



## Proses Inferensi Dinamis Pada Software Effort Estimation Menggunakan Case Based Reasoning

Maria Rosario Barroek<sup>1</sup> dan Edrian Hadinata<sup>2</sup>

<sup>1</sup>STIKOM Dinamika Bangsa Jambi, email: diamar\_ros@yahoo.com

<sup>2</sup>Sekolah Tinggi Teknik Harapan Medan, email: edrianhadinata@gmail.com

### Abstrak

*Software effort estimation biasanya menjadi penentu sukses atau tidaknya kontrak negosiasi dan sepatat atau tidak sepatatnya sebuah proyek dijalankan. Keakurasian dari model estimasi biaya pengembangan software dibutuhkan untuk keefektifan prediksi, monitoring kontrol pengembangan software dan menaksir pengembangan software. Software tersebut akan menyampaikan berapa usaha yang dibutuhkan, biaya untuk menjalankan proyek, tenaga kerja yang diperlukan serta waktu kerja yang akan dihabiskan untuk menjalankannya. Penelitian ini bertujuan untuk melakukan proses inferensi secara dinamis dengan mengadopsi kemampuan Case Based Reasoning (CBR) untuk mengestimasi upaya pengembangan software. Perhitungan besaran fungsional dengan menggunakan COSMIC dataset yang dikemas dengan CBR akan memberikan hasil maksimal dalam hal menentukan kecepatan dalam proses estimasi khususnya pada tahap proposal atau penawaran proyek software dimana perhitungan lebih efisien terhadap waktu karena memiliki referensi terhadap kasus-kasus lama. Pengambilan dataset COSMIC yang sesuai berdasarkan tingkat kesamaan kasus menggunakan k-Nearest Neighbor (KNN) menghasilkan akurasi yang baik dengan perhitungan yang sederhana.*

**Kata kunci:** *software effort estimation, case based reasoning, k-nearest neighbor*

### Abstract

*Software Effort Estimation is the determinant of the success or failure of the negotiations contract and the agreement of a project which is run. The accuracy of the software development cost estimation model is required for predictive effectiveness, monitoring software development controls and estimating software development. The software will relay how much the effort it takes, the cost to run the project, the necessary workforce and the time it takes to run it. This study aims to make the inference process dynamically by adopting CBR capabilities to estimate software development efforts. The calculation of functional quantity using COSMIC dataset which is packed with CBR will give maximum result in the speed of estimation process particularly at the proposal or bid stage of software project in which the calculation is more efficient in time because it has references to the old cases. The collection of appropriate COSMIC datasets based on case-level similarity using KNN produces good accuracy with simple calculations.*

**Keywords:** *software effort estimation, case based reasoning, k-nearest neighbor*

## 1. PENDAHULUAN

*Software Effort Estimation* memainkan peranan penting dalam software engineering. Oleh sebab itu *effort estimation* biasanya menjadi penentu sukses atau tidaknya kontrak negosiasi dan sepakat atau tidak sepakatnya sebuah proyek dijalankan. Pengembang software saat ini sudah menjadi investasi penting bagi berbagai macam perusahaan, keakuratan model estimasi biaya pengembangan software dibutuhkan untuk keefektifan prediksi, monitoring kontrol pengembangan software dan menaksir pengembangan software, Saxena <sup>1</sup>. Software tersebut akan menyampaikan berapa usaha yang dibutuhkan, biaya untuk menjalankan proyek, tenaga kerja yang diperlukan serta waktu kerja yang akan dihabiskan untuk menjalankannya. Beragam variabel akan digunakan sebagai kunci utama untuk membuat proposal penawaran proyek, analisa progress, perencanaan biaya, analisis monitoring proyek dan sebagainya.

Besaran adalah bobot karakteristik yang melekat pada sebuah software dan akan digunakan untuk mengestimasi karakteristik sebuah software seperti upaya (*effort*), penjadwalan, sumber daya, kecacatan sebuah software dan sebagainya. Terdapat dua jenis pengukuran besaran software, yaitu berbasiskan besaran fungsional (*functional size*) perspektif pengguna dan besaran yang bersifat teknis (*technical size*) perspektif pengembang seperti yang telah diutarakan, Bhardwaj et al <sup>2</sup>. Pengukuran yang termasuk kedalam *functional size* dan *technical size* adalah *function point*(FP), *line of code*(KLOC atau SLOC), *Use Case Count*(UCC), *Object Count*(OC).

Kesalahan memperhitungkan sumberdaya, waktu, biaya dan besaran yang selalu berubah pada proses negosiasi akan berdampak buruk pada proses produksi terlebih lagi terhadap penentuan besaran berdasarkan kebutuhan teknologi canggih yang tidak diketahui. Untuk itu akan sulit bagi manajer proyek dalam mengestimasi besaran software dan biaya yang dibutuhkan. Ini adalah ciri-ciri kelemahan *theory domains* dimana metode Case Base Reasoning (CBR) berpotensi sebagai solusinya seperti yang telah diutarakan Delany, Sarah Jane et al <sup>3</sup>.

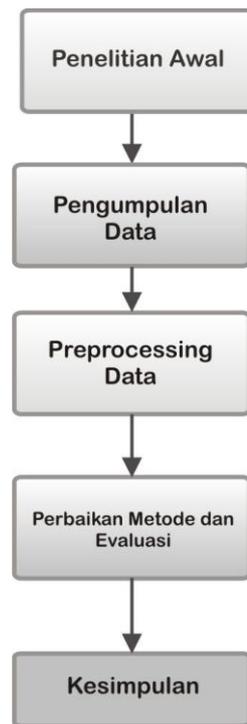
CBR menyelesaikan masalah dengan menemukan kembali permasalahan yang dinilai mirip dengan kasus sebelumnya kemudian menggunakan kembali solusi yang dimiliki kasus tersebut. Setiap pengalaman yang ada diingat sebagai kasus didalam basis kasus Sammut et al <sup>4</sup>. CBR bekerja berdasarkan pada adaptasi langsung terhadap kasus-kasus sebelumnya atas dasar kesamaan kasus tersebut dengan kasus saat ini. Dalam konteks *software effort estimation* sebuah sistem CBR didasari dengan asumsi bahwa setiap proyek software yang sama memiliki jumlah upaya(*effort*) yang sama. Oleh karena itu CBR menggunakan *k*-Nearest Neighbor (*k*-NN) dan menggunakan Euclidean Distance dalam mengecek tingkat kesamaan, Brady et al <sup>5</sup>.

Untuk itu dalam penelitian ini dikembangkan sebuah proses inferensi secara dinamis dengan menggunakan CBR yang akan diterapkan didalam bidang software engineering untuk mengestimasi upaya pengembangan software.

## 2. METODOLOGI PENELITIAN

### 2.1 Tahapan Penelitian

Dalam penelitian ini akan dilakukan beberapa tahapan kerja sebagai berikut :



Gambar 1 Tahapan Penelitian

### 1. Penelitian Awal

Pada tahapan ini dikumpulkan bahan penelitian dari berbagai sumber pustaka, seperti buku, jurnal (baik cetak maupun online), prosiding, majalah, artikel dan sumber lain yang relevan.

### 2. Pengumpulan Data

Dataset yang digunakan adalah cosmic10 yang berasal dari ISBSG, data ini merupakan data masalah tentang proyek yang telah berhasil dikerjakan software developer.

### 3. Preprocessing Data

Pada proses ini dilakukan prosedur imputasi atau teknik mengisi data kosong, cleaning serta reduksi untuk mengurangi data yang redundan jika ada.

### 4. Perbaikan Metode dan Evaluasi

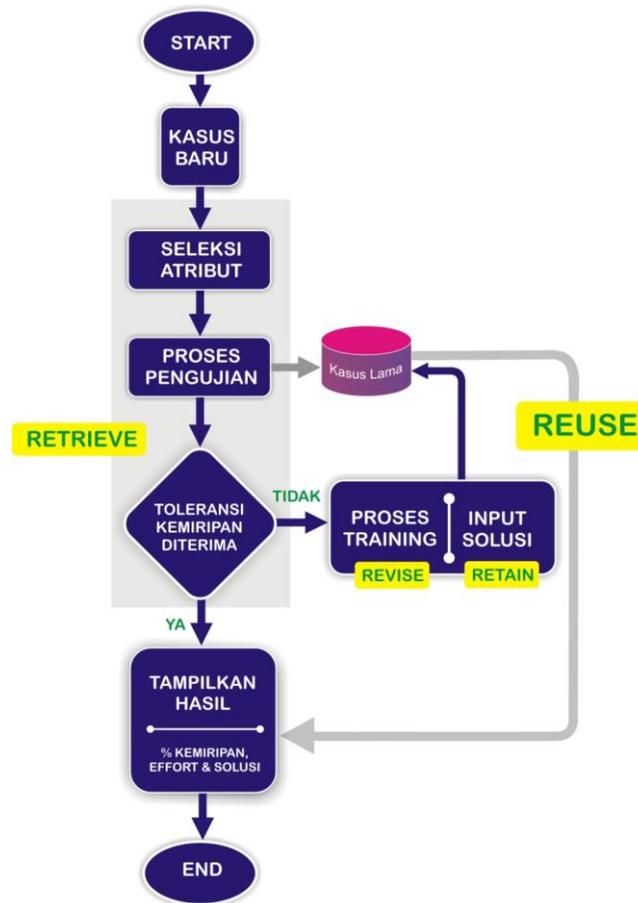
Pada tahapan ini dilakukan proses perhitungan dan perancangan aplikasi software effort estimation dengan menggunakan Metode Case Base Reasoning dengan menerapkan algoritma  $k$ -NN dengan melakukan klasifikasi berdasarkan jarak terdekat serta mengevaluasi tingkat akurasi dari hasil klasifikasi yang telah dibentuk.

### 5. Kesimpulan

Dari output yang sudah dihasilkan maka dapat diambil kesimpulan dengan cara mendokumentasikannya menjadi sebuah laporan dimulai dari teori pendukung, prosedur metodologi, pengujian dan hasil pengujian serta saran yang dapat dilakukan pada penelitian selanjutnya.

## 2.2 RANCANGAN PENELITIAN

Metode yang di ajukan pada penelitian ini diharapkan akan menghasilkan model dengan skala penelitian. Pengevaluasian serta hasil dengan akurasi yang baik adalah tolak ukur indikator pada penelitian ini yang dapat mengembangkan metode yang dapat di uji coba secara eksperimental.



Gambar 2 CBR Dalam Rancangan Penelitian

## 3. PEMBAHASAN

Penyediaan estimasi waktu yang tepat seperti pada pengembangan *software effort estimation* sudah menjadi fokus penelitian dalam ranah software engineering, secara khusus pada proyek software pada tingkat manajemen. Hasilnya adalah kemunculan beragam teknik pada pengembangan *software effort estimation*. Sebagai hasilnya jualah pada bab ini dipaparkan sebuah hasil kajian tentang pengembangan teknik tersebut.

### 3.1 Analisa Penggunaan Dataset dalam Preparasi Data

Data repository dalam bidang software engineering yang didefenisikan dengan baik sebagai sebuah set yang sangat berguna, merepresentasikan keadaan sebenarnya yang dihubungkan terhadap software proyek berisikan data kualitatif dan kuantitatif. Dataset tersebut mendeskripsikan informasi tentang sumberdaya, produk, proses, teknik, manajemen dan sebagainya. Dalam bidang keilmuan ilmiah dan engineering, data tersebut sangat berguna untuk benchmarking, eksperimen dan studi empiris.

Sejak tahun 2006 COSMIC telah mengumpulkan data dari berbagai pengembang software diseluruh dunia melalui database International Software Benchmarking Standards

Groups (ISBSG). Dari data yang ada terdapat beberapa kolom yang pada baris tertentu tidak memiliki nilai diantaranya kolom *Project Elapsed Time* (PET) dan kolom *Max Team Size* (MTS).

Pada proses selanjutnya *missing value* atau data kosong yang terdapat pada dataset akan dapat mempengaruhi nilai yang akan dihasilkan pada proses *case based reasoning*. Untuk itu proses imputasi akan dilakukan untuk meningkatkan akurasi.

### 3.2 Proses Imputasi Pada Preparasi Data

Penggunaan metode yang tepat pada proses imputasi akan dilakukan berdasarkan nilai kelengkapan pada atribut yang lain. Algoritma *k*-Nearest Neighbor (*k*-NN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut, Suguna & Thanushkodi<sup>6</sup>. Sebab itu penggunaan algoritma *K*-Nearest Neighbor (KNN) pada proses ini memperhitungkan struktur korelasi antar fieldnya. Selain itu jenis data yang terdapat pada dataset adalah integer, float dan string artinya algoritma KNN akan melakukan perhitungan kesamaan data menyertakan bobot setelah sebelumnya dilakukan normalisasi dengan berbagai macam tipe data.

Penggunaan data yang akan di jadikan data training dalam proses imputasi adalah merupakan data yang memiliki kelengkapan atribut PET dan MTS. Untuk proses pembobotan pada jenis data string diuraikan seperti pada tabel dibawah ini:

**Tabel 1 Pembobotan dan Tipe Data**

No	Nama Field	Type	Bobot
1	ISBSG Project ID	ID integer	-
2	Organisation Type (OT)	String	0.012926717
3	Application Type (AT)	String	0.032268029
4	Development Platform (DP)	String	0.060196632
5	Development Type (DT)	String	0.018950607
6	Language Type (LT)	String	0.134281725
7	Primary Programming Language (PPL)	String	0.705267983
8	Functional Size (FS)	Float	-
9	Normalised Work Effort (N_effort)	Float	-
10	Summary Work Effort (S_effort)	Float	-
11	Normalised Productivity Delivery Rate (N_PDR)	Float	-
12	Project Elapsed Time (PET)	Float	-
13	Max Team Size (MTS)	Integer	-
14	1st Operating System (OS1)	String	0.036108307

#### 3.2.1 Analisis Variansi dalam Efektifitas Perubahan *Maximum Time Size* (MTS) Terhadap Setiap Variabel

Untuk mengetahui apakah setiap variansi yang terdapat pada setiap kolom terpilih yang terdapat ISBSG dataset menyebabkan perubahan yang signifikan terhadap variabel *Maximum*

time size atau MTS, pada penelitian ini digunakan analisis variansi sebagai alat hitung dalam menentukan pengujian hipotesis.

Untuk rata-rata pada setiap populasi pada satu field atau variabel terpilih dalam ISBSG ditandai dengan  $\mu_1, \mu_2, \mu_3, \dots, \mu_k$  dan simpangan baku berturut  $\sigma^2_1, \sigma^2_2, \sigma^2_3, \dots, \sigma^2_k$ , akan diuji hipotesis nol  $H_0$  dengan tandingan  $H_1$ . Pada pengujian ini dilakukan pada semua kolom terpilih yang bertipe string seperti *Organisation Type (OT)*, *Application Type (AT)*, *Development Platform(DP)*, *Development Type(DT)*, *Language Type(LT)*, *Primary Programming Language(PPL)*, *1st Operating System (OS1)*.

**Tabel 2 Perubahan jumlah Development Type (DT) Terhadap Variasinya**

	<b>New Development</b>	<b>Enhancement</b>	<b>Re-development</b>
Data Pengamatan	5	74	1
	68	4	3
	21	7	
	30	3	
	6	4	
	5	5	
	3	1	
		6	
Jumlah	138	104	4
Jumlah data	7	8	2
rata-rata	19.71428571	13	2

**Tabel 3 Daftar Analisis Varians Untuk Pengujian Development Type (DT)**

Sumber Variasi	dk	JK	KT	F
rata-rata	1	3559.76	3559.7647	0.47859
antar kelompok	2	520.807	260.40336	
dalam kelompok	14	7617.43	544.10204	
total	17	11698	-	-

Untuk harga F didapat dari pembagian kuadrat tengah antar kelompok dengan kuadrat tengah dalam kelompok.

$$F = 260.40336/544.10204 = 0.47859$$

Dari distribusi F dengan pembilang (dk antar kelompok) 2 dan penyebut (dk dalam kelompok) 14 dan peluang 0.95 (untuk  $\alpha = 0.05$ ) didapat  $F = 3.74$ . Ternyata bahwa F-hasil yang terdapat pada tabel proses kalkulasi bernilai 0.47859 lebih kecil dari F-Tabel 3.74, jadi Hipotesis  $H_0 = \mu_1 = \mu_2 = \mu_3$  diterima dalam taraf nyata 0.05. Ketiga variasi *New Development*, *Enhancement* dan *Re-development* menyebabkan perubahan jumlah *Development Type(DT)* yang tidak berbeda secara nyata. Dengan kata lain ketiga variasi tersebut sama efektifnya sehingga variasi mana saja yang digunakan memberikan hasil yang secara nyata tidak berbeda.

Selanjutnya tentukan nilai bobot berdasarkan perhitungan perbandingan F-hasil dengan F-Tabel pada analisis variansi.

$$\text{Bobot}(w) = F\text{-hasil}/F\text{-tabel} = 0.12796601$$

Kemudian untuk kolom yang lain lakukan perhitungan yang sama seperti perhitungan diatas.

**Tabel 4 Daftar Analisis Varians Untuk Pengujian Organisation Type (OT)**

Sumber Variasi	dk	Jk	KT	F-Hasil	F-Tabel	Bobot
rata-rata	1	4034.4	4034.4	0.300274	3.44	0.08729
antar kelompok	8	1769.77	221.22083			
dalam kelompok	8	5893.83	736.72917			
total	17	11698	-	-		

Untuk pengujian OT jumlah F-Hasil lebih kecil dibandingkan dengan F-Tabel berarti dengan sembilan variasi tidak memberikan perbedaan yang secara nyata. Begitu juga dengan AT, DT, DP, LT, PPL dan OS1

Untuk pengujian PPL jumlah F-Hasil lebih besar dibandingkan dengan F-Tabel berarti dengan delapan variasi memberikan perbedaan yang secara nyata. Untuk PPL dengan besarnya F-Hasil dibandingkan F-Tabel maka Hipotesis apakah variasi didalam kolom tersebut homogen ditolak. Ini juga membuktikan bahwa perubahan yang terdapat pada salah satu variasi juga memberikan perubahan yang signifikan terhadap nilai MTS.

Setelah perhitungan-perhitungan tersebut diperoleh, secara langsung pengambilan nilai bobot dapat dipergunakan untuk menghitung dalam penentuan jumlah MTS pada data testing.

### 3.2.2 Kombinasi Algoritma K-NN Dan Persentase Pembobotan Dalam Teknik Imputasi Data

Perhitungan pembobotan diberikan agar dapat digunakan sebagai pengali terhadap nilai kesamaan antara data testing dengan data training. Jika data testing sama dengan data training maka nilainya sama dengan 1 jika berbeda maka bernilai 0. Kemudian nilai tersebut dikali dengan bobot yang terdapat pada kolom tersebut. Tetapi sebelum memberikan pembobotan nilai pembobotan terlebih dahulu harus dinormalisasi.

**Tabel 5 Daftar Hasil Normalisasi Perhitungan Bobot**

OT	AT	DT	DP	LT	OS1	PPL
0.01293	0.03227	0.01895	0.0602	0.13428	0.03611	0.70527

Setelah hasil normalisasi diperoleh selanjutnya adalah melakukan kecocokan terhadap data testing. Berikut adalah tabel kecocokan antara ID 25081 dengan ID 24302.

**Tabel 6 Daftar Hasil Kecocokan Dan Kali Bobot**

Field	Data Testing	Data Training	Hasil	Kali Bobot
OT	Banking	Banking	1	0.0129267
AT	Financial transaction process/accounting	Financial transaction process/accounting	1	0.032268

DT	New Development	New Development	1	0.0189506
DP	MF	PC	0	0
LT	3GL	3GL	1	0.134281725
PPL	Java	C++	0	0
OS1	UNIX/AIX	Windows	0	0

**Tabel 7 Tabel Perbandingan Kesamaan Jarak**

ID	FS	N_effort	S_effort	N_PDR	PET	SUM	DIST	SIM	ERROR	TRUE
25081	1386.47	40573902	40500496	159.057	11.6201	81075956	9004.22	1.11279	0.11279	0.887211288
29310	58194.5	1.64E+09	1.64E+09	44.7325	12.8965	3.28E+09	57313.1	7.08305	6.08305	-5.083053533
31397	27.4083	17599999	17648401	440.506	73.8083	35248942	5937.08	0.73374	0.26626	0.733735811
25271	19534.2	2.39E+08	2.4E+08	12.1678	19.4377	4.79E+08	21888.4	2.70508	1.70508	-0.705083223
27553	10356.1	1.52E+08	1.52E+08	991.509	12.8965	3.05E+08	17450.5	2.15662	1.15662	-0.156621295
29311	196458	7539224	7507600	47.4478	2.53184	15243331	3904.27	0.48251	0.51749	0.482510333
29471	19255.6	39699636	39627025	151.58	11.6201	79346080	8907.64	1.10085	0.10085	0.899146796
29834	37933.3	20791454	20738916	0.0972	6.7142	41568310	6447.35	0.7968	0.2032	0.79679731
32005	298373	9288870	9253764	1245.26	0.08478	18842252	4340.77	0.53645	0.46355	0.536454729
32064	12048.3	39009577	38937600	237.522	12.8965	77959475	8829.47	1.09119	0.09119	0.908808115
27934	80222.2	36273695	36204289	16.0943	31.2613	72558253	8518.11	1.05271	0.05271	0.947286765
30333	27477.9	46060175	46457856	240.615	11.6201	92545761	9620.07	1.1889	0.1889	0.81110106
31662	12942.4	33475073	33408400	144.282	11.6201	66896572	8179.03	1.01081	0.01081	<b>0.989192155</b>
30658	37933.3	38263685	38192400	146.695	11.6201	76494177	8746.09	1.08089	0.08089	0.919111589
30911	39110.9	48021639	47941776	106.332	37.9311	96002670	9798.1	1.2109	0.2109	0.78909986
27505	6685.47	30049744	29986576	0.01249	5.80243	60043012	7748.74	0.95763	0.04237	0.957630362
31895	13868.5	36805634	37503376	102.248	0.34949	74322981	8621.08	1.06544	0.06544	0.934561873

Untuk tabel perhitungan dapat dilihat pada uraian dibawah ini.

**Tabel 8 Tabel Global Persentase Kesamaan terhadap ID 24302**

ID	STR SIM	TRUE	GLOBAL	MTS
25081	0.198427078	0.8872113	0.5428192	5
29310	0.018950607	-	-	68
31397	0.858500316	0.7337358	0.7961181	21
25271	0.045194746	-	-	74
27553	0.858500316	-	-	30
29311	0.858500316	0.4825103	0.6705053	6

29471	0.858500316	0.8991468	0.8788236	5
29834	0.194478357	0.7967973	0.4956378	4
32005	0.839549708	0.5364547	0.6880022	7
32064	0.194478357	0.9088081	0.5516432	3
27934	0.134281725	0.9472868	0.5407842	1
30333	0.194478357	0.8111011	0.5027897	4
31662	0.858500316	0.9891922	0.9238462	3
30658	0	0.9191116	0.4595558	5
30911	0	0.7890999	0.3945499	1
27505	0.012926717	0.9576304	0.4852785	3
31895	0.134281725	0.9345619	0.5344218	6

Untuk beberapa nilai PET yang masih kosong dilakukan perhitungan Analysis of variance terhadap beberapa atribut seperti perhitungan anova pada kolom MTS. Tabel hasil perhitungan Analysis varian PET terhadap beberapa kolom adalah sebagai berikut.

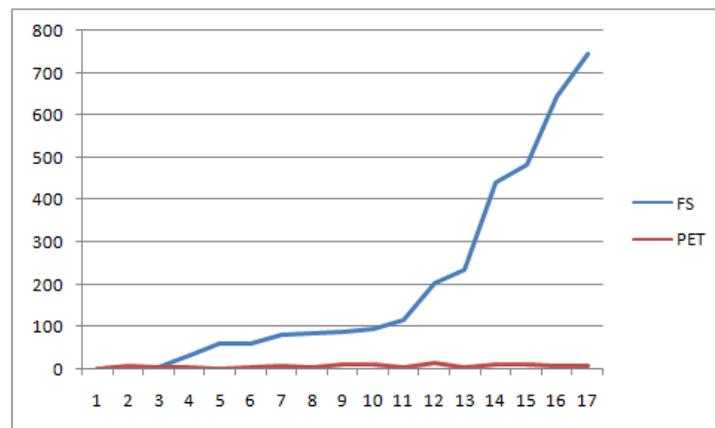
**Tabel 9 Hasil Analisis Variansi**

No	Nama Field	Type	Hasil	Bobot
1	ISBSG Project ID	ID integer	-	-
2	Organisation Type (OT)	String	Terima $H_0$	0.01523
3	Application Type (AT)	String	Terima $H_0$	0.103
4	Development Platform (DP)	String	Terima $H_0$	0.032
5	Development Type (DT)	String	Terima $H_0$	0.1034
6	Language Type (LT)	String	Terima $H_0$	0.1325
7	Primary Programming Language (PPL)	String	Terima $H_0$	0.264

Tidak ada perubahan yang signifikan terhadap variasi yang terdapat pada setiap kolom seperti yang terlihat pada tabel diatas, artinya apa pun variasi yang dipilih tidak menghasilkan perubahan yang signifikan terhadap nilai PET.

Untuk membuktikan hal tersebut dilakukan perhitungan pencarian PET terhadap nilai yang ada berdasarkan perhitungan COSMIC CFP. Dimana untuk C adalah sebuah konstanta 0.543 dan E1 sebuah konstanta bernilai 0.408.

Dengan menggunakan gambar menunjukkan perbandingan besaran PET dengan FS, berikut seperti gambar dibawah ini:



**Gambar 3 Perbandingan Besaran PET Terhadap FS**

Perbedaan antar estimasi PET dengan PET tersebut mungkin disebabkan oleh tertundanya proyek pengerjaan software oleh hal-hal tertentu. Untuk itu proses imputasi nilai PET pada data testing menggunakan nilai rata-rata PET yang terdapat pada data Training. Setelah nilai PET yang missing value di isi dengan rata-rata, kemudian lakukan perhitungan untuk menentukan nilai MTS kembali sehingga nilai MTS yang kosong dapat terisi semua.

Hitung nilai SDR dengan menggunakan perhitungan COSMIC :

$$SDR = C \times Size^{EI} \quad (1)$$

$$SDR = 6.368576216$$

### 3.3 Implementasi Case Based Reasoning

Metode ini merupakan satu model pola *problem solving* yang mendapatkan banyak pengakuan yang pada dasarnya berbeda dari model *Artificial Intelligence* (AI) lainnya. Sebuah masalah baru di selesaikan dengan menemukan kasus yang sama di masa yang lampau. Hal yang paling menarik lainnya dari CBR adalah metode ini merupakan metode pembelajaran yang meningkat secara terus menerus dan secara proses terdapat empat proses yang dapat di representasikan secara silindrikal, Aamodt & Plaza <sup>7</sup>.

Setelah proses imputasi dilakukan, Proses Estimasi dengan menggunakan Case Based Reasoning atau CBR sudah siap untuk dimulai. Penggunaan Case Based Reasoning atau CBR pada sistematika penulisan ini terbagi beberapa bagian, yaitu : Retrieve, Reuse, Revise dan Retain.

#### 3.3.1 Retrieve

Proses retrieve pada CBR dilakukan dengan memasukkan kasus baru sesuai dengan variabel yang telah ditentukan berdasarkan variabel yang terdapat di dataset. Variabel-variabel yang disertakan dalam penelitian ini adalah ID, OT, AT, DT, DP, LT, PPL, FS, N\_effort, S\_effort, N\_PDR, SDR, PET, MTS dan OS1.

Proses pengujian pada bagian retrieve dalam CBR hanya menggunakan beberapa variabel saja, diantaranya adalah Organisation Type(OT), Application Type(AT), Development Type(DT), Development Platform (DP), Language Type (LT), Primary Language (PPL), Project Elapsed Time (PET) dan Operating System (OS1). Penggunaan sebagian variabel string dalam

proses retrieve disebabkan oleh orientasi sistem yang dibuat akan dipergunakan sebagai pengestimasi upaya dalam pembuatan software pada tahap penawaran sehingga tidak menggunakan perhitungan yang terlalu mendalam.

Proses retrieve pada CBR dilakukan dengan memasukkan kasus baru sesuai dengan variabel yang telah ditentukan berdasarkan variabel yang terdapat di dataset. Variabel-variabel yang disertakan dalam penelitian ini adalah ID, OT, AT, DT, DP, LT, PPL, FS, N\_effort, S\_effort, N\_PDR, SDR, PET, MTS dan OS1.

Proses pengujian pada bagian retrieve dalam CBR hanya menggunakan beberapa variabel saja, diantaranya adalah Organisation Type(OT), Application Type(AT), Development Type(DT), Development Platform (DP), Language Type (LT), Primary Language (PPL), Project Elapsed Time (PET) dan Operating System (OS1). Penggunaan sebagian variabel string dalam proses retrieve disebabkan oleh orientasi sistem yang dibuat akan dipergunakan sebagai pengestimasi upaya dalam pembuatan software pada tahap penawaran sehingga tidak menggunakan perhitungan yang terlalu mendalam.

Jika diketahui data testing dari sebuah kasus seperti tabel dibawah ini:

**Tabel 10 Data Testing Studi Kasus**

No	Nama Field	Data Testing
1	Organisation Type (OT)	Medical & Health Care
2	Application Type (AT)	Online analysis and reporting
3	Development Platform (DP)	Multi
4	Development Type (DT)	New Development
5	Language Type (LT)	4GL
6	Primary Programming Language (PPL)	Java
7	Operating System (OS1)	Windows XP
8	Project Elapsed Time (PET)	7 Months

Maka proses yang pertama pada proses retrieve adalah menemukan kesamaan data testing pada kasus yang ada dengan data training yang tersedia.

Tentukan jumlah pembobotan yang diinginkan sesuai dengan prioritas kepentingan untuk mendahulukan pencarian pada variabel-variabel tertentu. Dalam penelitian ini prioritas ditentukan pada tabel berikut ini.

**Tabel 11 Penentuan Bobot Prioritas Setiap Variabel**

No	Nama Field	Bobot Prioritas
1	Organisation Type (OT)	1
2	Application Type (AT)	2
3	Development Platform (DP)	1
4	Development Type (DT)	1
5	Language Type (LT)	1
6	Primary Programming Language (PPL)	2
7	Operating System (OS1)	1
8	Project Elapsed Time (PET)	3

Langkah selanjutnya adalah menghitung kedekatan antara data testing dengan data training dengan menggunakan algoritma *k-nearest neighbor retrieval*.

$$\text{Similarity}(\text{Kasus}, \text{ID24302}) = \frac{0*1+0*2+1*1+0*1+0*1+1*2+0*3+0*1}{1+2+1+1+1+2+1+3}$$

$$\text{Similarity}(\text{Kasus}, \text{ID24302}) = 0.25$$

Kemudian dengan menggunakan persamaan algoritma *k-nearest neighbor retrieval* tentukan nilai kedekatan untuk semua ID. Tabel hasil perhitungan ditampilkan berikut ini.

**Tabel 12 Pengujian Kesamaan**

ID	OT	AT	DT	DP	LT	PPL	PET	OSI	SIM
24302	0	0	1	0	0	2	0	0	0.25
25081	0	0	1	0	0	0	0	0	0.083333
28884	0	0	1	1	1	0	0	0	0.25
29310	0	0	1	0	1	0	0	1	0.25
29604	0	0	0	0	1	0	0	0	0.083333
30611	0	0	1	0	0	2	0	0	0.25
31240	1	0	1	1	0	0	0	0	0.25
31397	0	0	1	1	0	2	0	0	0.333333
25271	0	0	0	0	1	0	0	0	0.083333
27553	0	0	1	0	0	2	0	0	0.25
29311	0	0	1	1	0	2	0	0	0.333333
29331	0	0	1	1	0	2	0	0	0.333333
29471	0	0	1	1	0	2	0	0	0.333333
29834	0	0	0	0	0	0	0	0	0
29893	0	2	0	0	0	0	0	0	0.166667
31333	1	0	0	0	0	2	0	0	0.25
31512	0	0	1	0	0	0	0	0	0.083333
32005	0	0	0	0	0	2	0	0	0.166667
32064	0	0	0	0	0	0	0	0	0
27934	0	0	0	0	0	0	0	0	0
30333	0	2	0	0	0	0	0	0	0.166667
31448	0	0	1	0	0	0	0	1	0.166667
31662	0	0	1	0	0	2	0	0	0.25
32081	0	0	0	0	0	2	0	0	0.166667
27917	0	0	0	1	0	0	0	0	0.083333
28026	0	0	0	0	0	2	0	0	0.166667
30658	0	2	0	1	1	0	0	0	0.333333
30677	0	0	0	1	1	0	0	0	0.166667
30719	0	2	0	1	1	0	0	0	0.333333
30879	0	0	0	1	1	0	0	0	0.166667
30911	0	2	0	1	1	0	0	0	0.333333

31451	0	0	1	0	0	2	0	0	0.25
27505	0	2	0	1	1	0	0	0	0.333333
27537	0	0	1	0	0	2	0	0	0.25
27547	0	2	1	1	1	0	0	0	0.416667
27913	0	0	1	0	0	0	0	0	0.083333
31895	0	0	0	0	0	0	3	1	0.333333
MAX									0.416667

Pencarian jumlah kesamaan yang terbesar sesuai tabel telah ditemukan dengan besar persentase kesamaan 0.416667 yang terdapat pada ID 27547. Maka proses selanjutnya adalah menampilkan informasi pada data training yang terpilih tersebut.

**Tabel 13 Data Training ID Terpilih**

No	Nama Field	Data Training
1	ISBSG Project ID	27547
2	Organisation Type (OT)	Banking
3	Application Type (AT)	Online analysis and reporting
4	Development Platform (DP)	Multi
5	Development Type (DT)	New Development
6	Language Type (LT)	4GL
7	Primary Programming Language (PPL)	Script Language
8	Operating System (OS1)	UNIX/AIX
9	Project Elapsed Time (PET)	3

Proses ini adalah proses terakhir yang terdapat pada tahap retrieve dalam CBR.

### 3.3.2 Reuse

Pada tahapan reuse, proses mempergunakan kasus yang lama dilakukan, berdasarkan tingkat kedekatan yang paling cocok. Proses ini dilakukan agar memberikan solusi terbaik dari kasus yang lama. Hasil proses tersebut ditampilkan pada tabel dibawah ini.

**Tabel 14 Data Training Solusi Terbaik**

No	Nama Field	Data Training
1	ISBSG Project ID	27547
2	Organisation Type (OT)	Banking
3	Application Type (AT)	Online analysis and reporting
4	Development Platform (DP)	Multi
5	Development Type (DT)	New Development
6	Language Type (LT)	4GL
7	Primary Programming Language (PPL)	Script Language
8	Operating System (OS1)	UNIX/AIX
9	Project Elapsed Time (PET)	3

<b>10</b>	<b>Functional Size (FS)</b>	<b>175</b>
<b>11</b>	<b>Normalisation Effort (N_Effort)</b>	<b>3633</b>
<b>12</b>	<b>Summary Effort (S_Effort)</b>	<b>3306</b>
<b>13</b>	<b>Normalisation PDR (N_PDR)</b>	<b>9.9</b>
<b>14</b>	<b>Maximum Team Size (MTS)</b>	<b>3</b>
<b>15</b>	<b>Speed of Delivery (SDR)</b>	<b>13.06</b>

Dari tabel pada perhitungan sebelumnya didapat nilai kesamaan tertinggi adalah 0.416667 atau tingkat kepercayaan sebesar 41.6667%, jika perhitungan tidak memperhitungkan batas error maksimum maka solusi 41.6667% tersebut akan diterima sebagai solusi yang terbaik. Jika menggunakan batas error, misal 90% maka proses revise akan dilakukan dalam rangka memperbaiki solusi yang ada.

### 3.3.3 Revise

Pada tahapan revise proses tinjau kasus kembali dilakukan. Jika prosedur penggunaan batas error maksimum  $\alpha = 0.1$ , atau tingkat kepercayaan yang di izinkan adalah sebesar 90% maka solusi yang diperoleh pada perhitungan sebelumnya sebesar 41.6667% ditolak.

Oleh karena itu, tahapan revise akan memperbaiki solusi yang diberikan agar relevan dengan kasus yang baru tersebut. Caranya adalah dengan melakukan perhitungan untuk mencari *effort estimation* sesuai dengan prosedur yang ada.

Dari perhitungan COSMIC Function Points, CFP yang didapat sebesar 73 CFP, hasil seperti tertera pada tabel dibawah ini:

**Tabel 15 Jumlah Entry, Read, Write, Exit dan Total Functional Size**

Entry	Read	Write	Exit	CFP
11	24	16	22	73

Perhitungan jumlah *Entry* diperoleh dari berapa jumlah form yang ada pada aplikasi termasuk form edit, login dan masukan yang diproses didalam sistem, jumlah *Read* didapat dari berapa jumlah query yang mengambil data hingga diproses didalam sistem, jumlah *Write* diperoleh dengan menghitung jumlah query yang memproses data dari sistem hingga disimpan didalam database. Untuk nilai *Exit* diperoleh dengan menghitung jumlah proses yang terjadi yang terdapat pada sistem hingga ditampilkan kehalaman aplikasi.

Proses selanjutnya adalah menghitung sumber daya yang diperoleh dari ISBSG dengan hanya menggunakan ukuran fungsionalitas(FP) untuk proyek pengembangan software tanpa melihat platform dan bahasa pemograman yang digunakan. Dengan Max team size = 3

#### a. Project Work Effort

$$\text{Project Work Effort} = 23.25 \times 73^{0.814}$$

$$\text{Project Work Effort} = 764.1309 \text{ hours}$$

$$\text{Jika ingin mengubah ke N\_PDR} = 764.1309/73 = 10.46754665 \text{ hours/CFP}$$

#### b. Project Duration

$$\text{Duration} = 0.543 \times 73^{0.408}$$

$$\text{Duration} = 3.126335316 \text{ Months}$$

### c.Speed of Delivery(SDR)

$$\text{Speed of Delivery} = 1.842 \times 73^{0.592}$$

$$\text{Speed of Delivery} = 23.35483262 \text{ FP/Months}$$

Untuk perhitungan Normalisation Effort atau N\_Effort dan juga S\_Effort belum dapat dilakukan karena perhitungan tersebut ada berdasarkan laporan dari masing-masing anggota tim ketika proses pengerjaan software sedang berjalan. Dari perhitungan diatas digabungkan dengan data pengujian, maka diperoleh tabel dibawah ini.

**Tabel 16 Perubahan Data Training Pada Tahapan Revise**

No	Nama Field	Data Training
1	ISBSG Project ID	27547
2	Organisation Type (OT)	Medical & Health Care
3	Application Type (AT)	Online analysis and reporting
4	Development Platform (DP)	Multi
5	Development Type (DT)	New Development
6	Language Type (LT)	4GL
7	Primary Programming Language (PPL)	Java
8	Operating System (OS1)	Windows XP
9	Project Elapsed Time (PET)	3
10	Functional Size (FS)	73
11	Normalisation Effort (N_Effort)	-
12	Summary Effort (S_Effort)	-
13	Normalisation PDR (N_PDR)	10.46755
14	Maximum Team Size (MTS)	3
15	Speed of Delivery (SDR)	23.35483

Setelah dilakukan proses perhitungan, nilai yang dihasilkan diharapkan dapat menjadi solusi pada kasus yang serupa.

### 3.3.4 Retain

Ketika proses pencarian solusi terbaik pada proses revise selesai dilakukan, maka data pada tabel 17 dapat dimasukkan sebagai data training dalam basis data untuk menambah jumlah solusi yang ada. Untuk menggunakannya lakukan tahapan retrieve dengan memasukkan variabel dan nilai yang tertera pada tabel17 tersebut

## 3.4 Hasil Akhir Analisis

Dari beberapa pengujian yang dilakukan disimpulkan bahwa proses inferensi dinamis pada software effort estimation menggunakan CBR bergantung kepada ketersediaan dan

keadaan nilai dari pada dataset tersebut. Nilai kosong atau missing value yang ditandai dengan “?” pada ISBSG dataset khususnya COSMIC dataset membuat perhitungan algoritma  $k$ -NN dan  $K$ -NN Retrieval tidak memiliki akurasi yang baik. Proses imputasi data terhadap dataset dilakukan dengan menggunakan  $k$ -NN Retrieval dan  $k$ -NN dengan *Euclidean Distance* menghasilkan tingkat akurasi yang baik. Penggunaan rasio perbandingan F-Tabel dengan F-Hasil pada analisis variansi dapat dijadikan pembobotan dalam algoritma  $K$ -NN Retrieval. Penentuan bobot untuk menentukan prioritas pada tahap retrieve bergantung kepada kebutuhan developer tersebut. Untuk pengujian terhadap metoda menggunakan MAE sebesar 0.8775.

Penentuan Effort Estimation dan besaran software tahap awal *development life cycle* dengan menggunakan tahapan CBR lebih efektif dibandingkan menghitung besaran software berdasarkan tolak ukur COSMIC CFP secara langsung.

### 3.5 Kesimpulan dan Saran

#### 3.5.1 Kesimpulan

Perhitungan besaran fungsional dengan menggunakan COSMIC dataset yang dikemas dengan Case Based Reasoning (CBR) akan memberikan hasil maksimal dalam hal kecepatan dalam proses estimasi khususnya pada tahap proposal atau penawaran proyek software dimana perhitungan lebih efisien terhadap waktu karena memiliki referensi terhadap kasus-kasus lama. Pengambilan dataset yang sesuai berdasarkan tingkat kesamaan kasus menggunakan  $k$ -Nearest Neighbor menghasilkan akurasi yang baik dengan perhitungan yang sederhana. Penggunaan dataset COSMIC dalam perhitungan tidak dapat dilakukan secara langsung disebabkan oleh banyaknya missing value yang akan mengganggu akurasi perhitungan. Proses Imputasi data pada saat preparasi dirasa penting untuk menghindari kesalahan-kesalahan pada proses analisis begitu juga proses normalisasi dataset sangat efektif untuk data yang memiliki berbagaimacam tipe data, satuan dan skala yang akan dikalkulasi menggunakan satu algoritma.

#### 3.5.2 Saran

Proses imputasi data dilakukan lebih hati-hati disebabkan oleh berbagai macam variabel pada kolom yang lain mungkin saja sangat berpengaruh terhadap besaran nilai tertentu. Perlunya pengujian statistik untuk memeriksa korelasi diantaranya. Penggunaan dataset COSMIC pada ISBSG beberapa kolom tidak relevansi terhadap nilai estimasinya. Disebabkan oleh data yang disubmit kedalam dataset adalah laporan data asli pada waktu pengerjaan.

### Daftar Pustaka

- [1] Saxena, Urvashi Rahul & Singh, SP, Software Effort Estimation Using Neuro-Fuzzy Approach, *Software Engineering (CONSEG), CSI Sixth International Conference on*, IEEE Publisher, pp 1 – 6, 2012
- [2] Bhardwaj, Mridul. & Ajay, Rana, Estimation of Testing and Rework Effort for Software Development Projects, *Asian Journal of Computer Science and Information Technology*. Vol 5, No 5, pp 33 – 37, 2015
- [3] Delany, Jane Sarah et al., The Limit of CBR in Software Project Estimation, *Germany Workshop on Case-Based Reasoning*, pp 1-10, 1998,
- [4] Sammut, C. & Webb, G.I, *Encyclopedia of Machine Learning*,. Springer, 2011
- [5] Brady, Adam et al, , Case-Based Reasoning for Reducing Software Development Effort, *Journal Software Engineering & Application (JSEA)*, Vol 3, pp 1005 – 1014, 2010

- [6] Suguna, N. and Thanushkodi, K, An Improved k-Nearest Neighbor Classification Using Genetic Algorithm, *IJCSI International Journal of Computer Science Issues*, Vol. 7, Issue 4, No 2, pp 18 – 21, 2010
- [7] Aamodt, A. and Plaza, E., Case-based reasoning: foundational issues, methodological variations and system approaches, *Artificial Intelligence Communications*, Vol 7, No.1, p.3059, 1994

