SELEKSI FITUR MENGGUNAKAN ALGORITMA *CHI SQUARE* UNTUK PREDIKSI CACAT PERANGKAT LUNAK

SKRIPSI

Oleh:
MUHAMMAD FARIS RURI RAMADLANI
NIM. 14650052

JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2019

SELEKSI FITUR MENGGUNAKAN ALGORITMA *CHI SQUARE* UNTUK PREDIKSI CACAT PERANGKAT LUNAK

SKRIPSI

Diajukan kepada:
Fakultas Sains dan Teknologi
Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan Dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh:
MUHAMMAD FARIS RURI RAMADLANI
NIM. 14650052

JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2019

LEMBAR PERSETUJUAN

SELEKSI FITUR MENGGUNAKAN ALGORITMA *CHI SQUARE* UNTUK PREDIKSI CACAT PERANGKAT LUNAK

SKRIPSI

Oleh :
MUHAMMAD FARIS RURI RAMADLANI
NIM. 14650052

Telah Diperiksa dan Disetujui untuk Diuji

Tanggal : Juni 2019

Dosen Pembimbing I	Dosen Pembimbing II
Fatchurrochman, M.Kom	Irwan Budi Santoso, M.Kom
NIP. 19700731 200501 1 002	NIP. 19770103 201101 1 004

Mengetahui,
Ketua Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malan

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

HALAMAN PENGESAHAN

SELEKSI FITUR MENGGUNAKAN ALGORITMA *CHI SQUARE* UNTUK PREDIKSI CACAT PERANGKAT LUNAK

SKRIPSI

Oleh:
MUHAMMAD FARIS RURI RAMADLANI
NIM. 14650052

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: Juni 2019

Susunan Dewan Penguji			Tanda Tangan
Penguji Utama	:	Ajib Hanani, M.T NIDT. 19840731 20160801 1 076	( )
Ketua Penguji	:	A'la Syauqi, M.Kom NIP. 19771201 200801 1 007	( )
Sekretaris Penguji	:	Fatchurrochman, M.Kom NIP. 19700731 200501 1 002	( )
Anggota Penguji	:	Irwan Budi Santoso, M.Kom NIP. 19770103 201101 1 004	( )

Mengetahui,
Ketua Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

HALAMAN PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan dibawah ini,

Nama : Muhammad Faris Ruri Ramadlani

NIM : 14650052

Jurusan : Teknik Informatika

Fakultas : Sains dan Teknologi

Menyatakan dengan sebenarnya benarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambilan tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, Juni 2019
Yang membuat pernyataan,

M. Faris Ruri Ramadlani
NIM. 14650052

MOTTO

"Urip Iku Urup"

HALAMAN PERSEMBAHAN

Bismillahirrahmanirrahim

Alhamdulillahi Rabbil 'Alamin Segala puji bagi Allah atas nikmat-Nya serta dukungan dan doa yang tiada henti dari orang-orang tercinta sehingga skripsi ini dapat terselesaikan . Oleh karena itu kupersembahkan karya tulis yang masih jauh dari kata sempurna ini kepada:

Ayah ,ibu dan istri tersayang yang selalu memberikan semangat, meluangkan waktu dan doa yang tiada henti untuk saya.

Bapak Fatchurrochman, M.kom dan Bapak Irwan Budi Santoso, M.Kom selaku dosen pembimbing yang selama ini dengan sabar dan ikhlas memberikan bimbingan kepada saya, dan untuk seluruh dosen Teknik Informatika yang selama ini telah memberikan saya banyak ilmu juga pengalaman yang sangat berguna untuk saya. Semoga keberkahan selalu terlimpah untuk beliau-beliau semua.

Saya ucapkan rasa terimakasih dengan teramat dalam untuk teman-teman Tenik Informatika, khususnya warga Biner (TI angkatan 2014) yang sudah memberi pengalaman berharga selama ini. Semoga kita bisa dipertemukan lagi di lain kesempatan dengan kesuksesan diantara kalian.

Tak lupa juga kepada seluruh teman-temanku yang tak bisa disebutkan satu-satu yang juga berjasa dan membuat saya bisa sampai seperti ini.

Terimakasih Banyak.

KATA PENGANTAR

Assalamualaikum Wr. Wb.

Alhamdulillah segala puji bagi Allah tuhan semesta alam, dengan segala rahmat dan kasih saying-Nya sehingga penulis mampu menyelesaikan naskah skripsi ini yang semoga bias bermanfaat untuk orang lain di kemudian hari. Tidak lupa juga shalawat serta salam penulis ucapkan untuk junjungan kita, suri tauladan kita yakni Rasulullah Muhammad SAW yang telah menuntun umatnya dari zaman kegelapan menuju zaman terang dengan agama Islam yang mulia.

Selanjutnya penulis haturkan ucapan terima kasih karena dalam penyelesaian skripsi ini tidak lepas dari bantuan, bimbingan serta dukungan dari beberapa pihak. Ucapan terima kasih ini penulis sampaikan kepada:

1. Prof. DR. H. Abd. Haris, M.Ag, selaku rektor UIN Maulana Malik Ibrahim Malang beserta seluruh staf. Bakti Bapak dan Ibu sekalian terhadap UIN Maliki Malang yang menaungi segala kegiatan di kampus UIN Maliki Malang.
2. Dr. Sri Harini, M.Si selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang beserta seluruh staf. Bapak dan ibu sekalian sangat berjasa memupuk dan menumbuhkan semangat untuk maju kepada penulis.
3. Bapak Dr. Cahyo Crysdian, selaku Ketua Jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang, yang telah banyak memberi masukan dan motivasi untuk tidak mudah menyerah.
4. Bapak Fatchurrochman, M.Kom selaku dosen pembimbing I yang telah bersedia meluangkan waktu untuk membimbing, memberi arahan dan masukan kepada penulis dalam pengerjaan skripsi ini hingga akhir.
5. Bapak Irwan Budi Santoso, M.T selaku dosen pembimbing II yang juga senantiasa memberi masukan dan nasihat serta petunjuk dalam penyusunan skripsi ini.

6. Bapak Totok Chamidi, M.Kom selaku dosen wali yang telah dan selalu memberi arahan dan bimbingan hingga saat ini.
7. Ayah, Ibu, Istri tercinta serta keluarga besar yang selalu memberi dukungan doa maupun semuanya demi kelancaran pengerjaan skripsi ini sampai selesai.
8. Seluruh Dosen Teknik Informatika yang telah memberikan keilmuan serta pengalaman yang berarti kepada penulis selama ini.
9. Seluruh teman-teman Teknik Informatika UIN Maulana Malik Ibrahim Malang yang telah banyak berbagi ilmu, pengalaman dan menjadi inspirasi untuk terus semangat belajar.
10. Teman-teman seperjuangan angkatan 2014 khususnya teman-teman TI-B dan Biner Teknik Informatikayang telah berjuang bersama dan saling mendukung selama ini.
11. Para peneliti yang telah melakukan penelitian tentang prediksi cacat perangkat lunak dan seleksi fitur yang menjadi acuan penulis dalam pembuatan skripsi ini. Serta semua pihak yang telah membantu yang tidak bisa disebutkan satu persatu. Terimakasih banyak.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna dikarenakan terbatasnya pengalaman dan pengetahuan yang dimiliki penulis. Oleh karena itu, penulis sangat mengharapkan saran, masukan serta kritik yang membangun dari berbagai pihak. Dan harapan penulis, semoga skripsi ini dapat memberikan manfaat bagi pembaca dan mendorong peneliti selanjutnya agar lebih baik dalam lagi kedepannya. Amin.

Wassalamualaikum Wr. Wb.

Malang, Mei 2019

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAK

Ramadlani, M.F.R. 2018. **Seleksi Fitur Menggunakan Algoritma Chi Square Untuk Prediksi Cacat Perangkat Lunak.** Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Fatchurrochman, M.Kom. (II) Irwan Budi Santoso, M.Kom.

Kata kunci: prediksi cacat perangkat lunak, seleksi fitur *Chi Square.*

Cacat perangkat lunak atau biasa yang dikenal dengan istilah bug atau kesalahan di dalam aplikasi yang sudah dibuat oleh seorang pengembang atau programmer. Atau sesuatu yang muncul ketika hasil yang diharapkan dari sebuah perangkat lunak bertolak belakang hasil yang sebenarnya. Namun , bisa juga saat pengujian menemukan sesuatu dalam sistem yang menyimpang dari perilaku yang diharapkan , bukan berarti bisa dikatkan sepenuhnya bahwa hal tersebut merupakan bug. Data yang digunakan adalah *dataset* dari NASA, yang berjumlah 5 data, yaitu CM1, JM1, KC1, KC2, PC1.

Penelitian ini melakukan pemilihan fitur menggunakan seleksi fitur dengan algoritma *Chi Square*. Uji coba dilakukan dengan menyeleksi fitur fitur yang ada kemudian menghitung akurasi dengan metode klasifikasi *Naïve Bayes*. Diambilnya berbagai jumlah yang berbeda pada setiap data, akan diketahui nantinya jumlah fitur dan fitur apa saja yang memiliki pengaruh signifikan pada penelitian ini.

Hasil dari penelitian ini menyimpulkan bahwa fitur-fitur yang memiliki pengaruh signifikan terhadap prediksi cacat perangkat lunak berdasarkan seleksi fitur *Chi Square* pada *dataset* CM1 adalah fitur *i* (*Intelligence*), *t* (*Time to write program*) dengan akurasi **93,47%**. *Dataset* JM1 adalah fitur *i (Intelligence), l (Program length), total_Opnd (Total operand), d (Difficulty), v (Volume), b (Error estimate), loc (Line of code), v(g) (Cyclomatic complexity), ev(g) (Essential complexity), n (Total operator + operand), total_Op (Total operator), branchCount, uniq_Opnd (Unique operands), lOCode (Count of Statement Lines), t (Time to write program), e (Effort to write program), uniq_Op (Unique operators), iv(g) (Design complexity), lOComment (Count of lines of comments)* dengan akurasi **94,10** %. Pada *dataset* KC1 adalah fitur *d* (*Difficulty)* dengan akurasi **92,06%**. Pada *dataset* KC2 adalah fitur *d* (*Difficulty),* dengan akurasi **96,65%**. Pada *dataset* PC1 adalah fitur lOComment (*Count of lines of comments),v (Volume), branchCount, v(g) (Cyclomatic complexity),lOCode lOCode (Count of Statement Lines), t (Time to write program), b (Error estimate), e (Effort to write program)* dengan akurasi **93,02%.**

ABSTRACT

Ramadlani, M.F.R. 2018. **Feature Selection Using Chi Square Algorithms For Software Defect Prediction**. Undergraduate Thesis. Department of Informatics Engineering Faculty of Science and Technology Islamic State Maulana Malik Ibrahim University, Malang. Supervisor: (I) Fatchurrochman, M. Kom. (II), Irwan Budi Santoso, M. Kom.

Keyword : software defect prediction, chi square feature selection.

Software defect or regularly known as bug or errors in aplications already created by a developer or programmer. Or Something that appears when the expectation from the software is not like with actual result. However, it can also when the testing finds something else from expectation, it's not mean can be fully claimed as a bug. The data used is a dataset from NASA MDP, which amounts to 5 data, are CM1, JM1, KC1, KC2, and PC1.

This study performs feature selectin using Chi Square algorithm. The testing is done by selecting the features and then calculating the accuracy of prediction using Naive Bayes classification. with the take of different amounts on each data, it will be known later the number of features and features that have significant influence on this study.

The result of this study concluded the features that have a significant influence on the software defect prediction based on Chi Square feature selection on the CM1 dataset are features I (Intelligence), T (Time to write program) with accuracy 93.47%. Dataset JM1 are features I (Intelligence), L (Program length), total_Opnd (Total operand), D (Difficulty), V (Volume), B (Error estimate), loc (Line of Code), V (g) (Cyclomatic complexity), Ev (g) (Essential complexity), N (Total operator + operand), total_Op (Total operator), branchCount, uniq_Opnd (Unique operands), lOCode (Count of Statement Lines), T (Time to write program), E (Effort to write program), uniq_Op (Unique operators), iv(g) (Design complexity), lOComment (Count of Lines of comments) with accuracy 94.10%. In the KC1 dataset is the feature D (Difficulty) with accuracy 92.06%. In the KC2 Dataset is the feature D (Difficulty), with accuracy 96.65%. In the PC1 dataset are features lOComment (Count of lines of comments), V (Volume), branchCount, V (g) (Cyclomatic complexity), lOCode lOCode (Count of Statement Lines), T (Time to write program), B (Error estimate), E (Effort to write program) with accuracy 93.02%.

ملخص

رمضان, م.ف.ر. 2019. **اختيار الميزة باستخدام تشي سكوير في التنبؤ من عيوب البرمجيات، البحث العلمي.** قسم المعلوماتية كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف : (1) فتح الرحمن الماجستير، (2) إروان بودي سانتوسا الماجستير.

الكلمات الرئيسية : التنبؤ عيب البرمجيات ، تشي سكوير ميزه الاختيار.

البرامج المعيبة أو الأخطاء في التطبيق الذي تم إنشاؤه بالفعل من قبل مطور. أو اي شيء ينشا عندما تكون النتيجة المتوقعة أو المرغوبة للبرمجيات مخالفه للنتائج الفعلية. ومع ذلك ، فانه يمكن أيضا ان يكون عند اختبار يجد شيئا في النظام الذي ينحرف عن السلوك المتوقع ، فانه لا يعني انه يمكن تعزيزها بشكل كامل انها أعاقه. البيانات المستخدمة هي مجموعه بيانات من ناسا ، والتي تصل إلى ه معطيات ، وهي *CM1*، *JM1*، *KC1*، *KC2*، *PC1*.

هذا البحث ينفذ ميزة التحديد باستخدام اختيار ميزة مع خوارزمية ساحة تشي. يتم الاختبار عن طريق تحديد ميزات الميزة ثم حساب الدقة مع طريقة تصنيف السذاجة بايز. إذا أخذنا كمية مختلفة على كل بيانات، فإنه سوف يعرف في وقت لاحق عدد من الميزات والميزات التي لها تأثير كبير على هذا البحث.

وخلصت نتائج هذه الدراسة إلى أن الميزات التي لها تأثير كبير على التنبؤ بعيوب البرمجيات استنادإلى اختيار ميزات تشي سكوير على مجموعة البيانات *CM1* هي ميزة (*Intelligence*) *i*، (Time to write program) *t*. مجموعات البيانات *JM1* هي الميزات *i (Intelligence), l (Program length), total_Opnd (Total operand), d (Difficulty), v (Volume), b (Error estimate), loc (Line of code), v(g) (Cyclomatic complexity), ev(g) (Essential complexity), n (Total operator + operand), total_Op (Total operator), branchCount, uniq_Opnd (Unique operands), lOCode (Count of Statement Lines), t (Time to write program), e (Effort to write program), uniq_Op (Unique operators), iv(g) (Design complexity), lOComment (Count of lines of comments)*. في *KC1* هو ميزة (*Difficulty*) *d*. في *KC2*هو ميزة (*Difficulty*) *d* . في مجموعة البيانات *PC1* هي ميزات *lOComment* (*Count of lines of comments),v (Volume), branchCount, v(g) (Cyclomatic complexity),lOCode lOCode (Count of Statement Lines), t (Time to write program), b (Error estimate), e (Effort to write program)*

BAB I
PENDAHULUAN

1.1. Latar Belakang Masalah

Perangkat lunak memiliki berbagai fungsi dan peranan yang penting untuk mempermudah aktivitas manusia. Dalam masa yang serba modern seperti ini, peran perangkat lunak yang optimal sangat diperlukan, yakni perangkat lunak yang minim akan *bug* saat digunakan.

Software Development Life Cycle (SDLC) mawakili berbagai tahapan dalam produk perangkat lunak. SDLC mencakup perencanaan, analisis, desain, pengembangan, dan fase uji, baik sebagai model linier (*water fall*) atau siklikal (*incremental, iterative, agile*). Pelayo dan Dick (2007) menyebutkan bahwa perbaikan yag dilakukan saat perangkat lunak sudah dipakai atau digunakan untuk beroperasi dan di terynyata ditemukan kecacatan di dalamnya, maka akan memakan biaya yang tinggi, sedangkan jika perangkat lunak masih dalam proses pengembangan dan ditemukan cacat di dalamnya, biaya yang dikeluarkan lebih sedikit.

Dengan menggunakan metode mesin pembelajaran, para peneliti telah berhasil menerapkan dengan berbagai metode yang berbeda dengan tema prediksi cacat perangkat lunak. Arar (2017) menyebutkan bahwa dalam proses prediksi cacat, tidak semua fitur yang terdapat dalam *dataset* akan digunakan, melainkan akan memilih diantara fitur-fitur tersebut. Pemilihan tersebut biasa didapat melalui proses *feature selection* yang mana akan diambil beberapa fitur yang berpengaruh selama proses prediksi.

Beberapa penelitian terdahulu tentang prediksi cacat perangkat lunak-terkait dengan penerapan pemilihan fitur, oleh Karabulut dkk. (2012) penelitiannya yang menggunakan 15 yang diambil dari *Data Mining Repository of University of California Irvine* (UCI), dimana penelitian dilakukan dengan menggunakan 6 fitur seleksi dan 3 pengklasifikasi. Hasilnya, pada klasifikasi dengan *Naive Bayes* , fitur seleksi yang memberi nilai efektif dengan menggunakan *Gain Ratio*, pada *Multilayer Perceptron* (MLP) dengan *Chi Square* , dan pada *Decision Tree* dengan menggunakan *Information Gain*.

Di dalam islam di jelaskan bahwa manusia pasti memiliki kekurangan, begitu pula dengan perangkat lunak yang dibuat oleh manusia, seperti yang dijelaskan pada surah Ar Rum ayat 54 dibawah ini.

اللَّهُ الَّذِي خَلَقَكُمْ مِنْ ضَعْفٍ ثُمَّ جَعَلَ مِنْ بَعْدِ ضَعْفٍ قُوَّةً ثُمَّ جَعَلَ مِنْ بَعْدِ قُوَّةٍ ضَعْفًا وَشَيْبَةً ۚ يَخْلُقُ مَا يَشَاءُ ۖ وَهُوَ الْعَلِيمُ الْقَدِيرُ

Artinya : *"Allah, Dialah yang menciptakan kamu dari keadaan lemah, kemudian Dia menjadikan (kamu) sesudah keadaan lemah itu menjadi kuat, kemudian Dia menjadikan (kamu) sesudah kuat itu lemah (kembali) dan beruban. Dia menciptakan apa yang dikehendaki-Nya dan Dialah Yang Maha Mengetahui lagi Maha Kuasa " (Q.S Ar Rum : 54)*

Menurut tafsir Ibnu Katsir, dalam ayat ini Allah SWT mengingatkan manusia akan fase-fase yang telah dilaluinya dalam penciptaannya, dari suatu keadaan menuju keadaan yang lain. Asal mulanya manusia itu berasal dari tanah liat, kemudian dari air mani, kemudian menjadi 'alaqah, kemudian menjadi segumpal daging, kemudian menjadi tulang yang dilapisi dengan daging, lalu ditiupkan roh ke dalam tubuhnya.

Setelah itu ia dilahirkan dari perut ibunya dalam keadaan lemah, kecil, dan tidak berkekuatan. Kemudian menjadi besar sedikit demi sedikit hingga menjadi anak, setelah itu berusia balig dan masa puber, lalu menjadi pemuda. Inilah yang dimaksud dengan keadaan kuat sesudah lemah.

Kemudian mulailah berkurang dan menua, lalu menjadi manusia yang lanjut usia dan memasuki usia pikun; dan inilah yang dimaksud keadaan lemah sesudah kuat. Di fase ini seseorang mulai lemah keinginannya, gerak, dan kekuatannya; rambutnya putih beruban, sifat-sifat lahiriah dan batinnya berubah pula.

Dan terdapat pada sebuah hadits yang menunjukkan bahwa setiap manusia pasti pernah melakukan perbuatan dosa, berikut haditsnya;

يا عبادي إنكم تخطئون في الليل والنهار وأنا أغفر الذنوب جميعاً فاستغفروني أغفر لكم (صحيح مسلم 4674)

Artinya:*"Wahai hamba-hamba Ku, sesungguhnya kalian berbuat salah pada malam dan siang, dan Aku mengampuni semua dosa, maka minta mapunlahkepada-Ku niscaya Aku akan mengampuni kalian."* (Shahih Muslim : 4674)

Menurut hadits shahih Muslim diatas, dapat kita ketahui bahwa manusia tidak luput dari kesalahan disetiap harinya, baik kesalahan yang di sengaja maupun tidak di sengaja. Dosa bisa dilakukan antara sesama makhluknya maupun kepada sang pencipta. Mulai bangun tidur sampai mejelang tidur kembali, sangat penting untuk selalu meminta ampunan kepad Allah. Untuk itu, dengan memohon ampunan kepada Allah lah manusia bisa menghapus dosa-dosa yang telah dilakukannya.

Selain itu dijelaskan juga bahwa sebaik-baik manusia adalah manusia yang bisa memberi manfaat bagi manusia yang lain. Seperti pada potongan surah Al Isra' ayat 7 dibawah ini :

إِنْ أَحْسَنْتُمْ أَحْسَنْتُمْ لِأَنْفُسِكُمْ...

Artinya : *"Jika kalian berbuat baik, sesungguhnya kalian berbuat baik bagi diri kalian sendiri"* (QS. Al-Isra:7)

Sesuai seperti hadits dibawah ini :

خَيْرُ النَّاسِ أَنْفَعُهُمْ لِلنَّاسِ

Artinya : *"Sebaik-baik manusia adalah yang paling bermanfaat bagi manusia".* (Hadits dihasankan oleh al-Albani di dalam Shahihul Jami' , no. 3289).

Berdasarkan dalil-dalil yang sudah dibahas diatas, dapat di simpulkan bahwa tidak ada sesuatu yang sempurna di dunia ini, kecuali atas izin-Nya. Manusia sebagai makhluk pasti tidak luput dari salah dan dosa, begitu juga dengan sesuatu yang dikerjakan nya, pasti juga akan ada kekurangan-kerungan didalamnya, seperti sebuah program bagi seorang *progammer*. Oleh karena itu, kita sebagai manusia bias memberi manfaat kepada yang lain dan bahwa pentingnya pemilihan fitur dimana fitur-fitur yang akan dipilih atau diseleksi merupakan fitur yang memiliki peran penting untuk proses prediksi cacat prangkat lunak.

Pemilihan fitur tersebut menggunakan algortima *Chi Square* dan pengujian klasifikasi akan menggunakan algorima *Naive Bayes*. Dengan demikian,penelitian kali ini akan menggunakan fitur seleksi *Chi Square* dengan pengujian menggunakan algoritma *Naïve Bayes* pada *dataset Software Defect Prediction*.

1.2. Identifikasi Masalah

Berdasarkan uraian pada latar belakang masalah diatas manakah fitur-fitur yang memiliki pengaruh signifikan terhadap proses prediksi cacat perangkat lunak menggunakan *Chi Square* dengan metode klasifikasi *Naïve Bayes*?

1.3. Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk mengetahui fitur yang memiliki pengaruh signifikan terhadap prediksi cacat perangkat lunak menggunakan algoritma seleksi fitur *Chi Square*.

1.4. Manfaat Penelitian

Adapun manfaat yang diharapkan dari penelitian ini adalah meningkatkan hasil akurasi menggunakan pemilihan fitur *Chi Square* dengan klasifikasi *Naïve Bayes*.

1.5. Batasan Masalah

Berdasarkan masalah-masalah yang sudah dijelaskan pada sub bagian sebelumnya, pengembangan sistem dapat dibatasi dengan batasan data yang digunakan berupa *dataset metrics data program* yang diunduh dari *website* PROMISE (*Predictor Models in Software Engineering*) *repository* melalui http://promise.site.uottawa.ca/SERepository (yang_di_unduh_pada_tanggal 12 Februari 2018) yang terdiri dari 5 *dataset* yaitu *dataset* CM1, JM1, KC1, KC2 dan PC1.

1.6. Sistematika Penulisan

Dalam penulisan skripsi ini, secara keseluruhan terdiri dari lima bab yang masing-masing bab disusun dengan sistematika sebagai berikut :

BAB I PENDAHULUAN

Bab ini merupakan bagian awal, dalam bab ini berisi latar belakang, pernyataan masalah, tujuan penelitian, batasan masalah, manfaat penelitiaandan sistematika penulisan laporan.

BAB II TINJAUAN PUSTAKA

Bab ini berisi tentang teori-teori yang berhubungan dengan permasalahan yang diangkat dari penelitian ini, antara lain penelitian-penelitian terkait, cacat perangkat lunak, prediksi cacat perangkat lunak, algoritma diskritisasi data *Equal Width Binning,* algoritma seleksi fitur *Chi Square* , algoritma klasifikasi *Naïve Bayes.*

BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan tentang analisis dan perancangan sistem yang akan dibuat sekaligus batasan-batasan sistem serta di dalamnya juga terdapat beberapa hitungan manual algoritma-algoritma yang dipakai.

BAB IV UJI COBA DAN PEMBAHASAN

Bab ini berisi mengenai pengujian dan analisis dari hasil pengujian dari sistem yang telah dibangun berdasarkan hasil perancangan pada bab 3 sebelumnya.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dan saran seluruh penelitian yang telah dilakukan.

BAB II
TINJAUAN PUSTAKA

2.1 Cacat Perangkat Lunak

Kesalahan dalam perangkat lunak atau biasa yang dikenal dengan istilah bug atau kesalahan di dalam aplikasi yang sudah dibuat oleh seorang pengembang atau programmer. Sebuah yang muncul ketika hasil yang diharapkan atau di inginkan dari sebuah perangkat lunak bertolak belakang hasil yang sebenarnya. Namun , bisa juga saat tester menemukan sesuatu dalam sistem yang menyimpang dari perilaku yang diharapkan , bukan berarti bisa dikatkan sepenuhnya bahwa hal tersebut merupakan bug.

Dalam sebuah pengujian, terkadang tester tidak melaporkan tentang cacat, tapi lebih kepada insiden yang terjadi saat seorang tester menguji produk dan tidak menemukan hasil seperti yang diinginkan. Oleh sebab itu, hal ini tidak secara langsung bisa disebut dengan sebuah kecacatan. Hal tersebut bisa saja karena data untuk pengujian yang salah, melakukan pengaturan yang salah, atau bahkan bisa karena kesalahpahaman dari seorang tester.

Runeson dkk. (2006) menjelaskan bahwa dalam cacat perangkat lunak terdiri dari cacat kebutuhan, cacat desain, dan cacat kode. Sedangan Pressman (2001) menyebutkan cacat perangkat lunakdidefinisikan sebagai cacat pada perangkat lunak yang mungkin terjadi cacat pada kode program, dokumentasi, desain, dan pada ha-hal lain yang menyebabkan kegagalan perangkat lunak. Cacat pada perangkat lunak bisa muncul di berbagai tahapan proses pengembangan perangkat lunak.

Cacat pada perangkat lunak merupakan salah satu faktor penting yang dapat mempengaruhi kualitas dari perangkat lunak. Dengan mengurangi atau bahkan bisa mencegah timbulnya cacat pada sebuah perangkat lunak dapat meningkatkan kualitas dari perangkat lunak itu sendiri (Boehm dan Basili , 2001).

2.2 Prediksi Cacat Perangkat Lunak

Prediksi pada cacat perangkat lunak oleh Okumoto (2011) diasumsikan sebagai jumlah cacat pada perangkat lunak yang dapat ditentukan pada kondisi tertentu. Jadi apabila telah ditemukan kecacatan pada perangkat lunak maka penentuan solusi akan diketahui agar tidak menyebabkan kegagalan sistem yang lebih besar lagi. Jika defect atau kecacatan tidak kunjung diimbangi dengan solusi, permasalah yang akan muncul lebih besar lagi yaitu kegagalan sistem dimana terjadinya aktifitas sistem secara keseluruhan.

Menurut Pelayo dan Dick (2007) biaya untuk memperbaiki cacat yang terdeteksi salama proses pengembangan sistem perangkat lunak jauh lebih murah dibandingkan jika perangkat lunak sudah dipakai oleh pengguna. Jika perangkat lunak sudah dalam tahapan pakai oleh pengguna dan ditemukan sebuah kecacatan atau *bug* bisa berakibat fatal jika produk tersebut sangat penting. Contohnya , sebuah pesawat luar angkasa NASA yang bernilai 125 juta dollar Amerika hilang di luar angkasa karena ada nya *bug* dalam mengkonversi data kecil dan pesawat kargo militier *airbus* A400M yang jatuh saat uji coba penerbangan yang ternyata disebabkan oleh bug pernagkat lunak untuk pengontrolan mesin.

Czibula dkk. (2014) menyebutkan bahwa prediksi cacat perangkat lunak adalah proses yang yang mengidentifikasi cacat yang terdapat pada modul atau bagian yang terdapat pembaharuan dari sistem perangkat lunak yang mungkin

terjadi sebuah kesalahan. Prediksi cacat perangkat lunak merupakan salah satu kegiatan yang menjadi fokus para peneliti dan komunitas untuk memperoleh keakuratan yang baik dalam melakukan prediksi cacat perangkat lunak.

Selama lebih dari sepuluh tahun, masalah prediksi untuk cacat perangkat lunak ini telah banyak dilakukan penelitian karena kontribusinya yang sangat besar untuk proyek dalam perangkat lunak. Studi yang dilakukan oleh Wahono (2015) dan Malhotra (2015) secara sistematis telah banyak mengulas penelitian berupa makalah, jurnal penelitian dalam bidang ini. Dalam Menzies dkk. (2007), menurutnya teknik klasifikasi penting untuk digunakan dalam perancangan prediksi cacat perangkat lunak daripada penggunaan seleksi fitur dan teknik pemrosesan data lainnya. Dalam studi yang sudah banyak dilakukan oleh para peneliti, menunjukkan bahwa adanya hubungan antara kualitas perangkat lunak dan prediksi cacat perangkat lunak.

Dalam penelitian ini, *dataset* yang digunakan berasal dari NASA MDP (*Metric Data Program*) yang dapat di unduh dari http://promise.site.uottawa.ca/SERepository/*dataset*-page.html dengan memilih *dataset* untuk Software Defect Prediction. Terdapat 5 *dataset* yang akan digunakan dalam penelitian ini, antara lain *dataset* CM1, JM1, KC1, KC2, dan PC1. Mengenai informasi atribut yang ada dalam tiap *dataset*, dapat dilihat pada tabel 2.1 dibawah ini

Tabel 2. 1 Dataset CM1

No.	Simbol	Keterangan
1.	loc	**McCabe's "*line count of code*"**
2.	v(g)	**McCabe "*Cyclomatic complexity*"**

3.	ev(g)	**McCabe "*Essential complexity*"**
4.	iv(g)	**McCabe "*Design complexity*"**
5.	UniqOp	***unique operators***
6.	UniqOpnd	***unique operands***
7.	TotalOp	**total operator**
8.	TotalOpnd	**total operand**
9.	n	**Halstead "Total operator + operand"**
10.	v	**Halstead "Volume"**
11.	L	**Halstead "*Programlength*"**
12.	d	**Halstead "*Difficulty*"**
13.	i	**Halstead "*Intelligence*"**
14.	e	**Halstead " *Effort to write program*"**
15.	b	**Halstead "*Error estimate*"**
16.	t	**Halstead "*Time to write program*"**
17.	IOCode	***Count of Statement Lines***
18.	IOComment	***Count of lines of comments***
19.	IOBlank	***Count of blank lines***
20.	lOCodeAndComment	***Count of Code and Comments Lines***
21.	branchCount	***Branch count***
22.	*defect*	**Hasil prediksi {*false,true*}**

Berdasarkan tabel 2.1 terlihat bahwa atribut yang berbentuk data matrik yang digunakan berjumlah 21 fitur yang berbeda, berikut adalah penjelasan dari setiap fitur yang digunakan (Menzies, 2007; Akbar, 2017);

1. *Line of code* (loc)

   Loc ini merupakan teknik tradisional untuk mengukur kompleksitas sebuah kode program . Matrik ini bekerja dengan cara menghitung jumlah baris kode program yang ada.

2. *Cyclomatic complexity* (v(g))

   Matrik v(g) adalah nilai matrik yang menyediakan ukuran kuantitatif dari kompleksitas logika dari sebuah program. Dengan nilai matrik ini kita dapat menentukan apakah sebuah program termasuk kedalam program yang sederhana atau kompleks berdasar logika yang ada pada program tersebut. Contohnya logika percabangan (*if)* atau perulangan *(for,while,do while)* yang diilustrasikan sebagai sebuah *graph* yang terdiri dari jumlah *edge* (e) dan jumlah *node*(n). Rumus perhitungannya adalah v(g) = e – n + 2.

3. *Essential complexity* (ev(g))

   Matrik ev(g) adalah nilai matrik kompleksitas yang dihitung dengan mengurangi nilai *cyclomatix complexity* v(g) dengan jumlah *subflowgraph* yang terdiri dari *single entry* dan *single exit*. Rumus perhitungan *Essential complexity* adalah ev(g) = v(g) – m, m adalah jumlah *subflowgraph* yang merupakan *single entry* dan *single exit.*

4. *Design complexity* (iv(g))

   Design complexity adalah *cyclomatic complexity* dari suatu modul yang mengurangi *flowgraph*. *Flowgraph* "g" dari sebuah modul dikurangi untuk menghilangkan kompleksitas yang tidak mempengaruhi keterkaitan antara modul desain. *Design complexity* dihitung dengan mengurangi modul pada *flowgraph* dengan cara menghilangkan *node decisicon* yang tidak memiliki dampak pada kontrol sebuah program.

5. *Unique operators* (Uniq Op)

   Matrik *Uniq Op* merupakan nilai matrik yang didapat dengan cara menghitung jumlah operator yang berbeda dalam sebuah kode program.

6. *Unique operands* (*Uniq Opnd*)

   Matrik *Uniq Opnd* merupakan nilai matrik yang didapat dengan cara menghitung jumlah operand yang berbeda dalam sebuah kode program.

7. Total operator (*Total Op*)

   Matrik *Total Op* merupakan nilai matrik yang didapat dengan cara menghitung jumlah operator dalam sebuah kode program.

8. *Total operand (Total Opnd*)

   Matrik *Total Opnd* merupakan nilai matrik yang didapat dengan cara menghitung jumlah operand dalam sebuah kode program.

9. Total operator + operand (n)

   Matrik "n" merupakan nilai matrik yang didapat dengan cara menghitung jumlah *Total Op* dengan *Total Opnd* dalam sebuah kode program. Rumus untuk menghitung "n" adalah n = Total Op + Total Opnd.

10. Volume (v)

    Matrik "v" merupakan nilai matrik yang didapat dengan cara menghitung sesuai rumus v = n * $\log_2$(Uniq Op + Uniq Opnd)

11. *Program length* (L)

    Matrik "L" merupakan nilai matrik yang didapat dengan cara menghitung sesuai rumus L = v/ n. v adalah volume didapat dengan perhitungan v = n * $\log_2$(Uniq Op + Uniq Opnd), *Uniq Op* adalah merupakan nilai matrik yang didapat dengan cara menghitung jumlah operator yang

berbeda dalam sebuah kode program. *Uniq Opnd* merupakan nilai matrik yang didapat dengan cara menghitung jumlah operand yang berbeda dalam sebuah kode program. Sedangkan untuk n sendiri di dapat dari perhitungan n = *Total Op* + *Total Opnd*.

12. *Difficulty* (d)

    Matrik "d" merupakan nilai matrik hasil perhitungan dari d = (*UniqOp* / 2) * (*TotalOpnd* / *UniqOpnd*).

13. *Intelligence* (i)

    Matrik i digunakan untuk mengukur kompleksitas algoritma yang diterapkan pada kode program dan tidak bergantung pada bahasa pemrograman yang dipakai. Rumus dari matrik i = v / d.

14. *Effort to write program* (e)

    Matrik e adalah matrik yang digunakan untuk mengukut *effort* atau usaha untuk menulis sebuah kode program. Rumus untuk menghitung nilai matrik ini dengan persamaan e = d * v.

15. *Error estimate* (b)

    Metric b merupakan matrik yang menunjukkan jumlah *bug* validasi, perhitungan b dilakukan dengan rumus b = V / 3000.

16. *Time to write program* (t)

    Matrik T merupakan matrik yang digunakan untuk mengukur perkiraan waktu yang proportional dari setiap *effort*. Rumus untuk menghitung nilai T dengan T = e / 18(*seconds*).

17. *Count of Statement Lines* (IOCode)

    Matrik *IOCode* merupakan matrik yang menghitung baris statement *(if, while, for*) dalam baris kode program.

18. *Count of lines of comments* (IOComment)

    Matrik *IOComment* merupakan matrik yang menghitung jumlah baris komen dalam kode program.

19. *Count of blank lines* (IOBlank)

    Matrik *IOBlank* merupakan matrik yang menghitung jumlah baris yang kosong dalam kode program.

20. *Count of Code and Comments Lines* (lOCodeAndComment)

    Matrik *IOCodeAndComment* merupakan matrik yang menghitung jumlah baris kode yang terdapat komen.

21. *Branch count* (branchCount)

    Matrik *branchCount* adalah matrik yang menghitung jumlah percabangan dalam kode program. Jumlah percabangan oleh statement *if* dan *case* dalam kode program.

2.3 Diskritisasi Data dengan *Equal Width Binning*

Diskritisasi adalah proses pengolahan data yang mengubah data kuantitatif menjadi data kualitatif (Yang *dkk.*, 2010). Diskritisasi data dalam proses prediksi cacat perangkat lunak dapat meningkatkan akurasi, metode *machine learning* yang digunakan Naïve Bayes (Singh & Verma, 2009).

Sebelum ke perhitungan nilai dengan menggunakan *Chi Square*, data atribut dalam *dataset* yang berbentuk data kontinyu harus dirubah terlebih dahulu menjadi data kategorikal. Dengan menggunakan metode *Equal Width Binning*

kita dapat mengkategorikan data dalam beberapa kategori untuk perhitungan dengan *Chi Square*.

Equal Width Binning adalah salah satu metode diskritisasi unsupervised, metode ini banyak digunakan karena *simple* sehingga mudah diterapkan. Adapaun tahapan melakukan diskritisasi data menuurut Yang dkk. (2010) meliputi;

a. Menentukan nilai K.

   K disini berarti interval yang membagi nilai Vmax dan Vmin dari setiap atribut pada *dataset*. Berikut rumus untuk menentukan K.

   $K = max\{1,2 \quad \log L\}$, L disini adalah jumlah nilai pengamatan yang berbeda untuk setiap fitur.

b. Menghitung lebar interval (W)

$$W = \frac{Vmax - Vmin}{K}$$

c. Menentukan titik potong dari tiap data dengan rumus sebagai berikut

   Vmin+W, Vmin+2W,…, Vmin+(K-1)W

   Memberikan nilai dari *dataset* sesuai rentang yang di dapat.

2.4 Seleksi Fitur

Pemilihan fitur merupakan salah satu hal yang penting dalam melakukan proses prediksi cacat pada perangkat lunak. Kesalahan dalam pemilihan fitur dapat menyebabkan penurunan tingkat prediksi. Beberapa masalah yang dapat menghambat data cacat pada software adalah redudansi, korelasi, fitur yang tidak relevan dan sampel yang tidak lengkap (Ghouti 2015).

Dalam penelitian sebelumnya penggunaan semua fitur yang ada di *dataset* tidak selalu dapat meningkatkan nilai hasil prediksi, hal itu disebabkan karena setiap data pada setiap fitur memiliki kualitas yang berbeda.

Menurut Chandrashekar and Sahin (2014), seleksi fitur dibagi menjadi 2 yakni *Filter* dan *Wrapper*. Pemilihan fitur yang di maksud disini adalah proses memilih fitur-fitur yang memiliki pengaruh dalam proses prediksi cacat perangkat lunak. Dimana dalam *dataset* di atas terdapat berbagai macam fitur dan akan dipilih fitur-fitur yang memiliki pengaruh untuk proses prediksi dengan menggunakan metode *Chi Square*.

Gao (2011) dalam penelitiannya menyebutkan terdapat 2 seleksi fitur, yaitu *filter* dan *wrapper*. Dalam penelitiannya, teknik seleksi fitur dengan menggunakan metode *filter* lebih baik daripada *wrapper*. oleh karena itu teknik *filter* seperti algoritma *Chi Square* di gunakan dalam penelitian ini dimana hasil fitur diambil berdasarkan perangkingan.

2.5 Seleksi Fitur dengan *Chi Square*

Setelah data berhasil di bentuk dalam kategori – kategori hasil dari diskritisasi pada langkah sebelumnya, selanjutnya melakukan proses pecarian nilai *Chi Square*. *Chi Square* yang digunakan untuk pemilihan fitur, akan merangking fitur yang memiliki yang memiliki rangking tertinggi sampai fitur yang memiliki rangking terendah. Berikut rumus *Chi Square* menurut Liu (2014);

$$x^2 = \sum \frac{(Oi - Ei)^2}{Ei}$$

Dimana ;

Oi = Frekuensi Kenyataan *(f0)*

Ei = Merupakan Frekuensi Harapan *(fh)*

Langkah-langkah untuk perhitungan *Chi Square* sebagai berikut;

Tahap 1, menentukan kelas untuk setiap atribut

Tahap 2, membuat frekuensi kenyataan (*f0*)

Tahap 3, menghitung frekuensi harapan (*fh*)

Tahap 4, menghitung nilai $(f0\text{-}fh)^2$

Tahap 5, menghitung nilai *Chi Square* ($\frac{(f0-fh)^2}{fh}$)

Tahap 6, Perangkingan nilai *Chi Square*.

2.6 Klasifikasi dengan *Naïve Bayes*

Pada Proses klasifikasi merupakan tahapan untuk mengolah input yang berupada kumpulan data dimana dalam penelitian ini, data yang digunakan adalah *dataset software defect prediction*. Setiap data memiliki atribut dan kelas pada setiap *dataset*nya yang dimisalkan atribut x dan kelas y. Data yang memiliki atribut x akan diprediksi nilai y nya. Model prediksi adalah algortima yang memiliki fungsi untuk mengklasifikasi nilai kategori kelas y berdasar nilai atribut x.

Teknik klasifikasi digunakan untuk membangun model prediksi dari *dataset input.* Banyak sekali teknik untuk klasifikasi, diantaranya SVM, Jaringan Saraf Tiruan, *Naïve Bayes, Decision Tree*, dan lain sebagainya. Pembuatan model prediksi meliputi 2 tahap, yaitu *training* dan *testing.*

Dalam menyelesaikan malasah klasifikasi, algoritma *Naïve Bayes* merupakan salah satu algoritma yang paling banyak digunakan karena algoritma tersebut lebih sederhana , lebih efektif. Menurut Akbar (2017), *Naïve Bayes* merupakan metode yang digunakan dalam bidang statistika untuk menghitung peluang dari suatu hipotesis, yaitu denga menghitung peluang pada setiap kelas berdasark pada atribut-atirbut yang dimiliki dan menentukan kelas yang memiliki probabilitas paling tinggi.

Berikut rumus umum dari teorema Bayes (Suyanto, 2017);

$$P(H|X) = \frac{P(H)P(X|H)}{P(X)}$$

$P(H|X)$: Probabilitas akhir bersyarat, suatu hipotesis H terjadi jika diberikan bukti X terjadi

$P(X|H)$: Probabilitas sebuah bukti X terjadi akan mempengaruhi hipotesis H

$P(H)$: Probabilitas awal hipotesis H terjadi tanpa memandang bukti apapun

$P(X)$: Probabilitas awal bukti X terjadi tanpa memandang hipotesis/bukti yang lain

Dan berikut merupakan rumus Naïve Bayes untuk proses klasifikasi :

$$P(H_k|X) = \frac{P(H_k)\prod_{i=1}^{n} P(X_i|H_k)}{P(X)}$$

Dimana $P(H_k|X)$ merupakan kemungkinan data dengan vektor X pada kelas H_k. $P(H_k)$ adalah kemungkinan awal kelas H_k. $\prod_{i=1}^{n} P(X_i|H_k)$ kemungkinan ketidak tergantungan (independensi) kelas H_k dari semua fitur - fitur n dalam vektor X. Mengingat bahwa P(X) bernilai sama untuk semua kelas yang artinya tuple X

memiliki probabilitas yang sama untuk masuk ke dalam kelas manapun, maka hanya $P(H_k)\prod_{i=1}^{n} P(X_i|H_k)$ yang akan di maksimalkan.

Untuk atribut yang bernilai kontinyu, yang umumnya diasumsikan memiliki distribusi Gaussian, $P(X_i|H_k)$ didefinisikan sebagai :

$$P(X_i|H_k) = \frac{1}{\sigma_{H_k}\sqrt{2\pi}} e^{-\frac{\left(x-\mu_{H_k}\right)^2}{2\sigma_{H_k}^2}}$$

Rumus utuk mencari Rata-rata adalah sebagai berikut :

$$\mu = \frac{x_1 + x_2 + x_3 + \ldots + x_n}{n}$$

Berikut rumus untuk mencari nilai varian :

$$s^2 = \frac{n\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_1)^2}{n(n-1)}$$

Sedangkan rumus untuk mencari standar deviasi adalah sebagi berikut :

$$\sigma = \sqrt{\frac{n\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_1)^2}{n(n-1)}}$$

Dimana :

P = Probabilitas

Xi = Atribut ke-i

Hk = Kelas yg dicari

μ = Rata-rata dari seluruh atribut

s^2 = Varian

σ = Standar deviasi , hasil penjumlahan varian dari seluruh atribut

x_i = Nilai x ke –i

n = Jumlah sampel.

Untuk memprediksi label kelas dari tupel X, maka harus dihitung terlebih dahulu probabilitas $P(X_i|H_k)P(H_k)$ untuk setiap kelas H_k. Lalu dipilihlah nilai terbesar dari probabilitas kelas hasil perhitungan $P(H_k)\prod_{i=1}^{n} P(X_i|H_k)$

2.7 Evaluasi Klasifikasi

Tahap Evaluasi ini bertujuan untuk mengetahui tingkat akurasi dari hasil penggunaan metode klasfikasi *Naïve Bayes.* Pada tahap evaluasi, akan didapatkan seberapa besar akurasi yang telah diperoleh. Teknik evaluasi yang akan digunakan dikenal sebagai *confusion matrix*. *Confusion matrix* ini akan merepresentasikan kebenaran dari sebuah prediksi.

Arar (2017) juga menjelaskan bahwa perhitungan *confusion matrix* adalah sebagai berikut;

Tabel 2. 2 Confusion Matrix

		Kenyataan	
		+	-
Hasil Prediksi	+	***True Positive***	***False Positive***
	-	***False Negative***	***True Negative***

- ***True Positive* (TP)** adalah jumlah data positif yang terklasifikasi dengan benar oleh sistem.
- ***False Positive* (FP)** adalah jumlah data positif namun terklasifikasi salah oleh sistem.
- ***False Negative* (FN)** adalah jumlah data negatif namun terklasifikasi salah oleh sistem.
- ***True Negative* (TN)** adalah jumlah data negatif yang terklasifikasi dengan benar oleh sistem.

Berdasarkan Nilai dari *confusion matrix* tersebut bisa dihitung nilai akurasi suatu prediksi. Untuk perhitungannya didefinisikan sebagai berikut:

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN}$$

BAB III
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Seperti yang sudah dituliskan pada bagian identifikasi masalah, penelitian ini berupaya memilih mana dinatara bebagai macam fitur atau atribut yang ada pada ke-5 *dataset* NASA MDP yang memiliki pengaruh signifikan terhadap proses prediksi cacat perangkat lunak.

Penulis mengkaji dari beberapa sumber atau referensi yang ada kaitannya dengan pemilihan fitur untuk prediksi cacat perangkat lunbak dengan berbagai metode yang digunakan dari masing-masing penelitian sebelumnya. Dengan adanya studi literatur ini diharapkan dapat memberikan gambaran secara lengkap terkait penelitian dan dapat memberikan dasar kontribusi dalam pemilihan fitur untuk prediksi cacat perangkt lunak yang akan dilakukan pada penelitian ini. Berikut ini merupakan beberapa referensi studi literatur yang dilakukan:

a. Penelitian sebelumnya yang telah melakukan prediksi cacat perangkat lunak berdasarkan *dataset* kode program dan menerapkan seleksi fitur terhadap *dataset* yang digunakan.

b. Metode pemilihan fitur dengan *Chi Square*.

c. Klasifikasi *Naïve Bayes* dan teknik evaluasi klasifikasi.

Data yang digunakan pada penelitian ini diunduh dari website PROMISE (*Predictor Models in Software Engineering*) *repository* melalui http://promise.site.uottawa.ca/SERepository dan pilih data untuk *software defect prediction* (prediksi cacat perangkat lunak). Terdapat 5 data terpilih yang berupa *dataset* yang menyimpan log cacat perangkat lunak. Dari *dataset* tersebut bisa

terlihat setiap baris menunjukkan modul terkecil dari sebuah program yaitu berupa *method* atau *function* dan setiap kolom menunjukkan nilai metrik. *Dataset* yang akan digunakan pada penelitian ini yaitu *dataset* CM1, JM1, KC1, KC2 dan PC1.

Data *input* yang akan digunakan pada penelitian kali ini berbentuk *dataset* yang berisikan data numerik. Berikut tampilan *dataset* yang akan digunakan dalam proses prediksi;

Tabel 3. 1 Dataset CM1

loc	v(g)	ev(g)	iv(G)	. . .	total Op	total Opnd	branchCount	defects
1.1	1.4	1.4	1.4	. . .	1.2	1.2	1.4	false
1.0	1.0	1.0	1.0	. . .	1.0	1.0	1.0	true
24.0	5.0	1.0	3.0	. . .	44.0	19.0	9.0	false
20.0	4.0	4.0	2.0	. . .	31.0	16.0	7.0	false
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
28.0	6.0	5.0	5.0	. . .	67.0	37.0	11.0	true

Tabel 3. 2 Dataset JM1

loc	v(g)	ev(g)	iv(G)	. . .	total Op	total Opnd	branch Count	defects
1.1	1.4	1.4	1.4	. . .	1.2	1.2	1.4	false
706.0	94.0	3.0	67.0	. . .	1918.0	784.0	163.0	false
20.0	2.0	1.0	2.0	. . .	26.0	21.0	3.0	false
609.0	6.0	1.0	6.0	. . .	1496.0	1378.0	17.0	false
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
1129.0	128.0	14.0	104.0	. . .	0.0	0.0	0.0	true

Tabel 3. 3 Dataset KC1

loc	v(g)	ev(g)	iv(G)	. . .	total Op	total Opnd	branchCount	defects

1.1	1.4	1.4	1.4	. . .	1.2	1.2	1.4	false
5.0	1.0	1.0	1.0	. . .	7.0	4.0	1.0	false
3.0	1.0	1.0	1.0	. . .	1.0	0.0	1.0	False
3.0	1.0	1.0	1.0	. . .	1.0	0.0	1.0	false
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
116.0	12.0	6.0	12.0	. . .	509.0	285.0	89.0	true

Tabel 3. 4 Dataset KC2

loc	v(g)	ev(g)	iv(G)	. . .	total Op	total Opnd	branchCount	defects
1.1	1.4	1.4	1.4	. . .	1.2	1.2	1.4	false
1.0	1.0	1.0	1.0	. . .	1.0	1.0	1.0	true
415.0	59.0	50.0	51.0	. . .	692.0	467.0	106.0	true
230.0	33.0	10.0	16.0	. . .	343.0	232.0	65.0	true
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
3.0	1.0	1.0	1.0	. . .	1.0	0.0	1.0	true

Tabel 3. 5 Dataset PC1

loc	v(g)	ev(g)	iv(G)	. . .	total Op	total Opnd	branchCount	defects
1.1	1.4	1.4	1.4	. . .	1.2	1.2	1.4	false
10.0	1.0	1.0	1.0	. . .	14.0	11.0	1.0	false
29.0	9.0	6.0	3.0	. . .	48.0	31.0	8.0	false
8.0	2.0	1.0	1.0	. . .	8.0	3.0	3.0	false
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
27.0	7.0	1.0	4.0	. . .	55.0	39.0	12.0	true

Dari ke -5 *dataset* diatas, jika di rangum berdaser informasinya bisa dilihat pada tabel 3.6 di bawah ini;

Tabel 3. 6 Perbedaan dataset

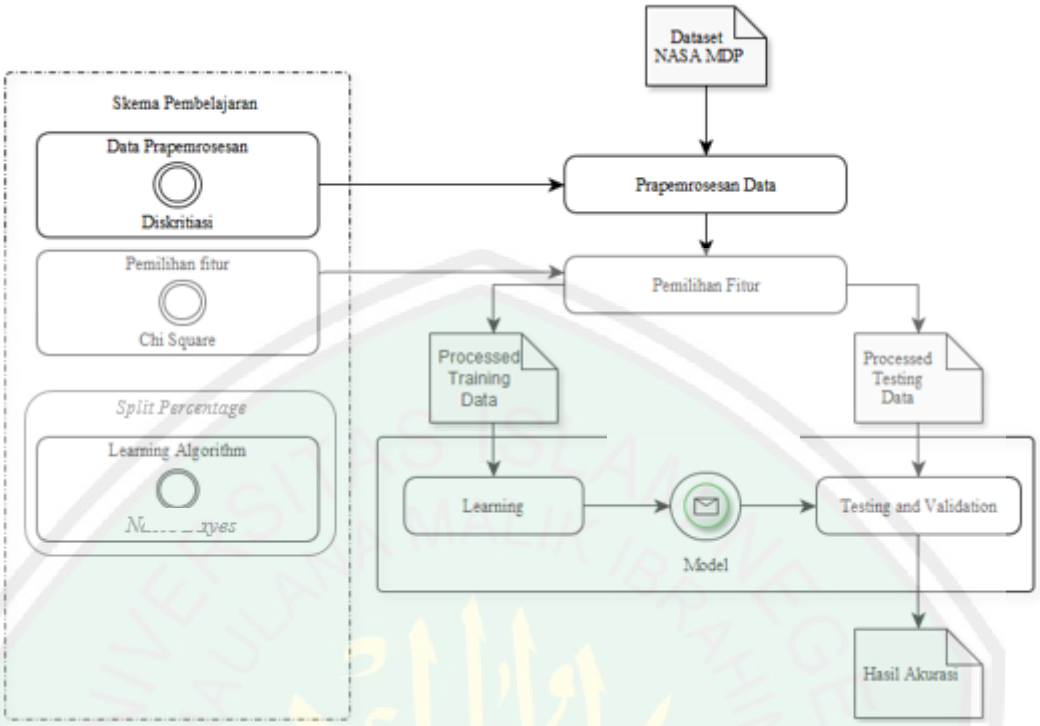
Dataset	Jumlah Fitur/Atribut	Jumlah Data Berbeda
CM1	22	498
JM1	22	10885
KC1	22	2109
KC2	22	522
PC1	22	1109

Untuk perbedaan di setiap *dataset*nya dapat dilihat pada tabel 3.6 , disana dijelaskan bahwa *dataset* yang digunakan sama-sama menggunakan 5 *dataset* yang berasal dari NASA MDP yang terdiri dari 22 atribut (*5 different lines of code measure, 3 McCabe metrics, 4 base Halstead measures, 8 derived Halstead measures, a branch-count, and 1 goal field)* . Jumlah instan menunjukkan jumlah data pada setiap atributnya. Dimana setiap atribut memiliki jumlah instan yang berbeda-beda dan bisa dilihat pada tabel 3.6 diatas.

Dari fitur-fitur yang ada pada setiap *dataset* diatas, akan dilakukan dilakukan proses pemilihan fitur-fitur yang mempunyai pengaruh signifikan pada proses prediksi cacat perangkat lunak. Pemilihan fitur dilakukan menggunakan Algoritma *Chi Square.* Setelah didapatkan fitur-fitur dengan menggunakan *Chi Square*, fitur-fitur yang terpilih akan digunakan untuk proses prediksi yang akan dilakukan dengan menggunakan *Naïve Bayes* untuk mendapatkan hasil akurasi.

3.2 Perancangan Sistem

Pada bagian ini akan dibahas tentang alur sistem yang diusulkan, alur sistem dapat dilihat pada gambar 3.1



Gambar 3. 1 Desain Sistem Yang Diusulkan

3.2.1. Proses Diskritisasi dengan *Equal Width Binning*

Langkah yang pertama sebelum perhitungan dengan menggunakan *Chi Square*, data di kategorikan dengan menggunakan proses diskritisasi. Data dalam bentuk kategori itulah yang nanti akan dibutuhkan dalam perhitungan *Chi Square*.

Proses diskritisasi data yang digunakan disini adalah dengan *Equal Width Binning*. Alasan menggunakan diskritisasi data dengan *Equal Width Binning* karena mampu meningkatkan akurasi klasifikasi. Adapaun tahapan melakukan diskritisasi data meliputi;

a. Menentukan nilai K.

   K disini berarti interval yang membagi nilai Vmax dan Vmin dari setiap atribut pada *dataset*. Berikut rumus untuk menentukan K.

   $K = max\{1,2 \ \log L\}$, L disini adalah jumlah nilai pengamatan yang berbeda untuk setiap fitur.

b. Menghitung lebar interval (W)

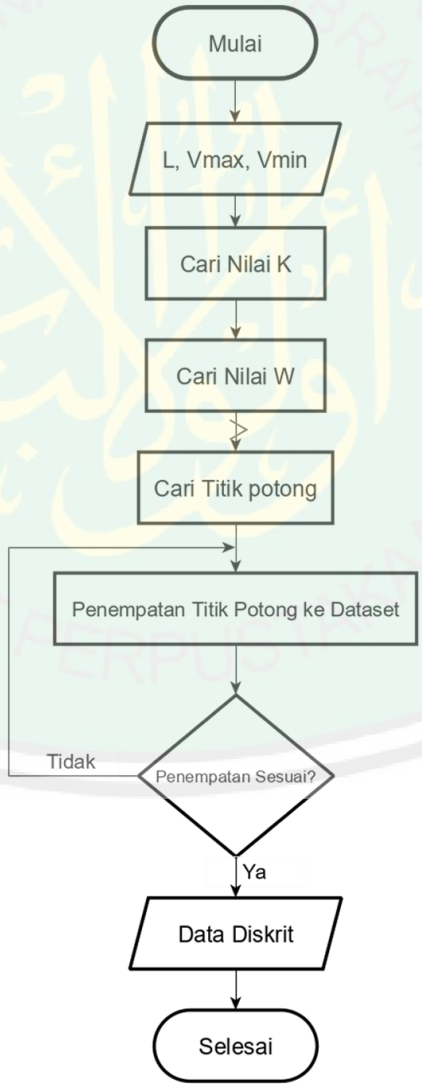
$$W = \frac{Vmax - Vmin}{K}$$

c. Menentukan titik potong dari tiap data dengan rumus sebagai berikut

Vmin+W, Vmin+2W,…, Vmin+(K-1)W

d. Memberikan nilai dari *dataset* sesuai rentang yang di dapat .

Berdasarkan alur tahap perhitungan yang telah dijelaskan jika dibentuk dalam sebuah *flowchart*, seperti berikut;



Gambar 3. 2 Flowchart diskritisasi Equal Width Binning

Berikut adalah beberapa contoh data dari *dataset* CM1.

Tabel 3. 7 contoh dataset CM1

loc	v(g)	ev(g)	...	i	*defect*
1.1	1.4	1.4	...	1.3	*FALSE*
1	1	1	...	1	*TRUE*
24	5	1	...	32.54	*FALSE*
20	4	4	...	13.47	*FALSE*
24	6	6	...	19.97	*FALSE*
...	...	...	...	...	...

Berikut contoh dari perhitungan dari diskritisasi dengan *Equal Width Binning* pada atribut i *dataset* CM1.

Diketahui:

L (Jumlah Nilai berbeda) untuk atribut loc=102, v(g)=34, i=396, .., ke-n

Vmin atribut loc = 1, v(g) = 1, i = 0, …, ke-N

Vmax atribut loc = 423, v(g) = 180, i = 293,68 , …, ke-N

Perhitungan diskritisasi untuk atribut i;

a. Hitung nilai K

K = max(1,2 log L)

K = max(1,2 log 396)

K = max(3,117)

K = 4

b. Hitung nilai W

W = (Vmax – Vmin) / K

W = (293,68 – 0) / 4

W = 73,42

c. Tentukan titik potong Vmin+W, Vmin+2W,…, Vmin + (K-1)W

Vmin + W → 0 + 73,42 = 73,42

Vmin + 2W → 1 + (2 x 73,42) = 146,84

Vmin + 2W → 1 + (3 x 73,42) = 220,26

Maka hasil titik potongnya;

1. Interval (-inf – 73,42)
2. Interval (73,42– 146,84)
3. Interval (146,84– 220,26)
4. Interval (220,26 - inf)

d. Memberikan nilai *dataset* sesui interval diatas

Tabel 3. 8 Hasil perhitungan diskritisasi atribut i pada CM1

loc	v(g)	ev(g)	...	i	*defect*
1,1	1.4	1.4	...	(-inf – 73,42)	*FALSE*
1	1	1	...	(-inf – 73,42)	*TRUE*
24,5	5	1	...	(-inf – 73,42)	*FALSE*
20,4	4	4	...	(-inf – 73,42)	*FALSE*
24,6	6	6	...	(-inf – 73,42)	*FALSE*
...	...	...	...	...	...

Ulangi langkah 1 sampai 4 untuk semua atribut sehingga hasil akhir dari proses diskritisasi bisa dilihat pada tabel 3.9.

Tabel 3. 9 Hasil perhitungan diskritisasi CM1

loc	v(g)	ev(g)	...	i	*defect*
(-inf-425.66666)	(-inf-48.5)	(-inf-15.5)	...	(-inf-73.42)	*FALSE*
(-inf-425.66666)	(-inf-48.5)	(-inf-15.5)	...	(-inf-73.42)	*TRUE*
(-inf-425.66666)	(-inf-48.5)	(-inf-15.5)	...	(-inf-73.42)	*FALSE*
(-inf-425.66666)	(-inf-48.5)	(-inf-15.5)	...	(-inf-73.42)	*FALSE*

(-inf-425.66666)	(-inf-48.5)	(-inf-15.5)	...	(-inf-73.42)	*FALSE*
...	...	...	...	...	...

3.2.2. Pemilihan Fitur dengan Menggunakan *Chi Square*

Dalam *dataset* terdapat berbagai fitur, dan dari sekian banyak fitur tersebut, ada beberapa fitur yang memiliki pengaruh besar da nada juga fitur yang kurang berpengaruh pada proses pengujian prediksi cacat perangkat lunak. Oleh karena itu, perlu dilakukan pemilihan fitur dengan tujuan mimilih dari sekian banyak fitur tersebut yang memiliki pengaruh besar untuk proses pengujian prediksi cacat perangkat lunak. Yaitu dengan menggunakan metode *Chi Square* sebagai proses pemilihan fitur dengan memilih fitur – fitur yang memiliki rangking atau nilai terbesar pada proses pemilihan fitur nanti.

Seleksi fitur dengan *Chi Square* menggunakan teori statistika untuk menguji independensi sebuah term dengan kategorinya. Berdasarkan teori statistika, dua peristiwa di antaranya adalah, kemunculan dari fitur dan kemunculan dari kategori, yang kemudian setiap nilai termdiurutkan dari yang tertinggi.

Seleksi fitur dilakukan untuk mereduksi fitur-fitur yang tidak relevan dalam proses klasifikasi oleh *Naïve Bayes*. Seleksi fitur menggunakan metode *Chi Square* yang berbasis statistika untuk menguji independensi sebuah term dengan kategorinya. Salah satu tujuan penggunaan seleksi fitur adalah untuk menghilangkan fitur-fitur yang kurang optimal dalam klasifikasi

Adapun proses algoritma *Chi Square* adalah sebagai berikut;

Tahap 1, menentukan kelas untuk setiap atribut

Tahap 2, membuat frekuensi kenyataan (*f0*)

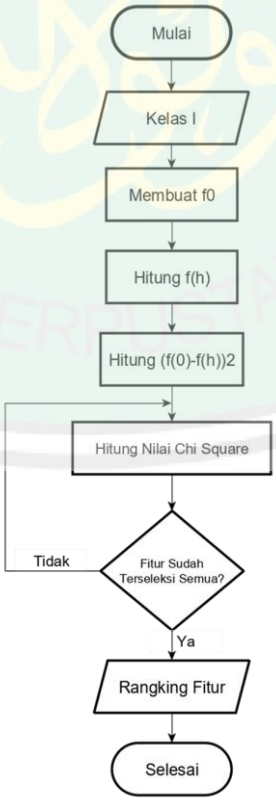
Tahap 3, menghitung frekuensi harapan (*fh*)

Tahap 4, menghitung nilai $(f0\text{-}fh)^2$

Tahap 5, menghitung nilai *Chi Square* ($\frac{(f0-fh)^2}{fh}$)

Tahap terakhir adalah dengan melakukan perankingan berdasarkan nilai *Chi* terbesar dan mengambil beberapa atribut yang dianggap memiliki pengaruh terhadap proses prediksi. Jumlah atribut yang di pilih bisa dilakukan uji dengan berbagai pilihan , semisal 3 atribut, 5 atribut, 7 atribut, atau 10 atribut dimana akan dihasilkan hasil prediksi dengan jumlah atribut yang sesuai.

Jika dilihat dari tahapan prosesnya, berikut *flowchart* dari metode *Chi Square*.



Gambar 3. 3 Flowchart Algoritma Chi Square

Contoh perhitungan nilai *Chi Square* dapat dilihat pada proses dibawah ini.

Tahap 1, menentukan kelas. Pada tahap ini, kelas ditentukan pada langkah sebelumnya dengan menggunakan diskritisasi, dimana atribut yang dihitung adalah atribut i, untuk lebih jelasnya, lihat pada tabel 3.10

Tabel 3. 10 Penentuan Kelas pada atribut i

loc	Kelas *Defect*		Total
	Ya	Tidak	
Interval (-inf – 73,42)	35	401	436
Interval (73,42 – 146,84)	8	41	49
Interval (146,84 – 220,26)	4	6	10
Interval (220,26 – inf)	2	1	3
Total	49	449	498

Tahap 2, membuat frekuensi kenyataan (*f0*)

Berikut cara untuk membuat frekuensi kenyataan (*f0)*, seperti pada tabel 3.11

Tabel 3. 11 Membuat frekuensi kenyataan (*f0*)

Cell	*f0*
a	35
b	401
c	8
d	41
e	4
f	6
g	2
h	1

Tahap 3, menghitung nilai frekuensi harapan (*fh*) per *cell*. Rumus untuk menghitung frekuensi harapan adalah sebagai berikut

$$fh = \frac{\text{total baris}}{\text{total semua}} \text{x total kolom}$$

maka, akan menghasilkan ;

$$fh\ cell\ \text{a} = \frac{436}{498} \text{x } 49 = 42.89959839$$

$$fh\ cell\ \text{b} = \frac{436}{498} \text{x } 449 = 393.1004016$$

$$fh\ cell\ \text{c} = \frac{49}{498} \text{x } 49 = 4.821285141$$

$$fh\ cell\ \text{d} = \frac{436}{498} \text{x } 449 = 44.17871486$$

$$fh\ cell\ \text{e} = \frac{10}{498} \text{x } 49 = 0.983935743$$

$$fh\ cell\ \text{f} = \frac{10}{498} \text{x } 449 = 9.016064257$$

$$fh\ cell\ \text{g} = \frac{3}{498} \text{x } 49 = 0.295180723$$

$$fh\ cell\ \text{h} = \frac{3}{498} \text{x } 449 = 2.704819277$$

Untuk lebih mudahnya lihat tabel 3.12

Tabel 3. 12 Perhitungan frekuensi harapan (*fh*)

Cell	*f0*	***fh***
a	35	42.89959839
b	401	393.1004016
c	8	4.821285141
d	41	44.17871486
e	4	0.983935743
f	6	9.016064257
g	2	0.295180723
h	1	2.704819277

Tahap ke 4, menghitung hasil kuadrat dari pengurangan frekuensi kenyataan dikurangi dengan frekuensi harapan $(f0 - fh)^2$.

cell a = $(35 - 42.89959839)^2$ = -7.899598394

cell b = $(401 - 393.1004016)^2$ = 7.899598394

lakukan perhitungan seperti diatas sampai pada *cell* h.Untuk lebihmudahnya, pada tahap ini, lihat perhitungan $(f0 - fh)^2$ pada tabel dibawah ini.

Tabel 3. 13 perhitungan $(f0 - fh)^2$

Cell	*f0*	*fh*	*f0 - fh*	$(f0 - fh)^2$
a	35	42.89959839	-7.899598394	62.40365478
b	401	393.1004016	7.899598394	62.40365478
c	8	4.821285141	3.178714859	10.10422816
d	41	44.17871486	-3.178714859	10.10422816
e	4	0.983935743	3.016064257	9.096643603
f	6	9.016064257	-3.016064257	9.096643603
g	2	0.295180723	1.704819277	2.906408768
h	1	2.704819277	-1.704819277	2.906408768

Kita menggunakan kolom *f0 – fh* sebagai kolom bantuan untuk mempermudah perhitungan hasil kuadrat dari pengurangan frekuensi kenyataan dikurangi dengan frekuensi harapan $(f0 - fh)^2$.

Tahap ke 5, yaitu menghitung nilai *Chi Square* dari atribut i. Rumus untuk menghitung nilai *Chi Square* adalah sebagai berikut:

$$x^2 = \sum \frac{(Oi - Ei)^2}{Ei}$$

Dimana O sama dengan frekuensi kenyataan, dan E sama dengan frekuensi harapan. Maka perhitungannya bisa dilihat seperti dibawah ini:

$$Cell\ \text{a} = \frac{(f0-fh)^2}{fh}$$

$$= \frac{62.40365478}{42.89959839}$$

$$= 1.454644265$$

$$Cell\ \text{b} = \frac{(f0-fh)^2}{fh}$$

$$= \frac{0.15874737}{1.454644265}$$

$$= 0.15874737$$

Lakukan perhitungan seperti di atas hingga cell h, makan hasilnya bisa di lihat seperti tabel 3.14

Tabel 3. 14 Hasil perhitungan nilai Chi Square

Cell	*f0*	*fh*	*f0 - fh*	$(f0 - fh)^2$	$\frac{(f0-fh)^2}{fh}$
a	1.454644265	1.454644265	1.454644265	1.454644265	1.454644265
b	0.15874737	0.15874737	0.15874737	0.15874737	0.15874737
c	2.095754112	2.095754112	2.095754112	2.095754112	2.095754112
d	0.228712587	0.228712587	0.228712587	0.228712587	0.228712587
e	9.245160233	9.245160233	9.245160233	9.245160233	9.245160233
f	1.008937308	1.008937308	1.008937308	1.008937308	1.008937308
g	9.846201131	9.846201131	9.846201131	9.846201131	9.846201131
h	1.074529745	1.074529745	1.074529745	1.074529745	1.074529745
Nilai *Chi Square*					25.11268675

Ulangi perhitungan diatas untuk mencari nilai *Chi Square* untuk atribut yang lainnya sehingga di dapat nilai dari *Chi Square* dari seluruh fitur yang ada.

Tahap terakhir, yaitu melakukan perankingan sesuai dengan nilai *Chi Square* setiap atribut. Perangkingan ini untuk memilih fitur-fitur yang berpengaruh untuk proses klasifikas. Hasil dari proses ini dapat dilihat pada Tabel 3.15 yang menunjukkan hasil perankingan atribut sesuai dengan nilai *Chi Square* pada *dataset* CM1.

Tabel 3. 15 Hasil perankingan perhitungan Chi Square pada dataset CM1

No.	*Chi Square*	Atribut
1.	25.112686	i
2.	19.403292	t
3.	19.403292	e
4.	17.889837	uniq_Opnd
5.	15.650128	v
6.	14.040755	uniq_Op
7.	13.643305	d
8.	10.465001	lOComment
9.	9.261633	b
10.	7.6635766	n
11.	6.6203456	loc
12.	6.620345	total_Opnd
13.	5.795301	total_Op
14.	3.817894	lOBlank
15.	3.778721	ev(g)
16.	1.8779099	branchCount
17.	1.8779099	iv(g)
18.	1.8779099	v(g)
19.	0.15822342	lOCode
20.	0.06936351	l
21.	0.0	locCodeAndComment

Pada penelitian terdahulu oleh Wang dkk (2012) dijelaskan bahwa jumlah seleksi fitur yang direkomendasikan adalah $\log_2 n$ dengan nilai n adalah seluruh atribut. Dimana hasil fitur yang terpilih sejumlah 5 fitut dengan rangking tertinggi tiap *dataset.* Hasil seleksi fitur dari keseluruhan atribut pada *dataset* CM1 hasilnya bisa dilihat pada tabel 3.16 berikut ini.

Tabel 3. 16 Lima fitur terpilih pada dataset CM1

No.	Nilai	Fitur
1.	25.112686	i
2.	19.403292	t
3.	19.403292	e
4.	17.889837	uniq_Opnd
5.	15.650128	v

3.2.3 Klasifikasi *Naïve Bayes*

Pada tahapan ini, data dipecah terlebih dahulu menjadi 2 bagian , yaitu data *training* dan data *testing.* Teknik *split percentage* sebesar 60% digunakan untuk memecah data tersebut, sehingga dari contoh *dataset* CM1 yang memiliki data sebanyak 498 baris dibagi menjadi 2 bagian , yaitu *testing* dan *training*.

Dimana 60% persen dari 498 adalah 299 baris, akan dijadikan menjadi data *training* , dan 40 persen sisanya akan dijadikan data *testing*. Karena persebaran label pada *dataset* yang tidak merata , maka di ambil data mulai 60% dari *dataset* yang dimulai dari bawah , yaitu mulai dari baris 201-498 akan dijadikan data *training*, . sedangkan baris 1-200 akan dijadikan data *testing*. Dan dari data *testing* tersebut akan didapatkan akurasinya.

Selanjutnya untuk data training terlebih dahulu dicari nilai probabilitasnya dari setiap fitur. Karena data yang digunakan dalam penelitian ini bertipe data numerik kontinu, maka perhitungan *Naïve Bayes* sesuai dengan distribusi *Gaussian* dengan rumus sebagai berikut :

$$P(X_i|H_k) = \frac{1}{\sigma_{H_k}\sqrt{2\pi}} e^{-\frac{\left(x-\mu_{H_k}\right)^2}{2\sigma_{H_k}^2}}$$

Dari hasil seleksi fitur menggunakan *Chi Square* ditemukan sudah terseleksi fitur yang memiliki nilai tertinggi hingga terendah. Kita ambil 5 fitur dengan nilai tertinggi secara berurutan , yaitu fitur i, t, e , Uniq_Opnd, dan v.

Rumus utuk mencari Rata-rata adalah sebagai berikut :

$$\mu = \frac{x_1 + x_2 + x_3 + \ ... + x_n}{n}$$

Berikut rumus untuk mencari nilai varian :

$$s^2 = \frac{n\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_1)^2}{n(n-1)}$$

Sedangkan rumus untuk mencari standar deviasi adalah sebagi berikut :

$$\sigma = \sqrt{\frac{n\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_1)^2}{n(n-1)}}$$

Kemudian langkah selanjutnya adalah mencari nilai rata-rata dan standar deviasi dari fitur-fitur tersebut, perhitungannya sebagai berikut :

1. Menghitung nilai probabilitas rata-rata pada fitur i pada kelas *defect* "*True*".

$$Rata - rata = \frac{38{,}55 \ + \ 52{,}03 \ + \ ... \ + \ 35{,}08}{48} = \ 69{,}816875$$

$$Varian = \frac{((48x750004674{,}2) - 39254090190)}{48 * 47} = 3541{,}011435$$

$$Standar\ Deviasi = \sqrt{3541{,}011435} = 59{,}50639827$$

2. Menghitung nilai probabilitas rata-rata pada fitur i pada kelas *defect* "*False*".

$$Rata - rata = \frac{13 + 7{,}86 + \ldots + 21{,}83}{251} = 33{,}90545817$$

$$Varian = \frac{((251x527665{,}1025) - 72424695{,}47)}{251 * 250} = 956{,}4819961$$

$$Standar\ Deviasi = \sqrt{956{,}4819961} = 30{,}92704312$$

3. Menghitung nilai probabilitas rata-rata pada fitur t pada kelas *defect* "*True*".

$$Rata - rata = \frac{991{,}46 + 439{,}7 + \ldots + 504{,}35}{48} = 4228{,}398$$

$$Varian = \frac{((48x3442602743) - 41194015902)}{48 * 47} = 54987108{,}05$$

$$Standar\ Deviasi = \sqrt{54987108{,}05} = 7415{,}329261$$

4. Menghitung nilai probabilitas rata-rata pada fitur t pada kelas *defect* "*False*".

$$Rata - rata = \frac{6{,}5 + 2{,}73 + \ldots + 131{,}62}{251} = 1807{,}451155$$

$$Varian = \frac{((251x17258779044) - 2{,}05817E + 11)}{251 * 250} = 65755168{,}98$$

$$Standar\ Deviasi = \sqrt{65755168{,}98} = 8108{,}956097$$

5. Menghitung nilai probabilitas rata-rata pada fitur e pada kelas *defect* "*True*".

$$Rata - rata = \frac{117 + 49{,}13 + \ldots + 2369{,}07}{48} = 32534{,}11398$$

$$Varian = \frac{((48x1{,}1154E + 12) - 1{,}33469E + 13)}{48 * 47} = 17815825255$$

$$Standar\ Deviasi = \sqrt{17815825255} = 133475{,}9351$$

6. Menghitung nilai probabilitas rata-rata pada fitur e pada kelas *defect* "*False*".

$$Rata - rata = \frac{17846{,}19 + 7914{,}68 + \ldots + 9078{,}38}{251} = 76111{,}15354$$

$$Varian = \frac{((48x5{,}59184E + 12) - 6{,}66846E + 13)}{251 * 250} = 21304674351$$

$$Standar\ Deviasi = \sqrt{21304674351} = 145961{,}2084$$

7. Menghitung nilai probabilitas rata-rata pada fitur Uniq_Opnd pada kelas *defect* "*True*".

$$Rata - rata = \frac{32 + 33 + \ldots + 23}{48} = 53{,}35416667$$

$$Varian = \frac{((48x278871) - 6558721)}{48 * 47} = 3026{,}191046$$

$$Standar\ Deviasi = \sqrt{3026{,}191046} = 55{,}01082663$$

8. Menghitung nilai probabilitas rata-rata pada fitur Uniq_Opnd pada kelas *defect* "*False*".

$$Rata - rata = \frac{4 + 2 + \ldots + 12}{251} = 22{,}27888446$$

$$Varian = \frac{((251x326682) - 31270464)}{251 * 250} = 808{,}3939124$$

$$\text{Standar Deviasi} = \sqrt{808{,}3939124} = 28{,}43226886$$

9. Menghitung nilai probabilitas rata-rata pada fitur v pada kelas *defect* "*True*".

$$Rata - rata = \frac{829{,}45 + 641{,}73 + \ldots + 564{,}33}{48} = 1997{,}402917$$

$$Varian = \frac{((48x481012914{,}3) - 9192080820)}{48 * 47} = 6159813{,}415$$

$$Standar\ Deviasi = \sqrt{6159813{,}415} = 2481{,}89714$$

10. Menghitung nilai probabilitas rata-rata pada fitur v pada kelas *defect* "*False*".

$$Rata - rata = \frac{39 + 19{,}65 + \ldots + 227{,}43}{251} = 789{,}3484064$$

$$Varian = \frac{((251 * 750004674{,}2) - 39254090190)}{251 * 250} = 2374455{,}507$$

$$Standar\ Deviasi = \sqrt{2374455{,}507} = 1540{,}926834$$

Jika hasil rata-rata dan standar deviasi pada setiap fitur sudah dihitung, maka langkah selanjutnya akan dilakukan pengujian dengan data *testing* yang ada. Data testing diambil pada baris ke - 1 sampai baris ke – 199, sebagai contoh data testing diambil pada baris pertama, untuk datanya bias dilihat pada table dibawah ini:

Tabel 3. 17 Baris Ke-48 Set Pertama Dataset CM1

i	t	e	uniq_Opnd	v	defects
1,3	1,3	1,3	1,2	1,3	?

1. Perhitungan nilai probabilitas menggunakan rumus distribusi Gaussian :

$$P(\text{i} = 1{,}3|TRUE) = \frac{1}{59{,}50639827\sqrt{2\pi}} e^{\frac{-(1{,}3 - 69{,}816875)^2}{(2\times 59{,}50639827)^2}} = 0{,}003455966$$

$$P(\text{i} = 1{,}3|FALSE) = \frac{1}{30{,}92704312\sqrt{2\pi}} e^{\frac{-(1{,}3 - 33{,}90545817)^2}{(2\times 30{,}92704312)^2}} = 0{,}007401603$$

$$P(t = 1{,}3|TRUE) = \frac{1}{7415{,}329261\sqrt{2\pi}} e^{\frac{-(1{,}3 - 4228{,}397708)^2}{(2\times 7415{,}329261)^2}} = 4{,}57432$$

$$P(t = 1{,}3|FALSE) = \frac{1}{8108{,}956097\sqrt{2\pi}} e^{\frac{-(1{,}3 - 1807{,}451155)^2}{(2\times 8108{,}956097)^2}} = 4{,}80045$$

$$P(e = 1{,}3|TRUE) = \frac{1}{133475{,}9351\sqrt{2\pi}} e^{\frac{-(1{,}3 - 76111{,}15354)^2}{(2\times 133475{,}9351)^2}} = 2{,}54105$$

$$P(e = 1{,}3|FALSE) = \frac{1}{145961{,}2084\sqrt{2\pi}} e^{\frac{-(1{,}3 - 32534{,}11398)^2}{(2\times 145961{,}2084)^2}} = 2{,}66683$$

$$P(\text{uniq\_Opnd} = 1{,}2|TRUE) = \frac{1}{55{,}01082663\sqrt{2\pi}} e^{\frac{-(1{,}2 - 53{,}35416667)^2}{(2\times 55{,}01082663)^2}} = 0{,}0046279$$

$$P(\text{uniq\_Opnd} = 1{,}2|FALSE) = \frac{1}{28{,}43226886\sqrt{2\pi}} e^{\frac{-(1{,}3 - 22{,}27888446)^2}{(2\times 28{,}43226886)^2}} = 0{,}010662463$$

$$P(\text{v} = 1{,}3|TRUE) = \frac{1}{2481{,}89714\sqrt{2\pi}} e^{\frac{-(1{,}3 - 1997{,}402917)^2}{(2\times 2481{,}89714)^2}} = 0{,}000116353$$

$$P(\text{v} = 1{,}3|FALSE) = \frac{1}{28{,}43226886\sqrt{2\pi}} e^{\frac{-(1{,}3 - 789{,}3484064)^2}{(2\times 1540{,}926834)^2}} = 0{,}000227219$$

2. Mencari nilai probabilitas akhir untuk setiap kelas

- P(X | *TRUE*) = P(i = 1,3|*TRUE*) X P(t = 1,3|*TRUE*) X P(e = 1,3|*TRUE*) X P(Uniq_Opnd = 1,2|*TRUE*) X P(v = 1,3|*TRUE*) = 2,1631E-19
- P(X | *FALSE*) = P(i = 1,3| *FALSE*) X P(t = 1,3| *FALSE*) X P(e = 1,3| *FALSE*) X P(Uniq_Opnd = 1,2| *FALSE*) X P(v = 1,3| *FALSE*) = 2,29565E-18

P(*DEFECT = TRUE)* = 48/299 = 0,160535117

P(*DEFECT = FALSE)* = 251/299 = 0,839464883

P(*X | TRUE)* = 2,1631E-19 x 0,160535117 = 3,47254E-20

P(*X | FALSE)* = 2,29565E-18 x 0,839464883 = 1,92712E-18

Dari hasil perhitungan diatas dapat kita lihat bahwa hasil dari pengujian yang kita lakukan terhadap data *testing* pada baris pertama, menunjunkkan hasil nilai terbesar terdapat pada kelas *Defect = False*. Dengan demikian pada baris pertama pengujian ini mengahasilkan prediksi *False*. Lakukan tahapan tersebut sampai dengan data uji ke 199.

3.2.4 Perhitungan Akurasi

Untuk mendapat nilai akurasi dari hasil pemilihan fitur dengan *Chi Square* dan proses klasifikasi dengan ,maka dilakukan perhitungan dengan menggunakan *confusion matrix.* Nilai yang ada pada *confusion matrix* diperoleh dari data prediksi yang sudah dilakukan sebelumnya.

Dengan melihat hasil prediksi pada klasifikasi *Naïve Bayes*, dapat kita kelompokkan hasil prediksi kedalam *confusion matrix*. Berikut salah satu contoh perhitungan akurasi dari proses prediksi cacat perangkat lunak pada *dataset* CM1.

Tabel 3. 18 Confusion Matrix dataset CM1

		Kenyataan	
		+	-
Hasil Prediksi	+	**182**	**1**
	-	**16**	**0**

Pada tabel *confusion matrix* diatas sudah kita masukkan data hasil prediksi. dimana 182 data didapat dari hasil prediksi oleh pengklasifikasi banyaknya jumlah data *"true"* yang terklasifikasi *"true"* oleh pengklasifikasi. Angka 16 diperoleh dari banyaknya data *"true"* yang terklasifikasi *"false"* oleh pengklasifikasi. Angka 1 diperoleh dari banyaknya data *"false"* yang terlasifikasi *"true"* oleh pengklasifikasi. Dan yang terakhir, angka 0 diperoleh dari banyaknya data *"false"* yang benar terklasifikasi *"false"* oleh pengklasifikasi. Sesuai dengan rumus pada sub bab sebelumnya, maka perhitungan akurasi untuk data testingnya sebagai berikut;

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Akurasi} = \frac{182+0}{182+0+1+16} = 0{,}91457$$

BAB IV

UJI COBA DAN PEMBAHASAN

Pada bab ini akan membahas hasil dari uji coba rancangan atau desain sistem yang telah dibuat pada bab sebelumnya. Penjelasan pertama pada bab ini adalah tentang uji coba penelitian yang berisi tentang lingkungan uji coba, data yang digunakan, tampilan sistem yang berhasil dibuat, hasil uji coba seleksi fitur dan hasil uji coba klasifikasi.

4.1 Uji Coba

Sebelum menjelaskan proses uji coba terlebih dahulu akan dijelasan hal-hal yang berkaitan terhadap proses uji coba yaitu lingkungan uji coba dan juga data yang digunakan untuk uji coba.

4.1.1 Lingkungan Uji Coba

Lingkungan uji coba menjelaskan tentang spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam penelitian ini, adapun spesifikasi perangkat keras yang digunakan adalah sebagai berikut;

a. *Processor 2,3 GHz Intel Core i5.*
b. *Memory 12 GB 2133 MHz DDR4.*
c. *Hard Disk 1TB*
d. *VGA NVIDIA GEFORCE 930MX*

Sedangkan spesifikasi perangkat lunak yang digunakan adalah sebagai berikut;

a. Sistem operasi *Windows 10 Pro*
b. *Netbeans 8.2*
c. *Microsoft Excel 2013*

4.1.2 Data Uji coba

Data uji yang digunakan pada penelitian ini merupakan *dataset* yang didapat dari NASA *Metric Data program* (MDP) yang berjumlah 5 *dataset* yaitu CM1, JM1, KC1, KC2 dan PC1. Berikut adalah 5 *dataset* yang digunakan pada penelitian ini;

Tabel 4. 1 Dataset CM1

No	loc	v(g)	ev(g)	iv(g)	….	defect
1	1.1	1.4	1.4	1.4	…	False
2	1	1	1	1	…	True
3	24	5	1	3	…	False
…	…	…	…	…	…	…
498	28	6	5	5	…	True

Tabel 4. 2 Dataset JM1

No	loc	v(g)	ev(g)	iv(g)	….	defect
1	1.1	1.4	1.4	1.4	…	False
2	706.0	94.0	13.0	67.0	…	False
3	20.0	2.0	1.0	2.0	…	False
…	…	…	…	…	…	…
10885	3442.0	470.0	113.0	385.0	…	True

Tabel 4. 3 Dataset KC1

No	loc	v(g)	ev(g)	iv(g)	….	defect
1	1.1	1.4	1.4	1.4	…	False
2	5.0	1.0	1.0	1.0	…	False
3	3.0	1.0	1.0	1.0	…	False

…	…	…	…	…	…	…
2109	205.0	26.0	16.0	19.0	…	True

Tabel 4. 4 Dataset KC2

No	loc	v(g)	ev(g)	iv(g)	….	defect
1	1.1	1.4	1.4	1.4	…	False
2	1.0	1.0	1.0	1.0	…	True
3	415.0	59.0	50.0	51.0	…	True
…	…	…	…	…	…	…
522	3..0	1.0	1.0	1.0	…	True

Tabel 4. 5 Dataset PC1

No	loc	v(g)	ev(g)	iv(g)	….	defect
1	1.1	1.4	1.4	1.4	…	False
2	10.0	1.0	1.0	1.0	…	False
3	29.0	6.0	1.0	3.0	…	False
…	…	…	…	…	…	…
1109	6.0	1.0	1.0	1.0	…	True

Berdasarkan data setiap *dataset* maka Tabel 4.6 dan tabel 4.7 menunjukkan informasi 5 *dataset* yang digunakan pada penelitian ini. Informasi yang ditunjukkan antara lain jumlah total modul setiap *dataset*, jumlah modul yang cacat dan jumlah modul yang dinyatakan tidak cacat serta informasi tentang fitur-fitur yang ada pada setiap *dataset*.

Tabel 4. 6 Dataset Penelitian

Dataset	Jumlah total	Jumlah modul cacat	Jumlah modul tidak cacat	Jumlah fitur
CM1	498	449	49	21
JM1	10885	8779	2106	21
KC1	2109	1783	326	21
KC2	522	415	107	21
PC1	1109	1032	77	21

Tabel 4. 7 Fitur setiap dataset

No	Fitur	*Dataset*				
		CM1	JM1	KC1	KC2	PC1
1	loc	√	√	√	√	√
2	v(g)	√	√	√	√	√
3	ev(g)	√	√	√	√	√
4	iv(g)	√	√	√	√	√
5	uniq_Op	√	√	√	√	√
6	uniq_Opnd	√	√	√	√	√
7	total_Op	√	√	√	√	√
8	total_Opnd	√	√	√	√	√
9	n	√	√	√	√	√
10	v	√	√	√	√	√
11	l	√	√	√	√	√
12	d	√	√	√	√	√
13	i	√	√	√	√	√
14	e	√	√	√	√	√
15	b	√	√	√	√	√
16	t	√	√	√	√	√
17	lOCode	√	√	√	√	√
18	lOComment	√	√	√	√	√
19	lOBlank	√	√	√	√	√
20	locCodeAndComment	√	√	√	√	√

21	branchCount	√	√	√	√	√

4.1.3. Tampilan Program

Berikut tampilan program yang udah di buat, yang terdiri dari proses diskritisasi dengan *Equal Width Binning*, proses seleksi fitur dengan *Chi Square*, dan proses klasifikasi dengan *Naïve Bayes* untuk perhitungan akurasinya. Dan kita dapat menyimpan akurasi hasil klasifikias denan men-klik tombol simpan, dan data akurasi akan di simpan kedalam file berformatkan *excel*. Program tersebut diimplementasikan dalam bahasa pemrograman java yang bisa dilihat pada gambar dibawah ini.

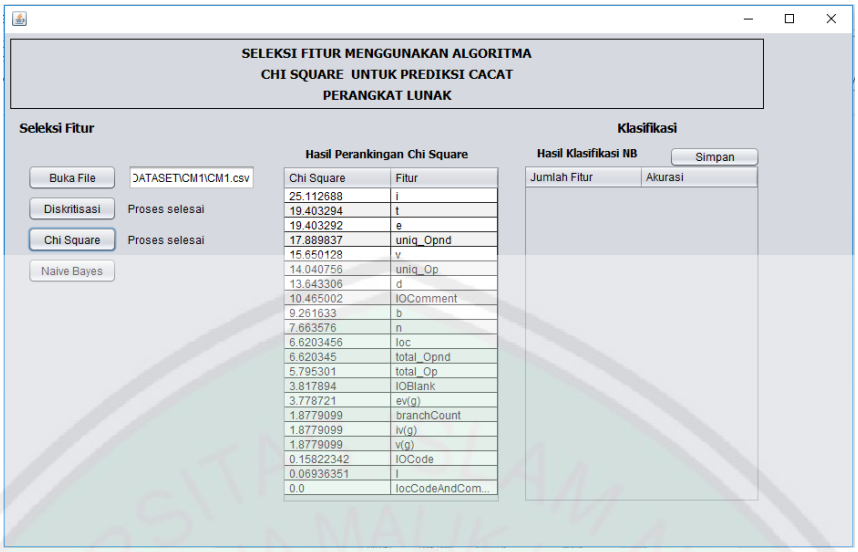
Gambar 4. 1 Tampilan Awal Program

Pada gambar 4.1 dapat dilihat bahwa terdapat beberapa tombol yang berguna untuk menjalankan sebuah proses sesuai dengan algoritma yang telah dijelaskan pada bab sebelumnya. Pertama-tama jalankan programnya terlebih dahulu agar sistem dapat berjalan. Jika program sudah dijalankan, maka kta buka file yang akan kita gunakan untuk proses klasifikasi dengan menklik tombol Buka File. Kemudian

pilih file di tampat anda menyimpan file tersebut. Jika sudah maka jalankan proses Diskritisasi data yang akan mengubah data yang sudah kita olih tadi menjadi data-data dalam bentuk data diskrit. Proses diskritisasi berhasil jika terdapat teks yang bertuliskan proses selesai di samping tombol tersebut. Berikut tampilan lebih jelasnya dapat dilihat pada gambar 4.2.

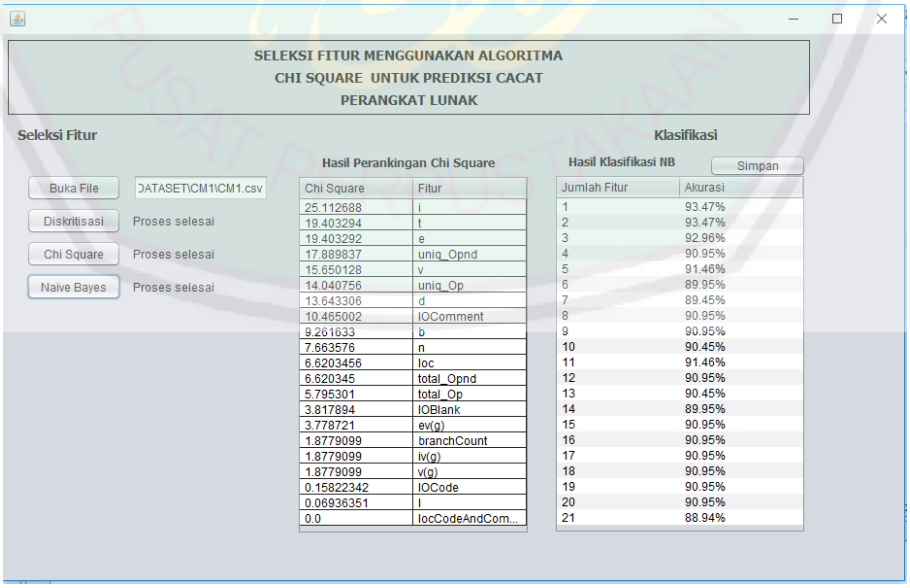
Gambar 4. 2 Proses Diskritisasi

Proses selanjutnya yaitu seleksi fitur dengan menggunakan *Chi Square.* Dengan menklik tombol *Chi Square,* maka proses seleksi fitur akan dijalankan oleh system, dimana yang dihasilkan pada proses ini adalah perangkingan fitur-fitur terbaik menurut algoritma Chi Square. Proses seleksi fitur selesai ditandai dengan teks yang bertuliskan proses selesai di samping tombol tersebut. Berikut tampilan programnya:

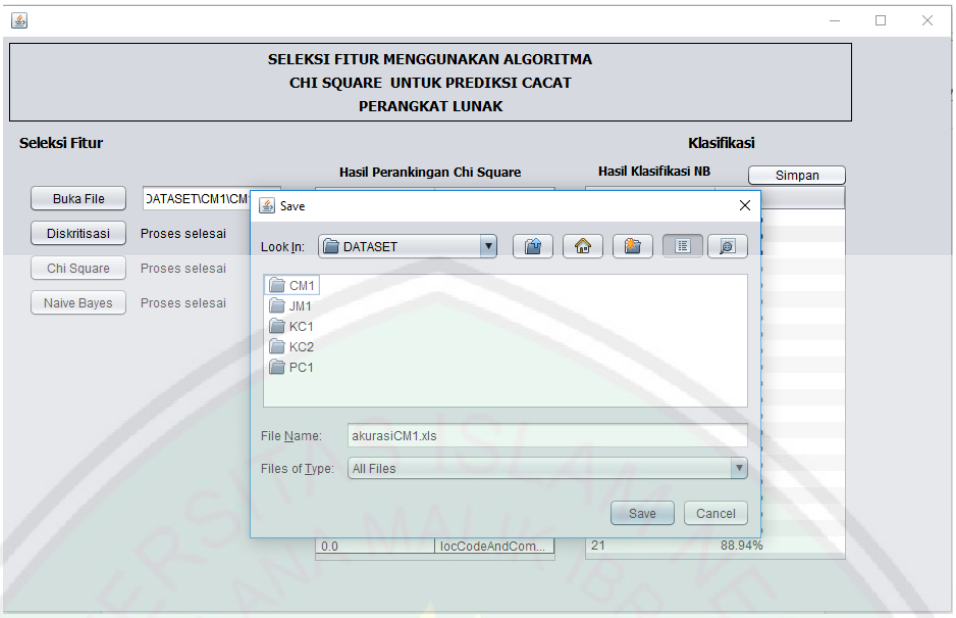


Gambar 4. 3 Tampilan Proses Seleksi Fitur Chi Square

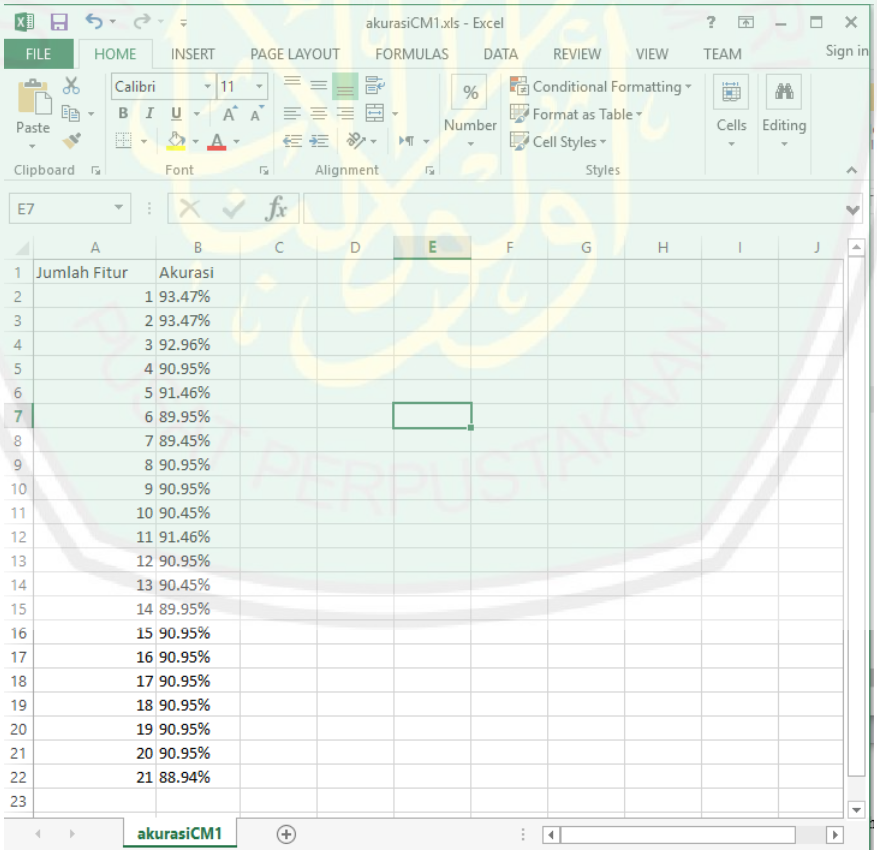
Proses selanjutnya adalah proses klasifikasi dengan menggunakan algoritma *Naïve Bayes* dimana akan dihasilkan akurasi pada setiap jumlah fitur dan pada setiap datasetnya. Dan kemudian kita dapat menyimpan file hasil dari klasifikasi tersebut dengan menggunakan tombol simpan yang berada di atas tabel hasil klasifikasi. Berikut tampilan Programnya:



Gambar 4. 4 Tampilan Proses Klasifikasi Naïve Bayes



Gambar 4. 5 Tampilan Proses Simpan



	A	B
1	Jumlah Fitur	Akurasi
2	1	93.47%
3	2	93.47%
4	3	92.96%
5	4	90.95%
6	5	91.46%
7	6	89.95%
8	7	89.45%
9	8	90.95%
10	9	90.95%
11	10	90.45%
12	11	91.46%
13	12	90.95%
14	13	90.45%
15	14	89.95%
16	15	90.95%
17	16	90.95%
18	17	90.95%
19	18	90.95%
20	19	90.95%
21	20	90.95%
22	21	88.94%
23		

Gambar 4. 6 Hasil Proses Simpan

4.1.4 Hasil Pengujian

Dalam penelitian ini, dilakukan pengujian terhadap 5 *dataset* yaitu *dataset* CM1, JM1, KC1, KC2, PC1. Pada tahap awal adalah menyeleksi setiap fitur dengan menggunakan algoritma *Chi Square* pada setiap *dataset* tersebut kemudian diurutkan mana fitur yang memiliki nilai paling tinggi dan rendah. Selanjutnya dilakukan proses klasifikasi menggukan Naïve Bayes untuk mengetahui tingkat akurasi dari *dataset* tersebut.

4.1.4.1. Seleksi Fitur menggunakan *Chi Square*

Berikut hasil seleksi fitur dengan menggunakan Chi Square, dimana setiap atribut diurutkan dari nilai tertinggi hingga paling rendah.

Tabel 4. 8 Hasil Seleksi Fitur Dataset CM1

Chi Square	Atribut
25.112686	i
19.403292	t
19.403292	e
17.889837	uniq Opnd
15.650128	v
14.040755	uniq Op
13.643305	d
10.465001	lOComment
9.261633	b
7.6635766	n
6.6203456	loc
6.620345	total Opnd
5.795301	total Op
3.817894	lOBlank
3.778721	ev(g)
1.8779099	branchCount

1.8779099	iv(g)
1.8779099	v(g)
0.15822342	lOCode
0.06936351	l
0.0	locCodeAndComment

Dapat dilihat pada tabel diatas, pada tahap uji coba seleksi fitur pada *dataset* CM1, fitur i (*Intelligence)* memiliki nilai tertinggi kemudian disusul dengan fitur t (*Time to write program),* e (*Effort to write program),* uniq Opnd (*Unique operands),*v (Volume), uniq Op (*Unique operators),* (d (*Difficulty),* lOComment (*Count of lines of comments),* b (*Error estimate),* n (Total operator + operand), loc (*Line of code),* total Opnd (Total operand), total Op (Total operator), lOBlank (*Count of blank lines),* ev(g) (*Essential complexity),* branchCount, iv(g) (*Design complexity),* v(g) (*Cyclomatic complexity),* lOCode (*Count of Statement Lines),* l (*Program length),* dan locCodeAndComment (*Count of Code and Comments Lines).*

Tabel 4. 9 Hasil Seleksi Fitur Dataset JM1

Chi Square	**Atribut**
214.64569	i
91.126625	l
61.47448	total_Opnd
54.03862	d
39.28376	v
37.718513	b
37.718513	loc
36.399395	v(g)
36.082554	ev(g)
33.78586	n
32.649338	total_Op

32.102173	branchCount
29.301697	uniq_Opnd
29.301693	lOCode
28.141212	t
28.141212	e
20.904879	uniq_Op
20.154142	iv(g)
12.791038	lOComment
6.582422	lOBlank
4.1710434	locCodeAndComment

Pada uji coba seleksi fitur untuk *dataset* JM1, fitur i (*Intelligence)* memiliki rangking atau nilai paling tinggi diantara fitur lainnya, kemudian fitur selanjutnya sebagai berikut l (*Program length)*, total_Opnd *(Total operand)*, d (*Difficulty)*, v (*Volume*), b (*Error estimate)*, loc (*Line of code)*, v(g) (*Cyclomatic complexity)*, ev(g) (*Essential complexity)*, n *(Total operator + operand)*, total_Op *(Total operator)*, branchCount, uniq_Opnd (*Unique operands)*, lOCode (*Count of Statement Lines)*, t (*Time to write program)*, e (*Effort to write program)*, uniq_Op (*Unique operators)*, iv(g) (*Design complexity)*, lOComment (*Count of lines of comments)*, lOBlank (*Count of blank lines)*, locCodeAndComment (*Count of Code and Comments Lines)*.

Tabel 4. 10 Hasil Seleksi Fitur Dataset KC1

Chi Square	**Atribut**
259.57108	d
150.37274	uniq_Op
148.3774	i
93.88463	l
89.57417	uniq_Opnd
81.3644	loc

60.301056	v
54.95382	lOComment
50.782734	total_Op
50.469402	lOCode
49.7902	b
42.84056	total_Opnd
42.042156	n
32.62228	t
32.62228	e
14.7508545	ev(g)
11.259989	branchCount
11.259989	v(g)
10.872311	lOBlank
4.0347247	iv(g)
0.007380225	locCodeAndComment

Hasil uji coba seleksi fitur pada *ataset* KC1 adalah sebagai berikut ; fitur d (*Difficulty)*, uniq_Op (*Unique operators)*, i (*Intelligence)*, l (*Program length)*, uniq_Opnd (*Unique operands)*, loc (*Line of code)*, v (*Volume*), lOComment (*Count of lines of comments)*, total_Op *(Total operator)*, lOCode (*Count of Statement Lines)*, b (*Error estimate)*, total_Opnd *(Total operand)*, n *(Total operator + operand)*, t (*Time to write program)*, e (*Effort to write program)*, ev(g) (*Essential complexity)*, branchCount, v(g) (*Cyclomatic complexity)*, lOBlank (*Count of blank lines)*, iv(g) (*Design complexity)*, locCodeAndComment (*Count of Code and Comments Lines).*

Tabel 4. 11 Hasil Seleksi Fitur dataset KC2

Chi Square	**Atribut**
38.420925	d
26.901974	i
25.026638	lOCodeAndComment

22.149889	uniq_Op
18.80953	lOComment
12.062008	lOCode
12.062007	total_Opnd
12.062007	uniq_Opnd
12.062007	t
12.062007	e
12.062007	n
12.062007	loc
7.944794	lOBlank
7.9447937	total_Op
7.9447937	b
7.9447937	v
3.9250174	branchCount
3.925017	iv(g)
3.925017	ev(g)
3.925017	v(g)
7.685324E-4	l

Hasil uji coba seleksi fitur pada *ataset* KC2 adalah sebagai berikut ; fitur d (*Difficulty),* i (*Intelligence),* locCodeAndComment (*Count of Code and Comments Lines),* (*Unique operators),* lOCode (Count of Statement Lines),total_Opnd (Total operand), uniq_Opnd (Unique operands), t (Time to write program), e (Effort to write program), n (Total operator + operand), loc (Line of code), lOBlank (Count of blank lines), total_Op (Total operator), b (Error estimate), v (Volume), branchCount,iv(g) (Design complexity), ev(g) (Essential complexity), v(g) (Cyclomatic complexity), l (*Program length).*

Tabel 4. 12 Hasil Seleksi Fitur dataset PC1

Chi Square	**Atribut**
50.156048	lOComment

30.442928	V
28.23413	branchCount
28.23413	v(g)
27.646458	lOCode
27.646458	T
27.646458	B
27.646458	E
27.646458	loc
27.573595	uniq_Opnd
20.291636	I
18.441465	total_Opnd
17.035946	total_Op
16.895039	lOBlank
16.768406	N
16.606588	iv(G)
11.518756	uniq_Op
8.494679	locCodeAndComment
5.748972	ev(g)
3.2424536	L
1.8475873	D

Hasil uji coba seleksi fitur pada d*ataset* PC1 adalah sebagai berikut ; fitur lOComment (*Count of lines of comments)*, V (*Volume*), branchCount, v(g) (*Cyclomatic complexity)*, lOCode (*Count of Statement Lines)*, T (*Time to write program)*, B (*Error estimate)*, E (*Effort to write program)*, loc (*Line of code)*, uniq_Opnd (*Unique operands)*, l, total_Opnd *(Total operand)*, total_Op *(Total operator)*, lOBlank (*Count of blank lines)*, N *(Total operator + operand)*, iv(G) (*Design complexity)*, uniq_Op (*Unique operators)*, locCodeAndComment (*Count of Code and Comments Lines)*, ev(g) (*Essential complexity)*, L (*Program length)*, D (*Difficulty)*.

4.1.4.2. Uji coba Klasifikasi

Pada bagian ini dilakukan pengujian hasil dari pemilihan fitur diatas dengan algoritma pengklasifikasi. Hasil dari proses pengujian ini berupa akurasi dari data *testing* dan algoritma yang digunakan untuk klasifiksi adalah algoritma klasifikasi *Naïve Bayes* dengan pembagian data menggunakan *Split Percentage*. Jumlah fitur-fitur akan di ujicoba yaitu dengan menggunakan 1 fitur, 2 fitur, 3 fitur, dan sampai 21 fitur.

Confusion matrix berupa tabel yang berisi perbandingan atau pencocokan antara data hasil prediksi dengan kenyataan dari data yang ada. .

Berikut hasil dari proses uji coba kelima *dataset* yaitu *dataset* CM1, JM1, KC1, KC2 dan PC1.

Tabel 4. 13 Hasil Klasifikasi Dataset CM1

Jumlah fitur	Akurasi klasifikasi data *testing* NB(%)
1	93.47%
2	93.47%
3	92.96%
4	90.95%
5	91.46%
6	89.95%
7	89.45%
8	90.95%
9	90.95%
10	90.45%
11	91.46%
12	90.95%
13	90.45%
14	89.95%

15	90.95%
16	90.95%
17	90.95%
18	90.95%
19	90.95%
20	90.95%
21	88.94%

Pada tabel 4. 13 dapat dilihat hasil klasifikasi untuk prediksi cacat perangkat lunak menggunakan *Naïve Bayes* pada *dataset* CM1 menunjukkan hasil akurasi terbesar didapatkan dengan mengklasifikasi sejumlah 1 dan 2 fitur dengan persentase akurasi 93.47%. Dan hasil rata-rata akurasi nya sebesar 91,02%.

Tabel 4. 14 Hasil Klasifikasi Dataset JM1

Jumlah fitur	**Akurasi klasifikasi data *testing* NB(%)**
1	93.68%
2	91.87%
3	91.80%
4	91.32%
5	92.35%
6	92.93%
7	93.02%
8	92.99%
9	93.04%
10	93.06%
11	93.09%
12	93.11%
13	92.93%
14	93.06%
15	93.59%
16	94.05%
17	94.01%

18	94.07%
19	94.10%
20	94.07%
21	80.09%

Pada tabel 4. 14 dapat dilihat hasil klasifikasi untuk prediksi cacat perangkat lunak menggunakan *Naïve Bayes* pada *dataset* JM1 menunjukkan hasil akurasi terbesar didapatkan dengan mengklasifikasi sejumlah 19 fitur dengan persentase akurasi 94.10%. Dan hasil rata-rata akurasi nya sebesar 92.48%.

Tabel 4. 15 Hasil Klasifikasi Dataset KC1

Jumlah fitur	**Akurasi klasifikasi data *testing* NB(%)**
1	92.06%
2	89.22%
3	86.61%
4	83.18%
5	83.41%
6	84.83%
7	86.37%
8	87.44%
9	87.56%
10	87.56%
11	87.68%
12	87.91%
13	88.15%
14	88.98%
15	89.57%
16	89.57%
17	89.81%
18	90.05%
19	90.05%
20	90.17%

21	81.52%

Pada tabel 4. 15 dapat dilihat hasil klasifikasi untuk prediksi cacat perangkat lunak menggunakan *Naïve Bayes* pada *dataset* KC1 menunjukkan hasil akurasi terbesar didapatkan dengan mengklasifikasi sejumlah 1 fitur dengan persentase akurasi 92.06%. Dan hasil rata-rata akurasi nya sebesar 87.7%.

Tabel 4. 16 Hasil Uji Coba Klasifikasi Dataset KC2

Jumlah fitur	**Akurasi klasifikasi data *testing* NB(%)**
1	96.65%
2	91.39%
3	92.82%
4	91.39%
5	92.34%
6	93.30%
7	93.78%
8	93.78%
9	95.22%
10	96.17%
11	95.69%
12	96.17%
13	96.17%
14	96.17%
15	95.69%
16	96.17%
17	96.17%
18	96.17%
19	96.65%
20	96.17%
21	83.73%

Pada tabel 4.16 dapat dilihat hasil klasifikasi untuk prediksi cacat perangkat lunak menggunakan *Naïve Bayes* pada *dataset* KC2 menunjukkan hasil akurasi terbesar didapatkan dengan mengklasifikasi sejumlah 1 fitur dan 19 fitur dengan persentase akurasi 96.65%. Dan hasil rata-rata akurasi nya sebesar 94.37%.

Tabel 4. 17 Hasil Klasifikasi Dataset PC1

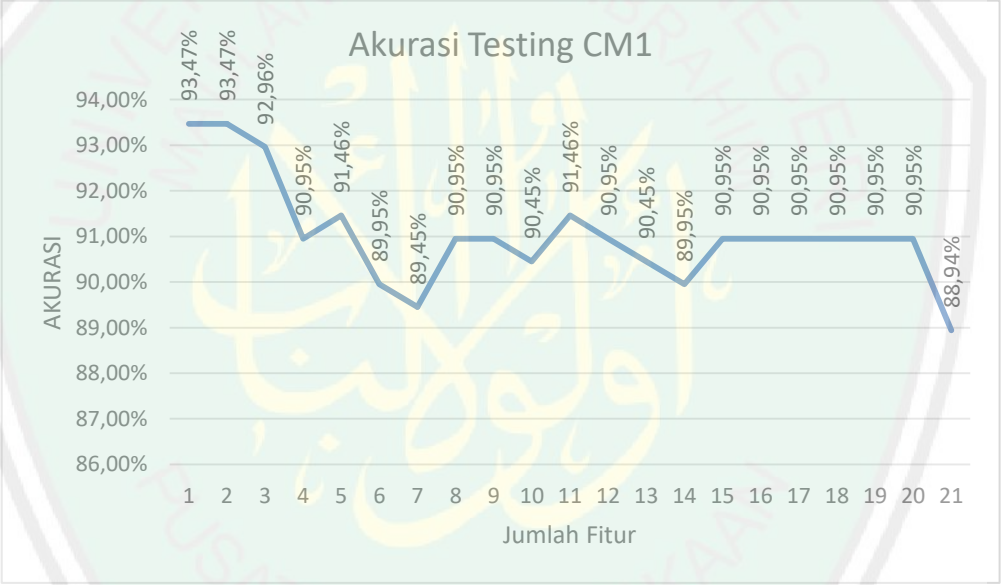
Jumlah fitur	Akurasi klasifikasi data *testing* NB(%)
1	91.22%
2	90.77%
3	91.44%
4	91.44%
5	90.54%
6	92.79%
7	92.12%
8	93.02%
9	92.79%
10	92.12%
11	91.67%
12	91.67%
13	91.22%
14	90.32%
15	90.54%
16	90.32%
17	90.09%
18	89.86%
19	90.09%
20	90.09%
21	88.74%

Pada tabel 4. 17 dapat dilihat hasil klasifikasi untuk prediksi cacat perangkat lunak menggunakan *Naïve Bayes* pada *dataset* PC1 menunjukkan hasil akurasi

terbesar didapatkan dengan mengklasifikasi sejumlah 8 fitur dengan persentase akurasi 93.02%. Dan hasil rata-rata akurasi nya sebesar 91.08%.

4.2 Pembahasan

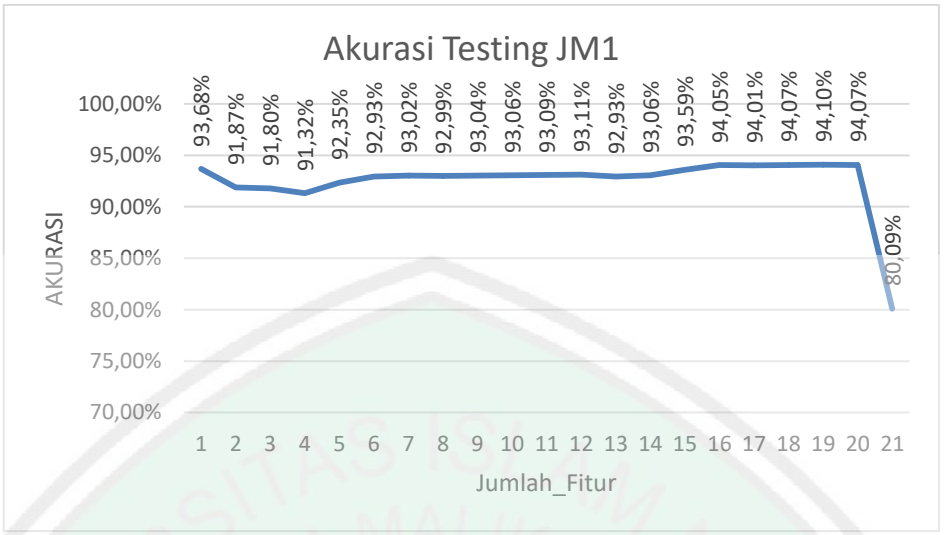
Pada sub bab ini akan dibahas mengenai hasil dari pengujian seleksi fitur menggunakan *Chi Square* dan klasifikasi mengunakan *Naïve Bayes* untuk prediksi cacat perangkat lunak yang telah dilakukan pada ke lima *dataset* yaitu CM1, JM1, KC1, KC2 dan PC1. yang pertama pembahasan pada pengujian *dataset* CM1.



Gambar 4. 7 Grafik akurasi data testing dataset CM1

Pada gambar 4.7 menunjukkan grafik hasil akurasi klasifikasi *Naïve Bayes* dengan seleksi fitur *Chi Square* didapatkan hasil rangking tertinggi pada jumlah 1 fitur dan 2 fitur pengujian dengan akurasi sebesar 93,47%.

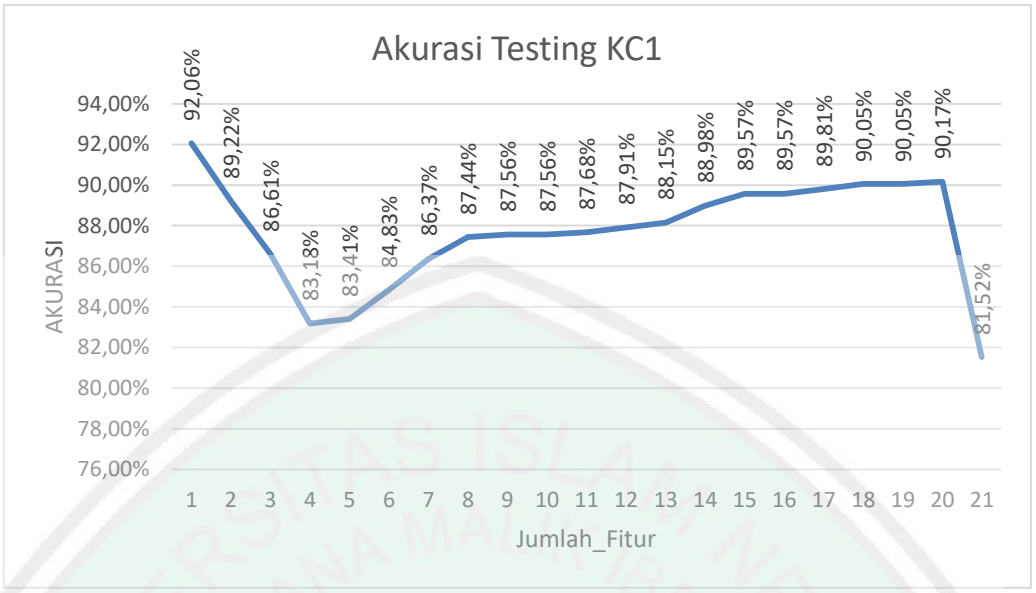
Uji coba selanjutnya dilakukan pada *dataset* JM1, dimana akan dilakukan pengujian setiap fitur sesuai hasil pemilihan fitur yang sudah dilakukan pada proses sebelumnya. Berikut hasil akurasi yang diperoleh pada *dataset* JM1:



Gambar 4. 8 Grafik akurasi data testing dataset JM1

Pada gambar 4.8 menunjukkan grafik hasil akurasi klasifikasi *Naïve Bayes* dengan seleksi fitur *Chi Square* didapatkan hasil rangking tertinggi pada jumlah 18 fitur dan 20 fitur pengujian dengan akurasi sebesar 94,07%.

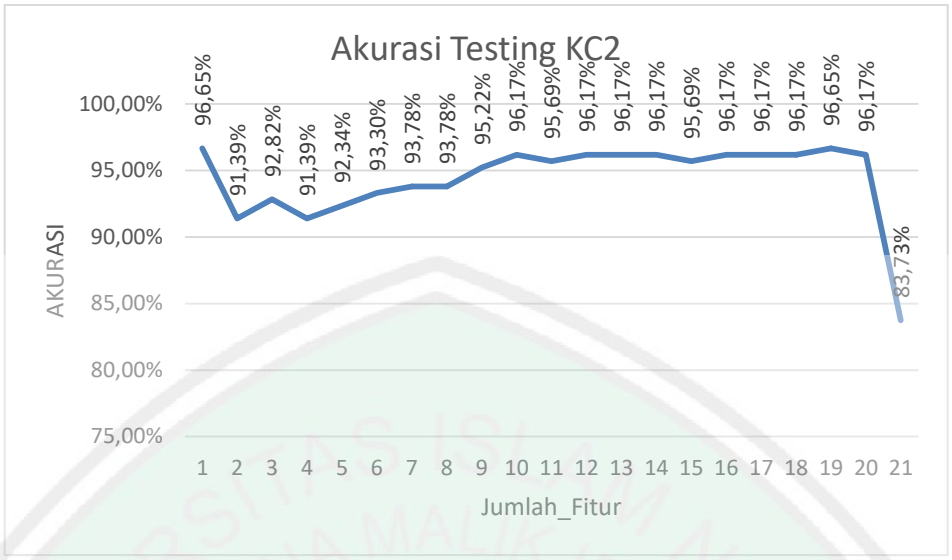
Uji coba selanjutnya dilakukan pada *dataset* KC1, dimana akan dilakukan pengujian setiap fitur sesuai hasil pemilihan fitur yang sudah dilakukan pada proses sebelumnya. Berikut hasil akurasi yang diperoleh pada *dataset* KC1:



Gambar 4. 9 Grafik akurasi data testing dataset KC1

Pada gambar 4.9 menunjukkan grafik hasil akurasi klasifikasi *Naïve Bayes* dengan seleksi fitur *Chi Square* didapatkan hasil rangking tertinggi pada jumlah 1 fitur pengujian dengan akurasi sebesar 92,06%.

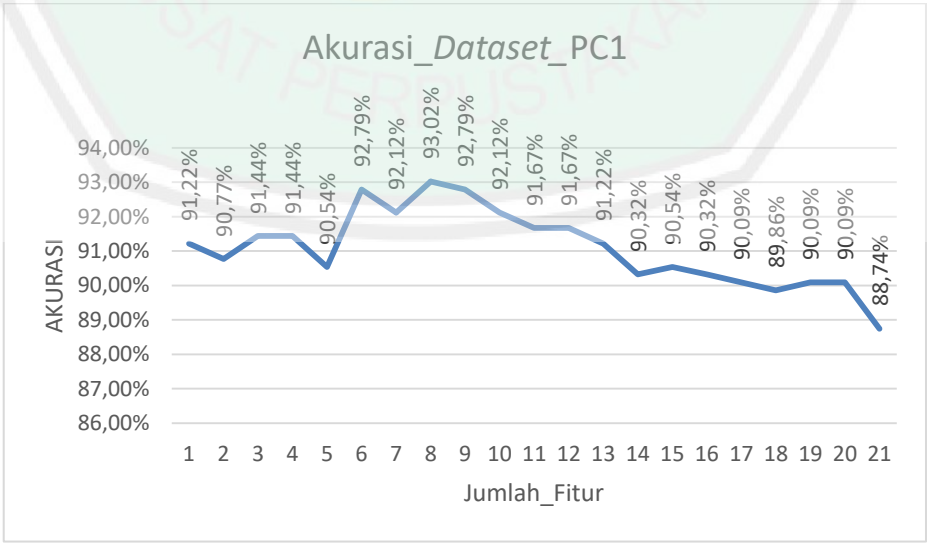
Uji coba selanjutnya dilakukan pada *dataset* KC2, dimana akan dilakukan pengujian setiap fitur sesuai hasil pemilihan fitur yang sudah dilakukan pada proses sebelumnya. Berikut hasil akurasi yang diperoleh pada *dataset* KC2:



Gambar 4. 10 Grafik akurasi data testing dataset KC2

Pada gambar 4.10 menunjukkan grafik hasil akurasi klasifikasi *Naïve Bayes* dengan seleksi fitur *Chi Square* didapatkan hasil rangking tertinggi pada jumlah 1 fitur dan 19 fitur pengujian dengan akurasi sebesar 96,65%.

Uji coba selanjutnya dilakukan pada *dataset* PC1, dimana akan dilakukan pengujian setiap fitur sesuai hasil pemilihan fitur yang sudah dilakukan pada proses sebelumnya. Berikut hasil akurasi yang diperoleh pada *dataset* PC1:



Gambar 4. 11 Grafik akurasi data testing dataset PC1

Pada gambar 4.6 menunjukkan grafik hasil akurasi klasifikasi *Naïve Bayes* dengan seleksi fitur *Chi Square* didapatkan hasil rangking tertinggi pada jumlah 1 fitur pengujian dengan akurasi sebesar 93,02%.

Setelah dilakukan pengujian menggukan seleksi fitur menggunakan Chi Square dan klasifikasi menggunakan Naïve Bayes, berikut jumlah fitur dan fitur yang menghasilkan akurasi tertinggi pada setiap *datasetnya*.

Tabel 4. 18 Daftar Dataset Terpilih

Dataset	Fitur	Akurasi Testing
CM1	➢ i (*Intelligence)* ➢ t (*Time to write program)*	93,47%
JM1	➢ i (*Intelligence)* ➢ l (*Program length)* ➢ total_Opnd *(Total operand*) ➢ d (*Difficulty)* ➢ v (Volume) ➢ b (*Error estimate)* ➢ loc (*Line of code)* ➢ v(g) (*Cyclomatic complexity)* ➢ ev(g) (*Essential complexity)* ➢ n (Total operator + operand) ➢ total_Op (Total operator) ➢ branchCount ➢ uniq_Opnd (*Unique operands)* ➢ lOCode (*Count of Statement Lines)* ➢ t (*Time to write program)*	94,10%

	➢ e (*Effort to write program)* ➢ uniq_Op (*Unique operators)* ➢ iv(g) (*Design complexity)* ➢ lOComment (*Count of lines of comments)*	
KC1	➢ d (*Difficulty)*	92,06%
KC2	➢ d (*Difficulty)*	96,65%
PC1	➢ lOComment (*Count of lines of comments)* ➢ v (Volume) ➢ branchCount ➢ v(g) (*Cyclomatic complexity)* ➢ lOCode lOCode (*Count of Statement Lines)* ➢ t (*Time to write program)* ➢ b (*Error estimate)* ➢ e (*Effort to write program)*	93,02%

4.3 Integrasi dengan Islam

Al Qur'an adalah kitab suci yang harus diyakini kebenarannya, itulah salah satu ciri orang yang beriman. Selain itu, Al Qur'an juga sumber pedoman kehidupan, banyak nilai-nilai yang terkandung di dalamnya. Seperti ayat dibawah ini, bahwasanya manusia adalah makhluk yang lemah dimana akan membutuhkan bantuan yang lainnya. Berikut menurut suah Ar Rum ayat 54;

اللَّهُ الَّذِي خَلَقَكُمْ مِنْ ضَعْفٍ ثُمَّ جَعَلَ مِنْ بَعْدِ ضَعْفٍ قُوَّةً ثُمَّ جَعَلَ مِنْ بَعْدِ قُوَّةٍ ضَعْفًا وَشَيْبَةً ۚ يَخْلُقُ مَا يَشَاءُ ۖ وَهُوَ الْعَلِيمُ الْقَدِيرُ

Artinya : *"Allah, Dialah yang menciptakan kamu dari keadaan lemah, kemudian Dia menjadikan (kamu) sesudah keadaan lemah itu menjadi kuat, kemudian Dia menjadikan (kamu) sesudah kuat itu lemah (kembali) dan beruban. Dia menciptakan apa yang dikehendaki-Nya dan Dialah Yang Maha Mengetahui lagi Maha Kuasa.".(Q.S Ar Rum : 54)*.

Pada potongan ayat tersebut Allah SWT memerintahkan untuk saling tolong-menolong dalam aktivitas kebaikan dan dilarang untuk tolong-menolong dalam perbuatan yang salah dan mengakibatkan dosa, potongan ayat tersebut dalam Tafsir Ibnu Katsir dijelaskan Allah SWT memerintahkan kepada hamba-hamba-Nya yang beriman untuk saling menolong dalam berbuat kebaikan yaitu kebajikan dan meninggalkan hal-hal yang mungkar; hal ini dinamakan ketakwaan. Allah SWT melarang mereka bantu-membantu dalam kebatilan serta tolong- menolong dalam perbuatan dosa dan hal- hal yang diharamkan. Ibnu Jarir mengatakan bahwa dosa itu ialah meninggalkan apa yang diperintahkan oleh Allah untuk dikerjakan. Pelanggaran itu artirnya melampaui apa yang digariskan oleh Allah dalam agama kalian, serta melupakan apa yang difardukan oleh Allah atas diri kalian dan atas diri orang lain. Penjelasan tersebut kemudian diperkuat dengan hadits yang diriwayatkan oleh Imam Bukhari secara munfarid melalui hadis Hasyim dengan sanad yang sama dan lafaz yang semisal. Keduanya mengetengahkan hadis ini melalui jalur Sabit, dari Anas yang menceritakan bahwa Rasulullah Saw. telah bersabda:

انْصُرْ أَخَاكَ ظَالِمًا أَوْ مَظْلُومًا". قِيلَ: يَا رَسُولَ اللَّهِ، هَذَا نَصَرْتُهُ مَظْلُومًا، فَكَيْفَ أَنْصُرُهُ ظَالِمًا؟ قَالَ:

"تَمْنَعُهُ مِنَ الظُّلْمِ، فَذَاكَ نَصْرُكَ إِيَّاهُ

Artinya : *"Tolonglah saudaramu, baik dia berbuat aniaya ataupun diani-aya." Ditanyakan, "Wahai Rasulullah, orang ini dapat aku tolong bila dalam keadaan teraniaya, tetapi bagaimana menolongnya jika dia berbuat aniaya?" Rasulullah Saw. menjawab, "Kamu cegah dia dari perbuatan aniaya, itulah cara kamu menolongnya.".*

Menurut tafsir Ibnu Katsir, dalam ayat ini Allah SWT mengingatkan manusia akan fase-fase yang telah dilaluinya dalam penciptaannya, dari suatu keadaan menuju keadaan yang lain. Asal mulanya manusia itu berasal dari tanah liat, kemudian dari air mani, kemudian menjadi 'alaqah, kemudian menjadi segumpal daging, kemudian menjadi tulang yang dilapisi dengan daging, lalu ditiupkan roh ke dalam tubuhnya.

Setelah itu ia dilahirkan dari perut ibunya dalam keadaan lemah, kecil, dan tidak berkekuatan. Kemudian menjadi besar sedikit demi sedikit hingga menjadi anak, setelah itu berusia balig dan masa puber, lalu menjadi pemuda. Inilah yang dimaksud dengan keadaan kuat sesudah lemah.

Kemudian mulailah berkurang dan menua, lalu menjadi manusia yang lanjut usia dan memasuki usia pikun; dan inilah yang dimaksud keadaan lemah sesudah kuat. Di fase ini seseorang mulai lemah keinginannya, gerak, dan kekuatannya; rambutnya putih beruban, sifat-sifat lahiriah dan batinnya berubah pula.

Dan terdapat pada sebuah hadits yang menunjukkan bahwa setiap manusia pasti pernah melakukan perbuatan dosa, berikut haditsnya;

يا عبادي إنكم تخطئون في الليل والنهار وأنا أغفر الذنوب جميعاً فاستغفروني أغفر لكم (صحيح مسلم 4674)

Artinya : *" Wahai hamba-hamba Ku, sesungguhnya kalian berbuat salah pada malam dan siang, dan Aku mengampuni semua dosa, maka minta mapunlah kepada-Ku niscaya Aku akan mengampuni kalian."* (Shahih Muslim : 4674)

Menurut hadits shahih Muslim diatas, dapat kita ketahui bahwa manusia tidak luput dari kesalahan disetiap harinya, baik kesalahan yang di sengaja maupun tidak di sengaja. Dosa bisa dilakukan antara sesama makhluknya maupun kepada sang pencipta. Mulai bangun tidur sampai mejelang tidur kembali, sangat penting untuk selalu meminta ampunan kepad Allah. Untuk itu, dengan memohon ampunan kepada Allah lah manusia bisa menghapus dosa-dosa yang telah dilakukannya.

Dengan demikian, dapat kit petik dari beberapa dalil Al – Qur'an dan hadits di atas bahwasanya manusia masih adalah makhluk yang lemah dan membutuhkan yang lainnya di kehidupan ini. Dan juga kesalahan-kesalahan manusia adalah hal yang maklum. Contohnya saja saat manusia sedang membuat sebuah program, yang dimana akan ada kecacatan atau *bug* dalam program yang dibuatnya.

BAB V

KESIMPULAN DAN SARAN

Pada bab penutup ini menjelaskan tentang kesimpulan dari penelitian ini serta memberi saran bagi pembaca untuk pengembangan penelitian.

5.1 Kesimpulan

Berdasarkan hasil uji coba yang telah dilakukan pada ke 5 *dataset* yaitu CM1, JM1, KC1, KC2, dan PC1 dapat disimpulkan bahwa pengklasifikasian dengan jumlah fitur yang berbeda menghasilkan nilai akurasi yang berbeda pula. Dan berdasarkan statistik dari hasil pengujian fitur yang memiliki pengaruh signifikan terhadap proses prediksi cacat perangkat lunak terhadap 5 dataset tersebut dari hasil proses untuk untuk prediksi cacat perangkat lunak untuk didapatkan dengan hasil *dataset* CM1 sebanyak 2 fitur , yaitu fitur i (*Intelligence)* dan fitur t (*Time to write program)* dengan akurasi sebesar 93,47%. Untuk *dataset* JM1 sebanyak 19 fitur yaitu *i (Intelligence), l (Program length), total Opnd (Total operand), d (Difficulty), v (Volume), b (Error estimate), loc (Line of code), v(g) (Cyclomatic complexity), ev(g) (Essential complexity), n (Total operator + operand), total Op (Total operator), branchCount, uniq Opnd (Unique operands), lOCode (Count of Statement Lines), t (Time to write program), e (Effort to write program), uniq Op (Unique operators), iv(g) (Design complexity), lOComment (Count of lines of comments)* dengan akurasi sebesar 94,10%. Untuk *dataset* KC1 adalah sebanyak 1 yaitu fitur *d (Difficulty)* dengan akurasi sebesar 92,06%. Untuk *dataset* KC2 sebanyak 1 dan 19 fitur , yaitu fitur *d (Difficulty)* dengan akurasi sebesar 96,65%. Dan untuk *dataset* PC1 sebanyak 8 fitur , yaitu fitur *lOComment (Count of lines of comments),v (Volume), branchCount,v(g) (Cyclomatic complexity), lOCode lOCode (Count of Statement Lines), t (Time to write program), b (Error estimate), e (Effort to write program)* dengan akurasi sebesar 93,02%.

5.2 Saran

Dalam penelitian ini, peneliti menyadari bahwa masih banyak kekurangan dan jauh dari kata sempurna, dengan demikian perlu adanya pengembangan untuk mendapatkan hasil yang lebih baik di kemudian hari, beberapa saran dari peneliti antara lain:

a. Menguji dengan menggunakan seleksi fitur yang berbeda.
b. Melakukan uji coba dengan *dataset* yang berbeda.
c. Menambahakn algoritma untuk mengatasi ketidakseimbangan kelas pada *dataset.*

DAFTAR PUSTAKA

Akbar , M. S. (2017). *Prediksi Cacat Perangkat Lunak dengan Optimasi Naive Bayes Menggunakan Gain Ratio.* Surabaya, Jawa Timur, Indonesia: Fakultas Teknologi Informasi. Institut teknologi sepuluh november .

Al-Sheikh, A. b. (1994). *Tafsir Ibnu Katsir Jilid 1.* Bogor: Pustaka Imam asy-Syafi'i.

Arar, O. F., & Ayan , K. (2015). Software defect prediction using cost-sensitive neural network. *Applied Soft Computing*.

Arar, O. F., & Ayan, K. (2017, Mei). A Feature Dependent Naive Bayes Approach and Its Application to the Software Defect Prediction Problem. *Soft Computing Journal*.

Bratu, C.V., Muresan, T., Potolea, R. 2008. *Improving Classification Accuracy through Feature Selection*. In : Proceding of the 4th IEEE Internaional Conference on Intelligent Computer Communication and Processing, pp. 25-32. IEEE Press, New York.

Czibula, G., Marian, Z. & Czibula, I.G. 2014. Software Defect Prediction Using Relational Association Rule Mining. Inf. Sci. (Ny)., vol. 264, pp. 260–278, April 2014.

Dougherty, J., Kohavi, R., Sahami, M. 1995. *Supervised and unsupervised discretization of continuous features*. In Proceedings of the 12th International Conference on Machine Learning, pages 194–202.

Beniwal , S., & Arora , J. (2012). Classification and Feature Selection Techniques in Data Mining. *International Journal of Engineering Research & Technology (IJERT)*, 1.

Boehm , B., & Basili , R. V. (2001). Software Defect Reduction Top 10 List .

Dougherty , J., Kohavi , R., & Sahami , M. (1995). Supervised and Unsupervised Discretization of Continuous Features.

G. Chandrashekar, F. Sahin, A survey on feature selection methods. Journal of Computers and Electrical Engineering, 2014, 40(1):16-28.

Han , J., Kamber , M., & Pei , J. (2012). *Data Mining Concepts and Techniques Third Edition.* United States of America , United States of America: Morgan Kaufmann .

I. H. I. Laradji, M. Alshayeb, and L. Ghouti, "Software defect prediction using ensemble learning on selected features," Inf. Softw. Technol., vol. 58, pp. 388–402, 2015

Karabulut, E. M., Özel, S. A., & İbrikçi, T. (2012). A comparative study on the effect of feature selection on classification accuracy. *Procedia Technology*.

Kumar, C. S., & Sree, R. R. (2014, OCTOBER). APPLICATION OF RANKING BASED ATTRIBUTE SELECTION FILTERS TO PERFORM AUTOMATED EVALUATION OF DESCRIPTIVE ANSWERS THROUGH SEQUENTIAL MINIMAL OPTIMIZATION MODELS. *Journal on Soft Computing* .

Kumaresh , S., R , B., & Sivaguru , M. (2014). Software Defect Classification using Bayesian Classification Techniques. *International Conference on Communication, Computing and Information Technology* .

Menzies , Greenwald , J., & Frank , A. (2007, Januari). Data Mining Static Code Attributes to Learn Defect Predictors. *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING* .

Menzies, T. (n.d.). *http://promise.site.uottawa.ca/SERepository/datasets-page.html*. Retrieved Maret Senin, 2018, from Promise Software Engineering Repository: http://promise.site.uottawa.ca/

Okumoto, K. 2011. Software Defect Prediction Based on Stability Test Data. Quality, Reliability, Risk, Maintenance, and Safety Engineering (ICQR2MSE), 2011 International Conference on , vol., no., pp.385,387, 17-19 June 2011.Pelayo , L., & Dick , S. (2007). Applying Novel Resampling Strategies To Software Defect Prediction. *Conference Fuzzy Information Processing Society*.

Prasetyo, E. (2012). *Data Mining - Konsep dan Aplikasi Menggunakan Matlab.* Yogyakarta: Andi.

Pressman , R. S. (2001). *Software Engineering A PRACTITIONER'S APPROACH.* (N. T. C. L. Liu, Ed.) New York San Francisco : McGraw-Hill.

Runeson , P., Andersson , C., Thelin , T., Andrews , A., & Berling , T. (2006). What Do We Know about Defect Detection Methods? *IEEE SOFTWARE* .

Santoso, I. B. (2013). *STATISTIKA UNTUK TEKNIK INFORMATIKA.* Malang, Jawa Timur, Indonesia: UIN-Maliki Press.

Singh , P., & Verma , S. (2009). An Investigation of the Effect of Discretization on Defect Prediction Using Static Measures. *2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies*.

Song , Q., Jia , Z., Shepperd , M., Ying , S., & Liu , J. (2011). A General Software Defect-Proneness Prediction Framework. *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*.

Subiyakto , A. (2008, April). Penggunaan Algoritma Klasifikasi dalam Data Mining.

Suyanto. (2017). *Data Mining Untuk Klasifikasi dan Klasterisasi Data.* Bandung: Informatika.

Trabelsi, M. 2017. *A New Feature Selection Method for Nominal Classifier based on Formal Concept Analysis*. Procedia Computer Science, 186–194.

Wahono , R. S. (2015). A Systematic Literature Review of Software Defect Prediction: Research Trends, *Datasets*, Methods and Frameworks. *Software Engineering* .

Wang, Huajing, Taghi M. K., Amri Napolitano. 2012. *Software measurement data reduction using ensemble techniques.* Neurocomputing, 124-132.

Wang , H., Khoshgoftaar , T. M., & Napolitano , A. (2012). Software measurement data reduction using ensemble techniques. *Neurocomputing*.

Yang, Y., Webb, G. I., & Wu, X. (2010). Discretization Methods. *Data Mining and Knowledge Discovery Handbook, 2nd ed.*

LAMPIRAN 1

Dataset CM1

No	Loc	V(g)	Ev(g)	Iv(g)	N	v	l	defect
1	1.1	1.4	1.4	1.4	1.3	1.3	1.3	FALSE
2	1.0	1.0	1.0	1.0	1.0	1.0	1.0	TRUE
3	24.0	5.0	1.0	3.0	63.0	309.13	0.11	FALSE
4	20.0	4.0	4.0	2.0	47.0	215.49	0.06	FALSE
5	24.0	6.0	6.0	2.0	72.0	346.13	0.06	FALSE
6	24.0	6.0	6.0	2.0	72.0	346.13	0.06	FALSE
7	7.0	1.0	1.0	1.0	11.0	34.87	0.5	FALSE
8	12.0	2.0	1.0	2.0	23.0	94.01	0.16	FALSE
9	25.0	5.0	5.0	5.0	107.0	548.83	0.07	FALSE
10	46.0	15.0	3.0	1.0	239.0	1362.41	0.04	FALSE
11	34.0	5.0	5.0	1.0	155.0	856.15	0.05	FALSE
12	10.0	2.0	1.0	1.0	35.0	143.06	0.11	FALSE
13	23.0	7.0	5.0	1.0	157.0	770.38	0.04	FALSE
14	23.0	7.0	5.0	1.0	105.0	474.97	0.04	FALSE
15	31.0	5.0	1.0	2.0	231.0	1303.73	0.04	FALSE
16	24.0	5.0	1.0	2.0	120.0	655.13	0.07	FALSE
17	13.0	2.0	1.0	1.0	57.0	271.03	0.11	FALSE
18	6.0	1.0	1.0	1.0	15.0	55.51	0.25	FALSE
19	33.0	2.0	1.0	2.0	135.0	745.68	0.09	FALSE
20	6.0	1.0	1.0	1.0	15.0	55.51	0.25	FALSE
21	7.0	2.0	1.0	1.0	27.0	105.49	0.08	FALSE
...	...	...	...	...	...	...	...	...
478	22.0	5.0	5.0	3.0	84.0	440.83	0.08	TRUE
479	59.0	7.0	1.0	7.0	349.0	2318.71	0.05	TRUE
480	19.0	1.0	1.0	1.0	118.0	561.08	0.08	TRUE
481	89.0	24.0	9.0	19.0	402.0	2603.24	0.02	TRUE
482	32.0	4.0	1.0	4.0	121.0	648.26	0.05	TRUE
483	74.0	11.0	11.0	9.0	354.0	2274.9	0.03	TRUE
484	32.0	4.0	1.0	4.0	117.0	574.11	0.04	TRUE
485	107.0	18.0	9.0	10.0	516.0	3366.16	0.02	TRUE
486	7.0	1.0	1.0	1.0	30.0	122.62	0.18	TRUE
487	19.0	3.0	1.0	1.0	85.0	439.44	0.1	TRUE
488	18.0	3.0	1.0	3.0	60.0	297.25	0.11	TRUE
489	22.0	3.0	1.0	3.0	81.0	439.53	0.09	TRUE
490	20.0	3.0	1.0	3.0	64.0	330.88	0.1	TRUE
491	8.0	1.0	1.0	1.0	24.0	103.73	0.2	TRUE
492	44.0	8.0	4.0	7.0	159.0	971.26	0.08	TRUE
493	91.0	18.0	7.0	16.0	591.0	4176.06	0.05	TRUE
494	47.0	3.0	1.0	3.0	256.0	1563.78	0.04	TRUE
495	24.0	4.0	3.0	3.0	107.0	587.63	0.05	TRUE
496	82.0	11.0	3.0	10.0	475.0	3155.83	0.02	TRUE
497	10.0	2.0	1.0	1.0	32.0	150.41	0.15	TRUE
498	28.0	6.0	5.0	5.0	104.0	564.33	0.06	TRUE

No	D	I	E	B	T	IOCode	IOComment	IOBlank	defect
1	1.3	1.3	1.3	1.3	1.3	2.0	2.0	2.0	FALSE
2	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	TRUE
3	9.5	32.54	2936.77	0.1	163.15	1.0	0.0	6.0	FALSE
4	16.0	13.47	3447.89	0.07	191.55	0.0	0.0	3.0	FALSE
5	17.33	19.97	5999.58	0.12	333.31	0.0	0.0	3.0	FALSE
6	17.33	19.97	5999.58	0.12	333.31	0.0	0.0	3.0	FALSE
7	2.0	17.43	69.74	0.01	3.87	0.0	0.0	1.0	FALSE
8	6.43	14.62	604.36	0.03	33.58	0.0	0.0	7.0	FALSE
9	14.25	38.51	7820.87	0.18	434.49	12.0	16.0	13.0	FALSE
10	22.3	61.1	30377.95	0.45	1687.66	8.0	35.0	22.0	FALSE
11	20.76	41.24	17773.08	0.29	987.39	11.0	28.0	16.0	FALSE
12	9.0	15.9	1287.55	0.05	71.53	2.0	4.0	4.0	FALSE
13	28.12	27.4	21659.58	0.26	1203.31	10.0	17.0	23.0	FALSE
14	27.22	17.45	12929.85	0.16	718.32	10.0	17.0	23.0	FALSE
15	27.5	47.41	35852.6	0.43	1991.81	2.0	15.0	40.0	FALSE
16	15.2	43.1	9958.0	0.22	553.22	3.0	20.0	23.0	FALSE
17	9.15	29.61	2480.95	0.09	137.83	6.0	5.0	8.0	FALSE
18	4.0	13.88	222.03	0.02	12.33	1.0	1.0	2.0	FALSE
19	11.65	64.01	8684.99	0.25	482.5	2.0	0.0	19.0	FALSE
20	4.0	13.88	222.03	0.02	12.33	1.0	1.0	0.0	FALSE
21	12.0	8.79	1265.83	0.04	70.32	0.0	0.0	1.0	FALSE
...	...	...	...	...	...	...	...	...	...
478	22.14	104.75	51324.87	0.77	2851.38	16.0	23.0	32.0	TRUE
479	12.0	46.76	6732.92	0.19	374.05	1.0	3.0	10.0	TRUE
480	47.6	54.69	123914.45	0.87	6884.14	4.0	62.0	30.0	TRUE
481	18.57	34.91	12037.08	0.22	668.73	3.0	24.0	13.0	TRUE
482	36.74	61.92	83581.43	0.76	4643.41	7.0	28.0	31.0	TRUE
483	22.75	25.24	13060.92	0.19	725.61	0.0	4.0	17.0	TRUE
484	64.92	51.85	218524.86	1.12	12140.27	9.0	73.0	51.0	TRUE
485	5.63	21.8	689.76	0.04	38.32	0.0	0.0	0.0	TRUE
486	9.64	45.57	4237.49	0.15	235.42	2.0	22.0	12.0	TRUE
487	9.47	31.39	2815.15	0.1	156.4	3.0	34.0	14.0	TRUE
488	11.08	39.66	4871.43	0.15	270.63	3.0	11.0	10.0	TRUE
489	10.5	31.51	3474.19	0.11	193.01	2.0	19.0	8.0	TRUE
490	5.0	20.75	518.63	0.03	28.81	0.0	7.0	7.0	TRUE

491	13.06	74.36	12685.78	0.32	704.77	2.0	31.0	10.0	TRUE
492	21.41	195.05	89412.4	1.39	4967.36	4.0	18.0	4.0	TRUE
493	28.0	55.85	43785.9	0.52	2432.55	2.0	13.0	2.0	TRUE
494	19.13	30.72	11241.58	0.2	624.53	1.0	7.0	4.0	TRUE
495	44.71	70.59	141084.24	1.05	7838.01	9.0	59.0	35.0	TRUE
496	6.5	23.14	977.69	0.05	54.32	1.0	12.0	4.0	TRUE
497	16.09	35.08	9078.38	0.19	504.35	2.0	7.0	0.0	TRUE
498	1.3	1.3	1.3	1.3	1.3	2.0	2.0	2.0	TRUE

No.	locCode AndComment	Uniq_ Op	Uniq_ Opnd	Total_ Op	Total_Opnd	Branch Count	D	defect
1	2.0	1.2	1.2	1.2	1.2	1.4	1.3	FALSE
2	1.0	1.0	1.0	1.0	1.0	1.0	1.0	TRUE
3	0.0	15.0	15.0	44.0	19.0	9.0	9.5	FALSE
4	0.0	16.0	8.0	31.0	16.0	7.0	16.0	FALSE
5	0.0	16.0	12.0	46.0	26.0	11.0	17.33	FALSE
6	0.0	16.0	12.0	46.0	26.0	11.0	17.33	FALSE
7	0.0	4.0	5.0	6.0	5.0	1.0	2.0	FALSE
8	0.0	10.0	7.0	14.0	9.0	3.0	6.43	FALSE
9	0.0	15.0	20.0	69.0	38.0	9.0	14.25	FALSE
10	0.0	15.0	37.0	129.0	110.0	29.0	22.3	FALSE
11	0.0	19.0	27.0	96.0	59.0	9.0	20.76	FALSE
12	0.0	9.0	8.0	19.0	16.0	3.0	9.0	FALSE
13	0.0	17.0	13.0	114.0	43.0	13.0	28.12	FALSE
14	0.0	14.0	9.0	70.0	35.0	13.0	27.22	FALSE
15	0.0	22.0	28.0	161.0	70.0	9.0	27.5	FALSE
16	0.0	19.0	25.0	80.0	40.0	9.0	15.2	FALSE
17	0.0	14.0	13.0	40.0	17.0	3.0	9.15	FALSE
18	0.0	8.0	5.0	10.0	5.0	1.0	4.0	FALSE
19	0.0	12.0	34.0	69.0	66.0	3.0	11.65	FALSE
20	0.0	8.0	5.0	10.0	5.0	1.0	4.0	FALSE
21	0.0	10.0	5.0	15.0	12.0	3.0	12.0	FALSE
...	...	...	...	...	...	...	...	...
478	0.0	17.0	21.0	53.0	31.0	9.0	22.14	TRUE
479	0.0	26.0	74.0	223.0	126.0	13.0	12.0	TRUE
480	0.0	8.0	19.0	61.0	57.0	1.0	47.6	TRUE
481	0.0	34.0	55.0	248.0	154.0	32.0	18.57	TRUE
482	0.0	19.0	22.0	78.0	43.0	7.0	36.74	TRUE
483	0.0	32.0	54.0	230.0	124.0	21.0	22.75	TRUE
484	0.0	14.0	16.0	65.0	52.0	5.0	64.92	TRUE
485	0.0	37.0	55.0	323.0	193.0	35.0	5.63	TRUE
486	0.0	9.0	8.0	20.0	10.0	1.0	9.64	TRUE
487	0.0	15.0	21.0	58.0	27.0	5.0	9.47	TRUE

488	0.0	14.0	17.0	37.0	23.0	5.0	11.08	TRUE
489	0.0	19.0	24.0	53.0	28.0	5.0	10.5	TRUE
490	0.0	18.0	18.0	43.0	21.0	5.0	5.0	TRUE
491	0.0	10.0	10.0	14.0	10.0	1.0	1.3	TRUE
492	0.0	20.0	49.0	95.0	64.0	15.0	1.3	TRUE
493	0.0	22.0	112.0	373.0	218.0	35.0	1.0	TRUE
494	0.0	23.0	46.0	144.0	112.0	5.0	9.5	TRUE
495	0.0	22.0	23.0	67.0	40.0	7.0	16.0	TRUE
496	0.0	32.0	68.0	285.0	190.0	21.0	17.33	TRUE
497	0.0	13.0	13.0	19.0	13.0	3.0	17.33	TRUE
498	0.0	20.0	23.0	67.0	37.0	11.0	2.0	TRUE