

**PENGEMBANGAN SISTEM UJIAN ONLINE  
BERBASIS OBJEK**

**TUGAS AKHIR**

Oleh :

**Lies Handayani      3310801090**

**Ferri Martin         3310801110**

Disusun untuk memenuhi syarat kelulusan Program Diploma III



**PROGRAM STUDI TEKNIK INFORMATIKA  
POLITEKNIK NEGERI BATAM  
BATAM  
2011**

# **LEMBAR PENGESAHAN**

Batam, 21 Februari 2011

**Pembimbing,**

**Riwinoto, M. Kom**

**NIK. 103025**

## LEMBAR PERNYATAAN

Dengan ini, saya:

NIM : 3310801090

Nama : Lies Handayani

adalah mahasiswa Teknik Informatika Politeknik Negeri Batam yang menyatakan bahwa tugas akhir dengan judul:

PENGEMBANGAN SISTEM UJIAN *ONLINE* BERBASIS OBJEK

disusun dengan:

1. tidak melakukan plagiat terhadap naskah karya orang lain
2. tidak melakukan pemalsuan data
3. tidak menggunakan karya orang lain tanpa menyebut sumber asli atau tanpa ijin pemilik

Jika kemudian terbukti terjadi pelanggaran terhadap pernyataan di atas, maka saya bersedia menerima sanksi apapun termasuk pencabutan gelar akademik.

Lembar pernyataan ini juga memberikan hak kepada Politeknik Negeri Batam untuk mempergunakan, mendistribusikan ataupun memproduksi ulang seluruh hasil Tugas Akhir ini.

Batam, 21 Februari 2011

**Lies Handayani**  
3310801090

## LEMBAR PERNYATAAN

Dengan ini, saya:

NIM : 3310801110

Nama : Ferri Martin

adalah mahasiswa Teknik Informatika Politeknik Negeri Batam yang menyatakan bahwa tugas akhir dengan judul:

PENGEMBANGAN SISTEM UJIAN *ONLINE* BERBASIS OBJEK

disusun dengan:

1. tidak melakukan plagiat terhadap naskah karya orang lain
2. tidak melakukan pemalsuan data
3. tidak menggunakan karya orang lain tanpa menyebut sumber asli atau tanpa ijin pemilik

Jika kemudian terbukti terjadi pelanggaran terhadap pernyataan di atas, maka saya bersedia menerima sanksi apapun termasuk pencabutan gelar akademik.

Lembar pernyataan ini juga memberikan hak kepada Politeknik Negeri Batam untuk mempergunakan, mendistribusikan ataupun memproduksi ulang seluruh hasil Tugas Akhir ini.

Batam, 21 Februari 2011

**Ferri Martin**  
3310801110

## KATA PENGANTAR

Puji syukur kehadirat Allah SWT karena atas rahmat dan limpahan karunia-Nya penyusun dapat menyelesaikan Laporan Tugas Akhir yang berjudul “Pengembangan Sistem Ujian *Online* Berbasis Objek”.

Dalam kesempatan ini, penyusun ingin menyampaikan ucapan ribuan terima kasih kepada pihak-pihak yang telah membantu penyusun dalam menyelesaikan laporan ini, baik dalam memberikan semangat, dukungan, bimbingan, waktu dan biaya, antara lain:

1. Orang tua dan keluarga yang telah memberikan dukungan baik moral maupun materi serta ribuan kata maaf jika dari dulu sampai sekarang sering mengecewakan dan menyia-nyiakan kesempatan yang telah diberikan.
2. Bapak Uuf Brajawidagda selaku Koordinator Tugas Akhir
3. Bapak Riwinoto selaku Pembimbing Tugas Akhir ini. Terima kasih atas bimbingan dan motivasi yang bapak berikan dalam penyelesaian laporan ini dan kami juga mohon maaf apabila terdapat kesalahan selama proses bimbingan.
4. Andria Gutama, A.Md yang telah membantu dan memberi saran dalam pembuatan aplikasi ini.
5. Buat sahabat dan teman-teman yang tidak dapat penyusun sebutkan satu per satu yang telah membantu dalam menyelesaikan laporan ini. Terima kasih atas diskusi-diskusi serta kritik dan saran yang telah diberikan.

Penyusun juga menyadari bahwa masih terdapat kekurangan bahkan jauh dari kesempurnaan dalam penyusunan Tugas Akhir ini. Untuk itu, penyusun mengharap kritik dan saran yang konstruktif dari pihak-pihak lain.

Semoga laporan ini dapat bermanfaat bagi pembaca, khususnya bagi yang ingin mengembangkan aplikasi seperti ini.

Batam, 21 Februari 2011

Penyusun

## **ABSTRAKSI**

Tujuan Tugas Akhir ini adalah untuk mengatasi permasalahan ujian-ujian yang masih dilakukan secara *offline* dan manual. Tugas Akhir ini mengembangkan penelitian sebelumnya yang telah dilakukan oleh Januardi dan Jonathan RK yang menggunakan rancangan prosedural.

Tahap pertama adalah melakukan analisis fitur-fitur dari penelitian sebelumnya. Kemudian dilakukan identifikasi terhadap sistem yang diusulkan. Selanjutnya dilakukan tahapan analisa, perancangan, implementasi dan pengujian berbasis objek.

Penelitian ini membuktikan bahwa aplikasi yang dikembangkan ini merupakan aplikasi ujian *online general* dengan model kelulusan berbasis nilai dan kuota.

Kata kunci: Ujian *Online General*, Perancangan Objek, Kelulusan Berbasis Nilai dan Kuota

## **ABSTRACT**

The purpose of final project is to solve problem of tests that are done offline and manually. The final project developed previous research that had been done by Januardi and Jonathan RK using procedural design.

The first step is to analysis the features of previous research. Then do the identify of the proposed system. Next do the analysis, design, implementation and object-based testing.

The research proved that the developed application was general application online exam with passing model based value and quota.

Key words: General Online Exam, Object Design, Graduation based value and quota.

## DAFTAR ISI

LEMBAR PENGESAHAN.....	i
LEMBAR PERNYATAAN .....	ii
KATA PENGANTAR .....	iv
ABSTRAKSI.....	v
ABSTRACT .....	vi
DAFTAR ISI .....	iii
DAFTAR GAMBAR.....	vi
DAFTAR TABEL .....	vii
Bab I   Pendahuluan .....	1
I.1   Latar Belakang .....	1
I.2   Rumusan Masalah .....	2
I.3   Batasan Masalah.....	3
I.4   Tujuan.....	3
I.5   Sistematika Penulisan.....	3
Bab II   Tinjauan Pustaka .....	5
II.1   Deskripsi Umum Ujian Online pada Tiap Pengembang Sebelumnya .....	5
II.2   Ujian Online.....	7
II.3   Object Oriented Programming (OOP).....	8
II.4   Object Oriented Programming (OOP) dalam PHP .....	12
II.5   Pendekatan dengan Metode UML (Unified Model Language) .....	17
II.6   PostgreSQL.....	21
Bab III  Analisis .....	22
III.1  Analisis Perbandingan Ujian Online pada Setiap Pengembang Sebelumnya.....	22
III.2  Spesifikasi Pengembangan Sistem Ujian Online Berbasis Objek.....	23
III.3  Deskripsi Umum Sistem .....	25
III.4  Fitur Utama Perangkat Lunak.....	27
III.5  Kebutuhan Fungsional.....	27
III.6  Kebutuhan Non Fungsional .....	28

III.7	Use Case .....	29
III.7.1	Diagram Use Case .....	29
III.7.2	Skenario Use Case.....	30
III.8	Analisis Kelas .....	36
Bab IV	Perancangan .....	38
IV.1	Interaction Diagram.....	38
IV.1.1	Interaction Diagram Use Case Login .....	38
IV.1.2	Interaction Diagram Use Case Mengelola Profil Organisasi.....	39
IV.1.3	Interaction Diagram Use Case Mengelola Daftar User.....	39
IV.1.4	Interaction Diagram Use Case Mengelola Program Studi.....	40
IV.1.5	Interaction Diagram Use Case Mengelola Profil Ujian.....	41
IV.1.6	Interaction Diagram Use Case Mengelola Soal dan Jawaban.....	42
IV.1.7	Interaction Diagram Use Case Menyiapkan Soal Siap Uji.....	43
IV.1.8	Interaction Diagram Use Case Mengelola Daftar Peserta.....	43
IV.1.9	Interaction Diagram Use Case Menentukan Kelulusan dan Ranking.....	44
IV.1.10	Interaction Diagram Use Case Menentukan Status Ujian.....	45
IV.1.11	Interaction Diagram Use Case Melakukan Registrasi.....	45
IV.1.12	Interaction Diagram Use Case Melaksanakan Ujian.....	46
IV.1.13	Interaction Diagram Use Case Menampilkan Hasil Ujian.....	47
IV.2	Diagram Kelas.....	47
IV.3	Rancangan Kelas Rinci.....	48
IV.3.1	Kelas Admin.....	48
IV.3.2	Kelas Prodi.....	49
IV.3.3	Kelas Profil Ujian.....	49
IV.3.4	Kelas Subjek.....	50
IV.3.5	Kelas Soal.....	50
IV.3.6	Kelas Peserta.....	51
IV.3.7	Kelas Ujian.....	51
IV.3.8	Kelas Query.....	52
IV.3.9	Algoritma.....	53

IV.4	Perancangan Antarmuka .....	69
IV.4.1	Admin.....	69
IV.4.2	Program Studi.....	71
IV.4.3	Subjek.....	71
IV.4.4	Profil Ujian.....	72
IV.4.5	Soal.....	74
IV.4.6	Peserta.....	75
IV.4.7	Ujian.....	76
IV.5	Diagram Entity Relationship (ER) .....	76
Bab V	Implementasi dan Pengujian .....	81
V.1	Implementasi Kelas .....	81
V.2	Implementasi Antarmuka.....	82
V.3	Implementasi Instalasi .....	84
V.4	Skenario Pengujian.....	89
V.5	Dokumen Rinci Testing.....	95
V.5.1	Tim Penguji.....	95
V.5.2	Hasil Rinci Pengujian.....	95
V.6	Analisa Hasil Pengujian.....	118
Bab VI	Kesimpulan dan Saran .....	119
VI.1	Kesimpulan .....	119
VI.2	Saran .....	119
	DAFTAR PUSTAKA .....	120

## DAFTAR GAMBAR

Gambar 1 Contoh Pewarisan Kelas .....	11
Gambar 2 Konsep Kerja HTML.....	13
Gambar 3 Skema PHP.....	13
Gambar 4 Deskripsi Umum Sistem .....	25
Gambar 5 Use Case Ujian Online.....	29
Gambar 6 Analisis Kelas.....	36
Gambar 7 Interaction Diagram Use Case Login .....	38
Gambar 8 Interaction Diagram Use Case Mengelola Profil Organisasi .....	39
Gambar 9 Interaction Diagram Use Case Mengelola Daftar User .....	39
Gambar 10 Interaction Diagram Use Case Mengelola Program Studi.....	40
Gambar 11 Interaction Diagram Use Case Mengelola Subjek.....	41
Gambar 12 Interaction Diagram Use Case Mengelola Konfigurasi Kelulusan ....	41
Gambar 13 Interaction Diagram Use Case Mengelola Soal dan Jawaban.....	42
Gambar 14 Interaction Diagram Use Case Menyiapkan Soal Siap Uji.....	43
Gambar 15 Interaction Diagram Use Case Mengelola Daftar Peserta .....	43
Gambar 16 Interaction Diagram Use Case Menentukan Kelulusan dan Peringkat... .....	44
Gambar 17 Interaction Diagram Use Case Menentukan Status Ujian.....	45
Gambar 18 Interaction Diagram Use Case Melakukan Registrasi .....	45
Gambar 19 Interaction Diagram Use Case Melaksanakan Ujian .....	46
Gambar 20 Interaction Diagram Use Case Menaampirkan Hasil Ujian .....	47
Gambar 21 Diagram Kelas .....	47
Gambar 22 Rancangan Tampilan Profil Organisasi dan Pengaturan Hak akses...	69
Gambar 23 Rancangan Tampilan Program Studi .....	71
Gambar 24 Rancangan Tampilan Subjek dan Tingkat Kesulitan.....	71
Gambar 25 Rancangan Tampilan Profil Ujian .....	72
Gambar 26 Rancangan Tampilan Soal dan Jawaban.....	74
Gambar 27 Rancangan Tampilan Peserta .....	75
Gambar 28 Rancangan Tampilan Penentuan Status Ujian .....	76

Gambar 29 Diagram Entity Relationship (ER) .....	78
Gambar 30 Skenario Pengujian .....	89

## DAFTAR TABEL

Tabel 1 Analisis Perbandingan Ujian Online Pada Tiap Pengembang Sebelumnya .....	22
Tabel 2 Spesifikasi Pengembangan Ujian Online Berbasis Objek .....	23
Tabel 3 Spesifikasi analisis Kelas .....	37
Tabel 4 Rancangan Rinci Setiap Kelas .....	48
Tabel 5 Deskripsi Rancangan Tampilan Profil Organisasi dan Pengaturan Hak Akses .....	70
Tabel 6 Deskripsi Rancangan Tampilan Program Stud .....	71
Tabel 7 Deskripsi Rancangan Tampilan Subjek dan Tingkat Kesulitan .....	72
Tabel 8 Deskripsi Rancangan Tampilan Profil Ujian .....	73
Tabel 9 Deskripsi Rancangan Tampilan Soal dan Jawaban .....	74
Tabel 10 Deskripsi Rancangan Tampilan Peserta .....	75
Tabel 11 Deskripsi Rancangan Tampilan Penentuan Status Ujian .....	77
Tabel 12 Implementasi Kelas .....	81
Tabel 13 Implementasi Antarmuka .....	82
Tabel 14 Hasil Rinci Pengujian .....	95

## **Bab I   Pendahuluan**

Bab ini menjelaskan tentang latar belakang, rumusan masalah, batasan masalah, tujuan, dan sistematika penulisan.

### **I.1   Latar Belakang**

Di era globalisasi seperti sekarang ini, ujian *online* sangat diperlukan baik untuk penerimaan pegawai baru, mahasiswa baru, dan sebagainya.

Di Politeknik Negeri Batam, sudah banyak Mahasiswa yang membahas Tugas Akhirnya tentang sistem ujian *online*, diantaranya Muhammad Tarmizi (2005) Aplikasi Tes *Online*, Karmila (2008) Aplikasi Tes UMPB Berbasis Komputer, Muhammad Irfan Fahmi (2008) Aplikasi Simulasi Ujian SPMB, Januardi dan Jonathan RK (2010) Aplikasi Ujian *Online*. Dari aplikasi yang dibuat oleh pengembang-pengembang tersebut masih memiliki beberapa kekurangan, seperti dari segi perancangan yang masih menggunakan rancangan prosedural yang mengakibatkan sistem tidak *reusable*. Selain itu, pada aplikasi tersebut masih memiliki aturan-aturan yang bersifat *static* yang mengakibatkan sistem menjadi kaku dan tidak dapat diperbaharui.

Tugas Akhir dengan judul Pengembangan Sistem Ujian *Online* Berbasis Objek ini dibuat dengan dua tipe ujian, yaitu Ujian Nasional (UN) dan Sertifikasi. Hal ini dikarenakan masih terdapat ujian *online* seperti UN dan Sertifikasi yang dilakukan secara manual dan *offline*. Hal tersebut mengakibatkan ketidakefisienan dan keefektifan, karena peserta harus membawa berbagai peralatan, seperti pensil 2B, penghapus, *ballpoint*, dan sebagainya. Selain itu, hal tersebut juga membutuhkan waktu yang cukup lama untuk mengetahui informasi kelulusan karena proses pengoreksian jawaban masih dilakukan secara manual. Ketidakefisienan dan keefektifan ini dapat terjadi baik untuk instansi atau Perguruan Tinggi yang melaksanakan ujian maupun untuk peserta ujian.

Oleh karena itu, Tugas Akhir ini dirancang secara *general* agar instansi manapun dapat menggunakan aplikasi ini dalam menerapkan ujian *online* baik untuk ujian dengan tipe UN atau Sertifikasi. Selain itu, aplikasi ini juga dirancang secara *dynamic* agar dapat diperbaharui sesuai dengan kebutuhan instansi yang akan melaksanakan ujian *online* ini.

Perkembangan model pemrograman telah mencapai model pemrograman berorientasi objek atau *Object-Oriented Programming* (OOP) setelah era pemrograman prosedural (*procedural programming*). Pemrograman berorientasi objek (OOP) merupakan suatu pendekatan pemrograman yang menggunakan *object* dan *class*. Keuntungan yang diperoleh dari pendekatan OOP ini, antara lain:

1. Mudah dimodifikasi dan dirawat (*maintenance*).
2. *Problem Oriented Approach*, OOP merupakan teknik pemrograman yang melakukan pendekatan terhadap masalah yang terjadi, sehingga lebih mudah menciptakan aplikasi untuk menangani masalah yang terjadi.
3. *Information Hiding*, data dalam sebuah kelas belum tentu dapat diakses atau diketahui pihak luar.
4. *Reusability dan Extensibility*, yaitu komponen yang sudah dibuat dapat digunakan lagi pada sistem yang lain yang membutuhkan komponen tersebut.
5. *Low probability errors*, dengan OOP *programmer* dapat mengurangi kemungkinan kesalahan.

Oleh karena itu, dengan adanya kelebihan-kelebihan tersebut Tugas Akhir ini dikembangkan dengan menggunakan rancangan berorientasi objek.

## **I.2 Rumusan Masalah**

Dari uraian diatas dapat dirumuskan bahwa masalah dalam tugas akhir ini adalah:

1. Melakukan analisis pada aplikasi yang telah ada.
2. Mengembangkan Aplikasi Ujian *Online* yang sebelumnya menggunakan rancangan prosedural menjadi rancangan berorientasi objek.

### **I.3 Batasan Masalah**

Adapun batasan masalah dari aplikasi ini adalah:

1. Hanya menangani soal dalam bentuk pilihan berganda.
2. Tidak menangani penambahan waktu ujian.
3. Tidak melakukan pengambilan keputusan otomatis untuk kelulusan pada tipe ujian UN.
4. Hanya menangani ujian *online* yang dilakukan oleh satu organisasi.

### **I.4 Tujuan**

Tujuan pembuatan aplikasi ini adalah:

1. Bebas disparitas geografis yang dapat mempermudah proses pelaksanaan ujian.  
Dengan menggunakan jaringan dan internet aplikasi komputer berbasis web dapat diakses dari manapun dan kapanpun.
2. Aplikasi mampu menangani ujian dengan tipe UN dan Sertifikasi.
3. Dapat mempermudah proses pengoreksian jawaban karena dilakukan oleh sistem.
4. Aplikasi dapat menampilkan hasil ujian dan peringkat yang diraih peserta.
5. Aplikasi dikelola secara *general*, agar dapat digunakan oleh seluruh instansi yang akan melaksanakan ujian *online*.
6. Aplikasi ini dikembangkan dengan menggunakan rancangan berorientasi objek.

### **I.5 Sistematika Penulisan**

Adapun sistematika penulisan dalam aplikasi ini adalah:

#### **BAB 1 Pendahuluan**

Berisi tentang penjelasan latar belakang, rumusan masalah, batasan masalah, tujuan penelitian serta sistematika penulisan untuk memberikan gambaran isi laporan tugas akhir ini.

## **BAB II Tinjauan Pustaka**

Berisi tentang teori-teori yang berhubungan dengan ujian *online*, program berorientasi objek, PHP berdasarkan penelitian-penelitian yang sudah ada serta penelitian yang dilakukan.

## **BAB III Analisis**

Berisi tentang deskripsi hubungan antara perangkat keras dan perangkat lunak, terdiri dari deskripsi umum sistem, fitur utama perangkat lunak, kebutuhan fungsional, kebutuhan non fungsional, diagram *usecase* beserta skenarionya serta analisis kelas.

## **BAB IV Perancangan**

Berisi tentang *Interaction Diagram*, diagram kelas, rancangan kelas rinci dan perancangan antarmuka.

## **BAB V Implementasi dan Pengujian**

Berisi tentang uraian langkah implementasi dan pengujian, baik implementasi kelas maupun antarmuka.

## **BAB VI Kesimpulan dan Saran**

Berisi tentang kesimpulan yang merupakan rangkuman dari hasil analisis kinerja pada bagian sebelumnya serta saran-saran pengembangan dari penelitian yang dibuat dan aspek yang belum terselesaikan.

## **Bab II      Tinjauan Pustaka**

Bab ini berisi tentang teori-teori yang berhubungan dengan ujian *online*, program berorientasi objek dan PHP berdasarkan penelitian-penelitian yang sudah ada serta penelitian yang dilakukan.

### **II.1 Deskripsi Umum Ujian Online pada Pengembang Sebelumnya**

Di Politeknik Negeri Batam, sudah banyak Mahasiswa yang membahas Tugas Akhirnya tentang sistem ujian *online*, diantaranya Muhammad Tarmizi (2005) Aplikasi Tes *Online*, Karmila (2008) Aplikasi Tes UMPB Berbasis Komputer, Muhammad Irfan Fahmi (2008) Aplikasi Simulasi Ujian SPMB, Januardi dan Jonathan RK (2010) Aplikasi Ujian *Online*.

Deskripsi umum pada Aplikasi Tes *Online* yang dibuat oleh Muhammad Tarmizi (2005) terdiri dari:

1. Aplikasi mampu menampilkan soal tes secara acak untuk tiap *user*.
2. Aplikasi mampu menyimpan hasil tes peserta.
3. Aplikasi mampu menghitung nilai tes peserta.

Dari deskripsi umum Aplikasi Tes *Online* yang dibuat oleh Muhammad Tarmizi masih memiliki kekurangan, seperti:

1. Pemilihan level pada soal yang masih statis.
2. Pengaturan soal ujian yang belum teratur menurut level.
3. Penyajian nilai ujian yang masih statis.

Deskripsi umum pada Aplikasi Tes UMPB Berbasis Komputer yang dibuat oleh Karmila (2008) terdiri dari:

1. Aplikasi dapat menampilkan soal-soal ujian secara acak.
2. Aplikasi dapat menentukan jumlah soal yang akan dikerjakan oleh peserta.
3. Aplikasi dapat menghitung nilai dari hasil tes UMPB secara cepat.

4. Aplikasi dapat melihat peserta-peserta yang diterima dan ditolak pada program studi tertentu berdasarkan hasil tes.
5. Aplikasi dapat menampilkan *report*/laporan hasil tes UMPB.
6. Aplikasi dapat melakukan pengelolaan data soal yang terdiri dari *insert*, *update* dan *delete* soal.
7. Aplikasi dapat melakukan penambahan mata pelajaran.
8. Aplikasi mampu mengelola data soal bertipe suara dan gambar.
9. Aplikasi dapat melakukan pengelolaan data peserta yang terdiri dari *insert*, *update* dan *delete* peserta ujian.
10. Aplikasi dapat melakukan pencarian *user* berdasarkan nama, *id\_user*, dan program studi.

Dari deskripsi umum Aplikasi Tes UMPB Berbasis Komputer yang dibuat oleh Karmila masih memiliki kekurangan, seperti:

1. Tidak memiliki bank soal.
2. Penyediaan level masih statis.
3. Pengaturan nilai ujian yang masih statis.

Deskripsi umum pada Aplikasi Simulasi Ujian SPMB yang dibuat oleh Muhammad Irfan Fahmi (2008) terdiri dari:

1. Aplikasi dapat berjalan secara *client-server*.
2. Aplikasi dapat menampilkan soal secara acak.
3. Aplikasi dapat memberikan laporan kelulusan.

Dari deskripsi umum Aplikasi Simulasi Ujian SPMB yang dibuat oleh Muhammad Irfan Fahmi masih memiliki kekurangan, seperti:

1. Aplikasi tidak dapat memberikan informasi peringkat.
2. Aplikasi tidak dapat memberikan pembahasan jawaban.

Deskripsi umum pada Aplikasi Ujian *Online* yang dibuat oleh Januardi dan Jonathan RK (2010) terdiri dari:

1. Aplikasi dapat membuat mata pelajaran sesuai kebutuhan.

2. Aplikasi dapat memberikan informasi hasil jawaban dan kunci jawaban dari soal-soal yang telah dikerjakan.
3. Aplikasi dapat membuat level dan nilai yang dapat diubah sesuai kebutuhan.
4. Aplikasi dapat melakukan pengaturan soal yang akan ditampilkan.
5. Aplikasi dapat memberikan informasi peringkat.
6. Aplikasi dapat membuat aturan waktu pelaksanaan ujian.
7. Aplikasi dapat menampilkan soal ujian dengan pembagian level yang ditentukan.

Dari deskripsi umum Aplikasi Ujian *Online* yang dibuat oleh Januardi dan Jonathan RK masih memiliki kekurangan, seperti:

1. Tidak dapat membuat random pada pilihan jawaban.
2. Tidak dapat menangani registrasi peserta ujian yang dilakukan secara *online*.
3. Tidak menangani tempat pemberian *checklist* pada soal ujian yang digunakan sebagai tanda jika peserta ujian merasa ragu-ragu pada jawaban soal tersebut.

## **II.2 Ujian *Online***

Ujian *online* merupakan suatu ujian yang dilaksanakan dengan memanfaatkan jaringan dan internet. Dengan memanfaatkan sistem ini, peserta ujian dapat mengakses dan melakukan ujian dimanapun dan kapanpun. Ujian *online* memiliki beberapa tipe, antara lain Ujian Nasional (UN), yaitu ujian yang dilakukan sebagai evaluasi standar pendidikan secara nasional oleh Departemen Pendidikan Nasional (Depdiknas). Selain itu, juga ada tipe ujian sertifikasi yaitu ujian yang dilaksanakan sebagai solusi untuk pemenuhan sumber daya manusia yang berkualitas.

Latar belakang diadakannya ujian *online* ini karena adanya lonjakan minat para pelajar untuk melanjutkan pendidikannya ke jenjang yang lebih tinggi. Selain itu, ujian *online* ini sangat membantu dalam peningkatan kualitas sumber daya manusia. Dengan adanya ujian *online* berbasis *web* yang ditunjang dengan

teknologi pemanfaatan jaringan dan internet maka proses penerimaan siswa baru ataupun mahasiswa baru akan lebih efektif dan efisien.

“Salah satu tujuan ujian *online* adalah mengatasi masalah dan kekhawatiran pada kertas yang ada pada ujian pilihan ganda. Idealnya, orang akan menginginkan sebuah sistem yang diperlukan dalam pekerjaan siswa dan memberikan nilai berdasarkan solusi seluruh proses dan bukan hanya angka dalam nilai akhir. Sistem ujian *online* adalah sistem pengujian baru yang memungkinkan melakukan *generate* soal secara dinamis.”(Alimudin: 2009).

### **II.3 Object Oriented Programming (OOP)**

Pemrograman berorientasi objek (OOP) yang populer pada masa kini menjadikan objek sebagai komponen utama dalam program. Objek menggabungkan data dan fungsi sebagai satu kesatuan, sedangkan pada pemrograman yang tidak berorientasi objek, fungsilah yang menjadi perhatian utama. Perlu diketahui, fungsi adalah subprogram atau sekumpulan kode yang diberi satu nama tersendiri. Pendekatan pemrograman berorientasi objek membuat pengembangan program dapat dilakukan dengan lebih mudah, mengurangi duplikasi kode dan mengurangi kesalahan.

OOP adalah suatu program yang dapat mengorganisasikan program sebagai kumpulan komponen yang disebut objek. Objek-objek ini ada secara independen, mempunyai aturan-aturan berkomunikasi dengan objek lain dan untuk memerintahkan objek lain guna meminta informasi tertentu atau meminta objek lain untuk mengerjakan sesuatu.

OOP bukanlah sekedar cara penulisan sintak program yang berbeda, namun lebih dari itu, OOP merupakan cara pandang dalam menganalisa sistem dan permasalahan pemrograman. Dalam OOP, setiap bagian dari program adalah objek. Sebuah objek mewakili suatu bagian program yang akan diselesaikan.

Istilah-istilah yang terdapat pada pemrograman berorientasi objek adalah:

## 1. Objek

Objek merupakan elemen pada saat *run time* yang akan diciptakan, dimanipulasi dan dihancurkan. Sebuah objek memiliki *attribute* dan *method*.

*Attribute* adalah data yang terkandung di dalam objek. Sedangkan *method* adalah operasi-operasi yang disediakan oleh objek untuk mengakses atau memodifikasi *attribute* yang dimilikinya.

Contoh:

Sebuah objek **Ujian** memiliki *attribute*:

- Nomor Ujian
- Nama Peserta

Objek **Ujian** ini menyediakan *method* berupa:

- Informasi kelulusan
- Melihat Nilai

## 1. Kelas

Kelas adalah deskripsi statik dari himpunan objek yang mungkin lahir atau diciptakan. Di dalam kelas terdapat penjelasan tentang suatu objek termasuk *attribute* yang dimilikinya serta *method* yang bisa dilakukan oleh objek. Kelas pada *Java* didefinisikan dengan menggunakan kata kunci **class**, contoh **class Orang**. Pada kelas tersebut memiliki *attribute* seperti tangan, kaki, mata, dan sebagainya. Selain itu, kelas tersebut juga memiliki *method* seperti bisa berjalan, lari, melompat, dan sebagainya.

## 2. Konstruktor

Konstruktor merupakan *method* yang dapat digunakan untuk memberikan nilai awal pada saat objek diciptakan. *Method* ini akan dipanggil secara otomatis oleh *Java* ketika **new** dipakai untuk menciptakan *instance* kelas. Di dalam konstruktor

ada yang disebut dengan istilah *overloading constructor*, yaitu dalam sebuah kelas dapat memiliki lebih dari satu konstruktor, namun masing-masing konstruktor harus memiliki *argument list* yang berbeda-beda sebagai penandanya.

Konstruktor memiliki sifat:

- Namanya sama dengan nama kelas.
- Tidak memiliki nilai balik (termasuk tidak boleh ada kata kunci *void*).

### **3. Package**

*Package* adalah suatu tempat yang digunakan untuk mengelompokkan kelas-kelas yang memiliki kesamaan fungsionalitas. Kelas yang berada pada direktori yang sama, otomatis berada dalam *package* yang sama (berhubungan langsung dengan *filesystem*). Selain itu, *package* juga membantu mengelola sistem *software* yang berukuran besar. Nama *package* harus disusun secara hirarki dan dipisahkan dengan dot.

Selain istilah-istilah di atas, OOP juga memiliki beberapa konsep dasar, antara lain:

#### **1. Encapsulation**

*Encapsulation* adalah proses pemaketan atau pembungkusan data dalam sebuah kelas yang terdiri dari atribut, konstruktor dan metode-metodenya. Manfaat dari proses ini adalah:

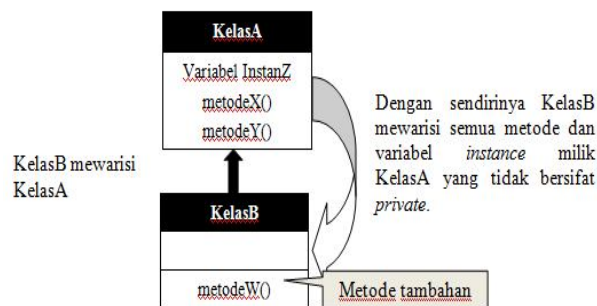
- Menyembunyikan implementasi (*Hiding Implementation*), yaitu perlindungan implementasi internal objek.
- Modularitas (*Modularity*), yaitu objek dapat dikelola secara independen.
- Membuat kode lebih terpelihara.
- Memaksa pengguna untuk menggunakan *method* untuk mengakses data.

## 2. Inheritance

“*Inheritance* adalah proses penciptaan kelas baru dengan mewarisi karakteristik kelas yang telah ada, ditambah karakteristik unik kelas baru itu. Pewarisan adalah mekanisme ampuh yang memungkinkan kelas mewarisi fungsionalitas kelas yang telah ada. Untuk menciptakan kelas baru, kita hanya perlu menspesifikasikan cara kelas baru itu berbeda dari kelas yang telah ada. Pewarisan merupakan mekanisme yang ampuh dan alami untuk mengorganisasikan program.

Dengan pewarisan, dimungkinkan penciptaan klasifikasi berhirarki. Selain itu, kita dapat menciptakan kelas umum yang mendefinisikan perilaku umum dari item-item yang saling berhubungan. Kelas ini kemudian dapat diwarisi kelas-kelas lain, kelas yang lebih spesifik menambahkan hal-hal yang unik di kelas itu. Dalam terminologi *Java*, kelas yang diwarisi disebut *superclass*, sedangkan kelas yang mewarisi disebut *subclass*. Dengan demikian, *subclass* adalah versi spesial dari *superclass*.”(Esensi-esensi Pemrograman Java, 2003: 370-371).

Contoh:



Gambar 1 Pewarisan Kelas

Pada Gambar 1 dijelaskan bahwa KelasA disebut *superclass*, sedangkan KelasB disebut *subclass*.

## 3. Polymorphism

*Polymorphism* berarti mempunyai banyak bentuk. Dua objek atau lebih dikatakan *polymorphic* apabila objek-objek tersebut mempunyai antarmuka-antarmuka

identik, namun memiliki perilaku-prilaku yang berbeda. Ada dua macam bentuk *polymorphism*, yaitu *overloading* dan *overriding*.

*Overloading* adalah penggunaan satu nama untuk beberapa *method* yang berbeda dalam suatu kelas. *Method-method* yang namanya sama ini dibedakan satu dengan yang lain berdasarkan parameter yang disediakan masing-masing *method*.

*Overriding* adalah mendeklarasikan sebuah *method* dengan nama dan parameter yang sama dengan suatu *method* dari *superclassnya*. *Method* yang dideklarasikan di *subclass* akan *mengoverride method* dari *superclassnya*.

## **II.4 Object Oriented Programming (OOP) dalam PHP**

PHP (PHP *Hypertext Preprocessor*) merupakan bahasa berbentuk *script* yang ditempatkan dalam *server* dan diproses di *server*, kemudian hasil dikirimkan ke *browser* dalam bentuk HTML biasa. Oleh karena itu, PHP termasuk program web untuk *server side* yang merupakan *website* yang menggunakan bahasa pemrograman yang tergantung kepada tersedianya *server*. Jika tidak ada *server*, *website* yang dibangun tidak dapat berfungsi sebagaimana mestinya.

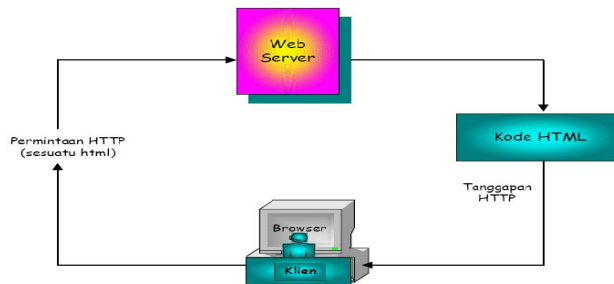
PHP adalah suatu bahasa pemrograman *open source* yang digunakan secara luas terutama untuk pengembangan web. Secara khusus PHP dirancang untuk membentuk web dinamis. Contohnya, seseorang dapat membentuk suatu tampilan berdasarkan permintaan terkini, misal menampilkan isi *database* ke halaman web.

Kelebihan PHP antara lain:

1. *Script* PHP sederhana, mudah dan memiliki kecepatan akses.
2. Dapat berjalan dalam *server* web yang berbeda-beda dan juga dapat berjalan di sistem operasi yang berbeda-beda pula.
3. PHP termasuk bahasa pemrograman yang *embedded*, yaitu *script* PHP bisa ditempel dan diletakkan dalam tag HTML.

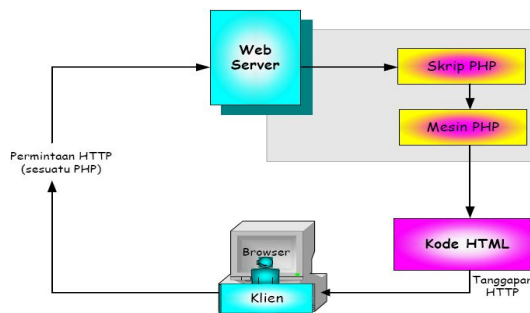
Konsep kerja HTML diawali dengan permintaan suatu halaman web oleh *browser*. Berdasarkan URL (*Uniform Resource Locator*) atau dikenal dengan

internet, *browser* mendapat alamat dari *web server*, mengidentifikasi halaman yang dikehendaki, dan menyampaikan segala informasi yang dibutuhkan oleh *web server*. Selanjutnya, *web server* akan mencari berkas yang diminta dan memberikan isinya ke *browser*. *Browser* yang mendapatkan isinya segera melakukan proses penerjemahan kode HTML dan menampilkannya ke layar *client*. Hal ini seperti terlihat pada Gambar 2.



Gambar 2 Konsep Kerja HTML

Jika yang diminta adalah sebuah halaman PHP, prinsip kerjanya sama, hanya saja ketika berkas PHP yang diminta didapatkan oleh *web server*, isinya segera dikirimkan ke mesin PHP dan mesin inilah yang memproses dan memberikan hasilnya (berupa kode HTML) ke *web server*. Selanjutnya, *web server* menyampaikannya ke *client*, seperti yang terlihat pada Gambar 3.



Gambar 3 Skema PHP

*Script* PHP diawali dengan tag awal berupa tanda lebih kecil (<) dan diakhiri dengan tag akhir berupa tanda lebih besar (>). Contoh:

```
<?...?>
<?php...?>
<script language="PHP">...</script>
Contoh script PHP sederhana:
<?php
Echo ("Hello PHP, Aku akan menaklukkanmu!!");
?>
```

Seiring dengan perkembangan internet yang ditandai dengan maraknya *dotcom* dan *E-Commerce*, aplikasi *web* juga berkembang menjadi semakin kompleks, kemudian timbul inisiatif dari komunitas pengguna PHP untuk menyempurnakan struktur bahasa pemrograman PHP. Dimulai dengan PHP 3 yang mulai memperkenalkan penggunaan objek dan kelas meskipun dari segi konsep. Kemudian dilanjutkan dengan lahirnya PHP 4, meskipun penerapan konsep *object oriented* masih sangat terbatas.

Uniknya, segala keterbatasan konsep OOP yang ada dalam PHP 3 dan PHP 4 tidak menyurutkan niat pengguna PHP untuk mencoba membuat program PHP-nya mengikuti konsep *object oriented*, bahkan jumlahnya semakin meningkat.

PHP 5 menggunakan model pengelolaan objek yang baru dimana sebuah objek tidak lagi diolah berdasarkan *value*, melainkan berdasarkan *reference* sehingga saat dibuat *instance* sebuah objek tidak lagi dilakukan proses penyalinan terhadap objek yang sesungguhnya, melainkan menggunakan objek *handler*. Objek *handler* inilah yang mengaitkan *instance* tersebut ke *table global object*, dimana objek sesungguhnya disimpan.

PHP 5 juga memiliki fitur-fitur baru untuk mendukung implementasi *object oriented programming* seperti penerapan *public*, *private*, dan *protected class member*, *static variables*, *abstract class*, dan *interface*.

Dasar pertimbangan untuk mengembangkan kemampuan *object oriented* dari PHP adalah perkembangan aplikasi *web* sebagai sebuah *platform* yang terus meluas dengan cepat karena ruang lingkup aplikasi *web* semakin luas sehingga aplikasi

*web* yang dibangun juga semakin besar, rumit, dan kompleks. Aplikasi *web* telah diimplementasikan mulai dari tingkatan yang paling sederhana seperti *online news* hingga ke tingkatan *enterprise* seperti *online banking*, *Supply Chain Management* (SCM), *Customer Relationship Management* (CRM), dan sebagainya.

Dengan kemampuan untuk mengimplementasikan konsep *object oriented*, PHP akan lebih mudah untuk dapat berhubungan dan mengakses komponen-komponen yang dibuat dengan bahasa-bahasa *object oriented* lain seperti C++ dan bahkan *Visual Basic*.

#### Mengelola kompleksitas aplikasi

Pemrograman berorientasi objek memandang aplikasi perangkat lunak (*software*) sebagai kumpulan objek yang saling berinteraksi di dalam suatu sistem. Merancang sebuah aplikasi menggunakan teknik *object oriented* dilakukan dengan membagi fungsi-fungsi berdasarkan pembagian tanggung jawab. Pembagian tanggung jawab ditetapkan kepada setiap kelas yang dibuat. Setiap kelas menyediakan pelayanan untuk mengerjakan operasi tertentu. Operasi-operasi ini dilaksanakan oleh objek-objek yang dibuat dari kelas tersebut. Dengan memfokuskan setiap kelas pada tanggung jawab tertentu dengan tingkatan kompleksitas yang memadai tentu akan sangat membantu untuk mereduksi kompleksitas aplikasi secara keseluruhan.

Keuntungan yang diperoleh dari pendekatan *object oriented* ini adalah kemudahan untuk mengelola kompleksitas aplikasi yang dibuat dan kemudahan untuk melakukan perubahan dan pengembangan dari aplikasi tersebut.

Dengan teknik *object oriented*, jalinan komunikasi antar *programmer* juga dipermudah karena masing-masing kelas yang dibuat akan dilengkapi dengan *interface* yang menjadi standar bagaimana kelas tersebut bisa dihubungi.

Pada akhirnya dengan pendekatan *object oriented* ini akan dihasilkan aplikasi dengan struktur yang kokoh. Operasi dan data dibungkus dengan rapi di dalam kelas. Data hanya dapat dimanipulasi menggunakan operasi-operasi yang disediakan di dalam kelas. Bila ditemukan kesalahan (*bugs*), dapat ditelusuri kesalahan terjadi pada kelas mana, dan perbaikan pada kelas tersebut tidak akan mengakibatkan terjadinya kesalahan baru pada kelas yang lain.

Mengantisipasi adanya perubahan

Salah satu prinsip dalam perancangan aplikasi dengan pendekatan *object oriented* adalah pemisahan antara tampilan (*user interface*) dan implementasi (*business logic*). Masalah utama dalam melakukan perubahan dan pengembangan dari sebuah aplikasi adalah saling ketergantungan yang tinggi antara bagian-bagian aplikasi tersebut. Pemisahan antara bagian tampilan dengan bagian implementasi ini bertujuan untuk mengurangi saling ketergantungan antar bagian di dalam aplikasi.

Tingkat saling ketergantungan antar modul yang rendah akan memudahkan penggantian, perbaikan, atau pengembangan suatu modul dengan tidak menimbulkan efek pada modul lainnya. Modul-modul yang independen juga memungkinkan penggunaan ulang (*reuse*) dari modul-modul tersebut. Contoh skrip PHP OOP adalah:

```
<?php
    // nama class
    Class KendBermotor {
    public $mesin;
    private $roda;
    protected $jalur;
        //konstruktor
        function __construct() {
            $this->mesin ="Kendaraan Bermotor Punya Mesin <br>";
            $this->roda ="Kendaraan Bermotor Punya Roda <br>";
            $this->jalur ="Kendaraan Bermotor Punya Jalur <br>";
        }
    }
}
```

```

    }
    // method
    function getMesin() {
    return $this->mesin;
    }
    function getJalur() {
    return $this->jalur;
    }
    function getRoda() {
    return $this->roda;
    }
    }
    $KB = new KendBermotor();
    echo "<p><b> Kriteria Kendaraan Bermotor : </b><br>";
    echo $KB->getMesin();
    echo $KB->getRoda();
    echo $KB->getJalur();
?>

```

## II.5 Pendekatan dengan metode UML (*Unified Model Language*)

UML (*Unified Modeling Language*) adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan *software* berbasis OOP (*Object-Oriented*). UML juga memberikan standar penulisan sebuah sistem *blue print*, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema *database*, dan komponen-komponen yang diperlukan dalam sistem *software*. UML dapat secara langsung dihubungkan ke berbagai bahasa pemrograman, seperti JAVA, C++, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk model aplikasi *procedural* dalam VB atau C.

UML mendefinisikan diagram-diagram seperti:

## 1. Use case

*Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem.

*Use case diagram* terdiri dari:

- *Use case* dinotasikan dengan gambar (*horizontal ellipse*), biasanya menggunakan kata kerja dan nama *use case* boleh terdiri dari beberapa kata serta tidak boleh ada dua *use case* yang memiliki nama yang sama.
- *Actor* menggambarkan sebuah tugas/peran, bukan posisi sebuah jabatan. *Actor* memberi *input* atau menerima informasi dari sistem, biasanya menggunakan kata benda dan tidak boleh ada komunikasi langsung antar *actor*. *Actor* dalam *use case* terbagi menjadi dua yaitu:
  - *Primary Actor* (Aktor utama), yaitu aktor yang memprakarsai sebuah *use case*, memegang peran proaktif dan yang memulai aksi dalam sistem.
  - *Secondary Actor*, yaitu aktor yang ikut berperan atau terlibat dalam *use case*.
- *Relationship* merupakan penghubung yang digunakan untuk menggambarkan bagaimana *actor* terlibat dalam *use case*.
- *System boundary boxes* digambarkan dengan kotak disekitar *use case*, untuk menggambarkan jangkauan sistem (*scope of your system*).
- *Include* menjelaskan bahwa fungsionalitas sebuah *use case* ada di dalam *use case* lainnya. *Included use case* tidak pernah berdiri sendiri, tetapi hanya merupakan bagian dari beberapa *use case* yang lebih besar yang diikutinya.
- *Extend* menjelaskan bahwa fungsionalitas sebuah *use case* merupakan perluasan fungsionalitas *use case* lainnya. *Use case* asal dapat berdiri sendiri, tetapi untuk kondisi tertentu perilaku *use case* tersebut dapat diperluas oleh perilaku dari *use case* lain. Hubungan perluasan digunakan untuk memodelkan

bagian dari use case yang dapat dilihat oleh user sebagai perilaku yang dapat dipilih dari system.

- *Inheritance* berarti suatu elemen memiliki sifat yang sama dengan elemen lainnya. *Inheritance* antara use case berarti suatu use case merupakan versi khusus dari use case lainnya.

Contoh : *pay(card: CreditCard), pay(cheque: Cheque)*.

## 2. Kelas Analisis

Kelas analisis merupakan elemen/benda yang terdapat pada model analisis. Kelas analisis menggambarkan konsep awal mengenai benda yang terdapat dalam sistem (yang akhirnya memiliki tanggung jawab dan perilaku). Digunakan sebagai dasar kelas pada model desain. Kelas analisis terbagi menjadi 3 yaitu boundary, control, dan entity.

- **Kelas Bondary**

kelas yang memodelkan interaksi antara satu atau lebih actor dengan sistem. Kelas Boundary memodelkan bagian dari sistem yang bergantung pada pihak lain disekitarnya dan merupakan pembatas dengan dunia luar. Kelas Boundary dapat berupa:

- *User interface* yang merupakan sarana komunikasi antara sistem dengan user, misalnya jendela (window) dalam GUI.
- *System Interface* yang merupakan sarana komunikasi antara sistem dengan sistem informasi lainnya misalnya *communication protocol*.
- *Device interface* yang merupakan sarana komunikasi antara sistem dengan device (alat), seperti printer, sensor dan sebagainya.

- **Kelas Control**

Kelas control digunakan untuk memodelkan perilaku mengatur, khusus untuk satu atau beberapa use case saja. Bersifat optional, umumnya terdapat satu control pada satu use case yang mengontrol urutan events dalam use case

tersebut. Tidak semua use case memerlukan kelas control. Kelas control menghubungkan kelas bondary dengan kelas entity.

- **Kelas Entity**

Kelas entity memodelkan informasi yang harus disimpan oleh sistem. Kelas ini memperlihatkan struktur data dari sebuah sistem. Tanggung jawab utama objek ini adalah untuk menyimpan dan mengatur informasi dalam sistem.

### **3. 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).

*Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu.

Istilah-istilah dalam *sequence diagram*, antara lain:

- *Participant* merupakan objek yang terkait dengan sebuah urutan proses.
- *Lifeline* merupakan gambaran daur hidup sebuah objek.
- *Activation* merupakan suatu titik waktu dimana sebuah objek mulai berpartisipasi didalam sebuah *sequence* yang ditandai dengan sebuah *bar*.
- *Time* merupakan elemen penting dalam *sequence diagram* yang menggambarkan urutan waktu.
- *Return* merupakan suatu hasil kembalian sebuah operasi.

### **4. 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). Sedangkan *class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

## II.6 PostgreSQL

PostgreSQL merupakan database server cukup dikenal di kalangan administrator database, programmer aplikasi, maupun pengembang web. PostgreSQL merupakan database RDBMS (*Relational Database Management System*) bebas yang powerful dan tidak kalah dengan database komersil sekelasnya Oracle, maupun Sybase, dan Informix. Berbagai fitur yang ada di PostgreSQL semacam transaction, merupakan jawaban untuk mereka yang memerlukan fasilitas tersebut tapi belum menemukannya di MySQL. Database Server PostgreSQL yang kini mendukung platform NT, Solaris, Sun OS, HPUX, AIX, Linux, Irix, FreeBSD dan kebanyakan Unix lainnya telah dikembangkan sejak 1985. PostgreSQL didisain untuk skala besar, memiliki ciri berorientasi objek, namun tetap dapat digunakan layaknya RDBMS pada umumnya. Para pengembangnya (dulu) mengutamakan kelengkapan feature ketimbang dukungan penuh akan standar SQL. Dalam perjalanannya, kelebihan feature ini justru membuat Postgres semakin powerful dan membuatnya dapat disejajarkan dengan database besar lainnya seperti Oracle. Fasilitas standar lain yang telah didukung adalah trigger dan function. Function dalam PostgreSQL sangat unik karena PostgreSQL tidak menetapkan built-in language untuk penulisan function. PostgreSQL menyerahkan kepada pemakai untuk memilih sendiri bahasa yang diinginkan yang disebut Procedural Language. Tools yang tersedia cukup banyak dan sangat berguna seperti:

- `pg_dump`: konversi objek database menjadi script SQL
- `pgaccess`: drag & drop-nya cukup menyenangkan

### Bab III Analisis

Bab ini Berisi tentang analisis perbandingan ujian *online* pada setiap pengembang sebelumnya, spesifikasi pengembangan ujian *online* berbasis objek, deskripsi umum sistem, fitur utama perangkat lunak, kebutuhan fungsional, kebutuhan non fungsional, diagram *use case* beserta skenarionya serta analisis kelas.

#### III.1 Analisis Perbandingan Ujian Online pada Setiap Pengembang Sebelumnya

Analisis perbandingan ujian *online* pada setiap pengembang sebelumnya akan disajikan pada tabel 1.

Tabel 1 Analisis Perbandingan Ujian *Online* pada Setiap Pengembang Sebelumnya

No	Fitur	Aplikasi Ujian Test Online	Aplikasi Simulasi Ujian SPMB	Aplikasi Tes UMPB Berbasis Komputer	Aplikasi Ujian Online
1.	Materi ujian	Static	Static	Dinamic	Dinamic
2.	Pengaturan tingkat kesulitan pada setiap materi ujian	1.Mudah 2.Sedang 3.Sulit	1.Mudah 2.Sedang 3.Sulit	Level 1 sampai 5	Dinamic
3.	Penyimpanan soal pada bank soal	Ada	Ada	Tidak ada	Ada
4.	Penilaian bobot soal	Static	Static	Static	Dinamic
5.	Pengaturan soal yang akan di tampilkan	Tidak ada	Ada	Tidak ada	Ada
6.	Pembuatan aturan pelaksanaan ujian	Tidak ada	Ada (pelaksanaan ujian berdasarkan tanggal )	Tidak ada	Ada (pelaksanaan ujian berdasarkan tanggal dan waktu)

No	Fitur	Aplikasi Ujian Test Online	Aplikasi Simulasi Ujian SPMB	Aplikasi Tes UMPB Berbasis Komputer	Aplikasi Ujian Online
7.	Penyajian hasil ujian kepada peserta	Tidak ada	Tidak ada	Tidak ada	Hasil jawaban peserta dan kunci jawaban
8.	Penyajian hasil ujian kepada admin/penguji	Tidak ada	Tidak ada	Tidak ada	Ada, kepada penguji berupa nilai dan peringkat peserta
9	Informasi peringkat	Tidak ada	Tidak ada	Tidak ada	Ada
10	Pengelolaan data peserta	Ada	Ada	Ada	Ada
11.	Pendaftaran peserta	Ada	Ada	Ada	Ada, hanya admin
12.	Profil ujian	Ada	Ada	Ada	Ada, <i>setting</i> dilakukan oleh penguji
13.	Profil organisasi	Tidak ada	Tidak ada	Tidak ada	Tidak ada

### III.2 Spesifikasi Pengembangan Sistem Ujian Online Berbasis Objek

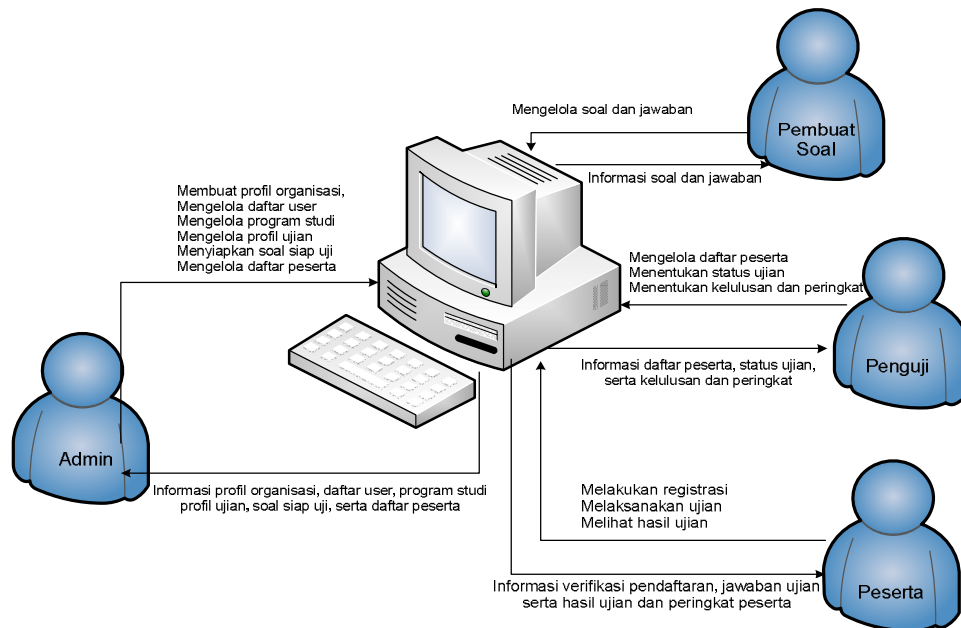
Spesifikasi pengembangan system ujian *online* berbasis objek akan disajikan pada tabel 2.

Tabel 2 Spesifikasi Pengembangan Ujian *Online* Berbasis Objek

No	Fitur	Spesifikasi	Keterangan
1.	Pengaturan subjek	Dinamic	-
2.	Pengaturan tingkat kesulitan pada setiap subjek	Dinamic	-
3.	Penyimpanan soal pada bank soal	Ada	-
4.	Penilaian bobot soal	Dinamic	-
5.	Pengaturan soal yang akan di	Ada	Soal ditampilkan secara random pada setiap peserta.

No	Fitur	Spesifikasi	Keterangan
	tampilkan		
6.	Pembuatan aturan pelaksanaan ujian	Ada	<b>Terdapat pengembangan.</b> Setelah peserta masuk maka akan tampil informasi nama ujian, tipe ujian (UN atau Sertifikasi), keterangan, waktu ujian, nomor ujian, dan program studi yang dipilih (jika ada). Ketika peserta memilih tombol Mulai Ujian, maka akan tampil aturan pelaksanaan ujian.
7.	Penyajian hasil ujian kepada peserta	Ada	<b>Terdapat Pengembangan.</b> Aplikasi dapat menampilkan subjek dan tingkat kesulitannya masing-masing dengan keterangan “LULUS”, “TIDAK LULUS”, dan “BELUM DINILAI”. Jika tipe ujian UN, maka aplikasi dapat menampilkan program studi yang diputuskan lulus oleh manajemen dan penguji. Aplikasi dapat menampilkan hasil jawaban peserta.
8.	Penyajian hasil ujian kepada admin/penguji	Ada	<b>Terdapat Pengembangan.</b> Hasil ujian peserta dapat disajikan kepada admin dan penguji. Isi penyajian hasil ujian kepada admin dan penguji sama seperti penyajian kepada peserta, namun pada admin dan penguji ditampilkan juga nilai pada masing-masing subjek serta tingkat kesulitannya.
9.	Informasi peringkat	Ada	-
10.	Pengelolaan data peserta	Ada	<b>Terdapat Pengembangan.</b> Admin dapat mengubah kata sandi peserta. Hal ini digunakan jika peserta lupa dengan kata sandinya..
11.	Pendaftaran peserta	Ada	<b>Terdapat Pengembangan.</b> Pendaftaran peserta dapat dilakukan sendiri oleh peserta. Peserta juga dapat merubah kata sandinya sendiri.
12.	Profil ujian	Ada	<b>Terdapat Pengembangan.</b> Pengaturan profil ujian dilakukan oleh admin.
13.	Profil organisasi	Ada	<b>Terdapat Pengembangan.</b>

### III.3 Deskripsi Umum Sistem



Gambar 4 Deskripsi Umum Sistem

Gambar 4 menunjukkan deskripsi umum sistem yang menjelaskan tentang pengguna dalam aplikasi ini. Pengguna dalam aplikasi ini dibagi menjadi empat yaitu Admin, Pembuat Soal, Penguji dan Peserta.

Pada awal menggunakan aplikasi ini admin harus melakukan masuk terlebih dahulu untuk mengelola profil organisasi, mengelola daftar *user* (pembuat soal dan penguji) yang akan digunakan untuk menentukan hak akses dari masing-masing *user* tersebut. Selain itu, admin juga dapat mengelola program studi, mengelola profil ujian berupa mengatur subjek masing-masing dengan tingkat kesulitannya dan konfigurasi kelulusan berdasarkan nilai (UN atau Sertifikasi) atau kuota, menyiapkan soal siap uji, serta dapat mengelola daftar peserta.

Pembuat soal dan penguji juga harus melakukan masuk terlebih dahulu. Pembuat soal dapat mengelola soal dan jawaban yang terdiri dari tambah serta ubah soal dan jawaban untuk subjek tertentu sesuai dengan hak akses yang diberikan oleh admin. Kemudian aplikasi akan menampilkan informasi berupa soal dan jawaban.

Penguji dapat mengelola daftar peserta, membuka profil ujian yang telah dibuat oleh admin untuk mengatur beberapa ketentuan-ketentuan seperti menentukan status suatu ujian yang diselenggarakan oleh suatu organisasi (membuka atau menutup ujian). Apabila penguji telah menutup suatu ujian, maka nama ujian tersebut tidak akan tampil dalam formulir registrasi peserta. Selain itu, Penguji juga dapat menentukan kelulusan peserta dan melakukan pengaturan peringkat untuk ujian tertentu sesuai dengan hak akses yang diberikan oleh admin. Kemudian aplikasi akan menampilkan informasi berupa status ujian, daftar peserta serta kelulusan dan peringkat peserta.

Untuk hak akses peserta, sebelum masuk menggunakan aplikasi ini, peserta harus melakukan registrasi terlebih dahulu dengan mengisi formulir pendaftaran. Pada saat mengisi formulir, peserta juga diminta untuk mengisi nama *user* dan kata sandi, dimana nama *user* dan kata sandi ini digunakan untuk masuk pada saat peserta menggunakan aplikasi ini, baik pada saat pelaksanaan ujian maupun melihat hasil ujian peserta. Selain itu, peserta juga dapat merubah kata sandinya sendiri. Pada pengisian formulir, khususnya pada *field* “Pilih Ujian”, peserta dapat memilih salah satu ujian yang diinginkan, kemudian peserta harus memilih tombol “Rinci” untuk mengetahui nama ujian, tipe ujian, batas waktu suatu ujian. Jika tipe ujian UN maka akan tampil formulir tambahan untuk beberapa informasi terkait dengan identitas peserta, kemudian dapat memilih tombol “Simpan”. Namun jika tipe ujian Sertifikasi, setelah peserta memilih tombol “Rinci”, peserta dapat langsung memilih tombol “Simpan”. Setelah mengisi formulir pendaftaran, peserta dapat masuk sesuai dengan nama *user* dan kata sandi pada saat registrasi. Setelah masuk, aplikasi akan menampilkan beberapa informasi untuk ujian yang pernah diikuti dan ujian yang akan diikuti antara lain nama ujian, tipe ujian, keterangan, waktu mulai dan waktu selesai, nomor ujian, program studi yang dipilih. Perbedaannya, pada ujian yang akan diikuti ketika peserta memilih tombol “Mulai Ujian” kedua kali maka akan tampil soal ujian yang harus dikerjakan oleh peserta. Apabila peserta sudah yakin dengan semua jawaban pada soal-soal

tersebut, maka dapat memilih tombol “Selesai Ujian”. Selain itu, peserta juga dapat melihat hasil ujian dan peringkat yang diraih.

### **III.4 Fitur Utama Perangkat Lunak**

Aplikasi ini telah disesuaikan dengan kebutuhan pokok permasalahan yang telah diangkat. Aplikasi dapat digunakan secara umum untuk semua instansi-instansi yang ingin menyelenggarakan ujian *online* untuk berbagai keperluan, misalnya dalam penerimaan mahasiswa baru bagi Perguruan Tinggi, *test* pegawai bagi sebuah perusahaan, dan sebagainya. Hal ini dapat mempermudah dalam proses pelaksanaan ujian karena ujian dilakukan secara *Online*.

### **III.5 Kebutuhan Fungsional**

Kebutuhan Fungsional dalam aplikasi ini adalah :

- |       |  |
|-------|--|
| F-001 | Sistem mampu mengelola informasi profil organisasi   |
| F-002 | Sistem mampu mengelola data <i>user</i>  |
| F-003 | Sistem mampu mengelola program studi   |
| F-004 | Sistem mampu mengelola konfigurasi profil ujian berupa pengaturan subjek beserta tingkat kesulitannya                          |
| F-005 | Sistem mampu mengelola konfigurasi profil ujian berupa pengaturan kelulusan berdasarkan nilai (UN atau sertifikasi) atau kuota |
| F-006 | Sistem mampu membuat aturan distribusi   |
| F-007 | Sistem mampu membuat aturan nilai  |
| F-008 | Sistem mampu membuat soal dan jawaban serta kunci jawaban  |
| F-009 | Sistem mampu menangani registrasi peserta ujian  |
| F-010 | Sistem menyediakan fasilitas konfigurasi pelaksanaan ujian berupa soal siap uji  |

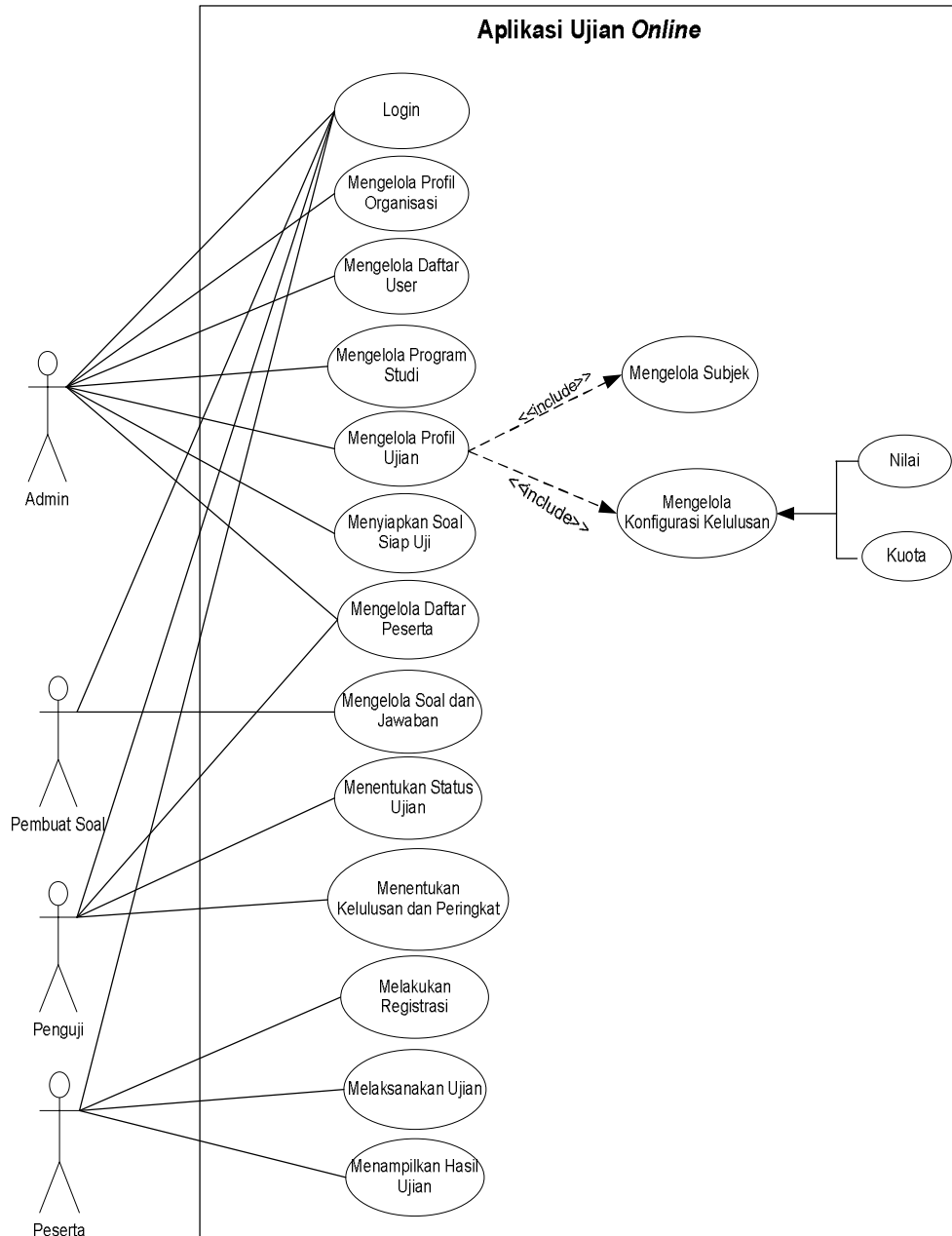
- F-011 Sistem mampu melakukan pengelolaan pelaksanaan ujian berupa pengaturan waktu ujian
- F-012 Sistem menyediakan hak akses soal dan menyimpan jawaban peserta
- F-013 Sistem mampu melakukan kalkulasi nilai peserta
- F-014 Sistem mampu menampilkan informasi kelulusan peserta ujian sesuai konfigurasi yang telah ditentukan (F-004 – F-005).
- F-015 Sistem mampu menampilkan informasi peringkat peserta ujian

### **III.6 Kebutuhan Non Fungsional**

- NF-001 Sistem ini dapat digunakan di PC lain.
- NF-002 Mudah diakses (*online*).

### III.7 Use Case

#### III.7.1 Diagram Use Case



Gambar 5 Use Case Ujian Online

### III.7.2 Skenario Use Case

Gambar 5 menjelaskan tentang interaksi aktor dengan sistem. Aplikasi ini memiliki empat aktor, yaitu Admin, Pembuat Soal, Penguji dan Peserta. Setiap aktor yang akan menggunakan aplikasi ini harus melakukan *login* (masuk) terlebih dahulu, dan apabila ingin mengakhiri penggunaan aplikasi ini harus melakukan *logout* (keluar). Berikut spesifikasi interaksi antara admin dengan sistem, antara lain:

1. Mengelola profil organisasi
2. Mengelola daftar *user*
3. Mengelola program studi
4. Mengelola profil ujian, yaitu mengelola subjek dan tingkat kesulitannya serta mengelola konfigurasi kelulusan berdasarkan nilai (UN atau Sertifikasi) atau kuota.
5. Menyiapkan soal siap uji
6. Mengelola daftar peserta

Interaksi antara pembuat soal dengan sistem adalah pembuat soal hanya mampu mengelola soal dan jawaban.

Berikut spesifikasi interaksi antara penguji dengan sistem antara lain:

1. Mengelola daftar peserta
2. Menentukan status suatu ujian (membuka atau menutup)
3. Menentukan kelulusan dan peringkat peserta

Berikut spesifikasi interaksi Peserta dengan sistem antara lain:

1. Melakukan registrasi
2. Melaksanakan ujian
3. Menampilkan hasil ujian

### III.7.2.1 Use Case Login

- **Aktor** : Admin, Pembuat Soal, Penguji (*User*), dan Peserta
- **Kondisi Awal** : *User* dan peserta memasukkan data berupa nama *user* dan kata sandi
- **Skenario** : 1. Aplikasi akan mengecek apakah data tersebut valid atau tidak  
1. Aplikasi akan menampilkan hak akses *user*
- **Kondisi Akhir** : Tampil hak akses masing-masing *user*

### III.7.2.2 Use Case Mengelola Profil Organisasi

- **Aktor** : Admin
- **Kondisi Awal** : Admin memilih tombol “Profil Organisasi”
- **Skenario** : Admin mengisi formulir pengisian identitas atau profil suatu organisasi yang akan melaksanakan ujian
- **Kondisi Akhir** : Tampil profil organisasi sesuai data yang dimasukkan

### III.7.2.3 Use Case Mengelola Daftar User

- **Aktor** : Admin
- **Kondisi Awal** : Admin memilih tombol “Daftar *User*”
- **Skenario** : Admin dapat menentukan hak akses setiap *user* (admin, pembuat soal dan penguji). Selain itu, admin juga dapat menambah dan mengubah *user* serta dapat mengubah kata sandi.
- **Kondisi Akhir** : Informasi berupa daftar *user* yang menggunakan aplikasi ini sesuai hak aksesnya masing-masing

### III.7.2.4 Use Case Mengelola Program Studi

- **Aktor** : Admin
- **Kondisi Awal** : Admin memilih tombol “Daftar Program Studi”

- **Skenario** : Admin dapat menambah serta mengubah program studi yang akan ditampilkan pada saat peserta melakukan pendaftaran dimana peserta dapat memilih program studi tersebut
- **Kondisi Akhir** : Tampil daftar program studi yang ada pada suatu organisasi yang akan melaksanakan ujian

### III.7.2.5 Use Case Mengelola Profil Ujian

#### III.7.2.5.1 Use Case Mengelola Subjek

- **Aktor** : Admin
- **Kondisi Awal** : Admin memilih tombol “Daftar Subjek”
- **Skenario** : Admin dapat menambah serta mengubah nama subjek dan tingkat kesulitannya
- **Kondisi Akhir** : Tampil daftar subjek dan tingkat kesulitannya

#### III.7.2.5.2 Use Case Mengelola Konfigurasi Kelulusan

- **Aktor** : Admin
- **Kondisi Awal** : Admin memilih tombol “Daftar Profil Ujian”
- **Skenario** :
  1. Admin akan melakukan pengaturan profil ujian berupa spesifikasi ujian yang akan dilaksanakan antara lain nama ujian, tipe ujian, keterangan, waktu mulai dan waktu selesai ujian, serta kuota (jika ada)
  2. Admin juga dapat menghapus profil ujian yang telah dibuat
  3. Admin dapat mencari spesifikasi ujian berdasarkan nama ujiannya
- **Kondisi Akhir** : Tampil daftar profil ujian yang telah diatur. Daftar profil ujian ini juga ditampilkan dalam bentuk *file* dokumen *spreadsheet*

### III.7.2.6 Use Case Mengelola Soal dan Jawaban

- **Aktor** : Pembuat Soal
- **Kondisi Awal** : Pembuat soal memilih tombol “Daftar Soal”
- **Skenario** :
  1. Pembuat soal dapat memasukkan dan menambah serta mengubah soal dan jawaban sesuai dengan subjek dan tingkat kesulitannya
  2. Pembuat soal dapat mencari soal dan jawaban berdasarkan subjek atau subjek dengan tingkat kesulitannya.
- **Kondisi Akhir** : Tampil daftar soal dan jawaban yang telah disimpan. Daftar soal juga ditampilkan dalam bentuk *file* dokumen *spreadsheet*

### III.7.2.7 Use Case Menyiapkan Soal Siap Uji

- **Aktor** : Admin
- **Kondisi Awal** : Admin memilih tombol “Daftar Profil Ujian”
- **Skenario** : Admin mengatur komposisi soal berdasarkan subjek, tingkat kesulitan, jumlah soal, nilai minimum, nilai benar dan nilai salah
- **Kondisi Akhir** : Tampil daftar soal siap uji. Daftar soal siap uji juga ditampilkan dalam bentuk *file* dokumen *spreadsheet*

### III.7.2.8 Use Case Mengelola Daftar Peserta

- **Aktor** : Admin dan Penguji
- **Kondisi Awal** : Admin dan penguji memilih tombol “Daftar Peserta”
- **Skenario** :
  1. Admin dan penguji dapat melihat data-data peserta yang mendaftar untuk melaksanakan ujian
  2. Admin dan penguji dapat mencari jumlah atau data peserta berdasarkan nama ujian

3. Admin juga dapat mengubah kata sandi peserta dimana hal ini dapat dilakukan jika peserta lupa dengan kata sandi yang dibuat
- **Kondisi Akhir** : Tampil daftar peserta. Daftar peserta juga ditampilkan dalam bentuk *file* dokumen *spreadsheet*

### III.7.2.9 Use Case Menentukan Status Ujian

- **Aktor** : Penguji
- **Kondisi Awal** : Penguji memilih tombol “Daftar Profil Ujian”
- **Skenario** : Penguji dapat menentukan status suatu ujian dengan memilih “Buka” berarti masih dibuka atau “Tutup” berarti telah ditutup
- **Kondisi Akhir** : Tampil daftar profil ujian beserta statusnya. Daftar ini juga ditampilkan dalam bentuk *file* dokumen *spreadsheet*

### III.7.2.10 Use Case Menentukan Kelulusan dan Peringkat

- **Aktor** : Penguji
- **Kondisi Awal** : Penguji memilih tombol “Daftar Hasil Ujian”
- **Skenario** :
  1. Penguji dapat menentukan hasil ujian peserta dengan memilih status hasil ujian yaitu “LULUS”, “TIDAK LULUS” atau “BELUM DINILAI” berdasarkan subjek dan tingkat kesulitannya
  2. Penguji dapat mencari hasil ujian peserta berdasarkan nama ujian atau nama ujian dengan hasil ujian
  3. Penguji juga dapat menentukan peringkat peserta berdasarkan nama ujian
- **Kondisi Akhir** : Tampil daftar hasil ujian peserta dan peringkatnya. Daftar hasil ujian juga ditampilkan dalam bentuk *file* dokumen *spreadsheet*

### III.7.2.11 Use Case Melakukan Registrasi

- **Aktor** : Peserta
- **Kondisi Awal** : Peserta memilih tombol “Registrasi Peserta Ujian”
- **Skenario** : Peserta mengisi formulir pendaftaran yang telah disediakan kemudian memilih tombol “Simpan”. Setelah itu peserta dapat masuk menggunakan aplikasi ini
- **Kondisi Akhir** : Tampil daftar ujian yang pernah diikuti dan ujian yang akan diikuti, masing-masing memberikan informasi tentang nama ujian, tipe ujian, keterangan, waktu mulai dan waktu selesai ujian, serta nomor ujian

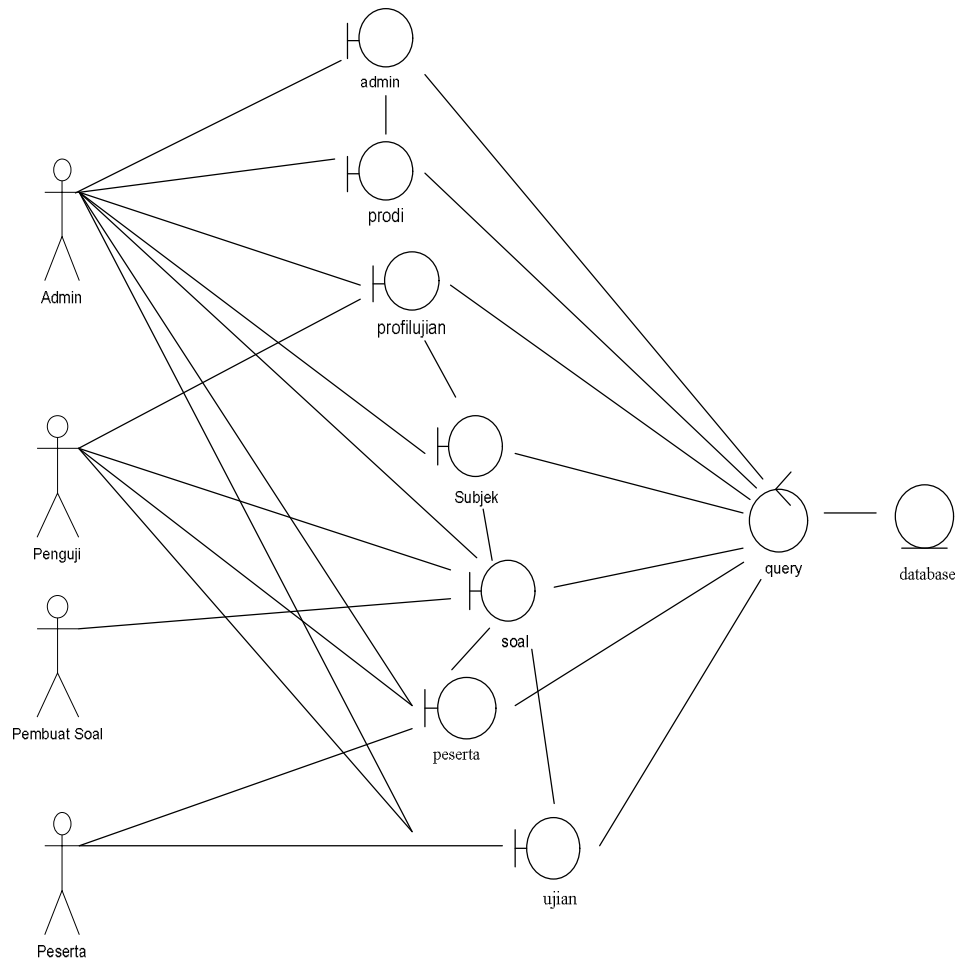
### III.7.2.12 Use Case Melaksanakan Ujian

- **Aktor** : Peserta
- **Kondisi Awal** : Peserta masuk ke aplikasi ini
- **Skenario** :
  1. Peserta dapat memilih tombol “Rinci” untuk mengetahui informasi tentang ujian
  2. Peserta dapat memilih tombol “Mulai Ujian” dan mulai mengerjakan soal-soal yang ditampilkan sesuai batas waktu yang diberikan. Kemudian memilih tombol “Selesai Ujian” apabila peserta sudah yakin dengan jawabannya
- **Kondisi Akhir** : Aplikasi akan menampilkan informasi bahwa peserta telah mengikuti ujian tersebut dan peserta dapat memilih tombol kembali ke daftar ujian untuk mengetahui informasi hasil ujian serta jawaban peserta atau pilih tombol “Keluar” untuk keluar

### III.7.2.13 Use Case Menampilkan Hasil Ujian

- **Aktor** : Peserta
- **Kondisi Awal** : Peserta masuk ke aplikasi ini
- **Skenario** : Setelah masuk, peserta dapat memilih tombol “Daftar Ujian” untuk mengetahui informasi hasil ujian yang diraih. Selain itu peserta juga dapat mengetahui peringkat dalam ujian yang diikuti
- **Kondisi Akhir** : Tampil daftar hasil ujian

### III.8 Analisis Kelas



Gambar 6 Analisis kelas

Gambar 6 menjelaskan tentang analisis kelas pada aplikasi ini. Adapun rincian dari kelas-kelas pada gambar tersebut dijelaskan pada tabel 3, seperti berikut:

Tabel 3 Spesifikasi Analisis Kelas

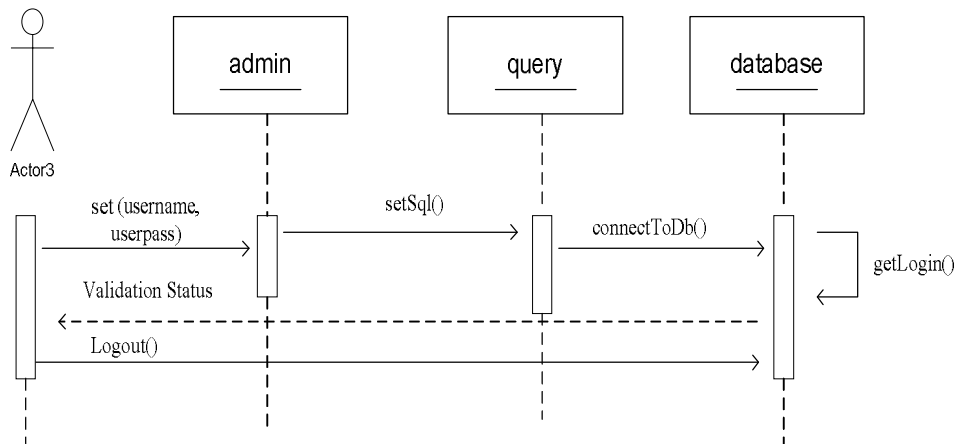
<b>Jenis Kelas</b>	<b>Nama Kelas</b>	<b>Deskripsi</b>
Kelas Boundary	Admin	Menyimpan informasi data <i>user</i> (Admin, Pembuat Soal dan Penguji) serta profil organisasi
	Prodi	Menyimpan informasi program studi pada sebuah organisasi
	Profil Ujian	Menyimpan data profil ujian suatu ujian
	Subjek	Menyimpan nama-nama subjek beserta tingkat kesulitan
	Soal	Menyimpan data soal dan jawaban yang akan ditampilkan pada saat ujian serta kunci jawaban
	Peserta	Menyimpan data peserta yang telah mendaftar ujian <i>online</i>
	Ujian	Menyimpan nama ujian yang diikuti oleh peserta serta jawaban peserta dan hasil ujian peserta
Kelas Control	Query	Mengatur akses ke <i>database</i>
Kelas Entity	Database	Menyimpan dan mengatur informasi dalam sistem

## Bab IV Perancangan

Bab ini menjelaskan tentang *Interaction Diagram*, Diagram Kelas dan Kelas Detail.

### IV.1 Interaction Diagram

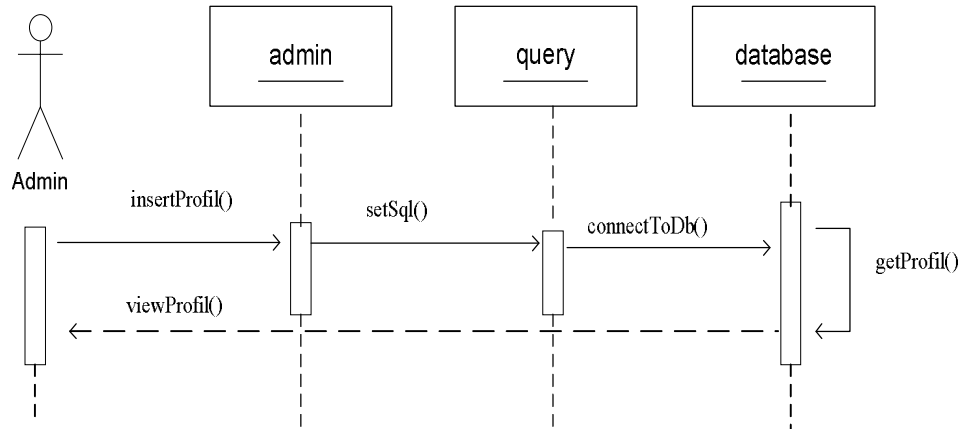
#### IV.1.1 Interaction Diagram Use Case Login



Gambar 7 *Intercation Diagram Use Case Login*

Pada Gambar 7 menjelaskan proses *login*. Setiap *user* (admin, pembuat soal dan penguji) serta peserta yang menggunakan aplikasi ini, maka harus terlebih dahulu *login*. Aplikasi akan mencari nama *user* dan kata sandi yang dimasukkan *user* dan peserta ke kelas admin. Kemudian kelas *query* akan mengkoneksikan ke kelas *database*. Apabila data yang dimasukkan valid, maka aplikasi akan menampilkan layar masing-masing hak akses *user* dan peserta. Di samping itu, setiap akan mengakhiri aktifitas penggunaan aplikasi ini, setiap *user* dan peserta harus melakukan proses *logout* atau keluar.

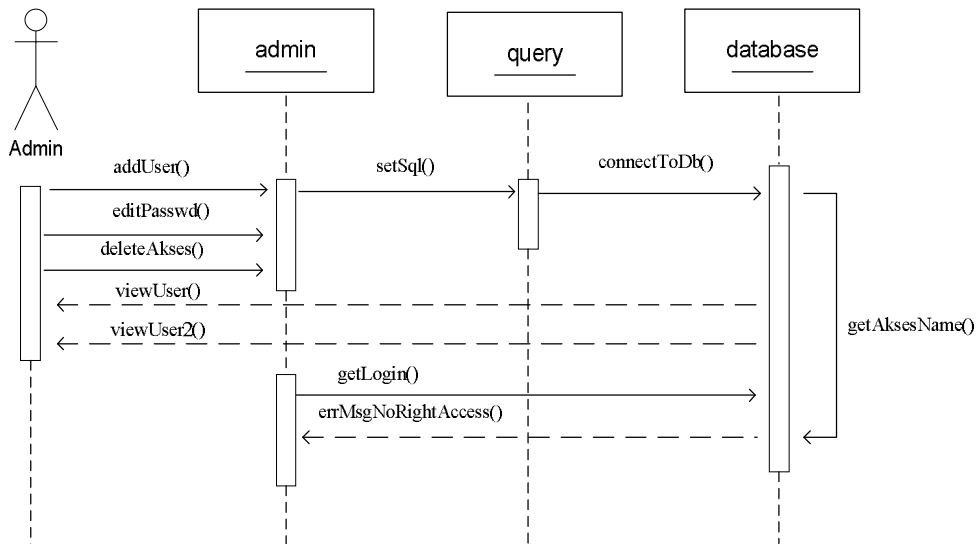
### IV.1.2 Interaction Diagram Use Case Mengelola Profil Organisasi



Gambar 8 *Intercation Diagram Use Case* Mengelola Profil Organisasi

Pada Gambar 8 menjelaskan proses mengelola profil organisasi. Admin akan memasukkan identitas suatu organisasi yang akan melaksanakan ujian *online* ke kelas admin. Kemudian kelas *query* akan mengkoneksikannya ke kelas *database* dan selanjutnya akan tampil profil organisasi yang telah dimasukkan.

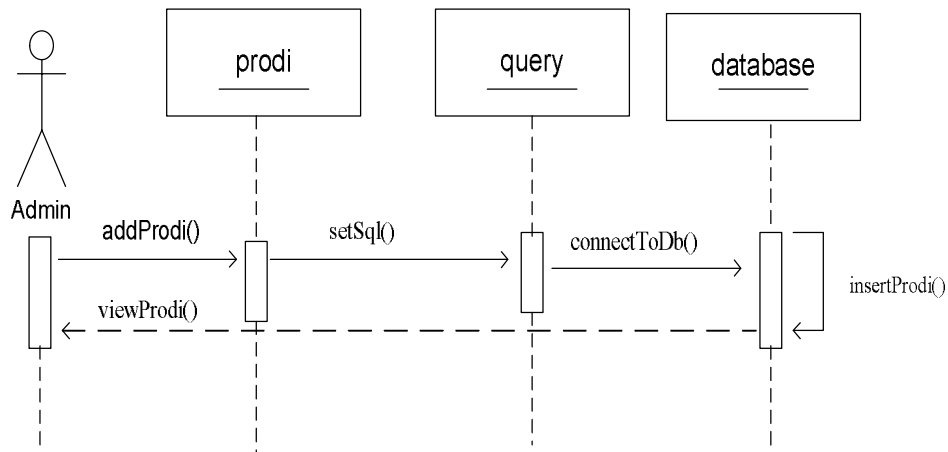
### IV.1.3 Interaction Diagram Use Case Mengelola Daftar User



Gambar 9 *Intercation Diagram Use Case* Mengelola Daftar User

Pada Gambar 9 menjelaskan proses mengelola daftar *user*. Admin akan memberi hak akses kepada dirinya sendiri, pembuat soal dan penguji. Selain itu, admin juga dapat mengubah kata sandi *user*. Hal ini dilakukan apabila *user* (pembuat soal dan penguji) lupa dengan kata sandi yang diberikan admin sebelumnya. Dari pengelolaan ini, maka kelas *query* akan mengkoneksikan ke kelas *database* kemudian akan tampil daftar *user* sesuai hak akses yang diberikan oleh admin, baik secara *general* (*viewUser2*), maupun secara spesifik (*viewUser*). Dari kelas admin, *user* dapat melakukan *login*. *User* hanya dapat mengakses fitur-fitur yang diizinkan oleh admin, jika tidak maka akan tampil pesan *error* “Maaf!! Anda tidak punya hak akses”. Selain itu, admin juga dapat menghapus hak akses *user*.

#### IV.1.4 Interaction Diagram Use Case Mengelola Program Studi

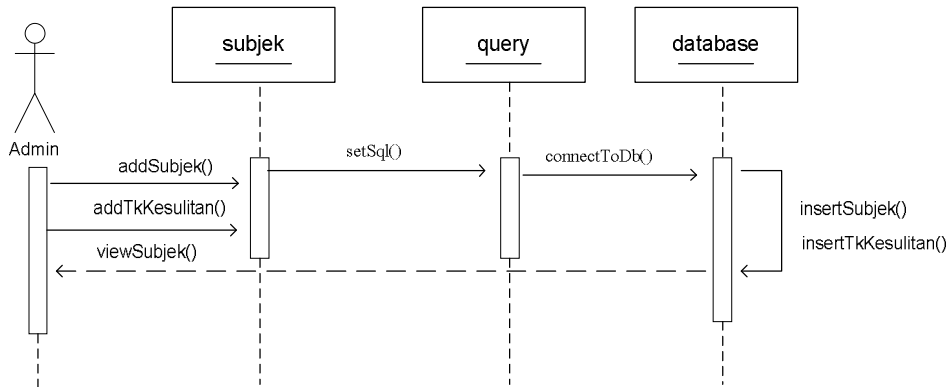


Gambar 10 *Intercation Diagram Use Case Mengelola Program Studi*

Pada Gambar 10 menjelaskan proses mengelola program studi. Admin akan memasukkan program studi apa saja yang akan ditampilkan pada saat peserta mendaftar untuk memilih program studi yang diinginkan dan yang terdapat pada organisasi tersebut. Kemudian kelas *query* akan mengkoneksikan ke kelas *database* dan menyimpannya. Kemudian akan tampil daftar program studi. Program studi hanya akan ditampilkan apabila tipe ujian UN.

## IV.1.5 Interaction Diagram Use Case Mengelola Profil Ujian

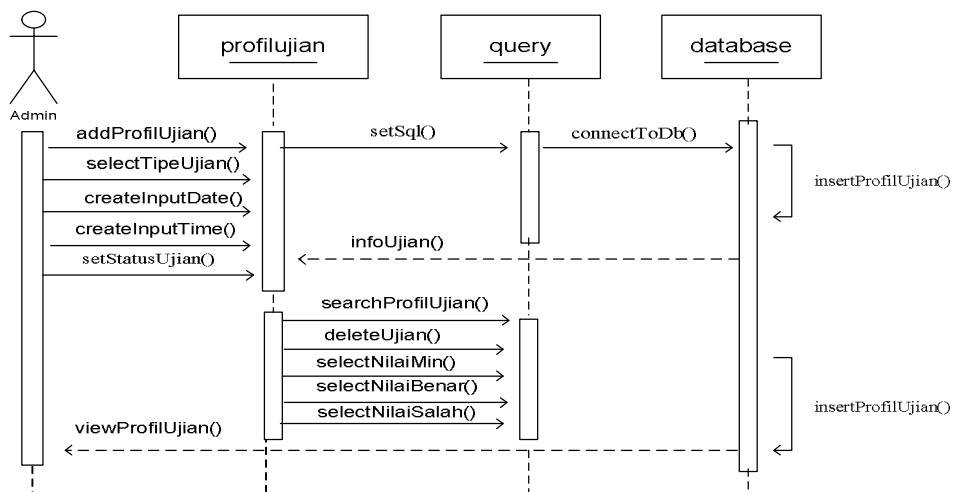
### IV.1.5.1 Interaction Diagram Use Case Mengelola Subjek



Gambar 11 *Intercation Diagram Use Case Mengelola Subjek*

Pada gambar 11 menjelaskan proses mengelola profil ujian pada subjek. Admin akan memasukkan subjek beserta tingkat kesulitan yang akan ditampilkan pada saat ujian. Kelas *query* akan mengkoneksikan ke kelas *database* lalu menyimpannya. Selanjutnya akan tampil daftar subjek dan tingkat kesulitannya.

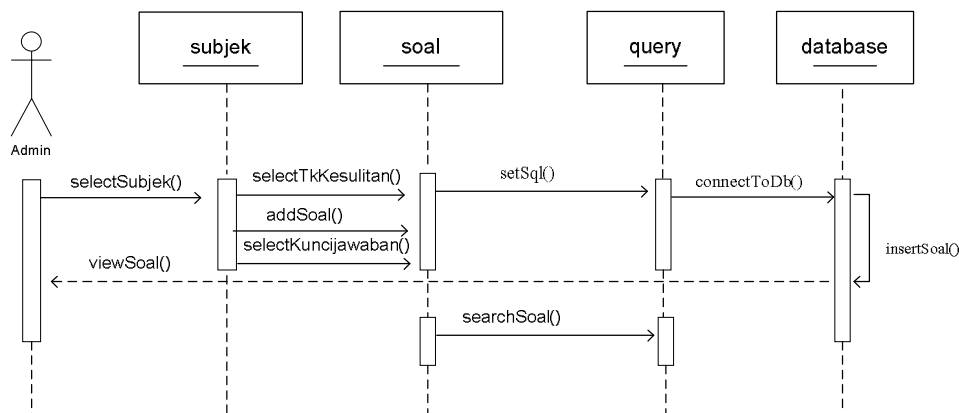
### IV.1.5.2 Interaction Diagram Use Case Mengelola Konfigurasi Kelulusan



Gambar 12 *Intercation Diagram Use Case Mengelola Konfigurasi Kelulusan*

Pada Gambar 12 menjelaskan proses mengelola profil ujian pada konfigurasi kelulusan. Admin akan memasukkan nama ujian, memilih tipe ujian (UN atau Sertifikasi), memasukkan batas tanggal ujian, memasukkan batas waktu ujian serta menyediakan *field* untuk penguji menentukan status suatu ujian. Kelas *query* akan mengkoneksikan ke kelas *database* kemudian menyimpannya. Admin juga dapat mencari suatu ujian berdasarkan namanya dan menghapus ujian. Kemudian akan tampil informasi ujian. Pada saat pengaturan konfigurasi kelulusan, admin juga harus memasukkan nilai minimal, nilai benar dan nilai salah dari suatu subjek dan tingkat kesulitannya kemudian disimpan ke kelas *database* dan akan menampilkan daftar profil ujian.

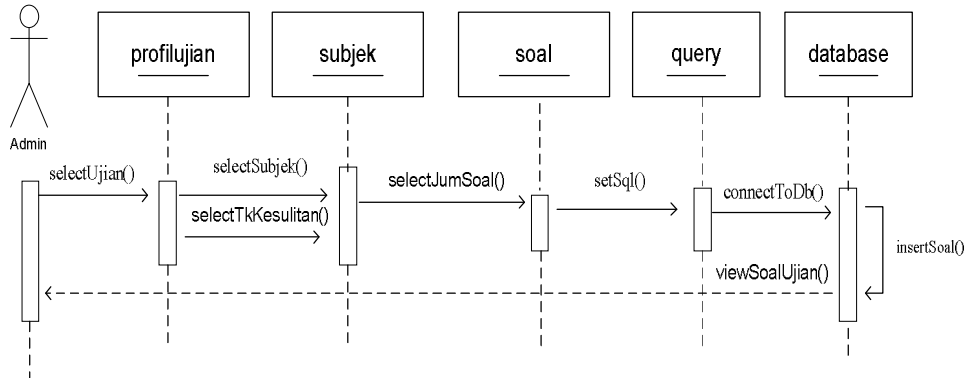
#### IV.1.6 Interaction Diagram Use Case Mengelola Soal dan Jawaban



Gambar 13 *Intercation Diagram Use Case Mengelola Soal dan Jawaban*

Pada Gambar 13 menjelaskan proses mengelola soal dan jawaban. Admin akan memilih subjek dan tingkat kesulitan. Kemudian akan menambahkan atau memasukkan soal pada subjek dan tingkat kesulitan tersebut serta memilih kunci jawaban pada soal tersebut. Selanjutnya akan dikoneksikan ke kelas *database* oleh kelas *query* dan disimpan. Maka akan tampil daftar soal dan jawaban. Admin juga dapat mencari soal berdasarkan subjek atau subjek dengan tingkat kesulitannya.

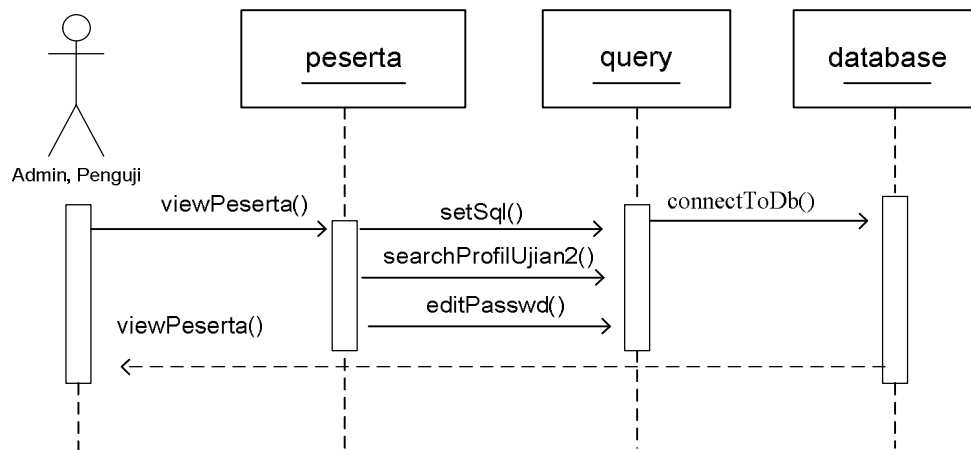
#### IV.1.7 Interaction Diagram Use Case Menyiapkan Soal Siap Uji



Gambar 14 *Interaction Diagram Use Case* Mengelola Soal Siap Uji

Pada Gambar 14 menjelaskan proses menyiapkan soal siap uji. Pertama kali admin harus memilih nama ujian, subjek, tingkat kesulitan dan jumlah soal yang akan ditampilkan dalam satu ujian. Kemudian akan dikoneksikan ke kelas *database* oleh kelas *query* lalu disimpan. Maka akan tampil daftar soal siap uji.

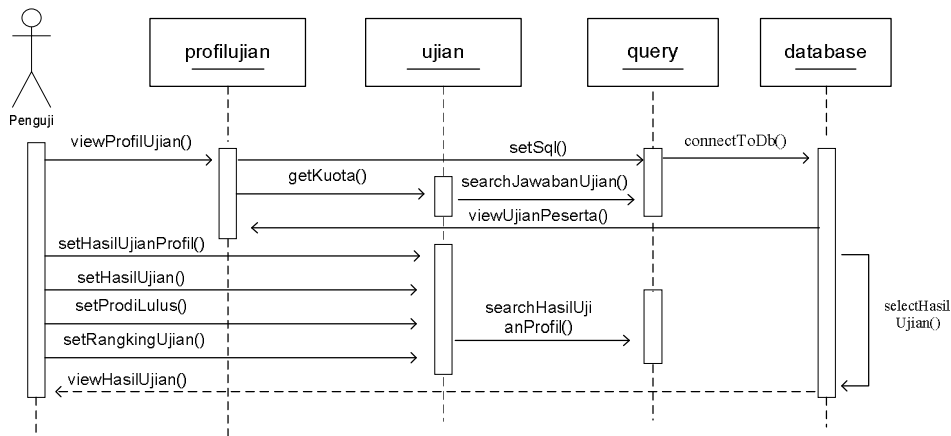
#### IV.1.8 Interaction Diagram Use Case Mengelola Daftar Peserta



Gambar 15 *Interaction Diagram Use Case* Mengelola Daftar Peserta

Pada Gambar 15 menjelaskan proses mengelola daftar peserta. Admin dapat mengubah kata sandi peserta apabila peserta lupa dengan kata sandi sebelumnya. Sedangkan penguji hanya dapat melihat daftar peserta yang mengikuti suatu ujian.

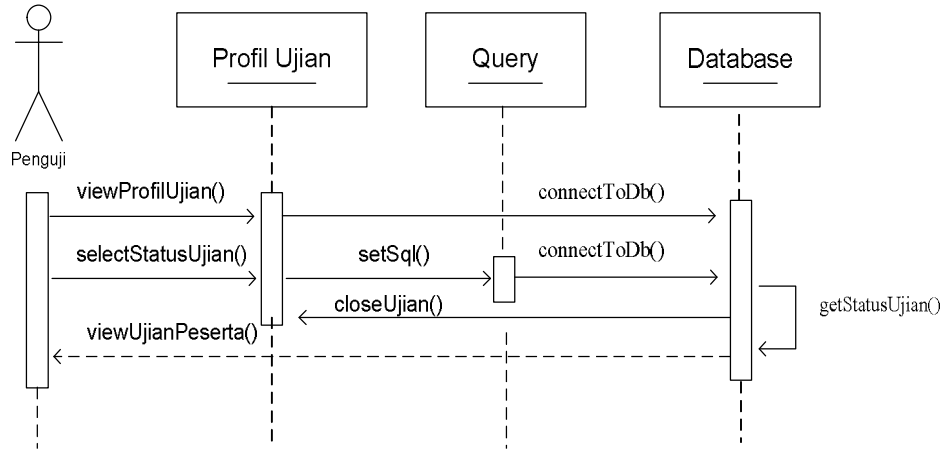
#### IV.1.9 Interaction Diagram Use Menentukan Kelulusan dan Peringkat



Gambar 16 *Intercation Diagram Use Case Melaksanakan Ujian*

Pada Gambar 16 menjelaskan proses menentukan kelulusan dan peringkat. Pertama kali penguji akan melihat profil ujian yang diikuti oleh peserta. Aplikasi secara otomatis akan mencari jawaban peserta dan mengoreksinya. Penguji hanya melakukan pengaturan kelulusan sesuai nilai minimum yang sudah di atur oleh admin pada profil ujian. Selain itu, jika tipe ujian UN, maka penguji dapat menentukan peserta lulus pada salah satu prodi yang dipilih oleh peserta pada saat pendaftaran. Pengaturan peringkat diatur berdasarkan nama ujian, jadi penguji harus mencari berdasarkan nama ujian terlebih dahulu kemudian dapat menentukan peringkat. Selanjutnya seluruh data dikoneksikan ke kelas *database* melalui kelas *query*, kemudian hasil ujian dapat ditampilkan.

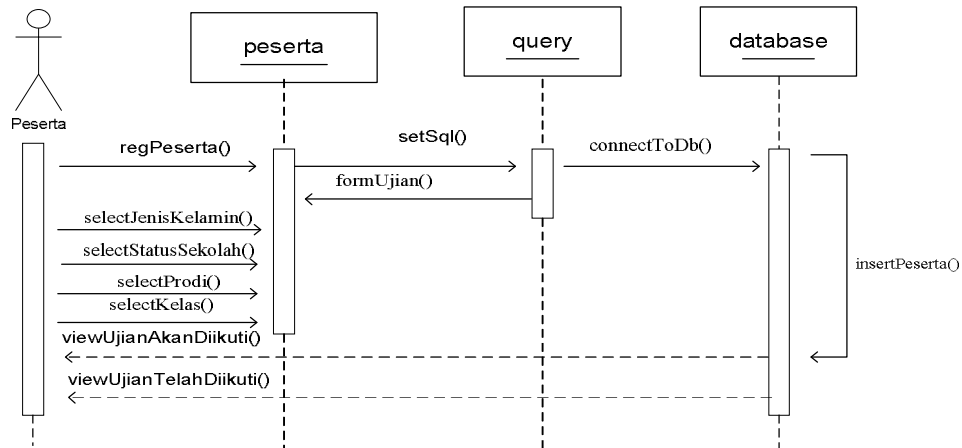
#### IV.1.10 Interaction Diagram Use Case Menentukan Status Ujian



Gambar 17 *Interaction Diagram Use Case Menentukan Status Ujian*

Pada Gambar 17 menjelaskan proses menentukan status suatu ujian. Penguji dapat menentukan status ujian “Buka” berarti masih dibuka atau “Tutup” berarti telah ditutup. Kemudian akan dikoneksikan ke kelas *database* melalui kelas *query* lalu akan tampil status suatu ujian.

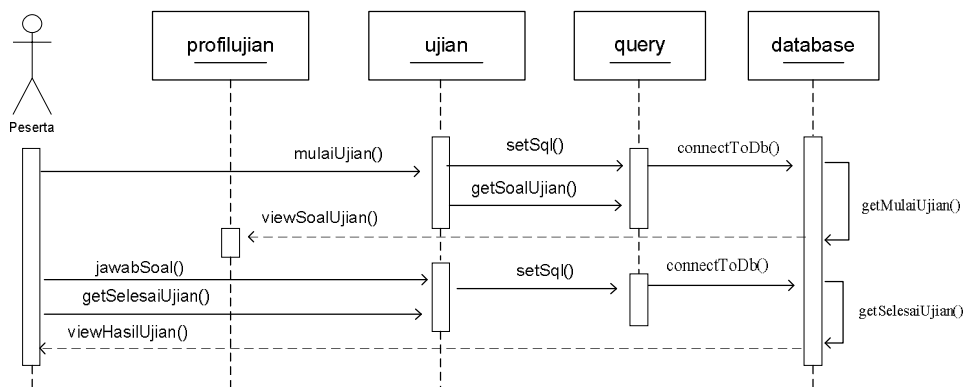
#### IV.1.11 Interaction Diagram Use Case Melakukan Registrasi



Gambar 18 *Interaction Diagram Use Case Melakukan Registrasi*

Pada gambar 18 menjelaskan proses melakukan registrasi. Peserta dapat mengisi formulir yang telah disediakan. Pada pilihan ujian, jika tipe ujian UN, maka akan ada *form* tambahan berupa status sekolah, pilihan prodi, dan jenis kelas. Setelah selesai, maka akan dikoneksikan ke kelas *database* melalui kelas *query* kemudian data-data tersebut disimpan. Maka akan tampil informasi ujian yang akan diikuti dan yang telah diikuti. Informasi tersebut berisi nama ujian, tipe ujian, batas waktu ujian, nomor ujian, serta program studi yang telah dipilih (jika ada).

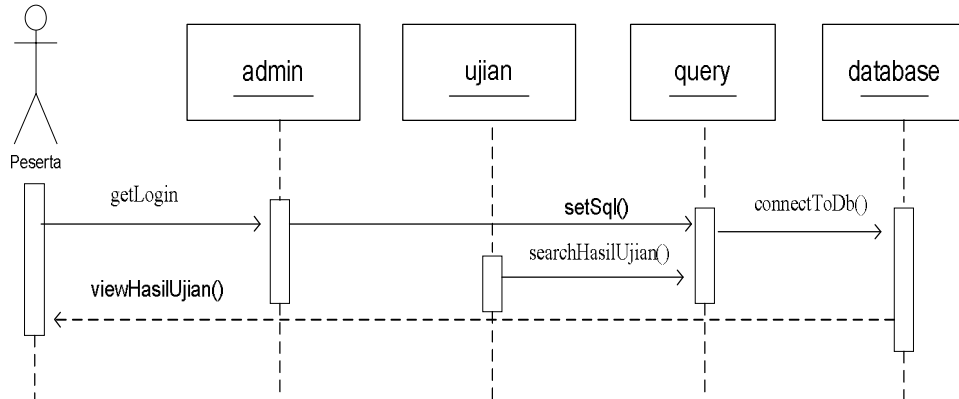
#### IV.1.12 Interaction Diagram Use Case Melaksanakan Ujian



Gambar 19 *Interaction Diagram Use Case Melaksanakan Ujian*

Gambar 19 menjelaskan proses melaksanakan ujian. Peserta dapat memilih tombol “Mulai Ujian” dan akan dikoneksikan ke kelas *database* melalui kelas *query*. Kemudian akan tampil soal-soal yang harus dikerjakan oleh peserta. Selanjutnya peserta dapat menjawab soal-soal tersebut. Apabila peserta sudah selesai dan merasa sudah yakin dengan jawabannya, maka peserta dapat memilih tombol “Selesai Ujian”. Kemudian secara otomatis akan tersimpan ke kelas *database* dan akan ditampilkan hasil ujian peserta berupa hasil jawaban peserta.

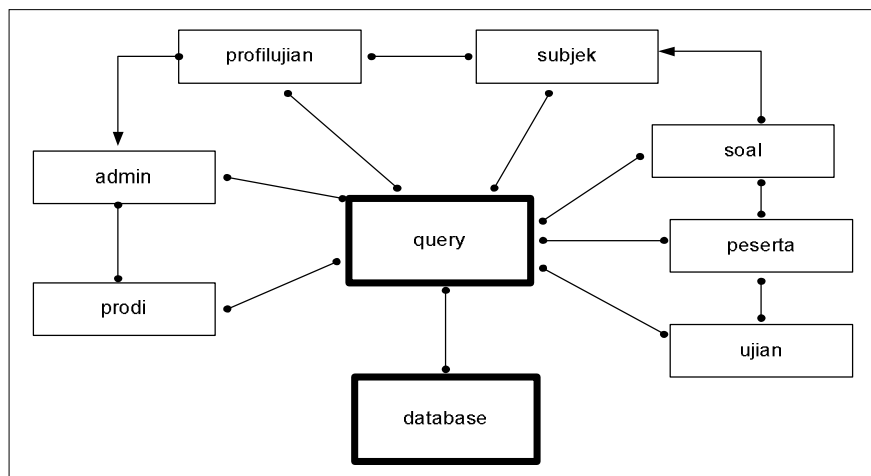
### IV.1.13 Interaction Diagram Use Case Menampilkan Hasil Ujian



Gambar 20 Interaction Diagram Use Case Menampilkan Hasil Ujian

Gambar 20 menjelaskan proses melihat hasil ujian. Peserta dapat melihat hasil ujian setelah peserta melakukan *login* atau masuk ke aplikasi ini. Aplikasi akan secara otomatis mencari hasil ujian peserta kemudian menampilkan ke layar peserta. Informasi hasil ujian tersebut berupa keterangan “LULUS”, “TIDAK LULUS”, atau “BELUM DINILAI”. Selain itu, aplikasi juga akan menampilkan peringkat peserta berdasarkan suatu ujian yang diikuti dan menampilkan program studi yang dinyatakan lulus oleh manajemen dan penguji (jika ada).

## IV.2 Diagram Kelas



Gambar 21 Diagram Kelas

Pada gambar 21 menggambarkan diagram kelas. Setiap kelas yang ada pada aplikasi ini terhubung ke kelas *database* melalui kelas *query*.

### IV.3 Rancangan Kelas Rinci

#### IV.3.1 Kelas Admin

Rancangan rinci setiap kelas disajikan pada tabel 4.

Tabel 4 Rancangan Rinci Setiap Kelas

<b>Admin</b>		
+id_org	:	serial
+title	:	text
+subtitle1	:	text
+subtitle2	:	text
+motd	:	text
+address1	:	text
+address2	:	text
+gbr_header	:	text
+id_int	:	serial
+username	:	text
+userpass	:	text
+nama	:	text
+alamat	:	text
+kontak	:	text
+email	:	text
+jk	:	character(1)
+tgl_lahir	:	date
+user_id	:	integer
+level_akses	:	character varying(10)
+subjek_id	:	integer
+addUser()	:	void
+deleteAkses()	:	void
+editPasswd()	:	void
+errMsgNoRightAccess()	:	void
+getAksesName()	:	character varying(10)
+getLogin()	:	text
+getProfil()	:	text
+getUserId()	:	integer
+insertProfil()	:	void
+insertUser()	:	void
+viewProfil()	:	void
+viewUser()	:	void

+viewUser2()	:	void
--------------	---	------

### IV.3.2 Kelas Prodi

<b>Prodi</b>		
+id	:	serial
+nama	:	text
+ket	:	text
+addProdi()	:	void
+insertProdi()	:	void
+selectProdi()	:	void
+viewProdi()	:	void

### IV.3.3 Kelas Profil Ujian

<b>Profilujian</b>		
+id	:	serial
+profil_ujian_id	:	integer
+user_id	:	integer
+nilai_benar	:	integer
+nilai_salah	:	integer
+nilai_kosong	:	integer
+nilai	:	numeric
+hasil	:	text
+ujian_id	:	integer
+subjek_id	:	integer
+tk_kesulitan	:	integer
+jum_soal	:	integer
+nilai_min	:	integer
+nilai_benar	:	integer
+nilai_salah	:	integer
+nilai_kosong	:	integer
+addProfilUjian()	:	void
+closeUjian()	:	void
+createInputDate()	:	timestamp without time zone
+createInputTime()	:	timestamp without time zone
+deleteUjian()	:	void
+getStatusUjian()	:	integer
+infoUjian()	:	void
+insertProfilUjian()	:	void
+searchProfilUjian()	:	void
+searchProfilUjian2()	:	void
+selectNilaiBenar()	:	integer
+selectNilaiMin()	:	integer
+selectNilaiSalah()	:	integer

+selectStatusUjian()	:	integer
+selectTipeUjian()	:	text
+selectUjian()	:	text
+setStatusUjian()	:	integer
+viewProfilUjian()	:	void
+viewSoalUjian()	:	void
+viewUjianAkanDiikuti()	:	void
+viewUjianPeserta()	:	void
+viewUjianTelahDiikuti()	:	void

#### IV.3.4 Kelas Subjek

<b>Subjek</b>		
+id	:	serial
+nama	:	character varying (255)
+id_tk_kesulitan	:	serial
+nama_tk_kesulitan	:	character varying (255)
+nomor	:	integer
+addSubjek()	:	void
+addTkKesulitan()	:	void
+insertSubjek()	:	void
+insertTkKesulitan()	:	void
+selectSubjek()	:	character varying (255)
+viewSubjek()	:	void

#### IV.3.5 Kelas Soal

<b>Soal</b>		
+id	:	serial
+subjek_id	:	integer
+soal	:	text
+opsi_a	:	text
+opsi_b	:	text
+opsi_c	:	text
+opsi_d	:	text
+opsi_e	:	text
+kunci	:	character varying(1)
+gbr	:	text
+user_id	:	integer
+tk_kesulitan	:	integer
+mp3	:	text
+addSoal()	:	void
+insertSoal()	:	void

+playMp3()	:	void
+playMp3_2()	:	void
+searchSoal()	:	void
+selectJumSoal()	:	integer
+selectKunciJawaban()	:	character(1)
+selectTkKesulitan()	:	integer
+viewSoal()	:	void

### IV.3.6 Kelas Peserta

<b>Peserta</b>		
+id	:	serial
+user_id	:	integer
+nama_sekolah	:	text
+status_sekolah	:	text
+alamat_sekolah	:	text
+tahun	:	character varying(4)
+nilai_ijasah	:	numeric
+rata2_ijasah	:	numeric
+ujian_id	:	integer
+gel	:	integer
+prodi1	:	integer
+prodi2	:	integer
+kelas	:	text
+no_ujian	:	text
+status	:	integer
+hasil	:	text
+prodi	:	integer
+peringkat	:	integer
+formUjian()	:	void
+insertPeserta()	:	void
+regPeserta()	:	void
+selectJenisKelamin()	:	character(1)
+selectKelas()	:	text
+selectProdi()	:	integer
+selectStatusSekolah()	:	text
+viewPeserta()	:	void

### IV.3.7 Kelas Ujian

<b>Ujian</b>		
+id	:	serial
+soal_ujian_id	:	integer
+kunci	:	character(1)

+user_id	:	integer
+ujian_id	:	integer
+profil_ujian	:	integer
+soal_id	:	integer
+startdate	:	timestamp without time zone
+enddate	:	timestamp without time zone
+nama	:	text
+ket	:	text
+status	:	integer
+jenis	:	text
+nilai_benar	:	integer
+nilai_salah	:	integer
+nilai_kosong	:	integer
+tipe_ujian	:	text
+kuota	:	integer
+getKuota()	:	Integer
+getMulaiUjian()	:	timestamp without time zone
+getSelesaiUjian()	:	timestamp without time zone
+jawabSoal()	:	void
+mulaiUjian()	:	timestamp without time zone
+searchHasilUjian()	:	void
+searchHasilUjianProfil()	:	void
+searchJawabanUjian()	:	void
+selectHasilUjian()	:	text
+selesaiUjian()	:	timestamp without time zone
+setHasilUjian()	:	text
+setHasilUjianProfil()	:	integer
+setProdiLulus()	:	integer
+setRangkingUjian()	:	integer
+setTidakLulus()	:	integer
+viewHasilUjian()	:	void

### IV.3.8 Kelas Query

Query		
+sql	:	+integer
+result	:	+integer
+connectToDb()	:	void
+setSql()	:	void
+closeDb()	:	void

### IV.3.9 Algoritma

Bagian ini hanya diisi untuk kerangka algoritma untuk operasi/method yang dianggap cukup penting.

#### IV.3.9.1 Algoritma Pada Kelas Admin

Method : addUser()

```
function addUser()  
    Tampilkan "User Admin"  
    Tampilkan "Tambah/Ubah User Admin"  
    if admin.ifAdmin() Then  
        import modules.profilujian.profil.ujian  
        ProfilUjian=new ProfilUjian(this.config)  
        Panggil Query.connectToDb()  
        column= new array()  
        column[][] <= Query.querySelect()  
        Panggil Query "SELECT * FROM  
            ujian.view_user(Admin.getUserId(), id) "  
  
        Tampilkan field "Nama User"  
        Tampilkan field "Sandi"  
        Tampilkan field "Nama"  
        Tampilkan field "Jenis Kelamin"  
        Tampilkan field "Alamat"  
        Tampilkan field "Kontak"  
        Tampilkan textarea "Email"  
    End if
```

Method : getAksesName()

```
function getAksesName()  
    column = ""  
    Panggil Query.connectToDb()  
    Panggil Query "SELECT ujian.get_akses_name(akses, id) "  
    Panggil Query.querySingle()  
    return column
```

Method : getProfil()

```
function getProfil()  
    Panggil Query.connectToDb()  
    Panggil Query "SELECT ujian.get_profil(name,id)";  
    column [][]<= Query.querySingle()  
    return column
```

Method : viewUser()

```
function viewUser()  
    Tampilkan "Administrasi"  
    Tampilkan "Daftar User Admin"  
    if Admin.ifAdmin() Then  
    Panggil Query.connectToDb()  
    Panggil Query "SELECT * FROM  
        ujian.view_user(Admin.getUser(),id)"  
    jum<=Panggil Query.getNumRows()  
  
    Tampilkan Tombol "Tambah User" onclick=AddUser(0)  
    Tampilkan Tombol "lihat semua user" onclick=ViewUser2()  
    Tampilkan Tombol "Perbaharui" onclick=ViewUser()  
    Tampilkan "Jumlah User Admin" + jum  
    Tampilkan "No" "Nama User" "Nama" "Alamat" "Kontak" "Email"  
        "Jenis Kelamin" "Tanggal Lahir" "Aksi"  
    if jum>0 Then  
    column[] [] <= Panggil Query.querySelect()  
    row<=0;  
    no<=1;  
    while column[row][0] IS NOT NULL Then  
    Loop  
        Tampilkan no  
        Tampilkan column[row][1] // Nama User  
        Tampilkan column[row][2] // Nama  
        Tampilkan column[row][3] // Alamat  
        Tampilkan column[row][4] // Kontak  
        Tampilkan column[row][5] // Email  
        Tampilkan column[row][6] // Jenis Kelamin
```

```

        Tampilkan    column[row][7]    // Tanggal Lahir
        Tampilkan    Tombol                                "Ubah"
                    onclick=AddUser(column[row][0])
        Tampilkan    Tombol                                "Sandi"
                    onclick=EditPasswd(column[row][0])
        no <= no+1
        row <= row+1
    End Loop
    else
        Tampilkan    pesan    "Anda tidak memiliki hak
                        akses!!"
    End if

```

Method : viewUser2()

```

function ViewUser2()
    Tampilkan "Administrasi"
    Tampilkan "Daftar User Admin"
    if Admin.ifAdmin() Then
        Koneksi ke database Query.connectToDb()
        Memanggil    Query    "SELECT * FROM
            ujian.view_user(Admin.getUserId(), id)"
        jum<=Panggil Query.getNumRows()

        Tampilkan Tombol "Tambah User" onclick=AddUser(0)
        Tampilkan Tombol "lihat semua user" onclick=ViewUser2()
        Tampilkan Tombol "Perbaharui" onclick=ViewUser()
        Tampilkan "Jumlah User Admin : "+jum
        Tampilkan "No" "Nama User" "Nama" "Alamat" "Kontak" "Email"
            "Jenis Kelamin" "Tanggal Lahir" "Aksi"
        if jum>0 then
            column[][]Query.querySelect()
            row=0;
            no=1;
            while column[row][0] is not null then
                loop
                Tampilkan column[row][0]    // No
                Tampilkan column[row][1]    // Nama User
            end while
        end if
    end if
end function

```

```

Tampilkan column[row][2]      // Nama
Tampilkan column[row][3]      // Alamat
Tampilkan column[row][4]      // Kontak
Tampilkan column[row][5]      // Email
Tampilkan column[row][6]      // Jenis Kelamin
Tampilkan column[row][7]      // Tanggal Lahir
Tampilkan Tombol "Ubah" onclick=AddUser(column[row][0])
Tampilkan Tombol "Sandi" onclick=EditPasswd(column[row][0])
no++
row+
end loop
else
Tampilkan pesan "Anda Tidak memiliki hak akses!!"
End if

```

### IV.3.9.2 Algoritma Pada Kelas Prodi

Method : addProdi()

```

function addProdi()

    Tampilkan "Program Studi"
    Tampilkan "Tambah/Ubah Program Studi"
    if Admin.ifAdmin() Then
        Panggil Query.connectToDb()
        Panggil Query "SELECT * FROM ujian.select_prodi($id)"
            column<=Panggil Query querySelect()

        Tampilkan field "Nama Program Studi"
        Tampilkan field "Keterangan"
        Tampilkan tombol "Simpan" //untuk menyimpan program studi
        Tampilkan tombol "Tutup" onclick=window.close() //untuk
            menutup jendela tampilan
    else
        Tampil pesan ""Anda tidak memiliki hak akses!!"
    End if

```

Method : viewProdi()

```
function viewProdi()

    Tampilkan "Program Studi"
    Tampilkan "Daftar Program Studi"
    ifAdmin.ifAdmin()
    Tampilkan tombol "Tambah Program Studi" onclick=AddProdi(0)
    Tampilkan tombol "Perbaharui" onclick=ViewProdi()
    Tampilkan "No" //Nomor urut
    Tampilkan "Nama Program Studi"
    Tampilkan "Keterangan"

    Panggil Query.connectToDb()
    Panggil Query "SELECT * FROM ujian.select_prodi(0)"
    Panggil Query.setSql(sql)
    if jum>0 Then
    column[] [] <= Panggil Query.querySelect()
    row<=0
    no<=1;
    while column[row][0] IS NOT NULL Then
    Loop
        Tampilkan column[row][0] //No
        Tampilkan column[row][1] //Nama Program Studi
        Tampilkan column[row][2] // Keterangan
        Tampilkan Tombol "Ubah"
            onclick=AddProdi(column[row][0])
        no <= no+1
        row <= row+1
    End Loop
    else
        Tampilkan pesan "Anda tidak memiliki hak akses!!"
    End if
```

### IV.3.9.3 Algoritma Pada Kelas Profil Ujian

Method : addProfilUjian()

```
function addProfilUjian()  
  
    Tampilkan "Profil Ujian"  
    Tampilkan "Tambah Profil Ujian"  
    if Admin.ifAdmin()  
    Tampilkan "Lihat Profil Ujian" onclick=ViewProfilUjian()  
    Tampilkan "Perbaharui" onclick=AddProfilUjian()  
    Tampil "Profil Ujian" action=index.php  
    Tampilkan field "Nama Ujian"  
    Tampilkan field "Tipe Ujian"  
    Tampilkan field "Keterangan"  
    Tampilkan field "Waktu Mulai"  
    Tampilkan field "Waktu Selesai"  
    Tampilkan field "Kuota"  
    Tampilkan          "Tambah          Komposisi          Soal"  
        onclick=AddKomposisiSoal()  
    Tampilkan field "Subjek"  
    Tampilkan field "Tingkat Kesulitan"  
    Tampilkan field "Jumlah Soal"  
    Tampilkan field "Nilai Minimum"  
    Tampilkan field "Nilai Benar"  
    Tampilkan field "Nilai Salah"  
    Tampilkan tombol "Simpan">//untuk menyimpan profil ujian  
    else  
        Tampilkan pesan "Anda tidak memiliki hak akses!!"  
    End if
```

Method : viewProfilUjian()

```
Function viewProfil Ujian()  
  
    Tampilkan "Daftar Ujian"  
    Tampilkan "Daftar Profil Ujian"  
    If Admin.ifAdmin() OR Admin.ifPenguji() Then
```

```

Tampilkan Tombol "Tambah Profil Ujian"
    OnClick=addProfilUjian()
Tampilkan Tombol "Cari" OnClick=searchProfilUjian()
Tampilkan Tombol "Perbaharui" OnClick=viewProfilUjian()
Panggil searchProfilUjian()
Else
    Tampilkan pesan "Anda tidak punya hak akses!!"
End If

```

Method : viewSoalUjian()

```

function viewSoalUjian()

    ifAdmin.ifAdmin()
    Panggil Query.connectToDb()
    Panggil Query "SELECT
        ujian.search_soal_ujian_numrows(ujian_id)"
    Panggil Query setSql(sql)
    if jum>0 Then
    Panggil Query.querySingle()

    if limit=0;
    Panggil Query "SELECT * FROM ujian.search_soal_ujian(
        ujian_id,limit,offset)

    Panggil Query setSql(sql)
    if jumrows>0 Then
    Panggil Query.Query.getNumRows();
    Tampilkan "Daftar Soal Ujian" onclick=SearchSoalUjian(0,
        ujian_id)
    Tampil pesan "Daftar Soal dibawah ini akan ditampilkan
        secara random pada saat peserta memulai ujian"
    Tampilkan tombol "Lihat Dalam Spreadsheets"
    Tampilkan "No" //nomor urut
    Tampilkan "Subjek"
    Tampilkan "Tingkat Kesulitan"
    Tampilkan "Soal"

```

```

Tampilkan "Opsi A"
Tampilkan "Opsi B"
Tampilkan "Opsi C"
Tampilkan "Opsi D"
Tampilkan "Opsi E"
Tampilkan "Kunci Jawaban"

if jumrows>0 Then
column[] [] <= Panggil Query.querySelect()
no=offset+1;
row=0;
while column[row][0]IS NOT NULL Then
Loop
Tampilkan column[row][0] //nomor urut
Tampilkan column[row][1] //Subjek
Tampilkan column[row][3] //Tingkat Kesulitan
Tampilkan column[row][2] //Soal
Tampilkan column[row][4] //Opsi A
Tampilkan column[row][5] //Opsi B
Tampilkan column[row][6] //Opsi C
Tampilkan column[row][7] //Opsi D
Tampilkan column[row][8] //Opsi E
Tampilkan column[row][9] //Kunci Jawaban
no++;
row++;
End loop
else
Tampil "Tidak ada soal ujian..."

//halaman
if jum>=limit
numRows=ceil(jum);
perPages=limit-1;
numPages=ceil(numRows/limit)

offsetFrst=0;
offsetLast=((numPages-1)*perPages+numPages)-1;

```

```

        offsetPrev=offset-limit;
    if(offsetPrev<0)
        {
            offsetPrev=0;
        }
    offsetNext=offset+limit;

    if(offsetNext>offsetLast)
        {
            offsetNext=offsetLast;
        }
        Tampilkan halaman "First"
        Tampilkan halaman "Prev"
        Tampilkan halaman "Next"
        Tampilkan halaman "Last"

    for(i=1;i<=$numPages;i++)
        {
            //newoffset=((i-1)*perPages+i)-1;
            newoffset=(((i-1)*perPages)+i)-1;
            if(offset==newoffset)
                {
                    Tampil Soal
                }else
                    Tampil Soal
                }
            Tampil baris selanjutnya
        }
        Tampil halaman
    End if

```

#### IV.3.9.4 Algoritma Pada Kelas Subjek

Method : addSubjek()

```

function addSubjek()

    Tampilkan "Subjek"

```

```

Tampilkan "Tambah Subjek"
if Admin.ifAdmin()
Panggil Query.connectToDb()
Panggil Query "SELECT * FROM ujian.select_subjek (id)"
column [] [] <= Panggil Query.querySelect()

Tampilkan field "Nama Subjek"
Tampilkan tombol "Simpan" //untuk menyimpan subjek
Tampilkan tombol "Tutup" onclick=window.close() // untuk
menutup jendela tampilan

else
Tampil pesan "Anda tidak memiliki hak akses!!"
End if

```

Method : addTkKesulitan()

```

function addTkKesulitan()

Tampilkan "Tingkat Kesulitan"
Tampilkan "Tambah Tingkat Kesulitan"
if Admin.ifAdmin()
Panggil Query.connectToDb()
Panggil Query "SELECT * FROM ujian.select_tk_kesulitan(id)"
column [] [] <= Panggil Query.querySelect()

Tampilkan field "Nama Tingkat Kesulitan"
Tampilkan tombol "Simpan" //untuk menyimpan tingkat
kesulitan
Tampilkan tombol "Tutup" onclick=window.close() // untuk
menutup jendela tampilan

else
Tampil pesan "Anda tidak memiliki hak akses!!"
End if

```

Method : viewSubjek()

```
function viewSubjek()

    Tampilkan "Subjek"
    Tampilkan "Daftar Subjek"
    if Admin.ifAdmin()
    Tampilkan Tombol "Tambah Subjek" onclick=AddSubjek(0)
    Tampilkan Tombol "Perbaharui" onclick=ViewSubjek()
    Tampilkan "No" //nomor urut
    Tampilkan "Nama Subjek"
    Tampilkan "Jumlah Soal Tersedia"
    Panggil Query.connectToDb()
    Panggil Query "SELECT * FROM ujian.view_subjek()"
    Panggil Query.setSql(sql)
    if jum>0 Then
    column[] [] <= Panggil Query.querySelect()
    row<=0
    no<=1;
    while column[row][0] IS NOT NULL Then
    Loop
        Tampilkan column[row][0] //No
        Tampilkan column[row][1] //Nama Subjek
        Tampilkan column[row][2] //Jumlah Soal Tersedia
        Tampilkan Tombol "Ubah"
            onclick=AddSubjek(column[row][0])
        row++;
        no++;
    End loop
    else
        Tampil "Tidak ada baris"
    else
        Tampil pesan tidak memiliki hak akses
    End if
    Tampilkan "Tingkat Kesulitan"
    Tampilkan "Daftar Tingkat Kesulitan"
    if Admin.ifAdmin()
    Tampilkan Tombol "Tambah Tingkat Kesulitan"
```

```

        onclick=AddTkKesulitan(0)
Tampilkan Tombol "Perbaharui" onclick= ViewSubjek()
Tampilkan "No" //nomor urut
Tampilkan "Nama Tingkat Kesulitan"
Tampilkan "Nomor Tingkat Kesulitan"
Panggil Query.connectToDb()
Panggil Query "SELECT * FROM
        ujian.select_tk_kesulitan(0)"
Panggil Query.setSql(sql)
if jum>0 Then
column[] [] <= Panggil Query.querySelect()
row<=0
no<=1;
while column[row][0] IS NOT NULL Then
Loop
Tampilkan column[row][0] //No
Tampilkan column[row][1] //Nama Tingkat Kesulitan
Tampilkan column[row][2]//Nomor Tingkat Kesulitan
Tampilkan Tombol "Ubah"
        onclick=AddTkKesulitan(column[row][0])
row++;
no++;
End loop
else
        Tampil "Tidak ada baris"
else
        Tampil pesan "Anda tidak memiliki hak akses!!"
End if

```

### IV.3.9.5 Algoritma Pada Kelas Soal

Method : addSoal()

```

function addSoal()
import modules subjek.subjek
Subjek=new Subjek(this.config);
Tampilkan "Soal"

```

```

Tampilkan "Menambah Soal"
if Admin.ifAdmin() OR Admin.ifPembuatSoal() Then
    column= new array()
    Buka Koneksi ke database
    Query.connectToDb()
    Memanggil                                Query"SELECT*FROM
        ujian.search_soal(Admin.getUserId(),id)"
    column [][] <= Panggil querySelect()
    Pilih Nama Subjek
    Pilih Tingkat Kesulitan
    Masukkan Soal
    Masukkan Gambar
    Masukkan Suara
    Masukkan Opsi A
    Masukkan Opsi B
    Masukkan Opsi C
    Masukkan Opsi D
    Masukkan Opsi E
    Pilih Kunci Jawaban
else
    Tampil errMsg "Anda Tidak Punya Hak Akses!!"
End if

```

Method : viewSoal()

```

function viewSoal()
    Tampilkan "Soal"
    Tampilkan "Daftar Soal"
    if Admin.ifAdmin() OR Admin.ifPembuatSoal() Then
    import modules subjek.subjek
    Subjek=new Subjek(this.config)
    Tampilkan "Menu"
    if Admin.ifPembuatSoal() Then
    Tampilkan Tombol "Tambah Soal" onclick=AddSoal(0)
        Pilih Subjek
        Pilih TkKesulitan
    Tampilkan Tombol "Cari" onclick=SearchSoal(0)
    Tampilkan Tombol "Perbaharui" onclick=ViewSoal()

```

```
Else
    Tampil ErrMsg "Anda Tidak memiliki hak akses!!"
End if
```

#### IV.3.9.6 Algoritma Pada Kelas Peserta

Method : regPeserta()

```
function regPeserta()
    Import modules profilujian.profil.ujian
    ProfilUjian=new ProfilUjian(this.config)
    Tampilkan "Peserta Ujian"
    Tampilkan "Registrasi Peserta Ujian"
    Tampilkan "Setelah proses registrasi, silahkan login untuk
        mengetahui nomor ujian Anda dan info mengenai ujian yang
        akan Anda ikuti"
    Masukkan Nama User
    Masukkan Nama
    Pilih Jenis Kelamin
    Pilih Tanggal Lahir
    Masukkan Alamat
    Masukkan Kontak
    Masukkan Email
    Pilih Ujian
```

Method : viewPeserta()

```
function viewPeserta()
    Tampilkan "Peserta Ujian"
    Tampilkan "Daftar Peserta Ujian"
    if Admin.ifAdmin() OR Admin.ifPenguji() Then
    Import modules.profilujian.profil.ujian
    ProfilUjian=new ProfilUjian(this.config)
    Tampil Tombol "Tambah Peserta" onclick=RegPeserta()
    Tampil Tombol "Cari" onclick=SearchPeserta(0)
    Tampil Tombol "Perbaharui" onclick='ViewPeserta()
    Else
    Tampil errMsg "Anda Tidak Memiliki Hak akses!!"
```

```
End if
```

### IV.3.9.7 Algoritma Pada Kelas Ujian

Method : getMulaiUjian()

```
function getMulaiUjian()

    column=0;
    if(user_id==0) {
        user_id=$_SESSION['the_user_id_uo'];
    }
    if Admin->ifPeserta(user_id,ujian_id)
    Panggil Query.connectToDb()
    Panggil Query "SELECT ujian.mulai_ujian(user_id,ujian_id)"

    Panggil Query.setSql(sql)
    column [] [] <= Panggil Query.querySingle()
    Panggil Query.closeDb()

    // -1 : ujian belum selesai ttp peserta tlh menyelesaikan
        ujiannya
    // 0 : belum dapat dimulai
    // 1 : sedang berlangsung
    // 2 : telah selesai
    return $column;
End if
```

Method : getSelesaiUjian()

```
function getSelesaiUjian()

    if Admin->ifPeserta(user_id,ujian_id)
    Panggil Query.connectToDb()
    Panggil Query "SELECT
        ujian.selesai_ujian(user_id,ujian_id)"
    Panggil Query.setSql(sql)
    Panggil Query "SELECT
```

```

        ujian.penilaian_ujian(user_id,ujian_id)"
    Panggil Query.setSql(sql)
    Panggil Query.closeDb()
    End if

```

Method : jawabSoal()

```

function jawabSoal()

    if(user_id==0)
    Panggil user_id=Admin->getUserid()
    if(ujian_id==0)
    Panggil ujian_id=Peserta->getUjianId()

    Panggil Query.connectToDb()
    Panggil Query "SELECT
        ujian.jawab_soal(id,ujian_id, ".user_id.", 'kunci')
    Panggil Query.closeDb();
    End if

```

Method : setHasilUjian()

```

function setHasilUjian()

    if (Admin->ifLogin() || Admin->ifAdmin() || Admin-
        >ifPenguji())
    Panggil Query.connectToDb()
    Panggil Query "SELECT ujian.set_hasil_ujian(".Admin-
        >getUserid().", id, 'hasil')
    Panggil Query.closeDb()
    End if

```

Method : setTidakLulus()

```

function setTidakLulus()

    if(Admin->ifLogin() || Admin->ifAdmin() || Admin-
        >ifPenguji())
    Panggil Query.connectToDb()

```

```

Panggil Query "SELECT
    ujian.set_tidak_lulus(ujian_id, ".this->Admin-
    >getUserId().")"
Panggil Query.closeDb();
End if

```

Method : viewHasilUjia()

```

function viewHasilUjian()

    Tampilkan "Ujian"
    Tampilkan "Daftar Hasil Ujian"
    if(Admin.ifAdmin() || Admin->ifPenguji())
        Tampilkan ujian
        Pilih hasil ujian

        Tampilkan Tombol "Cari" onclick=SearchHasilUjian(0)
        Tampilkan Tombol "Perbaharui" onclick=ViewHasilUjian()

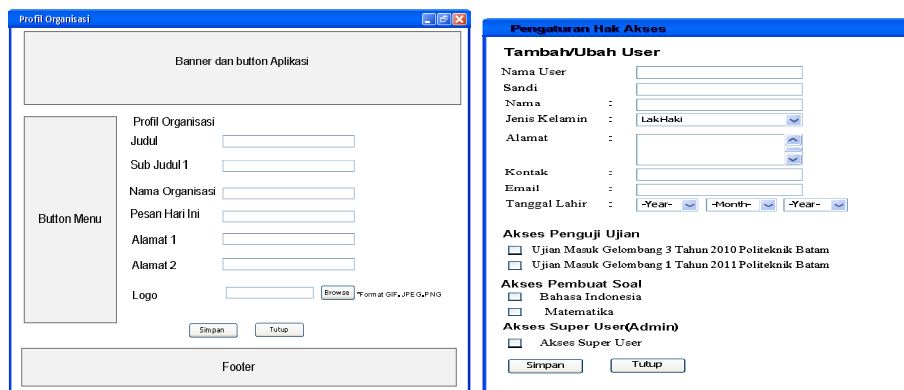
    else
        Tampil pesan "Anda tidak memiliki hak akses!!"
    End if

```

## IV.4 Perancangan Antarmuka

### IV.4.1 Admin

#### IV.4.1.1 Rancangan Tampilan



Gambar 22 Rancangan Tampilan Profil Organisasi dan Pengaturan Hak Akses

#### IV.4.1.2 Deskripsi

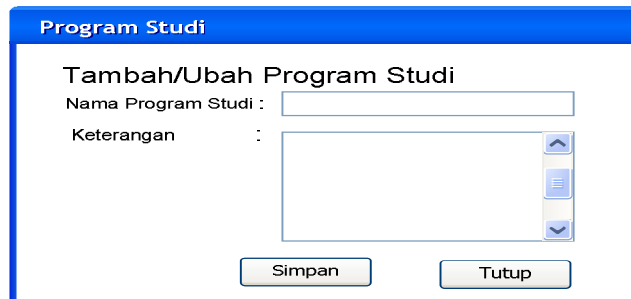
Deskripsi rancangan tampilan profil organisasi dan pengaturan hak akses *user* disajikan pada tabel 5.

Tabel 5 Deskripsi Rancangan Tampilan Profil Organisasi dan Pengaturan Hak Akses *User*

<b>Id_Objek</b>	<b>Jenis</b>	<b>Nama</b>	<b>Keterangan</b>
Banner	Banner	Banner	Menampilkan banner aplikasi yang berisi gambar yang dimasukkan.
Button	Button	Button	Button-button yang akan mengacu ke file yang terhubung dengan aplikasi.
TF	Text Field	tittle	Memasukkan judul aplikasi
TF2	Text Field	subtitle1	Memasukkan sub bagian judul aplikasi
TF3	Text Field	subtitle2	Memasukkan nama profil organisasi
TF4	Text Field	motd	Memasukkan info tentang profil organisasi
TF5	Text Field	address1	Memasukkan alamat organisasi
TF6	Text Field	address2	Memasukkan tambahan alamat organisasi
Button2	Button	gbr_header	Memasukkan gambar yang akan tampil pada <i>header</i> aplikasi ini
TF7	Text Field	username	Memasukkan <i>username user</i>
TF8	Text Field	userpass	Memasukkan <i>password user</i>
TF9	Text Field	nama	Memasukkan nama <i>user</i>
CB	Combo Box	jk	Memilih jenis kelamin <i>user</i>
TF10	Text Field	alamat	Memasukkan alamat <i>user</i>
TF11	Text Field	kontak	Memasukkan nomor telepon atau <i>handphone user</i>
TF12	Text Field	email	Memasukkan <i>email user</i>
CB2	Combo Box	tgl_lahir	Memilih tanggal, bulan dan tahun lahir <i>user</i>
Check	Check Box	level_akses	Memilih hak akses <i>user</i>
Button3	Button	getAksesName	Jika di <i>klik</i> , maka hak akses tersimpan di <i>t_user_akses</i>

## IV.4.2 Program Studi

### IV.4.2.1 Rancangan Tampilan



The screenshot shows a window titled "Program Studi" with a subtitle "Tambah/Ubah Program Studi". It contains two input fields: "Nama Program Studi" (a single-line text field) and "Keterangan" (a multi-line text area). Below the text area are two buttons: "Simpan" and "Tutup".

Gambar 23 Rancangan Tampilan Program Studi

### IV.4.2.2 Deskripsi

Deskripsi rancangan tampilan program studi disajikan pada tabel 6.

Tabel 6 Deskripsi Rancangan Tampilan Program Studi

<b>Id_Objek</b>	<b>Jenis</b>	<b>Nama</b>	<b>Keterangan</b>
TF13	Text Field	nama	Memasukkan nama program studi
TF14	Text Field	ket	Memasukkan keterangan dari nama program studi yang dimasukkan
Button4	Button	insertProdi	Jika di klik, maka data tersebut akan disimpan ke t_prodi

## IV.4.3 Subjek

### IV.4.3.1 Rancangan Tampilan



The image shows two separate form windows. The left window is titled "Daftar Subjek" and "Tambah/Ubah subjek". It has a text field labeled "Nama Subjek" and two buttons: "Simpan" and "Tutup". The right window is titled "Tingkat Kesulitan" and "Tambah/Ubah Tingkat Kesulitan". It has a text field labeled "Nama Tingkat Kesulitan" and two buttons: "Simpan" and "Tutup".

Gambar 24 Rancangan Tampilan Subjek dan Tingkat Kesulitan

### IV.4.3.2 Deskripsi

Deskripsi rancangan tampilan subjek dan tingkat kesulitan disajikan pada tabel 7.

Tabel 7 Deskripsi Rancangan Tampilan Subjek dan Tingkat Kesulitan

<b>Id_Objek</b>	<b>Jenis</b>	<b>Nama</b>	<b>Keterangan</b>
TF15	Text Field	nama	Memasukkan nama subjek yang akan ditampilkan pada pembuat soal untuk memasukkan soal
Button5	Button	insertSubjek	Apabila <i>diklik</i> , maka subjek akan disimpan ke t_subjek
TF16	Text Field	nama	Memasukkan tingkat kesulitan yang akan ditampilkan pada saat memilih subjek
Button6	Button	insertTkKesulitan	Apabila <i>diklik</i> , maka tingkat kesulitan akan disimpan ke t_tk_kesulitan

### IV.4.4 Profil Ujian

#### IV.4.4.1 Rancangan Tampilan

Profil Ujian

Profil Ujian

Nama Ujian :

Type Ujian : UN

Keterangan :

Waktu Mulai : -Tahun- -Bulan- +Hari- +waktu-

Waktu Selesai : -Tahun- -Bulan- +Hari- +waktu-

Komposisi Soal dan Penilaian

Tambah Komposisi Soal

Subjek	Tingkat Kesulitan	Jumlah Soal	Nilai Minimum(%)	Nilai Benar	Nilai Salah

Simpan

Gambar 25 Rancangan Tampilan Profil Ujian

#### IV.4.4.2 Deskripsi

Deskripsi rancangan tampilan profil ujian disajikan pada tabel 8.

Tabel 8 Deskripsi Rancangan Tampilan Profil Ujian

<b>Id_Objek</b>	<b>Jenis</b>	<b>Nama</b>	<b>Keterangan</b>
TF17	Text Field	nama	Memasukkan nama ujian yang akan ditampilkan
CB3	Combo Box	tipe_ujian	Memilih tipe ujian sesuai nama ujian
TF18	Text Field	ket	Memasukkan keterangan dari nama ujian yang dimasukkan
CB4	Combo Box	stastdate	Memilih waktu mulai pelaksanaan ujian
CB5	Combo Box	enddate	Memilih waktu selesai pelaksanaan ujian
CB6	Combo Box	nama	Memilih subjek yang akan ditampilkan ke peserta pada saat ujian
CB7	Combo Box	nama	Memilih tingkat kesulitan subjek
CB8	Combo Box	jum_soal	Memilih jumlah soal yang akan ditampilkan ke peserta
CB9	Combo Box	nilai_min	Memilih nilai minimum setiap subjek dan stiap tingkat kesulitan
CB10	Combo Box	nilai_benar	Memilih nilai benar setiap subjek pada tingkat kesulitannya
CB11	Combo Box	nilai_salah	Memilih nilai salah setiap subjek pada tingkat kesulitannya
Button7	Button	insertProfil Ujian	Apabila diklik, maka profil ujian tersebut akan disimpan ke t_profil_ujian

## IV.4.5 Soal

### IV.4.5.1 Rancangan Tampilan

The screenshot shows a web form titled "Soal" with the sub-header "Menambah Soal". The form contains the following elements:

- Nama Subjek**: A dropdown menu with the placeholder text "-Pilih Subjek-".
- Tingkat kesulitan**: A dropdown menu with the placeholder text "-Pilih Tk.Kesulitan-".
- Soal**: A large text area for entering the question text.
- Suara**: A text input field followed by a "Browse" button for uploading an audio file.
- Opsi A**, **Opsi B**, **Opsi C**, **Opsi D**, and **Opsi E**: Five separate text input fields for entering multiple-choice options.
- Kunci Jawaban**: A dropdown menu with the placeholder text "-Pilih Kunci Jawaban-".
- At the bottom of the form are two buttons: "Simpan" (Save) and "tutup" (Close).

Gambar 26 Rancangan Tampilan Soal dan Jawaban

### IV.4.5.2 Deskripsi

Deskripsi rancangan tampilan soal dan jawaban disajikan pada tabel 9.

Tabel 9 Deskripsi Rancangan Tampilan Soal dan Jawaban

<b>Id_Objek</b>	<b>Jenis</b>	<b>Nama</b>	<b>Keterangan</b>
CB12	Combo Box	nama	Memilih subjek
CB13	Combo Box	nama	Memilih tingkat kesulitan
Button8	Button	mp3	Memasukkan soal ujian dalam bentuk file suara
TF19	Text Field	soal	Memasukkan pertanyaan atau soal
TF20	Text Field	opsi_a	Memasukkan pilihan jawaban a
TF21	Text Field	opsi_b	Memasukkan pilihan jawaban b
TF22	Text Field	opsi_c	Memasukkan pilihan jawaban c
TF23	Text Field	opsi_d	Memasukkan pilihan jawaban d
TF24	Text Field	opsi_e	Memasukkan pilihan jawaban e
CB14	Combo Box	kunci	Memilih kunci jawaban dalam soal

<b>Id_Objek</b>	<b>Jenis</b>	<b>Nama</b>	<b>Keterangan</b>
			tersebut
Button8	Button	insertSoal	Apabila <i>diklik</i> , maka soal tersebut akan disimpan ke t_soal

## IV.4.6 Peserta

### IV.4.6.1 Rancangan Tampilan

Gambar 27 Rancangan Tampilan Peserta

### IV.4.6.2 Deskripsi

Deskripsi rancangan tampilan peserta disajikan pada tabel 10.

Tabel 10 Deskripsi Rancangan Tampilan Peserta

<b>id_Objek</b>	<b>Jenis</b>	<b>Nama</b>	<b>Keterangan</b>
F25	Text Field	username	Memasukkan <i>username</i> peserta
TF26	Text Filed	userpass	Memasukkan <i>password</i> peserta
TF27	Text Field	nama	Memasukkan nama peserta
CB15	Combo Box	jk	Memilih jenis kelamin peserta
CB16	Combo Box	tgl_lahir	Memilih tanggal, bulan dan tahun lahir peserta
F28	Text Field	alamat	Memasukkan alamat peserta
F29	Text Field	kontak	Memasukkan nomor telepon atau

id_Objek	Jenis	Nama	Keterangan
			<i>handphone</i> peserta
TF30	Text Field	email	Memasukkan alamat <i>email</i> peserta
CB17	Combo Box	nama	Memilih nama ujian yang diinginkan.
Button9	Button	formUjian	Apabila <i>diklik</i> , maka akan tampil form tambahan. Form ini akan tampil jika tipe ujian UN
TF31	Text Field	nama_sekolah	Memasukkan nama sekolah peserta
CB18	Combo Box	status_sekolah	Memilih status sekolah peserta
TF32	Text Field	alamat_sekolah	Memasukkan alamat sekolah peserta
TF33	Text Field	nilai_ijasah	Memasukkan nilai ijasah
TF34	Text Field	rata2_ijasah	Memasukkan nilai rata-rata ujian
TF35	Text Field	tahun	Memasukkan tahun lulus peserta
CB19	Combo Box	prodi1	Memilih program studi 1 peserta
CB20	Combo Box	prodi2	Memilih program studi 2 peserta yang dianggap sebagai alternatif dari program studi 1
CB21	Combo Box	kelas	Memilih jenis kelas yang diinginkan peserta
Button10	Button	regPeserta	Jika <i>diklik</i> , maka data-data tersebut akan disimpan ke <i>t_profil_peserta</i>

## IV.4.7 Ujian

### IV.4.7.1 Rancangan Tampilan

**Hasil Ujian**

Menu  Nama Ujian

Jumlah Ujian : 2

No	Nama Ujian	Keterangan	Tipe Ujian	Waktu Mulai	Waktu Selesai	Kuota	Status	Aksi
								<input type="button" value="Buka"/> <input type="text" value="v"/>

Gambar 28 Rancangan Tampilan Penentuan Status Ujian

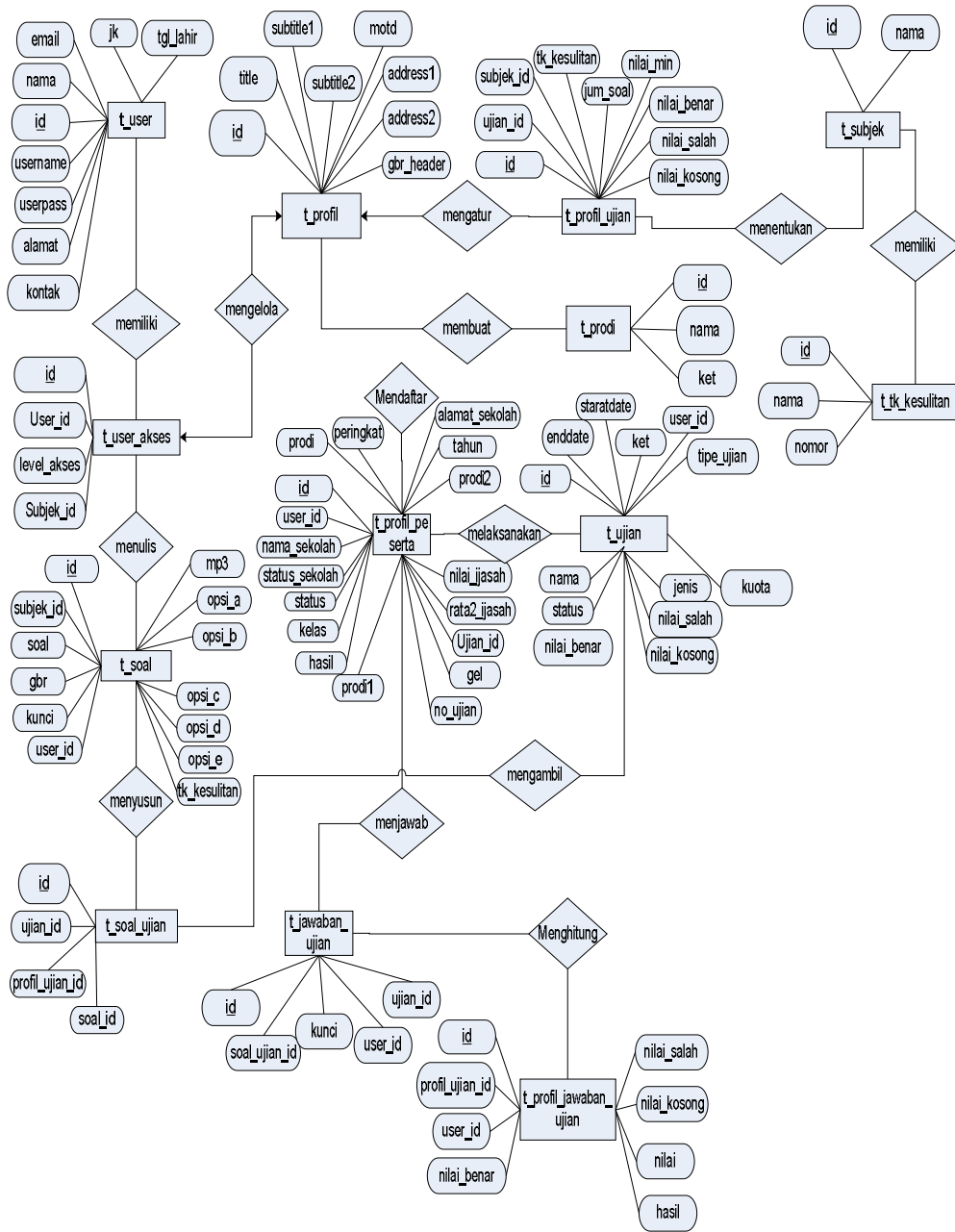
#### IV.4.7.2 Deskripsi

Deskripsi rancangan tampilan ujian disajikan pada tabel 11.

Tabel 11 Deskripsi Rancangan Tampilan Ujian

<b>Id_Objek</b>	<b>Jeniis</b>	<b>Nama</b>	<b>Keterangan</b>
Button11	Button	addProlUjian	Jika diklik, maka akan tampil layar form tambah profil ujian
CB22	Combo Box	nama	Memilih nama ujian yang akan dicari
Button12	Button	searchProfilUjian	Jika diklik, maka akan tampil profil ujian yang dicari
Button13	Button	addProfilUjian	Jika diklik, maka akan tampil layar form tambah profil ujian
Button14	Button	closeUjian	Jika diklik, maka ujian yang telah selesai akan ditutup
Button15	Button	viewProfilUjian	Jika diklik, maka akan tampil dokumen profil ujian dalam bentuk <i>excel</i>
TF36	Text Field	nama	Berisi nama-nama ujian
TF37	Text Field	ket	Berisi keterangan dari nama ujian
TF38	Text Field	tipe_ujian	Berisi tipe ujian
TF39	Text Field	startdate	Berisi waktu mulai suatu ujian
TF40	Text Field	enddate	Berisi waktu selesai suatu ujian
TF41	Text Field	kuota	Berisi jumlah kuota suatu ujian
CB23	Combo Box	status	Memilih status suatu ujian

## IV.5 Diagram Entity Relationship (ER)



Gambar 29 Diagram Entity Relationship (ER)

Pada Gambar 25 yang menggambarkan *diagram entity relationship* (ER) menjelaskan bahwa dalam aplikasi ini ada tiga belas (13) entitas atau tabel. Berikut ini penjelasan entitas tersebut bersama atribut-atribut dan kardinalitasnya.

1. t\_user memiliki atribut id, nama, *username*, *userpass*, jk, tgl\_lahir, alamat, kontak, *email*. *Primary key*-nya adalah id. Tabel ini memiliki kardinalitas *many-to-many* dengan tabel t\_user\_akses.
2. t\_user\_akses memiliki id, user\_id, level\_akses dan subjek\_id. *Primary key*-nya adalah id. Tabel ini memiliki kardinalitas *many-to-many* dengan tabel t\_soal. Selain itu, tabel ini juga memiliki kardinalitas *one-to-one* dengan tabel t\_profil.
3. t\_soal memiliki atribut id, subjek\_id, tk\_kesulitan, user\_id, soal, gbr, mp3, opsi\_a, opsi\_b, opsi\_c, opsi\_d, opsi\_e, dan kunci. *Primary key*-nya adalah id. Tabel ini memiliki kardinalitas *many-to-many* dengan tabel t\_soal ujian.
4. t\_soal ujian memiliki atribut id, ujian\_id, profil\_ujian\_id, dan soal\_id. *Primary key*-nya adalah id. Tabel ini memiliki kardinalitas *many-to-many* dengan tabel t\_ujian.
5. t\_profil memiliki atribut id, *title*, *subtitle1*, *subtitle2*, motd, *address1*, *address2*, dan gbr\_header. *Primary key*-nya adalah id. Tabel ini memiliki kardinalitas *one-to-many* dengan tabel t\_profil\_ujian. Selain itu, tabel ini juga memiliki kardinalitas *many-to-many* dengan tabel t\_prodi.
6. t\_profil\_ujian memiliki atribut id, ujian\_id, subjek\_id, tk\_kesulitan, jum\_soal, nilai\_min, nilai\_benar, nilai\_salah, dan nilai\_kosong. *Primary key*-nya adalah id. Tabel ini memiliki kardinalitas *many-to-many* dengan tabel t\_subjek.
7. t\_subjek memiliki atribut id dan nama. *Primary key*-nya adalah id. Tabel ini memiliki kardinalitas *many-to-many* dengan tabel t\_tk\_kesulitan.
8. t\_tk\_kesulitan memiliki atribut id, nama dan nomor. *Primary key*-nya adalah id.
9. t\_prodi memiliki atribut id, nama, dan ket. *Primary key*-nya adalah id.
10. t\_profil\_peserta id, user\_id, nama\_sekolah, status\_sekolah, alamat\_sekolah, tahun, nilai\_ijasah, rata2\_ijasah, ujian\_id, gel, prodi1, prodi2, kelas, no\_ujian,

status, hasil, prodi, dan peringkat. *Primary key*-nya adalah id. Tabel ini memiliki kardinalitas *many-to-many* dengan tabel t\_ujian. Selain itu, tabel ini juga memiliki kardinalitas *many-to-many* dengan table t\_jawaban\_ujian.

11. t\_ujian memiliki atribut id, *user\_id*, startdate, enddate, ket, status, jenis, nilai\_benar, nilai\_slah, nilai\_kosong, tipe\_ujian, dan kuota. *Primary key*-nya adalah id
12. t\_jawaban\_ujian memiliki atribut id, soal\_ujian\_id, kunci, *user\_id*, dan ujian\_id. Tabel ini memiliki kardinalitas *many-to-many* dengan table t\_profil\_jawaban\_ujian. *Primary key*-nya adalah id.
13. t\_profil\_jawaban\_ujian memiliki atribut id, profil\_ujian\_id, *user\_id*, nilai\_benar, nilai\_salah, nilai\_kosong, nilai, dan hasil. *Primary key*-nya adalah id.

## Bab V Implementasi dan Pengujian

Setelah dilakukan tahap perancangan aplikasi ini maka tahap selanjutnya adalah implementasi aplikasi tersebut. Untuk itu perlu dilakukan pengujian aplikasi.

### V.1 Implementasi Kelas

Implementasi kelas disajikan pada tabel 12.

Tabel 12 Implementasi Kelas

No	Nama Kelas	Nama File Fisik	Nama File Executable
1.	Admin	class.admin.php	class.admin.php
2.	Prodi	class.prodi.php	class.prodi.php
3.	Subjek	class.subjek.php	class.subjek.php
4.	profil ujian	class.profil.ujian.php	class.profil.ujian.php
5.	Soal	class.soal.php	class.soal.php
6.	Peserta	class.peserta.php	class.peserta.php
7.	Ujian	class.ujian.php	class.ujian.php
8.	Query	class.query.php	class.query.php

## V.2 Implementasi Antarmuka

Implementasi antarmuka disajikan pada tabel 13.

Tabel 13 Implementasi Antarmuka

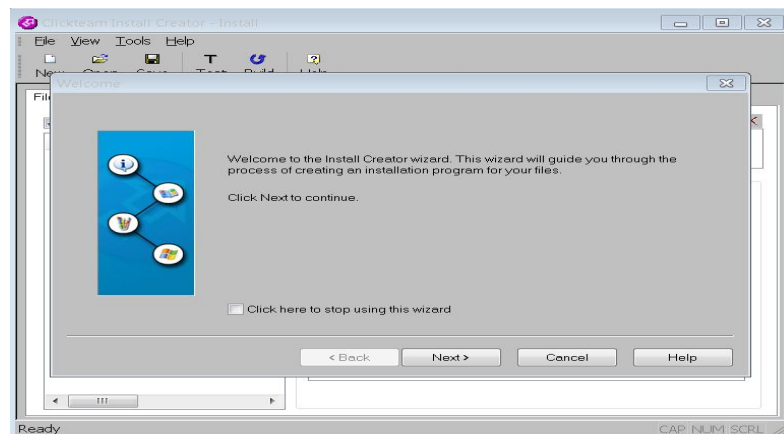
No	Antarmuka	Nama File Fisik	Nama File Executable
1.	Layar Menu Utama	index.php	index.php
		content_center.php	content_center.php
		footer.php	footer.php
		menu.php	menu.php
		loginForm.php	loginForm.php
2.	Layar Program Studi	viewProdi.php	viewProdi.php
		add.Prodi.php	add.Prodi.php
3.	Layar Subjek	addSubjek.php	addSubjek.php
		addTkKesulitan.php	addTkKesulitan.php
		selectSubjek.php	viewSubjek.php
4.	Layar Profil Ujian	addProfilUjian.php	addProfilUjian.php
		searchProfilUjian.php	searchProfilUjian.php
		viewSoalUjian.php	viewSoalUjian.php
		viewProfilUjian.php	viewProfilUjian.php
		searchProfilUjian.php	searchProfilUjian.php

No	Antarmuka	Nama File Fisik	Nama File Executable
		selectNilaiBenar.php	selectNilaiBenar.php
		selectNilaiMin.php	selectNilaiMin.php
		selectNilaiSalah.php	selectNilaiSalah.php
5.	Layar Soal	addSoal.php	addSoal.php
		searchSoal.php	searchSoal.php
		selectJumSoal.php	selectJumSoal.php
		selectTkKesulitan.php	selectTkKesulitan.php
		viewSoal.php	viewSoal.php
6.	Layar Peserta	searchPeserta.php	searchPeserta.php
		viewPeserta.php	viewPeserta.php
7.	Layar User	addUser.php	addUser.php
		viewUser.php	viewUser.php
		viewUser2.php	viewUser2.php
8.	Layar Profil Organisasi	viewProfil.php	viewProfil.php
9.	Layar Master Reset	viewMasterReset.php	viewMasterReset.php

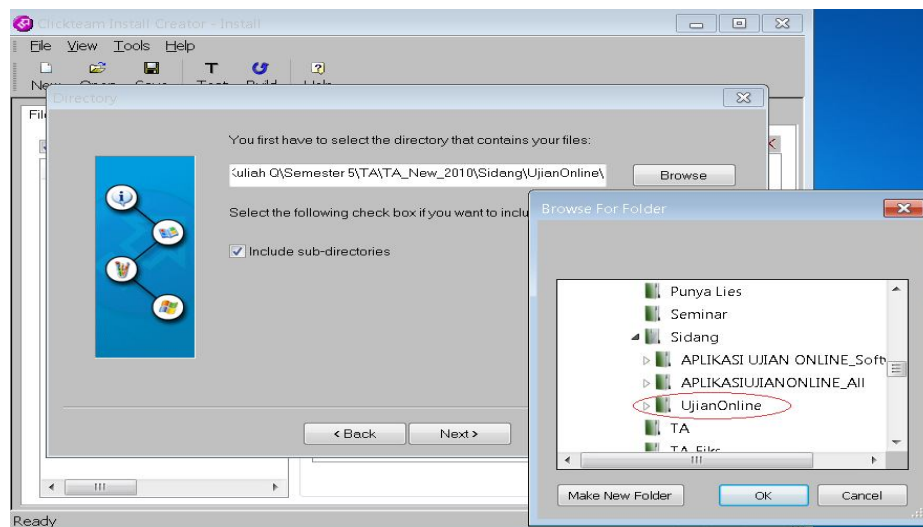
### V.3 Implementasi Instalasi

Aplikasi Pengembangan Sistem Ujian *Online* Berbasis Objek ini juga menyediakan instalasi. Berikut ini tata cara instalasi aplikasi ini:

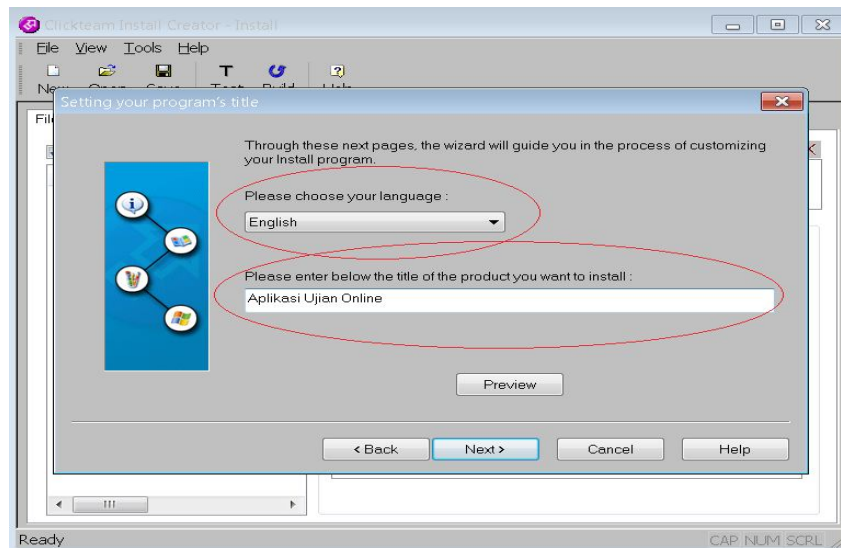
1. Pertama *install* terlebih dahulu xampp 1.7.0, PostgreSQL 8.4 dan *icinst.exe (Install Creator)* di komputer Anda.
2. Klik dua kali pada aplikasi *Install Creator*, akan tampil seperti gambar berikut, kemudian pilih *next*.



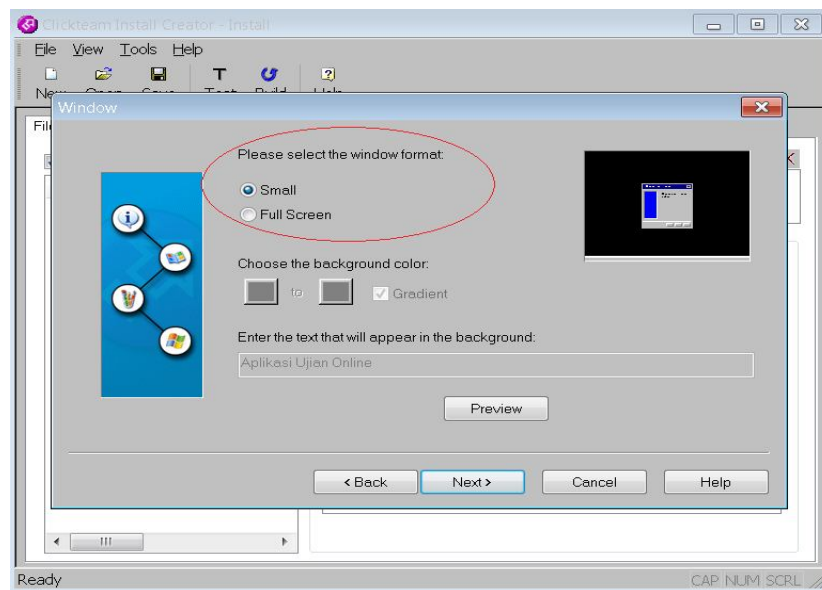
3. *Browse file* UjianOnline pada *directory* dimana *file* UjianOnline tersimpan, kemudian pilih OK dan *next*, terlihat seperti gambar berikut:



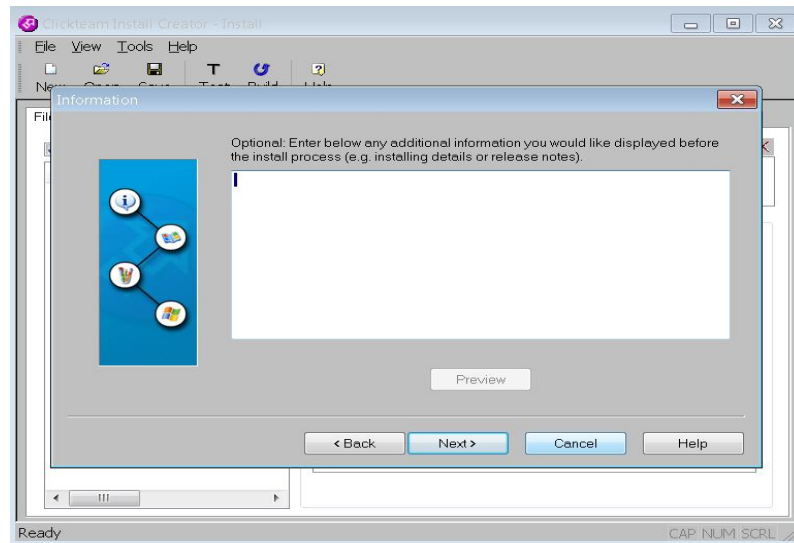
4. Pilih bahasa dan beri nama produk yang akan di *install*, misalnya Aplikasi Ujian Online, kemudian pilih *next*, terlihat seperti gambar berikut:



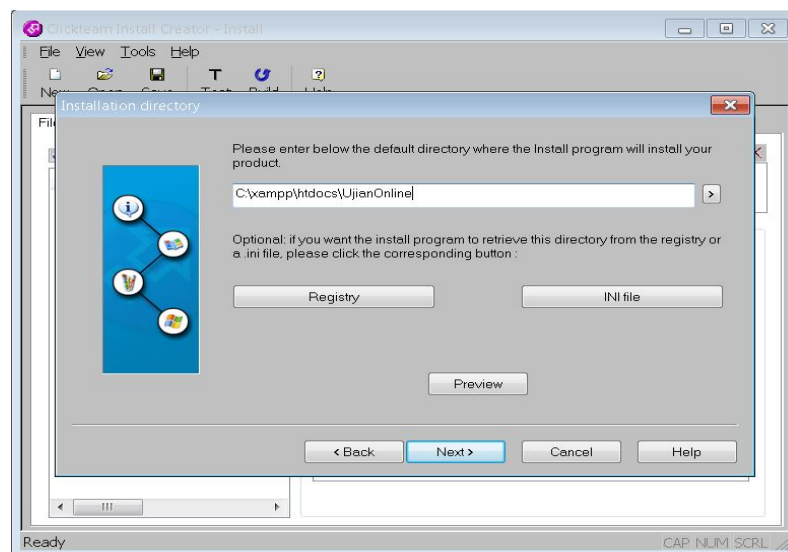
5. Pilih tampilan jendela (*small* atau *full screen*), kemudian pilih *next* saja, terlihat seperti gambar berikut:



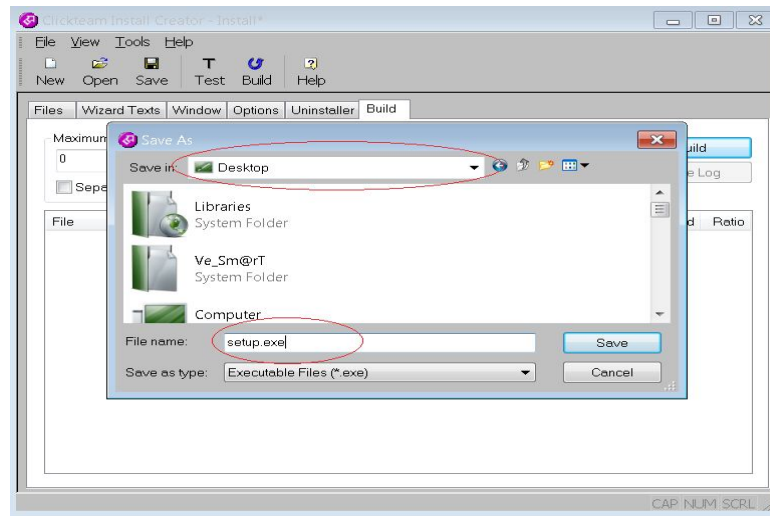
6. Masukkan informasi yang perlu ditampilkan pada saat menjalankan aplikasi ujian *online* (kosongkan saja jika dianggap tidak perlu), kemudian pilih *next*, terlihat seperti gambar berikut:



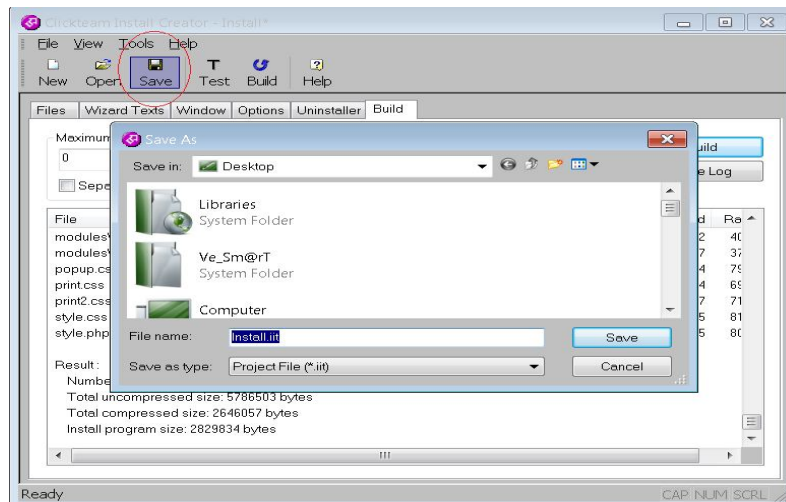
7. Pilih *directory* dimana *file* UjianOnline akan masuk ketika aplikasi ujian *online* dijalankan (C:\xampp\htdocs\UjianOnline), kemudian pilih *next* saja sampai tampil tombol *finish*, terlihat seperti gambar berikut:



8. Pilih *directory* untuk menyimpan *file* instalasi ujian *online*. Kemudian beri nama *file* tersebut dengan *setup.exe* dan pilih *save*. Tunggu beberapa saat sampai proses *copy file* selesai, terlihat seperti gambar berikut:

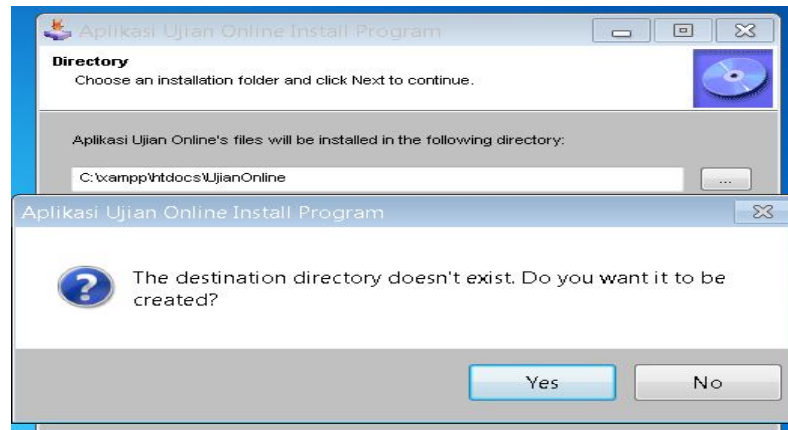


9. Pilih *icon save* untuk menyimpan *file* Instal.iit, terlihat seperti gambar berikut:



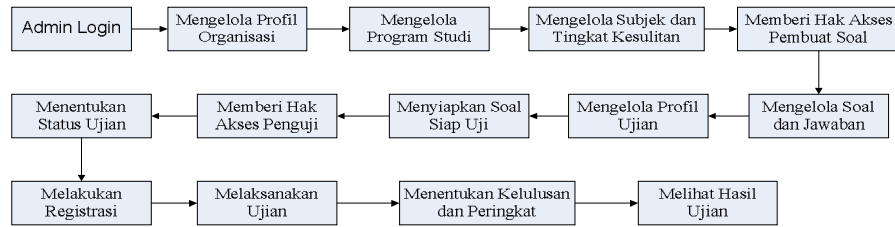
Untuk menginstal aplikasi ujian *online* diatas, ikuti cara di bawah ini:

1. *Klik* 2 kali pada *file* setup.exe untuk menjalankan instalasi aplikasi ujian *online*. Kemudian pilih *next* saja pada setiap jendela instalasi yang muncul.
2. Pilih *radio button* "I agree...", kemudian pilih *next* saja, jika tampil jendela dengan pertanyaan "do you want it to be created?", maka pilih *yes*. Kemudian pilih *start*.



3. Tunggu beberapa saat, karena aplikasi tersebut sedang mengcopy *file-file* aplikasi ujian *online*. Jika sudah selesai, maka pilih *next* dan *exit*.
4. Coba lihat folder UjianOnline di C:\xampp\htdocs. Jika *file* tersebut sudah ada, berarti sudah berhasil melakukan instalasi aplikasi ini.
5. Selanjutnya, Anda dapat memilih *file* setup-db.bat untuk menginstal *database* dari aplikasi ujian *online* ini.
6. Jika tampil layar cmd dan meminta *password*, masukkan *password* dengan *postgres*.
7. Tunggu beberapa saat, karena aplikasi sedang mengcopy *file-file* tersebut ke *database*. Kemudian ulangi memasukkan *password postgres*. Tunggu beberapa saat.
8. Buka aplikasi PgAdminIII yang telah diinstall. Coba lihat apakah *database* db\_ujian\_online sudah ada atau belum. Jika ada berarti instalasi *database* aplikasi ini telah berhasil.
9. Buka *Browser* dan masukkan halaman web berikut <http://localhost/ujianonline/>. Jika halaman web tersebut dapat diakses, maka aplikasi ini telah berhasil diinstall.

## V.4 Skenario Pengujian



Gambar 30 Skenario Pengujian

Gambar 30 menjelaskan tentang skenario aplikasi Pengembangan Sistem Ujian *Online* Berbasis Objek, antara lain:

### 1. Login

Setelah aplikasi ini di *install*, maka admin dapat *login* dengan nama *user*: admin dan kata sandi: admin. Setelah *login*, maka aplikasi akan menampilkan menu-menu yang ada di dalam aplikasi ini, antara lain daftar subjek, daftar program studi, daftar soal, daftar profil ujian, daftar peserta, daftar hasil ujian, daftar *user*, profil organisasi, dan *master reset*.

### 2. Mengelola Profil Organisasi

Pertama kali setelah *login*, admin dapat memilih menu profil organisasi untuk mengisi profil organisasi sesuai *form* yang telah disediakan, antara lain: judul, Sub Judul, Nama Organisasi, Pesan Hari Ini, Alamat 1, Alamat 2 dan logo. Kemudian dapat memilih tombol “Simpan”, maka secara otomatis akan tersimpan ke *database* dan aplikasi akan menampilkan profil organisasi yang telah dimasukkan.

### 3. Mengelola Program Studi

Jika aplikasi ini digunakan pada instansi pendidikan, admin dapat memasukkan program studi sesuai dengan kebutuhan organisasi atau instansi tersebut. Admin dapat memilih menu daftar program studi untuk memasukkan beberapa program studi yang ada pada organisasi tersebut. Contoh:

- Teknik Informatika
- Akuntansi, dan lain-lain.

Jika sudah memasukkan beberapa program studi, coba pilih tombol “Perbaharui”, maka akan tampil daftar program studi yang telah dimasukkan. Admin juga dapat menambah dan mengubah program studi.

#### 4. Mengelola Subjek dan Tingkat Kesulitan

Admin dapat memilih menu daftar subjek untuk memasukkan beberapa subjek yang akan dimasukkan soal-soal oleh pembuat soal. Contoh:

- Matematika
- Bahasa Indonesia, dan lain-lain

Selain itu, admin juga dapat memasukkan tingkat kesulitan untuk subjek-subjek tersebut. Contoh:

- Mudah
- Sulit, dan lain-lain.

Jika sudah memasukkan beberapa subjek dan tingkat kesulitan, coba pilih tombol “Perbaharui”, maka akan tampil daftar subjek dan tingkat kesulitan yang telah dimasukkan. Admin juga dapat menambah dan mengubah subjek serta tingkat kesulitan.

#### 5. Memberi Hak Akses Pembuat Soal

Admin dapat memilih menu daftar *user* untuk memberi hak akses pembuat soal sesuai subjek yang dibutuhkan agar pembuat soal dapat membuat atau memasukkan beberapa soal ke subjek tersebut. Admin dapat memilih tombol tambah *user* untuk menambah *user*. Kemudian isi beberapa informasi yang dibutuhkan, seperti nama *user*, sandi, nama, jenis kelamin, alamat, kontak, *email*, dan tanggal lahir. Setelah itu admin dapat menentukan hak akses pembuat soal dengan memberi tanda *checkbox* pada *list* akses pembuat soal.

Kemudian pilih tombol “Simpan” untuk menyimpan data-data tersebut, dan pilih tombol “Tutup” untuk menutup *form* tersebut.

## 6. Mengelola Soal dan Jawaban

Setelah diberi hak akses oleh admin sebagai pembuat soal dengan subjek tertentu, maka pembuat soal dapat *login* sesuai nama *user* dan sandi yang diberi oleh admin. Selanjutnya, setelah *login* sukses, maka akan tampil menu daftar soal dan pembuat soal dapat mengelola soal dan jawaban pada subjek tersebut dengan memilih tombol tambah soal. Setelah itu kita dapat memasukkan soal-soal sesuai subjek dan tingkat kesulitannya. Contoh:

- Nama Subjek : Matematika
- Tingkat Kesulitan : Mudah
- Soal : Berapa hasil dari  $(3 \times 20 + 20 - 10) : 10$  ?
- Suara : -
- Opsi a : 100
- Opsi b : 1000
- Opsi c : 10000
- Opsi d : 100000
- Opsi e : 99
- Kunci : A

Setelah itu, pilih tombol “Perbaharui” untuk melihat daftar soal-soal yang telah dimasukkan. Selain itu, jika ingin mencari soal-soal yang telah dimasukkan dengan memilih subjek dan tingkat kesulitan kemudian pilih tombol cari, maka aplikasi akan mencari soal-soal sesuai subjek dan tingkat kesulitan tersebut.

## 7. Mengelola Profil Ujian

Mengelola profil ujian adalah tugas admin. Admin dapat mengelola profil dengan memilih menu daftar profil ujian. Setelah tampil daftar profil ujian, pilih tombol tambah profil ujian. Admin dapat memasukkan beberapa

informasi untuk profil ujian berupa nama ujian, tipe ujian, keterangan dari nama ujian, waktu mulai dan waktu selesai serta kuota. Dari profil ujian inilah, peserta dapat memilih ujian mana yang akan diikuti. Contoh:

- Nama Ujian : Ujian Masuk Gel. 1 PolBat
- Tipe Ujian : UN
- Keterangan : Ujian Masuk Gelombang 1 Politeknik Negeri Batam
- Waktu Mulai : 2011-Feb-16 08:00
- Waktu Selesai : 2011-Feb-18 08:00
- Kuota : 50 Orang

Menyimpan daftar profil ujian harus bersamaan dengan pengaturan soal siap uji. Setelah mengisi *form* profil ujian, maka dapat mengisi *form* komposisi soal.

#### 8. Menyiapkan Soal Siap Uji

Admin melakukan pengaturan soal siap uji yang akan ditampilkan dalam suatu ujian pada menu daftar profil ujian. Admin dapat mengisi *form* komposisi soal dengan memasukan data berupa subjek, tingkat kesulitan, jumlah soal, nilai minimum, nilai benar, dan nilai salah. Contoh:

- Subjek : Matematika
- Tingkat Kesulitan : Sulit
- Jumlah Soal : 10
- Nilai Minimum : 50
- Nilai Benar : 5
- Nilai Salah : -1

Menambahkan komposisi soal dapat dilakukan berulang-ulang sesuai berapa banyak soal yang akan kita tampilkan sesuai dengan subjek dan tingkat kesulitannya. Setelah selesai, kemudian pilih tombol “Simpan”, maka akan tampil daftar profil ujian yang telah dimasukkan atau pilih tombol “Perbaharui”.

## 9. Memberi Hak Akses Penguji

Admin dapat memberi hak akses penguji setelah memiliki profil ujian. Untuk memberi hak akses kepada penguji, caranya sama seperti memberi hak akses kepada pembuat soal. Jika hak akses pembuat soal admin hanya memberi *checklist* pada hak akses pembuat soal, maka untuk penguji juga demikian. Admin dapat memberi *checklist* pada hak akses penguji.

## 10. Menentukan Status Ujian

Setelah diberi hak akses sebagai penguji oleh admin, penguji dapat *login* dengan menggunakan nama *user* dan sandi yang diberi oleh admin. Kemudian admin dapat menentukan status ujian (“Buka” atau “Tutup”) pada ujian tertentu sesuai hak akses yang diberikan oleh admin.

## 11. Melaksanakan Registrasi

Setelah semua pengelolaan sebuah organisasi yang akan melaksanakan ujian telah selesai, maka peserta dapat melakukan registrasi dengan memilih tombol registrasi peserta ujian. Peserta dapat mengisi formulir pendaftaran yang telah disediakan dan memilih tombol “Simpan” jika sudah selesai. Kemudian, peserta dapat *login* dengan menggunakan nama *user* dan sandi yang telah dimasukkan pada saat registrasi. Setelah *login*, peserta dapat memilih menu daftar ujian dan aplikasi akan menampilkan beberapa informasi berupa nama ujian, tipe ujian, keterangan dari nama ujian, waktu mulai, waktu selesai, nomor ujian, program studi1 dan program studi2 (jika ada).

## 12. Melaksanakan Ujian

Setelah *login* peserta dapat memilih menu daftar ujian dan memilih tombol “Mulai Ujian” untuk mengetahui aturan penilaian ujian yang akan diikuti oleh peserta. Setelah mengetahui aturan penilaian ujian, peserta dapat memilih tombol “Mulai Ujian”. Aplikasi akan menampilkan soal-soal yang harus dikerjakan oleh peserta dan peserta dapat mulai menjawab soal-soal tersebut.

Setelah selesai, peserta dapat memilih tombol “Selesai Ujian”. Peserta dapat melihat hasil jawabannya pada soal-soal yang telah dikerjakan, namun tidak dapat mengulangi ujian kembali.

### 13. Menentukan Kelulusan dan Peringkat

Untuk ujian dengan tipe sertifikasi, ketika peserta selesai mengikuti ujian, maka peserta dapat mengetahui nilai dan peringkatnya secara langsung. Sedangkan untuk tipe ujian UN, apabila peserta telah selesai melaksanakan ujian, maka hasilnya adalah “BELUM DINILAI”. Penguji yang akan menilai hasil ujian peserta (“LULUS” atau “TIDAK LULUS”) serta jika hasil ujian peserta lulus, maka penguji juga harus menentukan peserta lulus pada program studi yang mana. Namun untuk peringkat akan tampil secara langsung, tetapi peringkat tersebut adalah peringkat sementara. Setelah ujian tersebut telah ditutup, maka penguji dapat menentukan peringkat peserta dalam ujian tersebut.

Jika tipe ujian UN dan memandang kuota (kapasitas peserta yang diterima pada suatu ujian dan organisasi), maka penguji dapat menentukan peserta yang lulus berdasarkan peringkat, jika sudah mencapai kuota, maka aplikasi akan secara otomatis memberi hasil kelulusan “TIDAK LULUS”.

### 14. Melihat Hasil Ujian

Untuk tipe ujian sertifikasi, peserta dapat melihat hasil ujian secara langsung setelah selesai melaksanakan ujian. Sedangkan untuk tipe ujian UN, hasil ujian akan tampil dengan status “BELUM DINILAI”. Peserta dapat melihat hasil ujian (“LULUS” atau “TIDAK LULUS”) setelah penguji memeriksa hasil ujian peserta.

## V.5 Dokumen Rinci Testing

### V.5.1 Tim Penguji

1. Riwinoto (RW)

### V.5.2 Hasil Rinci Pengujian

Hasil rinci pengujian pada aplikasi Pengembangan Sistem Ujian *Online* Berbasis Objek disajikan pada tabel 14.

Tabel 14 Hasil Rinci Pengujian

No	User	Use Case	Fungsi yang Diuji	Kondisi Awal	Input	Hasil Yang diharapkan	Output	Hasil Test
1.	Admin, Pembuat Soal, Penguji dan Peserta	<i>Login</i>	<i>Login</i> dengan nama <i>user</i> dan sandi yang benar	Tampil halaman beranda	- Admin: Nama <i>user</i> : admin Sandi: admin2011 - Pembuat Soal: Nama <i>user</i> : ve	Tampil menu-menu sesuai hak akses <i>user</i> . - Menu pada admin (admin) antara lain: daftar subjek, daftar program studi, daftar soal, daftar profil ujian, daftar peserta, daftar hasil ujian,	Tampil menu-menu sesuai hak akses <i>user</i> . - Menu pada admin (admin) antara lain: daftar subjek, daftar program studi, daftar soal, daftar	Diterima

No	User	Use Case	Fungsi yang Diuji	Kondisi Awal	Input	Hasil Yang diharapkan	Output	Hasil Test
					Sandi: ve2011 - Penguji: Nama user: lis Sandi: lis2011 - Peserta: Nama user: ani Sandi: ani	daftar <i>user</i> , profil organisasi, dan <i>master reset</i> - Menu pada ve (pembuat soal) adalah daftar soal - Menu pada lis (penguji) antara lain: daftar profil ujian, daftar peserta dan daftar hasil ujian - Menu pada ani (peserta) adalah daftar ujian	profil ujian, daftar peserta, daftar hasil ujian, daftar <i>user</i> , profil organisasi, dan <i>master reset</i> - Menu pada ve (pembuat soal) adalah daftar soal - Menu pada lis (penguji) antara lain: daftar profil ujian, daftar peserta dan daftar hasil ujian - Menu pada ani (peserta) adalah daftar ujian	
			<i>Login</i> dengan nama <i>user</i> dan	Tampil halaman	Nama <i>user</i> : ani Sandi: ani2011	<i>Login</i> tidak akan berhasil dan tampilan	<i>Login</i> tidak akan berhasil dan	Diterima

No	User	Use Case	Fungsi yang Diuji	Kondisi Awal	Input	Hasil Yang diharapkan	Output	Hasil Test
			sandi yang salah	beranda		layar kembali ke tampilan beranda	tampilan layar kembali ke tampilan beranda	
2.	Admin	Mengelola Profil Organisasi	Mengisi <i>form</i> profil organisasi secara lengkap	Memilih menu profil organisasi	<ul style="list-style-type: none"> <li>- Judul: Sertifikasi TUK Politeknik Negeri Batam</li> <li>- Sub judul: Selamat Datang!!</li> <li>- Nama Organisasi: Politeknik Negeri Batam</li> <li>- Pesan Hari ini: Jujurlah dalam Ujian!!</li> </ul>	<p>Tampil profil organisasi sesuai data yang dimasukkan.</p> <ul style="list-style-type: none"> <li>- Judul: tampil di halaman atas samping logo</li> <li>- Sub Judul: tampil di samping menu</li> <li>- Nama Organisasi: tampil di bawah sub judul</li> <li>- Pesan Hari Ini: tampil pada halaman peserta ketika akan melaksanakan ujian</li> <li>- Alamat1: tampil di bawah nama</li> </ul>	<p>Tampil profil organisasi sesuai data yang dimasukkan.</p> <ul style="list-style-type: none"> <li>- Judul: tampil di halaman atas samping logo</li> <li>- Sub Judul: tampil di samping menu</li> <li>- Nama Organisasi: tampil di bawah sub judul</li> <li>- Pesan Hari Ini: tampil pada halaman peserta ketika akan melaksanakan</li> </ul>	Diterima

No	User	Use Case	Fungsi yang Diuji	Kondisi Awal	Input	Hasil Yang diharapkan	Output	Hasil Test
					<ul style="list-style-type: none"> <li>- Alamat 1: Parkway Street Batam Centre</li> <li>- Alamat 2: Batam, Indonesia</li> <li>- Logo: poltek.jpg</li> </ul>	organisasi - Alamat 2: tampil di bawah alamat 1 - Logo: tampil gambar yang dimasukkan pada bagian halaman atas.	ujian - Alamat1: tampil di bawah nama organisasi - Alamat 2: tampil di bawah alamat 1 - Logo: tampil gambar yang dimasukkan pada bagian halaman atas.	
			Mengisi <i>form</i> profil organisasi secara tidak lengkap	Memilih menu profil organisasi	Sub judul: - (dikosongkan)	<i>Field</i> sub judul tidak akan tampil pada tempatnya	<i>Field</i> sub judul tidak akan tampil pada tempatnya	Diterima
3.	Admin	Mengelola Daftar <i>User</i>	Memberi hak akses pembuat soal	Memilih tombol Tambah <i>User</i>	<ul style="list-style-type: none"> <li>- Nama <i>user</i>: ve</li> <li>- Sandi: ve2011</li> </ul>	Ve sebagai pembuat soal dapat <i>login</i> dan dapat membuat soal	Ve sebagai pembuat soal dapat <i>login</i> dan dapat	Diterima

No	User	Use Case	Fungsi yang Diuji	Kondisi Awal	Input	Hasil Yang diharapkan	Output	Hasil Test
				pada menu daftar <i>user</i>	<ul style="list-style-type: none"> <li>- Nama: Ve</li> <li>- Jenis Kelamin: Perempuan</li> <li>- Alamat: Batam Centre</li> <li>- Kontak: 081370690983</li> <li>- Email: <a href="mailto:verri_ve@yahoo.com">verri_ve@yahoo.com</a></li> <li>- Tanggal Lahir: 1990 Mar 3</li> <li>- Akses Pembuat Soal: <i>checklist</i> pada Bahasa Indonesia dan</li> </ul>	pada subjek Bahasa Indonesia dan Matematika	membuat soal pada subjek Bahasa Indonesia dan Matematika	

No	User	Use Case	Fungsi yang Diuji	Kondisi Awal	Input	Hasil Yang diharapkan	Output	Hasil Test
					Matematika			
			Memberi hak akses pengujian	Memilih tombol tambah <i>user</i> pada menu daftar <i>user</i>	<ul style="list-style-type: none"> <li>- Nama <i>user</i>: lis</li> <li>- Sandi: lis2011</li> <li>- Nama: Lis</li> <li>- Jenis Kelamin: Perempuan</li> <li>- Alamat: Batam Centre</li> <li>- Kontak: 08566099032</li> <li>- Email: <a href="mailto:lis@yahoo.com">lis@yahoo.com</a></li> <li>- Tanggal Lahir: 1990 Jan 31</li> <li>- Akses Pengujian</li> </ul>	Pengujian dapat <i>login</i> dan dapat menguji pada Ujian Masuk Gel. 1 PolBat	Pengujian dapat <i>login</i> dan dapat menguji pada Ujian Masuk Gel. 1 PolBat	Diterima

No	User	Use Case	Fungsi yang Diuji	Kondisi Awal	Input	Hasil Yang diharapkan	Output	Hasil Test
					Ujian: <i>checklist</i> pada Ujian Sertifikasi TUK PolBat dan Ujian Masuk Gel. 1 PolBat			
			Menghapus hak akses pembuat soal	Memilih tombol Ubah pada nama Ve (pembuat soal) pada menu daftar <i>user</i>	Hapus tanda <i>checklist</i> pada Akses Pembuat Soal pada Bahasa Indonesia	Ve (pembuat soal) tidak dapat memasukkan soal-soal pada subjek Bahasa Indonesia	Ve (pembuat soal) tidak dapat memasukkan soal-soal pada subjek Bahasa Indonesia	Diterima
			Menghapus hak akses penguji	Memilih tombol Ubah pada nama Lis (penguji) pada	Hapus tanda <i>checklist</i> pada Akses Penguji Ujian pada	Lis (penguji) tidak dapat menguji pada Ujian Sertifikasi TUK PolBat	Lis (penguji) tidak dapat menguji pada Ujian Sertifikasi TUK PolBat	Diterima

No	User	Use Case	Fungsi yang Diuji	Kondisi Awal	Input	Hasil Yang diharapkan	Output	Hasil Test
				menu daftar <i>user</i>	Ujian Sertifikasi TUK PolBat			
4.	Admin	Mengelola Program Studi	Menambah program studi	Memilih tombol Tambah Program Studi pada menu daftar program studi	- Nama Program Studi: Akuntansi dan Teknik Informatika - Keterangan: Akuntansi dan Teknik Informatika	Tampil program studi Akuntansi dan Teknik Informatika pada daftar program studi dan pada <i>form</i> registrasi peserta ketika akan memilih program studi	Tampil program studi Akuntansi dan Teknik Informatika pada daftar program studi dan pada <i>form</i> registrasi peserta ketika akan memilih program studi	Diterima
			Mengubah program studi	Memilih tombol Ubah pada program studi Teknik Informatika	- Nama Program Studi: Teknik Informatika diganti dengan	Tampil program studi Teknik Elektro pada daftar program studi dan pada <i>form</i> registrasi peserta	Tampil program studi Teknik Elektro pada daftar program studi dan pada <i>form</i> registrasi	Diterima

No	User	Use Case	Fungsi yang Diuji	Kondisi Awal	Input	Hasil Yang diharapkan	Output	Hasil Test
					Teknik Elektro - Keterangan: Teknik Informatika diganti dengan Teknik Elektro	ketika akan memilih program studi	peserta ketika akan memilih program studi	
5.	Admin	Mengelola Subjek dan Tingkat Kesulitan	Menambah subjek	Memilih tombol Tambah Subjek pada menu daftar subjek	Nama Subjek: Bahasa Indonesia dan Matematika	Tampil subjek Bahasa Indonesia dan Matematika pada daftar subjek	Tampil subjek Bahasa Indonesia dan Matematika pada daftar subjek	Diterima
			Menambah tingkat kesulitan	Memilih tombol Tambah	Nama Tingkat Kesulitan: Mudah,	Tampil tingkat kesulitan mudah, sedang dan sulit pada	Tampil tingkat kesulitan mudah, sedang dan sulit	Diterima

No	User	Use Case	Fungsi yang Diuji	Kondisi Awal	Input	Hasil Yang diharapkan	Output	Hasil Test
				Tingkat Kesulitan pada menu daftar subjek	Sedang dan Sulit	daftar tingkat kesulitan	pada daftar tingkat kesulitan	
			Mengubah subjek	Memilih tombol Ubah pada subjek Bahasa Indonesia	Nama Subjek: Bahasa Indonesia diganti dengan Bahasa Inggris	Tampil subjek Bahasa Inggris pada daftar subjek	Tampil subjek Bahasa Inggris pada daftar subjek	Diterima
			Mengubah tingkat kesulitan	Memilih tombol Ubah pada tingkat kesulitan Sedang	Nama Tingkat Kesulitan: Sedang diganti dengan Cukup Sulit	Tampil tingkat kesulitan cukup sulit pada daftar tingkat kesulitan	Tampil tingkat kesulitan cukup sulit pada daftar tingkat kesulitan	Diterima
6.	Admin	Mengelola	Menambah	Memilih	- Nama Ujian:	Tampil data profil	Tampil data profil	Diterima

No	User	Use Case	Fungsi yang Diuji	Kondisi Awal	Input	Hasil Yang diharapkan	Output	Hasil Test
		Konfigurasi Kelulusan	profil ujian	tombol Tambah Profil Ujian pada menu daftar profil ujian	Ujian Sertifikasi TUK PolBat - Tipe Ujian: Sertifikasi - Keterangan: Ujian Sertifikasi TUK PolBat - Waktu Mulai: 2011 Feb 23 08:00 - Waktu Selesai: 2011 Feb 25 12:00 - Kuota: 0	ujian tersebut pada daftar profil ujian dan nama ujian tampil pada <i>form</i> registrasi peserta pada saat memilih ujian	ujian tersebut pada daftar profil ujian dan nama ujian tampil pada <i>form</i> registrasi peserta pada saat memilih ujian	
			Menghapus profil ujian	Admin memilih menu daftar profil ujian	Admin memilih tombol hapus pada Ujian	Nama ujian “Ujian Sertifikasi TUK PolBat” tidak ada lagi pada daftar profil ujian	Nama ujian “Ujian Sertifikasi TUK PolBat” tidak ada lagi pada daftar	Diterima

No	User	Use Case	Fungsi yang Diuji	Kondisi Awal	Input	Hasil Yang diharapkan	Output	Hasil Test
					Sertifikasi TUK PolBat		profil ujian	
			Mencari profil ujian	Admin memilih menu daftar profil ujian	Admin memilih nama ujian “Ujian Sertifikasi TUK PolBat” kemudian memilih tombol cari	Tampil profil ujian Ujian Sertifikasi TUK PolBat	Tampil profil ujian Ujian Sertifikasi TUK PolBat	Diterima
7.	Pembuat Soal	Mengelola Soal dan Jawaban	Menambah soal dan jawaban	Memilih menu daftar soal	- Nama Subjek: Bahasa Indonesia - Tingkat Kesulitan: Mudah - Soal: Antonim dari	Tampil soal tersebut pada daftar soal dan pada daftar soal siap uji pada ujian tertentu	Tampil soal tersebut pada daftar soal dan pada daftar soal siap uji pada ujian tertentu	Diterima

No	User	Use Case	Fungsi yang Diuji	Kondisi Awal	Input	Hasil Yang diharapkan	Output	Hasil Test
					kata pintar adalah... - Suara: - (dikosongkan) - Opsi A: Bodoh - Opsi B: Cerdas - Opsi C: Brilliant - Opsi D: Jenius - Opsi E: Biasa-biasa saja - Kunci Jawaban: A			
			Mencari soal dan jawaban	Memilih menu daftar soal	- Nama Subjek: Bahasa	Tampil soal-soal Bahasa Indonesia	Tampil soal-soal Bahasa Indonesia	Diterima

No	User	Use Case	Fungsi yang Diuji	Kondisi Awal	Input	Hasil Yang diharapkan	Output	Hasil Test
					Indonesia - Tingkat Kesulitan: Mudah kemudian pilih tombol cari	dengan tingkat kesulitan mudah pada daftar soal	dengan tingkat kesulitan mudah pada daftar soal	
			Mengubah soal dan jawaban	Memilih tombol Ubah pada soal “Antonim dari kata...” pada daftar soal	Opsi C: Brilliant diganti dengan Pintar	Tampil soal pada opsi C yang baru pada daftar soal	Tampil soal pada opsi C yang baru pada daftar soal	Diterima
8.	Admin	Menyiapkan Soal Siap Uji	Menambah komposisi soal	Memilih menu daftar profil ujian	- Nama Subjek: Bahasa Indonesia - Tingkat Kesulitan:	Tampil daftar komposisi soal siap uji yang akan ditampilkan kepada peserta pada daftar profil ujian	Tampil daftar komposisi soal siap uji yang akan ditampilkan kepada peserta pada daftar	Diterima

No	User	Use Case	Fungsi yang Diuji	Kondisi Awal	Input	Hasil Yang diharapkan	Output	Hasil Test
					Mudah - Jumlah Soal: 10 - Nilai Minimum: 50 - Nilai Benar: 4 - Nilai Salah: - 1		profil ujian	
			Menghapus komposisi soal	Tampil daftar komposisi soal	Memilih tombol hapus pada: - Nama Subjek: Bahasa Indonesia - Tingkat Kesulitan: Mudah - Jumlah Soal:	Komposisi soal tersebut tidak akan tampil pada daftar profil ujian pada suatu ujian	Komposisi soal tersebut tidak akan tampil pada daftar profil ujian pada suatu ujian	Diterima

No	User	Use Case	Fungsi yang Diuji	Kondisi Awal	Input	Hasil Yang diharapkan	Output	Hasil Test
					10 - Nilai Minimum: 50 - Nilai Benar: 4 - Nilai Salah: - 1			
9.	Admin dan Penguji	Mengelola Daftar Peserta	Mencari daftar peserta	Memilih menu daftar peserta	Nama Ujian: Ujian Sertifikasi TUK PolBat kemudian pilih tombol cari	Tampil daftar peserta yang mengikuti Ujian Sertifikasi TUK	Tampil daftar peserta yang mengikuti Ujian Sertifikasi TUK	Diterima
			Mengubah sandi peserta	Admin memilih menu daftar peserta	Admin memilih tombol Sandi pada salah satu	Ani dapat melakukan <i>login</i> dengan menggunakan sandi yang baru	Ani dapat melakukan <i>login</i> dengan menggunakan sandi	Diterima

No	User	Use Case	Fungsi yang Diuji	Kondisi Awal	Input	Hasil Yang diharapkan	Output	Hasil Test
					peserta, misal Ani. - Nama User: ani - Sandi Baru: ani2011		yang baru	
10.	Penguji	Menentukan Status Ujian	Menambah profil ujian	Memilih menu daftar profil ujian	Memilih tombol Tambah Profil Ujian	Tampil pesan “Maaf!! Anda tidak punya hak akses!!”	Tampil pesan “Maaf!! Anda tidak punya hak akses!!”	Diterima
			Membuka ujian	Memilih menu daftar profil ujian	Memilih status Ujian Sertifikasi TUK PolBat “Buka”	Ujian Sertifikasi TUK PolBat akan tampil pada <i>form</i> registrasi peserta	Ujian Sertifikasi TUK PolBat akan tampil pada <i>form</i> registrasi peserta	Diterima
			Menutup ujian	Memilih menu daftar profil	Memilih status Ujian	Ujian Sertifikasi TUK PolBat tidak akan	Ujian Sertifikasi TUK PolBat tidak	Diterima

No	User	Use Case	Fungsi yang Diuji	Kondisi Awal	Input	Hasil Yang diharapkan	Output	Hasil Test
				ujian	Sertifikasi TUK PolBat “Tutup”	tampil pada <i>form</i> registrasi peserta	akan tampil pada <i>form</i> registrasi peserta	
11.	Penguji	Menentukan Kelulusan dan Peringkat	Memberi status kelulusan peserta	Memilih: - Nama Ujian: Ujian Masuk Gel. 1 PolBat - Status Hasil Ujian: Belum Dinilai kemudian pilih tombol cari	- Nama Peserta: Ani - Status Hasil Ujian: Lulus - Program Studi yang lulus: Akuntansi	Ani dapat mengetahui hasil ujiannya yaitu “Lulus” pada program studi Akuntansi	Ani dapat mengetahui hasil ujiannya yaitu “Lulus” pada program studi Akuntansi	Diterima
			Menentukan peringkat peserta	Memilih Nama Ujian: Ujian Masuk Gel. 1 PolBat kemudian memilih	Memilih tombol atur peringkat peserta	Peserta dapat mengetahui peringkatnya masing-masing pada Ujian Masuk Gel. 1 PolBat	Peserta dapat mengetahui peringkatnya masing-masing pada Ujian Masuk Gel. 1 PolBat	Diterima

No	User	Use Case	Fungsi yang Diuji	Kondisi Awal	Input	Hasil Yang diharapkan	Output	Hasil Test
				tombol cari				
			Menentukan peserta yang diterima pada suatu ujian	Memilih Nama Ujian: Ujian Masuk Gel. 1 PolBat kemudian memilih tombol cari	Memilih peserta yang lulus, kemudian memilih tombol atur hasil ujian = tidak lulus	Peserta pada Ujian Masuk Gel. 1 PolBat dapat mengetahui hasil ujian yang telah diikuti (“Lulus” atau “Tidak Lulus”) dan jika lulus dapat mengetahui pada program studi mana diterima	Peserta pada Ujian Masuk Gel. 1 PolBat dapat mengetahui hasil ujian yang telah diikuti (“Lulus” atau “Tidak Lulus”) dan jika lulus dapat mengetahui pada program studi mana diterima	Diterima
12.	Peserta	Melakukan Registrasi	Mengisi <i>form</i> pendaftaran secara lengkap	Tampil <i>form</i> registrasi	- Nama User: cinta - Sandi: cinta2011 - Nama: Cinta - Jenis	Cinta dapat <i>login</i> untuk mengetahui informasi-informasi tentang Ujian Sertifikasi TUK	Cinta dapat <i>login</i> untuk mengetahui informasi-informasi tentang Ujian Sertifikasi TUK	Diterima

No	User	Use Case	Fungsi yang Diuji	Kondisi Awal	Input	Hasil Yang diharapkan	Output	Hasil Test
					Kelamin: Perempuan - Tanggal Lahir: 2011 Feb 23 - Alamat: Bandung - Kontak: 08137556957 5 - Email: <a href="mailto:cinta@ymail.com">cinta@ymail.com</a> - Pilih Ujian: Ujian Sertifikasi TUK PolBat	PolBat. Selain itu, aplikasi juga akan menampilkan informasi tentang ujian yang telah diikuti oleh Cinta	PolBat. Selain itu, aplikasi juga akan menampilkan informasi tentang ujian yang telah diikuti oleh Cinta	
			Mengisi <i>form</i> pendaftaran secara tidak	Tampil <i>form</i> registrasi	Misal <i>field</i> alamat dikosongkan	Tampil layar <i>error</i>	Tampil layar <i>error</i>	Diterima

No	User	Use Case	Fungsi yang Diuji	Kondisi Awal	Input	Hasil Yang diharapkan	Output	Hasil Test
			lengkap					
			Memilih ujian dengan tipe UN	Tampil <i>form</i> registrasi	Memilih salah satu nama ujian yang ditampilkan	Tampil <i>form</i> tambahan berupa nama sekolah, status sekolah, alamat sekolah, tahun, nilai ijazah, rata-rata ijazah, prodi 1, prodi 2, dan kelas	Tampil <i>form</i> tambahan berupa nama sekolah, status sekolah, alamat sekolah, tahun, nilai ijazah, rata-rata ijazah, prodi 1, prodi 2, dan kelas	Diterima
			Melihat informasi waktu pelaksanaan ujian	Tampil <i>form</i> registrasi	Misal peserta memilih nama ujian “Ujian Sertifikasi TUK PolBat” kemudian memilih	Tampil informasi berupa: - Nama Ujian: Ujian Sertifikasi TUK PolBat - Tipe Ujian: Sertifikasi - Keterangan: Ujian	Tampil informasi berupa: - Nama Ujian: Ujian Sertifikasi TUK PolBat - Tipe Ujian: Sertifikasi - Keterangan: Ujian	Diterima

No	User	Use Case	Fungsi yang Diuji	Kondisi Awal	Input	Hasil Yang diharapkan	Output	Hasil Test
					tombol rinci	Sertifikasi TUK PolBat - Tanggal: 23-Feb-2011 08:00 s.d 25-Feb-2011 12:00 - Kuota: 0	Sertifikasi TUK PolBat - Tanggal: 23-Feb-2011 08:00 s.d 25-Feb-2011 12:00 - Kuota: 0	
13.	Peserta	Melaksanakan Ujian	Peserta mulai ujian	Tampil soal-soal yang harus dikerjakan oleh peserta	Peserta memilih tombol mulai ujian dan memilih jawaban dari masing-masing soal yang ditampilkan	Peserta dapat memilih jawaban pada setiap soal yang ditampilkan	Peserta dapat memilih jawaban pada setiap soal yang ditampilkan	Diterima
			Peserta selesai ujian	Tampil soal-soal yang harus	Peserta memilih	Peserta dapat melihat hasil jawaban yang	Peserta dapat melihat hasil	Diterima

No	User	Use Case	Fungsi yang Diuji	Kondisi Awal	Input	Hasil Yang diharapkan	Output	Hasil Test
				dikerjakan oleh peserta	tombol selesai ujian	telah dijawab pada setiap soal yang ditampilkan	jawaban yang telah dijawab pada setiap soal yang ditampilkan	
14.	Peserta	Menampilkan Hasil Ujian	Hasil ujian pada tipe ujian UN	Tampil menu daftar ujian	Peserta memilih menu daftar ujian	Tampil hasil ujian “Belum Dinilai” dan peringkat sementara peserta	Tampil hasil ujian “Belum Dinilai” dan peringkat sementara peserta	Diterima
			Hasil ujian pada tipe ujian Sertifikasi	Tampil menu daftar ujian	Pilih tombol Kembali ke Daftar Ujian	Tampil hasil ujian peserta (“Lulus” atau “Tidak Lulus”) dan peringkat sementara peserta	Tampil hasil ujian peserta (“Lulus” atau “Tidak Lulus”) dan peringkat sementara peserta	Diterima

## V.6 Analisa Hasil Pengujian

Dari hasil pengujian yang telah dilakukan dapat diambil kesimpulan sebagai analisa dari hasil pengujian, antara lain:

1. Aplikasi dapat melakukan proses *login*, baik untuk admin, pembuat soal, penguji dan peserta.
2. Aplikasi mampu mengelola informasi profil organisasi.
3. Aplikasi mampu mengelola data *user* (admin, pembuat soal dan penguji).
4. Jika tipe ujian UN, aplikasi mampu mengelola program studi suatu organisasi (jika ada).
5. Aplikasi mampu mengelola konfigurasi profil ujian berupa pengaturan subjek beserta tingkat kesulitannya.
6. Aplikasi mampu mengelola konfigurasi profil ujian berupa pengaturan kelulusan berdasarkan nilai (UN atau Sertifikasi) atau kuota.
7. Aplikasi mampu mengelola aturan distribusi.
8. Aplikasi mampu membuat aturan nilai suatu ujian setiap subjek beserta tingkat kesulitannya.
9. Aplikasi mampu mengelola soal dan jawaban beserta kunci jawabannya.
10. Aplikasi mampu menangani registrasi peserta yang dilakukan secara *online*.
11. Aplikasi mampu menampilkan soal siap uji.
12. Aplikasi mampu mengelola waktu pelaksanaan ujian
13. Aplikasi menyediakan hak akses soal dan menyimpan jawaban peserta
14. Aplikasi mampu melakukan kalkulasi nilai hasil ujian peserta.
15. Aplikasi mampu menampilkan hasil ujian peserta sesuai konfigurasi profil ujian yang telah ditentukan.
16. Aplikasi mampu menampilkan *ranking* atau peringkat yang diraih peserta

## **Bab VI Kesimpulan dan Saran**

### **VI.1 Kesimpulan**

Setelah di lakukan tahap analisis, perancangan, implementasi dan pengujian pada aplikasi Pengembangan Sistem Ujian *Online* Berbasis Objek, maka kesimpulan yang didapat adalah:

1. Aplikasi ini dapat mempermudah peserta yang ingin mengikuti suatu ujian tanpa terbatas oleh letak geografis karena proses ujian dilakukan secara *online*.
2. Aplikasi ini dapat mengelola tipe ujian UN dan Sertifikasi.
3. Aplikasi dapat melakukan pengoreksian jawaban secara otomatis karena dilakukan oleh sistem.
4. Aplikasi dapat menampilkan hasil ujian dan peringkat yang diraih peserta.
5. Aplikasi ini dapat digunakan oleh instansi manapun yang akan melaksanakan ujian dengan terlebih dahulu melakukan instalasi aplikasi ini.
6. Aplikasi ini dikembangkan dengan menggunakan rancangan berorientasi objek.

### **VI.2 Saran**

Penyusun menyadari masih terdapat kekurangan-kekurangan dalam aplikasi Pengembangan Sistem Ujian *Online* Berbasis Objek ini. Oleh karena itu, aplikasi ini dapat dikembangkan kembali. Adapun saran yang dapat diberikan untuk penyempurnaan dari aplikasi ini adalah:

1. Aplikasi mampu mengelola soal ujian dalam bentuk *essay*.
2. Aplikasi dapat melakukan pengaturan waktu ujian pada setiap soal yang ditampilkan kepada peserta.

## DAFTAR PUSTAKA

1. Hariyanto, Bambang, "*Esensi-esensi Bahasa Pemrograman Java*", Informatika Bandung, Bandung: Oktober 2003.
2. Azis, M. Farid, "*Object Oriented Programming dengan PHP 5*", PT Elex Media Komputindo, Jakarta: 2005.
3. Achmad Solichin, "*Diktat Kuliah Pemrograman Web 2 ver 1.0*". Tersedia: [achmatim@bl.ac.id](mailto:achmatim@bl.ac.id)
4. Purbasari, Ayi, S.T., M.T., "*Materi 4 Kuliah IF-681 RPLT*". Tersedia: <http://if-unpas.org/home/downloads/Handout/Hand%20Out%20RPLT%202008/Materi%20Kuliah%20IF-681%20-%204%20%28Model%20Analisis%29.pdf>
5. Tarmizi, Muhammad, "*Aplikasi Ujian Test Online*", Tugas Akhir, Batam: Politeknik Batam: 2005
6. Fahmi, Irfan M., "*Aplikasi Simulasi Ujian SPMB*", Tugas Akhir, Batam: Politeknik Batam: 2006
7. Karmila, "*Aplikasi Tes UMPB Berbasis Komputer*", Tugas Akhir, Batam: Politeknik Batam: 2008
8. Januardi dan Jonatan, "*Aplikasi Ujian Online*", Tugas Akhir, Batam: Politeknik Batam: 2010
9. Slide Diktat Kuliah, "*Business Modeling*", Minggu 12 dan 13
10. <http://misdek.wordpress.com/2006/11/16/mysql-vs-postgresql/>