

PEMBUATAN APLIKASI PENGOLAH KATA SEDERHANA

Rila Mandala¹, Debby Paseru², Alfonso Mario Tumewu³

¹Jurusan Teknik Informatika, ITB, Jl. Ganesha No. 10 Bandung

^{2,3}Teknik Informatika, Unika De La Salle Manado

e-mail rila@if.itb.ac.id¹, bydeb@hotmail.com², alfonso_mt@yahoo.com³

ABSTRAKSI

Teknologi yang berkembang saat ini, membuat perubahan yang besar bagi kehidupan manusia. Kemampuan komputer untuk mengolah data dan informasi sudah sangat baik. Contoh aplikasi yang sering digunakan dalam mengolah kata adalah Microsoft Word milik Microsoft. Dan untuk penggunaannya pun, tidak lepas dari pembayaran lisensi kepada Microsoft. Hal ini tentu merupakan hal yang berat bagi sebagian kalangan, mengingat biaya yang dibutuhkan untuk membayar lisensi ini tidaklah kecil. Salah satu cara untuk mengatasi masalah sebagian kalangan ini (pembayaran lisensi), adalah dengan membuat suatu aplikasi perangkat lunak pengolah kata sederhana dengan fitur-fitur yang umum digunakan dan yang bersifat freeware, artinya tidak perlu membayar lisensi jika ingin menggunakannya. Dalam membangun suatu aplikasi perangkat lunak, dibutuhkan suatu metodologi pengembangan sistem. Untuk itu, dalam mengembangkan aplikasi perangkat lunak pengolah kata sederhana ini, digunakanlah metodologi Rapid Application Development (RAD) dengan Unified Modelling Language (UML) sebagai alat bantu pemodelan datanya. Aplikasi perangkat lunak ini dibuat dengan menggunakan bahasa pemrograman Microsoft Visual Basic 6.0 Enterprise Edition.

Aplikasi perangkat lunak pengolah kata sederhana ini dibuat untuk membantu pengguna dalam mengolah kata/mengetik artikel dengan menyediakan fitur-fitur yang umum digunakan dan menambahkan fitur keamanan untuk data hasil olahan pengguna, serta memberikan kebebasan bagi pengguna dari segi biaya.

Kata kunci: Perangkat lunak, Pengolah kata, RAD, UML, lisensi.

1. LATAR BELAKANG

Aplikasi pengolah kata yang umum digunakan oleh masyarakat Indonesia saat ini untuk sistem operasi Windows adalah Microsoft Word milik perusahaan raksasa Microsoft. Penggunaan aplikasi ini, tidak lepas dari pembayaran lisensi kepada Microsoft. Bagi pengguna dalam skala kecil, misalnya oleh mahasiswa dan pelajar, pembayaran lisensi produk aplikasi ini dirasakan cukup berat, karena harus membayar mahal baru dapat menggunakannya, sedangkan mengingat, yang akan menggunakannya adalah pelajar yang belum memiliki penghasilan untuk mem-bayar lisensi tersebut dan juga harga CD software asli per-kepingnya sudah sangat mahal. Hal ini mengakibatkan banyaknya penggunaan software bajakan di kalangan pelajar. Lain halnya jika digunakan oleh perusahaan, hal ini tentu tidak menimbulkan masalah untuk membayar lisensi tersebut, walaupun kenyataannya, masih banyak perusahaan yang menggunakan software bajakan. Lepas dari pembayaran lisensi dan pembelian software asli yang mahal, kerugian lain yang dirasakan adalah penggunaan fasilitas pengolah kata yang tidak maksimal.

Seringkali kita menggunakan program pengolah kata dengan fasilitas-fasilitas yang pada umumnya hanya untuk membantu pengetikan dan pembuatan artikel saja. Tidak semua fasilitas yang ada diguna-kan untuk pengetikan, bahkan banyak fasilitas lain yang telah tersedia, namun tidak digunakan sama sekali. Penggunaan program aplikasi pengolah kata yang tidak maksimal ini, menimbulkan ketidak-efektifan dari segi manfaat. Kita telah mengeluarkan banyak uang untuk membayar lisensi, tetapi tidak memanfaatkan seluruh fasilitas pengolah kata secara maksimal.

Untuk mengatasi masalah seperti ini, maka akan dibuat sebuah program pengolah kata sederhana yang ditujukan untuk membantu dalam pengetikan dan pembuatan artikel. Program ini hanya akan menyediakan fasilitas-fasilitas yang umum diguna-kan/dibutuhkan dalam pembuatan atau pengetikan artikel (seperti Save, Find, mengubah ukuran font, dll), sehingga penggunaannya dirasakan lebih maksimal oleh pengguna. Program ini bersifat freeware, yang artinya tidak perlu membayar lisensi jika ingin menggunakannya. Program pengolah kata yang akan dibuat menggunakan Microsoft Visual Basic 6.0.

2. DASAR TEORI

2.1 Perangkat Lunak (Software)

Perangkat Lunak (Software) adalah sekumpulan program yang dilengkapi dengan dokumentasi yang berhubungan secara langsung ke komputer, yang digunakan untuk menjalankan fungsi-fungsi yang diinginkan [MW98]. Singkatnya, perangkat lunak adalah kumpulan instruksi-instruksi untuk sebuah komputer. Misalnya, perangkat lunak untuk manaje-men data, inventorisasi, ataupun untuk pembuatan dokumentasi. Pengembangan perangkat lunak di sini erat hubungannya dengan perkembangan sistem operasi dan aplikasi yang dijalankan di atasnya.

2.2 Aplikasi

Dari sisi Aplikasi, perangkat lunak dapat dibedakan atas tiga kategori, yaitu Personal Packaged Software, Workgroup Computing, dan Enterprise Applications [MW98]. Berikut penjelasan singkat mengenai ketiga kategori tersebut:

- *Personal Packaged Software* adalah sekumpulan perangkat lunak tertentu yang telah diintegrasikan dengan sengaja pada perangkat keras yang ada (dalam hal ini PC) dengan tujuan untuk kepentingan yang sifatnya *personal*. *Word Processor* merupakan salah satu kategori dari *Personal Packaged Software*. *Word Processor* adalah kategori perangkat lunak yang mencakup fungsi lengkap dari suatu perangkat lunak pengolah kata dengan fungsi dasar yang memperbolehkan si pengguna untuk membuat dokumentasi berupa teks. Yang termasuk dalam kategori ini yaitu *Novell's, WordPerfect for Windows* dan *Microsoft Word for Windows*.
- *Workgroup Computing* adalah perangkat lunak yang memiliki fungsi *workgroup* (kolaborasi) dalam suatu lingkungan kerja. *Dataquest* menelusuri dan melaporkan tujuh kategori utama di dalam perangkat lunak komputasi *workgroup*.
- *Enterprise Applications*, adalah serangkaian aplikasi yang didesain untuk membantu pengguna dalam hal pengelolaan data dan mendukung fungsi-fungsi bisnis dari suatu organisasi. Aplikasi-aplikasi berskala *enterprise* menggunakan kombinasi dari transaksi dan sistem pengelolaan *database*. Saat ini terdapat paket-paket aplikasi *enterprise* yang berbasis pada *client/server* dari berbagai vendor perangkat lunak, diantaranya *SAP, Oracle, Peoplesoft* dan *Dun & Bradstreet*.

2.3 Sistem Informasi: Office Automation System

Menurut Whitten, Bentley, dan Dittman [WBD04], **Sistem Informasi (SI)** dirumuskan sebagai pengaturan manusia, data, proses-proses, dan antarmuka yang saling berinteraksi untuk mendukung dan meningkatkan operasi harian suatu bisnis termasuk mendukung kebutuhan penyelesaian masalah dan pengambilan keputusan oleh manajemen dan para pengguna.

Seperti yang diketahui, sistem informasi terbagi atas beberapa level dengan tujuan yang berbeda. Menurut Kendall dan Kendall [KK03], "*sistem informasi dikembangkan untuk tujuan-tujuan yang berbeda-beda, tergantung pada kebutuhan bisnis.*" Secara komputerisasi, untuk mengolah data menjadi sebuah sistem informasi, dibutuhkan aplikasi-aplikasi pendukung. Proses pengolahan ini dapat digolongkan dalam level *knowledge*, yaitu *Office Automation System (OAS)*. *Office Automation System* hanya mendukung pekerja data, yang biasa-nya tidak menciptakan pengetahuan baru melainkan hanya menganalisis informasi sedemikian rupa untuk mentransformasikan data atau memanipulasi-nya dengan cara-cara tertentu sebelum membaginya atau menyebarkannya secara keseluruhan, dengan organisasi dan, kadang-kadang di luar itu. Aspek-aspek OAS yang sudah dikenal seperti *word processing, spreadsheets, desktop publishing, electronic scheduling*, dan komunikasi melalui *voice mail, e-mail, dan video conferencing* [KK03].

2.4 Pengolah Kata (Word Processor)

Pengolah kata atau *word processor* merupakan salah satu aplikasi pendukung yang biasa digunakan untuk mengolah data, yang umumnya disajikan dalam bentuk teks. Menurut O'Leary dan O'Leary [TL97], "*Word processing software is a program that helps you create any type of written communication. A word processor can be used to manipulate text data to produce a letter, a report, a memo, or any other type of correspondence. Text data is any letter, number, or symbol that you can type on a keyboard. The grouping of the text data to form word, sentences, paragraph, and pages of text result in the creation of a document.*

Through a word processor you can create, modify, store, retrieve, and sprint part or all of a document".

Menurut Wikipedia [WF06], "*A word processor (also more formally known as a document preparation system) is a computer application used for the production (including composition, editing, formatting, and possibly printing) of any sort of viewable or printed material*".

2.5 Enkripsi dan Dekripsi Data

Kebanyakan aplikasi pengolah kata telah menyertakan fitur-fitur yang dapat membantu dan membuat pengguna merasa nyaman. Penyimpanan pesan, data, atau informasi penting untuk diperhatikan, apalagi jika pesan, data, atau informasi tersebut bersifat rahasia yang tidak boleh diketahui oleh orang lain. Salah satu cara untuk menjaga kerahasiaan data dari pihak yang tidak berkepentingan adalah dengan menggunakan teknik kriptografi. Fitur inilah yang nantinya akan disertakan dalam aplikasi pengolah kata sederhana ini.

Kriptografi merupakan ilmu dan seni penyimpanan pesan, data, atau informasi secara aman. Dalam kriptografi proses menjaga kerahasiaan data dilakukan dengan 2 cara yaitu proses enkripsi dan proses dekripsi. Kriptografi mentransformasikan data jelas (*plaintext*) ke dalam bentuk data sandi (*ciphertext*) yang tidak dapat dikenali. *Ciphertext* inilah yang kemudian dikirimkan oleh pengirim (*sender*) kepada penerima (*receiver*). Setelah sampai di penerima, *ciphertext* tersebut ditransformasikan kembali ke dalam bentuk *plaintext* agar dapat dikenali. Proses transformasi dari *plaintext* menjadi *ciphertext* disebut proses *Encipherment* atau enkripsi (*encryption*), sedangkan proses mentransformasikan kembali *ciphertext* menjadi *plaintext* disebut proses dekripsi (*decryption*).

Tujuan dari sistem kriptografi adalah:

- *Confidentiality*
Memberikan kerahasiaan pesan dan menyimpan data dengan menyembunyikan informasi lewat teknik-teknik enkripsi.
- *Message Integrity*
Memberikan jaminan untuk tiap bagian bahwa pesan tidak akan mengalami perubahan dari saat ia dibuat sampai saat ia dibuka. Contoh untuk keutuhan (*integrity*) atas data pembayaran dilakukan dengan fungsi *hash* (pengacakan) satu arah.

- *Non-repudiation*
Memberikan cara untuk membuktikan bahwa suatu dokumen datang dari seseorang apabila ia mencoba menyangkal memiliki dokumen tersebut dengan memanfaatkan tanda tangan digital dan sertifikat digital.
- *Authentication*
Memberikan dua layanan. Pertama mengidentifikasi keaslian suatu pesan dan memberikan jaminan keotentikannya. Kedua untuk menguji identitas seseorang apabila ia akan memasuki sebuah sistem. Contohnya : Untuk jaminan atas identitas dan keabsahan (*authenticity*) pihak-pihak yang melakukan transaksi dilakukan dengan menggunakan *password* atau sertifikat digital. Sedangkan keotentikan data transaksi dapat dilakukan dengan tanda tangan digital.
- *Data Signature*
Dapat disebut juga sebagai tanda tangan digital. Berguna untuk menandatangani data digital. Contohnya adalah *Digital Signature Algorithm (DSA)*.
- *Access Control*
Untuk mengontrol akses terhadap suatu *entity*.

2.6 Metodologi Pengembangan Sistem: RAD

Whitten, Bentley, dan Dittman [WBD04] mendefinisikan metodologi *Rapid Application Development (RAD)* sebagai sebuah strategi pengembangan sistem yang menekankan kecepatan pengembangan melalui keterlibatan pengguna sistem yang ekstensif dalam konstruksi cepat, berulang, dan bertambah dengan serangkaian *prototype* yang bekerja dari sebuah sistem yang pada akhirnya berkembang ke dalam sebuah sistem *final*.

Perancangan sistem dengan menggunakan metodologi RAD memiliki lima fase, yaitu:

Fase I: Analisis kebutuhan

1. Analisis permasalahan
 - a. Identifikasi masalah, kesempatan dan titik tolak
2. Analisis sistem dan kebutuhan pengguna
 - a. Identifikasi sistem dan kebutuhan pengguna
 - b. Analisis kelayakan
 - c. Memvalidasi kebutuhan sistem dan pengguna
 - d. Memprioritaskan kebutuhan
3. Mengidentifikasi fitur dan fungsi
4. Daftar *actor* dan *use case*

Fase II: Analisis

1. Mengidentifikasi objek bisnis
 - a. Identifikasi *actor* dan *use case*
 - b. Membangun model *use case*
 - c. Mendokumentasikan *naratif use case*
 - d. Identifikasi ketergantungan *usecase*
 - e. Mengidentifikasi Objek-objek Potensial
 - f. Penyeleksian Objek-objek Potensial
2. Pengorganisasian Objek dan Pengidentifikasi Hubungan antar Objek

Fase III: Perancangan

1. Peninjauan kembali pemodelan *use case*
 - a. Mengalihkan *use case*, dari *use case* analisis ke *use case design*
 - b. Periksa kembali *use case* diagram dan dokumentasi lainnya
 - c. Pembuatan Diagram Aktivitas
2. Memodelkan antarmuka dan karakteristik objek yang mendukung skenario *use case*
 - a. Identifikasi dan klasifikasi objek pada desain *use case*
 - b. Pemodelan *high-level* interaksi objek untuk tiap *use case*
 - c. Identifikasi *behaviour* dan tanggung jawab dari setiap *usecase*
 - d. Pemodelan secara detail interaksi antara objek pada setiap *use case*
3. Perancangan antarmuka
4. Perancangan algoritma

Fase IV: Implementasi

1. Pembuatan objek
2. Pembuatan antarmuka sistem
3. Pembangunan *code*

Fase V: Pengujian

1. Menampilkan Pemeriksaan dan Pengujian Sistem
2. Kesimpulan Pengujian Aplikasi

2.7 Tools Pengembangan Sistem: Unified Modelling Language

Unified Modelling Language (UML) adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun.

Adapun komponen-komponen UML yang akan digunakan dalam pembuatan proyek ini adalah sebagai berikut:

- Use Case Diagram

Menurut Dharwiyanti dan Wahono [DW03], *Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Sebuah *use case* mempresentasikan sebuah interaksi antara *actor* dengan sistem. *Use case* merupakan sebuah pekerja-an tertentu, misalnya *login* ke sistem, *meng-create* sebuah daftar belanja, dan sebagainya. Sebuah *use case* dapat di-include oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsi-onalitas yang *common*. Sebuah *use case* juga dapat mengextend *use case* lain dengan *behaviournya* sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

- **Class Diagram**
Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. Class menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).
Class Diagram menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi dan lain-lain.
- **Activity Diagram**
Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka ber-akhir. Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.
Sebuah aktivitas dapat direalisasikan oleh satu use case atau lebih. Aktivitas menggambarkan proses yang sedang berjalan, sementara use case menggambarkan bagaimana actor menggunakan sistem untuk melakukan aktivitas.
- **Sequence Diagram**
Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu. Sequence diagram terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

3. ANALISIS

3.1 Identifikasi, Analisis Masalah & Kesempatan

Tabel 1. Cause-Effect Matrix

No.	Problems, opportunities and directives	Cause-effect	
		Causes	Effects
1.	User kekurangan pengetahuan dalam menjalankan aplikasi pengolahan kata.	Kurangnya membaca buku-buku dan tutorial mengenai aplikasi pengolahan kata membuat user bingung dalam menjalankan aplikasi.	User tidak dapat menggunakan aplikasi pengolahan kata dengan nyaman, karena harus mencoba setiap fitur yang ada.
2.	User tidak dapat mengetahui bagaimana menggunakan setiap fitur yang ada dalam aplikasi pengolahan kata serta fungsinya.	Kurangnya informasi yang tersedia mengenai fitur-fitur pengolahan kata serta fungsinya.	User kebingungan dalam menentukan fitur-fitur mana saja yang bisa digunakan, dan apa kegunaan fitur-fitur tersebut.

No.	Problems, opportunities and directives	Cause-effect	
		Causes	Effects
3.	Aplikasi yang akan dibangun ini hanya untuk pengetikan dan pembuatan artikel saja, sehingga fitur-fitur yang tersedia hanya yang berkaitan saja.	Tidak semua fitur-fitur yang ada pada aplikasi pengolahan kata akan digunakan dalam pembuatan artikel atau pengetikan.	Penggunaan aplikasi pengolahan kata akan dirasakan lebih maksimal /efektif.
4.	Adanya fitur Enkripsi dan Dekripsi untuk keamanan data.	Data dapat dengan mudah dibaca oleh sembarang orang karena tidak adanya fitur Enkripsi.	Dengan adanya fitur Enkripsi dan Dekripsi, data menjadi lebih aman dan tidak mudah dibaca oleh siapapun.

3.2 Analisis Kelayakan

Kelayakan Operasional:

1. Apakah sistem yang dibuat dapat mengakibatkan kerusakan?
Tidak.
2. Apakah dalam menjalankan aplikasi ini, dapat menimbulkan pengaruh buruk bagi user?
Tidak

Kelayakan Ekonomi

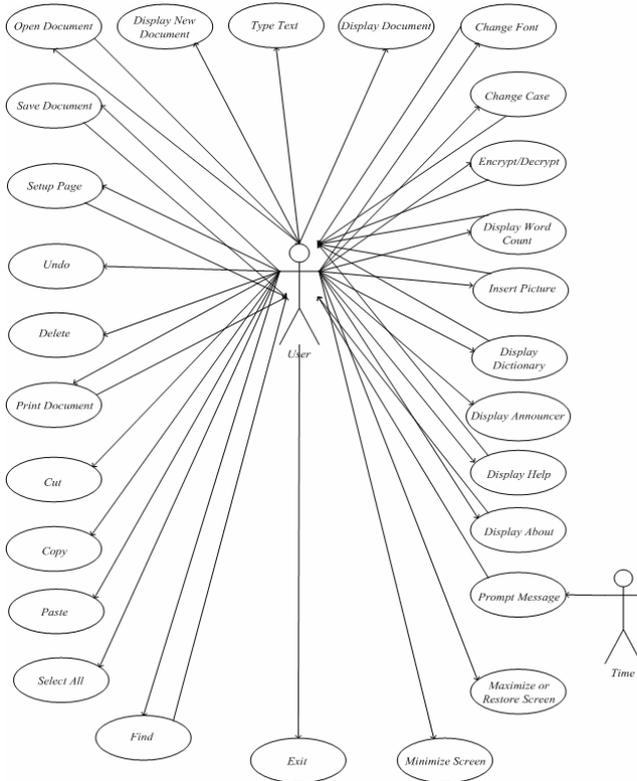
Hardware dan Software pendukung untuk melaksanakan proyek ini pada umumnya telah tersedia dan tidak memerlukan biaya tambahan yang cukup besar untuk perancangan aplikasi pengolahan kata ini.

3.3 Identifikasi Fitur dan Fungsi

Tabel 2. Features and Functions

No	Features	Description
1.	Menu Utama	Menampilkan Icon Toolbars, menu Dokumen, Edit, Format, Alat Bantu dan Bantuan.
2.	Icon Toolbars	Berisi icon's yang dapat digunakan untuk mempermudah penggunaan aplikasi.
3.	Dokumen	Berisi submenu Dokumen Baru, Buka Dokumen, Simpan, Pengaturan Halaman, Cetak dan Keluar
4.	Edit	Berisi submenu Batalkan, Potong Kata, Salin, Tempelkan, Hapus, Pilih Semua dan Cari.
5.	Format	Berisi submenu Pengaturan Huruf dan Ubah Kata.
6.	Alat Bantu	Berisi submenu Sisip Gambar, Enkripsi/Dekripsi, Penghitungan Kata, Kamus dan Pembaca Dokumen.
7.	Bantuan	Berisi submenu Bantuan dan Tentang Program.

3.4 Analisis Sistem



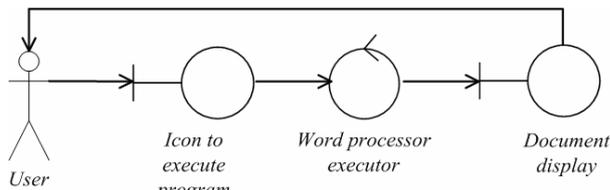
Gambar 1. Diagram Model Use case

Select document name
Press "Open"
Saved document display

- Pseudocode untuk Cari**
 Display document
 Select submenu "Cari"
 Cari dialog box display
 Type the search text
 Press "Cari"
 If word is found
 Word selected
 else
 prompt message "<word> tidak ditemukan!"
 end if
- Pseudocode untuk Enkripsi/Dekripsi**
 Display document
 Select submenu "Enkripsi/Dekripsi"
 Enkripsi/Dekripsi dialog box display
 Press "Ambil Teks Dari Dokumen"
 All text from document active display in result box
 Type password
 Select the option
 If selected option is Encrypt then
 Text encrypted
 Update text in result box
 else if password=0 then
 prompt message "Silahkan masukkan password."
 Press "Salin Hasil Ke Dokumen"
 All text in result box has copied to document active
 end if
- Pseudocode untuk Pembaca Dokumen**
 Display document
 Select submenu "Pembaca Dokumen"
 Animation character display
 If text=0
 Animation character hide
 else
 read and sepak text
 end if
 animation character hide

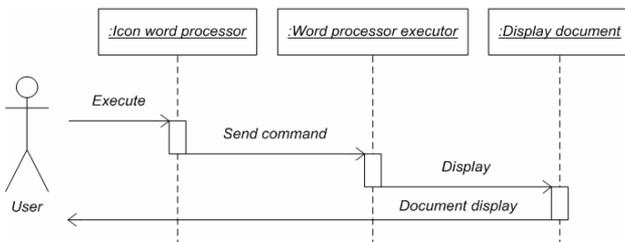
4. PERANCANGAN

4.1 Pemodelan Interaksi Objek Tingkat Tinggi untuk Setiap Use Case



Gambar 2. Ideal Object Model Diagram untuk "Display Document" Use Case

4.2 Pemodelan Secara Detil Interaksi Antara Objek Pada Setiap Use Case



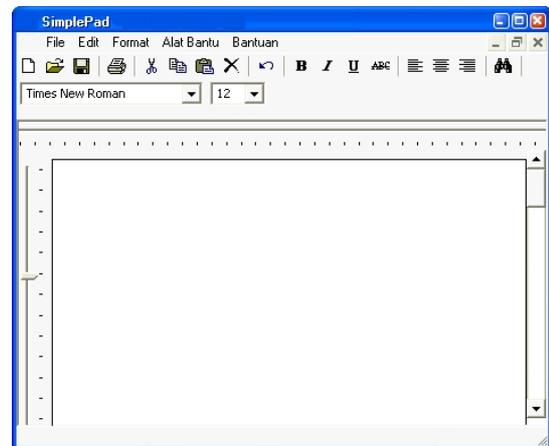
Gambar 4.2 Object Interaction Diagram untuk "Display Document" Use Case

4.3 Perancangan Algoritma

Pseudocode

- Pseudocode untuk Buka Dokumen**
 Select submenu "Buka Dokumen"
 Buka Document dialog box display
 Chose document type

4.4 Perancangan Antarmuka



Gambar 4.3 Template Interface Menu Utama

5. PENGIMPLEMENTASIAN

5.1 Lingkungan Implementasi

Operating system

Operating system yang dapat digunakan untuk menjalankan aplikasi ini adalah:

- *Windows Millennium Edition* (ME)
- *Windows 2000*
- *Windows XP Professional Edition*
- *Windows XP Professional Home Edition*

Hardware

Spesifikasi *hardware* yang dibutuhkan sebagai berikut :

- Komputer minimal Pentium III (disarankan Pentium IV).
- Memori RAM 256 MB (disarankan).
- Ruang *hard disk* yang tersedia minimal 50 MB.
- Resolusi monitor 1024 x 768 pixels.
- *Sound card* (disarankan).
- *Speaker*.
- *Keyboard*.
- *Mouse*.

5.2 Menulis Prosedur

• Prosedur Aplikasi untuk Buka Dokumen

```
Dim sFile As String
If ActiveForm Is Nothing Then
LoadNewDoc
With dlgCommonDialog
.DialogTitle = "Buka Dokumen"
.CancelError = False
.Filter = "SimplePad Files
(*.rtf; *.txt)|*.rtf;*.txt|Rich Text
Documents_
(*.rtf)|*.rtf|Plain Text Files
(*.txt)|*.txt|All Files (*.*)|*.*"
.ShowOpen
If Len.FileName) = 0 Then
Exit Sub
sFile = .FileName
End With
ActiveForm.rtfText.LoadFile sFile
ActiveForm.Caption = sFile
ClearFilename
```

• Prosedur Aplikasi untuk Simpan Dokumen

```
Dim sFile As String
If Right$(ActiveForm.Caption, 14) =
"Dokumen Kosong" Then
MnuFileSimpanSebagai_Click
Else
sFile = ActiveForm.Caption
ActiveForm.rtfText.SaveFile
sFile
End If
Changed = False
ClearFilename
```

6. PENGUJIAN

Dari pengujian yang telah dilakukan, maka dapat disimpulkan sebagai berikut:

- Aplikasi telah menyediakan beberapa fitur yang dilengkapi dengan fungsi fungsi tertentu agar mempermudah *user* dalam menjalankan aplikasi.
- Pada bagian Pengaturan Halaman masih perlu untuk dilakukan perbaikan/penyempurnaan.

7. KESIMPULAN DAN SARAN

7.1 Kesimpulan

Berdasarkan penelitian yang dilakukan, penulis dapat menyimpulkan beberapa hal sebagai berikut:

1. Aplikasi Pengolah Kata yang dibuat dapat mengerjakan tugas-tugas pengetikan seperti pembuatan artikel, walaupun hanya menggunakan fitur-fitur yang umum digunakan dalam pengolah kata lainnya.
2. Aplikasi Pengolah Kata yang dibuat bersifat *freeware*, sehingga bisa didistribusikan dengan bebas tanpa harus membayar *licence* kepada siapapun.

7.2 Saran

Untuk pengembangan lebih lanjut Aplikasi Pengolah Kata yang dibangun, disarankan untuk memiliki fungsi yang dapat *me-load* dokumen dari aplikasi pengolah kata lain, misalnya *Microsoft Word*, dan menambahkan beberapa fitur, yaitu *Print Preview*, *Insert Table*, *Bullet and Numbering*, *Page Number*, *Header and Footer* dan *Autosave*, agar Aplikasi Pengolah Kata ini lebih lengkap.

DAFTAR PUSTAKA

- [DW03] Dharwiyanti, S., dan Wahono, R.S., **Pengantar Unified Modeling Language(UML)**, 2003 from <http://ilmukomputer.com/umum/yanti/yanti-uml.zip>, Tanggal Akses 05/03/2006 Pukul 13:50 WITA.
- [KH06] Karl, H., **Dictionary English-Indonesian**, 2006 from <http://www.lzesoftware.com/dictionary/engind/dict.html>, Tanggal Akses 07/04/2006 Pukul 15:25 WITA.
- [KK03] Kendall, K.E., dan Kendall, J.E., **Analisis Dan Perancangan Sistem**, Jilid 1, PT. Prenhallin Do, Jakarta, 2003, hal 2-3.
- [MW98] Martin Wibisono, **Pengenalan Industri Perangkat Lunak**, 1998 from <http://statistic.gunadarma.ac.id/idkf/idkf/aplikasi/n21/n21-software-bab2-industrisoftware-05-1998.doc>, Tanggal Akses 05/03/2006 Pukul 00:40 WITA.
- [TL97] Timothy J.O'Leary, dan Linda L.O'Leary, **Microsoft Word 7.0a for Windows 95**, McGraw-Hill, New York, 1997, pg. 5.
- [WBD04] Whitten, J.L., Bentley, L.D., and Dittman, K.C., **System Analysis And Design Methods**, 6th edition, McGraw-Hill, New York, 2004.
- [WF06] Wikipedia, **Word Processor**, Wikimedia Foundation, 2006 From http://en.wikipedia.org/wiki/word_processor.htm, Tanggal Akses 25/02/2006 Pukul 13:47 WITA.