

**Perancangan Model Berorientasi Objek
Menggunakan *Unified Modeling Language* (UML)
Studi Kasus Sistem Pengolahan Parkir Pada PT. TRIKARYA ABADI**

T. Sumarta¹⁾, B. Siswoyo¹⁾, N. Juhana¹⁾

¹ Jurusan Teknik Informatika, FT, UNIKOM,
email : www.tatang_sumarta@plasa.com

Abstraksi

Dunia bisnis, khususnya pada perparkiran yang semakin kompleks dengan bertambah luasnya ruang lingkup dan tahapan proses yang bergantung pada sistem informasi berbasis komputer, telah mempengaruhi peningkatan pasar perangkat lunak sekaligus juga menjadi tantangan bagi dunia rekayasa perangkat lunak untuk memiliki teknik rekayasa dalam membangun sistem perangkat lunak tersebut yang dapat meningkatkan kualitas serta mengurangi biaya dan waktu. Teknik rekayasa perangkat lunak pemodelan visual dengan bahasa *Unified Modeling Language* (UML) yang bekerja dalam *Object Oriented Analysis and Design* (OOAD) dan tool pemodelan visual berupa Rational Rose yang mendukung *round-trip engineering* dan *reverse engineering* pada kode bahasa pemrograman seperti Visual Basic 6.0, dapat memberikan gambaran visual mengenai sistem yang akan dibangun dengan membuat suatu himpunan semantik dan notasi yang mampu menangani kerumitan arsitektural dalam segala ruang lingkup sehingga dapat dimiliki sebuah cetak biru dari sistem perangkat lunak tersebut.

Model yang dikerjakan dengan UML pada penyusunan skripsi ini ada dua yaitu model bisnis dan model untuk rekayasa perangkat lunak. Model bisnis menggambarkan proses bisnis perparkiran yang terjadi pada Perusahaan PT. TRIKARYA ABADI. Model untuk rekayasa perangkat lunak menggambarkan model sistem perangkat lunak untuk perangkat lunak pengolahan parkir, seperti model analisis, model desain, model basis data, model fisik perangkat lunak dan model perangkat keras. Setelah model dibuat dapat diterapkan pada pemrograman.

Abstract

Business world, specially at parking which progressively complex with increasing scope broadness and process step base on information system base on computer, have influenced the make-up of market of software at the same time also become challenge to world engineer software to have technique engineer in developing system of software which can improve quality and also lessen time and expense. Technique engineer software for visual modeling with language of Unified Modeling Language (UML) laboring in Object Oriented Analysis and Design (OOAD) and visual Modeling tool in the form of is Rational Rose supporting round-trip engineering and reserve engineering at language code of making of software like Visual Basic 6.0, can give visual picture concerning system to be

woke up by making semantic gathering and notation capable to handle complication of architecture in all scope so that can have a printing blue from system of software.

There are two model with UML is business model and model for the engineering of software. Business model depict parking business process at Company Of TRIKARYA ABADI. Model for the engineering of software depict software system model for the software of processing park, consist of analysis model, design model, database model, software physical model and hardware model. After model there is can be applied at making of software.

1. PENDAHULUAN

Membangun sistem *software* yang kompleks memerlukan perancangan model yang sistematis dalam mengerjakan pekerjaan analisis dan desainnya. Dengan Rational Rose yang merupakan *tool* dari bahasa *Unified Modeling Language* (UML) maka pekerjaan tersebut dapat dengan mudah dirancang karena merupakan pemodelan secara visual dan memiliki semantik dan notasi UML yang bekerja dalam *Object Oriented Analysis and Design* (OOAD).[5]

2. TEORI

Sistem adalah sebuah tatanan (keterpaduan) yang terdiri atas sejumlah komponen fungsional yang saling berhubungan dan secara bersama-sama bertujuan untuk memenuhi suatu proses atau pekerjaan tertentu.[6]

UML adalah bahasa pemodelan yang konsisten, dengan sistem arsitektur yang bekerja dalam OOAD untuk menentukan visualisasi, konstruksi dan mendokumentasikan *artifact* dari sistem *software*. Model yang dikerjakan dengan UML ada dua yaitu model bisnis dan model untuk rekayasa *software*. UML memiliki diagram grafis seperti *use case*

diagram, class diagram, statechart diagram, activity diagram, sequence diagram, collaboration diagram, component diagram dan deployment diagram.[13]

Tujuan penelitian ini yaitu menganalisa Sistem Pengolahan Parkir agar didapat model bisnis dan model sistem *software* sehingga memberikan solusi yang baik untuk kelancaran sistem perusahaan dan diterima oleh pekerja dalam memudahkan pengolahan parkir dalam aktivitasnya.

3. METODE PENELITIAN

Objek dari penelitian penyusun yaitu pada Sistem Pengolahan Parkir dan UML. Penelitiannya yaitu menganalisa Sistem Pengolahan Parkir kemudian merancang model dengan bahasa UML yang berorientasi objek dan mengimplementasikannya terhadap *software* dengan bahasa pemrograman Visual Basic 6.0.

Tempat penelitian dilakukan di Tempat Parkir Mall Istana Plaza pada PT. TRIKARYA ABADI dan Jurusan Teknik Informatika UNIKOM terhitung dari bulan Maret sampai dengan Juni 2004.

Metode penelitian yang digunakan yaitu metode deskriptif jenis studi

kasus yang memusatkan perhatian pada suatu kasus secara intensif dan mendetail, subyek yang diteliti terdiri dari satu unit (atau satu kesatuan unit) yang dipandang sebagai kasus. Studi kasus juga memperlihatkan kebulatan dan keseluruhan kasus, termasuk bila diperlukan kebulatan siklus hidup kasus dan keseluruhan interaksi faktor-faktor dalam kasus. Kemudian metode pemecahan masalah yang diambil menggunakan metode *object oriented Unified Modeling Language (UML)* yang merupakan gabungan dari metode *object oriented* BOOCH, OMT dan OOSE.

Bahan dan alat penelitian yang digunakan diantaranya CD *software* pendukung program aplikasi seperti Sistem Operasi Windows, Rasional Rose, Visual Basic 6.0, Crystal Report 8.0, Microsoft Office 2000, seperangkat komputer AMD Duron 1.1 GHz, memori 160 MB, Hardisk 40 Giga, Monitor 15 inci.

4. HASIL DAN PEMBAHASAN

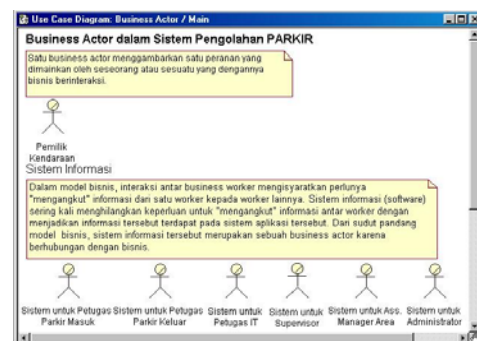
Untuk merancang model, rasional rose telah mengeluarkan satu proses yang bernama *Unified Rational Process (RUP)*. RUP memberikan pedoman secara umum dalam pembuatan model bisnis dan model sistem *software*.

a. Model bisnis

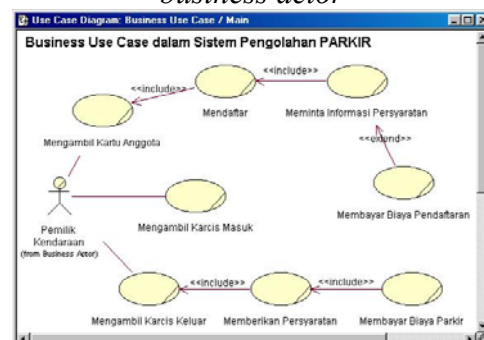
Model bisnis memberikan gambaran mengenai orang yang menjalankan bisnis, proses dari bisnisnya dan objek yang mendukung bisnisnya. Model bisnis dikelompokkan menjadi dua pemodelan yaitu *business use case model* dan *business object model*.

Pada *Business use case model* dijelaskan hubungan antara aktor

yang melaksanakan bisnis dengan proses-proses bisnisnya sehingga terdapat *business actor* dan *business use case*. *Business actor* digambarkan dengan sebuah *use case diagram* pada gambar 3.1 dan *Business use case* digambarkan dengan sebuah *use case diagram* seperti pada gambar 3.2

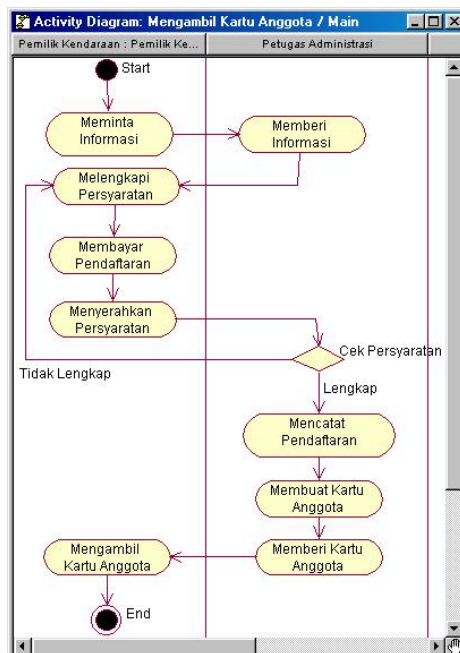


Gambar 3.1 Use case diagram
business actor



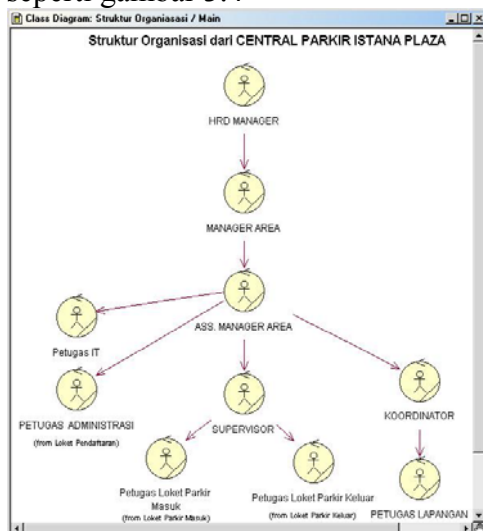
Gambar 3.2 Use case diagram
business use case

Setiap proses bisnis atau rangkaian *business use case* seperti rangkaian mengambil kartu anggota untuk proses pendaftaran digambarkan proses bisnisnya dengan sebuah *activity diagram*. *Activity diagram* untuk rangkaian mengambil kartu anggota pada proses pendaftaran anggota parkir terlihat pada gambar 3.3



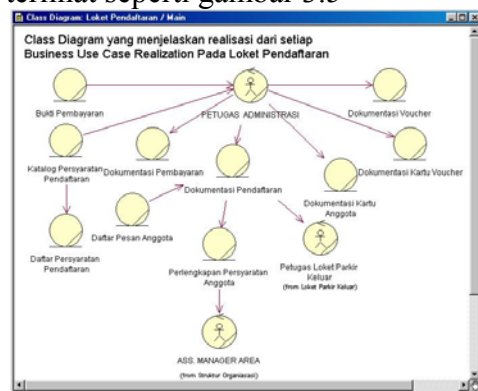
Gambar 3.3 Activity diagram pada proses pendaftaran

Pada *business object model* dijelaskan objek-objek yang mendukung proses bisnis. Objek-objek tersebut dikelompokkan menjadi dua kelompok diantaranya struktur organisasi dan unit organisasi. Struktur organisasi digambarkan dengan *class diagram* seperti gambar 3.4



Gambar 3.4 Class diagram Struktur Organisasi

Unit organisasi terdiri dari Loker Pendaftaran dan Loker Pemarkiran. Loker Pemarkiran terdiri dari Loker Parkir Masuk dan Loker Parkir Keluar. Setiap unit organisasi yang didalamnya terdapat proses bisnis sehingga terdapat objek yang digunakan, maka direalisasikan dengan beberapa *class diagram*, seperti untuk unit organisasi Loker Pendaftaran pada proses pendaftaran terlihat seperti gambar 3.5



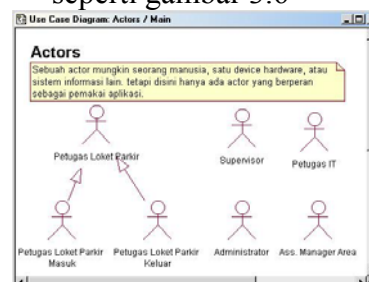
Gambar 3.5 Class diagram yang menggambarkan objek proses pendaftaran

b. Model software

Model *sistem software* (dalam rekayasa *software*) menggambarkan :

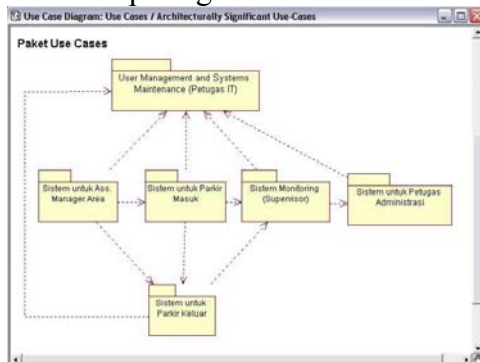
1. Interaksi pengguna *software* (*user*) atau disebut *actor* dengan perilaku *software* aplikasi yang disebut *use case*.

Aktor digambarkan oleh sebuah *use case diagram* seperti gambar 3.6



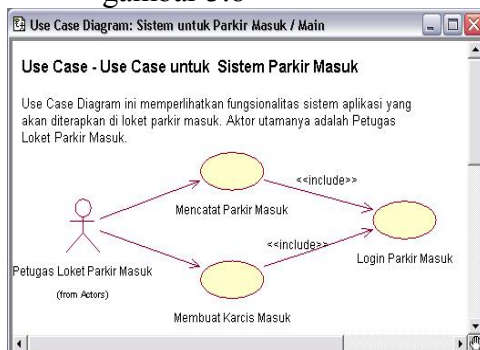
Gambar 3.6 Use case diagram yang menggambarkan pengguna *software*

Melihat pada *business actor* yang berupa sistem informasi yang mempunyai hubungan dengan pengguna *software* maka *use case* dikelompokkan menjadi beberapa paket seperti gambar 3.7



Gambar 3.7 Use case diagram Paket use case

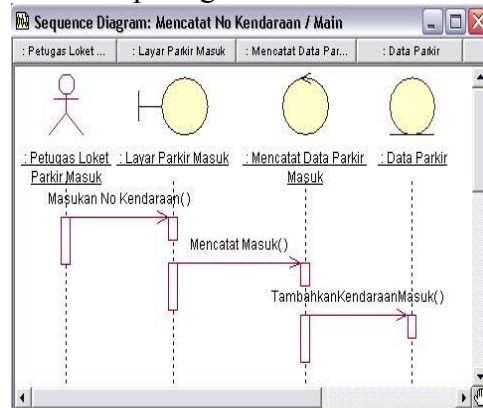
Setiap paket pada gambar 3.7 mempunyai hubungan antara pengguna *software* (*actor*) dengan *use case*, seperti pada paket Sistem Parkir Masuk mempunyai hubungan seperti gambar 3.8



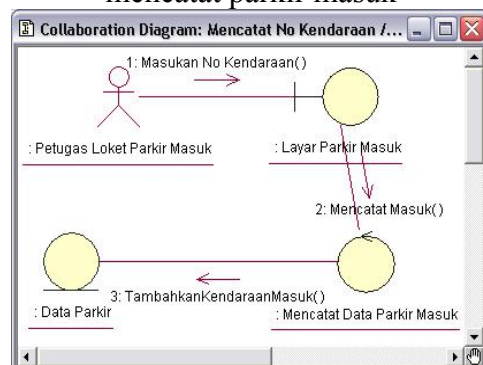
Gambar 3.8 Use case diagram dalam paket use case sistem parkir masuk

- Analisa dari *software* aplikasi tentang realisasi dari perilaku *software* aplikasi (*use case*). Setiap *use case* direalisasikan dengan *class diagram*, *sequence diagram* dan *collaboration diagram*. Pada *use case* Mencatat Parkir

Masuk mempunyai *sequence diagram* seperti gambar 3.9 dan *collaboration diagram* seperti gambar 3.10

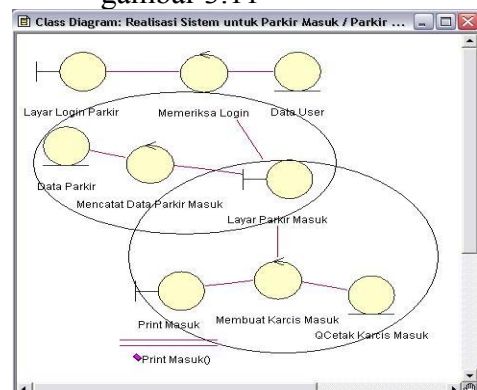


Gambar 3.9 sequence diagram mencatat parkir masuk



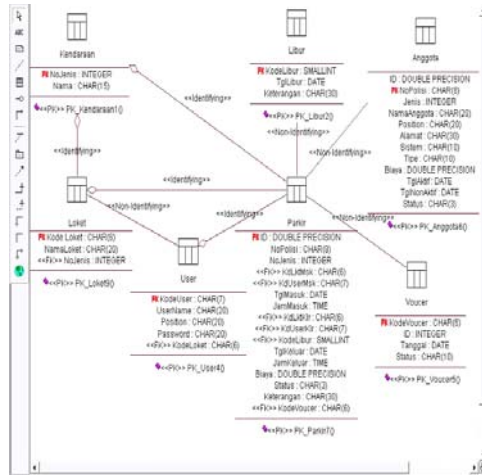
Gambar 3.10 collaboration diagram mencatat nomor kendaraan

Class diagram dalam setiap *Use case* dapat disatukan menjadi *class diagram* seperti gambar 3.11



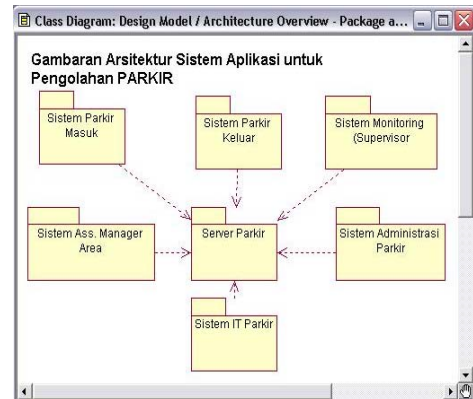
Gambar 3.11 Class diagram realisasi Sistem Parkir Masuk

3. *Database* yang terdiri dari tabel-tabel yang mempunyai atribut dan saling berhubungan seperti gambar 3.12



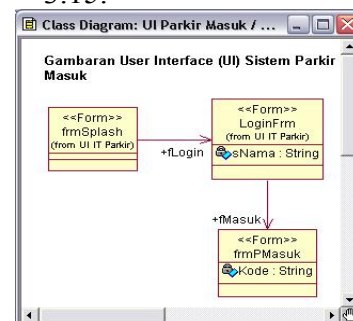
Gambar 3.12 Database parkir

4. Desain *software* yang merupakan abstraksi dari penerapan sistem *software* atau merupakan gambaran dari arsitektur *software* yang didalamnya terdapat *class* ber-stereotype yang menunjukkan *source code* seperti form berupa *user interface* dan modul. Desain *software* juga di kelompokkan menjadi beberapa paket diantaranya seperti gambar 3.13, pada setiap paket memiliki *user interface* dan *business object* *software*.

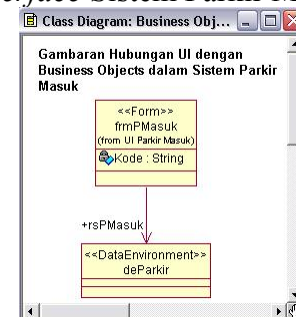


Gambar 3.13 Paket-paket pada desain *software*

Class diagram yang menggambarkan *user interface* pada paket Sistem Parkir Masuk terlihat pada gambar 3.14 dan *Class diagram* yang menggambarkan *business object* *software* beserta hubungannya dengan *user interface* terlihat pada gambar 3.15.

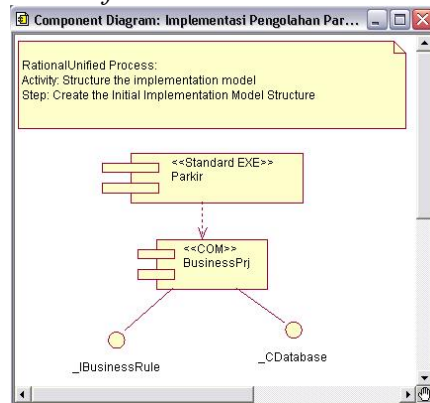


Gambar 3.14 *class diagram* *user interface* Sistem Parkir Masuk



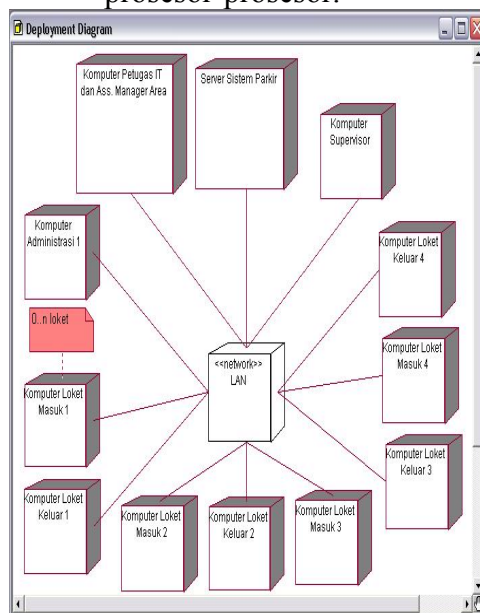
Gambar 3.15 *class diagram* hubungan UI dengan BO Sistem Parkir Masuk

5. Model fisik *software* yang terdiri dari kumpulan komponen yang mendukung *software*.



Gambar 3.16 *Component diagram software* Pengolahan Parkir

6. Model *hardware* yang terdiri dari prosesor-prosesor sehingga didapat gambaran bagaimana proses yang terjadi dalam sistem dialokasikan ke dalam prosesor-prosesor.



Gambar 3.17 *Deployment diagram* sistem *software* Pengolahan Parkir

Setelah model dibuat maka model dapat di *generate* ke dalam sebuah *project*. Model yang di *generate* hanya model dalam tahap desain *software* yang terdiri dari kelas yang ber-*stereotype* form, modul, class modul, objek data environment, dan lainnya.

5. KESIMPULAN

- Hasil pengujian telah menunjukkan bahwa model dengan *software* dapat memenuhi keserasian dalam kode.
- Pemodelan dengan UML dapat menghasilkan gambaran yang jelas dan memberikan kemudahan dalam menganalisis, desain dan implementasinya.
- Teknologi pemodelan dan pemograman berorientasi objek memudahkan dalam pengembangan *software* baik itu merubah, menambah dan memperbaikinya arsitektur *software*.
- Dalam rekayasa *software* harus dapat mengikuti perkembangan teknologi sehingga kualitas *software* akan terus lebih baik.

Daftar Pustaka

1. Abdul Kadir, 1990, **Trik dan Teknik Pemrograman Turbo Basic**, edisi 1, cetakan 1, Andi Offset Yogyakarta, Yogyakarta.
2. Agus Kurniawan, 2003, **Pemrograman COM, DCOM, dan COM + dengan Visual Basic 6.0**, Elex Media Komputindo, Jakarta.
3. Anonim, 2002, **Seri Panduan Pemrograman Database Visual Basic 6.0 dengan Crystal Reports**, kerjasama Madcoms dan andi Yogyakarta, Madiun.
4. Budi Permana, 2000, **Seri Penuntun Praktis Microsoft Access**, Elex Media Komputindo, Jakarta.
5. Eric J. Naiburg and Robert A. Maksimchuk, 2001, **UML for Database Design**, 119-147, 257, NJ.
6. Fathansyah, Ir. , 1999, **Basis Data**, cetakan 1, Informatika Bandung , Bandung.
7. Hengky Alexander Mangkulo, 2004, **Tip dan Trik Pemrograman Database Visual Basic 6.0 dan Microsoft Access**, ,Cetakan 1, 131-141, Elex Media Komputindo, Jakarta.
8. Henry Pandia, 2002, **Visual Basic 6 Tingkat Lanjut**, Andi Yogyakarta. Yogyakarta.
9. Kartono Susanto, 1995, **Pemrograman Berorientasi pada Objek dengan Borland C++**, edisi 1, cetakan 1, Andi Offset Yogyakarta, Yogyakarta.
10. Mulyadi, 2001, **Sistem Akuntansi**, edisi ke-3, Cetakan ke-3, Salemba Empat, Jakarta.
11. Peter Erik Mellquist, 2002, **SNMP ++ Pendekatan Berorientasi Objek untuk Pengembangan Aplikasi Manajemen Jaringan**, Ed. 1, 41-52, Andi Yogyakarta, Yogyakarta.
12. Panitia Tugas Akhir Jurusan Teknik Informatika, 2002, **Buku Pedoman dan Penulisan Tugas Akhir**, Universitas Komputer Indonesia (Unikom) Bandung.
13. Suhendar A. dan Hariman G., 2002, **Visual Modeling Menggunakan UML dan Rational Rose**, Cetakan 1, Informatika, Bandung.