

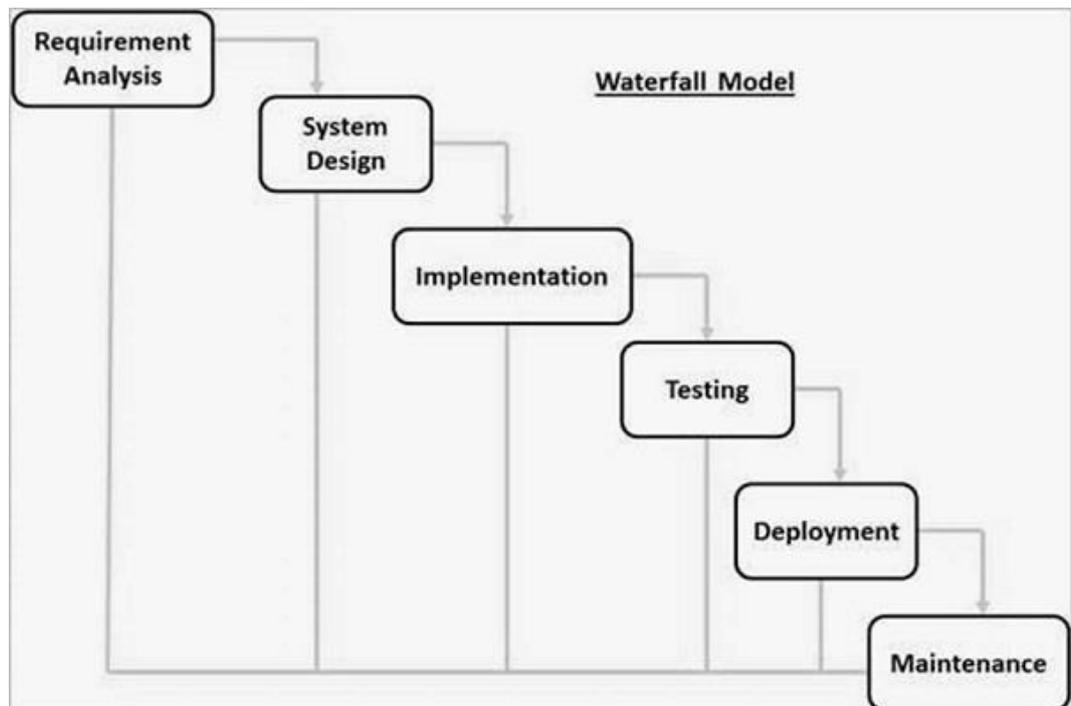
## BAB III

### METODE PENELITIAN

#### 3.1 Metode Pengembangan Sistem

Pada penelitian ini diterapkan metode SDLC (*Software Development Life Cycle*) untuk pengembangan sistemnya. Metode ini merupakan siklus pengembangan sistem pada perangkat lunak. Terdapat beberapa tahapan di dalamnya, di antaranya: perencanaan, analisis, desain, implementasi, pengujian dan pemeliharaan.

Model proses yang akan digunakan adalah model *waterfall*. Model ini merupakan model yang diperkenalkan oleh Wiston Royce pada tahun 1970. Proses yang ada pada model ini bersifat sekuensial dan dibagi menjadi beberapa fase seperti yang ditunjukkan pada gambar di bawah ini:



Gambar 3. 1 Model *Waterfall*

### **3.1.1 Analisis Kebutuhan**

Pada tahap ini peneliti mengumpulkan data terkait kebutuhan pembangunan aplikasi. Data diambil secara langsung dari narasumber berdasarkan permasalahan yang ada. Data yang terkumpul digunakan untuk menganalisis kebutuhan sistem dan fungsi pada *software* yang akan dibangun. Dalam tahapan ini peneliti juga menentukan target pengguna aplikasi yang merupakan tingkatan SMA keatas.

### **3.1.2 Desain Sistem**

Dalam tahapan ini hasil dari analisis kebutuhan diolah kemudian disajikan dalam sebuah rancangan sistem yang akan digunakan untuk kebutuhan proses pengembangan *software*. Di tahap ini juga dilakukan perencanaan terkait bahasa pemrograman dan *database* yang akan digunakan.

### **3.1.3 Implementasi**

Hasil dari tahap perancangan diimplementasikan pada tahap ini. Setiap fungsi dibangun berdasarkan kebutuhan yang ada pada rancangan sistem. Sistem dibangun menggunakan bahasa pemrograman dan *database* sesuai dengan yang direncanakan pada tahap sebelumnya.

### **3.1.4 Pengujian**

Dalam tahapan ini hasil dari pembangunan sistem diuji berdasarkan rancangan dan kebutuhan yang ada pada tahapan-tahapan sebelumnya untuk memastikan bahwa sistem sudah sesuai dengan kebutuhan dan tidak ada *error*.

### 3.2 Alat dan Bahan

#### 3.2.1 Alat

##### A. Perangkat Keras

Spesifikasi perangkat keras yang digunakan oleh peneliti untuk mengembangkan sistem, disajikan pada tabel 3.1 berikut:

**Tabel 3. 1** Perangkat keras yang digunakan.

Perangkat Keras	Informasi
RAM	4 GB
Prosesor	Intel(R) Core(TM)i3-4005U CPU @1.70 GHz
<i>Hard Disk</i>	500 GB

##### B. Perangkat Lunak

Tabel 3.2 dibawah menunjukkan perangkat lunak yang digunakan oleh peneliti dalam pengembangan sistem.

**Tabel 3. 2** Perangkat lunak yang digunakan.

No	Nama Perangkat Lunak	Keterangan
1	Windows 10 Enterprise	Sistem operasi yang digunakan untuk menjalankan perangkat keras.
2	Android Studio	Perangkat ini digunakan untuk mengembangkan aplikasi berbasis <i>mobile</i> Android.

No	Nama Perangkat Lunak	Keterangan
4	PHP Storm	Perangkat ini digunakan untuk membangun <i>service</i> menggunakan <i>script</i> PHP.
5	XMPP Server	Perangkat ini digunakan sebagai <i>host server</i> lokal untuk mendistribusikan servis secara lokal sekaligus penyedia MySQL <i>database</i> .
6	Microsoft Visio 2013	Perangkat ini digunakan dalam hal perancangan sistem menyediakan berbagai <i>template</i> untuk berbagai rancangan sistem.
7	Balsamiq Mockups 3	Perangkat ini digunakan untuk membuat rancangan antarmuka untuk <i>client</i> .

### 3.2.2 Bahan

Bahan yang digunakan adalah kumpulan data yang didapat melalui wawancara terhadap narasumber secara langsung dan hasil dari observasi. Peneliti juga melakukan obeservasi terhadap sistem yang sama yang bersumber dari *web*.

### **3.3 Metode Pengumpulan Data**

#### **3.3.1 Observasi**

Observasi merupakan metode pengumpulan data yang dilakukan dengan melakukan pengamatan terhadap objek penelitian yang sudah ada. Pada kasus ini peneliti menerapkan observasi partisipasi yang merupakan pengamatan yang dilakukan dengan cara peneliti mengambil bagian dalam objek observasi. Kegiatan ini bertujuan untuk mengamati sebuah proses dari sistem yang sudah ada.

Objek yang diamati oleh peneliti adalah *web e-learning Moodle*. Peneliti mengambil bagian secara langsung sebagai siswa dan instruktur untuk mengamati proses serta menganalisa sistem yang ada pada objek observasi untuk mengumpulkan data yang dibutuhkan.

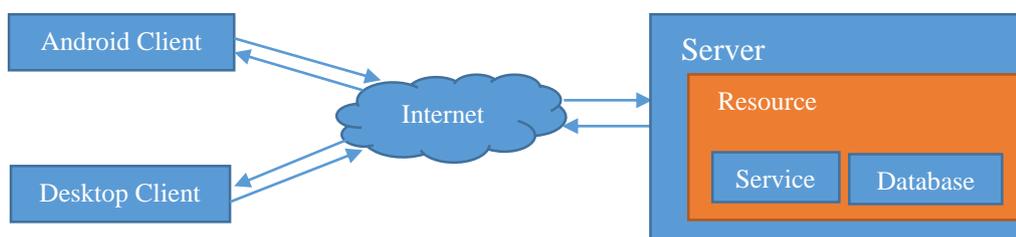
#### **3.3.2 Wawancara**

Peneliti melakukan wawancara terhadap narasumber secara langsung untuk mengumpulkan data yang dibutuhkan guna proses pengembangan sistem yang akan dilaksanakan.

### 3.4 Analisis Data

#### 3.4.1 Arsitektur Sistem

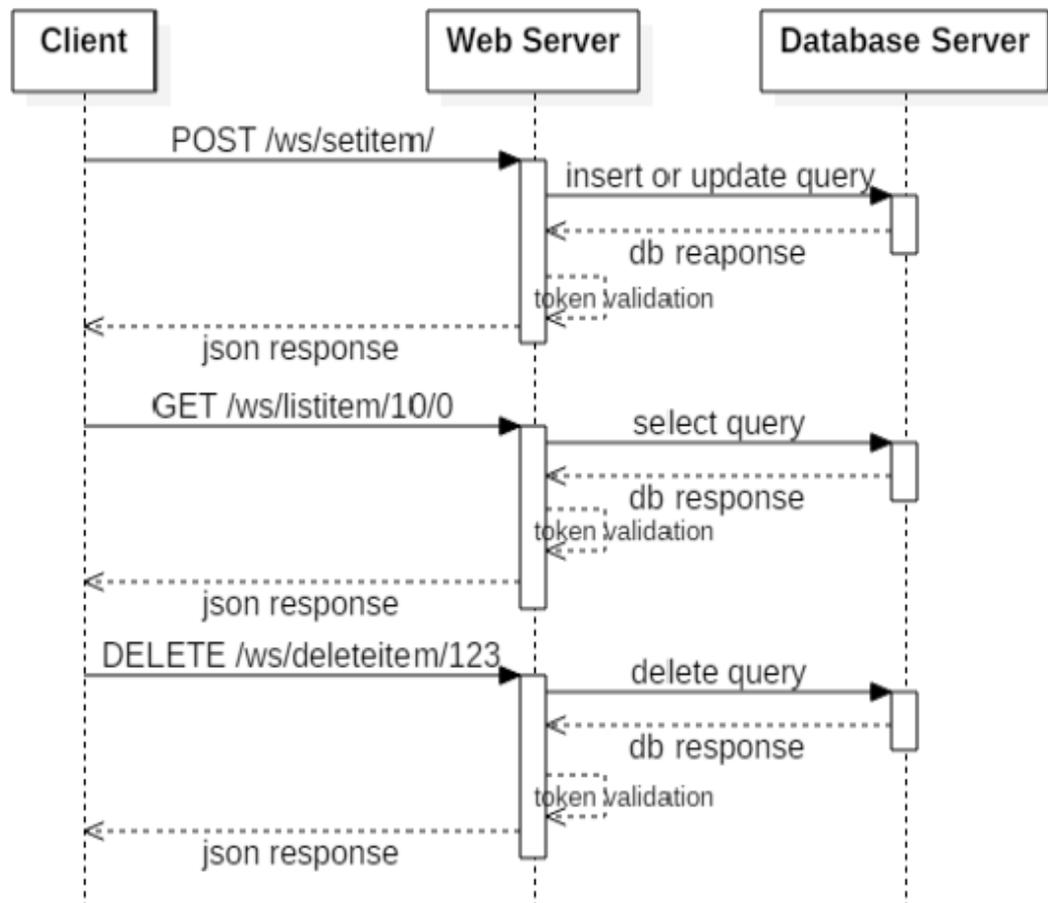
Sistem yang dibangun terdiri dari *server* dan *client*. Secara garis besar *server* bertugas menjalankan *service* dan *database*. *Service* berfungsi mengelola setiap data yang ada pada *database* dan mengubahnya menjadi *resource*. *Resource* tersebut didistribusikan oleh *server* kepada *client*, sehingga *client* yang terhubung pada *server* dapat menggunakan dan mengelola *resource* tersebut. Terdapat dua jenis *client* pada sistem ini, yaitu *client mobile* Android dan *client desktop*. Setiap *client* memiliki fungsi dan tugasnya masing-masing. Bila digambarkan maka sistem ini akan menjadi seperti gambar 3.2 di bawah ini:



**Gambar 3. 2** Desain Arsitektur Sistem

*Server* yang digunakan adalah XAMPP *server* dan basis data yang digunakan adalah MySQL *server* yang secara otomatis terpasang di XAMPP *server*. Sistem distribusi *service* menggunakan RESTful yang dibangun dengan *script* PHP sehingga *resource* disajikan dalam bentuk *web http* dengan format JSON. Alasan peneliti memilih protokol ini karena sistem yang dibangun memiliki *resource* yang sangat besar sehingga akan sangat optimal jika komunikasi antara *client* dengan *server* menggunakan RESTful. Format JSON sendiri dipilih karena sangat mendukung untuk perangkat Android yang sudah banyak penyedia *library*-nya.

Adapun gambaran mengenai RESTful beserta gambaran proses transaksinya adalah sebagai berikut:



**Gambar 3. 3** Proses Transaksi RESTful Web Service

RESTful memberikan abstraksi untuk mempublikasikan informasi dan memberikan akses *remote* ke lapisan *Data Access*. Isi *Data Access* yang dianggap sebagai sumber daya yang direferensikan oleh URI sederhana dan diwakili oleh JSON. Misal *POST/ws/setitem* adalah URI yang digunakan *client* untuk melakukan *request* menyimpan atau mengubah data dari *item* setelah itu *web server* memproses *request* tersebut dengan mencocokkan URI yang dipanggil terhadap *service* kemudian *service* akan menjalankan *query* untuk eksekusi ke basis data dan menjalankan perintah sesuai fungsi.

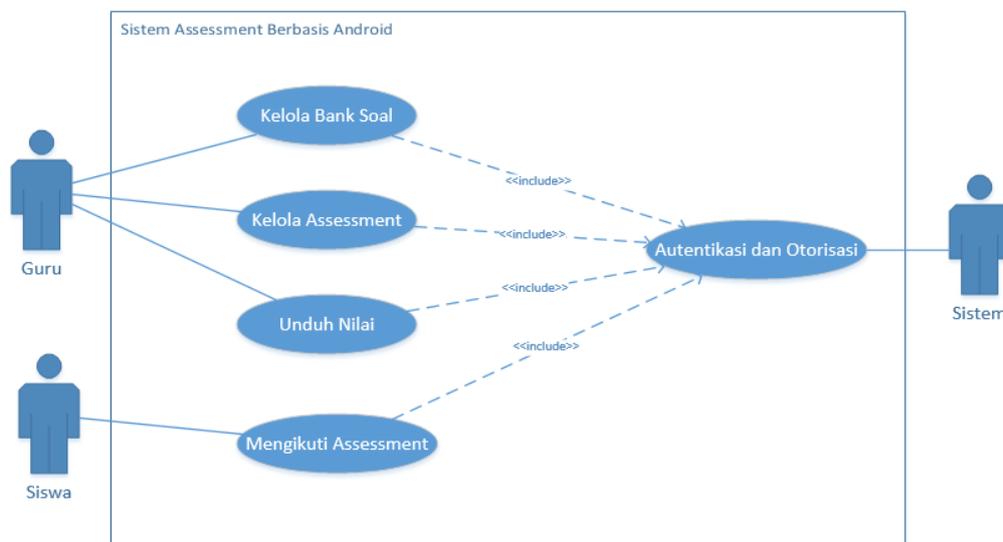
### 3.4.2 Analisis Kebutuhan Sistem

Berikut ini adalah kebutuhan sistem yang didapat dari hasil analisis yang telah dilakukan:

1. Sistem harus memiliki fitur unggah bank soal dalam bentuk *file excel* terkait pekerjaan instruktur lebih mudah melalui *desktop*.
2. Sistem harus memiliki 2 jenis *client* yang berbeda agar pengguna dapat mengoperasikan fungsi berdasarkan tugas dan *role*-nya masing-masing.
3. Sistem harus memiliki fitur cari *assessment* berdasarkan *id assessment* agar siswa mengerjakan *assessment* sesuai dengan yang diinstruksikan oleh instruktur.
4. Sistem harus memiliki fitur *enrollment* ke *assessment* untuk memastikan bahwa *assessment* yang harus dikerjakan sudah sesuai dengan instruksi dari instruktur.

### 3.5 Rancangan Sistem

#### 3.5.1 Diagram *Use Case*

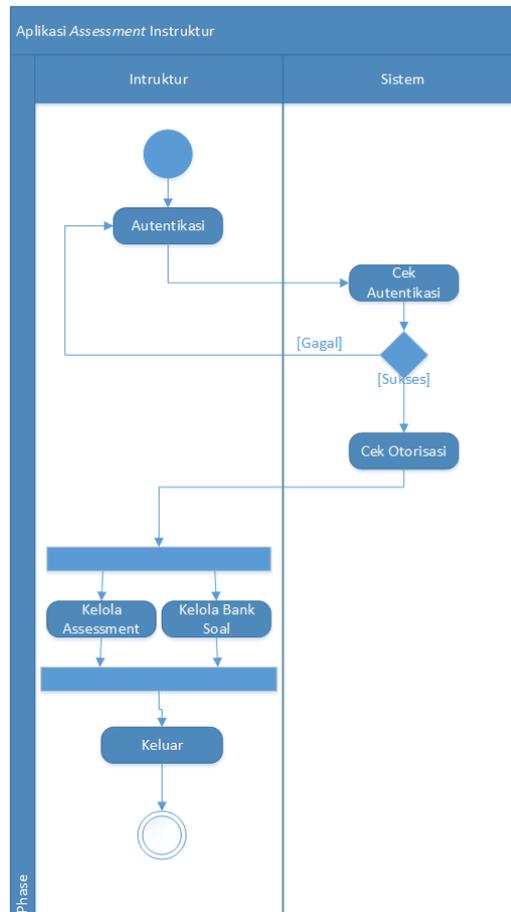


**Gambar 3. 4** Diagram *Use Case* Sistem

Seperti yang ditunjukkan pada gambar 3.4 sistem ini memiliki 3 aktor, yaitu Instruktur, Siswa, dan Sistem itu sendiri. Setiap aktor memiliki perannya masing-masing. Instruktur pada sistem ini mampu mengelola bank soal, *assesment* dan mengunduh nilai *assesment*. Siswa berperan sebagai aktor yang melaksanakan ujian. Khusus untuk sistem, aktor ini berperan menangani autentikasi dan otorisasi sehingga instruktur dan siswa mendapatkan peran yang sesuai.

### 3.5.2 Diagram Activity

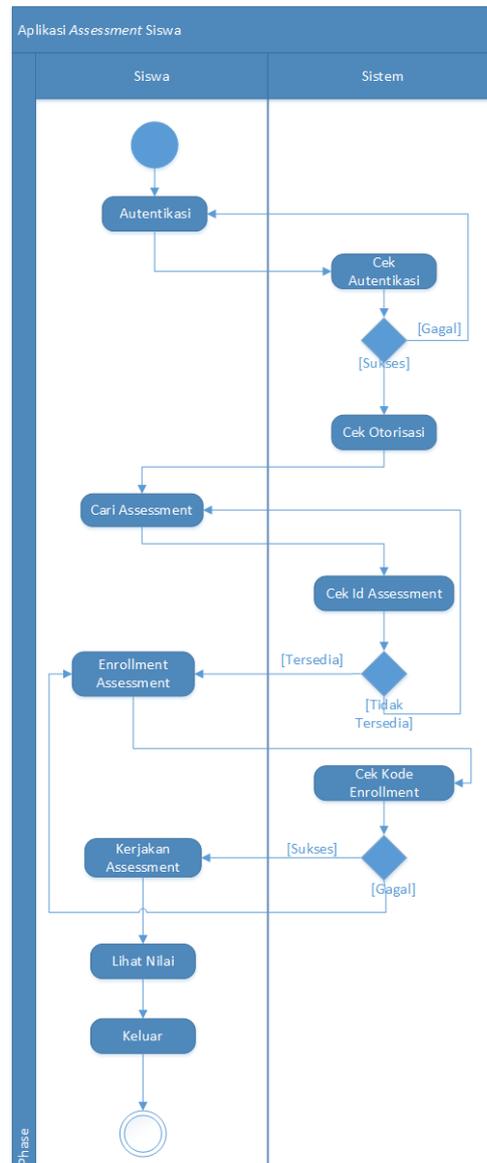
#### A. Instruktur



**Gambar 3.5** Diagram Activity Instruktur

Seperti yang ditunjukkan pada gambar 3.5 proses instruktur memiliki beberapa aktivitas akan tetapi harus melalui proses otorisasi dan autentikasi yang merupakan bagian dari aktivitas pada proses sistem agar dapat menjalankan aktivitas lainnya. Setelah melalui proses tersebut terdapat pilihan kelola *assessment* dan kelola bank soal. Kelola *assessment* sudah mencakup proses tambahkan *assessment* dan unduh nilai sedangkan proses kelola bank soal sudah termasuk proses tambahkan bank soal dan unggah bank soal dari *file excel*.

## B. Siswa

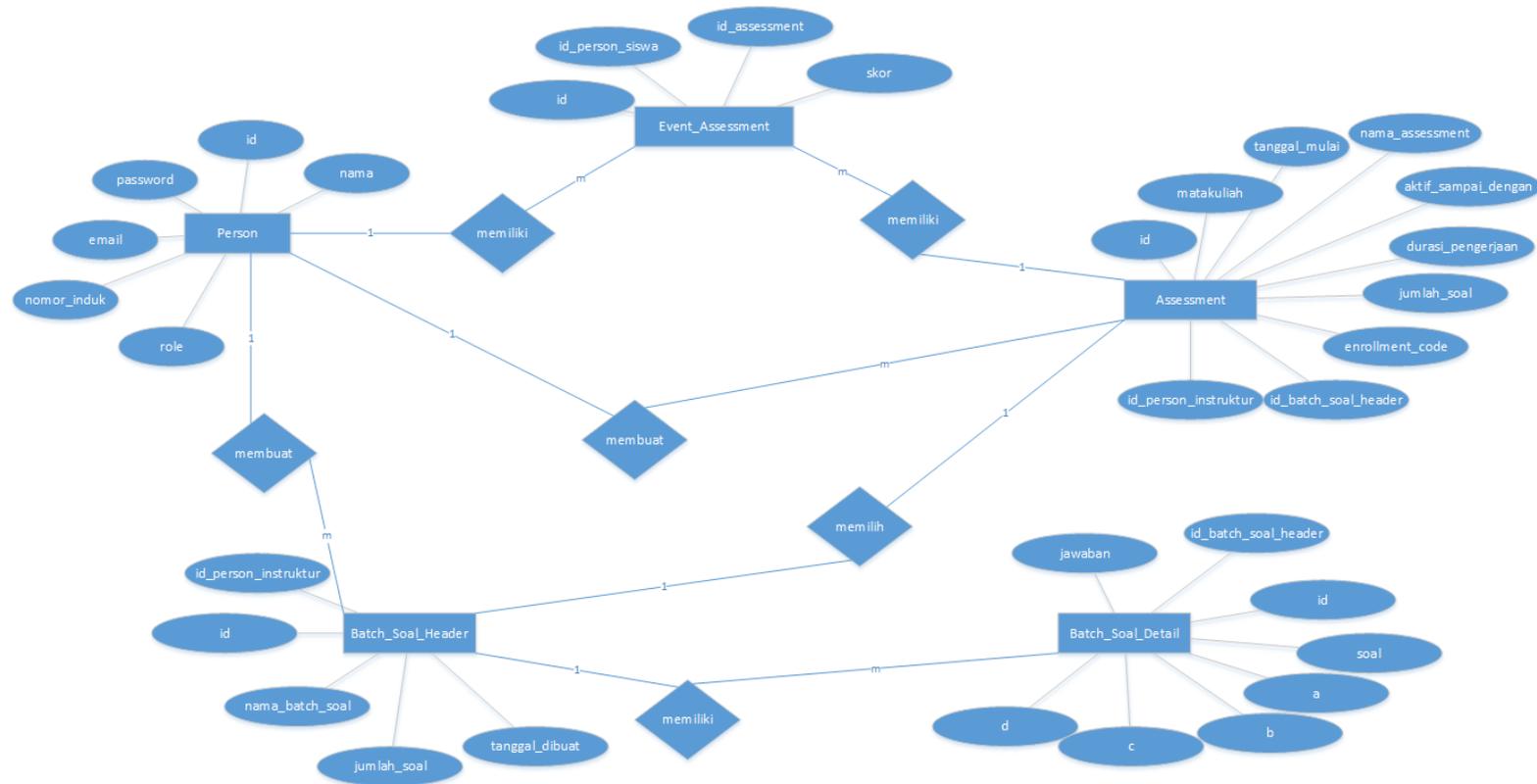


**Gambar 3. 6** Diagram Activity Siswa

Pada proses siswa memiliki aturan yang sama dengan instruktur yaitu melalui autentikasi dan otorisasi pada proses sistem. Seperti yang ditunjukkan pada gambar 3.6 setelah melalui proses tersebut siswa dapat melakukan pencarian *assessment* berdasarkan *id assessment* setelah itu sistem bekerja untuk mencari *assessment* sesuai dengan *id* yang dimasukkan, bila tersedia user dapat melanjutkan proses ke aktivitas *enrollment assessment* bila tidak tersedia

siswa akan dikembalikan ke aktivitas cari *assessment*. Aktivitas *enrollment assessment* adalah verifikasi terhadap *assessment* yang akan dikerjakan. Setiap *assessment* memiliki kode unik yang digunakan oleh sistem untuk proses verifikasi tersebut. Sistem akan mencocokkan kode yang dimasukkan oleh siswa dengan kode yang ada pada *assessment* jika verifikasi gagal atau tidak sesuai siswa akan diminta memasukkan ulang kode, jika berhasil siswa akan melanjutkan ke aktivitas kerjakan *assessment*. Aktivitas lihat nilai dapat dilakukan setelah siswa menyelesaikan *assessment*.

### 3.5.3 Diagram ERD



Gambar 3. 7 Diagram ERD Sistem

Berikut adalah penjelasan terkait diagram relasi entitas yang telah dipaparkan pada gambar 3.7:

### 1. Entitas

- a. Person
- b. Assessment
- c. Event\_Assessment
- d. Batch\_Soal\_Header
- e. Batch\_Soal\_Detail

### 2. Atribut

- a. Person: id, nama, password, email, nomor\_induk, role.
- b. Assessment: id, nama\_assessment, matakuliah, tanggal\_mulai, aktif\_sampai\_dengan, durasi\_pengerjaan, jumlah\_soal, enrollment\_code. id\_person\_instruktur, id\_batch\_soal\_header.
- c. Event\_Assessment: id, skor, id\_person\_siswa, id\_assessment.
- d. Batch\_Soal\_Header: id, nama\_batch\_soal, jumlah\_soal, tanggal\_dibuat, id\_person\_instruktur.
- e. Batch\_Soal\_Detail: id, soal, a, b, c, d, jawaban, id\_batch\_soal\_header.

### 3. Relasi

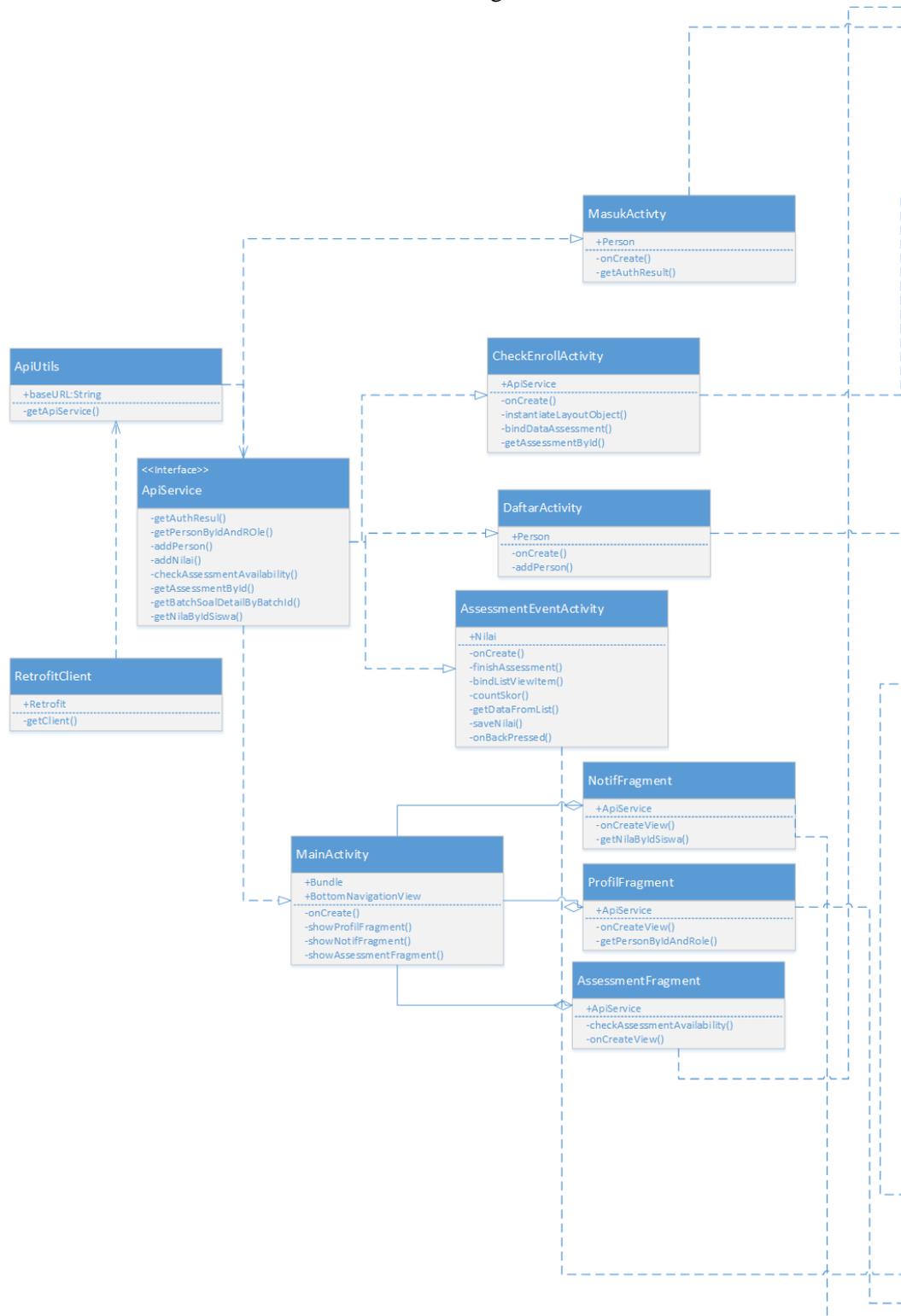
- a. Entitas Person memiliki relasi *one to many* terhadap entitas Event\_Assessment.
- b. Entitas Person memiliki relasi *one to many* terhadap Entitas Assessment.
- c. Entitas Person memiliki relasi *one to many* terhadap Entitas Batch\_Soal\_Header.

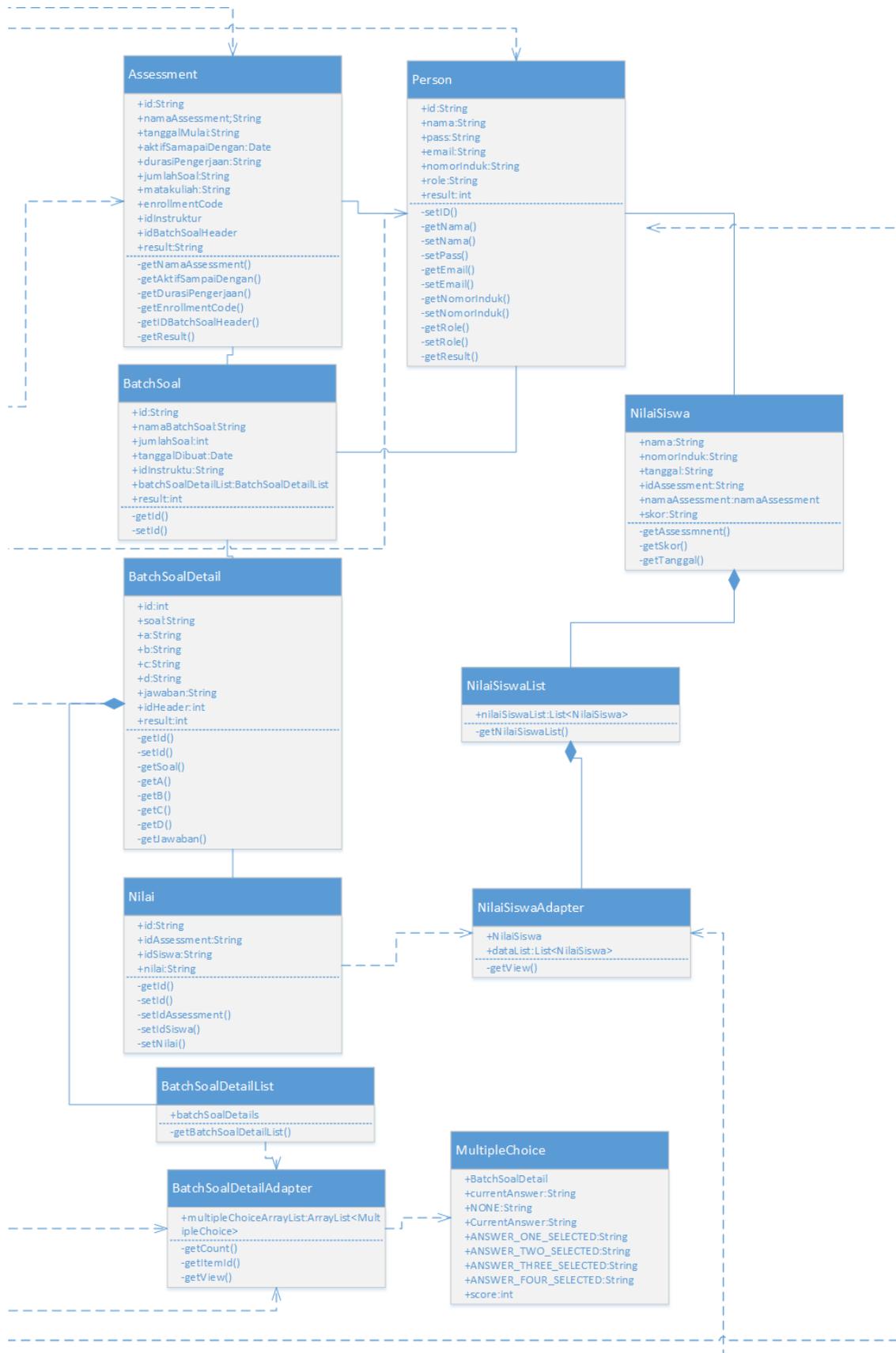
- d. Entitas Assessment memiliki relasi *one to many* terhadap entitas Event\_Assessment.
- e. Entitas Assessment memiliki relasi *one to one* terhadap entitas Batch\_Soal\_Header.
- f. Entitas Batch\_Soal\_Header memiliki relasi *one to many* terhadap entitas Batch\_Soal\_Detail.

### 3.5.4 Diagram Class

#### A. Diagram Class Siswa

Gambar 3. 8 Diagram Class Siswa

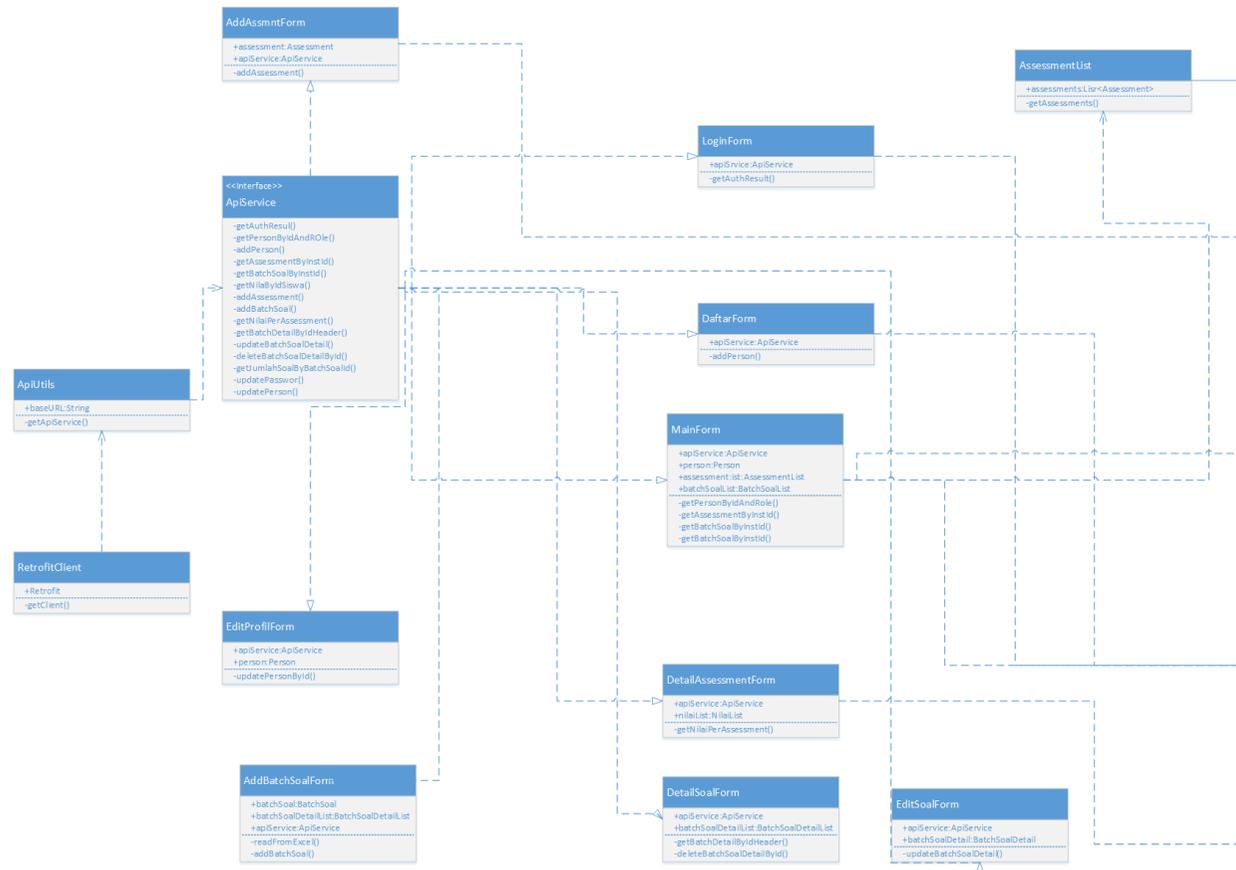


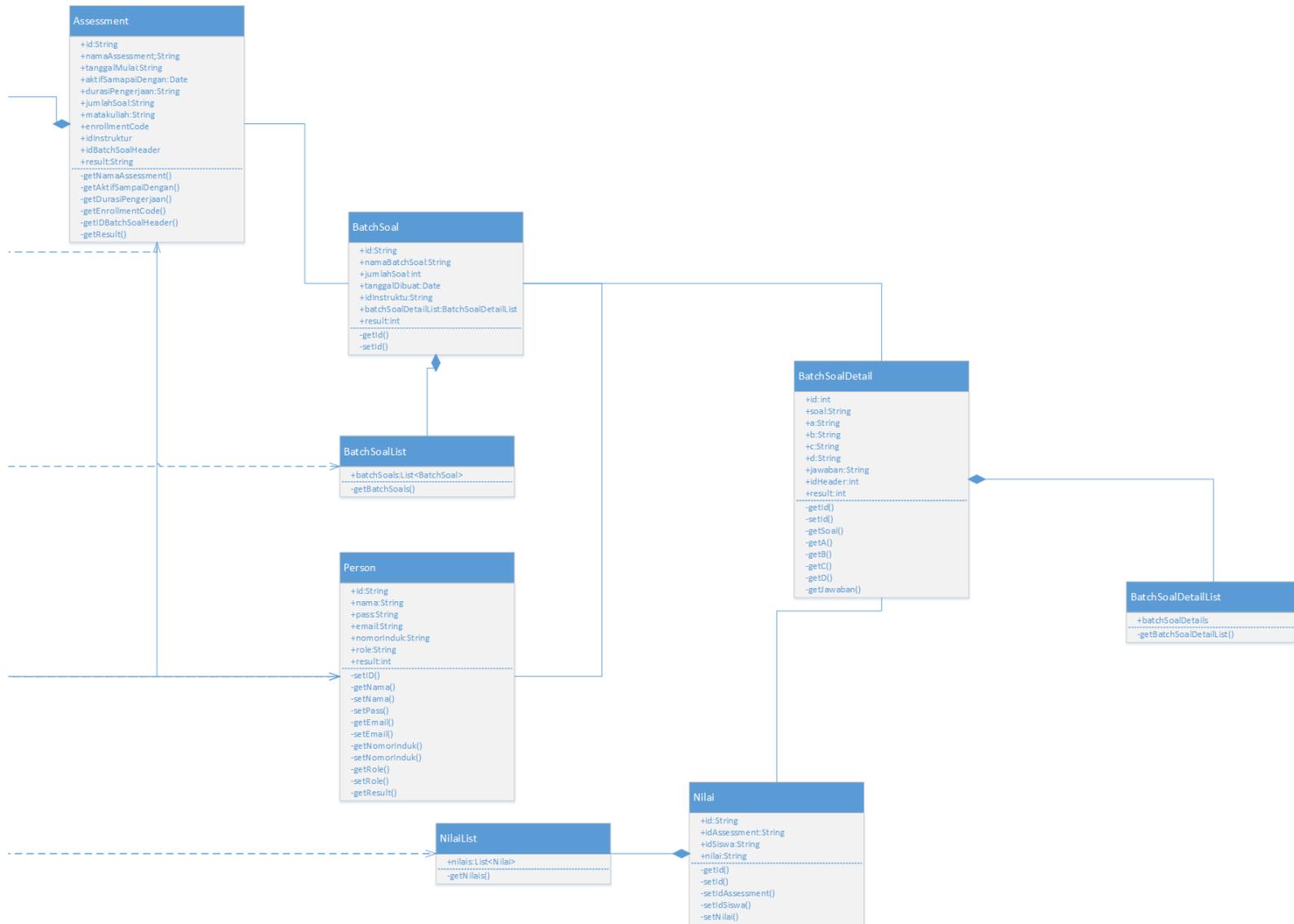


Gambar 3.8 menunjukkan Diagram *Class* Siswa. *Class* untuk bagian ini digolongkan menjadi beberapa bagian diantaranya; *service*, *model*, dan *activity*. *Class service* bertanggung jawab sebagai yang menangani API untuk *client*, jadi segala macam transaksi terhadap *resource* yang ada pada *web service* dikelola secara langsung pada *class* ini. *Service* direpresentasikan dalam bentuk *interface class* setiap fungsi yang ada pada *class* ini digunakan oleh *class* lainnya untuk mengakses *web service*. Untuk *class model* sendiri digunakan sebagai mapping terhadap *resource* yang dikelola misal, data *person* yang diterima dalam bentuk JSON akan dipetakan terhadap *class model person*. Adapun *class activity* difungsikan sebagai representasi *resource* untuk bagian antarmuka *client android*.

B. Diagram Class Instruktur

Gambar 3.9 Diagram Class Instruktur



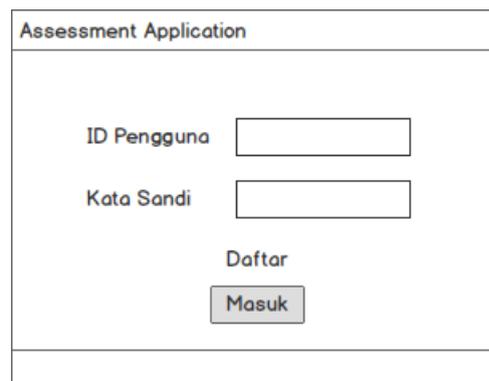


Sama halnya dengan Diagram *Class* Siswa, Diagram *Class* Instruktur yang ditunjukkan pada gambar 3.9 di atas digolongkan menjadi beberapa bagian yang sama yaitu; *service*, *model* dan *form*. Sedikit berbeda dengan yang sebelumnya, terdapat *class form* yang digunakan untuk *client* yang berbentuk *desktop*. Secara keseluruhan *class service* maupun *model* memiliki fungsi dan pekerjaan yang sama.

### 3.6 Rancangan Antarmuka

#### 3.6.1 Aplikasi *Desktop* Instruktur

##### A. *Form Login*



The image shows a login form window titled "Assessment Application". It contains two text input fields: "ID Pengguna" and "Kata Sandi". Below these fields are two buttons: "Daftar" and "Masuk". The "Masuk" button is highlighted with a grey background.

**Gambar 3. 10** *Form Login* Instruktur

*Form login* yang ditunjukkan pada gambar 3.10 akan muncul saat pertamakali aplikasi dibuka. *User* akan diminta untuk memasukkan *ID* dan kata sandi. Setelah *field* dilengkapi, dengan menekan tombol masuk, sistem akan langsung memvalidasi data yang dimasukkan jika berhasil maka *form* utama akan ditampilkan. Terdapat teks daftar yang berfungsi untuk mengalihkan *user* ke *form* daftar.

## B. Form Daftar

Window Name

**Daftar**

ID Pengguna

Nama Pengguna

Nomor Induk

E-mail

Kata Sandi

Konfirmasi Kata Sandi

**Gambar 3.11** Form Daftar Instruktur

Form daftar yang ditunjukkan pada gambar 3.11 digunakan oleh *user* untuk mendaftarkan diri apabila *user* belum memiliki akun untuk mengakses aplikasi. Setelah *field* diisi dengan lengkap dan benar, dengan menekan tombol daftar sistem akan langsung memasukkan data ke dalam basisdata.

## C. Form Utama (Tab Assessment)

Window Name

Pengaturan

Assessment	ID	Nama Assessment	Tanggal Dibuat	Enrollment Code	.....
Batch Soal					
Keluar					

**Gambar 3.12** Form Utama (Tab Assessment)

Gambar 3.12 menunjukkan *form* utama bagian *assessment*. Pada bagian ini *user* dapat melihat daftar *assessment* yang telah dibuat. Selain itu terdapat 2 aksi yang dapat dilakukan seperti tambah *assessment* dan buka *detail*

*assessment*. Ketika *user* memilih tambah *assessment* maka akan dialihkan ke *form* tambah *assessment*. Tombol *detail assessment* akan berfungsi untuk membuka rincian atau informasi *assessment* yang dipilih dari daftar

#### D. Form Utama (Tab Batch Soal)

Assessment	ID	Nama Bank Soal	Jumlah Soal	.....
Batch Soal				

**Gambar 3. 13** Form Utama (Tab Batch Soal)

Gambar 3.13 diatas menunjukkan *form* utama bagian *batch* soal. Pada bagian ini *user* dapat melihat daftar *batch* soal yang pernah dibuat. Terdapat 2 aksi yang dapat dipilih tambah *batch* soal dan *detail batch* soal. Sama seperti bagian *assessment*, dengan menekan tombol tambah, *user* akan dialihkan ke *form* tambah *batch* soal, sedangkan dengan memilih *tombol* detail, *user* akan dialihkan ke *form detail batch* soal yang berisi rincian atau informasi mengenai *batch* soal yang dipilih.

### E. Form Utama (Tab Keluar)

The screenshot shows a web application window titled "Window Name". At the top left, there is a circular profile picture placeholder. To its right, the text "Nama Pengguna", "Nomor Induk", and "ID Pengguna" is displayed. A "Pengaturan" button is located in the top right corner. Below this is a table with three tabs: "Assessment", "Batch Soal", and "Keluar". The "Keluar" tab is currently selected. A modal dialog box titled "Informasi" is overlaid on the table, asking "Apakah anda yakin ingin keluar?" with "Yes" and "No" buttons. At the bottom of the window, there are "Tambah" and "Detail" buttons.

**Gambar 3.14** Form Utama (Tab Keluar)

Form utama bagian keluar yang ditunjukkan pada gambar 3.14 adalah opsi bila *user* ingin keluar dari aplikasi. Ketika *user* memilih *tab* ini maka akan langsung muncul *dialog* untuk konfirmasi apakah *user* ingin keluar aplikasi. Jika memilih ya maka *user* akan dikembalikan ke *form login* sebaliknya jika tidak *user* akan tetap berada di *form* utama.

### F. Form Detail Assessment

The screenshot shows a form titled "Detail Assessment". It contains four text input fields: "Kode Assessment", "Nama Assessment", "Tanggal Selesai", and "Kode Enrollment". A "Unduh Excel" button is positioned to the right of the "Kode Enrollment" field. Below these fields is a section titled "Daftar siswa yang sudah mengerjakan" followed by a table. The table has five columns: "ID", "Nama Siswa", "Nomor Induk", "Skor", and an empty column. The table is currently empty. At the bottom of the form, there are "Refresh" and "Tutup" buttons.

**Gambar 3.15** Form Detail Assessment

Gambar 3.15 menunjukkan *detail* dari *assessment* yang berisi informasi dari *assessment* yang sedang dibuka. Pada *form* ini juga terdapat tabel yang

berisi daftar mahasiswa yang telah mengikuti *assessment*. Tabel tersebut dapat diunduh dalam bentuk format *file excel*.

### G. Form Tambah Assessment

**Gambar 3. 16** Form Tambah Assessment

Gambar 3.16 merupakan *form* tambah *assessment*. *Form* ini memungkinkan *user* untuk menambah *assessment* baru. Terdapat beberapa isian yang harus dilengkapi oleh *user* diantaranya nama *assessment*, aktif sampai dengan, durasi, jumlah soal, matakuliah, dan *batch* soal.

### H. Form Detail Batch Soal

**Gambar 3. 17** Form Detail Batch Soal

*Form detail batch soal* berisi informasi mengenai *batch soal* yang sedang dibuka. Seperti yang ditunjukkan pada gambar 3.17 terdapat tabel yang berisi daftar soal dari *batch soal* yang sedang dibuka. Setiap soal dapat dihapus dan diubah.

### I. *Form Tambah Batch Soal*

The screenshot shows a window titled "Tambah Batch Soal". Inside, there is a text input field labeled "Nama Batch Soal". Below it is a file selection field labeled "Pilih File" with a "Browse" button to its right. At the bottom of the window, there are two buttons: "Simpan" and "Batal".

**Gambar 3.18** *Form Tambah Batch Soal*

Gambar 3.18 menunjukkan *form* tambah *batch soal* yang. Di dalam *form* ini terdapat 2 *field* yang harus diisi yaitu nama *batch soal* dan *file batch soal*. Khusus untuk *batch soal* terdapat tombol *browse* yang akan mengalihkan *user* ke jendela pilih *file*. Format *file* yang dipilih harus dalam bentuk *excel*.

### J. *Form Ubah Soal*

The screenshot shows a window titled "Tambah Batch Soal" (though the caption says it's the 'Form Ubah Soal'). It contains a large text area labeled "Soal". Below it are four radio button options labeled "A", "B", "C", and "D". At the bottom left, there is a "Jawaban" dropdown menu with a "ComboBox" label and a list of options: "A", "B", "C", and "D". At the bottom right, there are two buttons: "Simpan" and "Batal".

**Gambar 3.19** *Form Ubah Soal*

Gambar 3.19 menunjukkan *form* ubah soal. Di dalam *form* ini *user* dapat mengubah soal apabila terdapat kesalahan pada soal.

### K. Form Ubah Profil

**Gambar 3. 20** *Form* Ubah Profil

Seperti yang ditunjukkan pada gambar 3.12 *form* utama sebelumnya terdapat tombol pengaturan pada bagian kanan atas *form*. Tombol tersebut mengalihkan *user* menuju *form* ubah profil seperti yang ditunjukkan pada gambar 3.20. Ketika terbuka semua *field* pada *form* ini akan secara otomatis terisi sesuai data *user*. Semua *field* dapat diubah kecuali bagian kata sandi. Agar kata sandi dapat diubah, *user* harus menekan tombol *ubah* yang berfungsi untuk membuka *form* ubah kata sandi.

### L. Form Ubah Kata Sandi

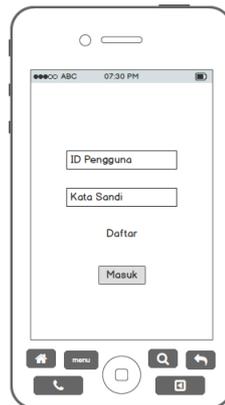
**Gambar 3. 21** *Form* Ubah Kata Sandi

Gambar 3.21 menunjukkan *form* ubah kata sandi. Dengan memasukkan kata sandi lama lalu kata sandi baru setelah itu lakukan konfirmasi terhadap kata

sandi baru maka sistem akan melakukan validasi terhadap data yang dimasukkan ketika *user* menekan tombol simpan.

### 3.6.2 Aplikasi Mobile Siswa

#### A. Login Activity



Gambar 3. 22 Login Activity

Gambar 3.22 menunjukkan bagian *login activity*. Pada *activity* ini *user* akan diminta untuk memasukkan *ID* dan kata sandi untuk proses autentikasi dan otorisasi. Dalam *activity* ini juga terdapat teks *link* daftar yang akan mengalihkan *user* ke *daftar activity*.

#### B. Daftar Activity



Gambar 3. 23 Daftar Activity

Gambar 3.23 menunjukkan *daftar activity* dimana *user* melakukan registrasi apabila ingin membuat akun baru.

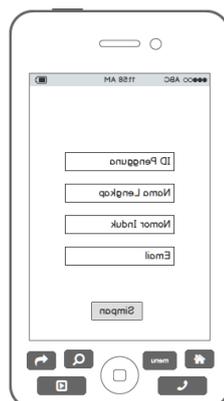
### C. Main Activity (Tab Profil)



**Gambar 3. 24** Main Activity (Tab Profil)

Gambar 3.24 menunjukkan *main activity* bagian *tab* profil. *Activity* ini menampilkan informasi akun milik *user*. Pada bagian kanan atas terdapat *menu bar*. Didalam *menu bar* tersebut terdapat 3 pilihan yaitu ubah profil, ubah kata sandi, dan keluar. Ubah profil memungkinkan *user* untuk mengakses ubah profil *activity*. Ubah kata sandi digunakan untuk mengalihkan *user* ke *activity* ubah kata sandi. Sedangkan keluar digunakan *user* untuk *logout*.

### D. Ubah Profil Activity



**Gambar 3. 25** Ubah Profil Activity

Gambar 3.25 menunjukkan bagian ubah profil *activity*. *User* dapat mengubah semua keterangan profilnya pada bagian ini. Agar bisa masuk ke

*activity* ini, *user* harus mengaksesnya melalui *menu bar* yang berada di bagian kanan atas pada *main activity* seperti yang ditunjukkan pada gambar 3.22

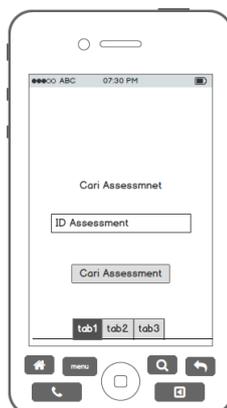
#### E. Ubah Kata Sandi *Activity*



**Gambar 3. 26** Ubah Kata Sandi *Activity*

Pada bagian *activity* ini *user* dapat mengubah kata sandi lama dengan kata sandi baru. Seperti yang ditunjukkan oleh gambar 3.26, *user* harus memasukkan kata sandi lama, lalu masukkan kata sandi baru dan lakukan konfirmasi terhadap kata sandi baru. Setelah itu dengan menekan tombol simpan kata sandi akan di validasi. Jika berhasil perubahan akan tersimpan.

#### F. *Main Activity (Tab Assessment)*

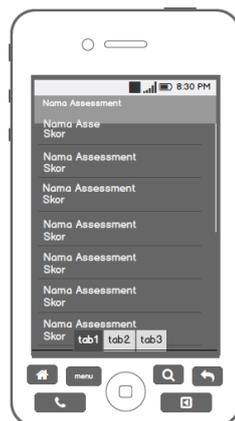


**Gambar 3. 27** *Main Activity (Tab Assessment)*

Gambar 3.27 menunjukkan *main activity* bagian *tab assessment*. Pada *activity* ini terdapat sebuah *field* yang digunakan *user* untuk mencari *assessment*

berdasarkan *ID assessment*. Terdapat tombol cari *assessment* yang digunakan untuk memulai identifikasi terhadap *ID* yang dimasukkan. Jika *assessment* tersedia maka *user* akan dialihkan ke bagian cek *enrollment activity*.

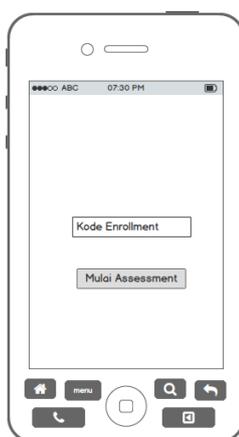
### G. Main Activity (Tab Nilai)



Gambar 3. 28 Main Activity (Tab Nilai)

*Activity* ini menampilkan daftar nilai dari setiap *assessment* yang sudah dikerjakan oleh *user*. Seperti yang ditunjukkan pada gambar 3.28 terdapat *list* yang berisi informasi nilai *assessment*. Setiap item berisi nama *assessment* dan skor.

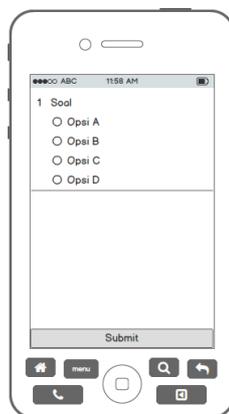
### H. Cek Enrollment Activity



Gambar 3. 29 Cek Enrollment Activity

Gambar 3.29 menunjukkan cek *enrollment activity*. *Activity* ini adalah bagian dimana *user* melakukan verifikasi ulang terhadap *assessment* yang akan dikerjakan. Dengan memasukkan kode *enrollment* yang diberikan oleh instruktur dan menekan tombol mulai *assessment*, *user* akan langsung dialihkan ke bagian kerjakan soal *activity*.

### I. Kerjakan Soal *Activity*



**Gambar 3. 30** Kerjakan Soal *Activity*

Gambar 3.30 menunjukkan *activity* kerjakan soal, *Activity* ini merupakan bagian dimana *user* mengerjakan *assessment* yang berhasil melalui proses cek *enrollment*. Terdapat tombol *submit* yang berfungsi untuk menyimpan hasil *assessment* apabila *user* telah selesai mengerjakan soal. Nilai dari *assessment* yang sudah dikerjakan dapat dilihat di *main activity* bagian *tab* nilai.