

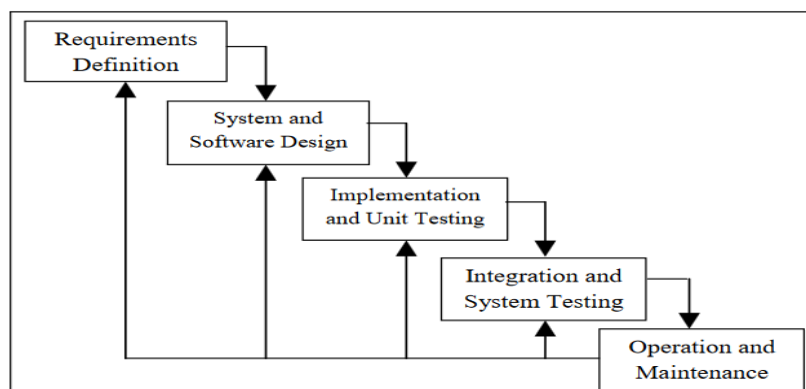
BAB III

METODOLOGI PENELITIAN

3.1 Metode Pengembangan Sistem

Proses pengembangan sistem pada penelitian ini menggunakan model SDLC (*Software Development Life Cycle*). SDLC merupakan sebuah siklus pengembangan perangkat lunak yang terdiri dari beberapa tahapan-tahapan penting dalam membangun perangkat lunak yang dilihat dari segi pengembangannya. Tahapan-tahapan tersebut diantaranya : perencanaan (*planning*), analisis (*analysis*), desain (*design*), implementasi (*implementation*), dan uji coba (*testing*). Selain untuk proses pembuatan, SDLC juga penting untuk proses *maintenance* (pemeliharaan) *software*.

Model SDLC yang digunakan dalam penelitian ini adalah model *Waterfall*. Model ini Pertama kali diperkenalkan oleh Winston Royce pada tahun 1970. Model ini merupakan model klasik yang sederhana dengan aliran sistem yang linear, yang artinya suatu tahapan harus selesai terlebih dahulu, sebelum memulai tahapan yang lain. Output dari setiap tahap menjadi input bagi tahap berikutnya. Dalam pembuatan sistem ini penulis menggunakan metode waterfall menurut referensi Sommerville. Akan tetapi, tahapan model *Waterfall* ini hanya sampai pada tahap *testing*, dikarenakan pada penelitian ini tidak sampai pada tahap *maintenance*. Untuk lebih jelasnya dapat dilihat pada gambar berikut :



Gambar 3. 1 Metode Waterfall

(Sommerville, 2011)

3.1.1 Requirement analysis and Definition

Sebelum melakukan analisis, penulis terlebih dahulu melakukan observasi terhadap siswa dan guru pada beberapa SMA di Sumatera dan Jawa yang nantinya sebagai *user*. Berdasarkan hasil observasi, peneliti menganalisis data dari permasalahan *user*. Analisis difokuskan pada kebutuhan fungsi *software* untuk memenuhi kendala yang dialami oleh *user*.

3.1.2 System and Software Design

Desain sistem merupakan tahap penyusunan proses, data, aliran proses, dan hubungan antar data yang paling optimal untuk menjalankan proses bisnis dan memelihara kebutuhan sesuai dengan analisis kebutuhan. Setelah peneliti mendapatkan dokumen hasil dari analisis, maka peneliti mengubah kebutuhan-kebutuhan fungsi *software* tersebut menjadi sebuah bentuk desain sistem dan aliran proses dari sistem yang akan dirancang.

3.1.3 Implementation and Unit Testing

Tahap ini merupakan tahap untuk mengubah desain yang telah dibuat menjadi sebuah sistem yang dapat berjalan sesuai dengan kebutuhan. Untuk dapat dimengerti oleh mesin, dalam hal ini adalah komputer, maka desain tersebut harus diubah menjadi bentuk yang dapat dimengerti oleh mesin, yaitu ke dalam bahasa pemrograman melalui proses *coding*. Bahasa pemrograman yang digunakan adalah bahasa pemrograman *C#*.

Pada tahap ini, peneliti membangun sebuah aplikasi berdasarkan desain sistem yang telah dibuat. Pengembangan aplikasi ini dilakukan dari awal hingga aplikasi siap dijalankan. Dari fungsi-fungsi yang dibutuhkan hingga tampilan untuk pengguna (*user*).

3.1.4 Integration and System Testing

Dalam tahap *testing* dilakukan pengujian *software* yang sudah dibuat. Semua fungsi-fungsi *software* harus diuji coba agar *software* bebas dari *error* atau kesalahan dan hasilnya harus benar-benar sesuai dengan kebutuhan yang sudah didefinisikan sebelumnya.

3.2 Alat dan Bahan

3.2.1 Alat

Alat yang digunakan dalam penelitian ini mencakup perangkat keras dan perangkat lunak.

- **Perangkat keras untuk pengembang**

Spesifikasi laptop yang digunakan untuk pengembang dapat dilihat pada tabel berikut:

Tabel 3. 1 Tabel perangkat keras pengembang

Perangkat Keras	Tipe
<i>Sistem Operasi</i>	Windows Embedded 8.1 Pro
<i>Processor</i>	Intel® Core™ i5-3317U Processor 1.70 GHz
<i>RAM</i>	4 GB
<i>Hard Disk</i>	500 GB

- **Perangkat lunak untuk pengembang**

Daftar perangkat lunak yang digunakan pengembang dapat dilihat pada tabel berikut :

Tabel 3. 2 Tabel perangkat lunak pengembang

No	Perangkat Lunak	Fungsi
1.	Microsoft Visio 2015	Digunakan untuk pembuatan perancangan sistem.
2.	SQL Server Management Studio 2014	Digunakan untuk pengolahan <i>database</i> yang akan digunakan dalam aplikasi presensi.
3.	IIS Server	Digunakan sebagai web server pada aplikasi.

3.2.2 Bahan

Bahan yang digunakan adalah hasil pengumpulan data berupa hasil wawancara dan data yang diperoleh melalui kuesioner berdasarkan penelitian sebelumnya yang masih memiliki keterkaitan dengan aplikasi yang dikembangkan.

3.3 Metode Pengumpulan Data

3.3.2 Wawancara

Dalam penelitian ini wawancara disajikan dalam bentuk tanya jawab secara langsung kepada narasumber, wawancara dilaksanakan kepada orang-orang yang pernah memiliki pengalaman dalam mengikuti persiapan ujian tertulis masuk perguruan tinggi, yang meliputi siswa kelas 12, mahasiswa baru di perguruan tinggi, dan guru SMA yang bersangkutan dengan mata pelajaran ujian masuk perguruan tinggi. Data dari hasil wawancara dapat dilihat di lampiran 1. Sehingga data yang diperoleh dapat membantu dalam pembuatan fitur reporting pada sistem asesmen dan pemetaan hasil asesmen.

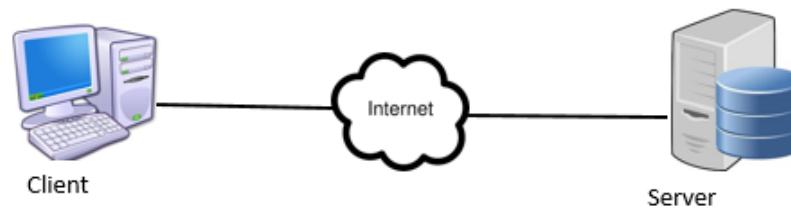
3.3.2 Kuesioner

Dalam penelitian ini kuesioner disajikan dalam bentuk pertanyaan berupa pilihan ganda, jawaban singkat dan skala linier sehingga mempermudah responden untuk mengisi kuesioner tersebut sesuai keadaan yang dialami. Kuesioner disebar kepada orang-orang yang pernah memiliki pengalaman dalam mengikuti persiapan ujian tertulis masuk perguruan tinggi, yang meliputi siswa kelas 12, mahasiswa baru di perguruan tinggi, dan guru SMA melalui sebuah komunitas *online*. Data dari hasil kuesioner dapat dilihat di lampiran. Sehingga data yang diperoleh dapat membantu dalam perancangan fitur mappin pada Sistem Asesmen dan Pemetaan Hasil Asesmen.

3.4 Analisis Penelitian

3.4.1 Arsitektur

Aplikasi ini akan digunakan pada setiap sekolah dengan web server dan basis data yang berbeda untuk setiap sekolah, sehingga peneliti mendapatkan sebuah analisis rancangan arsitektur web secara umum seperti pada gambar berikut :



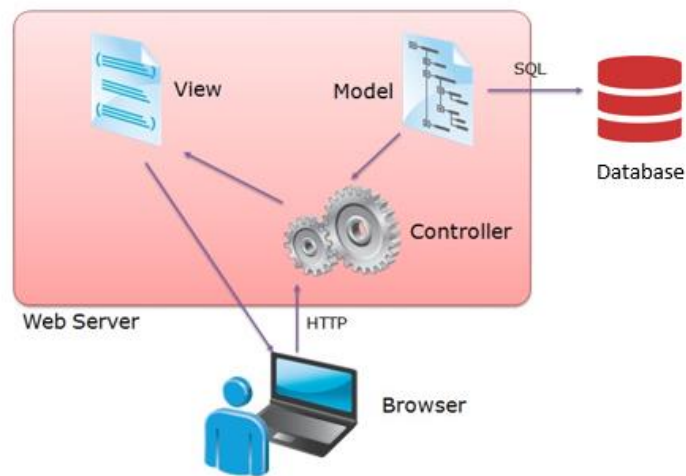
Gambar 3. 2 Arsitektur Aplikasi

Database yang digunakan pada database server adalah Microsoft SQL Server 2014, kemudian akan terhubung melalui web server. User mengakses web dengan menggunakan komputer client, saat user mengakses web, sistem akan memuat antarmuka web dan melakukan pengambilan data dari database server. Melalui antarmuka web, user dapat menambahkan data dan disimpan kedalam database.

Dalam pembuatan aplikasi ini, pola yang akan digunakan adalah MVC. Pola MVC memisahkan sebuah aplikasi, khususnya aplikasi website menjadi tiga bagian, yaitu:

- a. Model : Model berperan untuk mengelola data, menyimpan dan mengambil *entity* yang digunakan aplikasi dari sebuah database.
- b. View : View memiliki fungsi untuk menampilkan data yang sebelumnya diolah oleh model. Berhubungan kepada interface akhir aplikasi yang akan dilihat oleh user.
- c. Controller : Controller berguna sebagai penghubung antara model dan view. Controller menerima permintaan dari *client*, lalu memanggil model untuk melakukan operasi, kemudian mengirimkan data pada view. Akhirnya view akan menampilkan data tersebut dalam bentuk sebuah halaman html.

Arsitektur proses kontrol dapat dilihat pada **Gambar 3.3** :



Gambar 3.3 Arsitektur Proses Kontrol Model MVC

3.4.2 Analisis Pengguna

Website pada fitur reporting yang dibuat hanya oleh guru dan siswa, yang nantinya berperan sebagai user web e-learning. User guru memiliki hak akses melihat data nilai menurut yang dia buat, seperti :

1. Data nilai rata-rata pertahun.
2. Data nilai rata-rata perkelas.
3. Data nilai rata-rata perujian.
4. Data nilai persiswa.
5. *Review* ujian siswa.
6. Print review ujian siswa.

Untuk user siswa memiliki hak akses melihat data nilai menurut yang dia kerjakan, seperti :

1. Data nilai rata-rata permatapelajaran.
2. Data nilai rata-rata perujian.
3. Semua data nilai ujian yang dia kerjakan.
4. *Review* ujian.
5. Print review ujian.

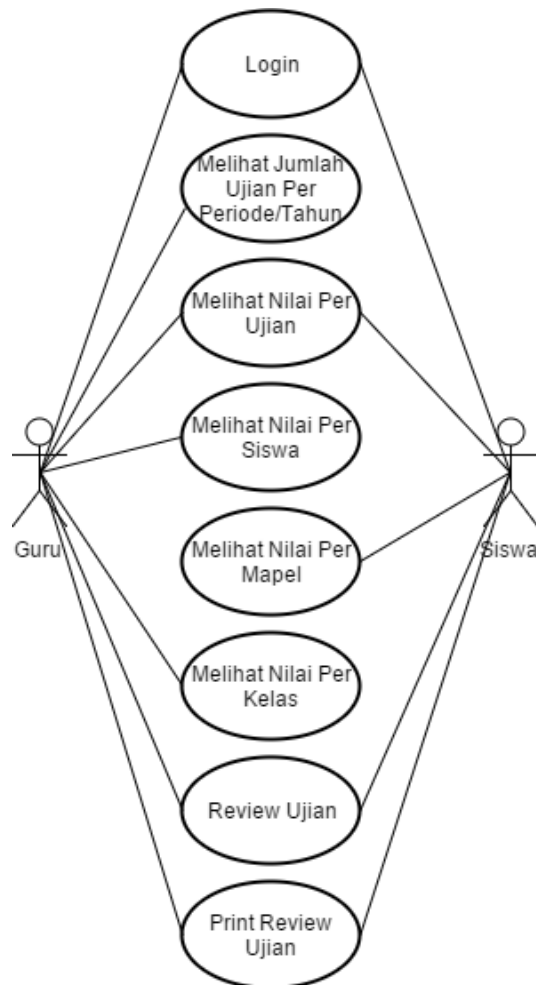
3.5 Rancangan Sistem

3.5.1 Rancangan Sistem

1. Use Case Diagram

Use case diagram untuk fitur reporting guru pada aplikasi, dapat dilihat pada gambar berikut :

Berikut penjelasan **Gambar 3.4** :



Gambar 3.4 Use Case Diagram Guru & Siswa

Tugas guru secara umum adalah pemantauan data yang digunakan dalam aplikasi, diantaranya :

1. Login : Guru harus login terlebih dahulu sebelum masuk kedalam halaman *reporting* nilai.
2. Lihat Data Jumlah Ujian Pertahun : Guru dapat melihat data pertahunnya atau per periode, yang jika melakukan filer chart tahun

akan berfungsi memantau perkembangan hasil ujian yang dibuat pertahun.

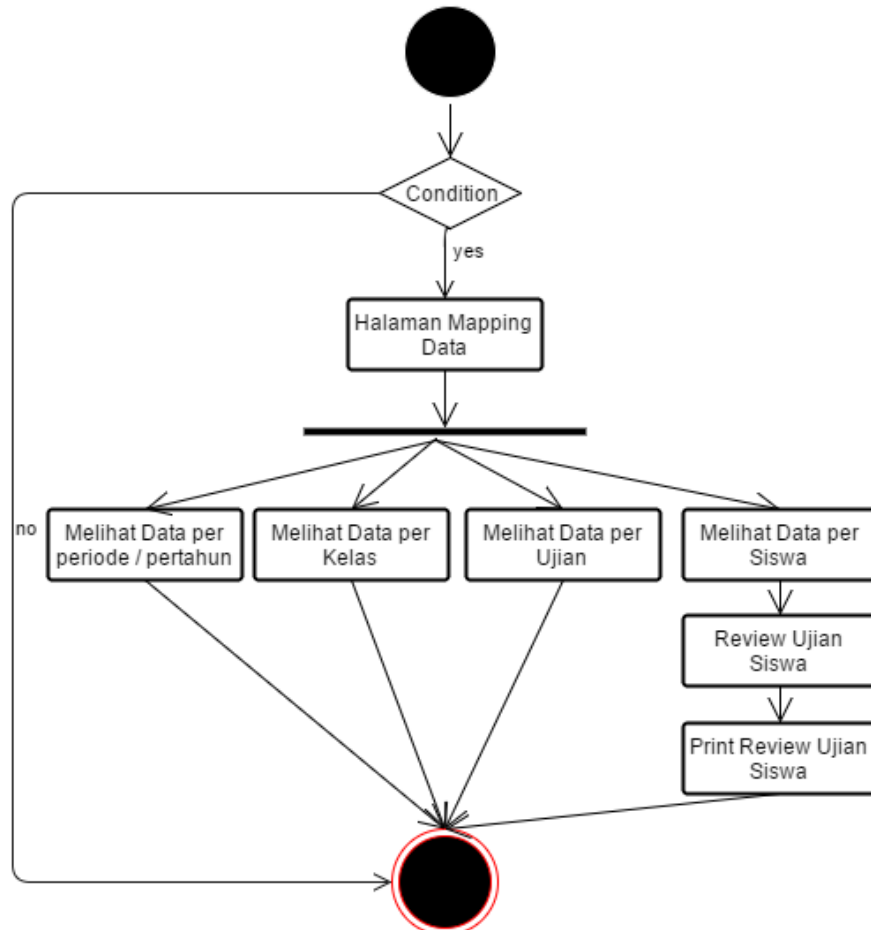
3. Lihat Data Nilai Perujian : Guru dapat melihat data perujian, yang jika melakukan filer chart ujian akan berfungsi memantau perkembangan hasil ujian yang dibuat setiap ujian.
4. Lihat Data Nilai Persiswa : Guru dapat melihat data persiswa, yang jika melakukan filer chart siswa akan berfungsi memantau perkembangan hasil ujian siswa.
5. Lihat Data Nilai Perkelas : Guru dapat melihat data perkelas, yang jika melakukan filer chart ujian akan berfungsi memantau perkembangan hasil ujian yang dibuat setiap kelas.
6. *Review Ujan*: Guru dapat melihat ujian yang sudah dikerjakan oleh siswa.
7. *Print Review Ujian*: Guru dapat mencetak data soal serta jawaban ujian yang siswa kerjakan.

Tugas siswa secara umum adalah pemantauan data yang digunakan dalam aplikasi, diantaranya :

1. *Login* : Siswa harus login terlebih dahulu sebelum masuk kedalam halaman *reporting* nilai.
2. Lihat Data Jumlah Ujian Permatapelajaran : Siswa dapat melihat data permatapelajaran, yang jika melakukan filer chart mapel akan berfungsi memantau perkembangan hasil ujian menurut matapelajaran.
3. Lihat Data Nilai Perujian : Siswa dapat melihat data perujian, yang jika melakukan filer chart ujian akan berfungsi memantau perkembangan hasil ujian yang dibuat setiap ujian.
4. *Review Ujan*: Siswa dapat melihat ujian yang sudah dikerjakan oleh siswa.
5. *Print Review Ujian*: Siswa dapat mencetak data soal serta jawaban ujian yang dikerjakan.

2. Activity Diagram

Berdasarkan use case yang telah dibuat, maka dapat diperoleh activity diagram berdasarkan aktor yang terlibat pada use case diagram, yang dapat dilihat pada gambar berikut :

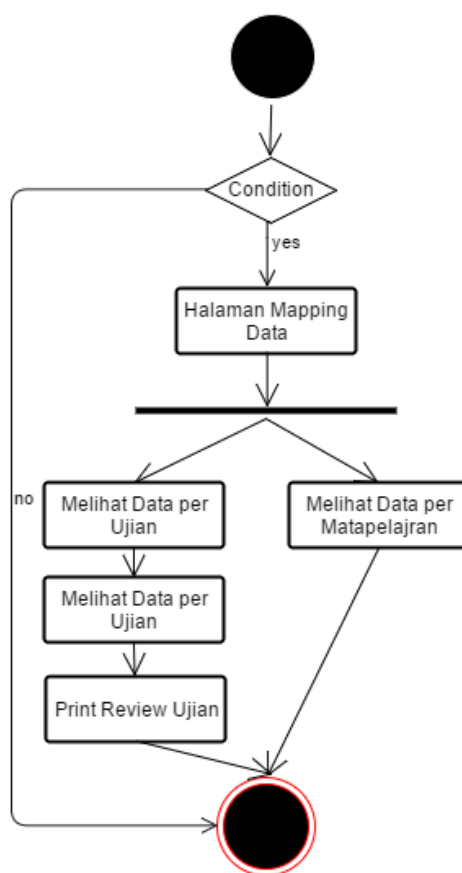


Gambar 3.5 Activity Diagram Guru

Berikut penjelasan terhadap langkah-langkah yang terdapat pada **Gambar 3.5** :

- a. Terdapat beberapa proses yang harus dilewati ketika guru menggunakan aplikasi. Sebelum masuk ke halaman reporting, user sudah login dahulu sebagai guru.
- b. Apabila user adalah guru, maka akan ditampilkan halaman reporting akan tetapi apabila gagal, berarti user login dari awal sebagai siswa dan data salah.

- c. Pada halaman reporting, terdapat beberapa chart dan satu buah tabel yang berfungsi menampilkan data nilai menurut tahun, kelas, ujian, dan siswa.
- d. Apabila user melihat *review* hasil ujian dari siswa, maka user tinggal masuk halaman *review* pada *link review* digrid nilai.
- e. Apabila user melihat *print* hasil ujian dari siswa, maka user tinggal *click button print*.



Gambar 3. 6 Activity Diagram Guru

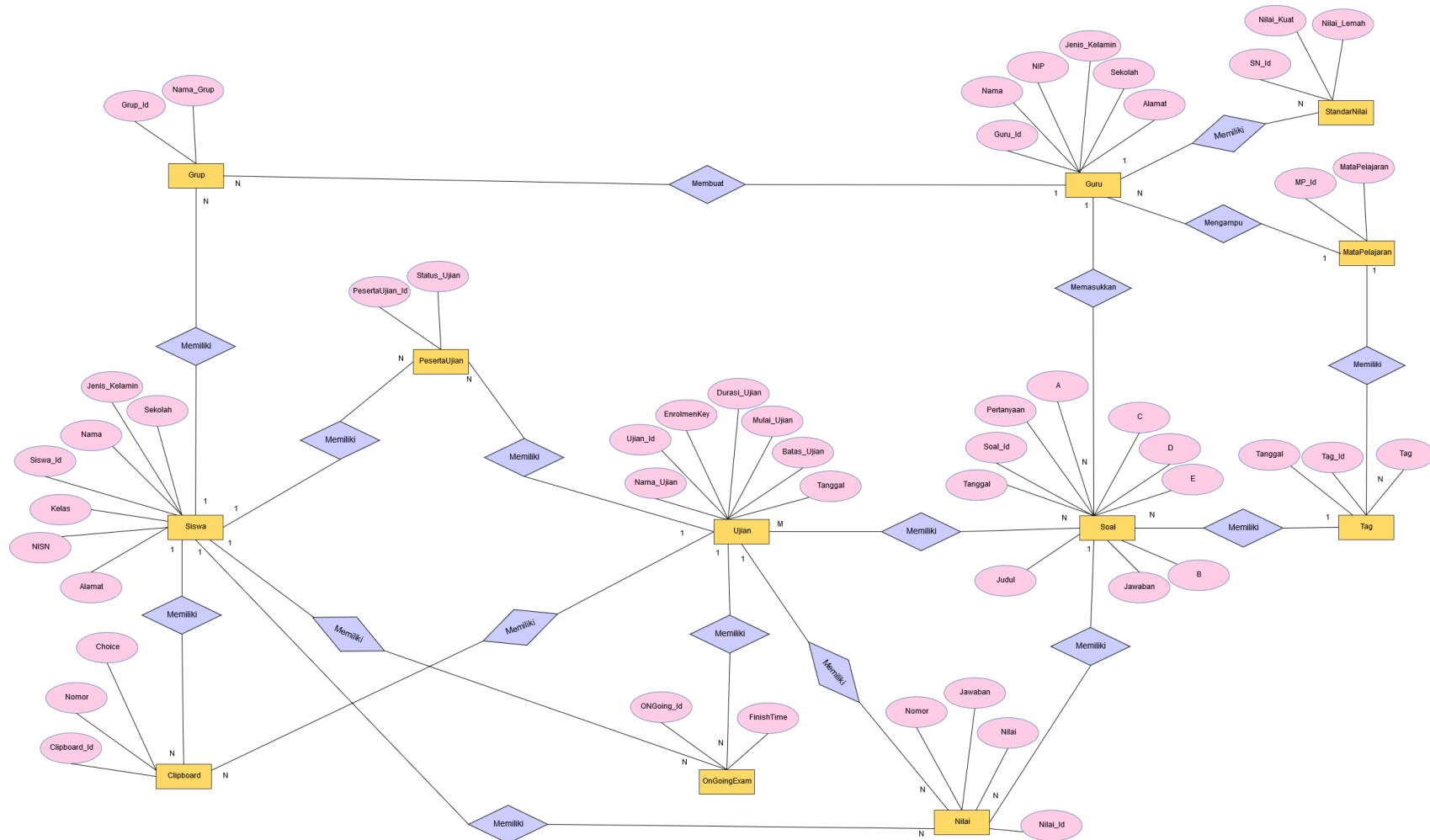
Berikut penjelasan terhadap langkah-langkah yang terdapat pada **Gambar 3.6** :

- a. Terdapat beberapa proses yang harus dilewati ketika siswa menggunakan aplikasi. Sebelum masuk ke halaman reporting, user sudah login dahulu sebagai siswa.

- b. Apabila user adalah siswa, maka akan ditampilkan halaman reporting akan tetapi apabila gagal, berarti user login dari awal sebagai guru dan data salah.
- c. Pada halaman reporting, terdapat beberapa chart dan satu buah tabel yang berfungsi menampilkan data nilai menurut matapelajaran, ujian, dan semua data nilai milik user.
- d. Apabila user melihat review hasil ujian, maka user tinggal masuk halaman review pada *link* review digrid nilai.
- e. Apabila user melihat *print* hasil ujian, maka user tinggal *click button print*.

3. Entity Relationship Diagram (ERD)

Untuk mengetahui hubungan antar obyek (Entity) dalam basis data yang digunakan, dapat dilihat dari ERD pada **Gambar 3.7**



Gambar 3.7 Entity Relationship Diagram

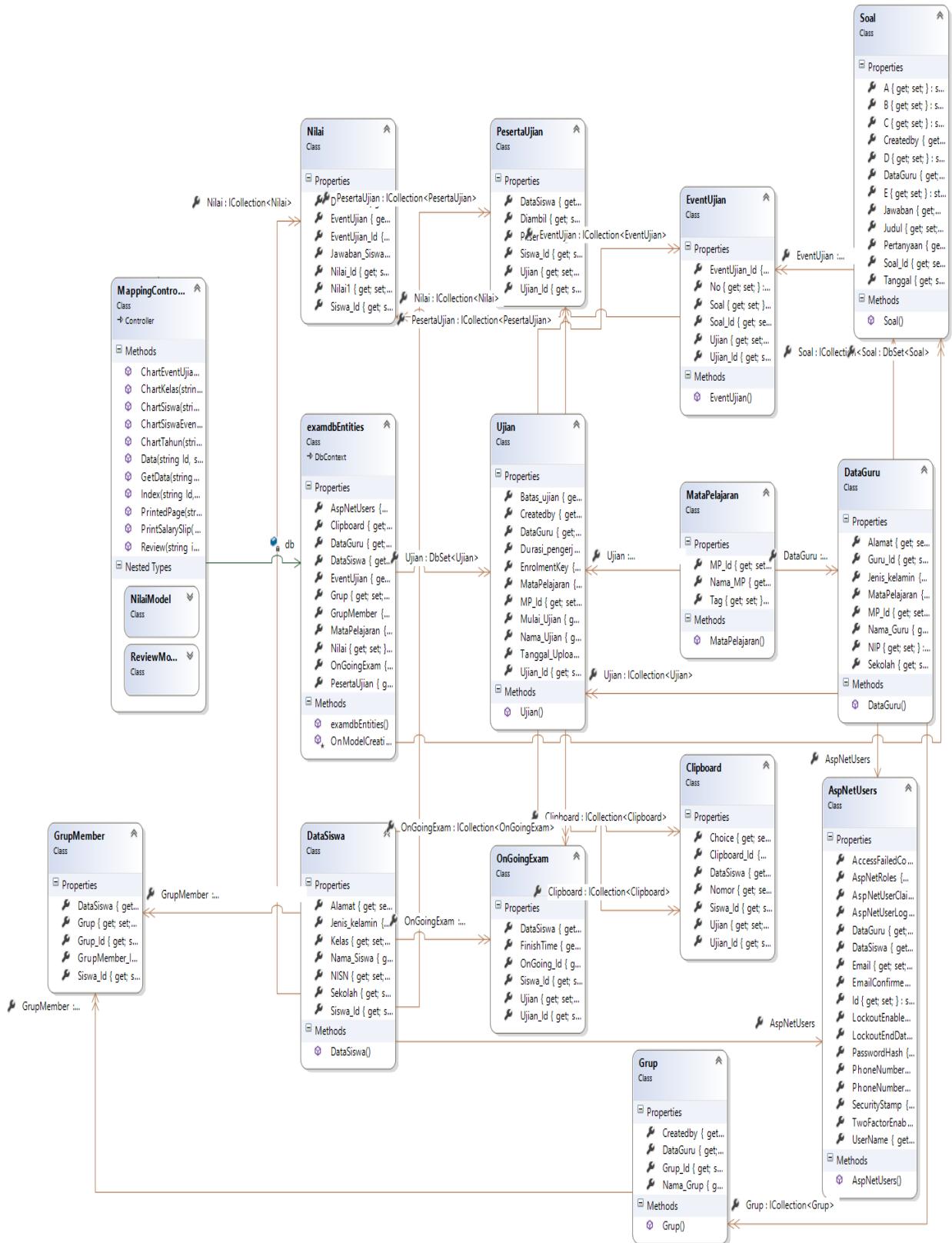
Penjelasan ERD pada **Gambar 3.7** adalah sebagai berikut :

- a. Entity Guru mempunyai relasi many to 1 terhadap entity MataPelajaran.
- b. Entity Guru mempunyai relasi 1 to many terhadap entity StandarNilai.
- c. Entity Guru mempunyai relasi 1 to many terhadap entity Soal.
- d. Entity Ujian mempunyai relasi many to many dengan entity Soal.
- e. Entity Soal mempunyai relasi many to 1 dengan entity Tag.
- f. Entity Tag mempunyai relasi many to 1 terhadap entity MataPelajaran.
- g. Entity Ujian mempunyai relasi many to 1 dengan entity Guru.
- h. Entity Grup mempunyai relasi many to 1 dengan entity Guru.
- i. Clipboard relasi many to 1 dengan entity Ujian.
- j. Entity Clipboard relasi many to 1 dengan entity Siswa.
- k. Entity OnGoingExam mempunyai relasi many to 1 dengan entity Siswa
- l. Entity OnGoingExam mempunyai relasi many to 1 dengan entity Ujian.
- m. Entity PesertaUjian mempunyai relasi many to 1 dengan entity Siswa.
- n. Entity PesertaUjian mempunyai relasi many to 1 dengan entity Ujian.
- o. Entity Siswa mempunyai relasi 1 to many dengan entity Grup.
- p. Entity Nilai mempunyai relasi many to 1 dengan entity Siswa.

- q. Entity Nilai mempunyai relasi many to 1 dengan entity Ujian.
- r. Nilai mempunyai relasi many to 1 dengan entity Soal.

4. *Class Diagram*

Rancangan *Class diagram* yang akan digunakan dalam proses pembuatan aplikasi dapat dilihat pada gambar berikut :



Gambar 3.8 Class Diagram untuk Reporting

Berikut penjelasan **Gambar 3.8**:

1. Proses pembuatan aplikasi akan menggunakan *Class Asp.net MVC* untuk pengaturan user.
2. *Class Data Guru* berfungsi untuk menampung daftar identitas guru yang digunakan dalam aplikasi.
3. *Class Data Siswa* berfungsi untuk menampung daftar identitas siswa yang digunakan dalam aplikasi.
4. *Class Group* berfungsi untuk menyimpan daftar kelas,
5. *Class Member Group* berfungsi untuk menyimpan kelas yang diikuti oleh siswa.
6. *Class Asp Net User* berfungsi untuk menyimpan data login user.
7. *Class Mata Pelajaran* berfungsi untuk menampung daftar mata pelajaran yang digunakan dalam aplikasi.
8. *Class Clipboard* berfungsi untuk menampung daftar guru yang telah membuat ujian.
9. *Class Soal* berfungsi untuk menampung daftar soal yang digunakan dalam aplikasi.
10. *Class Ujian* berfungsi untuk menampung daftar ujian yang dilakukan dalam aplikasi.
11. *Class Event Ujian* berfungsi untuk menampung data daftar soal yang digunakan pada ujian yang sudah dibuat.
12. *Class Peserta Ujian* berfungsi untuk menampung data daftar siswa yang sudah mengikuti ujian.
13. *Class Nilai* berfungsi untuk menyimpan data nilai ujian persoa atau perevent ujian, yang coloum nilainya tersimpan dalam angka 1 jika benar dan 0 salah.
14. *Class Reporting Controller* memiliki *Nilai Model* didalamnya sebagai class bantuan untuk menyimpan hasil pengolahan data *Class Nilai*.
15. *Class examdb* berfungsi sebagai media koneksi database yang digunakan oleh controller.

3.5.2 Rancangan Antarmuka

Interface atau antarmuka merupakan tampilan dari aplikasi yang menjadi sarana komunikasi antara pengguna dengan sistem, serta memberikan kemudahan kepada pengguna dalam melakukan aktivitas didalam penggunaan *web*.

1. Rancangan Halaman Reporting Guru

Rancangan antarmuka ini merupakan halaman menampilkan data reporting yang user-nya adalah guru, data yang di tampilkan berupa data chart dan grid yang berisi rata-rata nilai, antara lain:

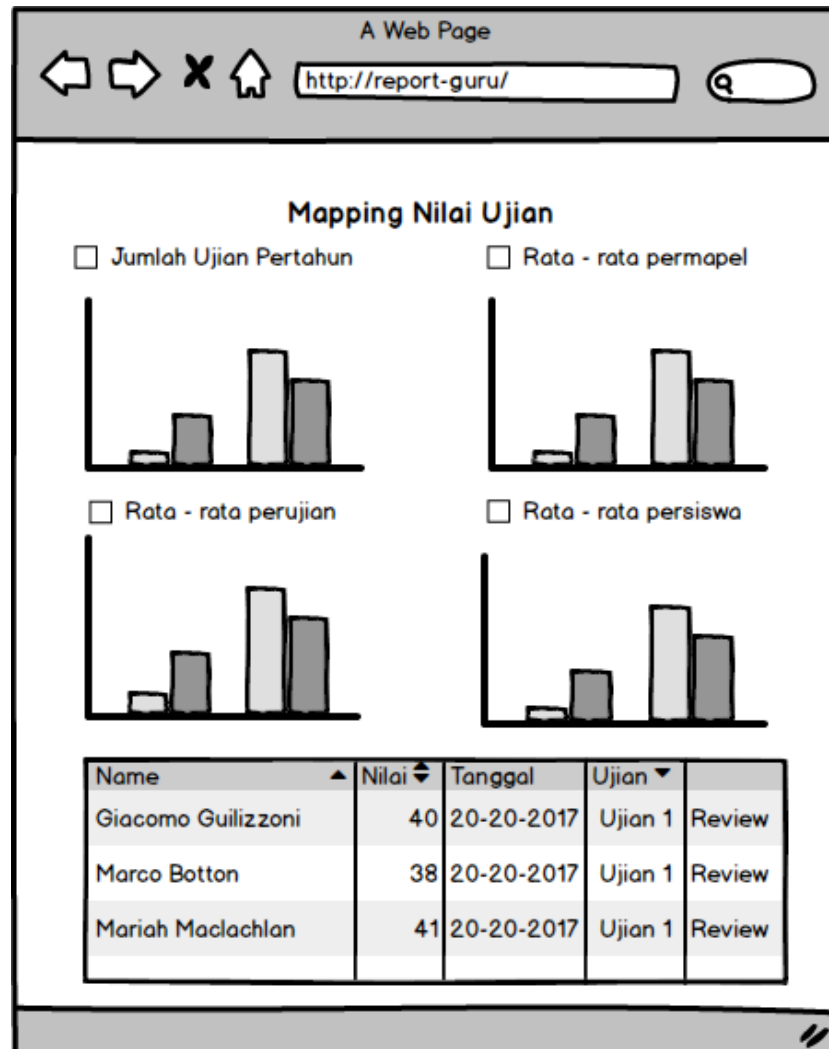
- a. Chart Bar menampilkan data jumlah ujian yang diikuti pertahun.
- b. Chart Bar menampilkan data rata-rata nilai kelas, ujian dan siswa.
- c. Grid menampilkan data siswa lebih detail.
- d. Review menampilkan hasil ujian yang sudah dikerjakan.

Halaman reporting guru juga dilengkapi dengan fitur *filtering* pada chart, yang memudahkan guru dalam mencari data. Dimana filter tersebut terdapat :

- a. Click data pada chart bar yang berisi data jumlah ujian yang diikuti pertahun akan mengfilter data rata-rata ujian pada chart bar, data siswa pada chart bar dan data siswa pada grid menurut tahun yang dipilih.
- b. Click data pada chart bar yang berisi data rata-rata nilai perujian akan mengfilter data rata-rata pertahun pada chart bar, data siswa pada chart bar dan data siswa pada grid menurut ujian yang dipilih.
- c. Click data pada chart bar yang berisi data siswa akan mengfilter data rata-rata pertahun pada chart bar, data rata-rata ujian pada chart bar, dan data siswa pada grid menurut siswa yang dipilih.

- d. Click *link review* pada grid bar akan menuju ke halaman *review* ujian yang filter menurut ujian id dan siswa id.

Gambaran rancangan antarmuka halaman ini dapat dilihat pada **Gambar 3.9**



Gambar 3. 9 Rancangan antarmuka halaman Reporting Data Guru

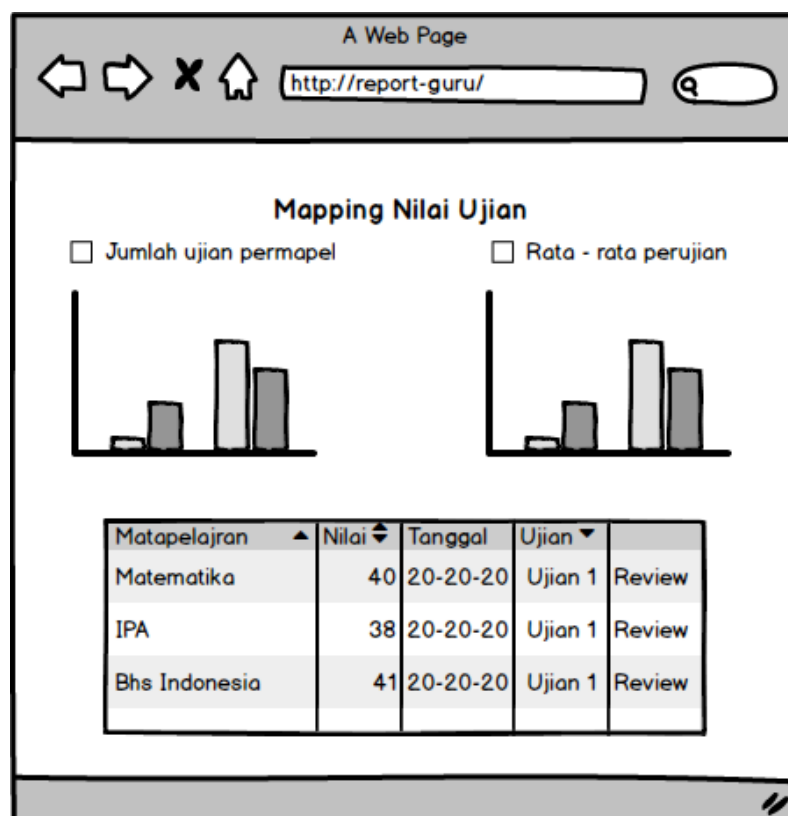
Rancangan antarmuka ini merupakan halaman menampilkan data reporting yang usernya adalah siswa, data yang di tampilkan berupa data chart dan grid yang berisi rata-rata nilai, antara lain:

- Chart Bar menampilkan data jumlah ujian peramatapelajaran.
- Chart Bar menampilkan data siswa.
- Grid menampilkan data siswa lebih detail.
- Review ujian.

Halaman reporting guru juga dilengkapi dengan fitur *filtering* pada chart, yang memudahkan guru dalam mencari data. Dimana filter tersebut terdapat :

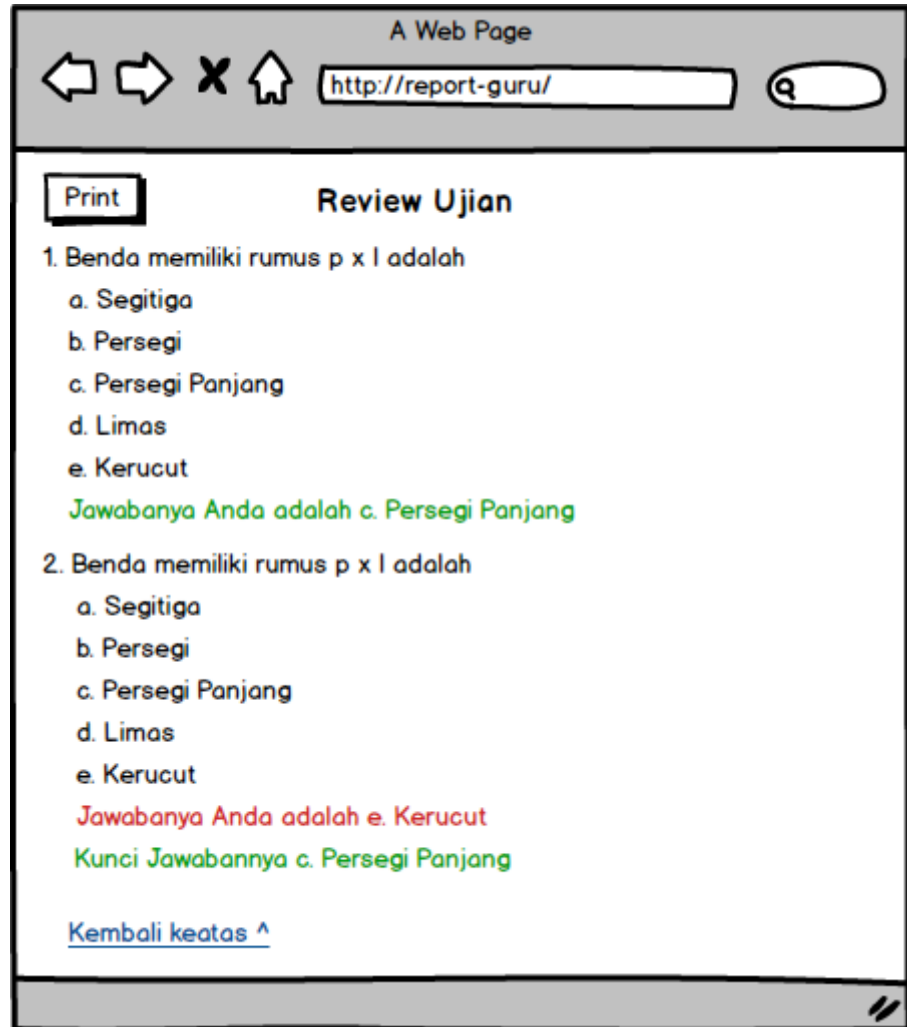
- Click data pada chart bar yang berisi data jumlah ujian permatapelajaran akan mengfilter data rata-rata ujian pada chart bar, dan data siswa pada grid menurut matapelajaran yang dipilih.
- Click data pada chart bar yang berisi data rata-rata nilai perujian akan mengfilter data rata-rata permatapelajaran pada chart bar, dan data siswa pada grid menurut ujian yang dipilih.
- Click *link review* pada grid bar akan menuju ke halaman *review* ujian yang filter menurut ujian id dan siswa id.

Gambaran rancangan antarmuka halaman ini dapat dilihat pada **Gambar 3.10**



Gambar 3. 10 Rancangan antarmuka halaman Reporting Data Siswa.

Halaman review ujian menampilkan data ujian yang sudah dia kerjakan beserta jawaban yang dia isi dan *button click print* untuk mencetak data. Data ditampilkan menurut filter ujian id dan siswa id dari *link* yang diclick pada grid. Gambaran rancangan antarmuka halaman ini dapat dilihat pada **Gambar 3.11**



Gambar 3. 11 Rancangan antarmuka halaman Review