)))*(Math.exp(-
1*((Math.pow((blank-
4.82213438735177),2))/(2*(Math.pow(6.87790228708995,2))))));
hasil2=(1/(Math.sqrt(2*Math.PI*0.664471456940689)))*(Math.exp(-
1*(Math.pow((code-
0.118715083798882),2))/(2*(Math.pow(0.664471456940689,2))))));
hasil3=(1/(Math.sqrt(2*Math.PI*1.99327666341805)))*(Math.exp(-
1*(Math.pow((comment-
0.630586592178771),2))/(2*(Math.pow(1.99327666341805,2))))));
hasil4=(1/(Math.sqrt(2*Math.PI*11327.8369059779)))*Math.exp(-
1*(Math.pow((effort-
3054.76275837989),2))/(2*(Math.pow(11327.8369059779,2)))));
hasil5=(1/(Math.sqrt(2*Math.PI*0.122763797252566)))*Math.exp(-
1*(Math.pow((est-
0.0593156424581007),2))/(2*(Math.pow(0.122763797252566,2)))));
hasil6=(1/(Math.sqrt(2*Math.PI*62.6447405836545)))*Math.exp(-
1*(Math.pow((length-
36.6536312849162),2))/(2*(Math.pow(62.6447405836545,2)))));
hasil7=(1/(Math.sqrt(2*Math.PI*0.330595446699861)))*Math.exp(-
1*(Math.pow((level-
0.355656424581006),2))/(2*(Math.pow(0.330595446699861,2)))));
hasil8=(1/(Math.sqrt(2*Math.PI*629.324166717279)))*Math.exp(-
1*(Math.pow((time-
169.709546089386),2))/(2*(Math.pow(629.324166717279,2)))));
hasil9=(1/(Math.sqrt(2*Math.PI*4.96117996152181)))*Math.exp(-
1*(Math.pow((operators-
6.69483240223463),2))/(2*(Math.pow(4.96117996152181,2)))));
hasil10=(1/(Math.sqrt(2*Math.PI*22.5666319101371)))*Math.exp(-
1*(Math.pow((total-
15.7032122905027),2))/(2*(Math.pow(22.5666319101371,2)))));
hasilallno1=hasil1*hasil2*hasil3*hasil4*hasil5*hasil6*hasil7*hasil8*hasil9*hasil
10*0.844;

hasil1=(1/(Math.sqrt(2*Math.PI)*6.89322717624868))*Math.exp(-
1*Math.pow((blank-
4.86792452830188),2)/(2*Math.pow(6.89322717624868,2)));
hasil2=(1/(Math.sqrt(2*Math.PI)*0.53787375479812))*Math.exp(-
1*Math.pow((code-
0.118715083798882),2)/(2*Math.pow(0.53787375479812,2)));
hasil3=(1/(Math.sqrt(2*Math.PI)*5.89016342364392))*Math.exp(-
1*Math.pow((comment-
2.6377358490566),2)/(2*Math.pow(5.89016342364392,2)));
hasil4=(1/(Math.sqrt(2*Math.PI)*24912.5541276703))*Math.exp(-
1*Math.pow((effort-
```

```

15357.9259245282),2)/(2*Math.pow(24912.5541276703, 2)));
hasil5=(1/(Math.sqrt(2*Math.PI)*0.243873256502795))*Math.exp(-
1*Math.pow((est-
0.221962264150943),2)/(2*Math.pow(0.243873256502795, 2)));
hasil6=(1/(Math.sqrt(2*Math.PI)*114.351937870255))*Math.exp(-
1*Math.pow((length-
119.241509433962),2)/(2*Math.pow(114.351937870255, 2)));
hasil7=(1/(Math.sqrt(2*Math.PI)*0.165587498181568))*Math.exp(-
1*Math.pow((level-
0.137849056603773),2)/(2*Math.pow(0.165587498181568, 2)));
hasil8=(1/(Math.sqrt(2*Math.PI)*1384.03079512391))*Math.exp(-
1*Math.pow((time-
853.217660377358),2)/(2*Math.pow(1384.03079512391, 2)));
hasil9=(1/(Math.sqrt(2*Math.PI)*6.50586444337087))*Math.exp(-
1*Math.pow((operators-
12.8792452830188),2)/(2*Math.pow(6.50586444337087, 2)));
hasil10=(1/(Math.sqrt(2*Math.PI)*42.6177976012864))*Math.exp(-
1*Math.pow((total-
44.8943396226415),2)/(2*Math.pow(42.6177976012864,
2)));
hasilalloyes1=hasil1*hasil2*hasil3*hasil4*hasil5*hasil6*hasil7*hasil8*hasil9*hasi
l10*0.156; if (hasilallno1>hasilalloyes1)
{
hasil_t=hasil_t+1;
}
else if (hasilallno1<hasilalloyes1)
{
hasil_y=hasil_y+1;
}
}

//model2
hasil1=(1/(Math.sqrt(2*Math.PI)*2.68843420084555))*Math.exp(-
1*Math.pow((blank-
1.21394064872325),2)/(2*Math.pow(2.68843420084555, 2)));
hasil2=(1/(Math.sqrt(2*Math.PI)*0.67877250307027))*Math.exp(-
1*Math.pow((code-
0.125603864734299),2)/(2*Math.pow(0.67877250307027, 2)));
hasil3=(1/(Math.sqrt(2*Math.PI)*2.05603247080098))*Math.exp(-
1*Math.pow((comment-
0.646652864044168),2)/(2*Math.pow(2.05603247080098, 2)));
hasil4=(1/(Math.sqrt(2*Math.PI)*14116.9141854623))*Math.exp(-
1*Math.pow((effort-
3307.9554658385),2)/(2*Math.pow(14116.9141854623, 2)));
hasil5=(1/(Math.sqrt(2*Math.PI)*0.140132021671918))*Math.exp(-
1*Math.pow((est-
0.0617322291235336),2)/(2*Math.pow(0.140132021671918, 2)));
hasil6=(1/(Math.sqrt(2*Math.PI)*68.7206796448543))*Math.exp(-
1*Math.pow((length-
37.8219461697722),2)/(2*Math.pow(68.7206796448543, 2)));

```



```

hasil7=(1/(Math.sqrt(2*Math.PI)*0.325742992983394))*Math.exp(-
1*Math.pow((level-
0.355797101449276),2)/(2*Math.pow(0.325742992983394, 2)));
hasil8=(1/(Math.sqrt(2*Math.PI)*784.27287956944))*Math.exp(-
1*Math.pow((time-
183.775776397515),2)/(2*Math.pow(784.27287956944, 2)));
hasil9=(1/(Math.sqrt(2*Math.PI)*4.97648702360752))*Math.exp(-
1*Math.pow((operators-
6.76259489302967),2)/(2*Math.pow(4.97648702360752, 2)));
hasil10=(1/(Math.sqrt(2*Math.PI)*23.5236226640102))*Math.exp(-
1*Math.pow((total-
15.9440993788819),2)/(2*Math.pow(23.5236226640102, 2)));
hasilallno2=hasil1*hasil2*hasil3*hasil4*hasil5*hasil6*hasil7*hasil8*hasil9*hasil
10*0.854;

```

```

hasil1=(1/(Math.sqrt(2*Math.PI)*7.07702980922972))*Math.exp(-
1*Math.pow((blank-
4.81854838709677),2)/(2*Math.pow(7.07702980922972, 2)));
hasil2=(1/(Math.sqrt(2*Math.PI)*0.644531248454078))*Math.exp(-
1*Math.pow((code-
0.133064516129032),2)/(2*Math.pow(0.644531248454078, 2)));
hasil3=(1/(Math.sqrt(2*Math.PI)*6.52097884352546))*Math.exp(-
1*Math.pow((comment-
2.91532258064516),2)/(2*Math.pow(6.52097884352546, 2)));
hasil4=(1/(Math.sqrt(2*Math.PI)*27753.313571283))*Math.exp(-
1*Math.pow((effort-
16171.1501612903),2)/(2*Math.pow(27753.313571283, 2)));
hasil5=(1/(Math.sqrt(2*Math.PI)*0.259967850455032))*Math.exp(-
1*Math.pow((est-
0.226935483870967),2)/(2*Math.pow(0.259967850455032, 2)));
hasil6=(1/(Math.sqrt(2*Math.PI)*121.079013246473))*Math.exp(-
1*Math.pow((length-
121.149193548387),2)/(2*Math.pow(121.079013246473, 2)));
hasil7=(1/(Math.sqrt(2*Math.PI)*0.170012156837796))*Math.exp(-
1*Math.pow((level-
0.145564516129032),2)/(2*Math.pow(0.170012156837796, 2)));
hasil8=(1/(Math.sqrt(2*Math.PI)*1541.85071293541))*Math.exp(-
1*Math.pow((time-
898.396653225806),2)/(2*Math.pow(1541.85071293541, 2)));
hasil9=(1/(Math.sqrt(2*Math.PI)*6.4587162557225))*Math.exp(-
1*Math.pow((operators-
12.633064516129),2)/(2*Math.pow(6.4587162557225, 2)));
hasil10=(1/(Math.sqrt(2*Math.PI)*46.146718791453))*Math.exp(-
1*Math.pow((total-
46.2258064516129),2)/(2*Math.pow(46.146718791453,
2)));
hasilalloyes2=hasil1*hasil2*hasil3*hasil4*hasil5*hasil6*hasil7*hasil8*hasil9*hasi
110*0.146; if (hasilallno2>hasilalloyes2)
{

```



```

hasil_t=hasil_t+1;
}
else if (hasilallno2<hasilallyes2)
{
hasil_y=hasil_y+1;
}

//model3
hasil1=(1/(Math.sqrt(2*Math.PI)*2.74363725914337))*Math.exp(-
1*Math.pow((blank-
1.23574408901251),2)/(2*Math.pow(2.74363725914337, 2)));
hasil2=(1/(Math.sqrt(2*Math.PI)*0.727943455699066))*Math.exp(-
1*Math.pow((code-
0.125173852573018),2)/(2*Math.pow(0.727943455699066, 2)));
hasil3=(1/(Math.sqrt(2*Math.PI)*2.090136139231))*Math.exp(-
1*Math.pow((comment-
0.666203059805285),2)/(2*Math.pow(2.090136139231, 2)));
hasil4=(1/(Math.sqrt(2*Math.PI)*13462.9912826757))*Math.exp(-
1*Math.pow((effort-
3245.41945757997),2)/(2*Math.pow(13462.9912826757, 2)));
hasil5=(1/(Math.sqrt(2*Math.PI)*0.139328566667417))*Math.exp(-
1*Math.pow((est-
0.0618289290681503),2)/(2*Math.pow(0.139328566667417, 2)));
hasil6=(1/(Math.sqrt(2*Math.PI)*68.2726018795086))*Math.exp(-
1*Math.pow((length-
37.8289290681502),2)/(2*Math.pow(68.2726018795086, 2)));
hasil7=(1/(Math.sqrt(2*Math.PI)*0.324421703030615))*Math.exp(-
1*Math.pow((level-
0.353977746870654),2)/(2*Math.pow(0.324421703030615, 2)));
hasil8=(1/(Math.sqrt(2*Math.PI)*747.9439051915))*Math.exp(-
1*Math.pow((time-
180.301481223923),2)/(2*Math.pow(747.9439051915, 2)));
hasil9=(1/(Math.sqrt(2*Math.PI)*4.97647793722628))*Math.exp(-
1*Math.pow((operators-
6.75799721835883),2)/(2*Math.pow(4.97647793722628, 2)));
hasil10=(1/(Math.sqrt(2*Math.PI)*23.9842028999842))*Math.exp(-
1*Math.pow((total-
16.0757997218358),2)/(2*Math.pow(23.9842028999842, 2)));
hasilallno3=hasil1*hasil2*hasil3*hasil4*hasil5*hasil6*hasil7*hasil8*hasil9*hasil
10*0.847;
hasil1=(1/(Math.sqrt(2*Math.PI)*6.90806190808757))*Math.exp(-
1*Math.pow((blank-
4.76833976833976),2)/(2*Math.pow(6.90806190808757, 2)));
hasil2=(1/(Math.sqrt(2*Math.PI)*0.692879073369926))*Math.exp(-
1*Math.pow((code-
0.166023166023166),2)/(2*Math.pow(0.692879073369926, 2)));
hasil3=(1/(Math.sqrt(2*Math.PI)*6.37630054998353))*Math.exp(-
1*Math.pow((comment-

```

```

2.77220077220077),2)/(2*Math.pow(6.37630054998353, 2)));
hasil4=(1/(Math.sqrt(2*Math.PI)*27189.430943005))*Math.exp(-
1*Math.pow((effort-
15748.6522779922),2)/(2*Math.pow(27189.430943005, 2)));
hasil5=(1/(Math.sqrt(2*Math.PI)*0.255484435042608))*Math.exp(-
1*Math.pow((est-
0.221158301158301),2)/(2*Math.pow(0.255484435042608, 2)));
hasil6=(1/(Math.sqrt(2*Math.PI)*119.207930246371))*Math.exp(-
1*Math.pow((length-
118.305019305019),2)/(2*Math.pow(119.207930246371, 2)));
hasil7=(1/(Math.sqrt(2*Math.PI)*0.169464990978138))*Math.exp(-
1*Math.pow((level-
0.145714285714285),2)/(2*Math.pow(0.169464990978138, 2)));
hasil8=(1/(Math.sqrt(2*Math.PI)*1510.52387176074))*Math.exp(-
1*Math.pow((time-
874.924633204633),2)/(2*Math.pow(1510.52387176074, 2)));
hasil9=(1/(Math.sqrt(2*Math.PI)*6.64340769204876))*Math.exp(-
1*Math.pow((operators-
12.5868725868725),2)/(2*Math.pow(6.64340769204876, 2)));
hasil10=(1/(Math.sqrt(2*Math.PI)*45.0959678251634))*Math.exp(-
1*Math.pow((total-
45.069498069498),2)/(2*Math.pow(45.0959678251634,
2)));
hasilalloyes3=hasil1*hasil2*hasil3*hasil4*hasil5*hasil6*hasil7*hasil8*hasil9*hasi
l10*0.153; if (hasilallno3>hasilalloyes3)
{
hasil_t=hasil_t+1;
}
else if (hasilallno3<hasilalloyes3)
{
hasil_y=hasil_y+1;
}

//model4
hasil1=(1/(Math.sqrt(2*Math.PI)*2.73816913018312))*Math.exp(-
1*Math.pow((blank-
1.21990257480862),2)/(2*Math.pow(2.73816913018312, 2)));
hasil2=(1/(Math.sqrt(2*Math.PI)*0.742616789453064))*Math.exp(-
1*Math.pow((code-
0.129436325678496),2)/(2*Math.pow(0.742616789453064, 2)));
hasil3=(1/(Math.sqrt(2*Math.PI)*2.09914136569346))*Math.exp(-
1*Math.pow((comment-
0.674321503131524),2)/(2*Math.pow(2.09914136569346, 2)));
hasil4=(1/(Math.sqrt(2*Math.PI)*14162.4647548881))*Math.exp(-
1*Math.pow((effort-
3338.42607515657),2)/(2*Math.pow(14162.4647548881, 2)));
hasil5=(1/(Math.sqrt(2*Math.PI)*0.141420883697082))*Math.exp(-
1*Math.pow((est-

```



```

hasil6=(1/(Math.sqrt(2*Math.PI)*69.434211324101))*Math.exp(-
1*Math.pow((length-
38.1503131524008),2)/(2*Math.pow(69.434211324101, 2)));
hasil7=(1/(Math.sqrt(2*Math.PI)*0.327802337254059))*Math.exp(-
1*Math.pow((level-
0.354585942936674),2)/(2*Math.pow(0.327802337254059, 2)));
hasil8=(1/(Math.sqrt(2*Math.PI)*786.80350128501))*Math.exp(-
1*Math.pow((time-
185.468538622129),2)/(2*Math.pow(786.80350128501, 2)));
hasil9=(1/(Math.sqrt(2*Math.PI)*4.9237161252461))*Math.exp(-
1*Math.pow((operators-
6.73347251217814),2)/(2*Math.pow(4.9237161252461, 2)));
hasil10=(1/(Math.sqrt(2*Math.PI)*23.8968140027683))*Math.exp(-
1*Math.pow((total-
16.1398747390396),2)/(2*Math.pow(23.8968140027683, 2)));
hasilallno4=hasil1*hasil2*hasil3*hasil4*hasil5*hasil6*hasil7*hasil8*hasil9*hasil
10*0.847;

```

```

hasil1=(1/(Math.sqrt(2*Math.PI)*7.02668341212835))*Math.exp(-
1*Math.pow((blank-
4.98461538461538),2)/(2*Math.pow(7.02668341212835, 2)));
hasil2=(1/(Math.sqrt(2*Math.PI)*0.689735511435416))*Math.exp(-
1*Math.pow((code-
0.161538461538461),2)/(2*Math.pow(0.689735511435416, 2)));
hasil3=(1/(Math.sqrt(2*Math.PI)*6.41796719688586))*Math.exp(-
1*Math.pow((comment-
2.86538461538461),2)/(2*Math.pow(6.41796719688586, 2)));
hasil4=(1/(Math.sqrt(2*Math.PI)*27181.2422439513))*Math.exp(-
1*Math.pow((effort-
16317.4843846153),2)/(2*Math.pow(27181.2422439513, 2)));
hasil5=(1/(Math.sqrt(2*Math.PI)*0.25374620864584))*Math.exp(-
1*Math.pow((est-
0.227346153846153),2)/(2*Math.pow(0.25374620864584, 2)));
hasil6=(1/(Math.sqrt(2*Math.PI)*118.241378969562))*Math.exp(-
1*Math.pow((length-
121.584615384615),2)/(2*Math.pow(118.241378969562, 2)));
hasil7=(1/(Math.sqrt(2*Math.PI)*0.167408114156622))*Math.exp(-
1*Math.pow((level-
0.139923076923076),2)/(2*Math.pow(0.167408114156622, 2)));
hasil8=(1/(Math.sqrt(2*Math.PI)*1510.068887751))*Math.exp(-
1*Math.pow((time-
906.526423076922),2)/(2*Math.pow(1510.068887751, 2)));
hasil9=(1/(Math.sqrt(2*Math.PI)*6.64545730270707))*Math.exp(-
1*Math.pow((operators-
12.9923076923076),2)/(2*Math.pow(6.64545730270707, 2)));
hasil10=(1/(Math.sqrt(2*Math.PI)*44.8891706850064))*Math.exp(-
1*Math.pow((total-

```



```

45.75),2)/(2*Math.pow(44.8891706850064,
2)));
hasilallyes4=hasil1*hasil2*hasil3*hasil4*hasil5*hasil6*hasil7*hasil8*hasil9*hasi
l10*0.153; if (hasilallno4>hasilallyes4)
{
hasil_t=hasil_t+1;
}
else if (hasilallno4<hasilallyes4)
{
hasil_y=hasil_y+1;
}

//model5
hasil1=(1/(Math.sqrt(2*Math.PI)*2.78073431664639))*Math.exp(-
1*Math.pow((blank-
1.21438547486033),2)/(2*Math.pow(2.78073431664639, 2)));
hasil2=(1/(Math.sqrt(2*Math.PI)*0.72834313504331))*Math.exp(-
1*Math.pow((code-
0.123603351955307),2)/(2*Math.pow(0.72834313504331, 2)));
hasil3=(1/(Math.sqrt(2*Math.PI)*1.99659466229907))*Math.exp(-
1*Math.pow((comment-
0.616620111731843),2)/(2*Math.pow(1.99659466229907, 2)));
hasil4=(1/(Math.sqrt(2*Math.PI)*14335.7853450659))*Math.exp(-
1*Math.pow((effort-
3313.84117318436),2)/(2*Math.pow(14335.7853450659, 2)));
hasil5=(1/(Math.sqrt(2*Math.PI)*0.142055174398488))*Math.exp(-
1*Math.pow((est-
0.0610195530726258),2)/(2*Math.pow(0.142055174398488, 2)));
hasil6=(1/(Math.sqrt(2*Math.PI)*69.5712682263653))*Math.exp(-
1*Math.pow((length-
37.3526536312849),2)/(2*Math.pow(69.5712682263653, 2)));
hasil7=(1/(Math.sqrt(2*Math.PI)*0.330194049312648))*Math.exp(-
1*Math.pow((level-

hasil8=(1/(Math.sqrt(2*Math.PI)*796.432417227296))*Math.exp(-
1*Math.pow((time-
184.102765363129),2)/(2*Math.pow(796.432417227296, 2)));
hasil9=(1/(Math.sqrt(2*Math.PI)*4.93179572837506))*Math.exp(-
1*Math.pow((operators-
6.67597765363128),2)/(2*Math.pow(4.93179572837506, 2)));
hasil10=(1/(Math.sqrt(2*Math.PI)*23.8745144748757))*Math.exp(-
1*Math.pow((total-
15.7856145251396),2)/(2*Math.pow(23.8745144748757, 2)));
hasilallno5=hasil1*hasil2*hasil3*hasil4*hasil5*hasil6*hasil7*hasil8*hasil9*hasil
10*0.844;

```

```

hasil1=(1/(Math.sqrt(2*Math.PI)*6.80463930464529))*Math.exp(-
1*Math.pow((blank-
4.8377358490566),2)/(2*Math.pow(6.80463930464529, 2)));
hasil2=(1/(Math.sqrt(2*Math.PI)*0.685406983470609))*Math.exp(-
1*Math.pow((code-
0.162264150943396),2)/(2*Math.pow(0.685406983470609, 2)));
hasil3=(1/(Math.sqrt(2*Math.PI)*6.26654927325781))*Math.exp(-
1*Math.pow((comment-
2.80754716981132),2)/(2*Math.pow(6.26654927325781, 2)));
hasil4=(1/(Math.sqrt(2*Math.PI)*26734.1600856878))*Math.exp(-
1*Math.pow((effort-
15924.513245283),2)/(2*Math.pow(26734.1600856878, 2)));
hasil5=(1/(Math.sqrt(2*Math.PI)*0.245033993744413))*Math.exp(-
1*Math.pow((est-
0.222679245283018),2)/(2*Math.pow(0.245033993744413, 2)));
hasil6=(1/(Math.sqrt(2*Math.PI)*114.803312297946))*Math.exp(-
1*Math.pow((length-
119.720754716981),2)/(2*Math.pow(114.803312297946, 2)));
hasil7=(1/(Math.sqrt(2*Math.PI)*0.160059960534377))*Math.exp(-
1*Math.pow((level-
0.138226415094339),2)/(2*Math.pow(0.160059960534377, 2)));
hasil8=(1/(Math.sqrt(2*Math.PI)*1485.23106094947))*Math.exp(-
1*Math.pow((time-
884.694716981132),2)/(2*Math.pow(1485.23106094947, 2)));
hasil9=(1/(Math.sqrt(2*Math.PI)*6.48785644905943))*Math.exp(-
1*Math.pow((operators-
12.9509433962264),2)/(2*Math.pow(6.48785644905943, 2)));
hasil10=(1/(Math.sqrt(2*Math.PI)*43.5469167940068))*Math.exp(-
1*Math.pow((total-
45.588679245283),2)/(2*Math.pow(43.5469167940068, 2)));
hasilallyes5=hasil1*hasil2*hasil3*hasil4*hasil5*hasil6*hasil7*hasil8*hasil9*hasi
l10*0.156; if (hasilallno5>hasilallyes5)
{
hasil_t=hasil_t+1;
}
else if (hasilallno5<hasilallyes5)
{
hasil_y=hasil_y+1;
}

//model6
hasil1=(1/(Math.sqrt(2*Math.PI)*2.78471203305847))*Math.exp(-
1*Math.pow((blank-
1.21653819201121),2)/(2*Math.pow(2.78471203305847, 2)));
hasil2=(1/(Math.sqrt(2*Math.PI)*0.739916873192768))*Math.exp(-
1*Math.pow((code-
0.133146461107217),2)/(2*Math.pow(0.739916873192768, 2)));

```



```

hasil3=(1/(Math.sqrt(2*Math.PI)*2.045046734438))*Math.exp(-
1*Math.pow((comment-
0.66082690960056),2)/(2*Math.pow(2.045046734438, 2)));
hasil4=(1/(Math.sqrt(2*Math.PI)*14359.253184504))*Math.exp(-
1*Math.pow((effort-
3349.76784863349),2)/(2*Math.pow(14359.253184504, 2)));
hasil5=(1/(Math.sqrt(2*Math.PI)*0.142523518589285))*Math.exp(-
1*Math.pow((est-
0.0618009810791872),2)/(2*Math.pow(0.142523518589285, 2)));
hasil6=(1/(Math.sqrt(2*Math.PI)*69.7935600396447))*Math.exp(-
1*Math.pow((length-
37.7421163279607),2)/(2*Math.pow(69.7935600396447, 2)));
hasil7=(1/(Math.sqrt(2*Math.PI)*0.328544952324315))*Math.exp(-
1*Math.pow((level-
0.354414856341976),2)/(2*Math.pow(0.328544952324315, 2)));
hasil8=(1/(Math.sqrt(2*Math.PI)*797.736205546187))*Math.exp(-
1*Math.pow((time-
186.09866152768),2)/(2*Math.pow(797.736205546187, 2)));
hasil9=(1/(Math.sqrt(2*Math.PI)*4.92979975704657))*Math.exp(-
1*Math.pow((operators-
hasil10=(1/(Math.sqrt(2*Math.PI)*24.0369438007679))*Math.exp(-
1*Math.pow((total-
15.9334267694463),2)/(2*Math.pow(24.0369438007679, 2)));
hasilallno6=hasil1*hasil2*hasil3*hasil4*hasil5*hasil6*hasil7*hasil8*hasil9*hasil
10*0.841;
hasil1=(1/(Math.sqrt(2*Math.PI)*6.59330139582277))*Math.exp(-
1*Math.pow((blank-
4.688888888888888),2)/(2*Math.pow(6.59330139582277, 2)));
hasil2=(1/(Math.sqrt(2*Math.PI)*0.645054476330152))*Math.exp(-
1*Math.pow((code-
0.137037037037037),2)/(2*Math.pow(0.645054476330152, 2)));
hasil3=(1/(Math.sqrt(2*Math.PI)*5.78353857629089))*Math.exp(-
1*Math.pow((comment-
2.644444444444444),2)/(2*Math.pow(5.78353857629089, 2)));
hasil4=(1/(Math.sqrt(2*Math.PI)*26042.7481044745))*Math.exp(-
1*Math.pow((effort-
15501.82922222222),2)/(2*Math.pow(26042.7481044745, 2)));
hasil5=(1/(Math.sqrt(2*Math.PI)*0.240629094839373))*Math.exp(-
1*Math.pow((est-
0.217592592592592),2)/(2*Math.pow(0.240629094839373, 2)));
hasil6=(1/(Math.sqrt(2*Math.PI)*113.245493087053))*Math.exp(-
1*Math.pow((length-
117.274074074074),2)/(2*Math.pow(113.245493087053, 2)));
hasil7=(1/(Math.sqrt(2*Math.PI)*0.169467600208147))*Math.exp(-
1*Math.pow((level-
0.14237037037037),2)/(2*Math.pow(0.169467600208147, 2)));
hasil8=(1/(Math.sqrt(2*Math.PI)*1446.81926111153))*Math.exp(-
1*Math.pow((time-

```



```

861.212296296296),2)/(2*Math.pow(1446.81926111153, 2)));
hasil9=(1/(Math.sqrt(2*Math.PI)*6.64511593158335))*Math.exp(-
1*Math.pow((operators-
12.7925925925925),2)/(2*Math.pow(6.64511593158335, 2)));
hasil10=(1/(Math.sqrt(2*Math.PI)*42.810056485039))*Math.exp(-
1*Math.pow((total-
44.4407407407407),2)/(2*Math.pow(42.810056485039,
2)));
hasilalloyes6=hasil1*hasil2*hasil3*hasil4*hasil5*hasil6*hasil7*hasil8*hasil9*hasi
l10*0.159; if (hasilallno6>hasilalloyes6)
{
hasil_t=hasil_t+1;
}
else if (hasilallno6<hasilalloyes6)
{
hasil_y=hasil_y+1;
}

//model7
hasil1=(1/(Math.sqrt(2*Math.PI)*2.76265583021302))*Math.exp(-
1*Math.pow((blank-
1.22330097087378),2)/(2*Math.pow(2.76265583021302, 2)));
hasil2=(1/(Math.sqrt(2*Math.PI)*0.727896400505417))*Math.exp(-
1*Math.pow((code-
0.130374479889043),2)/(2*Math.pow(0.727896400505417, 2)));
hasil3=(1/(Math.sqrt(2*Math.PI)*2.00853708144343))*Math.exp(-
1*Math.pow((comment-
0.637309292649098),2)/(2*Math.pow(2.00853708144343, 2)));
hasil4=(1/(Math.sqrt(2*Math.PI)*14339.3530969646))*Math.exp(-
1*Math.pow((effort-
3343.77042302357),2)/(2*Math.pow(14339.3530969646, 2)));
hasil5=(1/(Math.sqrt(2*Math.PI)*0.141852693101706))*Math.exp(-
1*Math.pow((est-
0.0614493758668517),2)/(2*Math.pow(0.141852693101706, 2)));
hasil6=(1/(Math.sqrt(2*Math.PI)*69.4771063287401))*Math.exp(-
1*Math.pow((length-
37.5700416088765),2)/(2*Math.pow(69.4771063287401, 2)));
hasil7=(1/(Math.sqrt(2*Math.PI)*0.32823656531661))*Math.exp(-
1*Math.pow((level-
0.35497226074896),2)/(2*Math.pow(0.32823656531661, 2)));
hasil8=(1/(Math.sqrt(2*Math.PI)*796.630645297007))*Math.exp(-
1*Math.pow((time-
185.765464632455),2)/(2*Math.pow(796.630645297007, 2)));
hasil9=(1/(Math.sqrt(2*Math.PI)*4.97554655385741))*Math.exp(-
1*Math.pow((operators-
6.72191400832177),2)/(2*Math.pow(4.97554655385741, 2)));
hasil10=(1/(Math.sqrt(2*Math.PI)*23.8934571888131))*Math.exp(-
1*Math.pow((total-
15.879334257975),2)/(2*Math.pow(23.8934571888131, 2)));

```

```

hasilallno7=hasil1*hasil2*hasil3*hasil4*hasil5*hasil6*hasil7*hasil8*hasil9*hasil
10*0.850;
hasil1=(1/(Math.sqrt(2*Math.PI)*6.97117291523538))*Math.exp(-
1*Math.pow((blank-
4.81960784313725),2)/(2*Math.pow(6.97117291523538, 2)));
hasil2=(1/(Math.sqrt(2*Math.PI)*0.674715958169612))*Math.exp(-
1*Math.pow((code-
0.145098039215686),2)/(2*Math.pow(0.674715958169612, 2)));
hasil3=(1/(Math.sqrt(2*Math.PI)*6.36810700161397))*Math.exp(-
1*Math.pow((comment-
2.76078431372549),2)/(2*Math.pow(6.36810700161397, 2)));
hasil4=(1/(Math.sqrt(2*Math.PI)*27546.2840683259))*Math.exp(-
1*Math.pow((effort-
16042.6183137254),2)/(2*Math.pow(27546.2840683259, 2)));
hasil5=(1/(Math.sqrt(2*Math.PI)*0.255325111298266))*Math.exp(-
1*Math.pow((est-
0.222588235294117),2)/(2*Math.pow(0.255325111298266, 2)));
hasil6=(1/(Math.sqrt(2*Math.PI)*118.878217129763))*Math.exp(-
1*Math.pow((length-
119.16862745098),2)/(2*Math.pow(118.878217129763, 2)));
hasil7=(1/(Math.sqrt(2*Math.PI)*0.145292696229308))*Math.exp(-
1*Math.pow((level-
0.136588235294117),2)/(2*Math.pow(0.145292696229308, 2)));
hasil8=(1/(Math.sqrt(2*Math.PI)*1530.34902165581))*Math.exp(-
1*Math.pow((time-
891.256117647059),2)/(2*Math.pow(1530.34902165581, 2)));
hasil9=(1/(Math.sqrt(2*Math.PI)*6.58666761052337))*Math.exp(-
1*Math.pow((operators-
12.8078431372549),2)/(2*Math.pow(6.58666761052337, 2)));
hasil10=(1/(Math.sqrt(2*Math.PI)*45.0383803527791))*Math.exp(-
1*Math.pow((total-
45.1686274509803),2)/(2*Math.pow(45.0383803527791, 2)));
hasilallyes7=hasil1*hasil2*hasil3*hasil4*hasil5*hasil6*hasil7*hasil8*hasil9*hasi
110*0.150; if (hasilallno7>hasilallyes7)
{
hasil_t=hasil_t+1;
}
else if (hasilallno7<hasilallyes7)
{
hasil_y=hasil_y+1;
}

//model8
hasil1=(1/(Math.sqrt(2*Math.PI)*2.72306295717564))*Math.exp(-
1*Math.pow((blank-
1.2134909596662),2)/(2*Math.pow(2.72306295717564, 2)));
hasil2=(1/(Math.sqrt(2*Math.PI)*0.747235403391362))*Math.exp(-
1*Math.pow((code-

```



```

0.133518776077885),2)/(2*Math.pow(0.747235403391362, 2)));
hasil3=(1/(Math.sqrt(2*Math.PI)*1.96619438172516))*Math.exp(-
1*Math.pow((comment-
0.63490959666203),2)/(2*Math.pow(1.96619438172516, 2)));
hasil4=(1/(Math.sqrt(2*Math.PI)*14209.5143494184))*Math.exp(-
1*Math.pow((effort-
3336.86630041724),2)/(2*Math.pow(14209.5143494184, 2)));
hasil5=(1/(Math.sqrt(2*Math.PI)*0.141387913490776))*Math.exp(-
1*Math.pow((est-
0.0619541029207234),2)/(2*Math.pow(0.141387913490776, 2)));
hasil6=(1/(Math.sqrt(2*Math.PI)*69.3232354346341))*Math.exp(-
1*Math.pow((length-
37.8810848400556),2)/(2*Math.pow(69.3232354346341, 2)));
hasil7=(1/(Math.sqrt(2*Math.PI)*0.326057390207606))*Math.exp(-
1*Math.pow((level-
0.354026425591099),2)/(2*Math.pow(0.326057390207606, 2)));
hasil8=(1/(Math.sqrt(2*Math.PI)*789.417332067501))*Math.exp(-
1*Math.pow((time-
185.381912378303),2)/(2*Math.pow(789.417332067501, 2)));
hasil9=(1/(Math.sqrt(2*Math.PI)*4.9882473035728))*Math.exp(-
1*Math.pow((operators-
6.74339360222531),2)/(2*Math.pow(4.9882473035728, 2)));
hasil10=(1/(Math.sqrt(2*Math.PI)*23.8033423361333))*Math.exp(-
1*Math.pow((total-
15.9923504867872),2)/(2*Math.pow(23.8033423361333, 2)));
hasilallno8=hasil1*hasil2*hasil3*hasil4*hasil5*hasil6*hasil7*hasil8*hasil9*hasil
10*0.847;

```

```

hasil1=(1/(Math.sqrt(2*Math.PI)*6.92877561475037))*Math.exp(-
1*Math.pow((blank-
4.87644787644787),2)/(2*Math.pow(6.92877561475037, 2)));
hasil2=(1/(Math.sqrt(2*Math.PI)*0.689088974210513))*Math.exp(-
1*Math.pow((code-
0.158301158301158),2)/(2*Math.pow(0.689088974210513, 2)));

```

```

hasil3=(1/(Math.sqrt(2*Math.PI)*6.20938269958631))*Math.exp(-
1*Math.pow((comment-
2.77220077220077),2)/(2*Math.pow(6.20938269958631, 2)));
hasil4=(1/(Math.sqrt(2*Math.PI)*26647.6963119445))*Math.exp(-
1*Math.pow((effort-
15643.7706949806),2)/(2*Math.pow(26647.6963119445, 2)));
hasil5=(1/(Math.sqrt(2*Math.PI)*0.250694309879136))*Math.exp(-
1*Math.pow((est-
0.219806949806949),2)/(2*Math.pow(0.250694309879136, 2)));
hasil6=(1/(Math.sqrt(2*Math.PI)*117.117161165245))*Math.exp(-
1*Math.pow((length-
118.127413127413),2)/(2*Math.pow(117.117161165245, 2)));

```



```

hasil7=(1/(Math.sqrt(2*Math.PI)*0.15302540936761))*Math.exp(-
1*Math.pow((level-
0.137104247104246),2)/(2*Math.pow(0.15302540936761, 2)));
hasil8=(1/(Math.sqrt(2*Math.PI)*1480.42747168747))*Math.exp(-
1*Math.pow((time-
869.097992277992),2)/(2*Math.pow(1480.42747168747, 2)));
hasil9=(1/(Math.sqrt(2*Math.PI)*6.42841612188894))*Math.exp(-
1*Math.pow((operators-
12.8108108108108),2)/(2*Math.pow(6.42841612188894, 2)));
hasil10=(1/(Math.sqrt(2*Math.PI)*44.7716720583224))*Math.exp(-
1*Math.pow((total-
45.2432432432432),2)/(2*Math.pow(44.7716720583224,
2)));
hasilallyes8=hasil1*hasil2*hasil3*hasil4*hasil5*hasil6*hasil7*hasil8*hasil9*hasi
110*0.153; if (hasilallno8>hasilallyes8)
{
hasil_t=hasil_t+1;
}
else if (hasilallno8<hasilallyes8)
{
hasil_y=hasil_y+1;
}

//model9
hasil1=(1/(Math.sqrt(2*Math.PI)*2.69506492690203))*Math.exp(-
1*Math.pow((blank-
1.21548117154811),2)/(2*Math.pow(2.69506492690203, 2)));
hasil2=(1/(Math.sqrt(2*Math.PI)*0.744063670717665))*Math.exp(-
1*Math.pow((code-
0.131101813110181),2)/(2*Math.pow(0.744063670717665, 2)));
hasil3=(1/(Math.sqrt(2*Math.PI)*2.05962447621718))*Math.exp(-
1*Math.pow((comment-
0.648535564853556),2)/(2*Math.pow(2.05962447621718, 2)));
hasil4=(1/(Math.sqrt(2*Math.PI)*14368.1970778443))*Math.exp(-
1*Math.pow((effort-
3360.38723152022),2)/(2*Math.pow(14368.1970778443, 2)));
hasil5=(1/(Math.sqrt(2*Math.PI)*0.142450036277719))*Math.exp(-
1*Math.pow((est-
0.0619665271966529),2)/(2*Math.pow(0.142450036277719, 2)));
hasil6=(1/(Math.sqrt(2*Math.PI)*69.7473722578665))*Math.exp(-
1*Math.pow((length-
37.9135285913528),2)/(2*Math.pow(69.7473722578665, 2)));
hasil7=(1/(Math.sqrt(2*Math.PI)*0.324856650013185))*Math.exp(-
1*Math.pow((level-
0.353905160390516),2)/(2*Math.pow(0.324856650013185, 2)));
hasil8=(1/(Math.sqrt(2*Math.PI)*798.233067085152))*Math.exp(-
1*Math.pow((time-
186.688640167364),2)/(2*Math.pow(798.233067085152, 2)));

```

```

hasil9=(1/(Math.sqrt(2*Math.PI)*4.93757245093874))*Math.exp(-
1*Math.pow((operators-
6.75941422594142),2)/(2*Math.pow(4.93757245093874, 2)));
hasil10=(1/(Math.sqrt(2*Math.PI)*23.9938667045447))*Math.exp(-
1*Math.pow((total-
16.0516039051603),2)/(2*Math.pow(23.9938667045447, 2)));
hasilallno9=hasil1*hasil2*hasil3*hasil4*hasil5*hasil6*hasil7*hasil8*hasil9*hasil
10*0.845;

```

```

hasil1=(1/(Math.sqrt(2*Math.PI)*6.7182043638834))*Math.exp(-
1*Math.pow((blank-
4.71863117870722),2)/(2*Math.pow(6.7182043638834, 2)));
hasil2=(1/(Math.sqrt(2*Math.PI)*0.655305725816485))*Math.exp(-
1*Math.pow((code-
0.144486692015209),2)/(2*Math.pow(0.655305725816485, 2)));
hasil3=(1/(Math.sqrt(2*Math.PI)*5.46029222338651))*Math.exp(-
1*Math.pow((comment-
2.46768060836501),2)/(2*Math.pow(5.46029222338651, 2)));
hasil4=(1/(Math.sqrt(2*Math.PI)*26428.7384358408))*Math.exp(-
1*Math.pow((effort-
hasil5=(1/(Math.sqrt(2*Math.PI)*0.244774014259036))*Math.exp(-
1*Math.pow((est-
0.216539923954372),2)/(2*Math.pow(0.244774014259036, 2)));
hasil6=(1/(Math.sqrt(2*Math.PI)*114.741165951224))*Math.exp(-
1*Math.pow((length-
116.615969581749),2)/(2*Math.pow(114.741165951224, 2)));
hasil7=(1/(Math.sqrt(2*Math.PI)*0.162271467427729))*Math.exp(-
1*Math.pow((level-
0.141520912547528),2)/(2*Math.pow(0.162271467427729, 2)));
hasil8=(1/(Math.sqrt(2*Math.PI)*1468.26317385962))*Math.exp(-
1*Math.pow((time-
849.560494296577),2)/(2*Math.pow(1468.26317385962, 2)));
hasil9=(1/(Math.sqrt(2*Math.PI)*6.57258766835194))*Math.exp(-
1*Math.pow((operators-
12.6463878326996),2)/(2*Math.pow(6.57258766835194, 2)));
hasil10=(1/(Math.sqrt(2*Math.PI)*42.6654484664737))*Math.exp(-
1*Math.pow((total-
44.2205323193916),2)/(2*Math.pow(42.6654484664737,
2)));
hasilalloyes9=hasil1*hasil2*hasil3*hasil4*hasil5*hasil6*hasil7*hasil8*hasil9*hasi
110*0.155; if (hasilallno9>hasilalloyes9)
{
hasil_t=hasil_t+1;
}
else if (hasilallno9<hasilalloyes9)
{
hasil_y=hasil_y+1;
}

```



```
//model10
hasil1=(1/(Math.sqrt(2*Math.PI)*2.75556866758956))*Math.exp(-
1*Math.pow((blank-
1.19529085872576),2)/(2*Math.pow(2.75556866758956, 2)));
hasil2=(1/(Math.sqrt(2*Math.PI)*0.746749434286372))*Math.exp(-
1*Math.pow((code-
0.135041551246537),2)/(2*Math.pow(0.746749434286372, 2)));
hasil3=(1/(Math.sqrt(2*Math.PI)*2.00904267719062))*Math.exp(-
1*Math.pow((comment-
0.625346260387811),2)/(2*Math.pow(2.00904267719062, 2)));
hasil4=(1/(Math.sqrt(2*Math.PI)*14287.5295225557))*Math.exp(-
1*Math.pow((effort-
3274.39761080332),2)/(2*Math.pow(14287.5295225557, 2)));
hasil5=(1/(Math.sqrt(2*Math.PI)*0.141781435780191))*Math.exp(-
1*Math.pow((est-
0.0604155124653741),2)/(2*Math.pow(0.141781435780191, 2)));
hasil6=(1/(Math.sqrt(2*Math.PI)*69.3327511645765))*Math.exp(-
1*Math.pow((length-
36.9896121883656),2)/(2*Math.pow(69.3327511645765, 2)));
hasil7=(1/(Math.sqrt(2*Math.PI)*0.325651350714546))*Math.exp(-
1*Math.pow((level-
0.352548476454294),2)/(2*Math.pow(0.325651350714546, 2)));
hasil8=(1/(Math.sqrt(2*Math.PI)*793.751539116347))*Math.exp(-
1*Math.pow((time-
181.911433518006),2)/(2*Math.pow(793.751539116347, 2)));
hasil9=(1/(Math.sqrt(2*Math.PI)*4.92503231429013))*Math.exp(-
1*Math.pow((operators-
6.64335180055401),2)/(2*Math.pow(4.92503231429013, 2)));
hasil10=(1/(Math.sqrt(2*Math.PI)*23.8633992149632))*Math.exp(-
1*Math.pow((total-
15.7195290858725),2)/(2*Math.pow(23.8633992149632, 2)));
hasilallno10=hasil1*hasil2*hasil3*hasil4*hasil5*hasil6*hasil7*hasil8*hasil9*hasi
l10*0.851;
```

```
hasil1=(1/(Math.sqrt(2*Math.PI)*6.87790228708995))*Math.exp(-
1*Math.pow((blank-
4.82213438735177),2)/(2*Math.pow(6.87790228708995, 2)));
hasil2=(1/(Math.sqrt(2*Math.PI)*0.663317391396526))*Math.exp(-
1*Math.pow((code-
0.142292490118577),2)/(2*Math.pow(0.663317391396526, 2)));
hasil3=(1/(Math.sqrt(2*Math.PI)*6.48835990287599))*Math.exp(-
1*Math.pow((comment-
2.8893280632411),2)/(2*Math.pow(6.48835990287599, 2)));
hasil4=(1/(Math.sqrt(2*Math.PI)*26063.7590890116))*Math.exp(-
1*Math.pow((effort-
15866.1787351778),2)/(2*Math.pow(26063.7590890116, 2)));
hasil5=(1/(Math.sqrt(2*Math.PI)*0.248281602849102))*Math.exp(-
1*Math.pow((est-
```



```

0.22391304347826),2)/(2*Math.pow(0.248281602849102, 2)));
hasil6=(1/(Math.sqrt(2*Math.PI)*115.992400060623))*Math.exp(-
1*Math.pow((length-
hasil7=(1/(Math.sqrt(2*Math.PI)*0.14456269199879))*Math.exp(-
1*Math.pow((level-
0.134268774703557),2)/(2*Math.pow(0.14456269199879, 2)));
hasil8=(1/(Math.sqrt(2*Math.PI)*1447.98657338833))*Math.exp(-
1*Math.pow((time-
881.453952569169),2)/(2*Math.pow(1447.98657338833, 2)));
hasil9=(1/(Math.sqrt(2*Math.PI)*6.54406847249157))*Math.exp(-
1*Math.pow((operators-
12.9328063241106),2)/(2*Math.pow(6.54406847249157, 2)));
hasil10=(1/(Math.sqrt(2*Math.PI)*44.6262683527402))*Math.exp(-
1*Math.pow((total-
45.4664031620553),2)/(2*Math.pow(44.6262683527402, 2)));

hasilallyes10=hasil1*hasil2*hasil3*hasil4*hasil5*hasil6*hasil7*hasil8*hasil9*ha
sil10*0.149;
if (hasilallno10>hasilallyes10)
{
hasil_t=hasil_t+1;
}
else if (hasilallno10<hasilallyes10)
{
hasil_y=hasil_y+1;
}

if (hasil_y>hasil_t)
{
jTextArea1.setText("Class Yes ");
}
else if (hasil_y<hasil_t)
{
jTextArea1.setText("Class No ");
}
else
{
jTextArea1.setText("Undefined Class");
}

```

DAFTAR RIWAYAT HIDUP



Personal Information

Name	Rakhmat Purnomo, S.Pd, S.Kom, M.Kom
NIK	32750622108201
Place, Date of Birth	Jakarta, October 22 th 1982
Gender	Male
Religion	Moslem
Nationality	Indonesia (WNI)
Hobby	IT Innovation, Camping, and Traveling
Address	Jl. Pandawa No. 18 Rt. 02 Rw. 06 Kel. Harapan Mulya Kec. Medan Satria Kota Bekasi 17143
Phone	0812 9947 4407
e-mail	Rakhmat313@gmail.com

Education Background

2013 – 2015	STMIK Nusa Mandiri, Jakarta Master in Computer Sciences (M.Kom) Research Interest : Software Engineering, Intelligent Systems GPA = 3,58 / 4
2007 – 2012	STMIK Mitra Karya, Bekasi Bachelor of Degree (Information Technology) GPA = 3,28 / 4
2000 – 2006	Universitas Negeri Jakarta Bachelor of Degree (Electronics Engineering Education) GPA = 3,14 / 4

Demikian daftar riwayat hidup ini dibuat dengan sebenarnya.

Jakarta, 1 Desember 2015

Rakhmat Purnomo