

PENILAIAN KEMAHIRAN PENGATURCARAAN: EKSPERIMEN TERHADAP PELAJAR MENGGUNAKAN PROSES KEJURUTERAAN SEMULA.

Zhamri Che Ani, ¹Zauridah Abdul Hamid, Mohd Zabidin Husin,
Mohamed Ali Saip

Fakulti Teknologi Maklumat
Universiti Utara Malaysia
{[@uum.edu.my](mailto:zhamri,zabidin,mdali)}

¹Fakulti Komunikasi dan Bahasa Moden
Universiti Utara Malaysia
zauree@uum.edu.my

ABSTRAK

Agak sukar untuk kita menilai tahap kemahiran pengaturcaraan setiap pelajar jika hanya merujuk kepada keputusan peperiksaan semata-mata. Oleh yang demikian, kertas kerja ini akan melaporkan hasil eksperimen yang telah dilakukan terhadap pelajar Program Sarjana Muda Teknologi Maklumat, pengkhususan dalam bidang Kejuruteraan Perisian. Seramai 33 orang pelajar yang mempunyai keputusan yang baik dalam Pengaturcaraan Java telah dipilih dan diuji dalam makmal menggunakan proses kejuruteraan semula. Pelajar diminta untuk membangunkan semula program menggunakan Java berpandukan program Ada yang disediakan. Hanya keputusan pengaturcaraan yang memenuhi spesifikasi ujian sahaja dinilai dan masa setiap pelajar yang berjaya direkodkan.

Kata Kunci: Kemahiran pengaturcaraan, eksperimen, proses kejuruteraan semula.

1. PENGENALAN

Kebanyakan kursus bidang Teknologi Maklumat dan Sains Komputer menekankan kepada teori dan kemahiran. Keberkesanan dalam teori dilihat dapat meningkatkan kemahiran seseorang pelajar dalam usaha memenuhi keperluan industri. Salah satu cara untuk memenuhi keperluan tersebut, pelajar-pelajar didedahkan dengan projek berkumpulan. Menurut Anewalt, Polack-Wahl, Beidler dan Smarkusky (2003), bekerja secara berkumpulan dapat mengurangkan beban kerja, membolehkan pelajar-pelajar berkongsi idea, membina kemahiran komunikasi dan meningkatkan kemahiran penyelesaian masalah dalam sesuatu projek.

Kemahiran berkomunikasi boleh diketahui ketika pelajar mempersembahkan projek. Begitu juga dengan penguasaan ilmu terutamanya daripada aspek teori. Ia boleh diukur daripada keputusan peperiksaan. Walaubagaimana pun, agak sukar untuk kita menilai tahap kemahiran pengaturcaraan setiap pelajar jika hanya merujuk kepada keputusan peperiksaan atau projek berkumpulan yang mereka hasilkan. Ini adalah kerana terdapat juga dikalangan pelajar yang mengambil kesempatan daripada projek berkumpulan ini. Kajian menunjukkan terdapat pelajar yang hanya bertindak sebagai penumpang dalam

sesuatu projek berkumpulan setelah penilaian antara rakan dilakukan (Dunsmore, Moffett, & Ward, 1989).

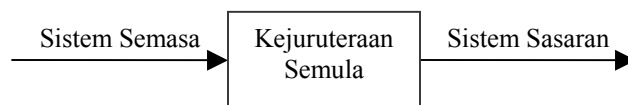
Justeru itu, eksperimen ini bertujuan untuk menilai kemahiran pengaturcaraan di kalangan pelajar. Pengkaji memilih proses kejuruteraan semula supaya pelajar boleh memberi lebih tumpuan kepada pengaturcaraan dengan adanya panduan daripada aturcara yang lain. Pelajar diberikan program yang ditulis dalam bahasa Ada dan diminta untuk membangunkan semula program berkenaan menggunakan bahasa Java. Menurut Prasad dan Fielden (2002), selain daripada bahasa yang diajar secara formal, pengaturcara yang baik boleh menguasai apa juga bahasa pengaturcaraan yang diberikan.

Seterusnya, kerja kerja ini akan membincangkan proses kejuruteraan semula di bab kedua, latarbelakang pelajar di bab ketiga, kaedah penilaian di bab keempat, keputusan dan perbincangan di bab kelima, kesimpulan dan cadangan di bab yang keenam.

2. PROSES KEJURUTERAAN SEMULA

Menurut (Pressman, 2004), proses kejuruteraan semula perisian adalah satu siri aktiviti yang melibatkan analisis inventori, penstrukturan semula dokumentasi, kejuruteraan sonsang (*reverse engineering*), aturcara, pengstruktur data dan kejuruteraan ke depan (*forward engineering*). Kejuruteraan semula akan meneliti sistem semasa untuk distrukturkan dan dibangunkan semula supaya dapat mempamerkan perisian yang lebih berkualiti (Byrne & Gustafson, 1992). Fungsi perisian tersebut tidak berubah dan kebiasaannya senibina sistem juga adalah sama (Sommerville, 2004). Ia tidak sepatutnya melibatkan penambahan ciri baru daripada pengguna (Lethbridge & Laganiere, 2001).

Terdapat tiga pendekatan dalam kejuruteraan semula perisian iaitu pendekatan *big bang* atau *lump sum*, pendekatan tokokan (*incremental*) dan pendekatan *evolutionary* (Rosenberg, 1996). Walau bagaimanapun, dalam eksperimen ini pengkaji memilih pendekatan *big bang* kerana pelajar membangunkan semula keseluruhan sistem semasa dan menggantikan dengan sistem yang baru dalam satu masa tertentu. Pendekatan ini selalu digunakan bagi sistem yang memerlukan penyelesaian dengan kadar segera (Rosenberg, 1996). Rajah 1 menunjukkan pendekatan *big bang*.



Rajah 1: Pendekatan *Big Bang*

Disebabkan aturcara yang diberikan bersaiz kecil, eksperimen ini hanya fokus kepada penjana kod baru sahaja. Menurut Rosenberg (1996), penjana kod baru memerlukan 26% daripada usaha proses kejuruteraan semula.

3. LATARBELAKANG PELAJAR

Eksperimen ini telah dijalankan terhadap pelajar Program Sarjana Muda Teknologi Maklumat, pengkhususan dalam bidang Kejuruteraan Perisian. Seramai 16 pelajar lelaki dan 17 pelajar perempuan yang mempunyai keputusan yang baik iaitu mendapat sekurang-kurangnya gred 'B' dan ke atas dalam kursus TA1023 Pengaturcaraan Lanjutan telah dipilih. Jadual 1 menunjukkan jantina dan gred pengaturcaraan lanjutan pelajar yang terlibat dalam eksperimen ini.

Jadual 1: Jantina dan Gred Pengaturcaraan Lanjutan

Jantina	Gred Pengaturcaraan Lanjutan				Jumlah
	A	A-	B+	B	
Lelaki	4	2	5	5	16
Perempuan	4	3	6	4	17
Jumlah	8	5	11	9	33

Pelajar-pelajar yang dipilih ini telah didedahkan dengan pengaturcaraan Java selama setahun semasa mengikuti kursus TA1013 Pengaturcaraan Awalan dan TA1023 Pengaturcaraan Lanjutan. Mereka juga telah diperkenalkan dengan bahasa Ada dalam kursus TW3053 Pengaturcaraan Masa-Nyata.

4. KAEDAH PENILAIAN

Untuk memastikan pelajar-pelajar ini lebih bersedia, mereka telah dimaklumkan dua minggu lebih awal sebelum eksperimen dilakukan. Semua pembantu pengkaji juga diberi taklimat awal mengenai tujuan eksperimen dan tugas-tugas yang perlu mereka lakukan. Juruteknik komputer diminta untuk memeriksa semua komputer yang bakal digunakan sehari sebelum eksperimen dijalankan dan mereka juga perlu bersedia pada waktu eksperimen dijalankan.

Pelajar-pelajar diingatkan sekali lagi setengah jam sebelum eksperimen bermula. Mereka diminta untuk menyemak komputer masing-masing dan bahasa pengaturcaraan Java yang telah siap dipasang. Selain dibekalkan dengan disket, pelajar juga diwajibkan mencipta folder bernama TW3053_<no.matrik> supaya penyimpanan fail lebih tersusun dan memudahkan proses pengumpulan data. Buku pengaturcaraan Java dan Ada dibenarkan untuk dibawa ke dalam makmal. No PC yang digunakan oleh setiap pelajar direkodkan.

Setelah soalan dan kod aturcara Ada diedarkan, masa mula eksperimen dicatatkan. Rajah 2 menunjukkan kod aturcara Ada yang diberikan kepada pelajar.

```

with text_io;
use text_io;

procedure count_chars is

    ch: character;
    counter: integer;

    package int_io is new integer_io(integer);
    use int_io;

begin
    counter := 0;
    ch := ' ';
    while ch /= '.' loop
        get(ch) ;
        counter := counter + 1;
    end loop;

    put("Bilangan aksara ialah ");
    put(counter);
    new_line;
end count_chars;

```

Rajah 2: Kod Aturcara Bahasa Ada

Secara ringkasnya, bahasa Ada yang mempunyai 18 baris kod aturcara ini akan mengira dan memaparkan bilangan aksara yang dimasukkan oleh pengguna. Pelajar dikehendaki membangunkan semula kod berkenaan menggunakan bahasa Java dan memastikan output yang sama dihasilkan. Pelajar tidak dibenarkan menambah fungsi baru dan mereka dikehendaki memaklumkan kepada pengkaji jika aturcara mereka telah berjaya. Pengkaji atau pembantu akan menguji aturcara tersebut dan sekiranya ia memenuhi keperluan seperti aturcara Ada yang diberikan, masa tamat pelajar akan direkodkan. Walau bagaimanapun, masa yang diambil oleh setiap pelajar hanya akan dikira setelah ditolak masa gangguan yang pelajar hadapi ketika eksperimen dijalankan seperti kerosakan komputer, perisian Java tidak berfungsi dan sebagainya. Sebagai langkah kawalan, kamera video juga digunakan untuk memastikan tiada peniruan dikalangan pelajar.

5. KEPUTUSAN DAN PERBINCANGAN

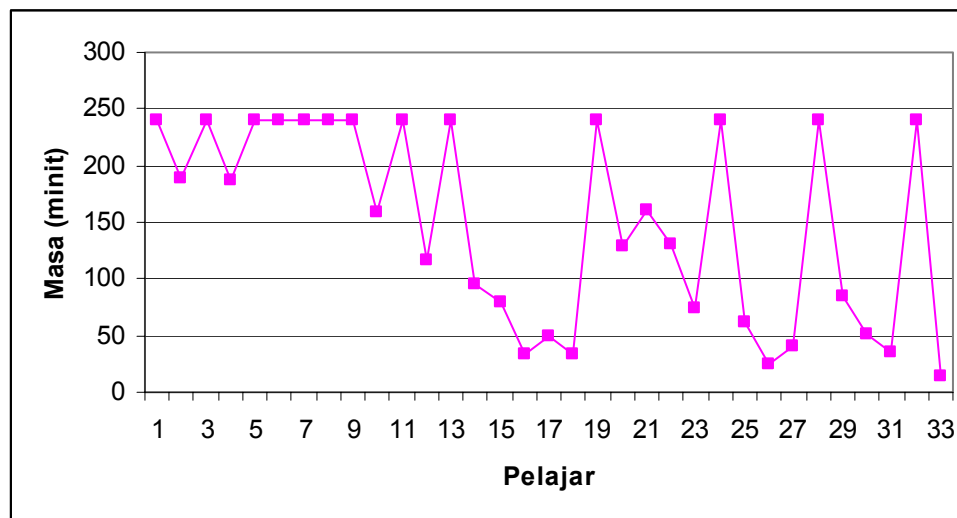
Setelah eksperimen dijalankan selama empat jam, didapati hanya 20 (60.6%) orang pelajar sahaja yang dapat menghasilkan keputusan yang sama dengan aturcara Ada yang telah diberikan manakala 13 (39.4%) orang pelajar lagi gagal menghabiskan aturcara tersebut. Jadual 2 menunjukkan bilangan pelajar yang berjaya dan pelajar yang gagal.

Jadual 2: Pelajar Berjaya dan Gagal

Jantina	Status		Jumlah
	Berjaya	Gagal	
Lelaki	9	7	16
Perempuan	11	6	17
Jumlah	20	13	33

Daripada jadual 2 di atas, didapati daripada 13 orang pelajar yang gagal, tujuh orang adalah pelajar lelaki manakala enam orang adalah perempuan.

Rajah 3 pula menunjukkan masa dalam kiraan minit yang diambil oleh setiap pelajar (data pelajar disusun mengikut nombor matrik) ketika membangunkan semula aturcara berkenaan. Walau bagaimanapun, 240 minit menunjukkan masa pelajar yang gagal menyiapkan aturcara tersebut. Ini adalah kerana pengkaji mendapati masa yang diambil oleh pelajar-pelajar berkenaan terlalu lama iaitu melebihi 50 minit berbanding dengan pelajar ke-20 yang berjaya. Jika dibandingkan pula dengan 18 baris kod aturcara Ada yang diberikan, secara puratanya pelajar-pelajar yang gagal hanya menulis satu baris kod aturcara dalam masa 13 minit.



Rajah 3: Masa yang diambil oleh setiap pelajar

Disebabkan 13 pelajar gagal membangunkan semula aturcara berkenaan, pengkaji akan menganalisis kemahiran pengaturcaraan pelajar berdasarkan masa yang dicatatkan oleh 20 orang pelajar yang berjaya sahaja. Didapati min masa bagi 20 orang pelajar yang berjaya adalah 87.40 minit. Jika dibandingkan pula dengan 18 baris kod aturcara Ada yang diberikan, ini bermakna secara puratanya pelajar yang diuji ini memperuntukan hampir lima minit untuk membaca kod aturcara Ada dan menulis semula menggunakan bahasa Java. Dan jika dibandingkan min masa yang dicatatkan, didapati secara puratanya pelajar lelaki lebih pantas menulis semula aturcara berkenaan berbanding dengan pelajar perempuan. Analisis menunjukkan 75.44 minit adalah min masa dicatatkan oleh pelajar

lelaki manakala pelajar perempuan hanya mencatatkan 97.18 minit bagi min masa. Masa yang paling pantas juga dicatatkan oleh pelajar lelaki iaitu 14 minit. Jadual 3 menunjukkan perbandingan masa bagi pelajar lelaki dan pelajar perempuan dari segi min, median, masa minimum dan masa maksimum yang dicatatkan.

Jadual 3: Perbandingan Masa bagi Pelajar Lelaki dan Pelajar Perempuan

Jantina	Bilangan	Min	Median	Minimum	Maksimum
Lelaki	9	75.44	62.00	14	188
Perempuan	11	97.18	96.00	24	187
Jumlah	20	87.40	77.00	14	188

Untuk menguji perkaitan antara gred yang diperolehi dan masa yang dicatatkan, pengkaji juga menganalisis 20 orang pelajar yang berjaya sahaja. Hasil analisa yang dibuat, didapati enam pelajar yang berjaya dengan masa yang pantas mempunyai keputusan 'A' dalam kursus TA1023 Pengaturcaraan Lanjutan dan tiada pelajar yang mendapat gred 'A', gagal dalam ujian tersebut. Jadual 4 menunjukkan kepantasan menghabiskan aturcara dengan jayanya mengikut gred kursus TA1023 Pengaturcaraan Lanjutan.

Jadual 4: Masa dan Gred Pengaturcaraan Lanjutan

Masa (minit)	Gred Pengaturcaraan Lanjutan				Jumlah
	A	A-	B+	B	
14 - 48	6	0	0	0	6
49 - 83	1	1	2	1	5
84 - 118	0	0	2	1	3
119 - 153	1	0	0	1	2
154 - 188	0	2	2	0	4
Jumlah	8	3	6	3	20

Jadual 5 menunjukkan ujian Korelasi Spearman yang dilakukan bagi melihat perkaitan antara gred pengaturcaraan lanjutan dan masa yang dicatatkan oleh pelajar untuk membangunkan semula aturcara tersebut. Kajian mendapati wujud perkaitan yang signifikan antara gred kursus TA1023 Pengaturcaraan Lanjutan dan masa yang dicatatkan oleh pelajar ($r = 0.563$, $p < 0.01$). Ia menunjukkan bahawa kebanyakan pelajar yang mempunyai gred yang baik mampu membangunkan semula aturcara dengan pantas.

Jadual 5: Perkaitan antara Gred Pengaturcaraan Lanjutan dan Masa yang dicatatkan

	Masa	
	r	p
Gred Pengaturcaraan Lanjutan	0.563	0.010

6. KESIMPULAN DAN CADANGAN

Daripada keputusan di atas, dapatlah disimpulkan bahawa kejuruteraan semula bukanlah tugas yang mudah. Selain daripada perlu membangunkan aturcara yang baru menggunakan bahasa Java, pelajar juga perlu memahami kod aturcara Ada yang sedia ada. Pengaturcaraan memerlukan kemahiran seperti belajar membaca dan menulis (Prasad & Fielden, 2002). Ia memberi kelebihan jika pelajar berkenaan dapat menguasai kedua-duanya dengan baik.

Keputusan eksperimen juga menunjukkan kebanyakan pelajar yang mempunyai keputusan yang baik dalam Pengaturcaraan Lanjutan mampu menulis semula aturcara dengan pantas. Menurut Prasad dan Fielden (2002), orang yang mempunyai pemikiran logik dan analitikal yang baik, mempunyai kelebihan untuk membangunkan aturcara dengan jayanya.

Pengkaji juga berpendapat faktor persekitaran juga perlu diambil kira ketika melakukan eksperimen dalam makmal. Maklumbalas daripada pelajar menyatakan bahawa gangguan bunyi telefon bimbit, pendingin udara yang tidak sejuk, kehadiran pembantu yang ramai dan penggunaan kamera video menyebabkan pelajar berasa tidak selesa untuk menulis aturcara tersebut.

7. RUJUKAN

- Anewalt, K., Polack-Wahl, J. A., Beidler, J., & Smarkusky, D. L. (2003). Group Projects Across The Curriculum. *Journal of Computing Sciences in Colleges*, 19(2), 232-237.
- Byrne, E. J., & Gustafson, D. A. (1992). *A Software Re-Engineering Process Model*. Paper presented at the Computer Software and Applications Conference, Chicago.
- Dunsmore, H. E., Moffett, D. P., & Ward, S. T. (1989). Software Engineering Team Project Group Member Evaluations: Some Empirical Results. *ACM SIGCSE Bulletin*, 21(2), 40-45.
- Lethbridge, T. C., & Laganriere, R. (2001). *Object Oriented Software Engineering: Practical Software Development Using UML and Java*. London: McGraw-Hill.
- Prasad, C., & Fielden, K. (2002). *Introducing Programming: A Balanced Approach*. Paper presented at the 15th Annual NACCQ, Hamilton, New Zealand.
- Pressman, R. (2004). *Software Engineering: A Practitioner's Approach* (6 ed.). Boston: McGraw-Hill.
- Rosenberg, L. H. (1996). *Software Re-engineering*. Retrieved 5 July, 2005, from [http://cisas.unipd.it/STS_course/Software_development/SW_re-engineering\(NASA%20doc\).pdf](http://cisas.unipd.it/STS_course/Software_development/SW_re-engineering(NASA%20doc).pdf)
- Sommerville, I. (2004). *Software Engineering* (7 ed.). London: Addison-Wesley.