



Rancang Bangun Aplikasi Pembelajaran Siswa SMA Berbasis Android

TUGAS AKHIR



Oleh:

Zayed Elfasa

11410100127

FAKULTAS TEKNOLOGI DAN INFORMATIKA

INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA

2018

ABSTRAK

Proses pembelajaran siswa SMA dari tahun ke tahun mengalami perkembangan yang sangat pesat diikuti dengan perkembangan teknologi yang cepat. Proses perkembangan tersebut dimulai dari guru sebagai sumber kebutuhan belajar untuk siswa yang kemudian berubah menjadi siswa yang membutuhkan untuk belajar. Hal ini menjadi kebutuhan siswa saat ini berbeda dengan siswa masa lalu. Siswa saat ini ada pada era *digital native* yang dekat dengan *gadget*. *Gadget* dapat digunakan oleh guru dan siswa untuk belajar secara *mobile* kapan saja dan dimana saja.

Aplikasi pembelajaran berbasis *mobile* yang beredar saat ini tidak memiliki standar pembelajaran. Standar pembelajaran siswa terdiri dari kompetensi dasar, kompetensi inti, dan indikator sebagai acuan untuk materi, soal, dan tugas. Selain itu guru maupun siswa tidak memiliki forum dalam sebuah kelas untuk saling berdiskusi. Solusi dari permasalahan tersebut adalah dibuat sebuah aplikasi pembelajaran berbasis *mobile* yang memiliki standar kompetensi untuk menampilkan materi, tugas, dan soal yang juga dapat digunakan untuk berdiskusi melalui pesan teks. Aplikasi yang dibuat dapat menilai soal dan tugas yang telah dikerjakan oleh siswa ketika ujian atau tugas telah selesai dikerjakan.

Hasil dari pembuatan aplikasi ini adalah semua materi, tugas, dan soal yang ditampilkan berdasarkan standar kompetensi mulai dari kompetensi dasar, kompetensi inti, dan indikator sehingga materi yang disebar kepada siswa telah sesuai dengan standar. Aplikasi ini akan dikembangkan untuk versi iOS dan akan dimasukkan sebanyak 22 mata pelajaran.

Kata Kunci : Aplikasi, Siswa SMA, *Mobile Learning*, Standar Kompetensi

DAFTAR ISI

	Halaman
ABSTRAK	vi
KATA PENGANTAR	vii
DAFTAR ISI.....	ix
DAFTAR TABEL.....	xii
DAFTAR GAMBAR	xvii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Masalah.....	1
1.2 Perumusan Masalah.....	3
1.3 Batasan masalah	3
1.4 Tujuan.....	4
1.5 Sistematika Penulisan.....	4
BAB II LANDASAN TEORI.....	7
2.1 Belajar dan Pembelajaran.....	7
2.2 Generasi <i>Digital Native</i>	7
2.3 <i>Mobile Learning</i>	8
2.4 Penelitian Terdahulu.....	10
2.5 Android.....	11
2.6 Analisa Berorientasi Object.....	12
2.7 Pengembangan Perangkat Lunak Menggunakan <i>Scrum</i>	12

2.8	Model Penelitian.....	13
BAB III ANALISIS DAN PERANCANGAN SISTEM.....		17
3.1	Analisis Sistem.....	17
3.1.1	Identifikasi Masalah.....	17
3.1.2	Analisis Kebutuhan Pengguna.....	17
3.2	Perancangan Sistem.....	20
3.2.1	<i>Use Case Diagram</i>	20
3.2.2	<i>Flow Of Event</i>	22
3.2.3	<i>Sequence Diagram</i>	52
3.2.4	<i>Class Diagram</i>	67
3.2.5	<i>Class Model</i>	67
3.2.6	<i>Class Controller</i>	83
3.2.7	Relasi <i>Class Diagram</i> dengan Entitas.....	92
3.2.8	Diagram Komponen.....	114
3.2.9	Diagram <i>Deployment</i>	116
3.2.10	Disain <i>Input / Output</i>	116
BAB IV IMPLEMENTASI DAN EVALUASI.....		130
4.1	Implementasi <i>Scrum</i>	130
4.1.1	<i>Product Backlog</i>	130
4.1.2	<i>Sprint Ke-1</i>	131
4.1.3	<i>Sprint Ke-2</i>	145

4.1.4	<i>Sprint Ke-3</i>	164
4.1.5	<i>Sprint Ke-4</i>	174
BAB V PENUTUP.....		181
5.1.	Kesimpulan.....	181
5.2.	Saran	181
DAFTAR PUSTAKA		182
BIODATA PENULIS		183



BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Proses belajar mengajar memiliki perubahan paradigma di beberapa tahun belakangan ini. Menurut Indra (2001) bahwa proses belajar mengajar dimana paradigma mengajar seperti yang selama ini diterapkan perlu diubah menjadi belajar. Maksudnya adalah selama ini yang dikenal adalah proses belajar mengajar akan diubah menjadi proses bagaimana belajar bersama antara guru dan siswa. Antara guru dan siswa (setiap individu) akan aktif belajar bersama-sama.

Perubahan zaman akan membuat kebutuhan pada siswa menjadi berbeda yang disesuaikan zamannya. Zaman ini disebut dengan zaman globalisasi yang membuka pandangan manusia terhadap dunia luar. Setiap individu dapat terhubung melalui sebuah alat secara *realtime*. Saat inilah setiap individu dalam hal ini siswa dapat membuka dunia karena media seperti *smartphone* menjadi alat untuk mencari pengetahuan dalam jumlah yang tidak terbatas. Menurut Suyanto (2017) perbedaan antara pembelajaran secara tradisional dan *web based learning* maupun *mobile learning* adalah secara tradisional fokus pada pendidik yang menyalurkan ilmu pada siswa sedangkan *web based learning* maupun *mobile learning* adalah fokus utamanya akan memaksa para siswa secara mandiri untuk belajar dan bertanggung jawab atas apa yang telah dipelajarinya.

E-Learning yang berkembang saat ini bergerak menuju pembelajaran berbasis *mobile*. Ada beberapa aplikasi berbasis *mobile* yang telah digunakan saat ini seperti *Quipper*, *Edmodo* dan *Google Classes* yang berjalan di Android. *Quipper* adalah aplikasi *e-learning* yang digunakan oleh siswa dimana di dalam aplikasi terdapat materi, tugas, dan soal. *Edmodo* adalah aplikasi yang menjadi pihak penghubung antara guru, orang tua, dan siswa agar siswa dapat belajar secara terkendali. Sedangkan *Google Classes* adalah aplikasi yang dapat menghubungkan guru dan siswa dalam sebuah kelas digital. Dalam kelas digital tersebut guru memberikan materi untuk dipelajari oleh siswa dan tugas yang dikerjakan juga oleh siswa.

Pembelajaran menggunakan aplikasi *Quipper*, *Edmodo*, dan *Google Classes* tidak memiliki adaptasi dan keterjangkauan dalam arti tidak ada standar kebutuhan materi dan soal untuk didistribusikan kepada para siswa. Standarisasi materi yang didistribusikan dibutuhkan forum diskusi yang mencakup guru untuk saling berkomunikasi untuk membuat materi ataupun soal. Standar materi tersebut akan menggunakan kompetensi mata pelajaran yang telah dibuat oleh pemerintah. Guru membutuhkan standar materi untuk pembuatan materi dan soal yang dapat didistribusikan kepada para siswa yang diajarkan pada sebuah kelas ajar.

Usulan dari media pembelajaran yang sesuai dengan kebutuhan tersebut adalah membuat aplikasi pembelajaran *Mobile Learning Smartphone* berbasis Android yang dinamakan "*MoLearn* berbasis Android". Aplikasi ini sebagai wadah untuk pembelajaran mandiri yang berfokus pada siswa dengan materi yang sesuai dengan standar kompetensi. *MoLearn* akan memiliki materi untuk belajar mandiri siswa dan dapat berkolaborasi antar individu bersama guru.

Pembuatan aplikasi pembelajaran ini akan bekerja sama dengan Musyawarah Guru Mata Pelajaran (MGMP). MGMP adalah forum para guru yang mengajar pada mata pelajaran yang sama untuk saling berkoordinasi memecahkan masalah secara umum yang masih berkaitan dengan mata pelajaran. MGMP inilah yang akan memberikan masukan-masukan untuk pembuatan aplikasi ini. MGMP yang akan bekerja sama adalah MGMP Biologi dan Geografi Surabaya.

1.2 Perumusan Masalah

Berdasarkan uraian latar belakang masalah yang telah dipaparkan maka telah dirumuskan masalahnya adalah bagaimana merancang dan membangun aplikasi *MoLearn* berbasis Android?

Sub masalah di atas adalah sebagai berikut:

1. Bagaimana membuat aplikasi yang dapat menampilkan materi pembelajaran pada Android?
2. Bagaimana membuat aplikasi yang dapat menampilkan soal-soal pada Android dan memberikan nilai ujian pada siswa?
3. Bagaimana membuat aplikasi yang dapat menampilkan tugas dan memberikan nilai tugas pada Android?
4. Bagaimana membuat aplikasi untuk berdiskusi melalui forum pada Android?

1.3 Batasan masalah

Berdasarkan perumusan masalah tersebut batasan masalah pada penelitian adalah sebagai berikut :

1. Aplikasi yang dikembangkan pada sistem berbasis Android.

2. Aplikasi memiliki tiga pengguna yaitu MGMP, guru, dan siswa.
3. Data yang digunakan untuk aplikasi ini berasal dari MGMP Biologi dan Geografi.
4. Aplikasi Android ini untuk menampilkan materi dari MGMP dan guru.
5. Aplikasi Android ini untuk menampilkan tugas dan soal ujian.
6. Aplikasi Android ini untuk menilai tugas siswa dan ujian siswa.
7. Aplikasi Android ini untuk melakukan diskusi melalui forum.

1.4 Tujuan

Dengan melihat perumusan masalah yang ada, maka tujuan yang ingin dicapai dalam penelitian ini adalah menghasilkan aplikasi *MoLearn* untuk MGMP, guru, dan siswa. Sub tujuan dari penelitian ini adalah :

1. Menghasilkan aplikasi berbasis Android yang dapat menampilkan materi yang berasal dari MGMP dan guru.
2. Menghasilkan aplikasi berbasis Android yang dapat menampilkan soal-soal dan memberikan nilai ujian pada siswa.
3. Menghasilkan aplikasi berbasis Android yang dapat menampilkan tugas dan menilai tugas yang dikerjakan oleh siswa.
4. Menghasilkan aplikasi berbasis Android yang dapat melakukan diskusi melalui forum.

1.5 Sistematika Penulisan

Laporan Tugas Akhir ini memiliki lima bagian yang dibagi tiap bab. Sistematika penulisan bab sebagai berikut:

BAB I : Pendahuluan

Bab ini menjelaskan tentang latar belakang dari topik tugas akhir ini, rumusan masalah, batasan masalah (ruang lingkup), dan tujuan dari pembuatan tugas akhir ini.

BAB II : Landasan Teori

Bab ini menjelaskan gambaran apa itu *Mobile Learning* dan menjelaskan teori-teori yang berhubungan dengan *mobile learning* dan pengembangan sistem dimulai dari belajar dan pembelajaran, generasi *digital native*, *mobile learning*, *Android*, dan *scrum*.

BAB III : Metode Penelitian

Bab ini diisi dengan analisis sistem, identifikasi masalah, analisis kebutuhan pengguna, perancangan sistem, dan disain *input/output*, disain uji coba berdasarkan kerangka kerja *Scrum*.

BAB IV : Implementasi dan Evaluasi

Bab ini menjelaskan tentang implementasi dari analisis dimulai dari *product backlog*, dan *sprint planning* yang telah dibuat secara keseluruhan serta melakukan pengujian secara UAT yang telah dibuat untuk mengetahui apakah aplikasi yang dibuat telah sesuai dengan kebutuhan atau tidak.

BAB V : Penutup

Bab ini menjelaskan kesimpulan dan saran. Kesimpulan didapatkan dari rumusan masalah. Kesimpulan akan dijelaskan dari Bab IV hasil evaluasi aplikasi dan saran dijelaskan tentang

masuk dari aplikasi untuk tahap lanjut dari pengembangan aplikasi.



BAB II

LANDASAN TEORI

2.1 Belajar dan Pembelajaran

Belajar adalah kegiatan saling berinteraksi antara tenaga pendidik dan peserta didik. Belajar merupakan suatu proses memperoleh berbagai kecakapan, keterampilan, dan sikap melalui berbagai pengalaman peserta didik sehingga terjadi perubahan tingkah laku (Hariadi, 2011). Pembelajaran adalah upaya membelajarkan peserta didik (Hariadi, 2000). Pemahaman di atas menunjukkan bahwa fokus pada pembelajaran adalah peserta didik untuk merancang disain pembelajaran.

Menurut Degeng (Hariadi, 2011) bahwa adanya pengaruh lingkungan agar peserta didik dapat belajar dengan baik. Lingkungan akan menjadi hal utama dalam kebebasan (fleksibel) kepada peserta didik dalam melakukan proses belajar.

Belajar di setiap zaman memiliki cara yang berbeda-beda tetapi dari pengertian di atas bahwa belajar pada intinya secara aktif mencari ilmu pengetahuan yang diberikan oleh orang lain atau diberikan secara pasif. Zaman dahulu belajar dilakukan dengan cara tradisional yaitu dengan alat-alat yang sangat terbatas. Berbeda dengan zaman ini dimana belajar dapat melalui media mana saja. Belajar generasi saat ini dijelaskan pada sub berikutnya.

2.2 Generasi *Digital Native*

Seorang ilmuwan bernama Jim Marteney yang dikutip oleh (Hasugian, 2011) dibagi menjadi 6 generasi hingga saat ini yaitu:

- a) *The Greatest Generation* (word war II, 1901-1924)

- b) *The Silent Generation* (1925-1945)
- c) *The Baby Boomers* (1943-1960)
- d) *X Generation* (1961 – 1981)
- e) Generasi Millennial (1982 – 2002)
- f) *Digital Native* (1994 – sekarang)

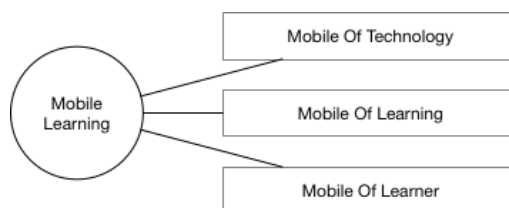
Generasi *Digital Native* inilah namanya lahir karena generasi era digital yang dekat dengan *gadget*. Generasi ini menganggap bahwa *gadget* adalah kebutuhan primer yang tidak dapat dipisahkan dari kehidupan. Contohnya generasi ini adalah seorang anak yang telah mendapatkan pendidikan dari sekolah dasar yang diberikan perangkat komputer maka generasi ini disebut dengan generasi *digital native*.

Ciri khas dari individu digital native adalah pertama identitas. Individu tersebut ingin identitas yang sangat ingin keberadaannya diakui. Kedua adalah privasi. Privasi sangat sempit pada seorang individu karena pemikiran mereka terbuka. Ketiga adalah kebebasan. Keempat adalah proses belajar yang berbeda dengan generasi lainnya. Individu generasi *digital native* tidak membaca buku seri ensiklopedia tetapi mereka tetap tahu isi ensiklopedia dengan cara mencari melalui mesin pencari.

2.3 Mobile Learning

Mobile learning adalah revolusi dari *e-learning*. Pembelajaran *mobile* adalah produk dari revolusi wireless (Keegan, 2004). *Mobile learning* disebut akan menjadi pembelajaran di masa depan. Menurut Traxler (2007) pada dasarnya Mobile Learning tidak dapat diartikan sendiri sendiri. *Mobile Learning* adalah sebuah disiplin ilmu yang mempelajari metode baru cara pembelajaran. *Mobile*

Learning secara garis besar dibagi menjadi tiga yang dijelaskan pada konsep pohon pada Gambar 2.1:



Gambar 2.1 Konsep Pohon *Mobile Learning* (Traxler, 2007)

Mobile Learning juga dapat diartikan sebagai *Mobile E-Learning* karena perkembangan dari *E-Learning* yang diterapkan menjadi *Mobile*. *E-Learning* selama ini dianggap salah satu bagian dari cara pembelajaran secara konvensional kemudian pada era digital native ini dikembangkan menjadi *Mobile Learning* (Medipour & Zerehkafi, 2013).

Dari pengertian diatas bahwa *Mobile Learning* merupakan sebuah bagian baru dalam perkembangan pembelajaran untuk siswa yang diterapkan secara *mobile* di era *digital native* ini.

Mobile Of Technology

Mobile Of Technolgy pada pengertian ini adalah bentuk dari sebuah perangkat keras digital seperti *smartphones* untuk mendukung fasilitas yang akan digunakan untuk melakukan pembelajaran. Fasilitas pendukung tersebut seperti suara yang dihasilkan, video yang dihasilkan, dan sebagainya (Laouris & Eteokleous, 2017).

Mobile Of Learning

Mobile of learning pada pengertian ini adalah metode baru untuk pembelajaran yang tidak kaku di dalam kelas ataupun laboratorium. Dalam

sebuah konferensi MLearn tahun 2004 Metode ini menyediakan fasilitas “belajar kapan saja dan dimana saja” (Laouris & Eteokleous, 2017).

Mobile Of Learner

Mobile Of Learner pada pengertian ini adalah evolusi dari *E-Learning* yang sebagian besar terikat oleh waktu, tempat, dan konfigurasi dari sebuah komputer. *E-Learning* yang berubah menjadi *mobile learning* yang bersifat *nomaden* dan fleksibel tanpa harus terpaku pada suatu tempat yang sama (Laouris & Eteokleous, 2017).

Kemampuan dan penerapan *Mobile Learning* pada *smartphone* saat ini dapat dilihat pada Tabel 2.1:

Tabel 2.1 Kemampuan dan penerapan *Mobile Learning* pada *smartphone* (Medipour & Zerehkafi, 2013)

<i>Subject</i>	<i>M-Learning</i>
Tempat	<ol style="list-style-type: none"> 1. Belajar dimana saja. 2. Tidak ada batasan geografis.
Pedagogi (keilmuan guru)	<ol style="list-style-type: none"> 3. Lebih atraktif pada sisi suara, grafik, animasi, dan instruksi. 4. Belajar terjadi di lapangan atau secara <i>mobile</i>. 5. Fleksibel waktu akses selama 24 jam.
Komunikasi	<ol style="list-style-type: none"> 6. Memungkinkan untuk <i>video teleconference</i> 7. Memungkinkan untuk saling kirim pesan teks 8. Tidak membutuhkan <i>travel time</i> karena adanya hubungan internet.

2.4 Penelitian Terdahulu

Penelitian terdahulu sangat membantu dalam pembuat aplikasi *MoLearn*. Ada dua jurnal terdahulu yang menjelaskan tentang pembelajaran di era *digital*

native ini dan pengaruhnya. Jurnal yang dimaksud adalah:

1. *Development of Web-Based Learning Application for Generation Z* (2016).

Jurnal ini menjelaskan tentang implementasi dari sebuah aplikasi hybrid learning dengan menggunakan *Google Apps For Education* yang dinamakan “*Brillian*”. *Brillian* mengintegrasikan *Google Apps* untuk memberikan materi dari dosen, *upload* tugas, membuat tugas dan lain-lain.

2. *Web-Based Cooperative Learning, Learning Styles, and Student’s Learning Outcomes* (2015).

Jurnal ini menjelaskan tentang pengaruh strategi pembelajaran kooperatif tipe STAD berbasis *web* dan kooperatif tipe STAD berbasis teks. Hasil penelitian pada jurnal ini adalah nilai signifikan antara kedua cara tersebut sehingga disarankan untuk menerapkan strategi pembelajaran kooperatif tipe STAD dengan cara bersamaan atau saling melengkapi.

Penelitian terdahulu ini akan menjadi acuan untuk mengembangkan aplikasi pembelajaran berbasis Android yang di dalamnya terdapat materi ajar dari guru, standar kompetensi, dan lainnya. Untuk lebih jelasnya, fitur-fiturnya dijelaskan pada metode penelitian.

2.5 Android

Menurut Suprianto & Agustina (2012) menerangkan dalam bukunya bahwa Android adalah sistem operasi bergerak (*mobile operating system*) yang mengadopsi sistem operasi *Linux*, namun telah dimodifikasi. Android diambil alih oleh *Google* pada tahun 2005 oleh *Android, Inc* sebagai bagian strategi untuk mengisi pasar sistem operasi bergerak. *Google* mengambil alih seluruh hasil kerja Android termasuk tim yang mengembangkan Android.

Keuntungan utama dari Android adalah adanya pendekatan aplikasi secara terpadu. Pengembang hanya berkonsentrasi pada pembuatan aplikasi saja. Aplikasi tersebut bisa berjalan pada berbeda selama masih dalam system operasi Android (Suprianto & Agustina, 2012).

2.6 Analisa Berorientasi Object

Tohari (2014) sistem berorientasi objek adalah mengorganisasikan sistem perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya secara sistematis.

Tahap analisis dilakukan setelah tahap perencanaan sistem dan sebelum tahap perancangan sistem. Tahap ini merupakan tahap yang kritis, sangat penting, karena kesalahan dalam tahap ini menyebabkan kesalahan pada tahap selanjutnya. Langkah-langkah dasar dalam melakukan analisis sistem:

1. *Identify*, mengidentifikasi masalah dan sasaran dari sistem yang hendak dicapai.
2. *Understand*, mempelajari dan memahami secara terperinci bagaimana sistem yang ada beroperasi, dapat melakukan pengumpulan data untuk mencari kelemahan dan kebutuhan dari pemakai sistem.
3. *Analyze*, setelah data dari pengguna sistem dikumpulkan, data-data tersebut dianalisa untuk menentukan langkah-langkah yang harus dikerjakan.
4. *Report*, dalam tahap ini hasil dan temuan dari tahap analisis sistem dibuatkan laporan agar pihak pengguna dapat mengetahuinya.

2.7 Pengembangan Perangkat Lunak Menggunakan Scrum

Scrum adalah kerangka kerja untuk pembuatan produk yang kompleks. Kompleks artinya di sini adalah produk yang masih diproses yang dikembangkan

akan terus berubah bentuknya dimana saat selesai akan menghasilkan produk dengan nilai setinggi mungkin dilihat dari sisi kreatifitas dan produktifitas.

Scrum ada semenjak tahun 1990-an. *Scrum* bukan proses dalam pengembangan produk tetapi ditekankan pada kerangka kerja. Scrum menggunakan pendekatan berkala dan bertahap untuk meningkatkan prediktibilitas dan mengendalikan resiko (Schwaber & Sutherland, 2013).

Ada tiga pilar dari setiap kontrol proses empiris yaitu:

1. Transparansi

Transaparansi merupakan proses peninjauan dari penanggung jawab.

Transaparansi memiliki standar definisi yang sama.

2. Inspeksi

Sebuah cara untuk meninjau kembali liniwaktu scrum yang telah dibuat beserta perkembangannya agar perubahan dapat dikendalikan. Peninjauan ini diperlukan waktu rutin oleh peninjau yang kompeten saat pekerjaan sedang dalam proses.

3. Adaptasi

Apabila proses pengerjaan mengalami deviasi yang diluar dugaan yang hasilnya tidak dapat diterima, maka proses scrum perlu diatur ulang.

Pengaturan ulang harus dibuat sesegera mungkin untuk menghindari deviasi yang sangat jauh.

2.8 Model Penelitian

Model penelitian untuk pengembangan aplikasi *MoLearn* ini adalah menggunakan model pengembangan *scrum*. Seperti yang telah dijelaskan pada landasar teori bahwa *scrum* adalah sebuah kerangka kerja dalam pengembangan

aplikasi. Pada buku Schwaber & Sutherland (2013) Secara formal, scrum menyediakan empat acara formal dalam kerangka kerjanya untuk inspeksi dan adaptasi. Empat cara tersebut adalah:

1. *Sprint Planning*

Sprint planning adalah pertemuan untuk menentukan *sprint* yang direncanakan secara kolaboratif oleh seluruh anggota tim scrum. Pertemuan ini akan dilakukan oleh tim pengembang, *scrum master*, dan MGMP biologi dan geografi jawa timur. *Sprint planning* ini dijadwalkan oleh MGMP sebagai *product owner* untuk melakukan peninjauan/*monitoring* dan masukan untuk *sprint planning* selanjutnya. *Sprint* akan dilakukan setiap tiga minggu sekali antara *scrum master*, pengembang, dan *product owner* dalam ruang rapat yang bergantian.

2. *Daily Scrum*

Daily scrum adalah dilakukan sesama tim dalam waktu 15 menit yang dilakukam setiap pagi hari untuk berdiskusi antara tim yang lainnya. *Daily scrum* akan dilakukan oleh peneliti untuk untuk peninjauan pengembangan aplikasi setiap harinya.

3. *Sprint Review*

Sprint review dilakukan di akhir *sprint* untuk meninjau perubahan *Product Backlog*. *Sprint review* dapat dilakukan ketika *daily sprint* telah selesai dikerjakan.

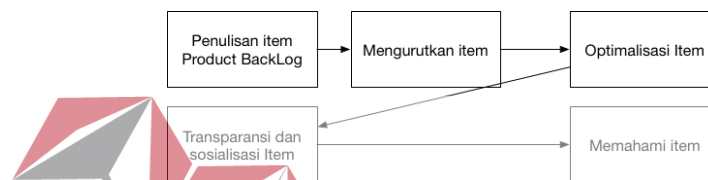
4. *Sprint Retrospective*

Sprint Retrospective adalah kesempatan bagi tim *scrum* untuk meninjau dirinya dan membuat perencanaan mengenai peningkatan yang akan dilakukan sprint berikutnya.

Pengembangan *software* menggunakan *scrum* melibatkan bermacam-macam tim. Tim *scrum* meliputi sebagai berikut :

1. *Product Owner*

Product owner bertanggung jawab disini adalah tim MGMP untuk mengelola *Product Backlog*. Gambar 2.2 adalah cakupan pengelolaan *product backlog*.



Gambar 2.2 Langkah pengelolaan *product backlog*

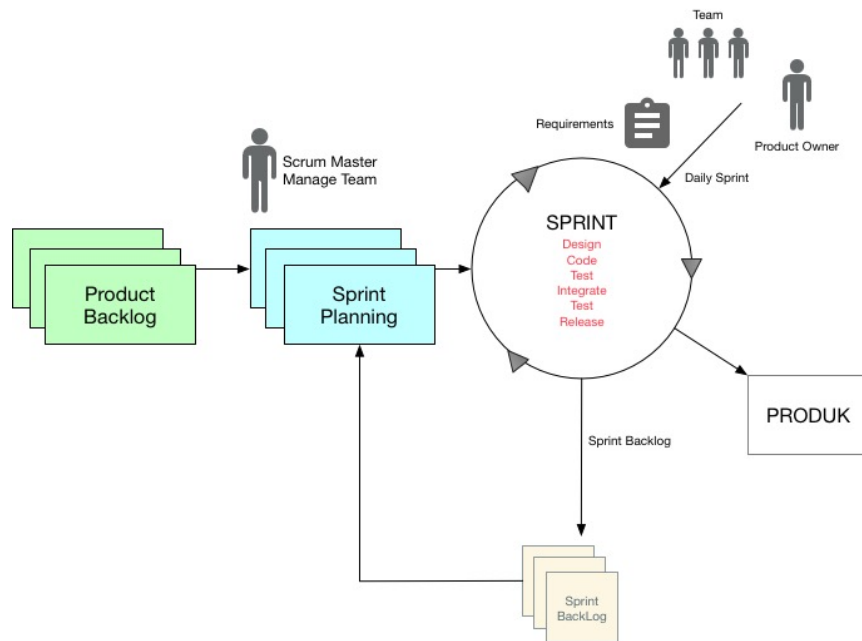
2. Tim Pengembang

Tim pengembang adalah saya sendiri sebagai peneliti sebagai pembuat aplikasi Android.

3. Scrum Master

Scrum master dalam cakupan penelitian ini adalah peneliti sendiri.

Kerangka kerja *scrum* dapat dilihat pada Gambar 2.3 di bawah ini:



Gambar 2.3 Kerangka Kerja Scrum



BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Aplikasi *MoLearn* ini telah dijelaskan pada bab sebelumnya yaitu menggunakan model pengembangan *scrum*. Pengembangan *scrum* akan dimulai dari *product backlog* kemudian menuju *sprint planning*, *daily scrum* yang di dalamnya terdapat *sprint review* dan *sprint retrospective*, kemudian *sprint* tersebut akan menghasilkan produk dalam bentuk aplikasi. Tim *scrum* yang terlibat dalam pengembangan sistem ini adalah:

1. Product Owner: Tim MGMP Biologi dan Surabaya Jawa Timur.
2. Tim Pengembang: Zayed Elfasa
3. Scrum Master: Zayed Elfasa

3.1.1 Identifikasi Masalah

Pembelajaran yang dilakukan oleh generasi digital *native* telah berkembang di era globalisasi ini. Dimana ada aplikasi yang telah diterapkan pada generasi digital *native*. Aplikasi yang telah berjalan saat ini tidak memiliki standar pembelajaran dalam pemberian materi kepada siswa. Standar pembelajaran ini sangat perlu digunakan untuk tugas, materi, dan ujian agar hal-hal tersebut dapat menjadi standar.

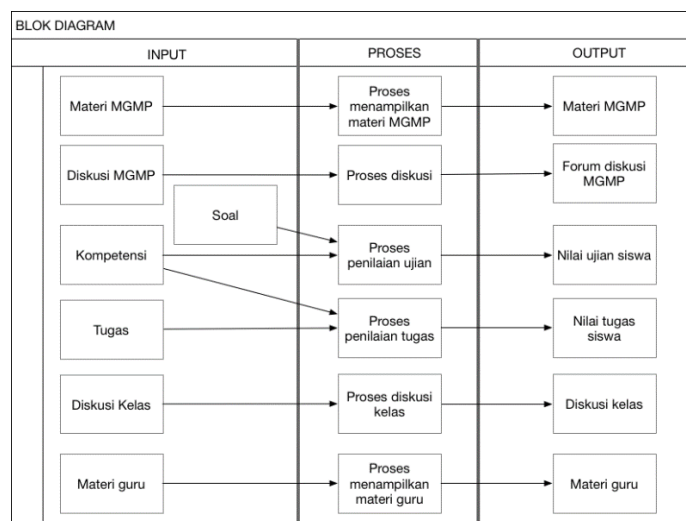
3.1.2 Analisis Kebutuhan Pengguna

Aplikasi *MoLearn* ini dirancang untuk guru dan siswa untuk saling belajar dan berkolaborasi untuk mencapai tujuan dari pembelajaran abad ini.

Aplikasi *Molearn* memiliki pengguna sebagai siswa yang dapat melihat materi dari MGMP dan materi dari guru yang mengajar, dapat melihat tugas yang diberikan oleh guru ajar, dan diberikan fasilitas untuk berkolaborasi antar siswa dan guru ajar.

Tenaga pendidik dibagi menjadi dua yaitu koordinator yang disebut dengan nama “MGMP” dan guru. Fungsi MGMP pada aplikasi *MoLearn* yaitu melihat kompetensi yang merupakan standar belajar secara nasional, mengolah standar materi yang dipublikasikan kepada guru dan siswa, melihat materi yang dipublikasikan oleh guru mata pelajaran, menampilkan ujian berdasarkan kompetensi dasar, kompetensi inti, dan indikator, dan untuk berkolaborasi antara MGMP dan guru difasilitasi dengan sebuah forum diskusi.

Tenaga pendidik yang lain yaitu guru dapat melihat materi yang diberikan oleh MGMP, melihat materi guru yang telah dipublikasikan, menampilkan soal ujian berdasarkan kompetensi dasar, kompetensi inti, dan indikator, membuat tugas untuk siswa, menilai tugas, dan untuk berkolaborasi dengan siswa akan diberikan fasilitas forum diskusi guru dan siswa.



Gambar 3.1 Blok Diagram Aplikasi *MoLearn*

Dari *block diagram* pada gambar 3.1 diatas terlihat aplikasi *MoLearn* akan memiliki tiga pengguna yang berbeda dimana di setiap pengguna memiliki proses sesuai dengan kebutuhannya. Proses-proses yang terjadi di atas dapat juga terjadi pada pengguna lainnya seperti siswa. Siswa tidak hanya mendapatkan tugas dari guru tetapi juga dapat melihat materi yang telah dibuat oleh guru dan MGMP. Untuk lebih jelas peran dari setiap pengguna dapat dilihat pada Tabel 3.1:

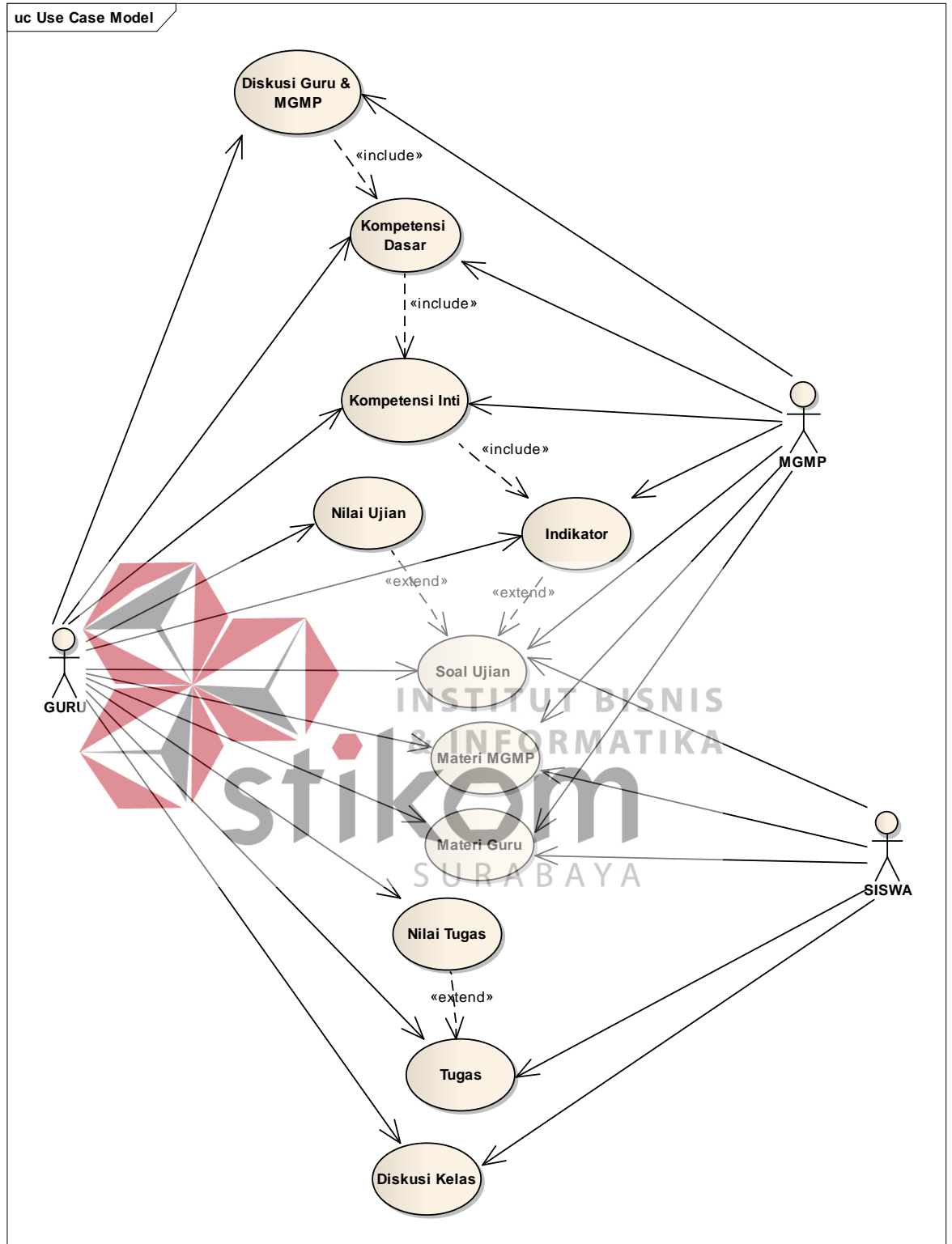
Tabel 3.1 Tabel Fitur Aplikasi *MoLearn*

No.	Pengguna	Fitur
1	MGMP	Menampilkan kompetensi dasar
2		Menampilkan kompetensi inti
3		Menampilkan indikator
4		Menampilkan soal berdasarkan kompetensi
5		Menampilkan materi MGMP
6		Menampilkan materi guru.
7		Forum diskusi (guru dan MGMP).
8	Guru	Menampilkan kompetensi dasar
9		Menampilkan kompetensi inti
10		Menampilkan indikator
11		Menampilkan soal ujian
12		Menampilkan materi MGMP
13		Menampilkan materi guru.
14		Forum diskusi (MGMP, guru, dan kelas).
15		Forum diskusi (Guru dan Siswa)
16		Menampilkan tugas (guru)
17		Menilai tugas
18		Menilai soal ujian
19	Siswa	Menampilkan materi MGMP
20		Menampilkan materi guru.
21		Menampilkan tugas (siswa)
22		Forum diskusi (kelas)
23		Menampilkan nilai dan soal ujian

3.2 Perancangan Sistem

3.2.1 Use Case Diagram

Use Case digunakan untuk melakukan identifikasi apa saja yang dapat dilakukan oleh pengguna. Di dalam *use case* terdapat siapa yang akan menjadi pengguna aplikasi (aktor) dan apa yang dilakukan oleh aktor (*use case*). Pada kasus pembuatan Aplikasi *MoLearn* berbasis Android terdapat tiga aktor yang akan menggunakan aplikasi. Ketiga aktor tersebut memiliki *use case* yang nantinya akan memiliki detail pada *flow of event* yang selanjutnya proses tersebut akan diterjemahkan dalam *Sequence Diagram*. Ketiga aktor tersebut adalah MGMP, Guru, dan Siswa. Aktor MGMP memiliki *use case* Diskusi Guru & MGMP, Kompetensi Dasar, Kompetensi Inti, Indikator, Soal Ujian, Materi MGMP, dan Materi Guru. Aktor Guru akan memiliki Diskusi Guru & MGMP, Kompetensi Dasar, Kompetensi Inti, Indikator, Nilai Ujian, Soal Ujian, Materi MGMP, Materi Guru, Nilai Tugas, Tugas, Diskusi Kelas. Aktor Siswa akan memiliki *use case* Soal Ujian, Materi MGMP, Materi Guru, Tugas, dan Diskusi Kelas. *Use case* Aplikasi *MoLearn* berbasis Android dapat dilihat pada Gambar 3.2:



Gambar 3.2 Use Case Aplikasi MoLearn berbasis Android

3.2.2 Flow Of Event

1. Flow Of Event Menampilkan Kompetensi Inti (MGMP)

Kompetensi Inti memiliki kegunaan untuk MGMP karena kompetensi inti digunakan sebagai inti untuk membuat klasifikasi materi dan soal yang ditampilkan. Kompetensi yang ditampilkan berdasarkan mata pelajaran yang dipilih ketika masuk aplikasi. Proses menampilkan kompetensi dasar untuk lebih jelasnya dapat dilihat pada Tabel 3.2:

Tabel 3.2 *Flow of Event* Menampilkan Kompetensi Dasar (MGMP)

Nama Use Case	Menampilkan Kompetensi Inti (MGMP)		
Kebutuhan terkait	Proses untuk menampilkan kompetensi inti berdasarkan mata pelajaran yang dipilih.		
Tujuan	Menangani tampilan kompetensi inti.		
Prasyarat	-		
Kondisi Akhir Sukses	Kompetensi inti tampil berdasarkan mata pelajaran yang telah dipilih		
Kondisi Akhir Gagal	Kompetensi inti tidak tampil karena koneksi internet tidak ada.		
Aktor Utama	MGMP		
Aktor Sekunder	-		
Pemicu	Aktor utama memiliki menu kompetensi dasar		
Alur Utama	Langkah	Aktor	Aplikasi
	1	Memilih menu kompetensi	Menampilkan halaman kompetensi
	2		Meminta data kompetensi
	3		Mengembalikan data berformat JSON.
	4		Menampilkan Kompetensi Inti dalam bentuk <i>list</i> .
	5		<i>Use Case</i> selesai.
Alur Perluasan	Langkah	Aktor	Aplikasi
	3.1		Jika Aplikasi tidak dapat mengembalikan data maka tidak ada tampilan.

Nama Use Case	Menampilkan Kompetensi Inti (MGMP)		
		Kembali ke menu utama	Menampilkan menu utama
		Klik/ pilih kompetensi	Kembali pada langkah 1.

2. Flow Of Event Menampilkan Kompetensi Dasar (MGMP)

Kompetensi dasar adalah detil dari kompetensi inti yang memiliki kebutuhan sama seperti kompetensi dasar yaitu sebagai klasifikasi materi dan soal yang ditampilkan. Kompetensi dasar akan ditampilkan berdasarkan kompetensi inti yang ditampilkan sebelumnya. Proses menampilkan lebih jelasnya dapat dilihat pada Tabel 3.3:

Tabel 3.3 *Flow of Event* Menampilkan Kompetensi Dasar (MGMP)

Nama Use Case	Menampilkan Kompetensi Dasar (MGMP)		
Kebutuhan terkait	Proses untuk menampilkan kompetensi dasar berdasarkan mata pelajaran yang dipilih.		
Tujuan	Menangani tampilan kompetensi dasar.		
Prasyarat	Aktor telah memilih salah satu kompetensi dasar.		
Kondisi Akhir Sukses	Kompetensi dasar tampil berdasarkan mata pelajaran yang telah dipilih.		
Kondisi Akhir Gagal	Kompetensi dasar tidak tampil karena koneksi internet tidak ada.		
Aktor Utama	MGMP		
Aktor Sekunder	-		
Pemicu	Aktor memilih salah satu kompetensi dasar.		
Alur Utama	Langkah	Aktor	Aplikasi
	1	Memilih salah satu kompetensi dasar yang ada di <i>ListView</i> .	Menuju halaman kompetensi inti
	2		Meminta data kompetensi inti berdasarkan satu kompetensi dasar yang telah dipilih pada

Nama Use Case	Menampilkan Kompetensi Dasar (MGMP)		
			halaman sebelumnya.
	3		Mengembalikan data dalam format JSON.
	4		Menampilkan kompetensi inti dalam bentuk <i>list</i> .
	5		<i>Use Case</i> selesai.
Alur Perluasan	Langkah	Aktor	Aplikasi
	3.1		Jika Aplikasi tidak dapat mengembalikan data maka tidak ada tampilan.
		Kembali ke menu utama	Menampilkan menu utama.
	Klik / pilih salah satu kompetensi dasar.	Kembali pada langkah 1.	

3. *Flow Of Event* Menampilkan Indikator (MGMP)

Indikator merupakan detail dari kompetensi dasar. Indikator juga dibutuhkan sebagai klasifikasi dari materi dan tugas yang akan ditampilkan.

Indikator ditampilkan berdasarkan kompetensi yang dipilih sebelumnya. Proses menampilkan lebih jelasnya dapat dilihat pada Tabel 3.4:

Tabel 3.4 *Flow of Event* Menampilkan Indikator (MGMP)

Nama Use Case	Menampilkan Indikator (MGMP)
Kebutuhan terkait	Proses untuk menampilkan indikator berdasarkan mata pelajaran yang dipilih.
Tujuan	Menangani tampilan indikator
Prasyarat	Aktor telah memilih salah satu kompetensi inti.
Kondisi Akhir Sukses	Indikator tampil berdasarkan mata pelajaran yang telah dipilih
Kondisi Akhir Gagal	Indikator tidak tampil karena koneksi internet tidak ada.
Aktor Utama	MGMP

Nama Use Case	Menampilkan Indikator (MGMP)		
Aktor Sekunder	-		
Pemicu	Aktor memilih salah satu kompetensi dasar.		
Alur Utama	Langkah	Aktor	Aplikasi
	1	Memilih salah satu kompetensi inti yang ada di ListView.	Menuju halaman indikator
	2		Meminta data indikator berdasarkan satu kompetensi inti yang telah dipilih pada halaman sebelumnya.
	3		Mengembalikan data dalam format JSON.
	4		Menampilkan indikator dalam bentuk <i>list</i> .
	5		<i>Use Case</i> selesai.
Alur Perluasan	Langkah	Aktor	Aplikasi
	3.1	Kembali ke menu utama	Jika Aplikasi tidak dapat mengembalikan data maka tidak ada tampilan. Menampilkan menu utama.
		Klik / pilih salah satu kompetensi dasar.	Kembali pada langkah 1.

4. Flow Of Event Menampilkan Soal Berdasarkan Kompetensi (MGMP)

Soal ini adalah soal yang ditampilkan berdasarkan kompetensi yang dipilih. Pertama kompetensi dasar dipilih, kemudian kompetensi inti, kemudian indikator. Setelah semua kompetensi dipilih, soal dapat dilihat. Proses menampilkan soal berdasarkan kompetensi lebih jelasnya dapat dilihat pada Tabel 3.5:

Tabel 3.5 *Flow of Event* Menampilkan Soal Berdasarkan Kompetensi (MGMP)

Nama Use Case	Menampilkan Soal Berdasarkan Kompetensi		
Kebutuhan terkait	Proses untuk menampilkan ujian berdasarkan kompetensi dasar, kompetensi inti, dan indikator untuk pengguna sebagai MGMP dimana soal ini akan digunakan untuk standar soal ujian siswa.		
Tujuan	Menangani tampilan soal ujian yang telah disaring berdasarkan kompetensi.		
Prasyarat	Aktor telah memilih kompetensi dasar, kompetensi inti, indikator.		
Kondisi Akhir Sukses	Soal ujian sukses tampil berdasarkan kompetensi.		
Kondisi Akhir Gagal	Soal ujian tidak tampil, isi halaman soal ujian kosong.		
Aktor Utama	MGMP		
Aktor Sekunder	-		
Pemicu	Aktor memilih kompetensi kemudian klik button lihat soal.		
Alur Utama	Langkah	Aktor	Aplikasi
	1	Aktor memilih kompetensi dasar.	Menampilkan dialog kompetensi dasar.
	2	Aktor memilih kompetensi inti.	Menampilkan dialog kompetensi inti.
	3	Aktor memilih indikator.	Menampilkan indikator.
	4	Aktor menekan tombol lihat soal.	Pindah halaman baru menuju soal.
	5		Meminta soal ujian berdasarkan kompetensi di <i>server</i> .
	6		Menampilkan soal dalam bentuk <i>timeline</i> .
	7		<i>Use Case</i> selesai.
Alur Perluasan	Langkah	Aktor	Aplikasi

Nama Use Case	Menampilkan Soal Berdasarkan Kompetensi		
	4.1		Jika aplikasi tidak dapat menampilkan soal, maka halaman akan menjadi kosong.
		Aktor kembali ke halaman sebelumnya.	
		Aktor menekan tombol lihat soal.	Kembali ke langkah 4.

5. Flow Of Event Menampilkan Materi MGMP (MGMP)

Materi MGMP ini adalah materi yang ditampilkan pada aktor MGMP, yang sebelumnya telah dimasukkan oleh MGMP itu sendiri. Tentunya materi MGMP ini adalah materi yang akan ditampilkan pada guru dan siswa. Materi MGMP yang ditampilkan berdasarkan mata pelajaran yang dipilih. Proses menampilkan lebih jelasnya dapat melihat Tabel 3.6 *flow of event* menampilkan materi MGMP.

Tabel 3.6 *Flow of Event* Menampilkan Materi MGMP (MGMP)

Nama Use Case	Menampilkan Materi MGMP (MGMP)		
Kebutuhan terkait	Proses untuk menampilkan materi MGMP yang menjadi standar materi yang diberikan kepada siswa.		
Tujuan	Menangani materi MGMP dalam bentuk <i>youtube, file</i> , dan bentuk teks.		
Prasyarat	Tidak ada.		
Kondisi Akhir Sukses	Materi MGMP telah sukses ditampilkan dalam bentuk <i>youtube, file</i> , atau teks.		
Kondisi Akhir Gagal	Materi MGMP tidak sukses ditampilkan dalam bentuk <i>youtube, file</i> , atau teks karena tidak dapat mengambil data melalui internet.		
Aktor Utama	MGMP		
Aktor Sekunder	-		
Pemicu	Aktor memilih tombol materi MGMP.		
Alur Utama	Langkah	Aktor	Aplikasi

Nama Use Case	Menampilkan Materi MGMP (MGMP)		
	1	Aktor memilih tombol materi MGMP	Menuju halaman materi MGMP.
	2		Meminta data materi dari MGMP berdasarkan mata pelajaran dan kelas.
	3		Menampilkan list dari materi MGMP.
	4	Aktor memilih salah satu materi MGMP	Membuka halaman baru.
	5		Menampilkan materi MGMP yang telah dipilih oleh aktor.
	6		Use case selesai.
Alur Perluasan	Langkah	Aktor	Aplikasi
	4.1		Jika aktor memilih materi berjenis <i>youtube</i> , maka langsung akan menampilkan video <i>youtube</i> .
			Jika aktor memilih materi dalam bentuk file, maka langsung menampilkan isi dari <i>file</i> .
	Jika aktor memilih materi dalam bentuk teks, maka langsung menampilkan teks.		

6. Flow Of Event Menampilkan Materi Guru (MGMP)

Materi guru dapat ditampilkan untuk MGMP karena materi guru ini juga dibutuhkan oleh MGMP untuk monitoring jika materi guru terdapat kesalahan.

Materi guru yang ditampilkan berdasarkan mata pelajaran yang dipilih. Proses menampilkan lebih jelasnya proses ini dapat dilihat pada Tabel 3.7:

Tabel 3.7 *Flow of Event* Menampilkan Materi Guru (MGMP)

Nama Use Case	Menampilkan Materi Guru (MGMP)		
Kebutuhan terkait	Proses untuk menampilkan materi guru yang akan diberikan kepada siswa di kelas masing-masing.		
Tujuan	Menangani materi guru dalam bentuk <i>youtube</i> , <i>file</i> , dan bentuk teks.		
Prasyarat	Tidak ada.		
Kondisi Akhir Sukses	Materi guru telah sukses ditampilkan dalam bentuk <i>youtube</i> , <i>file</i> , atau teks.		
Kondisi Akhir Gagal	Materi guru tidak sukses ditampilkan dalam bentuk <i>youtube</i> , <i>file</i> , atau teks karena tidak dapat mengambil data melalui internet.		
Aktor Utama	Guru		
Aktor Sekunder	-		
Pemicu	Aktor memilih tombol materi MGMP.		
Alur Utama	Langkah	Aktor	Aplikasi
	1	Aktor memilih tombol materi guru	Menuju halaman materi guru
	2		Meminta data materi dari guru berdasarkan mata pelajaran dan kelas.
	3		Menampilkan list dari materi guru
	4	Aktor memilih salah satu materi guru.	Membuka halaman baru.
	5		Menampilkan materi guru yang telah dipilih oleh aktor.
	6		<i>Use case</i> selesai.
Alur Perluasan	Langkah	Aktor	Aplikasi

Nama Use Case	Menampilkan Materi Guru (MGMP)		
	4.1		Jika aktor memilih materi berjenis <i>youtube</i> , maka langsung akan menampilkan video <i>youtube</i> .
	4.2		Jika aktor memilih materi dalam bentuk file, maka langsung menampilkan isi dari <i>file</i> .
	4.3		Jika aktor memilih materi dalam bentuk teks, maka langsung menampilkan teks.

7. Flow Of Event Forum Diskusi Guru dan MGMP (MGMP)

Forum diskusi ini digunakan oleh guru dan MGMP untuk berdiskusi pada semua masalah dan solusi yang dihadapi selama proses belajar mengajar. Proses diskusi guru dan MGMP lebih jelasnya dapat dilihat pada Tabel 3.8:

Tabel 3.8 *Flow of Event* Forum Diskusi Guru dan MGMP (MGMP)

Nama Use Case	Forum Diskusi Guru dan MGMP (MGMP)		
Kebutuhan terkait	Proses untuk melakukan diskusi oleh tim guru dan tim MGMP pada sebuah halaman forum.		
Tujuan	Menangani diskusi antara guru dan MGMP apabila terjadi kesalah-pahaman pada materi-materi yang diberikan kepada siswa.		
Prasyarat	Tidak ada.		
Kondisi Akhir Sukses	Forum diskusi sukses dijalankan.		
Kondisi Akhir Gagal	Forum diskusi tidak dapat menampilkan teks-teks diskusi sebelumnya dan sesudahnya.		
Aktor Utama	MGMP		
Aktor Sekunder	Guru		
Pemicu	Aktor memilih forum diskusi guru dan MGMP		
Alur Utama	Langkah	Aktor	Aplikasi

Nama Use Case	Forum Diskusi Guru dan MGMP (MGMP)		
	1	Aktor menekan tombol forum diskusi guru dan MGMP	Menuju form diskusi.
	2		Meminta isi diskusi - diskusi sebelumnya dalam bentuk JSON.
	3		Menampilkan diskusi-diskusi sebelumnya dari seluruh guru yang tergabung dalam forum.
	4	Aktor mengetik teks.	Menampilkan teks yang telah di-input-kan oleh aktor.
	5	Aktor mengirim tombol "kirim".	Mengirim data ke server
	6		Menampilkan teks yang telah dikirim oleh aktor.
	7		Use Case selesai.
Alur Perluasan	Langkah	Aktor	Aplikasi
	4.1		Jika data tidak dapat diambil dari internet, maka halaman tidak akan menampilkan apa-apa.
		Aktor kembali ke halaman sebelumnya.	Menuju langkah 1.

8. Flow Of Event Menampilkan Soal Ujian (Guru)

Soal ujian ini adalah soal yang ditampilkan beserta jawabannya yang akan ditampilkan pada siswa kelas yang diajar oleh guru. Soal ujian yang akan ditampilkan pada siswa hanya soalnya saja tanpa jawaban. Proses menampilkan ujian lebih jelasnya dapat dilihat pada Tabel 3.9:

Tabel 3.9 *Flow of Event* Menampilkan Soal Ujian (MGMP)

Nama Use Case	Menampilkan Soal Ujian (MGMP)		
Kebutuhan terkait	Proses untuk menampilkan soal ujian yang telah dibuat oleh guru.		
Tujuan	Menangani tampilan soal ujian.		
Prasyarat	Guru telah membuat ujian.		
Kondisi Akhir Sukses	Soal ujian dapat ditampilkan dengan baik.		
Kondisi Akhir Gagal	Soal ujian tidak dapat ditampilkan karena tidak dapat mengambil data dari internet.		
Aktor Utama	Guru.		
Aktor Sekunder	Tidak ada.		
Pemicu	Guru memilih tombol melihat soal ujian.		
Alur Utama	Langkah	Aktor	Aplikasi
	1	Aktor memilih menu soal ujian.	Menuju halaman ujian
	2		Menampilkan ujian-ujian yang telah dibuat oleh guru.
	3	Memilih ujian yang telah dibuat sebelumnya.	Menuju halaman soal.
	4		Mengambil data dari server.
	5		Menampilkan soal-soal dari ujian yang telah dipilih sebelumnya.
	6		<i>Use case</i> selesai.
Alur Perluasan	Langkah	Aktor	Aplikasi
	4.1	Aktor kembali ke halaman sebelumnya.	Jika soal tidak dapat ditampilkan, halaman soal ujian akan kosong. Menuju langkah 3.

9. Flow Of Event Menampilkan Kompetensi Inti (GURU)

Kompetensi inti dibutuhkan oleh guru sebagai standar untuk ditampilkan pada tugas dan soal yang sebelumnya telah dibuat. Kompetensi inti yang ditampilkan berdasarkan mata pelajaran yang dipilih ketika masuk aplikasi. Proses menampilkan kompetensi inti lebih jelasnya dapat dilihat pada Tabel 3.10:

Tabel 3.10 *Flow of Event* Menampilkan Kompetensi Inti (GURU)

Nama Use Case	Menampilkan Kompetensi Inti (GURU)		
Kebutuhan terkait	Proses untuk menampilkan kompetensi inti berdasarkan mata pelajaran yang dipilih.		
Tujuan	Menangani tampilan kompetensi inti.		
Prasyarat	-		
Kondisi Akhir Sukses	Kompetensi dasar tampil berdasarkan mata pelajaran yang telah dipilih.		
Kondisi Akhir Gagal	Kompetensi inti tidak tampil karena koneksi internet tidak ada.		
Aktor Utama	Guru		
Aktor Sekunder	-		
Pemicu	Aktor utama memiliki menu kompetensi inti.		
Alur Utama	Langkah	Aktor	Aplikasi
	1	Memilih menu kompetensi	Menampilkan halaman kompetensi
	2		Meminta data kompetensi
	3		Mengembalikan data berformat JSON.
	4		Menampilkan Kompetensi Inti dalam bentuk <i>list</i> .
	5		<i>Use Case</i> selesai.
Alur Perluasan	Langkah	Aktor	Aplikasi
	3.1		Jika aplikasi tidak dapat mengembalikan data maka tidak ada tampilan.

Nama Use Case	Menampilkan Kompetensi Inti (GURU)		
		Kembali ke menu utama	Menampilkan menu utama
		Klik/ pilih kompetensi	Kembali pada langkah 1.

10. Flow Of Event Menampilkan Kompetensi Dasar (Guru)

Kompetensi dasar dibutuhkan oleh guru sebagai standar yang ditampilkan pada soal, tugas, dan Materi Guru yang sebelumnya telah dibuat. Kompetensi dasar adalah detil dari kompetensi inti. Kompetensi dasar ditampilkan berdasarkan kompetensi inti yang dipilih sebelumnya. Proses menampilkan lebih jelasnya dapat dilihat pada Tabel 3.11:

Tabel 3.11 *Flow of Event* Menampilkan Kompetensi dasar (GURU)

Nama Use Case	Menampilkan Kompetensi Dasar (GURU)		
Kebutuhan terkait	Proses untuk menampilkan kompetensi dasar berdasarkan mata pelajaran yang dipilih.		
Tujuan	Menangani tampilan kompetensi inti.		
Prasyarat	Aktor telah memilih salah satu kompetensi inti.		
Kondisi Akhir Sukses	Kompetensi dasar tampil berdasarkan mata pelajaran yang telah dipilih		
Kondisi Akhir Gagal	Kompetensi dasar tidak tampil karena koneksi internet tidak ada.		
Aktor Utama	Guru		
Aktor Sekunder	-		
Pemicu	Aktor memilih salah satu kompetensi inti.		
Alur Utama	Langkah	Aktor	Aplikasi
	1	Memilih salah satu kompetensi inti yang ada di ListView.	Menuju halaman kompetensi dasar
	2		Meminta data kompetensi dasar berdasarkan satu kompetensi inti yang telah dipilih

Nama Use Case	Menampilkan Kompetensi Dasar (GURU)		
			pada halaman sebelumnya.
	3		Mengembalikan data dalam format JSON.
	4		Menampilkan kompetensi dasar dalam bentuk <i>list</i> .
	5		<i>Use Case</i> selesai.
Alur Perluasan	Langkah	Aktor	Aplikasi
	3.1		Jika Aplikasi tidak dapat mengembalikan data maka tidak ada tampilan.
		Kembali ke menu utama	Menampilkan <i>menu</i> utama.
	Klik / pilih salah satu kompetensi dasar.	Kembali pada langkah 1.	

11. Flow Of Event Menampilkan Indikator (Guru)

Indikator dibutuhkan sebagai standar untuk soal, tugas, dan materi guru. Indikator adalah detil dari kompetensi dasar. Indikator yang ditampilkan berdasarkan kompetensi dasar yang dipilih. Proses menampilkan indikator lebih jelasnya dapat dilihat pada Tabel 3.12:

Tabel 3.12 *Flow of Event* Menampilkan Indikator (GURU)

Nama Use Case	Menampilkan Indikator (GURU)
Kebutuhan terkait	Proses untuk menampilkan indikator berdasarkan mata pelajaran yang dipilih.
Tujuan	Menangani tampilan indikator
Prasyarat	Aktor telah memilih salah satu kompetensi dasar.
Kondisi Akhir Sukses	Indikator tampil berdasarkan mata pelajaran yang telah dipilih

Nama Use Case	Menampilkan Indikator (GURU)		
Kondisi Akhir Gagal	Indikator tidak tampil karena koneksi internet tidak ada.		
Aktor Utama	Guru		
Aktor Sekunder	-		
Pemicu	Aktor memilih salah satu kompetensi dasar.		
Alur Utama	Langkah	Aktor	Aplikasi
	1	Memilih salah satu kompetensi dasar yang ada di <i>ListView</i> .	Menuju halaman indikator
	2		Meminta data indikator berdasarkan satu kompetensi dasar yang telah dipilih pada halaman sebelumnya.
	3		Mengembalikan data dalam format JSON.
	4		Menampilkan indikator dalam bentuk <i>list</i> .
	5		<i>Use Case</i> selesai.
Alur Perluasan	Langkah	Aktor	Aplikasi
	3.1		Jika Aplikasi tidak dapat mengembalikan data maka tidak ada tampilan.
		Kembali ke menu utama	Menampilkan <i>menu</i> utama.
	Klik / pilih salah satu kompetensi dasar.	Kembali pada langkah 1.	

12. Flow Of Event Menampilkan Materi MGMP (Guru)

Materi MGMP yang ditampilkan pada guru adalah materi untuk para siswa di kelas ajar guru. Materi ini adalah materi standar dari MGMP yang digunakan oleh guru sebagai tambahan bahan ajar. Materi MGMP yang ditampilkan

berdasarkan mata pelajaran yang telah dipilih ketika masuk aplikasi. Proses menampilkan materi MGMP lebih jelasnya dapat dilihat pada Tabel 3.13:

Tabel 3.13 *Flow of Event* Menampilkan Materi MGMP (GURU)

Nama Use Case	Menampilkan Materi MGMP (GURU)		
Kebutuhan terkait	Proses untuk menampilkan materi MGMP yang menjadi standar materi yang diberikan kepada siswa.		
Tujuan	Menangani materi MGMP dalam bentuk <i>youtube</i> , <i>file</i> , dan bentuk teks.		
Prasyarat	Tidak ada.		
Kondisi Akhir Sukses	Materi MGMP telah sukses ditampilkan dalam bentuk <i>youtube</i> , <i>file</i> , atau teks.		
Kondisi Akhir Gagal	Materi MGMP tidak sukses ditampilkan dalam bentuk <i>youtube</i> , <i>file</i> , atau teks karena tidak dapat mengambil data melalui internet.		
Aktor Utama	Guru		
Aktor Sekunder	-		
Pemicu	Aktor memilih tombol materi MGMP.		
Alur Utama	Langkah	Aktor	Aplikasi
	1	Aktor memilih tombol materi MGMP	Menuju halaman materi MGMP.
	2		Meminta data materi dari MGMP berdasarkan mata pelajaran dan kelas.
	3		Menampilkan list dari materi MGMP.
	4	Aktor memilih salah satu materi MGMP	Membuka halaman baru.
	5		Menampilkan materi MGMP yang telah dipilih oleh aktor.
	6		<i>Use case</i> selesai.
Alur Perluasan	Langkah	Aktor	Aplikasi

Nama Use Case	Menampilkan Materi MGMP (GURU)		
	4.1		Jika aktor memilih materi berjenis <i>youtube</i> , maka langsung akan menampilkan video <i>youtube</i> .
			Jika aktor memilih materi dalam bentuk file, maka langsung menampilkan isi dari <i>file</i> .
			Jika aktor memilih materi dalam bentuk teks, maka langsung menampilkan teks.

13. Flow Of Event Menampilkan Materi Guru (Guru)

Materi guru ini adalah materi yang akan ditampilkan kepada siswa pada sebuah kelas pembelajaran. Materi guru yang ditampilkan berdasarkan mata pelajaran yang telah dipilih ketika masuk aplikasi dan menampilkan semua materi guru milik guru itu sendiri. Proses menampilkan materi guru lebih jelasnya dapat dilihat pada pada Tabel 3.14:

Tabel 3.14 *Flow of Event* Menampilkan Materi Guru (GURU)

Nama Use Case	Menampilkan Materi Guru (GURU)
Kebutuhan terkait	Proses untuk menampilkan materi guru yang akan diberikan kepada siswa di kelas masing-masing.
Tujuan	Menangani materi guru dalam bentuk <i>youtube</i> , <i>file</i> , dan bentuk teks.
Prasyarat	Tidak ada.
Kondisi Akhir Sukses	Materi guru telah sukses ditampilkan dalam bentuk <i>youtube</i> , <i>file</i> , atau teks.
Kondisi Akhir Gagal	Materi guru tidak sukses ditampilkan dalam bentuk <i>youtube</i> , <i>file</i> , atau teks karena tidak dapat mengambil data melalui internet.

Nama Use Case	Menampilkan Materi Guru (GURU)		
Aktor Utama	MGMP		
Aktor Sekunder	-		
Pemicu	Aktor memilih tombol materi MGMP.		
Alur Utama	Langkah	Aktor	Aplikasi
	1	Aktor memilih tombol materi guru	Menuju halaman materi guru
	2		Meminta data materi dari guru berdasarkan mata pelajaran dan kelas.
	3		Menampilkan list dari materi guru
	4	Aktor memilih salah satu materi guru.	Membuka halaman baru.
	5		Menampilkan materi guru yang telah dipilih oleh aktor.
	6		Use case selesai.
Alur Perluasan	Langkah	Aktor	Aplikasi
	4.1		Jika aktor memilih materi berjenis <i>youtube</i> , maka langsung akan menampilkan video <i>youtube</i> .
	4.2		Jika aktor memilih materi dalam bentuk file, maka langsung menampilkan isi dari <i>file</i> .
	4.3		Jika aktor memilih materi dalam bentuk teks, maka langsung menampilkan teks.

14. Flow Of Event Forum Diskusi Guru dan MGMP (Guru)

Diskusi ini akan digunakan oleh guru untuk berdiskusi secara *online* bersama MGMP lainnya melalui pesan teks. Proses diskusi lebih jelasnya dapat dilihat pada Tabel 3.15:

Tabel 3.15 *Flow of Event* Forum Diskusi Guru dan MGMP (Guru)

Nama Use Case	Forum Diskusi Guru dan MGMP (GURU)		
Kebutuhan terkait	Proses untuk melakukan diskusi oleh tim guru dan tim MGMP pada sebuah halaman forum.		
Tujuan	Menangani diskusi antara guru dan MGMP apabila terjadi kesalah-pahaman pada materi-materi yang diberikan kepada siswa.		
Prasyarat	Tidak ada.		
Kondisi Akhir Sukses	Forum diskusi sukses dijalankan.		
Kondisi Akhir Gagal	Forum diskusi tidak dapat menampilkan teks-teks diskusi sebelumnya dan sesudahnya.		
Aktor Utama	Guru		
Aktor Sekunder	MGMP		
Pemicu	Aktor memilih forum diskusi guru dan MGMP		
Alur Utama	Langkah	Aktor	Aplikasi
	1	Aktor menekan tombol forum diskusi guru dan MGMP	Menuju form diskusi.
	2		Meminta isi diskusi - diskusi sebelumnya dalam bentuk JSON.
	3		Menampilkan diskusi-diskusi sebelumnya dari seluruh guru yang tergabung dalam forum.
	4	Aktor mengetik teks.	Menampilkan teks yang telah di-inputkan oleh aktor.
	5	Aktor mengirim tombol	Mengirim data ke server

Nama Use Case	Forum Diskusi Guru dan MGMP (GURU)		
		"kirim".	
	6		Menampilkan teks yang telah dikirim oleh aktor.
	7		<i>Use Case</i> selesai.
Alur Perluasan	Langkah	Aktor	Aplikasi
	4.1		Jika data tidak dapat diambil dari internet, maka halaman tidak akan menampilkan apa-apa.
		Aktor kembali ke halaman sebelumnya.	Menuju langkah 1.

15. *Flow Of Event* Forum Diskusi Kelas (Guru)

Diskusi ini digunakan oleh guru untuk berinteraksi dengan siswa secara *online* melalui pesan teks. Proses diskusi lebih jelasnya dapat dilihat pada Tabel 3.16:

Tabel 3.16 *Flow of Event* Forum Diskusi Kelas (GURU)

Nama Use Case	Forum Diskusi Kelas (GURU)		
Kebutuhan terkait	Proses untuk melakukan diskusi oleh guru dan siswa pada sebuah halaman forum.		
Tujuan	Menangani interaksi antara guru dan siswa di kelas apabila kelas telah selesai tatap muka dalam bentuk pesan teks.		
Prasyarat	Tidak ada.		
Kondisi Akhir Sukses	Forum diskusi sukses dijalankan.		
Kondisi Akhir Gagal	Forum diskusi tidak dapat menampilkan teks-teks diskusi sebelumnya dan sesudahnya.		
Aktor Utama	Siswa		
Aktor Sekunder	Guru		
Pemicu	Aktor memilih diskusi kelas.		
Alur Utama	Langkah	Aktor	Aplikasi

Nama Use Case	Forum Diskusi Kelas (GURU)		
	1	Aktor memilih diskusi kelas.	Menuju forum diskusi.
	2		Meminta isi diskusi - diskusi sebelumnya dalam bentuk JSON dari server.
	3		Menampilkan diskusi-diskusi sebelumnya dari seluruh guru yang tergabung dalam forum.
	4	Aktor mengetik teks.	Menampilkan teks yang telah di-inputkan oleh aktor.
	5	Aktor mengirim tombol "kirim".	Mengirim data ke server
	6		Menampilkan teks yang telah dikirim oleh aktor.
	7		Use Case selesai.
Alur Perluasan	Langkah	Aktor	Aplikasi
	4.1		Jika data tidak dapat diambil dari internet, maka halaman tidak akan menampilkan apa-apa.
		Aktor kembali ke halaman sebelumnya.	Menuju langkah 1.

16. Flow Of Event Menampilkan Tugas (Guru)

Tugas yang ditampilkan adalah tugas yang diberikan kepada siswa sesuai dengan kelas pembelajaran oleh guru yang bersangkutan. Proses menampilkan tugas lebih jelasnya dapat dilihat pada Tabel 3.17:

Tabel 3.17 *Flow of Event* Menampilkan Tugas (GURU)

Nama Use Case	Menampilkan Tugas (GURU)		
Kebutuhan terkait	Proses untuk menampilkan tugas dari guru yang diberikan kepada siswa.		
Tujuan	Menangani tampilan tugas yang diberikan kepada siswa.		
Prasyarat	Guru telah membuat tugas.		
Kondisi Akhir Sukses	Tugas yang telah dibuat oleh guru dapat ditampilkan.		
Kondisi Akhir Gagal	Tugas tidak dapat ditampilkan karena tidak dapat diambil dari server.		
Aktor Utama	Guru.		
Aktor Sekunder	Tidak ada.		
Pemicu	Aktor telah memilih menu tugas.		
Alur Utama	Langkah	Aktor	Aplikasi
	1	Memilih menu tugas.	Menuju halaman tugas.
	2	Memilih kelas ajar.	Menampilkan kelas ajar yang diambil oleh guru.
	3		Mengambil tugas berdasarkan kelas ajar.
	4		Mengembalikan data tugas.
	5		Menampilkan daftar tugas yang pernah diberikan kepada siswa.
	6		<i>Use case</i> selesai.
Alur Perluasan	Langkah	Aktor	Aplikasi
	2.1		Jika tugas tidak dapat tampil dalam bentuk daftar, maka tampilan akan kosong.
		Kembali ke halaman yang sebelumnya.	Menuju langkah 1.

17. *Flow Of Event* Menilai Tugas (Guru)

Tugas yang telah dikumpulkan oleh siswa akan dinilai oleh guru yang mengajar kelas tersebut. Tugas ini akan diunduh terlebih dahulu kemudian dapat

dinilai setelah melihat hasil dari pekerjaan masing-masing siswa. Proses menilai tugas lebih jelasnya dapat dilihat pada Tabel 3.18:

Tabel 3.18 *Flow of Event* Menilai Tugas (Guru)

Nama Use Case	Menilai Tugas (Guru)		
Kebutuhan terkait	Proses untuk melakukan penilaian setiap tugas yang telah dikumpulkan oleh siswa.		
Tujuan	Menangani penilaian tugas yang telah diunggah oleh siswa.		
Prasyarat	Siswa telah mengumpulkan tugas.		
Kondisi Akhir Sukses	Nilai tugas telah tersimpan.		
Kondisi Akhir Gagal	Nilai tugas tidak dapat tersimpan dan tidak dapat langsung menampilkan tugas.		
Aktor Utama	Guru.		
Aktor Sekunder	Tidak ada.		
Pemicu	Aktor memilih salah satu tugas siswa.		
Alur Utama	Langkah	Aktor	Aplikasi
	1	Memilih tugas siswa yang akan dinilai.	Menuju halaman penilaian.
	2	Mengisi nilai.	Menampilkan isi nilai.
	3	Klik tombol simpan.	Mengirim paramter ke server.
	4		Memberikan pesan kembali.
	5		Menampilkan pesan telah tersimpan.
	6		Kembali ke halaman sebelumnya
	7		<i>Use case</i> selesai.
Alur Perluasan	Langkah	Aktor	Aplikasi
	4.1		Jika nilai tidak tersimpan maka menuju langkah 3.

18. *Flow Of Event* Menilai Soal Ujian (Guru)

Ujian yang telah dikerjakan oleh siswa dapat dinilai secara langsung.

Ujian yang dapat dinilai adalah ujian yang soalnya bersifat esai karena tipe soal

yang lain yaitu soal pilihan ganda dan multiple answer dapat dinilai secara otomatis oleh sistem. Proses menilai ujian lebih jelasnya dapat dilihat Tabel 3.19:

Tabel 3.19 *Flow of Event* Menilai Soal Ujian (Guru)

Nama Use Case	Menilai Soal Ujian (Guru)		
Kebutuhan terkait	Proses untuk melakukan penilaian setiap soal yang telah dikerjakan oleh siswa. Proses ini dikerjakan apabila aktor menemukan soal dalam bentuk esai		
Tujuan	Menangani penilaian soal-soal ujian yang telah dikerjakan oleh siswa.		
Prasyarat	Siswa telah melaksanakan ujian <i>online</i> .		
Kondisi Akhir Sukses	Nilai soal-soal ujian telah tersimpan.		
Kondisi Akhir Gagal	Nilai soal-soal ujian tidak tersimpan.		
Aktor Utama	Guru.		
Aktor Sekunder	Tidak ada.		
Pemicu	Aktor mengisi <i>field</i> penilaian pada soal esai.		
Alur Utama	Langkah	Aktor	Aplikasi
	1	Memilih salah satu ujian yang telah dilaksanakan.	Menuju halaman soal-soal ujian.
	2	Mengisi ujian yang bertipe esai.	Menampilkan isi nilai.
	3	Klik tombol simpan.	Mengirim parameter ke server.
	4		Memberikan pesan kembali.
	5		Menampilkan pesan telah tersimpan.
	6		Kembali ke halaman sebelumnya
	7		<i>Use case</i> selesai.
Alur Perluasan	Langkah	Aktor	Aplikasi
	4.1		Jika nilai tidak tersimpan maka menuju langkah 3.

19. Flow Of Event Menampilkan Materi MGMP (Siswa)

Materi MGMP dapat ditampilkan pada siswa. Materi MGMP sifatnya adalah umum dimana siswa dari sekolah mana saja dapat mengakses materi dari MGMP berdasarkan tingkat yang sedang ditempuh oleh siswa dan mata pelajaran yang dipilih ketika masuk aplikasi. Untuk lebih jelasnya dapat dilihat pada Tabel 3.20:

Tabel 3.20 *Flow of Event* Menampilkan Materi MGMP (Siswa)

Nama Use Case	Menampilkan Tugas (Siswa)		
Kebutuhan terkait	Proses untuk menampilkan tugas dari guru yang mengajar siswa.		
Tujuan	Menangani tampilan tugas yang akan ditampilkan sebagai pengguna siswa.		
Prasyarat	Guru telah membuat tugas.		
Kondisi Akhir Sukses	Tugas yang telah dibuat oleh guru dapat ditampilkan.		
Kondisi Akhir Gagal	Tugas tidak dapat ditampilkan karena tidak dapat diambil dari server.		
Aktor Utama	Siswa		
Aktor Sekunder	Tidak ada.		
Pemicu	Aktor telah memilih menu tugas.		
Alur Utama	Langkah	Aktor	Aplikasi
	1	Memilih menu tugas.	Menuju halaman tugas.
	2	Memilih kelas ajar.	Menampilkan kelas ajar yang diajarkan oleh guru yang bersangkutan.
	3		Mengambil tugas berdasarkan kelas dan guru ajar.
	4		Mengembalikan data tugas.
	5		Menampilkan daftar tugas yang pernah diberikan kepada siswa.
	6		<i>Use case</i> selesai.
Alur Perluasan	Langkah	Aktor	Aplikasi

Nama Use Case	Menampilkan Tugas (Siswa)		
	2.1		Jika tugas tidak dapat tampil dalam bentuk daftar, maka tampilan akan kosong.
		Kembali ke halaman yang sebelumnya.	Menuju langkah 1.

20. Flow Of Event Menampilkan Materi Guru (Siswa)

Materi guru pada yang ditampilkan pada siswa adalah materi yang berasal dari guru yang mengajarnya. Materi guru yang lainnya tidak ditampilkan pada siswa yang dimana materi ini ditampilkan berdasarkan mata pelajaran yang dipilih ketika masuk aplikasi. Proses untuk menampilkan materi guru lebih jelasnya dapat dilihat pada Tabel 3.21:

Tabel 3.21 *Flow of Event* Menampilkan Materi Guru (Siswa)

Nama Use Case	Menampilkan Materi Guru (Siswa)		
Kebutuhan terkait	Proses untuk menampilkan materi guru yang akan diberikan kepada siswa di kelas masing-masing.		
Tujuan	Menangani materi guru dalam bentuk <i>youtube</i> , <i>file</i> , dan bentuk teks.		
Prasyarat	Tidak ada.		
Kondisi Akhir Sukses	Materi guru telah sukses ditampilkan dalam bentuk <i>youtube</i> , <i>file</i> , atau teks.		
Kondisi Akhir Gagal	Materi guru tidak sukses ditampilkan dalam bentuk <i>youtube</i> , <i>file</i> , atau teks karena tidak dapat mengambil data melalui internet.		
Aktor Utama	Siswa		
Aktor Sekunder	-		
Pemicu	Aktor memilih tombol materi MGMP.		
Alur Utama	Langkah	Aktor	Aplikasi
	1	Aktor memilih tombol materi guru	Menuju halaman materi guru

Nama Use Case	Menampilkan Materi Guru (Siswa)		
	2		Meminta data materi dari guru berdasarkan mata pelajaran dan kelas.
	3		Menampilkan list dari materi guru
	4	Aktor memilih salah satu materi guru.	Membuka halaman baru.
	5		Menampilkan materi guru yang telah dipilih oleh aktor.
	6		<i>Use case</i> selesai.
Alur Perluasan	Langkah	Aktor	Aplikasi
	4.1		Jika aktor memilih materi berjenis <i>youtube</i> , maka langsung akan menampilkan video <i>youtube</i> .
	4.2		Jika aktor memilih materi dalam bentuk file, maka langsung menampilkan isi dari <i>file</i> .
	4.3		Jika aktor memilih materi dalam bentuk teks, maka langsung menampilkan teks.

21. Flow Of Event Menampilkan Tugas (Siswa)

Tugas yang ditampilkan pada siswa adalah tugas berdasarkan guru yang mengajar siswa yang bersangkutan dan berdasarkan mata pelajaran ketika siswa masuk aplikasi. Proses menampilkannya dapat dilihat pada Tabel 3.22:

Tabel 3.22 *Flow of Event* Menampilkan Tugas (Siswa)

Nama Use Case	Menampilkan Tugas (Siswa)
Kebutuhan terkait	Proses untuk menampilkan tugas dari guru yang mengajar siswa.

Nama Use Case	Menampilkan Tugas (Siswa)		
Tujuan	Menangani tampilan tugas yang akan ditampilkan sebagai pengguna siswa.		
Prasyarat	Guru telah membuat tugas.		
Kondisi Akhir Sukses	Tugas yang telah dibuat oleh guru dapat ditampilkan.		
Kondisi Akhir Gagal	Tugas tidak dapat ditampilkan karena tidak dapat diambil dari server.		
Aktor Utama	Siswa		
Aktor Sekunder	Tidak ada.		
Pemicu	Aktor telah memilih menu tugas.		
Alur Utama	Langkah	Aktor	Aplikasi
	1	Memilih menu tugas.	Menuju halaman tugas.
	2	Memilih kelas ajar.	Menampilkan kelas ajar yang diajarkan oleh guru yang bersangkutan.
	3		Mengambil tugas berdasarkan kelas dan guru ajar.
	4		Mengembalikan data tugas.
	5		Menampilkan daftar tugas yang pernah diberikan kepada siswa.
	6		<i>Use case selesai.</i>
Alur Perluasan	Langkah	Aktor	Aplikasi
	2.1		Jika tugas tidak dapat tampil dalam bentuk daftar, maka tampilan akan kosong.
		Kembali ke halaman yang sebelumnya.	Menuju langkah 1.

22. Flow Of Event Forum Diskusi Kelas (Siswa)

Forum diskusi kelas dapat digunakan untuk berdiskusi antara siswa lainnya bersama guru berdasarkan kelas pembelajaran siswa secara *online* melalui sebuah pesan teks. Proses forum diskusi kelas dapat dilihat pada Tabel 3.23:

Tabel 3.23 *Flow of Event* Forum Diskusi Kelas (Siswa)

Nama Use Case	Forum Diskusi Kelas (Siswa)		
Kebutuhan terkait	Proses untuk melakukan diskusi oleh guru dan siswa pada sebuah halaman forum.		
Tujuan	Menangani interaksi antara guru dan siswa di kelas apabila kelas telah selesai tatap muka dalam bentuk pesan teks.		
Prasyarat	Tidak ada.		
Kondisi Akhir Sukses	Forum diskusi sukses dijalankan.		
Kondisi Akhir Gagal	Forum diskusi tidak dapat menampilkan teks-teks diskusi sebelumnya dan sesudahnya.		
Aktor Utama	Guru		
Aktor Sekunder	Siswa		
Pemicu	Aktor memilih diskusi kelas.		
Alur Utama	Langkah	Aktor	Aplikasi
	1	Aktor memilih diskusi kelas.	Menuju forum diskusi.
	2		Meminta isi diskusi - diskusi sebelumnya dalam bentuk JSON dari server.
	3		Menampilkan diskusi-diskusi sebelumnya dari seluruh guru yang tergabung dalam forum.
	4	Aktor mengetik teks.	Menampilkan teks yang telah di-input-kan oleh aktor.
	5	Aktor mengirim tombol "kirim".	Mengirim data ke server
	6		Menampilkan teks yang telah dikirim oleh aktor.
	7		<i>Use Case</i> selesai.
Alur Perluasan	Langkah	Aktor	Aplikasi
	4.1		Jika data tidak dapat diambil dari internet, maka halaman tidak akan menampilkan apa-apa.

Nama Use Case	Forum Diskusi Kelas (Siswa)		
		Aktor kembali ke halaman sebelumnya.	Menuju langkah 1.

23. Flow Of Event Menampilkan Nilai dan Soal Ujian (Siswa)

Siswa dapat melihat nilai dari ujian yang diberikan oleh guru. Di setiap soal terdapat nilai yang apabila guru telah selesai menilai. Proses menampilkan nilai dan soal ujian dijelaskan pada Tabel 3.24:

Tabel 3.24 *Flow of Event* Menampilkan Nilai dan Soal Ujian (Siswa)

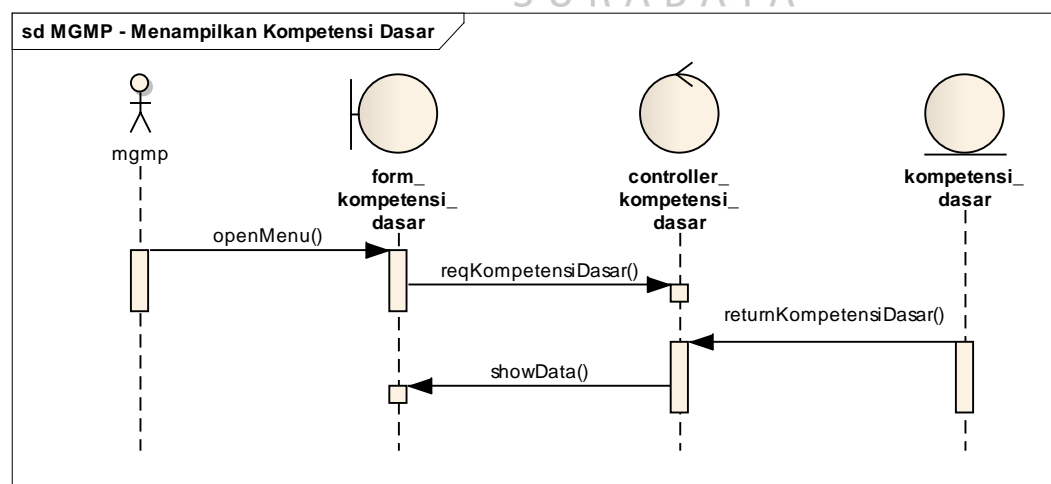
Nama Use Case	Menampilkan Nilai dan Soal Ujian (Siswa)		
Kebutuhan terkait	Proses untuk menampilkan soal ujian yang telah dibuat oleh guru.		
Tujuan	Menangani tampilan soal ujian dan nilai yang telah dinilai oleh guru (untuk esai).		
Prasyarat	Guru telah menilai ujian		
Kondisi Akhir Sukses	Soal ujian dapat ditampilkan dengan baik.		
Kondisi Akhir Gagal	Soal ujian tidak dapat ditampilkan karena tidak dapat mengambil data dari internet.		
Aktor Utama	Siswa.		
Aktor Sekunder	Tidak ada.		
Pemicu	Aktor memilih tombol melihat soal ujian.		
Alur Utama	Langkah	Aktor	Aplikasi
	1	Aktor memilih menu soal ujian.	Menuju halaman ujian
	2		Menampilkan ujian-ujian yang telah dibuat oleh guru.
	3	Memilih ujian yang telah dibuat sebelumnya.	Menuju halaman soal.
	4		Mengambil data dari server.

Nama Use Case	Menampilkan Nilai dan Soal Ujian (Siswa)		
	5		Menampilkan soal-soal dari ujian yang telah dipilih sebelumnya.
	6		<i>Use case</i> selesai.
Alur Perluasan	Langkah	Aktor	Aplikasi
	4.1		Jika soal tidak dapat ditampilkan, halaman soal ujian akan kosong.
		Aktor kembali ke halaman sebelumnya.	Menuju langkah 3.

3.2.3 Sequence Diagram

1. Sequence Diagram Menampilkan Kompetensi Dasar (MGMP)

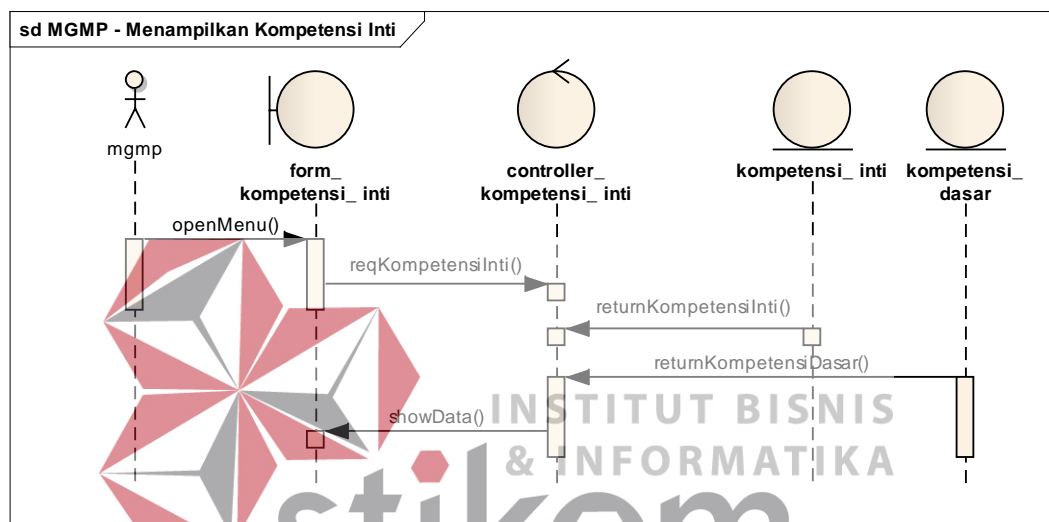
Aktor MGMP akan membuka kompetensi dasar melalui halaman menu dengan cara klik kompetensi dasar. Setelah halaman pada Android terbuka maka `controller_kompetensi_dasar` akan mengambil data kompetensi dasar yang dimasukkan ke dalam `kompetensi_dasar`. Setelah pengambilan data selesai, maka kompetensi dasar langsung ditampilkan dalam bentuk *list*.



Gambar 3.3 Sequence Diagram Menampilkan Kompetensi Dasar (MGMP)

2. Sequence Diagram Menampilkan Kompetensi Inti (MGMP)

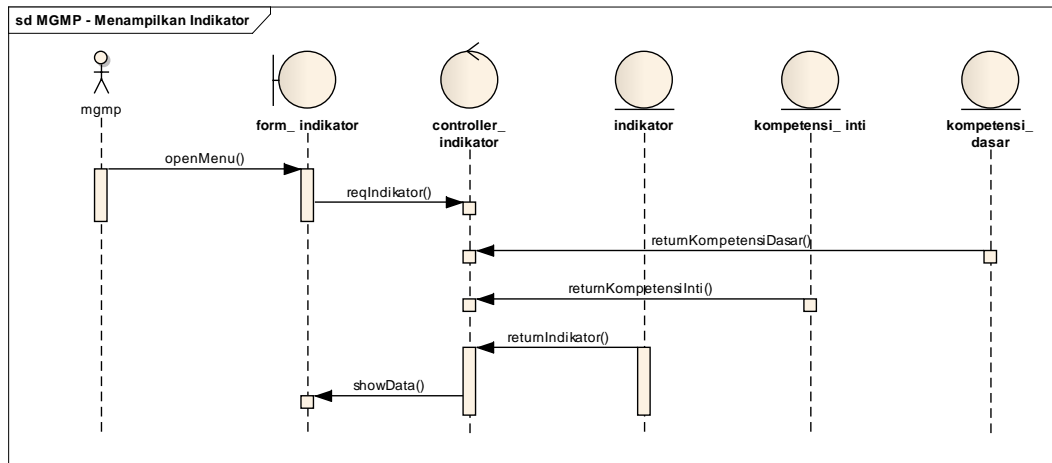
Aktor MGMP membuka halaman kompetensi dasar kemudian memilih salah satu kompetensi dasar untuk diklik. Setelah itu akan berpindah halaman kemudian `controller_kompetensi_inti` akan bekerja untuk mengambil data yang dimasukkan ke dalam `kompetensi_inti` dan `kompetensi_dasar`. Setelah diambil maka kompetensi inti ditampilkan dalam bentuk *list*.



Gambar 3.4 Sequence Diagram Menampilkan Kompetensi Inti (MGMP)

3. Sequence Diagram Menampilkan Indikator (MGMP)

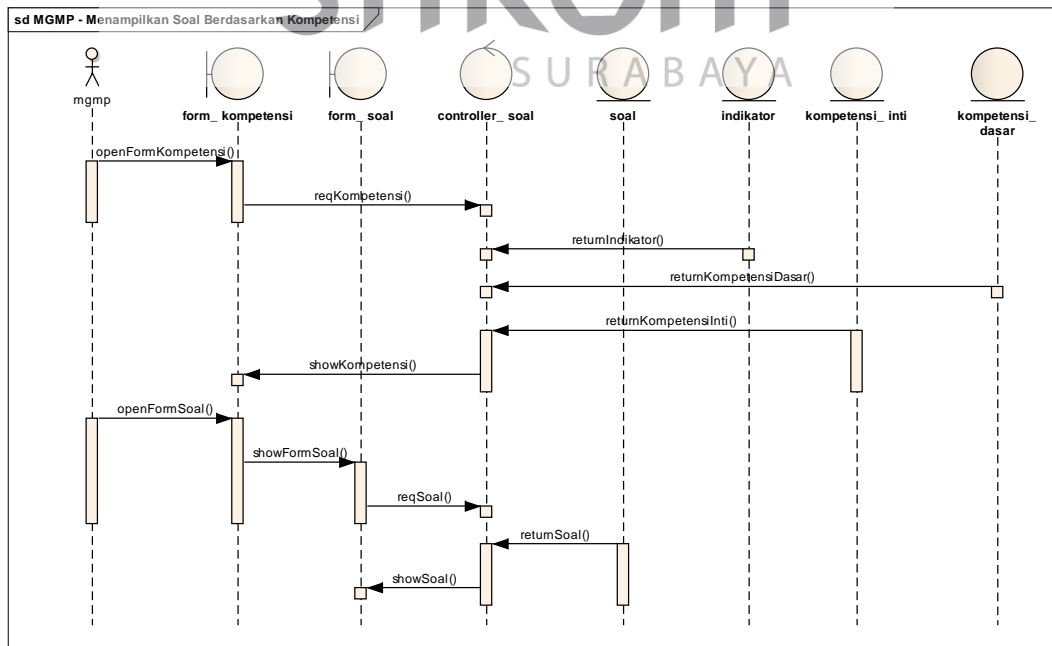
Aktor MGMP membuka *form* kompetensi inti setelah itu memilih salah satu kompetensi inti untuk diklik. Setelah itu aplikasi akan pindah halaman kemudian `controller_indikator` akan mengambil data indikator. Setelah itu data tersebut akan disimpan dalam indikator kemudian ditampilkan pada *form* indikator dalam bentuk *list*.



Gambar 3.5 *Sequence Diagram* Menampilkan Indikator (MGMP)

4. *Sequence Diagram* Menampilkan Soal Berdasarkan Kompetensi (MGMP)

Aktor MGMP membuka form_kompetensi kemudian controller_soal meminta data untuk kompetensi dasar, kompetensi inti, dan indikator yang dimasukkan ke dalam indikator, kompetensi_inti, dan kompetensi_dasar. Setelah itu kompetensi ditampilkan. MGMP memilih kompetensi lalu aplikasi akan menuju form_soal untuk menampilkan soal.

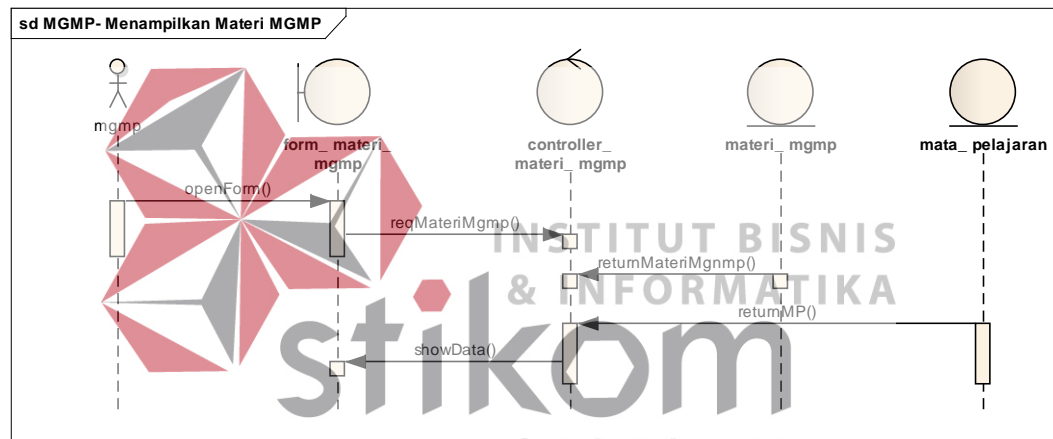


Gambar 3.6 *Sequence Diagram* Menampilkan Soal Berdasarkan Kompetensi (MGMP)

controller_soal memanggil soal yang kompetensinya telah dipilih sebelumnya. Setelah itu disimpan dalam soal kemudian ditampilkan pada form_soal.

5. Sequence Diagram Menampilkan Materi MGMP (MGMP)

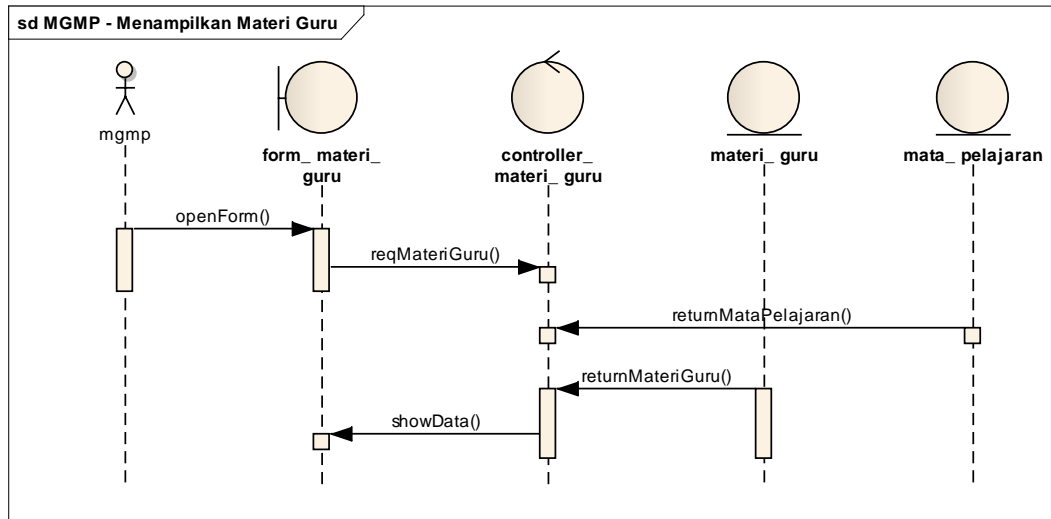
Aktor MGMP membuka form_materi_mgmp kemudian controller_materi_mgmp akan mengambil materi yang disimpan dalam materi_mgmp berdasarkan mata_pelajaran. Setelah selesai diambil, materi MGMP langsung ditampilkan pada form_materi_mgmp.



Gambar 3.7 Sequence Diagram Menampilkan Materi MGMP (MGMP)

6. Sequence Diagram Menampilkan Materi Guru (MGMP)

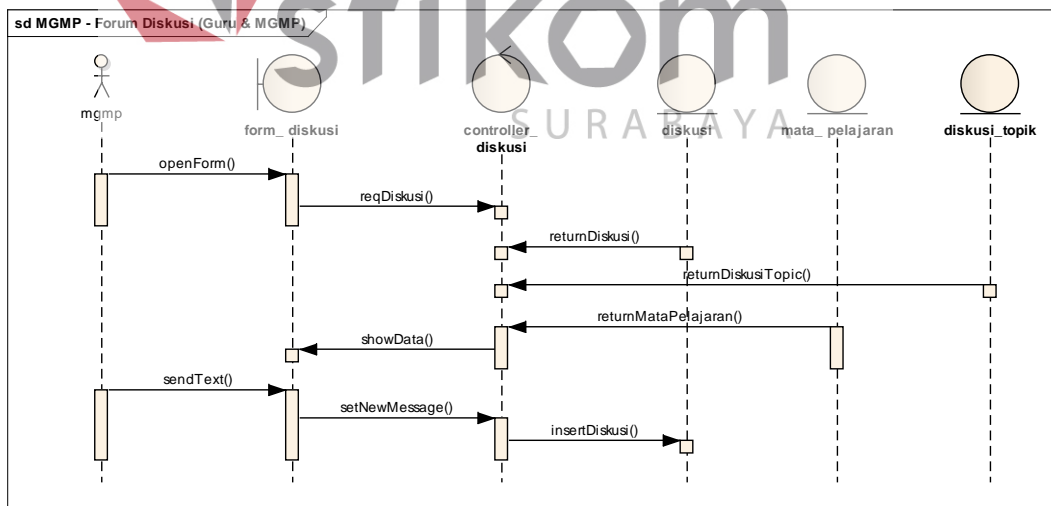
Aktor MGMP membuka form_materi_guru kemudian controller_materi_guru mengambil data melalui server. Setelah itu disimpan pada materi_guru berdasarkan mata_pelajaran. Setelah itu langsung ditampilkan pada form_materi_guru.



Gambar 3.8 *Sequence Diagram* Menampilkan Materi Guru (MGMP)

7. *Sequence Diagram* Forum Diskusi Guru dan MGMP (MGMP)

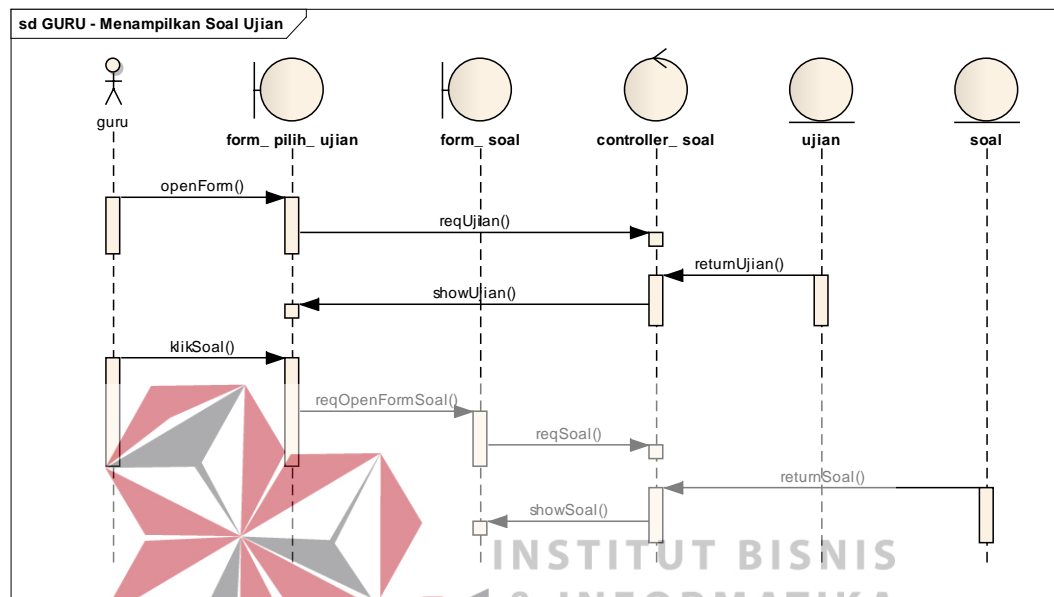
Aktor MGMP membuka form_diskusi. Setelah itu controller_diskusi akan mengambil data yang akan disimpan pada diskusi. Diskusi akan ditampilkan berdasarkan mata_pelajaran, dan diskusi topik. Diskusi kemudian ditampilkan pada form_diskusi.



Gambar 3.9 *Sequence Diagram* Forum Diskusi Guru dan MGMP (MGMP)

8. Sequence Diagram Menampilkan Soal Ujian (MGMP)

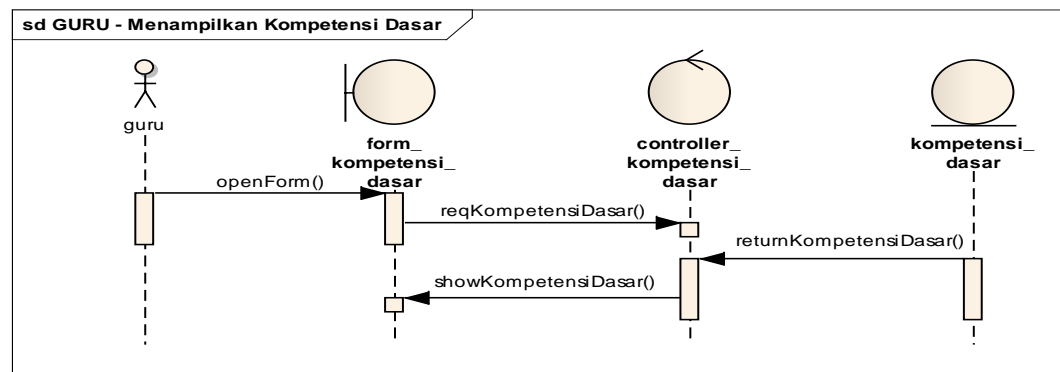
Aktor MGMP memilih ujian melalui `form_pilih_ujian` kemudian `controller_soal` akan memanggil soal yang disimpan pada ujian. Setelah itu soal pada ujian akan ditampilkan pada `form_soal`.



Gambar 3.10 Sequence Diagram Menampilkan Soal Ujian (GURU)

9. Sequence Diagram Menampilkan Kompetensi Dasar (GURU)

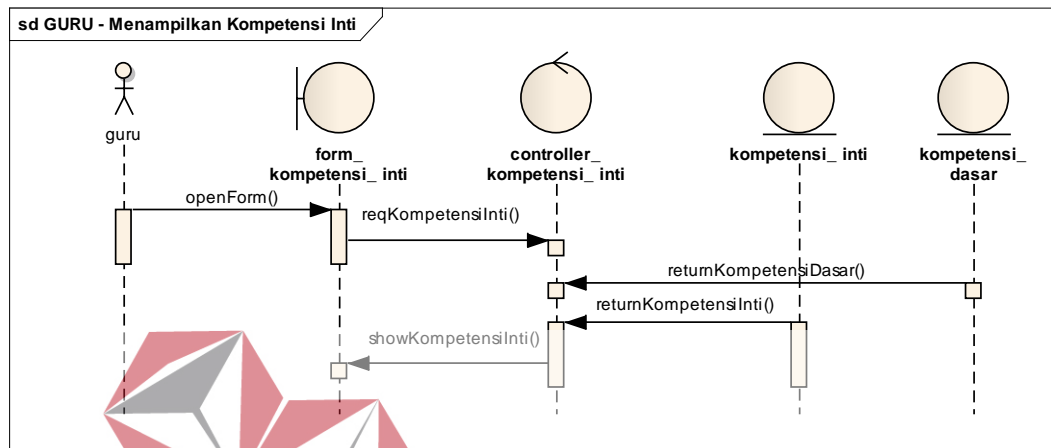
Aktor GURU membuat `form_kompetensi_dasar`. Setelah itu `controller_kompetensi_dasar` akan mengambil data yang dimasukkan ke dalam `kompetensi_dasar` kemudian ditampilkan pada `form_kompetensi_dasar`.



Gambar 3.11 Sequence Diagram Menampilkan Kompetensi Dasar (GURU)

10. Sequence Diagram Menampilkan Kompetensi Inti (GURU)

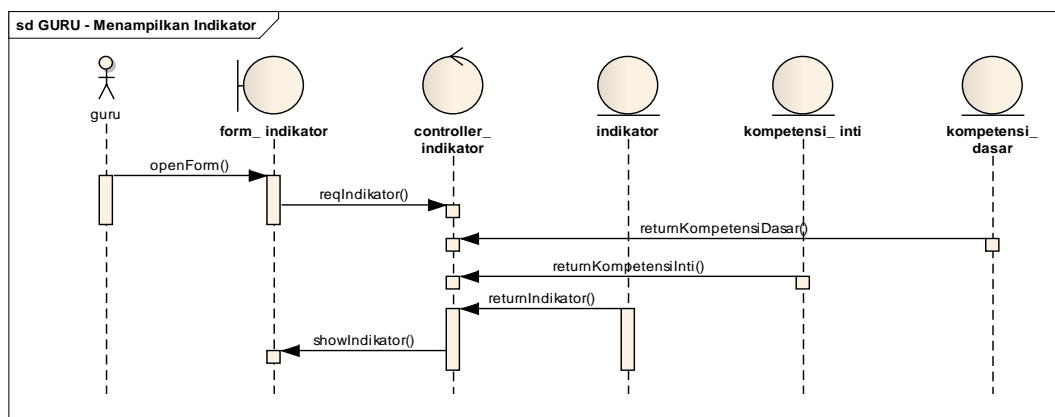
Aktor GURU klik salah satu kompetensi dasar kemudian *form* akan menuju *form_kompetensi_inti*. Setelah *form* tersebut dibukalangkah selanjutnya *controler_kompetensi_inti* akan memanggil data yang kemdian disimpan dalam *kompetensi_inti*. Setelah itu ditampilkan dalam *form*.



Gambar 3.12 Sequence Diagram Menampilkan Kompetensi Inti (GURU)

11. Sequence Diagram Menampilkan Indikator (GURU)

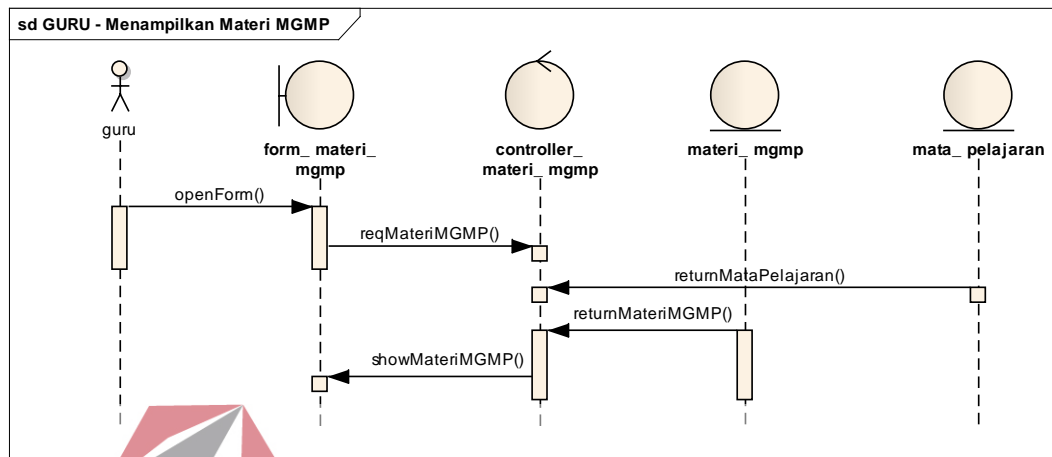
Aktor GURU klik salah satu kompetensi inti kemudian akan menuju halaman *form_indikator*. Setelah itu *controler_indikator* akan memanggil data yang dimasukkan ke dalam indikator berdasarkan *kompetensi_dasar* dan *kompetensi_inti*. Setelah disimpan langsung akan ditampilkan pada *form_indikator*.



Gambar 3.13 Sequence Diagram Menampilkan Indikator (GURU)

12. Sequence Diagram Menampilkan Materi MGMP (Guru)

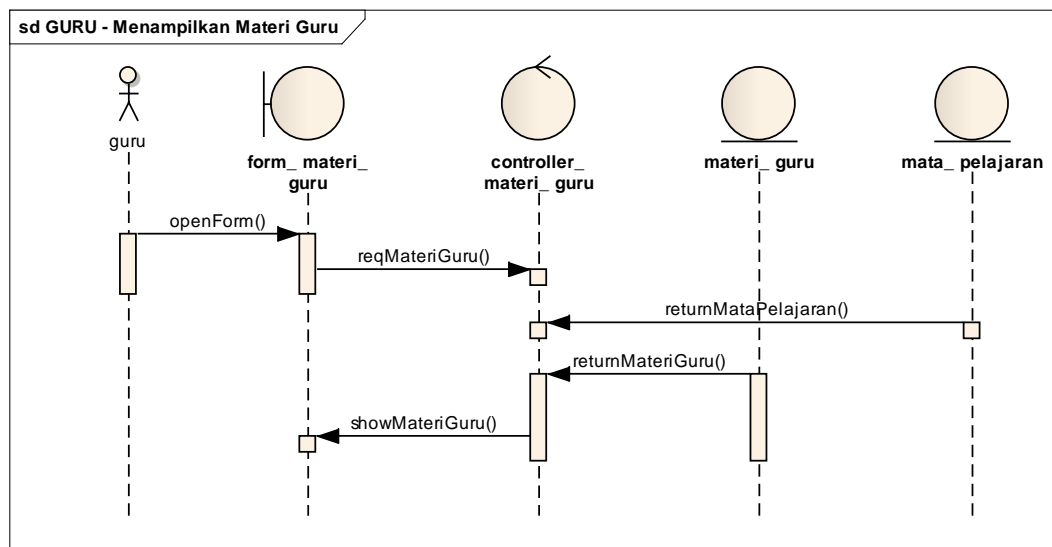
Aktor GURU membuka form_materi_mgmp kemudian controller_materi_mgmp akan mengambil data yang disimpan pada materi_mgmp. Pengambilan data berdasarkan mata_pelajaran.



Gambar 3.14 Sequence Diagram Menampilkan Materi MGMP (Guru)

13. Sequence Diagram Menampilkan Materi Guru (Guru)

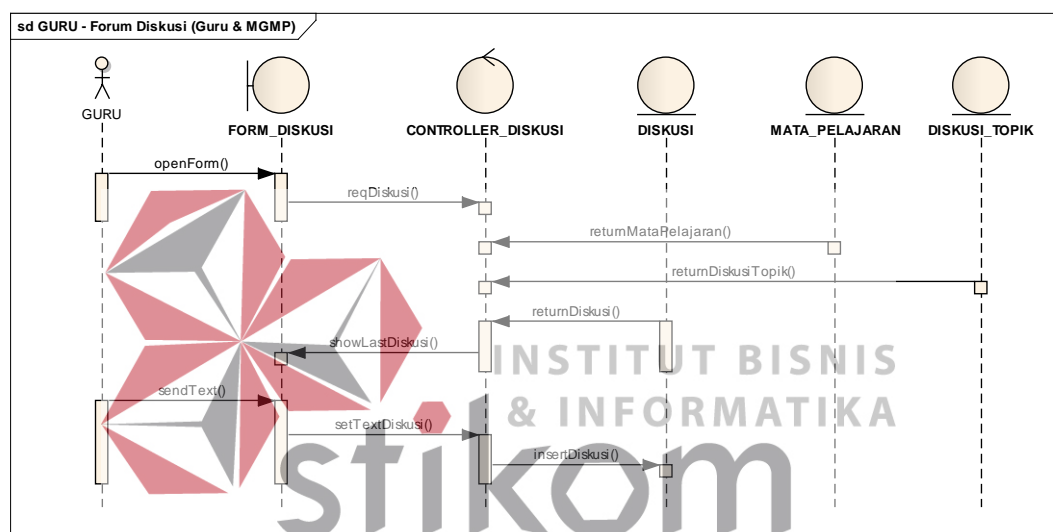
Aktor GURU membuka form_materi_guru kemudian controller_materi_guru akan mengambil data. Setelah itu disimpan pada materi guru berdasarkan mata_pelajaran. Kemudian ditampilkan pada form_materi_guru.



Gambar 3.15 Sequence Diagram Menampilkan Materi Guru (Guru)

14. Sequence Diagram Forum Diskusi Guru dan MGMP (GURU)

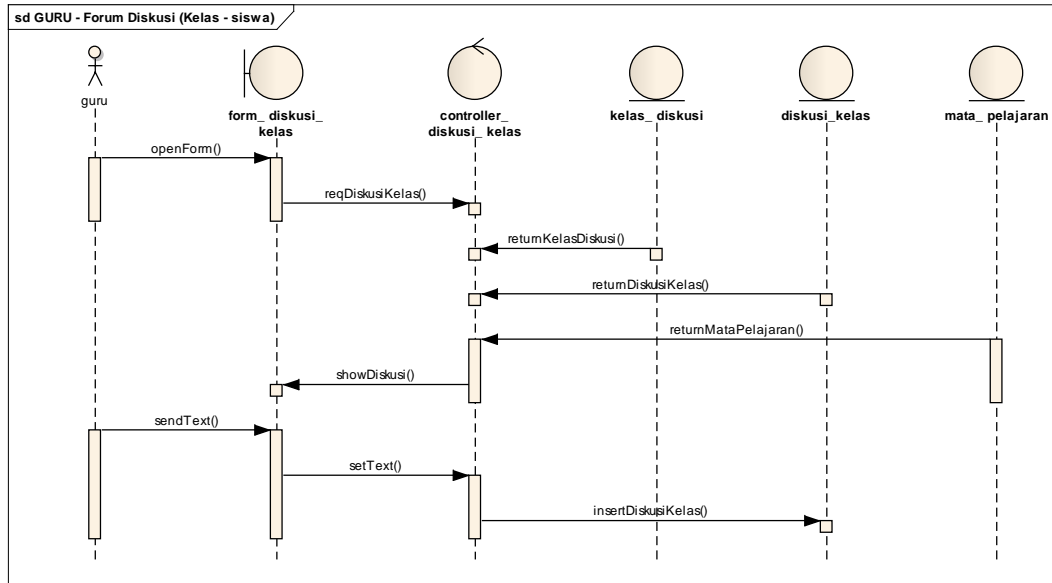
Aktor GURU membuka form_diskusi. Setelah itu controller_diskusi akan mengambil data yang kemudian disimpan pada diskusi. Diskusi tersebut diambil berdasarkan mata_pelajaran dan diskusi_topik. Setelah itu pesan teks pengguna lainnya dapat ditampilkan. Untuk mengirim pesan, aktor GURU mengirim pesan teks melalui form_diskusi kemudian akan dikirim lewat controller_diskusi menuju server.



Gambar 3.16 Sequence Diagram Forum Diskusi-Guru dan MGMP (GURU)

15. Sequence Diagram Forum Diskusi Kelas (GURU)

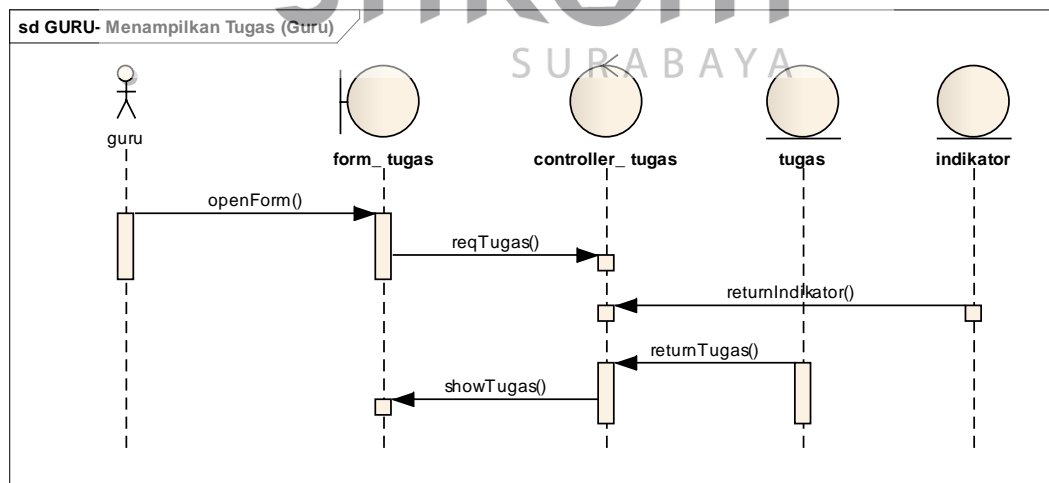
Aktor GURU akan masuk pada form_diskusi_kelas kemudian controller_diskusi_kelas akan mengambil pesan dan disimpan dalam diskusi_kelas. Diskusi_kelas akan ditampilkan berdasarkan kelas_diskusi dan mata_pelajaran. Setelah diambil maka pesan teks akan ditampilkan pada form_diskusi_kelas. Mengirim pesan teks dikirim melalui controller_diskusi_kelas ke server.



Gambar 3.17 *Sequence Diagram* Forum Diskusi Kelas (GURU)

16. *Sequence Diagram* Menampilkan Tugas (GURU)

Aktor GURU membuka form_tugas kemudian controller_tugas akan mengambil data yang akan disimpan pada tugas. Tugas yang ditampilkan dibutuhkan indikator sebagai standar dari tugas yang telah dibuat. Setelah pengambilan data selesai data langsung ditampilkan pada form_tugas.

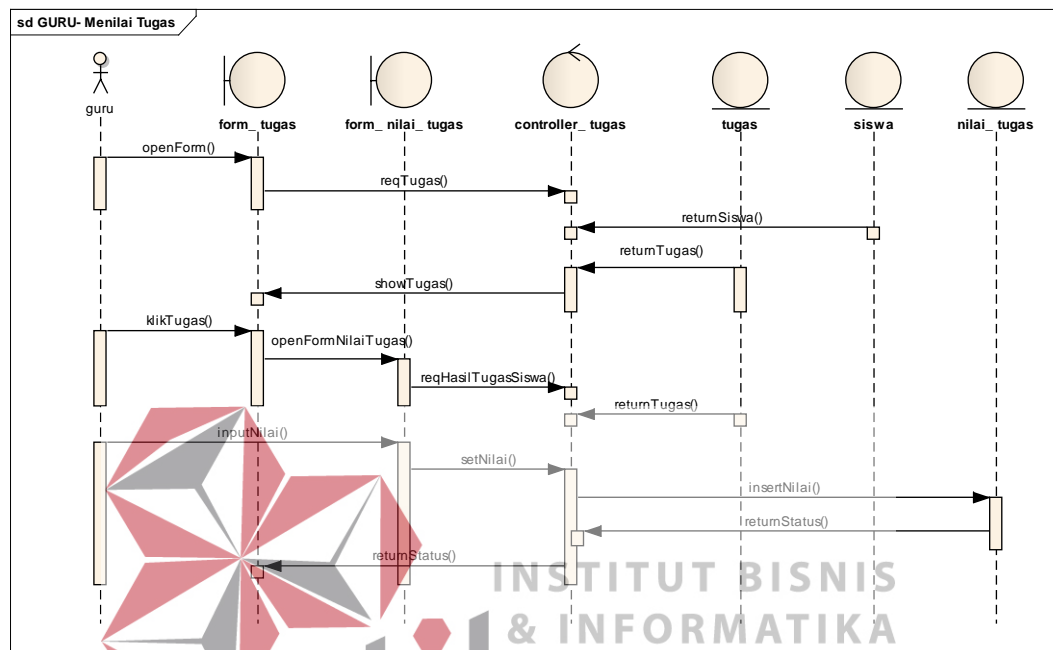


Gambar 3.18 *Sequence Diagram* Menampilkan Tugas (GURU)

17. *Sequence Diagram* Menilai Tugas (Guru)

Aktor GURU membuka form_tugas yang isinya adalah *List* tugas SISWA yang telah mengumpulkan. Kemudian aktor akan klik salah satu siswa yang telah

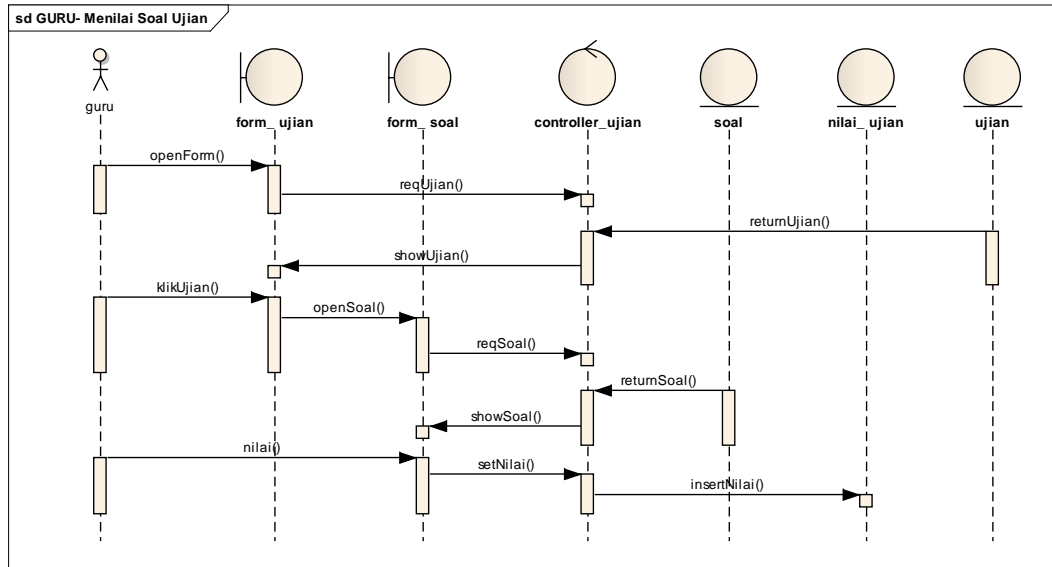
mengumpulkan kemudian aplikasi akan menuju form_nilai_tugas. Pada form_nilai_tugas, GURU akan entri nilai sesuai dengan hasil kerjaan SISWA yang diunduh oleh GURU. Setelah melakukan entri nilai kemudian nilai tersebut akan dikirim pada controller_tugas kemudian dimasukkan ke dalam server.



Gambar 3.19 Sequence Diagram Menilai Tugas (Guru)

18. Sequence Diagram Menilai Soal Ujian (GURU)

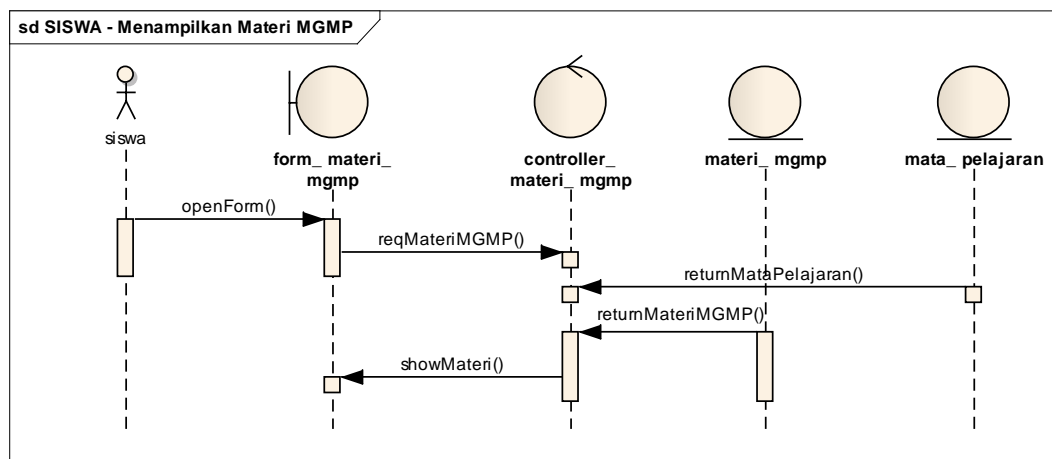
Aktor GURU membuka form_ujian yang telah terisi dengan siswa yang telah mengikuti ujian. Setelah itu GURU akan klik salah satu siswa untuk menilai ujian. Aplikasi akan menuju form_soal kemudian controller_ujian mengambil soal yang disimpan dalam soal. Setelah itu GURU entri nilai untuk soal esai. Setelah itu disimpan melalui controller_ujian.



Gambar 3.20 Sequence Diagram Menilai Soal Ujian (GURU)

19. Sequence Diagram Menampilkan Materi MGMP (Siswa)

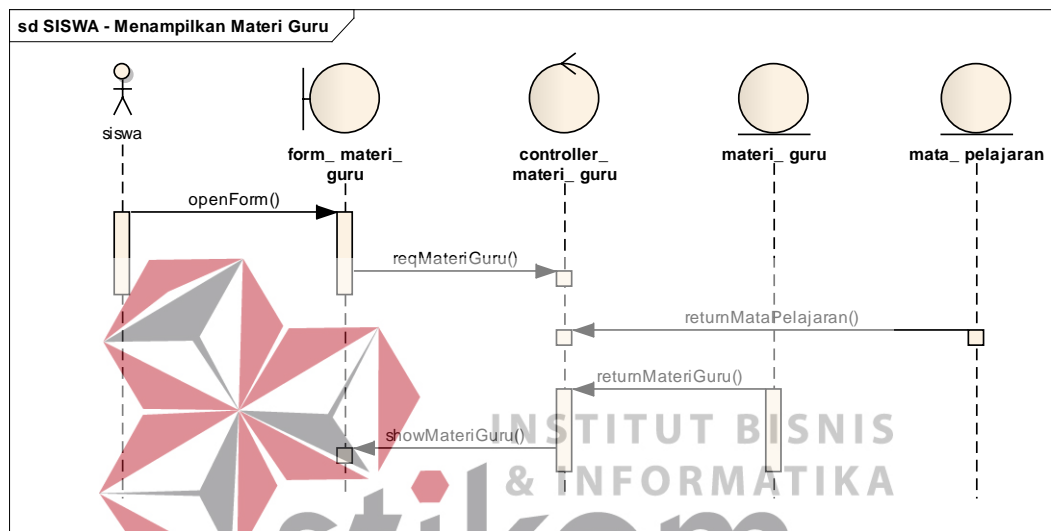
Aktor SISWA membuka form_materi_mgmp kemudian controller_materi_mgmp mengambil data dan dimasukkan ke dalam materi_mgmp. Materi MGMP ini akan ditampilkan berdasarkan mata_pelajaran dan kelas yang sedang ditempuh oleh SISWA. Setelah itu materi_mgmp akan ditampilkan pada form_materi_mgmp.



Gambar 3.21 Sequence Diagram Menampilkan Materi MGMP (Siswa)

20. Sequence Diagram Menampilkan Materi Guru (Siswa)

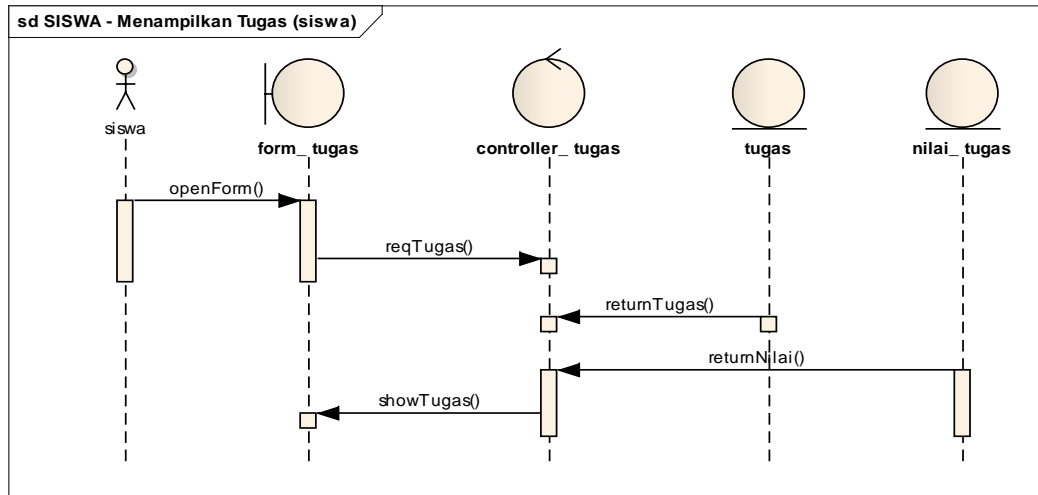
Aktor SISWA membuka form_materi_guru kemudian controller_materi_guru akan mengambil data yang dimasukkan ke dalam materi_guru. Materi guru ini akan ditampilkan berdasarkan mata_pelajaran dan kelas yang ditempuh oleh siswa. materi_guru yang ditampilkan pada SISWA adalah materi_guru yang mengajar SISWA di kelas tersebut.



Gambar 3.22 Sequence Diagram Menampilkan Materi Guru (Siswa)

21. Sequence Diagram Menampilkan Tugas (Siswa)

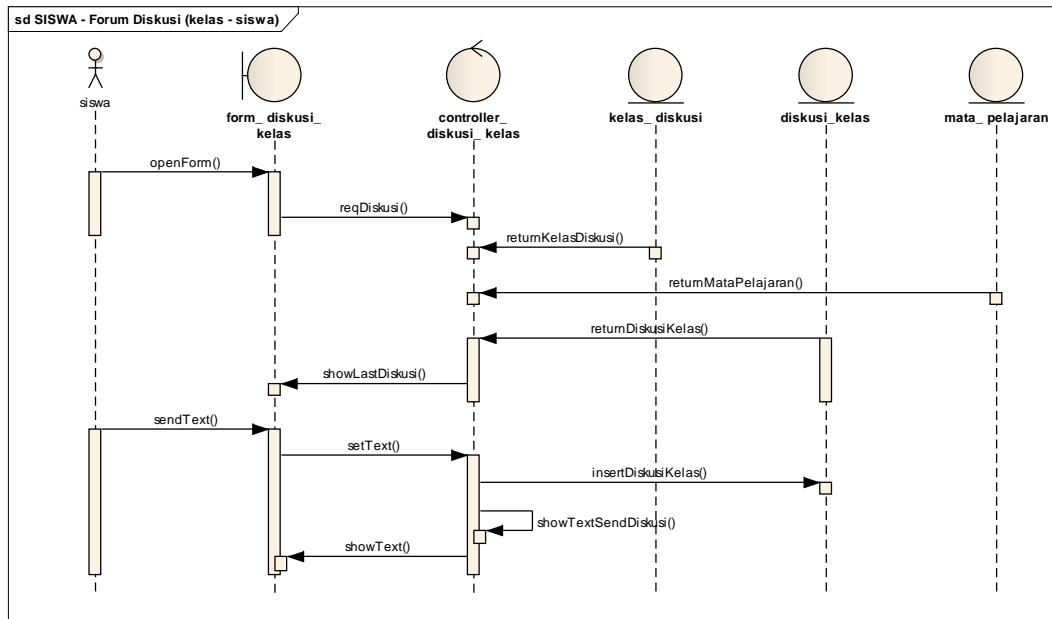
Aktor SISWA membuka form_tugas kemudian controller_tugas akan mengambil data tugas yang disimpan dalam tugas. Ketika menampilkan tugas, nilai_tugas juga ditampilkan. Setelah pengambilan data selesai maka ditampilkan pada form_tugas.



Gambar 3.23 *Sequence Diagram* Menampilkan Tugas (Siswa)

22. *Sequence Diagram* Forum Diskusi Kelas (Siswa)

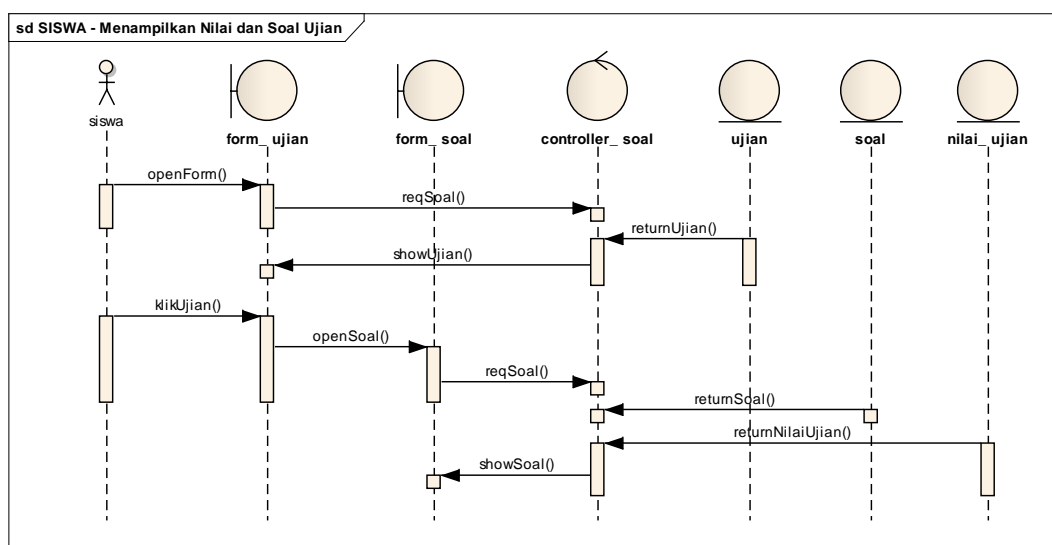
Aktor SISWA membuka form_diskusi_kelas kemudian controller_diskusi_kelas akan mengambil pesan teks yang akan disimpan pada diskusi_kelas. Pesan teks tersebut akan ditampilkan berdasarkan mata_pelajaran dan kelas_diskusi. Setelah itu pesan teks milik pengguna lain yang masih dalam satu kelas_diskusi akan ditampilkan di form_diskusi_kelas. Untuk mengirim pesan teks, siswa menulis pesan lalu disimpan melalui controller_diskusi_kelas dan dikirim ke server. Setelah itu pesan yang telah masuk akan langsung ditampilkan pada form_diskusi_kelas.



Gambar 3.24 *Sequence Diagram* Forum Diskusi Kelas (Siswa)

23. *Sequence Diagram* Menampilkan Nilai dan Soal Ujian (Siswa)

Aktor SISWA membuka form_ujian yang berisi ujian yang diambil melalui controller_soal. Setelah ujian tampil, siswa bisa klik salah satu dari ujian kemudian aplikasi akan langsung menuju form_soal. controller_soal akan memanggil soal-soal yang disimpan dalam soal. Ketika ditampilkan nilai_ujian juga masuk untuk ditampilkan pada form_ujian.



Gambar 3.25 *Sequence Diagram* Menampilkan Nilai dan Soal Ujian (Siswa)

3.2.4 Class Diagram

Class diagram digunakan untuk membentuk paket-paket dalam sistem yang di dalamnya terdapat atribut dan operasi yang dibutuhkan untuk memproses data. *Class diagram* salah satu cara untuk merancang perangkat lunak untuk membuat objek-objek setiap benda ataupun entitas.

Perancangan aplikasi berorientasi objek pada *use case*, dibutuhkan *class diagram* untuk memroses dan membangun aplikasi berbasis Android. *Class diagram* yang dibangun akan ditampilkan secara terpisah dan dihubungkan dengan relasi bertujuan untuk mempermudah pembacaan.

3.2.5 Class Model

Class Model memiliki isi atribut-atribut yang berfungsi untuk memudahkan menampilkan atribut dari sebuah entitas jika dibutuhkan untuk ditampilkan. Pada aplikasi *MoLearn* siswa memiliki *Class Model* yang terbagi menjadi 23 *class Model*.

1. Class modelGuru

Class modelGuru menangani pengguna sebagai guru ketika *login*. Semua atribut yang ada di *class modelGuru* dapat ditampilkan di aplikasi Android. *Class modelGuru* terdapat atribut `guru_id`, `gu_id`, `sekolah_id`, `sekolah_nama`, `guru_nama`, `guru_nuptk`, `guru_jk`, `kota_id`, `kota_nama`, `guru_tgl_lahir`, `guru_email`, `guru_telp`, `guru_password`, `guru_status_aktif`, `guru_sekolah_alamat`. Selain itu terdapat dua *constructor* dan *setter getter*.

modelGuru
<pre> - guru_id: String - gu_id: String - sekolah_id: String - sekolah_nama: String - guru_nama: String - guru_nuptk: String - guru_jk: String - kota_id: String - kota_nama: String - guru_tgl_lahir: String - guru_email: String - guru_telp: String - guru_password: String - guru_status_aktif: String - guru_sekolah_alamat: String </pre>
<pre> + modelGuru() + modelGuru(String, String, String, String, String, String, String, String, String, String, String, String, String) + setGuru_id(String) : void + setGu_id(String) : void + setSekolah_id(String) : void + setSekolah_nama(String) : void + setGuru_nama(String) : void + setGuru_nuptk(String) : void + setGuru_jk(String) : void + setKota_id(String) : void + setKota_nama(String) : void + setGuru_tgl_lahir(String) : void + setGuru_email(String) : void + setGuru_telp(String) : void + setGuru_status_aktif(String) : void + setGuru_sekolah_alamat(String) : void + getGuru_id() : String + getGu_id() : String + getSekolah_id() : String + getSekolah_nama() : String + getGuru_nama() : String + getGuru_nuptk() : String + getGuru_jk() : String + getKota_id() : String + getKota_nama() : String + getGuru_tgl_lahir() : String + getGuru_email() : String + getGuru_telp() : String + getGuru_password() : String + getGuru_status_aktif() : String + getGuru_sekolah_alamat() : String </pre>

Gambar 3.26 Class modelGuru

2. Class modelSiswa

Class modelSiswa berguna untuk menangani pengguna sebagai siswa ketika login pada aplikasi Android. Atribut-atribut yang ada pada class modelSiswa yaitu siswa_tingkat, siswa_id, sekolah_id, sekolah_nama, kota_id, gu_id, guru_nama, siswa_nisn, siswa_nama, siswa_jk, siswa_tgl_lahir, siswa_alamat, siswa_email, siswa_telp_ortu, siswa_email_ortu, siswa_status_aktif beserta dua *constructor* dan *setter getter*.

modelSiswa	
<ul style="list-style-type: none"> - siswa_tingkat: int - siswa_id: String - sekolah_id: String - sekolah_nama: String - kota_id: String - gu_id: String - guru_nama: String - siswa_nisn: String - siswa_nama: String - siswa_jk: String - siswa_tgl_lahir: String - siswa_alamat: String - siswa_email: String - siswa_telp_ortu: String - siswa_email_ortu: String - siswa_status_aktif: String 	<ul style="list-style-type: none"> + modelSiswa() + modelSiswa(String, String, String, String, String, String, String, String, String, String, String, String, String, String, String, int) + setSiswa_tingkat(int) : void + setSiswa_id(String) : void + setSekolah_id(String) : void + setSekolah_nama(String) : void + setKota_id(String) : void + setGu_id(String) : void + setGuru_nama(String) : void + setSiswa_nisn(String) : void + setSiswa_nama(String) : void + setSiswa_jk(String) : void + setSiswa_tgl_lahir(String) : void + setSiswa_alamat(String) : void + setSiswa_email(String) : void + setSiswa_telp_ortu(String) : void + setSiswa_email_ortu(String) : void + setSiswa_status_aktif(String) : void + getSiswa_tingkat() : int + getSiswa_id() : String + getSekolah_id() : String + getSekolah_nama() : String + getKota_id() : String + getGu_id() : String + getGuru_nama() : String + getSiswa_nisn() : String + getSiswa_nama() : String + getSiswa_jk() : String + getSiswa_tgl_lahir() : String + getSiswa_alamat() : String + getSiswa_email() : String + getSiswa_telp_ortu() : String + getSiswa_email_ortu() : String + getSiswa_status_aktif() : String

Gambar 3.27 Class modelSiswa

3. Class modelMgmp

Class modelMgmp digunakan untuk menangani ketika pengguna sebagai MGMP login pada aplikasi Android. Atribut-atribut yang ada pada *class modelMgmp* yaitu *user_id*, *user_nama*, *user_status_aktif* beserta dua *constructor* dan *setter getter*.

modelMgmp
<ul style="list-style-type: none"> - user_id: String - user_nama: String - user_status_aktif: String
<ul style="list-style-type: none"> + modelMgmp() + modelMgmp(String, String, String) + setUser_id(String) : void + setUser_nama(String) : void + setStatus_aktif(String) : void + getUser_id() : String + getUser_nama() : String + getStatus_aktif() : String

Gambar 3.28 Class modelMgmp

4. Class modelChatDiskusiKelas

Class *modelChatDiskusiKelas* berguna untuk menangani atribut-atribut ketika menampilkan pesan teks untuk forum diskusi kelas. Atribut yang ada pada class *modelChatDiskusiKelas* yaitu *kpId*, *kdiId*, *kdiPosting*, *kdiWaktu*, *kdiUser*, *kdiAs*, *kdiUserName* beserta dua *constructor* dan *setter*.

modelChatDiskusiKelas
<ul style="list-style-type: none"> - kpId: String - kdiId: String - kdiPosting: String - kdiWaktu: String - kdiUser: String - kdiAs: String - kdiUserName: String
<ul style="list-style-type: none"> + modelChatDiskusiKelas() + setKpId(String) : void + setKdiId(String) : void + setKdiPosting(String) : void + setKdiWaktu(String) : void + setKdiUser(String) : void + setKdiAs(String) : void + setKdiUserName(String) : void + getKpId() : String + getKdiId() : String + getKdiPosting() : String + getKdiWaktu() : String + getKdiUser() : String + getKdiAs() : String + getKdiUserName() : String + modelChatDiskusiKelas(String, String, String, String, String, String, String)

Gambar 3.29 Class modelChatDiskusiKelas

5. Class modelChatDiskusiOnline

Class *modelChatDiskusiOnline* digunakan untuk menangani atribut-atribut yang berhubungan dengan pesan teks forum diskusi guru dan MGMP. Atribut yang ada pada class *modelChatDiskusiOnline* yaitu *dkId*, *doId*, *doPosting*,

doUser, doWaktu, doAs, doNama, doStatus beserta dua *constructor* dan *setter* *getter*.

modelChatDiskusiOnline	
-	dkId: String
-	dold: String
-	doPosting: String
-	doUser: String
-	doWaktu: String
-	doAs: String
-	doNama: String
-	doStatus: String
+	modelChatDiskusiOnline()
+	modelChatDiskusiOnline(String, String, String, String, String, String, String, String)
+	setDkId(String) : void
+	setDold(String) : void
+	setDoPosting(String) : void
+	setDoUser(String) : void
+	setDoWaktu(String) : void
+	setDoAs(String) : void
+	setDoNama(String) : void
+	setDoStatus(String) : void
+	getDkId() : String
+	getDold() : String
+	getDoPosting() : String
+	getDoUser() : String
+	getDoWaktu() : String
+	getDoAs() : String
+	getDoNama() : String
+	getDoStatus() : String

Gambar 3.30 Class modelChatDiskusiOnline

6. Class modelMataPelajaran

Class *modelMataPelajaran* digunakan untuk menangani atribut-atribut mata pelajaran ketika menampilkan mata pelajaran. Atribut yang ada pada *class modelMataPelajaran* yaitu MP_ID dan MP_NAMA beserta dua *constructor* dan *setter* *getter*.

modelMataPelajaran	
-	MP_ID: String
-	MP_NAMA: String
+	modelMataPelajaran()
+	modelMataPelajaran(String, String) : void
+	setMp_id(String) : void
+	setMp_nama(String) : void
+	getMp_id() : String
+	getMp_nama() : String

Gambar 3.31 Class modelMataPelajaran

7. Class modelDiskusiOnline

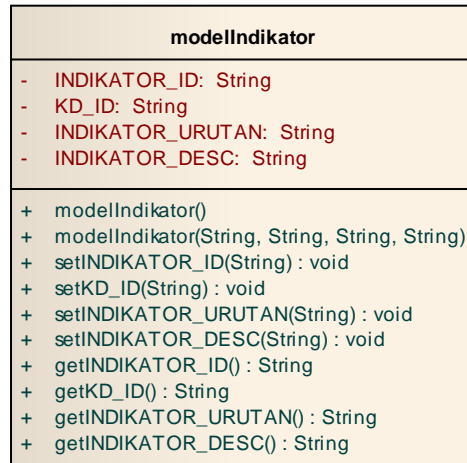
Class modelDiskusiOnline digunakan untuk menangani atribut ketika menampilkan daftar dari diskusi guru dan MGMP. Atribut yang ada pada *class modelDiskusiOnline* yaitu *dkId*, *mpId*, *dkNama*, *dkDesc*, *dkStatus* beserta dua *constructor* dan *setter getter*.

modelDiskusiOnline	
-	<i>dkId</i> : String
-	<i>mpId</i> : String
-	<i>dkNama</i> : String
-	<i>dkDesc</i> : String
-	<i>dkStatus</i> : String
+	<i>modelDiskusiOnline</i> ()
+	<i>modelDiskusiOnline</i> (String, String, String, String, String)
+	<i>setDkId</i> (String) : void
+	<i>setMpId</i> (String) : void
+	<i>setDkNama</i> (String) : void
+	<i>setDkDesc</i> (String) : void
+	<i>setDkStatus</i> (String) : void
+	<i>getDkId</i> () : String
+	<i>getMpId</i> () : String
+	<i>getDkNama</i> () : String
+	<i>getDkDesc</i> () : String
+	<i>getDkStatus</i> () : String

Gambar 3.32 *Class modelDiskusiOnline*

8. Class modelIndikator

Class modelIndikator digunakan untuk menangani atribut yang berhubungan dengan indikator dari mata pelajaran. *Class modelIndikator* memiliki atribut yaitu *indikator_id*, *kd_id*, *indikator_urutan*, *indikator_desc* beserta dua *constructor* dan *setter getter*.



Gambar 3.33 Class modelIndikator

9. Class modelJawabanSoal

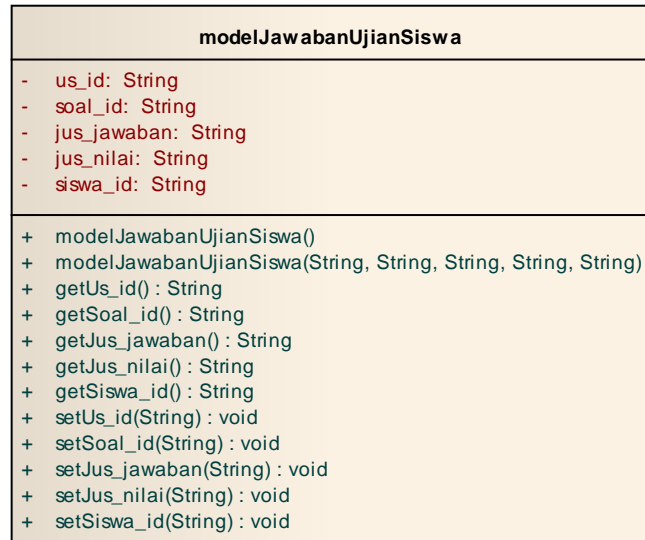
Class *modelJawabanSoal* digunakan untuk menangani atribut dari jawaban soal ketika soal ditampilkan. Class *modelJawabanSoal* memiliki atribut yaitu *jsID*, *soalID*, *jsISI*, *jsStatusAktif* beserta dua *constructor* dan *setter getter*.



Gambar 3.34 Class modelJawabanSoal

10. Class modelJawabanUjianSiswa

Class *modelJawabanUjianSiswa* digunakan untuk menangani atribut jawaban soal dari siswa yang telah melakukan ujian. Atribut yang ada pada class *modelJawabanUjianSiswa* yaitu *us_id*, *soal_id*, *jus_jawaban*, *jus_nilai*, *siswa_id* beserta dua *constructor* dan *setter getter*.



Gambar 3.35 Class modelJawabanUjianSiswa

11. Class modelKelasPembelajaran

Class modelKelasPembelajaran digunakan untuk menangani atribut dari kelas pembelajaran. Atribut kelas pembelajaran terdiri dari kp_id, mp_id, guru_id, kp_nama, kp_tingkat, kp_status_aktif, jumlah_siswa beserta dua constructor dan setter getter.



Gambar 3.36 Class modelKelasPembelajaran

12. Class `modelKelasTugas`

Class `modelKelasTugas` digunakan untuk menangani atribut kelas yang diberikan tugas oleh guru kepada siswa dalam satu kelas. *Class `modelKelasTugas`* memiliki atribut yaitu `kt_id`, `kp_id`, `guru_id`, `mp_id`, `kt_nama`, `kt_deadline`, `kt_kelompok`, `mIndikator` beserta dua *constructor* dan *setter getter*.



Gambar 3.37 *Class `modelKelasTugas`*

13. Class `modelKelasUjian`

Class `modelKelasUjian` digunakan untuk menangani atribut kelas ujian dari siswa ketika ada ujian class `modelKelasUjian` ini berhubungan ketika guru ingin menampilkan siswa yang telah melaksanakan ujian. Atribut *class `modelKelasUjian`* terdiri dari `ku_id`, `kp_id`, `ku_nama`, `ku_durasi`, `ku_expired`, `ku_publish` beserta dua *constructor* dan *setter getter*.

modelKelasUjian	
-	ku_id: String
-	kp_id: String
-	ku_nama: String
-	ku_durasi: int
-	ku_expired: String
-	ku_publish: String
+ modelKelasUjian()	
+ modelKelasUjian(String, String, int, String, String, String)	
+	setKu_id(String) : void
+	setKp_id(String) : void
+	setKu_nama(String) : void
+	setKu_durasi(int) : void
+	setKu_expired(String) : void
+	setKu_publish(String) : void
+	getKu_id() : String
+	getKp_id() : String
+	getKu_nama() : String
+	getKu_durasi() : int
+	getKu_expired() : String
+	getKu_publish() : String

Gambar 3.38 Class modelKelasUjian

14. Class modelKelompokUjianSiswa

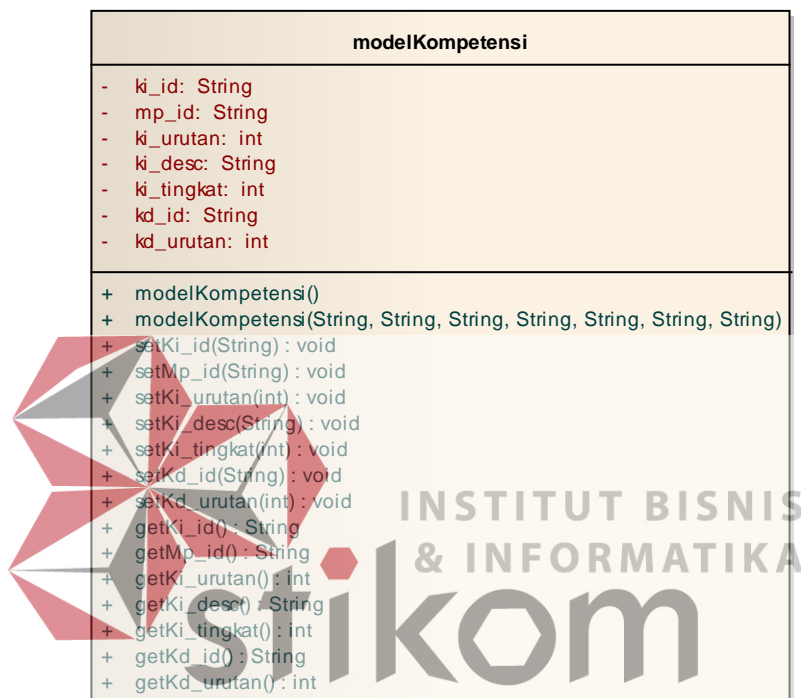
Class *modelKelompokUjianSiswa* digunakan untuk menangani atribut yang berhubungan dengan ujian yang dilaksanakan oleh siswa. Atribut class *modelKelompokUjianSiswa* terdiri dari *us_id*, *soal_id*, *jus_jawaban*, *jus_nilai*, *siswa_id* beserta dua *constructor* dan *setter* *getter*.

modelKelompokUjianSiswa	
-	us_id: String
-	siswa_id: String
-	siswa_nama: String
-	ku_id: String
-	us_mulai: String
-	us_selesai: String
-	us_nilai: int
+ modelKelompokUjianSiswa()	
+ modelKelompokUjianSiswa(int, String, String, String, String, String, String)	
+	setUs_id(String) : void
+	setSiswa_id(String) : void
+	setSiswa_nama(String) : void
+	setKu_id(String) : void
+	setUs_mulai(String) : void
+	setUs_selesai(String) : void
+	setUs_nilai(int) : void
+	getUs_id() : String
+	getSiswa_id() : String
+	getSiswa_nama() : String
+	getKu_id() : String
+	getUs_mulai() : String
+	getUs_selesai() : String
+	getUs_nilai() : int

Gambar 3.39 Class modelKelompokUjianSiswa

15. Class modelKompetensi

Class modelKompetensi digunakan untuk menangani atribut dari kompetensi yang ditampilkan ketika ingin menampilkan kompetensi dasar dan kompetensi inti. Atribut *class modelKompetensi* yaitu *ki_id*, *mp_id*, *ki_urutan*, *ki_desc*, *ki_tingkat*, *kd_id*, *kd_urutan* beserta dua *constructor* dan *setter getter*.



Gambar 3.40 *Class modelKompetensi*

16. Class modelMateriGuru

Class modelMateriGuru digunakan untuk menangani atribut materi dari guru. *Class modelMateriGuru* memiliki atribut yaitu *mg_id*, *guru_id*, *guru_nama*, *mp_id*, *mp_nama*, *mg_tingkat*, *mg_judul*, *mg_file*, *mg_tipe*, *mg_waktu_upload*, *mg_share*, *mg_download*, *sekolah_nama*, *kompetensi* beserta dua *constructor* dan *setter getter*.

modelMateriGuru
<pre> - mg_id: String - guru_id: String - guru_nama: String - mp_id: String - mp_nama: String - mg_tingkat: int - mg_judul: String - mg_file: String - mg_tipe: String - mg_waktu_upload: String - mg_share: String - mg_download: int - sekolah_nama: String - kompetensi: String </pre>
<pre> + modelMateriGuru() + modelMateriGuru(String, String, boolean, String, String, String, String, String, int, String, String, String, String) + setMg_id(String) : void + setGuru_id(String) : void + setGuru_nama(String) : void + setMp_id(String) : void + setMp_nama(String) : void + setMg_tingkat(int) : void + setMg_judul(String) : void + setMg_file(String) : void + setMg_tipe(String) : void + setMg_waktu_upload(String) : void + setMg_share(String) : void + setMg_download(int) : void + setSekolah_nama(String) : void + setKompetensi(String) : void + getMg_id() : String + getGuru_id() : String + getGuru_nama() : String + getMp_id() : String + getMp_nama() : String + getMg_tingkat() : int + getMg_judul() : String + getMg_file() : String + getMg_tipe() : String + getMg_waktu_upload() : String + getMg_share() : String + getMg_download() : int + getSekolah_nama() : String + getKompetensi() : String </pre>

Gambar 3.41 *Class modelMateriGuru*

17. *Class modelMateriMgmp*

Class modelMateriMGMP digunakan untuk menangani atribut yang berhubungan dengan materi yang berasal dari MGMP. Atribut yang ada pada *class modelMateriMGMP* yaitu *mf_id*, *kd_id*, *user_id*, *mf_judul*, *mf_file*, *mf_waktu_upload*, *kompetensi*, *mf_share*, *mf_download*, dan *mf_tipe* beserta dua *constructor* dan *setter getter*.

modelMateriMgmp	
<ul style="list-style-type: none"> - mf_id: String - kd_id: String - user_id: String - mf_judul: String - mf_file: String - mf_waktu_upload: String - kompetensi: String - mf_share: int - mf_download: int - mf_tipe: int 	
<ul style="list-style-type: none"> + modelMateriMgmp() + modelMateriMgmp(int, int, int, String, String, String, String, String, String, String) : void + setMf_id(String) : void + setKd_id(String) : void + setUser_id(String) : void + setMf_judul(String) : void + setMf_file(String) : void + setMf_waktu_upload(String) : void + setKompetensi(String) : void + setMf_share(int) : void + setMf_download(int) : void + setMf_tipe(int) : void + getMf_id() : String + getKd_id() : String + getUser_id() : String + getMf_judul() : String + getMf_file() : String + getMf_waktu_upload() : String + getKompetensi() : String + getMf_share() : int + getMf_download() : int + getMf_tipe() : int 	

Gambar 3.42 Class *modelMateriMgmp*

18. Class *modelNamaSekolah*

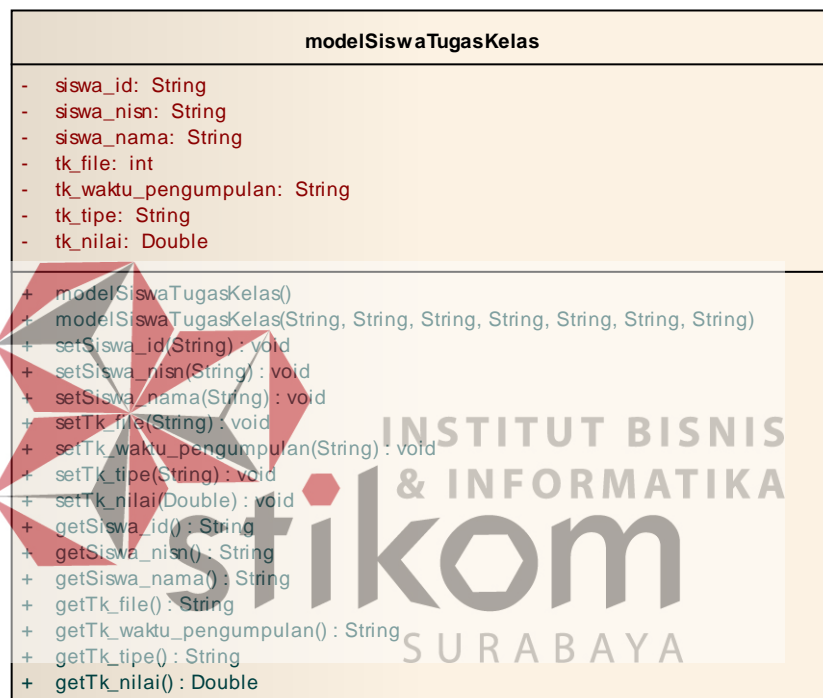
Class *modelNamaSekolah* digunakan untuk menangani nama sekolah ketika *login* ke dalam aplikasi. Atribut yang ada pada class *modelNamaSekolah* yaitu *sekolah_id* dan *sekolah_nama* beserta dua constructor dan setter getter.

modelNamaSekolah	
<ul style="list-style-type: none"> - sekolah_id: String - sekolah_nama: String 	
<ul style="list-style-type: none"> + modelNamaSekolah() + modelNamaSekolah(String, String) : void + setSekolah_id(String) : void + setSekolah_nama(String) : void + getSekolah_id() : String + getSekolah_nama() : String 	

Gambar 3.43 Class *modelNamaSekolah*

19. Class `modelSiswaTugasKelas`

Class `modelSiswaTugasKelas` digunakan untuk menangani atribut dari tugas kelas siswa yang telah mengumpulkan tugas. Class `modelSiswaTugasKelas` memiliki atribut yaitu `siswa_id`, `siswa_nisn`, `siswa_nama`, `tk_file`, `tk_waktu_pengumpulan`, `tk_tipe`, `tk_nilai` beserta dua *constructor* dan *setter getter*.



Gambar 3.44 Class `modelSiswaTugasKelas`

20. Class `modelSoal`

Class `modelSoal` digunakan untuk menangani atribut soal yang akan ditampilkan. Atribut pada class `modelSoal` yaitu `soal_id`, `indikator_id`, `soal_pertanyaan`, `soal_tipe`, `soal_jawaban`, `soal_status_aktif`, `soal_level`, `arrayListModelJawabanoal`, `mJawabanUjianSiswa` beserta dua *constructor* dan *setter getter*.

modelSoal
<ul style="list-style-type: none"> - soal_id: String - indikator_id: String - soal_pertanyaan: String - soal_tipe: String - soal_jawaban: String - soal_status_aktif: String - soal_level: String - arrayListModelJawabanSoal: ArrayList<modelJawabanSoal> - mJawabanUjianSiswa: modelJawabanUjianSiswa
<ul style="list-style-type: none"> + modelSoal() + modelSoal(String, String, String, String, String, String, String) : void + setSoal_id() : String + setIndikator_id(String) : String + setSoal_pertanyaan(String) : void + setSoal_tipe(String) : void + setSoal_jawaban(String) : void + setSoal_status_aktif(String) : void + setSoal_level(String) : void + setArrayListModelJawabanSoal(ArrayList<ModelJawabanSoal>) : void + setmJawabanUjianSiswa(modelJawabanUjianSiswa) : void + getSoal_id() : String + getIndikator_id() : String + getSoalPertanyaan() : String + getSoalTipe() : String + getSoalJawaban() : String + getSoalStatusAktif() : String + getSoalLevel() : String + getArrayListModelJawabanSoal() : ArrayList<modelJawabanSoal> + getmJawabanUjianSiswa() : modelJawabanUjianSiswa

Gambar 3.45 Class modelSoal

21. Class modelTugasKelas

Class modelTugasKelas digunakan untuk menangani atribut dari tugas yang diberikan siswa oleh guru. Atribut pada *class modelTugasKelas* yaitu *kt_id*, *kp_id*, *kt_nama*, *kt_deadline*, *kt_kelompok*, *status_kumpul*, *tk_nilai* beserta dua *constructor setter* dan *getter*.

modelTugasKelas
<ul style="list-style-type: none"> - kt_id: String - kp_id: String - kt_nama: String - kt_deadline: String - kt_kelompok: String - status_kumpul: String - tk_nilai: String
<ul style="list-style-type: none"> + modelTugasKelas() + modelTugasKelas(String, String, String, String, String, String, String, String) : void + setKt_id(String) : void + setKp_id(String) : void + setKt_nama(String) : void + setKt_deadline(String) : void + setKt_kelompok(String) : void + setStatus_kumpul(String) : void + setTk_nilai(String) : void + getKt_id() : String + getKp_id() : String + getKt_nama() : String + getKt_deadline() : String + getKt_kelompok() : String + getStatus_kumpul() : String + getTk_nilai() : String

Gambar 3.46 Class modelTugasKelas

22. Class modelSoalMgmp

Class modelSoalMgmp digunakan untuk menangani atribut soal ketika login sebagai MGMP. Atribut dari *class modelSoalMgmp* yaitu soalID, indikatorID, soalPertanyaan, soalTipe, soalJawaban, soalStatusAktif, soalLevel, modelJawabanSoalList beserta dua *constructor* dan *setter getter*.



Gambar 3.47 *Class modelSoalMgmp*

23. Class modelUjianSiswa

Class modelUjianSiswa digunakan untuk menangani atribut ujian yang ditampilkan ketika login sebagai siswa. *Class modelUjianSiswa* memiliki atribut yaitu `us_id`, `siswa_id`, `siswa_nama`, `ku_id`, `us_mulai`, `us_selesai`, `us_nilai` beserta dua *constructor* dan *setter getter*.

modelUjianSiswa
<ul style="list-style-type: none"> - us_id: String - siswa_id: String - siswa_nama: String - ku_id: String - us_mulai: String - us_selesai: String - us_nilai: Double
<ul style="list-style-type: none"> + modelUjianSiswa() + modelUjianSiswa(Double, String, String, String, String, String, String, String) : void + setUs_id(String) : String + setSiswa_id(String) : void + setSiswa_nama(String) : void + setKu_id(String) : void + setUs_mulai(String) : void + setUs_selesai(String) : void + setUs_nilai(Double) : void + getUs_id() : String + getSiswa_id() : String + getSiswa_nama() : String + getKu_id() : String + getUs_mulai() : String + getUs_selesai() : String + getUs_nilai() : Double

Gambar 3.48 Class modelUjianSiswa

3.2.6 Class Controller

Class controller adalah kelas-kelas yang mengatur dan memberikan jembatan antara *class model* dan view (xml) pada aplikasi android. Pengecekan kesalahan juga akan diproses pada *controller* ini. *Class controller* dibagi menjadi tujuh belas *Class Controller*.

1. Class controllerChatGuruMgmp

Class controlleChatGuruMgmp digunakan untuk menangani pesan teks yang diambil dari *server* yang akan ditampilkan pada *client* antar pengguna sebagai guru dan pengguna sebagai MGMP. Selain itu controller ini juga digunakan untuk mengirim pesan teks yang akan tersimpan pada *server*.

controllerChatGuruMGMP
<ul style="list-style-type: none"> - modelChatDiskusiOnlineList: List<modelChatDiskusiOnline> - modelDiskusiOnlineList: List<modelDiskusiOnlineList> - mChatDiskusiOnline: modelChatDiskusiOnline
<ul style="list-style-type: none"> + controllerChatKelas() + setNewModelChatDiskusiOnlineList() : void + getModelChatDiskusiOnlienList() : List<modelChatDiskusiOnline> + getmChatDiskusiOnline() : modelChatDiskusiOnline + setmChatDiskusiOnline() : void + setParcelableChatDiskusiOnline() : void + addModelDiskusiOnlineList() : void + removaArray() : void + getChatDiskusiOnlineVolley() : void + getChatDiskusiOnlinePagingVolley() : void + setChatGuruMGMP() : void

Gambar 3.49 *Class controllerChatGuruMGMP*

2. Class controllerDownloadFile

Class controllerDownloadFile digunakan untuk menangani unduh materi ketika materi guru ataupun materi MGMP berbentuk berkas yang diunduh pada server.



controllerDownloadFile
<ul style="list-style-type: none"> - context: Context - activity: Activity - apkStorage: File - outputFile: File - callBackFile: CallBackFile
<ul style="list-style-type: none"> + controllerDownloadFile(Context) : void + onPreExecute() : void + doInBackGround() : String + onProgressUpdate() : void + onPostExecute() : void

Gambar 3.50 *Class controllerDownloadFile*

3. Class controllerDetilUjian

Class controllerUjian digunakan untuk menangani ketika menampilkan detail dari ujian sebagai pengguna guru ataupun pengguna siswa. Selain itu controller digunakan untuk menangani penilaian soal untuk pengguna guru.

controllerDetilUjian	
-	modelDetilUjianArrayList: List<modelKelasUjian>
-	mKelasPembelajaran: List<modelKelasPembelajaran>
+	getArrayListModelDetilUjian() : List<modelKelasUjian>
+	getmKelasUjianList() : List<modelKelasUjian>
+	getmKelasPembelajaran() : List<modelKelasPembelajaran>
+	addModelKelasPembelajaran() : void
+	getSizeArrayListModelDetilUjian() : void
+	addDetilUjian() : void
+	getKelasUjianForSiswaVolley() : void

Gambar 3.51 Class *controllerDetilUjian*

4. Class *controllerKelasTugas*

Class controllerKelasTugas digunakan untuk menangani pengambilan data dari *server* untuk menampilkan kelas yang berisi tugas-tugas siswa. Data kelas tugas akan diambil melalui *server*.

controllerKelasTugas	
-	modelKelasPembelajaranList: List<modelKelasPembelajaran>
-	modelKelasTugasList: List<modelKelasTugasList>
-	mKompetensiList: List<modelKompetensiList>
-	mIndikatorList: List<modelIndikatorList>
-	mKelasTugas: modelKelasTugas
-	mKelasPembelajaran: modelKelasPembelajaran
+	setParcelableGetIntentBefore() : void
+	getmIndikatorList() : ArrayList<modelIndikator>
+	setNewModelIndikatorList() : void
+	addmIndikatorList() : void
+	getModelKelasTugas() : modelKelasTugas
+	controllerKelasTugas()
+	addKompetensi() : void
+	addKelasPembelajaran() : void
+	addKelasTugas() : void
+	getModelKelasPembelajaranList() : List<modelKelasPembelajaran>
+	removeListKelasTugas() : void
+	removeKelasPembelajaran() : void
+	getModelKelasTugasList() : List<modelKelasTugas>
+	getmKompetensiList() : List<modelKompetensi>
+	getKelasPembelajaranVolley() : void
+	getKelasTugasVolley() : void
+	setKelasTugasVolley() : void
+	updateKelasTugasVolley() : void
+	getIndikator() : void

Gambar 3.52 Class *controllerKelasTugas*

5. Class *controllerChatKelas*

Class controllerChatKelas digunakan untuk menangani pengambilan data lewat *server* yang ditampilkan dalam bentuk pesan teks pada pengguna guru ataupun pengguna siswa.

controllerChatKelas
<ul style="list-style-type: none"> - modelChatDiskusiKelasList: List<modelDiskusiKelas> - modelKelasPembelajaranList: List<modelKelasPembelajaran> - mChatDiskusiKelas: modelChatDiskusiKelas
<ul style="list-style-type: none"> + controllerChatKelas() + getmChatDiskusiKelas() : modelChatDiskusiKelas + setmChatDiskusiKelas() : void + setParcelableDiskusiKelas() : void + setNewModelDiskusiKelasOnlineList() : void + addModelDiskusiOnlineList() : void + getModelDiskusiOnlineList() : List<modelDiskusiOnline> + setModelKelasPembelajaran() : void + getModelKelasPembelajaran() : List<modelKelasPembelajaran> + addNewModelKelasPembelajaran() : void + addNewChat() : void + getModelChatDiskusiKelasList() : List<modelChatDiskusiKelas> + removeChatList() : void + getChatDiskusiKelasLastVolley() : void + getChatDiskusiKelasLastVolley() : void + setChatDiskusiKelasVolley() : void + getListDiskusiKelasOnlineVolley() : void + getKelasPembelajaran() : void

Gambar 3.53 Class controllerChatKelas

6. Class controllerGuru

Class *controllerGuru* digunakan untuk menangani login pengguna guru yang dimana atributnya akan dicek dan diambil pada *server*. Atribut-atribut yang telah diambil akan dimasukkan ke dalam *session*.

controllerGuru
<ul style="list-style-type: none"> - pref: SharedPreferences - editor: SharedPreferences.Editor - context: Context
<ul style="list-style-type: none"> + controllerGuru() + getModelGuru() : modelGuru + setModelGuru() : void + setSessionGuru() : void + getSessionGuru() : HashMap<String, String> + setClearSessionGuru() : void + clearActivities() : void + isLoggedIn() : void + loginGuru() : void + getDataGuruVolley() : void

Gambar 3.54 Class controllerGuru

7. Class controllerKompetensi

Class *controllerKompetensi* digunakan untuk menangani pengambilan data dari server dan menampilkan isi kompetensi untuk kompetensi dasar, kompetensi inti, dan indikator.

controllerKompetensi
<ul style="list-style-type: none"> - <i>modelKompetensiList</i>: ArrayList<modelKompetensi> - <i>modelKompetensiDasarList</i>: ArrayList<modelKompetensiDasar> - <i>modelIndikatorList</i>: ArrayList<modelIndikator>
<ul style="list-style-type: none"> + <i>contollerKompetensi</i>() + <i>getArrayListModelKompetensi</i>() : ArrayList<modelKompetensi> + <i>addKompetensi</i>() : void + <i>setNewModelKompetensiDasar</i>() : void + <i>setNewIndikator</i>() : void + <i>addKompetensiDasar</i>() : void + <i>getModelKompetensiDasarList</i>() : ArrayList<modelKompetensiDasar> + <i>getModelIndikatorList</i>() : ArrayList<getModelIndikatorList> + <i>addIndikator</i>() : void + <i>getIndikatorList</i>() : void + <i>getKompetensiDasarVolley</i>() : void

Gambar 3.55 Class *controllerKompetensi*

8. Class controllerMataPelajaran

Class *controllerMataPelajaran* digunakan untuk menangani pengambilan data mata pelajaran dari server yang kemudian ditampilkan ketika login pada aplikasi Android.

controllerMataPelajaran
<ul style="list-style-type: none"> - <i>pref</i>: SharedPreferences - <i>editor</i>: SharedPreferences.Editor - <i>mataPelajaranArrayList</i>: ArrayList<modelMataPelajaran>
<ul style="list-style-type: none"> + <i>controllerMataPelajaran</i>(Context) + <i>viewMataPelajaran</i>() : void + <i>getSessionMataPelajaran</i>() : HashMap<String, String> + <i>setSessionMP</i>() : void + <i>setMataPelajaranArrayList</i>() : void + <i>getMataPelajaranArrayList</i>() : ArrayList<modelMataPelajaran>

Gambar 3.56 Class *controllerMataPelajaran*

9. Class controllerMateriGuru

Class *controllerMateriGuru* digunakan untuk menangani pengambilan materi guru dari server. Materi-materi tersebut akan ditampilkan untuk pengguna sebagai guru, pengguna sebagai siswa, dan pengguna sebagai MGMP.

controllerMateriGuru	
-	modelMateriGuruArrayList: ArrayList<modelMateriGuru> mMateriGuru: modelMateriGuru
+	getModelMateriGuru() : modelMateriGuru setParcelableGetIntentBefore() : void setModelMateriGuru() : void controllerMateriGuru() addMateriGuru() : void addMateriGuruMGMP() : void uploadMateriGuru() : void countMateriguru() : void

Gambar 3.57 Class *controllerMateriGuru*

10. Class *controllerMateriMgmp*

Class controllerMateriMgmp digunakan untuk menangani pengambilan materi MGMP dari *server*. Setelah materi MGMP diambil kemudian ditampilkan pada aplikasi Android untuk pengguna MGMP, pengguna guru, dan pengguna siswa.

controllerMateriMGMP	
-	modelMateriMGMPList: ArrayList<modelMateriMGMP> mMateriMGMP: modelMateriMGMP
+	getModelMateriMGMP() : modelMateriMGMP setParcelableIntentBefore() : void setModelMateriMGMP() : void controllerMateriMGMP() getArrayListModelMateriMGMP() : ArrayList<modelMateriMGMP> addMateriMGMP() : void countMateriMGMP() : void

Gambar 3.58 Class *controllerMateriMGMP*

11. Class *controllerMgmp*

Class controllerMgmp digunakan untuk menangani pengambilan data dari server ketika login MGMP dan *session* ketika pengguna sebagai MGMP membuka aplikasi.

controllerMGMP	
-	pref: SharedPreferences
-	editor: SharedPreferences.Editor
-	context: Context
+	controllerMGMP(Context) : void
+	setSesisonMgmp() : void
+	getSessionMgmp() : HashMap<String, String>
+	setClearSession() : void
+	clearActivities() : void
+	isLoggedIn() : void
+	loginMgmp() : void

Gambar 3.59 Class controllerMGMP

12. Class controllerNilaiUjian

Class controllerNilaiUjian digunakan untuk menangani dalam pengambilan nilai ujian berdasarkan soal yang telah dijawab oleh siswa. Selain itu menangani penilaian ketika pengguna guru ingin menilai soal esai dari seorang siswa.

contollerNilaiUjian	
-	mKelasPembelajaran: List<modelKelasPembelajaran>
-	mKelasUjian: List<modlKelasUjian>
-	mKelompokUjianSiswa: List<modelKelompokUjianSiswa>
-	modelKelasUjian: modelKelasUjian
-	modelKelompokUjianSiswa: modelKelompokUjianSiswa
+	controllerNilaiUjian()
+	getModelKelompokUjianSiswa() : modelKelompokUjianSiswa
+	setParcelableGetIntentBefore() : void
+	getmKelasPembelajaran() : List<modelKelasPembelajaran>
+	getmKelasUjian() : List<modelKelasUjian>
+	setParcelableKelasUjian() : void
+	getModelKelasUjian() : modelKelasUjian
+	getmKelompokUjianSiswa() : List<modelKelompokUjianSiswa>
+	addModelKelasPembelajaran() : void
+	addModelKelasUjian() : void
+	addModelKelasUjianSiswa() : void
+	removeArrayModelKelasUjian() : void
+	removeArrayModelKelompokUjianSiswa() : void
+	getModelKelasPembelajaranVolley() : void
+	getModelListDaftarUjianVolley() : void
+	getModelListUjianSiswaVolley() : void

Gambar 3.60 Class controllerNilaiUjian

13. Class controllerSekolah

Class controllerSekolah digunakan untuk menangani pengambilan data nama sekolah yang kemudian ditampilkan pada aplikasi Android. Pilihan sekolah ini digunakan untuk *login* sebagai guru maupun siswa.

controllerSekolah	
-	modelNamaSekolahArrayList : ArrayList<modelNamaSekolah>
+	controllerSekolah() initModelNamaSekolahArrayList() : void getModelNamaSekolahArrayList() : ArrayList<modelNamaSekolah> addNamaSekolah() : void removeNamaSekolahList() : void getNamaSekolahVolley() : void

Gambar 3.61 Class controllerSekolah

14. Class controllerSiswa

Class controllerSiwa digunakan untuk menangani pengambilan data dari *server* ketika login sebagai siswa. Setelah *login class controllerSiswa* akan menyimpan *session* sebagai pengguna siswa beserta atribut yang lainnya.

controllerSiswa	
-	pref_name: String mInstanceControllerSiswa: controllerSiswa pref: SharedPreferences editor: SharedPreferences.Editor context: Context private_mode: int modelSiswa: modelSiswa
+	getmInstanceContollerSiswa() : controllerSiswa setNewModelSiswa() : void getModelSiswa() : modelSiswa setModelSiswa() : void setSessionSiswa() : void setSessionGuruForSiswa() : void getSessionSiswa() : HashMap<String, String> setClearSession() : void checkLoginSiswa() : boolean isLoggedIn() : boolean clearActivities() : void loginSiswa() : void getDataSiswaVolley() : void getDataGuruForSiswaVolley() : void

Gambar 3.62 Class controllerSiswa

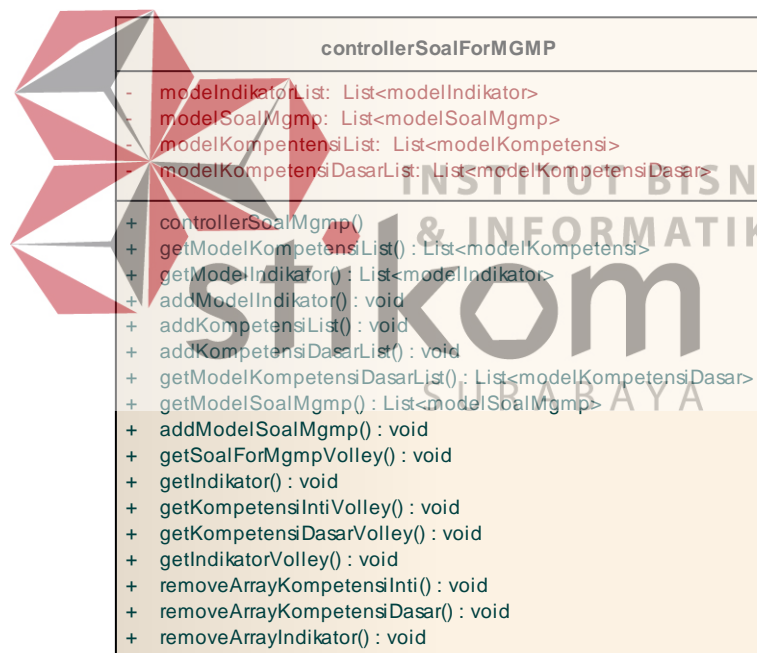
15. Class controllerSoal

Class controllerSoal digunakan untuk menangani pengambilan setiap soal yang ada di ujian. Soal yang ditampilkan pada pengguna guru akan menampilkan jawaban siswa dan jawaban sebenarnya, dan soal yang ditampilkan pada siswa hanya jawaban yang telah dijawab oleh siswa itu sendiri.

Gambar 3.63 Class *controllerSoal*

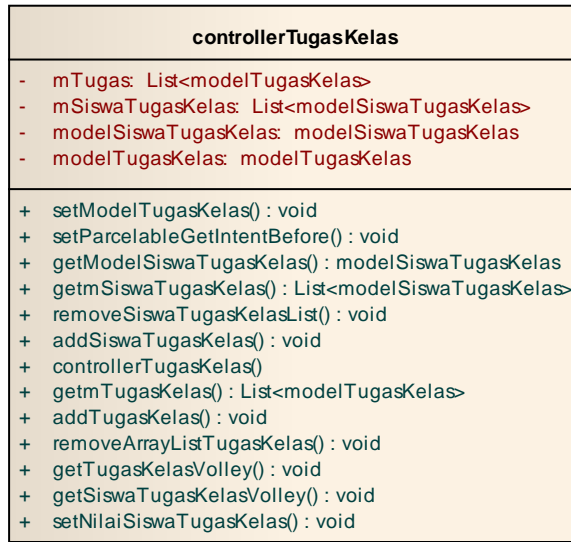
16. Class *contollerSoalForMgmp*

Class controllerSoalForMgmp digunakan untuk menangani dalam mengambil data soal dari *server*. Soal yang ditampilkan pada pengguna MGMP akan menampilkan jawaban yang sebenarnya.

Gambar 3.64 Class *controllerSoalForMGMP*

17. Class *controllerTugasKelas*

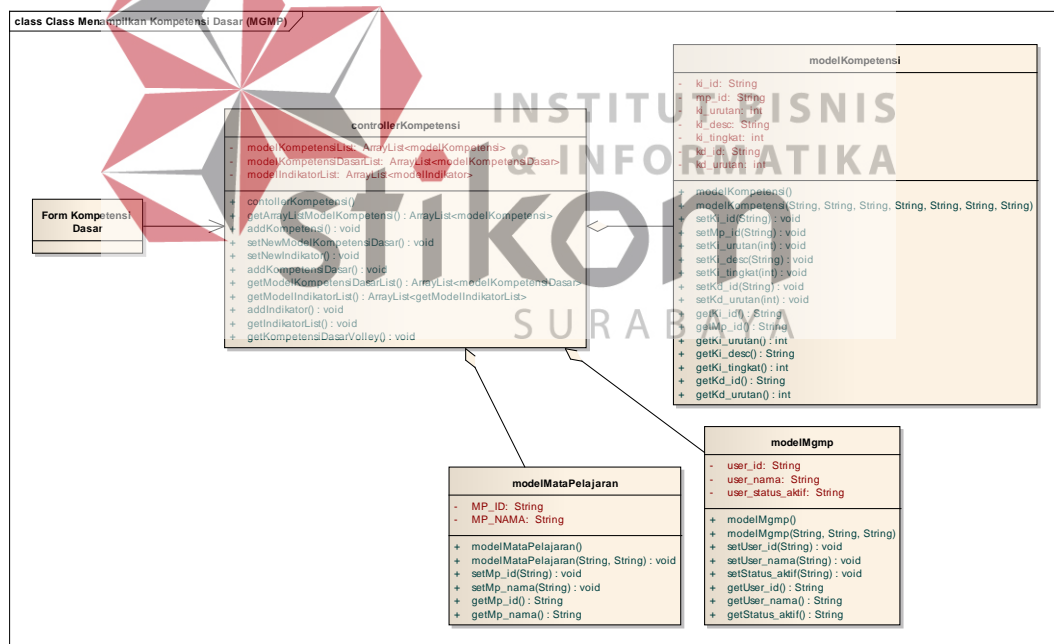
Class controllerTugasKelas digunakan untuk mengambil tugas-tugas berdasarkan kelas pembelajaran dari *server* kemudian ditampilkan dalam bentuk tugas kelas. Tugas kelas ini akan ditampilkan pada guru dan siswa sesuai dengan kelas pembelajaran.



Gambar 3.65 Class ControllerTugasKelas

3.2.7 Relasi Class Diagram dengan Entitas

1. Menampilkan Kompetensi Dasar (MGMP)



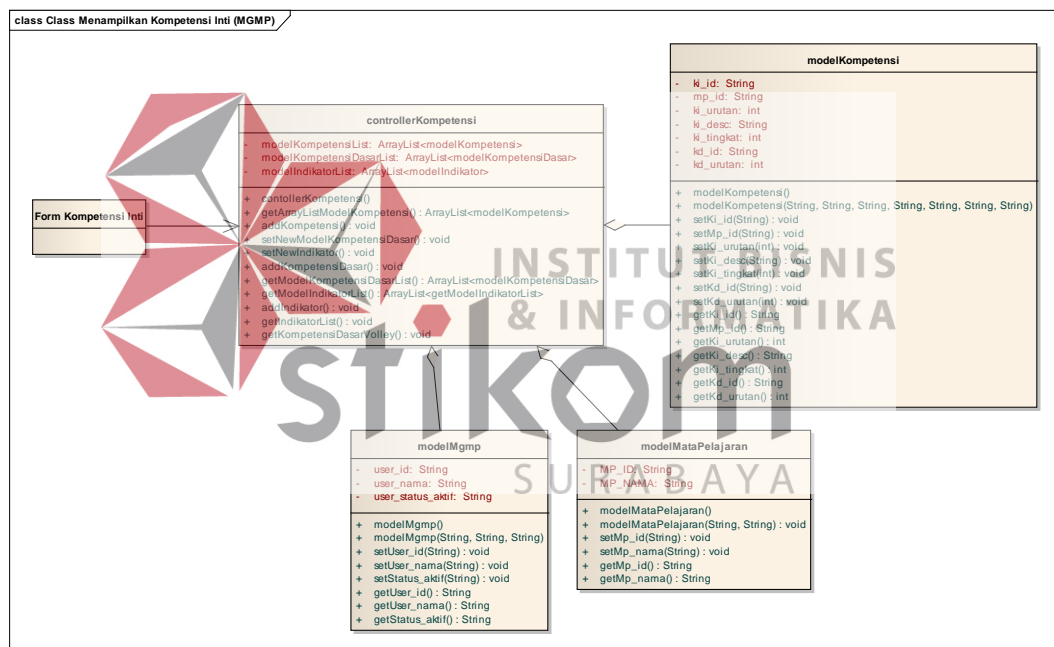
Gambar 3.66 Relasi class diagram menampilkan kompetensi dasar

Relasi class diagram “menampilkan kompetensi dasar” menjelaskan hubungan antara modelKompetensi, modelMgmp, dan modelMataPelajaran yang memiliki hubungan agregasi dengan controllerKompetensi. Semua entitas yang

berhubungan dengan kompetensi dasar sebagai model akan ditampilkan melalui `controllerKompetensi` pada `Form Kompetensi Dasar`.

2. Menampilkan Kompetensi Inti (MGMP)

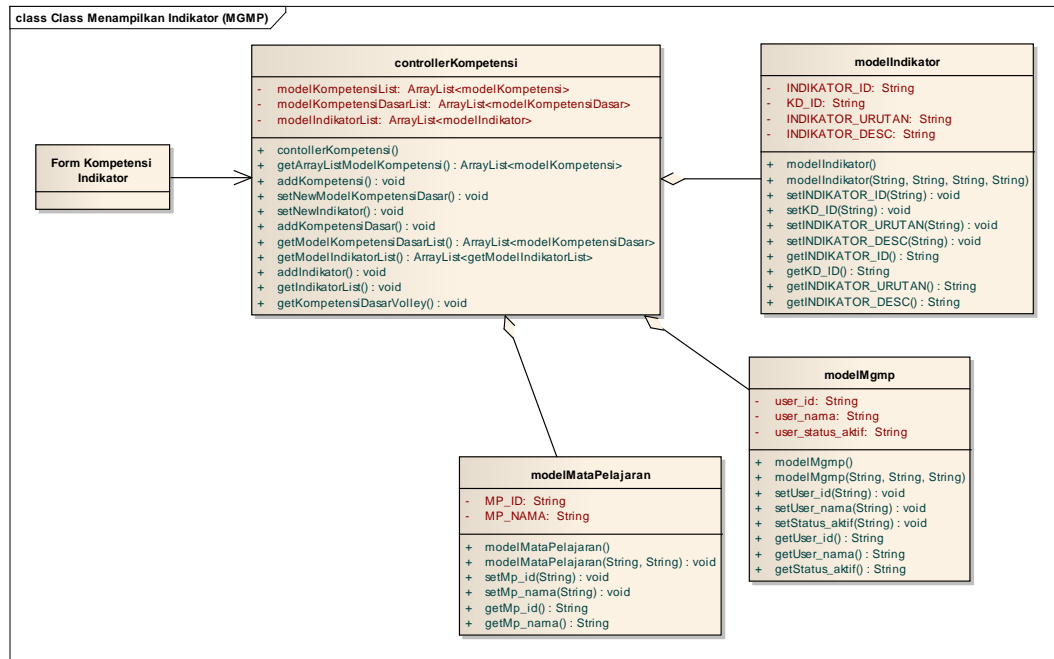
Relasi *class diagram* “menampilkan kompetensi inti” menjelaskan hubungan antara `modelKompetensi`, `modelMataPelajaran`, `modelMgmp` dengan `controllerKompetensi` yang memiliki hubungan agregasi. Semua entitas yang berhubungan dengan kompetensi inti akan ditampilkan melalui `controllerKompetensi` pada `Form Kompetensi Inti`.



Gambar 3.67 Relasi *class diagram* menampilkan kompetensi inti

3. Menampilkan Indikator (MGMP)

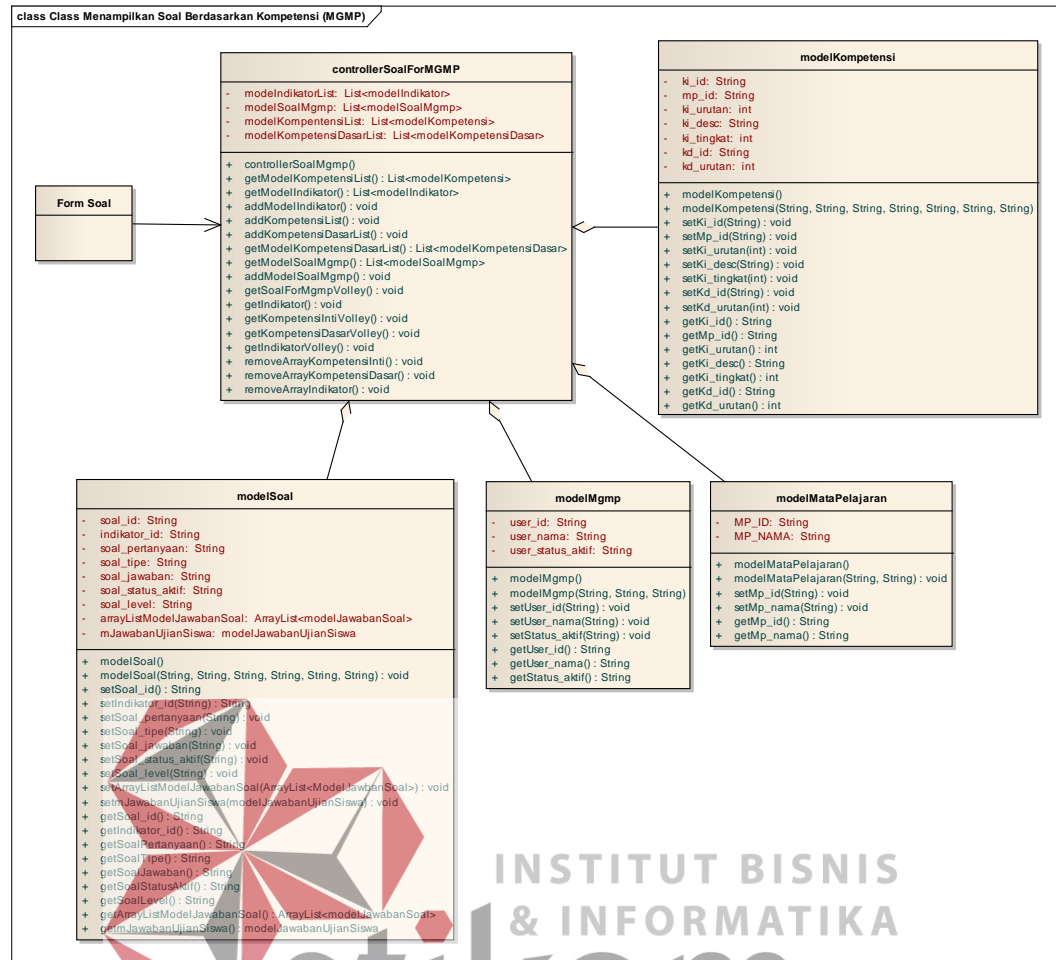
Relasi *class diagram* “menampilkan indikator” menjelaskan hubungan antara `modelIndikator`, `modelMgmp`, `modelMataPelajaran` yang memiliki hubungan agregasi dengan `controllerKompetensi`. Entitas model yang berhubungan dengan indikator akan ditampilkan melalui `controllerIndikator` pada `Form Kompetensi Indikator`.



Gambar 3.68 Relasi class diagram menampilkan indikator

4. Menampilkan Soal Berdasarkan Kompetensi (MGMP)

Relasi class diagram “menampilkan soal berdasarkan kompetensi” menjelaskan tentang hubungan antara modelKompetensi, modelSoal, modelMgmp, modelMataPelajaran yang memiliki hubungan agregasi dengan controllerSoalForMgmp yang nantinya soal-soal akan ditampilkan pada Form Soal.



Gambar 3.69 Relasi class diagram menampilkan soal berdasarkan kompetensi

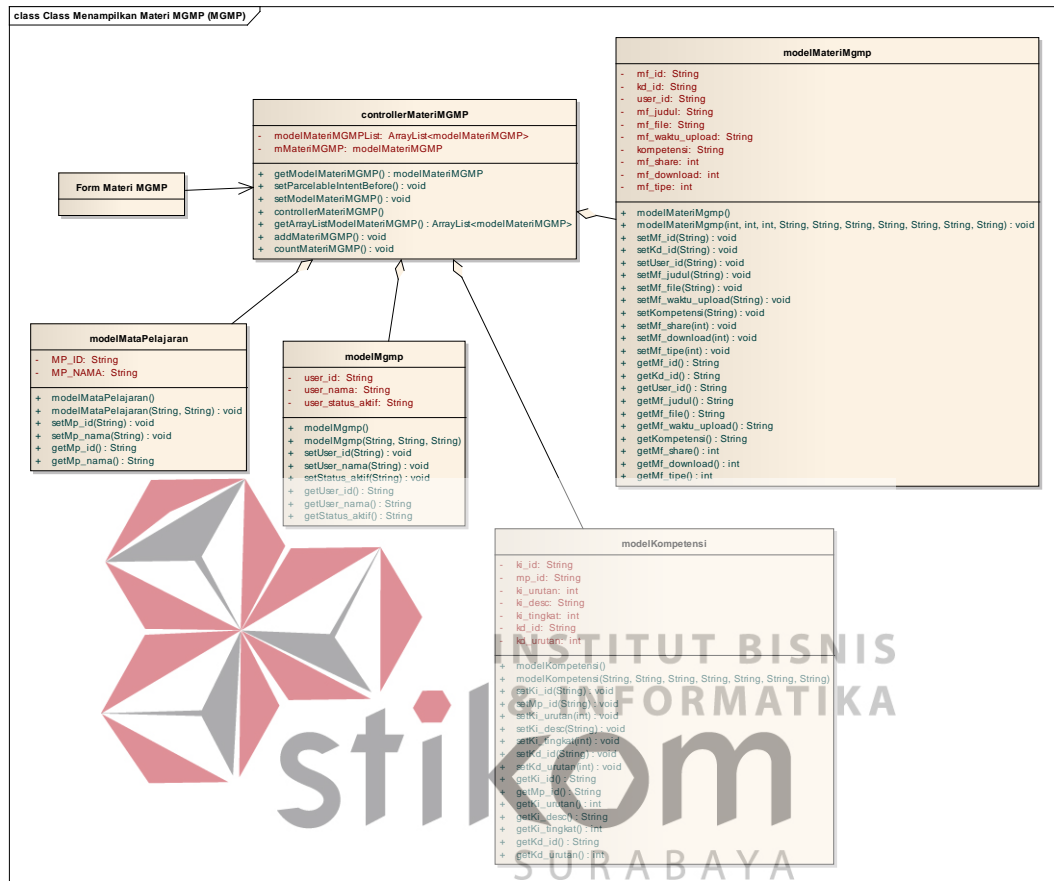
5. Menampilkan Materi MGMP (MGMP)

Relasi class diagram “menampilkan materi MGMP” menjelaskan hubungan antara **modelMateriMgmp**, **modelKompetensi**, **modelMgmp**, **modelMataPelajaran** yang memiliki hubungan agregasi dengan **controllerMateriMgmp**. Materi yang akan ditampilkan ada pada **Form Materi MGMP**.

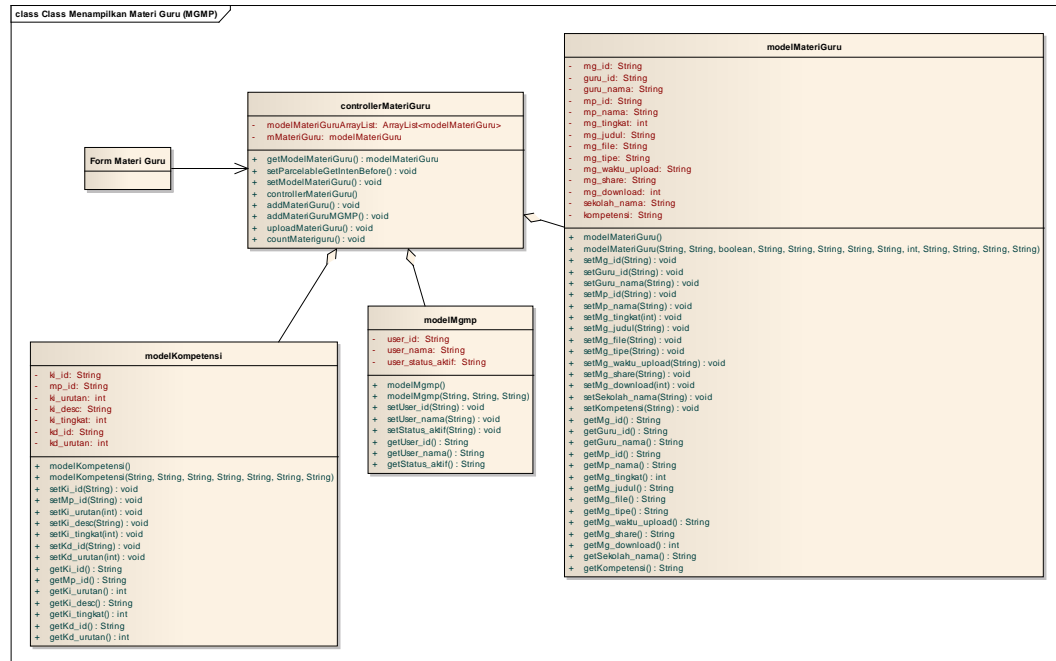
6. Menampilkan Materi Guru (MGMP)

Relasi class diagram “menampilkan materi guru” menjelaskan hubungan antara **modelMateriGuru**, **modelMgmp**, **modelKompetensi** yang memiliki hubungan agregasi dengan **controllerMateriGuru**. Semua entitas yang

berhubungan dengan materi guru akan ditampilkan melalui controllerMateriGuru yang nantinya akan ditampilkan ada pada *Form* Materi Guru. Relasi dapat dilihat di Gambar 3.71.



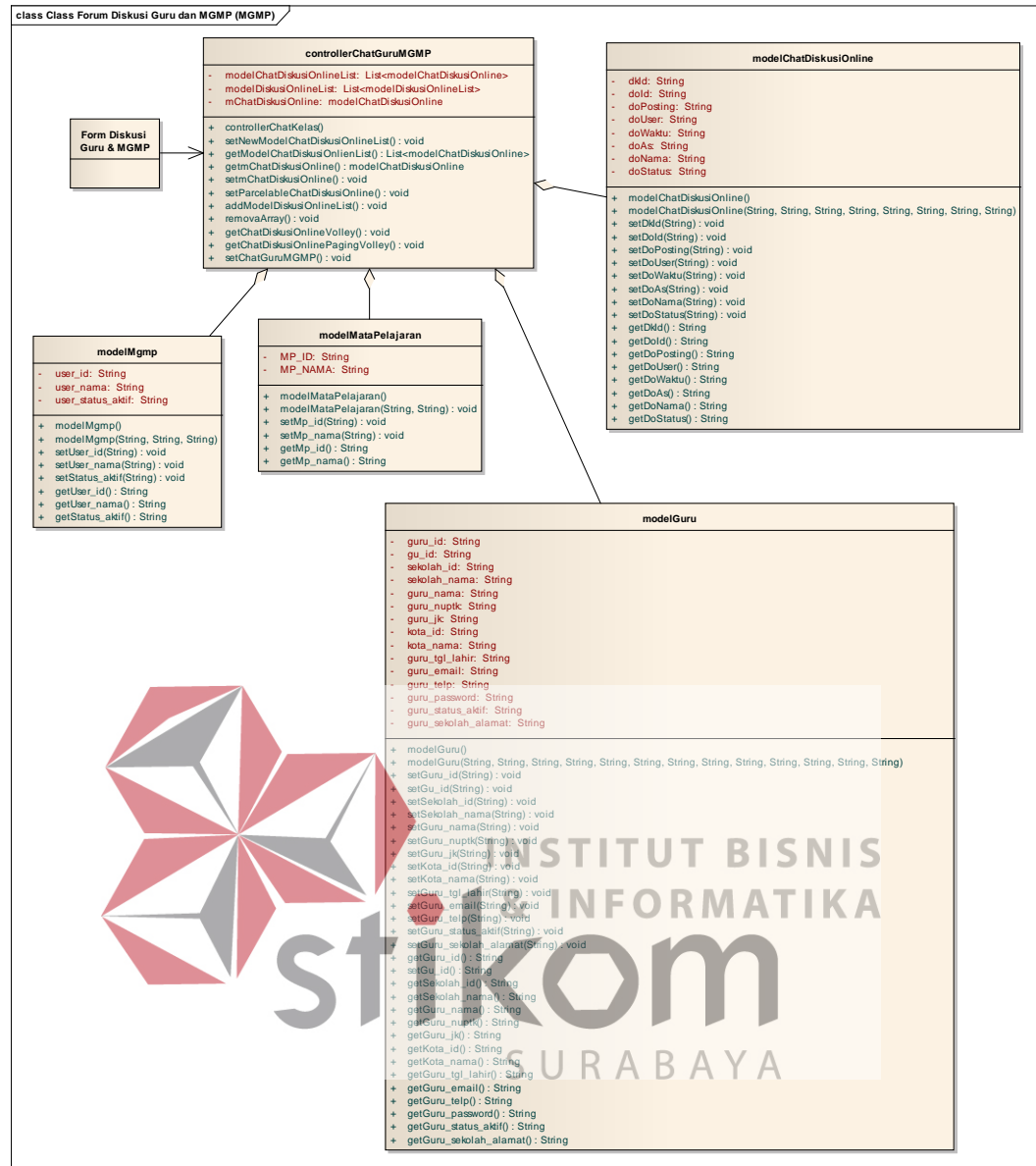
Gambar 3.70 Relasi class diagram menampilkan materi MGMP



Gambar 3.71 Relasi class diagram menampilkan materi guru

7. Forum Diskusi Guru dan MGMP (MGMP)

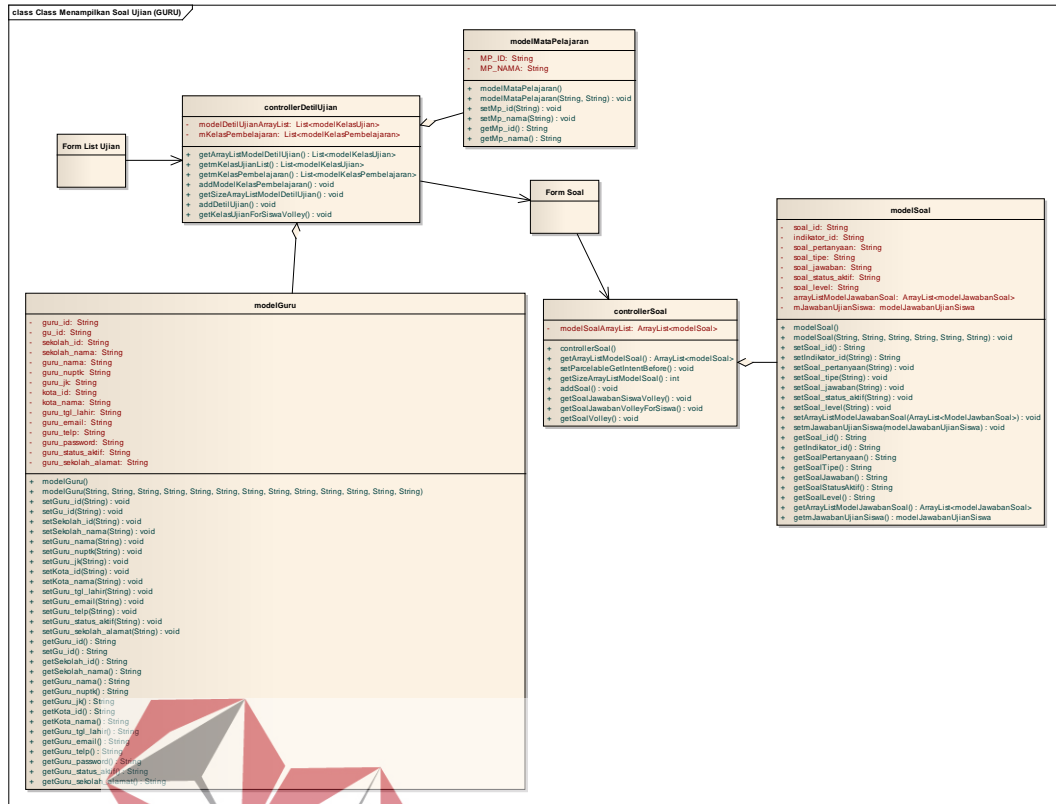
Relasi class diagram “forum diskusi guru dan MGMP” menjelaskan tentang hubungan antara modelChatDiskusiOnline, modelGuru, modelMataPelajaran, modelMgmp yang memiliki hubungan agregasi dengan controllerChatGuruMGMP. Semua entitas model yang berhubungan dengan controllerChatGuruMGMP akan ditampilkan pada Form Diskusi Guru & MGMP.



Gambar 3.72 Relasi *class diagram* forum diskusi guru dan MGMP

8. Menampilkan Soal Ujian (GURU)

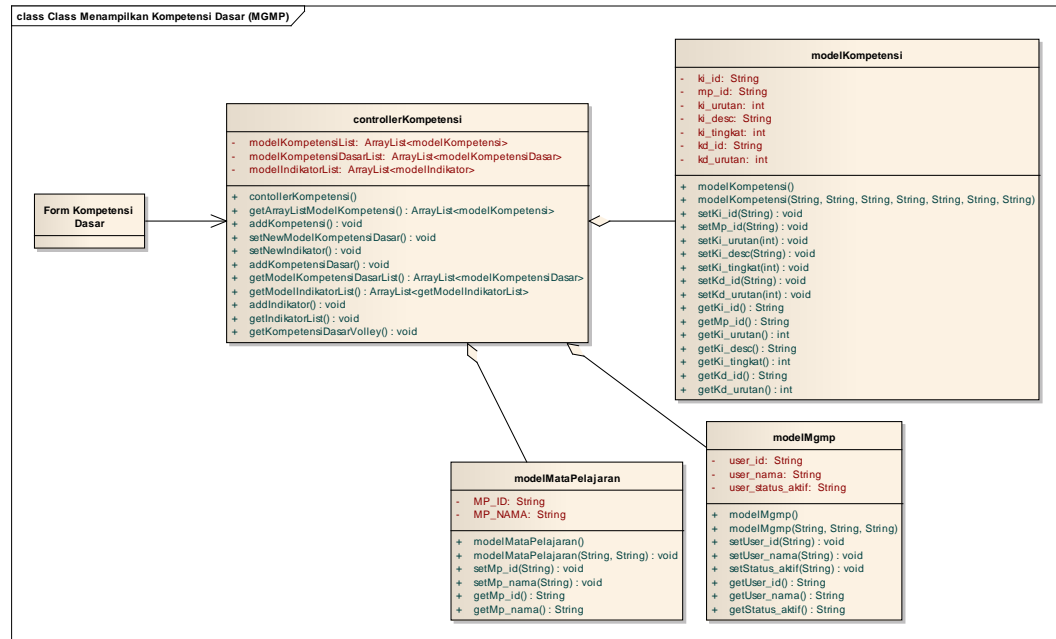
Relasi *class diagram* “menampilkan soal ujian” menjelaskan hubungan modelGuru, modelMataPelajaran yang terhubung secara agregasi controllerDetilUjian untuk menampilkan data pada Form Lihat Ujian. Form Soal akan terhubung asosiatif pada controllerDetilUjian sebagai parameter untuk menampilkan soal pada controllerSoal melalui modelSoal yang terhubung secara agregasi.



Gambar 3.73 Relasi class diagram menampilkan soal ujian

9. Menampilkan Kompetensi Dasar (GURU)

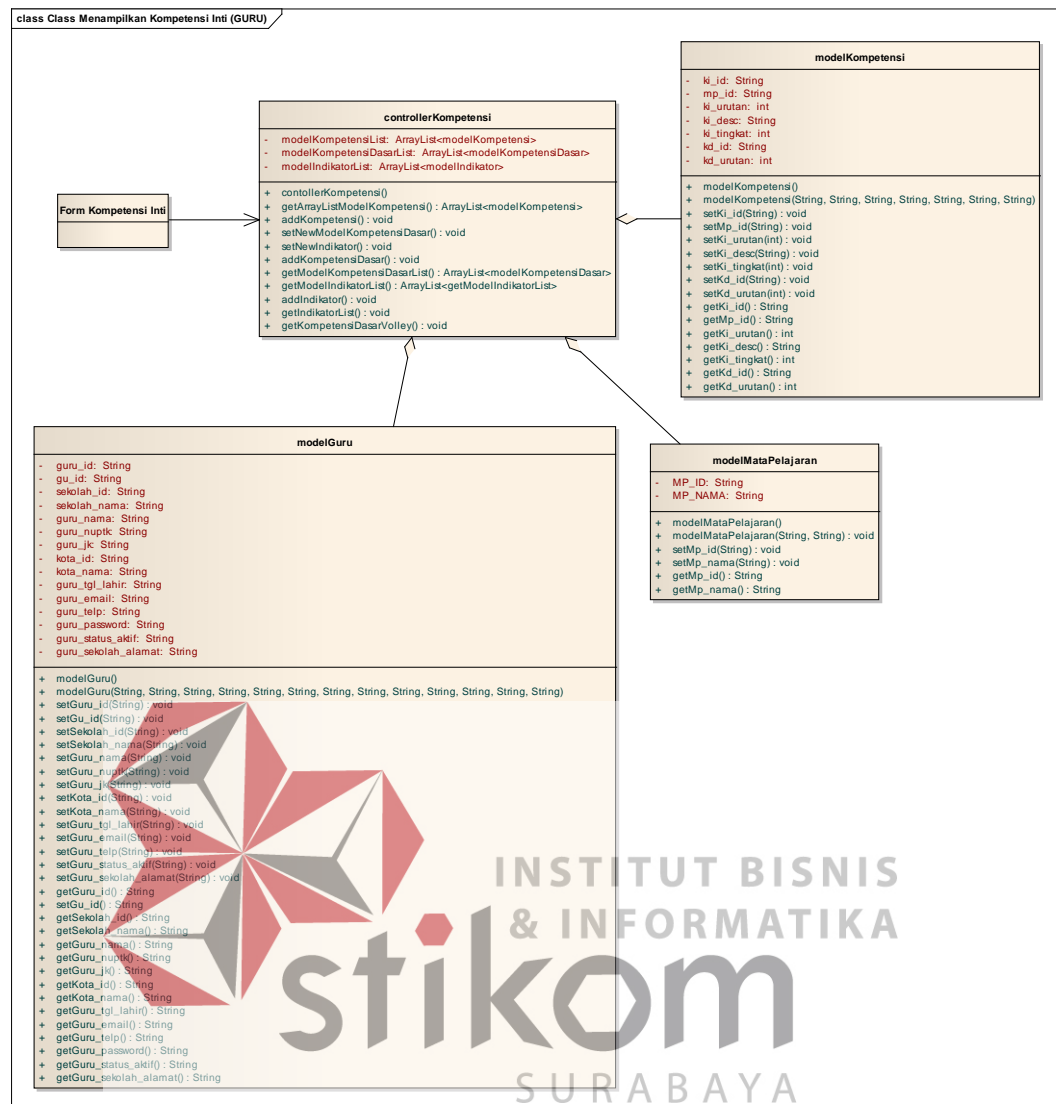
Relasi class diagram “menampilkan kompetensi dasar” menjelaskan tentang hubungan antara modelMataPelajaran, modelMgmp, modelKompetensi yang terhubung secara agregasi dengan controllerKompetensi.



Gambar 3.74 Relasi *class diagram* menampilkan kompetensi dasar

10. Menampilkan Kompetensi Inti (Guru)

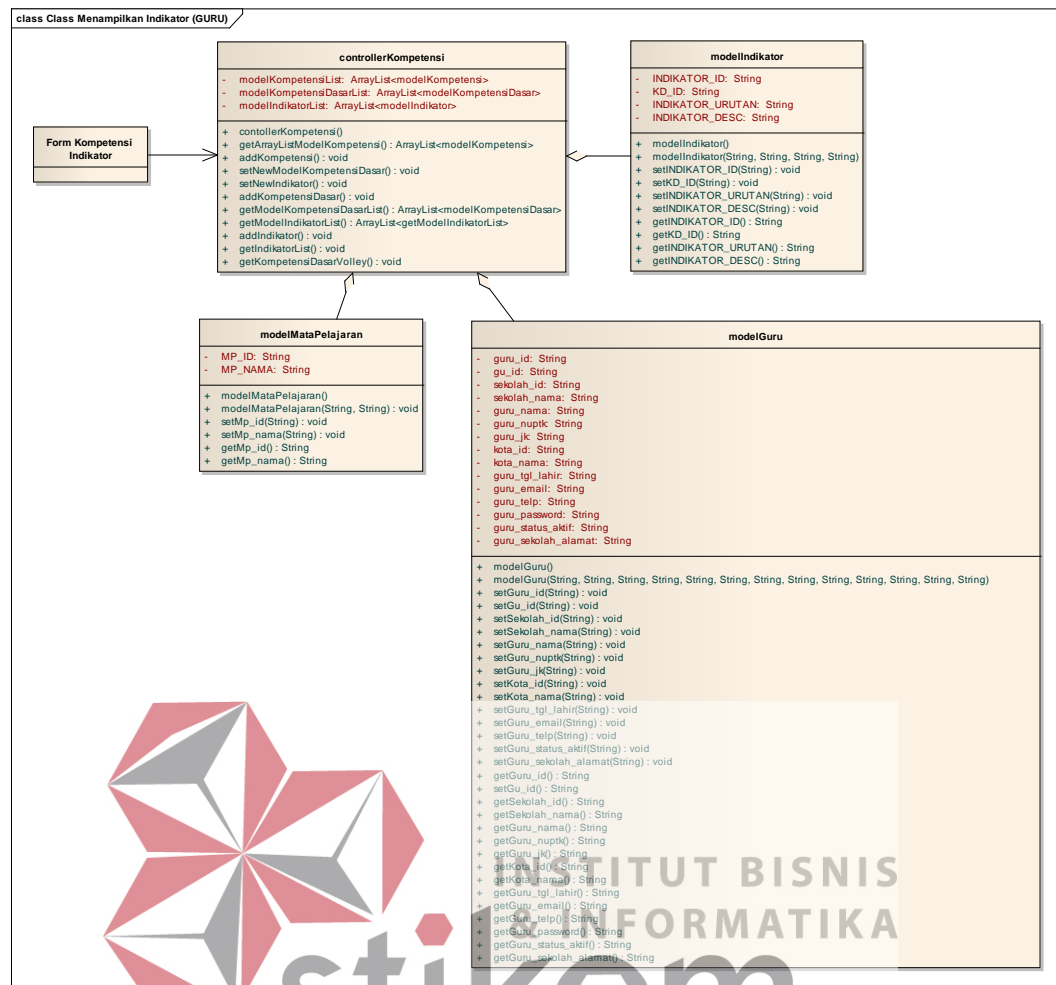
Relasi *class diagram* “menampilkan kompetensi inti” menjelaskan tentang hubungan modelKompetensi, modelMataPelajaran, modelGuru yang berhubungan secara agregasi dengan controllerKompetensi. Kompetensi akan ditampilkan pada Form Kompetensi Inti.



Gambar 3.75 Relasi *class diagram* menampilkan kompetensi inti

11. Menampilkan Indikator (Guru)

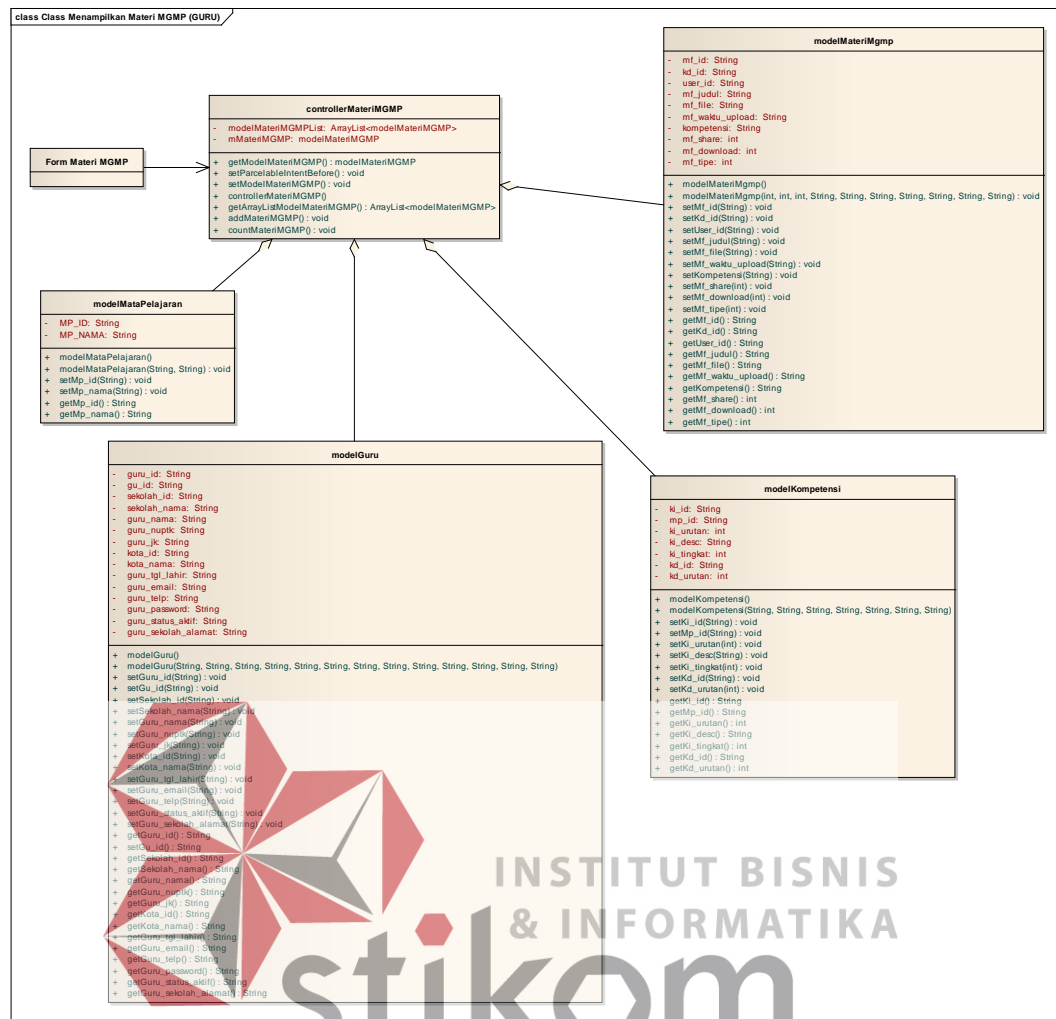
Relasi *class diagram* “menampilkan indikator” menjelaskan hubungan `modelIndikator`, `modelGuru`, `modelMataPelajaran` yang terhubung secara agregasi dengan `controllerKompetensi`. Indikator akan ditampilkan pada `Form Kompetensi Indikator`.



Gambar 3.76 Relasi *class diagram* menampilkan indikator

12. Menampilkan Materi MGMP (Guru)

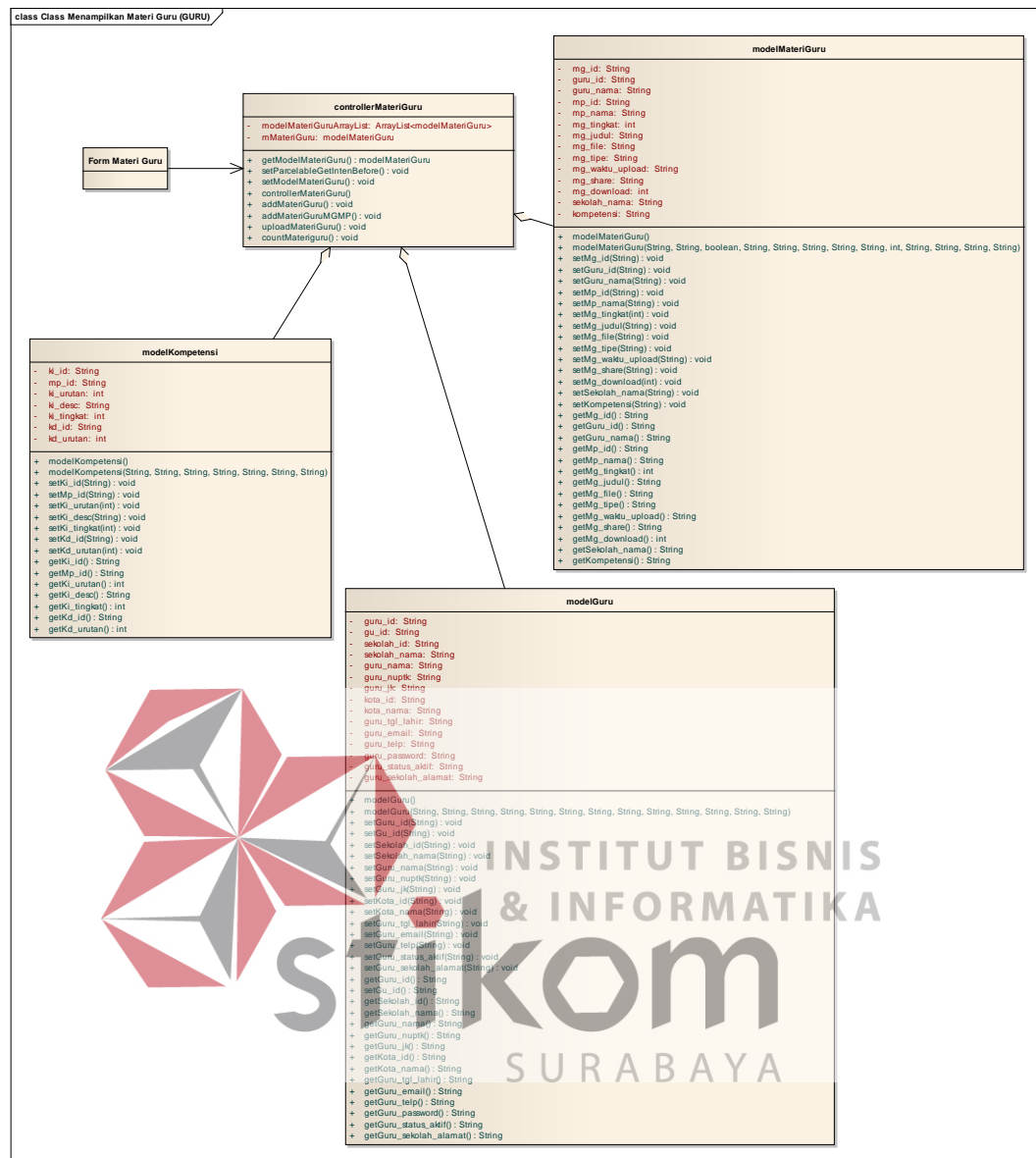
Relasi *class diagram* “menampilkan materi MGMP” menjelaskan hubungan modelMateriMgmp, modelKompetensi, modelGuru, modelMataPelajaran dengan controllerMateriMgmp secara agregasi. Materi MGMP akan ditampilkan pada Form Materi MGMP.



Gambar 3.77 Relasi *class diagram* menampilkan materi MGMP

13. Menampilkan Materi Guru (Guru)

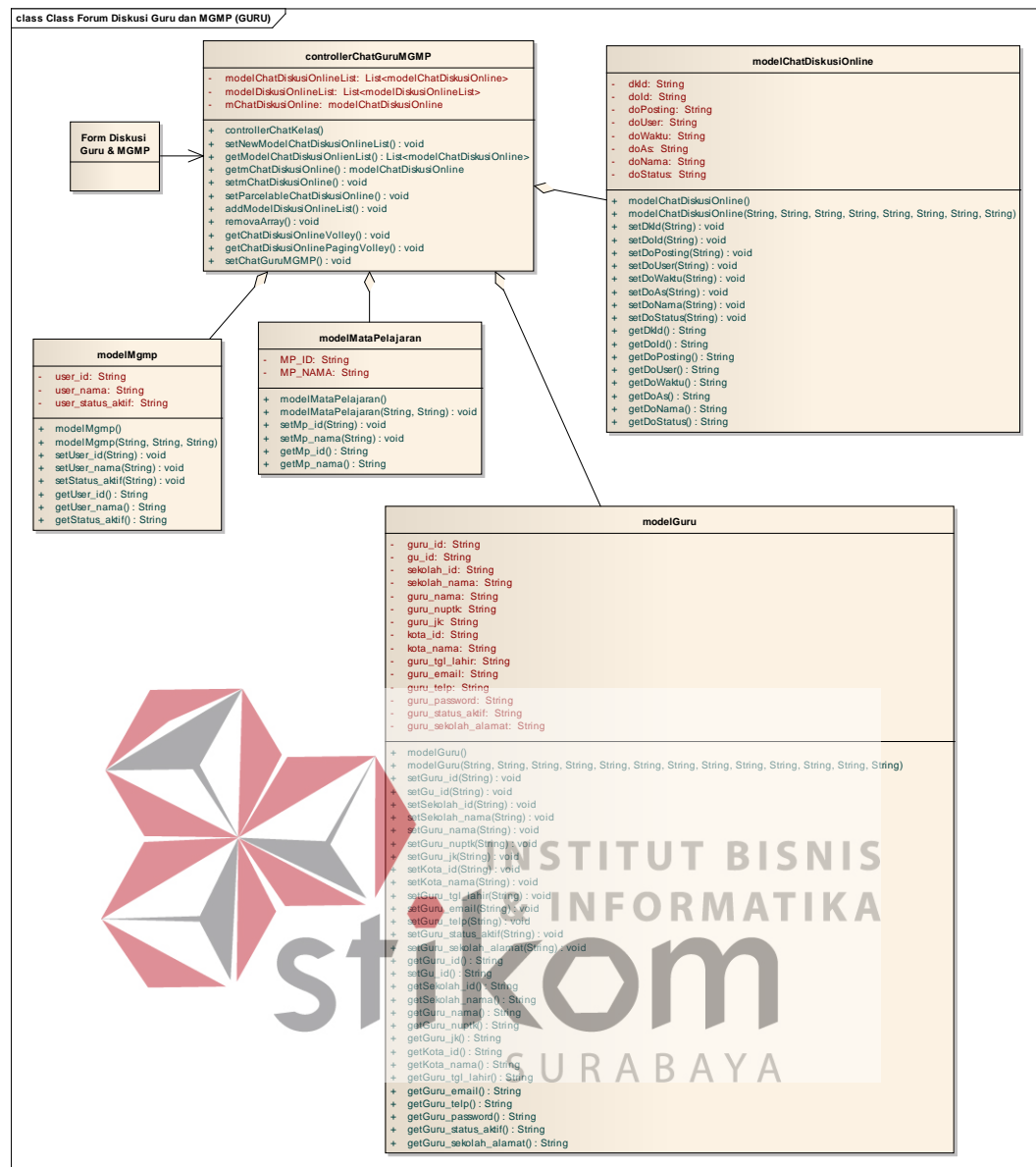
Relasi *class diagram* “menampilkan materi guru” menjelaskan tentang hubungan antara modelMateriGuru, modelGuru, modelKompetensi dengan controllerMateriGuru secara agregasi. Materi guru akan ditampilkan pada Form Materi Guru.



Gambar 3.78 Relasi *class diagram* menampilkan materi guru

14. Forum Diskusi Guru dan MGMP (Guru)

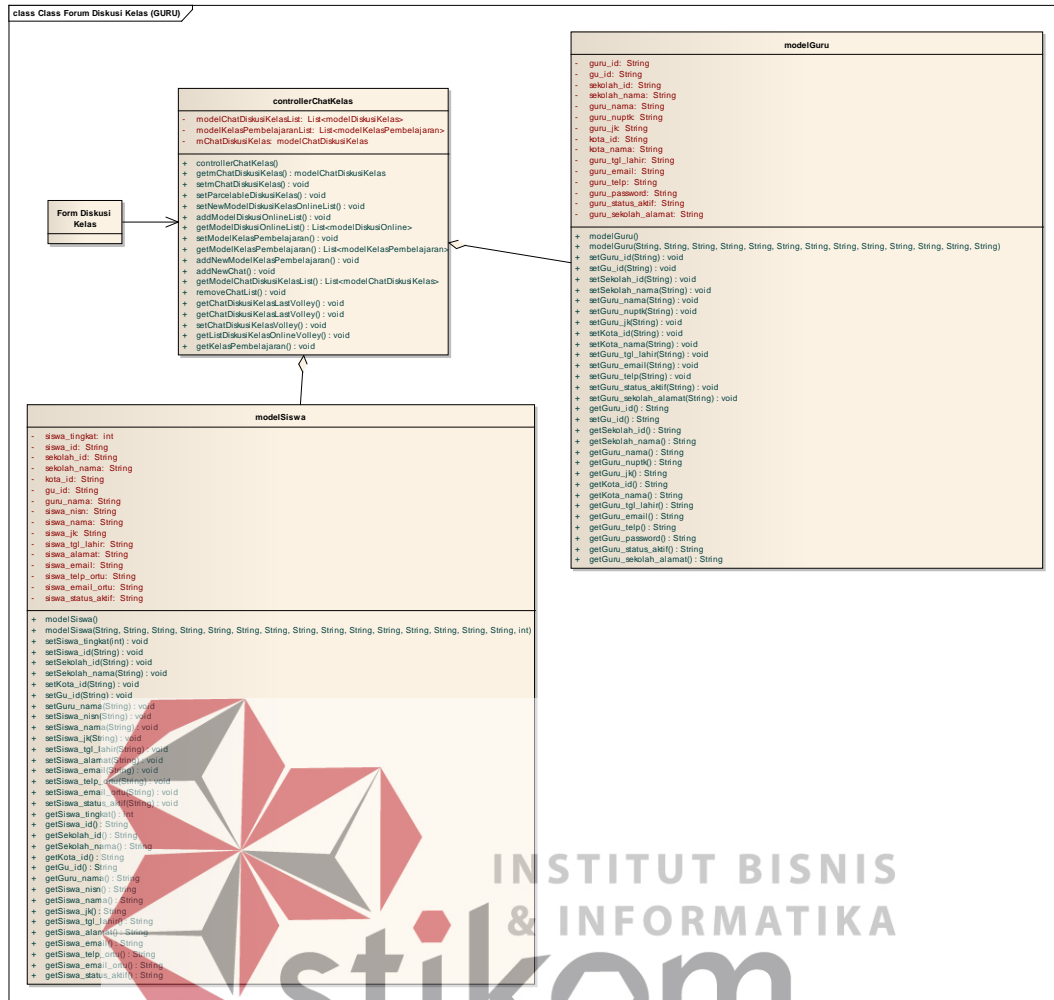
Relasi *class diagram* “forum diskusi guru dan MGMP” menjelaskan hubungan antara modelChatDiskusiOnline, modelGuru, modelMataPelajaran, modelMgmp yang terhubung dengan controllerChatGuruMgmp secara agregasi. Diskusi akan ditampilkan pada Form Diskusi Guru dan MGMP.



Gambar 3.79 Relasi class diagram forum diskusi guru dan MGMP

15. Forum Diskusi Kelas (Guru)

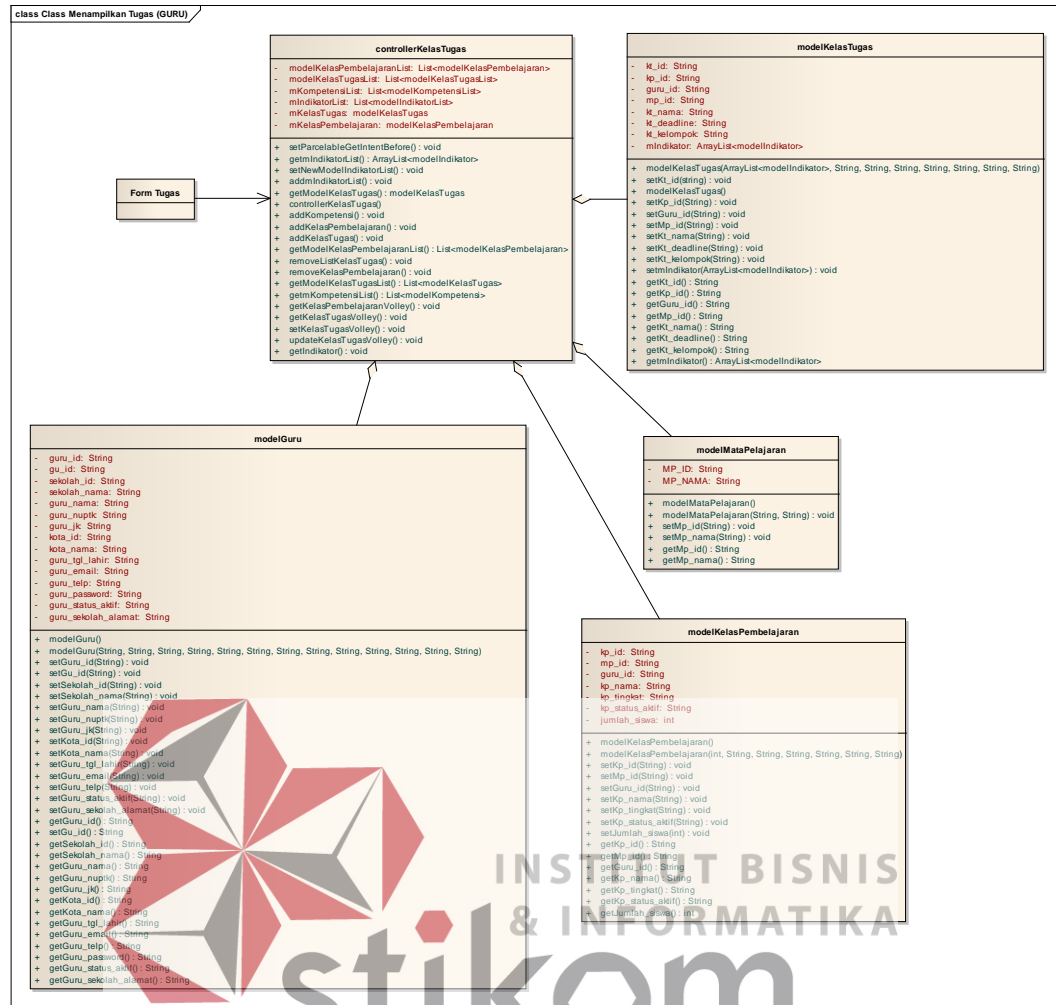
Relasi class diagram “forum diskusi kelas” menjelaskan tentang hubungan antara modelGuru, modelSiswa yang dihubungkan dengan controllerChatKelas secara agregasi. Forum akan ditampilkan pada *Form* Diskusi Kelas.



Gambar 3.80 Relasi class diagram forum diskusi kelas

16. Menampilkan Tugas (Guru)

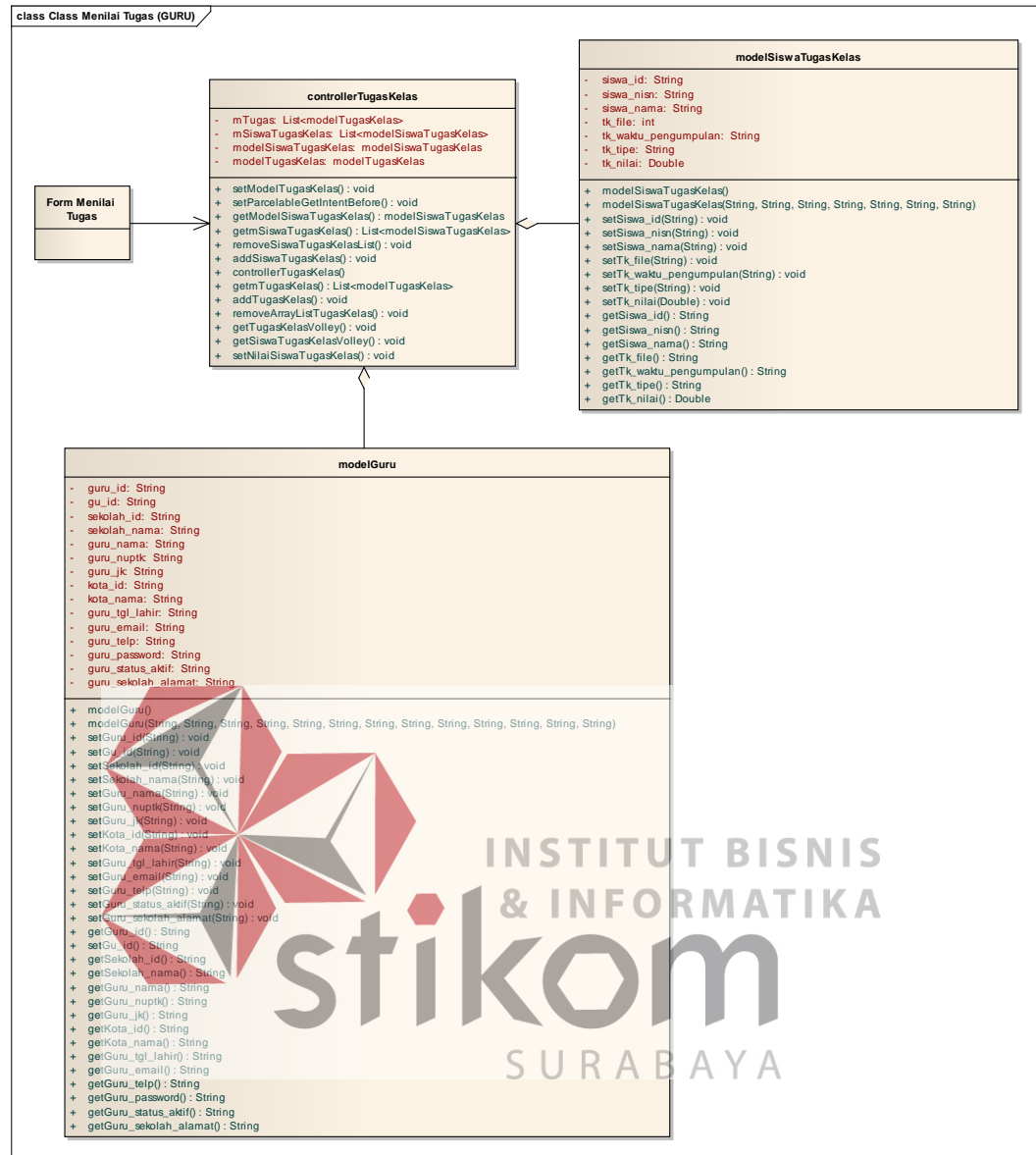
Relasi class diagram “menampilkan tugas” menjelaskan hubungan antara modelKelasTugas, modelMataPelajaran, modelKelasPembelajaran, modelGuru yang dihubungkan dengan controllerTugas secara agregasi. Tugas akan ditampilkan pada Form Tugas.



Gambar 3.81 Relasi class diagram menampilkan tugas

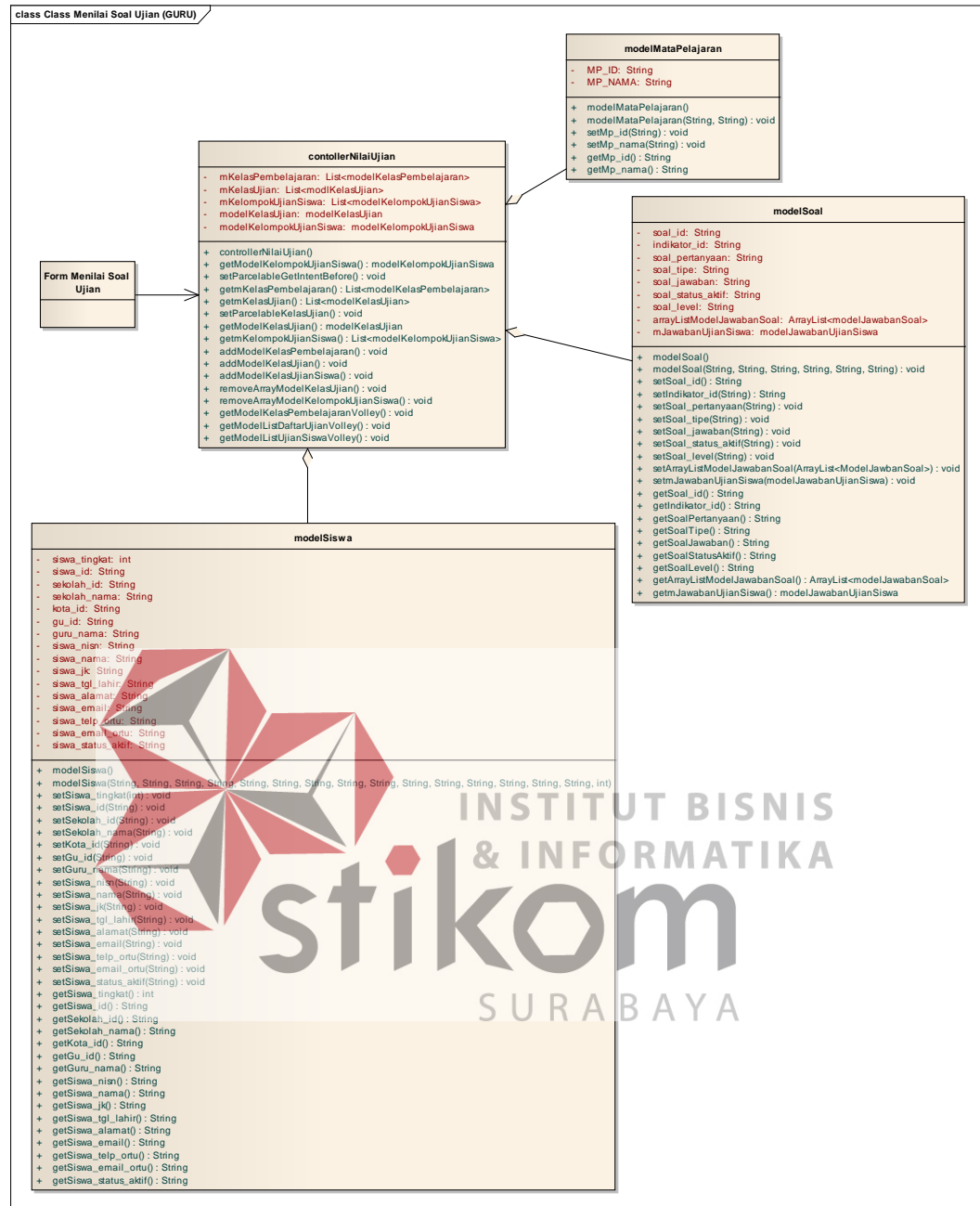
17. Menilai Tugas (Guru)

Relasi class diagram “menilai tugas” menjelaskan hubungan antara modelSiswaTugasKelas, modelGuru, dan controllerTugasKelas yang terhubung secara agrerasi. Menilai tugas akan dilakukan pada Form Menilai Tugas.

Gambar 3.82 Relasi *class diagram* menilai tugas

18. Menilai Soal Ujian (Guru)

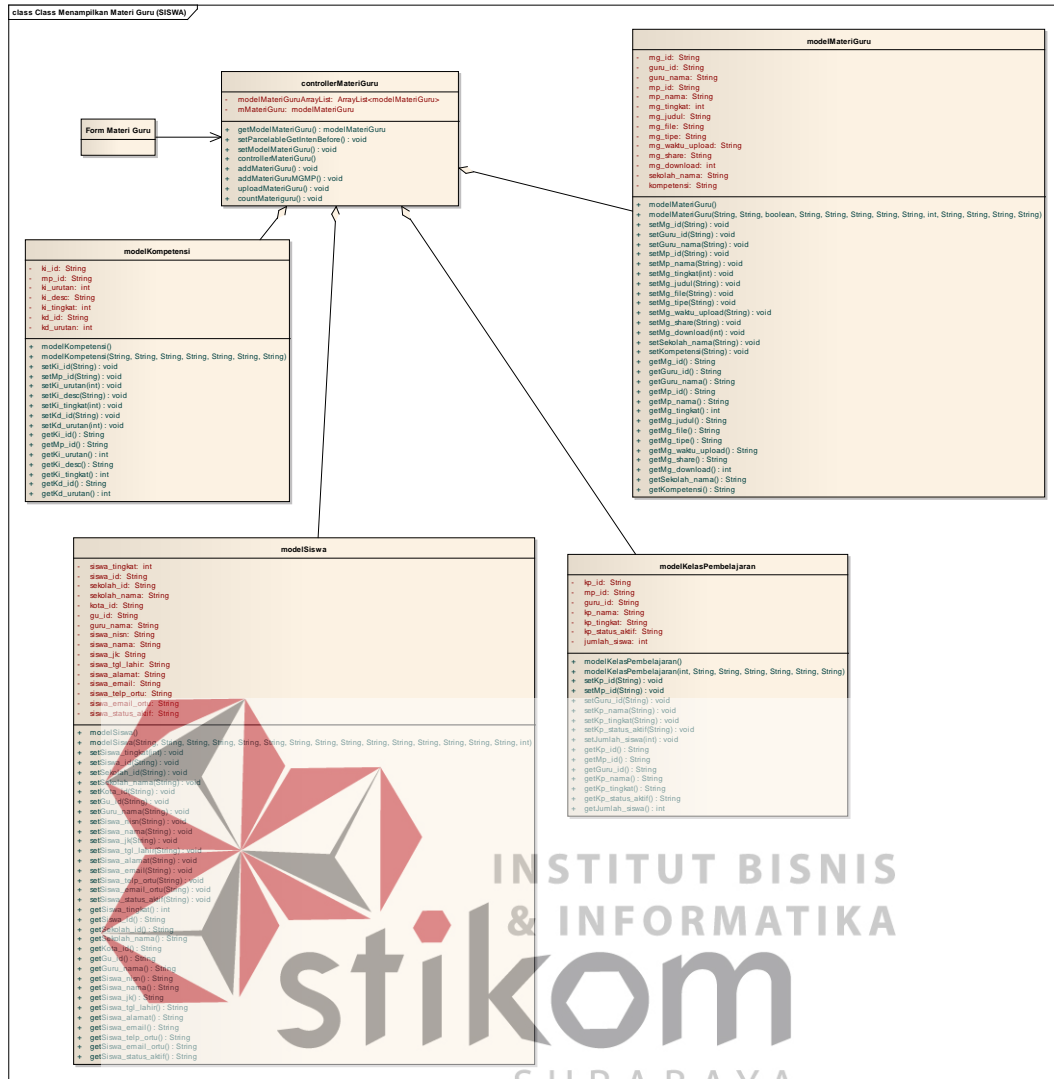
Relasi *class diagram* “menilai soal ujian” memiliki hubungan dengan *modelSoal*, *modelMataPelajaran*, *modelSiswa* yang berhubungan dengan *controllerNilaiUjian* secara agregasi. Menilai soal ujian aka dilakukan pada *Form Menilai Soal Ujian*.



Gambar 3.83 Relasi class diagram menilai soal ujian

19. Menampilkan Materi MGMP (Siswa)

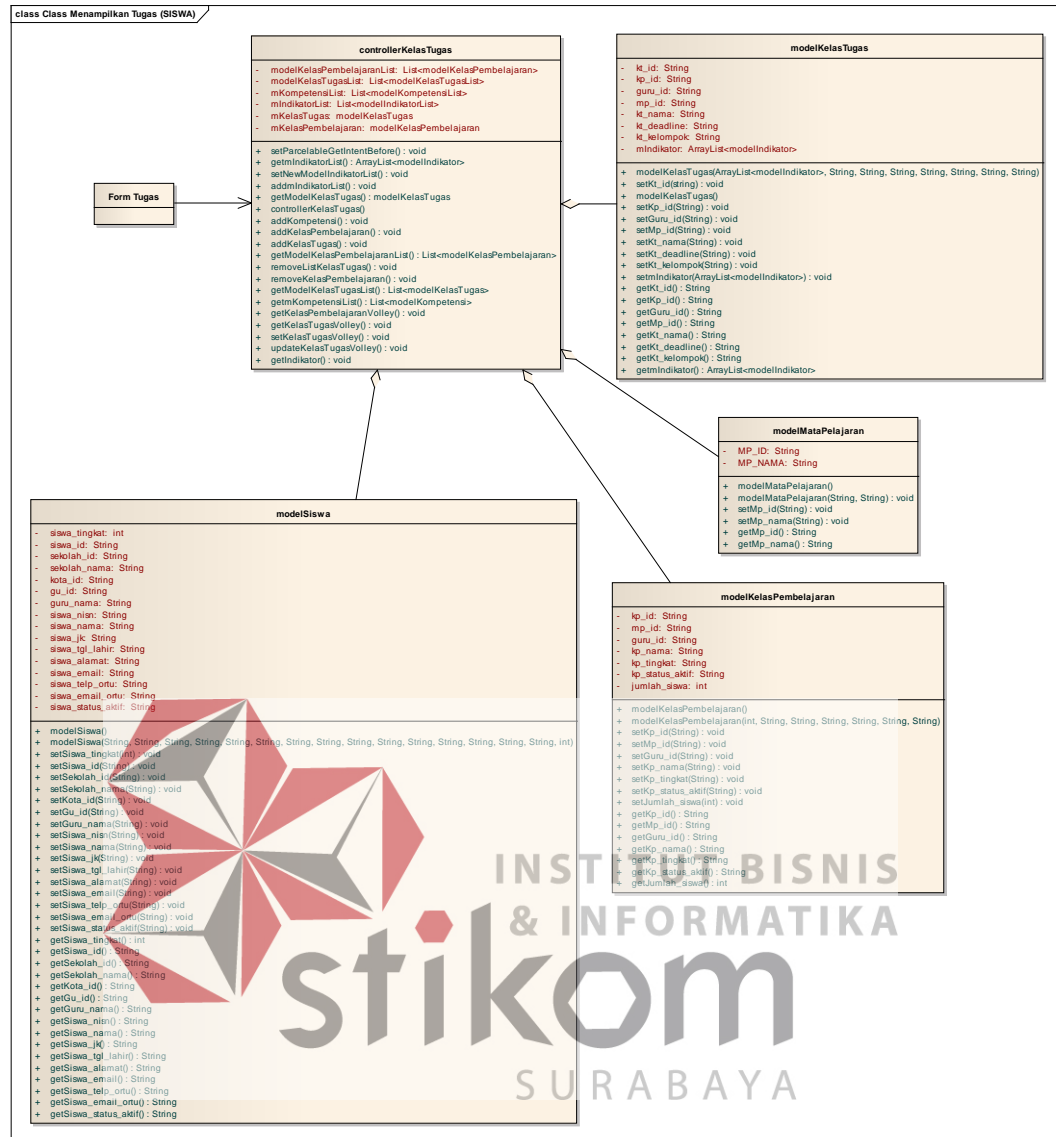
Relasi class diagram “materi MGMP” menjelaskan hubungan dengan modelMateriMgmp, modelKompetensi, modelSiswa, modelMataPelajaran, yang berhubungan dengan controllerMateriMGMP. Materi MGMP akan ditampilkan ketika Form Materi MGMP dibuka.



Gambar 3.85 Relasi class diagram relasi *class diagram* menampilkan materi guru

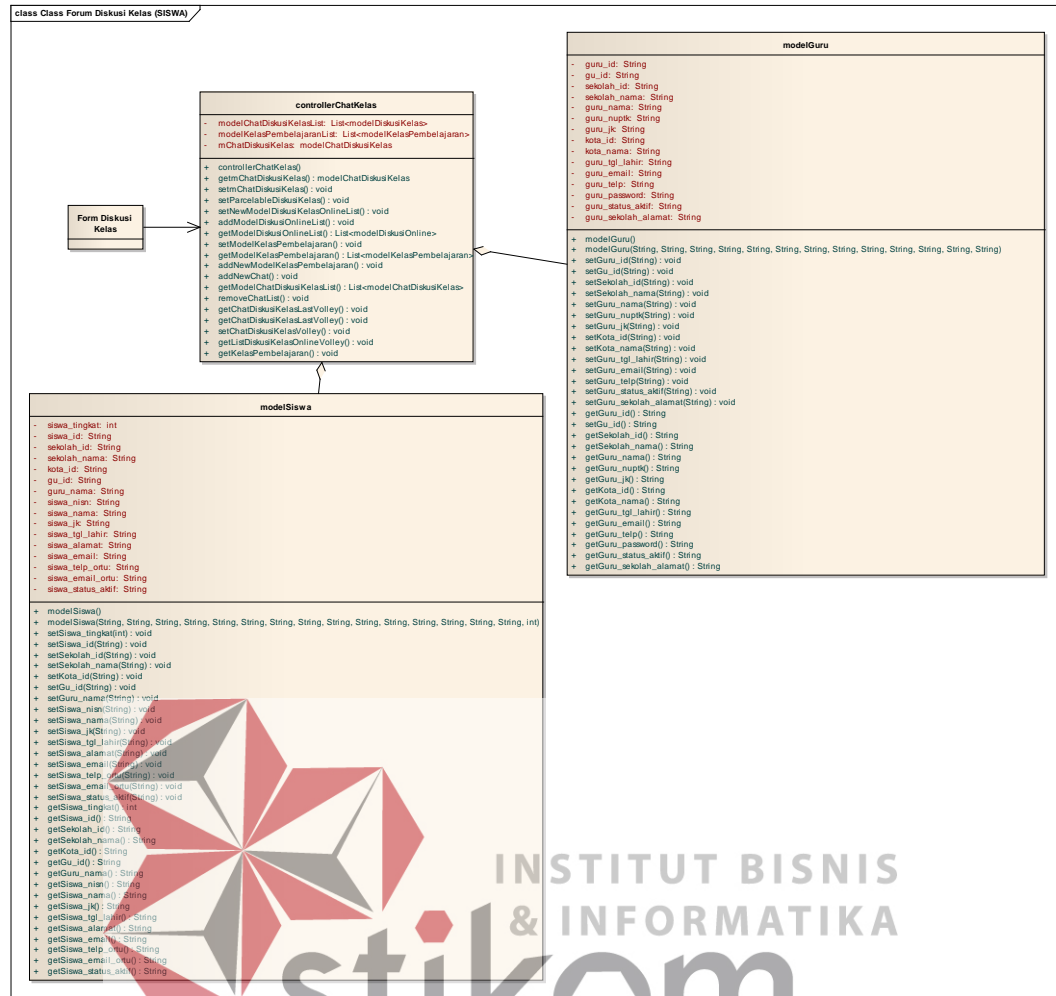
21. Menampilkan Tugas (Siswa)

Relasi *class diagram* “menampilkan tugas” menjelaskan hubungan modelKelasTugas, modelMataPelajaran, modelKelasPembelajaran, modelSiswa yang berhubungan dengan controllerKelasTugas secara agregasi. Tugas akan ditampilkan pada Form Tugas.

Gambar 3.86 Relasi *class diagram* menampilkan tugas

22. Forum Diskusi Kelas (Siswa)

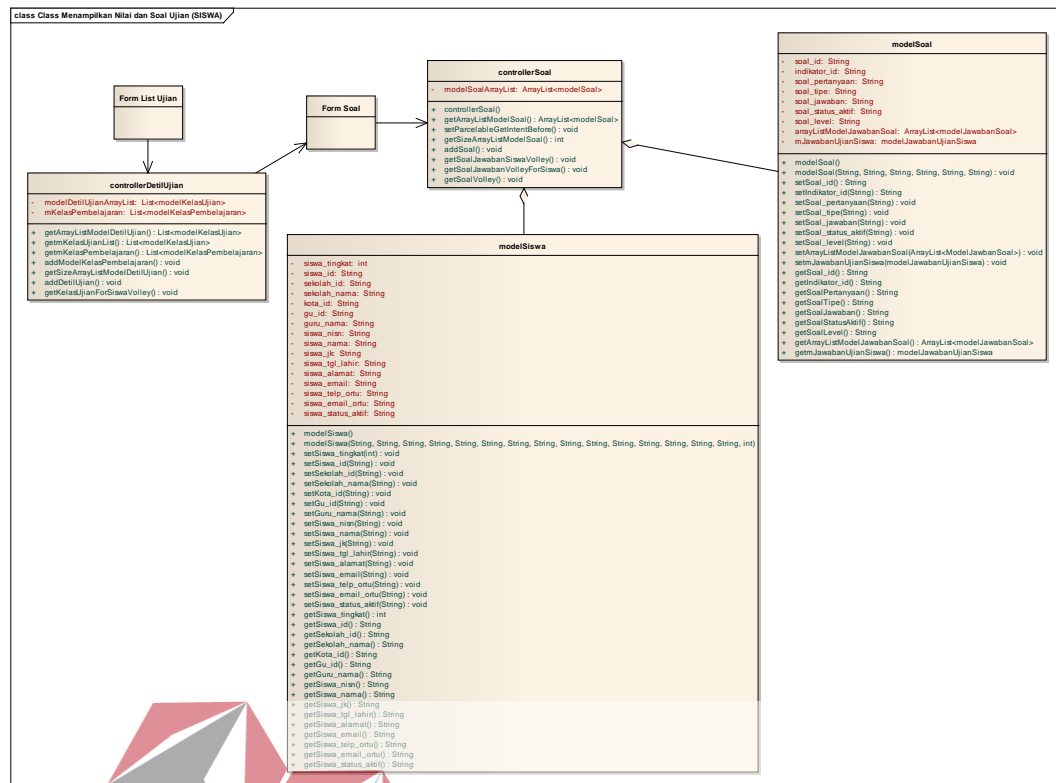
Relasi *class diagram* “forum diskusi kelas” menjelaskan hubungan modelGuru, modelSiswa, dan controllerChatKelas yang dihubungkan secara agregasi. Forum akan ditampilkan pada Form Kelas Diskusi.



Gambar 3.87 Relasi class diagram forum diskusi kelas

23. Menampilkan Nilai dan Soal Ujian (Siswa)

Relasi class diagram “menampilkan nilai dan soal ujian” menjelaskan hubungan antara modelSiswa, modelSoal, yang terhubung secara agregasi dengan controllerSoal. Form yang ditampilkan pertama kali yaitu Form List Ujian kemudian menuju Form Soal melalui controllerDetailUjian.

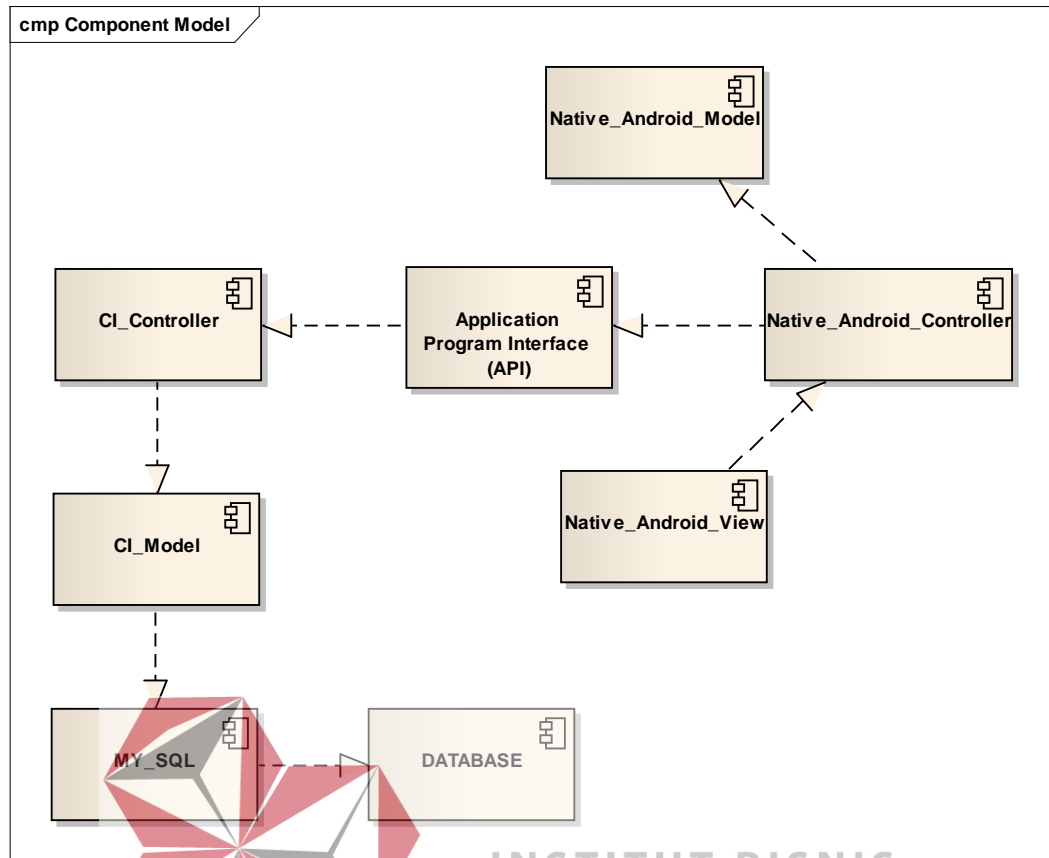


Gambar 3.88 Relasi *class diagram* menampilkan nilai dan soal ujian

3.2.8 Diagram Komponen

Diagram komponen menjelaskan hubungan dari kode-kode pemrograman yang akan dibangun. Garis-garis yang berhubungan akan menunjukkan dependensi komponen tersebut.

Halaman web yang ditandai dengan *CI_View* langsung berkomunikasi pada server melalui *controller* yang ditandai dengan *CI_Controller*. Komponen *controller* merupakan sekumpulan proses-proses yang dibutuhkan untuk manipulasi data dibantu dengan model yang ditandai dengan *CI_Model* untuk memasukkan data ke dalam *database*.

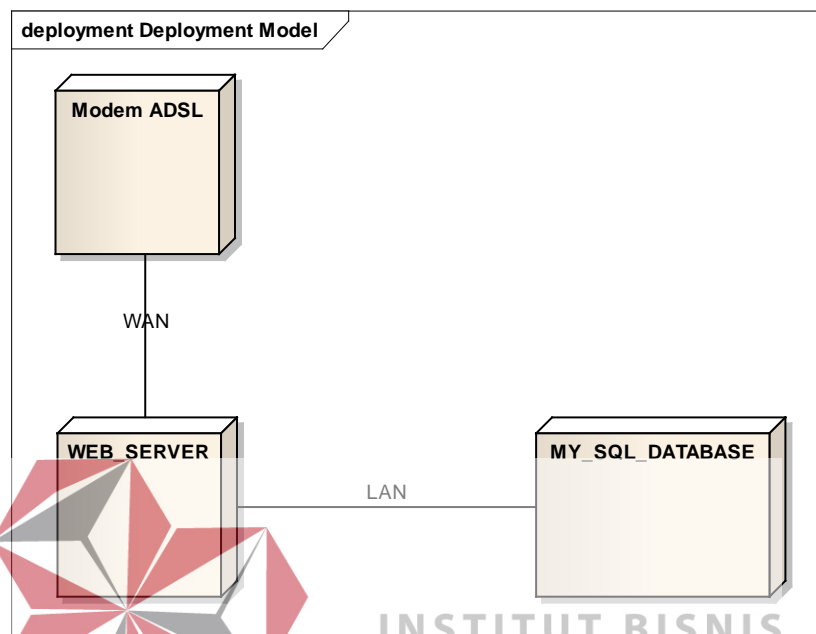


Gambar 3.89 Diagram Komponen

Halaman pada android yang ditandai dengan `Native_Android_view` berkomunikasi dengan controller yang ada pada android. Controller pada Android ditandai dengan `Native_Android_Controller` yang isinya proses-proses fungsi untuk manipulasi data yang disiapkan dikirim pada server. Manipulasi data pada controller tentunya dibantu dengan model sebagai atributnya. Aplikasi pada android dapat berkomunikasi dengan server melalui sebuah perantara yang disebut dengan `Application Program Interface (API)` dimana API tersebut langsung berinteraksi dengan `CI_Controller`. Setelah itu `CI_Controller` akan berfungsi sebagaimana mestinya untuk menyimpan data pada *database*.

3.2.9 Diagram *Deployment*

Diagram *deployment* berfungsi untuk menggambarkan peralatan-peralatan dan prosesor-prosesor yang saling berhubungan.



Gambar 3.90 Diagram *Deployment*

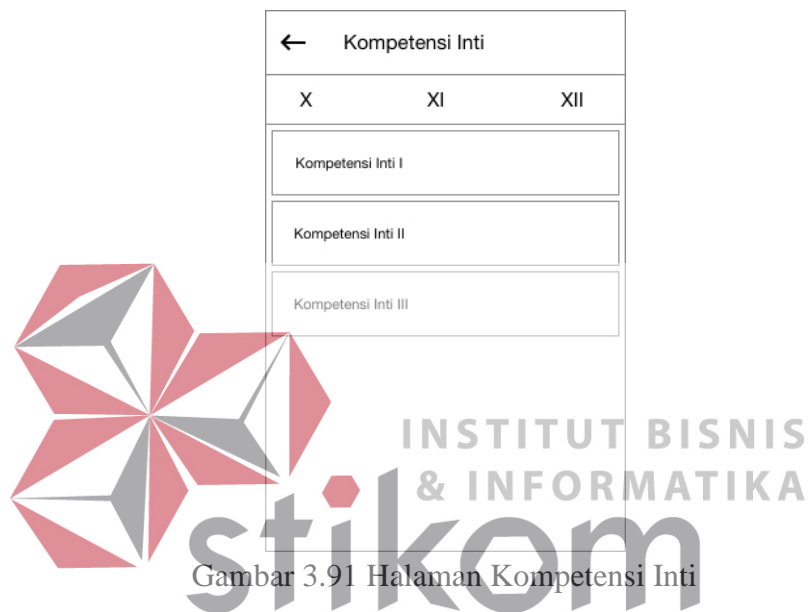
Diagram *deployment* diatas dapat dijelaskan bahwa kode pada *server side* dipasang pada *web server*. *Web server* tersebut terhubung dengan internet WAN. *Web server* juga terhubung pada *MySQL Database Server* melalui LAN.

3.2.10 Disain *Input / Output*

Disain *input/output* digunakan untuk menampilkan rancangan dari sebuah tampilan yang disesuaikan dengan kebutuhan pengguna. Fungsi dari *form-form* yang telah disesuaikan dengan kebutuhan akan dijelaskan disini. Tujuan adanya disain *input/output* adalah agar pengembang mengerti tampilan yang akan dibangun.

1. Disain *Input / Output* Halaman Kompetensi Inti (MGMP)

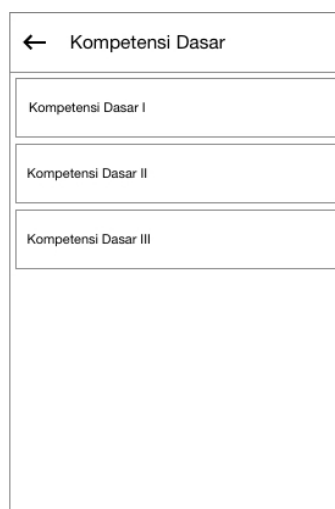
Halaman kompetensi dasar memiliki tampilan sebagai MGMP yang isinya adalah kompetensi berdasarkan kelas X, kelas XI, dan kelas XII. Kompetensi Inti akan ditampilkan ke bawah. Halaman ini dibuat dengan tujuan agar MGMP dapat melihat kompetensi dasar yang digunakan untuk membuat materi, tugas, dan soal ujian.



Gambar 3.91 Halaman Kompetensi Inti

2. Disain *Input / Output* Halaman Kompetensi Dasar (MGMP)

Halaman kompetensi dasar merupakan halaman detail dari kompetensi inti.

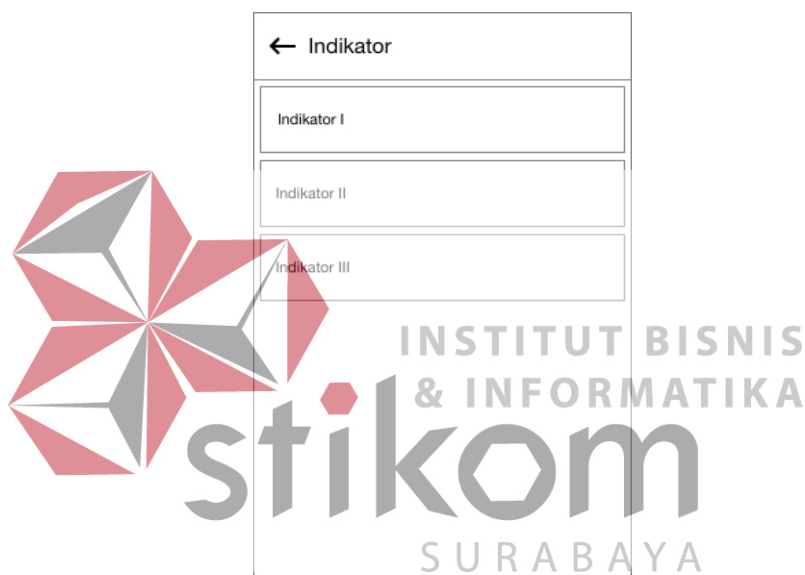


Gambar 3.92 Halaman Kompetensi Dasar

Tampilan kompetensi dasar untuk MGMP yang bertujuan untuk melihat kesesuaian kompetensi inti. Bentuk tampilannya dalam bentuk daftar ke bawah yang berisikan teks macam-macam kompetensi dasar.

3. Disain *Input / Output* Halaman Indikator (MGMP)

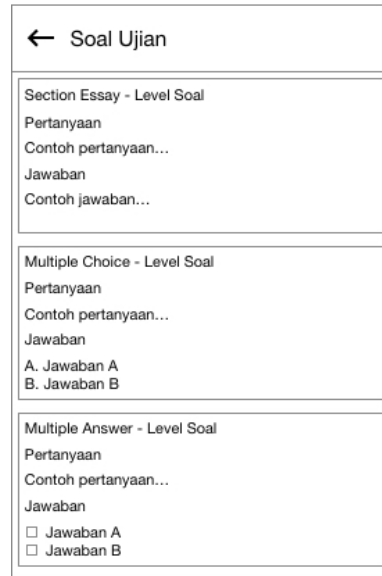
Halaman indikator adalah halaman yang berisikan detil dari kompetensi inti. Halaman indikator ini sama dengan kompetensi inti dalam bentuk daftar yang berisikan teks indikator.



Gambar 3.93 Halaman Indikator

4. Disain *Input / Output* Halaman Soal Berdasarkan Kompetensi (MGMP)

Halaman soal berdasarkan kompetensi ini adalah halaman yang dapat diakses oleh MGMP agar MGMP dalam melihat soal dan jawaban yang berdasarkan kompetensi. Soal dibagi menjadi tiga yaitu soal essay, soal *multiple choice*, soal *multiple answer* yang disetiap soal ditampilkan pertanyaan dan jawabannya.



← Soal Ujian

Section Essay - Level Soal
 Pertanyaan
 Contoh pertanyaan...
 Jawaban
 Contoh jawaban...

Multiple Choice - Level Soal
 Pertanyaan
 Contoh pertanyaan...
 Jawaban
 A. Jawaban A
 B. Jawaban B

Multiple Answer - Level Soal
 Pertanyaan
 Contoh pertanyaan...
 Jawaban
 Jawaban A
 Jawaban B

Gambar 3.94 Halaman Soal Ujian

5. Disain *Input / Output* Halaman Materi MGMP (MGMP)

Halaman materi MGMP ini adalah halaman untuk menampilkan seluruh materi dari MGMP. Setiap materi akan menampilkan judul materi, nama admin MGMP, tanggal unggah, dan jumlah klik oleh pengguna lainnya.



← Materi MGMP

X XI XII

Judul Materi MGMP
 Nama Admin MGMP
 Tanggal Upload XX Kali

Judul Materi MGMP
 Nama Admin MGMP
 Tanggal Upload XX Kali

Judul Materi MGMP
 Nama Admin MGMP
 Tanggal Upload XX Kali

Gambar 3.95 Halaman Materi MGMP

6. Disain *Input / Output* Halaman Materi Guru (MGMP)

Halaman materi guru digunakan untuk menampilkan guru yang dilihat oleh MGMP. Setiap materi diberikan informasi judul materi, nama guru, tanggal unggah, jumlah klik dari pengguna yang lain.

The screenshot shows a mobile application interface titled "Materi Guru". At the top, there is a back arrow and the title. Below the title are three tabs labeled "X", "XI", and "XII". The main content area displays a list of three material entries. Each entry consists of a square icon on the left, followed by a text box containing "Judul Materi Guru" and "Nama Guru". Below this, another text box shows "Tanggal Upload" followed by "XX Kali" and a toggle switch labeled "on off".

Gambar 3.96 Halaman Materi Guru

7. Disain *Input / Output* Halaman Forum Diskusi Guru dan MGMP (MGMP)

Halaman forum diskusi guru dan MGMP digunakan untuk saling berdiskusi seperti layaknya *chatting*

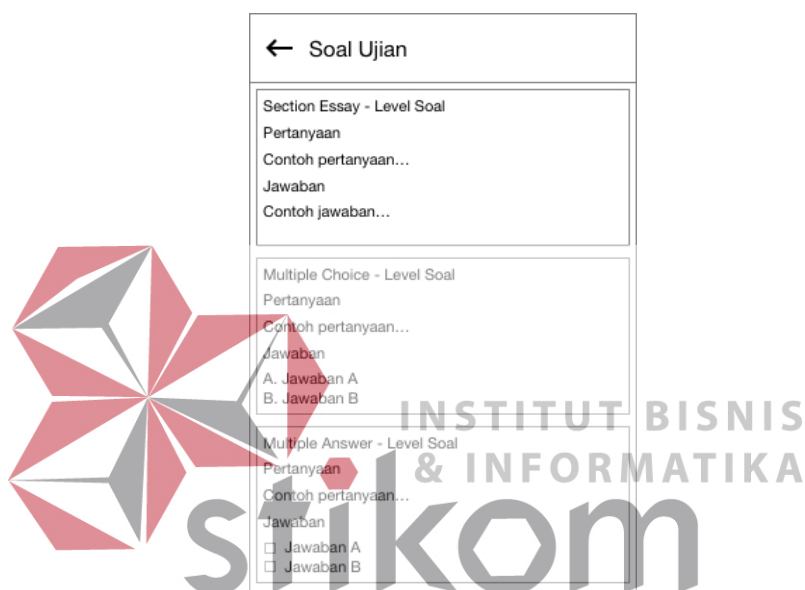
The screenshot shows a mobile application interface titled "Forum Diskusi". At the top, there is a back arrow and the title. The main content area is a chat interface. It features two message bubbles, each containing the text "Nama Pengguna" and "Isi pesan dan lainnya tanggal". At the bottom of the screen, there is a text input field labeled "Kirim Pesan...".

Gambar 3.97 Halaman Forum Diskusi Guru dan MGMP

Pesan yang ditampilkan pada halaman ini adalah pesan dari pengguna dan pesan dari pengguna lain yang tergabung dalam satu diskusi.

8. Disain *Input / Output* Halaman Soal Ujian (MGMP)

Halaman soal ujian digunakan oleh guru untuk melihat soal-soal dalam satu ujian. Soal-soal dibagi menjadi tiga jenis (Gambar 3.93) disertai pertanyaan, pilihan jawaban, dan jawaban yang benar.



Gambar 3.98 Halaman Soal Ujian

9. Disain *Input / Output* Halaman Kompetensi Inti (GURU)

Halaman kompetensi inti memiliki tampilan sebagai guru yang isinya adalah kompetensi berdasarkan kelas X, kelas XI, dan kelas XII. Kompetensi inti akan ditampilkan ke bawah. Halaman ini dibuat dengan tujuan agar guru dapat melihat kompetensi inti yang digunakan untuk membuat materi, tugas, dan soal ujian.

← Kompetensi Inti		
X	XI	XII
Kompetensi Inti I		
Kompetensi Inti II		
Kompetensi Inti III		

Gambar 3.99 Halaman Kompetensi Inti

10. Disain *Input / Output* Halaman Kompetensi Dasar (Guru)

Halaman kompetensi dasar merupakan halaman detail dari kompetensi inti. Tampilan kompetensi dasar untuk guru yang bertujuan untuk melihat kesesuaian kompetensi inti. Bentuk tampilannya dalam bentuk daftar ke bawah yang berisikan teks macam-macam kompetensi inti.

← Kompetensi Dasar	
Kompetensi Dasar I	
Kompetensi Dasar II	
Kompetensi Dasar III	

Gambar 3.100 Halaman Kompetensi Dasar

11. Disain *Input / Output* Halaman Indikator (Guru)

Halaman indikator adalah halaman yang berisikan detail dari kompetensi inti. Halaman indikator ini sama dengan kompetensi inti dalam bentuk daftar yang berisikan teks indikator.



← Indikator

Indikator I

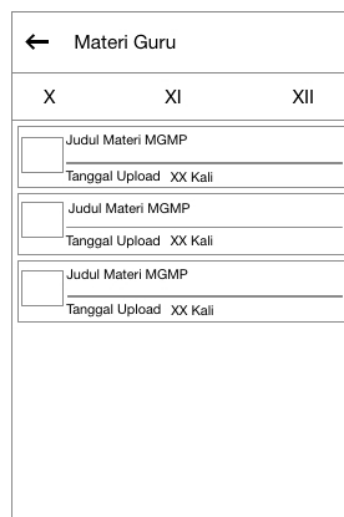
Indikator II

Indikator III

Gambar 3.101 Halaman Indikator

12. Disain *Input / Output* Halaman Materi MGMP (Guru)

Halaman materi MGMP ini adalah halaman untuk menampilkan seluruh materi dari MGMP. Setiap materi akan menampilkan judul materi, nama admin MGMP, tanggal unggah, dan jumlah klik oleh pengguna lainnya.



← Materi Guru

X XI XII

Judul Materi MGMP
Tanggal Upload XX Kali

Judul Materi MGMP
Tanggal Upload XX Kali

Judul Materi MGMP
Tanggal Upload XX Kali

Gambar 3.102 Halaman Materi MGMP

13. Disain *Input / Output* Halaman Materi Guru (Guru)

Halaman materi guru digunakan untuk menampilkan guru yang dilihat oleh guru. Setiap materi diberikan informasi judul materi, nama guru, tanggal unggah, jumlah klik dari pengguna yang lain.

← Materi Guru

X XI XII

Judul Materi Guru
Nama Guru
Tanggal Upload XX Kali on off

Judul Materi Guru
Nama Guru
Tanggal Upload XX Kali on off

Judul Materi Guru
Nama Guru
Tanggal Upload XX Kali on off

Gambar 3.103 Halaman Materi Guru

14. Disain *Input / Output* Halaman Forum Diskusi Guru dan MGMP (Guru)

Halaman forum diskusi guru dan MGMP digunakan untuk saling berdiskusi seperti layaknya *chatting*.

← Forum Diskusi

Nama Pengguna
Isi pesan dan lainnya
tanggal

Nama Pengguna
Isi pesan dan lainnya
tanggal

Kirim Pesan...

Gambar 3.104 Halaman Forum Diskusi Guru dan MGMP

Pesan yang ditampilkan pada halaman ini adalah pesan dari pengguna dan pesan dari pengguna lain yang tergabung dalam satu diskusi.

15. Disain *Input / Output* Halaman Forum Diskusi Kelas (Guru)

Halaman forum diskusi kelas digunakan untuk saling berdiskusi bersama siswa dalam satu kelas seperti layaknya *chatting*. Pesan yang ditampilkan pada halaman ini adalah pesan dari pengguna dan pesan dari pengguna lain yang tergabung dalam satu diskusi.



Gambar 3.105 Halaman Forum Diskusi Kelas

16. Disain *Input / Output* Halaman Tugas (Guru)

Halaman tugas digunakan untuk menampilkan tugas yang telah diberikan oleh guru dalam satu kelas.

← Tugas	
Tugas I	
Kelompok	Tgl Deadline
Tugas II	
Kelompok	Tgl Deadline
Tugas III	
Kelompok	Tgl Deadline

Gambar 3.106 Halaman Tugas

17. Disain *Input / Output* Halaman Menilai Tugas (Guru)

Halaman menilai tugas digunakan oleh guru untuk menilai setiap siswa yang telah mengumpulkan tugas. Halaman ini berisikan NIS / NISN, nama siswa, unduh *file* tugas, isi nilai, dan tombol simpan.

← Nilai Tugas
NIS / NISN
NAMA SISWA
Unduh File Tugas
Nilai
SIMPAN

Gambar 3.107 Halaman Menilai Tugas

18. Disain *Input / Output* Halaman Menilai Soal Ujian (Guru)

Halaman menilai soal ujian untuk guru menampilkan soal-soal dalam sebuah ujian beserta isi dari nilai untuk esai, Soal jenis *multipl choice* dan *multiple answer* telah diisi oleh sistem.

← Nilai Soal Ujian
Section Essay - Level Soal Pertanyaan Contoh pertanyaan... Jawaban Contoh jawaban... Nilai...
Multiple Choice - Level Soal Pertanyaan Contoh pertanyaan... Jawaban A. Jawaban A B. Jawaban B Nilai : 10
Multiple Answer - Level Soal Pertanyaan Contoh pertanyaan... Jawaban <input type="checkbox"/> Jawaban A <input type="checkbox"/> Jawaban B Nilai : 10

Gambar 3.108 Halaman Menilai Soal Ujian

19. Disain *Input / Output* Halaman Materi MGMP (Siswa)

Halaman materi MGMP ini adalah halaman untuk menampilkan seluruh materi dari MGMP. Setiap materi akan menampilkan judul materi, nama admin MGMP, tanggal unggah, dan jumlah klik oleh pengguna lainnya.

Gambar 3.109 Halaman Materi MGMP

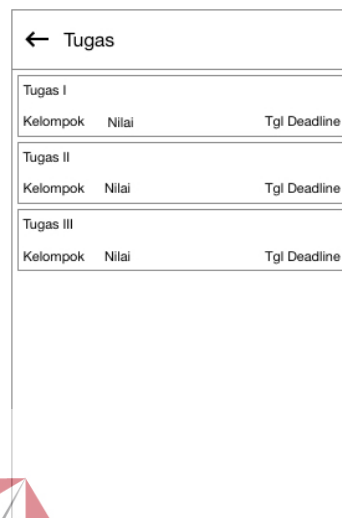
20. Disain *Input / Output* Halaman Materi Guru (Siswa)

Halaman materi guru digunakan untuk menampilkan guru yang dilihat oleh siswa. Setiap materi diberikan informasi judul materi, nama guru, tanggal unggah, jumlah klik dari pengguna yang lain.

Gambar 3.110 Halaman Materi Guru

21. Disain *Input / Output* Halaman Tugas (Siswa)

Halaman tugas digunakan untuk menampilkan tugas yang telah diberikan oleh guru dalam satu kelas. Tugas ditampilkan dalam bentuk list yang berisi nama tugas, kelompok, tanggal *deadline*.

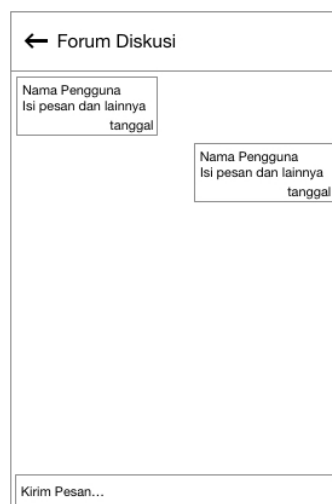


← Tugas		
Tugas I		
Kelompok	Nilai	Tgl Deadline
Tugas II		
Kelompok	Nilai	Tgl Deadline
Tugas III		
Kelompok	Nilai	Tgl Deadline

Gambar 3.111 Halaman Tugas

22. Disain *Input / Output* Halaman Forum Diskusi Kelas (Siswa)

Halaman forum diskusi kelas digunakan untuk saling berdiskusi bersama siswa dalam satu kelas seperti layaknya *chatting*. Pesan yang ditampilkan pada halaman ini adalah pesan dari pengguna dan pesan dari pengguna lain yang tergabung dalam satu diskusi.



← Forum Diskusi

Nama Pengguna
Isi pesan dan lainnya
tanggal

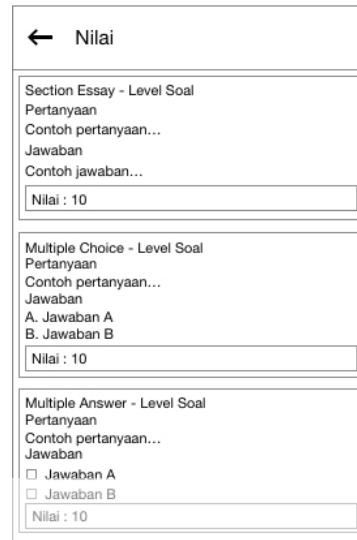
Nama Pengguna
Isi pesan dan lainnya
tanggal

Kirim Pesan...

Gambar 3.112 Halaman Forum Diskusi

23. Disain *Input / Output* Halaman Nilai dan Soal Ujian (Siswa)

Halaman nilai soal dan ujian ditampilkan pada siswa untuk melihat soal dan nilai yang telah diberikan oleh guru pengajar.



← Nilai

Section Essay - Level Soal
Pertanyaan
Contoh pertanyaan...
Jawaban
Contoh jawaban...
Nilai : 10

Multiple Choice - Level Soal
Pertanyaan
Contoh pertanyaan...
Jawaban
A. Jawaban A
B. Jawaban B
Nilai : 10

Multiple Answer - Level Soal
Pertanyaan
Contoh pertanyaan...
Jawaban
 Jawaban A
 Jawaban B
Nilai : 10

Gambar 3.113 Halaman Nilai dan Soal Ujian



BAB IV

IMPLEMENTASI DAN EVALUASI

4.1 Implementasi *Scrum*

Pada tahap ini akan diimplementasikan penggunaan kerangka *scrum* untuk pengembangan aplikasi *MoLearn* berbasis Android ini. Sesuai yang dijelaskan pada tahap ketiga dokumentasi *scrum* yang terdiri dari *product backlog*, *Sprint planning*, *Sprint*, dan *Sprint Backlog*.

4.1.1 *Product Backlog*

Product backlog yang telah direncanakan akan dikerjakan *sprint planning* selama tiga minggu sebanyak empat kali *sprint*. Sehingga total *sprint* sebanyak dua belas minggu. Isi dari *product backlog* pertama kali adalah fitur-fitur yang akan dimasukkan ke dalam *MoLearn*.

Fitur-fitur yang dimasukkan ke dalam *MoLearn* ada pada Gambar 3.1 Blok Diagram Aplikasi *MoLearn*. Fitur-fitur tersebut akan dikerjakan selama empat kali *sprint*. *Sprint* pertama akan fokus pada pembuatan fitur untuk pengguna sebagai MGMP. *Sprint* kedua akan fokus pada pembuatan fitur untuk pengguna guru. *Sprint* ketiga akan fokus pada pembuatan fitur untuk pengguna sebagai siswa. Kemudian *sprint* keempat akan dilaksanakan *User Acceptance Testing*.

Untuk lebih jelasnya *product backlog* akan ditampilkan dalam bentuk tabel yang fitur dari aplikasi yang akan dibangun.

Tabel 4.1 *Product Backlog*

No.	Fitur
1	Menampilkan kompetensi dasar

No.	Fitur
2	Menampilkan kompetensi inti
3	Menampilkan indikator
4	Menampilkan soal berdasarkan kompetensi
5	Menampilkan materi MGMP
6	Menampilkan materi guru.
7	Forum diskusi (guru dan MGMP).
8	Menampilkan kompetensi dasar
9	Menampilkan kompetensi inti
10	Menampilkan indikator
11	Menampilkan soal ujian
12	Menampilkan materi MGMP
13	Menampilkan materi guru.
14	Forum diskusi (MGMP, guru, dan kelas).
15	Forum diskusi (Guru dan Siswa)
16	Menampilkan tugas (guru)
17	Menilai tugas
18	Menilai soal ujian
19	Menampilkan materi MGMP
20	Menampilkan materi guru.
21	Menampilkan tugas (siswa)
22	Forum diskusi (kelas)
23	Menampilkan nilai dan soal ujian

4.1.2 *Sprint Ke-1*

A. *Sprint Planning*

Sprint planning ke-1 difokuskan pada pembuatan fitur-fitur yang tercatat pada *product backlog* yang akan digunakan oleh pengguna sebagai MGMP. Ada beberapa fitur yang akan digunakan sebagai pengguna MGMP yang akan dikembangkan pada *sprint planning* ke-1 ini.

Tabel 4.2 *Sprint Backlog* Menampilkan Kompetensi Dasar

Prioritas	Mendesak
Sprint ke	1 (Satu)
Tipe Pengguna	MGMP
Fitur	Menampilkan Kompetensi Dasar
Detil Pengerjaan	

Prioritas	Mendesak
Sprint ke	1 (Satu)
Tipe Pengguna	MGMP
Fitur	Menampilkan Kompetensi Dasar
<ul style="list-style-type: none"> • Membuat halaman untuk kompetensi dasar • Membuat tab untuk pembagian kelas. • Membuat <i>query</i> untuk menampilkan kompetensi dasar berdasarkan kelas • Membuat <i>end-point</i> untuk ditampilkan pada Android. • Membuat <i>model</i> untuk kompetensi dasar. • Membuat <i>controller</i> untuk kompetensi dasar. 	

Tabel 4.3 *Sprint Backlog* Menampilkan Kompetensi Inti

Prioritas	Mendesak
Sprint ke	1 (Satu)
Tipe Pengguna	MGMP
Fitur	Menampilkan Kompetensi Inti
Detil Pengerjaan <ul style="list-style-type: none"> • Membuat halaman untuk kompetensi inti. • Membuat <i>query</i> untuk menampilkan kompetensi inti berdasarkan kelas. • Membuat <i>end-point</i> untuk ditampilkan pada Android. • Membuat <i>model</i> untuk kompetensi inti. • Membuat <i>controller</i> untuk kompetensi inti. 	

Tabel 4.4 *Sprint Backlog* menampilkan indikator

Prioritas	Mendesak
Sprint ke	1 (Satu)
Tipe Pengguna	MGMP
Fitur	Menampilkan Indikator
Detil Pengerjaan <ul style="list-style-type: none"> • Membuat halaman untuk indikator. • Membuat <i>query</i> untuk menampilkan indikator berdasarkan kelas. • Membuat <i>end-point</i> untuk ditampilkan pada Android. • Membuat <i>model</i> untuk indikator. • Membuat <i>controller</i> untuk indikator. 	

Tabel 4.5 *Sprint Backlog* menampilkan soal berdasarkan kompetensi

Prioritas	Mendesak
Sprint ke	1 (Satu)
Tipe Pengguna	MGMP
Fitur	Menampilkan soal berdasarkan kompetensi
Detil Pengerjaan <ul style="list-style-type: none"> • Membuat halaman untuk soal. Soal dibagi menjadi tiga tipe yaitu soal esai, 	

Prioritas	Mendesak
Sprint ke	1 (Satu)
Tipe Pengguna	MGMP
Fitur	Menampilkan soal berdasarkan kompetensi
soal pilihan ganda, dan soal <i>multiple choice</i> . <ul style="list-style-type: none"> • Membuat <i>query</i> untuk menampilkan soal. • Membuat <i>model</i> untuk soal. • Membuat <i>controller</i> untuk soal. 	

Tabel 4.6 *Sprint Backlog* menampilkan materi MGMP

Prioritas	Mendesak
Sprint ke	1 (Satu)
Tipe Pengguna	MGMP
Fitur	Menampilkan materi MGMP
Detil Pengerjaan <ul style="list-style-type: none"> • Membuat halaman untuk menampilkan materi MGMP. • Membuat <i>query</i> untuk menampilkan materi MGMP. • Membuat <i>model</i> untuk materi MGMP. • Membuat <i>controller</i> untuk materi MGMP. 	

Tabel 4.7 *Sprint Backlog* menampilkan materi guru

Prioritas	Mendesak
Sprint ke	1 (Satu)
Tipe Pengguna	MGMP
Fitur	Menampilkan materi guru
Detil Pengerjaan <ul style="list-style-type: none"> • Membuat halaman untuk menampilkan materi guru. • Membuat <i>query</i> untuk menampilkan materi guru. • Membuat <i>model</i> untuk materi guru. • Membuat <i>controller</i> untuk materi guru. 	

Tabel 4.8 *Sprint Backlog* forum diskusi (Guru dan MGMP)

Prioritas	Mendesak
Sprint ke	1 (Satu)
Tipe Pengguna	MGMP
Fitur	Forum diskusi (Guru dan MGMP)
Detil Pengerjaan <ul style="list-style-type: none"> • Membuat halaman untuk fasilitas forum diskusi untuk MGMP. • Membuat <i>query</i> untuk menampilkan forum diskusi. • Membuat <i>end-point</i> POST untuk mengirim pesan. • Membuat <i>end-point</i> untuk menampilkan isi forum. 	

- Membuat notifikasi untuk ditampilkan pada Android.

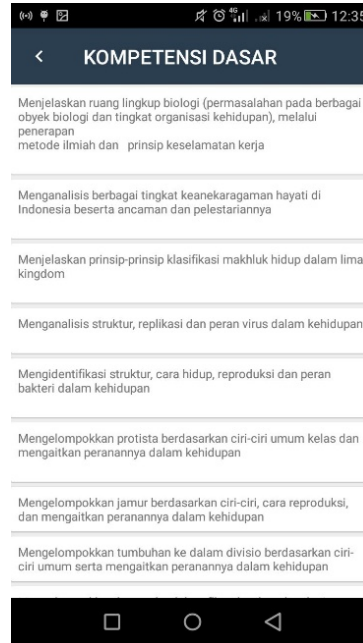
Setelah *Sprint Planning* ke-1 telah dilaksanakan maka langkah selanjutnya adalah masuk ke dalam *daily sprint*.

B. Sprint Review

Disain analisa telah dilakukan dengan mengacu pada perancangan sistem pada Bab III. Pada Bab III terdapat perancangan sistem berorientasi objek dimulai dari *use case diagram*, *flow of event*, dan *sequence diagram*. Setelah sinkronisasi maka akan digunakan untuk pembuatan aplikasi.

1. Halaman Kompetensi Dasar (MGMP)

Halaman kompetensi dasar memiliki *flow of event* pada Tabel 3.2 *Flow of Event* Menampilkan Kompetensi Dasar (MGMP), *sequence diagram* Gambar 3.3 *Sequence Diagram* Menampilkan Kompetensi Dasar (MGMP), dan disain aplikasi pada Gambar 3.91. Halaman kompetensi dasar ini memiliki isi kompetensi dasar berdasarkan mata pelajaran. Halaman ini dibangun agar pengguna MGMP dapat melihat kompetensi dasar.



Gambar 4.1 Halaman Kompetensi Dasar

Sesuai dengan prinsip *sprint*, maka ada evaluasi terhadap halaman kompetensi dasar yang mencakup Tabel 4.9:

Tabel 4.9 *Testing* halaman kompetensi dasar

Test Case	Tujuan	Input	Harapan Output	Status
1	Pengujian terhadap tampilan kompetensi dasar dalam bentuk list	Tidak Ada	Menampilkan <i>ListView</i> .	Sukses
2	Pengujian terhadap isi dari kompetensi dasar berdasarkan kelas.	Kelas 10, Kelas 11, Kelas 12	Menampilkan <i>ListView</i> berdasarkan kelas	Sukses
3	Pengujian terhadap isi dari kompetensi dasar berdasarkan mata pelajaran.	Tidak Ada	Menampilkan <i>ListView</i> berdasarkan kelas	Sukses

Sprint review untuk kompetensi dasar ini adalah:

Tabel 4.10 *Sprint review* kompetensi dasar

No.	Detil Pengerjaan	Status
1	Membuat halaman untuk kompetensi dasar	Selesai
2	Membuat tab untuk pembagian kelas.	Selesai
3	Membuat <i>query</i> untuk menampilkan kompetensi dasar berdasarkan kelas	Selesai

4	Membuat <i>end-point</i> untuk ditampilkan pada Android.	Selesai
5	Membuat <i>model</i> untuk kompetensi dasar.	Selesai
6	Membuat <i>controller</i> untuk kompetensi dasar.	Selesai

2. Halaman Kompetensi Inti (MGMP)

Halaman kompetensi inti memiliki *flow of event* pada Tabel 3.3, *sequence diagram* Gambar 3.4, dan disain aplikasi pada Gambar 3.92. Halaman kompetensi inti ini memiliki isi kompetensi inti berdasarkan kompetensi dasar yang dipilih sebelumnya. Halaman ini dibangun agar pengguna MGMP dapat melihat kompetensi inti.



Gambar 4.2 Halaman Kompetensi Inti

Sesuai dengan prinsip *sprint*, maka ada evaluasi terhadap halaman kompetensi dasar yang mencakup Tabel 4.11:

Tabel 4.11 *Testing* halaman kompetensi inti

Test Case	Tujuan	Input	Harapan Output	Status
1	Pengujian terhadap tampilan kompetensi dasar dalam bentuk list	Tidak Ada	Menampilkan ListView.	Sukses

Test Case	Tujuan	Input	Harapan Output	Status
2	Pengujian terhadap isi dari kompetensi dasar berdasarkan kelas.	Kelas 10, Kelas 11, Kelas 12	Menampilkan ListView berdasarkan kelas	Sukses
3	Pengujian terhadap isi dari kompetensi dasar berdasarkan mata pelajaran.	Tidak Ada	Menampilkan ListView berdasarkan kelas	Sukses

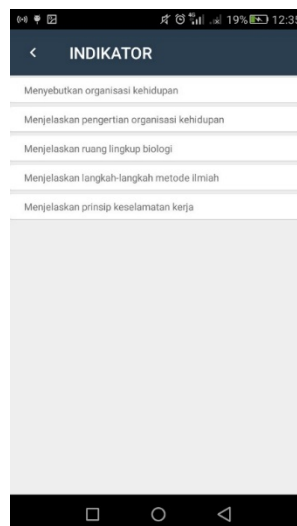
Sprint review untuk kompetensi inti adalah:

Tabel 4.12 *Sprint review* kompetensi inti

No.	Detil Pengerjaan	Status
1	Membuat halaman untuk kompetensi inti.	Selesai
2	Membuat <i>query</i> untuk menampilkan kompetensi inti berdasarkan kelas.	Selesai
3	Membuat <i>end-point</i> untuk ditampilkan pada Android.	Selesai
4	Membuat <i>model</i> untuk kompetensi inti.	Selesai
5	Membuat <i>controller</i> untuk kompetensi inti.	Selesai

3. Halaman Indikator (MGMP)

Halaman indikator memiliki *flow of event* pada Tabel 3.4, *sequence diagram* Gambar 3.5, dan disain aplikasi pada Gambar 3.93.



Gambar 4.3 Halaman Indikator

Halaman indikator ini memiliki isi indikator berdasarkan kompetensi inti yang dipilih sebelumnya. Halaman ini dibangun agar pengguna MGMP dapat melihat indikator. Sesuai dengan prinsip *sprint*, maka ada evaluasi terhadap halaman indikator yang mencakup Tabel 4.13:

Tabel 4.13 *Testing* halaman indikator

Test Case	Tujuan	Input	Harapan Output	Status
1	Pengujian terhadap tampilan indikator dalam bentuk list	Tidak Ada	Menampilkan ListView.	Sukses
2	Pengujian terhadap isi dari indikator berdasarkan kompetensi inti yang dipilih	ID Kompetensi Inti	Menampilkan ListView berdasarkan kompetensi dasar	Sukses

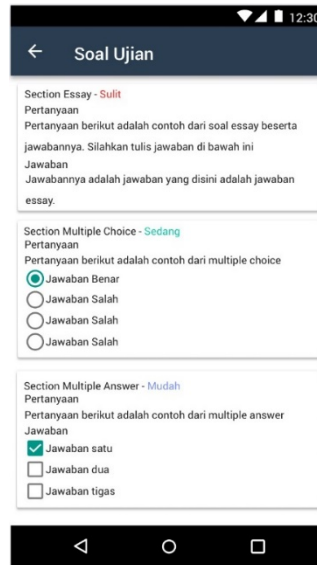
Sprint review untuk indikator adalah:

Tabel 4.14 *Sprint review* indikator

No.	Detil Pengerjaan	Status
1	Membuat halaman untuk indikator.	Selesai
2	Membuat <i>query</i> untuk menampilkan indikator berdasarkan kelas.	Selesai
3	Membuat <i>end-point</i> untuk ditampilkan pada Android.	Selesai
4	Membuat <i>model</i> untuk indikator.	Selesai
5	Membuat <i>controller</i> untuk indikator.	Selesai

4. Halaman Soal Berdasarkan Kompetensi (MGMP)

Halaman soal berdasarkan kompetensi memiliki *flow of event* pada Tabel 3.5, *sequence diagram* Gambar 3.6, dan disain aplikasi pada Gambar 3.94. Halaman indikator ini memiliki isi seluruh soal-soal berdasarkan kompetensi. Halaman ini dibuat agar pengguna MGMP agar pengguna dapat melihat soal dan jawabannya.



Gambar 4.4 Halaman Soal Berdasarkan Kompetensi

Sesuai dengan prinsip *sprint*, maka ada evaluasi terhadap halaman soal berdasarkan kompetensi yang mencakup Tabel 4.15:



Tabel 4.15 *Testing* halaman soal berdasarkan kompetensi

Test Case	Tujuan	Input	Harapan Output	Status
1	Pengujian terhadap tampilan soal dalam bentuk list	Tidak Ada	Menampilkan ListView soal.	Sukses

Sprint review untuk halaman soal berdasarkan kompetensi adalah:

Tabel 4.16 *Sprint review* halaman soal berdasarkan kompetensi

No.	Detil Pengerjaan	Status
1	Membuat halaman untuk soal. Soal dibagi menjadi tiga tipe yaitu soal esai, soal pilihan ganda, dan soal <i>multiple choice</i> .	Selesai
2	Membuat <i>query</i> untuk menampilkan soal.	Selesai
3	Membuat <i>model</i> untuk soal.	Selesai
4	Membuat <i>controller</i> untuk soal.	Selesai

5. Halaman Materi MGMP (MGMP)

Halaman materi MGMP memiliki *flow of event* pada Tabel 3.6 *Flow of Event* Menampilkan Materi MGMP (MGMP), *sequence diagram* Gambar 3.7, dan disain aplikasi pada Gambar 3.102.



Gambar 4.5 Halaman Materi MGMP

Halaman materi MGMP ini memiliki isi seluruh materi dari pengguna MGMP. Halaman ini dibangun agar pengguna MGMP dapat menampilkan seluruh materi MGMP.

Sesuai dengan prinsip *sprint*, maka ada evaluasi terhadap halaman materi MGMP yang mencakup Tabel 4.17:

Tabel 4.17 *Testing* halaman materi MGMP

Test Case	Tujuan	Input	Harapan Output	Status
1	Pengujian terhadap tampilan materi dalam bentuk list.	ID Mata Pelajaran	Menampilkan <i>ListView</i> .	Sukses
2	Pengujian terhadap isi dari materi yang terdiri dari tiga jenis.	ID Materi MGMP	Menampilkan materi sesuai dengan jenis materi.	Sukses

Sprint review untuk materi MGMP adalah:

Tabel 4.18 *Sprint review* materi MGMP

No.	Detil Pengerjaan	Status
1	Membuat halaman untuk menampilkan materi MGMP.	Selesai
2	Membuat <i>query</i> untuk menampilkan materi MGMP.	Selesai
3	Membuat <i>model</i> untuk materi MGMP.	Selesai
4	Membuat <i>controller</i> untuk materi MGMP.	Selesai

6. Halaman Materi Guru (MGMP)

Halaman materi guru memiliki *flow of event* pada Tabel 3.7 *Flow of Event* Menampilkan Materi Guru (MGMP), *sequence diagram* Gambar 3.8, dan disain aplikasi pada Gambar 3.96. Halaman materi guru ini memiliki isi seluruh materi yang berasal dari guru itu sendiri.



Gambar 4.6 Halaman Materi Guru

Sesuai dengan prinsip *sprint*, maka ada evaluasi terhadap halaman materi guru yang mencakup tabel Tabel 4.19:

Tabel 4.19 *Testing* halaman materi guru

Test Case	Tujuan	Input	Harapan Output	Status
1	Pengujian terhadap tampilan materi dalam bentuk list.	ID Mata Pelajaran	Menampilkan ListView.	Sukses
2	Pengujian terhadap isi dari materi yang terdiri dari tiga jenis.	ID Materi guru	Menampilkan materi sesuai dengan jenis materi.	Sukses
3	Pengujian terhadap materi guru yang tampil hanya materi yang telah diunggah	Tidak ada	Menampilkan materi dalam bentuk listView hanya materi yang telah diunggah oleh guru itu sendiri	Sukses

Sprint review materi guru adalah:

Tabel 4.20 *Sprint review* materi guru

No.	Detil Pengerjaan	Status
1	Membuat halaman untuk menampilkan materi guru.	Selesai

No.	Detil Pengerjaan	Status
2	Membuat <i>query</i> untuk menampilkan materi guru.	Selesai
3	Membuat <i>model</i> untuk materi guru.	Selesai
4	Membuat <i>controller</i> untuk materi guru.	Selesai

7. Halaman Forum Diskusi Guru dan MGMP (MGMP)

Halaman forum diskusi guru dan MGMP memiliki *flow of event* pada Tabel 3.8 *Flow of Event* Forum Diskusi Guru dan MGMP (MGMP), *sequence diagram* Gambar 3.9, dan disain aplikasi pada Gambar 3.97. Halaman ini digunakan untuk saling berinteraksi melalui pesan teks yang akan terhubung dengan seluruh MGMP dan guru yang terhubung dalam satu aplikasi. Pengguna mengirimkan pesan melalui sebuah teks yang telah disediakan kemudian tekan kirim maka pesan akan langsung tersebar pada smartphone masing-masing.



Gambar 4.7 Halaman Forum Diskusi Guru dan MGMP

Sesuai dengan prinsip *sprint*, maka ada evaluasi terhadap halaman forum diskusi guru dan MGMP yang mencakup Tabel 4.21:

Tabel 4.21 *Testing* halaman forum diskusi guru dan MGMP

Test Case	Tujuan	Input	Harapan Output	Status
1	Pengujian terhadap tampilan pesan teks	ID Forum diskusi, Mata Pelajaran	Menampilkan seluruh pesan teks dari berbagai pengguna guru dan MGMP	Sukses
2	Pengujian terhadap pengiriman teks ke forum.	Pesan teks	Menampilkan pesan teks terakhir.	Sukses

Sprint review forum diskusi guru dan MGMP adalah:

Tabel 4.22 *Sprint review* forum diskusi guru dan MGMP

No.	Detil Pengerjaan	Status
1	Membuat halaman untuk fasilitas forum diskusi untuk MGMP.	Selesai
2	Membuat <i>query</i> unuk menampilkan forum diskusi.	Selesai
3	Membuat <i>end-point</i> POST untuk mengirim pesan.	Selesai
4	Membuat <i>end-point</i> untuk menampilkan isi forum.	Selesai
5	Membuat notifikasi untuk ditampilkan pada Android.	Selesai

C. *Sprint Retrospective*

Sprint Retrospective untuk *sprint* pertama memiliki beberapa kendala karena baru pertama kali mengerjakan walaupun secara fungsional telah selesai semua. Beberapa kendala adalah:

1. Kendala pada emulator yang tidak lancar ketika *compile*. Kendala ini diselesaikan dengan cara menggunakan *device* asli ketika *compile*.
2. Kendala pada pembuatan notifikasi secara *realtime* untuk membuat sebuah forum diskusi. Kendala ini diselesaikan dengan cara mempelajari secara terus menerus cara menggunakan *firebase* untuk notifikasi.

3. Penulisan *syntax* yang tidak rapi. Penulisan *syntax* yang tidak rapi diselesaikan dengan cara mengubah pola pemrograman yang ditulis pada Android Studio.

4.1.3 *Sprint* Ke-2

A. *Sprint Planning*

Sprint planning ke-2 difokuskan pada pembuatan fitur-fitur yang dicatat pada *product backlog* yang akan digunakan oleh pengguna sebagai guru. Ada beberapa fitur yang akan digunakan sebagai pengguna guru yang akan dikembangkan pada *sprint planning* ke-2 ini.

Tabel 4.23 *Sprint Backlog* menampilkan kompetensi dasar

Prioritas	Mendesak
Sprint ke	2 (Dua)
Tipe Pengguna	Guru
Fitur	Menampilkan kompetensi dasar
Detil Pengerjaan	<ul style="list-style-type: none"> • Halaman kompetensi dasar telah dibuat sehingga dapat ditampilkan pada guru. • Membuat pengecekan pengguna yang <i>login</i> ke aplikasi dengan tujuan untuk mengetahui hak akses mana yang boleh mengakses kompetensi dasar.

Tabel 4.24 *Sprint Backlog* menampilkan kompetensi inti

Prioritas	Mendesak
Sprint ke	2 (Dua)
Tipe Pengguna	Guru
Fitur	Menampilkan kompetensi inti
Detil Pengerjaan	<ul style="list-style-type: none"> • Halaman kompetensi inti telah dibuat sehingga dapat ditampilkan pada guru. • Membuat pengecekan pengguna yang <i>login</i> ke aplikasi dengan tujuan untuk mengetahui hak akses mana yang dapat akses kompetensi inti.

Tabel 4.25 *Sprint Backlog* menampilkan indikator

Prioritas	Mendesak
Sprint ke	2 (Dua)
Tipe Pengguna	Guru
Fitur	Menampilkan indikator

Prioritas	Mendesak
Sprint ke	2 (Dua)
Tipe Pengguna	Guru
Fitur	Menampilkan indikator
Detil Pengerjaan	
<ul style="list-style-type: none"> • Halaman indikator telah dibuat sehingga dapat ditampilkan pada guru. • Membuat pengecekan pengguna yang <i>login</i> ke aplikasi dengan tujuan agar aplikasi dapat cek pengguna yang dapat akses indikator. 	

Tabel 4.26 *Sprint Backlog* menampilkan soal ujian

Prioritas	Mendesak
Sprint ke	2 (Dua)
Tipe Pengguna	Guru
Fitur	Menampilkan soal ujian
Detil Pengerjaan	
<ul style="list-style-type: none"> • Membuat halaman pemilihan kompetensi. • Membuat halaman menampilkan soal ujian. • Membuat <i>query</i> untuk menampilkan soal ujian. • Membuat <i>end-point</i> untuk menampilkan soal ujian. • Membuat <i>model</i> untuk soal ujian. • Membuat <i>controller</i> untuk soal ujian. 	

Tabel 4.27 *Sprint Backlog* menampilkan materi MGMP

Prioritas	Mendesak
Sprint ke	2 (Dua)
Tipe Pengguna	Guru
Fitur	Menampilkan materi MGMP
Detil Pengerjaan	
<ul style="list-style-type: none"> • Halaman materi MGMP telah dibuat sehingga dapat ditampilkan pada guru. • Membuat pengecekan pengguna yang <i>login</i> ke aplikasi dengan tujuan agar aplikasi dapat cek pengguna yang dapat akses materi MGMP. 	

Tabel 4.28 *Sprint Backlog* menampilkan materi guru

Prioritas	Mendesak
Sprint ke	2 (Dua)
Tipe Pengguna	Guru
Fitur	Menampilkan materi guru
Detil Pengerjaan	
<ul style="list-style-type: none"> • Halaman menampilkan materi guru telah dibuat sehingga dapat ditampilkan pada guru. • Membuat pengecekan pengguna yang <i>login</i> ke aplikasi dengan tujuan agar 	

aplikasi dapat cek pengguna yang dapat akses materi guru.

Tabel 4.29 *Sprint Backlog* forum diskusi (MGMP dan Guru)

Prioritas	Mendesak
Sprint ke	2 (Dua)
Tipe Pengguna	Guru
Fitur	Forum diskusi (MGMP dan Guru)
Detil Pengerjaan	
<ul style="list-style-type: none"> • Halaman forum diskusi MGMP telah dibuat sehingga dapat ditampilkan pada guru. • Membuat pengecekan pengguna yang <i>login</i> ke aplikasi dengan tujuan agar aplikasi dapat cek pengguna yang dapat forum diskusi ini. 	

Tabel 4.30 *Sprint Backlog* forum diskusi (Guru dan Siswa)

Prioritas	Mendesak
Sprint ke	2 (Dua)
Tipe Pengguna	Guru
Fitur	Forum diskusi (Guru dan Siswa)
Detil Pengerjaan	
<ul style="list-style-type: none"> • Halaman forum diskusi guru dan siswa telah dibuat sehingga dapat ditampilkan pada guru. • Membuat pengecekan pengguna yang <i>login</i> ke aplikasi dengan tujuan agar aplikasi dapat cek pengguna yang dapat akses forum diskusi ini. 	

Tabel 4.31 *Sprint Backlog* menampilkan tugas

Prioritas	Mendesak
Sprint ke	2 (Dua)
Tipe Pengguna	Guru
Fitur	Menampilkan tugas
Detil Pengerjaan	
<ul style="list-style-type: none"> • Membuat halaman menampilkan tugas. • Membuat <i>query</i> untuk menampilkan tugas. • Membuat <i>end-point</i> untuk menampilkan tugas. • Membuat <i>model</i> untuk menampilkan tugas. • Membuat <i>controller</i> untuk menampilkan tugas. 	

Tabel 4.32 *Sprint Backlog* menilai tugas

Prioritas	Mendesak
Sprint ke	2 (Dua)
Tipe Pengguna	Guru
Fitur	Menilai Tugas

Prioritas	Mendesak
Sprint ke	2 (Dua)
Tipe Pengguna	Guru
Fitur	Menilai Tugas
Detil Pengerjaan	
<ul style="list-style-type: none"> • Membuat halaman menilai tugas. • Membuat <i>query</i> untuk menampilkan nama siswa dan <i>file</i> untuk <i>download</i>. • Membuat <i>end-point</i> untuk menampilkan list siswa dalam sebuah tugas. • Membuat <i>model</i> untuk menilai tugas. • Membuat <i>controller</i> untuk menilai tugas. 	

Tabel 4.33 *Sprint Backlog* menilai soal ujian

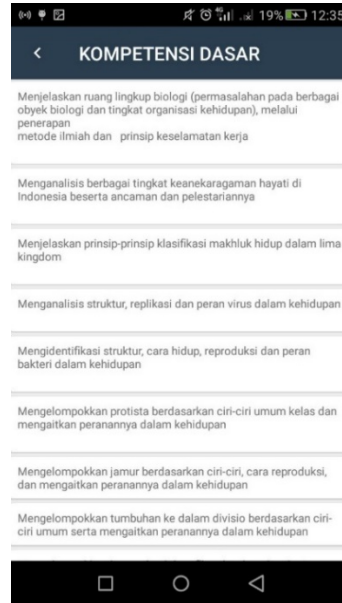
Prioritas	Mendesak
Sprint ke	2 (Dua)
Tipe Pengguna	Guru
Fitur	Menilai soal ujian
Detil Pengerjaan	
<ul style="list-style-type: none"> • Membuat halaman menilai soal ujian. • Membuat halaman untuk siswa yang telah melaksanakan ujian. • Membuat <i>query</i> untuk menampilkan nama siswa yang telah ujian. • Membuat <i>query</i> untuk soal yang digunakan untuk ujian. • Membuat <i>end-point</i> untuk menampilkan list siswa yang telah ujian. • Membuat <i>end-point</i> untuk menampilkan soal untuk ujian. • Membuat <i>model</i> untuk soal ujian. • Membuat <i>controller</i> untuk soal ujian. 	

Setelah *Sprint Planning* ke-2 telah dilaksanakan maka langkah selanjutnya adalah masuk ke dalam *daily sprint*.

B. *Sprint Review*

1. Halaman Kompetensi Dasar (GURU)

Halaman kompetensi dasar memiliki *flow of event* pada Tabel 3.10, *sequence diagram* Gambar 3.11, dan disain aplikasi pada Gambar 3.99.



Gambar 4.8 Gambar Kompetensi Dasar

Halaman kompetensi dasar ini memiliki isi kompetensi dasar berdasarkan mata pelajaran. Halaman ini dibangun agar pengguna guru dapat melihat standar kompetensi dasar.

Sesuai dengan prinsip *sprint*, maka ada evaluasi terhadap halaman kompetensi dasar yang mencakup Tabel 4.34:

Tabel 4.34 *Testing* halaman kompetensi dasar

Test Case	Tujuan	Input	Harapan Output	Status
1	Pengujian terhadap tampilan kompetensi dasar dalam bentuk list	Tidak Ada	Menampilkan ListView.	Sukses
2	Pengujian terhadap isi dari kompetensi dasar berdasarkan kelas.	Kelas 10, Kelas 11, Kelas 12	Menampilkan <i>ListView</i> berdasarkan kelas	Sukses
3	Pengujian terhadap isi dari kompetensi dasar berdasarkan mata pelajaran.	Tidak Ada	Menampilkan <i>ListView</i> berdasarkan kelas	Sukses

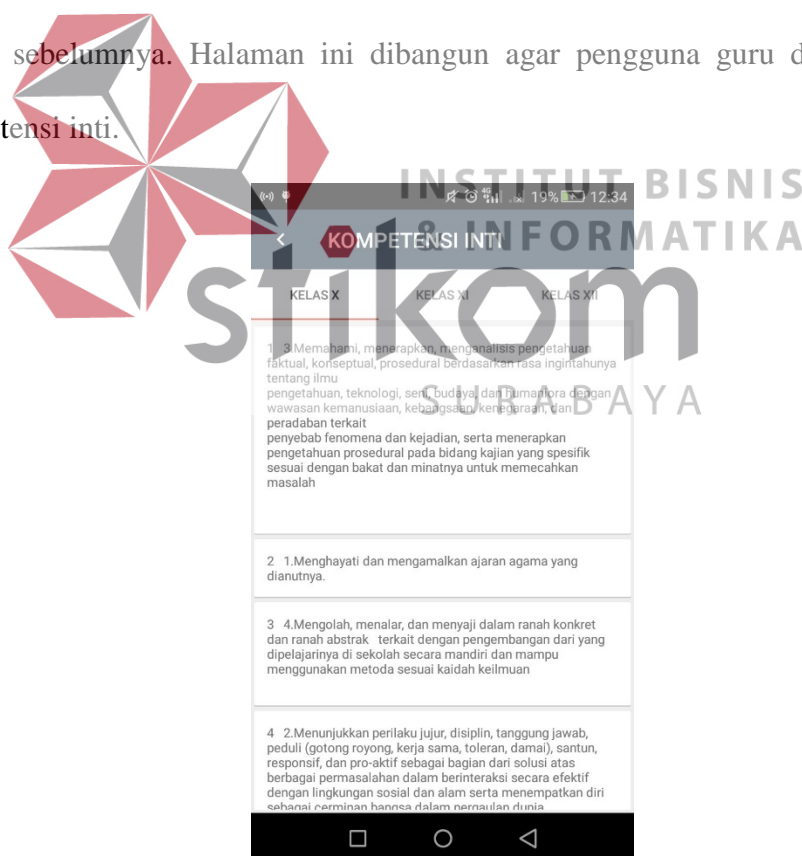
Sprint review untuk kompetensi dasar adalah:

Tabel 4.35 *Sprint review* kompetensi dasar untuk guru

No.	Detil Pengerjaan	Status
1	Halaman kompetensi dasar telah dibuat sehingga dapat ditampilkan pada guru.	Selesai
2	Membuat pengecekan pengguna yang <i>login</i> ke aplikasi dengan tujuan untuk mengetahui hak akses mana yang boleh mengakses kompetensi dasar.	Selesai

2. Halaman Kompetensi Inti (GURU)

Halaman kompetensi inti memiliki *flow of event* pada Tabel 3.11, *sequence diagram* Gambar 3.12, dan disain aplikasi pada Gambar 3.100. Halaman kompetensi inti ini memiliki isi kompetensi berdasarkan kompetensi dasar yang dipilih sebelumnya. Halaman ini dibangun agar pengguna guru dapat melihat kompetensi inti.



Gambar 4.9 Gambar Kompetensi Inti

Sesuai dengan prinsip *sprint*, maka ada evaluasi terhadap halaman kompetensi inti yang mencakup Tabel 4.36:

Tabel 4.36 *Testing* halaman kompetensi inti

Test Case	Tujuan	Input	Harapan Output	Status
1	Pengujian terhadap tampilan kompetensi inti dalam bentuk list	Tidak Ada	Menampilkan ListView.	Sukses
2	Pengujian terhadap isi dari kompetensi inti berdasarkan kompetensi dasar yang dipilih	ID Kompetensi dasar	Menampilkan ListView berdasarkan kompetensi dasar	Sukses

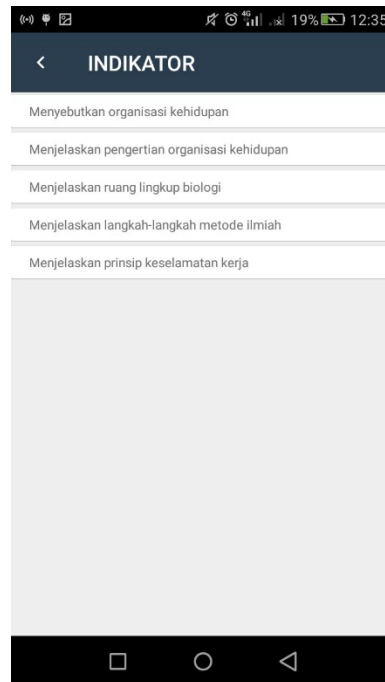
Sprint review kompetensi inti untuk guru adalah:

Tabel 4.37 *Sprint review* kompetensi inti untuk guru

No.	Detil Pengerjaan	Status
1	Halaman kompetensi inti telah dibuat sehingga dapat ditampilkan pada guru.	Selesai
2	Membuat pengecekan pengguna yang <i>login</i> ke aplikasi dengan tujuan untuk mengetahui hak akses mana yang dapat akses kompetensi inti.	Selesai

3. Halaman Indikator (GURU)

Halaman indikator memiliki *flow of event* pada Tabel 3.12, *sequence diagram* Gambar 3.13, dan disain aplikasi pada Gambar 3.101. Halaman indikator ini memiliki isi indikator berdasarkan kompetensi inti yang dipilih sebelumnya. Halaman ini dibangun agar pengguna MGMP dapat melihat indikator.



Gambar 4.10 Gambar Indikator

Sesuai dengan prinsip *sprint*, maka ada evaluasi terhadap halaman indikator yang mencakup Tabel 4.38:

Tabel 4.38 *Testing* halaman indikator

Test Case	Tujuan	Input	Harapan Output	Status
1	Pengujian terhadap tampilan indikator dalam bentuk list	Tidak Ada	Menampilkan ListView.	Sukses
2	Pengujian terhadap isi dari indikator berdasarkan kompetensi inti yang dipilih	ID Kompetensi Inti	Menampilkan ListView berdasarkan kompetensi dasar	Sukses

Sprint review menampilkan indikator untuk guru adalah:

Tabel 4.39 *Sprint review* menampilkan indikator untuk guru

No.	Detil Pengerjaan	Status
1	Halaman indikator telah dibuat sehingga dapat ditampilkan pada guru.	Selesai
2	Membuat pengecekan pengguna yang <i>login</i> ke aplikasi dengan tujuan agar aplikasi dapat cek pengguna yang	Selesai

No.	Detil Pengerjaan	Status
	dapat akses indikator.	

4. Halaman Materi MGMP (GURU)

Halaman materi MGMP memiliki *flow of event* pada Tabel 3.13, *sequence diagram* Gambar 3.14, dan disain aplikasi pada Gambar 3.109. Halaman materi MGMP ini memiliki isi seluruh materi dari pengguna guru. Halaman ini dibangun agar pengguna guru dapat menampilkan seluruh materi MGMP.



Gambar 4.11 Halaman Materi MGMP

Sesuai dengan prinsip *sprint*, maka ada evaluasi terhadap halaman materi MGMP yang mencakup Tabel 4.40:

Tabel 4.40 *Testing* halaman materi MGMP

Test Case	Tujuan	Input	Harapan Output	Status
1	Pengujian terhadap tampilan materi dalam bentuk list.	ID Mata Pelajaran	Menampilkan ListView.	Sukses
2	Pengujian terhadap isi dari materi yang terdiri	ID Materi MGMP	Menampilkan materi sesuai dengan jenis	Sukses

Test Case	Tujuan	Input	Harapan Output	Status
	dari tiga jenis.		materi.	

Sprint review materi MGMP untuk guru adalah:

Tabel 4.41 *Sprint review* materi MGMP untuk guru

No.	Detil Pengerjaan	Status
1	Halaman materi MGMP telah dibuat sehingga dapat ditampilkan pada guru.	Selesai
2	Membuat pengecekan pengguna yang <i>login</i> ke aplikasi dengan tujuan agar aplikasi dapat cek pengguna yang dapat akses materi MGMP.	Selesai

5. Halaman Materi Guru (GURU)

Halaman materi guru memiliki *flow of event* pada Tabel 3.14, *sequence diagram* Gambar 3.15, dan disain aplikasi pada Gambar 3.110.



Gambar 4.12 Halaman Materi Guru

Halaman materi guru ini memiliki isi seluruh materi yang berasal dari guru itu sendiri. Sesuai dengan prinsip *sprint*, maka ada evaluasi terhadap halaman materi guru yang mencakup Tabel 4.42:

Tabel 4.42 *Testing* halaman materi guru

Test Case	Tujuan	Input	Harapan Output	Status
1	Pengujian terhadap tampilan materi dalam bentuk list.	ID Mata Pelajaran	Menampilkan ListView.	Sukses
2	Pengujian terhadap isi dari materi yang terdiri dari tiga jenis.	ID Materi guru	Menampilkan materi sesuai dengan jenis materi.	Sukses
3	Pengujian terhadap materi guru yang tampil hanya materi yang telah diunggah	Tidak ada	Menampilkan materi dalam bentuk listView hanya materi yang telah diunggah oleh guru itu sendiri	Sukses

Sprint review halaman materi guru adalah:

Tabel 4.43 *Sprint review* materi guru

No.	Detil Pengerjaan	Status
1	Halaman menampilkan materi guru telah dibuat sehingga dapat ditampilkan pada guru.	Selesai
2	Membuat pengecekan pengguna yang <i>login</i> ke aplikasi dengan tujuan agar aplikasi dapat cek pengguna yang dapat akses materi guru.	Selesai

6. Halaman Forum Diskusi Guru dan MGMP (GURU)

Halaman forum diskusi guru dan MGMP memiliki *flow of event* pada Tabel 3.15, *sequence diagram* Gambar 3.16, dan disain aplikasi pada Gambar 3.104. Halaman ini digunakan untuk saling berinteraksi melalui pesan teks yang akan terhubung dengan seluruh MGMP dan guru yang terhubung dalam satu aplikasi. Pengguna mengirimkan pesan melalui sebuah teks yang telah disediakan kemudian tekan kirim maka pesan akan langsung tersebar pada smartphone masing-masing.



Gambar 4.13 Forum Diskusi Guru dan MGMP

Sesuai dengan prinsip *sprint*, maka ada evaluasi terhadap halaman forum diskusi guru dan MGMP yang mencakup Tabel 4.44:

Tabel 4.44 *Testing* halaman forum diskusi guru dan MGMP

Test Case	Tujuan	Input	Harapan Output	Status
1	Pengujian terhadap tampilan pesan teks	ID Forum diskusi, Mata Pelajaran	Menampilkan seluruh pesan teks dari berbagai pengguna guru dan MGMP	Sukses
2	Pengujian terhadap pengiriman teks ke forum.	Pesan teks	Menampilkan pesan teks terakhir.	Sukses

Sprint review forum diskusi guru dan MGMP adalah:

Tabel 4.45 *Sprint review* forum diskusi guru dan MGMP

No.	Detil Pengerjaan	Status
1	Halaman forum diskusi MGMP telah dibuat sehingga dapat ditampilkan pada guru.	Selesai
2	Membuat pengecekan pengguna yang <i>login</i> ke aplikasi	Selesai

No.	Detil Pengerjaan	Status
	dengan tujuan agar aplikasi dapat cek pengguna yang dapat forum diskusi ini.	

7. Halaman Forum Diskusi Kelas (GURU)

Halaman forum diskusi kelas memiliki *flow of event* pada Tabel 3.16, *sequence diagram* Gambar 3.17, dan disain aplikasi pada Gambar 3.105. Halaman ini digunakan untuk saling berinteraksi melalui pesan teks yang akan terhubung dengan seluruh siswa dalam satu kelas yang terhubung dalam satu aplikasi. Pengguna mengirimkan pesan melalui sebuah teks yang telah disediakan kemudian tekan kirim maka pesan akan langsung tersebar pada smartphone masing-masing.



Gambar 4.14 Diskusi Kelas

Sesuai dengan prinsip *sprint*, maka ada evaluasi terhadap halaman forum diskusi kelas yang mencakup Tabel 4.46:

Tabel 4.46 *Testing* halaman forum diskusi kelas

Test Case	Tujuan	Input	Harapan Output	Status
1	Pengujian terhadap tampilan pesan teks	ID kelas diskusi, Mata Pelajaran	Menampilkan seluruh pesan teks dari berbagai pengguna guru atau siswa dalam satu kelas pembelajaran	Sukses
2	Pengujian terhadap pengiriman teks ke forum.	Pesan teks	Menampilkan pesan teks terakhir.	Sukses

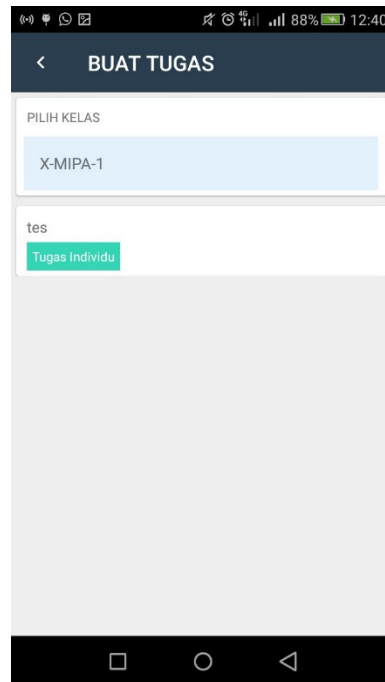
Sprint review forum diskusi kelas adalah:

Tabel 4.47 *Sprint review* forum diskusi kelas.

No.	Detil Pengerjaan	Status
1	Halaman forum diskusi guru dan siswa telah dibuat sehingga dapat ditampilkan pada guru.	Selesai
2	Membuat pengecekan pengguna yang <i>login</i> ke aplikasi dengan tujuan agar aplikasi dapat cek pengguna yang dapat akses forum diskusi ini.	Selesai

8. Halaman Tugas (GURU)

Halaman tugas memiliki *flow of event* pada Tabel 3.17, *sequence diagram* Gambar 3.18, dan disain aplikasi pada Gambar 3.106. Halaman tugas ini memiliki isi *list* tugas yang telah dibuat sebelumnya untuk kelas ajar. Halaman ini dibangun untuk menampilkan tugas-tugas dalam kelas ajar guru.



Gambar 4.15 Halaman Daftar Tugas

Sesuai dengan prinsip *sprint*, maka ada evaluasi terhadap halaman tugas yang mencakup Tabel 4.48:

Tabel 4.48 *Testing* halaman tugas

Test Case	Tujuan	Input	Harapan Output	Status
1	Pengujian terhadap tampilan tugas dalam bentuk list	Kelas pembelajaran, Mata Pelajaran	Menampilkan tugas dalam bentuk listView	Sukses
2	Pengujian terhadap Action Click pada salah satu listView	ID Tugas	Menampilkan detail tugas.	Sukses

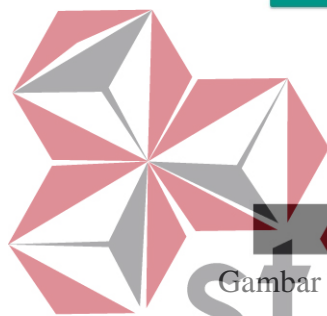
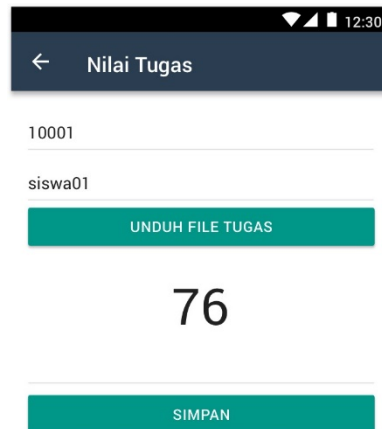
Sprint review halaman tugas adalah:

Tabel 4.49 *Sprint review* halaman tugas

No.	Detil Pengerjaan	Status
1	Membuat halaman menampilkan tugas.	Selesai
2	Membuat <i>query</i> untuk menampilkan tugas.	Selesai
3	Membuat <i>end-point</i> untuk menampilkan tugas.	Selesai
4	Membuat <i>model</i> untuk tugas.	Selesai
5	Membuat <i>controller</i> untuk tugas.	Selesai

9. Halaman Menilai Tugas (GURU)

Halaman menilai tugas memiliki *flow of event* pada Tabel 3.18, *sequence diagram* Gambar 3.19, dan disain aplikasi pada Gambar 3.107.



INSTITUT BISNIS
& INFORMATIKA

Gambar 4.16 Halaman Nilai Tugas

Halaman menilai tugas ini berbentuk *form* yang dapat diisi nilai mahasiswa. Halaman ini dibangun agar pengguna guru dapat menilai tugas siswa dalam satu kelas ajar.

Sesuai dengan prinsip *sprint*, maka ada evaluasi terhadap halaman menilai tugas yang mencakup Tabel 4.50:

Tabel 4.50 *Testing* halaman menilai tugas

Test Case	Tujuan	Input	Harapan Output	Status
1	Pengujian terhadap tampilan tugas dalam bentuk list	Kelas pembelajaran, Mata Pelajaran	Menampilkan tugas dalam bentuk listView	Sukses
2	Pengujian terhadap Action Click pada	ID Tugas	Menampilkan detail tugas.	Sukses

Test Case	Tujuan	Input	Harapan Output	Status
	salah satu listView			

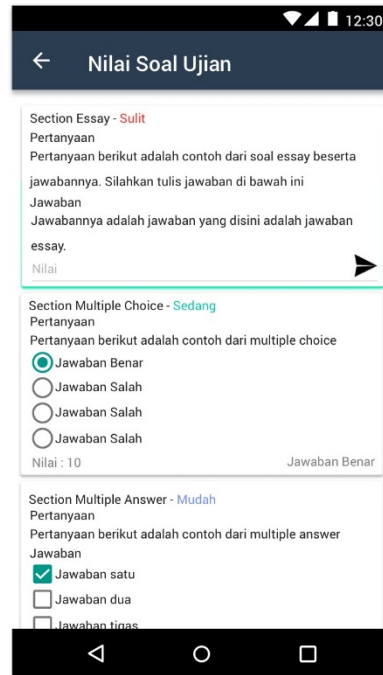
Sprint review menilai tugas adalah:

Tabel 4.51 *Sprint review* menilai tugas

No.	Detil Pengerjaan	Status
1	Membuat halaman menilai tugas.	Selesai
2	Membuat <i>query</i> untuk menampilkan nama siswa dan <i>file</i> untuk <i>download</i> .	Selesai
3	Membuat <i>end-point</i> untuk menampilkan list siswa dalam sebuah tugas.	Selesai
4	Membuat <i>model</i> untuk menilai tugas.	Selesai
5	Membuat <i>controller</i> untuk menilai tugas.	Selesai

10. Halaman Menilai Soal Ujian (GURU)

Halaman menilai soal ujian memiliki *flow of event* pada Tabel 3.19, *sequence diagram* Gambar 3.20, dan disain aplikasi pada Gambar 3.108. Halaman menilai soal ujian ini memiliki isi soal dan *form* untuk menilai khusus untuk esai. Halaman ini membantu pengguna guru untuk menilai soal ujian siswa yang telah dikerjakan sebelumnya.



Gambar 4.17 Halaman Nilai Soal Ujian

Sesuai dengan prinsip *sprint*, maka ada evaluasi terhadap halaman menilai soal ujian yang mencakup Tabel 4.52:

Tabel 4.52 *Testing* halaman menilai soal ujian

Test Case	Tujuan	Input	Harapan Output	Status
1	Pengujian terhadap tampilan soal dalam bentuk list	ID Ujian	Menampilkan soal dalam bentuk listView	Sukses
2	Pengujian terhadap <i>entry</i> nilai pada salah satu soal (esai)	Nilai soal	Nilai terkirim dan menampilkan hasil akhir.	Sukses

Sprint review menilai soal ujian adalah:

Tabel 4.53 *Sprint review* menilai soal ujian

No.	Detil Pengerjaan	Status
1	Membuat halaman menilai soal ujian.	Selesai
2	Membuat halaman untuk siswa yang telah melaksanakan ujian.	Selesai
3	Membuat <i>query</i> untuk menampilkan nama siswa yang telah ujian.	Selesai
4	Membuat <i>query</i> untuk soal yang digunakan untuk ujian.	Selesai

No.	Detil Pengerjaan	Status
5	Membuat <i>end-point</i> untuk menampilkan list siswa yang telah ujian	Selesai
6	Membuat <i>end-point</i> untuk menampilkan soal untuk ujian.	Selesai
7	Membuat <i>model</i> untuk soal ujian.	Selesai
8	Membuat <i>controller</i> untuk soal ujian.	Selesai

11. Halaman Menampilkan Soal Ujian (Guru)

Halaman menampilkan soal ujian memiliki *flow of event* pada Tabel 3.9, *sequence diagram* Gambar 3.10, dan disain aplikasi pada Gambar 3.94. Halaman ini digunakan untuk menampilkan soal ketika pengguna MGMP membutuhkan soal berdasarkan kompetensi.



Gambar 4.18 Halaman Menampilkan Soal Ujian

Sesuai dengan prinsip *sprint*, maka ada evaluasi terhadap halaman menampilkan soal ujian yang mencakup Tabel 4.54:

Tabel 4.54 *Testing* halaman menampilkan soal ujian

Test Case	Tujuan	Input	Harapan Output	Status
1	Pengujian terhadap tampilan soal dalam bentuk list	ID Ujian	Menampilkan soal dalam bentuk listView	Sukses

Sprint review menampilkan soal ujian adalah:

Tabel 4.55 *Sprint review* menampilkan soal ujian

No.	Detil Pengerjaan	Status
1	Membuat halaman pemilihan kompetensi.	Selesai
2	Membuat halaman menampilkan soal ujian.	Selesai
3	Membuat <i>query</i> untuk menampilkan soal ujian.	Selesai
4	Membuat <i>end-point</i> untuk menampilkan soal ujian.	Selesai
5	Membuat <i>model</i> untuk soal ujian.	Selesai
6	Membuat <i>controller</i> untuk soal ujian.	Selesai

C. *Sprint Retrospective*

Sprint Retrospective untuk sprint ke-2 ini mulai lancar. Catatan pada sprint ke-3 ini sebagai berikut:

1. Penulisan *syntax* yang sudah mulai rapi. Penulisan telah diubah menjadi pola pemrograman *model-view-controller*.
2. Sudah mulai terbiasa pembuatan notifikasi secara *realtime* di Android.

4.1.4 *Sprint Ke-3*

A. *Sprint Planning*

Sprint planning ke-3 difokuskan pada pembuatan fitur-fitur yang dicatat pada *product backlog* yang akan digunakan oleh pengguna sebagai siswa. Ada beberapa fitur yang akan digunakan sebagai pengguna siswa yang akan dikembangkan pada *sprint planning* ke-3 ini.

Tabel 4.56 *Sprint Backlog* menampilkan materi MGMP

Prioritas	Mendesak
Sprint ke	3 (Tiga)
Tipe Pengguna	Siswa
Fitur	Menampilkan materi MGMP
Detil Pengerjaan	
<ul style="list-style-type: none"> • Halaman materi MGMP telah dibuat sehingga dapat ditampilkan pada siswa. • Membuat pengecekan pengguna yang <i>login</i> ke aplikasi dengan tujuan agar aplikasi dapat cek pengguna yang dapat akses materi MGMP ini. 	

Tabel 4.57 *Sprint Backlog* menampilkan materi guru

Prioritas	Mendesak
Sprint ke	3 (Tiga)
Tipe Pengguna	Siswa
Fitur	Menampilkan materi guru
Detil Pengerjaan	
<ul style="list-style-type: none"> • Halaman materi guru telah dibuat sehingga dapat ditampilkan pada siswa. • Membuat pengecekan kelas yang ditempuh oleh siswa. • Membuat pengecekan pengguna yang <i>login</i> ke aplikasi dengan tujuan agar aplikasi dapat cek pengguna yang dapat akses materi guru ini. 	

Tabel 4.58 *Sprint Backlog* menampilkan tugas

Prioritas	Mendesak
Sprint ke	3 (Tiga)
Tipe Pengguna	Siswa
Fitur	Menampilkan tugas
Detil Pengerjaan	
<ul style="list-style-type: none"> • Membuat halaman untuk menampilkan tugas. • Membuat <i>query</i> untuk menampilkan tugas berdasarkan kelas dan guru yang mengajar. • Membuat <i>end-point</i> untuk menampilkan tugas. • Membuat <i>model</i> untuk tugas. • Membuat <i>controller</i> untuk tugas. 	

Tabel 4.59 *Sprint Backlog* forum diskusi

Prioritas	Mendesak
Sprint ke	3 (Tiga)
Tipe Pengguna	Siswa
Fitur	Forum diskusi siswa
Detil Pengerjaan	
<ul style="list-style-type: none"> • Halaman forum diskusi siswa telah dibuat sehingga dapat ditampilkan pada siswa. 	

Prioritas	Mendesak
Sprint ke	3 (Tiga)
Tipe Pengguna	Siswa
Fitur	Forum diskusi siswa
<ul style="list-style-type: none"> • Membuat pengecekan pengguna yang <i>login</i> ke aplikasi dengan tujuan agar aplikasi dapat cek pengguna yang dapat akses forum diskusi ini. 	

Tabel 4.60 *Sprint Backlog* menampilkan nilai dan soal ujian

Prioritas	Mendesak
Sprint ke	3 (Tiga)
Tipe Pengguna	Siswa
Fitur	Menampilkan nilai dan soal ujian
Detil Pengerjaan. <ul style="list-style-type: none"> • Membuat halaman untuk ujian yang telah dilaksanakan. • Membuat <i>query</i> untuk soal yang digunakan untuk ujian. • Membuat <i>end-point</i> untuk menampilkan soal untuk ujian. • Membuat <i>model</i> untuk soal ujian. • Membuat <i>controller</i> untuk soal ujian. 	

Setelah *Sprint Planning* ke-3 telah dilaksanakan maka langkah selanjutnya adalah masuk ke dalam *daily sprint*.

B. *Sprint Review*

1. Halaman Materi MGMP (SISWA)

Halaman materi MGMP memiliki *flow of event* pada Tabel 3.20, *sequence diagram* Gambar 3.21, dan disain aplikasi pada Gambar 3.109.



Gambar 4.19 Halaman Materi MGMP

Halaman materi MGMP ini memiliki isi seluruh materi dari pengguna siswa. Halaman ini dibangun agar pengguna guru dapat menampilkan seluruh materi MGMP sebagai bahan ajar selain dari materi guru. Sesuai dengan prinsip *sprint*, maka ada evaluasi terhadap halaman materi MGMP yang mencakup Tabel 4.61:

Tabel 4.61 *Testing* halaman materi MGMP

Test Case	Tujuan	Input	Harapan Output	Status
1	Pengujian terhadap tampilan materi dalam bentuk <i>list</i> .	ID Mata Pelajaran	Menampilkan <i>ListView</i> .	Sukses
2	Pengujian terhadap isi dari materi yang terdiri dari tiga jenis.	ID Materi MGMP	Menampilkan materi sesuai dengan jenis materi.	Sukses

Sprint review materi MGMP untuk siswa adalah:

Tabel 4.62 *Sprint review* materi MGMP untuk siswa

No.	Detil Pengerjaan	Status
1	Halaman materi MGMP telah dibuat sehingga dapat ditampilkan pada siswa.	Selesai
2	Membuat pengecekan pengguna yang <i>login</i> ke aplikasi dengan tujuan agar aplikasi dapat cek pengguna yang	Selesai

No.	Detil Pengerjaan	Status
	dapat akses materi MGMP ini.	

2. Halaman Materi Guru (SISWA)

Halaman materi guru memiliki *flow of event* pada Tabel 3.21, *sequence diagram* Gambar 3.22, dan disain aplikasi pada Gambar 3.110.



Gambar 4.20 Halaman Materi Guru

Halaman materi guru ini memiliki isi seluruh materi yang berasal dari guru ajar siswa dalam satu kelas.

Sesuai dengan prinsip *sprint*, maka ada evaluasi terhadap halaman materi guru yang mencakup Tabel 4.63:

Tabel 4.63 *Testing* halaman materi guru

Test Case	Tujuan	Input	Harapan Output	Status
1	Pengujian terhadap tampilan materi dalam bentuk list.	ID Mata Pelajaran	Menampilkan ListView.	Sukses
2	Pengujian terhadap isi dari materi yang terdiri dari tiga jenis.	ID Materi guru	Menampilkan materi sesuai dengan jenis materi.	Sukses
3	Pengujian terhadap	Tidak ada	Menampilkan materi	Sukses

Test Case	Tujuan	Input	Harapan Output	Status
	materi guru yang tampil hanya materi yang telah diunggah		dalam bentuk listView hanya materi yang telah diunggah oleh guru itu sendiri	

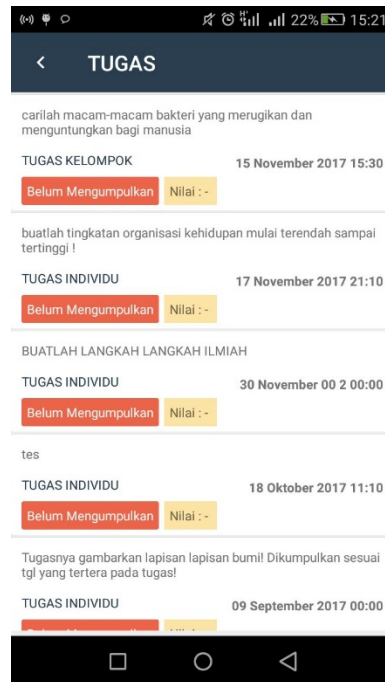
Sprint review materi guru untuk siswa adalah:

Tabel 4.64 *Sprint review* materi guru

No.	Detil Pengerjaan	Status
1	Halaman materi guru telah dibuat sehingga dapat ditampilkan pada siswa.	Selesai
2	Membuat pengecekan kelas yang ditempuh oleh siswa.	Selesai
3	Membuat pengecekan pengguna yang <i>login</i> ke aplikasi dengan tujuan agar aplikasi dapat cek pengguna yang dapat akses materi guru ini.	Selesai

3. Halaman Tugas (SISWA)

Halaman tugas siswa memiliki *flow of event* pada Tabel 3.22, *sequence diagram* Gambar 3.23, dan disain aplikasi pada Gambar 3.111. Halaman tugas siswa ini memiliki isi *list* tugas siswa yang diberikan oleh guru ajar siswa itu sendiri dalam sebuah kelas dengan satu mata pelajaran. Halaman tugas untuk siswa ini digunakan untuk siswa agar tetap mengingat tugas-tugas yang diberikan oleh guru.



Gambar 4.21 Halaman Tugas

Sesuai dengan prinsip *sprint*, maka ada evaluasi terhadap halaman tugas yang mencakup Tabel 4.65:

Tabel 4.65 *Testing* halaman tugas

Test Case	Tujuan	Input	Harapan Output	Status
1	Pengujian terhadap tampilan tugas dalam bentuk list	Kelas pembelajaran, Mata Pelajaran	Menampilkan tugas dalam bentuk listView	Sukses
2	Pengujian terhadap Action Click pada salah satu <i>listView</i>	ID Tugas	Menampilkan detail tugas.	Sukses

Sprint review halaman tugas untuk siswa adalah:

Tabel 4.66 *Sprint reiew* halaman tugas untuk siswa

No.	Detil Pengerjaan	Status
1	Membuat halaman untuk menampilkan tugas.	Selesai
2	Membuat <i>query</i> untuk menampilkan tugas berdasarkan kelas dan guru yang mengajar.	Selesai
3	Membuat <i>end-point</i> untuk menampilkan tugas.	Selesai
4	Membuat <i>model</i> untuk tugas.	Selesai

No.	Detil Pengerjaan	Status
5	Membuat <i>controller</i> untuk tugas.	Selesai

4. Halaman Forum Diskusi Kelas (Siswa)

Halaman forum diskusi kelas memiliki *flow of event* pada Tabel 3.23, *sequence diagram* Gambar 3.24, dan disain aplikasi pada Gambar 3.112. Halaman forum diskusi kelas ini digunakan untuk guru dan siswa untuk berinteraksi melalui pesan teks. Pengguna mengirim pesan dalam *form* yang telah disiapkan kemudian untuk mengirim pesan tersebut tekan tombol kirim maka pesan akan tersebar dalam satu kelas.



Gambar 4.22 Halaman Diskusi Kelas

Sesuai dengan prinsip *sprint*, maka ada evaluasi terhadap halaman forum diskusi kelas yang mencakup Tabel 4.67:

Tabel 4.67 *Testing* halaman forum diskusi kelas

Test Case	Tujuan	Input	Harapan Output	Status
1	Pengujian terhadap tampilan pesan teks	ID kelas diskusi, Mata	Menampilkan seluruh pesan teks dari berbagai	Sukses

Test Case	Tujuan	Input	Harapan Output	Status
		Pelajaran	pengguna guru atau siswa dalam satu kelas pembelajaran	
2	Pengujian terhadap pengiriman teks ke forum.	Pesan teks	Menampilkan pesan teks terakhir.	Sukses

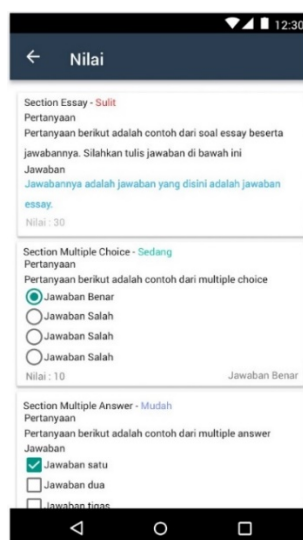
Sprint review forum diskusi siswa adalah:

Tabel 4.68 *Sprint review* forum diskusi siswa

No.	Detil Pengerjaan	Status
1	Halaman forum diskusi siswa telah dibuat sehingga dapat ditampilkan pada siswa.	Selesai
2	Membuat pengecekan pengguna yang <i>login</i> ke aplikasi dengan tujuan agar aplikasi dapat cek pengguna yang dapat akses forum diskusi ini.	Selesai

5. Halaman Nilai dan Soal Ujian (Siswa)

Halaman nilai dan soal ujian memiliki *flow of event* pada Tabel 3.24, *sequence diagram* Gambar 3.25, dan disain aplikasi pada Gambar 3.113. Halaman nilai dan soal ujian ini membantu siswa untuk melihat nilai dan jawaban benar atau salah dari soal yang telah dikerjakan.



Gambar 4.23 Halaman Nilai

Sesuai dengan prinsip *sprint*, maka ada evaluasi terhadap halaman nilai dan soal ujian yang mencakup

Tabel 4.69:

Tabel 4.69 Testing halaman nilai dan soal ujian

Test Case	Tujuan	Input	Harapan Output	Status
1	Pengujian terhadap tampilan soal dalam bentuk list	ID Ujian	Menampilkan soal dalam bentuk listView	Sukses
2	Pengujian terhadap teks nilai pada salah satu soal (esai)	Tidak ada	Menampilkan nilai setiap soal.	Sukses

Sprint review nilai dan soal ujian adalah:

Tabel 4.70 *Sprint review* nilai dan soal ujian

No.	Detil Pengerjaan	Status
1	Membuat halaman untuk ujian yang telah dilaksanakan.	Selesai
2	Membuat <i>query</i> untuk soal yang digunakan untuk ujian.	Selesai
3	Membuat <i>end-point</i> untuk menampilkan soal untuk ujian.	Selesai
4	Membuat <i>model</i> untuk soal ujian.	Selesai
5	Membuat <i>controller</i> untuk soal ujian.	Selesai

C. *Sprint Retrospective*

Sprint Retrospective pada *sprint* ke-3 sudah pada tahap penyelesaian aplikasi. Setelah ini akan dipresentasikan kepada tim MGMP. Pada *sprint* ini memiliki beberapa kendala ketika mengerjakan aplikasi yaitu:

1. Pembuatan *form* penilaian setiap soal yang cukup menyita waktu karena algoritmanya yang sulit. Setelah ditemukan proses *sprint* selanjutnya bisa dikerjakan dengan baik.
2. Beberapa fitur memiliki *bug* yang perlu diperbaiki karena adanya evaluasi dari tim MGMP. Melakukan perubahan code untuk menyelesaikan *bug*.

4.1.5 *Sprint Ke-4*

A. *Sprint Planning*

Sprint planning ke-4 akan fokus pada UAT dan *upload* aplikasi *MoLearn* pada *google play*. *Sprint planning* yang akan dilaksanakan adalah sebagai berikut:

Tabel 4.71 *Sprint Backlog* UAQ

Prioritas	Mendesak
Sprint ke	4 (Empat)
Tipe Pengguna	Admin
Fitur	UAQ
Detil Pengerjaan	
<ul style="list-style-type: none"> • Membuat quisioner. • Membagikan kepada para siswa. • Memproses hasil dari quisioner 	

Tabel 4. 72 *Sprint Backlog Upload Google Play*

Prioritas	Mendesak
Sprint ke	4 (Empat)
Tipe Pengguna	Admin
Fitur	<i>Upload Google Play</i>
Detil Pengerjaan	
<ul style="list-style-type: none"> • Mempersiapkan akun <i>Google</i>. • <i>Generate</i> file apk. • Upload ke <i>Google Play</i>. 	

B. *Sprint Review*

1. *User Acceptance Testing (UAT)*

Setelah *sprint* dalam analisa, disain UI/UX, pembuatan aplikasi *MoLearn* dan *testing*, maka diadakan UAT dalam bentuk kuisisioner dengan metode *User Experience Quesioner (UEQ)* untuk menilai penggunaan aplikasi.

Untuk melakukan evaluasi terhadap Aplikasi *MoLearn*, pengguna diminta untuk mengisi kuisisioner berikut ini. Kuisisioner terdiri dari pasangan atribut yang bertolak belakang yang dapat merepresentasikan Aplikasi *MoLearn*. Lingkaran-

lingkaran yang berada di tengah atribut merepresentasikan gradasi antar atribut yang ada dengan cara memilih lingkaran yang lebih dengan dengan impresi pengguna.

Tabel 4.73 Tabel Kuisisioner

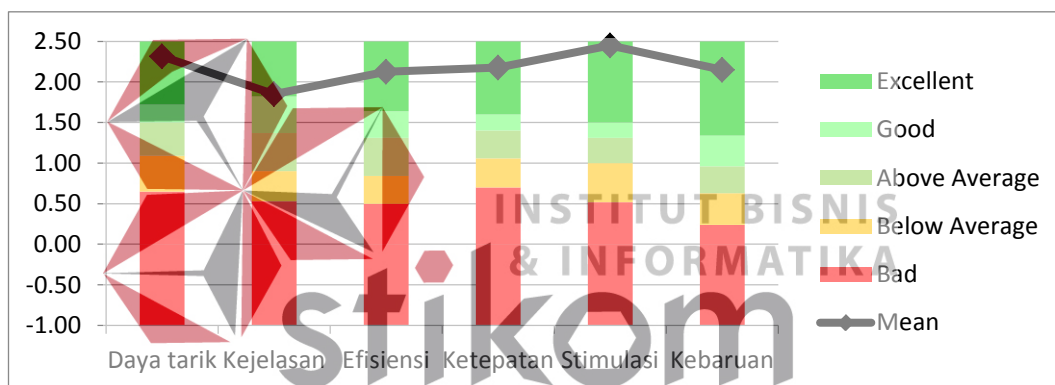
menyusahkan	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	menyenangkan
tak dapat dipahami	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	dapat dipahami
monoton	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	kreatif
sulit dipelajari	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	mudah dipelajari
kurang bermanfaat	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	bermanfaat
membosankan	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	mengasyikkan
tidak menarik	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	menarik
tak dapat diprediksi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	dapat diprediksi
lambat	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	cepat
konvensional	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	berdaya cipta
menghalangi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	mendukung
buruk	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	baik
rumit	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	sederhana
tidak disukai	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	menggembirakan
lazim	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	terdepan
tidak nyaman	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	nyaman
tidak aman	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	aman
tidak memotivasi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	memotivasi
tidak memenuhi ekspektasi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	memenuhi ekspektasi
tidak efisien	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	efisien
membingungkan	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	jelas
tidak praktis	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	praktis
berantakan	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	terorganisasi
tidak atraktif	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	atraktif
tidak ramah pengguna	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	ramah pengguna
konservatif	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	inovatif

Hasil dari kuisisioner dapat dilihat Tabel 4.74. Kuisisioner ini diisi oleh sepuluh guru / MGMP. Data yang diperoleh selanjutnya dikelompokkan menjadi enam kelompok yaitu daya tarik, kejelasan, efisiensi, ketepatan, stimulasi, dan kebaruan. Dari enam kelompok data diperoleh hasil sebagai berikut:

Tabel 4.74 Hasil UEQ

Graph to show the result relative to the benchmark							
Scale	Lower Border	Bad	Below Average	Above Average	Good	Excellent	Mean
Daya tarik	-1,00	0,65	0,44	0,41	0,22	0,78	2,316666667
Kejelasan	-1,00	0,53	0,37	0,47	0,45	0,68	1,85
Efisiensi	-1,00	0,5	0,34	0,47	0,33	0,86	2,125
Ketepatan	-1,00	0,7	0,36	0,34	0,2	0,9	2,175
Stimulasi	-1,00	0,52	0,48	0,31	0,19	1	2,45
Kebaruan	-1,00	0,24	0,39	0,33	0,38	1,16	2,15

Tabel 4.74 telah menunjukkan hasil dari UEQ yang kemudian dapat disampaikan dalam bentuk grafik seperti pada Gambar 4.24:



Gambar 4.24 Grafik hasil UEQ (Sagirani dkk, 2018)

Sprint review pada *sprint* ke-4 untuk *User Accaptance Testing* adalah:

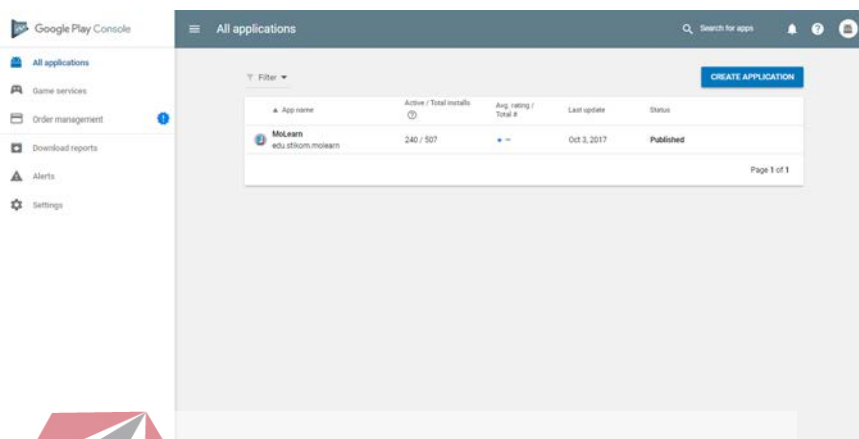
Tabel 4.75 *Sprint review* UAT

No.	Detil Pengerjaan	Status
1	Membuat quisioner.	Selesai
2	Membagikan kepada para siswa.	Selesai
3	Memproses hasil dari quisioner	Selesai

2. Upload ke Google Play

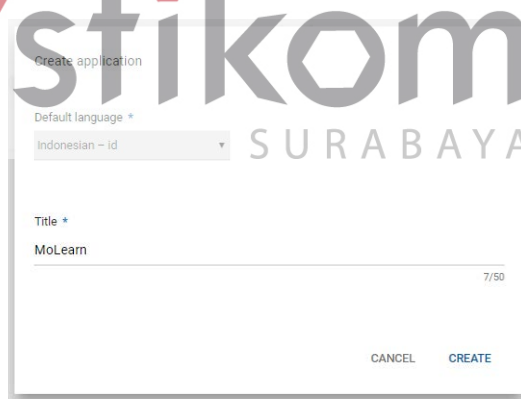
Setelah aplikasi telah selesai dikerjakan langkah selanjutnya adalah *upload* aplikasi ke *Google Play*. Tujuan *upload* ke *Google Play* agar pengguna Android lainnya dapat *download* aplikasi yang telah dikerjakan. Cara untuk *upload* ke dalam *Google Play* sebagai berikut:

1. Masuk pada link <https://play.google.com/apps/publish/> lewat browser.
2. Kemudian login menggunakan akun yang telah didaftarkan pada *Google Developer*. Tunggu sampai proses *login* selesai.
3. Kemudian akan muncul tampilan seperti Gambar 4.25:



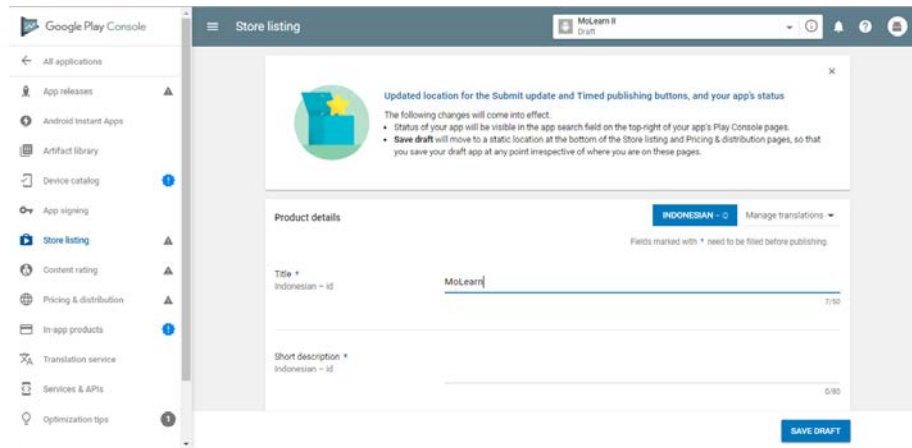
Gambar 4.25 Halaman Utama *Google Play*

4. Setelah itu pilihlah tombol “*Create Application*” untuk upload aplikasi baru kemudian akan muncul seperti Gambar 4.26:



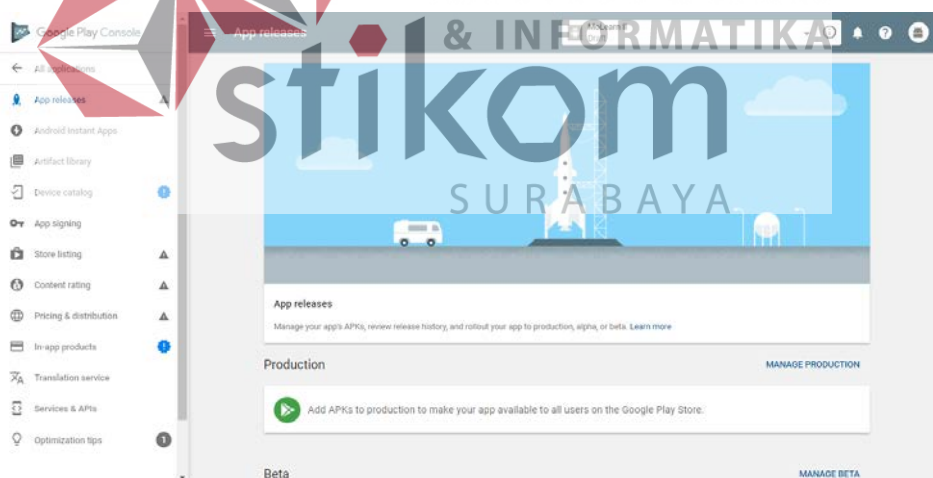
Gambar 4.26 *Create application*

Isi *Default language* Indonesian-ID dan *Title* “MoLearn”. Setelah itu pilih tombol “*create*”. Setelah itu halaman web akan menuju *store listing* pada Gambar 4.27:



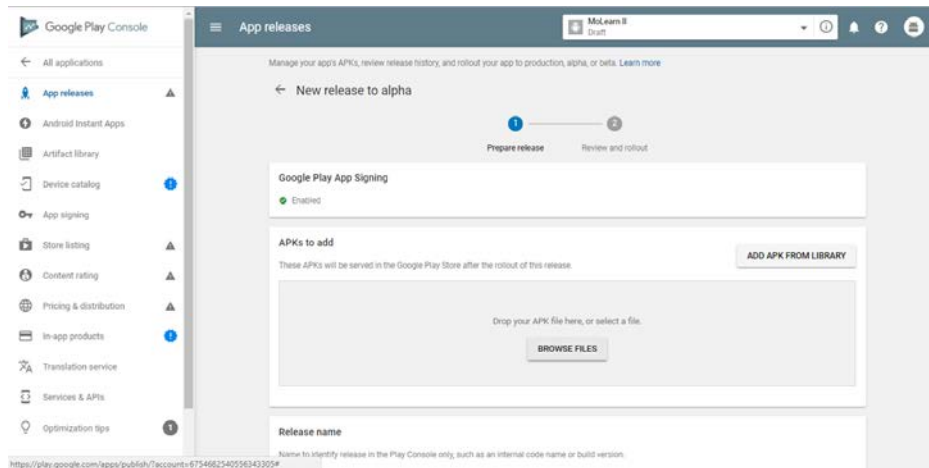
Gambar 4.27 Store Listing

5. Pada *store listing* terdapat *form* yang perlu diisi yaitu *Product Details*, *Graphic Assets*, *Categorization*, *Contact Details*, dan *Privacy Policy*. Setelah semua selesai diisi maka langkah selanjutnya masukkan file apk yang telah dibuat pada *Android Studio*.
6. Pilihlah menu *App Releases* untuk *upload* apk.



Gambar 4.28 App Releases

Pada *App Releases* terdapat tiga tipe yang memiliki *upload* yaitu *Production*, *Beta*, dan *Alpha*. Untuk pertama kali *upload* pilih *Alpha* kemudian akan masuk halaman *Alpha App Releases* kemudian pilih “*Create Release*”.



Gambar 4.29 Alpha App Releases

Setelah itu *upload file* apk yang ada pada *form APKs to add*. Setelah apk telah selesai *upload* langkah selanjutnya adalah tekan tombol *Review*. Aplikasi akan selesai *review* dalam waktu 24 jam. Setelah review maka aplikasi akan tampil di google play yang berstatus “Not Release”. Untuk menjadikan versi “Releases” maka perlu upload kembali di *Production*.



Gambar 4.30 Aplikasi MoLearn di Google Play

Sprint review upload ke Google Play adalah:

Tabel 4.76 Sprint review upload ke Google Play

No.	Detil Pengerjaan	Status
1	Mempersiapkan akun <i>Google</i> .	Selesai
2	<i>Generate</i> file apk.	Selesai
3	Upload ke <i>Google Play</i> .	Selesai

C. Sprint Retrospective

Sprint Retrospective untuk *sprint* ke-4 adalah:

1. Hasil UEQ perlu ditambahkan lagi penggunaanya untuk mendapatkan hasil yang lebih valid. Untuk pengembangan selanjutnya makan diperlukan UEQ yang lebih banyak.
2. *Upload* ke *Google Play* telah selesai tetapi perlu melakukan *update* aplikasi untuk pengembangan selanjutnya.



BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan hasil implmentasi dan evaluasi yang telah dikerjakan maka dapat disimpulkan bahwa:

1. Aplikasi *MoLearn* dapat menampilkan materi-materi pembelajaran untuk MGMP, guru, dan siswa.
2. Aplikasi *MoLearn* dapat menampilkan soal-soal berdasarkan standar kompetensi untuk MGMP, soal ujian untuk guru, dan soal ujian untuk siswa.
3. Aplikasi *MoLearn* dapat menampilkan tugas untuk siswa yang telah dibuat sebelumnya oleh guru.
4. Aplikasi *MoLearn* dapat digunakan untuk diskusi antar MGMP dan guru serta dapat digunakan untuk diskusi antar siswa dan guru.

5.2. Saran

Adapun saran untuk pengembangan aplikasi ini kedepannya yaitu:

1. Aplikasi *MoLearn* berbasis Android dapat dijalankan untuk sistem operasi iOS. Selain itu fitur untuk entri tugas siswa dapat dilakukan pada aplikasi *MoLearn* berbasis Android.
2. Penambahan mata pelajaran menjadi 22 mata pelajaran.

DAFTAR PUSTAKA

- Hariadi, B. (2000). *Pemanfaatan Sicyca sebagai Sumber Belajar di STIKOM Surabaya*. Tesis. Malang: Universitas Negeri Malang.
- Hariadi, B. (2011). *Pengaruh Strategi Pembelajaran Kooperatif tipe STAD Berbasis Web vs Teks dan Gaya Belajar Terhadap Hasil Belajar Mahasiswa*. Disertasi. Malang: Universitas Negeri Malang.
- Hasugian, J. (2011). *Perpustakaan Digital dan Digital Natives*. Medan: Universitas Nommensen.
- Indra, D. (2001). *Menuju Masyarakat Belajar. Menggagas Paradigma Baru Pendidikan*. Jakarta: Paramadina.
- Keegan, D. (2004). *Mobile Learning - The Next Generation of Learning*. Shanghai: The 18th Asian Association of Open Universities Annual Conference.
- Laouris, Y., & Eteokleous, N. (2017, Oktober 16). *We need an educational relevant definition of mobile learning*. Retrieved from MLearn: <http://www.mlearn.org.za/CD/papers/Laouris%20&%20Eteokleous.pdf>
- Medipour, Y., & Zerhkafi, H. (2013). Mobile Learning for Education: Benefits and Challenges. *International Journal of Computational Engineering Research Vol. 03*, 93-95.
- Sagirani, T., Sunarto, M., Hariadi, B., & Amelia, T. (2018). Model FRAME dalam Aplikasi Molearn Berbasis Android dengan Perhatian.
- Schwaber, K., & Sutherland, J. (2013). *Panduan Srum*. Newyork: ScrumInc.
- Suprianto, D., & Agustina, R. (2012). *Pemrograman Aplikasi Android*. Jakarta: PT. Buku Seru.
- Suyanto, A. S. (2017, Oktober 16). *Mengenal E-Learning*. Retrieved from <http://www.asep-hs.webugm.ac.id>
- Tohari, H. (2014). *Astah : Analisis serta Perancangan Sistem Informasi Melalui Pendekatan UML*. Yogyakarta: Andi.
- Traxler, J. (2007). Defining, Discussing and Evaluating Mobile Learning: The Moving Finger Writes and Having Writen. *The International Review in Open and Distance Learning, Vol. 8*, 1-13.