

## BAB 6

### HASIL PENELITIAN

Penelitian dilakukan pada bulan April sampai Mei 2009. Penelitian terbagi menjadi dua tahap yaitu berupa pengambilan data sekunder dan pengambilan data primer. Data sekunder diperoleh dari Pusat Administrasi FKUI (Bagian Pendidikan, Keuangan, SDM, Umum dan Fasilitas) untuk dicantumkan pada simulasi yang akan dilakukan sehubungan dengan pengambilan data primer. Data primer yang mencakup *usability* dan *performance* perangkat lunak penghitung *unit cost* diperoleh dari observasi, wawancara, dan pengisian formulir tanggapan pada peserta pelatihan perhitungan *unit cost* modul pendidikan kedokteran FKUI. Jumlah peserta mengikuti pelatihan sebanyak 30 peserta. Pelatihan diawali dengan penjelasan mengenai *unit cost* dan cara perhitungannya. Selanjutnya dilakukan simulasi perhitungan *unit cost*, dimana setiap peserta diberikan contoh soal dengan data yang sama untuk melakukan perhitungan *unit cost* berdasarkan data yang tersedia. Setiap peserta harus melakukan perhitungan secara manual menggunakan *spreadsheet* dan menggunakan perangkat lunak penghitung *unit cost* dalam waktu yang ditentukan. Sebelum simulasi dimulai, masing masing peserta diberi penjelasan cara menghitung dengan menggunakan *spreadsheet* dan cara menggunakan perangkat lunak.

### 6.1. Jumlah Peserta Pelatihan

Jumlah peserta pelatihan perhitungan *unit cost* modul pendidikan kedokteran FKUI pada tanggal 19 dan 26 Mei 2009 sebanyak 30 orang peserta. Peserta terdiri dari 17 orang perwakilan pengelola modul, 9 orang staf keuangan dan 4 staf administrasi non keuangan Pusat Administrasi FKUI.

### 6.2. Pendidikan Terakhir dan Profesi Utama

Pendidikan terakhir peserta pelatihan perhitungan *unit cost* modul pendidikan kedokteran adalah sebagai berikut.

SMA : 1 orang

Diploma 3 : 5 orang

Strata 1 : 8 orang

Strata 2 : 16 orang

Profesi utama peserta pelatihan perhitungan *unit cost* modul pendidikan kedokteran adalah sebagai berikut.

Staf pengajar : 14 orang

Staf administrasi : 16 orang

### 6.3. Keahlian khusus di bidang ilmu komputer

Berdasarkan hasil wawancara dan isian formulir tanggapan mengenai keahlian khusus di bidang ilmu komputer, dalam hal ini dibatasi kemampuan mengelola data menggunakan *spreadsheet* didapatkan data sebagai berikut:

- 1) tidak mampu mengoperasikan perintah sederhana: 4 orang;
- 2) mampu mengoperasikan perintah sederhana dan penggunaan fitur minimal: 16 orang;
- 3) menguasai sebagian besar fitur: 7 orang;
- 4) mahir menggunakan semua fitur: 3 orang.

### 6.4. Keahlian khusus di bidang keuangan

Berdasarkan hasil wawancara dan isian formulir tanggapan mengenai keahlian khusus di bidang keuangan, dalam hal ini dibatasi kemampuan membuat anggaran, melakukan perhitungan akuntansi, dan menghitung *unit cost* didapatkan data sebagai berikut:

- 1) tidak pernah mengikuti pendidikan formal atau kursus/pelatihan bersertifikat di bidang keuangan: 21 orang;
- 2) mengikuti pendidikan formal atau kursus/pelatihan bersertifikat di bidang keuangan: 8 orang;
- 3) ahli di bidang perhitungan *unit cost* pendidikan: 1 orang.

### 6.5. Sistem operasi yang digunakan

Seluruh peserta pelatihan menggunakan sistem operasi Microsoft Windows dalam beberapa versi sebagai berikut:

- 1) Windows.Vista: 4 orang;
- 2) Windows XP: 23 orang;
- 3) Windows 2000/NT: 2 orang;
- 4) Windows 98: 1 orang.

### 6.6. Instalasi Java Virtual Machine (JVM)

Perangkat lunak yang digunakan menggunakan bahasa pemrograman Java, untuk menjalankannya maka harus dilakukan instalasi terlebih dahulu. Sebanyak 20 buah *personal computer/note book* telah terinstalasi JVM, sementara 10 buah lainnya harus dilakukan instalasi JVM (versi 1.6) terlebih dahulu.

### 6.7. Perangkat *spreadsheet*

Seluruh peserta menggunakan *spreadsheet* Microsoft Excell dengan versi sebagai berikut:

- a) Versi 2003: 25 orang;
- b) Versi 2007: 5 orang.

## 6.8. Langkah Pengerjaan Perhitungan *Unit Cost* Modul Pendidikan Kedokteran

### 6.8.1. Menggunakan *spreadsheet*

Langkah pengerjaan melakukan perhitungan menggunakan *spreadsheet* adalah sebagai berikut.

1. Jalankan program *spreadsheet*
2. Buat file baru
3. Buat kolom kegiatan pendidikan dan input yang diperlukan
4. Isi kegiatan pendidikan dan tandai sesuai input yang dibutuhkan
5. Lakukan perincian biaya pada input yang telah ditandai
6. Lakukan konversi input ke dalam nilai moneter
7. Lakukan pengelompokan biaya
8. Masukkan rumus perhitungan

### 6.8.2. Menggunakan Perangkat Lunak Penghitung *Unit Cost*

Langkah pengerjaan melakukan perhitungan menggunakan perangkat lunak adalah sebagai berikut.

1. Jalankan perangkat lunak
2. Lakukan proses *log-in*
3. Buat *unit profile*
4. Buat *unit cost* dari *unit profile* yang dipilih
5. Masukkan jumlah kapasitas dan kuantitas pada *classification unit profile*, tekan *next*
6. Masukkan kegiatan pendidikan dan rinciannya pada *structure unit profile*, tekan *next*
7. Tandai sesuai dengan pengelompokan pada input pada *description unit profile*, tekan *next*

8. Masukan rincian pembiayaan dan pengelompokan biaya pada *entry data unit profile*, tekan *next*
9. Lihat hasil

### 6.9. Perhitungan *Unit Cost* Modul Pendidikan Kedokteran

Setiap peserta diberi waktu masing-masing satu jam untuk menyelesaikan perhitungan menggunakan *spreadsheet* dan satu jam terpisah untuk melakukan perhitungan dengan menggunakan perangkat lunak penghitung *unit cost*. Peserta yang tidak dapat menjalankan (*execute*) perangkat lunak penghitung *unit cost* dimasukkan kedalam kelompok “*Non Applicable*”, sementara peserta yang tidak dapat menyelesaikan karena menemui kesalahan/*fault* pada perangkat lunak penghitung *unit cost* ataupun karena sebab lainnya dimasukkan kedalam kelompok “Tidak Selesai”. Hasil perhitungan pada simulasi sesuai dengan perhitungan yang telah dilakukan oleh penulis adalah sebagai berikut.

*Total Fixed Cost* = Rp. 17.700.000

*Total Variable Cost* = Rp. 126.887.500

*Actual Unit Cost* = Rp. 628,641.30

*Normative Unit Cost* = Rp. 625,434.78

Kesimpulan *unit cost* tidak efisien karena *normative unit cost* lebih kecil dari pada *actual unit cost*

Penulis hanya mencantumkan pengelompokan hasil perhitungan simulasi berdasarkan ketepatan penyelesaian dan ketepatan perhitungan peserta pelatihan dalam waktu 1 jam. (Tabel 6.1).

**Tabel 6.1.**  
**Perhitungan Simulasi *Unit Cost* Modul Pendidikan Kedokteran FKUI**

<i>Spreadsheet</i>				Perangkat Lunak Penghitung Unit Cost				
Penyelesaian		Perhitungan		Penyelesaian			Perhitungan	
Selesai	Tidak Selesai	Benar	Salah	NA*	Selesai	Tidak Selesai	Benar	Salah
25	5	20	5	4	10	16	10	0

Keterangan:

\* NA = *non applicable* (program tidak dapat dijalankan)

#### 6.9.1. Kompatibilitas dan *Fault* Perangkat Lunak Penghitung *Unit Cost*

Sebanyak 4 peserta tidak dapat menjalankan (*execute*) perangkat lunak penghitung *unit cost*, tidak satupun peserta yang berhasil menjalankan perangkat lunak penghitung *unit cost* mengalami henti sementara. Adapun *fault* yang ditemukan oleh penulis dan peserta lainnya adalah sebagai berikut.

1. Tidak dapat melakukan proses *log-in*: 2 orang
2. Data “Structure unit profile” tidak dapat ditampilkan: 2 orang
3. Pada pengisian “entry data unit cost” data “variable cost” tidak hilang ketika data “fixed cost” untuk kelompok *entry* yang sama diisikan dan sebaliknya: 24 orang
4. Pada versi cetak nilai atau hasil perhitungan *unit cost* tidak tercantum: 24 orang
5. Terdapat penggandaan nilai *unit cost* yang tercantum apabila “view result” dijalankan kembali dalam satu sesi yang berurutan: 24 orang

### 6.9.2. Kesulitan Melakukan Perhitungan Simulasi

Berdasarkan hasil observasi dan wawancara, penulis menemukan beberapa peserta mengalami kesulitan dalam pengerjaan menggunakan *spreadsheet* dan perangkat lunak penghitung *unit cost*. Kesulitan yang dimaksud adalah sesuatu yang menyebabkan perhitungan simulasi tidak selesai sesuai dengan waktu yang diberikan. Beberapa kesulitan yang ditemui dalam pengguna secara keseluruhan berupa kesulitan menggunakan fitur atau fungsi yang ada pada *spreadsheet* maupun perangkat lunak *penghitung unit cost*. Beberapa peserta seringkali harus dijelaskan cara menggunakan fitur atau fungsi yang ada. Umumnya peserta yang harus dijelaskan tersebut tidak memiliki keterampilan komputer (khususnya *spreadsheet*) dan tidak terbiasa melakukan pekerjaan administratif. Tabel 6.2 menunjukkan kesulitan yang ditemui dalam menyelesaikan perhitungan *unit cost* menggunakan *spreadsheet* dan perangkat lunak penghitung *unit cost*:



**Tabel 6.2.**  
**Kesulitan Penyelesaian Perhitungan Unit Cost Menggunakan *Spreadsheet***  
**dan Perangkat Lunak Penghitung *Unit Cost***

Perhitungan			
<i>Spreadsheet</i>		Perangkat Lunak Penghitung <i>Unit Cost</i>	
Kesulitan	Peserta	Kesulitan	Peserta
Kesulitan memahami langkah-langkah perhitungan <i>unit cost</i>	5	Kesulitan memahami langkah-langkah perhitungan <i>unit cost</i>	5
Keterbatasan waktu yang tersedia	3	Keterbatasan waktu yang tersedia	10
Kesulitan mengelompokkan data dan melakukan perhitungan	5	Kesulitan menggunakan fitur yang tersedia	12
Tidak menguasai <i>spreadsheet</i>	4	Fitur tidak lengkap	14

### 6.9.3. Kesulitan *Spreadsheet* dan Perangkat Lunak Penghitung *Unit Cost*

Seluruh peserta menemukan kesulitan dalam melakukan perhitungan simulasi baik menggunakan *spreadsheet* ataupun menggunakan perangkat lunak penghitung *unit cost*. Kesulitan yang dijumpai pada penggunaan *spreadsheet* dan perangkat lunak penghitung *unit cost* untuk melakukan simulasi perhitungan menurut peserta adalah sebagai berikut (tabel 6.3).

**Tabel 6.3.**  
**Kesulitan Penggunaan *Spreadsheet* dan Perangkat Lunak Penghitung *Unit Cost***

Perhitungan			
<i>Spreadsheet</i>		Perangkat Lunak Penghitung <i>Unit Cost</i>	
Kesulitan	Peserta	Kesulitan	Peserta
Pengelompokan/pemasukan data dilakukan berkali kali	12	Instalasi JVM	6
Formulasi/perhitungan data yang ada dilakukan dengan memasukkan rumus	13	Tampilan kurang familiar	24
Perlu lebih banyak kolom isian (tidak praktis)	24	Penggabungan data tidak dapat dilakukan	24
Membutuhkan <i>sheet</i> yang lebih banyak (tampilan lebih rumit)	27	Perbaiki kesalahan tidak praktis	24
		Versi cetak hanya menampilkan hasil perhitungan jumlah <i>variable cost</i> dan <i>fixed cost</i> saja bukan hasil perhitungan <i>unit cost</i>	7

#### 6.9.4. Kemudahan *Spreadsheet* dan Perangkat Lunak Penghitung *Unit Cost*

Selain kesulitan yang dijumpai dalam melakukan proses perhitungan simulasi *unit cost*, peserta juga merasakan kemudahan yang ditemui baik menggunakan *spreadsheet* ataupun perangkat lunak penghitung *unit cost*. Kemudahan *spreadsheet* dan perangkat lunak penghitung *unit cost* untuk melakukan simulasi perhitungan menurut peserta adalah sebagai berikut (tabel 6.4).

**Tabel 6.4.**  
**Kemudahan Penggunaan *Spreadsheet* dan Perangkat Lunak Penghitung *Unit Cost***

Perhitungan			
<i>Spreadsheet</i>		Perangkat Lunak Penghitung <i>Unit Cost</i>	
Kemudahan	Peserta	Kemudahan	Peserta
Langsung dapat dijalankan tanpa menginstall program lain selain <i>spreadsheet</i>	30	Data terkelompok secara otomatis	26
Tampilan lebih familiar	30	Perhitungan dilakukan secara otomatis	26
Kesalahan pengisian lebih mudah ditelusuri dan diperbaiki	30	Input data tidak perlu berulang ulang	26
Dapat dilakukan penggabungan data dari file <i>/sheet spreadsheet</i> lainnya	30	Tampilan lebih sederhana	26
Versi cetak lebih lengkap dan lebih runtut	28		

#### 6.9.5. Ketertarikan Terhadap Penggunaan *Spreadsheet* dan Perangkat Lunak Penghitung *Unit Cost*

Hampir seluruh peserta (23 orang) menyukai perhitungan *unit cost* menggunakan *spreadsheet*, dengan alasan utama lebih familiar sehingga lebih mudah untuk melakukan pengelompokan data. Sebanyak 7 peserta tidak menyukai menggunakan *spreadsheet* dengan alasan utama tidak praktis. Sebanyak 20 orang peserta menyukai perhitungan dengan menggunakan perangkat lunak penghitung *unit cost* dengan alasan utama perhitungan lebih praktis. Sebanyak 10 peserta tidak menyukai menggunakan perangkat lunak penghitung *unit cost* dengan alasan utama tidak terlalu familiar dengan perangkat lunak tersebut dan beberapa kekurangan fitur yang terdapat di dalamnya.

#### 6.9.6. Kebutuhan Tenaga Khusus Untuk Melakukan Perhitungan *Unit Cost*

Sebanyak 21 peserta merasa perhitungan *unit cost* memiliki kesulitan tersendiri dalam pelaksanaannya mengingat latar belakang pendidikan dan profesi sehari-hari yang tidak berhubungan dengan keahlian di bidang keuangan ataupun komputer, sehingga memerlukan tenaga khusus. Sembilan peserta lainnya meski tidak mengalami kesulitan melakukan perhitungan juga berpendapat bahwa perhitungan *unit cost* perlu dilakukan oleh tenaga khusus. Tenaga khusus yang dibutuhkan untuk melakukan perhitungan *unit cost* menurut pendapat para peserta pelatihan adalah sebagai berikut.

1. Tenaga administrasi yang mengerti komputer (khususnya *spreadsheet* dan menjalankan suatu perangkat lunak tertentu): 12 orang.
2. Konsultan keuangan bidang anggaran/perencanaan: 10 orang.
3. Staf yang sudah pernah melakukan pelatihan atau mahir di bidang perhitungan *unit cost*: 22 orang.

## BAB 7

### PEMBAHASAN

#### 7.1. Proses Penelitian

Pengumpulan data dimulai dengan melakukan pengumpulan data sekunder yang akan digunakan sebagai bahan pembuatan soal simulasi perhitungan unit cost. Data diambil menggunakan wawancara dan formulir tanggapan dari Pusat Administrasi FKUI (Bagian Pendidikan, Keuangan, SDM, Umum dan Fasilitas). Selanjutnya pengambilan data primer dilakukan dalam bentuk wawancara, observasi, dan formulir tanggapan pada kegiatan pelatihan perhitungan *unit cost* modul pendidikan kedokteran FKUI.

#### 7.2. Keterbatasan Penelitian

Dalam proses penelitian ini ditemukan beberapa kendala antara lain sebagai berikut.

1. Jumlah kehadiran peserta pelatihan tidak sesuai dengan undangan yang disebarkan. Hal ini mengingat setiap pengelola modul terdiri dari staf pengajar yang saat pelaksanaan pelatihan juga harus melaksanakan kewajibannya sebagai tenaga pengajar.
2. Mengingat keterbatasan waktu dan sumber daya yang ada, penelitian hanya dibatasi dari segi *usability* yaitu menilai apakah perangkat ini dapat melakukan perhitungan dengan tepat, cepat, dan kesulitan atau kemudahan apa yang dijumpai pada penggunaan perangkat lunak tersebut. Dari segi verifikasi dinilai apakah perangkat lunak ini dapat dijalankan di berbagai sistem operasi, dan dari segi validasi dicari *fault* apakah yang ditemui selama perangkat lunak ini dijalankan. Pengujian observasi lain seperti *performance*,

*security, availability, reliability, dan capacity* tidak mungkin dilakukan karena membutuhkan waktu yang lebih lama dan observasi yang lebih cermat.

3. Kompatibilitas pada penelitian ini hanya terbatas pada sistem operasi yang digunakan. Kompatibilitas dengan program atau perangkat lunak lain tidak diteliti sehubungan dengan keterbatasan waktu dan subjek penelitian.
4. Waktu untuk mengerjakan simulasi perhitungan dibatasi selama satu jam, hal ini untuk membatasi sejauh mana peserta mampu mengerjakan dalam waktu yang ditentukan, bukan untuk mengetahui lamanya waktu yang dibutuhkan untuk mengerjakan simulasi. Lamanya waktu pengerjaan dapat menjadi tidak objektif karena banyak faktor yang mempengaruhinya seperti latar belakang pendidikan, ketrampilan yang dimiliki, serta profesi sehari-hari.

### **7.3. Pembahasan Hasil Penelitian**

#### **7.3.1. Kompatibilitas**

Dalam penelitian ini ditemukan sebanyak 4 peserta tidak dapat menjalankan perangkat lunak. Sebanyak 3 peserta dengan sistem operasi windows di bawah versi windows XP dan 1 peserta dengan sistem operasi versi windows XP tidak dapat menjalankan perangkat lunak. Satu peserta meski dengan sistem operasi Windows XP tidak dapat menjalankan perangkat lunak, disebabkan JVM yang terpasang pada komputer pengguna dibawah versi minimum yang dibutuhkan yaitu JVM versi 1.6. Ini menunjukkan ada suatu inkompatibilitas perangkat lunak dengan sistem operasi (atau program lain yang berhubungan) yang digunakan. Hal ini biasa terjadi pada pembuatan suatu perangkat lunak, apabila saat perancangannya, mengkoding atau melakukan pengujian

**Universitas Indonesia**

perangkat lunak dengan menggunakan sistem operasi versi terbaru atau satu sistem operasi saja. Konsekuensi yang ditimbulkan adalah perangkat lunak menjadi tidak sepenuhnya kompatibel dengan sistem operasi yang digunakan atau sistem operasi lain. Salah satu penyebab kegagalan perangkat lunak adalah tidak adanya kompatibilitas dengan program lain seperti sistem operasi atau program lain yang terkait dengan perangkat lunak. Sebenarnya hal ini tidak dapat dikatakan menjadi suatu kegagalan dari perangkat lunak apabila perancangan, koding, dan pengujian memang ditujukan untuk sebagian besar pengguna yang menggunakan sistem operasi yang sesuai. Suatu kegagalan terjadi apabila perangkat lunak ditujukan untuk kompatibel dengan berbagai sistem operasi dan dalam versi apapun atau secara signifikan tidak dapat digunakan oleh kebanyakan pengguna.<sup>1</sup>

### 7.3.2. *Fault*

Beberapa *fault* yang ditemukan pada sebagian peserta, menyebabkan peserta harus menghentikan proses pengerjaan, atau mengulang pengerjaan dari awal. *Fault* yang dijumpai berupa tidak munculnya data yang dimasukkan dan tidak dapat melakukan proses *log-in* dapat disebabkan oleh adanya ketidakmampuan program membaca atau menautkan sistem manajemen basis data MySQL yang ada. Idealnya sebelum penggunaan dipastikan terlebih dahulu apakah sudah terpasang MySQL pada komputer. Beberapa *fault* lain yang ditemukan adalah sebagai berikut.

---

<sup>1</sup> Diakses dari [http://en.wikipedia.org/wiki/Software\\_testing#Compatibility](http://en.wikipedia.org/wiki/Software_testing#Compatibility) [4 Mei 2009]

1. Pada pengisian “*entry data unit cost*” data “*variable cost*” tidak hilang ketika data “*fixed cost*” untuk kelompok *entry* yang sama diisikan dan sebaliknya.
2. Pada versi cetak nilai atau hasil perhitungan *unit cost* tidak tercantum.
3. Terdapat penggantian nilai *unit cost* yang tercantum apabila “*view result*” dijalankan kembali dalam satu sesi yang berurutan

*Fault* tersebut di atas kemungkinan muncul akibat adanya kesalahan coding (*coding errors*). Tidak semua *fault* pada suatu perangkat lunak disebabkan oleh kesalahan coding. Salah satu yang umumnya menyebabkan kecacatan atau defek adalah adanya kesenjangan kebutuhan (*requirements gaps*), misalkan adanya kebutuhan yang tidak terpenuhi sehingga menimbulkan kesalahan oleh perancang program tersebut. *Requirements gaps* yang umumnya digunakan adalah *non-functional requirements* yang terdiri dari *testability*, *scalability*, *maintanability*, *usability*, *performance*, dan *security*. *Fault* dapat muncul dalam berbagai proses tersebut. Seorang programmer dapat membuat kesalahan yang menghasilkan suatu *fault* pada kode sumber (*source code*). Jika *fault* ini tetap dijalankan (*execute*) pada situasi tertentu akan menimbulkan hasil yang salah yang menyebabkan suatu kegagalan (*failure*).<sup>2</sup>

### 7.3.3. Hasil Perhitungan

Dalam waktu satu jam yang diberikan, sebanyak 25 peserta yang berhasil menyelesaikan perhitungan dengan menggunakan *spreadsheet* hanya 5 orang yang tidak menghasilkan perhitungan yang tepat. Hal yang berbeda dijumpai pada perhitungan dengan menggunakan perangkat lunak dimana sebanyak 10 peserta yang berhasil menyelesaikan perhitungan sesuai dengan waktu yang

<sup>2</sup> Kolawa, Adam; Huizinga, Dorota (2007) Wiley-IEEE Computer Society Press. p. 86.



tersedia dengan hasil perhitungan tepat. Peserta di anggap selesai mengerjakan bukan hanya sesuai dengan waktu yang tersedia, tetapi juga dengan urutan pekerjaan yang sesuai dengan petunjuk pengerjaan. Hal ini menunjukkan bahwa dari segi ketepatan, perangkat lunak mampu menghitung data yang dimasukkan dengan tepat.

#### 7.3.4. Waktu Pengerjaan

Penelitian ini tidak mencari waktu yang dibutuhkan untuk melakukan simulasi perhitungan *unit cost* baik menggunakan *spreadsheet* ataupun perangkat lunak penghitung *unit cost*, akan tetapi untuk menilai seberapa banyak peserta yang mampu menyelesaikan soal dalam waktu yang disediakan. Sebanyak 25 peserta mampu menyelesaikan perhitungan menggunakan *spreadsheet* sesuai dengan waktu yang tersedia, sementara hanya 10 peserta yang menyelesaikan perhitungan menggunakan perangkat lunak penghitung *unit cost* dengan tepat waktu.

Salah satu keterbatasan dari penelitian ini adalah tidak dapat ditentukannya waktu yang dibutuhkan untuk melakukan perhitungan karena terdapat banyak faktor yang mempengaruhi. Adapun faktor yang mempengaruhi waktu pengerjaan antara lain.

1. Latar belakang ketrampilan/pendidikan atau profesi utama pengguna

Sebagian besar peserta merupakan staf pengajar dengan pendidikan terakhir setingkat strata dua, dengan latar belakang pendidikan tidak berhubungan dengan komputer maupun keuangan. Berdasarkan hasil observasi terdapat perbedaan yang mencolok antara peserta yang memiliki kemampuan di bidang komputer ataupun keuangan dengan yang tidak memiliki kemampuan di kedua bidang tersebut. Begitu pula

**Universitas Indonesia**

terlihat perbedaan pada peserta yang profesi utamanya sebagai tenaga administrasi dengan yang bukan tenaga administrasi.

Terdapat kesulitan pemahaman melakukan perintah atau urutan pengerjaan baik menggunakan *spreadsheet* atau perangkat lunak penghitung *unit cost* pada peserta yang tidak memiliki latar belakang di bidang komputer ataupun keuangan, serta yang bukan tenaga administrasi.

Saat pengerjaan simulasi perhitungan, seringkali terdapat interupsi dari peserta untuk menanyakan bagaimana melakukan perintah atau fungsi sederhana. Misalkan pada pengerjaan dengan menggunakan *spreadsheet* tidak jarang peserta yang menanyakan bagaimana cara melakukan penggandaan “*data cell*”, atau cara memasukan rumus ( $fx$ ), penggandaan “*sheet*”, melakukan penjumlahan, dan perintah sederhana lainnya. Sehingga waktu yang tersedia terbuang untuk menjelaskan kembali fungsi atau perintah sederhana. Di satu sisi hal ini juga menunjukkan bahwa pentingnya kemampuan komputer dasar dikalangan pengelola modul yang umumnya terdiri dari staf pengajar.

## 2. Penguasaan terhadap cara perhitungan *unit cost*

Menghitung *unit cost* khususnya yang berdasarkan *activity based costing* memiliki cara perhitungan sendiri dimana setiap data yang ada dikelompokkan, dilakukan perincian, serta memasukan nilai sesuai perincian, lalu dilakukan perhitungan sesuai rumus. Meskipun sebelum melakukan simulasi perhitungan, pemberi materi telah menunjukkan langkah langkah perhitungan secara rinci, akan tetapi langkah-langkah perhitungan tersebut tidak sepenuhnya dapat ditangkap dengan baik oleh

**Universitas Indonesia**

sebagian peserta. Hal inilah yang menjadi salah satu penyebab proses perhitungan menjadi lebih lama untuk sebagian peserta.

3. Penguasaan perangkat lunak penghitung *unit cost*

Perangkat lunak penghitung *unit cost* ini untuk pertama kalinya digunakan oleh semua peserta. Sebagaimana layaknya suatu program atau perangkat lunak lainnya perlu waktu untuk menguasai cara penggunaan program tersebut. Seperti halnya pada penggunaan perangkat lunak ini dimana tampilan kurang familiar dari pada penggunaan *spread sheet*, setidaknya membutuhkan waktu untuk mencerna terlebih dahulu, meskipun langkah langkah penggunaan telah diberikan dan diajarkan terlebih dahulu sebelum melakukan simulasi.

4. Adanya *fault* pada perangkat lunak

Seperti yang telah disebutkan dibagian sebelumnya, bahwa terdapat beberapa *fault* yang menghambat proses pengerjaan simulasi perhitungan ini. Mengulang langkah pekerjaan dari awal akibat *fault* yang ditimbulkan, atau tidak dapat meneruskan langkah pengerjaan akibat *fault* yang ada mengakibatkan pengerjaan memakan waktu lebih lama.

5. Kurangnya fitur tertentu pada perangkat lunak

Salah satu fitur yang tidak ada dari perangkat lunak ini adalah, tidak adanya fitur mundur (*back*) dan perbaikan kesalahan untuk setiap data yang dimasukkan sehingga jika terdapat kesalahan pemasukkan data, peserta harus mengulang langkah dari awal lagi. Hal ini menyebabkan pengerjaan menjadi lebih lama, mengingat peserta baru pertama kali menggunakan perangkat lunak tersebut, dapat dipastikan akan

**Universitas Indonesia**

melakukan kesalahan. Sehubungan dengan kemungkinan munculnya kesalahan tersebut, maka perlu adanya *fitur* yang menunjang perbaikan kesalahan tersebut.

### 7.3.5. Kesulitan Penggunaan Perangkat Lunak Penghitung *Unit Cost*

Kesulitan yang dijumpai oleh para peserta sebagai pengguna ini dianggap sebagai kekurangan dari perangkat lunak ini. Beberapa kesulitan yang ditemui dari penggunaan perangkat lunak penghitung ini.

#### 1. Harus dilakukan instalasi JVM terlebih dahulu

Pemasangan JVM diperlukan karena perangkat lunak dirancang dengan menggunakan bahasa pemrograman Java, dan membutuhkan suatu perangkat untuk menjalankan atau membaca bahasa pemrograman tersebut.

Hal ini menyebabkan perangkat lunak tidak praktis dapat digunakan begitu saja pada setiap komputer. Perangkat lunak dirancang menggunakan bahasa pemrograman Java mengingat bahasa pemrograman ini dapat dijalankan dimanapun dan disembarang *platform* apapun, sehingga menghemat biaya.

#### 2. Tampilan kurang familiar

Bila dibandingkan dengan *spreadsheet* tampilan perangkat lunak ini kurang familiar, sehingga peserta mengalami hambatan untuk menggunakannya. Hal ini akan segera teratasi dengan proses pengenalan dan pelatihan untuk pengguna yang lebih intensif, sehingga peserta menjadi lebih terbiasa.

#### 3. Tidak dapat dilakukan penggabungan data

Berbeda dengan menggunakan *spreadsheet*, data yang terdapat pada 1 file dari perangkat lunak ini tidak dapat digabungkan dengan data dari file lain.

**Universitas Indonesia**

Tidak adanya fitur *merge* atau penggabungan menjadi satu ketidakpraktisan dari perangkat lunak ini. Akibat dari ketidakpraktisan ini jika ada penambahan data baru atau perubahan masukan pada data sebelumnya tidak mungkin dilakukan, sehingga pengguna harus mengulang langkah pengerjaan dari awal kembali.

Penambahan fitur *merge* pada perangkat lunak ini dapat memberikan nilai tambah tersendiri untuk segi kemudahan proses pemasukan data, sehingga pengguna menjadi lebih mudah untuk melakukan perhitungan dengan menggunakan format data yang terdahulu.

4. Perbaiki kesalahan tidak praktis

Seperti yang telah dijelaskan sebelumnya, tidak adanya fitur mundur atau “*back*” menyebabkan pengguna harus mengulang langkah dari awal lagi jika melakukan kesalahan. Proses pengerjaan yang mengulang ini menyebabkan waktu pengerjaan jadi lebih lama, adanya fitur yang memungkinkan untuk kembali ke langkah pengerjaan sebelumnya memudahkan pengguna untuk melakukan perbaikan pada setiap kesalahan yang dilakukan.

5. Versi cetak hanya menampilkan hasil perhitungan jumlah *variable cost* dan *fixed cost* saja bukan hasil perhitungan *unit cost*

Jika *spreadsheet* menampilkan seluruh data yang dimasukkan, serta langkah langkah pengerjaannya secara detail, maka tidak demikian halnya pada versi cetak dari perangkat lunak. Versi cetak dari perangkat lunak ini hanya menampilkan hasil perhitungan jumlah *variable cost* dan *fixed cost* saja, sementara hasil perhitungan *unit cost* itu tidak muncul. Seperti yang

telah disebutkan sebelumnya, versi cetak ini merupakan salah satu *fault* yang ada pada perangkat lunak ini

### 7.3.6. Kemudahan Penggunaan Perangkat Lunak Penghitung *Unit Cost*

Kemudahan yang dijumpai oleh para peserta sebagai pengguna ini dianggap sebagai kelebihan dari perangkat lunak ini. Beberapa kemudahan yang ditemui dari penggunaan perangkat lunak penghitung ini.

#### 1. Data terkelompok secara otomatis

Berbeda dengan menggunakan *spreadsheet* data yang dimasukkan sudah terkelompokkan secara otomatis. Kolom isian pada perangkat lunak ini dirancang sedemikian rupa mengikuti langkah-langkah perhitungan *unit cost*. Pengguna cukup memasukkan data sesuai dengan kolom isian yang ada tanpa harus mengelompokkan kembali seperti halnya yang harus dilakukan pada penggunaan *spreadsheet*.

#### 2. Perhitungan dilakukan secara otomatis

Dalam perangkat lunak ini pengguna tidak perlu menghitung atau memasukkan rumus perhitungan. Perangkat lunak akan melakukan perhitungan *unit cost* secara otomatis. Dengan adanya perhitungan otomatis ini diharapkan dapat mengurangi kesalahan perhitungan yang disebabkan oleh kesalahan rumus yang dimasukkan. Jika pengguna telah menguasai penggunaan perangkat lunak ini, diharapkan dengan adanya perhitungan otomatis, perhitungan *unit cost* dapat lebih mudah dijalankan.

#### 3. Input data tidak perlu berulang

Menghitung *unit cost* modul pendidikan kedokteran, tidak hanya menghitung nilai-nilai yang tercantum, tetapi harus menampilkan urutan pengerjaan sesuai dengan kelompok data yang ada. Langkah pengerjaan seperti ini menyebabkan adanya input kelompok data yang berulang. Hal ini mempengaruhi terhadap banyaknya kolom baik *data cell* atau *sheet* ada. Semakin banyak kelompok data, semakin banyak *data cell* atau *sheet* yang harus diisikan. Berbeda dengan menggunakan perangkat lunak penghitung *unit cost*, kelompok data cukup sekali dimasukkan dan ditandai selanjutnya telah terkelompok secara otomatis dan siap untuk dimasukkan nilai yang akan dihitung.

#### 4. Tampilan lebih sederhana

Seperti yang telah disebutkan di atas, perhitungan menggunakan *spreadsheet* akan tergantung dengan jumlah kelompok data yang dimasukkan. Semakin banyak kelompok data, semakin banyak *data cell* atau *sheet* yang harus diisikan. Hal ini menyebabkan tampilan tampak lebih rumit. Pada perangkat lunak penghitung *unit cost*, pengisian data dilakukan secara bertahap, sehingga data yang ditampilkan lebih sedikit dan tampilan menjadi lebih sederhana.

### 7.4. Validasi dan Verifikasi

Pembuatan perangkat lunak baru tidak terlepas dari validasi, yaitu apakah program yang dibuat sudah sesuai dengan yang diinginkan oleh pengguna. Dengan adanya validasi dapat ditentukan apakah perangkat lunak layak untuk digunakan atau tidak. Selain validasi, perangkat lunak juga harus diverifikasi, yaitu sudah dipastikan terbuat dengan spesifikasi yang jelas. Jika tidak

memenuhi validasi dan verifikasi, maka suatu perangkat lunak akan disempurnakan kembali sehingga sesuai dengan yang diinginkan oleh pengguna.<sup>3</sup>

Verifikasi perangkat lunak penghitung unit cost ini adalah apabila dijalankan pada sistem operasi setingkat Windows XP atau di atasnya, dengan JVM yang digunakan setingkat versi 1.6 atau di atasnya, dan telah terinstalasi MySQL. Perangkat lunak penghitung unit cost pada prinsipnya mampu melakukan perhitungan seperti yang diharapkan pada tujuan perancangannya, akan tetapi masih dijumpai fault dan terdapat fitur yang belum lengkap sehingga menimbulkan kesulitan bagi penggunanya, untuk itu perlu diperbaiki kembali fault yang ditemui dan fitur dilengkapi.

#### **7.5. Usulan Perbaikan atau Penambahan Fitur**

Selain perbaikan *fault* yang ada, agar perangkat lunak ini lebih mudah dijalankan oleh pengguna perlu dilakukannya perbaikan fitur. Perbaikan fitur ini tentunya merupakan hasil kesepakatan antara analis sistem (berdasarkan hasil uji coba kepada pengguna dan konsultasi dengan ahli perhitungan *unit cost*), perancang program dan programmer yang membuat perangkat lunak tersebut.

Beberapa fitur yang perlu ditambahkan atau di perbaiki antara lain sebagai berikut.

1. Langkah mundur (*back*): adanya fitur *back* memudahkan pengguna untuk kembali ke langkah sebelumnya jika melakukan kesalahan, tanpa harus menyelesaikan seluruh perintah.

---

<sup>3</sup> Diakses dari [http://www.ece.cmu.edu/~koopman/des\\_s99/verification/index.html](http://www.ece.cmu.edu/~koopman/des_s99/verification/index.html) [4 Mei 2009]



2. Penggabungan data (*data merge*): untuk mempercepat proses penggabungan data, sehingga untuk data yang telah ada tidak perlu diinput ulang.
3. Adanya template rincian pembiayaan: pengguna cukup menandai rincian pembiayaan yang sesuai, serta memasukkan nilainya, dan menandai apakah pembiayaan itu termasuk *variable cost* atau *fixed cost*. Tentunya hal ini dapat dilakukan apabila struktur atau kelompok kegiatan beserta rincian kegiatan dan rincian pembiayaannya, sudah disepakati oleh FKUI beserta sistem analis, perancang program, dan programmer.
4. Terdapat pilihan *sharing* biaya: jika ada komponen biaya yang harus dikeluarkan memerlukan sistem *sharing*, pilihan ini akan memudahkan pengguna untuk menelusuri pola *sharing* yang dilakukan. Tentu saja hal ini dapat dilakukan apabila sistem *sharing* tersebut sudah disepakati oleh FKUI.
5. Perbaikan kesalahan yang lebih praktis: jika pengguna melakukan kesalahan input data, dapat dilakukan pada titik itu juga tanpa harus menyelesaikan terlebih dahulu, sehingga tidak perlu mengulang langkah pengerjaan.
6. Separasi angka dan pembatasan jumlah digit desimal: untuk memudahkan pembacaan nilai hasil perhitungan, maka sebaiknya angka yang tercantum diberi separasi koma atau titik, serta nilai desimal yang dicantumkan cukup dua digit dibelakang koma atau titik.

#### **7.6. Usulan Perbaikan Kualitas Sumber Daya Manusia**

Pada hasil observasi, didapatkan perbedaan yang mencolok antara peserta yang memiliki kemampuan komputer dasar ataupun peserta yang memiliki dasar pengetahuan keuangan. Adanya perangkat lunak apapun tidak dapat membantu banyak apabila tidak ditunjang dengan kemampuan dari pengguna itu sendiri.

**Universitas Indonesia**

Sehubungan dengan hal itu hendaknya pihak FKUI memperbaiki kualitas SDM pengelola modul, baik dengan pelatihan komputer dasar atau pelatihan pembiayaan. Selain itu sebaiknya terdapat tenaga administrasi khusus yang bertanggung jawab untuk melakukan perhitungan *unit cost*.

### **7.7. Usulan Pengujian Kompatibilitas Dengan Sistem Operasi Lainnya**

Sistem operasi windows, merupakan sistem operasi berlisensi. Untuk menghemat biaya dapat digunakan sistem operasi *open-source*. Beberapa komputer telah dilengkapi dengan sistem operasi sendiri yang juga tidak berbayar misalkan Appel Mac. Untuk memastikan apakah perangkat lunak ini dapat dijalankan pada berbagai sistem operasi, maka perlu dilakukan pengujian kembali di berbagai sistem operasi yang berbeda.

### **7.8. Usulan Pengujian Ulang Setelah Perbaikan**

Meskipun telah diperbaiki segala *fault* dan kekurangan fitur yang ada, tetap saja perangkat lunak ini harus diuji kembali penggunaannya. Hal ini untuk meyakinkan apakah perangkat lunak tersebut sudah sesuai dengan yang diinginkan. Jika waktu pengujian tidak terbatas, maka uji *non-functional* lainnya seperti *performance*, *security*, *availability*, *reliability*, dan *capacity* hendaknya dilakukan.

### **7.9. Usulan Pelatihan Penggunaan Perangkat Lunak**

Setelah dilakukan perbaikan dan sudah dapat diputuskan oleh pengguna bahwa perangkat lunak layak digunakan, maka hal yang perlu dilakukan selanjutnya adalah pelatihan penggunaan perangkat lunak tersebut. Pelatihan berguna agar pengguna dapat memahami segala fitur yang ada, sehingga kesulitan penggunaan dapat teratasi.

### **7.10. Pengembangan Aplikasi**

Perangkat lunak penghitung *unit cost* modul pendidikan kedokteran ini dapat dikembangkan atau dimanfaatkan juga pada perhitungan *unit cost* untuk menentukan tarif rumah sakit atau klinik, serta dikembangkan untuk terintegrasi dengan sistem informasi rumah sakit.