

PROSIDING SEMINAR NASIONAL

**Peran Ekonomi Kreatif dalam
Pembangunan Nasional yang Berkelanjutan**

Gresik, 29 - 30 Juni 2013



Diselenggarakan oleh
UNIVERSITAS MUHAMMADIYAH GRESIK
Bekerjasama dengan
ASOSIASI DOSEN INDONESIA (ADI) CABANG GRESIK

PROSIDING SEMINAR NASIONAL

Peran Ekonomi Kreatif dalam
Pembangunan Nasional yang Berkelanjutan

Gresik, 29 - 30 Juni 2013

EDITOR

Prof. Dr. Setyo Budi, MS.

Dr. Khoirul Anwar

Dr. Yudi Arifani

Dr. Nur Laily

Ir. Endah Sri Rejeki, MP. Phd.

PELAKSANA TEKNIS

Dian Kurnia Oktaviani

Universitas Muhammadiyah Gresik

Jl. Sumatra 101 GKB Gresik 61118

Telp. (031) 3951414 psw. 20

Fax. (031) 3952585

PROSIDING SEMINAR NASIONAL

DAFTAR ISI

	Halaman
1. Pengembangan Potensi Wisata Sebagai Katalisator Peningkatan Ekonomi Kreatif di Kabupaten Gresik Moh. Agung Surianto, Umaimah	1 - 10
2. Kreatif Interpretasi Brand Ritel Pada Usaha Kecil Menengah di Indonesia Apri Imanur Romadhon, SE, Abdurrahman Faris I.H., SE	11 - 23
3. Pengembangan dan Peningkatan Kinerja Industri Kreatif Melalui <i>Comprehensive Intellectual Capital Management</i> Sigit Hermawan	24 - 39
4. Model <i>Supply Chain Excellence</i> pada Usaha Mikro, Kecil, dan Menengah (umkm) sebagai Pelaku Industri Kreatif dan Pilar Ekonomi Bangsa Elly Ismiyah	40 - 50
5. Pengembangan Pariwisata Berbasis Budaya Lokal-budaya Religi sebagai Upaya Pendukung Peningkatan Industri Pariwisata Daerah Gresik (sebuah Konsep Pengembangan Budaya Lokal-budaya Religi dan Prospek Industri Pariwisata di Kabupaten Gresik) Hamim Farhan, R. Nazriyah	51 - 56
6. Analisa Metode Bayesian Network dan Multivariate Adaptive Regression Splines (mars) Untuk Memprediksi Object-oriented Software Maintainability Harunur Rosyid	57 - 78
7. Daya Dukung Lingkungan Perairan Berbasis <i>Blue Economy</i> Terhadap Produksi Budidaya Tambak Udang Vanname di Wilayah Pesisir Probolinggo Ichsan, Mohammad Mahmudi, Nuddin Harahap	79 - 92
8. Penguatan <i>Educational Community Based Development</i> sebagai Daya Dukung <i>Business Sustainability</i> Khoirul Anwar	93 - 98
9. The Analysis of Locutionary Acts in <i>Pinocchio</i> The Walt Disney's Picture Story Book For The Children Laila Rochmawati	99 - 104

ANALISA METODE BAYESIAN NETWORK DAN MULTIVARIATE ADAPTIVE REGRESSION SPLINES (MARS) UNTUK MEMPREDIKSI OBJECT-ORIENTED SOFTWARE MAINTAINABILITY

HARUNUR ROSYID

Teknik Informatika, Universitas Muhammadiyah Gresik

Jl. Sumatra 101 GKB-Gresik

Email : harun.umgresik@gmail.com

ABSTRACT:

Object-oriented metrics that have been proposed needs to be validated, One possible way to validate metrics is to do a statistical analysis of the size metrics and system maintainability. The problem is the number of metrics that are used to predict the maintainability of OO (Object Oriented) as long as it was not known metric that has the best prediction accuracy, so the analysis is needed to find a better metric. Evaluation model, is used to introduce the criteria for evaluating the accuracy of model predictions. Bayesian network is used as maintainability prediction model, one that is connected to the node CHANGE ten metric variables, Bayesian networks models that predict its output estimates in this paper, uses the interval estimation point. This is because the interval estimate is difficult to compare with other models suitable for the sample test used to test and compare the MARS models and other models. This paper evaluates and compares the OO software maintainability prediction models quantitatively, by using the prediction accuracy with a step size measures: absolute residuals (Ab.Res.), magnitude of relative error (MRE) and Pred measures. Analysis results can be concluded that the matrix model of MARS can effectively predict the maintainability of OO software systems. This is better than the matrix model of Bayesian network as the absolute value of residuals is significantly higher.

Key Words: *metric, Object Oriented, maintainability, Bayesian network, MARS. absolut residual. admistration*

PENDAHALUAN

Pemrograman berorientasi obyek merupakan fokus lain dari komunitas rekayasa perangkat lunak. Paradigma ini mengklaim pembangunan yang lebih cepat kecepatan dan kualitas perangkat lunak yang lebih tinggi dari paradigma prosedural. Akan tetapi, penggunaan metrik dalam paradigma berorientasi obyek masih harus dipelajari. Sejauh ini, metrik berorientasi objek telah diajukan dan diusulkan perlu divalidasi.. Satu kemungkinan cara untuk memvalidasi metrik adalah melakukan analisis statistik dari ukuran metrik dan sistem *maintainability*. Di antara semua faktor yang

memiliki potensi *maintainability* yang berpengaruh pada perangkat lunak adalah metrik perangkat lunak, terutama pengukuran system metrik dan *maintainability* dalam paradigma berorientasi objek tidak jelas. Selain itu, sangat sedikit yang diketahui tentang bagaimana dan dimana berbagai kegiatan pemeliharaan dilakukan dalam paradigma berorientasi objek.

Ada beberapa OO (*Object Oriented*) metrik yang digunakan untuk memprediksi *maintainability* dari OO sistem. Contoh metrik OO Chidamber dan Kemerer (K & K) metrik dan Li dan Henry (L & H). Itu menunjukkan bahwa L & H metrik memiliki korelasi engan jumlah

perubahan yang dibuat untuk kode sistem perangkat lunak OO. Itu juga menunjukkan bahwa beberapa model regresi linier yang terdiri dari C & K, L & H dan lainnya metrik OO mampu meramalkan perangkat lunak upaya pemeliharaan untuk beberapa sistem OO [1]. Dalam prakteknya, apakah suatu teknik pemodelan yang cocok untuk prediksi *maintainability* tidak hanya tergantung pada kemampuannya dalam menangkap hubungan sebab-akibat tetapi juga tergantung pada kemudahan membangun model prediksi. Permasalahannya adalah banyaknya metrik yang digunakan untuk memprediksi *maintainability* dari OO (*Object Oriented*) selama ini, belum diketahui metrik yang mempunyai akurasi prediksi yang terbaik, sehingga perlu dilakukan analisis untuk mencari metrik yang lebih baik.

Model prediksi dengan menggunakan teknik *Bayesian Network*. Teknik ini memperbolehkan user untuk membangun sebuah model prediksi yang didasarkan pada teori probabilitas Bayesian [2]. Namun, *Bayesian Network* juga bisa menjadi teknik baru yang menjanjikan untuk memprediksi *object-oriented software maintainability*. Ini karena kemampuan untuk mewakili ketidakpastian secara eksplisit menggunakan probabilitas, kemampuan untuk memasukkan keahlian manusia yang memiliki pengetahuan ke dalam data empiris, dan kemampuan untuk memperbarui model ketika informasi baru telah tersedia. Pada model *Bayesian Network* akurasi prediksi dievaluasi menggunakan *accuracy measures*, yang umumnya ditemukan dalam literature *software effort prediction*, ukuran itu digunakan untuk membetulkan, *magnitude of*

relative error (MRE) dan *prediction measures* [2].

Penerapan untuk memprediksi *software maintainability* yang dengan menggunakan teknik analisis multivarian, MARS (*Multivariate Adaptive Regression Splines*). MARS berbeda dari teknik yang lainnya, bahwa pendekatan dalam membangun model sebelumnya tidak memerlukan spesifikasi fungsional. Sebaliknya, MARS mencoba untuk beradaptasi dengan fungsional yang tidak diketahui bentuknya dengan menggunakan *regresi splines* [3]. Pengukuran pada metode ini, dilakukan dengan menjumlahkan perubahan yang dibuat pada *source code* selama periode *maintenance*. Model MARS memprediksinya dengan menggunakan *leave-one-out cross-validation* dan didukung dengan model vektor regresi.

Makalah ini menjelaskan dalam membangun *software maintainability* pada *object-oriented* dari sistem perangkat lunak dapat diukur dengan menggunakan cara yang berbeda yaitu dengan menggunakan model *Bayesian Network* dan MARS. Analisa ini dimaksudkan untuk menunjukkan adanya perbedaan akurasi yang signifikan dari setiap model prediksi *software maintainability*.

SOFTWARE MAINTENANCE

Studi ini dilakukan untuk mengumpulkan informasi untuk mendukung program pemahaman model atau untuk memvalidasi mereka, dan berkisar dari pengamatan (perilaku) studi eksperimen terkontrol. Beberapa model (misalnya, Pennington model) yang berdasarkan

pengalaman program yang sangat kecil atau bagian dari sebuah program dengan 200 *lines of code* (LOC) atau kurang.

Corritore dan Wiedenbeck menggambarkan studi empiris yang menyelidiki penggambaran mental dari pengelola ketika bekerja pada perangkat lunak yang dikembangkan oleh ahli prosedural dan berorientasi objek programmer. Dalam studi, program yang dikelola terdiri dari sekitar 800 *lines of code* (LOC), dan karena itu dianggap cukup besar. Studi empiris juga telah dilakukan pada skala besar sebenarnya sistem yang berisi lebih dari 40 *lines of code* (LOC) [4].

A. Metrik Perangkat Lunak

Metrik perangkat lunak mengukur aspek-aspek tertentu dari perangkat lunak. Metrik perangkat lunak secara umum dapat dibagi menjadi dua kategori; metrik produk perangkat lunak dan perangkat lunak metrik proses. Ukuran metrik produk perangkat lunak seperti kode sumber atau desain dokumen. perangkat lunak metrik proses merupakan proses pengembangan perangkat lunak, seperti jumlah jam kerja (*man-hours*) yang dibebankan pada kegiatan pembangunan dalam desain dan fase pengkodean. Metrik ini berfokus pada produk perangkat lunak.

Metric perangkat lunak pada paradigma Object-Oriented.

Memahami paradigma berorientasi obyek adalah langkah pertama untuk mendefinisikan metrik paradigma. Studi tentang paradigma berorientasi objek merupakan hasil konsep berorientasi objek seperti object, class,

attributes, inheritance, method, and message passing. Perilaku pemrograman yang ditunjukkan dalam paradigma berorientasi obyek berbeda dari prosedural paradigma. Sebagai contoh, penciptaan class dalam paradigma berorientasi obyek adalah perilaku pemrograman dibedakan dari penciptaan procedure atau functions dalam paradigma prosedural.

Langkah kedua dalam mendefinisikan metrik pada paradigma berorientasi objek adalah membangun landasan teoretis untuk metrik [1]. Usulan metrik berorientasi obyek tidak begitu banyak seperti yang terdapat dalam paradigma prosedural.

Karakteristik Dataset.

Perangkat lunak OO dataset menggunakan metrik yang digunakan oleh Li dan Henry. Yang terdiri dari lima metrik Chidamber and Kemerer (C&K): DIT, NOC, RFC, LCOM dan WMC, dan empat metrik Li and Henry (L&H): MPC, DAC, NOM dan SIZE2, serta SIZE1, yang merupakan metrik tradisional [1].

Data metrik dikumpulkan dari jumlah total 110 kelas dalam dua perangkat lunak OO sistem: *User Interface Management System* (UIMS) dan *Quality Evaluation System* (QUES). kode ditulis dalam *Classical-Ada*, UIMS dan QUES dataset masing-masing berisi 39 kelas dan 71 kelas. *Maintainability* diukur dengan metrik CHANGE dengan menghitung jumlah baris dalam kode, yang berubah selama 3-tahun periode pemeliharaan (*maintenance*). Baik UIMS maupun QUES dataset berisi data aktual sebagai upaya pemeliharaan. Deskripsi setiap metrik diberikan dalam Tabel 1 berikut

TABEL 1.
METRIK SOFTWARE

Name	Description
DIT	Depth of the inheritance tree (=inheritance level number of the class, 0 for the root class)
NOC	Number of children (= number of direct sub-classes that the class has)
MPC	Message-passing couple (=number of send statements defined in the class)
RFC	Response for a class (=total number of local methods and the number of methods, i.e. number of set of local methods that do not interact with each other, in the class)
LCOM	lack of cohesion of methods (=number of abstract data types defined in the class)
DAC	Data abstraction coupling (=number of abstract data types defined in the class)
WMC	Weighted method per class (=sum of McCabe's cyclomatic complexity of all local methods in the class)
NOM	Number of methods (=number of local methods in the class)
SIZE 1	Lines of cod (=number of semicolons in the class)
SIZE 2	Number of properties (=total number of attributes and the number of local methods in the class)
CHANGE	Number of lines changed in the class (insertion and deletion are independently counted as 1, change of the contents is counted as 2)

Korelasi antara CHANGE dan masing-masing metrik OO diperlihatkan dalam Tabel 2. Tabel 2 menunjukkan bahwa ada korelasi yang signifikan antara CHANGE dan metrik OO [2]. Namun, Tabel 2 juga menunjukkan bahwa korelasi di Dataset adalah UIMS berbeda dari

korelasi di QUES Dataset. Dengan demikian, makalah ini menganggap UIMS dan sebagai QUES dataset yang heterogen dan membangun model *Maintainability* secara terpisah untuk prediksi masing-masing Dataset

TABLE2.
KORELASI ANTRA CHANGE DAN
METRIK OO

	Pearson's correlation coefficient	
	UIMS dataset	QUES dataset
DIT	- 0.433**	- 0.090
NOC	0.559**	NA
MPC	0.454**	0.461**
RFC	0.643**	0.388**
LCOM	0.568**	0.050
DAC	0.629**	0.083
WMC	0.646**	0.425**
NOM	0.635**	0.142
SIZE 1	0.626**	0.635**
SIZE 2	0.666**	0.149

** Menunjukkan korelasi yang signifikan pada tingkat 0,01.

B. Bayesian Network

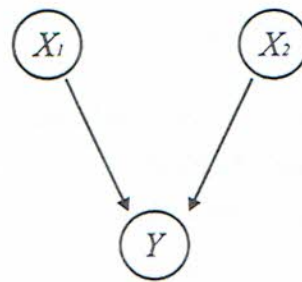
Bayesian Network (juga dikenal sebagai Bayes, kausal jaringan probabilistik, *Bayesian Network* kepercayaan, atau hanya kepercayaan jaringan) adalah *directed acyclic* graph (DAG) yang merepresentasikan *node* dalam sebuah domain. Node dalam sebuah domain ini terhubung dengan *directed links*, yang representasi dari asosiasi atau *causal relationship* antara *node* dalam sebuah domain [2]. Bila *link* representasi dari asosiasi, *directed* didefinisikan menurut dengan urutan waktu dimana peristiwa terjadi, yaitu, *link* dimulai dari event sebelumnya. Bila *link* representasi *causal relationship*, *relationship* sebab-akibat mulai

dari event. Gambar. 1 memperlihatkan sebuah contoh *Bayesian Network* yang terdiri dari tiga event, X_1 , X_2 , dan Y . Dalam Gambar. 1 [2], suatu event ditampilkan sebagai sebuah *elips* dan *directed link* dengan panah berarah. contoh *Bayesian Network* Ini menunjukkan bahwa X_1 dan X_2 memiliki asosiasi atau causal relationship dengan Y , yaitu hasil dari event X_1 dan X_2 mempunyai pengaruh pada hasil dari event Y .

Dalam sebuah *Bayesian Network*, hubungan antara event didefinisikan sebagai probabilitas bersyarat, $P(Y|X)$, yang merupakan probabilitas dari kejadian Y bersyarat pada hasil tertentu event X . probabilitas bersyarat dihitung dengan menggunakan Bayes' Theorem [2]:

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)} \quad (1)$$

di mana $P(X|Y)$ adalah probabilitas bersyarat dari event X ke event Y , dan $P(X)$ dan $P(Y)$ adalah probabilitas dari event X dan Y . Dari sudut pandang ini, Bayesian network dapat dianggap sebagai suatu event jaringan yang dihubungkan oleh ketergantungan probabilistik event X dan Y . Probabilistik dependensi maintained oleh *conditional probability table* (CPT), yang melekat ke event yang sesuai. CPT menunjukkan semua kemungkinan hasil dari event dan probabilitas bersyarat correspondensi dengan setiap outcome diberi outcome suatu associated atau causal event. Jika suatu event tidak associated atau causal event, event ini diberikan *unconditional (marginal)* distribusi probabilitas seperti ini bukannya CPT, baik CPT atau unconditional probability distribution menentukan untuk semua event, jaringan dapat dihitung dengan distribusi probabilitas $P(q, x)$ melalui jaringan, yang didefinisikan sebagai [2]:



Gambar1. Contoh Bayesian Network

$$P(\theta, x) = P(x|\theta)P(\theta) \quad (2)$$

Dalam Persamaan.2, q menunjukkan sebuah vektor yang komponen parameter menggambarkan distribusi probabilitas atas jaringan, dan x menunjukkan potongan data. Dalam Persamaan. 2 ini, $P(q)$ adalah disebut distribusi probabilitas sebelumnya dari jaringan dan $P(x|q)$ disebut kemungkinan data, Di sisi lain, Bayes teorema 1 ditunjukkan [2]:

$$P(\theta|x) = \frac{P(x|\theta)P(\theta)}{P(x)} \quad (3)$$

Dalam Persamaan. 3, $P(x|\theta)$ disebut *probabilitas posterior* distribusi jaringan. Menimbang bahwa $P(x)$ adalah konstan untuk suatu x , Persamaan 2 dan 3 menunjukkan bahwa:

$$\text{distributi posterior } \propto \text{kemungkinan } X \text{ prior distribusi} \quad (4)$$

serta

$$\text{distributi posterior } \propto \text{joint probabilitas distribusi} \quad (5)$$

Persamaan 4 di atas memberikan prinsip Bayesian updating, di mana *prior* probabilitas distribusi diperbarui menggunakan data yang diamati kemungkinan. Itu adalah, sebuah Bayesian Network dapat *recalibrated* setiap kali baru informasi menjadi tersedia. Efek informasi baru menyebar baik ke depan dan ke belakang pada jaringan dan itu memungkinkan jaringan

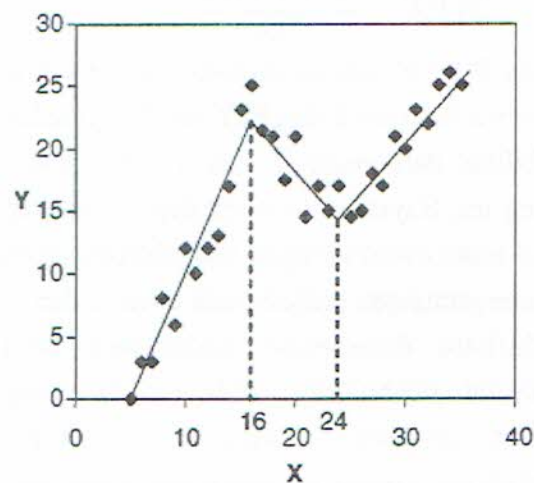
untuk memperbarui *joint probability* probabilitas.

Selain itu, *prior* probabilitas distribusi suatu Bayesian Network dapat ditetapkan subyektif, yaitu ditentukan berdasarkan pada awalnya hanya pengetahuan seorang pakar. Kemudian, jaringan diperbarui menggunakan data empiris. Hal ini memungkinkan pengetahuan seorang pakar untuk dimasukkan ke dalam data empiris. Atau, bila tidak ada pengetahuan seorang pakar untuk menentukan *prior* probabilitas distribusi, jaringan dapat melakukan *batch learning*. Dalam *batch learning* struktur jaringan dan semua CPTs dan unconditional distribusi probabilitas belajar dari data. Persamaan 5 adalah prinsip Bayesian inference, dimana perhitungan *joint probability* distribusi dilakukan dalam rangka untuk meringkas *posterior* probabilitas distribusi. Pada Bayesian Network, perhitungan dari *joint probability* distribusi disederhanakan menggunakan rantai aturan. Aturan rantai memungkinkan *joint probability* distribusi akan dihitung sebagai produk dari CPTs dan unconditional probabilitas distribusi melalui jaringan. Bayesian Network adalah sebuah model prediksi, hasil *posterior* distribusi probabilitas menyediakan distribusi probabilitas prediksi dari variabel, yaitu, daftar dari semua kemungkinan hasil dari variabel dan probabilitas terkait. Dengan kata lain, suatu model Bayesian Network memberikan interval estimasi prediksi variabel dengan terkait probabilitas. Ketika sebuah estimasi titik lebih disukai, yang hasil dengan nilai probabilitas tertinggi akan dipilih [2].

C. Multivariate adaptive regression splines (MARS).

MARS adalah regresi non-parametric merupakan teknik model *complex relationships* yang sulit, bukan tidak mungkin, untuk metode model lainnya mengungkapkan. Dalam arti, MARS didasarkan pada strategi *divide-and-conquer*, partisi set data pelatihan menjadi beberapa daerah, masing-masing mendapatkan persamaan regresi sendiri. Hal ini membuat MARS sangat cocok untuk masalah dengan masukan yang berdimensi tinggi [3].

Gambar 2, menunjukkan contoh sederhana bagaimana MARS akan menggunakan *piece-wise* regresi linear splines untuk mencoba untuk menyesuaikan data, dalam dua dimensi ruang (dimana Y adalah variabel dependen dan X adalah variabel bebas).



Gambar 2. Contoh knots pada MARS

Konsep utama adalah pengertian tentang *knot*, yang merupakan titik yang menandai akhir dari data daerah mana yang berbeda adalah dijalankan persamaan regresi, yaitu di mana perilaku dari model fungsi CHANGE. Gambar 2, menunjukkan dua knot: 16 dan 24. Knot tersebut membatasi tiga interval dimana *linear relationships* berbeda diidentifikasi [3].

MARS tidak membuat asumsi tentang mendasari *functional relationship* antara

dependen dan independen variabel. Untuk membangun model regresi itu fleksibel dengan tiba-tiba *splines* terpisah (atau fungsi dasar) untuk interval yang berbeda pada variabel independen. Kedua variabel yang akan digunakan dan titik akhir dari interval untuk setiap variabel (yaitu knot) ditemukan melalui pencarian intensif cepat tetapi prosedur. Selain mencari variabel satu per satu, MARS juga dapat melakukan pencarian untuk interaksi antara independen variabel, yang memungkinkan setiap tingkat interaksi untuk dipertimbangkan selama model yang dibangun dapat lebih sesuai dengan data. Model MARS umum dapat direpresentasikan menggunakan persamaan berikut:

$$\hat{y} = c_0 + \sum_{m=1}^M c_m \prod_{k=1}^{K_m} b_{km}(x_{v(k,m)}) \quad (6)$$

Dimana $\hat{y}$ adalah variabel dependen yang diperkirakan oleh model MARS, c_0 adalah konstanta, $b_{km}(x_{v(k,m)})$ adalah perpotongan fungsi dasar dengan $v(k, m)$ menjadi indeks independen variabel yang digunakan m th dalam jangka waktu k th produk, dan K_m adalah parameter yang membatasi urutan interaksi (model yang dihasilkan akan menjadi aditif untuk $K_m = 1$, dan berpasangan interaksi diperbolehkan untuk $K_m = 2$). *splines* b_{km} adalah didefinisikan dalam pasangan:

$$b_{km}(x) = (x - t_{km})_+^q = \begin{cases} (x - t_{km})^q & \text{if } x > t_{km} \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

$$b_{km+1}(x) = (t_{km} - x)_+^q = \begin{cases} (t_{km} - x)^q & \text{if } t_{km} > x \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

untuk m bilangan bulat, di mana t_{km} , salah satu dari nilai-nilai unik $x_{v(k, m)}$, dikenal sebagai simpul dari spline, $q \geq 0$ adalah nilai terkuat yang dibangkitkan untuk *splines* untuk

memanipulasi derajat resultant regression yang dihasilkan model. Ketika $q = 1$, *splines* linier sederhana diterapkan.

Model optimal MARS dibangun dalam dua tahap: *forward* proses seleksi bertahap diikuti oleh *backward* "pruning" proses. Pemilihan *forward* bertahap dimulai dari fungsi dasar dengan fungsi dasar konstan. Di setiap langkah, dari semua kemungkinan perpecahan di masing-masing fungsi dasar, proses memilih membagi diminimalkan kriteria "lack of fit". Pencarian ini berlanjut hingga mencapai beberapa model yang telah ditetapkan jumlah maksimum pada fungsi dasar. Dalam proses *backward* "pruning", kriteria yang sesuai "lack of fit" digunakan untuk mengevaluasi kontribusi masing-masing fungsi dasar deskriptif model. Fungsi Dasar memberikan kontribusi paling tidak untuk menghilangkan model secara bertahap. Model yang optimal secara bertahap dalam urutan kemudian dipilih. Di sini, kurangnya sesuai ukuran yang digunakan adalah berdasarkan pada kriteria generalized cross-validation (GCV), didefinisikan sebagai:

$$GCV(M) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y})^2 / \left[1 - \frac{C(M)}{n} \right]^2 \quad (9)$$

di mana n adalah jumlah pengamatan dalam data set, M jumlah non-constant istilah dalam model, dan $C(M)$ adalah fungsi *complexity penalty*. Tujuan $C(M)$ adalah model *complexity penalty*, untuk menghindari overfitting, dan untuk mempromosikan model parsimony. Biasanya didefinisikan sebagai:

$$C(M) = M + cd \quad (10)$$

di mana c adalah biaya yang ditetapkan pengguna faktor penalti untuk setiap basis fungsi optimasi, dan d adalah derajat kebebasan efektif,

yang sama dengan jumlah fungsi dasar independen dalam model. Semakin tinggi faktor c adalah, maka fungsi dasar semakin akan dikeluarkan. Dalam praktiknya, c meningkat selama langkah pemangkasan untuk memperoleh model yang lebih kecil.

Setelah model dibangun, kemungkinannya adalah untuk memperkirakan, pada skala antara 0 dan 100, kepentingan variabel relatif dalam hal kontribusi pada kecocokan model. Untuk menghitung kepentingan relatif dari suatu variabel, kita menghapus semua istilah yang berisi variabel tersebut, reparasi model, dan kemudian menghitung pengurangan yang sesuai. Terpenting variable dengan (skor tertinggi) adalah salah satu yang ketika dihapus, paling mengurangi kesesuaian model. Variable yang kurang penting menerima nilai yang lebih rendah. Skor ini sesuai dengan rasio dari pengurangan yang sesuai diproduksi oleh variabel-variabel ini dari variabel yang paling penting.

1. EVALUASI MODEL

Bagian ini, memperkenalkan kriteria untuk mengevaluasi akurasi prediksi model. Kemudian, akan dibahas Model construction yang digunakan dalam studi ini, serta dijelaskan juga metode pengujian yang signifikan.

A. Model Construction

Ada sejumlah software yang membantu dengan konstruksi *Bayesian network*, perangkat lunak yang dikembangkan memungkinkan pengguna untuk membangun sebuah type khusus yang disebut *Bayesian networks* (*Naive-Bayes classifier*), direpresentasikan dengan *single node* sebagai klasifikasi variable

yang terhubung keseluruhan *node* lainnya sebagai representasi dari prediksi variable. Ini mengasumsikan Bayesian network tentang pengetahuan propabilitas sebelumnya [2], tetapi melakukan *batch learning* untuk mempelajari data. Oleh sebab itu *Bayesian network* digunakan sebagai model prediksi *maintainability*, satu node CHANGE terhubung dengan 10 variabel metric. Pada table 1. tapi tidak cukup dengan pengetahuan yang tersedia, secara kuantitatif probabilitas distribusi lebih dahulu sebelum jaringan ini. Dengan demikian, jaringan perlu melakukan *batch learning* dengan menggunakan pengetahuan *subset*.

Setelah *batch learning*, memprediksi jaringan distribusi dengan probabilitas posterior CHANGE untuk masing-masing kasus dalam tes yang sesuai subset, dengan menghitung bersama distribusi probabilitas, dengan menghitung, probabilitas distribusi gabungan, seperti yang telah disebutkan sebelumnya, suatu model *Bayesian networks* outputnya memprediksi estimasi interval. namun pada paper ini menggunakan estimasi point. Hal ini karena estimasi interval sulit untuk membandingkan dengan model lainnya. Intinya Perkiraan ini dipilih dari interval untuk memperkirakan nilai yang memiliki probabilitas tertinggi. Apabila lebih dari satu nilai tertinggi dengan probabilitas yang sama, maka nilai rata-rata yang dipilih.

B. Ukuran Prediksi Akurasi

Makalah ini mengevaluasi dan membandingkan perangkat lunak OO *maintainability* model prediksi kuantitatif, dengan menggunakan **prediksi akurasi ukuran** berikut langkah-langkah: absolut residual

(Ab.Res.), magnitude of relative error (MRE) dan Pred measures.

Ab.Res. merupakan nilai absolut residual, diberikan oleh persamaan berikut:

$$\text{Ab.Res.} = |\text{actual value} - \text{predicted value}| \quad (11)$$

Dalam makalah ini digunakan, jumlah absolut residual (Sum Ab.Res.), Median dari absolute residuals (Med.Ab.Res.) dan standart deviasi dari absolute residuals (SD Ab.Res.). Sum Ab.Res. mengukur total residuals atas Datasct. Med.Ab.Res. mengukur tendensi sentral dari residual distribution. Med.Ab.Res. dipilih untuk menjadi ukuran kecenderungan pusat karena residual distribution biasanya tidak simetris dengan dataset perangkat lunak. SD Ab.Res. digunakan untuk mengukur dispersi residual distribution.

MRE adalah ukuran normalisasi ketidaksesuaian antara nilai actual dan nilai predicted, yang diberikan oleh persamaan berikut:

$$\text{MRE} = \frac{|\text{actual value} - \text{predicted value}|}{\text{actual value}} \quad (12)$$

Dalam makalah ini, nilai maksimum MRE (Max.MRE) adalah digunakan. Max.MRE ukuran yang maximum relative, yang setara dengan maximum error relatif terhadap usaha yang sebenarnya dalam prediksi. Mean MRE, mean magnitude of relative error (M.MRE):

$$\text{MMRE} = \frac{1}{n} \sum_{i=1}^n \text{MRE}_i \quad (13)$$

juga digunakan. MMRE mengukur rata-rata relative ketidak sesuaian, yang setara dengan kesalahan rata-rata relatif terhadap usaha yang sebenarnya dalam prediksi. Kadang-kadang MMRE dinyatakan dalam%. Namun, makalah ini mengikuti difinition diberikan dalam

Persamaan. (13) dan tidak mengekspresikan MMRE dalam%.

Pred adalah ukuran dari apa yang diperkirakan proporsi MRE nilai-nilai kurang dari atau sama dengan nilai tertentu, diberikan:

$$\text{Pred}(q) = \frac{k}{n} \quad (14)$$

dimana q adalah nilai tertentu, k adalah jumlah kasus MRE yang kurang dari atau sama dengan q, dan n adalah total jumlah kasus di Dataset. Dalam makalah ini, Pred (0,25) dan Pred (0,30) digunakan karena kedua tindakan Pred umum digunakan dalam upaya memprediksi perangkat lunak [2]. Laporan sebelumnya Perkiraan persentase dengan MRE dari 25% atau kurang, dan laporan terakhir perkiraan persentase dengan MRE 30% atau kurang [3].

C. Cross Validation

Ketika sebuah model dibangun, kita harus melakukan beberapa jenis *cross-validasi*. *Cross-validasi* adalah cara untuk mendapatkan perkiraan realistis pada model prediksi agar mempunyai kekuatan ketika itu diterapkan pada kumpulan data. Secara umum, sebuah kumpulan data dibagi menjadi dua himpunan bagian (subsets): training set dan test set. training set digunakan agar sesuai dengan model, dan tes set digunakan untuk memvalidasi model. Ini yang disebut dengan validasi *split-sampel*. Karena data set yang digunakan dalam upaya prediksi biasanya ukurannya terbatas, validasi *split-sampel* sulit dilakukan. Sebagai perbaikannya, *v-fold cross-validation* adalah cara memperoleh estimasi dengan *prediction error*. Untuk kumpulan data dengan n pengamatan, *v-fold cross-validation* membagi kumpulan data menjadi v kira-kira sama seperti partisi, dan

masing-masing pada gilirannya digunakan untuk pengujian sementara sisanya digunakan untuk training.

Leave-one-out (LOO) adalah *n-fold cross-validasi*, di mana *n* adalah jumlah pengamatan dalam kumpulan data. Kami memilih untuk menggunakan LOO validasi dalam kajian ini karena tiga alasan. **Pertama**, LOO cross-validation yang banyak digunakan adalah varian dari *v-fold cross-validation*. Lebih penting lagi, dari sudut pandang seorang praktisi adalah lebih dekat pada kenyataan *k-cross validation* ($k < n$). **Kedua**, tidak seperti *k-cross validation* ($k < n$), LOO adalah deterministik, yaitu tidak ada sampling terlibat. **Ketiga**, LOO menjamin bahwa besarnya jumlah data yang digunakan untuk training untuk setiap kasus, yang mungkin meningkatkan kemungkinan mendapatkan estimasi yang akurat mungkin dapat diperoleh. Kekurangan dari LOO adalah bahwa komputasi secara intensif, untuk seluruh pembelajaran prosedur harus dijalankan *n* kali. Namun, hal ini tidak masalah bagi penelitian kami karena relatif kecil untuk ukuran sampel [3].

D. Signifikansi Test

Dilakukan pengamatan apakah MARS (*Multivariate Adaptive Regression Spline*) lebih unggul untuk MLR (*Multivariate Linear Regression*), ANN (*Artificial Neural Network*), RT (*Regression Tree*), dan SVR (*Support Vector Regression*) untuk membangun maintainability dengan model prediksi [3]. Dengan demikian, Wilcoxon signedrank tes untuk sampel yang cocok digunakan untuk menguji dan membandingkan di MARS model dan model-model lain dalam bentuk dua jenis kesalahan,

ARE (*Absolute Residual Error*) and MRE (*Magnitude Of Relative Error*).

Prosedur pengujian pertama menghitung perbedaan antara pasangan pengamatan, peringkat mereka dari terkecil hingga terbesar dengan nilai absolut (pangkat ditugaskan mengikat pengamatan adalah mean dari barisan yang akan ditugaskan pengamatan itu mereka tidak terikat), dan kemudian menambahkan tanda perbedaan dari masing-masing yang sesuai peringkat. Jumlah dari peringkat memiliki tanda tambah disebut (T+) dan jumlah dari peringkat memiliki tanda minus disebut (T-). Bila ukuran sampel *n* adalah lebih besar dari 25, distribusi T (di mana baik T+ atau T- dapat digunakan untuk T) sangat erat didekati dengan distribusi normal dengan mean [3]:

$$\mu_T = \frac{n(n+1)}{4} \quad (15)$$

Dengan standart error :

$$\sigma_T = \sqrt{\frac{n(n+1)(2n+1)}{24}} \quad (16)$$

Dengan demikian, uji statistik dapat dihitung dari :

$$Z = \frac{|T - \mu_T| - 0.5}{\sigma_T} \quad (17)$$

dimana untuk T kita dapat menggunakan, dengan hasil yang identik, baik T+ atau T-. Kemudian, untuk pengujian dua sisi, jika Z adalah lebih besar dari atau sama dengan nilai kritis $Z_{\alpha(2)}$, maka hipotesis nol H_0 ($\mu_1 = \mu_2$) ditolak, di mana μ_1 dan μ_2 adalah dua populasi sarana pasangan cocok. Artinya, hipotesis alternative H_1 ($\mu_1 \neq \mu_2$) diterima. Secara khusus, jika $Z \leq -Z_{\alpha(2)}$ dan $T+ < T-$, kita menyimpulkan bahwa $\mu_1 < \mu_2$; jika $Z \geq Z_{\alpha(2)}$ dan $T+ > T-$, kita menyimpulkan bahwa $\mu_1 >$

μ_2 . Tingkat signifikansi statistik yang memberikan wawasan ke dalam akurasi pengujian. Pemilihan mungkin tergantung pada analisis dan properti dari masalah diselidiki. Dalam perangkat lunak teknik empiris, biasanya untuk menetapkan $\alpha = 0,05$ atau $0,1$.

Dalam studi ini, null dan hipotesis alternatif menggunakan ARE adalah sebagai berikut:

$$H_0: ARE_{MARS} = ARE_X$$

$$H_1: ARE_{MARS} \neq ARE_X$$

di mana X menunjukkan MLR, ANN, RT, atau SVR. Demikian pula hipotesis null dan alternatif menggunakan MRE adalah sebagai berikut:

$$H_0: MRE_{MARS} = MRE_X$$

$$H_1: MRE_{MARS} \neq MRE_X$$

Tingkat signifikansi minimum untuk menolak hipotesis null ditetapkan pada $\alpha = 0,05$.

2. ANALISA MODEL

Pada bagian ini, kami menganalisis hasil model Bayesian network dan MARS menggunakan dataset UIMS (*User Interface Management System*). Untuk menunjukkan adanya perbedaan akurasi yang signifikan dari setiap model prediksi *software maintainability*, dengan membandingkan prediksi model Bayesian network dengan model MARS.

A. Hasil Dari Dataset UIMS

Tabel 3 menunjukkan akurasi prediksi langkah-langkah model Bayesian network, dicapai oleh model MARS. LOO (*Leave-one-out*) kumpulan data dalam *cross-validation*. Meskipun hasil tidak memenuhi kriteria untuk model prediksi yang akurat, dijelaskan bahwa akurasi prediksi perangkat lunak model prediksi sebagai upaya pemeliharaan kecenderungan rendah, sehingga sangat sulit untuk memenuhi

kriteria [3]. Seperti dapat dilihat, model MARS memiliki nilai terkecil pada **Med.Ad.Res** sementara **Max.MRE**, **MMRE**, **Pred (0,25)**, **Pred (0,30)**, **Sum Ab.Res** dan **SD.Ab.Res** nilai secara signifikan lebih tinggi dibandingkan dengan model Bayesian network. Oleh karena itu, disimpulkan bahwa model MARS dapat memprediksi *maintainability* dari Dataset UIMS lebih baik dari pada yang model Bayesian network.

TABLE 3.
PREDIKSI AKURASI UNTUK DATA SET
UIMS

Absolut Residual	Bayesian Network	MARS
Max.MRE	7.039	14.06
MMRE	0.972	1.86
Pred (0,25)	0.446	0.28
Pred (0,30)	0.469	0.28
Sum Ab.Res	362.300	1532.78
Med.Ad.Res	10.550	9.26
SD.Ab.Res	46.652	59,76

KESIMPULAN

Dalam makalah ini, kami mempresentasikan hasil analisis yang berusaha untuk membangun model prediksi perangkat lunak berorientasi obyek *maintainability* dengan menggunakan model eksplorasi Bayesian network dan multivariate adaptive regression splines (MARS). Untuk membangun kedua model ini, memanfaatkan dari data set UIMS, diperoleh dari dua sistem berorientasi objek. Prediksi kinerja dari kedua model itu dinilai dan dibandingkan. Hasil menunjukkan bahwa model MARS dapat secara efektif memprediksi *maintainability* OO dari sistem perangkat lunak.

Dengan demikian, secara keseluruhan, dapat disimpulkan bahwa model MARS lebih baik daripada Bayesian network yang dibangun menggunakan teknik pemodelan yang khas dan dapat berguna untuk teknik *software maintainability prediction*. Salah satu batasan dari analisis ini adalah bahwa menggunakan data set UIMS untuk kedepannya dapat digunakan data set QUES atau yang lainnya. Studi semacam itu akan memungkinkan kita untuk menyelidiki lebih lanjut kemampuan MARS dalam *software maintainability prediction*.

DAFTAR PUSTAKA

Wei Li and Sallie Henry, "Object-Oriented Metrics that Predict Maintainability", Elsevier, J. Systems Software; vol.23, pp.111-122, 1993.

C. van Koten and A.R. Gray, "An application of Bayesian network for predicting object-oriented software maintainability", Elsevier, Information and Software Technology; vol. 48, pp59-67, 2006.

Yuming Zhou, Hareton Leung, "Predicting object-oriented software maintainability using multivariate adaptive regression splines", Elsevier, The Journal of Systems and Software; vol.80, pp.1349-1361, 2007.

Alf Inge Wang and Erik Arisholm, "The effect of task order on the maintainability of object-oriented software", Elsevier, Information and Software Technology; vol.51, pp.293-305, 2009.

ISBN 978-979-602 -261-8



9 789796 022618