



JURNAL TEKNOLOGI MAKLUMAT DAN SAINS KUANTITATIF

KANDUNGAN

Muka Surat

To (Start With) OOP, Or Not OOP: That is Not The Question <i>Syed Ahmad Aljunid</i>	1
Penilaian Tingkah Laku Taklinear Menggunakan Kaedah Empirik <i>Norazan Mohamed Ramli Habsah Midi</i>	19
Towards Developing A Risk Charter for Software Development Projects <i>Noor Habibah Arshad</i>	37
Comparative Performance of Computational Techniques in Retrieving Malay Text <i>Zainab Abu Bakar</i>	51
Algorithm of Magnetic Flux Density on a Plane Generated by a Finite Length Current Source <i>Rashdi Shah Ahmad, Tahir Ahmad, Chew Soon Leong</i>	63
Suatu Kaedah Menganggar Kos Perisian Berasaskan Spesifikasi Formal <i>Abdullah Mohd Zin, Maridah Mohamad Shah, Abd Malik Md Yusof</i>	75
Solving a Constraint Satisfaction Problem by Backtracking Intelligently: A Case Study <i>Muthukkaruppan Annamalai</i>	87
An Empirical Investigation into the Critical Success Factors Used by it Companies of Various Sizes to Adopt Internet Technology <i>Lloyd Tam Yew King</i>	99
Teknologi Maklumat dan Telekerja: Satu Tinjauan Awalan dan Implikasinya di Malaysia <i>Balakrishnan Parasuraman</i>	115

SUATU KAEDAH MENGANGGAR KOS PERISIAN BERASASKAN SPESIFIKASI FORMAL

Abdullah Mohd Zin

Maridah Mohamad Shah

Abd Malik Md Yusof

Kumpulan Penyelidikan Pengaturcaraan
Fakulti Teknologi dan Sains Maklumat
Universiti Kebangsaan Malaysia, Bangi

ABSTRAK

Suatu masalah yang sering dihadapi oleh pengurus pembangunan projek perisian ialah tentang anggaran kos bagi membangunkan perisian tersebut. Kertas ini mencadangkan suatu kaedah untuk menganggar kos berasaskan spesifikasi formal Z. Bagi membolehkan kaedah ini digunakan, satu teknik untuk membuat anggaran bilangan baris kod yang dihasilkan oleh suatu perisian telah dikemukakan. Daripada bilangan baris kod ini, anggaran kos dapat dilakukan dengan menggunakan teknik sedia ada, seperti COCOMO. Bagi menunjukkan keberkesanan teknik ini, lima kajian kes telah dilakukan. Kajian kes tersebut menunjukkan bahawa teknik yang dicadangkan ini memberikan hasil yang memberansangkan.

Katakunci: *Cocomo; Pembangunan perisian; Spesifikasi formal Z*

LATARBELAKANG

Seorang Pengurus Projek yang bertanggungjawab untuk mendirikan bangunan ataupun membina jambatan pasti inginkan projek beliau siap pada masa yang telah ditetapkan dan dalam lingkungan kos yang dianggarkan. Kelewatan yang dialami diiringi dengan kos yang bertambah akan hanya merugikan organisasi berkenaan. Begitu juga dengan Pengurus Projek perisian, beliau harus memiliki kebolehan dalam menganggar kos dan tenaga kerja yang diperlukan sebelum suatu projek dilaksanakan.

Terdapat pelbagai kos yang terlibat dalam membangunkan perisian. Antaranya ialah kos kakitangan secara langsung, kos kakitangan secara tidak langsung, kos perjalanan, kos grafik dan mencetak, kos ruang pejabat, kos perkakasan, kos pembekalan, kos latihan dan kos perisian yang khas. Kos kakitangan secara langsung merupakan kos yang membabitkan kakitangan pada setiap peringkat sepanjang jangka masa projek berkenaan dijalankan. Untuk mendapatkan kos kakitangan secara langsung tidaklah mudah jika dibandingkan dengan kos yang lain. Bagi memperolehi kos seorang juruaturcara menulis suatu aturcara, ia membabitkan jangka masa yang diperlukan oleh beliau untuk menyiapkan aturcara dan juga saiz aturcara yang telah dibangunkan.

Teknik bagi membuat anggaran kos yang popular pada masa ini ialah dengan berasaskan pengiraan bilangan baris kod atau berasaskan titik fungsi. Bilangan baris kod adalah berlandaskan saiz perisian manakala titik fungsi lebih menitikberatkan kepada fungsi perisian berkenaan. Dengan erti kata lain, titik fungsi mengambilkira input dan output pengguna, fail logikal, pertanyaan dan antaramuka luaran. Kebaikan dan keburukan penggunaan bilangan baris kod berbanding dengan titik fungsi telah dibincangkan oleh ramai penyelidik, misalnya oleh Low dan Jeffrey (1990) dan Jones (1986). Walaupun penggunaan titik fungsi telah menjadi semakin popular, namun dalam kebanyakan projek perisian, anggaran masih dilakukan dengan menggunakan bilangan baris kod memandangkan teori dan model yang bersangkutan dengannya adalah lebih kukuh.

Antara model yang banyak digunakan bagi menganggar kos ialah model COCOMO yang telah dikemukakan oleh Boehm (1981). Disebabkan model ini agak telah lama, beberapa usaha telah dilakukan untuk mengemaskinikan model ini dengan menghasilkan beberapa model anggaran kos yang baru. Walau bagaimanapun, model COCOMO masih popular dan masih digunakan sebagai asas bagi menganggar kos perisian. Dalam model COCOMO, Boehm telah mengelaskan projek perisian kepada tiga kelas iaitu mod organik, mod semiterpisah dan mod terbenam. Persamaan asas COCOMO adalah seperti berikut :-

$$E = a_p (KLOC) \exp(b_p)$$
$$D = c_p (E) \exp(d_p)$$

dengan E ialah usaha dalam bulan manusia, D ialah masa yang diambil untuk membangunkan sistem dalam kiraan bulan dan KLOC ialah anggaran bilangan baris kod yang dihantar dalam kiraan ribu. Koefisien a_p dan c_p berbeza-beza mengikut jenis perisian. Nilai koefisien dan contoh pengiraan COCOMO boleh didapati daripada Boehm (1981).

PERNYATAAN MASALAH

Teknik anggaran kos seperti COCOMO didapati berhasil membuat anggaran kos yang agak tepat. Anggaran tersebut dilakukan berdasarkan kepada bilangan baris kod telah diketahui, seperti yang ditunjukkan dalam Rajah 1.



Rajah 1: Kaedah anggaran kos perisian berasaskan COCOMO

Masalah utama dengan kaedah anggaran kos di atas ialah bilangan baris kod ini hanya akan diketahui apabila perisian tersebut telah selesai dibangunkan. Dalam kebanyakan pembangunan projek perisian, seperti juga pembangunan projek yang lain, anggaran kos perlu dilakukan pada awal-awal lagi, iaitu sebelum projek tersebut dibangunkan. Pihak pelanggan biasanya akan menyediakan satu dokumen spesifikasi perisian. Berdasarkan kepada dokumen spesifikasi tersebut, syarikat pemaju perisian perlu menyatakan anggaran kos bagi perisian tersebut, seperti yang digambarkan dalam Rajah 2.

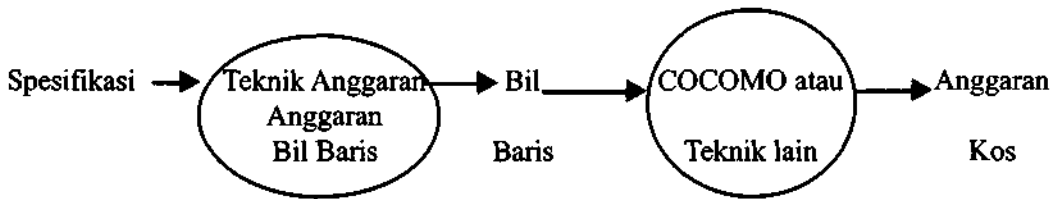


Rajah 2: Kaedah anggaran kos yang diperlukan

Syarikat pemaju perlulah mempunyai kemampuan untuk menganggarkan kos pembangunan setepat mungkin bagi meletakkan harga untuk pembangunan projek tersebut. Jika anggaran kos terlalu rendah, syarikat tersebut akan mengalami kerugian, sedangkan jika harganya terlampau tinggi, kemungkinan besar syarikat tersebut tidak akan memperolehi projek tersebut.

CADANGAN PENYELESAIAN

Disebabkan anggaran kos perisian berdasarkan bilangan baris kod seperti yang dicadangkan oleh COCOMO telah berupaya memberikan anggaran kos pembangunan perisian yang tepat, masalah sebenar yang dihadapi oleh pengurus pembangunan projek perisian ialah untuk mendapatkan anggaran bilangan baris kod yang akan dihasilkan oleh suatu perisian berdasarkan kepada spesifikasi perisian tersebut, seperti yang ditunjukkan dalam Rajah 3.



Rajah 3: Kaedah anggaran kos perisian

Teknik yang biasa diguna bagi mendapatkan anggaran bilangan baris kod ialah dengan menggunakan pengalaman, yakni berasaskan kepada kes yang lampau. Namun begitu, teknik ini tidak dapat memberikan anggaran yang tepat kerana projek baru tidak semestinya sama dengan projek yang lampau.

Teknik yang dicadangkan dalam penyelidikan ini adalah berdasarkan kepada suatu hipotesis bahawa jika spesifikasi perisian dinyatakan dengan menggunakan tatatanda formal, misalnya Z (Spivey,1988), maka bilangan baris kod bagi perisian tersebut akan dapat dianggarkan, dan justru itu anggaran kos perisian dapat diperolehi dengan lebih tepat.

Spesifikasi formal ialah suatu kaedah bagi menyatakan spesifikasi bagi suatu perisian dengan menggunakan tatatanda matematik. Selain daripada Z, tatatanda formal yang lain ialah seperti Vienna Development Method atau VDM (Jones,1990) dan Communicating Sequential Processes atau CSP (Hoare,1985).

TEKNIK ANGGARAN BILANGAN BARIS KOD BERASASKAN SPESIFIKASI FORMAL Z

Satu spesifikasi formal Z adalah satu model abstrak bagi satu perisian yang berasaskan kepada teori set. Dokumen spesifikasi ini terdiri daripada beberapa perenggan formal dan perenggan tidak formal. Perenggan formal dinyatakan dengan menggunakan skema, manakala perenggan tidak formal dinyatakan dengan menggunakan bahasa tabii. Terdapat dua jenis skema, iaitu skema keadaan dan juga skema operasi. Skema keadaan menggambarkan ciri-ciri sistem yang tidak berubah walaupun selepas suatu operasi dilaksanakan. Skema operasi pula menerangkan tindakan yang terjadi apabila suatu operasi dilaksanakan. Bilangan skema yang terdapat dalam satu spesifikasi bergantung kepada saiz masalah yang hendak diselesaikan.

Satu skema Z terdiri daripada beberapa pernyataan Z. Satu contoh skema untuk menambah pengguna bagi suatu sistem ditunjukkan dalam Rajah 4.

TambahPegguna Δ DSkemaKeadaan baru? : ID_STAF
baru? \in pengguna dalam' = dalam keluar' = keluar \cup {baru?} pengguna' = pengguna \cup {baru?}

Rajah 4: Satu contoh skema Z

Bahagian atas garisan tengah dalam skema ini dikenali sebagai "signatur", manakala dibahagian bawah garisan tengah menyenaraikan predikat yang berkaitan dengan skema tersebut.

Proses untuk menjelmakan suatu aturcara daripada suatu perisian dikenali sebagai proses penghalusan. Semasa proses ini, spesifikasi perisian yang berada dalam keadaan abstrak diperhalus sehingga kita mendapat spesifikasi yang konkrit. Spesifikasi ini kemudiannya diperhalus lagi dengan membuat pelbagai tambahan rekabentuk sehingga kita akhirnya memperolehi suatu aturcara (Morgan dan Woodcock, 1990).

Andaian

Daripada cerapan yang dilakukan ke atas proses penghalusan, kami mengemukakan enam andaian telah dibuat dalam mendapatkan bilangan baris kod daripada suatu spesifikasi Z. Kesemua andaian ini ialah:

Pertama, skema keadaan tidak akan menjanakan sebarang baris kod bagi suatu perisian. Hanya skema operasi sahaja yang diambilkira dalam menghitung bilangan baris kod. Andaian ini dibuat kerana suatu aturcara sebenarnya adalah suatu pengimplementasian bagi suatu operasi.

Kedua, bahagian signatur bagi skema operasi tidak diambilkira dalam pengiraan bilangan baris kod dan begitu juga bahagian deklarasi di dalam aturcara yang ditulis bagi sesuatu sistem. Andaian kedua ini dibuat memandangkan signatur hanya menyatakan mengenai deklarasi sesuatu aturcara.

Andaian ketiga melibatkan penggunaan komen dan baris kosong. Baris yang mengandungi komen tidak di ambilkira dalam menghitung bilangan baris kod bagi suatu tata tanda Z dan baris kod sebenar di dalam setiap kajian kes. Menurut Conte et al (1986), kebanyakan penyelidik bersetuju bahawa ukuran bilangan baris kod tidak termasuk komen atau baris kosong kerana ianya merupakan dokumentasi dalaman. Kehadiran atau kewujudannya tidak akan menjejaskan fungsi sesuatu aturcara.

Andaian keempat ialah arahan gelungan iaitu 'WHILE...DO', 'IF...THEN ... ELSE'

dan 'FOR...DO' adalah konsisten dari segi susunan baris aturcara yang ditulis. Selain daripada itu, semua pernyataan dalam bahasa pengaturcaraan (misalnya Pascal) yang berasingan ditulis pada baris yang berasingan.

Andaian kelima ialah antaramuka pengguna adalah yang paling mudah. Ini disebabkan spesifikasi formal mengutamakan fungsi sesuatu sistem dan bukannya dari segi paparan.

Andaian keenam menyatakan bahawa keadaan komponen tidak berubah selepas suatu operasi dilaksanakan ke atasnya. Pernyataan Z yang sedemikian tidak menjanakan sebarang baris kod. Berikut diberikan contoh pernyataan Z berkenaan:-

$$\text{pengguna}' = \text{pengguna}$$

Teknik yang Dicadangkan

Suatu spesifikasi sistem yang dinyatakan dalam spesifikasi formal Z biasanya terdiri daripada beberapa skema. Satu spesifikasi yang mempunyai bilangan skema yang banyak akan menghasilkan aturcara yang besar, iaitu aturcara yang mempunyai bilangan baris kod yang lebih banyak. Jelas di sini, bahawa terdapat suatu hubungan antara skema dan baris kod.

Sehubungan dengan itu, kami mencadangkan bilangan baris kod yang akan dihasilkan bagi perisian berkenaan adalah hasil tambah bilangan baris kod yang dihasilkan oleh semua skema bagi spesifikasi Z tersebut.

$$\text{Bilangan baris kod} = \sum l(\text{skema}), \forall \text{skema dalam spesifikasi}$$

dengan $l(\text{skema})$ ialah bilangan baris kod untuk suatu skema. Bilangan baris kod yang dihasilkan oleh suatu skema pula ialah hasil tambah bagi bilangan baris kod yang dihasilkan bagi setiap pernyataan dalam skema berkenaan.

$$l(\text{skema}) = \sum f(\text{baris}), \forall \text{baris dalam skema}$$

dengan $f(\text{baris})$ ialah jumlah bilangan baris kod di dalam suatu skema spesifikasi Z. Bilangan baris kod yang dihasilkan oleh suatu pernyataan Z adalah jumlah baris kod yang dijanakan oleh setiap operator Z yang wujud dalam pernyataan tersebut.

$$f(\text{baris}) = \sum h(\text{op}), \forall \text{op dalam baris}$$

Di sini, diperhatikan bahawa bilangan baris kod yang dihasilkan oleh setiap operator Z perlu diperolehi terlebih dahulu bagi mendapatkan bilangan baris kod bagi suatu baris pernyataan Z, diikuti oleh bilangan baris kod bagi suatu skema dan akhir sekali jumlah baris kod untuk perisian berkenaan.

Bilangan Baris kod untuk setiap Operator Z

Bagi memperolehi bilangan baris kod bagi setiap operator dalam Z, kami telah menulis aturcara bagi setiap operator Z yang sering digunakan semasa menulis spesifikasi Z. Ciri-ciri bagi operator Z ini diterapkan dalam aturcara berkenaan. Misalnya aturcara untuk operasi kesatuan akan membabitkan algoritma dengan dua set akan digabungkan dengan menghasilkan set baru. Ciri bagi kesatuan dikekalkan iaitu tidak terdapat dua ahli yang sama di dalam set berkenaan.

Jadual 1 merupakan sebahagian daripada hasil yang telah diperolehi. Penggunaan operator \in atau \notin , misalnya menghasilkan 13 baris kod dalam bahasa Pascal.

Jadual 1: Bilangan Baris kod Bagi Sebahagian Operator Z

Operator	Bilangan baris kod (dalam Pascal)
\in, \notin	13
$\subset, \not\subset$	24
\cup	14
\cap	30
\oplus	24
dom R	7
ran R	7
?	2

Bilangan Baris kod Untuk Setiap baris Pernyataan Z

$f(\text{baris})$ ialah jumlah baris kod yang dihasilkan oleh suatu baris pernyataan Z. Suatu baris pernyataan Z biasanya terdiri daripada beberapa tata tanda Z. Sebagai contoh diterangkan cara menghitung nilai $f(\text{baris})$ bagi pernyataan Z dalam skema **TambahPengguna** yang ditunjukkan dalam Rajah 4

baru? \in pengguna

Pernyataan Z di atas mempunyai dua tata tanda Z iaitu '?' dan ' \in ' dan hanya wujud satu kejadian bagi setiap tata tanda Z berkenaan. Berdasarkan kepada Jadual 1, bilangan baris kod yang sepadan dengan tata tanda '?' ialah 2 manakala tata tanda ' \in ' ialah 13. Maka jumlah bilangan baris kod yang dihasilkan oleh baris pernyataan Z ini ialah :-

$$\begin{aligned} f(\text{baris}) &= (2 \times 1) + (13 \times 1) \\ &= 15 \text{ bilangan baris kod} \end{aligned}$$

Bilangan Baris kod Untuk setiap Skema

Skema **TambahPengguna** mengandungi empat baris pernyataan Z. Pengiraan baris kod bagi setiap baris ditunjukkan dalam Jadual 2. Jumlah baris kod yang akan dihasilkan ialah hasil tambah kesemua bilangan baris kod bagi setiap baris, iaitu 43 baris.

Jadual 2: Kaedah Menghitung Bilangan Baris kod Bagi Suatu Skema

Baris	Operator	Bil * Bil baris kod	Jumlah baris kod
1	?	$1 * 2 = 2$	15
	∈	$1 * 13 = 13$	
2	=	0	0
3	∪	$1 * 14 = 14$	14
4	∪	$1 * 14 = 14$	14
			43

Bilangan Baris Kod bagi suatu Spesifikasi Z

Jumlah bilangan baris kod bagi suatu spesifikasi dapat diperolehi dengan menjumlahkan kesemua bilangan baris kod bagi setiap skema.

KAJIAN KES

Bagi menunjukkan keberkesanan teknik yang dicadangkan, lima kajian kes yang merupakan spesifikasi bagi lima sistem yang disediakan dengan menggunakan spesifikasi formal Z telah dikaji. Kesemua spesifikasi ini telah diambil daripada spesifikasi contoh yang diberikan dalam beberapa buku teks, antaranya ialah yang ditulis oleh Ince (1992) dan Norcliffe & Slatter (1991). Keterangan ringkas bagi kelima-lima spesifikasi bagi kajian kes ini adalah seperti berikut:

Kajian Kes 1:

Kajian Kes ini ialah mengenai suatu sistem perpustakaan yang membabitkan lima fungsi utama iaitu pendaftaran buku baru, pendaftaran pengguna, pinjaman, pemulangan dan penempahan buku. Pertanyaan mengenai senarai pinjaman dan penempahan yang telah dibuat oleh pengguna boleh juga dilakukan menerusi sistem ini. Terdapat beberapa skema daripada spesifikasi yang tidak dimasukkan, umpamanya skema yang membabitkan keperluan untuk menjana nombor buku pendaftaran dan senarai nombor pengguna. Namun demikian, secara keseluruhan ia masih tetap merupakan suatu sistem yang lengkap.

Kajian Kes 2:

Sistem Panduan Telefon telah dipilih bagi kes kajian yang kedua. Sistem ini membolehkan sama ada maklumat pengguna ditambah, disingkirkan dan juga dikemaskinikan. Seperti kebiasaan sistem yang lain, pertanyaan juga merupakan salah satu fungsi bagi sistem ini di mana pertanyaan mengenai nombor telefon bagi seseorang pengguna boleh juga diperolehi melalui sistem ini.

Kajian Kes 3:

Kajian kes ini merupakan spesifikasi Z mengenai sistem perbankan yang mudah. Di sini, keperluan utama bagi sistem ini ialah menyimpan wang dan mengeluarkan wang, pembukaan, penutupan akaun dan akhir sekali ialah pertanyaan mengenai senarai semua pemegang akaun bagi Bank berkenaan.

Kajian Kes 4:

Kajian kes yang keempat ialah spesifikasi mengenai sistem keselamatan bagi sebuah bangunan. Terdapat tiga ciri utama bagi keperluan sistem ini. Pertamanya sistem ini boleh mencatat kesemua aktiviti keluar masuk pengguna daripada bangunan berkenaan. Keduanya sistem ini boleh menambah dan menyingkir maklumat pengguna bangunan tersebut. Akhir sekali, tiga pertanyaan boleh dilakukan iaitu pertanyaan mengenai pengguna sama ada yang berada di dalam bangunan, di luar bangunan dan juga senarai semua pengguna bangunan tersebut.

Kajian Kes 5:

Kajian kes yang terakhir merupakan spesifikasi bagi suatu aturcara untuk menghasilkan laporan bagi senarai pelajar-pelajar yang tidak menduduki sesuatu peperiksaan dan yang sebaliknya.

Keputusan

Kesemua spesifikasi tersebut telah dianalisis dengan menggunakan teknik yang dinyatakan di atas bagi mendapatkan bilangan baris kod sebagai nilai anggaran. Bagi mendapatkan bilangan baris kod sebenar yang dihasilkan bagi setiap spesifikasi tersebut, beberapa orang juruaturcara telah diminta untuk menulis aturcara Pascal bagi setiap spesifikasi berkenaan. Seterusnya, peratus perbezaan di antara nilai baris kod sebenar dan anggaran dikira. Pengiraan peratus tersebut dilakukan seperti berikut:

$$\text{Peratus perbezaan} = \frac{\text{Bil baris dianggar} - \text{Bil baris sebenar}}{\text{Bil baris sebenar}} * 100\%$$

Hasil yang telah diperolehi ditunjukkan di dalam Jadual 3.

Jadual 3: Keputusan Bagi Kajian Kes

Kes	Nama Aturcara	Bil baris dianggar	Bil baris sebenar	Peratus Perbezaan
1	pustaka.pas	1300	1203	8.1%
2	telefon.pas	238	223	6.7%
3	akaunbank.pas	173	184	-5.9%
4	selamat.pas	299	285	2.41%
5	ujian.pas	150	144	4.2%

Daripada Jadual 3, dapat diperhatikan bahawa julat peratus perbezaan ialah di antara -5.9% dan 8.1 % iaitu secara keseluruhannya mencapai kurang daripada 10%. Hanya satu kajian kes iatu kajian kes 3 yang memberikan nilai negatif.

KESIMPULAN

Kertas ini telah mengemukakan suatu cadangan penyelesaian kepada masalah penganggaran kos perisian sebelum suatu perisian tersebut dibangunkan. Cadangan ini mengemukakan suatu teknik untuk menganggar bilangan baris kod yang dihasilkan oleh suatu aturcara. Berasaskan bilangan baris kod ini, anggaran kos perisian dapat dilakukan dengan menggunakan teknik yang sedia ada, misalnya COCOMO. Teknik anggaran bilangan baris kod yang dicadangkan ini membuat andaian bahawa spesifikasi bagi aturcara yang hendak dibangunkan telah disediakan dengan menggunakan spesifikasi formal Z.

Hasil daripada penyelidikan telah menunjukkan bahawa peratus perbezaan antara bilangan baris kod yang dianggarkan dengan menggunakan teknik yang dicadangkan dan yang sebenar adalah dalam julat -5.9% dan 8.1%. Kedua-dua nilai ini boleh digunakan untuk mendapatkan anggaran optimistik dan pesimistik masing-masing bagi menganggar suatu spesifikasi sistem yang telah dinyatakan dalam tatatanda spesifikasi formal Z.

Dalam kajian ini, kami telah mengira bilangan baris yang dihasilkan oleh aturcara yang ditulis menggunakan bahasa Pascal. Disebabkan ciri-ciri yang terdapat dalam Pascal adalah hampir sama dengan ciri-ciri bahasa imperatif yang lain, kami percaya teknik ini juga dapat memberikan hasil yang sama jika bahasa imperatif yang lain digunakan.

Walau bagaimanapun, sebelum suatu kesimpulan yang lebih konkrit dapat dilakukan, beberapa penyelidikan tambahan masih perlu dilakukan. Spesifikasi dan aturcara yang dikaji dalam penyelidikan ini adalah kecil. Beberapa eksperimen perlu dilakukan dengan melibatkan masalah yang lebih besar bagi menentukan sama ada teknik yang dicadangkan ini benar-benar dapat diterima pakai.

RUJUKAN

- Boehm, B. 1981. *Software Engineering Economics*, Prentice Hall.
- Conte S.D., Dunsmore H. E. and Shen V.Y. 1986. *Software Engineering Metrics Models*, Benjamin/Cummings Publishing Company, Inc.
- Hoare, C. A. R. 1985. *Communicating Sequential Processes*, Prentice-Hall.
- Ince, D.C. 1992. *An Introduction to Discrete Mathematics, Formal System Specification and Z. Ed. ke-2*, Oxford:Clarenton Press.
- Jones, C. 1986. *Programming Productivity*, McGraw-Hill.
- Jones, C. B. 1990. *Systematic Software Development Using VDM, Ed ke 2*, Prentice-Hall.
- Low, C.G. and Jeffrey, D. R., 1990. Function points in the estimation and evaluation of the software process. *IEEE Transactions of Software Engineering*, 16(1):64-71.
- Morgan, C. dan Woodcock, J.P. 1991. *3rd Refinement Workshop : Proceedings of the 3rd Refinement Workshop*, British Computer Sociiety.
- Norcliffe, A. dan Slater, G. 1991. *Mathematics of Software Construction*, Ellis Horwood.
- Spivey, J.M. 1988. *Understanding Z: A Specification Language and Its Formal Semantics*, Cambridge University Press.