

Rancang Bangun Spaced Repetition Software untuk Menghafal Huruf Jepang Menggunakan Algoritma SuperMemo 2 Berbasis iOS

Agustyan Hidayat¹, Ni Made Satvika Iswari²

Fakultas Teknik dan Informatika, Universitas Multimedia Nusantara, Tangerang, Indonesia
agustyan@hidayat.web.id
satvika@umn.ac.id

Diterima 26 April 2018

Disetujui 8 Juni 2018

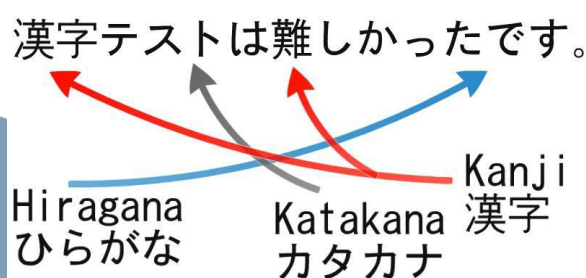
Abstract— During the study of another language such as Japanese, the problem that exists is in remembering each of the characters that is used as writing mechanism. This is happened because the difference with Roman characters and the number of character that exist, is far beyond that of alphabet which makes a visual impairment for those used to write in alphabet. There is a traditional way to remember something which is periodically learn or remember the characters and assign a visual image to each of the characters in order to be remembered easily. Spaced repetition is a method that give an interval during a repetition which is mainly used for someone who want to remember something periodically by learning it for a long term periodically. This research discusses the implementation of one of spaced repetition algorithm called SuperMemo 2 and using it along with a flashcard to learn Japanese characters periodically. Software development methodology used is Waterfall Model, because the requirements used to develop the software is clear at the beginning.

Index Terms—Flashcard, Japanese Characters, Language Learning, Spaced Repetition Software, SuperMemo

I. PENDAHULUAN

Dalam mempelajari sebuah bahasa asing seperti bahasa Jepang, kesulitan yang dihadapi adalah untuk menghafal karakter yang digunakan dalam penulisan. Bahasa Jepang sendiri memiliki 2 macam penulisan yang umum digunakan yakni Kana dan Kanji. Kana sendiri terbagi menjadi dua yaitu hiragana yang digunakan untuk kata-kata asli dan katakana yang digunakan untuk kata-kata serapan.

Kanji merupakan karakter Cina yang diserap dan digunakan bersamaan dengan Hiragana dan Katakana. Pada tahun 2010 jumlah kanji yang umum digunakan sudah mencapai sekitar 2.136 karakter. Dalam menghafalkan kanji terdapat berbagai macam teknik yang dapat dilakukan dan salah satunya adalah *flashcard*.



Gambar 1 Contoh Penulisan Bahasa Jepang

Faktor lain yang membuat kanji sulit untuk dihafal adalah sebuah kanji dapat memiliki lebih dari satu arti dan dalam melakukan penulisan kanji terdapat sebuah urutan yang baku. Seperti yang dijelaskan oleh Heisig dalam bukunya yang berjudul “Remembering the Kanji” pada tahun 2007, hal yang membuat kanji sulit untuk dipelajari adalah kurangnya kesamaan pola pada memori visual seseorang khususnya mereka yang terbiasa menulis menggunakan alphabet [5].

Cara tradisional yang digunakan untuk menghafal sebuah karakter adalah dengan mengulang mempelajari karakter tersebut secara berkala dan menggabungkan visual terhadap kanji tersebut agar dapat lebih mudah mengingat seperti menghubungkan sebuah kanji dengan objek-objek tertentu.

Spaced Repetition merupakan sebuah teknik yang digunakan untuk menghafal sesuatu dengan memberikan jeda yang berubah-ubah pada setiap pengulangan. Pendekatan ini menyerupai cara tradisional yang telah dibahas sebelumnya.

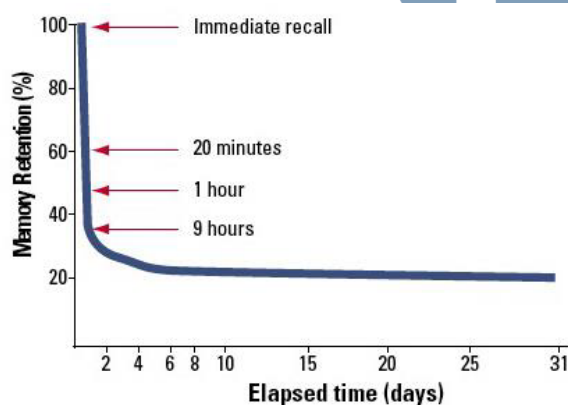
Sebelumnya pernah dilakukan penelitian oleh Jeff Hanks and Ping Zhan dengan judul “The Forgetting Curve and Learning Algorithms” yang membahas tentang algoritma SM-2 dan hasil penggunaannya [6]. Selain itu, Judy Yoneoka dalam paper yang berjudul “A Tailored Intensive Vocabulary Trainer Using an Online Flashcard Site” pada penerapannya digunakan metode Leitner [12]. Namun pada penelitian ini akan dilakukan dengan menggunakan algoritma

SuperMemo 2 dan dalam bentuk *mobile application*. Perbedaan Leitner dengan SuperMemo 2 terletak pada bagaimana penentuan jeda pada saat pengulangan (*repetition*), Metode Leitner hanya menggunakan variabel yang tetap untuk menentukan jeda sedangkan SuperMemo 2 melakukan perhitungan yang dibuat berdasarkan performa dari pengguna. Selain itu, SuperMemo 2 bersifat *Open Source* sehingga dapat digunakan dan dikembangkan oleh siapa saja.

II. LANDASAN TEORI

A. Spaced Repetition Software

Spaced Repetition Software merupakan sebuah aplikasi yang mengaplikasikan teknik *Spaced Repetition* yang merupakan sebuah teknik untuk menghafal yang dikembangkan pada tahun 1960. Teknik ini memberikan jeda waktu yang akan terus bertambah pada saat melakukan *review* pada sebuah informasi yang telah diketahui sebelumnya. Sebuah penelitian yang dilakukan oleh seorang psikolog Jerman, Hermann Ebbinghaus, menjelaskan bahwa ketika seseorang menerima sebuah informasi baru akan segera lupa dalam waktu yang sangat cepat jika tidak memperkuat informasi baru itu dalam ingatan mereka [2]. Berikut merupakan kurva hasil penelitian dari Hermann tentang jangka waktu yang dibutuhkan untuk melupakan sebuah informasi.



Gambar 2 Forgetting Curve

Dengan memberikan jeda waktu saat melakukan pengingatan terhadap sebuah informasi, maka akan terjadi sebuah *spacing effect* yang merupakan sebuah istilah psikologi dimana seseorang dapat lebih mudah mempelajari dan mengingat sebuah informasi.

B. Flashcard

Menurut kamus bahasa Merriam-Webster, Flashcard merupakan sebuah kartu yang memiliki kata atau gambar di salah satu atau kedua sisinya. *Flashcard* dapat digunakan untuk membantu seseorang dalam belajar, khususnya menghafal dengan mengasosiasikan sebuah gambar, kata maupun angka yang digunakan sebagai petunjuk [7].

C. Joyo Kanji (常用漢字)

Joyo Kanji yang dapat diartikan sebagai kanji yang digunakan secara umum atau kanji yang biasa dipakai. Joyo Kanji mulai digunakan pada tahun 1981 dan dijadikan standar berdasarkan Pengumuman Kabinet Jepang 1 Oktober. Jumlah kanji yang digunakan awalnya adalah 1945 [8]. Namun pada tahun 2010, total jumlah kanji yang terdapat pada Joyo Kanji ditambahkan hingga mencapai 2.136 karakter yang merupakan gabungan dari 1006 kanji yang diajarkan di sekolah dasar dan 1130 kanji yang diajarkan di sekolah menengah. Joyo Kanji sendiri banyak digunakan di masyarakat, surat kabar, majalah, media elektronik dan sebagainya.

D. Supermemo 2

SuperMemo 2 atau disebut juga SM-2 merupakan sebuah algoritma yang dibuat di Polandia oleh Piotr Wozniak [12]. Algoritma ini digunakan untuk memberikan sebuah interval pada sebuah *inter-repetition*.

Di dalam algoritma ini terdapat sebuah variabel yang disebut dengan E-Factor (*Easiness Factor*). E-Factor akan berperan memberikan sebuah nilai pada setiap *item* yang terdapat pada *list*. Nilai pada E-Factor memiliki variasi dari 1,1 (Tersulit) hingga 2,5 (Termudah). Untuk mendapatkan E-Factor, pada mulanya nilai E-Factor untuk semua data awal adalah 2,5, selanjutnya digunakan fungsi sebagai berikut [13]

$$EF' = EF + (0.1 - (5 - q) * (0.08 + (5 - q) * 0.02)) \dots \text{Rumus 1}$$

dengan q merupakan sebuah variabel yang melambangkan kualitas dari jawaban pada saat menjawab informasi dengan skala 0-5. E-Factor tidak berubah jika $q < 3$.

Selanjutnya perhitungan *interval* dilakukan dengan fungsi sebagai berikut dengan n adalah jumlah repetisi pembelajaran [13]

$$I(1) = 1, I(2) = 6, I(n) := I(n-1) * EF \dots \text{Rumus 2}$$

Nilai dari *interval* ini akan berubah-ubah seiring dengan repetisi yang dilakukan karena E-Factor akan berubah setiap kali repetisi dilakukan.

E. iOS

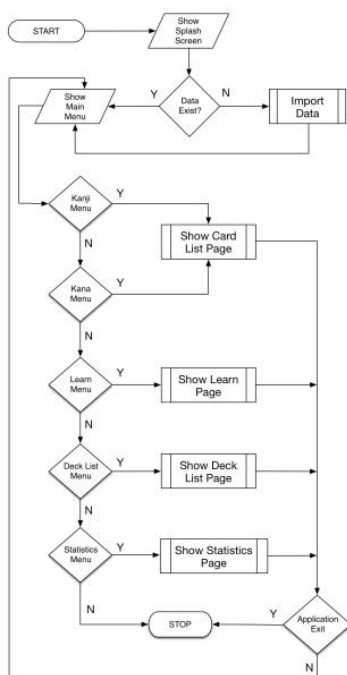
iOS adalah sistem operasi mobile yang dikembangkan oleh Apple. Pada awalnya bernama OS iPhone, namun kemudian diganti dengan iOS pada bulan Juni 2009. iOS saat ini berjalan pada iPhone, iPod touch, dan iPad. Seperti sistem operasi desktop modern, iOS menggunakan antarmuka pengguna grafis, atau GUI. Namun, karena merupakan sistem operasi mobile, iOS dirancang untuk menggunakan input *touchscreen*, bukan *keyboard* dan *mouse*.

III. PERANCANGAN SISTEM

Spaced Repetition Software adalah sebuah sistem yang memanfaatkan algoritma *spaced repetition* yang digunakan untuk memberikan jeda atau *interval* pada saat melakukan pembelajaran. Dalam aplikasi ini digunakan flashcard sebagai salah satu metode yang membantu *user* dalam mempelajari bahasa Jepang dengan algoritma Supermemo 2. Dengan menggunakan aplikasi ini *user* dapat mengatur flashcard yang ingin dipelajari dan pada saat melakukan pembelajaran *user* hanya perlu menjawab sesuai dengan menu yang telah disediakan. Aplikasi akan secara otomatis menghitung jeda pembelajaran agar dapat terjadi *Spacing Effect*. Adapun huruf Jepang yang digunakan dalam aplikasi ini adalah Hiragana, Katakana, dan *regular-use* Kanji (Joyo Kanji) sebanyak 2.136 karakter.

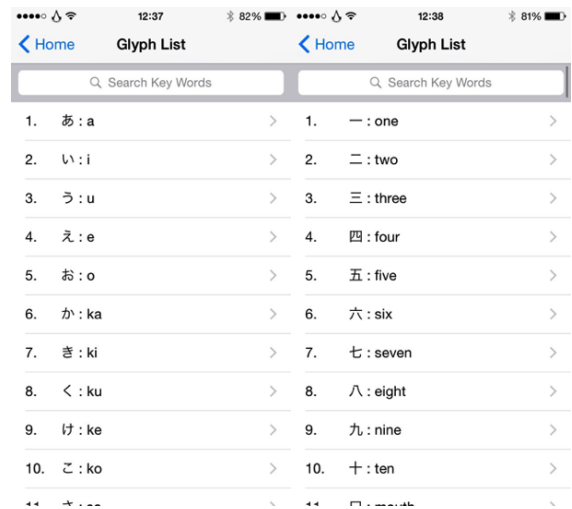
Metodologi pengembangan perangkat lunak yang digunakan adalah Waterfall Model, yang meliputi *requirements, design, implementation, verification, dan maintenance*. Metode ini dipilih karena *requirement* yang digunakan untuk mengembangkan perangkat lunak sudah jelas di awal, sehingga tidak perlu ada perubahan lagi.

Spaced Repetition Software ini memiliki dua menu utama, yaitu menu *Card List* dan menu *Learn*. Menu *Card List* digunakan untuk menampilkan data flashcard yang dapat *user* gunakan untuk dipelajari, sedangkan menu *Learn* digunakan untuk menampilkan daftar kartu yang telah *user* pilih dan juga menampilkan flashcard yang harus dipelajari pada saat itu.



Gambar 3 Diagram Alir Sistem Secara Garis Besar

IV. PENELITIAN DAN HASILNYA



Gambar 4 Tampilan Antarmuka Kana (kiri) dan Kanji (kanan)

Gambar 4 merupakan tampilan antarmuka aplikasi *Spaced Repetition Software*, saat *user* menekan tombol pada *section card list*. Disini *user* dapat melakukan *search* terhadap *flashcard* yang tersedia dan juga dapat melihat detail dari *flashcard* tersebut dengan melakukan klik pada salah satu *item* yang terdapat dalam tabel.



Gambar 5 Tampilan Antarmuka *Flashcard Detail Front* (kiri) dan *Flashcard Detail Back* (kanan)

Gambar 5 merupakan tampilan antarmuka aplikasi saat seorang *user* melakukan pilihan pada salah satu *flashcard* di menu *card list*. Aplikasi memungkinkan *user* untuk melakukan penambahan *flashcard* ke dalam *deck list*. Selain itu *user* juga dapat melihat detail *flashcard* dengan melakukan *tap* pada kartu

tampilan. Menekan tombol *close* maupun *Add to Deck* akan mengembalikan *user* ke menu Kana atau Kanji.

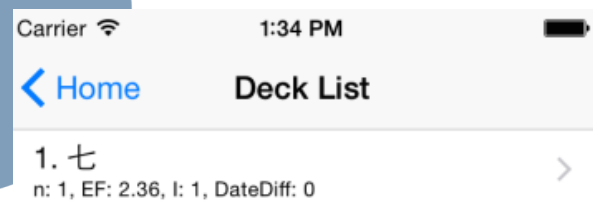
Di tahap ini, aplikasi yang dibuat diuji apakah sudah berhasil memberikan perhitungan menggunakan algoritma SM-2 dengan tepat, sehingga jeda waktu (*Interval*) yang dihasilkan sesuai dengan seharusnya sehingga jadwal yang dihasilkan tidak menjadi berantakan. Pengujian dilakukan dengan membandingkan perhitungan secara manual dan membandingkan hasil perhitungan melalui aplikasi yang telah dibangun. Sejumlah skenario perhitungan akan dibuat dimana setiap skenario akan memiliki jumlah iterasi yang berbeda-beda yaitu satu kali, dua kali dan tiga kali iterasi dengan masing-masing skenario memiliki tiga jenis *pattern* (kombinasi) yang berbeda-beda dan dipilih secara acak. Sebagai contoh *pattern* q342 berarti *user* pertama kali menjawab menggunakan *button* 3 dilanjutkan dengan *button* 4, dan terakhir dengan *button* 2.

Metode pengetesan adalah menggunakan *debugging button* yang dibuat untuk mengemulasikan agar waktu pengetesan adalah hari yang sama ketika tombol tersebut ditekan dengan membuat nilai variabel perbedaan hari yang menjadi *counter* pembanding bernilai sama dengan nilai variabel *Interval* yang merupakan hasil perhitungan dari algoritma.

Rumus 1 dan rumus 2 merupakan rumus dasar yang digunakan pada algoritma SM-2. Di dalam rumus tersebut terdapat variabel, seperti n yang merupakan urutan iterasi yang akan bertambah ketika *user* menjawab pada saat pengetesan secara berkala. Untuk nilai $n = 1$ dan $n = 2$, maka nilai interval yang dihasilkan statis yaitu 1 dan 6, selanjutnya nilai interval akan menggunakan rumus seperti yang tertera pada rumus 4.1. Setiap kali *user* menjawab, maka akan dihasilkan variabel q yang merupakan *quality* (kualitas jawaban) sesuai dengan nilai dari *user*. Ketika nilai $q < 3$, maka nilai EF tidak perlu diubah dan interval dikembalikan menjadi satu. Jika nilai $q > 2$ maka perhitungan EF dilakukan menggunakan rumus yang tersedia.

Input : $n = 1$ $q = 3$ $EF = 2.5$ Output : $I(1) = 1$ $EF' = 2.5 + (0.1 - (5 - 3) * (0.08 + (5 - 3) * 0.02))$ $EF' = 2.5 + (0.1 - 2 * (0.08 + 2 * 0.02))$ $EF' = 2.5 + (0.1 - 2 * (0.12))$ $EF' = 2.36$	#1
Input : $n = 2$ $q = 4$ $EF = 2.36$ Output : $I(2) = 3$ $EF' = 2.36 + (0.1 - (5 - 4) * (0.08 + (5 - 4) * 0.02))$ $EF' = 2.36 + (0.1 - 1 * (0.08 + 1 * 0.02))$ $EF' = 2.36 + (0.1 - 0.10)$ $EF' = 2.36$	#2
Input : $n = 3$ $q = 2$ $EF = 2.36$ Output : $n = 1$ $I = 1$ $EF' = EF$ $EF' = 2.36$	#3

Gambar 6 Contoh Hasil Perhitungan Manual



Gambar 7 Contoh Hasil Program

V. SIMPULAN DAN SARAN

Simpulan dari penelitian yang dilakukan adalah aplikasi *Spaced Repetition Software* telah berhasil dibangun dengan mengimplementasikan algoritma SM-2 untuk memberikan *inter-repetition* dari data *flashcard* yang diberikan.

Berdasarkan hasil pengujian, algoritma yang digunakan telah berhasil memberikan *interval* yang dinamis dimana nilai *easiness factor* yang dihasilkan mempengaruhi besarnya *interval* yang dihasilkan sehingga jeda pengulangan dapat berubah-ubah menyesuaikan dengan penggunaan dari *user*.

Saran yang dapat diberikan untuk penelitian selanjutnya, antara lain adalah masih diperlukan pengembangan pada sistem pembelajaran, sehingga *user* dapat memberikan inputan yang dapat diproses dan diberi nilai yang berupa variabel q atau *Quality* yang digunakan oleh algoritma SM-2. Sebagai contoh, *user* diberikan sebuah pilihan ganda atau isian untuk dijawab yang kemudian akan dikonversi menjadi variabel q dengan metode penilaian tertentu seperti misalnya *quiz* atau dengan kecepatan menjawab.

Selain itu, dapat juga ditambahkan sebuah fitur untuk menghafal bagaimana cara penulisan karakter dimana *user* diminta untuk menulis karakter tersebut di layar dan hasil *input* dari *user* akan dicocokkan dengan data penulisan yang ada. Selain itu, dapat juga ditambahkan *output* suara pada *flashcard* untuk mempelajari cara pengucapan.

DAFTAR PUSTAKA

- [1] Apple. 2014. *The Swift Programming Language*. Dalam <https://itunes.apple.com/us/book/swift-programming-language/id881256329?mt=11>. Diakses 7 Maret 2015.
- [2] Ebbinghaus, H. 2011. *Memory: A Contribution to Experimental Psychology*. German: Martino Fine Books.
- [3] Edge, D., Fitchett, S., Whitney, M., & Landay, J. 2012. MemReflex: Adaptive Flashcards for Mobile Microlearning. *ACM*.
- [4] Gwern. 2011. *Spaced Repetition*. Dalam <http://www.gwern.net/Spaced%20repetition>. Diakses 15 Maret 2015.
- [5] Heisig, J. W. 2007. *Remembering the Kanji*. Honolulu: University of Hawai'i Press.
- [6] Hanks, J., Zhan, P. 2012. *The Forgetting Curve and Learning Algorithms*. Dalam https://edo.repo.nii.ac.jp/index.php%3Faction%3Dpages_view_main%26active_action%3Drepository_action_common_download%26item_id%3D9. Diakses 21 Maret 2015.
- [7] Merriam-Webster. 2015. Dalam <http://www.merriam-webster.com/dictionary/flash%20card>. Diakses 20 Maret 2015.
- [8] Seeley, C. 1991. *A History of Writing in Japan*. BRILL.
- [9] SyndeonSoft. *The spacing effect*. 2012. Dalam <http://www.flashcardlearner.com/articles/the-spacing-effect/>. Diakses 15 Maret 2015.
- [10] You2. 1998. *Comparison of learning systems and programs*. Dalam <https://www.you2.de/karteikartensystem/lernprinzip.html>. Diakses 15 Maret 2015.
- [11] Yoneoka, J. 2006. A Tailored Intensive Vocabulary Trainer Using an Online Flashcard Site. Dalam <http://www2.kumagaku.ac.jp/teacher/~judy/letflashcardspaper.htm>. Diakses 21 Maret 2015.
- [12] Wozniak, P. 1990. *Optimization of learning*. Poznan: University of Technology.
- [13] Wozniak, P. 1998. *Application of a computer to improve the results obtained in working with the SuperMemo method*. Dalam <http://www.supermemo.com/english/ol/sm2.htm>. Diakses 21 Maret 2015.
- [14] Inet. 2016. *iOS*. Dalam <https://www.informasi-internet.com/2016/12/ios.html>. Diakses 21 Mei 2018.

