

Aplikasi Flashcard Berbasis Java Micro Edition

Edi Winarko, Prabowo Murti Saputro

Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Gadjah Mada
55281, Indonesia

ewinarko@ugm.ac.id murti@mail.ugm.ac.id

Abstrak— Saat ini jumlah aplikasi yang dapat dijalankan pada perangkat bergerak semakin banyak. Hal ini disebabkan oleh teknologi yang disebut JME atau Java Micro Edition yang dirilis oleh Sun Microsystems. JME memungkinkan pengembang aplikasi untuk membuat berbagai macam aplikasi *mobile* yang mendukung fitur *bluetooth* atau koneksi internet, pengaksesan terhadap *file system*, layanan perpesanan, pemrosesan gambar dari kamera, GPS, permainan, dan masih banyak lagi. Satu keuntungan besar dari penggunaan aplikasi *mobile* adalah dari sisi kepraktisan dan adalah mungkin untuk menjalankannya hampir di mana saja.

Sebuah aplikasi *flashcard* dapat membantu penggunaannya untuk mengingat materi pelajaran seperti matematika, rumus-rumus fisika, table periodik unsur (dalam bidang kimia), karakter dalam bahasa Jepang, atau kosakata dalam bahasa Inggris. Penelitian Piotr Wozniak membuktikan bahwa mempelajari materi tertentu dengan perulangan yang berjangka adalah cara paling efektif untuk belajar karena membuat penggunaannya cenderung fokus pada hal-hal yang sulit diingat dan tidak menghabiskan waktu pada hal-hal yang mudah dihafal. Wozniak lalu membuat sebuah aplikasi *flashcard* yang diberi nama Super Memo. Setelah itu Peter Bienstman memodifikasi algoritma yang dipakai pada Super Memo versi 2 dan membuat sebuah aplikasi *flashcard* baru bernama Mnemosyne yang ditulis dengan bahasa Python.

Hasil akhir dari penelitian ini adalah sebuah aplikasi *flashcard* menggunakan algoritma dari Bienstman dan berbasis JME. Lebih jauh, aplikasi ini dapat mempermudah pengguna aplikasi *flashcard* untuk belajar menggunakan perangkat bergerak yang mendukung teknologi MIDP 2.0 daripada menggunakan aplikasi *flashcard* versi *desktop*.

Kata kunci— aplikasi *flashcard*, perulangan berjangka, java micro edition

I. PENDAHULUAN

Perkembangan teknologi informasi dapat meningkatkan kinerja dan memungkinkan berbagai kegiatan dapat dilaksanakan dengan cepat, tepat dan akurat, sehingga akhirnya akan meningkatkan produktivitas [1]. Salah satu bentuk peningkatan produktivitas adalah dengan cara memanfaatkan aplikasi *flashcard* yang dapat membuat proses belajar menjadi lebih efektif. Sayangnya, aplikasi *flashcard* belum banyak dikenal dan dipakai di Indonesia. Padahal, berdasarkan data dari Badan Regulasi Telekomunikasi Indonesia (BRTI) tahun 2007, jumlah pengguna telepon genggam di Indonesia mencapai 80 juta orang, dan pasarnya terus bertambah dengan pesat [2]. Hal ini mengindikasikan bahwa telepon genggam bukan lagi termasuk barang mewah [3].

Untuk kepentingan aplikasi telepon seluler, Sun Microsystems mengeluarkan JME (Java Micro Edition) yang merupakan platform

untuk pengembangan aplikasi di perangkat bergerak dan berbagai perkakas rumah tangga seperti microwave, oven, kulkas, dsb. Penggunaan aplikasi pada telepon seluler membuat pengguna dapat melakukan proses belajar dengan aplikasi *flashcard* hampir di mana saja. Hal ini berbeda bila pengguna menggunakan aplikasi *flashcard* versi *desktop* dari segi kepraktisan.

II. LEITNER SYSTEM

Sebastian Leitner adalah seorang psikolog Eropa yang menemukan sistem kotak yang disebut *Leitner System* pada tahun 1960-an. Tujuannya adalah untuk mempermudah proses belajar dan mengingat kosakata memori jangka pendek maupun jangka panjang [4]. Leitner System menjadi penting karena merupakan cikal bakal dari banyak aplikasi *flashcard* [5]. Ide Leitner yang sederhana membuat efek yang sangat besar pada kinerja pembelajaran [6].

Leitner System terdiri dari sebuah kotak atau *box* yang dibagi menjadi beberapa bagian atau kompartemen. Kompartemen-kompartemen tersebut diisi dengan *flashcards* atau kartu yang disusun berdasar tingkat kesulitannya [6]. Sebuah *flash card* adalah sepotong kertas (kartu) yang memiliki pertanyaan di salah satu sisi dan jawaban di sisi sebaliknya [7]. *Flashcards* atau kumpulan kartu-kartu tersebut dapat berpindah kompartemen sesuai dengan tingkat pengetahuan kartu yang bersangkutan. Ketika sebuah *card* atau kartu dapat dijawab dengan baik, maka kartu tersebut dipromosikan ke kompartemen setelahnya. Sedangkan bila kartu tersebut gagal dijawab dengan baik maka kartu akan berpindah ke kompartemen paling depan [6]. Dengan demikian, cara *Leitner System* akan membuat penggunaannya menghafal kartu yang sulit diingat (kompartemen paling depan) lebih sering daripada kartu yang mudah diingat (kompartemen paling belakang) sampai semua kartu dapat dihafal dengan baik atau dengan kata lain kondisi semua kartu terletak di kompartemen paling akhir.

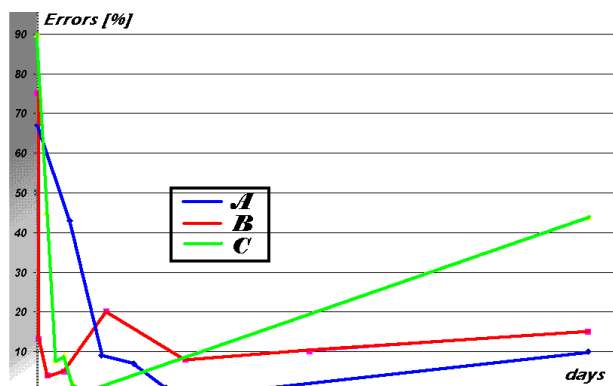
III. SPACED REPETITION

Pada tahun 1901 seorang psikolog Amerika bernama William James menyimpulkan bahwa review pada materi belajar seharusnya dilakukan dalam waktu yang berjangka. Kesimpulan yang sama juga dilakukan Hermann Ebbinghaus dan Jost. Pada tahun 1932, C.A. Mace memperkenalkan metode belajar yang efisien dalam bukunya yang berjudul "The Psychology of Study". Ia memperkenalkan istilah *active rehearsal* dan *repetitive revisions* yang menyebutkan bahwa proses belajar seharusnya berjangka menurut interval waktu yang meningkat, misalnya "interval 1 hari, 2 hari, 4 hari, dan seterusnya". Pendekatan ini diadopsi oleh beberapa penulis lain (termasuk Tony Buzan), namun tidak pernah diaplikasikan pada lingkungan studi dan penelitian. Barulah sekitar 50 tahun kemudian, ide ini diimplementasikan pada komputer sehingga menciptakan jalan yang lebih lebar bagi publik untuk melihat keuntungan *spaced repetition* [6].

Spaced repetition (perulangan berjangka) adalah teknik optimal dalam proses belajar dengan cara menghitung jangka waktu atau

interval yang memisahkan proses mengingat (mengambil informasi dari memori) dari pengetahuan tertentu. Program komputer yang menerapkan prinsip *spaced repetition* dinamakan aplikasi flashcard [6].

Ada beberapa metode yang digunakan dalam penjadwalan *item* yang akan dipelajari. Pada aplikasi SuperMemo, algoritma yang digunakan adalah algoritma SM (*SM-algorithm*). Sedangkan pada Trivial, sebuah aplikasi *flashcard* yang ditulis dengan bahasa C#, digunakan algoritma FSM atau *Finite State Machine* [7]. Perulangan yang berjangka secara meningkat (*increasing inter-repetition intervals*) tidak selamanya lebih efektif bila dibandingkan perulangan dengan jangka waktu antar review yang sama. Hal ini dibuktikan dengan percobaan yang dilakukan Wozniak sejak akhir Januari tahun 1985 hingga awal Agustus 1986 terhadap tiga macam kelompok pengetahuan yang diperlakukan berbeda. Kelompok A diulang dengan interval waktu yang sama (18 hari). Kelompok B diulang secara berjangka dan meningkat mulai 1 hari hingga 70 hari. Kelompok C diulang dengan jarak waktu yang sama, namun dengan masa yang relatif lebih singkat dari kelompok A, yakni setiap 5 hari. Hasilnya ditunjukkan oleh Gambar 1.



Gambar 1. Hasil percobaan pada 3 kelompok [2]

Gambar 1 menunjukkan bahwa hasil yang didapat dengan menggunakan perulangan dengan jangka waktu yang meningkat tidak lebih baik dari perulangan yang memakai jangka waktu yang tetap. Prinsip dalam menghitung interval waktu yang digunakan dalam proses belajar merujuk pada istilah *optimum repetition spacing principle* [8]. Permasalahannya adalah semakin lama jangka waktu untuk sebuah *item* agar dapat diulang membuat semakin sulitnya informasi bersangkutan diakses kembali [9].

Telah banyak penelitian yang menunjukkan bahwa variasi waktu pada perulangan berjangka berpengaruh pada kekuatan memori dan hasil penelitian dapat diaplikasikan para pembelajaran yang efektif. Dengan mengubah waktu interval dalam perulangan berjangka, efektifitas pembelajaran dapat diketahui. Kelemahan yang sering ditemukan pada penelitian tentang perulangan berjangka adalah kurangnya perhatian terhadap tingkat kesulitan *item* tertentu, sebagai contoh, jadwal perulangan yang sama digunakan terhadap *item* dengan tingkat kesulitan “sulit” dan tingkat kesulitan “mudah” [9].

IV. ALGORITMA SM

Tujuan eksperimen selanjutnya adalah membuktikan keberadaan nilai yang paling optimal untuk jarak waktu pada perulangan berjangka atau *optimum inter-repetition intervals* dan menghitung nilainya. Pada akhir eksperimen yang dilakukan Wozniak telah tercipta algoritma SM-0 dengan detail sebagai berikut [8]

1. Pilih informasi menjadi *item* terkecil yang mungkin.
2. Asosiasikan *item-item* menjadi grup yang terdiri dari 20-40

elemen.

3. Ulangi masing-masing grup menggunakan interval berikut

$$I(1) = 1$$

$$I(2) = 7$$

$$I(3) = 16$$

$$I(4) = 35$$

$$\text{Untuk } i > 4 : I(i) = I(i-1) * 2$$

Dimana $I(i)$ adalah interval waktu yang digunakan setelah perulangan ke- i .

Semua *item* yang terlupakan disalin setelah interval 35 hari ke grup baru (tanpa menghapusnya dari grup yang lama). Grup baru ini akan diulang dengan cara yang sama dengan *item* yang dipelajari pertama kali.

Setelah 2 tahun menggunakan algoritma SM-0, data yang cukup telah dikumpulkan untuk memastikan bahwa asumsi-asumsi yang dibuat pada algoritma SM-0 akurat dan beralasan [8].

V. MNEMOSYNE

Mnemosyne merupakan aplikasi *flashcard* lain yang menggunakan algoritma SM-2 dengan sedikit modifikasi. Aplikasi ini dibuat oleh Peter Bienstman, seseorang yang dulunya pernah bergabung dalam pembuatan aplikasi flashcard MemAid. Aplikasi MemAid sendiri bermula dari SuperMemo, dan kini menjadi aplikasi yang bersifat komersial dengan nama FullRecall. Walaupun berjenis aplikasi flashcard, Mnemosyne yang dikembangkan oleh Peter Bienstman memiliki karakter yang sedikit berbeda dari SuperMemo. Algoritma yang dipakai pada aplikasi Mnemosyne 1.0 adalah aplikasi SM-2 atau SM-algorithm versi 2 yang telah dimodifikasi. Selain itu, pada Mnemosyne dimungkinkan pengiriman log atau data mengenai card yang dipelajari oleh pengguna. Hal ini memungkinkan penelitian lebih lanjut mengenai penentuan optimum inter-repetition intervals yang lebih baik.

Masing-masing card dalam Mnemosyne memiliki 13 buah atribut [10]. Atribut-atribut tersebut adalah sebagai berikut

1. **ID**
Identifikasi unik dari tiap *card*. Dalam Mnemosyne, setiap *card* memiliki ID yang berupa *hash* dari data.
2. **Grade**
Grade adalah rentang integer dari 0 hingga 5. *Grade* 0 menandakan bahwa *card* belum dipelajari atau menggambarkan bahwa *card* sulit diingat. Pemberian *grade* 0 atau 1 pada *card* menandakan pengguna tidak mengetahui jawaban atau *answer* dari *card*, atau dengan kata lain telah melupakan *card*. Perbedaan *grade* 0 dan 1 hanya pada hal rentang waktu kapan *card* akan ditanyakan kembali. *Card* dengan *grade* 0 atau 1 akan terus ditanyakan hingga pengguna memberi *grade* 2 atau lebih pada *card*.
3. **Easiness**
Easiness menyatakan tingkat kesulitan dari *card*. Rentang nilainya dipengaruhi oleh *grade* baru yang diberikan oleh pengguna.
4. **Acquisition reps**
Istilah *acquisition phase* atau fase akuisisi digunakan untuk menggambarkan semua *card* yang memiliki *grade* 0 atau 1. *Acquisition reps* pada *card* menunjukkan seberapa sering proses pembelajaran *card* yang bersangkutan dengan hanya berkuat pada *grade* 0 atau 1.
5. **Retention reps**
Istilah *retention phase* atau fase retensi digunakan untuk menggambarkan semua *card* yang memiliki *grade* 2 sampai 5. Ini berarti bahwa pengguna dapat dengan baik mengingat *card*. *Retention reps* pada *card* menunjukkan seberapa sering proses pembelajaran *card* yang bersangkutan dengan diberi *grade* 2 hingga 5 oleh pengguna.
6. **Lapses**

Apabila sebuah *card* telah memasuki fase retensi, ada kemungkinan pengguna melupakan atau tidak dapat dengan baik mengingat *card*. Dengan demikian, *grade* dari *card* yang bersangkutan dapat berubah dari fase retensi ke fase akuisisi. Hal inilah yang disebut dengan istilah *lapse*. Nilai atribut *lapses* pada *card* akan meningkat bila hal ini terjadi.

7. *Acquisition reps since lapse*
Acquisition reps since lapse adalah rentang waktu *acquisition* sejak *lapse*.
8. *Retention reps since lapse*
Retention reps since lapse menyatakan rentang waktu *retention* sejak *lapse*.
9. *Last repetition*
Last repetition menggambarkan rentang waktu (dihitung sejak hari dimulainya proses belajar) *card* terakhir kali dimunculkan.
10. *Next repetition*
Next repetition menggambarkan rentang waktu *card* akan dimunculkan kembali.
11. *Question*
Berisi pertanyaan dari *card*.
12. *Answer*
Berisi jawaban dari *card*.
13. Kategori
Kategori *card* yang ditentukan sendiri oleh pengguna, misalnya: "katakana", "hiragana", atau "negara dan ibukotanya".

VI. HASIL PENELITIAN DAN PEMBAHASAN

Dari hasil penelitian yang telah dilakukan, didapat sebuah aplikasi atau produk berupa sebuah midlet (aplikasi MIDP) yang untuk selanjutnya diujicobakan pada beberapa merek dan tipe telepon seluler. Aplikasi hasil penelitian, yang diberi nama Syahfi, dibuat dengan berbagai macam tools seperti NetBeans sebagai

REFERENSI

- [1] Wardiana, W., 2002, "Perkembangan Teknologi Informasi di Indonesia". Pusat Penelitian Informatika, Lembaga Ilmu Pengetahuan Indonesia.
- [2] Antara, 2003, "Antara dan Ingo! Luncurkan Mobile News". Antara News.
- [3] Republika, 2003, "Menjual Daya Tarik Ponsel Secara Fisik". Republika Online.
- [4] MemoryLifter, <http://www.memorylifter.com/about-memorylifter.html>, diakses 28 Maret 2008
- [5] Wozniak, P.A., 2006, *History of SuperMemo*, <http://www.supermemo.com>, 9 Maret 2008.
- [6] InterFour, 2006, "Learning With Card Files", http://www.interfour.nl/downloads/Leitner_Cardsystem_imap.pdf, 3 September 2008.
- [7] Rochmatullah, A., 2005, <http://agro.web.ugm.ac.id/trivial.htm>, diakses tanggal 28 Maret 2008.
- [8] Wozniak, P.A., 1990, "Economics of Learning", Master's Thesis University of Technology in Poznan.
- [9] Wozniak, P.A., 1995, "Optimization of Learning", Doctoral Dissertation University of Economics, Wroclaw.
- [10] Bientsman, P., 2008, "Mnemosyne XML DTD", <http://mnemosyne-proj.org/help/mnemosyne-xml-dtd.php>, diakses 12 Februari 2009.