

VISUALISASI PENGARUH ELEMEN PERANCANGAN *CACHE* PADA *SYMMETRIC MULTIPROCESSORS*

Tati Erlina

Jurusan Sistem Komputer Fakultas Teknologi Informasi
Universitas Andalas
Kampus Limau Manis Unand, PADANG
tatierlina2014@gmail.com

Abstrak

Cache memory merupakan salah satu pokok pembahasan penting dalam matakuliah organisasi dan arsitektur komputer. Akan tetapi, *cache* tidak dapat diakses dalam proses pembelajaran. Oleh karena itu, mahasiswa menghadapi kesulitan dalam memahami topik ini secara komprehensif. Makalah ini membahas tentang penggunaan simulator *SMPCache* dalam pembelajaran *cache memory*. Beberapa contoh digunakan untuk mendemostrasikan bagaimana simulator *SMPCache* mendukung berbagai latihan terkait dengan elemen-elemen yang menjadi pertimbangan dalam perancangan *cache* dan berpengaruh terhadap kinerja keseluruhan sebuah *cache memory*. Penilaian pedagogi dilakukan terhadap visualisasi yang disediakan oleh simulator *SMPCache* dalam mendukung pembelajaran *cache memory* pada *symmetric multiprocessing*. Selain itu, untuk mengetahui respon mahasiswa terhadap peranan *SMPCache* dalam membantu proses pembelajaran, evaluasi berupa survey pendahuluan juga dilakukan terhadap simulator tersebut.

Kata kunci:

Cache, *SMPCache*, Simulator, Organisasi dan Arsitektur Komputer

Abstract

Cache memory is one of the important subject matter in the course of computer organization and architecture. However, the cache cannot be accessed in the learning process. Therefore, students of have difficulties in understanding this topic comprehensively. This paper focuses on the use of *SMPCache* simulator in learning *cache memory*. Some examples are used to demonstrate how *SMPCache* simulator supports various exercises associated with elements of cache design which affect

the overall performance of a cache memory. Pedagogy assessment is conducted on the visualization provided by the simulators *SMPCache* in supporting the learning cache memory on a *symmetric multiprocessing*. In addition, to evaluate the response of students to *SMPCache* role in helping the learning process, an evaluation form preliminary surveys were also conducted on the simulator.

Keywords:

Cache, *SMPCache*, Simulator, Computer Organization and Architecture

I. PENDAHULUAN

Organisasi dan Arsitektur Komputer merupakan salah satu bidang ilmu utama pembangun *body of knowledge* dalam teknik komputer/sistem komputer (ACM & IEEE, 2015). Hal ini terlihat dari besarnya proporsi yang dialokasikan untuk bidang ilmu ini yaitu 60 core hours dari 420 core hours yang tersedia. Matakuliah ini bertujuan untuk memberikan pemahaman pada mahasiswa tentang bagaimana cara komputer bekerja. Di Perguruan Tinggi di Indonesia, biasanya bidang ilmu ini ditawarkan dalam 6 sks dimana 3 SKS pertama dialokasikan untuk membahas konsep-konsep dasar yang terkait dengan struktur komputer, seperti *processor*, memori, subsistem input/output dan bus. Sedangkan 3 SKS kedua mencakup pembahasan tentang organisasi dan arsitektur processor CISC dan RISC, organisasi *pipeline processor*, sistem penyimpanan, interkoneksi jaringan dan sistem memori. Pada satu sisi, dapat diketahui bahwa matakuliah ini melibatkan banyak konsep dasar yang bersifat abstrak, sehingga seringkali mahasiswa hanya mengandalkan imajinasinya untuk memahami konsep-konsep dasar yang terkait dengan *hardware* (Hatfield et al, 2005).

Sedangkan pada sisi lain, dilingkungan pembelajaran di perguruan tinggi, seringkali pengelola kesulitan untuk menyediakan sebuah laboratorium yang mendukung pembelajaran organisasi dan arsitektur komputer yang selalu diperbaharui. Oleh karena itu, salah satu usaha yang dilakukan untuk membantu mahasiswa dalam memahami konsep *cache* serta elemen-elemen yang mempengaruhi kinerja dari *cache* tersebut pada pembelajaran Organisasi dan Arsitektur Komputer adalah dengan menggunakan *software* simulasi yang dapat memberikan visualisasi dari pengaruh elemen-elemen tersebut terhadap kinerja sebuah *cache memory*.

II. KAJIAN LITERATUR

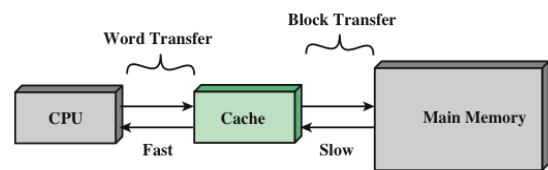
Sampai saat ini, banyak *software* yang sudah dikembangkan untuk membantu meningkatkan hasil pembelajaran pada matakuliah Organisasi dan Arsitektur Komputer. Diantara *software* tersebut, ada yang memasukkan *cache memory* sebagai salah satu elemen simulasi yang disediakan. Selain itu, beberapa *software* khusus dibuat sebagai simulator *cache memory* yang mencakup seluruh aspek yang menjadi pertimbangan dalam perancangan *cache memory*. Akan tetapi, perangkat-perangkat tersebut sangat berbeda (Bulić, Guštin, Šonc, & Štrancar, 2013) dalam hal jangkauan, kompleksitas arsitektur sasaran, level simulasi, *user interface*, kebutuhan *software* dan *hardware* serta pengetahuan yang dibutuhkan untuk dapat menggunakan simulator tersebut dengan baik.

EDUCache simulator (Ristov, Gusev, Atanasovski, & Anchev, 2013) merupakan sebuah simulator yang dapat mendukung proses pembelajaran *multilayer modern, multi-cache dan multi-core multi-processor* secara efektif dan efisien. EduMIPS64 (Fazzino, 2006) merupakan sebuah alat bantu pembelajaran tentang pipelining instruksi, *exception handling, interrupts* dan hirarki memori. Alat pembelajaran yang sangat berguna dan mensimulasikan arsitektur *pipeline processor MIPS64* secara menyeluruh. Akan tetapi, untuk dapat menggunakan simulator ini, mahasiswa sudah harus memahami MIPS64 ISA yang merupakan hal yang hampir mustahil bagi mahasiswa tingkat pertama dan kedua. DineroIV (Jan Edler, n.d.) adalah *cache simulator* yang mensimulasikan hierarki memori dengan beragam *cache*. Simulator ini merupakan perangkat yang akurat, akan tetapi sistem yang dapat disimulasikan terbatas pada sistem yang terdiri atas processor tunggal. Selain itu, DineroIV merupakan

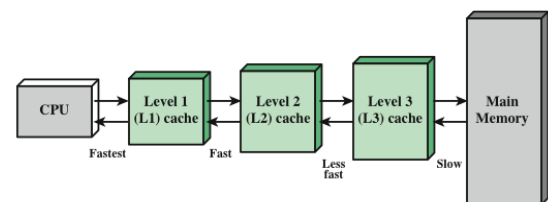
perangkat berbasis *command line* dan tidak menyediakan visualisasi.

II.1 Cache Memory dan Symmetric multiprocessors

Cache adalah sebuah memori kecil yang sangat cepat yang diposisikan diantara processor dan main memori. Tujuannya adalah agar main memori terkesan bekerja lebih cepat dari sudut pandang processor. Keefektifan pendekatan ini berdasarkan pada *property* program computer yang disebut *locality of reference* (Hamacher, Vranesic, Safwat, & Manjikian, 2012).



Gambar 1. Cache tunggal



Gambar 2. Organisasi cache tiga level

Cache memory berisi salinan sebagian isi dari memori (Patterson & Hennessy, 2014). Ketika *processor* berusaha untuk membaca sebuah *word* dari *memory*, maka terlebih dahulu dilakukan pencarian terhadap *word* tersebut di *cache*. Jika ada, maka *word* tersebut dikirim ke processor, jika tidak maka sebuah blok memori yang berisi *word* yang diminta processor tersebut dikirim ke processor, dan pada saat yang sama *word* tersebut juga disalin ke *cache*. Karena ada kecenderungan processor untuk mereferensi isi memori secara berurutan (*locality of reference*), maka kemungkinan besar *word* yang berikutnya diperlukan oleh processor sudah berada dalam *cache*, sehingga processor tidak perlu mencari *word* tersebut ke *main memory*. Dengan demikian, kinerja sistem secara keseluruhan dapat meningkat, karena processor tidak selalu harus meminta *word* yang dibutuhkannya ke *main memory* yang dalam hal kecepatan jauh lebih rendah dibanding dengan kecepatan *cache*.

Kinerja sebuah *cache* dipengaruhi oleh elemen-elemen perancangan *cache* yang terdiri atas beberapa elemen, yaitu alamat *cache* (logika dan fisik), ukuran *cache*, fungsi pemetaan (*direct*, *associative* dan *set associative*) ukuran baris, algoritma penimpaan (LRU, FIFO, LIFO, Random), aturan penulisan (*write through*, *write back*) dan jumlah *cache* (tunggal atau 2 tingkat, menyatu atau terpisah)(Stallings, 2013). Bagaimana elemen-elemen tersebut mempengaruhi kinerja sebuah *cache* adalah sebuah topik yang kompleks, apalagi *cache* dengan karakteristik tersebut tidak hanya terkait dengan processor tunggal, akan tetapi juga Symmetric multiprocessing (SMP)(Happe, Groenda, Hauck, & Reussner, 2010) yaitu processor yang terdiri atas beberapa inti processor yang memiliki properti yang sama, seperti kemampuan komputasi dan waktu yang dibutuhkan untuk mengakses memori.

II.2 SMPCache Simulator

SMPCache merupakan sebuah perangkat untuk menganalisa dan mensimulasikan sistem memori *cache* pada *symmetric mutiprocessors* (Rodriguez, Pérez, de la Montaña, & Gallardo, 2000). Simulator ini dibuat berdasarkan sebuah model yang sesuai dengan prinsip dasar arsitektur dari *symmetric mutiprocessor*, memiliki tampilan yang bersifat grafis, antarmuka yang ramah pengguna, serta dapat beroperasi pada sistem operasi Windows. Beberapa parameter pada *cache memory* dan multiprocessor yang dapat dipelajari menggunakan SMPCache adalah lokalitas program, pengaruh jumlah processor, *cache coherence protocol*, skema arbitrase bus, pemetaan, *replacement policies*, ukuran *cache* (blok dalam *cache*), jumlah set dalam *cache* (untuk *cache set associative*), dan jumlah *words* dalam blok (ukuran blok memori).

III. ANALISIS DAN PERANCANGAN

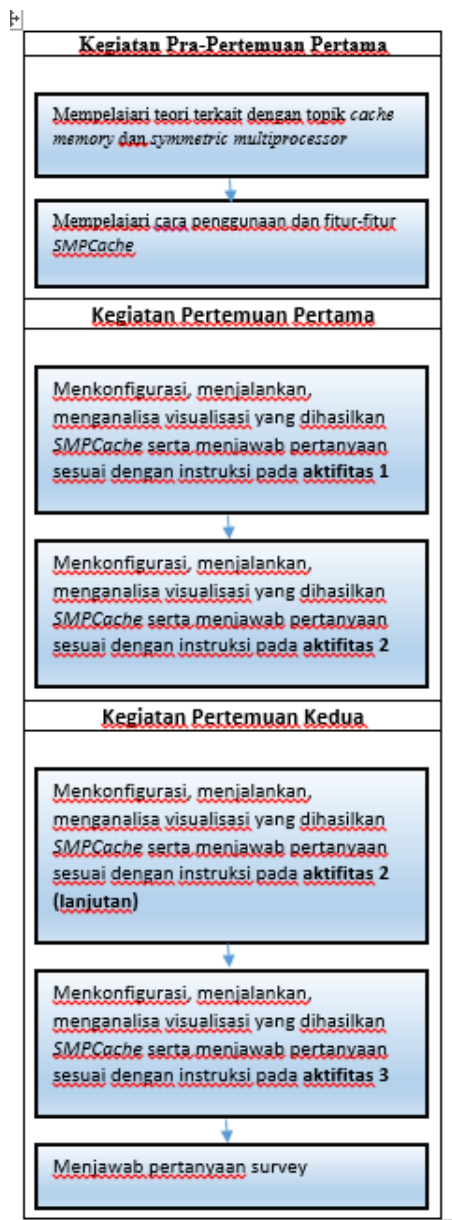
Secara umum, analisis dan perancangan dalam penelitian ini dilakukan melalui 2 (dua) langkah, yaitu: pemilihan strategi pembelajaran dan penilaian pedagogi.

III.1 Strategi Pembelajaran

Strategi pembelajaran pada topik ini adalah dengan melaksanakan ceramah di kelas yang berlangsung selama 2 SKS kemudian dilengkapi dengan 1 SKS sesi tutorial. Dimana tutorial ini

dilaksanakan pada hari yang sama dengan ceramah di kelas.

Agar kemampuan pedagogi melalui penggunaan SMPCache lebih terlihat, maka mahasiswa diminta untuk mengerjakan beberapa tugas selama sesi tutorial dimana tugas yang dikerjakan berdasarkan *student project* (Vega-Rodríguez, n.d.). Kegiatan ini dilakukan secara perorangan sebanyak 2 kali pertemuan yang berjarak 1 minggu dan masing-masingnya berdurasi 100 menit. Adapun rincian kegiatan yang dilakukan pada kedua pertemuan tersebut dapat dilihat pada Gambar 3.



Gambar 3. Langkah-Langkah Kegiatan

Persyaratan yang perlu dipenuhi oleh mahasiswa sebelum melaksanakan praktikum ini adalah mempelajari penggunaan SMPCache dan mempelajari teori yang terkait dengan sistem multiprocessor. Hal ini terkait dengan pelaksanaan praktek dimana SMPCache dikonfigurasi sedemikian rupa dengan menggunakan lebih dari satu processor. Sedangkan untuk referensi terhadap memori, digunakan 4 macam standar *multiprocessor traces* (FFT, Simple, Speech dan Weather).

Aktifitas 1: Mengevaluasi Kinerja Cache

Urutan kegiatan yang dilakukan mahasiswa pada aktifitas 1 adalah sebagai berikut :

1. Mengkonfigurasi sistem dengan menggunakan karakteristik tertentu yaitu jumlah processor 8, menggunakan MESI cache coherence protocol, skema arbitrase random, lebar word 32 bits, ukuran memori 1024 KB (jumlah blok dalam memori bervariasi), serta menggunakan aturan penimpaan LRU.
2. Menjalankan simulasi menggunakan semua kombinasi yang terdiri dari beberapa konfigurasi:
 - a. Mengkonfigurasi jumlah word menggunakan 10 macam konfigurasi yaitu: 2 (ukuran blok = 8 bytes) 4, 8, 16, 32, 64, 128, 256, 512 dan 1024 (ukuran blok = 4096).
 - b. Mengkonfigurasi jumlah blok dalam cache agar dapat memiliki ukuran sebagai berikut, yaitu 4 KB, 8 KB, 16 KB dan 32 KB
 - c. Mengkonfigurasi pemetaan menggunakan 5 macam konfigurasi yaitu, *direct*, *two way associative*, *four way associative*, *eight-way set associative* dan *fully associative*.
3. Menggunakan 10 konfigurasi yang sudah ditetapkan untuk mengevaluasi *replacement policy*. Pada masing-masing 10 konfigurasi terbaik tersebut, gunakan *replacement policy* menggunakan 4 konfigurasi: Random, LRU, LFU dan FIFO.
4. Membuat graf untuk mengilustrasikan perbandingan hit ratio pada semua konfigurasi yang sudah ditetapkan. Menganalisa dan menjelaskan hasil simulasi berdasarkan berdasarkan tabel dan graf yang dibuat oleh mahasiswa. Membuat analisa berdasarkan teori yang dipelajari di kelas. Menentukan konfigurasi yang terbaik dan berapa hit rasionya beserta alasannya, kemudian mengevaluasi dan mendiskusikan tingkat lokalitas dari *benchmark* yang digunakan.

Aktifitas 2: Pengaruh Cache Size terhadap Miss rate

Melalui aktifitas ini mahasiswa diharapkan mampu memahami pengaruh cache size terhadap miss rate selama eksekusi program parallel dalam sebuah SMP (*symmetric multiprocessor*). Urutan kegiatan yang dilakukan mahasiswa pada aktifitas 2 adalah sebagai berikut :

1. Mengkonfigurasi sebuah system dengan jumlah processor dalam SMP = 4, *cache coherence protocol*=MESI, skema arbitrase bus=LRU, lebar *word*=16, jumlah *word* dalam setiap blok=32 (ukuran blok=64 bytes), jumlah blok dalam main memori = 524288 (ukuran main memori=32 MB), *mapping* yang digunakan adalah *associative*. Sedangkan *cache sets* berbeda tergantung pada jumlah blok dalam *cache*, akan tetapi perlu dipastikan bahwa *cache* harus memiliki *four-way associative* serta menggunakan aturan penyimpanan LRU. Konfigurasi blok dalam *cache* adalah: 16 dengan ukuran *cache* 1 KB, 32, 64, 128, 256, 512, 1024 dan 2048 (ukuran *cache* = 128 KB). Untuk setiap konfigurasi tersebut perlu diketahui *miss rate* yang terjadi jika menggunakan trace files: FFT, Simple, Speech dan Weather.
2. Menjawab pertanyaan terkait dengan peningkatan atau penurunan *miss rate* ketika ukuran *cache* ditingkatkan beserta alasannya. Selain itu, mahasiswa juga diminta untuk membuat kesimpulan tentang hubungan antara ukuran *cache* dengan peningkatan kinerja sistem microprocessor dan seberapa besar perbedaan *miss rate* yang terjadi jika ukuran *cache* ditingkatkan.

Aktifitas 3: Pengaruh Fungsi Pemetaan terhadap berbagai ukuran *cache*

Kegiatan ini bertujuan untuk menganalisa pengaruh fungsi pemetaan terhadap *miss rate* pada *cache* dengan ukuran yang beragam. Urutan kegiatan yang dilakukan mahasiswa pada aktifitas 3 ini adalah sebagai berikut :

1. Menkonfigurasi sistem dengan jumlah processor yang digunakan dalam SMP adalah satu, *cache coherence* yang digunakan adalah MESI, skema arbitrase bus Random, lebar bus 32 bit, *words* dalam blok berjumlah 64 dimana setiap blok berukuran 256 bytes, dalam memori yang digunakan terdapat 4096 *blocks* serta menggunakan LRU *replacement policy*.
2. Mengkonfigurasi pemetaan dengan *direct*, *two way set associative*, *four way set associative*, *eight way associative* dan *fully associative*. Untuk setiap konfigurasi pemetaan tersebut, jumlah blok dalam *cache* dikonfigurasi agar dapat memperoleh ukuran *cache* berikut: 4 KB (16 blok dalam *cache*), 8 Kb, 16 KB, dan 32 KB (128 blok dalam *cache*). Untuk setiap konfigurasi, *miss rate*

diperoleh dengan menggunakan EAR *memory trace*.

3. Menjawab serangkaian pertanyaan: Apakah *miss rate* meningkat atau menurun sesuai dengan peningkatan asosiatifitas? Mengapa? Apa yang terjadi dengan *miss* tersebut ketika anda memperbesar tingkat asosiatifitas? Apakah pengaruh tingkat asosiatifitas meningkat atau menurun seiring dengan pertambahan ukuran *cache*? Mengapa?

III.2 Penilaian Pedagogi

Sebuah sumberdaya dalam proses pembelajaran tidak terlalu berarti jika tidak memenuhi kebutuhan pembelajaran yang diperlukan. Oleh karena itu, pengalaman belajar mahasiswa sangat dipengaruhi oleh peningkatan level keterlibatan dan kedalaman pembelajaran yang dapat diperolehnya melalui pengajaran dan lingkungan pembelajaran. Dalam penelitian ini, simulator SMP*Cache* yang digunakan dinilai dan dievaluasi dalam dua bentuk, yaitu: keterlibatan mahasiswa dan dukungan pembelajaran, serta evaluasi kualitatif.

Penilaian dilakukan dengan menggunakan dua metode, yaitu Taksonomi Keterlibatan (Naps, 2002) yang mendefinisikan enam kategori dalam keterlibatan mahasiswa dengan menggunakan teknologi visualisasi dan Taksonomi Bloom (Bloom, 1956) yang mengidentifikasi tingkatan kedalaman pemahaman mahasiswa. Untuk mengetahui sejauh mana peranan simulator SMP*Cache* efektif sebagai perangkat pembelajaran, maka perangkat ini dinilai menggunakan kedua metode tersebut diatas. Salah satu hal yang dilakukan adalah dengan memetakan kemampuan dan fungsionalitas simulator terhadap lima dari enam kategori yang terdapat pada Taksonomi Keterlibatan, dan juga memetakan variasi tugas-tugas yang didukung oleh Taksonomi Bloom. Penilaian ini diharapkan dapat memberikan tingkat kepercayaan terhadap penggunaan simulator tersebut, walaupun itu saja tidaklah cukup.

Tabel 1 memperlihatkan pemetaan contoh tugas yang dideskripsikan pada bagian sebelumnya ke lima kategori Taksonomi Keterlibatan. Sedangkan tabel 2 memperlihatkan pemetaan dari tugas yang sama ke lima level Taksonomi Bloom. Sebagaimana tergambar pada kedua tabel tersebut, simulator SMP*Cache* dapat mencakup semua tingkat keterlibatan dari Taksonomi Keterlibatan dan pada

saat yang sama juga memiliki kemampuan yang cukup untuk mendukung level pembelajaran yang berbeda dari Taksonomi Bloom.

Tabel 1. Pemetaan Terhadap Taksonomi Keterlibatan

Keterlibatan	Kategori	Aktifitas		
		1	2	3
	View	✓	✓	✓
	Respond	✓	✓	✓
	Change	✓	✓	✓
	Construct	✓	✓	✓
	Present	✓		✓

Tabel 2. Pemetaan Terhadap Taksonomi Bloom

Pembelajaran	Kategori	Aktifitas		
		1	2	3
	Knowlegde	✓	✓	✓
	Comprehension	✓	✓	✓
	Application	✓	✓	✓
	Analysis	✓	✓	✓
	Synthesis	✓	✓	✓
	Evaluation	✓	✓	✓

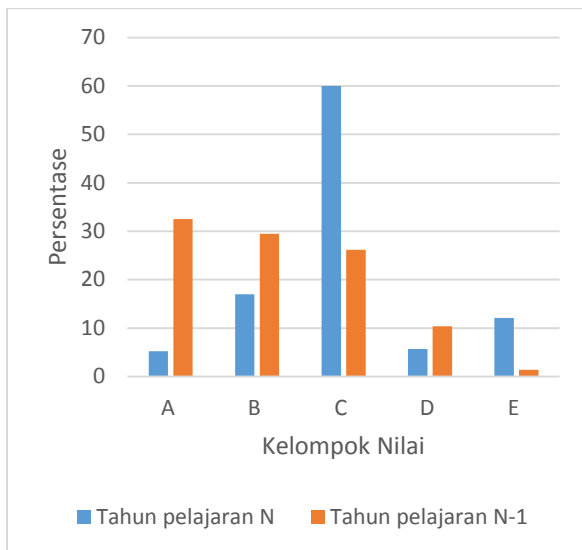
Untuk mengetahui efektifitas simulasi yang disediakan oleh simulator ini, sebuah evaluasi pendahuluan dilakukan dengan menggunakan survey kualitatif. Survey yang dilakukan menggunakan 5 nilai skala Likert. Evaluasi yang diadakan berdasarkan pada sejumlah sampel yang melibatkan mahasiswa tahun kedua sebanyak 45 orang, mahasiswa tahun 3 dan tahun 4 sebanyak 15 orang pada program studi sistem komputer. Tabel 3 memperlihatkan hasil dari survey tersebut. Kolom A dan kolom B merepresentasikan persentase gabungan dari mahasiswa tahun 2 yang menjawab sangat setuju dan setuju. Sedangkan kolom C dan D merepresentasikan nilai gabungan Sangat Setuju dan Setuju oleh mahasiswa tahun 3 dan tahun 4. Respon mereka secara umum adalah positif, hal ini menandakan bahwa mayoritas mahasiswa yang disurvei berpendapat bahwa simulator *SMPCache* bermanfaat dan mendukung pemahaman mahasiswa terhadap elemen-elemen perancangan *cache* serta setiap elemen tersebut terhadap kinerja *cache* secara umum, khususnya pada *symmetric multiprocessor*.

Tabel 3. Respon Mahasiswa Terhadap Penggunaan SMPCache

Pertanyaan Survey	A	B	C	D
Secara umum, simulator ini	93	0	98	0

membantu dalam memahami berapa konsep sulit terkait dengan <i>cache</i> memory				
Menurut saya simulator ini terlalu rumit untuk dimengerti dan digunakan secara efektif dalam tutorial yang berbasis simulator ini	13	27	10	8
Simulator ini lebih efektif membantu saya dalam memahami konsep sulit terkait dengan <i>cache</i> memory dibanding dengan membaca buku atau mencari lewat internet.	74	0	10	0
Saya menjadi lebih bersemangat dan lebih suka mengeksplorasi berbagai aspek dari <i>cache</i> memory dengan menggunakan simulator ini	83	7	90	0
Saya menghabiskan lebih banyak waktu untuk mempelajari bagaimana menggunakan simulator ini dibanding dengan pengerjaan latihan pada praktikum menggunakan simulator tersebut.	43	42	7	82
Simulator ini mendorong anggota grup saya untuk belajar bersama untuk menyelesaikan soal yang diberikan saat praktikum.	77	15	91	0
Menurut saya, latihan menggunakan simulator ini sangat mendukung dan tepat digunakan untuk memahami topik yang dibahas pada kuliah ini	93	0	10	0

Selain itu, keefektifan dari *SMPCache* dalam membantu mahasiswa dalam memahami topik *cache memory* beserta pengaruh elemen-elemen perancangan yang mempengaruhi performanya dapat dilihat dari gambar 4 yang memperlihatkan perbandingan nilai quiz yang diperoleh mahasiswa pada tahun pelajaran N (mahasiswa yang menggunakan *SMPCache* dalam proses pembelajaran) dan perolehan nilai mahasiswa pada tahun N-1 pada topik topik yang sama yaitu topik *cache memory*.



Gambar 4. Perbandingan Nilai Quiz topik *cache memory* Tahun Ajaran N dan Tahun Ajaran N-1

IV. KESIMPULAN DAN SARAN

Paper ini membahas peranan simulator bersifat visual yang berupa simulator dalam mendukung proses pembelajaran elemen-elemen perancangan *cache* pada *symmetric multiprocessor*. Pemetaan fungsionalitas simulator terhadap semua kategori dari taksonomi keterlibatan dan dukungannya terhadap proses pembelajaran level yang berbeda menunjukkan bahwa simulator ini dapat berperan sebagai sumberdaya pengajaran yang efektif. Pendapat positif yang diberikan mahasiswa yang tergambar pada hasil survey pada Tabel 3 memperkuat bahwa simulator tersebut memang efektif. Selain itu, peningkatan nilai quiz untuk topik *cache memory* dari Tahun Pelajaran N-1 ke Tahun Pelajaran N, terutama meningkatkan persentasi mahasiswa yang memperoleh nilai A dan B dari tahun pelajaran yang tersebut diatas.

REFERENSI

ACM, & IEEE. (2015). Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering, 142.

Bulić, P., Guštin, V., Šonc, D., & Štrancar, A. (2013). An FPGA-based integrated environment for computer architecture. *Computer Applications in Engineering Education*,

21(1), 26–35.
<http://doi.org/10.1002/cae.20448>

Fazzino, F. (2006). *EduMIPS64 : a Visual CPU Simulator for Teaching Computer Architecture*, 1–6.

Hamacher, C., Vranesic, Z., Safwat, Z., & Manjikian, N. (2012). *Computer Organization and Embedded Systems (Sixth Edit)*. New York: Mc Graw Hill.
<http://doi.org/10.1109/NOMS.2012.6211965>

Happe, J., Groenda, H., Hauck, M., & Reussner, R. H. (2010). A prediction model for *software* performance in symmetric multiprocessing environments. *Proceedings - 7th International Conference on the Quantitative Evaluation of Systems, QEST 2010*, 59–68.
<http://doi.org/10.1109/QEST.2010.15>

Hatfield, B., Rieker, M., & Jin, L. (2005). Incorporating simulation and implementation into teaching computer organization and architecture. *Proceedings - Frontiers in Education Conference, FIE, 2005*.

Jan Edler, M. D. H. (n.d.). *Dinero IV Trace-Driven Uniprocessor Cache Simulator*. Retrieved from <http://pages.cs.wisc.edu/~markhill/DineroIV/>

Patterson, D. A., & Hennessy, J. L. (2014). *In Praise of Computer Organization and Design : The Hardware / Software Interface , Fifth Edition (Fifth Edit)*. Waltham: Morgan Kaufmann.

Ristov, S., Gusev, M., Atanasovski, B., & Anchev, N. (2013). Using *EDUCache* Simulator for the Computer Architecture and Organization Course, 3(3), 47–57.

Rodríguez, M. A. V., Pérez, J. M. S., de la Montaña, R. M., & Gallardo, F. A. Z. (2000). Simulation of *Cache* Memory Systems on *Symmetric multiprocessors* with Educational Purposes. *Proceedings of the First International Congress in Quality and in Technical Education Innovation*, 3, 47–59. Retrieved from <http://arco.unex.es/smpcache/IICQTEI00.pdf>

Stallings, W. (2013). *Computer Organization and Architecture : Designing for Performance (Ninth Edit)*. Pearson.

-
- Vega-Rodríguez, M. A. (n.d.). Student Projects using *SMPCache* 2.0. Retrieved October 31, 2016, from /0017/171314/*SMPCache*-Projects.pdf
- Naps, T.L., et. al. 2002. Exploring the Role of Visualization and Engagement in ComputerScience Education. *ACM SIGCSE Bulletin* 35 (2), June 2003.
- Bloom, B. S., Krathwohl, D. R. 1956. Taxonomy of Educational Objectives; the Classification of Educational Goals, Handbook I: Cognitive Domain. Addison-Wesley