PENGGUNAAN *OBJECTIVE-C/COCOA FRAMEWORK* PADA PEMBUATAN PROTOTIPE APLIKASI KAMUS BAHASA INDONESIA PADA PERANGKAT BERGERAK

Arini[1], Andrew Fiade[2], Darma Triesavia Haryanto[3]

[1,2,3] Program Studi Teknik Informatika, Fakultas Sains dan Teknologi UIN Syarif Hidayatullah, Jakarta
Jl. Ir. H. Juanda No. 95 Ciputat 15412 Jakarta-Indonesia

[1]arinizoel@yahoo.com
[2]andrew_fiade@yahoo.co.id
[3]darma.haryanto@gmail.com

Abstrak: Manusia menggunakan bahasa sebagai alat berkomunikasi. Penggunaan bahasa yang benar, akan mengurangi kemungkinan si pembaca untuk memperkirakan maksud yang ingin disampaikan oleh si penulis, tanpa terkecuali bahasa Indonesia. Penelitian ini bertujuan untuk membangun sebuah bentuk prototipe aplikasi untuk perangkat bergerak berbasis iOS berupa kamus digital bahasa Indonesia. Pendekatan dalam pengembangan aplikasi ini adalah dengan metodologi berorientasikan objek (OOM) yang terdiri dari tahap prasyarat pengembangan, analisa dan desain, penerapan serta pengujian aplikasi. Aplikasi yang dihasilkan mampu menyediakan padanan kosakata yang ditampilkan secara lebih menarik, sederhana dan mudah digunakan.
Kata kunci: Bahasa Indonesia, *Cocoa framework*, iOS, Kamus, Metode Berorientasi Objek, MVC, Objective-C, Perangkat Bergerak, Prototipe Aplikasi

Abstract: Humans use language as a tool to communicate. Correct use of language, will reducing the chances of the reader to figure out the purpose to be conveyed by the author, and without exception for Bahasa. This study was aimed to develop a prototype of mobile application of Indonesian dictionary, in a digital form, for iOS-based devices. The approach used in the development of this application is object-oriented methodology (OOM) which consists of requirement, analysis and design, implementation and testing. Despite some existing limitations, the application that been produced is able to provide the equivalent of vocabulary displayed in a more attractive, simple way and user-friendly.
Keywords: Bahasa Indonesia, Cocoa framework, iOS, Kamus, Metode Berorientasi Objek, MVC, Objective-C, Perangkat Bergerak, Prototipe Aplikasi

I. PENDAHULUAN

Bahasa Indonesia, meskipun relatif sederhana jika dibandingkan dengan bahasa Inggris (karena dalam bahasa Indonesia tidak mengenal *tenses* dan pola pembentukan bunyi serta pembacaan kata-nya mempunyai aturan yang jelas dari huruf-huruf yang membentuk kata tersebut), telah dipengaruhi penggunaan-nya oleh banyak bahasa-bahasa serapan (mulai dari bahasa asing dan/atau bahasa daerah).

Sebelum penelitian ini, terdapat beberapa usaha atau penelitian yang pernah dilakukan perihal keprihatinan terhadap penggunaan bahasa Indonesia. Salah satunya adalah dengan merancang aplikasi pencarian makna kata bahasa Indonesia dengan J2ME [1] serta aplikasi pencarian istilah teknologi informasi berbahasa Indonesia. Namun sayangnya, aplikasi yang dibuat memiliki keterbatasan kosakata serta belum menyertakan konsep komunikasi yang baik antara sebuah perangkat lunak dengan manusia sehingga aplikasi tadi terkesan kurang ramah dan sedikit menyulitkan pengguna ketika menggunakannya.

Beberapa manfaat yang didapatkan dari penelitian ini adalah (1) Membantu si pengguna aplikasi untuk mencari makna dan pengertian dari kosakata atau istilah berbahasa Indonesia secara lebih mudah; (2) Menawarkan pemahaman secara

lengkap tentang keunggulan dan kekurangan dari iOS *development*; (3) Menambah pemahaman tentang pola rancangan arsitektur *Model View Controller* (MVC) dalam OOM.

Karena sangat luasnya ragam struktur bahasa Indonesia, maka ruang lingkup penelitian ini dibatasi pada: (1) Pengembangan aplikasi dengan *cocoa framework*, menggunakan bahasa pemrograman *objective-c*; (2) Rancangan sistem pada aplikasi bersifat statis, yang artinya adalah pengguna belum bisa ikut serta menambahkan kosakata/ istilah baru; (3) Kosakata/istilah yang terdapat dalam aplikasi adalah kosakata/istilah bahasa Indonesia yang merujuk kepada KBBI; (4)Pengujian aplikasi menggunakan program simulasi (iOS simulator).

II. TINJUAN PUSTAKA

Prototipe adalah, merupakan bentuk/ model yang mula-mula (model asal, asli) yang menjadi contoh [2] sebuah metode perancangan melalui pendekatan iteratif dalam pengembangan sistem yang berarti menggambarkan hal-hal penting sekaligus contoh terhadap pengembang-an bentuk/ model selanjutnya [3].

Aplikasi merupakan sebuah produk yang dikembangkan oleh pengembang perangkat lunak (*software engineer*) dalam bentuk program yang dapat dieksekusi oleh komputer dengan berbagai ukuran dan arsitekturnya [4].

Kata "kamus" memiliki makna buku acuan yang memuat kata dan ungkapan berikut dengan keterangan tentang makna, pemakaian atau terjemahannya [2]. Daftar kosa kata atau istilah dalam kamus, yang biasa dikenal dengan "lema" atau "entri", lazimnya disusun menurut abjad.

Kedudukan bahasa Indonesia sebagai bahasa nasional dalam tatanan kenegaraan (Seminar Politik Bahasa Nasional di Jakarta, 25-28 Februari 1975), berfungsi sebagai: (1) Lambang kebanggaan nasional; (2) Lambang identitas nasional; (3) Lambang pemersatu bangsa Indonesia yang memiliki latar belakang sosial budaya bahasa berbeda-beda; (4) Alat berkomunikasi antar-budaya dan antar-daerah.

Perangkat bergerak adalah sebuah perangkat elektronik yang oleh penggunanya bisa dibawa kemana-mana dan dapat digunakan tanpa harus diam statis pada suatu tempat. Contohnya adalah: telepon genggam (*handphone*), *smartphone*, *tablet-pc*, dan lain-lain

Sebelum dipasarkan dengan nama *objective-c*, pada awalnya dikembangkan ditahun 1985 oleh Tom Love dan Brad Cox (*Programming Technology Center*, ITT Corporation) dengan nama COOPC (*Object Oriented Programming in C*). Objective-c memisahkan deklarasi *interface* (berisikan sekumpulan metode dan properti yang menjadi gambaran struktur suatu modul terhadap modul lainnya) dari deklarasi *implementation*-nya (intruksi sesungguhnya terhadap komputer untuk mengeksekusi tugas-tugas yang telah "dipamerkan" dalam *interface*). Saat ini, *objcetive-c* menjelma sebagai sebuah bahasa pemrograman yang elegan serta secara unik pas digunakan pada pengetikan bahasa statis maupun dinamis [5].

Cocoa bermula dari sebuah proyek bernama NeXTSTEP yang dikembangkan oleh NeXTcomputer (perusahaan yang didirikan oleh Steve Jobs setelah dikeluarkan perihal konflik internal dalam perusahaan Apple Inc). NeXTSTEP ini pada awalnya adalah sebuah sistem operasi sekaligus perangkat pengembangan sistem yang menggunakan UNIX, atau lebih tepatnya BSD-UNIX yang dikembangkan oleh University of California di Berkeley, sebagai landasan inti, dengan alasan bahwa UNIX lebih jarang mengalami kerusakan (eror atau *crashed*), dan

memiliki kapabilitas untuk jaringan (*networking*) yang lebih baik serta dapat diandalkan [6].

III. METODE PENELITIAN

Secara umum, OOM terdiri dari dua proses yaitu: (1)Proses makro (*macro process*) yang merupakan tahap-tahap pengembangan secara berurutan relasional, terdiri dari: *requirement, analysis and design, implementation, testing* dan *deployment*; (2)Proses mikro (*micro process*) yang terjadi dalam tahapan *analysis and design* dimana dasar penganalisisan dan pembuatan desain sengaja dibuat buram (acak) bahkan dilakukan dengan beberapa tingkat abstraksi pada serangkaian kesatuan instruksi dalam keseluruhan proses menganalisis dan mendesain sistem [7].

Requirement, atau tahap prasyarat dan kebutuhan pengembangan aplikasi. Dilakukan dengan cara mengumpulkan, memeriksa, dan mengolah data-data terkait hasil objek observasi

Analysis and Design, atau tahap menganalisa dan merancang aplikasi. Dilakukan dengan cara meng-analisa data olahan dari tahapan *requirement* dan mulai merancang komponen kebutuhan aplikasi seperti: (1) *Identify the elements*, yaitu melakukan identifikasi bentuk-bentuk dari komponen aplikasi, termasuk didalamnya identifikasi kasus penggunaan, serta menentukan objek-objek terkait kebutuhan aplikasi; (2) *Define the colaborations between the elements*, yaitu menentukan bagaimana bentuk kerja sama antar tiap-tiap objek yang telah di-identifikasikan; (3) *Define the relationships between the elements*, yaitu menentukan bentuk hubungan antara satu objek terhadap objek lain; (4) *Define the semantics of the elements*, yaitu menentukan tingkah laku (*behaviour*) objek-objek hasil identifikasi.

Implementation, atau tahap mengimplementasikan analisa dan rancangan. Dilakukan dengan cara menerapkan hasil analisa dan rancangan ke bentuk kode-kode program.

Test, atau tahap pengujian implementasi yang telah dilakukan. Dilakukan dengan cara melakukan *compile* dan *debug* kode-kode program dan menguji fungsionalitas komponen menurut apa yang diharapkan dari hasil analisa dan desain.

Deployment, atau tahap memastikan kesiapan aplikasi, yaitu memastikan tahapan-tahapan yang dilakukan telah berjalan sebagaimana mestinya, dan dengan terlebih dahulu menggunakan bantuan program simulasi dan program pelacakan kinerja aplikasi.

IV. PEMBAHASAN

4.1 Prasyarat Kebutuhan Pembuatan Aplikasi

Spesifikasi perangkat keras (*hardware*) dan perangkat lunak yang digunakan pada saat pembuatan dan penerapan rancangan 3 rototype dari aplikasi adalah sebagai berikut

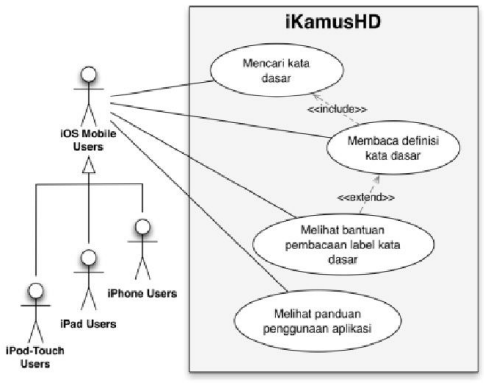
Tabel 1. Perangkat keras yang digunakan

Nama perangkat keras	:	Apple MacBook Pro (MC723LL/A)
Ukuran dan bentuk	:	2.41 x 36.4 x 24.9 (dalam cm) 15.4 inch (*diagonal*) LED-backlit *glossy widescreen display*. 1440 x 900 (*by resolution*)
Prosesor	:	2.2GHz quad-core Intel Core i7 6 MB shared L3 Cache
Memori	:	4GB (2 x 2GB SO-DIMM) memory module 1333MHz DDR3 SDRAM
Grafik	:	Intel HD Graphics 3000 AMD Radeon HD 6750M with *automatic graphic switching* 1GB DDR5
Media penyimpanan	:	750GB Serial ATA 5400 rpm

Tabel 2. Perangkat lunak yang digunakan

Spesifikasi sistem operasi yang digunakan		
Versi sistem	:	Mac OS X 10.7.5 (11G63B)
Versi kernel	:	Darwin 11.4.2 64-bit kernel and extension
Spesifikasi program untuk mengembangkan aplikasi		
Versi IDE	:	XCode 4.3.1 (4E1019)
Versi SDK	:	iPhone OS 5.1 (9B176)
Program simulasi	:	iOS Simulator 5.1
Spesifikasi program untuk mendesain aplikasi		
Pemrosesan grafis	:	Adobe Design Premium for Mac CS 5 (Adobe Photoshop CS5; Adobe Illustrator CS5)
Desain diagram & skema	:	Omni group OmniGraffle Pro 5.4
Spesifikasi basis data yang digunakan		
RDBMS	:	SQLite 3.7.7
Nama basis data	:	pustaka.sqlite
Program lainnya	:	OS X Terminal Firefox SQLite Manager add-on

4.2 Analisa dan Perancangan Aplikasi



Gambar 1. Identifikasi bentuk pemakaian aplikasi

Identifikasi bentuk pemakaian aplikasi yang akan dibuat, dapat dilihat dalam representasi diagram kasus penggunaan (*use case diagram*) dapat dilihat pada Gambar 1.

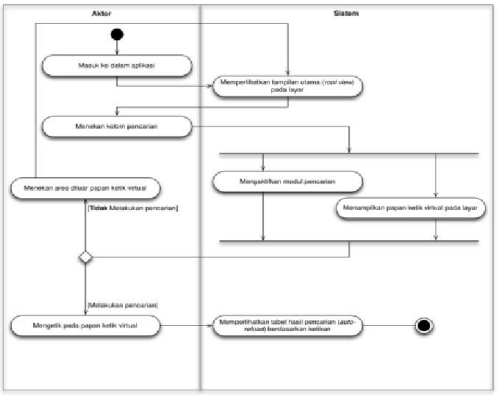
Aplikasi yang akan dibuat melibatkan aktor pengguna perangkat bergerak berbasis iOS. Pengguna perangkat bergerak berbasis iOS tersebut merupakan bentuk generalisasi dari pengguna-pengguna perangkat bergerak berupa *iPhone*, *iPod-Touch* ataupun *iPad*.

(1) Skenario utama dari identifikasi penggunaan aplikasi adalah dengan mengarahkan pengguna untuk melakukan pencarian kata dasar:

Tabel 3. Skenario penggunaan mencari kata dasar

Nama skenao :	Mencari kata dasar	
Aktor yang terlibt :	Pengguna perangkat bergerak iOS	
Deskrpsi :	Merupakan proses dimana aktor dituntut untuk mengetikkan kata dasar (yang ingin dicari definisinya) pada kolom pencarian atau *search bar*.	
Kondisi awal :	(1) Aktor menjalan aplikasi (2) Tampilan utama (*root view*) aplikasi telah terlihat pada layar	
Alur kerja dasar :	**Kegiatan pelaku**	**Respon sistem**
	[1] Aktor menekan kolom pencarian	[2] Meng-aktifkan modul pencarian
		[3] Me-nampilkan papan ketik virtual pada layar
	[4] Aktor mulai mengetik pada papan ketik virtual yang ada pada layar	[5] Me-nampilkan tabel hasil pencarian pada layar (*auto reload*) sesuai dengan kata dasar yang terbentuk dari huruf-huruf ketikan aktor
Alternatif :	Saat modul pencarian berstatus aktif, ditandai dengan masih tampak papan ketik virtual pada	

	layar, dan aktor tidak mengetikkan apapun didalam kolom pencarian (kolom pencarian kosong). Maka, tabel hasil pencarian tidak tampil pada layar dan modul dapat dinon-aktifkan dengan menekan area diluar papan ketik virtual.
Kondisi akhir :	Aktor menemukan kata dasar yang dicari tersebut dan dapat terlihat didalam tabel hasil pencarian.

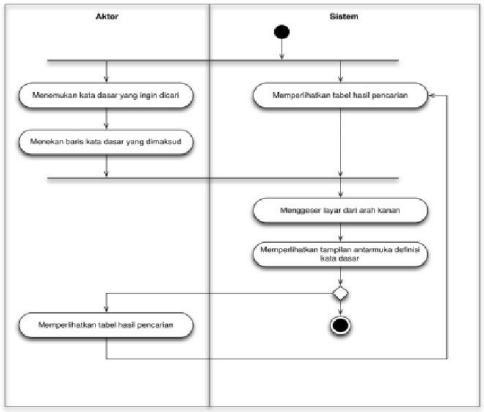


Gambar 2. Diagram aktivitas mencari kata dasar

(2) Skenario utama lainnya dari identifikasi penggunaan aplikasi adalah mengarahkan pengguna untuk melihat definisi kata dasar.

Tabel 4. Skenario penggunaan untuk membaca definisi kata dasar

Nama skenario :	Membaca definisi kata dasar	
Aktor yang terlibat :	Pengguna perangkat bergerak iOS	
Deskripsi :	Merupakan sebuah proses dimana aktor dapat membaca definisi dari kata dasar (yang ingin dicari)	
Kondisi awal	Aplikasi memperlihatkan tabel hasil pencarian yang berisikan baris-baris kata dasar pada layar dan Aktor menekan baris berisikan kata dasar (yang ingin dicari) pada tabel hasil pencarian tersebut	
Alur kerja dasar :	**Kegiatan pelaku**	**Respon sistem**
	[1] Aktor menekan baris berisikan kata dasar (yang ingin dicari) pada	[2] Layar pada aplikasi bergeser dari arah kanan dan memperlihatkan antarmuka definisi kata dasar
	tabel hasil pencarian tersebut	
Alternatif :	Aktor dapat kembali ke tampilan sebelumnya (antarmuka *root*), untuk melakukan pencarian kembali, dengan menekan tombol berbentuk indeks kamus pada sebelah kanan di tengah-tengah layar pada saat masih berada di tampilan antarmuka definisi	
Kondisi akhir :	Antarmuka definisi kata dasar ditampilkan pada layar	



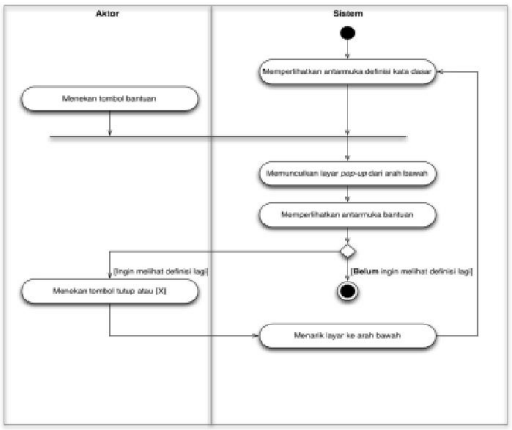
Gambar 3. Diagram aktifitas membaca definisi kata dasar

(3) Skenario selanjutnya dari identifikasi penggunaan aplikasi, dan aktif pada saat pengguna mengalami kesulitan pada saat melihat definisi kata dasar, adalah mengarahkan pengguna untuk melihat bantuan pembacaan label kata dasar.

Tabel 5. Skenario penggunaan untuk melihat bantuan pembacaan label kata dasar

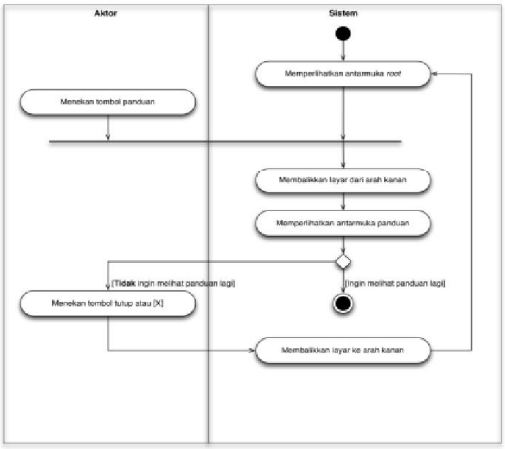
Nama scenario :	Melihat bantuan pembacaan label kata dasar	
Aktor yang terlibat :	Pengguna perangkat bergerak iOS	
Deskripsi :	Merupakan proses dimana aktor bisa melihat daftar penjelasan label-label yang digunakan pada definisi kata dasar	
Kondisi awal :	Aplikasi berada pada (memperlihatkan) tampilan antarmuka definisi kata dasar	
Alur kerja dasar :	**Kegiatan pelaku**	**Respon sistem**
	[1] Aktor menekan tombol bantuan	[2] Memperlihatkan antarmuka bantuan

	pada sebelah pojok kanan-bawah layar	yang muncul secara *pop-up* dari bawah layar
Alternatif :	Aktor dapat kembali melihat definisi kata dasar dengan menekan tombol tutup atau [X] di sebelah pojok kanan-atas layar, pada saat berada di antarmuka bantuan	
Kondisi akhir :	Antarmuka bantuan terlihat pada layar	



Gambar 4. Diagram aktifitas melihat bantuan pembacaan label kata dasar

	kanan-bawah layar	dengan membalikkan layar dari arah kanan
Alternatif :	Aktor dapat memulai pencarian dengan menekan tombol tutup atau [X] di sebelah pojok kanan-atas layar, pada saat berada di antarmuka panduan	
Kondisi akhir :	Antarmuka panduan terlihat pada layar	



Gambar 5. Diagram aktifitas melihat panduan penggunaan aplikasi

(4) Skenario selanjutnya dari identifikasi penggunaan aplikasi, dan aktif pada saat pengguna mengalami kesulitan pada saat akan menggunakan aplikasi, adalah mengarahkan pengguna untuk melihat panduan penggunaan aplikasi

Tabel 6. Skenario penggunaan untuk melihat panduan penggunaan aplikasi

Nama scenario :	Melihat panduan penggunaan aplikasi	
Aktor yang terlibat :	Pengguna perangkat bergerak iOS	
Deskripsi :	Melihat panduan penggunaan aplikasi	
Kondisi awal :	Aplikasi berada pada (memperlihatkan) tampilan antarmuka *root*	
Alur kerja dasar :	**Kegiatan pelaku**	**Respon sistem**
	[1] Aktor menekan tombol panduan pada sebelah pojok	[2] Memperlihatkan antarmuka panduan yang muncul

4.3 Implementasi dan Hasil

Dibawah ini merupakan gambar-gambar hasil tangkapan layar dari program simulasi yang sedang menjalankan aplikasi ini

Gambar 6. Gambar aplikasi ketika modul pencarian sedang aktif

Gambar 7. Gambar aplikasi saat pengguna mengetik pada keyboard virtual

Gambar 8. Proses yang terjadi pada aplikasi ketika pengguna ingin melihat definisi kata dasar yang dicari

Gambar 9. Proses pada aplikasi ketika pengguna ingin melihat bantuan pembacaan label kamus

Gambar 10. Proses pada aplikasi saat pengguna ingin melihat panduan penggunaan aplikasi

V. PENUTUP

5.1 Kesimpulan

Tahapan untuk membangun prototipe aplikasi kamus bahasa Indonesia menggunakan *Objective-C/ Cocoa framework* adalah dengan pertama-tama menentukan model atau kerangka kerja aplikasi tersebut melalui pendekatan antarmuka pengguna, kemudian merancang bentuk penggunaan basis data yang dipakai untuk menyimpan kosakata bahasa Indonesia dan menempatkan data tersebut dalam antarmuka yang telah dibuat, serta menampilkan keluaran (*output*) berupa terjemahan/ pengertian/ deskripsi berdasarkan hasil pencarian kosakata bahasa Indonesia yang dimasukkan (*input*).

5.2 Saran

Adapun saran untuk pengembangan berikutnya adalah: (1) aplikasi tidak lagi terbatas pada bentuk prototipe; (2) Diharapkan pada pengembangan selanjutnya jumlah data pada database lebih meluas lagi dan tidak hanya terbatas kepada bentuk kata dasar bahasa Indonesia.

REFERENSI

[1] Nurjaya, Mirwan. 2010. *Aplikasi Pencarian Makna Kata Dalam Bahasa Indonesia Dengan Teknologi Java 2 Micro Edition (J2ME)* (ISSN 0110-09-6645). Perpustakaan Utama UIN, Jakarta: vii + 110 t.d; 30 cm.

[2] Depdiknas, Pusat Bahasa. 2007. *Kamus Besar Bahasa Indonesia ed.3, cet.4* (ISBN 979-407-182-X). Balai Pustaka. Jakarta: xxxvi + 1386 hlm; 25 cm.

[3] Houde, Stephanie & Charles Hill. 2004. *What do Prototypes Prototype?*. e-artikel. link. http://www.apple.com. Senin, 11 Maret 2013, 03:02 pm (UTC +07:00).

[4] Pressman, Roger S, Ph.D. 2001. *Software Engineering: A Practitioner's Approach 5th ed* (ISBN 0-07-365578-3). McGraw Hill. New York: xxviii + 860 hlm; 23 cm.

[5] DeVoe, Jiva. 2011. *Objective-C: Developer Reference* (ISBN 978-0-470-47922-3). Wiley Publishing Inc. Indianapolis: pdf + 382 hlm; e-book.

[6] Hillegass, Aaron. 2008. *Cocoa Programming for Mac OS X 3rd ed* (ISBN 0-321-56273-9). Pearson Education Inc. Massachusetts: chm + 464 hlm; e-book.

[7] Booch, G, RA. Maksimchuk, MW. Engle, BJ Young, J Conallen & KA Houston. 2007. *Object Oriented Analysis and Design with Applications 3rd ed* (ISBN 0-201-89551-X). Pearson Education Inc. Massachusetts: xxiii + 691 hlm; 23 cm