Rancangan Estimasi Biaya dengan Teknik COCOMO II dan Neuro Fuzzy Studi Kasus: Sistem Informasi Rumah Sakit

[1]Hengki Suhartoyo dan [2]Tri Adhi Wijaya
[1, 2] Teknik Informatika Fakultas Teknik Universitas Dr. Soetomo Surabaya
e-mail: hengki@thegratsoft.com, tri.adhiwijaya@gmail.com

Abstrak— *Estimasi biaya pengembangan perangkat lunak merupakan proses pemrediksian biaya yang harus dikeluarkan dalam seluruh tahapan pembangunan perangkat lunak dengan menggunakan beberapa parameter. Terdapat beberapa metode dan kakas bantu dalam proses estimasi biaya. Salah satu kakas yang populer adalah COCOMO II. Demi meningkatkan keakuratan hasil estimasi biaya pembangunan perangkat lunak dengan menggunakan COCOMO II, diusulkan penggunaan metode-metode bantu dalam proses pengolahan data input maupun data output. Pemodelan Neuro-Fuzzy merupakan salah satunya. Metode ini digunakan dalam proses pengolahan data-data input, yakni pada proses penerjemahan nilai kualitatif menjadi nilai pengali kuantitatif yang kemudian digunakan sebagai nilai input bagi COCOMO II. Pada beberapa penelitian, metode ini terbukti dapat meningkatkan akurasi hasil estimasi biaya COCOMO II dengan dataset COCOMO '81. Namun demikian, diperlukan sebuah penelitian lanjutan dengan dataset proyek pembangunan perangkat lunak sesungguhnya untuk menguji peningkatan akurasi estimasi biaya dengan menggunakan COCOMO II yang dioptimasi dengan metode Neuro-Fuzzy.Pada makalah rancangan ini diusulkan penggunaan dataset pembangunan sistem informasi rumah sakit. Kontribusi yang diharapkan dari makalah ini adalah dapat diketahui signifikansi pengaruh penambahan satu cost driver baru yaitu tipe rumah sakit pada proses estimasi biaya pembangunan perangkat lunak dengan teknik COCOMO II dan Neuro-Fuzzy.*

Kata Kunci— Estimasi biaya, pembangunan perangkat lunak, COCOMO II, Neuro-fuzzy.

I. PENDAHULUAN

Biaya merupakan salah satu dari tiga pilar utama yang menentukan sukses tidaknya suatu proyek perangkat lunak, sedangkan dua yang lainnya adalah penjadwalan dan kinerja. Suatu proyek yang "*over budget*" seringkali terpaksa dihentikan di tengah jalan dikarenakan pihak terkait yang membiayai proyek tersebut kehabisan uang atau pihak terkait merasa jika proyek tersebut diteruskan maka ia harus mengeluarkan uang lebih banyak lagi. Inilah yang menyebabkan proses estimasi biaya (*cost estimation*) dalam suatu proyek pembangunan perangkat lunak memiliki peran yang sangat penting.

Berbagai macam teknik estimasi biaya perangkat lunak (*software cost estimation*) yang disingkat SCE telah dikembangkan dan diteliti selama beberapa dekade ini [1]. Menurut Boehm dkk[2], sedikitnya terdapat empat tujuan dalam aktivitas perkiraan biaya pembangunan perangkat lunak, diantaranya: *budgeting, tradeoff and risk analysis, project planning and control*, dan *improvement investment analysis*.Tujuan utama dari penelitian-penelitian mengenai teknik-teknik SCE ini adalah mengidentifikasi metode prediksi "terbaik", karena tiap model estimasi bertujuan untuk mengevaluasi sebuah fungsi yang dapat secara akurat melakukan prediksi terhadap biaya dari proyek yang baru.

Salah satu kakas yang populer dalam estimasi biaya pembangunan perangkat lunak adalah COCOMO II. COCOMO II merupakan pengembangan dari COCOMO dan dipublikasikan pada tahun 1997. Model COCOMO merupakan sebuah model estimasi biaya yang memanfaatkan sifat-sifat regresi dalam pemodelannya. COCOMO dikembangkan oleh Boehm pada 1981 dan merupakan pemodelan paling populer dari seluruh model prediksi tradisional. Model COCOMO dapat digunakan untuk menghitung jumlah dari usaha dan waktu dalam pembangunan perangkat lunak.Salah satu kelemahannya adalah ketidaksesuaian pemodelannya dengan lingkungan pengembangan perangkat lunak pada akhir 1990-an. Oleh karena itu, pada 1997, di publikasikan COCOMO II.

Meskipun cukup populer, COCOMO II mempunyai beberapa kelemahan [3]diantaranya: (1) hubungan antara metrik keluaran perangkat lunak dan factor kontribusi menunjukkan karakteristik nonlinear yang kompleks dan kuat, (2) pengukuran metrik perangkat lunak yang tidak presisi, dan (3) sulitnya penggunaan pengetahuan ahli dan data proyek numerik dalam satu model. Dari beberapa pendekatan yang digunakan untuk mengatasi permasalahan-permasalahan tersebut, neuro-fuzzy dapat menjadi solusi yang menjanjikan [3]. Hal ini didasari atas kemampuan belajar neuro-fuzzy yang tinggi, kehandalan ketika berhadapan dengan input yang kurang jelas, dan kemudahan dalam memanfaatkan informasi yang berbeda-beda yang berasal dari sumber informasi yang berbeda-beda pula [3].

Neuro-fuzzy merupakan teknik hibridasi antara teknik jaringan syaraf tiruan (neural network) dan teknik fuzzy. Teknik ini telah digunakan dalam banyak bidang. Salah satu penelitian yang menggunakan teknik ini adalah penelitian yang dilakukan oleh Kalbande dkk [5] dimana neuro-fuzzy digunakan dalam proses pemilihan teknologi yang tepat pada sistem pembayaran tol secara elektronik.

Penelitian mengenai COCOMO II yang di optimasi dengan teknik neuro-fuzzy dimulai oleh Huang dkk [3]. Tujuan utama dari penelitian ini adalah untuk mengetahui efektifitas dari penggunaan teknik logika fuzzy dalam peningkatan akurasi dari estimasi usaha. Pemodelan yang diajukan oleh Huang dkk, dapat di-interpretasikan dan divalidasi oleh ahli dan mempunyai kemampuan generalisasi yang baik. Pemodelan ini dapat menangani secara efektif masukan dengan presisi yang kurang baik dan tidak pasti, dan meningkatkan akurasi dari estimasi biaya pembangunan perangkat lunak. Selain itu, pemodelan yang diajukan memungkinkan data input berupa nilai kontinyu dan linguistik, sehingga dapat digunakan untuk menghindari masalah dari proyek-proyek serupa yang memiliki perbedaan estimasi biaya yang besar.

Pengunaan COCOMO II yang dioptimasi dengan neuro-fuzzy juga diusulkan oleh Attarzadeh dkk.[4]. Metode ini dilakukan dengan cara mengkarakteristikkan parameter-parameter input dengan menggunakan dua sisi fungsi Gaussian yang memberikan transisi superior dari satu interval ke interval lainnya. Dari penelitian ini didapatkan peningkatan akurasi estimasi usahasebesar 12,63%.

Penggunaan neuro-fuzzy untuk meningkatkan hasil akurasi estimasi biaya terbukti lebih baik daripada penggunaan metode-metode yang lain, seperti Fuzzy-Genetic Algorithm Hybrid Model, Halstead Model, Walston-Felix Model, Bailey-Basili Model, Doty Model, dan Genetic Algorithm Based Model[6]. Dalam perbandingannya, model neuro-fuzzy memiliki nilai Mean Magnitude of Relative Error (MMRE) dan Root Mean Square Error (RMSE) terendah yakni sebesar 0,015 dan 0,012.

Namun, penelitian-penelitian yang telah dikemukakan sebelumnya menggunakan dataset COCOMO ’81 sebagai data uji penelitian[2][3][4][6]. Dataset ini merupakan dataset pembangunan perangkat lunak National Aeronautics and Space Administration (NASA)[7]. Oleh karena itu, diperlukan sebuah penelitian lanjutan untuk menguji apakah penggunaan neuro-fuzzy dalam optimasi COCOMO II pada proses estimasi biaya juga dapat diaplikasikan pada proses pembangunan perangkat lunak lainnya. Rancangan penelitian ini diajukan untuk menjawab pertanyaan ini, khususnya dalam proses estimasi

biaya untuk sistem informasi rumah sakit. Terdapat tiga alasan utama dari penggunaan dataset rumah sakit, yakni (1) banyak proyek pembangunan perangkat lunak rumah sakit berbiaya sangat tinggi karena kompleksitas perangkat lunak yang dibangun, (2) banyak proyek pembangunan perangkat lunak rumah sakit yang gagal selesai akibat membengkaknya kebutuhan sehingga berakibat pada membengkaknya biaya, dan (3) terdapat perbedaan biaya pembangunan perangkat lunak rumah sakit yang besar antara satu *developer/vendor* dengan *developer/vendor* yang lain meskipun kebutuhan pada sistem informasi ini secara umum adalah sama.

II. DASAR TEORI

A. COCOMO II

Model COCOMO merupakan sebuah model estimasi biaya yang memanfaatkan sifat-sifat regresi dalam pemodelannya.COCOMO dikembangkan oleh Boehm pada 1981 dan merupakan pemodelan paling populer dari seluruh model prediksi tradisional. Model COCOMO dapat digunakan untuk menghitung jumlah dari usaha dan waktu dalam pembangunan perangkat lunak.Salah satu kelemahannya adalah ketidaksesuaian pemodelannya dengan lingkungan pengembangan perangkat lunak pada akhir 1990-an. Oleh karena itu, pada 1997, dipublikasikan COCOMO II. Fungsi utama yang ditambahkan pada COCOMO II adalah agar dapat digunakan untuk melakukan estimasi usahapada proyek-proyek non-sequential dan dengan menggunakan model proses *rapid development, re-engineering*, proyek-proyek dengan pendekatan *reuse driven*, serta pendekatan berorientasi objek. Terdapat tiga pemodelan pada COCOMO II yang merupakan pengembangan dari COCOMO[4], yakni:

• *Application Composition Model*: cocok digunakan untuk proyek yang dibangun dengan perkakas GUI-builder.

• *Early Design Model*: untuk mendapatkan estimasi kasar dari biaya dan waktu sebelum ditentukan arsitektur dari seluruh perangkat lunak yang akan dibangun. Pada pemodelan ini, digunakan sekumpulan kecil dari Cost Drivers baru dan persamaan estimasi baru yakni berdasarkan Unadjusted Function Point atau KSLOC.

• *Post-Architecture Model*: merupakan pemodelan paling detil dibandingkan dengan dua pemodelan lainnya, yang digunakan setelah seluruh arsitektur dari proyek yang telah di desain. Salah satu komponen yang digunakan adalah function points atau LOC. Pemodelan ini melibatkan proses pembangunan dan perawatan sesungguhnya dari sebuah produk perangkat lunak.

COCOMO II melibatkan 17 *cost drivers* yang digunakan dalam Post-Architecture model. Cost drivers pada COCOMO II berada pada kisaran level Very Low sampai Extra High, sama seperti pada COCOMO. Post-Architecture model COCOMO II memenuhi persamaan:

$$Effort = A \times [size]^B \times \prod_{i=1}^{17} Effort\ multiplier_i \qquad (1)$$

dimana

$$B = 1.01 + 0.01 \times \sum_{j=1}^{5} Scale\ factor_j \qquad (2)$$

A = konstanta multiplikasi

Size = ukuran dari proyek perangkat lunak yang diukur dengan terms dari KSLOC (jumlah dari *source line of code, function points, atau object points*)

Tabel 1: COCOMO II Cost Driver [4]

Cost Driver	***Range***
Required software reliability (RELY)	0.82-1.26
Database size (DATA)	0.90-1.28
Product complexity (CPLX)	0.73-1.74
Developed for reusability (RUSE)	0.95-1.24
Documentation match to life-cycle needs (DOCU)	0.81-1.23
Execution time constraint (TIME)	1.00-1.63
Main storage constraint (STOR)	1.00-1.46
Platform volatility (PVOL)	0.87-1.30
Analyst capability (ACAP)	1.42-0.71
Programmer capability (PCAP)	1.34-0.76
Personnel continuity (PCON)	1.29-0.81
Applications experience (APEX)	1.22-0.81
Platform experience (PLEX)	1.19-0.85
Language and tool experience (LTEX)	1.20-0.84
Use of software tools (TOOL)	1.17-0.78
Multi site development (SITE)	1.22-0.80
Required development schedule (SCED)	1.43-1.00

Seleksi dari *Scale Factors* (SF) berdasarkan pada logika rasional dimana faktor-faktor tersebut merupakan sumber yang signifikan dalam penentuan *project effort* atau *productivity variation*. Standar dari nilai numerik *cost drivers* tertera pada Tabel 1.

Pada Tabel 1, terdapat 22 *cost driver* yang menjadi acuan dalam penentuan biaya yang di-estimasi. Terdapat satu cost driver yang dapat merepresentasikan ukuran dari instansi pengguna perangkat lunak, yakni database size. Pada keadaan umum, semakin besar instansi yang menggunakan perangkat lunak, semakin besar pula ukuran database yang digunakan.

Namun demikian, tidak selalu ukuran database dapat dijadikan acuan besarnya instansi dan berkorelasi dengan besarnya biaya yang dibutuhkan dalam pembangunan perangkat lunak.Sebagai contoh, Rumah Sakit Umum Daerah mempunyai database yang lebih besar daripada Rumah Sakit Swasta, namun demikian harga perangkat lunak yang dibangun lebih tinggi.

B. Neural Network Model

Neural Network bekerja berdasarkan pada prinsip dari pembelajaran dari contoh dengan tanpa adanya penjelasan informasi terlebih dahulu. Tiga entitas utama dari neural networks adalah neurons, *interconnection structure*, dan *learning algorithms*. Penggunaan neural network model dimulai dengan sebuah tampilan yang sesuai dari neurons, atau koneksi antar network nodes. Proses ini meliputi pendefinisian dari jumlah lapisan dari neurons, jumlah neurons pada tiap-tiap lapisan, dan cara menghubungkan neuron-neuron tersebut. Fungsi aktivasi dari node-node dan algoritma training khusus

yang digunakan juga harus ditentukan.Setelah network telah dibangun, neural network tersebut harus dilatih dengan sekumpulan data historis. Kemudian pada neural network akan dilakukan iterasi sesuai dengan algoritma pelatihannya, dan secara otomatis menyesuaikan bobot (parameter) sampai bobot dari model menjadi konvergen. Setelah proses training telah selesai dilakukan dan bobot untuk tiap networklink telah ditentukan, input baru dapat dimasukkan kedalam pemodelan untuk dilakukan pengklasifikasian. Secara umum, kumpulan data yang besar dibutuhkan untuk meningkatkan akurasi dari neural network model.

C. Logika Fuzzy

Dengan semakin kompleksnya sistem, maka semakin sulit untuk mendefinisikan sebuah keadaan dengan tepat sesuai dengan perilakunya. Logika Fuzzy menawarkan pendekatan yang sangat tepat mengenai ketidakpastian yang terdapat pada kompleksitas perilaku manusia. Pada dasarnya, Logika Fuzzy adalah logika yang tepat dari ketidaktepatan dan penalaran perkiraan. Logika Fuzzy merupakan upaya formulasi/mekanisasi dua kemampuan manusia yang luar biasa, yakni kemampuan untuk membuat keputusan yang tepat dalam lingkungan informasi yang tidak sempurna, dan kemampuan untuk melakukan berbagai tugas-tugas fisik dan mental tanpa pengukuran dan setiap perhitungan.

Logika Fuzzy pertama kali diperkenalkan oleh Zadeh pada 1965. Metode ini mengembangkan permodelan yang lebih mirip dengan natural language processing dengan cara mendefinisikan kumpulan dari rule-rule IF, AND, THEN. Hasil dari pemodelan ini disebut sebagai Fuzzy Rules.

Logika Fuzzy dimulai dari teori kumpulan fuzzy. Teori ini merupakan sebuah teori dari kumpulan kelas-kelas dengan batasan yang kurang tajam (tidak jelas) dan dikenal sebagai perluasan dari classical set theory. Fungsi keanggotaan dari μA(x) dari sebuah elemen x dari sebuah classical set A, sebagai subset dari X, didefinisikan dengan persamaan (3) berikut ini:

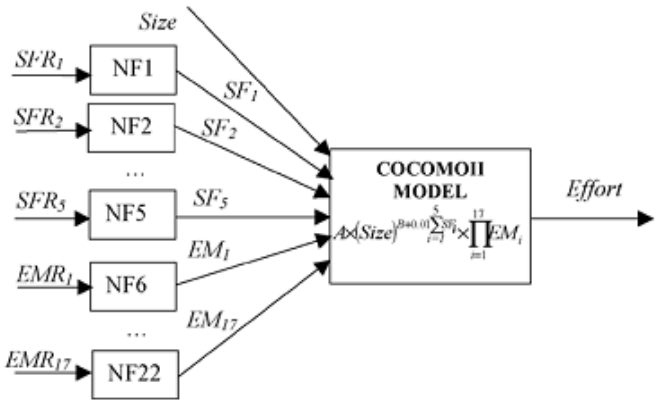
$$\mu_A(x) = \begin{cases} 1 \text{ if } x \in A \\ 0 \text{ if } x \notin A \end{cases} \qquad (3)$$

Terdapat empat komponen utama pada sistem fuzzy logic dengan fuzzifier dan defuzzifier seperti yang dijelaskan berikut:

1. *Fuzzification*: mengubah sebuah input crisp menjadi sebuah fuzy set.
2. *Fuzzy Rule Base*: fuzzy logic systems menggunakan fuzzy IF-THEN rules
3. *Fuzzy Inference Engine*: setelah seluruh nilai input crisp di fuzzi-fikasikan menjadi nilai linguistik yang sesuai, inference engine mengakses fuzzy rule base untuk menurunkan nilai linguistic untuk variabel linguistik intermediate dan variabel linguistik output.
4. *Defuzzification*: proses untuk mengubah output fuzzy menjadi output yang crisp.

D. Neuro Fuzzy model oleh Huang dkk

Gambar sistem Neuro-fuzzy model untuk estimasi biaya pembangunan perangkat lunak diperlihatkan pada Gambar 1. Input pada model ini adalah ukuran perangkat lunak dan ratings dari 22 *cost drivers* termasuk 5 Scale Factors (SFRi) dan 17 Effort Multipliers (EMRi). Output dari sistem ini adalah estimasi biaya pembangunan perangkat lunak.



Gambar 1: Model Neuro-Fuzzy dalam estimasi biaya pembangunan perangkat lunak oleh Huang dkk.[3]

Ratings dari cost drivers dapat berupa nilai numerical continuous atau linguistic terms seperti low, nominal, dan high. Parameter-parameter pada model ini dikalibrasikan dengan pembelajaran pada data proyek sesungguhnya. Terdapat dua komponen utama dari sistem neuro-fuzzy model, yakni sebagai berikut:

• Dua puluh dua sub-models NFi: untuk tiap sub-model, nilai input adalah rating value dari sebuah cost driver. Outputnya adalah nilai multiplier yang sesuai yang digunakan sebagai input COCOMO II.

• COCOMO II model: nilai input adalah ukuran dari perangkat lunak (SLOC), dan output dari NFi. Outputnya adalah hasil prediksi usaha yang dibutuhkan dalam pembangunan perangkat lunak.

E. Neuro Fuzzy Model oleh Sandhu dkk.

Sandhu dkk, membandingkan beberapa model untuk mengestimasi biaya pembuatan perangkat lunak, model- model yang digunakan adalah Halstead, Walston-Felix, Bailey-Basili, Doty, GA Based, Fuzzy-GA dan Neuro-Fuzzy (NF). Fuzzy dalam NF yang digunakan dalam penelitian Sandhu dkk, adalahSugeno Based Fuzzy Inference System, Fuzzy menginisialisasi membership function (mf) dari 16 atribut yang berbeda dan fungsi keanggotaan linier untuk output dan deduksi aturan fuzzy dari data. Sugeno Based fuzzy inference system dilatih dengan Neural Network menggunakan hybrid training algorithm. Algoritma pembelajaran Forward dan Backpropagation digunakan untuk memperbarui parameter non-linear dan linear dari sistem Neuro-fuzzy.

Beberapa model yang diuji dibandingkan menggunakan:

1. mean magnitude of relative error (MMRE),
2. root mean square error (RMSE),
3. PRED(30), dan
4. PRED(10).

RMSE sering digunakan untuk membandingkan perbedaan diantara nilai-nilai yang diprediksi oleh model atau estimator dengan nilai-nilai actual dari obyek yang diamati.

Shandu menggunakan dataset estimasi biaya perangat lunak NASA dan data tersebut akan ditraining dengan GA, data tersebut terdiri dari 16 atribut yaitu: *Analysts Capability, Programmers Capability, Application Experience, Modern Programming Practices, Use of Software Kakass, Virtual*

Machine Experience, Language Experience, Schedule Constraint, Main Memory Constraint, Data Base Size, Time Constraint for Cpu, Turnaround Time, Machine Volatility, Process Complexity, Required Software Reliability and Lines of Source Code.

Tabel 2: Project-wise results of the different model used for the effort dataset [6]

Actual Usaha	**Model Used**						
	Fuzzy-GA	**Neuro-Fuzzy**	**Halstead**	**Walston-Felix**	**Bailey-Basili**	**Doty**	**GA-Based**
958	956	958	729,54	14,049	38,9	166,7	54,61
237	237	237	364,48	9,2216	25,03	102,7	38,94
130	130	130	650	13,099	36,05	153,8	51,62
70	70	69,99	573,58	12,141	33,23	140,9	48,57
57	57	57	90,181	3,9521	12,13	38,74	19,71
50	50	49,98	770,44	14,522	40,34	173,2	56,08
38	38	37,99	142,75	5,2219	14,96	53,38	24,66

Tabel 3: Hasil perbandingan beberapa model untuk usaha dataset [6]

Performance Criteria	***Model Used***						
	Fuzzy-GA	***Neuro-Fuzzy***	***Halstead***	***Walston-Felix***	***Bailey-Basili***	***Doty***	***GA-Based***
MMRE	2.0722	0.014988	887.78	83.584	60.447	126.75	66.528
RMSE	1.6036	0.11651	26575	1880.9	1691.6	1382.1	1815.7
PRED (30)	100	100	3.1746	0	22.222	25.397	14.286
PRED (10)	100	100	1.5873	0	11.111	6.3492	1.5873

Hasil dari perhitungan beberapa model yang digunakan dalam usaha dataset ditunjukkan pada Tabel 2. Pada Tabel tersebut dapat dilihat bahwa Fuzzy-GA dan Nuero-Fuzzy memiliki hasil yang hampir seimbang, namun setelah diuji dengan membandingkan menggunakan MMRE dan RMSE serta dengan PRED(30) dan PRED(10) seperti pada Tabel 3, dapat kita lihat bahwa Neuro-Fuzzy Model yang paling unggul dengan nilai MMRE dan RMSE terendah serta PRED(30) dan PRED(10) yang mencapai 100.

F. Rumah Sakit

Menurut Peraturan Menteri Kesehatan Nomor 340/MENKES/PER/III/2010 tentang Klasifikasi Rumah Sakit[8], berdasarkan fugsi dan tugas dari Rumah Sakit, tipe-tipe Rumah Sakit berdasarkan kemampuan sebuah Rumah Sakit dalam memberikan pelayanan medis kepada para pasiennya dapat digolongkan menjadi 5 tipe, yaitu sebagai berikut.

1. Rumah Sakit Tipe A, adalah rumah sakit yang mampu memberikan pelayanan kedokteran spesialis dan subspesialis luas yang oleh pemerintah ditetapkan sebagai rujukan tertinggi (Top Referral Hospital) atau disebut pula sebagai rumah sakit pusat.Saat ini di Indonesia hanya terdapat 4 buah RSU Tipe A yaitu RSU Cipto Mangunkusumo di Jakarta, RSU Dr. Sutomo di Surabaya, RSUP Adam Malik di Medan, dan RSUP DR. Wahidin Sudiro Husodo di Ujung Pandang.
2. Rumah Sakit Tipe B, adalah rumah sakit yang mampu memberikan pelayanan kedokteran spesialis dan subspesialis terbatas.Rumah sakit ini didirikan disetiap Ibukota propinsi yang menampung pelayanan rujukan di rumah sakit kabupaten.

3. Rumah Sakit Tipe C, adalah rumah sakit yang mampu memberikan pelayanan kedokeran spesialis terbatas.Rumah sakit ini didirikan disetiap ibukota Kabupaten (Regency hospital) yang menampung pelayanan rujukan dari puskesmas.
4. Rumah Sakit Tipe D, adalah rumah sakit yang bersifat transisi dengan kemampuan hanya memberikan pelayanan kedokteran umum dan gigi. Rumah sakit ini menampung rujukan yang berasal dari puskesmas.
5. Rumah Sakit Tipe E, adalah rumah sakit khusus (spesial hospital) yang menyelenggarakan hanya satu macam pelayan kesehatan kedokteran saja. Saat ini banyak rumah sakit kelas ini ditemukan misal, rumah sakit kusta, paru, jantung, kanker, ibu dan anak.

Dalam pembangunan sistem informasi rumah sakit, terdapat banyak proyek pembangunan perangkat lunak rumah sakit berbiaya sangat tinggi karena kompleksitas perangkat lunak yang dibangun.sehingga banyak proyek pembangunan perangkat lunak rumah sakit yang gagal selesai akibat membengkaknya requirement sehingga berakibat pada membengkaknya biaya. Pembengkakan kebutuhan ini banyak disebabkan oleh kurangnya pengetahuan proses bisnis yang baku dan keinginan pihak terkaitrumah sakit yang tidak sesuai dengan kebutuhan fungsional dari tiap proses. Namun demikian, meskipun mempunyai proses bisnis yang mirip terdapatperbedaan biaya pembangunan perangkat lunak rumah sakit yang besar antara satu developer/vendor dengan developer/vendor yang lain meskipun requirement pada sistem informasi ini secara general adalah sama. Hal-hal inilah yang menjadi alasan penggunaan datasetpembangunan sistem informasi rumah sakit dalam rancangan penelitian yang diusulkan ini.

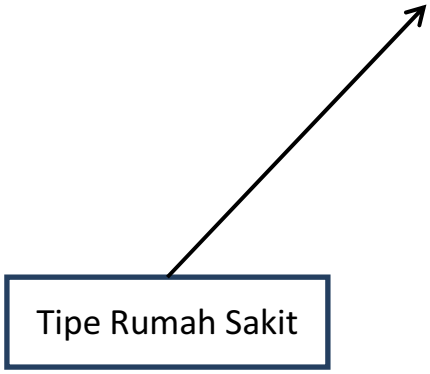
III. DESAIN DAN RANCANGAN SISTEM

Usulan Model Baru Neuro-Fuzzy

Model baru Neuro-Fuzzy yang diusulkan merupakan hasil modifikasi dari model Huang dkk.[3] seperti yang terlihat pada Gambar 2. Input pada model ini adalah ukuran perangkat lunak, tipe rumah sakit dan ratings dari 22 cost drivers termasuk 5 scale factors (SFRi) dan 17 effort multipliers (EMRi). Tipe-tipe rumah sakit yang digunakan adalah Tipe A, Tipe B dan Tipe C. Output dari sistem ini adalah estimasi biaya pembangunan perangkat lunak. Ratings dari cost drivers dapat berupa nilai numerical continuous atau linguistic terms seperti low, nominal, dan high. Parameter-parameter pada model ini dikalibrasikan dengan pembelajaran pada data proyek sesungguhnya, dan dalam penelitian ini adalah dataset proyek pembangunan sistem informasi rumah sakit.

Dataset dan Model Evaluasi

Pada penelitian ini, digunakan data-data yang berasal dari questioner yang diambil dari beberapa developer perangkat lunak rumah sakit yang terdapat di Surabaya dan sekitarnya. Questioner tersebut didasarkan pada *questioner* standard COCOMO II.



Gambar 2: Model Baru Neuro-Fuzzy hasil modifikasi dari Model Huang dkk.[3]

Pada data questioner, juga ditambahkan di rumah sakit mana perangkat lunak tersebut di bangun, apa tipe rumah sakitnya, dan berapa rencana biaya serta realisasi biayanya berapa. Hal ini penting, agar diketahui faktor-faktor lain yang mungkin mempengaruhi hasil estimasi biaya.

Metode yang digunakan untuk menghitung nilai error adalah Magnitude Of Relative Error (MRE) untuk tiap project i, yang memenuhi persamaan:

$$MRE_i = \frac{|estimated\ error_i - actual\ effort_i|}{actual\ effort_i} \qquad (4)$$

Untuk multiple project N digunakan Mean Magnitude of Relative Error (MMRE), yang memenuhi persamaan:

$$MMRE = \frac{1}{N}\sum_{i=1}^{N} MRE_i \qquad (5)$$

IV. ANALISIS DAN PEMBAHASAN

Dari analisis awal, didapatkan bahwa biaya yang dikeluarkan oleh rumah sakit untuk membangun sistem informasi rumah sakit yang terintegrasi sangat beragam jumlahnya. Kecenderungan yang diperoleh adalah semakin tinggi tipe rumah sakit, maka biaya yang dikeluarkan untuk membangun sistem informasi rumah sakit terintegrasi adalah semakin besar.

Dengan menggunakan komponen-komponen yang berpengaruh dalam penghitungan biaya yang sama seperti terlihat pada Tabel 1 ditambah dengan komponen size (SLOC) dari perangkat lunak yang dibangun, dan juga tipe dari rumah sakit, dilakukan penghitungan estimasi biaya dalam proyek pembangunan sistem informasi rumah sakit. Hasilnya akan dibandingkan dengan realisasi biaya akhir yang dikeluarkan oleh rumah sakit. Ketika hasil pembandingan menunjukkan nilai yang berbeda, sistem akan melakukan proses pelatihan kembali hingga nilai error yang dihasilkan sudah mencapai target yang telah didefinisikan sebelumnya.

Ketika iterasi dalam proses training dihentikana akan didapatkan pola neuro-fuzzy. Pola ini kemudian digunakan dalam data testing.

Tipe rumah sakit akan sangat berpengaruh dalam proses estimasi biaya pembangunan sistem informasi rumah sakit. Semakin tinggi tipe dari rumah sakit, biaya yang muncul akan semakin besar, meskipun komponen pembentuk biayanya sama. Hal ini dapat disebabkan oleh keinginan *vendor/developer* yang membangun perangkat lunak tersebut mendapatkan keuntungan yang lebih besar. *Vendor/developer* cenderung melihat bahwa tipe rumah sakit yang tinggi mempunyai jumlah dana yang tinggi pula. Sedangkan tipe rumah sakit yang rendah, mempunyai dana yang relatif sedikit.

Hal lain yang sering dijadikan alasan oleh vendor/developer aplikasi untuk menaikkan biaya adalah semakin besar tipe rumah sakit, maka database yang digunakan adalah semakin besar. Perangkat keras yang digunakan harus mempunyai kemampuan lebih tinggi dalam memproses dan menyimpan data karena proses dan data yang ditangani juga lebih banyak jumlahnya. Namun demikian, hal ini seharusnya tidak menjadikan perbedaan biaya yang sangat tinggi antara biaya pembangunan sistem informasi rumah sakit dengan tipe tinggi dan rendah, karena proses bisnisnya adalah sama.

Dalam pemodelan neuro-fuzzy yang diusulkan, tipe rumah sakit akan berfungsi sebagai faktor pengali (multiplier) dari proses estimasi biaya perangkat lunak rumah sakit. Seperti yang telah dijelaskan sebelumnya, kecenderungan yang diperoleh adalah semakin tinggi tipe rumah sakit, maka biaya yang dikeluarkan untuk membangun sistem informasi rumah sakit terintegrasi adalah semakin besar, meskipun komponen-komponenlain pembentuk biaya pembangunan sistem informasi rumah sakit adalah sama.

Faktor pengali ini juga muncul pada pembangunan sistem informasi pada sebuah organisasi/institusi yang proses bisnisnya adalah sama, namun mempunyai ukuran yang berbeda. Sebagai contoh adalah pembangunan sistem akademik di universitas. Biaya dalam pembangunan sistem informasi akademik di universitas dengan reputasi yang baik akan lebih besar daripada biaya pembangunan sistem informasi akademik di universitas dengan reputasi kurang baik. Yang membedakan keduanya adalah besaran data yang harus disimpan dan proses yang lebih besar. Sedangkan proses bisnisnya adalah sama.

Penambahan satu nilai input baru pada pemodelan neuro-fuzzyoleh Huang dkk.[3]akan menjadi faktor pengali yang sangat mempengaruhi hasil dari estimasi biaya pembangunan perangkat lunak. Pemodelan yang diusulkan dapat diterapkan tidak hanya pada proses estimasi biaya pembangunan sistem informasi rumah sakit, melainkan dapat pula digunakan dalam proses estimasi biaya pembangunan sistem informasi lainnya.

V. KESIMPULAN

Dalam pemodelan neuro-fuzzy yang diusulkan dalam penelitian ini, tipe rumah sakit berfungsi sebagai faktor pengali dalam proses estimasi biaya pembangunan sistem informasi rumah sakit. Penambahan faktor pengali, yakni tipe dari suatu organisasi, dapat menjadi salah satu faktor dalam

penentuan biaya pembangunan perangkat lunak. Tidak hanya dapat diterapkan pada proses estimasi biaya pembangunan sistem informasi rumah sakit, melainkan dapat pula digunakan dalam proses estimasi biaya pembangunan sistem informasi lainnya.

Dari rancangan penelitian ini, dapat diteruskan menjadi sebuah penelitian sesungguhnya yakni Estimasi Biaya dengan Teknik COCOMO II dan Neuro Fuzzy dengan memperhatikan ukuran organisasi/institusi.

DAFTAR PUSTAKA

[1] Jorgensen, M., Shepperd, M., "A systematic review of software development cost estimation studies". IEEE Transactions on Software Engineering 33 (1), 33–53, 2007

[2] Boehm, B., Abts, C., Chulani, S. "Software development cost estimation approaches – a survey". Annals of Software Engineering 10, 177–205, 2000.

[3] Huang, X., Ho, D., Ren, J., Capretz, L.F. "Improving the COCOMO model usinga neuro-fuzzy approach". Department of ECE, University of Western Ontario, London, Ont., Canada N6A 5B9 dan Toronto Design Center, Motorola Canada Ltd., Markham, Ont., Canada L6G 1B3, 2006.

[4] Attarzadeh, I. Ow, S.H. "A Novel Algorithmic Cost Estimation Model Based on Soft Computing Technique". Department of Software Engineering, Faculty of Computer Science and Information Technology, University of Malaya, 50603 Kuala Lumpur, Malaysia. Journal of Computer Science 6 (2): 117-125, 2010

[5] Kalbande, D.R. Singhlal, P. Deotale, N. Shah, S., Thampi, G.T. "An Advanced Technology Selection Model using Neuro Fuzzy Algorithm for Electronic Toll Collection System". University of Mumbai, India. International Journal of Advanced Computer Science and Applications, Vol. 2, No. 4, 2011.

[6] Sandhu, P.S., Prashar, M., Bassi, P., Bisht, A. "A Model for Estimation of efforts in Development of Software Systems". World Academy of Science, Engineering and Technology, 56, 2009

[7] http://promisedata.org/repository/data/nasa93/nasa93.arff

[8] Peraturan Menteri Kesehatan Republik Indonesia Nomor 340/Menkes/Per/Iii/2010, Tentang Klasifikasi Rumah Sakit