

Ringkasan Paper #10

Kelompok 313

Nama Peringkas : Ginanjar CK
NPM : 120500038X
Judul Artikel : *DO SEQUENCE AND CONCURRENCY MATTER?: An Investigation of Order and Timing Effects on Student Learning of Programming Languages*
Penulis : Andrew Urbaczewski, Bradley C. Wheeler
Publikasi : CAIS – Volume 5 – 2001
Jumlah Halaman : 22
Jenis Artikel : Penelitian
Metodologi : Survey

PENDAHULUAN

Kurikulum untuk bidang Sistem Informasi (SI) program S1 seringkali berubah mengikuti perkembangan teknologi. Para pengajar SI berusaha menyusun kurikulum untuk program S1 SI yang dapat membuat para mahasiswa mendapatkan pengajaran yang sesuai dengan kebutuhan di dunia kerja nantinya. Salah satu mata kuliah dasar yang penting di SI adalah Pemrograman. Bahasa pemrograman merupakan bagian dasar dalam kurikulum. Sekarang ini, kebanyakan perusahaan lebih suka merekrut sumber daya SI yang memiliki keahlian pada lebih dari satu bahasa pemrograman.

Dalam mengembangkan keahlian pemrograman, ada dua isu utama yaitu *sequence* (berurutan) dan *concurrency* (bersamaan). *Sequence* artinya mempelajari beberapa bahasa pemrograman secara bertahap. Misalkan pertama mempelajari bahasa pemrograman berkonsep prosedural, kemudian *object-oriented programming*, dst. Adapun *concurrency* artinya mempelajari dua atau lebih bahasa pemrograman secara simultan atau bersamaan. *Paper* ini melakukan investigasi apakah *sequence* dan *concurrency* berpengaruh dalam proses pembelajaran bahasa pemrograman pada mahasiswa.

STUDI PENELITIAN

Pada *paper* ini terdapat dua pertanyaan penelitian yang utama, yaitu:

1. Apakah mahasiswa belajar bahasa pemrograman lebih baik ketika mereka diajari dengan urutan per bagian tertentu, seperti 4GL (*fourth generation language*) kemudian 3GL (*third generation language*), atau sebaliknya?
2. Apakah mahasiswa belajar bahasa pemrograman lebih baik ketika mereka mengambilnya secara *sequence* daripada secara simultan (*concurrency*)?

EMPAT PERSPEKTIF SEQUENCE

Dalam menerapkan kurikulum pembelajaran secara *sequence*, penulis menawarkan empat alternatif perspektif *sequence*:

1. **Evolutionary sequence**
Bahasa pemrograman muncul dalam beberapa generasi. Urutan pengambilan/pembelajaran bahasa pemrograman berdasarkan pada kemunculan generasi bahasa pemrograman.
2. **Difficulty sequence**
Urutan pengambilan/pembelajaran bahasa pemrograman berdasarkan pada tingkat kesulitan bahasa tersebut. Di awal, mahasiswa sebaiknya mengambil bahasa pemrograman yang lebih mudah bagi mereka untuk mengenal dunia pemrograman.
3. **As needed sequence.**
Dalam dunia bisnis atau kerja, ada bahasa pemrograman tertentu yang sering digunakan atau dibutuhkan. Urutan pengambilan/pembelajaran bahasa pemrograman berdasarkan pada kebutuhan akan bahasa tersebut.
4. **Filtering sequence.**
Perspektif ini menawarkan mahasiswa untuk mempelajari bahasa pemrograman yang sulit terlebih dahulu, kemudian dilanjutkan dengan bahasa yang lebih mudah.

Tabel 1. Empat perspektif *sequence* (disadur dari *paper* asli)

Perspective	Sequence
Evolutionary	COBOL, C → C++, VB → PowerBuilder
Difficulty	PowerBuilder, VB → COBOL → C → C++
As Needed	VB, PowerBuilder → C++, C → COBOL
Filtering	C → C++ → COBOL → VB

HIPOTESA

Para pengajar bahasa pemrograman seringkali mendengar tanggapan mahasiswa berkaitan dengan kesulitan dalam mempelajari bahasa pemrograman. Tanggapan tersebut berasal dari mahasiswa yang belum memiliki pengetahuan tentang pemrograman sebelumnya maupun yang sudah pernah berpengalaman dengan satu bahasa pemrograman dan memiliki kesulitan dengan bahasa pemrograman lainnya. Berikut adalah hipotesa yang dibangun berdasarkan empat perspektif *sequence* dan *concurrency*. Hipotesa ini mengarah kepada nilai akhir yang didapat oleh mahasiswa.

- H1: *Students who took any prior programming course demonstrate greater performance with the language than those who have not taken any prior programming course (Sequence).*
- H2: *Students who are taking only one programming course demonstrate higher performance with the language than those who are taking multiple, concurrent programming courses (Concurrency).*

Selain itu, mahasiswa juga memiliki tingkat kenyamanan dan kepercayaan diri dengan bahasa pemrograman tertentu. Berikut adalah hipotesanya.

- H3: *Students who took any prior programming language course report greater comfort with the language than those who did not take a prior programming course (Sequence).*
- H4: *Students who take only one programming course will report greater comfort with the language than those who are taking multiple, concurrent programming courses (Concurrency).*

METODOLOGI

Penelitian ini melakukan investigasi terhadap mata kuliah pengenalan pemrograman dan 341 mahasiswa selama tiga semester pada *Midwestern University*. Adapun mata kuliah pengenalan pemrograman yang digunakan adalah:

1. *Introduction to Visual Programming*
2. *Introduction to COBOL Programming*, dan
3. *Introduction to C Programming*

Dari 341 mahasiswa yang merespon survey, ada yang mengambil satu, dua, atau tiga mata kuliah pemrograman tergantung jadwalnya.

Variabel dalam penelitian ini antara lain:

1. *Independent variable: prior programming course* dan *concurrent programming course*.
2. *Dependent variable: self-reported course grade* (nilai akhir) dan *perceived comfort level* (tingkat kenyamanan).
3. *Demographic data/variable: kehadiran di kelas, nilai akhir, pengalaman*.

HASIL PENELITIAN

Hasil yang didapat dari penelitian ini menunjukkan bahwa H1 dan H2 tidak terbukti. Adanya pengalaman mengambil mata kuliah pemrograman maupun pengambilan mata kuliah secara simultan tidak memberikan kontribusi dampak yang besar terhadap performa mahasiswa di mata kuliah pemrograman yang diukur dengan *self-reported course grade* (nilai akhir).

Adapun pada H3 dan H4, mahasiswa yang mengambil mata kuliah pemrograman sebelumnya menunjukkan *perceived comfort level* lebih tinggi dengan bahasa pemrograman Visual Basic daripada mahasiswa yang tidak mengambil kuliah sebelumnya. Adapun mahasiswa yang mengambil kuliah COBOL atau C bersamaan dengan Visual Basic menunjukkan *perceived comfort level* yang lebih tinggi daripada mahasiswa yang tidak mengambil

mata kuliah pemrograman lain secara bersamaan. Dari penelitian ini ditemukan bahwa Visual Basic cukup baik untuk dipelajari secara bersamaan dengan bahasa pemrograman lain atau setelahnya. Namun tidak demikian dengan bahasa pemrograman COBOL atau C.

Berikut adalah table deskriptif statistik dari ketiga bahasa pemrograman tersebut.

Tabel 2. Statistik untuk mata kuliah Visual Basic (disadur dari *paper* asli)

	No Prior Course	Had a Prior Course
<u>Course Grade</u>		
No Concurrent Course	n=99 3.0434 (.6974)	n=84 3.3107 (.5720)
With a Concurrent Course	n=42 3.2524 (.5071)	n=8 3.000 (.5345)
<u>Perceived Comfort</u>		
No Concurrent Course	n=102 4.5588 (1.1988)	n=85 5.1765 (.9534)
With a Concurrent Course	n=46 5.1304 (1.1471)	n=8 5.6250 (.9161)

Tabel 3. Statistik untuk mata kuliah COBOL (disadur dari *paper* asli)

<i>n, mean, (st. dev)</i>	No Prior Course	Had a Prior Course
<u>Course Grade</u>		
No Concurrent Course	n=31 3.3516 (.6501)	n=75 3.2760 (.6726)
With a Concurrent Course	n=12 2.9167 (1.1946)	n=5 3.000 (.7071)
<u>Perceived Comfort</u>		
No Concurrent Course	n=33 4.6970 (1.4892)	n=76 5.0921 (1.0976)
With a Concurrent Course	n=12 4.9167 (.7930)	n=6 5.3333 (.5164)

Tabel 4. Statistik untuk mata kuliah C (disadur dari *paper* asli)

<i>n, mean, (st. dev)</i>	No Prior Course	Had a Prior Course
Course Grade		
No Concurrent Course	n=140 3.43.14 (.6049)	n=62 3.4710 (.5756)
With a Concurrent Course	n=42 3.4095 (.6559)	n=4 3.3500 (.4041)
Perceived Comfort		
No Concurrent Course	n=144 4.9792 (1.0740)	n=62 5.0645 (1.0381)
With a Concurrent Course	n=53 5.1132 (1.3820)	n=6 5.8333 (.9832)

KESIMPULAN

Pertanyaan penelitian ini adalah apakah *sequence* dan *concurrency* dari mata kuliah pemrograman berpengaruh dalam pembelajaran bahasa pemrograman, dan jawaban dari penelitian ini adalah tidak. Pengajar tidak perlu terlalu memperhatikan urutan dalam kurikulum mata kuliah pemrograman. Mahasiswa menunjukkan bahwa mereka lebih nyaman belajar Visual Basic setelah belajar bahasa pemrograman lain atau bersamaan dengan bahasa lain.

KOMENTAR PERINGKAS

Berikut ini adalah komentar pribadi dari peringkasan yang subjektif:

1. Secara bahasa dan topik, *paper* ini cukup mudah dipahami.
2. Pada bagian metodologi, penulis tidak memberikan diagram atau gambar tentang proses penelitian dan model yang digunakan dalam penelitian sehingga peringkasan agak kesulitan dalam memahami bagaimana penelitian dilakukan. Padahal terdapat variabel-variabel dalam penelitian ini yang akan lebih mudah jika digambarkan daripada dijelaskan dengan kata-kata.
3. Mata kuliah pemrograman yang digunakan dalam penelitian ini hanya tiga.
4. Penelitian ini tidak memasukkan aspek pengajaran oleh pengajar sebagai salah satu variabel yang bisa jadi menentukan proses belajar mahasiswa.
5. Awalnya peringkasan berpikir *paper* ini akan sangat bermanfaat baik bagi mahasiswa dalam menentukan gaya belajarnya maupun bagi pengajar atau dosen dalam menyusun kurikulum di kampus. Namun, setelah mengetahui hasil penelitian ini, saya beranggapan bahwa *paper* ini tidak akan terlalu bermanfaat bagi mahasiswa maupun pengajar/dosen.

REFERENSI UTAMA

- [1] Campbell, D.T., and Stanley, J.C. (1963) *Experimental and Quasi-Experimental Designs for Research*. Boston: Houghton Mifflin Company
- [2] Couger, J.D., et al (1995) "IS '95: Guideline for Undergraduate IS Curriculum," *MIS Quarterly*, (19)3, pp. 341-359.