

2. BAB III ANALISIS DAN PERANCANGAN

1. Requirements Phase

1. Identifikasi Kebutuhan Pengguna untuk *Web Services*

Tahapan ini digunakan untuk mengidentifikasi kebutuhan pengguna dan fitur – fitur yang dibutuhkan oleh pengguna. Berikut langkah – langkah dalam mengidentifikasi kebutuhan pengguna:

Wawancara

Wawancara dilakukan di ruangan BPM yang terletak di Rektorat UII Lantai 3. Wawancara dilakukan dengan bersama Bu Kariyam selaku Ketua BPM UII. Wawancara pertama dilaksanakan pada tanggal 9 April 2019 dengan tujuan untuk mendapatkan gambaran proses audit di UII. Wawancara kedua dilakukan pada tanggal 24 April 2019 dengan tujuan untuk mengkaji berkas yang berada di BPM UII. Wawancara ketiga dilakukan pada tanggal 15 Mei 2019 dengan tujuan untuk menganalisis kebutuhan data yang sudah ada di Badan Sistem Informasi UII.

Studi Literatur

Pada tahapan ini dilakukan mencari berbagai referensi yang akan digunakan untuk membangun *web services*. Sumber referensi terdapat dari berbagai sumber seperti artikel, jurnal, buku dan sumber yang lainnya. Referensi yang digunakan untuk membangun *web services* BPM UII yaitu referensi yang berkaitan dengan sistem penjaminan mutu seperti pengertian sistem penjaminan mutu, cara kerja sistem penjaminan mutu pada umumnya, dan permasalahan yang sering terjadi di sistem penjaminan mutu.

Kajian Dokumen

Pada tahapan ini dilakukan analisis berkas – berkas yang berada di BPM UII. Berkas – berkas tersebut di analisis dengan tujuan untuk mengetahui data – data yang digunakan dalam proses audit di UII.

2. Identifikasi kebutuhan *Web Services*

Pada tahapan ini dilakukan dengan mengumpulkan kebutuhan fungsional *web services*, dan kebutuhan nonfungsional *web services*.

Kebutuhan Fungsional

Kebutuhan fungsional akan menentukan kebutuhan terhadap layanan yang disediakan oleh sistem.

1. Pemberkasan AMI

1. Pengguna (Auditee) dapat memasukkan data-data pemberkasan AMI sesuai dengan lingkup akses pekerjaan dan batas waktu pemberkasan AMI.
2. Pengguna (Auditee) dapat menghapus data pemberkasan AMI sesuai lingkup akses pekerjaannya.
3. Pengguna (Auditee) dapat memperbarui data pemberkasan AMI sesuai lingkup akses pekerjaan dan batas waktu pemberkasan AMI.
4. Pengguna (Auditee) dapat melihat data pemberkasan AMI yang telah dimasukkan ke dalam sistem.
5. Pengguna (Auditee) dapat mengetahui capaian setiap butir dari standar.
6. Pengguna (Auditee) dapat mengetahui nilai akhir setiap butir dari standar.
7. Pengguna (Auditee) dapat mengetahui target dari setiap butir dari standar.
8. Pengguna (Auditee) dapat melakukan validasi terhadap data pemberkasan AMI yang telah dimasukkan sebelumnya.

1. Audit Pemberkasan AMI

1. Pengguna (Auditor) dapat melihat data pemberkasan AMI yang telah dimasukkan oleh Auditee dengan sesuai lingkup aksesnya.
2. Pengguna (Auditor) dapat memberikan komentar terhadap kesalahan data yang telah dimasukkan oleh Auditee.
3. Pengguna (Auditor) dapat memberikan laporan temuan AMI jika nilai target pemberkasan AMI tidak memenuhi.
4. Pengguna (Auditor) dapat melakukan aksi selesai jika pemberkasan AMI telah sesuai.

5. Manajemen SIM BPM UII

1. Pengguna (Admin) dapat melihat seluruh data pemberkasan AMI
2. Pengguna (Admin) dapat memanajemen *user* yang ada di sistem BPM UII
3. Pengguna (Admin) dapat mengatur lingkup akses setiap *user* di sistem BPM UII

Kebutuhan Nonfungsional

Kebutuhan nonfungsional akan menentukan kebutuhan properti yang dibutuhkan dalam membangun *web services* BPM UII.

2. Operasional

1. Visual Studio Code (Text Editor)

Integrated Development Environment adalah sebuah software yang dapat membantu untuk membangun sebuah perangkat lunak yang memiliki beberapa fasilitas.

2. Postman

Postman merupakan aplikasi yang digunakan untuk menguji *web service* yang telah dibangun.

3. JMeter

JMeter merupakan aplikasi yang digunakan untuk menguji performa *web services* yang telah dibangun.

3. Keamanan

1. *Web service* yang akan dibangun dilengkapi otorisasi terhadap *endpoint* yang diakses.
2. *Web service* yang akan dibangun memberikan *response* data yang sesuai dengan lingkup akses pengguna.

3. Manajemen kebutuhan *Web Services*

No	Modul	Peran	Deskripsi
1	Login	Pengguna	Sebagai autentikasi terhadap pengguna yang ingin memasuki sistem. Ketika autentikasi berhasil maka akan mendapatkan token yang digunakan <i>endpoint</i> lain sebagai otorisasi
2	Pemilihan peran pengguna	Auditee, auditor, dan admin	Sebagai pemilihan peran pengguna, karena pengguna bisa mendapatkan lebih dari satu peran
3	Sasaran Mutu	Auditee, auditor dan admin	Sebagai standar/target yang akan dicapai Universitas, Fakultas, Jurusan, Program Studi, ataupun unit pendukung lain di lingkungan universitas

No	Modul	Peran	Deskripsi
4	Bukti Capaian	Auditee, auditor dan admin	Sebagai pemberkasan data bukti capaian AMI
5	Rekomendasi tindak lanjut	Auditee, auditor dan admin	Sebagai pemberkasan data rekomendasi tindak lanjut
6	Keranjang validasi	Auditee	Sebagai keranjang data validasi yang digunakan ketika data telah terisi maka akan masuk kedalam keranjang validasi
7	Laporan temuan AMI	Auditor	Sebagai pembuatan laporan temuan AMI, ketika auditor mendapatkan temuan AMI
8	Grafik	Auditee, auditor dan admin	Untuk menampilkan semua data sasaran mutu, yang nantinya oleh aplikasi klien akan ditampilkan dalam bentuk grafik
9	Otorisasi	Auditee, auditor dan admin	Proses pengecekan izin kepada pengguna untuk mengakses <i>web services</i>
11	<i>Cut Off</i>	Auditee, Auditor	Untuk melakukan pembatasan waktu proses audit

4. Mempersiapkan pengujian Sistem

Pada tahapan ini merupakan persiapan untuk pengujian sistem yang akan dibangun. Tujuan dilakukan pengujian agar memenuhi kebutuhan dari BPM UII dan kualitas peformas dari *web service*. Pada tahapan ini akan dilakukan 2 pengujian pada *web service* yang telah dibangun yaitu *endpoint testing* dan *endpoint performance testing*.

4. Analysis Phase

1. Pemilihan teknologi yang akan digunakan untuk *Web Services*

Pada tahapan ini dilakukan pemilihan teknologi untuk sistem *web service* yang akan dibangun. Teknologi yang akan digunakan yaitu *framework* Lumen. Lumen merupakan *microframework* dengan menggunakan bahasa pemrograman PHP. Lumen merupakan pecahan dari *framework* laravel, yang berfokus pada pembuatan *REST API*. Lumen menggunakan

konsep MVC (Model View Controller). MVC merupakan metode untuk membangun aplikasi dengan memisahkan data dengan tampilan dan prosesnya.

2. **Arsitektur Web Services**

Arsitektur *web services* yang dibangun menggunakan arsitektur komunikasi data *RESTful Web Services* dan menerapkan *layered architecture*. Pada Gambar 2.1 merupakan arsitektur dari BPM UII dibagi menjadi 3 bagian yaitu *presentation layer*, *business and persistence layer*, dan *database layer*. Berikut merupakan penjelasan dari masing – masing bagian *layered architecture*.

Presentation Layer

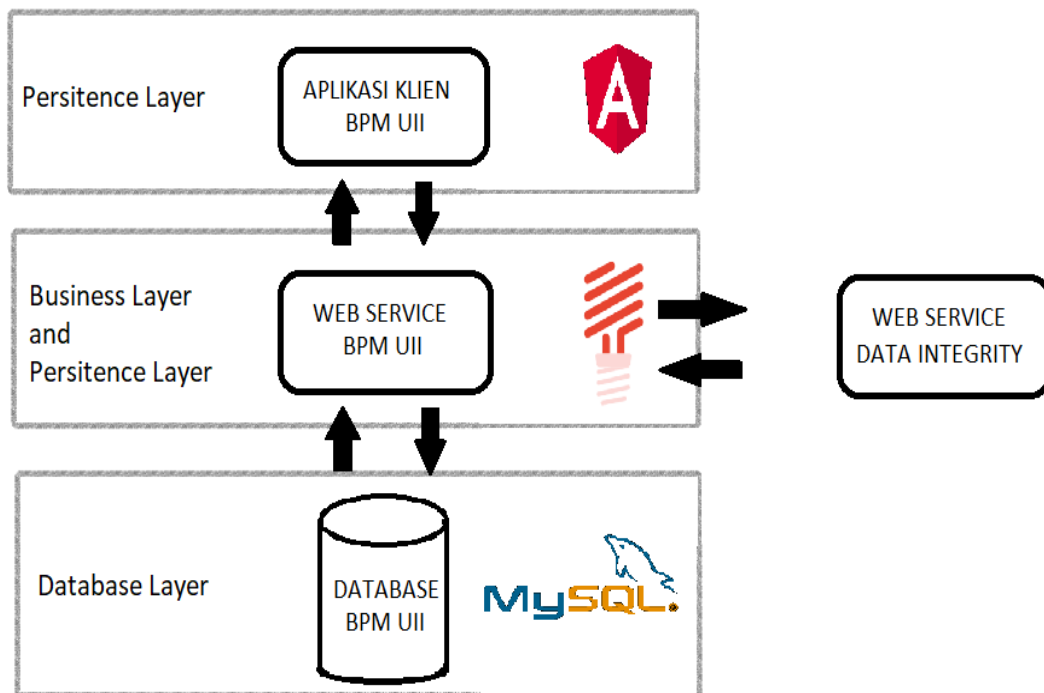
Layer ini merupakan *layer* yang bertugas sebagai penghubung antara pengguna dengan *business layer*. *Presentation layer* akan menerima masukan dari pengguna dan mengirimkan data tersebut ke *business layer*. *Business layer* akan memberikan *response* informasi dan status kode HTTP yang akan diterima oleh *presentation layer*. Pada penelitian ini *presentation layer* diisi oleh aplikasi klien BPM UII dengan menggunakan *framework Angular*.

Business dan Persistence Layer

Business layer merupakan *layer* yang mengolah data yang diterima sesuai fungsionalitas yang diminta dari *presentation layer* dan memberikan *response* informasi kepada *presentation layer*. *Persistence layer* yaitu *layer* yang digunakan untuk mengatur pemodelan data dan menghubungkannya ke *database layer*. *Persistence layer* akan dipanggil oleh *business layer*. Pada penelitian ini *business layer* juga akan mengirimkan permintaan data kepada *web service data integrity* sesuai fungsionalitas yang diminta oleh *persistence layer*.

Database Layer

Layer ini berfungsi sebagai penyimpanan dan mengirim data yang diminta sesuai dengan permintaan dari *business layer*. Pada penelitian ini menggunakan *database management system MySQL*.



Gambar 2.1 Arsitektur BPM UI

5. Design Phase

1. Mendefinisikan antarmuka *Web Services*

Perancangan struktur konten digunakan untuk menentukan struktur dari *response* dari *endpoint* yang dikirimkan oleh aplikasi server ke aplikasi klien. *Web service* BPM UI menggunakan format penulisan data JSON. Struktur *response web service* BPM UI memiliki dua jenis. Jenis *response web service* yang pertama yaitu *response* dari *method* GET yang mengembalikan koleksi datadan status *method* HTTP atau jika gagal maka akan mengembalikan pesan gagal yang dapat dilihat pada Gambar 2.2 dan Gambar 2.3 . Jenis *response web service* yang kedua yaitu *response* dari *method* PUT, POST, dan DELETE yang mengembalikan pesan berhasil atau gagal yang dapat dilihat pada Gambar 2.4 dan Gambar 2.5.

```
{
  "success": true,
  "result": {
    "tanggal_audit": "0000-00-00",
    "periodes_id": 1,
    "organisis_id": 90,
    "uuid": "17b86d97-0d5f-11ea-b95a-0cc47afc359e",
    "created_at": "2019-11-22 19:34:25",
    "updated_at": "2019-11-22 19:34:25",
    "uraian_temuan": []
  }
}
```

```

}
}

```

Gambar 2.2 Contoh stuktur konten *response* berhasil dari endpoint method GET

```

{
  "success": false,
  "error": "Data Not Found"
}

```

Gambar 2.3 Contoh stuktur konten *response* gagal dari endpoint method GET

```

{
  "success": false,
  "error": "User does not exist."
}

```

Gambar 2.4 Contoh stuktur konten *response* gagal dari endpoint method PUT/POST/DELETE

```

{
  "success": true,
  "message": "Save Success"
}

```

Gambar 2.5 Contoh stuktur konten *response* berhasil dari endpoint method
PUT/POST/DELETE

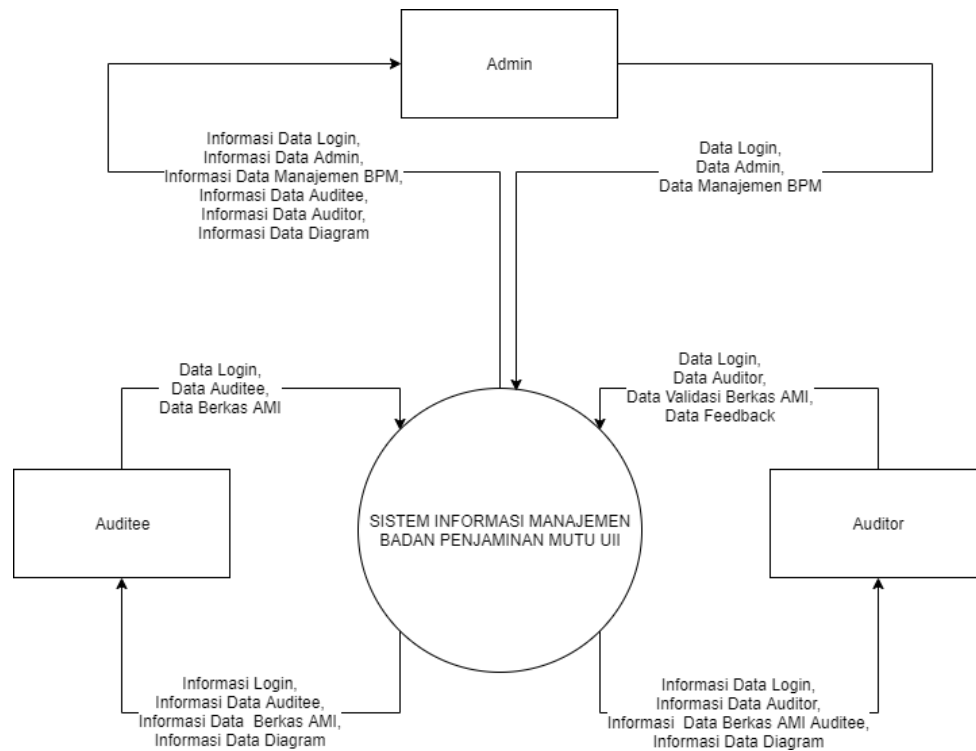
2. **Desain Web Services**

Pada tahapan ini merupakan perancangan *web service* yang bertujuan menghasilkan rancangan dari sistem yang akan dibangun berdasarkan dengan kebutuhan BPM UII. Berikut merupakan perancangan yang diperlukan untuk membangun *web service*:

Data Flow Diagram

Data Flow Diagram menjelaskan aliran/arus data pada sistem BPM UII atau menjelaskan aliran kerja pada BPM UII. DFD membantu untuk mengetahui lebih lanjut sub – sub sistem, mengetahui antar sub – sub sistem, dan mengetahui perkembangan aplikasi BPM secara efektif. Didalam DFD memiliki 3 level, yaitu

1. Diagram Konteks

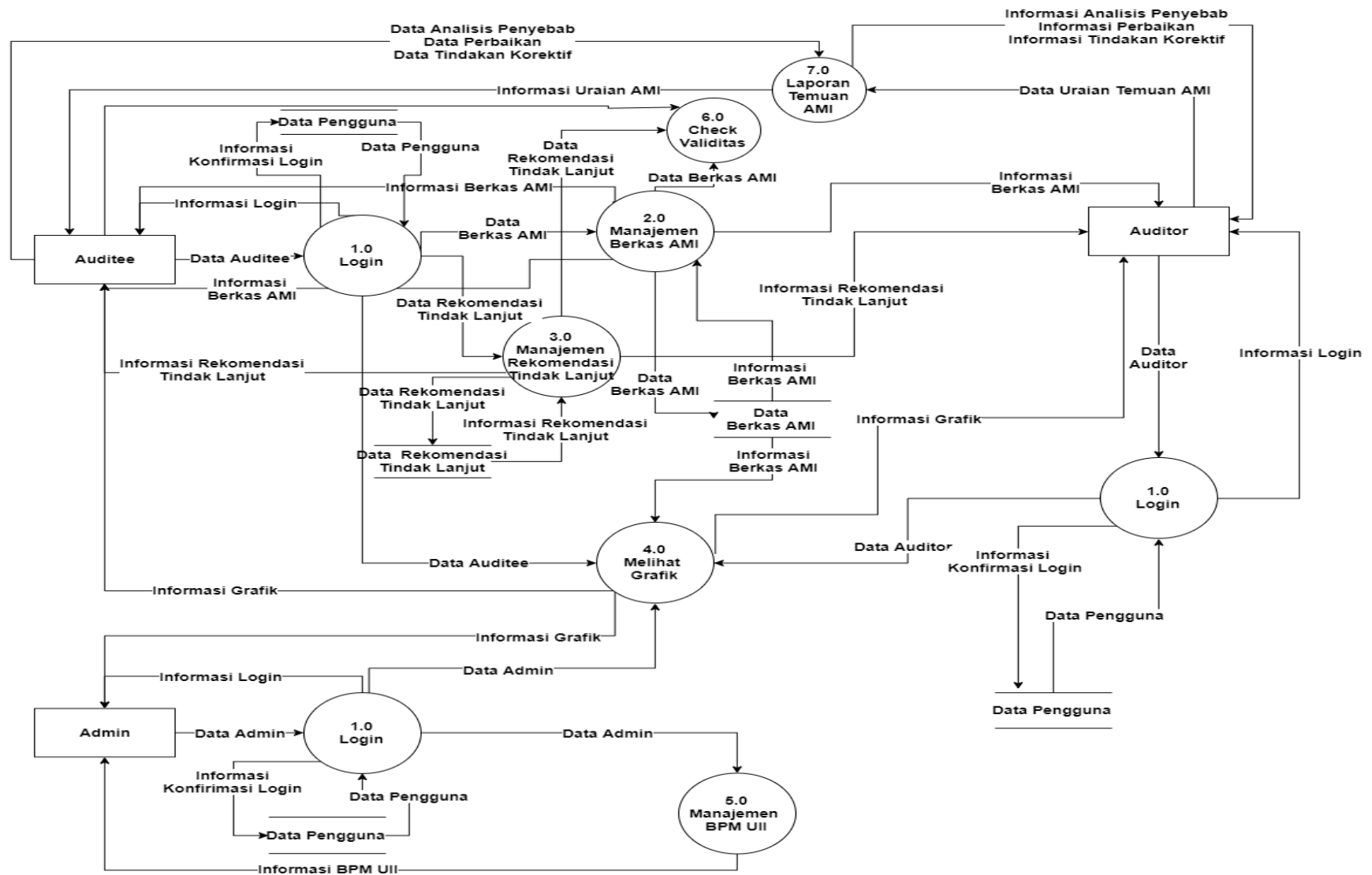


Gambar 2.6 DFD Konteks (DFD Level 0)

Diagram konteks akan menggambarkan gambaran secara besar pada suatu sistem, dan merupakan tingkatan tertinggi dalam DFD. DFD konteks biasanya diberi nomor sebagai DFD Level 0. Pada gambar 3.1 merupakan DFD Konteks yang menjelaskan sistem aliran data keseluruhan sistem. DFD konteks tersebut memiliki 3 terminator (entitas luar) yaitu Auditee, Auditor, dan Admin. Selain itu juga memiliki 1 proses besar dan 6 arus data. DFD Konteks dapat dilihat pada Gambar 2.6.

1. Diagram Level 1

Pada Gambar 2.7 DFD Level 1 merupakan diagram level 1 yang bertujuan untuk menjelaskan lebih rinci dari DFD konteks dan akan memuat penyimpanan data dari setiap proses yang memerlukan. Pada DFD Level 1 memiliki 4 proses yaitu login, manajemen ami, melihat grafik, dan manajemen BPM UII. Pada proses login bertujuan untuk memvalidasi data yang dimasukkan dengan memerlukan data penyimpanan pengguna. Proses manajemen AMI bertujuan untuk menambahkan data AMI, yang akan menghasilkan keluaran berupa nilai capaian, nilai akhir dan rekomendasi tindak. Proses melihat grafik yaitu menampilkan data grafik AMI. Proses yang terakhir yaitu proses manajemen BPM UII yang dilakukan oleh admin, disini admin akan mengawasi seluruh data dan proses yang ada di sistem BPM UII.

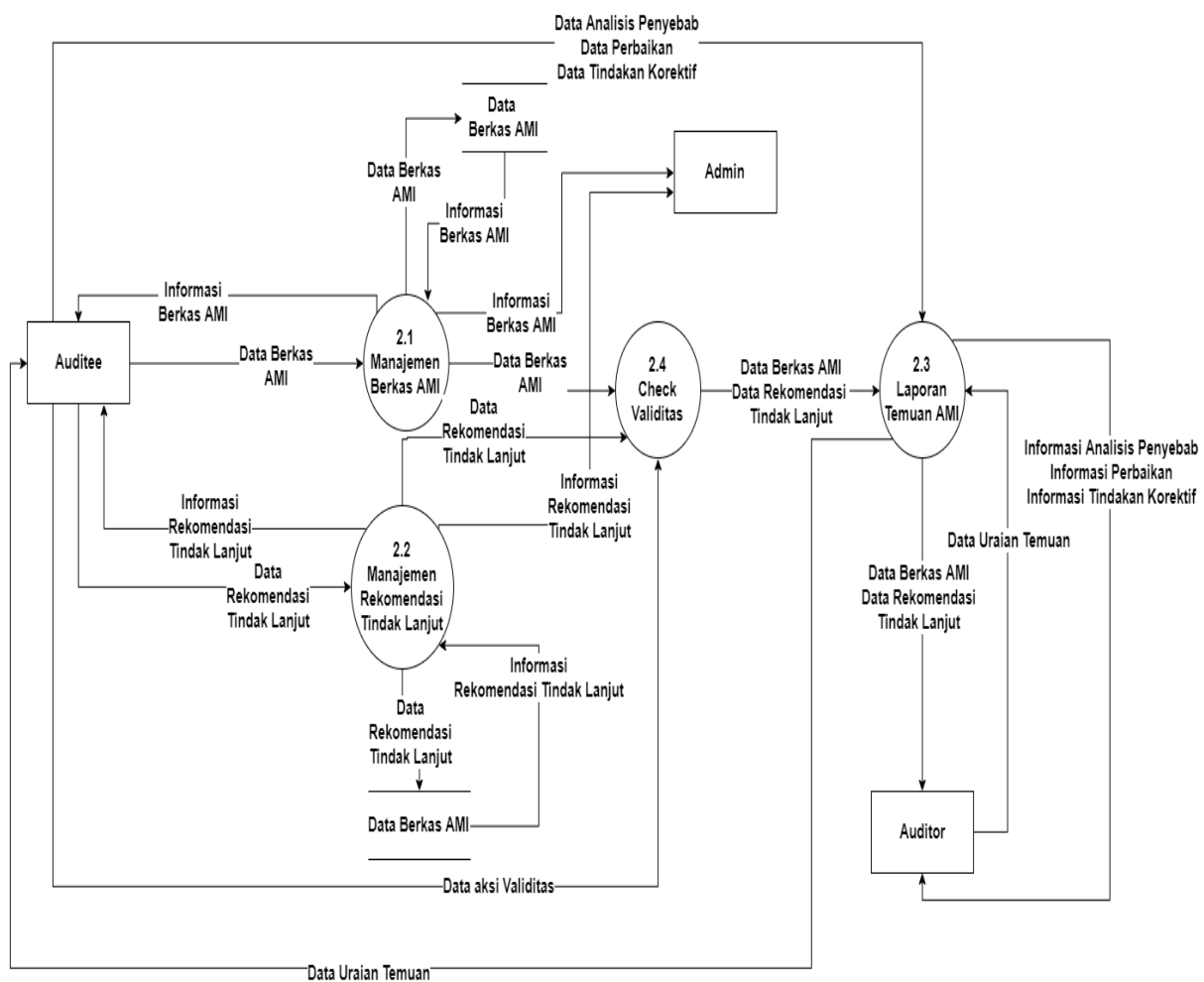


Gambar 2.7 DFD Level 1

2. Diagram Level 2 (Manajemen AMI)

Pada Gambar 2.7 merupakan detail dari proses manajemen AMI yang terdapat pada Gambar 2.7. DFD Level 2 manajemen AMI memiliki 4 proses yaitu

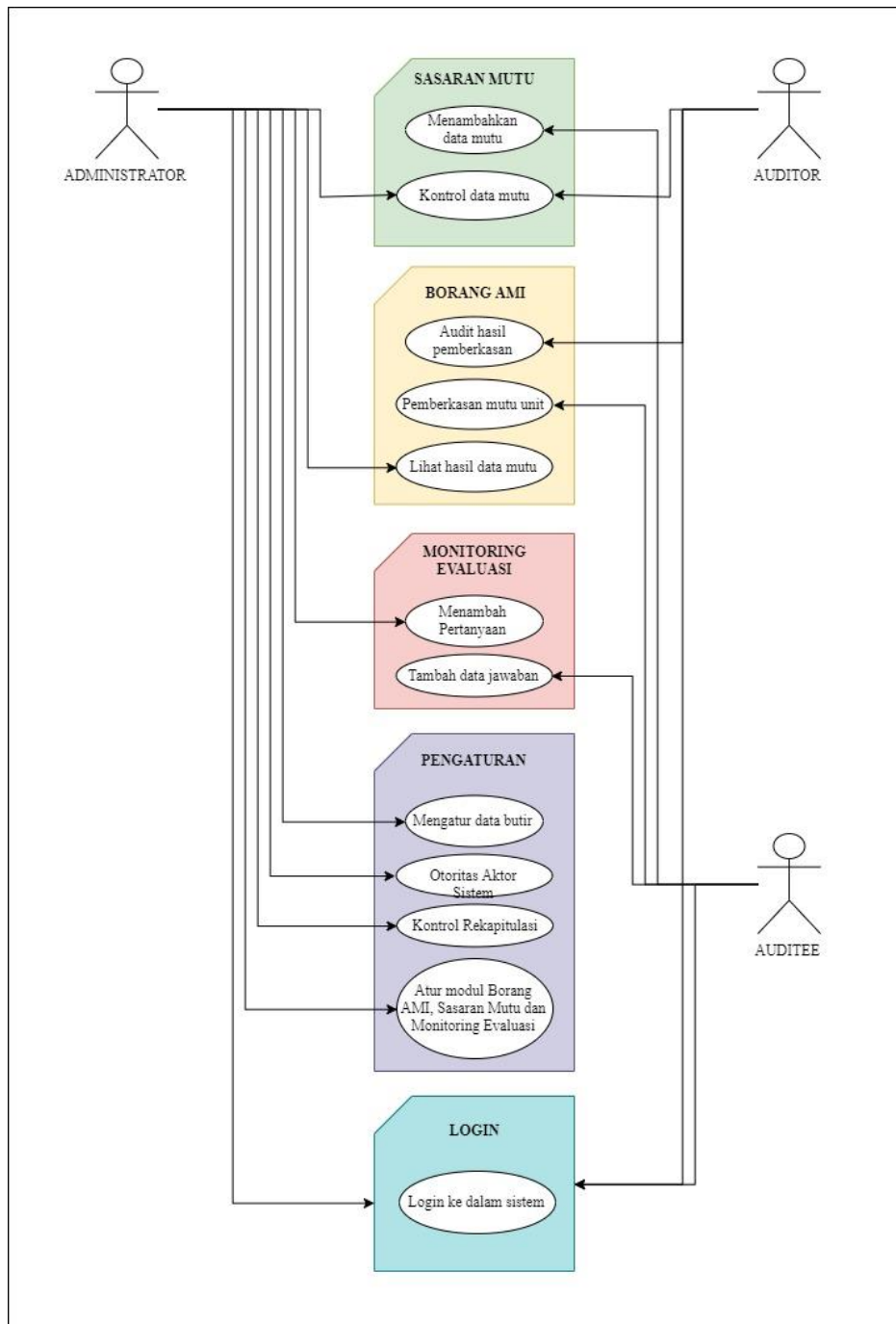
1. Proses manajemen berkas AMI yang berfungsi sebagai pemberkasan data AMI
2. Proses manajemen rekomendasi tindak lanjut bertujuan untuk pendataan data rencana tindak lanjut, waktu, dan Person In Charge (PIC).
3. *Check* validitas bertujuan untuk jika auditee sudah menambahkan semua data berkas AMI dan rekomendasi tindak lanjut maka status butir dari sasaran mutu akan berubah menjadi *check* validasi. *Check* validasi berguna sebagai untuk mengetahui butir sasaran mutu yang sudah diisi penuh.
4. Proses laporan temuan AMI merupakan jika terdapat temuan data AMI yang tidak sesuai maka akan diberi laporan AMI dari auditor ke auditee.



Gambar 2.8 DFD Level 2 Manajemen AMI

Use Case Diagram

Use case diagram akan menggambarkan proses interaksi antara aktor (*stakeholder*) dengan sistem. Dengan merancang *use case diagram* akan memudahkan dalam memahami fungsi yang ada didalam sistem serta aktor yang dapat menggunakan fungsi – fungsi tersebut. *Use case diagram* dapat dilihat pada Gambar 2.9.



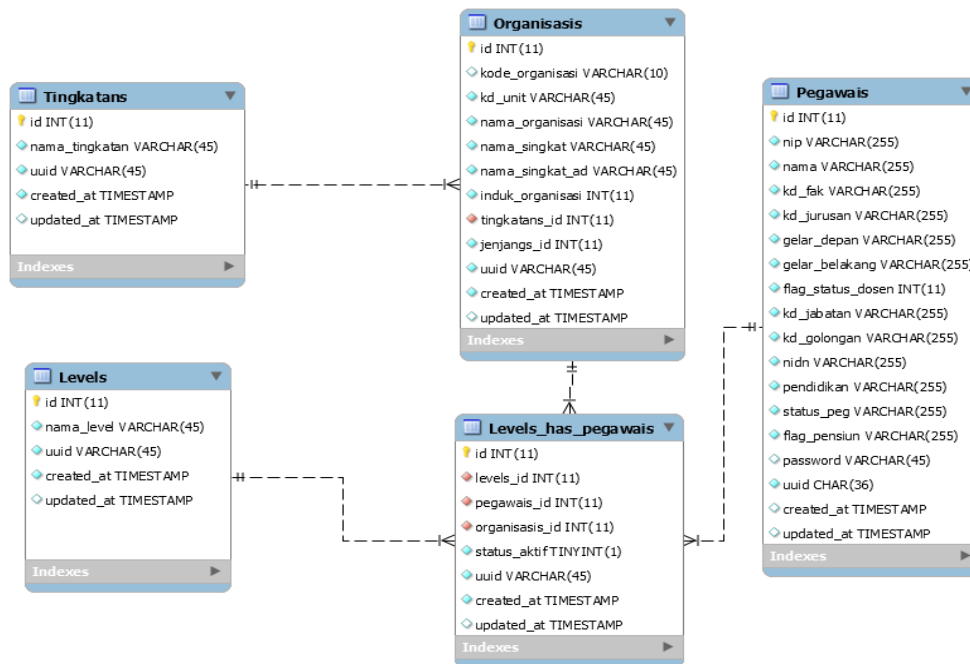
Gambar 2.9 Use Case Diagram

Entity Relationship Diagram (ERD)

ERD yang dirancang akan menggunakan aplikasi MySQL Workbench. MySQL Workbench merupakan salah satu tools aplikasi pembantu pemodelan hubungan antar data dalam database berdasarkan objek data yang memiliki hubungan antar relasi. Sistem BPM UII memiliki jumlah entitas yang begitu banyak karena setiap organisasi memiliki sasaran mutu yang berbeda – beda. Pada laporan Sistem BPM UII ini akan dibagi ke dalam beberapa fungsi dari setiap subjek ERD.

1. ERD Role Pengguna

Subjek ini berguna untuk mengatur segala role yang sudah di atur sedemikian rupa oleh admin. ERD role pengguna memiliki 4 entitas dan 1 entitas *pivot* yaitu tingkatans, organisasi, pegawais, levels, dan levels_has_pegawais (entitas *pivot*) hasil dari kardinalitas many to many antara tabel organisasi, pegawai dan levels. Entitas pegawais berguna untuk menyimpan data pegawai, entitas organisasi menyimpan data organisasi, entitas level menyimpan data level (Auditee, Auditor, dan Admin). ERD role pengguna dapat dilihat pada Gambar 2.10

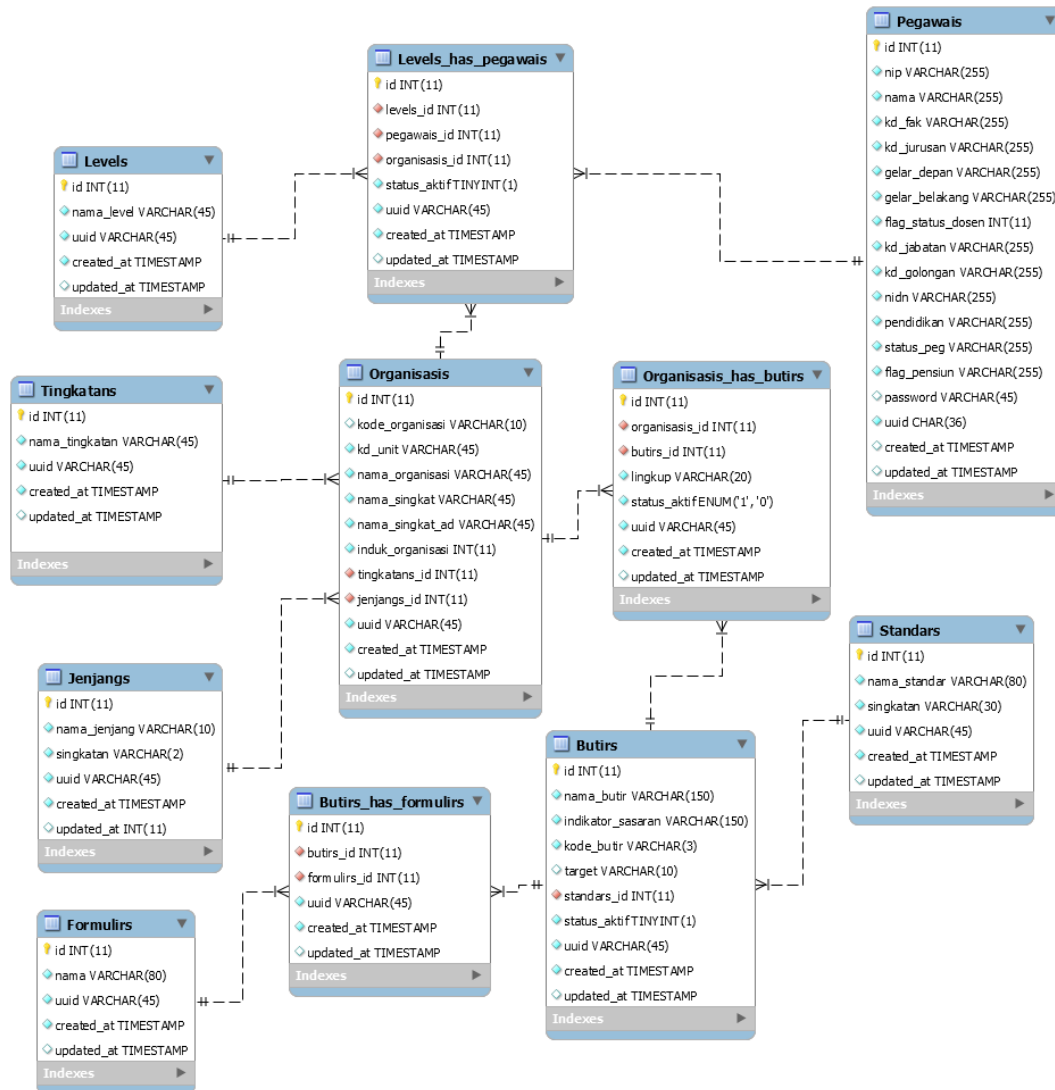


Gambar 2.10 ERD Role Pengguna

2. ERD Cakupan Organisasi

Subjek ini berguna untuk menyimpan *role* cakupan organisasi yang harus sesuai dengan yang diterimanya. ERD cakupan organisasi memiliki 8 entitas dan 3 entitas pivot yaitu entitas

levels, entitas tingkatans, entitas jenjangs, entitas formulirs, entitas butirs, entitas standars, entitas organisasi, entitas pegawais, entitas butirs_has_formulirs (entitas *pivot*) hasil dari kardinalitas *many to many* antara tabel formulir dengan tabel butirs, entitas organisasi_has_butirs hasil dari kardinalitas *many to many* antara tabel organisasi dengan tabel butirs dan entitas levels_has_pegawais hasil dari kardinalitas *many to many* antara tabel organisasi, pegawai dan level. ERD cakupan organisasi dapat dilihat pada Gambar 2.11.

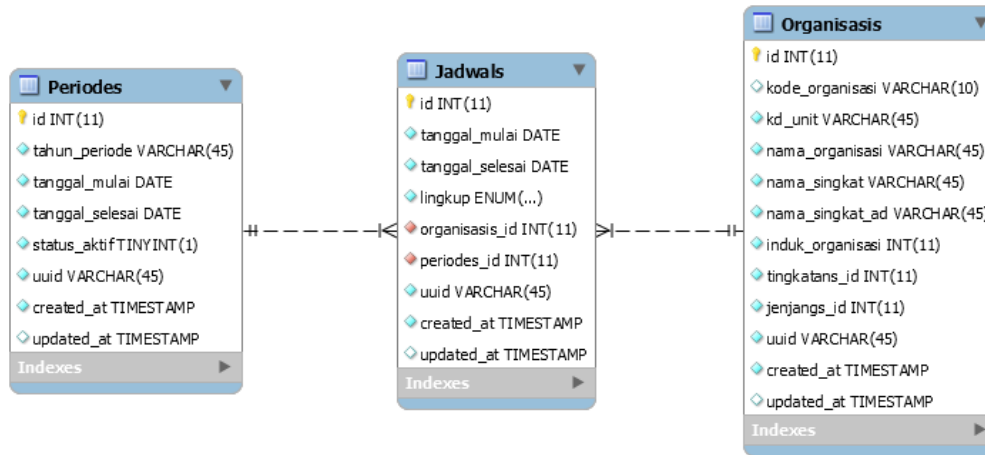


Gambar 2.11 ERD Cakupan Organisasi

3. ERD Jadwal

Subjek ini berguna untuk menyimpan data jadwal pemberkasan auditee, visitasi auditor, dan periode. ERD jadwal memiliki 3 entitas yaitu entitas periode yang berfungsi sebagai menyimpan data setiap periode, entitas jadwal yang berfungsi sebagai menyimpan data jadwal,

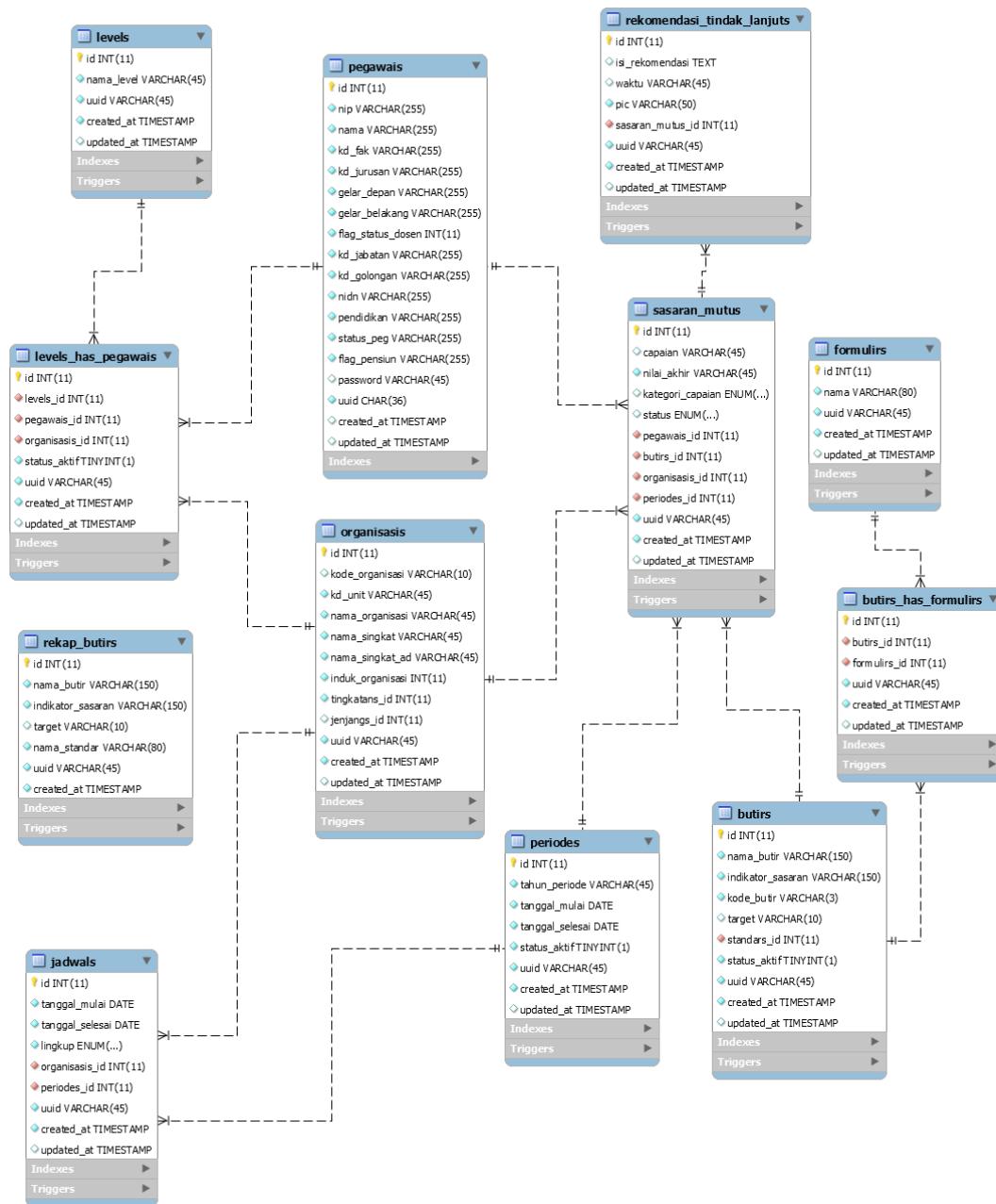
dan entitas organisasi menyimpan data organisasi. ERD jadwal dapat dilihat pada Gambar 2.12.



Gambar 2.12 ERD Jadwal

4. ERD Sasaran Mutu dan Rekomendasi Tindak Lanjut

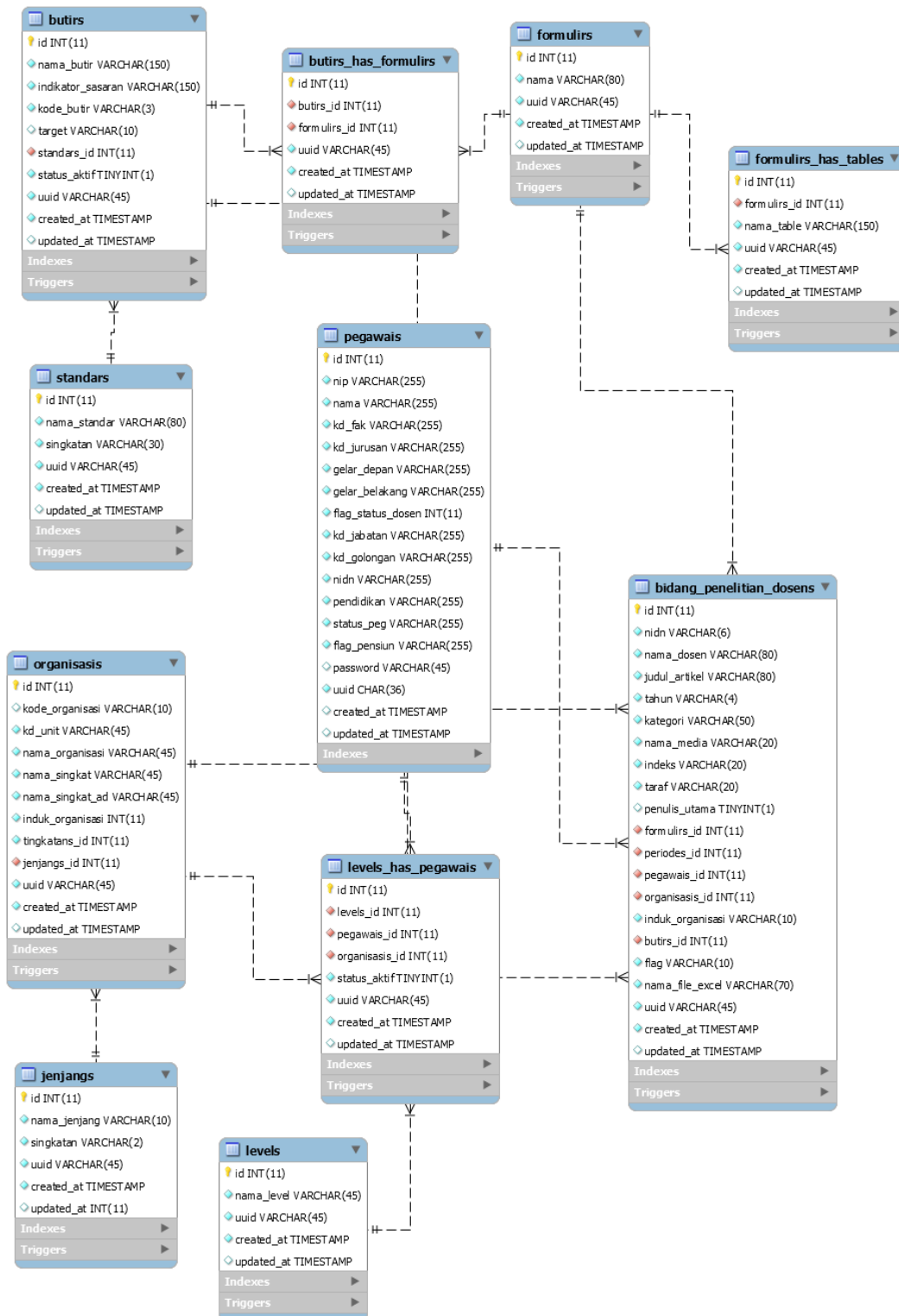
Subjek ini berguna untuk melakukan penyimpanan data sasaran mutu dan rekomendasi tindak lanjut. ERD sasaran mutu memiliki 6 entitas. Entitas sasaran_mutus berguna untuk menyimpan setiap data sasaran mutu dimana hasil tersebut untuk mengukur sasaran mutu setiap organisasi, butir, periode, dan pegawai. Sedangkan entitas rekomendasi_tindak_lanjuts berguna untuk menyimpan data rekomendasi tindak lanjut dari setiap sasaran mutu. Entitas rekap_butirs akan menyimpan data di entitas butir. Ketika entitas butir akan diubah atau dihapus maka data yang sebelum diubah dan dihapus akan disimpan di entitas rekap_butir. ERD sasaran mutu dapat dilihat pada Gambar 2.13.



Gambar 2.13 ERD Sasaran Mutu dan Rekomendasi Tindak Lanjut

5. ERD Bukti Capaian

Subjek ini berguna untuk melakukan pemberkasan AMI yang nantinya akan mendapatkan nilai capaian dari setiap organisasi per butir. Dari setiap pemberkasan akan di simpan di setiap tabel yang memiliki 40 tabel. Entitas pemberkasan borang ami memiliki 40 tabel, sehingga untuk ditampilkan di ERD sangat banyak maka hanya menampilkan 1 tabel contoh yaitu tabel bidang_penelitian_dosens. ERD bukti capaian dapat dilihat pada Gambar 2.14.

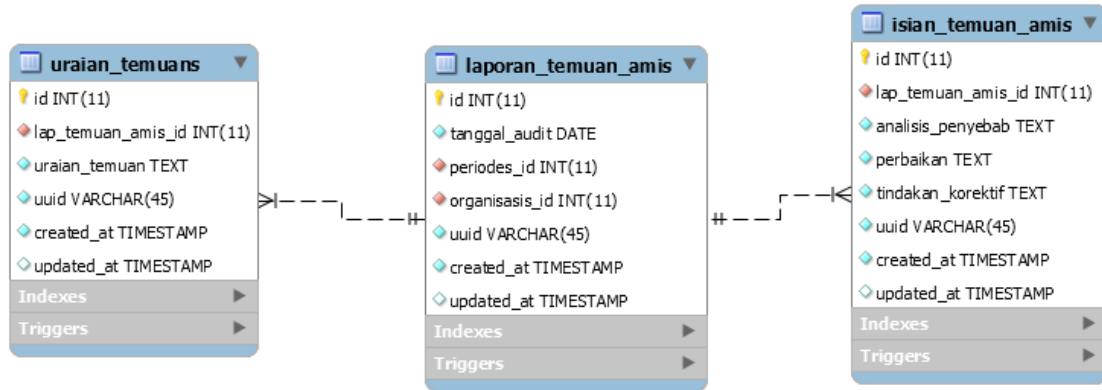


Gambar 2.14 ERD Bukti Capaian

6. ERD Laporan Temuan AMI

Pada rancangan iterasi ketiga terdapat penambahan fitur untuk laporan temuan AMI. laporan temuan AMI merupakan jika terdapat temuan data AMI yang tidak sesuai maka akan

diberi laporan AMI dari auditor ke auditee. Pada Gambar 2.15 merupakan ERD laporan temuan AMI yang memiliki 3 entitas /tabel.



Gambar 2.15 Laporan Temuan AMI

Struktur Tabel

1. Tabel Pegawaiis

Tabel pegawaiis berguna untuk melakukan penyimpanan data pegawai yang akan digunakan untuk proses pendataan pegawai dan validasi login. Tabel pegawai berisikan data yang akan didapatkan dari data Badan Sistem Informasi Universitas Islam Indonesia (BSI UII). Tabel pegawaiis dapat dilihat pada Tabel 2.1.

Tabel 2.1 Tabel Pegawaiis

No	Nama Kolom	Tipe Data	Constraint
1	id	Int(11)	Primary Key
2	nip	Varchar(255)	
3	nama	Varchar(255)	
4	kd_fak	Varchar(255)	
5	kd_jurusan	Varchar(255)	
6	gelar_depan	Varchar(255)	
7	gelar_belakang	Varchar(255)	
8	flag_status_dosen	Int(11)	
9	kd_jabatan	Varchar(255)	
10	kd_golongan	Varchar(255)	

No	Nama Kolom	Tipe Data	Constraint
11	nidn	Varchar(255)	
12	pendidikan	Varchar(255)	
13	status_peg	Varchar(255)	
14	flag_pensiun	Varchar(255)	
15	password	Varchar(45)	
16	uuid	Char(36)	
17	created_at	Timestamp	
18	updated_at	Timestamp	

2. Tabel Levels

Tabel levels berguna untuk melakukan penyimpanan data level yang akan digunakan untuk proses *role* pengguna yang akan masuk ke sistem. Tabel levels berisikan data yaitu Auditee, Auditor, dan Admin. Tabel levels dapat dilihat pada Tabel 2.2.

Tabel 2.2 Tabel Levels

No	Nama Kolom	Tipe Data	Constraint
1	Id	Int(11)	<i>Primary Key</i>
2	nama_level	Varchar(45)	
3	uuid	Varchar(45)	
4	created_at	Timestamp	
5	updated_at	Timestamp	

3. Tabel Tingkatans

Tabel tingkatans berguna untuk melakukan penyimpanan data tingkatans yang akan digunakan untuk proses *role* pengguna yang akan masuk ke sistem. Tabel tingkatans berisikan data yaitu Auditee, Auditor, dan Admin. Tabel tingkatans dapat dilihat pada Tabel 2.3.

Tabel 2.3 Tabel Tingkatans

No	Nama Kolom	Tipe Data	Constraint
1	id	Int(11)	<i>Primary Key</i>
2	nama_singkatan	Varchar(45)	
3	uuid	Varchar(45)	
4	created_at	Timestamp	

No	Nama Kolom	Tipe Data	Constraint
5	updated_at	Timestamp	

4. Tabel Organisasi

Tabel organisasi berguna untuk melakukan penyimpanan data organisasi UII yang akan digunakan untuk proses pendataan organisasi dan *role* pengguna. Tabel organisasi berisikan data yang akan didapatkan dari data Badan Sistem Informasi Universitas Islam Indonesia (BSI UII). Tabel organisasi dapat dilihat pada Tabel 2.4.

Tabel 2.4 Tabel Organisasi

No	Nama Kolom	Tipe Data	Constraint
1	id	Int(11)	<i>Primary Key</i>
2	kode_organisasi	Varchar(10)	
3	kd_unit	Varchar(45)	
4	nama_organisasi	Varchar(45)	
5	nama_singkat	Varchar(45)	
6	nama_singkat_ad	Varchar(45)	
7	induk_organisasi	Int(11)	
8	tingkatans_id	Int(11)	<i>Foregin Key</i>
9	jenjangs_id	Int(11)	
10	uuid	Varchar(45)	
11	created_at	Timestamp	
12	updated_at	Timestamp	

5. Tabel Levels_has_pegawais (Tabel Pivot)

Tabel *levels_has_pegawais* berguna untuk melakukan penyimpanan *role* data pengguna. Setiap pengguna memiliki organisasi dan level yang berbeda, sehingga untuk menyimpan *role* data tersebut membutuhkan tabel *levels_has_pegawais*. Tabel *levels_has_pegawais* dapat dilihat pada Tabel 2.5.

Tabel 2.5 Tabel Levels_has_pegawais

No	Nama Kolom	Tipe Data	Constraint
1	id	Int(11)	Primary Key
2	levels_id	Int(11)	Foreign Key
3	pegawais_id	Int(11)	Foreign Key
4	organisasis_id	Int(11)	Foreign Key
5	status_aktif	Tinyint(1)	
6	uuid	Varchar(45)	
7	created_at	Timestamp	
8	updated_at	Timestamp	

6. Tabel Butirs

Tabel butirs berguna untuk melakukan penyimpanan data butir dan untuk proses menampilkan butir sasaran mutu. Setiap butir memiliki beberapa standar yang berbeda. Tabel butirs dapat dilihat pada Tabel 2.6.

Tabel 2.6 Tabel Butirs

No	Nama Kolom	Tipe Data	Constraint
1	id	Int(11)	Primary Key
2	nama_butir	Varchar(150)	
3	indikator_sasaran	Varchar(150)	
4	kode_butir	Varchar(3)	
5	target	Varchar(10)	
6	standars_id	Int(11)	Foreign Key
7	status_aktif	Tinyint(1)	
8	uuid	Varchar(45)	
9	created_at	Timestamp	
10	updated_at	Timestamp	

7. Tabel Organisations_has_butirs (Tabel Pivot)

Tabel organisations_has_butirs berguna untuk melakukan penyimpanan *role* organisasi dengan butir, Setiap organisasi memiliki beberapa butir yang berbeda, sehingga untuk

menyimpan *role* data tersebut membutuhkan tabel `organisasi_has_butirs`. Tabel `organisasi_has_butirs` dapat dilihat pada Tabel 2.7.

Tabel 2.7 Tabel `Organisasi_has_butirs` (Tabel Pivot)

No	Nama Kolom	Tipe Data	Constraint
1	id	Int(11)	<i>Primary Key</i>
2	organisasi_id	Int(11)	<i>Foreign Key</i>
3	butirs_id	Int(11)	<i>Foreign Key</i>
4	lingkup	Varchar(20)	
5	status_aktif	Tinyint(1)	
6	uuid	Varchar(45)	
7	created_at	Timestamp	
8	updated_at	Timestamp	

8. Tabel Standars

Tabel standars berguna untuk melakukan penyimpanan data standar dan untuk proses menampilkan standar sasaran mutu. Tabel standars dapat dilihat pada Tabel 2.8.

Tabel 2.8 Tabel Standars

No	Nama Kolom	Tipe Data	Constraint
1	id	Int(11)	<i>Primary Key</i>
2	nama_standar	Varchar(80)	
3	singkatan	Varchar(30)	
4	uuid	Varchar(45)	
5	created_at	Timestamp	
6	updated_at	Timestamp	

9. Tabel Formulirs

Tabel formulirs berguna untuk melakukan penyimpanan data nama formulir. Tabel formulirs dapat dilihat pada Tabel 2.9.

Tabel 2.9 Tabel formulirs

No	Nama Kolom	Tipe Data	Constraint
1	id	Int(11)	<i>Primary Key</i>
2	nama	Varchar(80)	
3	uuid	Varchar(45)	
4	created_at	Timestamp	
5	updated_at	Timestamp	

10. Tabel Butirs_has_formulirs (Tabel Pivot)

Tabel butirs_has_formulirs berguna untuk melakukan penyimpanan *role* data butir dengan formulir. Setiap butir memiliki beberapa formulir yang berbeda, sehingga untuk menyimpan *role* data tersebut membutuhkan tabel butirs_has_formulirs. Tabel butirs_has_formulirs dapat dilihat pada Tabel 2.10.

Tabel 2.10 Tabel Butirs_has_formulirs (Tabel Pivot)

No	Nama Kolom	Tipe Data	Constraint
1	id	Int(11)	<i>Primary Key</i>
2	butirs_id	Int(11)	<i>Foreign Key</i>
3	formulirs_id	Int(11)	<i>Foreign Key</i>
4	uuid	Varchar(45)	
5	created_at	Timestamp	
6	updated_at	Timestamp	

11. Tabel Periodes

Tabel periodes berguna untuk melakukan penyimpan data periode sasaran mutu dan proses pergantian periode sasaran mutu. Tabel periodes dapat dilihat pada Tabel 2.11.

Tabel 2.11 Tabel Periodes

No	Nama Kolom	Tipe Data	Constraint
1	id	Int(11)	<i>Primary Key</i>
2	tahun_periode	Varchar(45)	
3	tanggal_mulai	Date	
4	tanggal_selesai	Date	

No	Nama Kolom	Tipe Data	Constraint
5	status_aktif	Tinyint(1)	
6	uuid	Varchar(45)	
7	created_at	Timestamp	
8	updated_at	Timestamp	

12. Tabel Jadwal

Tabel jadwal berguna untuk menyimpan data jadwal pemberkasan, visitasi auditor, dan revisi. Tabel jadwal dapat dilihat pada Tabel 2.12.

Tabel 2.12 Tabel Jadwal

No	Nama Kolom	Tipe Data	Constraint
1	id	Int(11)	<i>Primary Key</i>
2	tanggal_mulai	Date	
3	tanggal_selesai	Date	
4	lingkup	Enum(auditee, auditor, reivisi)	
5	organisasis_id	Int(11)	<i>Foreign Key</i>
6	periodes_id	Int(11)	<i>Foreign Key</i>
7	uuid	Varchar(45)	
8	created_at	Timestamp	
9	updated_at	Timestamp	

13. Tabel Sasaran Mutus

Tabel sasaran mutus berguna untuk menyimpan data sasaran mutus dari setiap pegawai, organisasi, dan periode yang berbeda – beda. Tabel sasaran mutus dapat dilihat pada Tabel 2.13.

Tabel 2.13 Tabel Sasaran Mutus

No	Nama Kolom	Tipe Data	Constraint
1	id	Int(11)	<i>Primary Key</i>
2	capaian	Varchar(45)	

No	Nama Kolom	Tipe Data	Constraint
3	nilai_akhir	Varchar(45)	
4	kategori_capaian	Varchar(45)	
5	status	Enum(auditee, auditor, reivisi)	
6	pegawais_id	Int(11)	<i>Foreign Key</i>
7	organisasis_id	Int(11)	<i>Foreign Key</i>
8	perodes_id	Int(11)	<i>Foreign Key</i>
9	uuid	Varchar(45)	
10	created_at	Timestamp	
11	updated_at	Timestamp	

14. Tabel Rekomendasi Tindak Lanjuts

Tabel rekomendasi tindak lanjut berguna untuk menyimpan data dari beberapa sasaran mutu yang berbeda. Tabel rekomendasi tindak lanjut dapat dilihat pada Tabel 2.14.

Tabel 2.14 Tabel Rekomendasi Tindak Lanjuts

No	Nama Kolom	Tipe Data	Constraint
1	id	Int(11)	<i>Primary Key</i>
2	isi_rekomendasi	Text	
3	waktu	Varchar(45)	
4	pic	Varchar(50)	
5	sasaran_mutus_id	Int(11)	<i>Foreign Key</i>
6	uuid	Varchar(45)	
7	created_at	Timestamp	
8	updated_at	Timestamp	

15. Tabel Formulirs_has_table

Tabel formulirs_has_table berguna untuk menyimpan data dari *role* formulirs dengan table. Setiap formulirs memiliki table yang berbeda, sehingga untuk mengatur *role* tersebut membutuhkan penyimpanan data. Tabel formulirs_has_table dapat dilihat pada Tabel 2.15.

Tabel 2.15 Tabel Formulir_has_table

No	Nama Kolom	Tipe Data	Constraint
1	id	Int(11)	Primary Key
2	formulirs_id	Int(11)	
3	nama_table	Varchar(150)	
4	uuid	Varchar(45)	
5	created_at	Timestamp	
6	updated_at	Timestamp	

16. Tabel Laporan_temuan_amis

Tabel laporan_temuan_amis berguna untuk menyimpan data tanggal audit dari temuan ami. Tabel laporan_temuan_amis dapat dilihat pada Tabel 2.16.

Tabel 2.16 Laporan_temuan_amis

No	Nama Kolom	Tipe Data	Constraint
1	id	Int(11)	Primary Key
2	tanggal_audit	Date	
3	perodes_id	Int(11)	Foreign Key
4	organisasis_id	Int(11)	Foreign Key
5	uuid	Varchar(45)	
6	created_at	Timestamp	
7	updated_at	Timestamp	

17. Tabel Uraian_temuans

Tabel uraian_temuans berguna untuk menyimpan data uraian temuan di laporan temuan ami. Tabel uraian_temuans berelasi dengan tabel laporan_temuan_amis dengan kardinalitas *many to one*.

Tabel 2.17 Tabel Uraian_temuans

No	Nama Kolom	Tipe Data	Constraint
1	id	Int(11)	Primary Key
2	lap_temuan_amis_id	Int(11)	Foreign Key

No	Nama Kolom	Type Data	Constraint
3	uraian_temuan	Text	
4	uuid	Varchar(45)	
5	created_at	Timestamp	
6	updated_at	Timestamp	

18. Tabel Isian_temuan_amis

Tabel isian_temuan_amis berguna untuk menyimpan data analisis penyebab, perbaikan, dan tindakan korektif di laporan temuan ami. Tabel uraian_temuans berelasi dengan tabel laporan_temuan_amis dengan kardinalitas *many to one*.

Tabel 2.18 Tabel Isian_temuan_amis

No	Nama Kolom	Type Data	Constraint
1	id	Int(11)	<i>Primary Key</i>
2	lap_temuan_amis_id	Int(11)	<i>Foreign Key</i>
3	analisis_penyebab	Text	
4	perbaikan	Text	
5	tindakan_korektif	Text	
6	uuid	Varchar(45)	
7	created_at	Timestamp	
8	updated_at	Timestamp	

1. Mempersiapkan studi kasus pengujian *Web Services*

Pada tahapan ini mendefinisikan kasus pengujian *web services* setelah dibangun. Berikut merupakan kasus pengujian *web services* yang dapat dilihat pada Tabel 2.19.

Tabel 2.19 Desain Kasus Pengujian

No	Endpoint	Deskripsi
1	Login	Melakukan pengujian terhadap <i>endpointlogin</i> dan penanganan kesalahan <i>endpointlogin</i>
2	List peran pengguna	Melakukan pengujian terhadap <i>endpoint listperan pengguna</i> dan penanganan kesalahan <i>endpoint list peran pengguna</i>

No	Endpoint	Deskripsi
3	Sasaran mutu	Melakukan pengujian terhadap semua <i>endpoint</i> sasaran mutu dan penanganan kesalahan semua <i>endpoint</i> sasaran mutu
4	Bukti capaian	Melakukan pengujian terhadap semua <i>endpoint</i> bukti capaian dan penanganan kesalahan semua <i>endpoint</i> bukti capaian
5	Rekomendasi tindak lanjut	Melakukan pengujian terhadap semua <i>endpoint</i> rekomendasi tindak lanjut dan penanganan kesalahan semua <i>endpoint</i> rekomendasi tindak lanjut
6	Tampil keranjang validasi	Melakukan pengujian <i>endpoint</i> keranjang validasi dan penanganan kesalahan <i>endpoint</i> keranjang validasi
7	Memvalidasi keranjang validasi	Melakukan pengujian <i>endpoint</i> memvalidasi keranjang validasi dan penanganan kesalahan <i>endpoint</i> keranjang validasi
8	Laporan temuan AMI	Melakukan pengujian <i>endpoint</i> laporan temuan AMI dan penanganan kesalahan <i>endpoint</i> laporan temuan AMI
11	Grafik	Melakukan pengujian <i>endpoint</i> grafik dan penanganan kesalahan <i>endpoint</i> grafik
12	Otorisasi	Melakukan pengujian ke salah satu <i>endpoint</i> yang memerlukan otorisasi
13	<i>Cut Off</i>	Melakukan pengujian jika sudah terjadi <i>cut off</i>
14	Peforma <i>Endpoint</i>	Melakukan pengujian peforma setiap <i>endpoint</i>