

# LAPORAN PENELITIAN

---

*Perancangan dan Pembuatan Perangkat Lunak untuk Ujian Elektronik  
(Aplikasi untuk menyusun dan melakukan pengujian terkomputerisasi)*



Disusun oleh:

**Laurentius Risal, ST**

Fakultas Teknologi Informasi

Universitas Kristen Maranatha

Bandung

2009

## LEMBAR IDENTITAS

1. Judul Penelitian:

**Perancangan dan Pembuatan Perangkat Lunak untuk Ujian Elektronik**

2. Ketua/Penanggung Jawab Pelaksana Kegiatan Penelitian:

- Nama / NIK : Laurentius Risal, ST / 710001.
- Jabatan Akademik / Golongan : Asisten Ahli / IIIB.
- Fakultas / Jurusan : Teknologi Informasi / D3 Teknik Informatika.

3. Anggota Pelaksana Kegiatan Penelitian:

- Nama / NIK : Erico Darmawan Handoyo / 720247
- Jabatan Akademik / Golongan : Asisten Ahli / IIIB
- Fakultas / Jurusan : Teknologi Informasi / S1 Teknik Informatika.

4. Jumlah Tim Peneliti : 2 (satu) orang.

5. Lokasi Pelaksanaan Penelitian : Fakultas Teknologi Informasi.  
Universitas Kristen Maranatha.

6. Lama Pelaksanaan Penelitian : 5 (lima) bulan, 3 (tiga) minggu.

7. Sumber Data Penelitian : Fakultas Teknologi Informasi.  
Universitas Kristen Maranatha.

8. Biaya Penelitian : Rp. 5.670.000,00

*TERBILANG* : #Lima Juta Enam Ratus Tujuh Puluh Ribu Rupiah#.

Bandung, 16 Desember 2009

Mengetahui,  
LPPM

Menyetujui,  
Fakultas Teknologi Informasi

Pelaksana,  
Ketua / Penanggung Jawab

Ir. Yusak Gunadi Santoso, MM.  
(Ketua LPPM)

Radiant Victor Imbar, SKom, MT  
(Dekan Fakultas Tek. Informasi)

Laurentius Risal, ST.  
(Ketua Peneliti)

## KATA PENGANTAR

Terima kasih dan puji syukur kepada Tuhan Yesus Kristus karena atas karunia dan penyertaan-Nya penulis dapat menyelesaikan penelitian yang berjudul “Perancangan dan Pembuatan Perangkat Lunak untuk Ujian Elektronik” di Universitas Kristen Maranatha.

Adapun penyelesaian penelitian ini tentunya tidak lepas dari bimbingan, bantuan, dan dukungan dari berbagai pihak. Oleh karena itu, peneliti ingin mengucapkan banyak terima kasih sebesar-besarnya kepada:

1. Bapak Radiant Victor Imbar selaku Dekan Fakultas Teknologi Informasi, Universitas Kristen Maranatha (FIT UK. Maranatha) yang telah memberikan kepercayaan bagi penulis untuk mengerjakan penelitian ini.
2. Bapak Erico Darmawan Handoyo selaku rekan kerja di Fakultas Teknologi Informasi yang telah memberikan banyak dukungan dan sebagai rekan untuk “brainstorming” dalam pembuatan antarmuka dari aplikasi ini.
3. Istri tercinta, Lea Sepvianty Suharso yang telah rela kehilangan banyak waktu ketika penulis menyelesaikan penelitian ini.
4. Anak tersayang, Raphael Gerald Law yang sering aku cuekin ketika membuat laporan dan penelitian ini.

Penulis juga tidak lupa mengucapkan terima kasih kepada semua pihak yang tidak dapat disebutkan satu-persatu. Semua pihak yang telah banyak membantu dan memberikan dukungan baik moril maupun lainnya sehingga penelitian ini dapat terwujud.

Penulis menyadari bahwa penelitian ini masih dapat dikembangkan lebih lanjut, oleh karena itu semua saran dan kritik yang membangun akan penulis terima dengan senang hati. Akhirnya penulis berharap agar penelitian ini bermanfaat dan dapat digunakan semaksimal mungkin untuk membangun kampus tercinta ini.

Bandung, 16 Desember 2009

Penulis

# DAFTAR ISI

LEMBAR IDENTITAS.....	ii
KATA PENGANTAR .....	iii
DAFTAR ISI .....	iv
DAFTAR GAMBAR .....	vii
DAFTAR TABEL .....	viii
INTISARI .....	ix
ABSTRACT .....	x
BAB I Pendahuluan .....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah .....	2
1.3. Tujuan Penelitian .....	2
1.4. Manfaat Penelitian.....	2
1.5. Metode Penelitian.....	3
1.6. Sistematika Penulisan .....	3
BAB II Landasan Teori .....	5
2.1. Bagaimana Membuat Pertanyaan Yang Baik.....	5
2.2. Tipe Soal Pilihan Ganda (Majemuk) .....	5
2.3. Tipe Soal Benar/Salah .....	6
2.4. Tipe Soal Isian .....	6
2.5. Diagram UML .....	7
2.6. Diagram Use Case .....	7
2.7. Diagram Class.....	8
2.8. Serialisasi Objek .....	9
2.9. Dekripsi/Enkripsi Data.....	10

2.10.	Fungsi Satu Arah dan Fungsi Hash Satu Arah .....	10
BAB III Analisis dan Perancangan .....		11
3.1.	Diagram Use Case Aplikasi QEase Author .....	11
3.2.	Diagram Use Case Aplikasi QEase Client .....	12
3.3.	Diagram Class Aplikasi QEase Author .....	12
3.4.	Diagram Class Aplikasi QEase Client.....	13
3.5.	Diagram Class QEase Library .....	13
3.6.	Desain Antarmuka Umum Window.....	14
3.7.	Desain Antarmuka QEase Author .....	15
3.8.	Desain Antarmuka QEase Client.....	15
BAB IV Implementasi Sistem .....		17
4.1.	Implementasi QEase Author .....	17
4.2.	Implementasi QEase Client .....	17
4.3.	Implementasi Sekuritas Data .....	18
BAB V Pengujian.....		19
5.1.	Jenis Pengujian.....	19
5.2.	Unit Testing QEase Shared Library .....	19
5.2.1.	Pengujian Abstract Class Problem .....	19
5.2.2.	Pengujian Class MultipleChoiceSingleAnswer .....	20
5.2.3.	Pengujian Class MultipleChoiceMultipleAnswer .....	20
5.2.4.	Pengujian Class TrueOrFalse .....	20
5.2.5.	Pengujian Class ProblemCollection .....	21
5.2.6.	Pengujian Class ProcessBusiness.....	21
5.3.	Blackbox Testing .....	21
5.3.1.	Blackbox Testing Aplikasi QEase Author.....	22
5.3.2.	Blackbox Testing Aplikasi QEase Client.....	22
BAB VI Penutup.....		24
6.1.	Simpulan .....	24

6.2. Saran .....	24
DAFTAR PUSTAKA .....	xi

## DAFTAR GAMBAR

Gambar 1. Point-of-Sales System.....	8
Gambar 2. POS Class Diagram.....	9
Gambar 3. Serialisasi dan Deserialisasi .....	9
Gambar 4. Use Case QEase Author .....	11
Gambar 5. Use Case QEase Client .....	12
Gambar 6. Diagram Class QEase Author .....	13
Gambar 7. Diagram Class QEase Client .....	13
Gambar 8. Diagram Class QEase Shared Library .....	14
Gambar 9. Desain Antarmuka Window.....	14
Gambar 10. Sketsa Antarmuka QEase Author .....	15
Gambar 11. Desain Antarmuka QEase Client.....	16
Gambar 12. Desain Antarmuka Report .....	16

## DAFTAR TABEL

Tabel 1. Unit Testing Abstract Class Problem.....	19
Tabel 2. Unit Testing Class MultipleChoiceSingleAnswer.....	20
Tabel 3. Unit Testing Class MultipleChoiceMultipleAnswer.....	20
Tabel 4. Unit Testing Class TrueOrFalse .....	20
Tabel 5. Unit Testing Class ProcessBusiness.....	21
Tabel 6. Blackbox Testing QEase Author .....	22
Tabel 7. Blackbox Testing QEase Client.....	23

## **INTISARI**

QEase adalah aplikasi yang sengaja didesain untuk kebutuhan penyelenggaraan ujian di Fakultas Teknologi Informasi. QEase terdiri dari dua aplikasi terpisah QEase Author dan QEase Client, yang keduanya mempunyai fungsi dan tujuan tersendiri.

QEase Author digunakan staf pengajar untuk menyusun soal. Sedangkan QEase Client digunakan siswa untuk mengerjakan soal (test).

Diharapkan dengan adanya kedua aplikasi ini penggunaan kertas untuk ujian dapat ditekan.

**KATA KUNCI: OOP, WPF, Test Elektronik**

## **ABSTRACT**

QEase is design to help Faculty Information Technology in managing electronic test. QEase was composed by two applications; QEase Author and QEase Client. These two applications have different goals and users.

QEase Author is supposed to be used by teachers to create or compose some problem to solved.

QEase Client is used by students to take some test.

Hopefully by creating these applications we manage to limit paper usage.

**KEYWORDS: OOP, WPF, Test Elektronik**

# **BAB I Pendahuluan**

## **1.1. Latar Belakang**

Pemanasan global merupakan masalah yang sering disinggung saat ini. Salah satu penyebab terjadinya pemanasan global adalah jumlah CO<sub>2</sub> di udara terlalu banyak sehingga tidak dapat diolah oleh tumbuhan. Industri dan kendaraan bermotor adalah penghasil CO<sub>2</sub> terbesar. Banyak usaha yang dapat dilakukan untuk mengurangi laju peningkatan CO<sub>2</sub> ini, salah satunya adalah dengan penghijauan dan pengurangan hasil industri.

Banyak perusahaan (baik besar maupun kecil) sudah memulai untuk peduli lingkungan dengan menghemat penggunaan kertas dan listrik. Dengan mengurangi konsumsi kertas tidak saja mengurangi jumlah pohon yang harus ditebang tetapi juga mengurangi pengolahan yang harus dilakukan pihak pabrik kertas, sehingga efek yang dihasilkan cukup besar.

Universitas Kristen Maranatha merupakan salah satu kampus yang sudah memulai penghematan penggunaan kertas semenjak tiga sampai empat tahun yang lalu. Salah satu cara yang dilakukan Universitas Kristen Maranatha adalah dengan menggunakan ulang kertas bekas untuk kebutuhan yang dinilai kurang penting. Pengeluaran kertas paling banyak sebenarnya terjadi ketika minggu ujian, karena pada minggu ini jutaan lembar kertas digunakan untuk mencetak soal ujian.

Fakultas Teknologi Informasi (salah satu fakultas yang bernaung di Universitas Kristen Maranatha) merasa perlu turut andil dalam memecahkan masalah ini. Salah satu caranya adalah dengan mengubah soal dalam bentuk digital, sehingga penggunaan kertas untuk soal ujian dapat ditekan (minimal). Saat ini banyak sekali metoda yang dilakukan oleh staf pengajar untuk meminimalkan penggunaan kertas, mulai dari memberikan soal elektronik (PDF) sampai dengan memberikan ujian elektronik (menggunakan software Exam).

Ujian elektronik menggunakan software tidak saja dapat membantu mengurangi kertas, tetapi juga dapat membantu staf pengajar untuk memeriksa ujian yang sifatnya objektif (contoh soal benar-salah dan soal pilihan ganda). Software yang saat ini digunakan (Exam) dinilai kurang mampu untuk memenuhi kebutuhan ujian di Fakultas Teknologi Informasi. Beberapa kekurangan dari software ini adalah keterbatasan jenis soal yang hanya berbentuk teks (soal berbentuk gambar atau multimedia tidak dapat ditampilkan), tampilan antarmuka yang kurang *user friendly*, dan cara distribusi ujian

elektronik yang merepotkan. Sebenarnya ada beberapa software sejenis yang menawarkan fitur ini, hanya saja harganya relatif mahal.

## 1.2. Rumusan Masalah

Rumusan masalah yang dapat diambil dari latar belakang di atas adalah:

1. Bagaimana membuat sebuah software yang tidak saja mudah digunakan oleh siswa tetapi juga mudah digunakan oleh staf pengajar?
2. Bagaimana cara agar soal yang diberikan dapat lebih bervariasi (tidak hanya teks tetapi juga dapat berbentuk multimedia)?
3. Bagaimana cara distribusi soal agar tidak saja distribusi soal menjadi mudah dan juga aman?

## 1.3. Tujuan Penelitian

Penelitian ini bertujuan untuk membuat sebuah alat bantu (perangkat lunak/*software*) yang memudahkan siswa dan staf pengajar. Perangkat lunak yang dibuat minimum mempunyai fitur-fitur sebagai berikut:

1. Tipe soal yang dapat dibuat (dan ditampilkan) adalah Pilihan Ganda, Pilihan Ganda Majemuk, dan Benar-Salah.
2. Soal/jawaban akan disimpan dalam bentuk terenkripsi, diharapkan dapat menyulitkan pihak yang tidak bertanggung jawab untuk merubah soal/jawaban. Selain itu hal ini dapat memudahkan distribusi soal, karena dapat disimpan di *server* terpusat tanpa harus takut data dicuri atau diubah.
3. Soal dapat berupa multimedia; teks, *audio*, *video*, dan *image*.
4. Akan dibuat dua jenis aplikasi yang mempunyai target pengguna yang berbeda. Aplikasi untuk membuat soal ujian (staf pengajar) dan aplikasi untuk mengerjakan soal ujian (siswa).

## 1.4. Manfaat Penelitian

Manfaat yang dapat dirasakan dengan terselesaikan penelitian ini adalah:

1. Staf pengajar: Memudahkan evaluasi soal tipe objektif sehingga waktu dapat digunakan untuk mengerjakan pekerjaan lain (evaluasi soal tipe essay). Selain itu juga dapat mempercepat penyerahan nilai ke Program Studi yang pada akhirnya akan meningkatkan kepuasan mahasiswa.
2. Mahasiswa: Membantu siswa untuk segera mengetahui hasil ujian (tipe objektif) sehingga mahasiswa dapat menyusun strategi dalam pengerjaan ujian tipe lainnya.
3. Program studi: Membantu mempercepat proses pengumpulan nilai, yang akan berimbas pada kepuasan mahasiswa.
4. Universitas: Mengurangi pengeluaran untuk pembelian perangkat lunak, karena perangkat lunak sengaja dirancang untuk kebutuhan lokal.

## 1.5. Metode Penelitian

Metode penelitian yang digunakan dalam pelaksanaan dan perancangan perangkat lunak ini adalah:

1. Tahap identifikasi: Pada tahap ini semua kebutuhan dari perangkat lunak dikumpulkan dan dianalisa agar perangkat lunak yang dibuat tepat guna. Sumber informasi adalah staf pengajar di Fakultas Teknologi Informasi (terutama staf pengajar yang sering menggunakan alat bantu uji elektronik).
2. Tahap desain: Pada tahap ini komponen pendukung dan antarmuka dari perangkat lunak dirancang. Diharapkan antarmuka yang dirancang dapat mudah untuk digunakan (*learning curve* yang pendek).
3. Tahap pengembangan: Pada tahap ini perangkat lunak dikembangkan dan setelah selesai akan diuji dengan menggunakan metode *whitebox*.
4. Tahap ujicoba: Pada tahap ini perangkat lunak diujicobakan di lapangan (dunia nyata) untuk mengetahui kendala-kendala yang tidak teridentifikasi sebelumnya.
5. Penyusunan laporan: Pada tahap ini laporan disusun. Laporan berisi langkah-langkah pengerjaan dari latar belakang sampai dengan kesimpulan yang didapat.

## 1.6. Sistematika Penulisan

Laporan penelitian disusun menggunakan sistematika penulisan sebagai berikut:

BAB I - Pendahuluan

Bab ini berisi uraian singkat mengenai latar belakang masalah, rumusan masalah, tujuan penelitian, manfaat, dan sistematika penulisan.

## BAB II - Landasan Teori

Bab ini berisi uraian teori dasar yang digunakan dengan penelitian. Setelah membaca bab ini diharapkan akan memudahkan dalam membaca bab-bab selanjutnya.

## BAB III - Analisis dan Perancangan

Bab ini berisi uraian tentang pemodelan system (aplikasi) yang dapat digunakan sebagai acuan dalam pembuatan perangkat lunak. Bab ini berisi diagram UML (use case diagram, activity diagram, dan static diagram) dan desain dari antarmuka.

## BAB IV - Implementasi Sistem

Bab ini berisi implementasi dari system berdasarkan rancangan yang telah dibuat sebelumnya. Bab ini berisi penjelasan dari implementasi perangkat lunak yang telah selesai dibuat.

## BAB V - Hasil dan Pengujian

Bab ini berisi *screenshot* dari perangkat lunak yang dibuat beserta pengujian yang dilakukan.

## BAB VI - Penutup

Bab ini berisi simpulan dan saran pengembangan dari system yang telah dirancang dan diimplementasikan.

## **BAB II Landasan Teori**

### **2.1. Bagaimana Membuat Pertanyaan Yang Baik**

Pertanyaan yang baik adalah pertanyaan yang tidak ambigu (dapat diartikan lebih dari satu tujuan). Selain itu tipe soal juga mempengaruhi apakah pertanyaan yang diberikan tepat sasaran. Tipe soal tertentu cocok untuk menguji kemampuan cognitive/pengertian seseorang, tipe soal lainnya cocok untuk menguji kemampuan daya ingat seseorang.

Sebelum menulis sebuah pertanyaan pertama kali yang harus dilakukan adalah memilih tipe soal yang akan diberikan. Tipe soal yang sering digunakan adalah pilihan ganda, pilihan ganda majemuk, benar/salah, isian, dan essay. Pilihlah tipe soal yang memang sesuai dengan tujuan yang ingin dicapai oleh mata kuliah (pelajaran) yang dimaksud.

Contoh bilamana ingin menguji kemampuan siswa dalam menghafal nama atau tempat tertentu, maka sebaiknya menggunakan tipe soal isian. Bilamana ingin menguji daya nalar seseorang untuk menyelesaikan suatu masalah tertentu, maka sebaiknya menggunakan tipe soal essay. Bilamana ingin menguji logika, analisa, dan pengertian umum, maka tipe soal pilihan ganda (majemuk) cocok digunakan.

Selain beberapa alasan yang telah dipaparkan yang perlu diperhatikan dalam memilih sebuah tipe soal adalah jumlah orang (siswa) yang mengikuti ujian tersebut. Bila jumlahnya cukup banyak, maka sangat tidak efektif bilamana memilih tipe ujian essay. Karena tidaklah mudah (dan membutuhkan banyak waktu) untuk memeriksa jawaban essay dalam jumlah yang besar.

### **2.2. Tipe Soal Pilihan Ganda (Majemuk)**

Tipe soal ini adalah tipe soal paling fleksibel. Tipe soal ini cocok untuk mengetahui tingkat pemahaman untuk suatu masalah tertentu. Staf pengajar akan (cukup) kesulitan dalam menyusun pertanyaan dengan tipe soal pilihan ganda, karena selain harus membuat pertanyaan, staf pengajar juga perlu untuk memikirkan pilihan jawaban yang harus ditawarkan. Berikut ini adalah beberapa saran yang dapat digunakan untuk membantu menyusun soal bertipe pilihan ganda.

1. Gunakan kalimat atau kata yang jelas (tidak ambigu).

2. Pastikan ada jawaban yang benar.
3. Berikan minimal 3 sampai dengan 5 pilihan jawaban.
4. Hindari (minimalkan) pilihan jawaban “semua pernyataan benar” atau “semua pernyataan salah”.
5. Distribusikan secara acak jawaban yang benar, hindari pola jawaban tertentu.
6. Pastikan (kumpulan) soal yang diberikan menguji pengetahuan, aplikatif, pemahaman, analisis, dan evaluasi agar dapat mengetahui tingkat penguasaan bahan yang diberikan.

### **2.3. Tipe Soal Benar/Salah**

Tipe soal ini digunakan untuk menguji daya ingat mengenai suatu fakta atau pengetahuan. Gunakan beberapa saran berikut untuk membantu menyusun soal bertipe benar/salah.

1. Pastikan ada jawaban yang benar (tidak membingungkan dan tidak dapat diartikan lain).
2. Hindari kalimat yang sifatnya negasi contoh “Patimura TIDAK lahir di ...” sebaiknya dituliskan menjadi “Patimura lahir di ...”.
3. Distribusikan secara acak antara jawaban benar dan salah sehingga tidak didapat pola-pola tertentu.
4. Buat soal dengan jawaban salah dibandingkan benar, karena hal ini menuntut tingkat cognitive (yang lebih tinggi) dari siswa.

### **2.4. Tipe Soal Isian**

Tipe soal ini mengharuskan untuk mengetahui jawaban yang tepat tanpa dapat memilih dari opsi yang ada. Berikut ini adalah beberapa saran yang dapat membantu menyusun soal bertipe isian.

1. Pastikan hanya ada satu jawaban pasti untuk menghindari kebingungan ketika mengisi atau mengoreksi.
2. Bagian yang harus dilengkapi (diisi) sedapat mungkin disimpan di akhir kalimat atau mendekati akhir kalimat.
3. Pertanyaan sebaiknya merupakan informasi penting yang harus diingat.

## 2.5. Diagram UML

Unified Modeling Language (UML) adalah “bahasa” yang digunakan untuk memodelkan sebuah sistem. Bahasa ini dibuat dalam bentuk kumpulan diagram agar lebih bersifat universal. Dari banyak UML Diagram yang ada UML dapat dibagi menjadi dua kelompok besar yaitu:

1. *Structural diagram*: Diagram jenis ini digunakan untuk menggambarkan struktur dari sistem yang akan dibuat. Ada dua buah jenis diagram yang digunakan untuk menggambarkan struktur dari sistem, diagram tersebut adalah *class diagram* dan *deployment diagram*.
2. *Behavioral diagram*: Diagram jenis ini digunakan untuk menggambarkan proses alur *component*, *class*, *object*, dan sistem. Ada lima jenis diagram yang termasuk dalam kategori ini, yaitu *use case diagram*, *activity diagram*, *sequence diagram*, *collaboration diagram*, dan *statechart diagram*.

Dari banyak diagram yang ada, dua diagram yang paling sering digunakan. Diagram tersebut adalah *class diagram* dan *use case diagram*.

## 2.6. Diagram Use Case

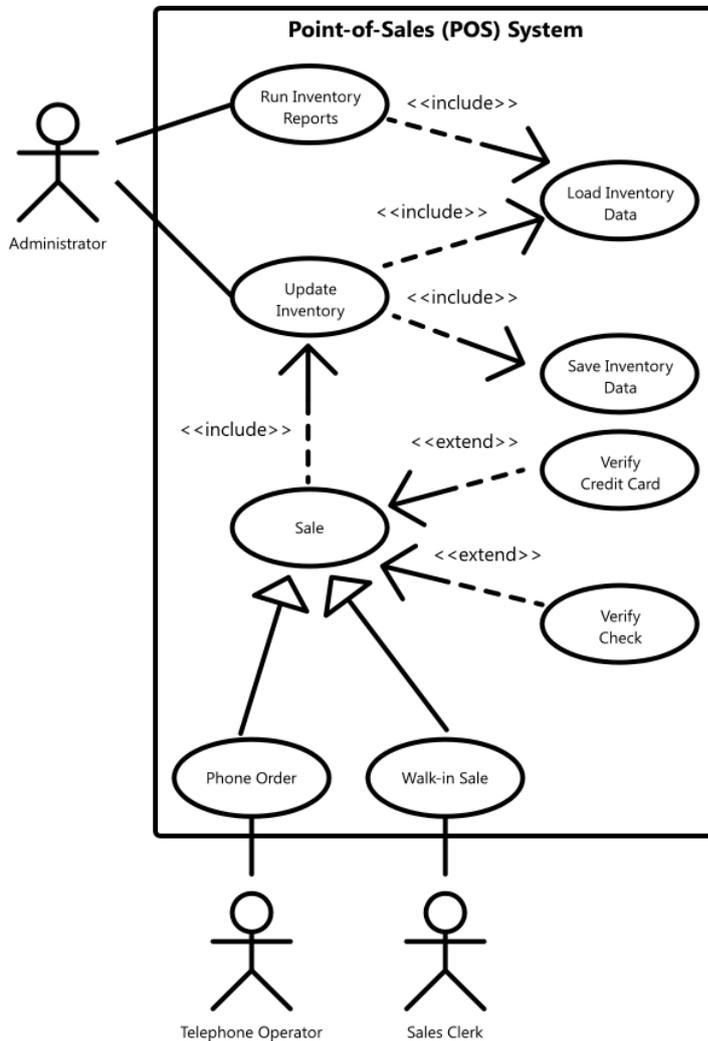
*Use case diagram* adalah diagram yang digunakan untuk menggambarkan kemampuan dari sebuah sistem. Diagram ini digambar dari cara pandang *user* menggunakan sistem. Dengan kata lain diagram ini digunakan untuk menggambarkan scenario yang mungkin dilakukan *user* ke dalam sistem.

Komponen pembentuk *use case diagram* terdiri dari empat bagian:

1. *Actor*: *Actor* merupakan pengguna dari sebuah sistem. *Actor* dapat berupa manusia atau sistem otomatis lainnya. *Actor* digambarkan dengan *stick figure*.
2. *Use case*: *Use case* merupakan aksi yang dapat dilakukan oleh sistem.
3. *System boundary*: Digunakan untuk memilah-milah sistem. Sering kali digambarkan bilamana sistem yang dibuat cukup besar.
4. *Relation*: *Relation* digunakan untuk menghubungkan relasi antara *actor* ke *actor*, *actor* ke *use case*, atau *use case* ke *use case* lainnya. Jenis relasi yang ada dalam diagram ini adalah:
  - a. *Generalization*: Menggambarkan bahwa *use case* atau *actor* yang ditunjuk lebih umum, sedangkan *use case* atau *actor* lainnya mempunyai sifat turunan dari yang ditunjuk.

- b. *Include*: Menggambarkan bahwa sebuah *use case* yang ditunjuk wajib (harus) dikerjakan terlebih dahulu agar *use case* utama dapat berjalan dengan baik.
- c. *Extend*: Menggambarkan bahwa sebuah *use case*  mungkin dijalankan ketika *use case* yang ditunjuk berjalan.

Diagram berikut (gambar 3) adalah sebuah contoh *use case diagram* dari sebuah sistem penjualan (*point-of-sale/POS*).



**Penjelasan dari diagram di samping:**

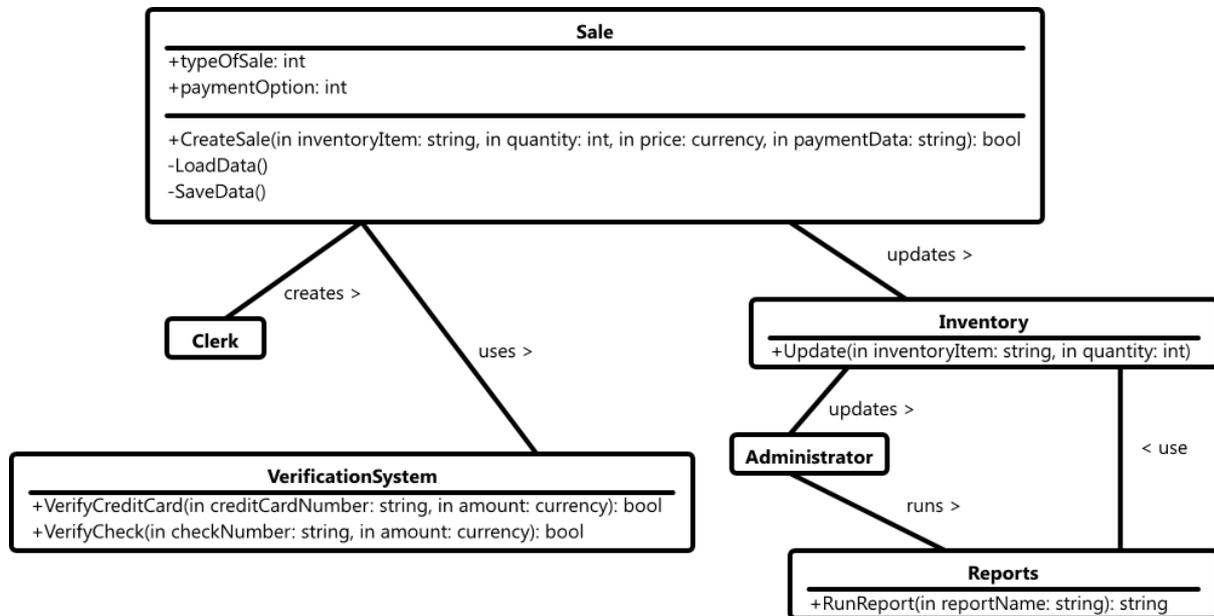
1. System memperbolehkan administrator untuk menjalankan inventory report dengan cara mengambil data dari disk.
2. Administrator dapat mengubah inventory dengan cara mengambil dan menyimpannya kembali dalam disk.
3. Sales clerk mencatat walk-in-sales.
4. Operator telepon adalah sejenis sales clerk yang mencatat pembelian via telepon.
5. Apapun jenis penjualannya, harus mengubah isi dari inventory.
6. Penjual harus memeriksa validitas credit card.
7. Penjual harus memeriksa validitas check.

**Gambar 1. Point-of-Sales System**

**2.7. Diagram Class**

*Class diagram* adalah diagram yang digunakan untuk menggambarkan struktur dari class pembentuk system. *Class diagram* merupakan hasil pemodelan yang didapat dari proses analisa dan desain. Biasanya pemodelan dimulai dari class diagram yang cukup dapat dimengerti oleh banyak orang

(tidak hanya *analyst/developer*), selanjutnya model ini diiterasi ulang sehingga mungkin dibuat ke dalam sistem. Dua model ini dikenal dengan *high-level model* (semua orang dapat mengerti), dan *low-level model* (untuk *analyst dan developer*). Contoh diagram berikut ini (gambar 4) menggambarkan class diagram dari sistem penjualan (*POS system*) sebelumnya.

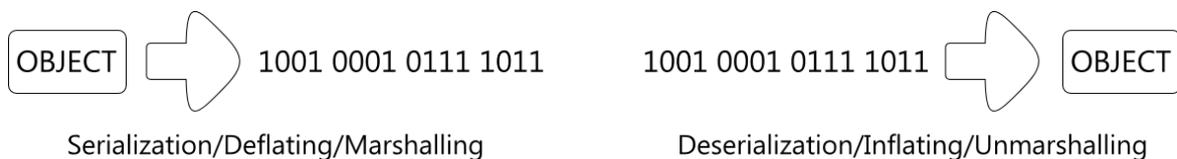


Gambar 2. POS Class Diagram

## 2.8. Serialisasi Objek

Serialisasi (*serialization*) objek adalah sebuah proses konversi objek menjadi urutan bit sehingga dapat disimpan dalam media penyimpanan (dalam sebuah file atau buffer memory) atau dikirimkan via network (jaringan) untuk selanjutnya objek tersebut dapat dibentuk ulang di komputer yang berbeda/sama.

Proses serialisasi ini dikenal dengan istilah lain *deflating/marshalling*, sedangkan proses sebaliknya (merubah urutan bit menjadi bentuk objek), dikenal dengan istilah deserialisasi (*deserialization*), *inflating*, atau *unmarshalling*. Ilustrasi berikut menggambarkan kedua proses ini.



Gambar 3. Serialisasi dan Deserialisasi

## 2.9. Dekripsi/Enkripsi Data

Enkripsi (encryption) adalah sebuah proses transformasi informasi menggunakan sebuah algoritma tertentu sehingga didapat sebuah informasi terenkripsi. Informasi awal (sebelum proses enkripsi) dikenal dengan istilah *plaintext*. Informasi terenkripsi dikenal dengan istilah *ciphertext*. Sedangkan algoritma enkripsi dikenal dengan istilah *cipher*. Proses dekripsi (decryption) merupakan proses kebalikan dari enkripsi, transformasi informasi terenkripsi menggunakan algoritma tertentu untuk mendapat informasi sebenarnya.

Kedua proses ini awalnya digunakan oleh pemerintah atau militer untuk mengirimkan informasi dari satu tempat ke tempat lain. Proses ini perlu dilakukan agar informasi tidak “bocor” ke pihak ke tiga. Saat ini kedua proses ini sudah digunakan umum, transaksi elektronik merupakan salah satu contoh yang menggunakan proses ini.

Proses dekripsi/enkripsi sudah berumur cukup lama. Julius Caesar (kerajaan Romawi) sudah menggunakannya untuk mengirimkan surat rahasia ke jendral-jendral nya. Algoritma yang digunakan oleh Julius Caesar dikenal dengan nama *Caesar Cipher*.

## 2.10. Fungsi Satu Arah dan Fungsi Hash Satu Arah

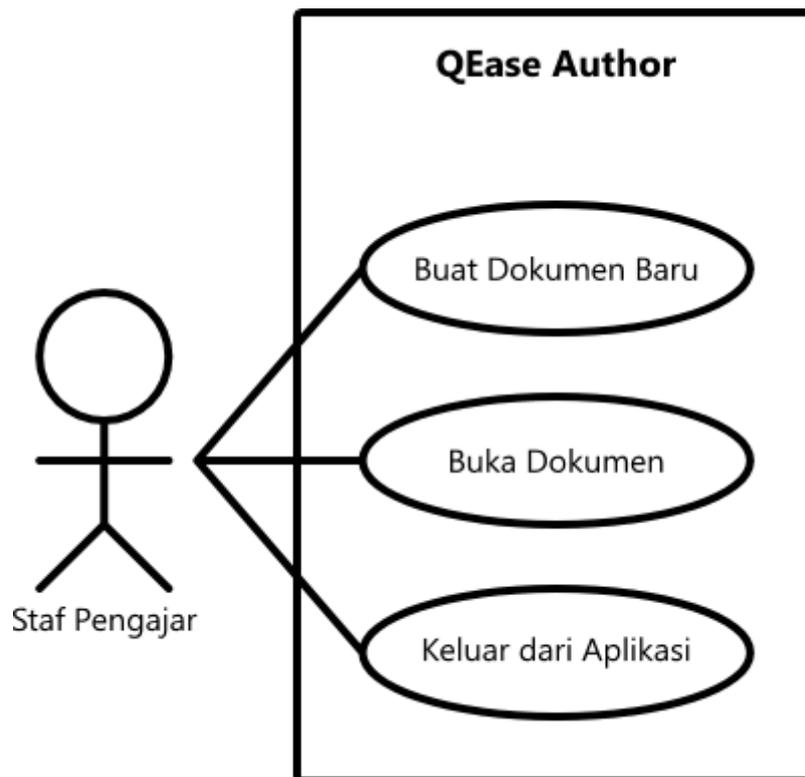
Fungsi satu arah adalah sebuah fungsi yang bilamana diberi input akan mendapatkan output dengan mudah, tetapi sulit dihitung hasil inverse-nya (ada output ingin mendapat nilai input). Fungsi satu arah ini merupakan dasar yang digunakan untuk mendapatkan fungsi hash satu arah.

Fungsi hash satu arah merupakan pengembangan dari fungsi satu arah perbedaannya fungsi hash satu arah merupakan fungsi satu-satu (satu input akan menghasilkan hanya satu output). Fungsi ini digunakan untuk mengetahui apakah sebuah informasi yang diterima valid (belum berubah). Fungsi ini juga dapat digunakan untuk menyimpan kata kunci (*password*).

## BAB III Analisis dan Perancangan

### 3.1. Diagram Use Case Aplikasi QEase Author

Aplikasi *QEase Author* adalah aplikasi yang digunakan oleh staf pengajar dalam menyusun soal ujian. Hasil dari aplikasi ini adalah sebuah *QEase document*. Berikut ini adalah diagram use case dari aplikasi *QEase Author*.



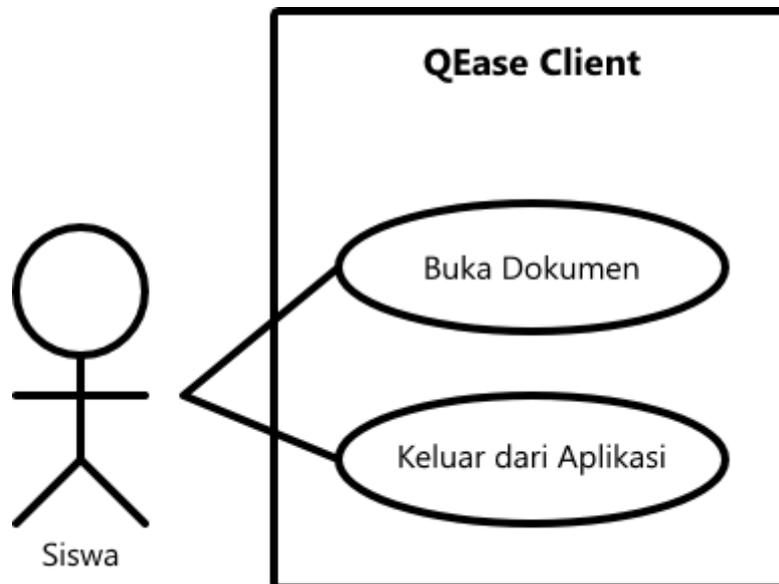
Gambar 4. Use Case QEase Author

Dari gambar 4 dapat dilihat bahwa (kelak) aplikasi pengguna dapat membuat dokumen baru, membuka dokumen (yang telah dibuat sebelumnya), dan keluar dari aplikasi. Ketika membuat dokumen baru dan membuka dokumen pengguna akan diminta untuk memasukkan kata kunci. Hal ini perlu dilakukan agar pengguna yang tidak berhak, tidak dapat mengubah isi dari sebuah dokumen. Sedangkan ketika pengguna keluar dari aplikasi, aplikasi diharapkan untuk menampilkan layar konfirmasi. Hal ini perlu dilakukan agar kesalahan pemilihan menu dapat diminimalisir dan dokumen yang telah disusun tidak hilang.

### 3.2. Diagram Use Case Aplikasi QEase Client

Aplikasi *QEase Client* adalah aplikasi yang digunakan oleh siswa ketika mengambil ujian. Aplikasi ini dapat membuka *QEase document* yang telah disusun sebelumnya menggunakan *QEase Author*.

Berikut ini adalah diagram use case dari aplikasi *QEase Client*.

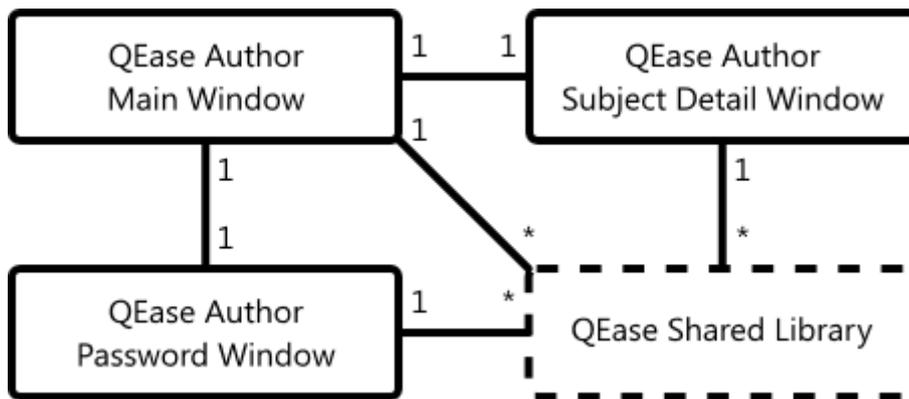


Gambar 5. Use Case QEase Client

*QEase Client* mempunyai use case yang jauh lebih sederhana daripada *QEase Author* (gambar 5). *QEase Client* hanya perlu membuka berkas dan menampilkan soal. Pengguna diharapkan untuk mengisi soal yang diberikan. Setelah selesai maka aplikasi yang menampilkan sebuah laporan (report) nilai mentah dan persentasi nilai keseluruhan. Laporan ini kemuidan dapat disimpan sebagai bukti telah menempuh ujian.

### 3.3. Diagram Class Aplikasi QEase Author

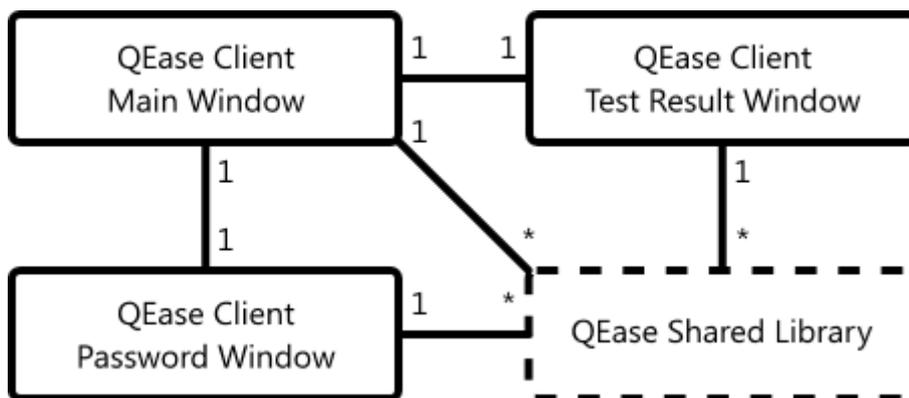
Aplikasi *QEase Author* mempunyai class diagram yang cukup sederhana, karena hampir sebagian besar (90%) class didefinisikan dalam pustaka (library) terpisah. Class diagram yang ada dalam pustaka akan dibahas terpisah (pada sub-bab selanjutnya).



Gambar 6. Diagram Class QEase Author

### 3.4. Diagram Class Aplikasi QEase Client

Aplikasi QEase Client mempunyai class diagram yang cukup mirip dengan QEase Author. Perbedaan paling mencolok adalah beberapa fungsionalitas yang ada pada QEase Author tidak tersedia pada QEase Client. Hal ini dikarenakan QEase Client hanya bertujuan untuk menampilkan dokumen yang dibuat dan meminta pengguna untuk mengisi jawaban pada tempat yang disediakan. Sama seperti QEase Author sebagian besar class menggunakan pustaka terpisah, sehingga membuat class diagram pada QEase Client sederhana.

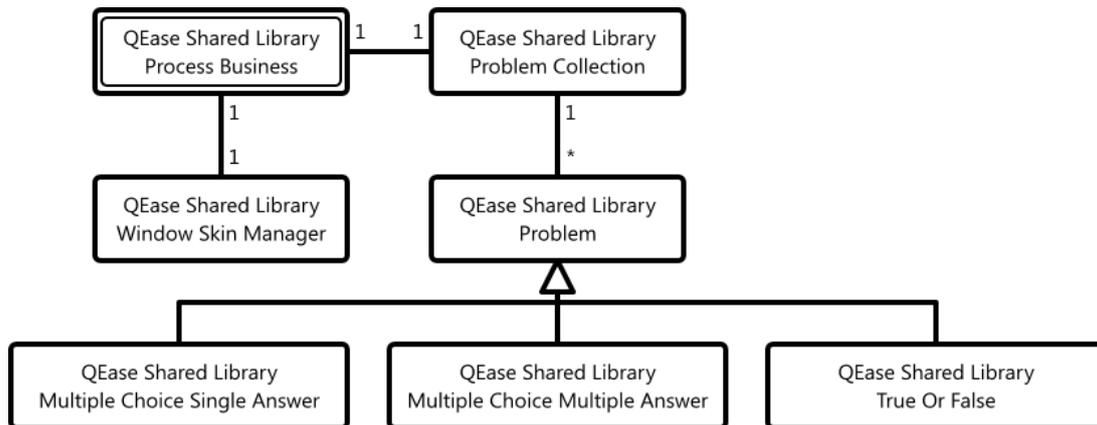


Gambar 7. Diagram Class QEase Client

### 3.5. Diagram Class QEase Library

QEase Library adalah pustaka class yang dibuat terpisah dari aplikasi utama. Hal ini dirasa perlu dilakukan karena class dan fungsionalitas yang digunakan oleh dua aplikasi utama adalah sama. Dengan memisahkan pustaka class diharapkan pemeliharaan dan penyempurnaan aplikasi lebih

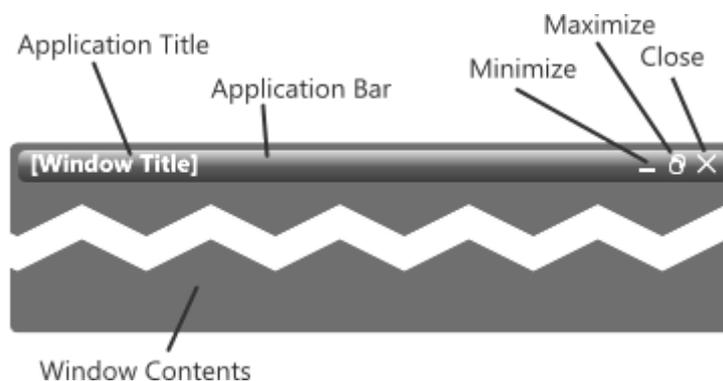
mudah dilakukan. Hal ini dikarenakan “engine” dari kedua aplikasi dikumpulkan dalam satu pustaka, bilamana pustaka ini diubah akan otomatis berimbas pada kedua aplikasi yang menggunakan.



Gambar 8. Diagram Class QEase Shared Library

### 3.6. Desain Antarmuka Umum Window

Window dari aplikasi sengaja diubah untuk memberikan kesan “cool” dan futuristik. Desain window dibuat sedemikian rupa agar tidak terlalu menyimpang dengan desain window pada umumnya. Pendekatan ini perlu dilakukan karena aplikasi tidak hanya digunakan oleh siswa (yang sifatnya lebih dinamis), tetapi juga digunakan oleh staf pengajar. Berikut ini adalah sketsa kasar dari desain window yang akan dibuat.

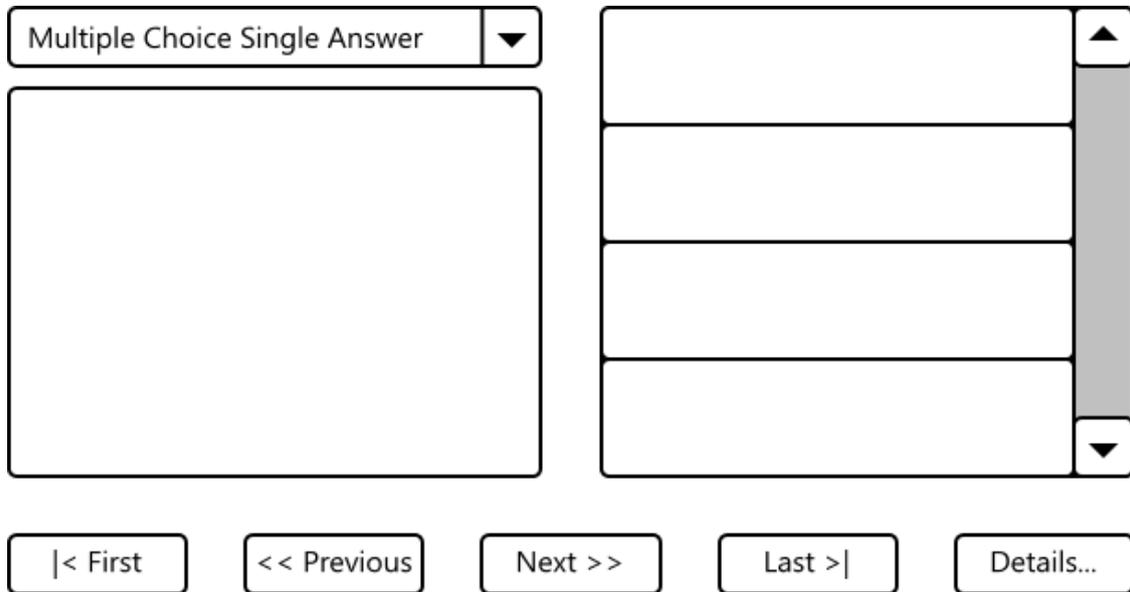


Gambar 9. Desain Antarmuka Window

Isi dari window akan ditempatkan pada bagian “Window Contents”. Window contents akan dibuat semi-transparan, sehingga terlihat efek kaca hitam. Application Bar pada tiap jenis aplikasi mempunyai warna gradasi yang berbeda, sehingga pengguna dapat langsung mengetahui jenis aplikasi tanpa harus membaca judul dari aplikasi. Aplikasi QEase Author akan menggunakan warna gradasi merah, sedangkan QEase Client akan menggunakan warna gradasi biru.

### 3.7. Desain Antarmuka QEase Author

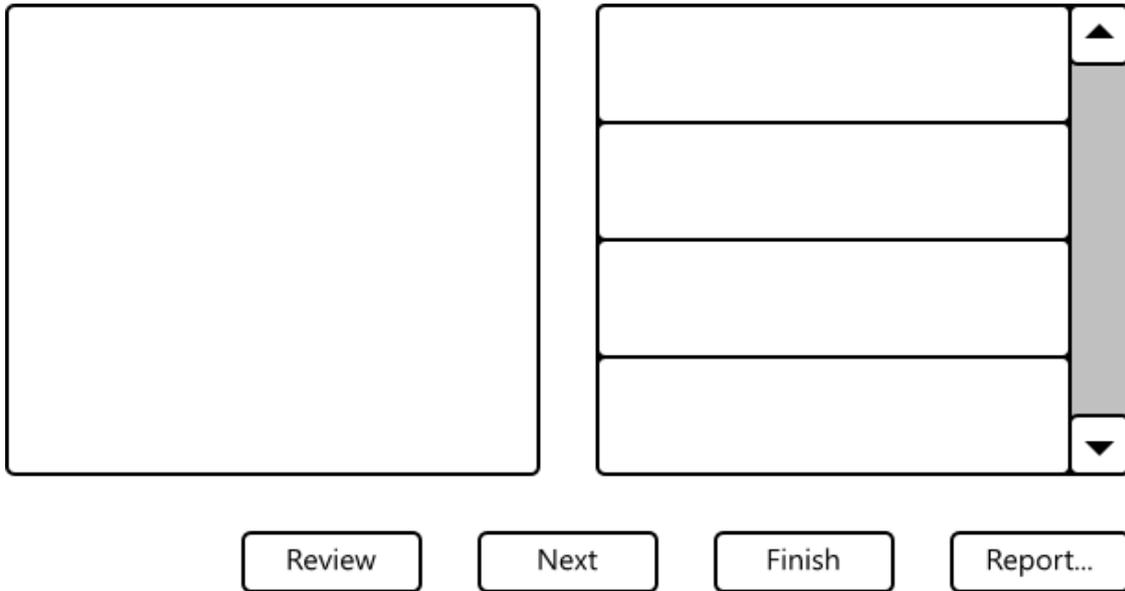
Aplikasi QEase Author adalah aplikasi yang digunakan oleh staf pengajar untuk membuat (menyusun) kumpulan soal. Aplikasi dibuat sederhana agar mudah digunakan. Pengguna dapat memilih tipe soal yang akan dibuat. Tampilan antarmuka akan berubah bergantung dari tipe soal yang dipilih. Berikut ini adalah sketsa dari tampilan antarmuka yang akan dibuat.



Gambar 10. Sketsa Antarmuka QEase Author

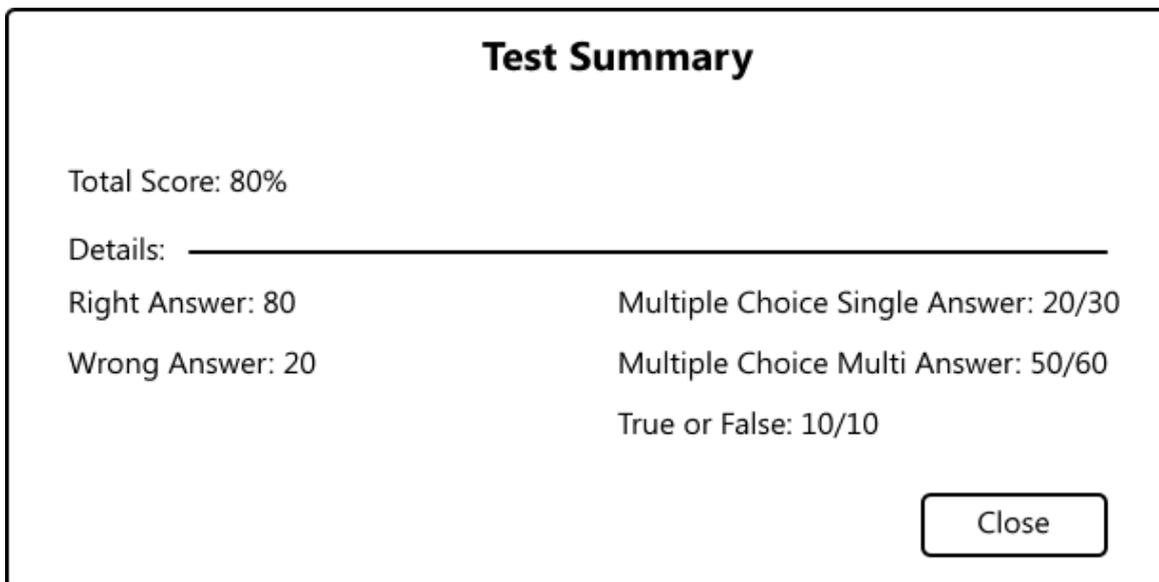
### 3.8. Desain Antarmuka QEase Client

Aplikasi QEase Client adalah aplikasi yang digunakan oleh siswa untuk mengerjakan soal yang sebelumnya telah dibuat oleh staf pengajar. Aplikasi mempunyai tampilan yang kurang lebih serupa dengan aplikasi QEase Author, perbedaannya pengguna hanya dapat memilih/mengisi jawaban, tanpa dapat mengubah soal dan pilihan yang ada. Berikut ini adalah sketsa dari tampilan antarmuka yang akan dibuat.



Gambar 11. Desain Antarmuka QEase Client

Setelah selesai mengerjakan soal aplikasi akan menampilkan laporan (*report*) keberhasilan pengguna tersebut. Desain laporan yang akan ditampilkan pada layar pengguna adalah sebagai berikut.



Gambar 12. Desain Antarmuka Report

## **BAB IV Implementasi Sistem**

### **4.1. Implementasi QEase Author**

Implementasi QEase Author tidak berbeda dengan yang telah direncanakan sebelumnya (Bab III). Hampir semua fitur yang diminta telah diimplementasikan. Berikut ini adalah fitur yang direncanakan dan telah diimplementasikan.

1. Terdapat 3 jenis tipe soal objektif: Pilihan Ganda (Multiple Choice Single Answer), Pilihan Ganda Majemuk (Multiple Choice Multiple Answer), dan Benar/Salah (True or False).
2. Terdapat mekanisme agar soal tidak dapat diubah oleh pihak yang tidak berhak (menggunakan password dan enkripsi).
3. Terdapat navigasi yang memudahkan pindah dari satu soal ke soal lainnya untuk proses editing.
4. Soal otomatis disimpan pada periode tertentu (kurang lebih 3 menit) untuk menghindari kehilangan data.

Terdapat satu fitur yang tidak diimplementasikan, yaitu menyimpan soal dalam bentuk XML. Hal ini tidak diimplementasikan karena menimbang ukuran teks yang jauh lebih besar dari pada bentuk biner. Selain itu dikarenakan adanya fitur enkripsi yang pada akhirnya membuat teks tidak dapat dibaca.

### **4.2. Implementasi QEase Client**

Implementasi QEase Client tidak berbeda jauh dari implementasi awal. Hanya saja QEase Client ternyata harus dapat menyimpan dokumen yang berbeda dari QEase Author. QEase Client harus mampu menyimpan semua jawaban dari pengguna dalam sebuah file yang pada akhirnya file ini dapat digunakan untuk memeriksa jawaban dan menghitung nilai dari pengguna tersebut.

Fitur lainnya tidak berubah dari perencanaan awal. Berikut ini adalah fitur dari aplikasi QEase Client setelah diperbaharui.

1. Terdapat 3 jenis tipe soal yang dapat ditampilkan: Pilihan Ganda, Pilihan Ganda Majemuk, dan Benar/Salah.

2. Pengguna wajib memasukan kata kunci (password) terlebih dahulu sebelum dapat mengerjakan soal yang diberikan. Kata kunci yang diminta berbeda dengan kata kunci untuk mengubah soal.
3. Terdapat beberapa bantuan navigasi yang bertujuan untuk memudahkan pengguna dalam mengerjakan soal yang diberikan.
4. Terdapat fitur untuk melihat laporan pencapaian yang dapat digunakan sebagai rambu dalam pengerjaan soal.
5. Jawaban otomatis disimpan dalam sebuah file terpisah yang dapat digunakan untuk melihat laporan nilai siswa. File inipun akan disimpan dalam keadaan terenkripsi untuk mencegah adanya perubahan data ke dalam file ini.

### **4.3. Implementasi Sekuritas Data**

Sekuritas dibuat dengan menggunakan logika yang mirip dengan aplikasi PGP (Pretty Good Privacy) yaitu dengan membuat sebuah enkapsulasi pada data yang akan disimpan. Cara penyimpanan datanya adalah seperti berikut.

1. Kumpulan soal (Problem Collection) diubah ke dalam bentuk array byte.
2. Array ini dihitung nilai hash-nya untuk memastikan validitas data.
3. Array dan nilai hash digabungkan menjadi satu kemudian dienkripsi dengan menggunakan algoritma tertentu dengan kata kunci yang sebelumnya telah diminta oleh aplikasi.
4. Hasilnya disimpan kedalam media penyimpanan.

Sedangkan cara untuk membaca kembali data yang telah disimpan adalah dengan membalik proses yang ada.

1. Ambil data dari media penyimpanan.
2. Dekripsi data dengan menggunakan kata kunci yang telah diminta sebelumnya.
3. Hitung hash dari data yang ada, kemudian bandingkan dengan hash yang disimpan. Bilamana sama maka data yang disimpan adalah valid.
4. Lakukan konversi data dari bentuk array byte menjadi kumpulan soal untuk selanjutnya diolah dan ditampilkan oleh aplikasi.

## BAB V Pengujian

### 5.1. Jenis Pengujian

Aplikasi yang baik adalah aplikasi yang lolos uji. Pengujian aplikasi pada umumnya dapat digolongkan menjadi dua jenis, yaitu *whitebox* dan *blackbox testing*. *Whitebox testing* adalah pengujian yang dilakukan oleh pembuat program untuk menjamin tidak ada *error* dari sisi algoritmik. Sedangkan *blackbox testing* adalah pengujian yang biasanya dilakukan oleh pengguna aplikasi untuk menjamin aplikasi relatif mudah digunakan dan tidak ada *error* dari sisi antarmuka.

Kedua pendekatan tersebut dilakukan pada pengujian aplikasi QEase Author dan Client dengan harapan tingkat *error* dapat ditekan seminim mungkin. Pengujian metode *whitebox* menggunakan pendekatan *unit testing* pengujian perbagian (unit), contoh *class/object*, *procedure*, *function*, dan *method*. Sedangkan pengujian metode *blackbox* menggunakan daftar antarmuka *checklist*.

### 5.2. Unit Testing QEase Shared Library

QEase Shared Library adalah pustaka yang dibuat untuk memenuhi kebutuhan dari dua buah aplikasi QEase Author dan QEase Client. Pustaka ini mempunyai kumpulan class yang cukup banyak dan rumit karena digunakan oleh dua jenis aplikasi tersebut.

#### 5.2.1. Pengujian Abstract Class Problem

Class ini merupakan class bapak dari semua jenis soal yang ditawarkan. Class ini hanya berisi sebuah property *Text* bertipe string dan dua buah constructor (default dan satu parameter).

Tabel 1. Unit Testing Abstract Class Problem

Method/Property	Argumen/Parameter	Hasil Harapan	Hasil Pengujian
Text	"Hello"	"Hello"	"Hello"
Text	NULL	""	""
Problem()	-	Text = ""	Text = ""
Problem(text)	"Hello"	Text = "Hello"	Text = "Hello"
Problem(text)	NULL	Text = ""	Text = ""

### 5.2.2. Pengujian Class MultipleChoiceSingleAnswer

Class ini merupakan class anak dari class Problem. Class ini digunakan untuk memodelkan soal pilihan ganda. Class ini menambah dua buah property, yaitu PossibleAnswers dan RightAnswerIndex.

Tabel 2. Unit Testing Class MultipleChoiceSingleAnswer

Method/Property	Argumen/Parameter	Hasil Harapan	Hasil Pengujian
PossibleAnswers	"A", "B", "C", "D", "E"	[0] = "A", ..., [4] = "E"	[0] = "A", ..., [4] = "E"
RightAnswerIndex	0	0	0
RightAnswerIndex	8 (> PossibleAnswers)	5	5
RightAnswerIndex	-4 (< PossibleAnswers)	0	0

### 5.2.3. Pengujian Class MultipleChoiceMultipleAnswer

Class ini merupakan class anak dari class Problem. Class ini digunakan untuk memodelkan soal pilihan ganda majemuk. Class ini menambah dua buah property, yaitu PossibleAnswers dan RightAnswerIndices.

Tabel 3. Unit Testing Class MultipleChoiceMultipleAnswer

Method/Property	Argumen/Parameter	Hasil Harapan	Hasil Pengujian
PossibleAnswers	"A", "B", "C", "D", "E"	[0] = "A", ..., [4] = "E"	[0] = "A", ..., [4] = "E"
RightAnswerIndices	0, 1, 2, 3	0, 1, 2, 3	0, 1, 2, 3
RightAnswerIndices	0, 1, 8, 5	0, 1, 4	0, 1, 4
RightAnswerIndices	-5, 1, 3	0, 1, 3	0, 1, 3

### 5.2.4. Pengujian Class TrueOrFalse

Class ini merupakan class anak dari class Problem. Class ini digunakan untuk memodelkan soal pilihan benar/salah. Class ini menambah dua buah property, yaitu PossibleAnswers (hanya ada true atau false tidak dapat diubah) dan RightAnswerIndex.

Tabel 4. Unit Testing Class TrueOrFalse

Method/Property	Argumen/Parameter	Hasil Harapan	Hasil Pengujian
PossibleAnswers	"Benar", "Salah"	"True", "False"	"True", "False"
RightAnswerIndex	0	0	0
RightAnswerIndex	-1	0	0

Method/Property	Argumen/Parameter	Hasil Harapan	Hasil Pengujian
RightAnswerIndex	3	1	1

### 5.2.5. Pengujian Class ProblemCollection

Class ini merupakan turunan dari class List<Problem>. Class ini digunakan untuk memodelkan kumpulan soal. Tidak ada method/property baru yang ditambahkan, sehingga tidak perlu dilakukan pengujian.

### 5.2.6. Pengujian Class ProcessBusiness

Class ini adalah class statis yang isinya adalah proses bisnis yang dapat dilakukan oleh aplikasi. Hampir sebagian besar proses bisnis yang ada berhubungan erat dengan manajemen data (soal). Berikut ini adalah proses bisnis utama yang ada pada class ProcessBusiness.

Tabel 5. Unit Testing Class ProcessBusiness

Method/Property	Argumen/Parameter	Hasil Harapan	Hasil Pengujian
Open(path)	"C:\..." (exists)	hasil ≠ NULL	hasil ≠ NULL
Open(path)	"C:\..." (not exists)	hasil = NULL	hasil = NULL
Save(path, data)	"C:\..." (exists), not NULL	file ditimpa dengan data baru, hasil = true	file ditimpa dengan data baru, hasil = true
Save(path, data)	"C:\..." (not exists), not NULL	buat file baru, isi dengan data, hasil = true	buat file baru, isi dengan data, hasil = true
Save(path, data)	"C:\..." (exists), NULL	tidak terjadi apapun, hasil = false	tidak terjadi apapun, hasil = false
Save(path, data)	"C:\..." (not exists), NULL	tidak terjadi apapun, hasil = false	tidak terjadi apapun, hasil = false

## 5.3. Blackbox Testing

Aplikasi QEase Author dan QEase Client keduanya menggunakan pustaka QEase Shared Library untuk menjalankan proses bisnisnya. Pada kedua aplikasi utama kode program yang ada kesemuanya hanya digunakan untuk mengatur tampilan dari antarmuka, sehingga jauh lebih tepat jika keduanya menggunakan pengujian dengan metode blackbox.

### 5.3.1. Blackbox Testing Aplikasi QEase Author

QEase Author adalah aplikasi yang digunakan oleh staf pengajar dalam membuat soal yang dapat disajikan lewat aplikasi QEase Client. Berikut ini adalah hasil pengujian blackbox yang dilakukan pada aplikasi QEase Author.

Tabel 6. Blackbox Testing QEase Author

Window/Screen	Keterangan	Hasil (Y/T)
MainWindow	Window aplikasi akan ditutup ketika menekan button "x" di ujung kanan atas.	Y
MainWindow	Ukuran layar aplikasi akan tampil maksimum/kembali ke ukuran asal ketika menekan button "[]" di ujung kanan atas.	Y
MainWindow	Window aplikasi akan disembunyikan (hanya tampil pada taskbar saja).	Y
PasswordWindow	Bilamana tidak diisi sama sekali, maka button OK akan dalam keadaan disabled.	Y
PasswordWindow	Bilamana isi Password dan ConfirmPassword tidak sesuai, maka ketika button OK diklik akan menampilkan pesan kesalahan.	Y
PasswordWindow	Bilamana isi Password dan ConfirmPassword sesuai maka setelah menekan button OK pengguna dapat mulai mengubah soal.	Y
EditWindow	Isi window akan berubah tergantung dari tipe soal yang disajikan.	Y
EditWindow	Ketika hanya ada satu soal, maka button navigasi akan dalam keadaan disabled.	Y

### 5.3.2. Blackbox Testing Aplikasi QEase Client

QEase Client adalah aplikasi yang digunakan oleh siswa untuk mengerjakan soal yang telah dibuat menggunakan aplikasi QEase Author. Berikut ini adalah hasil pengujian blackbox pada aplikasi QEase Client.

Tabel 7. Blackbox Testing QEase Client

Window/Screen	Keterangan	Hasil (Y/T)
<b>MainWindow</b>	Window aplikasi akan ditutup ketika menekan button “x” di ujung kanan atas.	Y
<b>MainWindow</b>	Ukuran layar aplikasi akan tampil maksimum/kembali ke ukuran asal ketika menekan button “[ ]” di ujung kanan atas.	Y
<b>MainWindow</b>	Window aplikasi akan disembunyikan (hanya tampil pada taskbar saja).	Y
<b>PasswordWindow</b>	Bilamana tidak diisi sama sekali, maka button OK akan dalam keadaan disabled.	Y
<b>PasswordWindow</b>	Bilamana isi Password dan ConfirmPassword tidak sesuai, maka ketika button OK diklik akan menampilkan pesan kesalahan.	Y
<b>PasswordWindow</b>	Bilamana isi Password dan ConfirmPassword sesuai maka setelah menekan button OK pengguna dapat mulai mengerjakan soal.	Y

## **BAB VI Penutup**

### **6.1. Simpulan**

Dari penelitian yang telah dilaksanakan, maka dapat diambil beberapa simpulan sebagai berikut.

1. Menjamin data tidak dapat diubah oleh pihak yang tidak berhak perlu dilakukan. Hal ini dapat dicapai dengan menggunakan algoritma enkripsi yang telah tersedia.
2. Melakukan enkripsi data tidak dapat dilakukan langsung ke dalam objek bersangkutan, hal ini dikarenakan hampir sebagian besar algoritma enkripsi/dekripsi membutuhkan data bertipe array byte. Maka dari itu objek harus ditransformasi terlebih dahulu menjadi bentuk array byte.
3. Agar password tidak dapat diketahui oleh siapapun juga maka diperlukan algoritma one-way hash function. Hal ini menjamin password tidak dapat terbaca sama sekali, karena one-way hash function tidak bersifat reversible.

### **6.2. Saran**

Bilamana ingin dikembangkan lebih lanjut, alangkah lebih baik bilamana aplikasi ini dibuat versi web-nya. Sehingga siswa yang berhalangan hadir di tempat ujian karena sakit, dapat tetap mengikuti ujian dari tempat lain.

## DAFTAR PUSTAKA

1. MacDonald, Matthew (2008). **Pro WPF in C# 2008: Windows Presentation Foundation with .NET 3.5, Second Edition**. New York: APRESS.
2. Andrade, Chris (2007). **Professional WPF Programming: .NET Development with the Windows® Presentation Foundation**. Canada: Willey Publishing.
3. Eisenberg, Rob (2009). **Sams Teach Yourself WPF in 24 Hours**. Sams Publishing.
4. Nathan, Adam (2007). **Windows Presentation Foundation Unleashed**. Sams Publishing.
5. Sells, Chris (2007). **Programming WPF, Second Edition**. O'Reilly Media.
6. Shoemaker, Martin L (2004). **UML Applied: A .NET Perspective**. New York: APRESS.
7. Pitone, Dan (2005). **UML 2.0 in a Nutshell**. O'Reilly Media.
8. Januszewski, Karsten (2007). **The New Iteration**. Microsoft.