

## **BAB II**

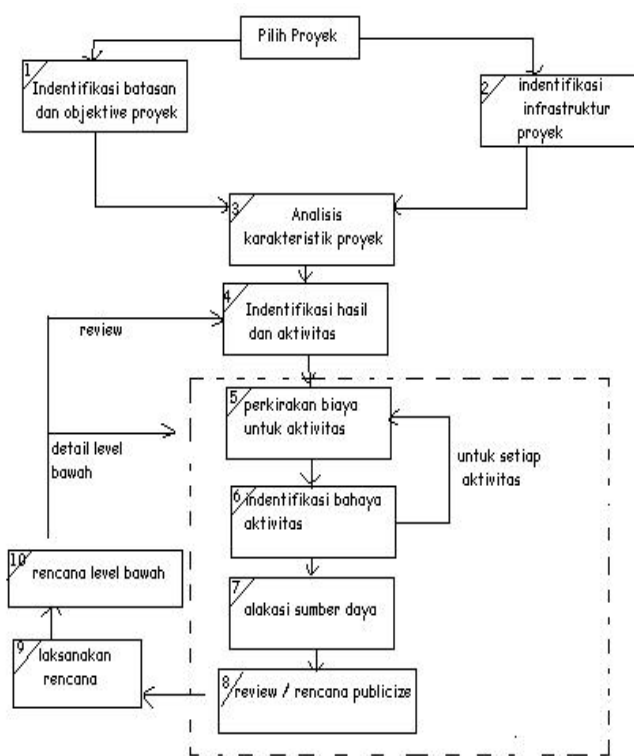
### **TINJAUAN PUSTAKA**

#### **2.1 Manajemen Proyek**

Proyek adalah sebuah usaha yang dilakukan secara sementara untuk menghasilkan sebuah produk, layanan, atau hasil yang unik (Schwalbe, 2010, pp 4). Manajemen proyek adalah penerapan dari pengetahuan, keahlian, alat, dan teknik untuk kegiatan proyek agar sesuai dengan kebutuhan proyek (Schwalbe, 2010, pp 10). Suatu proyek akan dikatakan sukses (Schwalbe, 2010, pp 14-15), jika :

1. Proyek selesai tepat waktu, sesuai dengan anggaran yang ditetapkan, dan sesuai dengan kualitas yang diinginkan.
2. Pelanggan atau sponsor puas dengan hasil proyek.
3. Hasil dari proyek sesuai dengan tujuan utama proyek tersebut.

Tahapan manajemen proyek dibagi menjadi lima kelompok proses yang terdiri dari proses inisialisasi, proses perencanaan, proses eksekusi, proses pemantauan dan pengendalian, dan proses penutupan (Schwalbe, 2010, pp 78-79).



**Gambar 2.1. Langkah-Langkah Manajemen Proyek *Software* (Hughes et al., 1999)**

## 2.2 Aplikasi Web

Aplikasi web merupakan suatu aplikasi yang dibuat dengan mengintegrasikan berbagai jenis elemen yang berbeda seperti komponen (contohnya DCOM, OLE, ActiveX), *interpreted scripting language*, komponen luar (seperti produk *third-party*, komponen *library*), file multimedia (seperti suara, video, objek 3D), file HTML/SGML/XML, gambar, campuran dari HTML dan program, dan database (Reifer, 2000) (Deshpande dan Hansen, 2001) (Offutt, 2002).

Berdasarkan tipenya, aplikasi web dapat dibedakan menjadi dua kategori yaitu aplikasi *hypermedia* berbasis web dan aplikasi *software* berbasis web (Christodoulou, et al., 2000). Mendes (2006) menambahkan satu jenis aplikasi web yang merupakan gabungan dari aplikasi *hypermedia* berbasis web dan aplikasi *software* berbasis web.

Aplikasi *hypermedia* berbasis web adalah aplikasi non konvensional yang memiliki ciri penyediaan informasi menggunakan node-node yang berisi informasi dan link antar node yang dikaitkan dengan struktur navigasi tertentu yang bisa diakses dalam jaringan web. Teknologi yang biasanya digunakan dalam aplikasi ini adalah HTML, *javascript* dan multimedia. Selain itu, pengembang yang biasanya terlibat adalah penulis artikel, seniman dan organisasi yang ingin mempublikasikan informasinya melalui web dan atau CD-ROM tanpa membutuhkan bahasa pemrograman seperti java. Aplikasi ini mempunyai potensi yang cukup besar dalam bidang rekayasa piranti lunak (Fielding dan Taylor, 2000), dan pendidikan (Michau et al, 2001).

Aplikasi *software* berbasis web merepresentasikan aplikasi software konvensional yang bergantung pada web atau menggunakan infrastruktur web untuk menjalankannya. Aplikasi dasarnya mencakup system informasi baku seperti sistem basis data, sistem pemesanan, sistem berbasis pengetahuan dan lain-lain. Berbagai aplikasi *e-commerce* termasuk dalam kategori ini. Biasanya aplikasi ini menggunakan teknologi pengembangan (seperti DCOM, ActiveX, dll), sistem basis data dan solusi pengembangan (seperti J2EE).

*Object Oriented Web Solutions* (OOWS) (Pastor et al., 2005) adalah sebuah metode *model-driven* untuk mengembangkan aplikasi web. Metode ini mengintegrasikan desain navigasi dan presentasi dengan pendekatan konseptual OO. Konseptual model tersebut berdasarkan pada lima pandangan yaitu *object*, *dynamic*, *functional*, *navigational*, dan *presentation*. Kelima pandangan tersebut menggambarkan fungsionalitas dari aplikasi web.

### 2.3 Metode Pengukuran Web

Berdasarkan cara mengukurnya, pendekatan untuk mengukur ukuran suatu *software* terbagi menjadi dua (Gencel, et al., 2006), yaitu :

1. Dengan menggunakan atribut panjang *code*. Panjang code bisa diukur dengan menggunakan *line of code* (LOC) atau menggunakan jumlah karakter, dan lainnya. Kekurangan dari LOC adalah LOC bersifat bergantung pada bahasa pemrograman yang digunakan sehingga program yang dibuat dengan bahasa pemrograman yang berbeda, tidak bisa dibandingkan secara langsung (Gencel, et al., 2006).
2. Dengan menggunakan jumlah fungsionalitas. Pendekatan ini pertama kali diteliti oleh Allan Albrecht pada tahun 1979. Albrecht mendesain metrik *Function Point* (FP) dan menghubungkannya dengan FPA untuk mengukur ukuran fungsional dari suatu sistem perangkat lunak (Albrecht, 1979). FPA ini kemudian dikembangkan lagi dan dipelihara oleh International Function Point User Group (IFPUG) (IFPUG, 1999).

### 2.3.1 Metrik Web

Penelitian terhadap metrik pengukuran aplikasi web telah dilakukan oleh Cowderoy (1998), Mendes et al. (1999), Cowderoy (2000), Mendes et al. (2000), Reifer (2000), Cleary (2000), Mendes et al. (2001). Cowderoy (1998;2000), Reifer (2000) dan Cleary (2000) hanya menggunakan data dari satu perusahaan web, yang mungkin akan mempengaruhi validasi eksternal dari hasil mereka (Mendes et al., 2003). Mendes (2000) mengusulkan metrik untuk aplikasi hypermedia berbasis web. Mendes et al. (2001) mengusulkan metrik untuk aplikasi web yang statis dan dinamis.

Mendes et al. (2003) melakukan survei terhadap 133 perusahaan web yang kemudian hasil survei tersebut divalidasi dengan survei ulang terhadap 32 perusahaan di New Zealand, mendapatkan hasil :

**Tabel 2.1 Metrik web (Mendes et al., 2003)**

Kategori	Metrik
Metrik aplikasi web statis	1. Jumlah Halaman Web
	2. Jumlah Gambar / Foto (termasuk icon dan button)
	3. Jumlah animasi (flash/gif/3D dll)
	4. Jumlah audio/video
Metrik aplikasi web dinamis	Fitur / Fungsionalitas
<i>Cost Drivers</i>	1. Aplikasi baru atau penambahan
	2. Pengalaman dari tim pengembang
	3. Alat yang digunakan
Metrik Proyek Web	1. Anggaran untuk proyek
	2. Estimasi tanggal terakhir proyek
	3. Tipe aplikasi web
	4. Estimasi tanggal awal proyek
	5. Apakah pelanggan akan memberikan <i>mock-up</i> halaman ?
	6. Apakah aplikasi akan diterjemahkan ke idiom yang berbeda ?
	7. Apakah pelanggan akan memberikan peta website ?

	8. Akses internet seperti apa yang dimiliki oleh pelanggan ?
Metrik perusahaan web	1. Deskripsi dari bisnis pelanggan
	2. Target pelanggan dari perusahaan
Metrik desain tampilan antarmuka dari web	1. Gaya yang diinginkan untuk site yang ingin dibuat
	2. Tipe warna untuk website
	3. Tipe warna untuk latar belakang
	4. Tipe warna untuk halaman

Dari hasil survei yang dilakukan oleh Mendes et al (2003), semua perusahaan yang disurvei memilih kategori metrik aplikasi web statis dan metrik aplikasi web dinamis, tidak semua perusahaan memilih kategori *cost drivers*, dan tidak ada perusahaan yang memilih kategori metrik proyek web, metrik perusahaan web, dan metrik desain tampilan antarmuka dari web. Dari hasil survei tersebut, Mendes (2003) menentukan variabel-variabel untuk mengumpulkan data aplikasi web. Variabel-variabel yang digunakan oleh Mendes adalah :

**Tabel 2.2. Variabel-variabel yang diajukan oleh Mendes et al (2003)**

Nama Variabel	Scale	Deskripsi
Country	Nominal	Negara tempat perusahaan berada
Established	Ordinal	Berapa lama perusahaan berdiri
Services	Nominal	Jenis layanan yang diberikan oleh perusahaan

ClientInd	Nominal	Jenis industri dari klien
TypeProj	Nominal	Tipe proyek
TypeApp	Nominal	Tipe aplikasi web yang dikembangkan
Languages	Nominal	Bahasa implementasi yang digunakan
Nlang	Ratio	Jumlah bahasa berbeda yang digunakan
DocProc?	Nominal	Apakah proyek diikuti dengan proses definisi dan dokumentasi?
ProcImpr?	Nominal	Apakah tim developer ikut dalam proses peningkatan program?
Metrics?	Nominal	Apakah tim developer bagian dari program software metrics.
Devteam	Ratio	Ukuran dari tim developer
Teamexp	Ratio	Rata-rata pengalaman tim dengan bahasa yang digunakan
Webpages	Ratio	Jumlah halaman web
newWP	Ratio	Jumlah halaman web baru
Wpcustom	Ratio	Jumlah halaman web yang diberikan oleh pelanggan
Wpout	Ratio	Jumlah halaman web yang dikembangkan oleh orang lain
wpOwnCo	Ratio	Jumlah halaman web yang digunakan kembali dari data perusahaan yang lama
txtTyped	Ratio	Jumlah halaman teks yang diketik (~600 kata)
txtElec	Ratio	Jumlah format elektronik dari halaman teks



txtScan	Ratio	Jumlah halaman teks yang di- <i>scan</i>
imgNew	Ratio	Jumlah gambar baru
Img3rdP	Ratio	Jumlah gambar yang dikembangkan oleh orang lain
imgScan	Ratio	Jumlah gambar yang di- <i>scan</i>
imgLib	Ratio	Jumlah gambar yang digunakan kembali dari <i>library</i>
imgOwnCo	Ratio	Jumlah gambar yang digunakan kembali oleh perusahaan sendiri
Animnew	Ratio	Jumlah animasi baru
animLib	Ratio	Jumlah animasi yang digunakan kembali dari sebuah <i>library</i>
AVNew	Ratio	Jumlah audio/video baru
AVLib	Ratio	Jumlah audio/video yang digunakan kembali
TotDiffPro	Ratio	Jumlah produk yang ditawarkan oleh aplikasi
HeffDev	Ratio	<i>Effort</i> yang dianggap tinggi untuk mengembangkan sebuah fungsi/fitur oleh seseorang
HeffAdpt	Ratio	<i>Effort</i> yang dianggap tinggi untuk mengadaptasikan sebuah fungsi/fitur oleh seseorang
Hfots	Ratio	Jumlah fitur/fungsi dengan <i>effort</i> tinggi yang digunakan kembali tanpa adaptasi
Hfotsa	Ratio	Jumlah fitur/fungsi dengan <i>effort</i> tinggi yang

		digunakan kembali
Hnew	Ratio	Jumlah fitur/fungsi dengan <i>effort</i> tinggi
Tothigh	Ratio	Total jumlah fitur/fungsi dengan <i>effort</i> tinggi
Fots	Ratio	Jumlah fitur/fungsi dengan <i>effort</i> rendah yang digunakan kembali tanpa adaptasi
Fotsa	Ratio	Jumlah fitur/fungsi dengan <i>effort</i> rendah yang digunakan kembali
New	Ratio	Jumlah fitur/fungsi dengan <i>effort</i> rendah
Totnhigh	Ratio	Total jumlah fitur/fungsi dengan <i>effort</i> rendah
Toteffor	Ratio	Total jumlah <i>effort</i> yang dikembangkan di aplikasi
Accuracy	Nominal	Prosedur yang digunakan untuk menyimpan data <i>effort</i>

Berdasarkan data-data yang sudah dikumpulkan pada tahun 2004, Mendes melakukan penelitian untuk model estimasi biaya dengan menggunakan *stepwise regression* untuk proyek web dengan menggunakan subset dari variabel-variabel pada tabel 2.3 (Mendes et al., 2004). Pada penelitian tersebut, Mendes juga melakukan perbandingan antara menggunakan data dari perusahaan sendiri untuk melakukan estimasi biaya dengan menggunakan data dari perusahaan-perusahaan lain untuk melakukan estimasi biaya. Pada tahun 2005, Mendes menginvestigasi pengukuran ukuran web pada tahap awal pengembangan proyek untuk estimasi biaya web (Mendes et al., 2005). Hal ini dilakukan

dengan melakukan 2 survei dan 1 case study. Pada tahun 2009, membuat model untuk estimasi effort web dengan menggunakan *bayesian networks* (Mendes et al., 2009). Variabel yang digunakan pada model tersebut juga merupakan subset dari variabel-variabel yg disebutkan pada tabel 2.3.

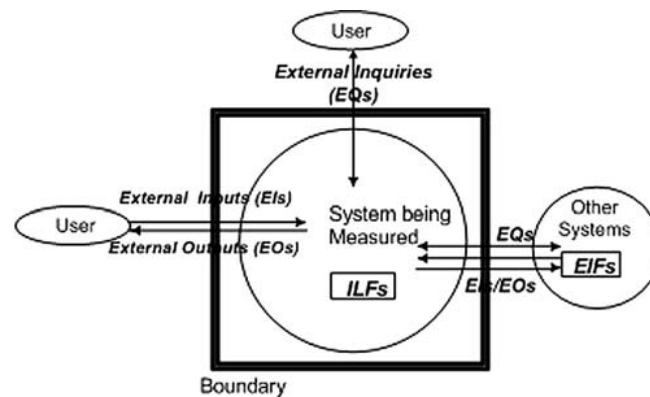
### **2.3.2 Function Point Analysis**

*Function Point* (FP) adalah suatu pengukuran standar dari perangkat lunak untuk kuantifikasi fungsionalitas yang ditawarkan oleh program ke pengguna (Fraternali dan Bongio, 2006). FPA pertama kali dikembangkan oleh Allan Albrecht untuk IBM (Albrecht, 1979), yang kemudian dikembangkan oleh IFPUG (IFPUG, 1999).

Metode perhitungan IFPUG berdasarkan pada identifikasi fungsi yang seharusnya dilakukan oleh sistem dan memberikan tingkat kompleksitas pada setiap fungsi. Setiap fungsi yang berbeda-beda diklasifikasikan (Fraternali dan Bongio, 2006) (Abrahao, Poels, dan Pastor, 2006) menjadi :

1. Fungsi tentang penyimpanan data (*Data Functions*) yang terdiri dari :
  - a. *Internal Logical Files* (ILF) jika data yang dibuat, dipelihara dan diatur oleh sistem.
  - b. *External Interface Files* (EIF) jika data yang digunakan diatur di luar ruang lingkup aplikasi.
2. Fungsi yang berinteraksi dengan pengguna (*Transaction Functions*) terdiri dari :

- a. *External Inputs* (EI), jika tujuannya adalah mengumpulkan informasi dari pengguna.
- b. *External Queries* (EQ), jika tujuannya adalah untuk mengambil informasi yang diminta oleh pengguna.
- c. *External Outputs* (EO), jika memberikan informasi kepada pengguna akibat dari suatu aksi atau elaborasi.



**Gambar 2.2. Pandangan FPA pada Ukuran Fungsional (Abrahao, Poels, dan Pastor, 2006)**

### 2.3.3 OOmFPWeb

Penelitian yang dilakukan terhadap pengukuran aplikasi web berdasarkan fungsional sistem telah dilakukan oleh *Web-Points* (Cleary, 2000), *Web Objects* (Reifer, 2000), *Internet Points* (Cost Xpert Group Inc, 2001), OOmFPWeb (Abrahao dan Pastor, 2006), *Data Web Points* (Ochoa et al., 2003), Candido dan Sanchez (2003), dan Fraternali et al. (2006). Beberapa penelitian seperti *Web Objects*, *Web-Points*, dan *Internet Points* tidak bisa digunakan pada awal tahap pengembangan proyek web karena

metode-metode tersebut bergantung pada jenis bahasa dan teknologi yang dilakukan pada saat implementasi (Abrahao dan Poels, 2009). Penelitian yang dilakukan oleh Fraternali et al. (2006) menghasilkan perbedaan 11% antara perhitungan secara manual dan secara otomatis. OOmFPWeb merupakan satu-satunya metode yang sudah divalidasi secara sistematis menggunakan evaluasi framework untuk metode atau prosedur FSM (Abrahao dan Poels, 2009). Selain itu, hasil dari evaluasi terhadap OOmFPWeb menunjukkan bahwa OOmFPWeb adalah efisien dibandingkan dengan praktis industri saat ini dan OOmFPWeb mudah untuk dipakai dan berguna untuk pengguna (Abrahao dan Poels, 2009). Selain itu, OOmFPWeb mempunyai keunggulan yaitu tidak bergantung pada teknologi yang digunakan untuk mengembangkan aplikasi web (Abrahao dan Poels, 2009).

*OO-Method Function Points for the Web* (OOmFPWeb) berdasarkan pada pengukuran skema konseptual dengan menggunakan metode berorientasi objek (Abrahao, Poels, dan Pastor, 2006). Pendekatan ini menghasilkan suatu nilai ukuran fungsional yang ada pada suatu sistem. Nilai ukuran fungsional mencerminkan jumlah fungsionalitas yang dimiliki dari spesifikasi kebutuhan pengguna sebagai model dalam skema konseptual *OO-method* (Abrahao, Poels, dan Pastor, 2006).

OomFPWeb memiliki beberapa pandangan perspektif untuk aplikasi web yang terdiri dari (Abrahao dan Poels, 2009):

1. *Data* : data yang digunakan dan dipelihara oleh aplikasi web. Data didefinisikan dengan menggunakan model objek. Model class

memrepresentasikan data logikal yang dipelihara oleh aplikasi dan *legacy views* memrepresentasikan data logikal yang direferensikan oleh aplikasi.

2. *Process* : komputasi yang seharusnya dilakukan oleh aplikasi web yang didefinisikan oleh model objek dengan definisi semantik yang jelas yang berhubungan dengan perubahan *state* pada model fungsional.
3. *Behavior* : interaksi dinamis dari sistem yang berhubungan dengan objek yang hidup dan interaksi antar objek yang didefinisikan di dalam model dinamis.
4. *Navigation* : struktur navigasi dari aplikasi yang didefinisikan di dalam model navigasi.
5. *Presentation* : interaksi pengguna dengan antarmuka aplikasi web, yang didefinisikan di model presentasi.



**Tabel 2.3. Mapping antara konsep FPA dan konsep OOWS (Abraham et al., 2006)**

<b>Fungsi</b>	<b>FPA</b>	<b>OO-Method</b>
ILF	Data logikal yang diidentifikasi oleh pengguna yang dipelihara di dalam ruang lingkup sistem.	Class yang mengenkapsulasi kumpulan atribut data yang merepresentasikan state dari objek dari setiap class.
EIF	Data logikal yang direferensikan oleh sistem dan dipelihara dalam ruang lingkup sistem.	<i>Legacy view</i> yang didefinisikan sebagai filter yang ditempatkan di class oleh sistem yang sudah ada sebelumnya.
EI	Sebuah proses yang memproses data atau informasi kontrol dari luar sistem.	<i>Service</i> yang didefinisikan di dalam class atau legacy view.
EO	Sebuah proses untuk mengirim data atau mengontrol informasi ke luar ruang lingkup sistem.	<i>Instance Interaction Unit</i> , <i>Population Interaction Unit</i> , dan <i>Master detail Interaction Unit</i> yang didefinisikan di model presentasi.
EQ	Sebuah proses yang memberikan informasi ke pengguna melalui penarikan data atau informasi kontrol.	Model di dalam model presentasi, memberikan informasi ke pengguna tanpa mengubah kelakuan sistem.

Kompleksitas dari fungsi transaksional adalah fungsi dari jumlah *Data Element Types* (DET\_Transaction) dan jumlah *File Types*



*Referenced* (FTR). FTR adalah fungsi data yang direferensikan selama eksekusi dari fungsi transaction (Abrahamo dan Poels, 2006).

Kompleksitas dari fungsi data adalah fungsi dari jumlah *Data Element Type* (DET) dan jumlah *Record Element Type* (RET). DET merupakan suatu field yang unik dan tidak berulang-ulang. RET adalah elemen data yang dikenalin oleh pengguna dalam file logikal (ILF atau EIF) (Abrahamo dan Poels, 2006).

Aturan perhitungan total DET dan RET untuk *class* dan *legacy view* bisa dilihat pada tabel 2.4 dan tabel 2.5. Sedangkan, aturan perhitungan total DET dan FTR untuk service pada *class* dan *legacy view*, IIU (Instance Interaction Unit), dan PIU (Population Interaction Unit) bisa dilihat pada tabel 2.6, tabel 2.7, tabel 2.8, dan tabel 2.98.

**Tabel 2.4. Aturan Pengukuran untuk Kompleksitas Class (Abrahamo et al., 2006)**

Tipe Element Data	<i>Record Element Type</i>
1 DET untuk setiap atribut dari class	1 RET untuk class
1 DET untuk setiap atribut di IF dari sebuah class atau legacy direferensikan oleh sebuah hubungan agregasi banyak.	1 RET untuk setiap hubungan <i>multi-valued</i> agregasi ( <i>class</i> atau <i>legacy view</i> )
1 DET untuk setiap atribut di IF dari class induk dari sebuah class	

**Tabel 2.5. Aturan Pengukuran Kompleksitas dari Legacy View  
(Abrahao et al., 2006)**

Tipe Element Data	Record Element Type
1 DET untuk setiap atribut dari <i>legacy view</i> .	1 RET untuk <i>legacy view</i> .
1 DET untuk setiap atribut di IF dari class yang berhubungan dengan hubungan agregasi <i>univalued</i> .	1 RET untuk setiap hubungan <i>multi-valued</i> agregasi dengan sebuah class.

**Tabel 2.6. Aturan Pengukuran untuk Kompleksitas dari Service  
(Abrahao et al., 2006)**

Tipe Element Data	File Type Referenced
1 DET untuk setiap argumen nilai data dari <i>service</i> .	1 FTR untuk class.
1 DET untuk kapasitas sistem untuk mengirim data.	1 FTR untuk setiap class baru yang direferensi dalam argumen nilai objek dari <i>service</i> .
1 DET untuk aksi (terima/batal) dari eksekusi <i>service</i> .	1 FTR untuk setiap class baru yang direferensi dalam formula dari nilai standar.
	1 FTR untuk setiap class baru yang direferensi dalam formula dari kejadian <i>destroy</i> .
	1 FTR untuk setiap class baru yang direferensi dalam formula transaksi.
	1 FTR untuk setiap class baru yang direferensi dalam formula kondisi dari spesialisasi.
	Untuks spesialisasi oleh kejadian,

	hitung 1 FTR untuk setiap class baru yang kejadian adalah <i>carrier</i> dan <i>liberator</i> .
	1 FTR untuk setiap class baru direferensi dalam formula batasan integritas.

**Tabel 2.7. Aturan Pengukuran untuk Kompleksitas dari Service Legacy View (Abrahao et al., 2006)**

Type Element Data	File Type Referenced
1 DET untuk setiap argumen nilai data dari service.	1 FTR untuk <i>legacy view</i> .
1 DET untuk kapasitas sistem untuk mengirim pesan.	1 FTR untuk setiap class baru direferensi dalam formula <i>precondition</i> .
1 DET untuk aksi (terima/batal) dari <i>service</i> .	1 FTR untuk setiap class baru direferensi dalam formula eksekusi dari sebuah <i>integrity constraint</i> .

**Tabel 2.8. Aturan Pengukuran untuk Instance Interaction Unit Pattern (Abrahao et al., 2006)**

Type Element Data	File Type Referenced
1 DET untuk setiap atribut yang unik.	1 FTR untuk setiap class atau <i>legacy view</i> dalam tampilan.
1 DET untuk setiap atribut unik yang ditampilkan (hanya jika atributnya berbeda dari OID).	
1 DET untuk setiap aksi yang ditawarkan.	
1 DET untuk setiap navigasi yang	

---

ditawarkan.

---

1 DET untuk kapasitas sistem untuk menampilkan pesan.

---

**Tabel 2.9. Aturan Pengukuran untuk Population Interaction Unit Pattern (Abrahao et al., 2006)**

Tipe Element Data	Record Element Type
1 DET untuk setiap variabel nilai data yang unik dari sebuah filter.	1 FTR untuk setiap <i>class</i> atau <i>legacy view</i> dalam tampilan.
1 DET untuk setiap atribut dalam tampilan (hanya jika atributnya berbeda dari filter).	1 FTR untuk setiap variabel nilai objek yang unik dari filter.
1 DET untuk setiap aksi yang ditawarkan (standarnya service class/ <i>legacy view</i> ).	1 FTR untuk setiap class baru direferensi dalam kriteria pengurutan.
1 DET untuk setiap navigasi yang ditawarkan.	1 FTR untuk setiap class baru direferensi dalam formula filter.
1 DET untuk kapasitas sistem untuk menampilkan pesan.	

Untuk menentukan tingkat kompleksitas dari ILF dan EIF bisa dilihat pada aturan yang disebutkan pada tabel 2.10. Untuk menentukan tingkat kompleksitas dari EI bisa dilihat pada aturan yang disebutkan pada tabel 2.11. Untuk menentukan tingkat kompleksitas dari EO dan EQ bisa dilihat pada aturan yang disebutkan pada tabel 2.12 (Alexander, 2004).

**Tabel 2.10. Level Kompleksitas untuk ILF dan EIF (Alexander, 2004)**

RETs	<i>Data Element Type (DETs)</i>		
	1 - 19	20 - 50	51+
1	<i>Low</i>	<i>Low</i>	<i>Average</i>
2 - 5	<i>Low</i>	<i>Average</i>	<i>High</i>
6 or more	<i>Average</i>	<i>High</i>	<i>High</i>

**Tabel 2.11. Level Kompleksitas untuk EI (Alexander, 2004)**

RETs	<i>Data Element Type (DETs)</i>		
	1 - 4	5 - 15	16+
0 - 1	<i>Low</i>	<i>Low</i>	<i>Average</i>
2	<i>Low</i>	<i>Average</i>	<i>High</i>
3 or more	<i>Average</i>	<i>High</i>	<i>High</i>

**Tabel 2.12. Level Kompleksitas untuk EO dan EQ (Alexander, 2004)**

RETs	<i>Data Element Type (DETs)</i>		
	1 - 5	6- 19	20+
0 - 1	<i>Low</i>	<i>Low</i>	<i>Average</i>
2 - 3	<i>Low</i>	<i>Average</i>	<i>High</i>
4 or more	<i>Average</i>	<i>High</i>	<i>High</i>

Untuk setiap fungsi yang ditemukan, diberikan bobot sesuai dengan tingkat kompleksitasnya yang bisa dilihat pada tabel 2.13.

**Tabel 2.13. Bobot Kompleksitas (Alexander, 2004)**

Tipe Fungsi	<i>Low</i>	<i>Average</i>	<i>High</i>
ILF	7	10	15
EIF	5	7	10
EI	3	4	6

EO	4	5	7
EQ	3	4	6

### 2.3.3.2 Perhitungan OOmFPWeb

Dengan diberikan sebuah skema konseptual yang dibuat pada saat langkah pemodelan *OO-Method* konseptual, OOmFP dapat dihitung dengan menggunakan rumus (Abraham et al., 2006) :

$$OOmFP = OOmFP_D + OOmFP_T \quad (1)$$

dimana :

$$OOmFP_D = \sum_{i=1}^n OOmFP_{class} + \sum_{j=1}^m OOmFP_{legView} \quad (2)$$

$$OOmFP_T = \sum_{i=1}^n OOmFP_{service} + \sum_{j=1}^m OOmFP_{IU|PIU|MDIU} \quad (3)$$

## 2.4 Metode Estimasi Usaha dan Biaya

### 2.4.1 Metode Estimasi Usaha

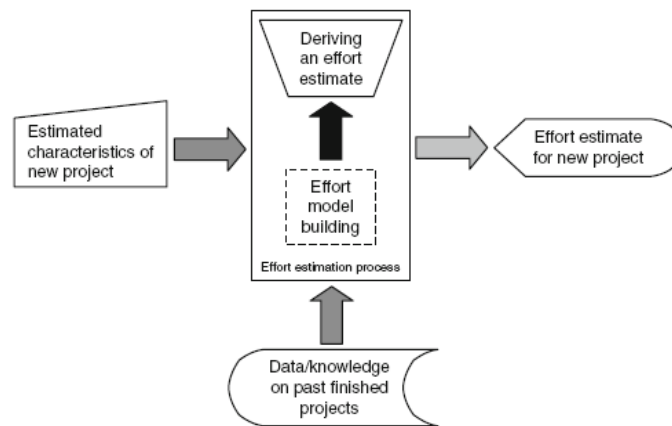
Estimasi usaha digunakan untuk memprediksi jumlah waktu yang diperlukan oleh seseorang atau sekelompok orang untuk mencapai tugas/aktivitas/proses yang diberikan (Mendes, 2010). Proses dalam mengestimasi usaha terdiri dari (Mendes, 2010) :

1. Membuat model usaha

Model usaha yang dibuat terdiri dari karakteristik proyek dan usaha yang sudah diketahui yang diambil dari data/pengetahuan dari proyek yang sudah selesai.

## 2. Membuat sebuah perkiraan usaha

Dalam proses ini, sebuah perkiraan usaha dari proyek baru didapat dengan menggunakan karakteristik proyek baru dibandingkan dengan karakteristik proyek-proyek lama yang sudah selesai.



**Gambar 2.4. Tahapan Estimasi Usaha (Mendes, 2010)**

Model usaha software dan perkiraan usaha membantu proyek manajer untuk mengalokasikan sumber daya, mengontrol biaya, mengatur jadwal dan meningkatkan praktis sekarang, mengarah ke proyek yang terselesaikan tepat waktu dan dalam anggaran yang ditetapkan (Mendes dan Mosley, 2006).

Beberapa teknik untuk melakukan estimasi usaha sudah diajukan lebih dari 30 tahun terakhir dalam bidang rekayasa piranti lunak. Teknik

untuk melakukan estimasi biaya dibagi menjadi tiga kategori (Shepperd, Schofield, dan Kitchenham, 1996), yaitu :

#### 1. Pendapat ahli

Proses estimasi teknik ini dilakukan dengan melalui pendapat para ahli secara subjektif berdasarkan pada pengalaman di masa lalu dari mengatur proyek yang sama sebelumnya. Proses estimasi usaha menggunakan pendapat ahli terdiri dari :

- b. Ahli melihat faktor-faktor yang mempengaruhi estimasi usaha dan biaya yang berhubungan dengan proyek, baru dimana usaha yang butuh diestimasi.
- c. Berdasarkan data yang diingat atau yang diambil dari proyek yang sudah selesai di masa yang usahnya sudah diketahui.
- d. Berdasarkan data dari poin (a) dan (b), dilakukan estimasi usaha untuk proyek baru secara subjektif. Estimasi usaha yang tepat bisa diberikan jika terdapat proyek yang sama dengan proyek di masa lalu.

#### 2. Model algoritma

Salah satu model algoritma adalah *COConstructive COst MOdel* (COCOMO) (Briand et al., 1999).

$$\textit{Estimated Effort} = a \textit{EstSizeNewproj}^b \textit{EAF} \quad (4)$$

dimana :

nilai a dan b berdasarkan pada jenis proyek yang sedang berjalan,.

EstSizeNewProj merupakan estimasi ukuran dari proyek baru,



*Effort Adjustment Factor* (EAF) berdasarkan pada 15 pemicu biaya yang dihitung dan dijumlahkan.

Selain itu, juga terdapat *regression-based algorithmic model*. Model ini cocok digunakan oleh satu perusahaan tertentu karena hasil estimasi usaha dari metode ini merupakan hasil yang didapat dari data proyek lama yang sudah selesai dari perusahaannya sendiri. Pada model ini terdapat prosedur untuk menemukan garis lurus yang paling baik antara usaha dan karakteristik proyek (Schroeder et al., 1986).

### 3. Teknik kecerdasan buatan

Salah satu jenis teknik kecerdasan buatan adalah *Case Based Reasoning* (CBR). CBR memberikan estimasi dengan membandingkan proyek baru yang akan diestimasi dengan proyek-proyek lama yang sudah diselesaikan di masa lalu dimana usaha sudah diketahui (Mendes dan Mosley, 2006, pp. 34). Pada saat menggunakan CBR terdapat enam parameter yang harus diperhatikan yang terdiri dari (Mendes, 2010) :

#### a. *Feature Subset Selection*

Seleksi subset fitur mencakup penentuan fitur-fitur yang optimum untuk menghasilkan estimasi yang lebih akurat menggunakan CBR. Fitur-fitur tersebut akan digunakan untuk mencari kasus yang paling mirip di masa lalu.

#### b. *Similarity Measure*

Pengukuran kemiripan digunakan untuk mencari kasus yang paling mirip dengan proyek baru, bisa menggunakan *Unweighted*

*Euclidean Distance*, *Weighted Euclidean Distance*, dan *Maximum Distance*. Rumus dari *Unweighted Euclidean Distance* adalah sebagai berikut :

$$d(x,y) = \sqrt{|x_0 - y_0|^2 + |x_1 - y_1|^2 + \dots + |x_{n-1} - y_{n-1}|^2 + |x_n - y_n|^2} \quad (5)$$

dimana  $x_0$  sampai  $x_n$  merupakan fitur 0 sampai fitur ke n dari kasus x dan  $y_0$  sampai  $y_n$  merupakan fitur 0 sampai fitur ke n dari kasus y. Sedangkan *Weighted Euclidean Distance* rumusnya adalah sebagai berikut :

$$d(x,y) = \sqrt{w_0|x_0 - y_0|^2 + w_1|x_1 - y_1|^2 + \dots + w_{n-1}|x_{n-1} - y_{n-1}|^2 + w_n|x_n - y_n|^2} \quad (6)$$

dimana  $w_0$  sampai  $w_n$  merupakan bobot untuk fitur 0 sampai fitur ke-n. Rumus dari *Maximum Distance* adalah sebagai berikut :

$$d = \sqrt{\max((x_0 - y_0)^2, (x_1 - y_1)^2)} \quad (7)$$

c. *Scaling*

*Scaling* merepresentasikan transformasi dari nilai fitur berdasarkan pada aturan yang sudah didefinisikan, seperti semua fitur merepresentasikan nilai dalam batasan yang sama maka semua fitur akan mempunyai derajat pengaruh yang sama dari hasil.

d. Jumlah Analogi

Jumlah analogi merepresentasikan jumlah kasus yang sama dengan kasus saat ini. Beberapa penelitian menggunakan satu kasus yang paling mirip untuk mengambil estimasi usaha untuk proyek baru (Mendes et al., 2002), sedangkan terdapat juga penelitian yang

menggunakan dua atau tiga analogi yang paling mirip (Angelis dan Stamelos, 2000) (Ruhe et al., 2003).

e. Adaptasi Analogi

Setelah kasus yang paling mirip didapatkan, dilakukan adaptasi analogi terhadap kasus yang mirip tersebut untuk mendapat estimasi usaha dari proyek baru.

f. Aturan Adaptasi

Aturan adaptasi merepresentasikan adaptasi dari estimasi usaha, berdasarkan pada kriteria yang diberikan seperti karakteristik dari proyek baru.

## 2.4.2 Metode Estimasi Biaya

Estimasi biaya yang akurat merupakan elemen yang penting untuk bisa memberikan tawaran yang kompetitif dan bertahan sukses di pasar. Estimasi biaya yang berlebihan atau kurang mempunyai pengaruh yang sangat besar dalam reputasi dan kompetitif dari perusahaan (Ruhe, Jeffery, dan Wieczorek, 2003).

Pendekatan yang digunakan untuk mengestimasi biaya dari suatu aplikasi web yang berbasis OO adalah dengan menggunakan variabel-variabel yang diusulkan oleh Sarno et al. (2002). Variabel-variabel yang digunakan oleh Sarno et al. (2002) untuk menghitung biaya pengembangan proyek perangkat lunak adalah jumlah orang yang diperlukan untuk pemrograman, biaya tenaga kerja untuk pemrograman, biaya listrik untuk pemrograman, biaya konsumsi untuk pemrograman,

dan biaya overhead untuk pemrograman. Untuk menghitung jumlah orang yang dibutuhkan bisa digunakan rumus (8), dimana  $E$  adalah effort yang dibutuhkan untuk menyelesaikan suatu proyek perangkat lunak dan  $t_d$  adalah target durasi yang dibutuhkan untuk menyelesaikan proyek perangkat lunak.

$$MP = E / t_d \quad (8)$$

Rumus untuk menghitung biaya yang diperlukan untuk pengembangan proyek software adalah :

$$B_{MB} = C_{MB} + C_{LMB} + C_{KMB} + BO_{MB} \quad (9)$$

$$C_{MB} = UR * E \quad (10)$$

dimana  $B_{MB}$  merupakan biaya pemrograman dan implementasi,  $C_{MB}$  merupakan biaya tenaga kerja untuk pemrograman,  $C_{LMB}$  merupakan biaya listrik untuk pemrograman, biaya konsumsi untuk pemrograman,  $BO_{MB}$  merupakan biaya overhead untuk pemrograman,  $UR$  merupakan upah regional,  $E$  merupakan effort yang dibutuhkan untuk menyelesaikan proyek perangkat lunak.

$$\begin{aligned} C_{LMB} &= E * L_{Rp} \\ L_{Rp} &= L_{komp} * J_H * H_B \end{aligned} \quad (11)$$

$$\begin{aligned} C_{KMB} &= E * K_{Rp} \\ K_{Rp} &= K_H * H_B \end{aligned} \quad (12)$$

$$\begin{aligned} BO_{MB} &= C_{Gfs} + C_{Dfs} + C_{Tfs} + C_{Afs} \\ C_{Gfs} &= E * G_{Rp} \\ C_{Dfs} &= E * D_{Rp} \\ C_{Tfs} &= E * T_{Rp} \\ C_{Afs} &= E * A_{Rp} \end{aligned} \quad (13)$$

dimana  $L_{Rp}$  merupakan ongkos listrik per unit komputer per bulan,  $J_H$  merupakan jumlah jam kerja per hari,  $H_B$  merupakan jumlah hari kerja per bulan,  $K_{Rp}$  merupakan biaya konsumsi per orang per bulan,  $K_H$  merupakan biaya konsumsi per orang per hari,  $C_{Gfs}$  merupakan biaya gedung dan listrik,  $C_{DFS}$  merupakan biaya depriasi mesin,  $C_{Tfs}$  merupakan biaya telpon,  $C_{Afs}$  merupakan biaya asuransi,  $G_{Rp}$  merupakan biaya sewa gedung dan ongkos listrik per orang per bulan,  $D_{Rp}$  merupakan biaya depriasi mesin per bulan,  $T_{Rp}$  merupakan biaya telpon per orang per bulan,  $A_{Rp}$  merupakan biaya asuransi per orang per bulan.