

PERKIRAAN BIAYA PERANGKAT LUNAK MENGGUNAKAN LOGIKA FUZZY

Denny Setia, Mulyadi Irman Hermadi, Wisnu Ananta Kusuma

ABSTRACT

Software cost estimation is an integral part of software development. Underestimated cost will most probable result low quality software. On the contrary, an overestimated one will cause resource wastages.

This research has developed a model of software cost estimation using fuzzy logic. It also compares two COCOMOs estimation methods: Intermediate COCOMO and Fuzzy Intermediate COCOMO. The data used is NASA Projects done in between 1980 and 1990. The Models developed are evaluated using MRE (Magnitude of Relative Error), Mean MRE (MMRE), Median MRE (MdMRE), Min MRE, Max MRE, and Standard Deviation MRE (SdMRE). However, we only present MMRE to the readers, because it represents overall model performances.

This experiment resulted MMREs of 59.50% for Intermediate COCOMO and 51.95% for Fuzzy Intermediate COCOMO. These results indicate that the fuzzy version of Intermediate COCOMO outperforms the non-fuzzy one. In addition, Fuzzy Intermediate COCOMO is sensitive to the values of Cost Drivers

Keywords: software cost estimation, fuzzy logic, COCOMO, MMRE

PENDAHULUAN

Latar Belakang

Perkiraan biaya perangkat lunak sangat penting dalam pengembangan perangkat lunak. Perkiraan yang terlalu kecil dari biaya aktualnya akan mengakibatkan perangkat lunak yang dihasilkan berkualitas rendah. Sebaliknya, apabila perkiraan biaya yang terlalu besar akan mengakibatkan pemborosan sumber daya yang digunakan.

Dalam pengembangan perangkat lunak terdapat banyak faktor yang mempengaruhi biaya pembuatan perangkat lunak. Faktor-faktor tersebut diukur dalam enam *linguistic value*, yaitu: 'very low', 'low', 'nominal', 'high', 'very high' dan 'extra high'. Dengan demikian setiap faktor hanya masuk dalam satu kelas (skala ordinal) tertentu. Hal ini akan menimbulkan perbedaan perkiraan biaya perangkat lunak yang cukup jauh pada dua proyek yang analogi. Masalah ini disebabkan karena penggunaan interval klasik dalam merepresentasikan variabel linguistik.

Pada penelitian yang dilakukan oleh Idri et al (2001), representasi variabel linguistik tidak dilakukan dengan interval klasik melainkan menggunakan himpunan *fuzzy* (*fuzzy sets*). Kinerja dari penggunaan konsep *fuzzy* tersebut menunjukkan hasil estimasi yang lebih akurat daripada menggunakan interval klasik.

Tujuan

Tujuan dari penelitian ini adalah memodelkan perkiraan biaya perangkat lunak menggunakan logika *fuzzy*. Dari pemodelan tersebut akan dianalisis kinerja penerapan logika *fuzzy* untuk perkiraan biaya perangkat lunak.

Ruang Lingkup

Ruang lingkup penelitian ini adalah sebagai berikut:

- 1 Data yang digunakan untuk pengujian adalah NASA *Project* yang terdiri dari 93 proyek dari tahun 1980 – 1990.
- 2 Atribut yang dipakai dalam perhitungan perkiraan biaya perangkat lunak ini adalah model pengembangan proyek, ukuran perangkat lunak, biaya aktual, dan 15 jenis *Cost Driver*.

Manfaat

Penelitian ini memberikan manfaat berupa pemodelan perkiraan biaya perangkat lunak menggunakan logika *fuzzy* yang digunakan untuk memprediksi biaya pada proyek yang selanjutnya.

Perkiraan Biaya Perangkat Lunak

Perkiraan biaya perangkat lunak adalah proses memprediksi biaya yang diperlukan dalam pengembangan sistem perangkat lunak (Leung 2001). Perkiraan biaya perangkat lunak mengacu pada prediksi jumlah biaya, waktu, dan *staffing level* yang diperlukan untuk membangun sistem perangkat lunak (Saliu 2003).

Perkiraan biaya perangkat lunak dapat diklasifikasi menjadi metode pemodelan algoritmik dan pemodelan non-algoritmik. Pemodelan algoritmik diturunkan dari analisis statistikal dari data proyek terdahulu (Saliu 2003). Contoh dari pemodelan algoritmik adalah model linear, model multiplikatif, dan model *power-function*. Sedangkan pemodelan non-algoritmik berdasarkan pengalaman pakar dan perbandingan dengan proyek sebelumnya yang analogi (Dickson 2007). Contoh yang paling umum adalah *expert judgment* dan *analogy costing*.

Proyek Analogi

Proyek yang analogi adalah proyek-proyek yang paling serupa (Briand et al. 1999). Proyek analogi dapat dicari dengan menggunakan persamaan jarak *Euclidean* n-dimensi dimana setiap dimensi mewakili satu atribut (Schofield dan Shepperd 1996). Persamaannya adalah sebagai berikut :

$$E = \sqrt{(a_1 - a_2)^2 + (b_1 - b_2)^2 + \dots + (z_1 - z_2)^2} \dots (1)$$

dimana E adalah jarak *Euclidean*, serta a,b,...,z adalah atribut-atribut proyek 1 dan proyek 2. Dua buah proyek dengan jarak terkecil akan dinyatakan sebagai proyek yang analogi.

Cost Driver

Dalam pengembangan perangkat lunak terdapat faktor-faktor yang mempengaruhi biaya pengembangan yang disebut dengan *Cost Driver* (CD). Menurut Saliu (2003) terdapat 15 jenis CD, yaitu : STOR, DATA, TIME, TURN, VIRT, CPLX, RELY, ACAP, PCAP, AEXP, MODP, TOOL, VEXP, LEXP dan SCED.

Intermediate COCOMO (Cost Constructive Model)

Model *Intermediate COCOMO* adalah ekstensi dari versi awal yang menghitung biaya perkiraan perangkat lunak hanya dengan menggunakan ukuran perangkat lunak dan model pengembangan proyek saja. Dalam *Intermediate COCOMO* terdapat penambahan atribut perhitungan yang digunakan yaitu CD, yang terdiri dari 15 jenis (Idri et al. 2001). Sehingga perkiraan biaya perangkat lunak tidak hanya berdasarkan ukuran dan model proyek tapi juga bergantung dari lingkungan pengembangan.

Persamaan perkiraan biaya perangkat lunak pada *Intermediate COCOMO* adalah sebagai berikut:

$$MM_{est} = A \times SIZE^B \prod_{i=1}^{i=15} C_{ij} \dots (2)$$

dimana MM_{est} adalah biaya estimasi (dalam *Man-Months*), *SIZE* adalah ukuran perangkat lunak dalam *KDSI (Kilo Delivery Source Instruction)*, A dan B adalah koefisien untuk tiap model proyek (*organic*, *semidetached*, dan *embedded*) dan C_{ij} adalah *effort multiplier* untuk tiap-tiap atribut pada *Cost Driver*.

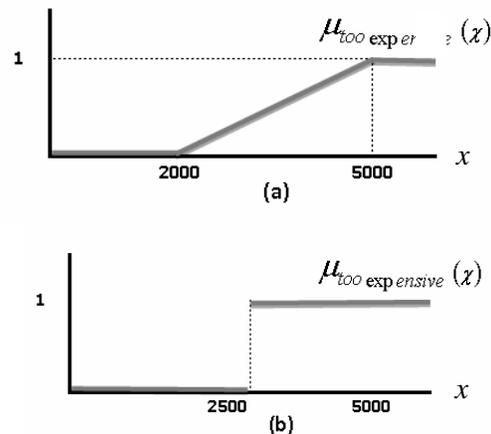
Logika Fuzzy

Zadeh (1996) menyatakan bahwa logika *fuzzy* berhubungan dengan prinsip-prinsip pemberian alasan formal mengenai suatu hal yang tidak mempunyai ketentuan atau pemberian alasan perkiraan. Penggunaan model logika *fuzzy* memungkinkan pembuatan keputusan yang relatif di dalam suatu lingkungan ketidakpastian dan ketidaktepatan.

Himpunan *fuzzy* berbeda dengan himpunan klasik (*crisp sets*). Dalam himpunan klasik, objek yang merupakan elemen dari suatu himpunan, maka fungsi keanggotaannya adalah 1. Sedangkan jika objek tersebut

bukan elemen dari suatu himpunan, maka fungsi keanggotaannya adalah 0. Dengan demikian himpunan klasik ini nilainya adalah {0,1}. Tipe pemikiran ini merupakan logika *bivalent* atau *two-value-logic*, yaitu logika yang hanya mempunyai dua nilai kebenaran, yaitu *true* (benar/1) dan *false* (salah/0). Problem dalam logika *bivalent* ini adalah dalam dunia nyata sesuatu pada umumnya tidak hanya ada dalam dua nilai kebenaran.

Dalam himpunan *fuzzy*, suatu objek mungkin menjadi elemen suatu himpunan secara sebagian. Tingkat keanggotaan dalam himpunan *fuzzy* disebut derajat keanggotaan, dimana nilainya diukur dari perluasan fungsi keanggotaan himpunan klasik. Fungsi keanggotaan pada himpunan *fuzzy* A dinyatakan dengan $\mu_A(x)$. Fungsi keanggotaan ini memetakan x kedalam *kodomain* dalam bentuk interval dari 0 sampai 1. Nilai 0 menunjukkan bahwa x bukan anggota himpunan, nilai 1 menunjukkan x anggota himpunan secara penuh, sedangkan nilai antara 0 sampai 1 (selain 0 dan 1) menunjukkan derajat keanggotaan x dalam suatu himpunan. Ilustrasi dari perbedaan antara himpunan *fuzzy* dan himpunan klasik dapat dilihat pada Gambar 1.



Gambar 1 (a) Himpunan *fuzzy*
(b) himpunan klasik.

Pada Gambar 1 terlihat bahwa pada himpunan *fuzzy* suatu nilai bisa menjadi anggota himpunan secara sebagian (misal $x = 3000$), sedangkan pada himpunan klasik nilai tersebut hanya masuk ke salah satu himpunan saja (*too expensive*)

Fuzzy Intermediate COCOMO

Proses dalam perkiraan biaya perangkat lunak menggunakan logika *fuzzy* adalah mendefinisikan fungsi keanggotaan untuk setiap variabel linguistik pada *Cost Driver* dengan fungsi keanggotaan yang bertipe *trapezoid* (Idri et al. 2001). Tetapi tidak semua faktor pada *Cost Driver* didefinisikan dalam *fuzzy sets* karena hanya merupakan sebuah deskripsi biasa (Saliu 2003). Faktor-faktor tersebut adalah RELY, CPLX, MODP, dan TOOL (Idri et al. 2001). Proses penghitungan perkiraan biaya perangkat lunak menggunakan logika *fuzzy* sama dengan COCOMO, tetapi *effort multipliers* yang digunakan berasal dari sebuah persamaan. Persamaannya adalah :

$$F_{-}C_{ij} = \sum_{j=1}^{k_i} \mu_{A_j}^{V_i}(P) \times C_{ij} \dots(3)$$

dimana $F_{-}C_{ij}$ adalah *effort multipliers* yang didapatkan dari himpunan *fuzzy*, $\mu_{A_j}^{V_i}$ adalah fungsi keanggotaan dari himpunan *fuzzy* A_j yang berasosiasi dengan *Cost Driver* V_i , P adalah urutan proyek dan K_i adalah jumlah *Cost Driver*.

Evaluasi Model Perkiraan Biaya Perangkat Lunak

Proses evaluasi model perkiraan biaya perangkat lunak adalah membandingkan akurasi dari biaya estimasi dengan biaya aktualnya (Boehm, 1981 dalam Indri et al, 2001). Dalam COCOMO digunakan lima jenis nilai untuk mengukur keakuratan dari sebuah model perkiraan biaya perangkat lunak, (Briand et all, 1998) yaitu:

- $MRE_i = \frac{|MM_{est}^i - MM_{act}^i|}{MM_{act}^i} \dots(4)$

dimana MRE_i adalah *Magnitude of Relative Error* dari proyek ke- i , MM_{est}^i adalah biaya estimasi untuk proyek ke- i , dan MM_{act}^i adalah biaya aktual untuk proyek ke- i .

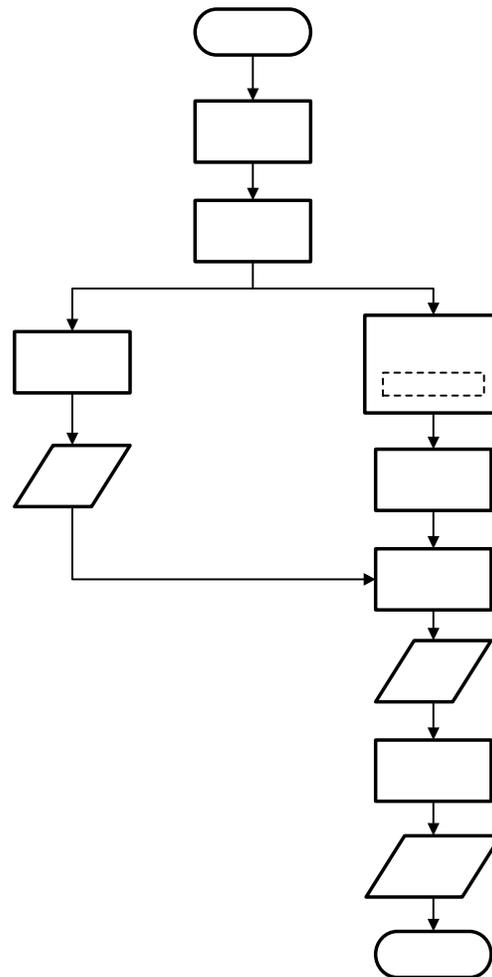
- $Mean\ MRE = \frac{1}{N} \sum_i MRE_i \dots(5)$

Mean MRE (MMRE) digunakan untuk menghitung agregat nilai MRE pada N proyek (pengujian).

- Median MRE (MdMRE) digunakan untuk menghitung nilai tengah dari sekumpulan MRE proyek.
- Standar Deviasi MRE (SdMRE) digunakan untuk mengukur seberapa besar penyimpangan nilai MRE dari nilai rata-ratanya.
- Pengukuran lainnya adalah MRE terkecil (*min* MRE) dan MRE terbesar (*max* MRE). Nilai tersebut digunakan untuk mengetahui seberapa baik dan buruk akurasi dari sistem.

METODE PENELITIAN

Tahapan penelitian yang digunakan dapat dilihat pada Gambar 2, dengan tahap – tahap sebagai berikut :



Gambar 2 Metode penelitian.

Studi Literatur

Pada tahap ini, dilakukan pengumpulan informasi tentang perkiraan biaya perangkat lunak menggunakan *Fuzzy Intermediate COCOMO*. Literatur berupa buku, jurnal dan artikel dari internet.

Pengumpulan Data

Data yang dipakai pada penelitian ini adalah *NASA Project*, yang didapat dari situs <http://www.promisedata.org>. Data tersebut berasal dari berbagai proyek yang dilakukan NASA dari tahun 1980 – 1990, yang terdiri dari 93 proyek. Data ini terdiri dari 24 atribut, yaitu 15 jenis CD yang bertipe numerik, 7 atribut tentang deskripsi proyek, ukuran perangkat lunak dalam KDSI, dan biaya aktual proyek dalam satuan *Man Month* (MM).

Tidak seluruh atribut pada *NASA Project* digunakan dalam penelitian ini. Atribut yang digunakan adalah 15 jenis CD, model pengembangan proyek, ukuran perangkat lunak dan biaya aktual proyek. Enam atribut lainnya tidak digunakan karena hanya berupa deskripsi proyek, seperti: nomor id, tahun pengembangan, nama proyek, kategori aplikasi yang dikembangkan, dan lokasi pengembangan.

Pengembangan Sistem

Proses pengembangan sistem pada penelitian ini mempunyai dilaksanakan dalam dua tahap, yaitu: pengembangan sistem, dan pengujian sistem.

Pada tahap pengembangan sistem, tiap-tiap CD berikut nilai dan derajat keanggotaannya akan dibuat fungsi keanggotaannya. Fungsi keanggotaan yang digunakan pada penelitian ini berasal dari jurnal *Cost Model Using Fuzzy Logic* (Idri at al. 2001). Kemudian kode program untuk fungsi keanggotaan ini dibuat menggunakan MATLAB.

Setelah sistem selesai dibuat, maka dilakukan pengujian terhadap sistem. Pengujian dilakukan dengan melakukan pengecekan terhadap kesalahan *syntax* ataupun logika pada kode program yang dibuat. Jika sistem sudah teruji barulah dapat memproses data.

Pengujian dan Analisis

Pengujian pada penelitian ini dibagi menjadi dua tahap. Tahap yang pertama adalah pengujian dan analisis sebelum penerapan logika *fuzzy*. Penghitungan biaya perangkat lunak pada pengujian ini dilakukan dengan menggunakan *Intermediate COCOMO*. Atribut yang digunakan dalam pengujian adalah 15 CD, model pengembangan, ukuran perangkat lunak dan biaya aktual.

Pengujian akan menghasilkan perkiraan besarnya biaya yang diperlukan untuk pengembangan perangkat lunak. Hasil pengujian tersebut dievaluasi dengan membandingkan dengan biaya aktualnya menggunakan MRE. Setelah dihitung nilai MRE dihitung juga nilai-nilai galat : MMRE, *min* MRE, *max* MRE, MdmMRE dan SdMRE.

Pengujian untuk kelompok yang kedua adalah pengujian setelah penerapan logika *fuzzy*. Pengujian

dilakukan dengan menggunakan metode *Fuzzy Intermediate COCOMO*. Atribut yang dipakai sama seperti pada pengujian kelompok pertama. Evaluasi juga dilaksanakan dengan menggunakan nilai-nilai galat yang sama dengan pengujian menggunakan *Intermediate COCOMO*.

Hasil pengujian *Fuzzy Intermediate COCOMO* akan dibandingkan dengan hasil pengujian *Intermediate COCOMO*. Perbandingan yang dilakukan adalah akurasi dan kinerja kedua metode tersebut. Analisis terhadap akurasi dilakukan dengan cara membandingkan besarnya nilai-nilai galat MMRE, *min* MRE dan *max* MRE antara kedua metode tersebut. Sedangkan untuk perbandingan kinerja dilakukan pada dua proyek yang analogi. Kedua proyek tersebut dihitung biaya perkiraannya menggunakan *Intermediate COCOMO* dan *Fuzzy Intermediate COCOMO*, kemudian hasilnya akan dibandingkan dan dianalisis.

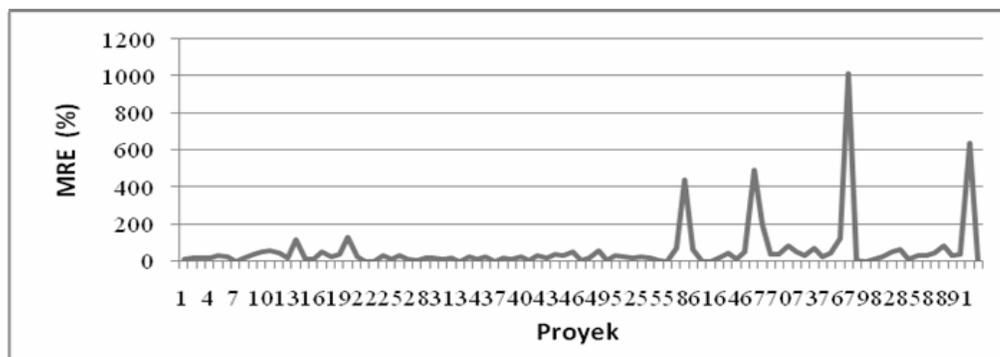
Lingkungan Pengembangan

Perangkat lunak yang digunakan dalam penelitian ini adalah sebagai berikut :

- Microsoft Windows XP Profesional,
- Microsoft Office Excel 2007,
- MATLAB 7.0.

Sedangkan perangkat keras yang digunakan adalah sebagai berikut :

- PC Intel Pentium IV 2.67 GHz,
- DDRAM 1024 MB,
- VGA,
- *hardisk* dengan kapasitas 120 GB,
- monitor VGA dengan resolusi 1024x768 *pixel*,
- *keyboard*,
- *mouse*.



Gambar 3 Grafik MRE NASA Project.

HASIL DAN PEMBAHASAN

Pengujian dan analisis perkiraan biaya menggunakan *Intermediate COCOMO*

Terhadap 93 Proyek dalam NASA Project akan dilakukan pengujian. Galat pada tiap pengujian dihitung dengan menggunakan MRE. Setelah dihitung MRE pada seluruh proyek, dihitung MMRE, *min* MRE, *max* MRE,

MdmMRE dan SdMRE. Grafik MRE untuk seluruh pengujian dapat dilihat pada Gambar 3.

Pada Gambar 3 terlihat bahwa terdapat 4 buah nilai pencilan tinggi. Nilai tersebut berada pada proyek ke-59, 67, 78 dan 92 dengan nilai masing-masing 442,25%, 492,68%, 1041,53% dan 638,09%. Keempat pencilan ini terlihat mencolok karena perbedaan nilai MRE yang sangat jauh dengan nilai MRE proyek lainnya. Pencilan-pencilan tersebut sangat mempengaruhi nilai rata-rata

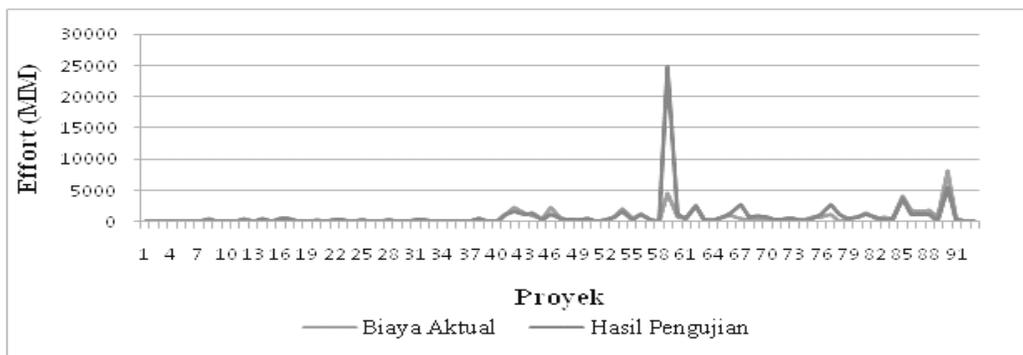
MRE. Hal ini terlihat dari selisih nilai yang besar antara MMRE dan MdMRE yang masing-masing sebesar 59,50% dan 29,53%.

Penyebab besarnya nilai MRE pada pencilan-pencilan tersebut adalah karena biaya aktual yang lebih kecil dibandingkan proyek lain yang sejenis. Sebagai contoh: proyek ke-92 memiliki biaya aktual 12 MM dan proyek ke-93 biaya aktualnya 36 MM. Seluruh CD pada kedua proyek tersebut memiliki kategori yang sama. Perbedaan kedua proyek tersebut terletak pada ukuran perangkat lunak, proyek ke-92 sebesar 6,2 KDSI dan proyek ke-93 sebesar 3 KDSI. Seharusnya makin besar ukuran perangkat lunak, makin besar biaya pengembangannya. Namun hal ini berkebalikan dengan biaya aktual pada proyek ke-92. Sehingga saat dibandingkan biaya perkiraan hasil pengujian dengan biaya aktualnya, didapatkan nilai MRE yang cukup

besar. Kondisi ini dialami juga oleh tiga pencilan lainnya.

Agar dapat diketahui selisih antara hasil pengujian dengan biaya aktual, Gambar 4 menampilkan grafik perbandingan antara biaya aktual dengan biaya perkiraan.

Pada Gambar 4 terlihat bahwa kedua grafik tersebut tidak jauh berbeda. Hanya tampak dua perbedaan yang mencolok pada kedua grafik tersebut, yaitu pada pengujian proyek ke-59 dan proyek ke-90. Dua proyek tersebut memiliki selisih antara biaya aktual dan biaya perkiraan yang cukup besar dibandingkan dengan proyek lainnya. Proyek yang memiliki selisih biaya yang paling besar adalah proyek ke-59 yang juga merupakan pencilan pada pengujian menggunakan *Intermediate COCOMO*.



Gambar 4 Grafik perbandingan biaya aktual dengan hasil pengujian.

Pengujian dan Analisis Fuzzy Intermediate COCOMO

Karena dalam basis data NASA Project pada tiap *Cost Driver* hanya dicantumkan informasi kategori saja yang berupa “very low”, “low”, “nominal”, “high” dan “very high”. Sedangkan pada pengujian yang menggunakan metode fuzzy diperlukan sebuah nilai dari setiap *Cost Driver*. Hal ini diperlukan sebagai input untuk $\mu_{A_j}^V(P)$ (derajat keanggotaan dari *Cost Driver*) pada persamaan 3. Oleh karena itu dibuatlah *dataset* (data tiruan) yang berasal dari basis data yang sebenarnya. *Dataset* ini berisi nilai *effort multipliers* yang sesuai dengan data yang sebenarnya (Idri et al. 2001). Sebagai contoh, misalkan proyek ke-1 dalam NASA Project *Cost Driver DATA* tertulis ‘low’, maka pada proyek ke-1 dalam *dataset* dengan *Cost Driver DATA* akan berisi nilai *random* antara 0 sampai 10. Pada pengujian ini akan digunakan sepuluh dataset.

Pengujian dari kesepuluh dataset tersebut dilakukan dengan menghitung MRE setiap proyek. Setelah dihitung nilai MRE, dihitung nilai MMRE, *min* MRE, *max* MRE, MdMRE dan SdMRE pada setiap pengujian *dataset*. Hasil pengujian dapat dilihat pada Tabel 3.

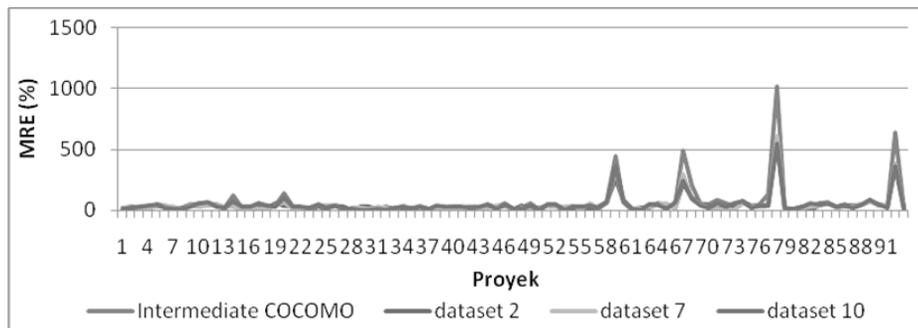
Pada Tabel 3 terlihat bahwa hasil pengujian kesepuluh *dataset* menghasilkan nilai galat yang

berbeda-beda. Namun jika kesepuluh dataset tersebut dihitung menggunakan *Intermediate COCOMO*, maka akan didapatkan nilai galat yang sama seperti pada pengujian data NASA Project. Hal ini terjadi karena pada *Intermediate COCOMO* yang digunakan sebagai inputan untuk menghitung biaya perkiraan adalah kategori dari kelima belas CD. Lain hal pada *Fuzzy Intermediate COCOMO*, yang digunakan sebagai inputan adalah nilai dari kelima belas CD. Dari nilai CD tersebut akan didapatkan derajat keanggotaan kategori tiap CD yang kemudian akan dikalikan dengan *effort multipliers*. Dengan demikian, walaupun kesepuluh *dataset* tersebut kategori CD nya sama, tapi hasil pengujian bisa saja berbeda jika nilai-nilai CD nya berbeda. Dapat disimpulkan bahwa hasil pengujian menggunakan konsep fuzzy sangat dipengaruhi oleh nilai CD.

Analisis yang akan dilakukan pada metode *Fuzzy Intermediate COCOMO* meliputi dua hal, yaitu akurasi dan kinerja. Namun karena sepuluh dataset pada Tabel 3 dihasilkan dan diuji dengan mekanisme yang sama, maka tidak semua dataset akan dibahas. Hanya pada *dataset* tertentu saja, yaitu pada *dataset* dengan nilai MMRE terbesar ataupun terkecil.

Tabel 3 Agregat Galat menggunakan *Fuzzy Intermediate COCOMO*

Galat (%) Dataset	Min MRE	Max MRE	Mean MRE	Median MRE	Standar Deviasi MRE
Dataset 1	0.40	978.12	52.42	29.80	116.35
Dataset 2	0.75	604.88	41.53	26.78	76.67
Dataset 3	0.89	891.14	51.71	32.28	103.21
Dataset 4	1.86	1023.16	61.30	24.90	143.33
Dataset 5	0.02	1014.53	61.61	24.65	140.78
Dataset 6	1.24	999.82	60.79	25.63	138.46
Dataset 7	0.01	604.88	44.35	27.53	81.62
Dataset 8	0.61	815.44	52.70	33.08	96.35
Dataset 9	0.27	933.84	50.41	30.81	104.77
Dataset 10	0.02	538.75	42.64	26.72	74.19



Gambar 5 Grafik perbandingan tiga MMRE terkecil.

a Akurasi *Fuzzy Intermediate COCOMO*

Nilai akurasi didapatkan dari nilai galat MMRE, *min* MRE dan *max* MRE pada tiap-tiap dataset. Makin kecil nilai-nilai tersebut maka makin baik akurasinya.

Pada Tabel 3 terlihat bahwa terdapat tiga *dataset* yang memiliki nilai galat MMRE terkecil yaitu: *dataset 2*, *dataset 7* dan *dataset 10* masing-masing sebesar 42,68%, 44,49%, dan 47,52%. Grafik perbandingan MMRE untuk ketiga *dataset* tersebut dapat dilihat pada Gambar 5. Sebagai pembanding dicantumkan pula grafik hasil pengujian *Intermediate COCOMO*.

Pada Gambar 5 terlihat bahwa grafik hasil pengujian *dataset 2*, *dataset 7*, *dataset 10* dan *Intermediate COCOMO* memiliki bentuk grafik yang hampir sama. Hal tersebut menunjukkan hasil pengujian antara *Intermediate COCOMO* dan *Fuzzy Intermediate COCOMO* menghasilkan MMRE yang tidak jauh berbeda. Perbedaan MMRE yang cukup jauh hanya pada empat proyek, yaitu proyek ke-59, 67, 78 dan 92, dan merupakan pencilon pada grafik-grafik di atas. Pengujian yang menggunakan *Fuzzy Intermediate COCOMO* menghasilkan nilai galat MRE yang lebih kecil dari pada *Intermediate COCOMO* pada empat pencilon tersebut. Hal tersebut yang menyebabkan nilai MMRE ketiga *dataset* lebih baik daripada nilai MMRE *Intermediate COCOMO*.

Tiga *dataset* yang memiliki nilai galat MMRE terbesar yaitu: *dataset 4*, *dataset 5* dan *dataset 6* masing-masing sebesar 59,46%, 61,74% dan 59,81%. Grafik perbandingan MMRE untuk ketiga *dataset* tersebut dapat

dilihat pada Gambar 6. Sebagai pembanding ditampilkan pula grafik hasil pengujian *Intermediate COCOMO*.

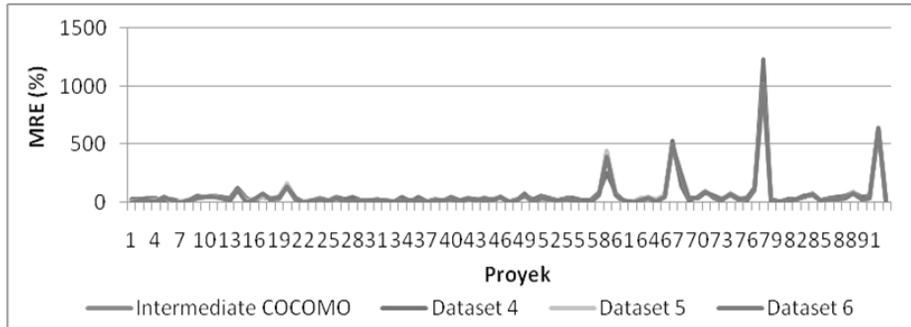
Pada Gambar 6 terlihat grafik hasil pengujian *dataset 4*, *dataset 5*, dan *dataset 6* memiliki bentuk yang sama dengan hasil pengujian *Intermediate COCOMO*. Pencilon-pencilon juga terletak pada proyek yang sama. Secara umum nilai galat MRE tiap proyek pada ketiga *dataset* tersebut nilainya sama dengan nilai galat *Intermediate COCOMO*. Sehingga nilai galat MMRE ketiga *dataset* tersebut tidak jauh berbeda dengan *Intermediate COCOMO*.

Nilai galat MMRE digunakan untuk menghitung akurasi dari sistem secara umum (seluruh proyek). Nilai galat *min* dan *max* MRE menunjukkan seberapa baik dan buruk akurasi dari sistem tersebut. Tiga *dataset* yang memiliki nilai galat MRE terkecil yaitu: *dataset 5*, *dataset 7* dan *dataset 10*, dengan masing-masing nilai *min* MRE yaitu: 0.02%, 0.01% dan 0.02% yang terdapat pada pengujian proyek ke-57, 21 dan 18. Nilai-nilai *min* MRE ini hampir sama dengan nilai *min* MRE pada pengujian menggunakan *Intermediate COCOMO* yaitu sebesar 0.02%. Hal ini menunjukkan bahwa perkiraan biaya perangkat lunak menggunakan logika *fuzzy* juga dapat memiliki akurasi sebaik *Intermediate COCOMO*.

Nilai galat MRE terbesar pada pengujian *Fuzzy Intermediate COCOMO* terletak pada *dataset 4* proyek ke-78 yaitu sebesar 1023.16%. Nilai galat ini lebih besar dari hasil pengujian dengan *Intermediate COCOMO*.

Penyebabnya adalah nilai *random* CD yang dihasilkan pada dataset 4 proyek ke-78 adalah nilai-nilai besar. Karena nilai 15 CD tersebut besar maka akan menghasilkan *fuzzy effort multiplie* yang besar. Sehingga

pengujian menghasilkan biaya perkiraan yang besar. Biaya perkiraan yang dihasilkan adalah 1089.50 MM sedangkan biaya aktualnya adalah 97 MM.



Gambar 6 Grafik perbandingan tiga MMRE terbesar.

b Kinerja Penggunaan Logika Fuzzy

Dalam menghitung biaya perangkat, semakin besar nilai CD: STOR, DATA, TIME, TURN, VIRT, CPLX dan RELY, semakin besar pula biaya perkiraan yang dihasilkan. Sedangkan untuk nilai CD: ACAP, PCAP, AEXP, MODP, TOOL, VEXP, dan LEXP yang semakin besar akan menurunkan biaya perkiraan.

Karena *Fuzzy Intermediate COCOMO* sangat dipengaruhi oleh nilai CD, akan dianalisis kinerja logika *fuzzy* berdasarkan perbedaan nilai-nilai CD. Analisis akan dilakukan pada proyek yang analogi (proyek yang serupa).

Salah satu contoh proyek yang analogi pada pengujian *Fuzzy Intermediate COCOMO* dengan menggunakan *dataset 1* adalah proyek ke-21 dan proyek ke-24.

Berdasarkan Tabel 4 pengujian dua proyek tersebut menghasilkan biaya perkiraan yang tidak jauh berbeda. Namun jika diuji menggunakan *Intermediate COCOMO* hasil pengujian dua proyek tersebut menghasilkan biaya perkiraan sebesar 74,25 MM untuk proyek ke-21 dan 62,22 MM untuk proyek ke-24. Penyebab besarnya selisih biaya perkiraan adalah walaupun dua proyek tersebut analogi tetapi kategori CD kedua proyek tersebut berbeda.

Dari hasil pengujian dua proyek tersebut terlihat bahwa hasil pengujian *Fuzzy Intermediate COCOMO* sangat sensitif terhadap perbedaan nilai CD, sedangkan pada *Intermediate COCOMO* hasil pengujian dipengaruhi oleh kategori kelima belas CD. Pengujian menggunakan *Intermediate COCOMO* akan menghasilkan biaya perkiraan yang sama apabila kategori kelima belas CD antara dua proyek tersebut sama. Sehingga walaupun dua proyek tersebut nilai CD nya tidak jauh berbeda, tapi kategori CD nya berbeda maka selisih biaya perkiraan antara dua proyek tersebut akan besar. Hal ini berbeda dengan *Fuzzy Intermediate COCOMO*, biaya perkiraan yang dihasilkan sesuai dengan nilai CD. Semakin besar perbedaan nilai CD antara dua proyek, maka semakin besar pula selisih

biaya perkiraan yang dihasilkan. Perbedaan kategori CD pada dua proyek tersebut tidak begitu mempengaruhi hasil pengujian.

Tabel 4 Nilai CD dan hasil pengujian proyek yang analogi

Nilai CD	Dataset 1 proyek ke-21	Dataset 1 proyek ke-24
DATA	9	11
TIME	25	50
STOR	29	25
VIRT	180	120
TURN	0.5	4
ACAP	54	49
AEXP	24	36
PCAP	53	55
VEXP	8	9
LEXP	17	12
SCED	81	95
Biaya Perkiraan	60,24 MM	63,18 MM

Pengujian menggunakan *Intermediate COCOMO* akan menghasilkan biaya perkiraan yang sama apabila kategori kelima belas CD antara dua proyek tersebut sama. Sehingga walaupun dua proyek tersebut nilai CDnya tidak jauh berbeda, tapi kategori CDnya berbeda maka selisih biaya perkiraan antara dua proyek tersebut akan besar. Hal ini berbeda dengan *Fuzzy Intermediate COCOMO*, biaya perkiraan yang dihasilkan sesuai dengan nilai CD. Semakin besar perbedaan nilai CD antara dua proyek, maka semakin besar pula selisih biaya perkiraan yang dihasilkan. Perbedaan kategori CD pada dua proyek tersebut tidak begitu mempengaruhi hasil pengujian.

KESIMPULAN DAN SARAN

Kesimpulan

Perkiraan biaya perangkat lunak menggunakan *Intermediate* COCOMO yang digunakan sebagai nilai input adalah kategori kelima belas CD sehingga akan menghasilkan biaya perkiraan dengan selisih yang besar pada proyek – proyek yang analogi. Begitu juga sebaliknya, pada proyek-proyek dengan nilai CD yang berbeda jauh akan menghasilkan biaya perkiraan yang sama

. Kekurangan dari metode *Intermediate* COCOMO diatasi dengan menggunakan *Fuzzy Intermediate* COCOMO. Dalam metode ini yang digunakan sebagai nilai input adalah nilai setiap CD sehingga besarnya biaya perkiraan yang dihasilkan sesuai dengan besarnya nilai CD. Semakin besar perbedaan nilai CD antara dua proyek maka semakin besar pula selisih biaya antara dua proyek tersebut.

Berdasarkan pengujian sepuluh *dataset* menggunakan *Fuzzy Intermediate* COCOMO didapatkan rata-rata nilai galat MMRE yaitu sebesar 51,95%. Nilai galat ini lebih kecil dari *Intermediate* COCOMO yaitu sebesar 59,50%.

Saran

Fuzzy Intermediate COCOMO dapat dikembangkan lebih jauh lagi dengan cara merepresentasikan ukuran perangkat lunak dan model pengembangan proyek dalam himpunan *fuzzy*.

DAFTAR PUSTAKA

- Briand, Lionel. et al. 1999. *An Assessment and Comparison of Common Software Cost Estimation Model Based on Integration of Multi-agent and Case-Based Reasoning*. Journal of Computer Science 2. ISSN, 3:276-282.
- Dickson, Gardner, 2007. *Software Cost Estimation*. Faculty of Computer Science – Faculty of Engineering, University of New Brunswick, Canada.
- Idri, Ali dan Abran, Alain. *COCOMO Cost Model Using Fuzzy Logic*. Department of Computer Science, UAQM. Montreal, Canada.
- Idri, Ali. et al. 2001. *Fuzzy Analogy: A New Approach for Software Cost Estimation*. International Workshop on Software Measurement. Montreal, Canada.
- Idri, Ali. et al. 2001. *La Logique Appliquee Aux Modeles d'Estimation de Development de Logiciels Cas du Modele COCOMO 81* : International Workshop on Software Measurement. Montreal, Canada.
- Leung, Hareton. dan Zhang Fan. 2001. *Software Cost Estimation*. Departement of Computing, The Hong Kong Polytechnic University.
- NASA Cost Estimating Web Site. 2005. <http://cost.jsc.nasa.gov/COCOMO.html>
- Promise Software Engineering Repository data set. 2004. <http://www.promisedata.org>
- Saliu, Moshood. 2003. *Adaptive Fuzzy Logic Based Framework For Software Development Effort Prediction*. King Fahd University of Petroleum & Minerals.
- Shepperd, M. dan Schofield, C. 1996. *Effort Estimation Using Analogy*. Proceedings of the 18th International Conference on Software Engineering. Berlin, Germany.
- Zadeh, L. 1996, *Fuzzy Logic: Computing With Words*, IEEE Transactions on Fuzzy Systems, Vol.4, No.2, pp. 103-111.