

BAB I

PENDAHULUAN

I.1 Latar Belakang

Lembaga pendidikan sebagai suatu wadah yang berfungsi sebagai tempat melaksanakan proses belajar mengajar, pelatihan dan pengembangan terhadap anak didiknya, di era globalisasi ini diharapkan bisa mengikuti perkembangan ilmu pengetahuan dalam bidang sistem pemantauan untuk menunjang kegiatan akademik anak maupun dapat memonitor aktivitas anak disekolah. Informasi yang diperlukan orang tua dapat membantu proses monitoring berjalan dengan efektif karena adanya komunikasi antara orang tua dan guru.

Menurut WHO *monitoring* merupakan suatu proses pengumpulan dan menganalisis informasi dari penerapan suatu program termasuk mengecek secara reguler untuk melihat apakah kegiatan atau program itu berjalan sesuai rencana sehingga masalah yang dilihat atau ditemui dapat diatasi. Penggunaan sistem monitoring bertujuan mengawasi serta mengecek sejumlah aktivitas yang dilakukan oleh siswa di lingkungan sekolah. Berdasarkan hasil observasi di SMPN 5 Pontianak mendapatkan suatu fakta bahwa selama ini kegiatan *monitoring* siswa kurang berjalan dengan optimal karena penggunaan buku penghubung memerlukan proses yang lama, sehingga informasi yang di terima tidak secepat dengan menggunakan teknologi. Buku penghubung di SMPN 5 Pontianak digunakan sebagai sarana monitoring dan evaluasi yang di lakukan oleh pihak sekolah. Diperlukan adanya suatu teknologi untuk mempercepat alur informasi yang diberikan buku penghubung. Setiap kegiatan siswa di sekolah perlu di informasikan kepada orang tua atau wali murid sebagai bahan pertimbangan untuk mendidik anak dirumah secara cepat.

Penggunaan teknologi dalam dunia pendidikan sudah banyak digunakan contohnya penggunaan ujian CBT (*Computer Based Test*). Pada era teknologi dan informasi banyak cara yang dapat dilakukan untuk mengakses informasi anak di sekolah salah satunya perangkat teknologi yang dapat digunakan adalah perangkat ponsel cerdas dengan menggunakan ponsel cerdas membuat orang tua dan guru dapat lebih mudah memberikan informasi yang berkaitan dengan siswa tersebut.

Sehingga *monitoring* yang dilakukan antara orang tua dan guru dapat berjalan dengan baik.

Berdasarkan uraian yang telah di paparkan dan dari hasil analisa di lapangan, maka perlunya dibangun aplikasi yang dibutuhkan pihak sekolah maupun orang tua dalam memonitor dan mengevaluasi aktivitas anak seperti absen siswa, nilai siswa, pengumuman yang berkaitan dengan sekolah agar menunjang proses belajar mengajar anak didik.

I.2 Perumusan Masalah

Berdasarkan latar belakang yang telah diuraikan dapat disimpulkan bahwa perumusan masalahnya adalah bagaimana membangun aplikasi *monitoring* dan evaluasi aktivitas anak seperti absen siswa, nilai siswa, pengumuman yang berkaitan dengan sekolah tersebut.

I.3 Tujuan Penelitian

Menghasilkan suatu aplikasi yang dapat memonitor dan mengevaluasi aktivitas anak di sekolah, sebagai media komunikasi antara orang tua dengan guru (wali kelas) yang diharapkan dapat membantu proses belajar mengajar.

I.4 Pembatasan Masalah

Pembatasan masalah dari penelitian yang akan dilakukan adalah :

1. Sistem menampilkan informasi tentang nilai rapot, nilai ulangan harian, absen dan pengumuman.
2. Admin menggunakan website untuk menginput data.
3. Orang tua dan guru (wali kelas), menggunakan android untuk mengakses aplikasi.

I.5 Sistematika Penulisan

Adapun sistematika dari penulisan tugas akhir ini disusun dalam lima bab yang terdiri dari Bab I Pendahuluan, Bab II Tinjauan Pustaka, Bab III Metodologi Penelitian dan Perancangan Sistem, Bab IV Hasil Perancangan dan Analisis Sistem, serta Bab V Penutup.

Bab I **Pendahuluan** adalah bab yang berisi latar belakang, perumusan masalah, tujuan penelitian, pembatasan masalah, dan sistematika penulisan.

Bab II **Tinjauan Pustaka** adalah bab yang berisi landasan teori berkaitan dengan penelitian yang akan dilakukan. Beberapa teori-teori yang terkait adalah pengertian aktivitas anak, monitoring, evaluasi, definisi Android, Android Studio, *Web Service*, *Unified Modeling Language* (UML), *Rapid Application Development* (RAD), Rest API, PHP, MySQL.

Bab III **Metodologi Penelitian dan Perancangan Sistem** adalah bab yang berisi tentang Bahan Penelitian, Alat yang Dipergunakan, Metode Penelitian, Variabel atau Data, Diagram Alir Penelitian, Diagram Alir Sistem, Perancangan Aplikasi, Perancangan UML, Perancangan Basis Data, Perancangan Antarmuka, serta Rencana Pengujian dengan menggunakan metode blackbox dan pengujian kuesioner.

Bab IV Implementasi dan Hasil Pengujian adalah bab yang berisi penjelasan mengenai implementasi pada sistem, screenshoot tampilan antarmuka sistem yang sudah jadi, serta analisis hasil uji coba. Setiap bagian sistem yang ditampilkan akan dilakukan analisis terlebih dahulu untuk mengarah kepada suatu kesimpulan.

Bab V Penutup adalah bab yang berisi kesimpulan dari penelitian yang telah dilakukan dan saran/rekomendasi untuk perbaikan, pengembangan atau kesempurnaan / kelengkapan penelitian yang telah dilakukan.

BAB II

TINJAUAN PUSTAKA

II.1 Aktivitas Anak

Aktivitas dalam proses belajar mengajar adalah rangkaian kegiatan yang meliputi keaktifan siswa dalam mengikuti pelajaran, bertanya hal yang belum jelas, mencatat, mendengar, berpikir, membaca dan segala kegiatan yang dilakukan yang dapat menunjang prestasi belajar. Proses perkembangan anak berkaitan dengan aktivitas yang dilakukan oleh anak, dengan semakin aktifnya anak perlunya peran orang tua dalam memonitoring hal tersebut, yang nantinya berguna dalam setiap proses peningkatan prestasi siswa (Sardiman, 2010).

Proses pembelajaran pada setiap satuan pendidikan dasar dan menengah harus interaktif, inspiratif, menyenangkan, menantang, dan memotivasi peserta didik untuk berpartisipasi aktif serta memberikan ruang yang cukup bagi prakarsa, kreativitas dan kemandirian sesuai dengan bakat, minat, dan perkembangan fisik serta psikologis peserta didik (Permendiknas RI No. 41, 2007). Apabila dicermati apa yang dikemukakan dalam Permen tersebut menunjukkan bahwa peran aktif siswa dalam pembelajaran merupakan suatu keharusan. Hal ini menunjukkan bahwa mengajar yang didesain guru harus berorientasi pada aktivitas siswa.

Interaksi yang baik antara orang tua dan guru yang benilai informasi tentang situasi dan kondisi setiap siswa, akan melahirkan suatu bentuk kerja sama yang dapat meningkatkan aktivitas belajar siswa baik di sekolah maupun di rumah. Hubungan kerja sama tersebut sangatlah penting. Sebab dengan adanya kerjasama tersebut orang tua dan guru dapat mengetahui kondisi siswa baik di lingkungan rumah maupun di lingkungan sekolah. Dalam hal ini guru dapat memperoleh informasi dari orang tua, bagaimana siswa tersebut ketika berada dirumah, apakah dirumah siwa mengulang pelajaran atau tidak dan sebagainya. Demikian juga orang tua juga dapat memperoleh informasi dari guru yaitu tentang bagaimana kemajuan siswa tersebut dalam belajar dan bagaimana sikap seorang siswa tersebut ketika dilingkungan sekolah.

II.2 ***Monitoring***

Monitoring adalah proses rutin pengumpulan data dan pengukuran kemajuan atas objektif program atau memantau perubahan yang fokus pada proses dan keluaran. *Monitoring* melibatkan perhitungan atas apa yang kita lakukan dan pengamatan atas kualitas dari layanan yang kita berikan. Fokus *monitoring* adalah pada komponen proses pelaksanaan program, baik menyangkut proses pengambilan keputusan, manajemen kelembagaan, maupun kegiatan belajar mengajar di sekolah. *Monitoring* adalah mendeterminasi apa yang telah dilaksanakan, maksudnya mengevaluasi prestasi kerja dan apabila perlu, menerapkan tindakan-tindakan korektif sehingga hasil pekerjaan sesuai dengan rencana yang telah ditetapkan (George R. Tery, 2006). *Monitoring* yang dilakukan untuk mengetahui perkembangan pelaksanaan program, apa hambatan yang dihadapi dan bagaimana cara mengatasi masalah tersebut. Dengan kata lain *monitoring* menekankan pada pemantauan proses pelaksanaan program dan pemberian saran untuk memperbaiki atau menyempurnakan masalah yang terjadi (Dit. SLTP Depdiknas, 2002).

Monitoring sekolah dapat dilakukan dengan dua cara, yaitu internal dan eksternal. Yang dimaksud dengan monitoring internal adalah yang dilakukan oleh sekolah sendiri yaitu kepala sekolah, guru, siswa, orang tua siswa, dan warga sekolah lainnya. Tujuan utamanya adalah untuk mengetahui tingkat kemajuan peserta didik sehubungan dengan sasaran-sasaran yang telah ditetapkan. Dengan cara ini diharapkan sekolah memahami tingkat ketercapaian sasaran, menemukan kendala-kendala yang dihadapi dan catatan-catatan bagi penyusunan program selanjutnya. Sedangkan monitoring eksternal dapat dilakukan oleh pihak luar sekolah, misalnya, pengawas, dinas pendidikan yang hasilnya dapat digunakan untuk penghargaan terhadap individu, sekolah dalam rangka meningkatkan iklim kompetisi sehat antar sekolah, kepentingan akuntabilitas publik, bagi perbaikan sistem yang ada keseluruhan dan membantu sekolah dalam mengembangkan diri.

II.3 **Evaluasi**

Evaluasi merupakan proses untuk menentukan tingkat keberhasilan dari pelaksanaan kegiatan yang telah diprogramkan melalui penentuan bobot kepentingan nilai dari suatu program, dan menentukan apakah hasil tersebut

efektif atau tidak.

Setiap hal yang berkaitan dalam proses pendidikan di sekolah selalu melibatkan unsur evaluasi. Namun keberadaan unsur ini tidak senantiasa dapat memberikan fungsi yang bersifat komprehensif bagi sekolah terutama yang menyangkut perbaikan dan pengembangannya. Banyak faktor yang berpengaruh berkenaan dengan fungsi evaluasi dalam peningkatan program sekolah, salah satunya adalah makna yang ditafsirkan dari konsep evaluasi itu sendiri. Menurut Arikunto (2003) mengungkapkan bahwa evaluasi adalah serangkaian kegiatan yang ditujukan untuk mengukur keberhasilan program pendidikan. Serta menurut Purwanto (2002) evaluasi adalah pemberian nilai terhadap kualitas sesuatu. Selain dari itu, evaluasi juga dapat dipandang sebagai proses merencanakan, memperoleh, dan menyediakan informasi yang sangat diperlukan untuk membuat alternatif-alternatif keputusan. Dengan demikian, Evaluasi merupakan suatu proses yang sistematis untuk menentukan atau membuat keputusan sampai sejauhmana tujuan-tujuan pengajaran telah dicapai oleh siswa

Evaluasi proses pembelajaran merupakan tahap yang perlu dilakukan oleh guru untuk menentukan kualitas pembelajaran. Kegiatan ini sering disebut juga sebagai refleksi proses pembelajaran, karena kita akan menemukan kelebihan dan kekurangan dari proses pembelajaran yang telah dilakukan. Dalam Permen No. 41 tahun 2007 tentang Standar proses dinyatakan bahwa evaluasi proses pembelajaran dilakukan untuk menentukan kualitas pembelajaran secara keseluruhan, mencakup tahap perencanaan proses pembelajaran, pelaksanaan proses pembelajaran, dan penilaian hasil pembelajaran. Dengan kerangka pemikiran ini tampak ada kaitan yang erat antara evaluasi dan mutu pendidikan di sekolah. Selanjutnya konsep evaluasi yang akan dibicarakan bertitik tolak dari tujuan evaluasi tersebut.

II.4 Android

Android merupakan *software environment* dibuat untuk perangkat *mobile*. Android menggunakan OS kernel Linux, UI yang berkualitas, *end-user application*, *code libraries*, *framework* aplikasi, mendukung multimedia dan banyak lagi. Aplikasi yang dibuat untuk Android menggunakan bahasa pemrograman Java.

Ada dua cara untuk membangun atau membuat aplikasi berbasis android, yaitu memiliki perangkat telepon seluler yang berbasis android langsung dan menggunakan emulator yang sudah disediakan oleh Google. Sebelum memulai membangun aplikasi berbasis android, diperlukan beberapa perangkat, antara lain :

- *Android Studio*
- *Sun's Java Development Kit (JDK)*.
- *The Android Software Developer's Kit (SDK)*.

Windows, Linux, dan Mac OS X merupakan sistem operasi yang dapat digunakan untuk pengembangan pembuatan aplikasi berbasis Android dengan memanfaatkan Android SDK (Elian, S., & Djanali, 2012).

Pada penelitian ini, akan ditampilkan data nilai, absen, dan pengumuman menggunakan Android Studio. Android pada penelitian ini berfungsi sebagai sistem operasi yang akan menampilkan data dari *mysql* dengan *php* dengan format *JSON (JavaScript Object Notation)*.

II.5 REST

REST merupakan singkatan dari REpresentative State Transfer. Pertama kali dikemukakan dalam disertasi seorang program doktor bernama Roy Thomas Fielding pada tahun 2000 (Rozali, 2011). REST adalah sebuah metode dalam menyampaikan resource melalui media web. Sedangkan resource sendiri didefinisikan sebagai segala sesuatu yang dapat disimpan didalam sebuah komputer dan ditampilkan sebagai urutan bit, misalnya sebuah dokumen, tabel dalam sistem basis data, atau hasil dari sebuah perhitungan (Sandoval, 2008). REST (REpresentational State Transfer) adalah model arsitektur yang pada dasarnya memanfaatkan teknologi dan protokol yang sudah ada seperti HTTP (Hypertext Transfer Protocol) dan XML (Riyadi, 2013). Pada arsitektur REST, REST server menyediakan resources (sumber daya/data) dan REST client mengakses dan menampilkan resource tersebut untuk penggunaan selanjutnya. Setiap resource diidentifikasi oleh URIs (Universal Resource Identifiers) atau global ID. Resource tersebut direpresentasikan dalam bentuk format teks, JSON atau XML. Pada umumnya formatnya menggunakan JSON dan XML.

II.6 PHP

PHP Hypertext Preprocessor (PHP) merupakan salah satu yang banyak digunakan sebagai *server-side scripting language* untuk mengembangkan aplikasi berbasis *web*. PHP adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu baru atau *up to date*. Semua *script PHP* dieksekusi pada server dimana *script* tersebut dijalankan (Anhar, 2010).

II.7 JSON (JavaScript Object Nation)

JSON (*JavaScript Object Nation*) adalah sebuah standar yang berbasis teks yang didesain untuk pertukaran data yang mudah dibaca manusia (Sriparasa, 2013).

Kegunaan dari JSON :

1. Digunakan saat menulis aplikasi berbasis *JavaScript* yang memerlukan ekstensi *browser* dan *website*.
2. Format JSON digunakan untuk mentransmisikan data yang berstruktur melalui jaringan.
3. Biasanya digunakan untuk mentransmisikan data antara server dan aplikasi *web*.
4. *Web Service* dan *API* menggunakan format JSON untuk menyediakan data publik.
5. Dapat digunakan pada bahasa pemrograman modern.

II.8 Web Service

Web service adalah aplikasi sekumpulan data (*database*), perangkat lunak (*software*) atau bagian dari perangkat lunak yang dapat diakses secara *remote* oleh berbagai piranti dengan sebuah perantara tertentu. Secara umum, *web service* dapat didefinisikan dengan menggunakan URL seperti hanya *web* pada umumnya. Namun yang membedakan *web service* dengan *web* pada umumnya adalah interaksi yang diberikan oleh *web service*. Berbeda dengan URL *web* pada umumnya, URL *web service* hanya mengandung kumpulan informasi, perintah, konfigurasi atau sintaks yang berguna membangun sebuah fungsi-fungsi tertentu

dari aplikasi.

Web service adalah sistem perangkat lunak yang di rancang untuk mendukung interaksi yang bisa beroperasi machine-to-machine di atas jaringan. Web service mempunyai alat penghubung yang di uraikan di dalam format machine procesable (secara spesifik WSDL). Sistem lain saling berhubungan dengan web service di dalam cara yang telah ditentukan oleh deskripsinya yang menggunakan pesan soap, secara khas disampaikan menggunakan HTTP dengan XML seirialization, bersama dengan standar lain yang terkait dengan web (Booth et al., 2004).

Web service dapat diartikan juga sebuah metode pertukaran data, tanpa memperhatikan dimana sebuah *database* ditanamkan, dibuat dalam bahasa apa sebuah aplikasi yang mengonsumsi data, dan di *platform* apa sebuah data itu dikonsumsi. *Web service* mampu menunjang interopabilitas. Sehingga *web service* mampu menjadi sebuah jembatan penghubung antara berbagai sistem yang ada.

II.9 MySQL

MySQL adalah *system management database* relasional. Suatu relasional data dalam tabel terpisah. Hal ini memungkinkan kecepatan dan fleksibilitas. Tabel-tabel yang dihubungkan dengan relasi yang ditentukan membuatnya bisa mengkombinasikan data dari beberapa tabel dari suatu permintaan. Bagian *SQL* dari kata *MySQL* berasal dari *Structured Query Language* yaitu bahasa yang paling umum digunakan untuk mengakses *database*. Konektivitas, kecepatan, dan keamanannya membuat *MySQL* cocok untuk pengaksesan *database* pada internet.

MySQL ini tergolong suatu software yang open source dan berlisensi GPL atau *General Public License*. Lisensi GPL ini hanya ditujukan pada perangkat lunak tertentu untuk keperluan proyek GNU, inilah yang menjadi faktor banyaknya pengguna *MySQL* di seluruh dunia. Selain mudah digunakan, anda dapat mengelola data dengan lebih efektif karena menggunakan script atau Bahasa tertentu dan secara otomatis akan menjadi perintah ke sistem.

II.9.1 Perintah-perintah MySQL

Query pada MySQL adalah *query* yang diciptakan dengan menggunakan pernyataan-pernyataan SQL. Pernyataan (*statement*) SQL dapat digolongkan atas tiga golongan, yaitu:

1. *Data Definition Language* (DDL) yang mendefinisikan struktur data. Perintah-perintah SQL yang termasuk DDL antara lain *create*, *alter*, dan *drop*.
2. *Data Manipulation Language* (DML) yang mencari (*query*) dan mengubah (*modify*) suatu tabel. Perintah-perintah SQL yang termasuk DML antara lain *select*, *insert*, *update*, dan *delete*.
3. *Data Control Language* (DCL) yang mengatur hak-hak (*privilege*) untuk seorang pemakai *database*. Perintah-perintah SQL yang termasuk DCL antara lain *grant* dan *revoke*.

II.9.2 Tipe Data MySQL

Tipe data MySQL adalah data yang terdapat dalam sebuah tabel berupa *field-field* yang berisi nilai dari data tersebut. Nilai data dalam *field* memiliki tipe sendiri-sendiri. MySQL mengenal beberapa tipe data *field* yaitu:

1. **Tipe Data Numerik**

Tipe numerik dibedakan dalam dua macam kelompok, yaitu *integer* dan *floating point*. *Integer* digunakan untuk data bilangan bulat sedangkan *floating point* digunakan untuk bilangan desimal. Tipe-tipe data yang termasuk dalam tipe data Numerik adalah *tinyint*, *smallint*, *mediumint*, *int*, *float*, *double*.

2. **Tipe Data String**

String adalah rangkaian karakter. Tipe-tipe data yang termasuk dalam tipe data *string* adalah *char*, *varchar*, *tinytext*, *text*, *mediumtext*, *longtext*.

3. **Tipe Data Tanggal**

Tipe data tanggal biasanya digunakan untuk tanggal atau waktu. Masing-masing tipe mempunyai kisaran nilai tertentu. MySQL akan memberikan peringatan kesalahan (*error*) apabila tanggal atau waktu yang dimasukkan salah. Tipe data yang termasuk dalam tipe data tanggal adalah *datetime*, *date*, *timestamp*, *time*, *year*.

II.10 Rapid Application Development (RAD)

Rapid Application Development (RAD) adalah strategi siklus hidup yang ditujukan untuk menyediakan pengembangan yang jauh lebih cepat dan mendapatkan hasil dengan kualitas yang lebih baik dibandingkan dengan hasil

yang dicapai melalui siklus tradisional (McLeod, 2002). RAD merupakan gabungan dari bermacam-macam teknik terstruktur dengan teknik *prototyping* dan teknik pengembangan *joint application* untuk mempercepat pengembangan sistem aplikasi (Bentley, 2004). Dari definisi-definisi konsep RAD ini, dapat dilihat bahwa pengembangan aplikasi dengan menggunakan metode RAD ini dapat dilakukan dalam waktu yang relatif lebih cepat.

Pemaparan konsep yang lebih spesifik lagi dijelaskan oleh Pressman (2005) dalam bukunya, “*Software Engineering: A Practitioner’s Approach*”. Ia mengatakan bahwa RAD adalah proses model perangkat lunak inkremental yang menekankan siklus pengembangan yang singkat. Model RAD adalah sebuah adaptasi “kecepatan tinggi” dari model *waterfall*, di mana perkembangan pesat dicapai dengan menggunakan pendekatan konstruksi berbasis komponen. Jika tiap-tiap kebutuhan dan batasan ruang lingkup projek telah diketahui dengan baik, proses RAD memungkinkan tim pengembang untuk menciptakan sebuah “sistem yang berfungsi penuh” dalam jangka waktu yang sangat singkat. Dari penjelasan Pressman (2012) ini, satu perhatian khusus mengenai metodologi RAD dapat diketahui, yakni implementasi metode RAD akan berjalan maksimal jika pengembang aplikasi telah merumuskan kebutuhan dan ruang lingkup pengembangan aplikasi dengan baik.

Sedangkan menurut Kendall (2010), RAD adalah suatu pendekatan berorientasi objek terhadap pengembangan sistem yang mencakup suatu metode pengembangan serta perangkat-perangkat lunak. RAD bertujuan mempersingkat waktu yang biasanya diperlukan dalam siklus hidup pengembangan sistem tradisional antara perancangan dan penerapan suatu sistem informasi. Pada akhirnya, RAD sama-sama berusaha memenuhi syarat-syarat bisnis yang berubah secara cepat.



Gambar 2.1 Gambar Alir RAD (Kendall, 2010)

II.10.1 Fase Dan Tahapan Pengembangan Aplikasi

Menurut Kendall (2010), terdapat tiga fase dalam RAD yang melibatkan penganalisis dan pengguna dalam tahap penilaian, perancangan, dan penerapan. Adapun ketiga fase tersebut adalah *requirements planning* (perencanaan syarat-syarat), *RAD design workshop* (*workshop desain RAD*), dan yang terakhir *implementation* (implementasi). Sesuai dengan metodologi RAD menurut Kendall (2010), berikut ini adalah tahap-tahap pengembangan aplikasi dari tiap-tiap fase pengembangan aplikasi.

1. *Requirements Planning* (Perencanaan Syarat-Syarat)

Dalam fase ini, pengguna dan penganalisis bertemu untuk mengidentifikasi tujuan-tujuan aplikasi atau sistem serta untuk megidentifikasi syarat-syarat informasi yang ditimbulkan dari tujuan-tujuan tersebut. Orientasi dalam fase ini adalah menyelesaikan masalah-masalah perusahaan. Meskipun teknologi informasi dan sistem bisa mengarahkan sebagian dari sistem yang diajukan, fokusnya akan selalu tetap pada upaya pencapaian tujuan-tujuan perusahaan (Kendall, 2010).

2. *RAD Design Workshop* (*Workshop Desain RAD*)

Fase ini adalah fase untuk merancang dan memperbaiki yang bisa digambarkan sebagai *workshop*. Penganalisis dan pemrogram dapat bekerja membangun dan menunjukkan representasi visual desain dan pola kerja kepada pengguna. *Workshop* desain ini dapat dilakukan selama beberapa hari tergantung dari ukuran aplikasi yang akan dikembangkan. Selama *workshop* desain RAD, pengguna merespon prototipe yang ada dan penganalisis memperbaiki modul-modul yang dirancang berdasarkan respon

pengguna. Apabila sorang pengembangnya merupakan pengembang atau pengguna yang berpengalaman, Kendall menilai bahwa usaha kreatif ini dapat mendorong pengembangan sampai pada tingkat terakselerasi (Kendall, 2010).

3. *Implementation* (Implementasi)

Pada fase implementasi ini, penganalisis bekerja dengan para pengguna secara intens selama *workshop* dan merancang aspek-aspek bisnis dan nonteknis perusahaan. Segera setelah aspek-aspek ini disetujui dan sistem-sistem dibangun dan disaring, sistem-sistem baru atau bagian dari sistem diujicoba dan

II.11 Modelling Language (UML)

Untuk membantu dalam pengembangan perangkat lunak dikenal istilah pemodelan. Salah satu pemodelan yang saat ini paling banyak digunakan oleh pengembang perangkat lunak adalah UML (*Unified Modelling Language*). UML adalah standar bahasa yang sering digunakan dalam bidang industri untuk mendefinisikan *requirement* (kebutuhan), membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.

Inilah beberapa tujuan atau fungsi dari penggunaan UML, yang diantaranya:

1. Dapat memberikan bahasa permodelan visual kepada pengguna dari berbagai macam pemrograman maupun proses rekayasa.
2. Dapat menyatukan praktek-praktek terbaik yang ada dalam permodelan.
3. Dapat memberikan model yang siap untuk digunakan, merupakan bahasa permodelan visual yang ekspresif untuk mengembangkan sistem dan untuk saling menukar model secara mudah.
4. Dapat berguna sebagai blue print, sebab sangat lengkap dan detail dalam perancangannya yang nantinya akan diketahui informasi yang detail mengenai koding suatu program.
5. Dapat memodelkan sistem yang berkonsep berorientasi objek, jadi tidak hanya digunakan untuk memodelkan perangkat lunak (*software*) saja.

6. Dapat menciptakan suatu bahasa permodelan yang nantinya dapat dipergunakan oleh manusia maupun oleh mesin.

UML terbagi atas 3 (tiga) kategori, yaitu diagram struktur (*structure diagram*), diagram kelakuan sistem (*behaviour diagram*), dan diagram interaksi (*interaction diagram*) (A. S & Shalahuddin, 2014).

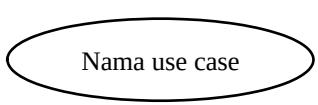
4. *Structure diagram*, yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan. *Structure diagram* terdiri dari *class diagram*, *object diagram*, *component diagram*, *composite diagram*, *package diagram*, dan *deployment diagram*.
5. *Behaviour diagram*, yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem. *Behaviour diagram* terdiri dari *use case diagram*, *activity diagram*, *state machine system*.
6. *Interaction diagram*, yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem. *Interaction diagram* terdiri dari *sequence diagram*, *communication diagram*, *timing diagram*, *interaction overview diagram*.

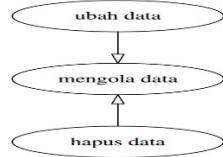
II.11.1 Use Case Diagram

Use case diagram merupakan pemodelan untuk kelakuan (*behaviour*) sistem yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi itu. *Use case* memiliki beberapa elemen antara lain adalah ***overview information, relationship, dan flow of event***.

Berikut adalah simbol-simbol yang ada pada diagram *use case* :

Tabel 2.1 Simbol-simbol Use Case Diagram

No	Simbol	Deskripsi
1.	<i>Use case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya

		dinyatakan dengan menggunakan kata kerja diawal fase nama <i>use case</i> .
2.	Aktor/ <i>actor</i> 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem yang akan dibuat di luar sistem yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
3.	Generalisasi/ <i>generalization</i> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya :  <p>Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum).</p>
4.	Asosiasi/ <i>association</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpatisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
5.	Ekstensi/ <i>extend</i> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan itu, mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek, biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan,

		<p>misal</p> <p>Arah panah mengarah pada <i>use case</i> yang ditambahkan, biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.</p>
--	--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Sumber : (A. S & Shalahuddin, 2014, p. 156).

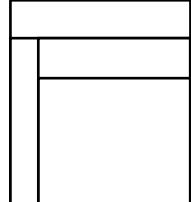
II.11.2 Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu di perhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Berikut adalah simbol-simbol yang ada pada diagram aktivitas :

Tabel 2.2 Simbol-simbol *Activity Diagram*

No	Simbol	Deskripsi
1.	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.	Percabangan/decision 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.

4.	Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.	Status akhir 	Status akhir yang dilakukan oleh sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.	Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Sumber : (A. S & Shalahuddin, 2014).

II.11.3 Class Diagram

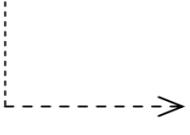
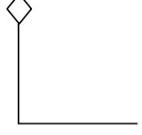
Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan *method* atau operasi. Berikut penjelasan atribut dan *method* :

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau *method* adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Berikut adalah simbol-simbol yang ada pada diagram kelas :

Tabel 2.3 Simbol-simbol *Class Diagram*

No	Simbol	Deskripsi
1.	Kelas 	Kelas pada struktur sistem.
2.	Antarmuka/ <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.

3.	Asosiasi/association 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4.	Asosiasi berarah/directed association 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5.	Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus).
6.	Kebergantungan /dependensi 	Relasi antar kelas dengan makna kebergantungan antar kelas.
7.	Agregasi/aggregation 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>).

Sumber : (A. S & Shalahuddin, 2014).

II.11.4 Sequence Diagram

Sequence Diagram adalah salah satu dari diagram - diagram yang ada pada UML, sequence diagram ini adalah diagram yang menggambarkan kolaborasi dinamis antara sejumlah object. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara object juga interaksi antara object. Sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

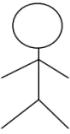
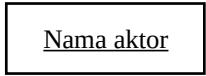
Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dengan *message* yang dikirimkan dan diterima antar objek. Menggambarkan diagram sekuen harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki

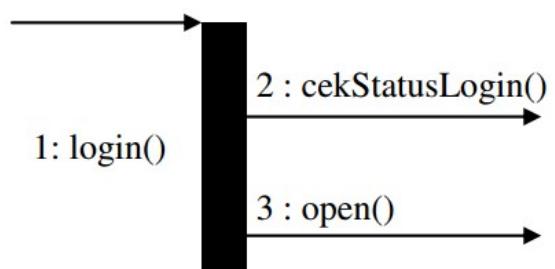
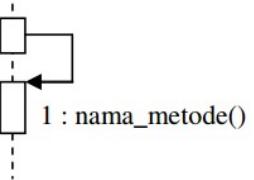
kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*. Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisan *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup dalam diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

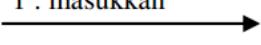
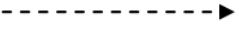
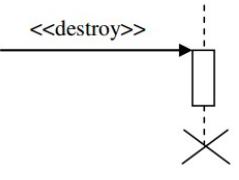
Dalam sequence diagram, setiap object hanya memiliki garis yang digambarkan garis putus-putus ke bawah. Pesan antar object digambarkan dengan anak panah dari object yang mengirimkan pesan ke object yang menerima pesan.

Berikut adalah simbol-simbol yang ada pada diagram sekuen :

Tabel 2.4 Simbol-simbol Sequence Diagram

No	Simbol	Deskripsi
1.	Aktor  Atau  Tanpa waktu aktif	Orang, proses, atau sistem lain yang berinteraksi dengan sistem yang akan dibuat diluar sistem yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar, tapi aktor belum tentu merupakan orang, biasanya dinyatakan dalam menggunakan kata benda diawali frase nama aktor.
2.	Garis hidup/ <i>lifeline</i> 	Menyatakan kehidupan suatu objek

3.	Objek Nama objek : nama kelas	Menyatakan objek yang berinteraksi pesan
4.	Waktu aktif 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya</p>  <p>Maka cekStatusLogin() dan open() dilakukan didalam metode login(). Aktor tidak memiliki waktu aktif.</p>
5.	Pesan tipe <i>create</i> →	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
6.	Pesan tipe <i>call</i> 1 : nama_metode()	 <p>Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada</p>

		pada diagram kelas sesuai dengan kelas objek yang berinteraksi.
7.	Pesan tipe <i>send</i> 1 : masukkan 	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8.	Pesan tipe <i>return</i> 1 : keluaran 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
9.	Pesan tipe <i>destroy</i> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada <i>create</i> maka ada <i>destroy</i> .

Sumber : (A. S & Shalahuddin, 2014).

II.12 Pengujian Perangkat Lunak

Ada beberapa metode yang dapat digunakan untuk memilih data pengujian, salah satunya adalah metode *black box*. Metode *black box* metode pengujian perangkat lunak yang menguji fungsionalitas aplikasi yang bertanggungjawab terhadap struktur internal atau kerja. Pengetahuan khusus dari kode aplikasi / struktur internal dan pengetahuan pemrograman pada umumnya tidak diperlukan. Uji kasus dibangun di sekitar spesifikasi dan persyaratan, yakni, aplikasi apa yang seharusnya dilakukan. Menggunakan deskripsi eksternal perangkat lunak, termasuk spesifikasi, persyaratan, dan desain untuk menurunkan uji kasus. Tes ini dapat menjadi fungsional atau non-fungsional, meskipun biasanya fungsional. Perancangan uji memilih input yang valid dan tidak valid dan menentukan output yang benar. Tidak ada pengetahuan tentang struktur internal

benda uji itu. Metode *Black-Box* yaitu data pengujian dipilih berdasarkan spesifikasi masalah tanpa memperhatikan detail internal dari program (Sukamto, 2009), untuk memeriksa apakah program dapat berjalan dengan benar. Pemilihan data pengujian paling tidak harus dipilih dengan cara berikut (Sukamto, 2009):

- a. *Extreme values*, banyak program *error* pada suatu batas *range* dari aplikasi.
- b. *Ilegal values*, yaitu suatu data atau nilai yang tidak diperbolehkan maupun data yang tidak berguna.
- c. *values*, yaitu data yang mudah diperiksa.
- d. *Typical realistic value*, yaitu mencoba program dengan data pengujian untuk melihat bagaimana program melakukannya. Data ini harus cukup sederhana sehingga hasilnya dapat dihitung secara manual.

Adapun teknik pengujian secara *Black Box* menurut Rosa A.S, (2009) sebagai berikut:

- a. *Boundary Value Analysisist (BVA)/Limit Testing* adalah banyak kesalahan terjadi pada kesalahan masukan. BVA mengijinkan untuk menyeleksi kasus uji yang menguji batasan nilai *input*. BVA merupakan komplemen dari *equivalence partitioning*. Lebih pada memilih elemen-elemen di dalam kelas ekivalen pada bagian sisi batas dari kelas.
- b. *Comparison Testing* adalah pengujian pada beberapa aplikasi *reliability* dari sebuah perangkat lunak. Redundansi perangkat keras dan perangkat lunak mungkin digunakan untuk meminimalisir kesalahan (*error*). Untuk redundansi perangkat lunak, gunakan tim yang terpisah untuk mengembangkan setiap versi perangkat lunak yang independen. Uji setiap versi dengan data yang sama untuk memastikan semua versi menghasilkan keluaran yang sama. Jalankan semua versi dengan paralel dan perbandingan keluaran secara *real-time*.
- c. *Sample Testing* adalah pengujian yang melibatkan beberapa nilai yang terpilih dari sebuah kelas ekivalen. Mengintegrasikan nilai pada kasus uji dan nilai-nilai yang terpilih mungkin dipilih dengan urutan tertentu atau interval tertentu.

- d. *Robustness Testing* adalah pengujian dengan data *input* dipilih diluar spesifikasi yang telah didefinisikan. Tujuan dari pengujian ini adalah membuktikan bahwa tidak ada kesalahan jika masukan tidak valid.
- e. *Behavior Testing* adalah pengujian dengan hasil uji tidak dapat dievaluasi jika hanya melakukan pengujian sekali, tapi dapat dievaluasi jika pengujian dilakukan beberapa kali, misalnya pada pengujian struktur data *stack*.
- f. *Requirement Testing* adalah spesifikasi kebutuhan yang terasosiasi dengan perangkat lunak (*input*, *output*, fungsi, performansi) diidentifikasi pada tahap spesifikasi kebutuhan dan desain. *Requirement testing* melibatkan pembuatan kasus uji untuk setiap spesifikasi kebutuhan yang terkait dengan program.
- g. *Performance Testing* adalah mengevaluasi kemampuan program untuk beroperasi dengan benar dipandang dari sisi acuan kebutuhan misalnya: aliran data, ukuran pemakaian memori, kecepatan eksekusi dan lain-lain. Untuk mencari tahu beban kerja atau kondisi konfigurasi program dan dapat digunakan untuk menguji batasan lingkungan program.
- h. Uji Ketahanan (*Endurance Testing*) adalah melibatkan kasus uji yang diulang-ulang dengan jumlah tertentu dengan tujuan untuk mengevaluasi program apakah sesuai dengan spesifikasi kebutuhan.
- i. *Equivalence Partitioning* adalah membagi input menjadi kelas-kelas data yang dapat digunakan untuk menggenerasi kasus uji dengan tujuan untuk menemukan kelas-kelas kesalahan. Selain itu, *equivalence partitioning* berdasarkan pada kesamaan kelas-kelas kondisi input. Sebuah kelas yang ekivalen merepresentasikan kumpulan status/kondisi yang valid atau tidak valid. Sebuah kondisi input dapat berupa nilai numerik yang spesifik, rentang nilai, kumpulan nilai yang berkaitan, atau kondisi *boolean*.
- j. Uji Sebab-Akibat (*Cause-Effect Relationship Testing*) adalah teknik yang merupakan suplemen dari *equivalence testing* dengan menyediakan cara untuk memilih kombinasi data *input* dan melibatkan kondisi *input* (*cause*) dan kondisi *output* (*effect*) untuk mencegah pendefinisian kasus uji yang terlalu banyak.
- k. *Acceptance testing* adalah pengujian terakhir sebelum sistem dipakai oleh *user* yang melibatkan pengujian dengan data dari pengguna sistem. *User*

acceptance test adalah uji terima perangkat lunak yang dilakukan ditempat pengguna oleh user perangkat lunak tersebut. Tujuan pengujian ini adalah untuk menguji apakah sistem sudah sesuai dengan apa yang tertuang dalam spesifikasi fungsional sistem (*validation*).

II.13 Kajian Terkait

Dalam hal penyusunan aplikasi *monitoring* dan evaluasi aktivitas anak di sekolah SMP 5 Kota Pontianak berbasis android di kaji dengan mengambil dari beberapa penelitian yang telah dilakukan sebelumnya, bisa kita lihat pada penelitian Hidayat menggunakan metode RAD dalam metode penelitiannya, kedua Utama, Nurhayaty dan Windsari mereka membuat aplikasi monitoring menggunakan BlackBox dan yang terakhir Neforawati, Fareza dan Juniarti mereka memonitoring anak menggunakan android. Kajian terkait dapat dilihat pada tabel 2.5.

Tabel 2.5 Tabel Perbandingan Penelitian

No	Penulis	Judul	Keterangan
1.	Hidayat, M. I. (2016), Program Studi Sistem Informasi, Fakultas Sains dan Teknologi UIN Alaudin Makasar	Aplikasi <i>Monitoring</i> Aktivitas Anak Santri Berbasis Android	<ul style="list-style-type: none"> - Aplikasi berbasis <i>mobile</i> dengan menggunakan Android Studio & MySQL sebagai <i>database</i>. - Menggunakan metode <i>Rapid Application Development</i> (RAD) - Menampilkan data santri seperti SPP, absen, hafidz, prestasi dan pelanggaran santri

No	Penulis	Judul	Keterangan
2.	Utama, H. P., Nurhayaty, O. K., & Windasari, I. P. (2016), Program Studi Sistem Komputer, Fakultas Teknik, Universitas Diponegoro	Pembuatan Aplikasi Memantau Anak Berbasis Android Menggunakan <i>Location Based Service</i>	<ul style="list-style-type: none"> - Aplikasi berbasis <i>mobile</i> dengan menggunakan Android Studio & MySQL sebagai <i>database</i>. - Pengujian sistem menggunakan metode <i>Blackbox</i>. - Aplikasi ini untuk memantau lokasi anak dengan <i>Google Maps</i>.
3.	Neforawati, I., Fareza, M. I., & Juniarti, V. (2015), Program Studi Teknik Informatika Politeknik Negeri Jakarta	Rancang Bangun Aplikasi Sistem Informasi Monitoring Absensi Mahasiswa Politeknik Negeri Jakarta Menggunakan Teknologi NFC Pada Android	<ul style="list-style-type: none"> - Menggunakan NFC (Near Field Communication) - Aplikasi ini berbasis <i>mobile</i> dengan menggunakan Android Studio & MySQL sebagai <i>database</i>. - Sistem ini menyimpan absensi dengan perantara NFC yang memungkinkan interaksi antar

			perangkat
--	--	--	-----------

Tabel 2.6 Tabel Penelitian yang Akan Dilakukan

No	Penulis	Judul	Keterangan
1.	Muhammad Rizki Firdaus (2017), Universitas Tanjungpura Pontianak	Sistem Monitoring Aktivitas Anak di Dinas Pendidikan Kota Pontianak Berbasis Android	<ul style="list-style-type: none"> - Sistem Menampilkan nilai siswa, absen siswa, pengumuman yang berkaitan dengan siswa dan fitur chat - Menggunakan metode <i>Rapid Application Development</i> (RAD) - Pengujian sistem menggunakan metode <i>Blackbox</i> dan Kuesioner - Menggunakan metode Rest Api dalam penarikan data JSON.

BAB III

METODOLOGI PENELITIAN

III.1 Data Penelitian

Data penelitian berupa data sekolah yang akan dijadikan studi kasus yaitu SMPN 5 Pontianak adapun data yang diambil dari sekolah SMPN 5 Pontianak adalah data siswa, data guru, nilai siswa, absen siswa serta data-data lainnya yang dianggap perlu.

III.1.1 Alat yang Digunakan

Alat penelitian yang digunakan dalam penelitian ini adalah:

- a. Diagram alir sistem, digunakan untuk menggambarkan bagaimana sistem dapat menghasilkan aplikasi yang dapat digunakan orang tua dan guru dalam monitoring dan evaluasi aktivitas anak di sekolah.
- b. *Unified Modelling Language* (UML). UML digunakan untuk menggambarkan batasan sistem dan fungsi-fungsi sistem secara umum.
- c. *Use Case Diagram*, untuk mengetahui fungsi apa saja yang ada di dalam sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

III.1.2 Perangkat Keras

Perangkat keras yang digunakan untuk merancang aplikasi dalam penelitian ini adalah:

- a. Satu unit laptop MSI GP62M, dengan spesifikasi Intel Core i7-7700HQ @1.80GHz, 8.00 GB RAM DDR3, Harddisk 1000 GB
- b. Smartphone Xiaomi Mi 5 dengan spesifikasi Qualcomm Snapdragon 640 octa-core, 32 GB internal memory, 3 GB RAM, GPS, Android versi 8.1.0.

III.1.3 Perangkat Lunak

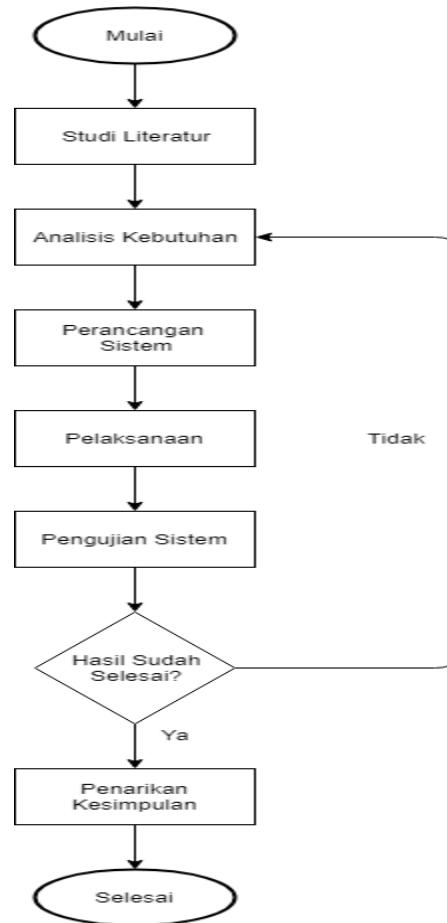
Perangkat lunak yang digunakan dalam penelitian ini adalah:

- a. Sistem Operasi Windows 10
- b. Android Studio 3.1.2 sebagai aplikasi untuk pengembangan aplikasi Android.
- c. Sublime Text 3 sebagai aplikasi untuk text editor.
- d. PHPMyAdmin 4.6.5.2 sebagai antarmuka MySQL.
- e. XAMPP version 3.2.2 sebagai aplikasi untuk membangun server local.
- f. PHP version 7.1.1 sebagai bahasa pemrograman yang digunakan.
- g. MySQL 5.14.1 sebagai pengelola sistem database.

- h. Laravel version 5.7 sebagai Framework bahasa pemrograman php.
- i. Google Chrome sebagai browser untuk menjalankan aplikasi.
- j. Adobe Xd sebagai mock up dalam perancangan design android.

III.2 Diagram Alir Penelitian

Tahapan pada penelitian yang akan dilakukan dijelaskan pada Gambar 3.1 sebagai berikut.



Gambar 3.1 Diagram Alir Penelitian

Berdasarkan pada Gambar 3.1 dapat dijelaskan diagram alir penelitian sebagai berikut.

III.2.1 Studi Literatur

Studi literatur dilakukan dengan mencari referensi ilmu pengetahuan yang berhubungan dengan sistem yang akan dibuat seperti referensi tentang kritik dan saran, UML (*Unified Modeling Language*) maupun melalui penelitian-penelitian sebelumnya.

III.2.2 Analisis Kebutuhan

Sistem dibangun berbasis android dan *website*, *website* bertugas sebagai server yang nanti nya setiap data yang masuk akan tersimpan di database. Pengguna pada aplikasi ini yaitu orang tua siswa yang dapat melihat data nilai rapot dan harian, data absen, data pengumuman, serta guru yang dapat melihat nilai dan memasukkan nilai harian di aplikasi android.

Pada aplikasi *web* pengguna yaitu admin bertugas memasukkan data-data yang diperlukan dalam aplikasi ini seperti data siswa, data guru, data nilai, data absen serta data pengumuman yang berkaitan dengan sekolah yang nantinya terhubung ke pengguna pada aplikasi android.

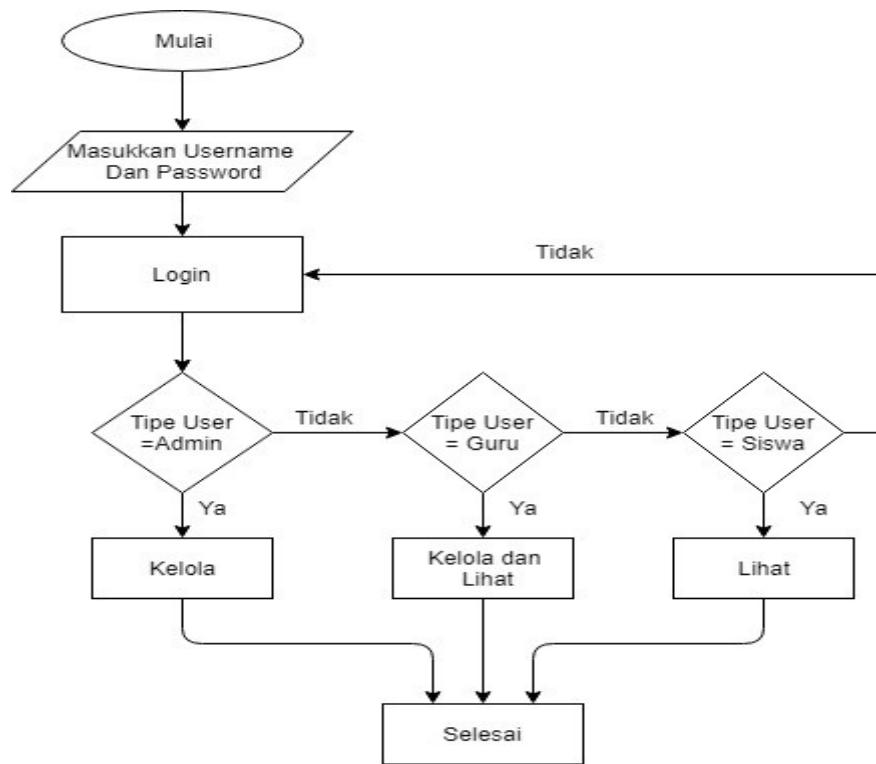
III.2.3 Perancangan Sistem

Dalam melakukan perancangan sistem terdapat tahapan-tahapan yang dilakukan yaitu:

III.2.3.1 Diagram Alir

Diagram alir sistem merupakan diagram yang menggambarkan sistem yang akan dibangun. Adapun diagram alir sistem yang akan dibangun dapat dilihat pada

Gambar 3.2 berikut:



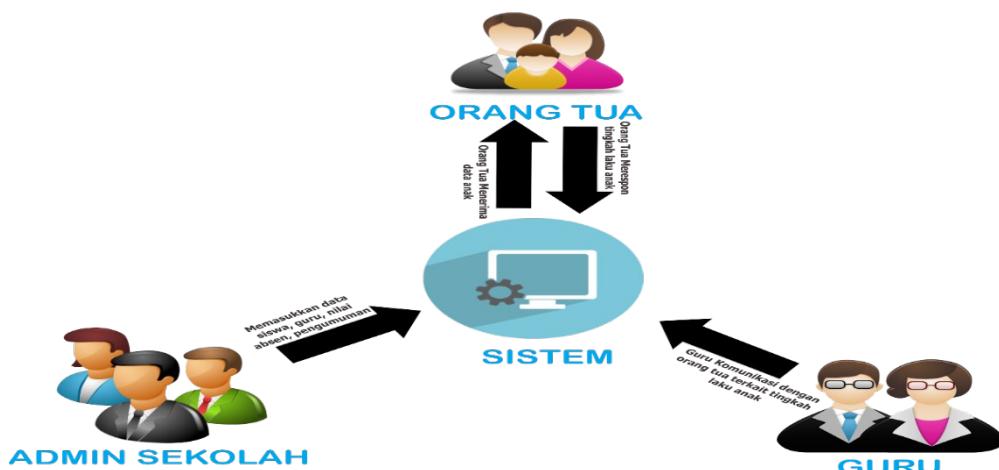
Gambar 3.2 Diagram Alir Sistem

Input data dilakukan oleh admin dari sekolah data yang di input kan adalah data guru, data siswa, data nilai, data absen dan pengumuman yang berkaitan dengan sekolah lalu data tersebut dimasukkan ke dalam database sistem melalui web service. Setelah data di masukkan ke database, data lalu diambil oleh interface pengguna dari database sistem, data yang di ambil dan ditampilkan meliputi data nilai, data absen, data pengumuman.

Pada penerapan aplikasi user menggunakan perangkat mobile yaitu android, user disini adalah orang tua siswa dan guru. Data yang telah ditampilkan lalu di lihat oleh guru dan orang tua yang nantinya data tersebut akan digunakan sebagai proses monitoring dan evaluasi terhadap siswa.

III.2.3.2 Arsitektur Sistem

Secara umum gambaran sistem dapat dibuat rancangan awal arsitektur sistem seperti Gambar 3.3.



Gambar 3.3 Arsitektur Sistem Usulan

Berdasarkan Gambar 3.3 proses yang terjadi adalah: Sistem yang dibangun memiliki dua pengguna yaitu orang tua dan pihak sekolah. Pihak sekolah di sistem ini mempunyai dua akses pengguna yang pertama admin sekolah untuk mengatur atau mengelola data yang akan ditampilkan di halaman website yang nantinya yang akan disimpan pada database Mysql dan kemudian disimpan ke dalam server ketika terhubung ke internet. Selanjutnya pengguna kedua adalah guru (wali kelas) yang bertugas memonitoring dan evaluasi aktivitas anak disekolah.

Pengguna berikutnya adalah orang tua di sistem ini orang tua dapat

mengakses aktivitas anak dan juga turut serta dalam monitoring dan evaluasi aktivitas anak bersama dengan guru (wali kelas).

III.2.3.3 Proses Bisnis Sistem

Proses bisnis merupakan kumpulan aktivitas yang menggambarkan alur bagaimana proses suatu bisnis dijalankan dan menghasilkan keluaran (*output*) yang bernilai bagi pelanggan atau *end-user*.



Gambar 3.4 Proses Bisnis Sistem

III.2.3.4 Perancangan Use Case Diagram

Diagram *use case* merupakan pemodelan untuk kelakuan (*behaviour*) sistem yang akan dibuat. Diagram ini digunakan untuk mendeskripsikan interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa yang sudah ada didalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

a. Definisi Aktor

Berikut adalah deskripsi pendefinisian aktor pada aplikasi rekomendasi harga produk terendah.

Tabel 3.1 Definisi Aktor

No.	Aktor	Deskripsi
1.	User	<i>User</i> merupakan orang yang melihat data yang dimasukkan oleh admin ke sistem, user dalam sistem ini adalah orang tua dan guru (wali kelas).
2.	Admin	Admin merupakan orang yang memasukkan data anak, guru, nilai, absen dan pengumuman. <i>User</i> memegang hak penuh proses kelola data sekolah.

b. Definisi Use Case

Berikut adalah tabel pendefinisian *use case* Andorid yang digunakan oleh orang tua dan guru serta *use case* Web yang digunakan oleh admin pada sistem monitoring aktivitas anak sekolah.

Tabel 3.2 Definisi *Use Case* Orang Tua dan Guru

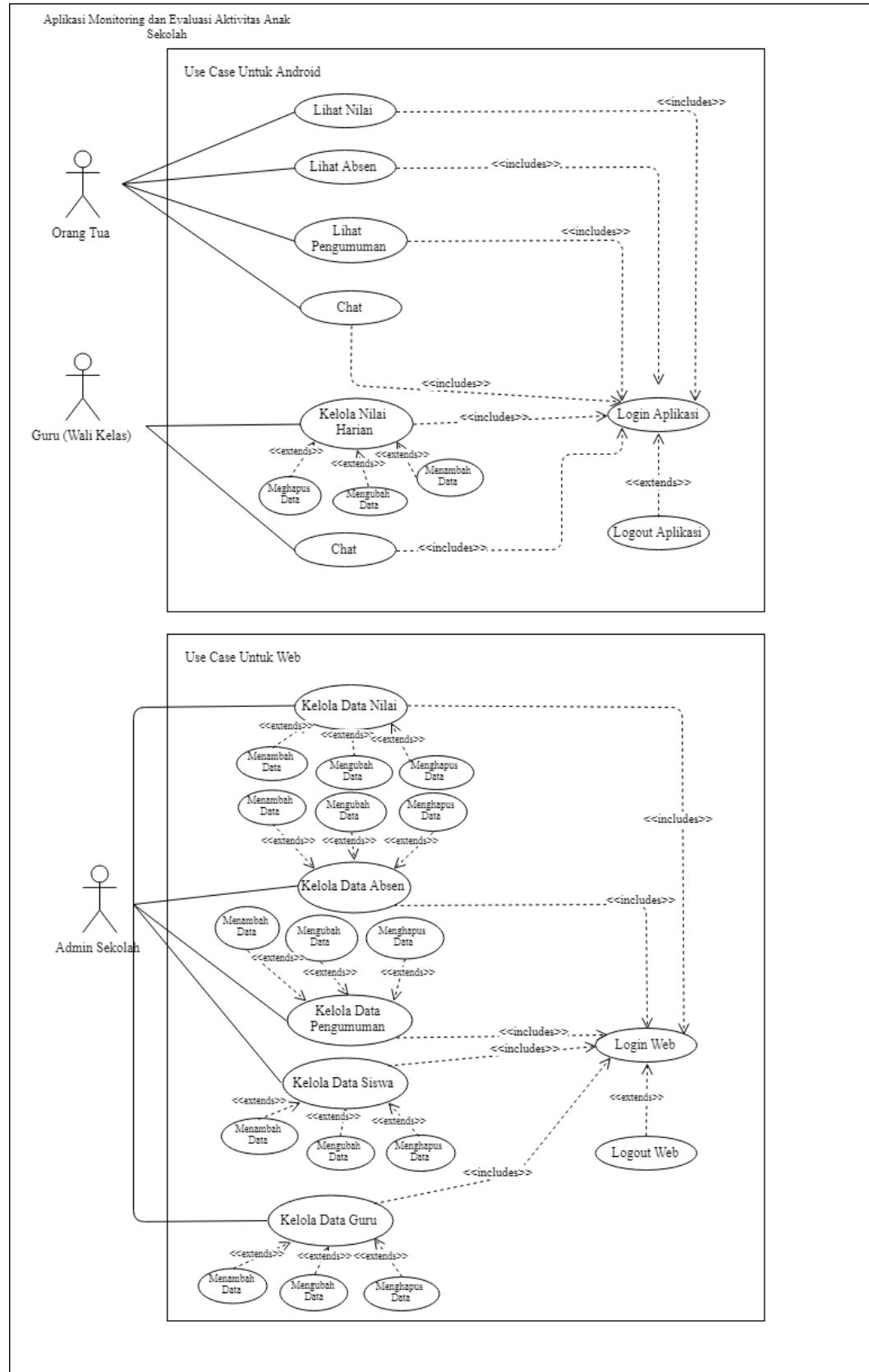
No.	<i>Use case</i>	Deskripsi
UC01	Login aplikasi	Merupakan proses untuk melakukan login guru dan orang tua.
UC02	Logout aplikasi	Merupakan proses untuk melakukan logout guru dan orang tua.
UC03	Lihat nilai	Merupakan proses menampilkan data nilai anak yang diakses orang tua dan guru
UC04	Lihat absen	Merupakan proses menampilkan data absen anak yang diakses orang tua (<i>user</i>).
UC05	Lihat pengumuman	Merupakan proses menampilkan data pengumuman yang berkaitan dengan anak yang diakses orang tua (<i>user</i>).
UC06	Menambah nilai harian	Merupakan proses menambah data nilai harian yang dilakukan oleh guru (wali kelas).
UC07	Mengubah nilai harian	Merupakan proses mengubah data nilai harian yang dilakukan oleh guru (wali kelas).
UC08		Merupakan proses menghapus data nilai harian yang dilakukan oleh guru (wali kelas).
UC09	<i>Chat</i>	Merupakan proses komunikasi antara orang tua dan guru (wali kelas) yang berkaitan dengan aktivitas anak disekolah.

Tabel 3.3 Definisi *Use Case* Admin

No.	<i>Use case</i>	Deskripsi
UC10	Login aplikasi	Merupakan proses untuk melakukan login

		<i>user.</i>
UC11	Logout aplikasi	Merupakan proses untuk melakukan logout <i>user</i> .
UC12	Login web	Merupakan proses untuk melakukan login admin.
UC13	Logout web	Merupakan proses untuk melakukan logout admin.
UC14	Menambah data nilai	Merupakan proses menambah data nilai.
UC15	Mengubah data nilai	Merupakan proses mengubah data nilai.
UC16	Menghapus data nilai	Merupakan proses menghapus data nilai.
UC17	Menambah data absen	Merupakan proses menambah data absen.
UC18	Mengubah data absen	Merupakan proses mengubah data absen.
UC19	Menghapus data absen	Merupakan proses menghapus data absen.
UC20	Menambah data pengumuman	Merupakan proses menambah data pengumuman.
UC21	Mengubah data pengumuman	Merupakan proses mengubah data pengumuman.
UC22	Menghapus data pengumuman	Merupakan proses menghapus data pengumuman.
UC23	Menambah data siswa	Merupakan proses menambah data siswa.
UC24	Mengubah data siswa	Merupakan proses mengubah data siswa.
UC25	Menghapus data siswa	Merupakan proses menghapus data siswa.
UC26	Menambah data guru	Merupakan proses menambah data guru.
UC27	Mengubah data guru	Merupakan proses mengubah data guru.
UC28	Menghapus data guru	Merupakan proses menghapus data guru.

c. Diagram Use Case



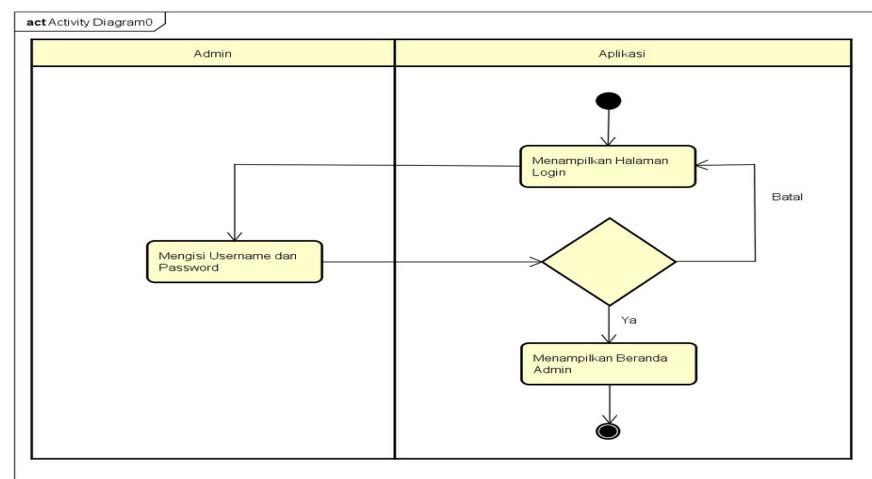
Gambar 3.5 Diagram Use Case Aplikasi

III.2.3.5 Perancangan Activity Diagram

Activity diagram menggambarkan alur kerja pada sistem yang bertujuan untuk melihat alur proses sistem secara bertahap. *Activity diagram* dibuat berdasarkan sebuah atau beberapa *use case* pada *use case diagram*.

a. Activity Diagram Login Admin

Berikut adalah *activity diagram* user dapat dilihat pada Gambar 3.6.

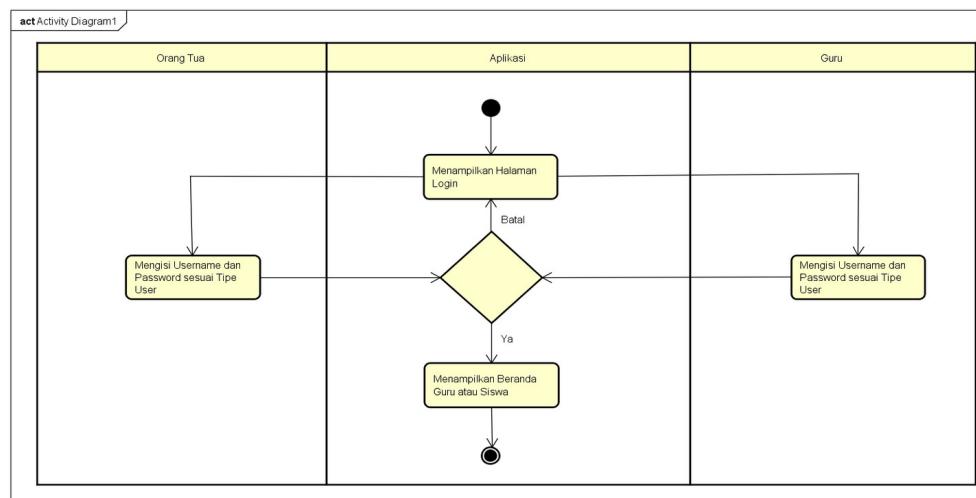


Gambar 3.6 Activity Diagram Login Admin

Pada Gambar 3.6, dapat dilihat bahwa saat login, admin mengisi username dan password jika berhasil tampil halaman beranda dan jika gagal kembali ke halaman login.

b. Activity Diagram Login Orang Tua Dan Guru

Berikut adalah *activity diagram* user dapat dilihat pada Gambar 3.7.

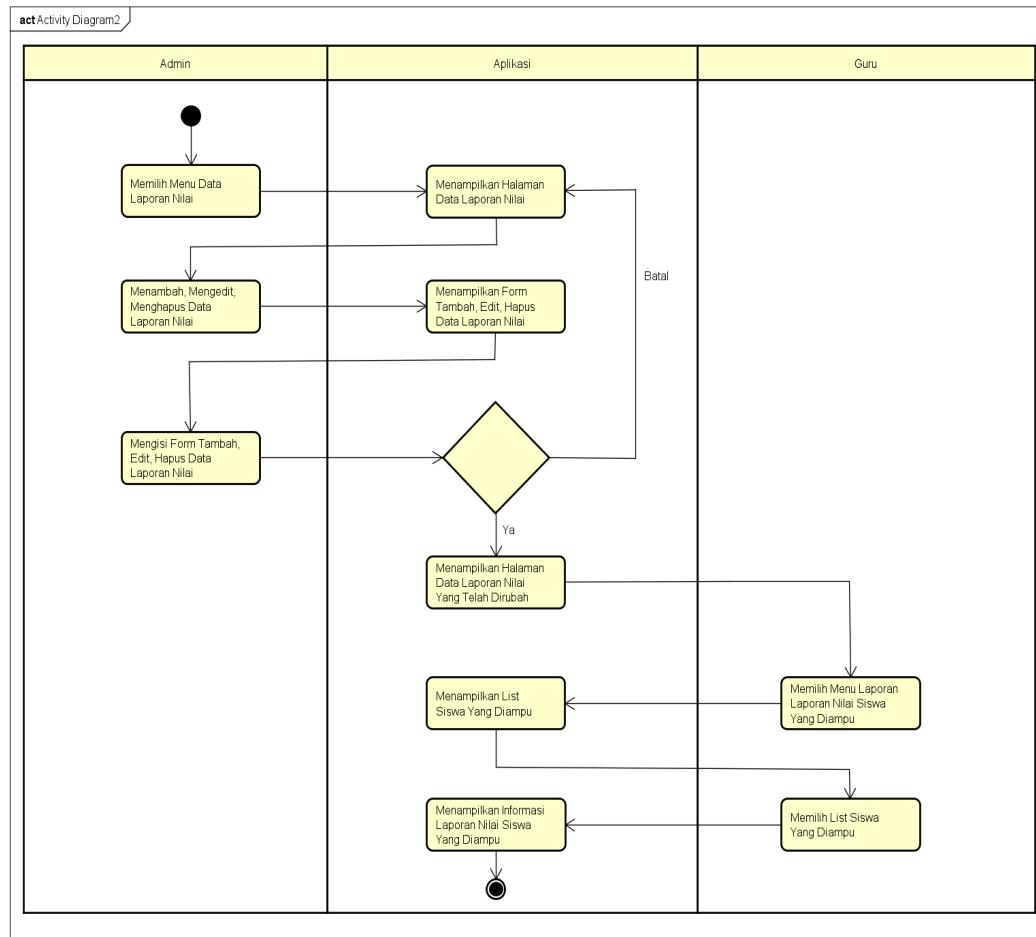


Gambar 3.7 Activity Diagram Login Orang Tua Dan Guru

Pada Gambar 3.7, dapat dilihat bahwa saat login, orang tua dan guru mengisi username dan password jika berhasil tampil halaman beranda siswa atau guru dan jika gagal kembali ke halaman login.

c. Activity Diagram Laporan Nilai Guru

Berikut adalah *activity diagram* laporan nilai guru dapat dilihat pada Gambar 3.8.

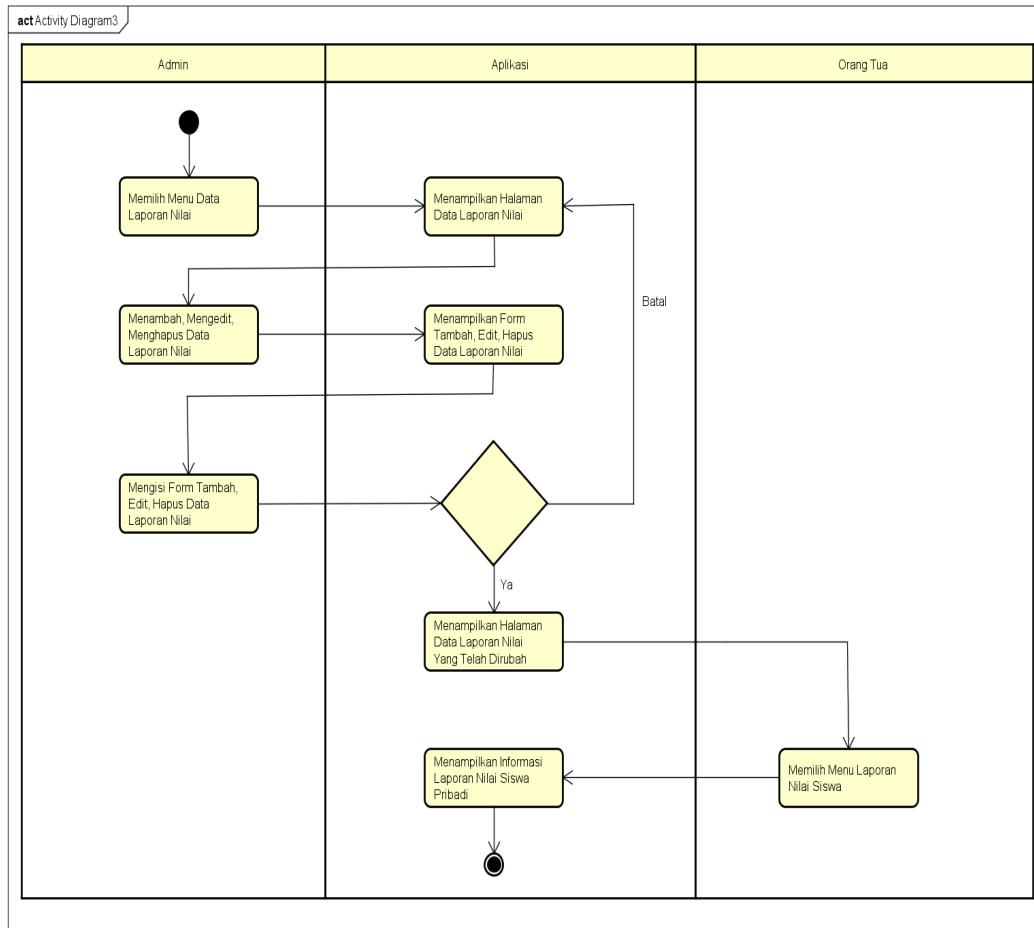


Gambar 3.8 Activity Diagram Laporan Nilai Guru

Pada Gambar 3.8, dapat dilihat bahwa saat admin dapat melakukan tambah, edit, hapus laporan nilai. Setelah itu admin mengisi *form* tambah, edit ataupun hapus laporan nilai siswa dan aplikasi akan menampilkan halaman data laporan nilai siswa yang telah ditambah atau dirubah, jika batal halaman data laporan nilai akan kembali tampil. Data yang dirubah akan tampil di halaman laporan guru, Data yang dapat dilihat seperti list siswa yang diampu, data pribadi laporan nilai per siswa dan absen per siswa.

d. *Activity Diagram* Laporan Nilai Siswa

Berikut adalah *activity diagram* laporan nilai siswa dapat dilihat pada Gambar 3.9.

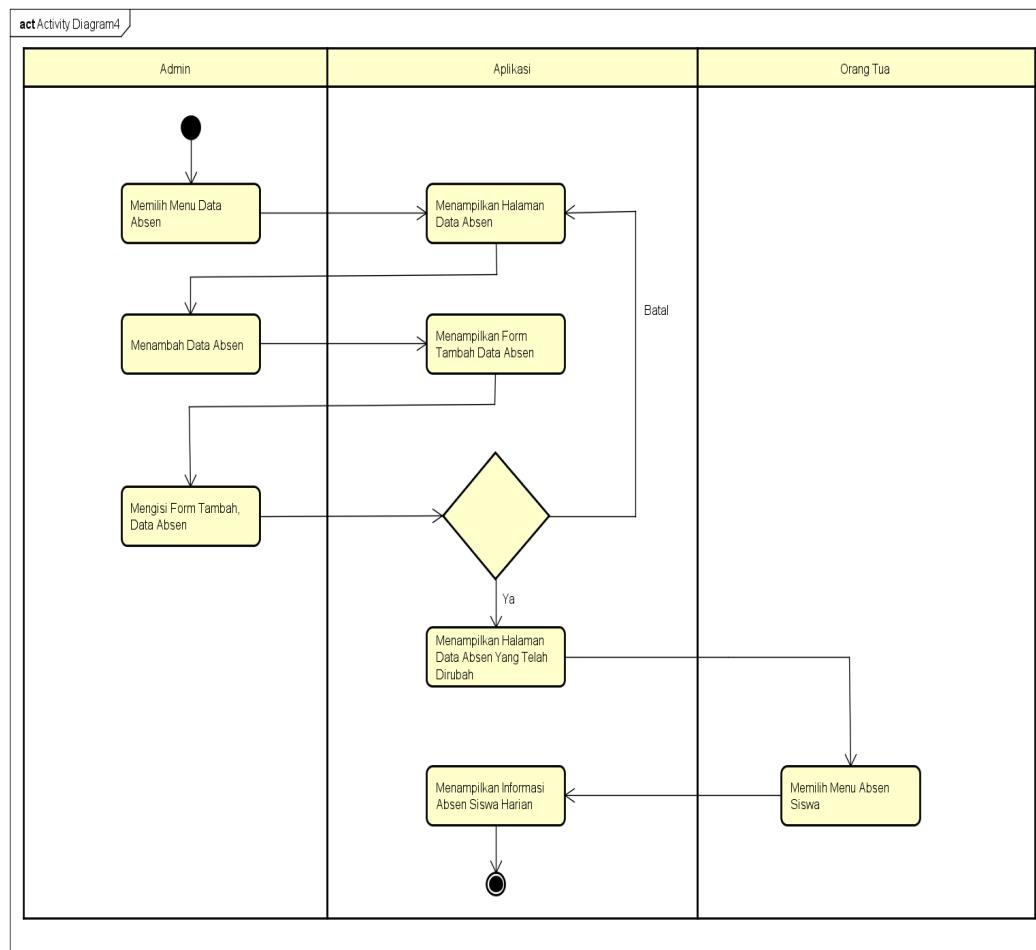


Gambar 3.9 Activity Diagram Laporan Nilai Siswa

Pada Gambar 3.9 dapat dilihat bahwa saat admin dapat melakukan tambah, edit, hapus laporan nilai. Setelah itu admin mengisi *form* tambah, edit ataupun hapus laporan nilai siswa dan aplikasi akan menampilkan halaman data laporan nilai siswa yang telah ditambah atau dirubah, jika batal halaman data laporan nilai akan kembali tampil. Data yang dirubah akan tampil di halaman laporan siswa, Data yang dapat dilihat seperti nilai pribadi siswa, absen pribadi siswa.

e. *Activity Diagram* Data Absen Siswa

Berikut adalah *activity diagram* data absen siswa dapat dilihat pada Gambar 3.10.

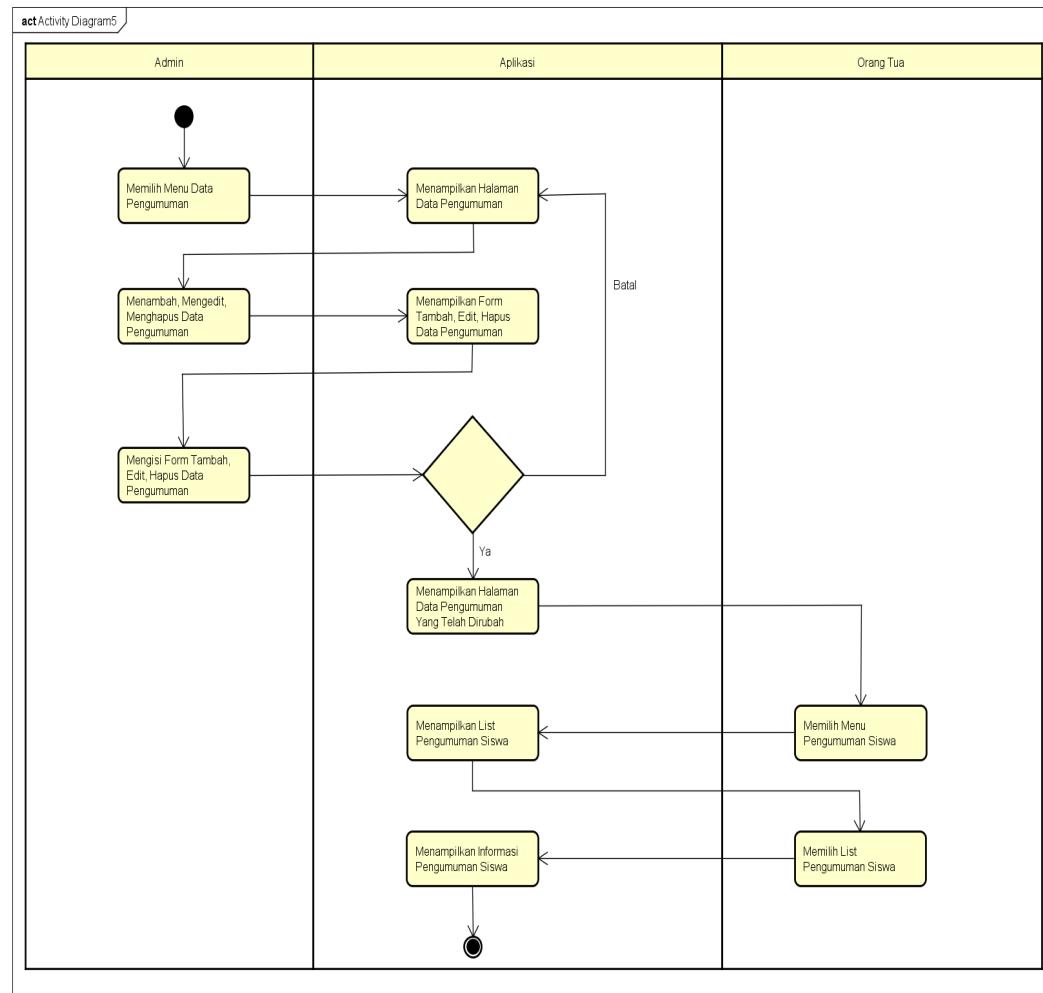


Gambar 3.10 Activity Diagram Data Absen Siswa

Pada Gambar 3.10, dapat dilihat bahwa saat admin dapat melakukan tambah, absen per hari. Setelah itu admin mengisi *form* tambah absen siswa perhari dan aplikasi akan menampilkan halaman data absen perhari siswa yang telah ditambah atau dirubah, jika batal halaman data absen perhari siswa akan kembali tampil. Data yang dirubah akan tampil di halaman absen perhari siswa, Data yang dapat dilihat seperti nilai list absen siswa perhari dan absen hari ini.

f. *Activity Diagram Data Pengumuman*

Berikut adalah *activity diagram* pengumuman dapat dilihat pada Gambar 3.11.



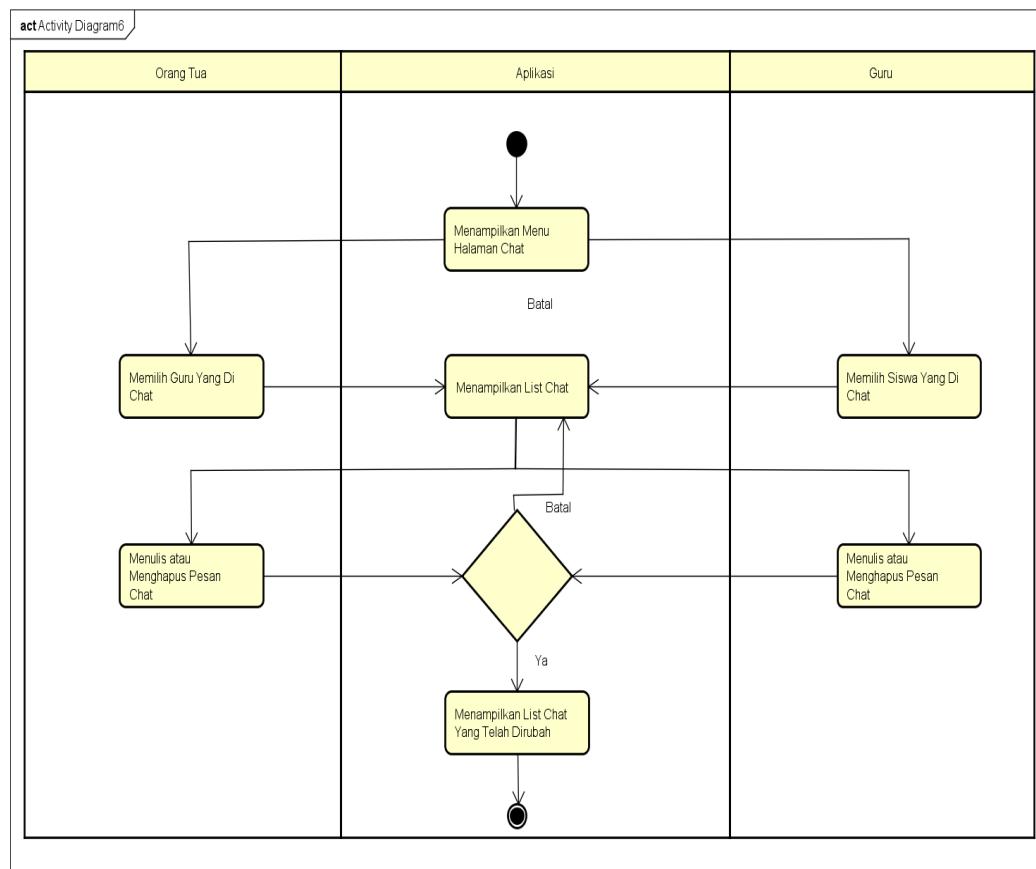
Gambar 3.11 *Activity Diagram Data Pengumuman*

Pada Gambar 3.11, dapat dilihat bahwa admin dapat melakukan tambah, edit, hapus data pengumuman untuk menambah, mengedit, dan menghapus data pengumuman, dengan cara memilih menu pengumuman. Aplikasi menampilkan *form* tambah, edit, dan hapus pengumuman. Setelah itu admin mengisi *form* tambah, edit ataupun hapus pengumuman dan aplikasi akan menampilkan halaman data pengumuman yang telah ditambah atau dirubah. Data yang telah

dirubah akan tampil di halaman menu pengumuman siswa. Data yang tampil seperti list pengumuman dan detail dari pengumuman.

g. Activity Diagram Chat

Berikut adalah *activity diagram* chat dapat dilihat pada Gambar 3.12.

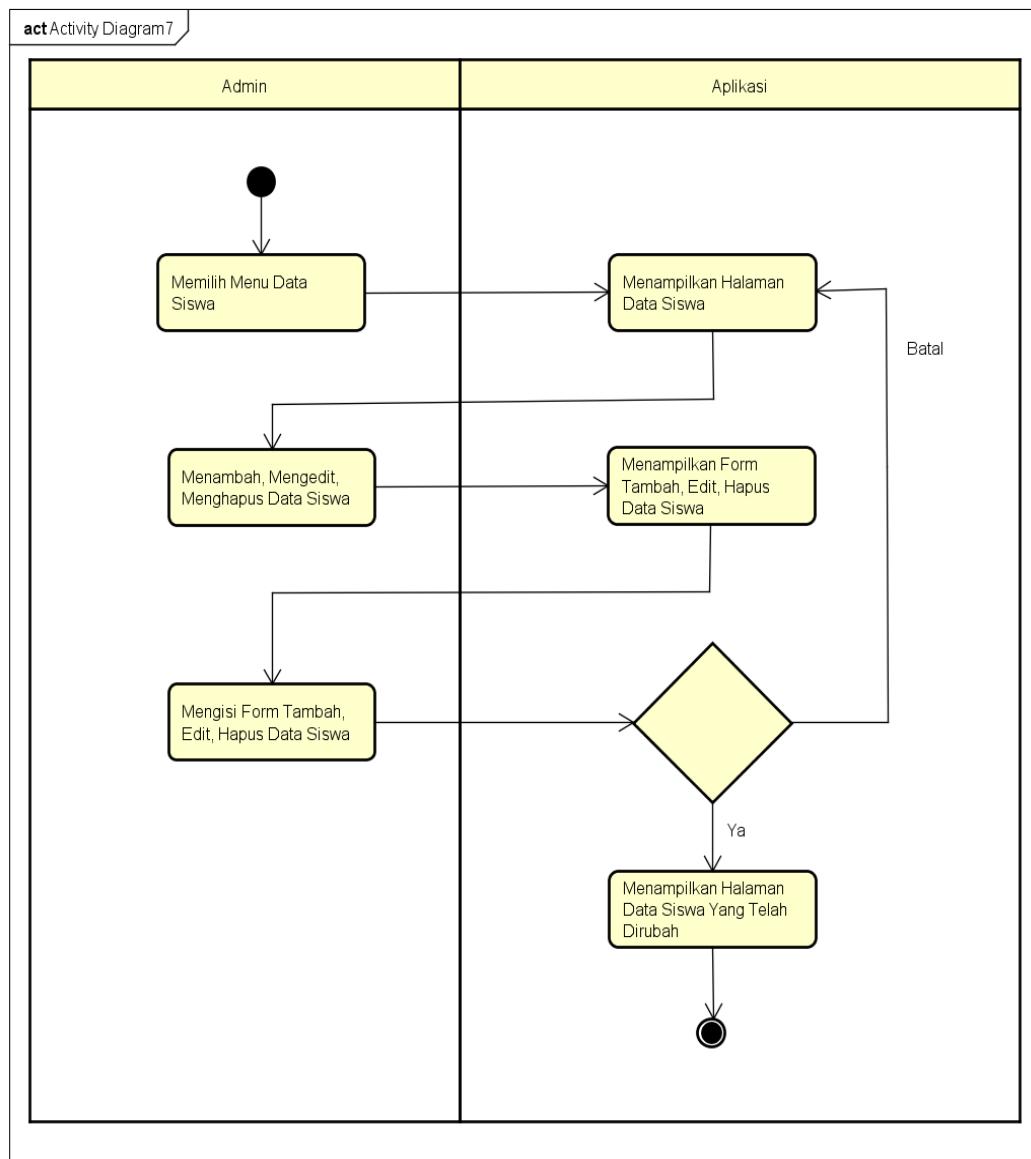


Gambar 3.12 *Activity Diagram Chat*

Pada Gambar 3.12, dapat dilihat bahwa orang tua dan guru dapat berkomunikasi dengan menggunakan fitur chat, pada saat menu halaman chat tampil guru atau orang tua memilih guru atau orang tua yang akan di chat, lalu orang tua dan guru dapat menulis atau menghapus pesan chat, jika berhasil list chat yang ditambah atau dirubah akan tampil.

h. Activity Diagram Data Siswa

Berikut adalah *activity diagram* data siswa dapat dilihat pada Gambar 3.13.



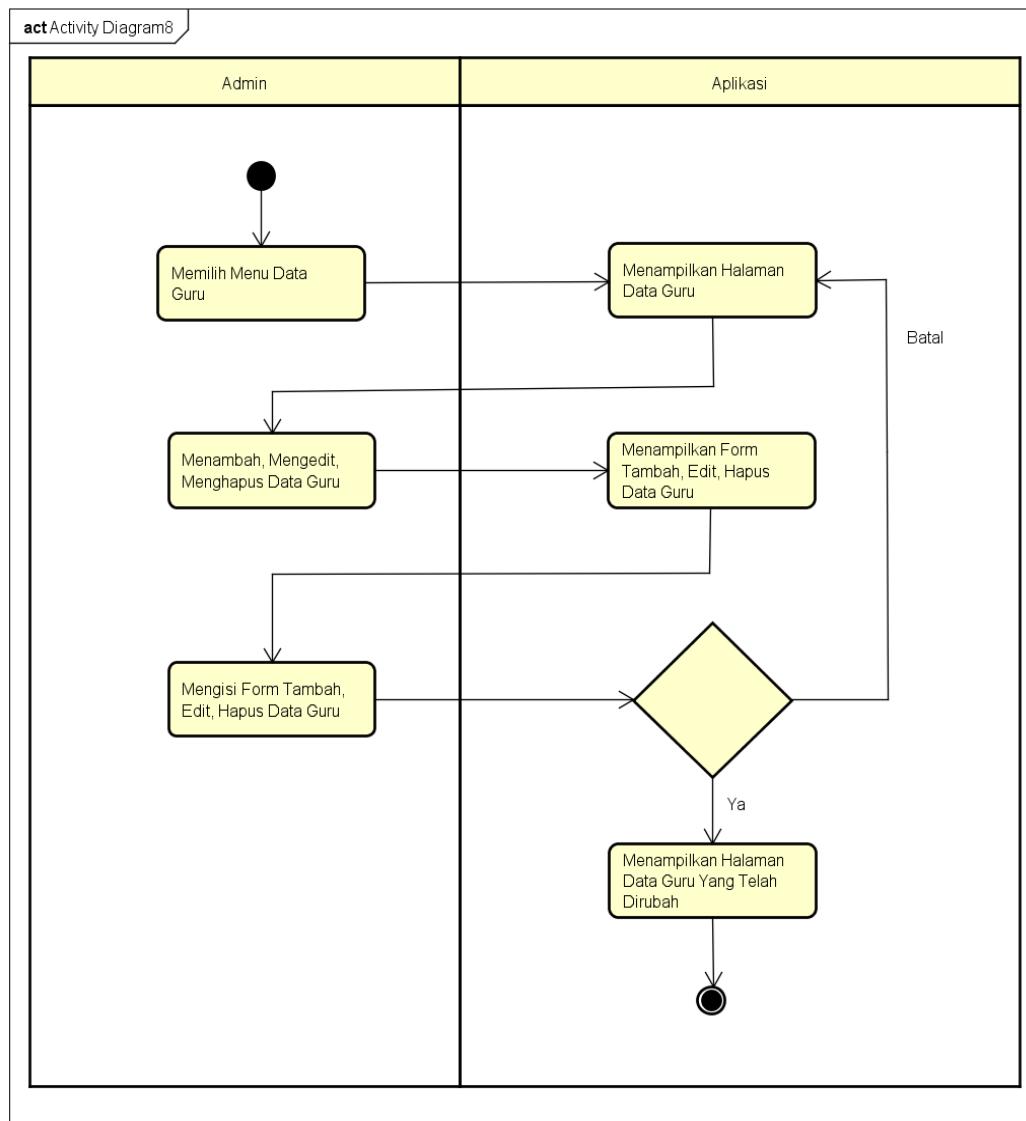
Gambar 3.13 Activity Diagram Data Siswa

Pada Gambar 3.13, dapat dilihat bahwa admin dapat melakukan tambah, edit, hapus data siswa untuk menambah, mengedit, dan menghapus data siswa, dengan cara memilih menu data siswa. Aplikasi menampilkan *form* tambah, edit, dan hapus data siswa. Setelah itu admin mengisi *form* tambah, edit ataupun hapus

data siswa dan aplikasi akan menampilkan halaman data siswa yang telah ditambah atau dirubah, jika batal halaman data siswa akan kembali tampil.

i. *Activity Diagram* Data Guru

Berikut adalah *activity diagram* data guru dapat dilihat pada Gambar 3.14.



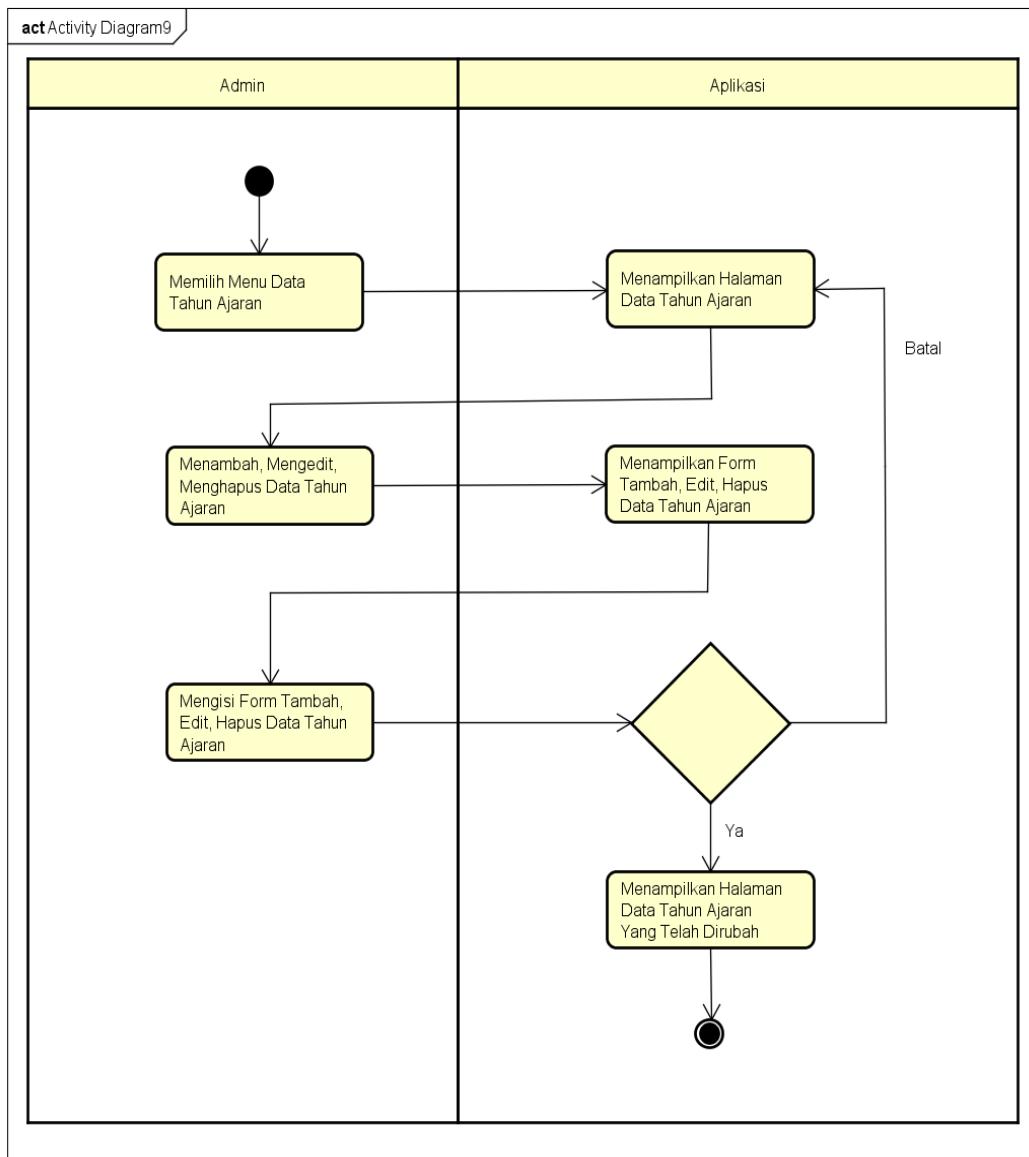
Gambar 3.14 *Activity Diagram* Data Guru

Pada Gambar 3.14, dapat dilihat bahwa admin dapat melakukan tambah, edit, hapus data guru untuk menambah, mengedit, dan menghapus data guru, dengan cara memilih menu data guru. Aplikasi menampilkan *form* tambah, edit, dan hapus data guru. Setelah itu admin mengisi *form* tambah, edit ataupun hapus

data guru dan aplikasi akan menampilkan halaman data guru yang telah ditambah atau dirubah, jika batal halaman data guru akan kembali tampil.

j. *Activity Diagram Data Tahun Ajaran*

Berikut adalah *activity diagram* data tahun ajaran dapat dilihat pada Gambar 3.15.



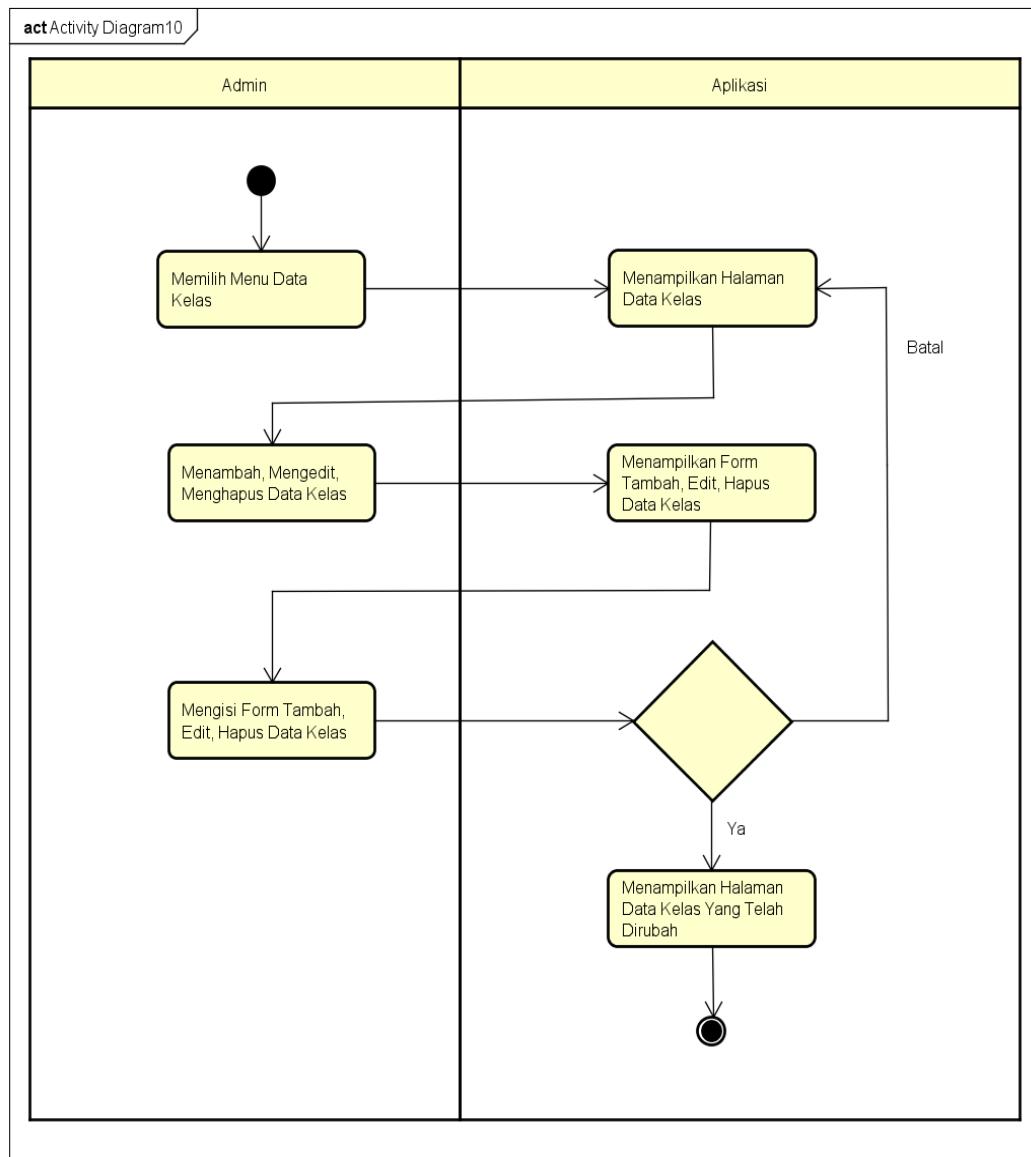
Gambar 3.15 *Activity Diagram Data Tahun Ajaran*

Pada Gambar 3.15, dapat dilihat bahwa admin dapat melakukan tambah, edit, hapus tahun ajaran untuk menambah, mengedit, dan menghapus tahun ajaran,

dengan cara memilih menu tahun ajaran. Aplikasi menampilkan *form* tambah, edit, dan hapus tahun ajaran. Setelah itu admin mengisi *form* tambah, edit ataupun hapus tahun ajaran dan aplikasi akan menampilkan halaman tahun ajaran yang telah ditambah atau dirubah, jika batal halaman tahun ajaran akan kembali tampil.

k. Activity Diagram Data Kelas

Berikut adalah *activity diagram* data kelas dapat dilihat pada Gambar 3.16.



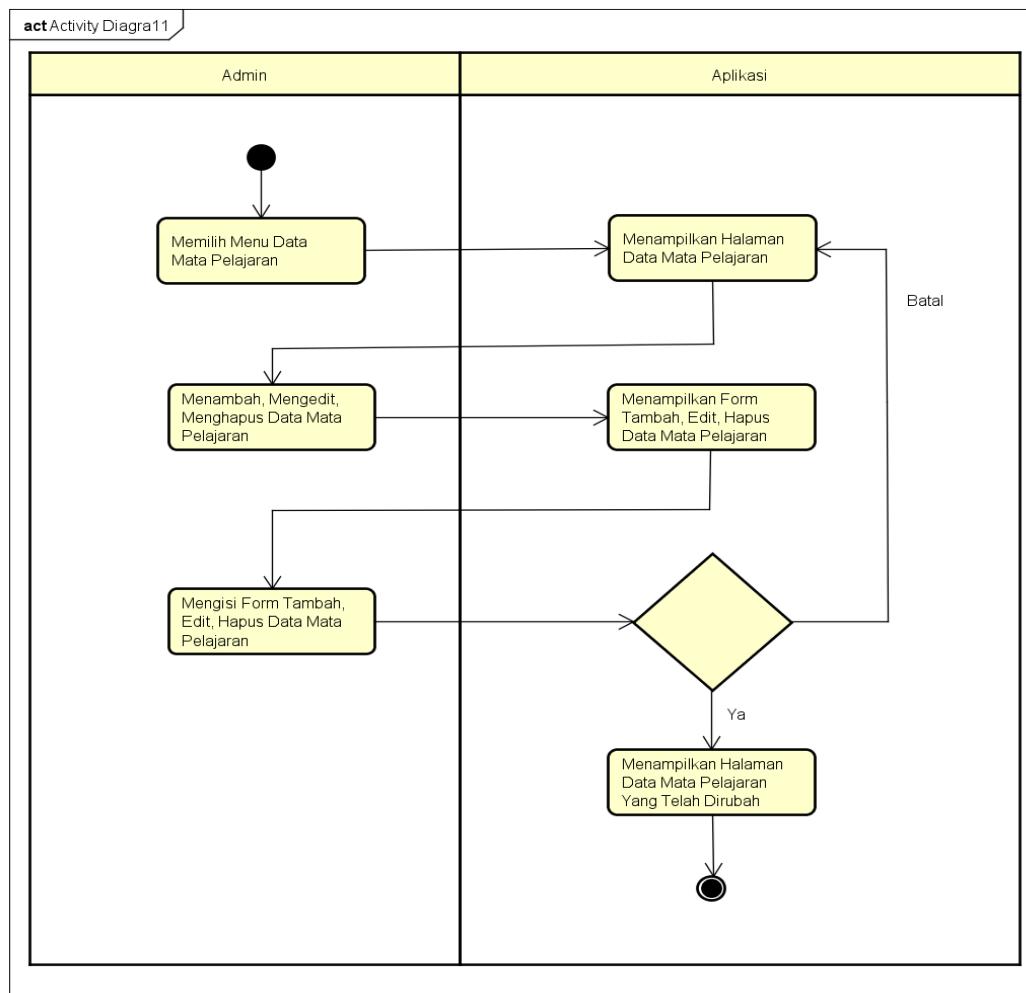
Gambar 3.16 Activity Diagram Data Kelas

Pada Gambar 3.16, dapat dilihat bahwa admin dapat melakukan tambah, edit, hapus kelas untuk menambah, mengedit, dan menghapus kelas, dengan cara

memilih menu kelas. Kemudian aplikasi menampilkan halaman kelas agar dapat menambah, mengedit, dan menghapus kelas dan aplikasi menampilkan *form* tambah, edit, dan hapus kelas. Setelah itu admin mengisi *form* tambah, edit ataupun hapus kelas dan aplikasi akan menampilkan halaman kelas yang telah ditambah atau dirubah, jika batal halaman kelas akan kembali tampil.

1. *Activity Diagram* Data Mata Pelajaran

Berikut adalah *activity diagram* data mata pelajaran dapat dilihat pada Gambar 3.17.



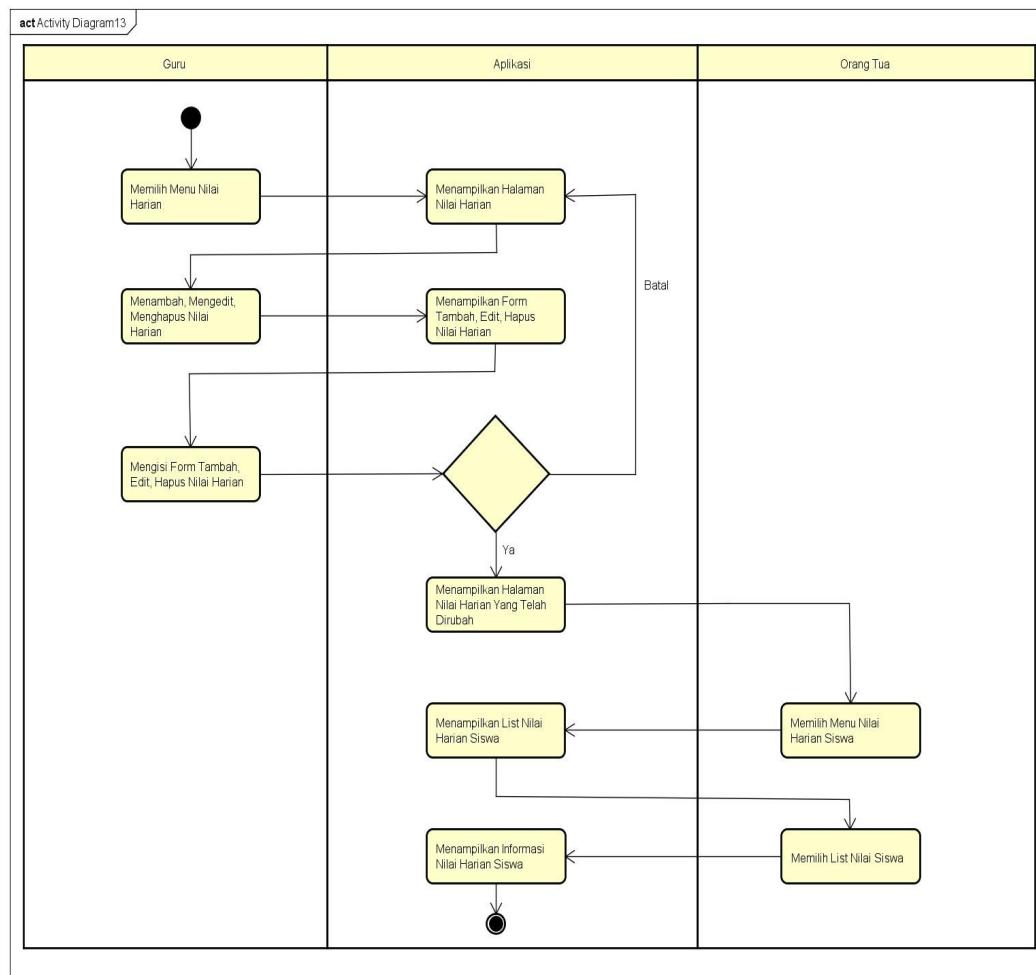
Gambar 3.17 *Activity Diagram* Data Mata Pelajaran

Pada Gambar 3.17, dapat dilihat bahwa admin dapat melakukan tambah, edit, hapus mata pelajaran untuk menambah, mengedit, dan menghapus mata pelajaran, dengan cara memilih menu mata pelajaran. Kemudian aplikasi menampilkan halaman mata pelajaran agar dapat menambah, mengedit, dan

menghapus mata pelajaran dan aplikasi menampilkan *form* tambah, edit, dan hapus mata pelajaran. Setelah itu admin mengisi *form* tambah, edit ataupun hapus mata pelajaran dan aplikasi akan menampilkan halaman mata pelajaran yang telah ditambah atau dirubah, jika batal halaman mata pelajaran akan kembali tampil.

m. *Activity Diagram* Data Nilai Harian

Berikut adalah *activity diagram* data mata pelajaran dapat dilihat pada Gambar 3.18.



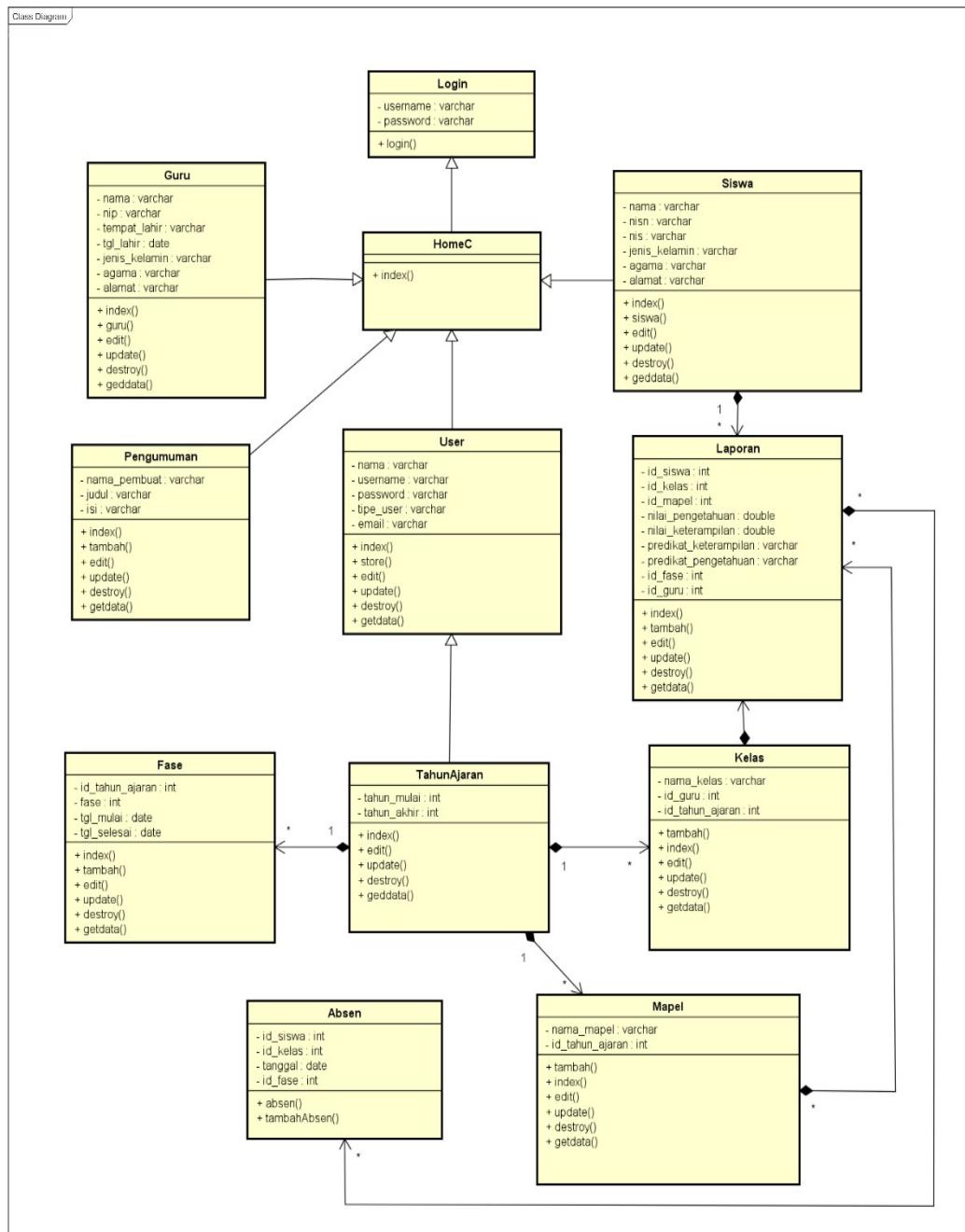
Gambar 3.18 *Activity Diagram* Data Nilai Harian

Pada Gambar 3.18, dapat dilihat bahwa guru dapat melakukan tambah, edit, hapus nilai harian untuk menambah, mengedit, dan menghapus nilai harian, dengan cara memilih menu nilai harian. Kemudian aplikasi menampilkan halaman nilai harian agar dapat menambah, mengedit, dan menghapus nilai harian dan aplikasi menampilkan *form* tambah, edit, dan hapus nilai harian. Setelah itu guru

mengisi *form* tambah, edit ataupun hapus nilai harian dan aplikasi akan menampilkan halaman nilai harian yang telah ditambah atau dirubah, jika batal halaman nilai harian akan kembali tampil.

III.2.3.6 Perancangan Class Diagram

Class diagram merupakan salah satu diagram utama dari UML untuk menggambarkan class atau *blueprint object* pada sebuah sistem. serta memiliki atribut dan operasi yang terdapat dalam sistem. Analis pembentukan class diagram, merupakan aktivitas inti yang sangat mempengaruhi arsitektur piranti lunak yang dirancang hingga ke tahap pengkodean.



Gambar 3.19 *Class Diagram* Aplikasi

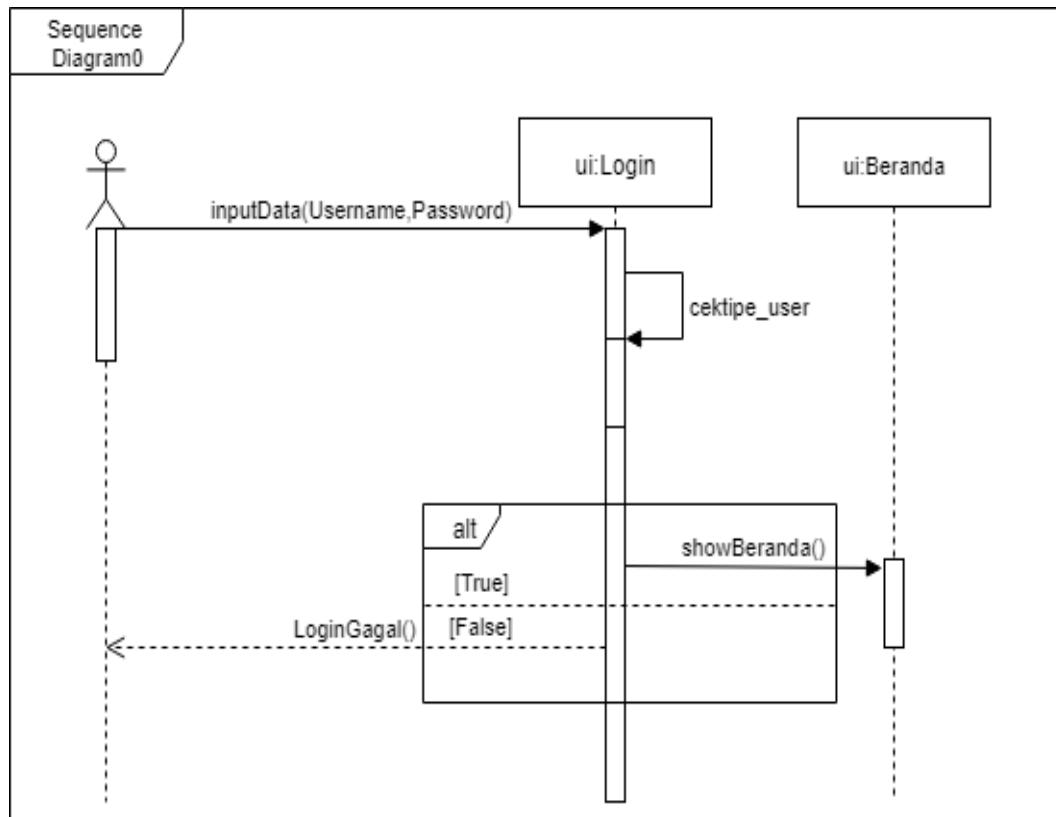
III.2.3.7 Perancangan Sequence Diagram

Sequence diagram adalah diagram yang menjelaskan interaksi objek berdasarkan urutan waktu. *Sequence diagram* menggambarkan urutan atau tahapan yang harus dilakukan pengguna. Diagram ini menunjukkan sejumlah contoh objek dan pesan yang melakukan satu tugas atau aksi tertentu. Komponen utama *sequence diagram* terdiri atas objek yang dituliskan dengan kotak persegi. Pesan diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan

progress vertikal.

a. *Sequence Diagram Login User*

Berikut adalah *sequence diagram* input data dapat dilihat pada Gambar 3.20.

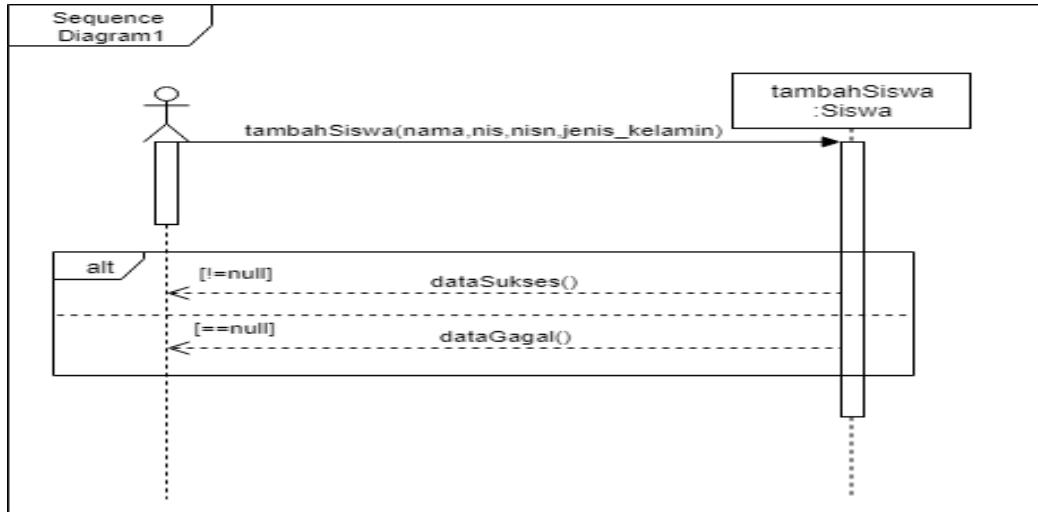


Gambar 3.20 Sequence Diagram Login User

Pada Gambar 3.20, dapat dilihat bahwa user *login* menggunakan fungsi *ceklogin()* dan *cektipeuser()* dengan mengambil data user agar aplikasi dapat menampilkan menu beranda user.

b. *Sequence Diagram Tambah Data Siswa*

Berikut adalah *sequence diagram* kelola data siswa dapat dilihat pada Gambar 3.21.

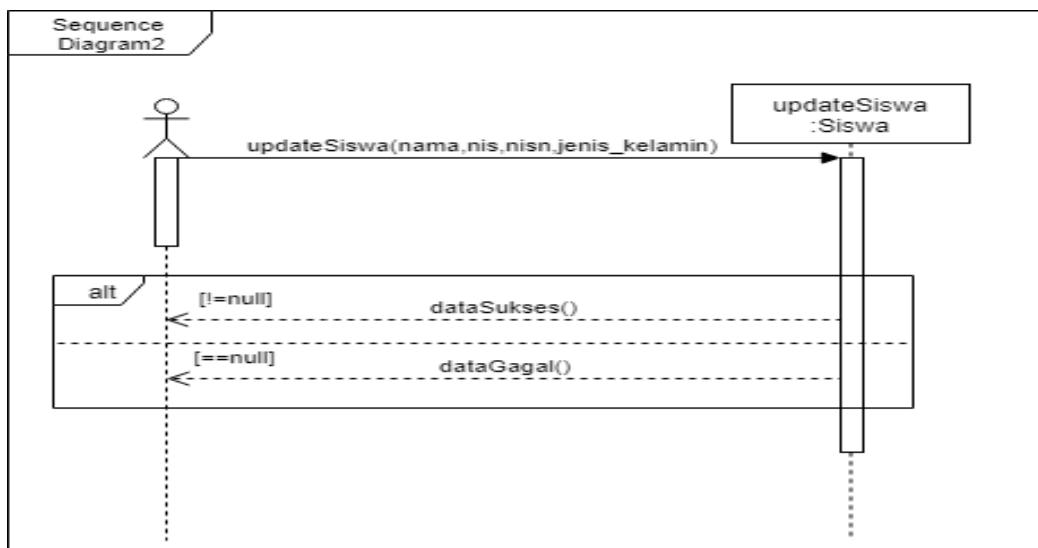


Gambar 3.21 Sequence Diagram Tambah Data Siswa

Pada Gambar 3.21, dapat dilihat bahwa admin *input* data menggunakan fungsi tambahSiswa() dengan menambah data siswa jika sukses maka data siswa akan bertambah.

c. Sequence Diagram Update Data Siswa

Berikut adalah *sequence diagram* kelola data siswa dapat dilihat pada Gambar 3.22.

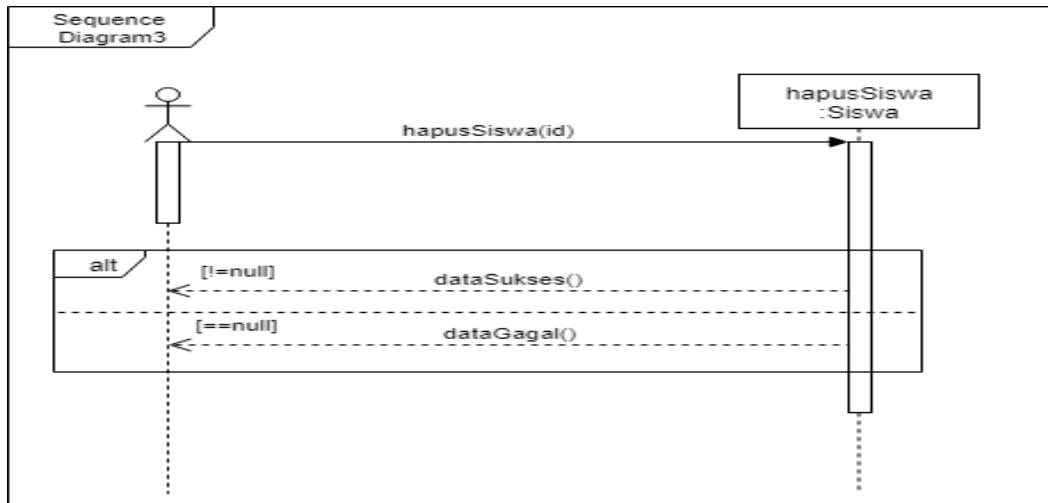


Gambar 3.22 Sequence Diagram Update Data Siswa

Pada Gambar 3.22, dapat dilihat bahwa admin *input* data menggunakan fungsi updateSiswa() dengan update data siswa jika sukses maka data siswa akan terupdate.

d. *Sequence Diagram* Hapus Data Siswa

Berikut adalah *sequence diagram* kelola data siswa dapat dilihat pada Gambar 3.23.

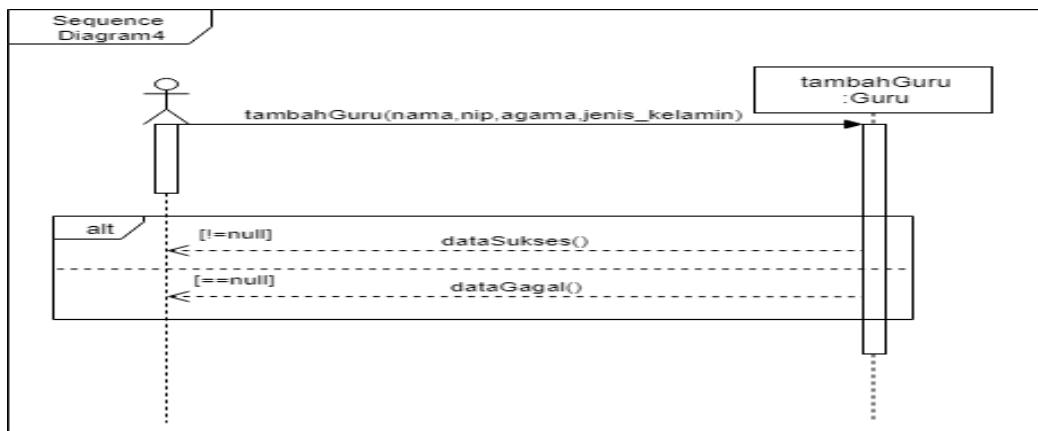


Gambar 3.23 Sequence Diagram Hapus Data Siswa

Pada Gambar 3.23, dapat dilihat bahwa admin *input* data menggunakan fungsi *hapusSiswa()* dengan menghapus data siswa jika sukses maka data siswa akan terhapus.

e. *Sequence Diagram* Tambah Data Guru

Berikut adalah *sequence diagram* kelola data Guru dapat dilihat pada Gambar 3.24.

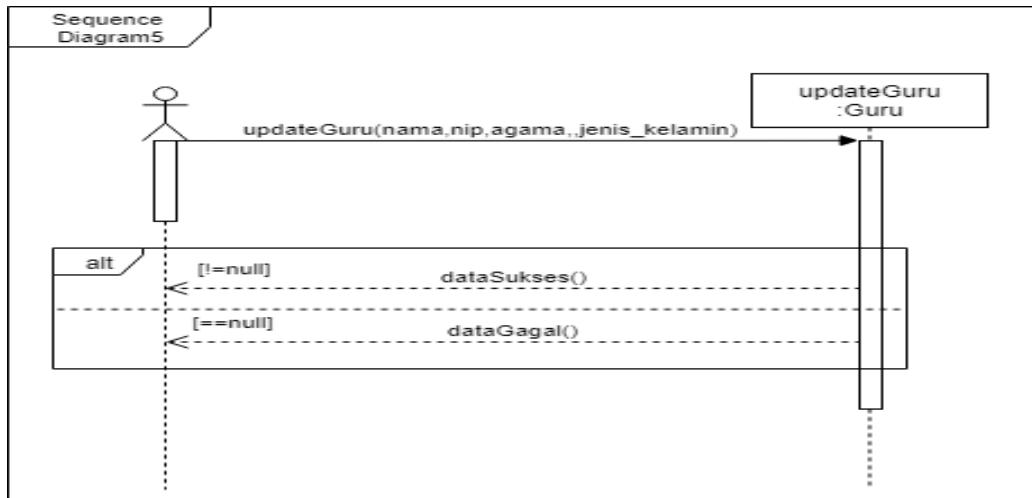


Gambar 3.24 Sequence Diagram Tambah Data Guru

Pada Gambar 3.24, dapat dilihat bahwa admin *input* data menggunakan fungsi *tambahGuru()* dengan menambah data guru jika sukses maka data guru akan bertambah.

f. *Sequence Diagram* Update Data Guru

Berikut adalah *sequence diagram* kelola data guru dapat dilihat pada Gambar 3.25.

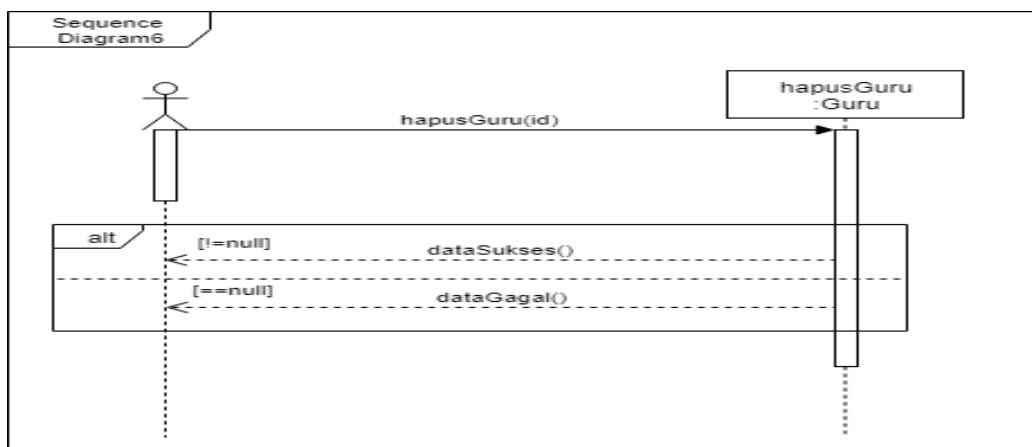


Gambar 3.25 Sequence Diagram Update Data Guru

Pada Gambar 3.25, dapat dilihat bahwa admin *input* data menggunakan fungsi `updateGuru()` dengan update data guru jika sukses maka data guru akan terupdate.

g. *Sequence Diagram* Hapus Data Guru

Berikut adalah *sequence diagram* kelola data guru dapat dilihat pada Gambar 3.26.

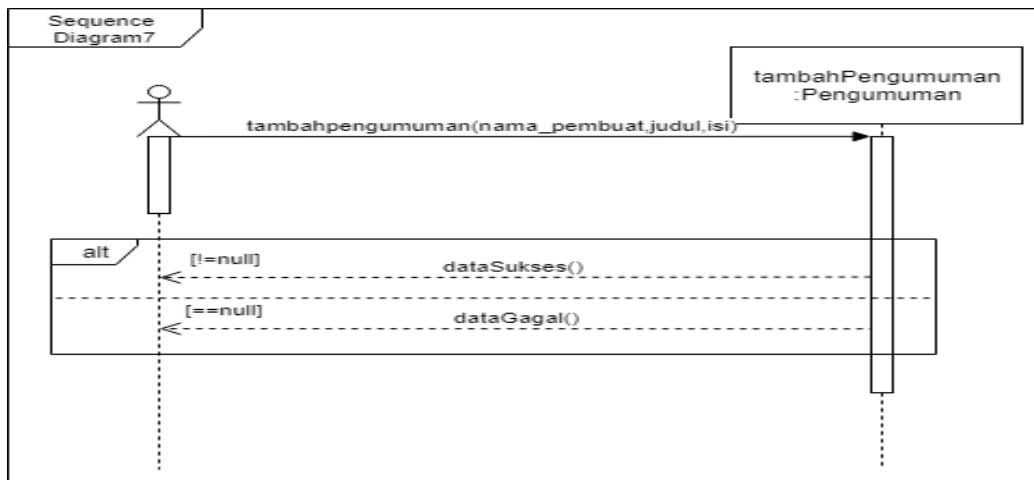


Gambar 3.26 Sequence Diagram Hapus Data Guru

Pada Gambar 3.26, dapat dilihat bahwa admin *input* data menggunakan fungsi `hapusGuru()` dengan hapus data guru jika sukses maka data guru akan terhapus.

h. Sequence Diagram Tambah Pengumuman

Berikut adalah *sequence diagram* kelola data pengumuman dapat dilihat pada Gambar 3.27.

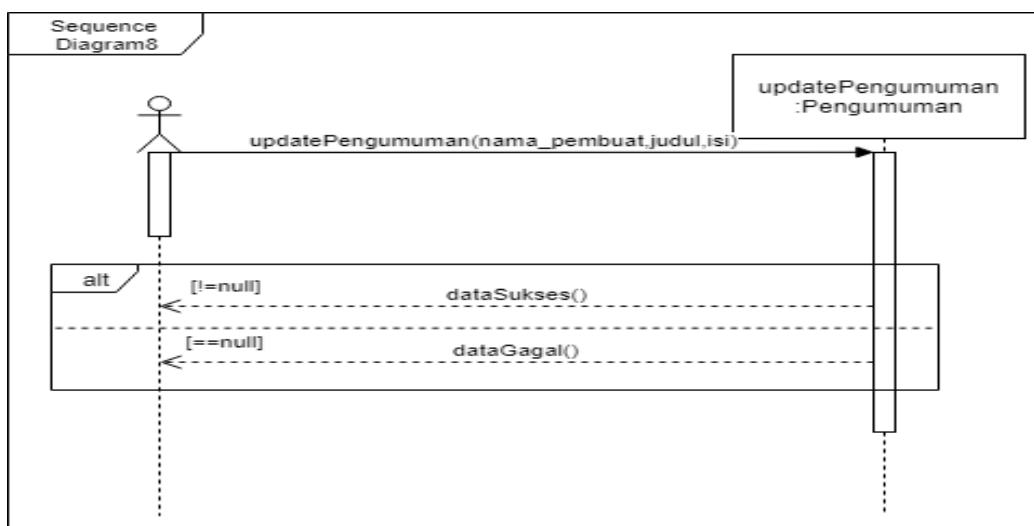


Gambar 3.27 Sequence Diagram Tambah Data Guru

Pada Gambar 3.27, dapat dilihat bahwa admin *input* data menggunakan fungsi `tambahPengumuman()` dengan menambah data pengumuman jika sukses maka data pengumuman akan bertambah.

i. Sequence Diagram Update Pengumuman

Berikut adalah *sequence diagram* kelola data pengumuman dapat dilihat pada Gambar 3.28.

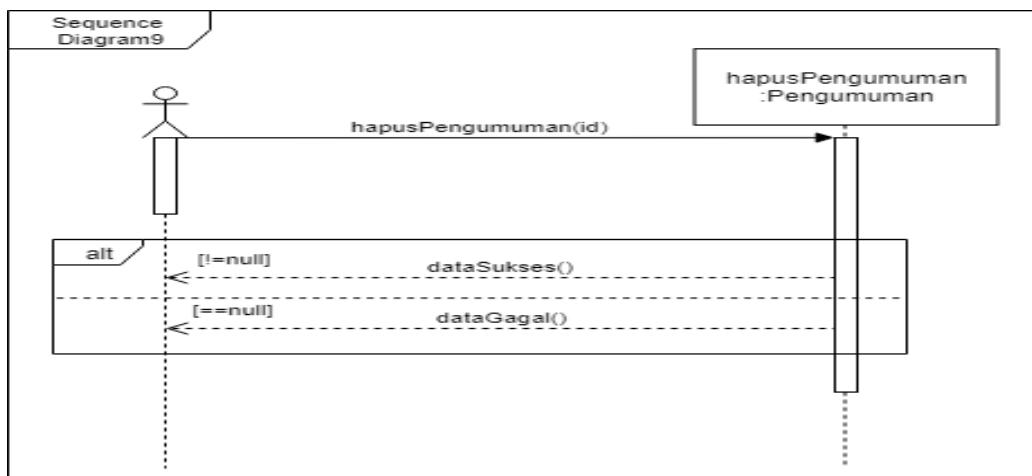


Gambar 3.28 Sequence Diagram Update Data Pengumuman

Pada Gambar 3.28, dapat dilihat bahwa admin *input* data menggunakan fungsi updatePengumuman() dengan update data pengumuman jika sukses maka data pengumuman akan terupdate.

j. *Sequence Diagram* Hapus Pengumuman

Berikut adalah *sequence diagram* kelola data pengumuman dapat dilihat pada Gambar 3.29.

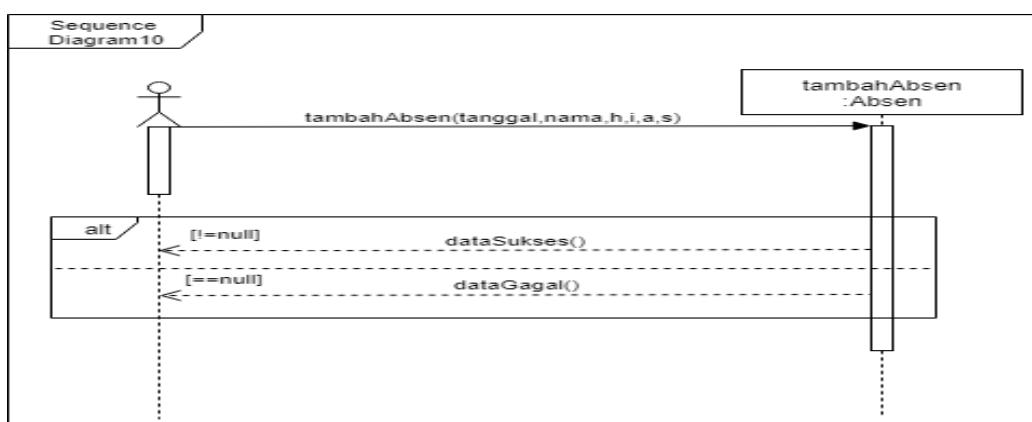


Gambar 3.29 Sequence Diagram Hapus Data Pengumuman

Pada Gambar 3.29, dapat dilihat bahwa admin *input* data menggunakan fungsi hapusPengumuman() dengan hapus data pengumuman jika sukses maka data pengumuman akan terhapus.

k. *Sequence Diagram* Tambah Absen

Berikut adalah *sequence diagram* kelola data absen dapat dilihat pada Gambar 3.30.

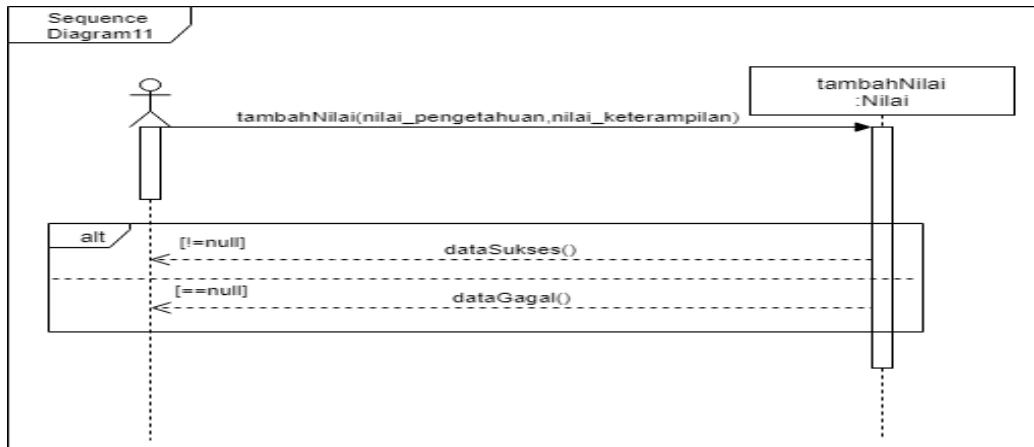


Gambar 3.30 Sequence Diagram Tambah Data Absen

Pada Gambar 3.30, dapat dilihat bahwa admin *input* data menggunakan fungsi tambahAbsen() dengan menambah data absen jika sukses maka data absen akan bertambah.

l. Sequence Diagram Tambah Nilai

Berikut adalah *sequence diagram* kelola data nilai dapat dilihat pada Gambar 3.31.

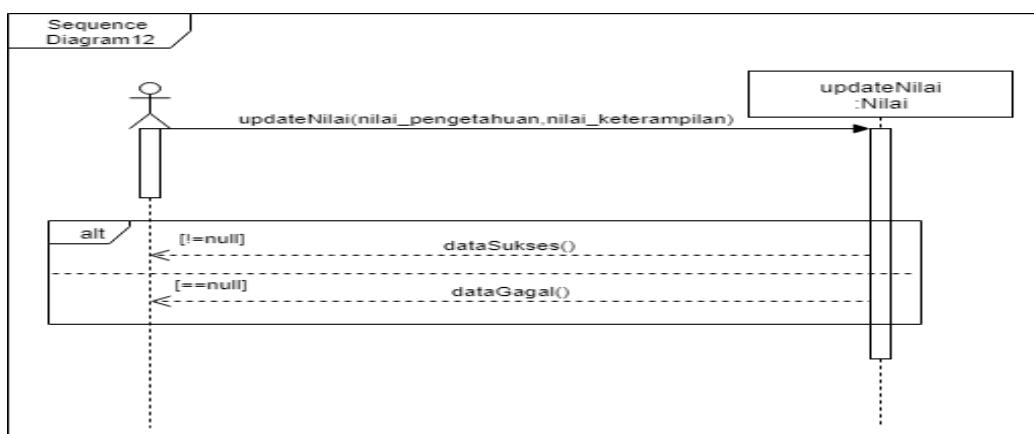


Gambar 3.31 Sequence Diagram Tambah Data Nilai

Pada Gambar 3.31, dapat dilihat bahwa admin *input* data menggunakan fungsi tambahNilai() dengan menambah data nilai jika sukses maka data nilai akan bertambah.

m. Sequence Diagram Update Nilai

Berikut adalah *sequence diagram* kelola data nilai dapat dilihat pada Gambar 3.32.

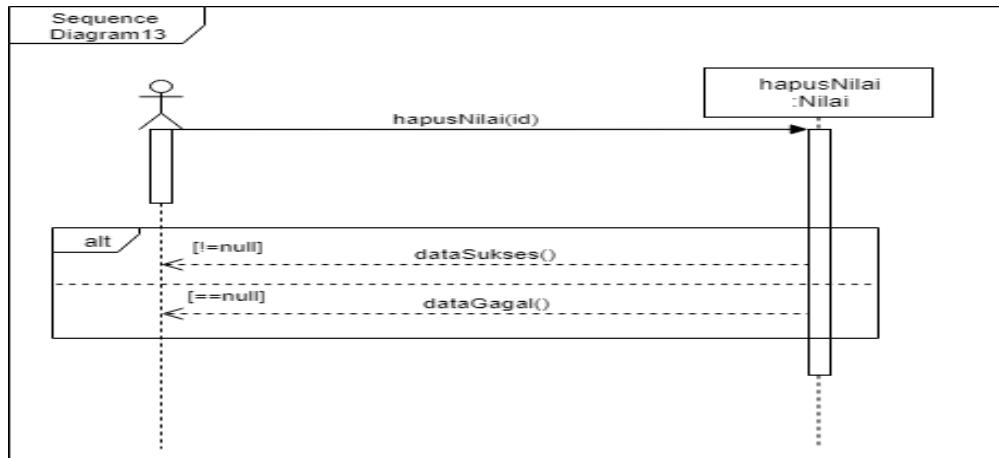


Gambar 3.32 Sequence Diagram Update Data Nilai

Pada Gambar 3.32, dapat dilihat bahwa admin *input* data menggunakan fungsi updateNilai() dengan update data nilai jika sukses maka data nilai akan terupdate.

n. *Sequence Diagram* Hapus Nilai

Berikut adalah *sequence diagram* kelola data nilai dapat dilihat pada Gambar 3.33.

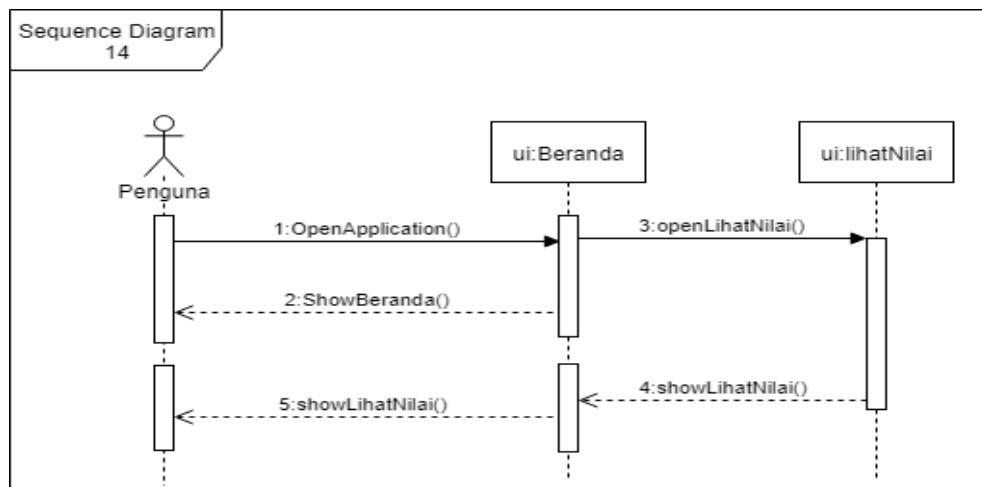


Gambar 3.33 Sequence Diagram Hapus Data Nilai

Pada Gambar 3.33, dapat dilihat bahwa admin *input* data menggunakan fungsi hapusNilai() dengan hapus data nilai jika sukses maka data nilai akan terhapus.

o. *Sequence Diagram* Lihat Nilai

Berikut adalah *sequence diagram* lihat data nilai dapat dilihat pada Gambar 3.34.

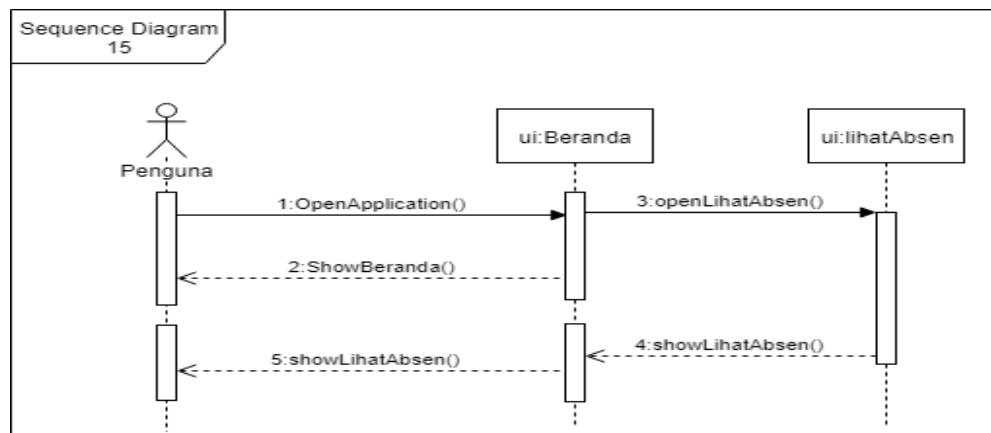


Gambar 3.34 Sequence Diagram Lihat Nilai

Pada Gambar 3.34, dapat dilihat bahwa pengguna membuka aplikasi menggunakan fungsi openApplication(), lalu aplikasi akan menampilkan beranda, setelah itu untuk melihat nilai menggunakan fungsi openLihatNilai(), jika berhasil akan menampilkan data nilai siswa.

p. *Sequence Diagram Lihat Absen*

Berikut adalah *sequence diagram* lihat data absen dapat dilihat pada Gambar 3.35.

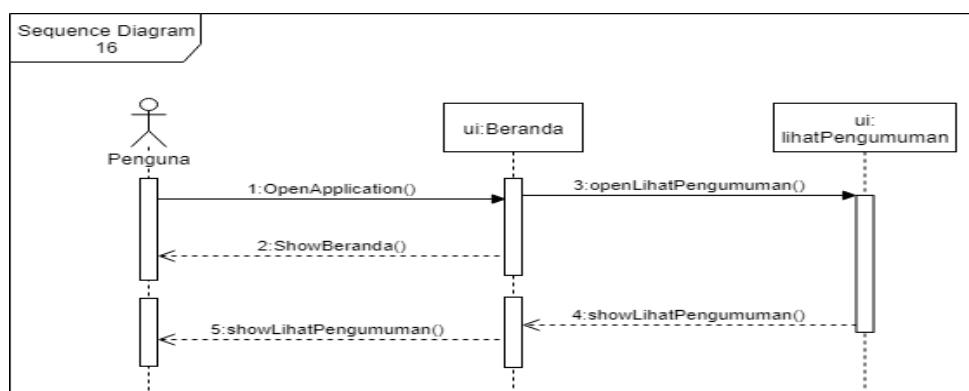


Gambar 3.35 Sequence Diagram Lihat Absen

Pada Gambar 3.35, dapat dilihat bahwa pengguna membuka aplikasi menggunakan fungsi openApplication(), lalu aplikasi akan menampilkan beranda, setelah itu untuk melihat absen menggunakan fungsi openLihatAbsen(), jika berhasil akan menampilkan data absen siswa.

q. *Sequence Diagram Lihat Pengumuman*

Berikut adalah *sequence diagram* lihat data absen dapat dilihat pada Gambar 3.36.

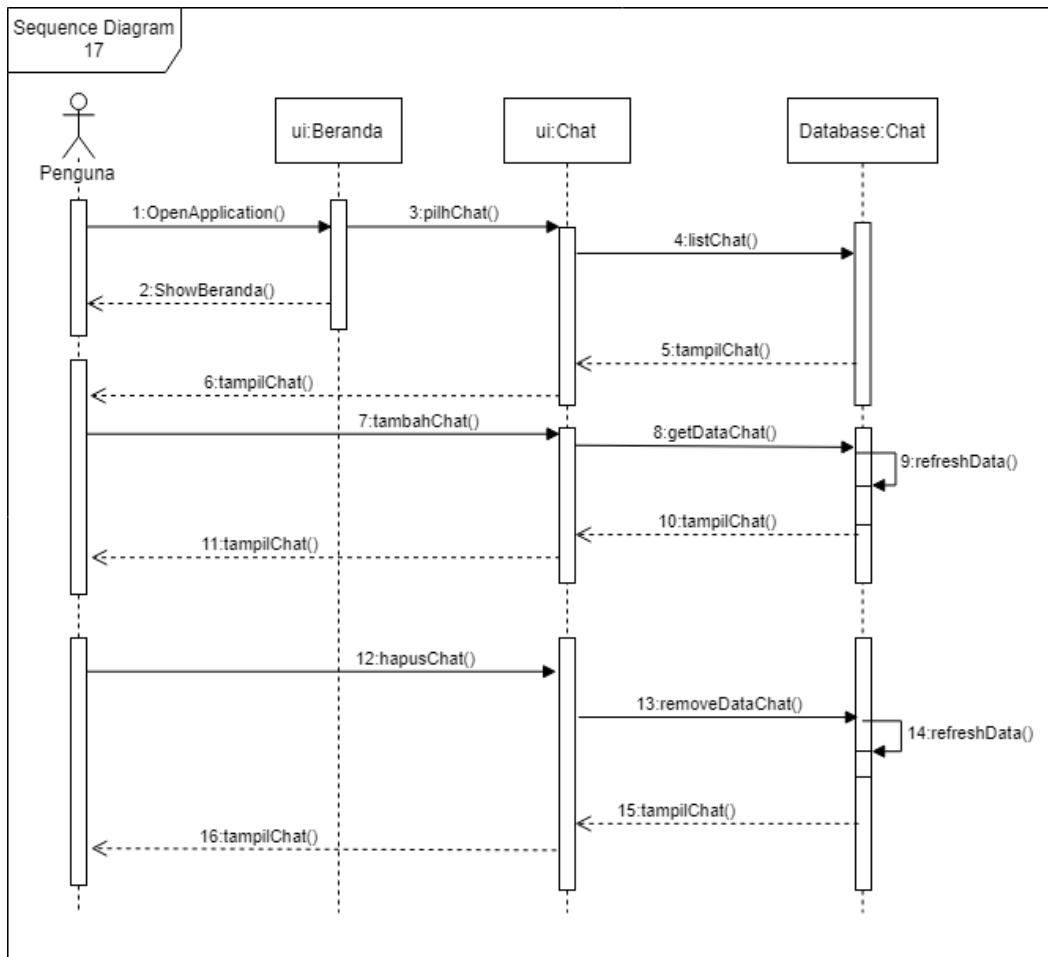


Gambar 3.36 Sequence Diagram Lihat Pengumuman

Pada Gambar 3.36, dapat dilihat bahwa pengguna membuka aplikasi menggunakan fungsi openApplication(), lalu aplikasi akan menampilkan menu beranda, setelah itu untuk melihat pengumuman menggunakan fungsi openLihatPengumuman(), jika berhasil akan menampilkan list dan detail dari pengumuman siswa.

r. *Sequence Diagram Chat*

Berikut adalah *sequence diagram* chat dapat dilihat pada Gambar 3.37.

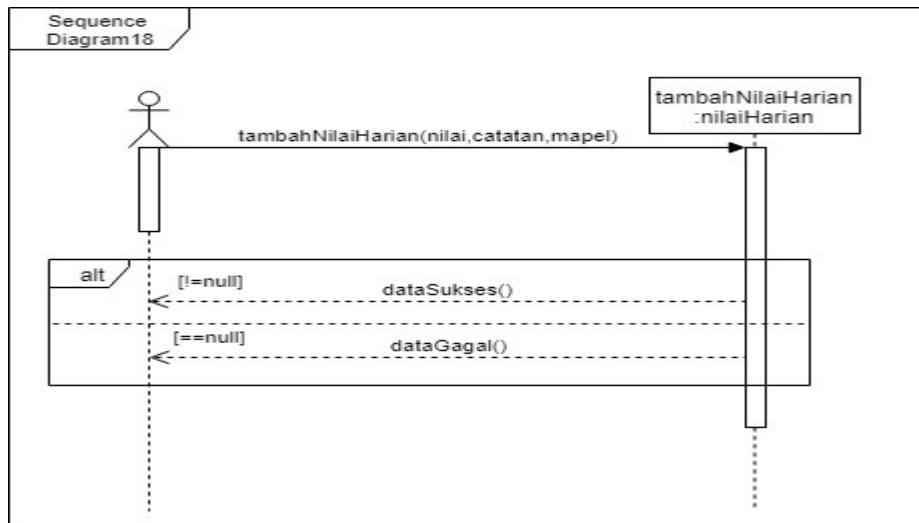


Gambar 3.37 Sequence Diagram Chat

Pada Gambar 3.37, dapat dilihat bahwa pengguna membuka aplikasi menggunakan fungsi openApplication(), lalu akan menampilkan beranda, setelah itu untuk memilih user yang akan dicheat menggunakan fungsi pilihUser(). Lalu tampil list dan form chat, setelah itu chat menggunakan fungsi tambahChat, sedangkan hapus chat menggunakan fungsi hhapusChat.

s. *Sequence Diagram* Tambah Nilai Harian

Berikut adalah *sequence diagram* chat dapat dilihat pada Gambar 3.38.

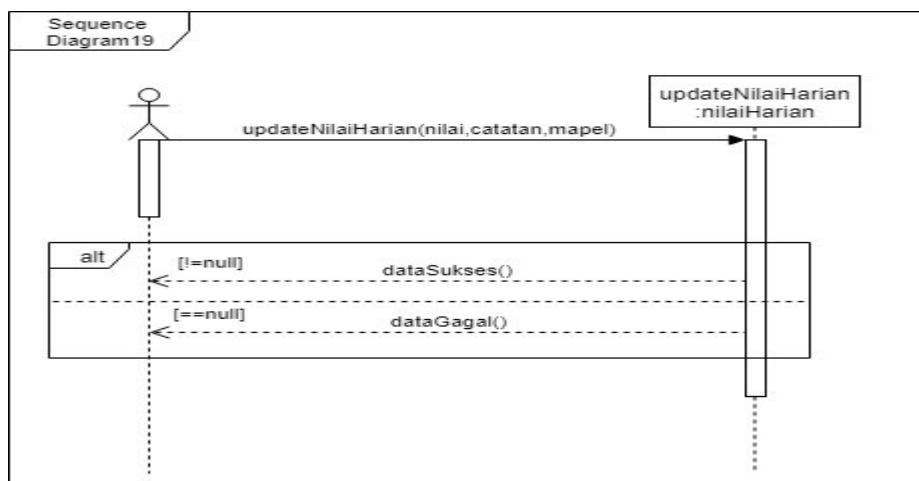


Gambar 3.38 Sequence Diagram Tambah Nilai Harian

Pada Gambar 3.38, dapat dilihat bahwa guru *input* data menggunakan fungsi `tambahNilaiHarian()` dengan menambah data nilai harian jika sukses maka data nilai akan bertambah.

t. *Sequence Diagram* Update Nilai Harian

Berikut adalah *sequence diagram* chat dapat dilihat pada Gambar 3.39.

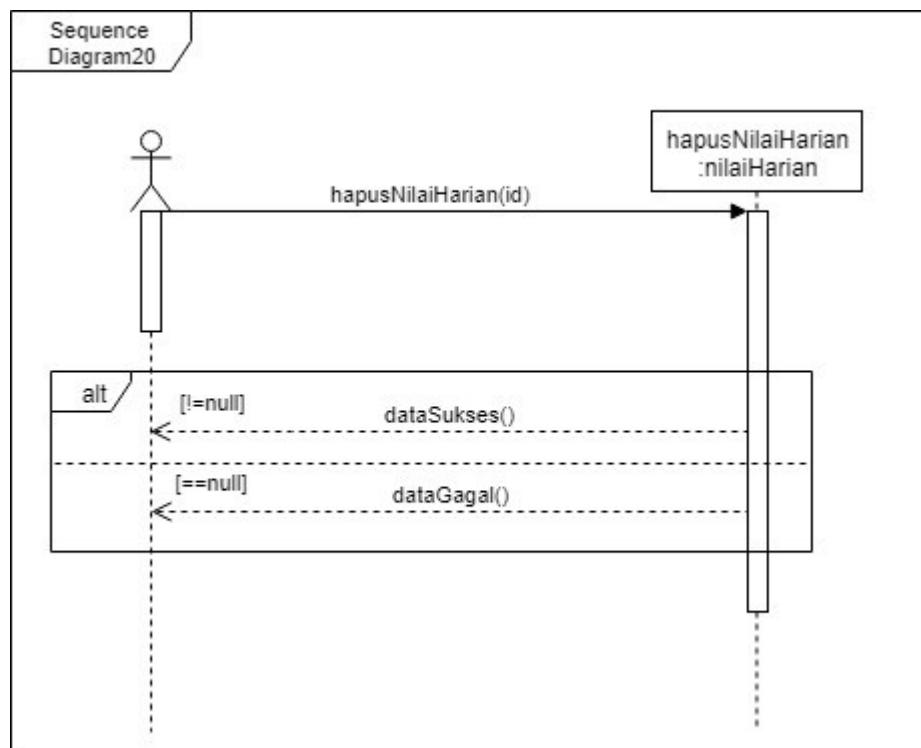


Gambar 3.39 Sequence Diagram Update Nilai Harian

Pada Gambar 3.39, dapat dilihat bahwa guru *input* data menggunakan fungsi `updateNilaiHarian()` dengan update data nilai harian jika sukses maka data nilai akan terupdate.

u. *Sequence Diagram* Hapus Nilai Harian

Berikut adalah *sequence diagram* yang dapat dilihat pada Gambar 3.40.



Gambar 3.40 Sequence Diagram Hapus Nilai Harian

Pada Gambar 3.40, dapat dilihat bahwa guru *input* data menggunakan fungsi `hapusNilaiHarian()` dengan hapus data nilai harian jika sukses maka data nilai akan terhapus.

III.2.3.8 Perancangan Basis Data

Sistem ini memiliki 13 tabel, yang dapat dilihat pada rincian berikut.

a. Spesifikasi Tabel User

Nama *database* : users

Nama tabel : sekolah

Keterangan : tabel yang digunakan untuk masuk ke dalam sistem.

Tabel 3.4 Spesifikasi Tabel User

Nama Field	Tipe	Boleh Kosong	Keterangan
id	<i>int</i> (10)	Tidak	Kunci Primer
name	<i>varchar</i> (191)	Tidak	
tipe_user	<i>varchar</i> (191)	Tidak	Level Pengguna
email	<i>varchar</i> (191)	Tidak	
password	<i>varchar</i> (191)	Tidak	

b. Spesifikasi Tabel Siswa

Nama *database* : siswa

Nama tabel : sekolah

Keterangan : tabel yang digunakan untuk masuk ke dalam sistem.

Tabel 3.5 Spesifikasi Tabel Siswa

Nama Field	Tipe	Boleh Kosong	Keterangan
id	<i>int</i> (11)	Tidak	Kunci Primer
nama	<i>varchar</i> (191)	Tidak	
nism	<i>varchar</i> (191)	Boleh	
nis	<i>varchar</i> (191)	Tidak	
jenis_kelamin	<i>varchar</i> (191)	Tidak	
agama	<i>varchar</i> (191)	Tidak	
alamat	<i>varchar</i> (191)	Tidak	

c. Spesifikasi Tabel Guru

Nama *database* : guru
 Nama tabel : sekolah
 Keterangan : tabel yang digunakan untuk masuk ke dalam sistem.

Tabel 3.6 Spesifikasi Tabel Guru

Nama Field	Tipe	Boleh Kosong	Keterangan
id	<i>int</i> (10)	Tidak	Kunci Primer
user_id	<i>int</i> (10)	Tidak	
name	<i>varchar</i> (191)	Tidak	
nip	<i>varchar</i> (191)	Boleh	
tempat_lahir	<i>varchar</i> (191)	Tidak	
tgl_lahir	<i>date</i>	Tidak	
jenis_kelamin	<i>varchar</i> (191)	Tidak	
agama	<i>varchar</i> (191)	Tidak	
alamat	<i>varchar</i> (191)	Tidak	

d. Spesifikasi Tabel Tahun Ajaran

Nama *database* : tahun_ajaran
 Nama tabel : sekolah
 Keterangan : tabel yang digunakan untuk masuk ke dalam sistem.

Tabel 3.7 Spesifikasi Tabel Tahun Ajaran

Nama Field	Tipe	Boleh Kosong	Keterangan
id	<i>int</i> (10)	Tidak	Kunci Primer
tahun_mulai	<i>varchar</i> (191)	Tidak	
tahun_akhir	<i>varchar</i> (191)	Tidak	

e. Spesifikasi Tabel Kelas

Nama *database* : kelas
 Nama tabel : sekolah
 Keterangan : tabel yang digunakan untuk masuk ke dalam sistem.

Tabel 3.8 Spesifikasi Tabel Kelas

Nama Field	Tipe	Boleh Kosong	Keterangan
id	<i>int</i> (10)	Tidak	Kunci Primer
id_tahun_ajaran	<i>int</i> (10)	Tidak	<i>Foreign Key</i>
user_id	<i>varchar</i> (191)	Tidak	<i>Foreign Key</i>
nama_kelas	<i>varchar</i> (191)	Tidak	

f. Spesifikasi Tabel Mapel

Nama *database* : mapel
 Nama tabel : sekolah
 Keterangan : tabel yang digunakan untuk masuk ke dalam sistem.

Tabel 3.9 Spesifikasi Tabel Mapel

Nama Field	Tipe	Boleh Kosong	Keterangan
id	<i>int</i> (10)	Tidak	Kunci Primer
id_tahun_ajaran	<i>int</i> (10)	Tidak	<i>Foreign Key</i>
nama_mapel	<i>varchar</i> (191)	Tidak	

g. Spesifikasi Tabel Guru Mapel

Nama *database* : guru_mapel
 Nama tabel : sekolah
 Keterangan : tabel yang digunakan untuk masuk ke dalam sistem.

Tabel 3.10 Spesifikasi Tabel Guru Mapel

Nama Field	Tipe	Boleh Kosong	Keterangan
id	<i>int</i> (10)	Tidak	Kunci Primer
id_kelas	<i>int</i> (10)	Tidak	<i>Foreign Key</i>
id_guru	<i>int</i> (10)	Tidak	<i>Foreign Key</i>
id_mapel	<i>int</i> (10)	Tidak	<i>Foreign Key</i>

h. Spesifikasi Tabel Pengumuman

Nama *database* : pengumuman

Nama tabel : sekolah

Keterangan : tabel yang digunakan untuk masuk ke dalam sistem.

Tabel 3.11 Spesifikasi Tabel Pengumuman

Nama Field	Tipe	Boleh Kosong	Keterangan
id	<i>int</i> (10)	Tidak	Kunci Primer
nama_pembuat	<i>varchar</i> (191)	Tidak	
judul	<i>varchar</i> (191)	Tidak	
isi	<i>text</i>	Tidak	

i. Spesifikasi Tabel Fase

Nama *database* : fase

Nama tabel : sekolah

Keterangan : tabel yang digunakan untuk masuk ke dalam sistem.

Tabel 3.12 Spesifikasi Tabel Fase

Nama Field	Tipe	Boleh Kosong	Keterangan

id	<i>int (10)</i>	Tidak	Kunci Primer
id_tahun_ajaran	<i>int (10)</i>	Tidak	<i>Foreign Key</i>
fase	<i>varchar(10)</i>	Tidak	
tgl_mulai	<i>date</i>	Tidak	
tgl_selesai	<i>date</i>	Tidak	

j. Spesifikasi Tabel Absen

Nama *database* : absens
 Nama tabel : sekolah
 Keterangan : tabel yang digunakan untuk masuk ke dalam sistem.

Tabel 3.13 Spesifikasi Tabel Absen

Nama Field	Tipe	Boleh Kosong	Keterangan
id	<i>int (10)</i>	Tidak	Kunci Primer
id_siswa	<i>int (10)</i>	Tidak	<i>Foreign Key</i>
id_kelas	<i>int(10)</i>	Tidak	<i>Foreign Key</i>
tanggal	<i>date</i>	Tidak	
fase	<i>date</i>	Tidak	
h	<i>varchar(191)</i>	Boleh	
i	<i>varchar(191)</i>	Boleh	
a	<i>varchar(191)</i>	Boleh	
s	<i>varchar(191)</i>	Boleh	

k. Spesifikasi Tabel Chat

Nama *database* : chat
 Nama tabel : sekolah
 Keterangan : tabel yang digunakan untuk masuk ke dalam sistem.

Tabel 3.14 Spesifikasi Tabel Chat

Nama Field	Tipe	Boleh Kosong	Keterangan
id	<i>int (10)</i>	Tidak	Kunci Primer
id_user1	<i>int (10)</i>	Tidak	<i>Foreign Key</i>
id_user2	<i>int (10)</i>	Tidak	<i>Foreign Key</i>
pesan	<i>double</i>	Tidak	

l. Spesifikasi Tabel Laporan Mapel

Nama *database* : laporan_mapel

Nama tabel : sekolah

Keterangan : tabel yang digunakan untuk masuk ke dalam sistem.

Tabel 3.15 Spesifikasi Tabel Laporan Mapel

Nama Field	Tipe	Boleh Kosong	Keterangan
id	<i>int (10)</i>	Tidak	Kunci Primer
id_laporan	<i>int (10)</i>	Tidak	<i>Foreign Key</i>
id_mapel	<i>int (10)</i>	Tidak	<i>Foreign Key</i>
nilai_pengetahuan	<i>double</i>	Tidak	
nilai_keterampilan	<i>double</i>	Tidak	

m. Spesifikasi Tabel Laporan

Nama *database* : laporan_mapel

Nama tabel : sekolah

Keterangan : tabel yang digunakan untuk masuk ke dalam sistem.

Tabel 3.16 Spesifikasi Tabel Laporan

Nama Field	Tipe	Boleh Kosong	Keterangan
id	<i>int (10)</i>	Tidak	Kunci Primer
id_kelas	<i>int (10)</i>	Tidak	<i>Foreign Key</i>
id_siswa	<i>int (10)</i>	Tidak	<i>Foreign Key</i>
fase	<i>int(10)</i>	Tidak	

n. Spesifikasi Tabel Player

Nama *database* : player
 Nama tabel : sekolah
 Keterangan : tabel yang digunakan untuk masuk ke dalam sistem.

Tabel 3.17 Spesifikasi Tabel Laporan

Nama Field	Tipe	Boleh Kosong	Keterangan
id	<i>int (10)</i>	Tidak	Kunci Primer
user_id	<i>int (10)</i>	Tidak	<i>Foreign Key</i>
player_id	<i>Varchar (191)</i>	Tidak	

o. Spesifikasi Tabel Laporan Nilai Harian

Nama *database* : laporan_harian
 Nama tabel : sekolah
 Keterangan : tabel yang digunakan untuk masuk ke dalam sistem.

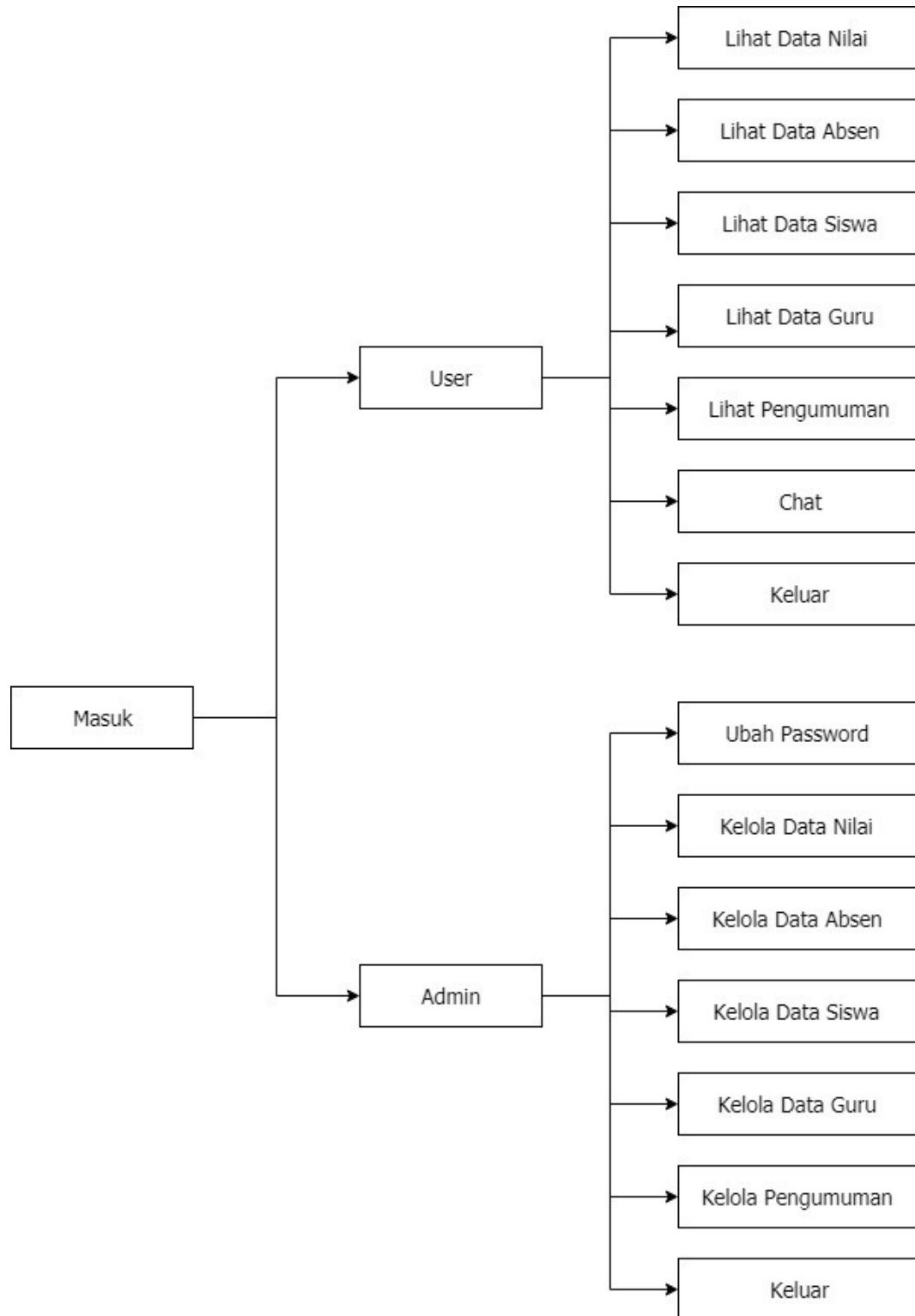
Tabel 3.18 Spesifikasi Tabel Laporan Nilai Harian

Nama Field	Tipe	Boleh Kosong	Keterangan
id	<i>int (10)</i>	Tidak	Kunci Primer
id_laporan	<i>int (10)</i>	Tidak	<i>Foreign Key</i>
id_mapel	<i>int (10)</i>	Tidak	<i>Foreign Key</i>

tanggal	<i>Date</i>	Tidak	
nilai	<i>double</i>	tidak	
catatan	Text	tidak	

III.2.4 Perancangan Antarmuka Sistem

III.2.4.1 Perancangan Struktur Antarmuka Sistem



Gambar 3.41 Perancangan Struktur Antarmuka Sistem

Sistem ini merupakan aplikasi berbasis android dan *website* sebagai *web service* yang nantinya setiap data yang masuk akan tersimpan di database. Antarmuka aplikasi yang dibangun memiliki beberapa tampilan yang disesuaikan dengan menu – menu aplikasi yang ada. Struktur antarmuka aplikasi dapat dilihat pada Gambar 3.41.

III.2.4.2 Perancangan *Layout* dan Komponen Antarmuka Sistem

a. Perancangan Halaman Login

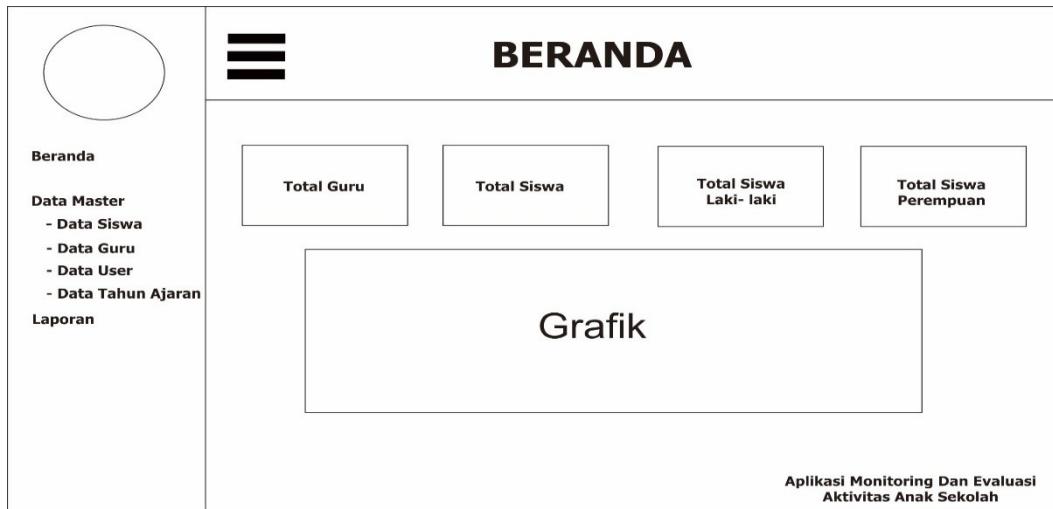
Halaman *login* berfungsi untuk menjaga keamanan sistem dan hanya yang berhak dalam mengelolah *website* yang diizinkan masuk ke dalam sistem. Halaman ini berisi *username* serta *password*, apabila pengguna mengisikan data *username* dan *password* dengan benar maka pengguna dapat mengakses halaman manajemen data. Rancangan Halaman *Login* dapat dilihat pada Gambar 3.42.



Gambar 3.42 Rancangan Halaman *Login*

b. Perancangan Halaman Beranda

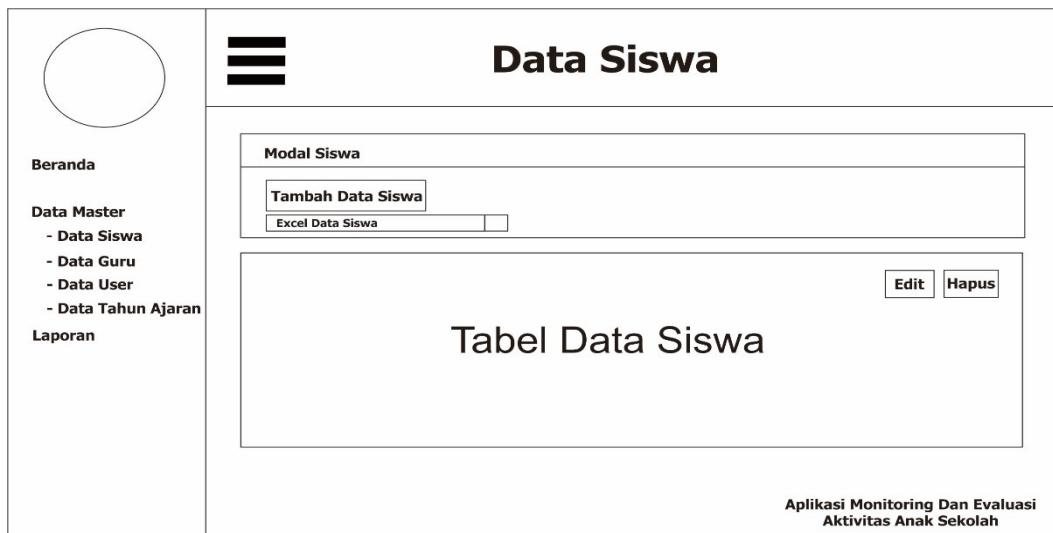
Halaman beranda berfungsi untuk menampilkan informasi data jumlah guru, jumlah siswa laki-laki, siswa perempuan dan menampilkan grafik. Rancangan Halaman Beranda dapat dilihat pada Gambar 3.43.



Gambar 3.43 Rancangan Halaman Beranda

c. Halaman Data Siswa

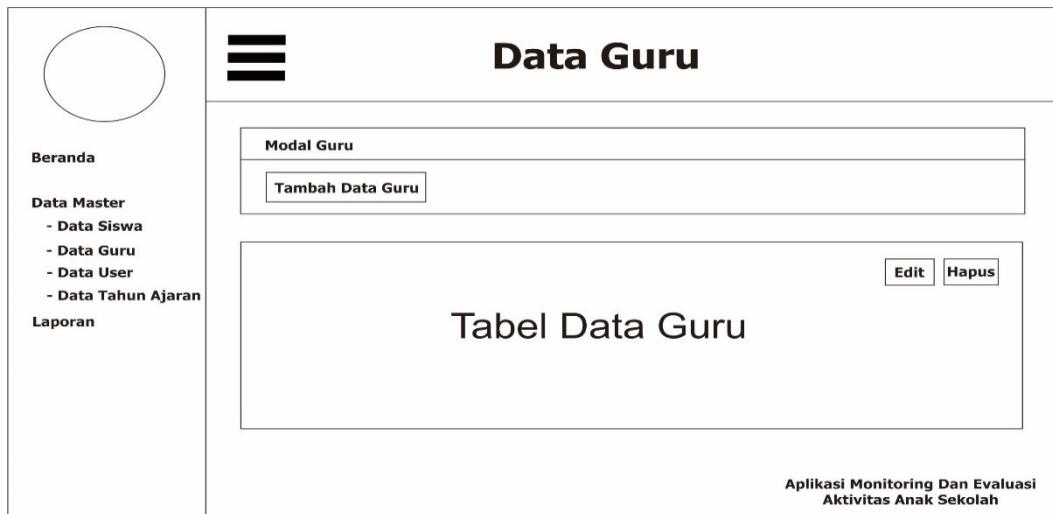
Halaman data siswa berfungsi untuk mengelola data siswa seperti tambah data siswa, edit data siswa dan hapus data siswa serta terdapat tombol button excel untuk memasukkan data siswa melalui microsoft excel. Rancangan Halaman Data Siswa dapat dilihat pada Gambar 3.44.



Gambar 3.44 Rancangan Halaman Data Siswa

d. Halaman Data Guru

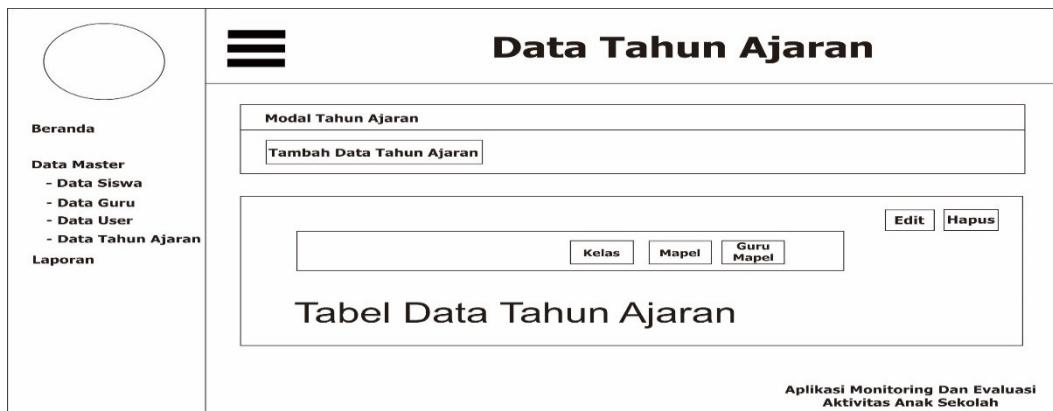
Halaman data guru berfungsi untuk mengelola data guru seperti tambah data guru, edit data guru dan hapus data guru. Rancangan Halaman Data Guru dapat dilihat pada Gambar 3.45.



Gambar 3.45 Rancangan Halaman Data Guru

e. Halaman Data Tahun Ajaran

Halaman data tahun ajaran berfungsi untuk mengelola data tahun ajaran seperti tambah data tahun ajaran, edit data tahun ajaran dan hapus data tahun ajaran. Pada halaman tahun ajaran saat data sudah di isi, tombol kelas, mapel dan guru mapel tampil sesuai dengan tahun ajaran. Rancangan Halaman Data Tahun Ajaran dapat dilihat pada Gambar 3.46.



Gambar 3.46 Rancangan Halaman Data Tahun Ajaran

f. Halaman Data Kelas

Halaman data kelas berfungsi untuk mengelola data kelas seperti tambah data kelas, edit data kelas dan hapus data kelas. Rancangan Halaman Data Kelas dapat dilihat pada Gambar 3.47.

The screenshot shows a web application interface titled "Data Kelas". On the left sidebar, there is a navigation menu with "Beranda", "Data Master" (which includes "Data Siswa", "Data Guru", "Data User", and "Data Tahun Ajaran"), and "Laporan". The main content area has a title "Data Kelas" with a three-line menu icon above it. Below the title is a modal window titled "Modal Kelas" with a "Tambah Data Kelas" button. To the right of the modal is a table header "Tabel Data Kelas". At the bottom right of the table area are "Edit" and "Hapus" buttons. Below the table is a footer bar with four buttons labeled "Ganjil", "Mid Ganjil", "Mid Genap", and "Genap". The footer also contains the text "Aplikasi Monitoring Dan Evaluasi Aktivitas Anak Sekolah".

Gambar 3.47 Rancangan Halaman Data Kelas

g. Halaman Data Mata Pelajaran

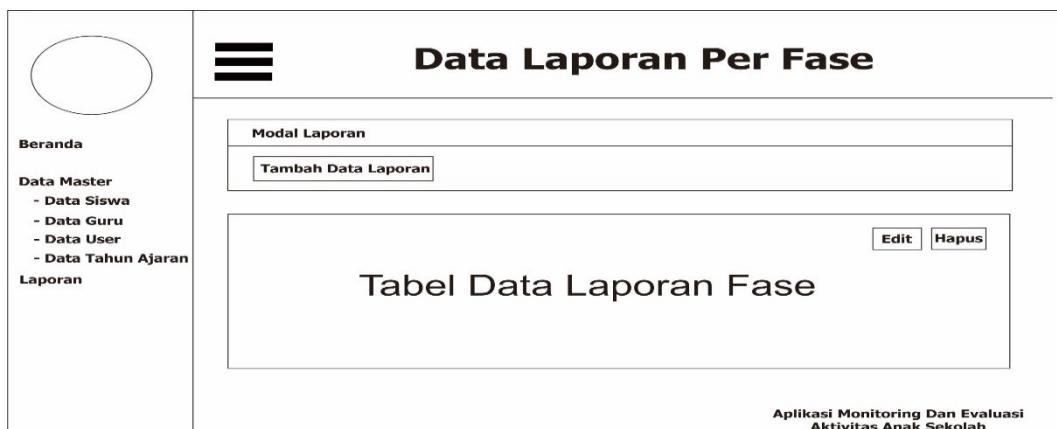
Halaman data mata pelajaran berfungsi untuk mengelola data mata pelajaran seperti tambah data mata pelajaran, edit data mata pelajaran dan hapus data mata pelajaran. Rancangan Halaman Data Mata Pelajaran dapat dilihat pada Gambar 3.48.

The screenshot shows a web application interface titled "Data Mapel". The left sidebar has the same navigation menu as in Gambar 3.47. The main content area has a title "Data Mapel" with a three-line menu icon above it. Below the title is a modal window titled "Modal Mapel" with a "Tambah Data Mapel" button. To the right of the modal is a table header "Tabel Data Mapel". At the bottom right of the table area are "Edit" and "Hapus" buttons. Below the table is a footer bar with four buttons labeled "Ganjil", "Mid Ganjil", "Mid Genap", and "Genap". The footer also contains the text "Aplikasi Monitoring Dan Evaluasi Aktivitas Anak Sekolah".

Gambar 3.48 Rancangan Halaman Data Mata Pelajaran

h. Halaman Data Laporan Perfase

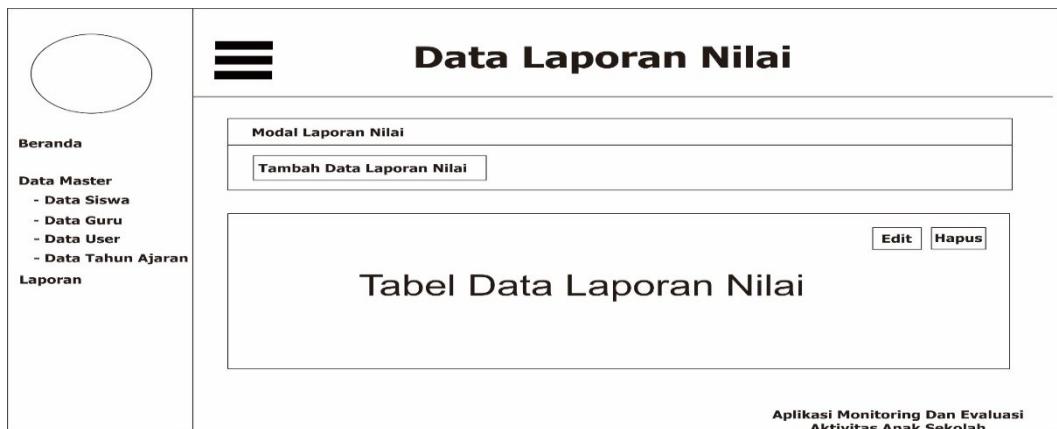
Halaman data laporan berfungsi untuk mengelola data laporan seperti tambah data laporan, edit data laporan dan hapus data laporan. Pada halaman data laporan terdapat empat fase yaitu laporan mid ganjil, laporan semester ganjil, laporan mid genap, laporan semester genap. Rancangan Halaman Data Laporan dapat dilihat pada Gambar 3.49.



Gambar 3.49 Rancangan Halaman Laporan Per Fase

i. Halaman Data Laporan Nilai

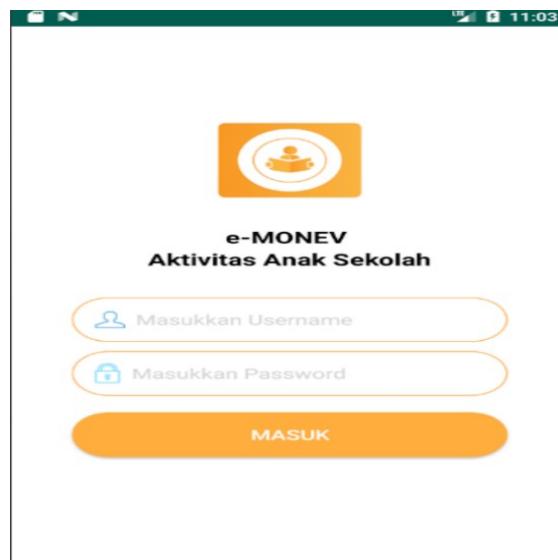
Halaman data laporan nilai berfungsi untuk mengelola data laporan nilai seperti tambah data laporan nilai, edit data laporan nilai dan hapus data laporan nilai. Rancangan Halaman Data Laporan Nilai dapat dilihat pada Gambar 3.50.



Gambar 3.50 Rancangan Halaman Laporan Nilai Siswa

j. Halaman Login Android

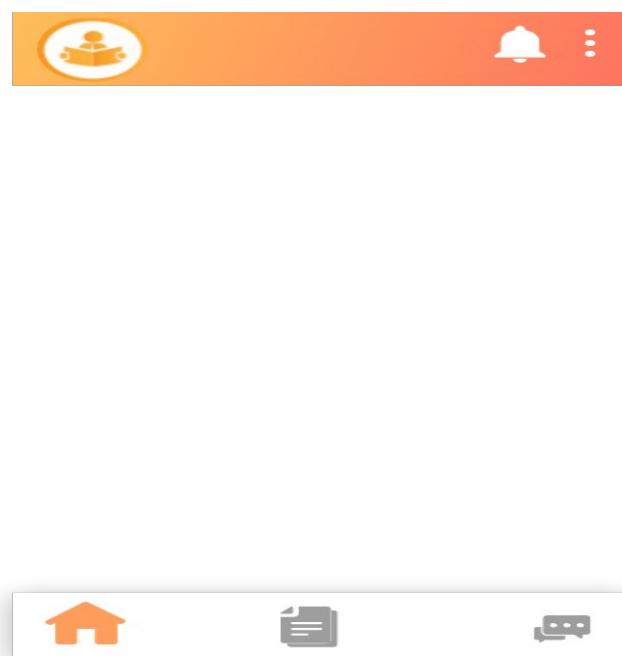
Halaman *login* berfungsi untuk menjaga keamanan sistem dan hanya yang berhak dalam mengelola android yang diizinkan masuk ke dalam sistem. Halaman ini berisi *username* serta *password*, apabila pengguna mengisikan data *username* dan *password* dengan benar maka pengguna dapat mengakses halaman dashboard sesuai dengan user. Rancangan Halaman *Login* Android dapat dilihat pada Gambar 3.51.



Gambar 3.51 Rancangan Halaman *Login* Android

k. Halaman *Dashboard* Guru Android

Halaman data *dashboard* guru di android terdapat beberapa menu yang dapat diakses sebagai informasi monitoring dan evaluasi siswa serta terdapat informasi profile dari guru tersebut. Rancangan Halaman Data *Dashboard* Guru dapat dilihat pada Gambar 3.52.



Gambar 3.52 Rancangan Halaman *Dashboard* Guru

l. Halaman *Dashboard* Guru Android

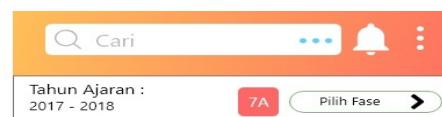
Halaman data *dashboard* siswa di android hanya dapat diakses oleh orang tua siswa, terdapat beberapa menu yang dapat diakses sebagai informasi monitoring dan evaluasi siswa serta terdapat informasi profile dari siswa tersebut. Rancangan Halaman Data *Dashboard* Siswa dapat dilihat pada Gambar 3.53.



Gambar 3.53 Rancangan Halaman *Dashboard* Siswa

m. Halaman Pilih Tahun Ajaran

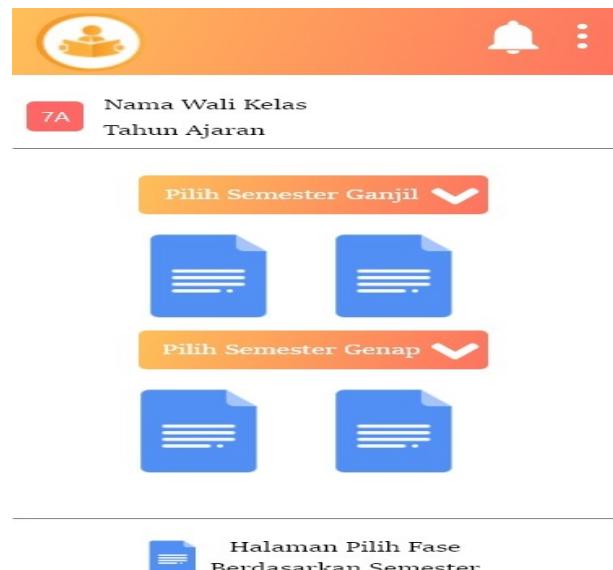
Halaman pilih tahun ajaran menampilkan informasi tentang list tahun ajaran dan kelas pada tampilan halaman guru dan siswa. Rancangan Halaman Pilih Tahun Ajaran dapat dilihat pada Gambar 3.54.



Gambar 3.54 Rancangan Halaman Pilih Tahun Ajaran

n. Halaman Pilih Fase

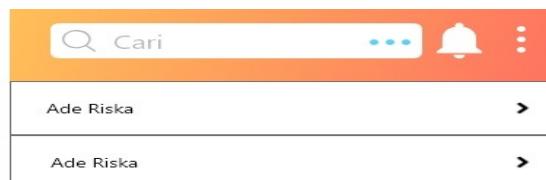
Halaman pilih fase menampilkan list fase berdasarkan semester ganjil dan semester genap, terdapat empat fase pertama mid semester ganjil, semester ganjil, mid semester genap dan semester genap. Rancangan Halaman Pilih Fase dapat dilihat pada Gambar 3.55.



Gambar 3.55 Rancangan Halaman Pilih Tahun Ajaran

o. Halaman Pilih Siswa

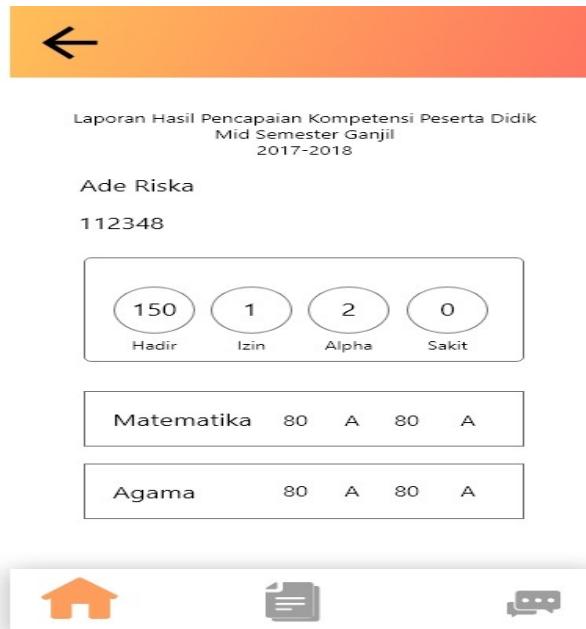
Halaman pilih Siswa menampilkan list siswa. Rancangan Halaman Pilih Siswa dapat dilihat pada Gambar 3.56.



Gambar 3.56 Rancangan Halaman Pilih Siswa

p. Halaman Laporan Siswa

Halaman Laporan Siswa menampilkan list nilai dan absen siswa. Rancangan Halaman Pilih Laporan Siswa dapat dilihat pada Gambar 3.57.



Gambar 3.57 Rancangan Halaman Laporan Siswa

q. Halaman List Absen Perhari Siswa

Halaman List Absen Perhari Siswa menampilkan list absen dengan kode absen h hadir, i izin, a alpa dan s sakit. Rancangan Halaman List Absen Perhari Siswa dapat dilihat pada Gambar 3.58.



Gambar 3.58 Rancangan Halaman List Absen Perhari Siswa

r. Halaman List Pengumuman

Halaman List Pengumuman menampilkan list pengumuman. Rancangan Halaman List Pengumuman dapat dilihat pada Gambar 3.59.



Gambar 3.59 Rancangan Halaman List Pengumuman

s. Halaman Chat

Halaman Chat menampilkan list chat antara orang tua dan guru. Rancangan Halaman List Absen Perhari Siswa dapat dilihat pada Gambar 3.60.



Gambar 3.60 Rancangan Halaman Chat

III.2.5 Pelaksanaan

III.2.5.1 Pelaksanaan Sistem

RAD adalah sebuah metode pengembangan software yang diciptakan untuk menekan waktu yang dibutuhkan untuk mendesain serta mengimplementasikan sistem informasi sehingga dihasilkan siklus pengembangan yang sangat pendek (-+ 60 sampai 90 hari). Pendekatan RAD pada aplikasi monitoring dan evaluasi aktivitas anak sekolah meliputi fase-fase berikut:

a. Rencana Kebutuhan (*Requirement Planning*)

Pada tahap ini, *programmer* bertemu dengan pihak Dinas Pendidikan dalam pertemuan ini bertujuan untuk mengidentifikasi tujuan dari sistem dan kebutuhan informasi pada aplikasi yang akan dibangun dan dirancang nantinya. Pada tahap ini merupakan hal terpenting yaitu adanya keterlibatan dari kedua belah pihak.

b. Proses Desain Sistem (*Design System*)

Pada tahap ini keaktifan *programmer* yang terlibat dalam pembuatan aplikasi dalam menentukan tujuan dari sistem yang akan dibangun, karena pada proses ini dilakukan proses perancangan, desain, dan perbaikan-perbaikan apabila masih terdapat ketidaksesuaian antara *programmer* dan pihak Dinas Pendidikan. Pihak Dinas Pendidikan dapat langsung memberikan komentar apabila terdapat ketidaksesuaian perancangan dan desain yang telah dirancang pada sistem dengan mengacu pada dokumentasi kebutuhan dari pihak Dinas Pendidikan yang dibuat pada tahap sebelumnya. Keluaran dari tahapan ini adalah spesifikasi software yang meliputi pemodelan sistem dan arsitektur sistem.

c. Implementasi (*Implementation*)

Pada tahap ini *programmer* yang mengembangkan dan mengerjakan proses *coding* dan desain yang telah disetujui sebelumnya oleh *programmer* dan pihak Dinas Pendidikan. Sebelum sistem diimplementasikan dilakukan proses pengujian terhadap program tersebut apakah ada kesalahan atau tidak. Pada tahap ini pihak Dinas Pendidikan, pihak Sekolah maupun Orang tua memberikan tanggapan akan sistem yang sudah dibuat serta mendapat persetujuan mengenai sistem tersebut.

III.2.5.2 Pengujian Sistem

Tahap pengujian dalam penelitian ini akan menggunakan pengujian *black-box* dan pengujian Kuesioner sebagai kompatibilitas aplikasi. Pengujian *black box* digunakan untuk menguji validitas dari integrasi dan konsistensi sistem.

a. Pengujian Black Box

Pengujian *black box* pada aplikasi ini menggunakan alpha testing. Pada pengujian kompatibilitas aplikasi, teknik pengujian blackbox yang digunakan yaitu *performance testing* dan *comparison testing*. pengujian akan dilakukan dengan memasang aplikasi pada berbagai jenis perangkat Android dan kemudian mengakses semua aktivitas dan fitur.

b. Pengujian Kuesioner

Pengujian kuesioner yaitu menguji tingkat penerimaan pengguna terhadap sistem melalui kuesioner. Kuesioner berisi beberapa pertanyaan yang dikelompokkan menjadi 3 aspek yang digunakan dalam pengujian aplikasi tersebut, yaitu aspek rekayasa perangkat lunak, aspek fungsionalitas, dan aspek komunikasi visual. Kuesioner dibagikan kepada 30 responden. Responden akan mencoba aplikasi untuk dapat mengetahui kapabilitas dan kelancaran jalannya aplikasi itu sendiri.

Perhitungan dilakukan dengan menggunakan skala likert, dimana setiap pertanyaan memiliki bobot nilai. Adapun bobot jawaban yang diberikan adalah sebagai berikut:

Tabel 3.19 Bobot jawaban

<i>Keterangan Jawaban</i>	<i>Bobot Nilai</i>
<i>Sangat Baik</i>	5
<i>Baik</i>	4
<i>Cukup Baik</i>	3
<i>Kurang Baik</i>	2
<i>Tidak Baik</i>	1

Selanjutnya aplikasi akan di uji coba kepada pengguna. Hasil pengujian dengan kuesioner akan diukur menggunakan metode Skala Likert, pengukuran akan dilakukan kepada pengguna yang sudah mencoba aplikasi. Tabel perhitungan skala likert dapat dilihat pada tabel 3.20

Tabel 3.20 Tabel perhitungan skala likert

No .	Aspek / Pertanyaan	Nilai Pengujian					Percentase Likert
		TB	KB	CB	B	SB	
Aspek Rekayasa Perangkat Lunak							
1.	Kemudahan menjalankan aplikasi						
2.	Kelancaran menjalankan aplikasi						
3.	Kenyamanan dalam menggunakan aplikasi secara keseluruhan						
Aspek Fungsionalitas							
4.	Kemudahan menjalankan tombol menu pada aplikasi						
5.	Tingkat kesesuaian menampilkan nilai siswa, absen perhari dan pengumuman yang berkaitan dengan sekolah, pada aplikasi android						
6.	Tingkat kesesuaian aplikasi dalam menampilkan dan mengirim chat antara orang tua dan guru						
7.	Kemudahan dalam mendapatkan informasi siswa seperti nilai siswa, absen dan pengumuman di aplikasi android						
Aspek Komunikasi Visual							
8.	Kemudahan memahami informasi yang ditampilkan pada aplikasi						
9.	Tampilan						

No .	Aspek / Pertanyaan	Nilai Pengujian					Persentase Likert
		TB	KB	CB	B	SB	
	antarmuka aplikasi						
10 .	Tingkat kemudahan melihat hasil yang ditampilkan aplikasi						
Total Persentase (%)							

Nilai persentase skala likert dihitung dengan rumus berikut ini:

$$L = \frac{(SB \times 5) + (B \times 4) + (CB \times 3) + (KB \times 2) + (TB \times 1)}{NB} \times 100$$

Keterangan :

- L : Nilai persentase likert
- SB : Jumlah responden yang memilih pilihan sangat baik
- B : Jumlah responden yang memilih pilihan baik
- CB : Jumlah responden yang memilih pilihan cukup baik
- KB : Jumlah responden yang memilih pilihan kurang baik
- TB : Jumlah responden yang memilih pilihan tidak baik
- NB : 5 (bobot tertinggi) x jumlah responden

Untuk menyimpulkan hasil akhir dari aplikasi, harus diketahui terlebih dahulu nilai interval total persentase dengan metode pencarian nilai interval pada persamaan berikut ini:

$$I = 100 / 5 = 20$$

Nilai 5 didapat dari jumlah pilihan yang diberikan dalam kuesioner.

Nilai interval persentase pada kuesioner ini yaitu:

- $X \leq 19,99\%$ dikategorikan tidak baik
- $20\% \geq X \leq 39,99\%$ dikategorikan kurang baik
- $40\% \geq X \leq 59,99\%$ dikategorikan cukup baik
- $60\% \geq X \leq 79,99\%$ dikategorikan baik
- $X \geq 80\%$ dikategorikan sangat baik

III.2.6 Kesimpulan

Kesimpulan dirumuskan berdasarkan tahapan-tahapan yang telah dilakukan sebelumnya apakah sistem yang dirancang dan dibangun dapat berjalan baik sesuai dengan yang diharapkan

BAB IV

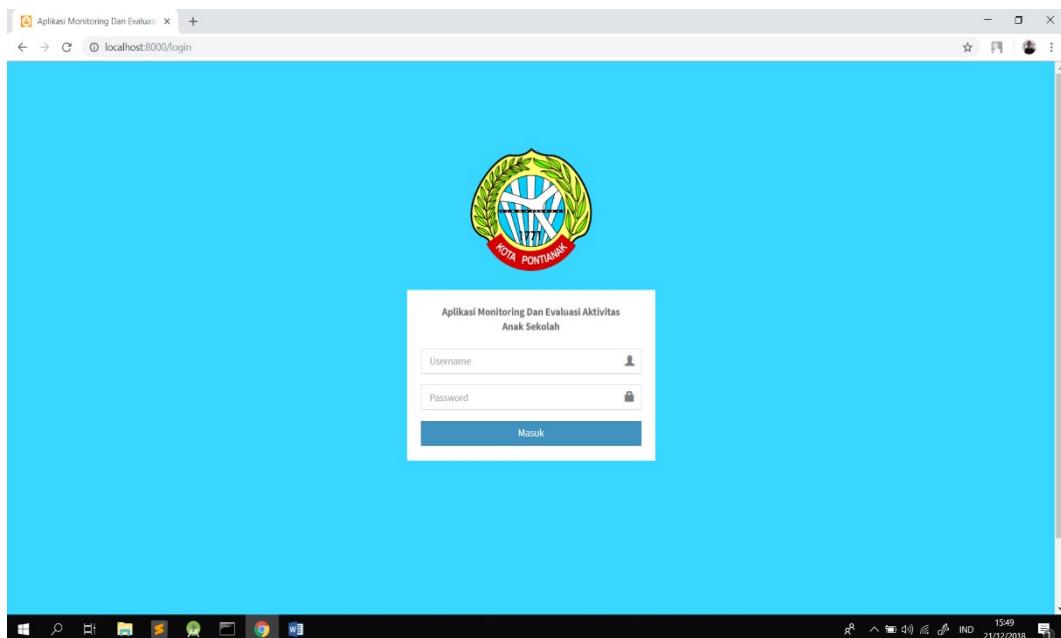
IMPLEMENTASI DAN HASIL PENGUJIAN

Berdasarkan hasil analisis dan pengujian terhadap aplikasi *monitoring* dan evaluasi aktivitas anak di sekolah SMP Negeri 5 Pontianak berbasis android, dapat disimpulkan bahwa aplikasi ini dapat menghasilkan informasi yang berkaitan dengan siswa seperti nilai yang dibagi menjadi 4 fase yang pertama mid semester ganjil, semester ganjil, mid semester genap, dan semester genap, selanjutnya absen perhari siswa disekolah, lalu terdapat pengumuman yang ditujukan kepada siswa. Berikut ini adalah penjelasan mengenai hasil perancangan antarmuka dari aplikasi yang dirancang.

IV.1 Implementasi

IV.1.1 Tampilan Antarmuka Halaman Login Website

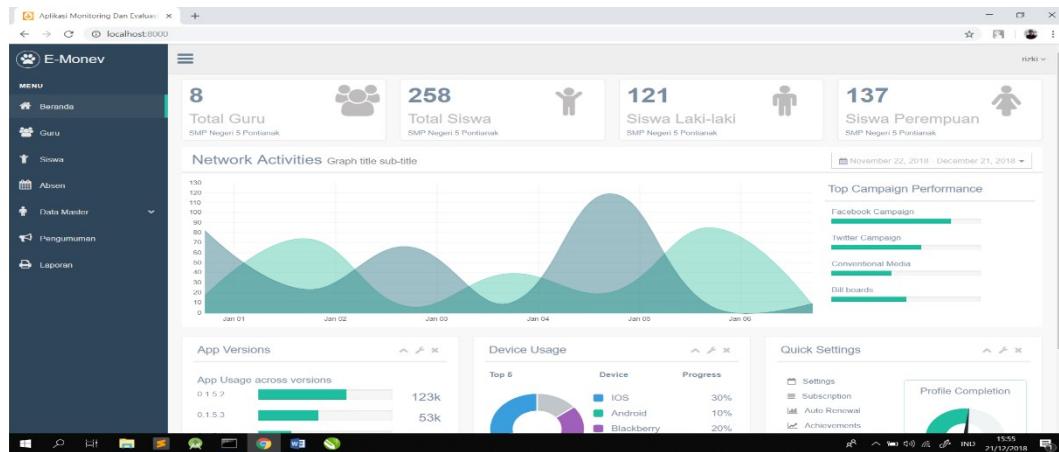
Pada tampilan halaman *login* website terdapat dua *tipe user* pengguna, yaitu admin dan guru. Pada *tipe user* admin, admin dapat melihat menu beranda, guru, siswa, pengumuman, laporan, serta master data yang digunakan untuk memasukkan data-data yang diperlukan. Sedangkan pada *tipe user*, guru dapat melihat menu beranda, guru, siswa, pengumuman dan laporan. Antarmuka halaman *login* dapat dilihat pada Gambar 4.1.



Gambar 4.1 Antarmuka Halaman Login

IV.1.2 Tampilan Antarmuka Halaman Beranda Website

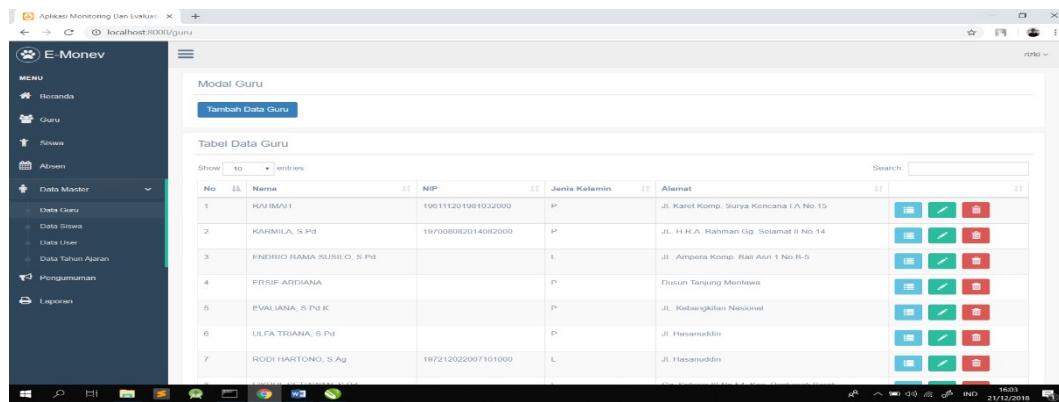
Apabila admin berhasil *login* dengan *username* dan *password* admin, maka admin akan diarahkan pada halaman beranda. Pada halaman beranda, admin dapat melihat informasi jumlah siswa. jumlah guru, jumlah siswa laki-laki dan jumlah siswa perempuan. Antarmuka halaman beranda dapat dilihat pada Gambar 4.2.



Gambar 4.2 Antarmuka Halaman Beranda

IV.1.3 Tampilan Antarmuka Halaman Master Guru

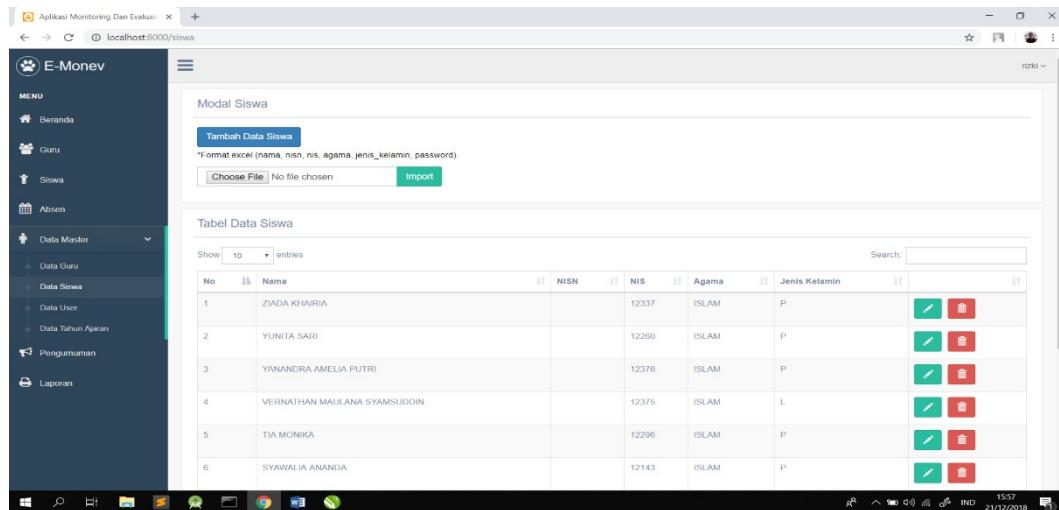
Pada halaman data master guru, admin dapat menambah, merubah maupun menghapus data master guru. Saat berhasil ditambahkan, data user untuk guru di tambah di data master user. Tampilan halaman data master guru dapat dilihat pada Gambar 4.3.



Gambar 4.3 Antarmuka Halaman Master Guru

IV.1.4 Tampilan Antarmuka Halaman Master Siswa

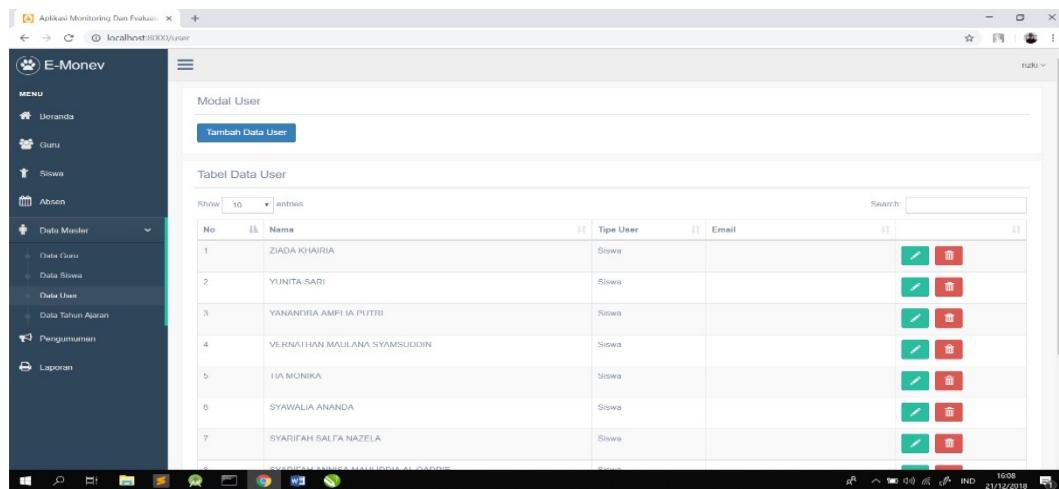
Pada halaman data master siswa, admin dapat menambah, merubah maupun menghapus data master siswa. Saat berhasil ditambahkan, data user untuk siswa di tambah di data master user. Tampilan halaman data master siswa dapat dilihat pada Gambar 4.3.



Gambar 4.4 Antarmuka Halaman Master Siswa

IV.1.5 Tampilan Antarmuka Halaman Master User

Pada halaman data master user, admin dapat menambah, merubah maupun menghapus data master user. Tampilan halaman data master siswa dapat dilihat pada Gambar 4.5.



Gambar 4.5 Antarmuka Halaman Master User

IV.1.6 Tampilan Antarmuka Halaman Master Tahun Ajaran

Pada halaman data master tahun ajaran, admin dapat menambah, merubah maupun menghapus data master tahun ajaran. Tampilan halaman data master tahun ajaran dapat dilihat pada Gambar 4.6.

No	Tahun Mulai	Tahun Akhir	Action
1	2018	2019	
2	2017	2018	

Gambar 4.6 Antarmuka Halaman Master Tahun Ajaran

IV.1.7 Tampilan Antarmuka Halaman Kelas

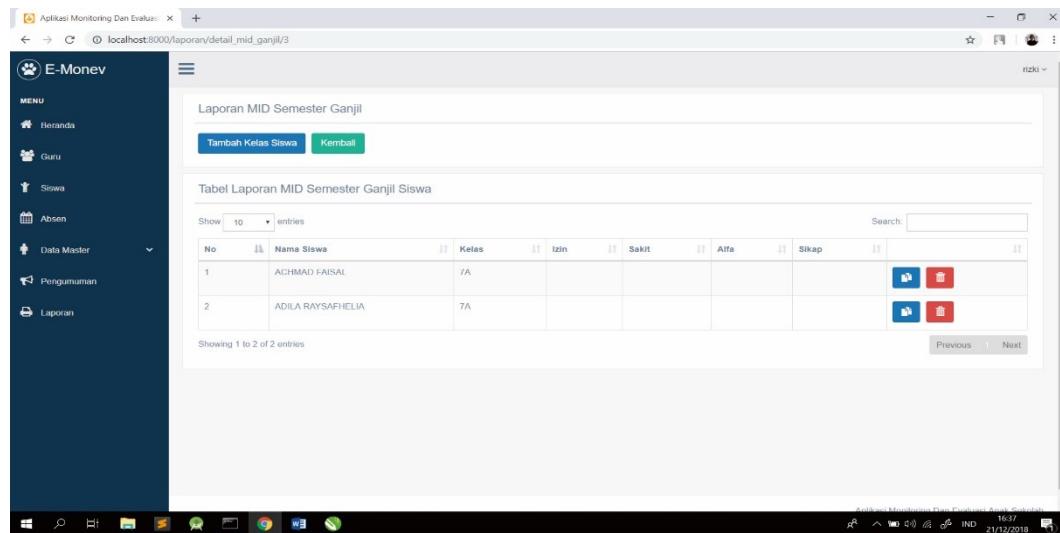
Pada halaman data kelas, admin dapat menambah, merubah maupun menghapus data kelas. Tampilan halaman data kelas dapat dilihat pada Gambar 4.7.

No	Nama Kelas	Guru	Action
1	7A	RODI HARTONO, S.Ag	
2	7B	FIKRUL SETIAWAN, S.Pd	
3	7C	ERSIE ARDIANA	

Gambar 4.7 Antarmuka Halaman Kelas

IV.1.8 Tampilan Antarmuka Halaman Laporan Siswa

Pada halaman data laporan, admin dapat menambah, merubah maupun menghapus data laporan siswa. Data laporan dibagi menjadi 4 fase mid semester ganjil, semester ganjil, mid semester genap dan semester genap. Tampilan halaman data laporan siswa dapat dilihat pada Gambar 4.8.



The screenshot shows a web-based application interface titled 'Applikasi Monitoring Dan Evaluasi'. The left sidebar has a dark blue theme with icons for Beranda, Guru, Siswa, Absen, Data Master, Pengumuman, and Laporan. The main content area is titled 'Laporan MID Semester Ganjil' and contains a table with two entries:

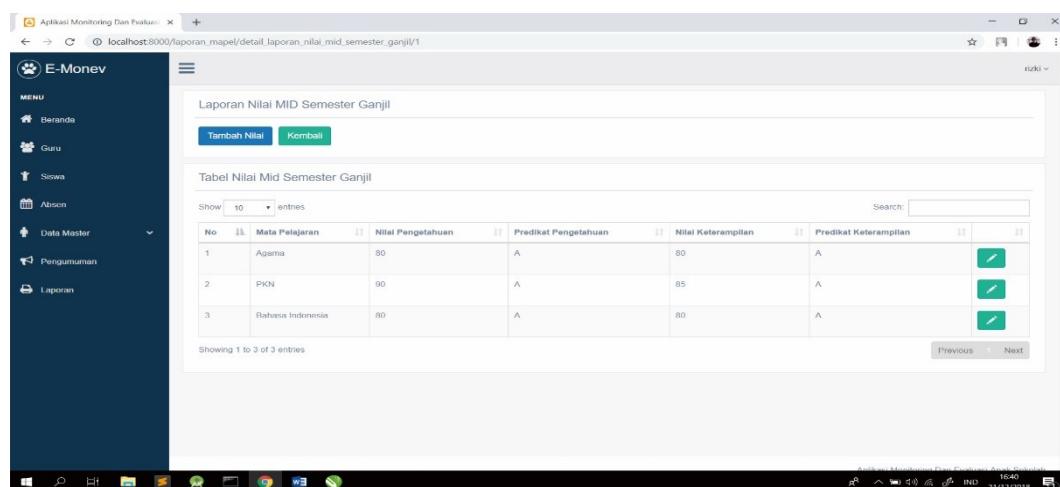
No	Nama Siswa	Kelas	Izin	Sakit	Alfa	Sikap
1	ACHMAD FAISAL	7A				
2	ADILA RAYSAFIELIA	7A				

At the bottom of the table, it says 'Showing 1 to 2 of 2 entries'. There are 'Previous' and 'Next' buttons at the bottom right. The browser address bar shows 'localhost:8000/laporan/detail_mid_ganjil/3'.

Gambar 4.8 Antarmuka Halaman Laporan Siswa

IV.1.9 Tampilan Antarmuka Halaman Nilai Siswa

Pada halaman data nilai, admin dapat menambah, merubah maupun menghapus data nilai siswa. Tampilan halaman data nilai siswa dapat dilihat pada Gambar 4.9.



The screenshot shows a web-based application interface titled 'Applikasi Monitoring Dan Evaluasi'. The left sidebar has a dark blue theme with icons for Beranda, Guru, Siswa, Absen, Data Master, Pengumuman, and Laporan. The main content area is titled 'Laporan Nilai MID Semester Ganjil' and contains a table with three entries:

No	Mata Pelajaran	Nilai Pengetahuan	Predikat Pengetahuan	Nilai Keterampilan	Predikat Keterampilan	
1	Agama	80	A	80	A	
2	PKN	90	A	85	A	
3	Bahasa Indonesia	80	A	80	A	

At the bottom of the table, it says 'Showing 1 to 3 of 3 entries'. There are 'Previous' and 'Next' buttons at the bottom right. The browser address bar shows 'localhost:8000/laporan.mapei/detail_laporan_nilai_mid_semester_ganjil/1'.

IV.1.10 Gambar 4.9 Antarmuka Halaman Nilai Siswa Tampilan Antarmuka Halaman Mata Pelajaran

Pada halaman data mata pelajaran, admin dapat menambah, merubah maupun menghapus data mata pelajaran. Tampilan halaman data mata pelajaran dapat dilihat pada Gambar 4.10.

No	Nama Mapel	
1	Agama	
2	PKN	
3	Bahasa Indonesia	
4	Matematika	
5	IPA	
6	IPS	
7	Bahasa Inggris	

Gambar 4.10 Antarmuka Halaman Mata Pelajaran

IV.1.11 Tampilan Antarmuka Halaman Pilih Fase

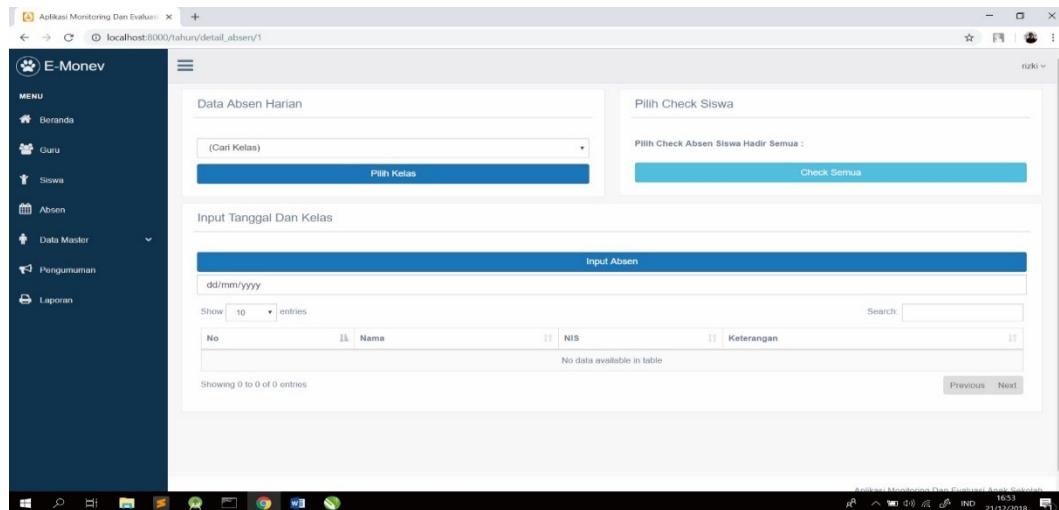
Pada halaman data fase, admin dapat menambah, merubah maupun menghapus data fase. Halaman pilih fase berfungsi menentukan tanggal dan jadwal akademik. Tampilan halaman data fase dapat dilihat pada Gambar 4.11.

No	Fase	Tanggal Mulai	Tanggal Selesai	
1	Mid Semester Ganjil	2017-09-15	2017-09-21	
2	Semester Ganjil	2017-09-20	2017-12-23	
3	Mid Semester Genap	2018-01-10	2018-03-25	
4	Semester Genap	2018-03-26	2018-06-10	

Gambar 4.11 Antarmuka Halaman Pilih Fase

IV.1.12 Tampilan Antarmuka Halaman Absensi Harian

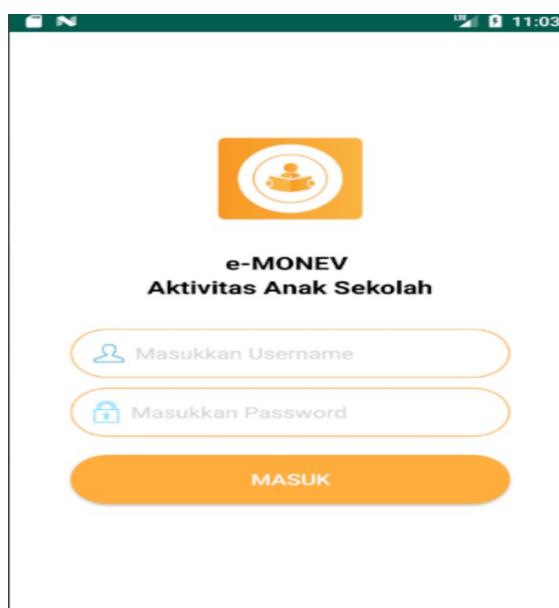
Pada halaman absensi harian, admin dapat menambah absensi harian. Tampilan halaman data fase dapat dilihat pada Gambar 4.12.



Gambar 4.12 Antarmuka Halaman Absensi Harian

IV.1.13 Tampilan Antarmuka Halaman Login Android

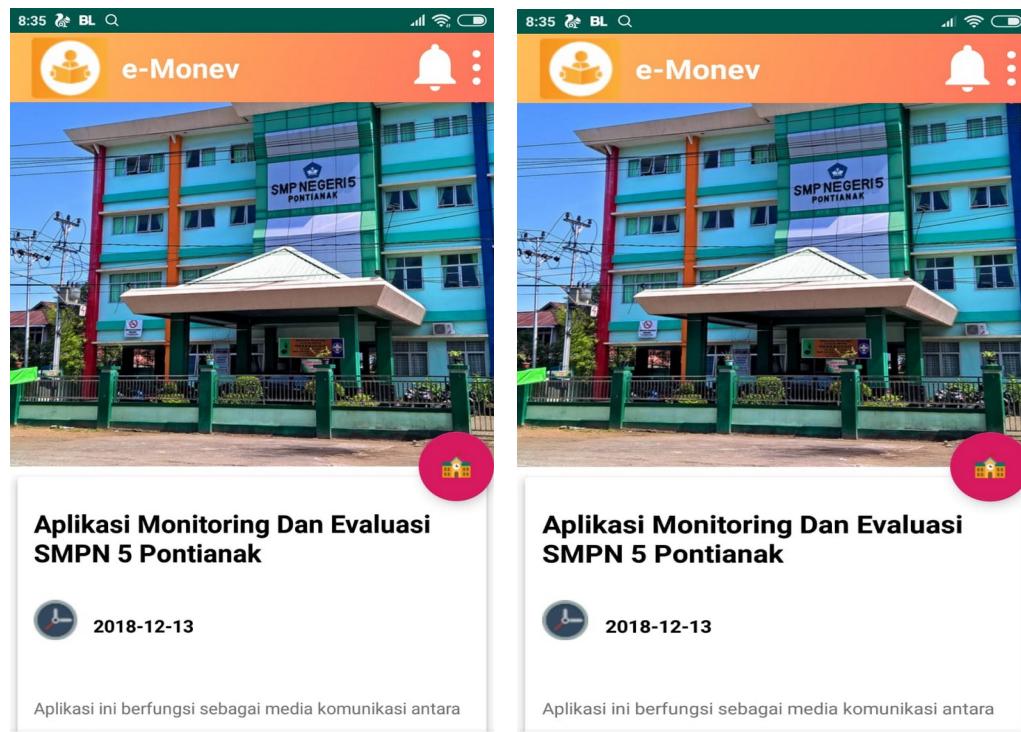
Pada tampilan halaman *login* android terdapat dua *tipe user* pengguna, yaitu orang tua siswa dan guru. Antarmuka halaman *login* dapat dilihat pada Gambar 4.13.



Gambar 4.13 Antarmuka Halaman Login Android

IV.1.14 Tampilan Antarmuka Halaman Beranda Guru Dan Siswa

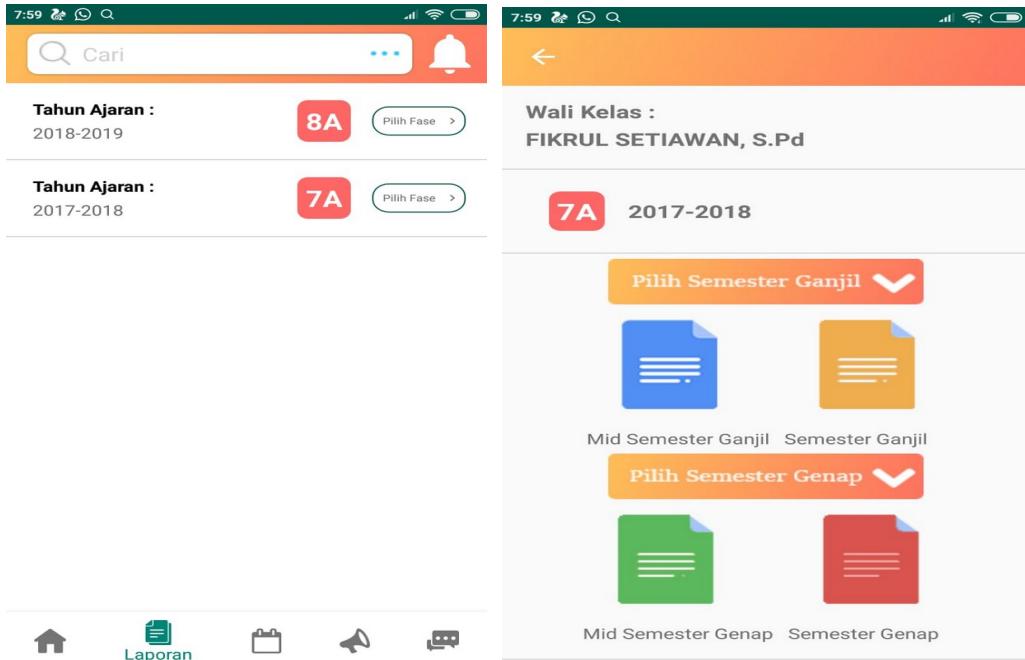
Pada tampilan halaman beranda guru terdapat informasi tentang aplikasi e-monev terdapat tiga menu di halaman guru yaitu beranda, laporan siswa, dan chat. Sedangkan pada halaman beranda siswa terdapat lima menu di halaman siswa yaitu beranda, laporan siswa, absen perhari, pengumuman dan chat. Antarmuka halaman beranda guru dapat dilihat pada Gambar 4.14.



Gambar 4.14 Antarmuka Halaman Beranda Guru Dan Siswa

IV.1.15 Tampilan Antarmuka Halaman List Tahun Ajaran

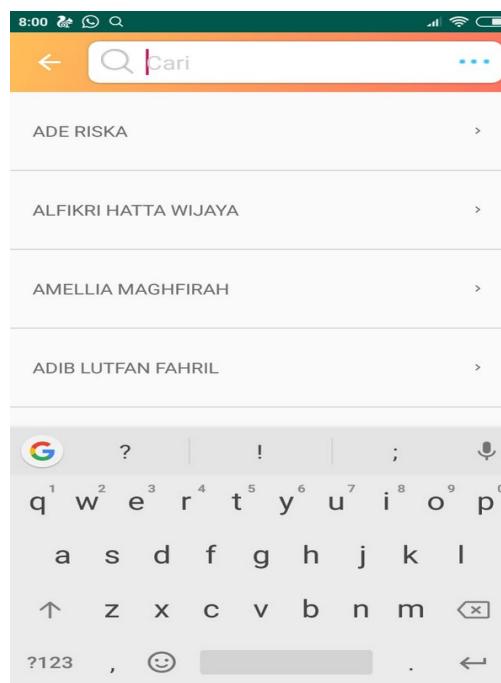
Pada tampilan halaman list tahun ajaran terdapat list yang menampilkan kelas dan tahun ajaran pada siswa dan guru. Pada saat list tahun ajaran di tekan, halaman fase tampil terdapat empat pilihan fase yaitu mid semester ganjil, semester ganjil, mid semester genap, semester genap. Antarmuka halaman list tahun ajaran dapat dilihat pada Gambar 4.15.



Gambar 4.15 Antarmuka Halaman List Tahun Ajaran

IV.1.16 Tampilan Antarmuka Halaman List Siswa

Pada tampilan halaman list siswa terdapat list siswa pada kelas dan tahun ajaran saat memilih list tahun ajaran. Antarmuka halaman list siswa dapat dilihat pada Gambar 4.16.



Gambar 4.16 Antarmuka Halaman List Siswa

IV.1.17 Tampilan Antarmuka Halaman Laporan Siswa

Pada tampilan halaman laporan siswa terdapat list nilai siswa dan absen perhari siswa. Antarmuka halaman laporan siswa dapat dilihat pada Gambar 4.17.



Gambar 4.17 Antarmuka Halaman Laporan Siswa

IV.1.18 Tampilan Antarmuka Halaman Absen Perhari

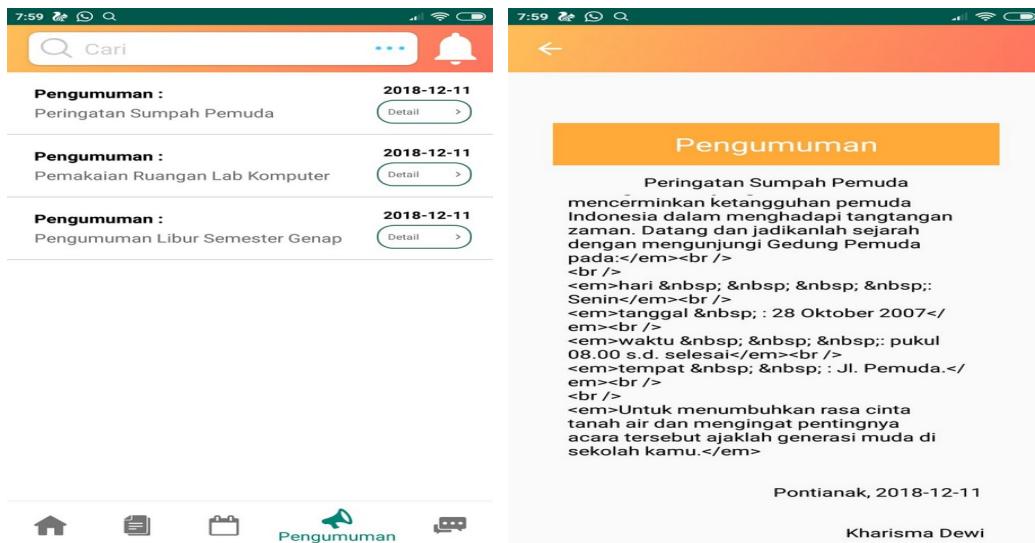
Pada tampilan halaman absen perhari terdapat absen terbaru dan riwayat list absen, pada list absen terdapat huruf H hadir, I izin, A alpa, S sakit. Antarmuka halaman absen perhari dapat dilihat pada Gambar 4.18.



Gambar 4.18 Antarmuka Halaman Absen Perhari

IV.1.19 Tampilan Antarmuka Halaman Pengumuman

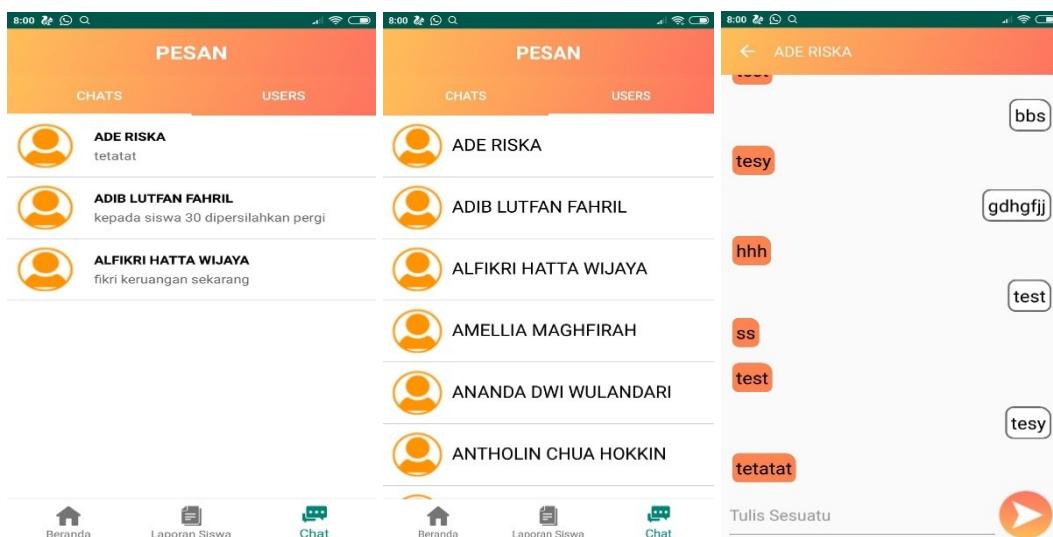
Pada tampilan halaman pengumuman terdapat list pengumuman. Pada saat list pengumuman ditekan tampil detail dari pengumuman, halaman detail pengumuman terdapat isi, nama pembuat pengumuman dan tanggal pengumuman dibuat. Antarmuka halaman list pengumuman dapat dilihat pada Gambar 4.19.



Gambar 4.19 Antarmuka Halaman Pengumuman

IV.1.20 Tampilan Antarmuka Halaman Chat

Pada tampilan halaman chat terdapat dua menu yaitu menu list chat terakhir dan menu list dari siswa maupun guru yang bisa digeser kenan dan ke kiri. Pada saat kedua menu di tekan tampil halaman list chat antara orang tua dan guru.. Antarmuka halaman chat dapat dilihat pada Gambar 4.20.



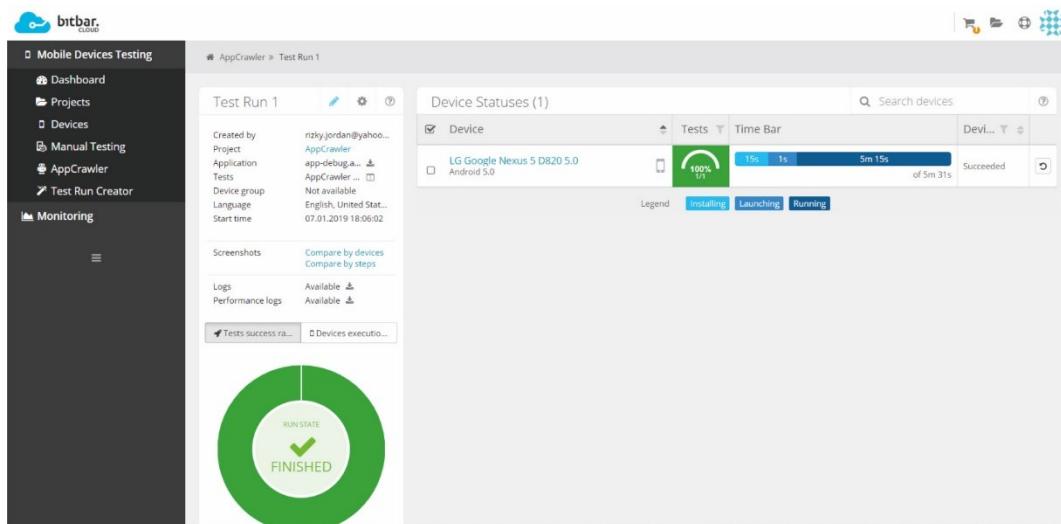
Gambar 4.20 Antarmuka Halaman Chat

IV.2 Hasil Pengujian

Aplikasi monitoring dan evaluasi aktivitas anak menggunakan dua pengujian yang pertama pengujian BlackBox dan yang kedua pengujian Kuesioner.

IV.2.1 Hasil Pengujian Black Box

Pengujian dilakukan untuk mengetes fungsionalitas, performa, dan memori yang terdapat dalam penelitian yang dilakukan. Teknik pengujian blackbox yang digunakan yaitu *performance testing* dan *comparison testing*. Alat yang digunakan adalah Bitbar Testing, berikut adalah tahapan pengujian yang telah dilakukan.



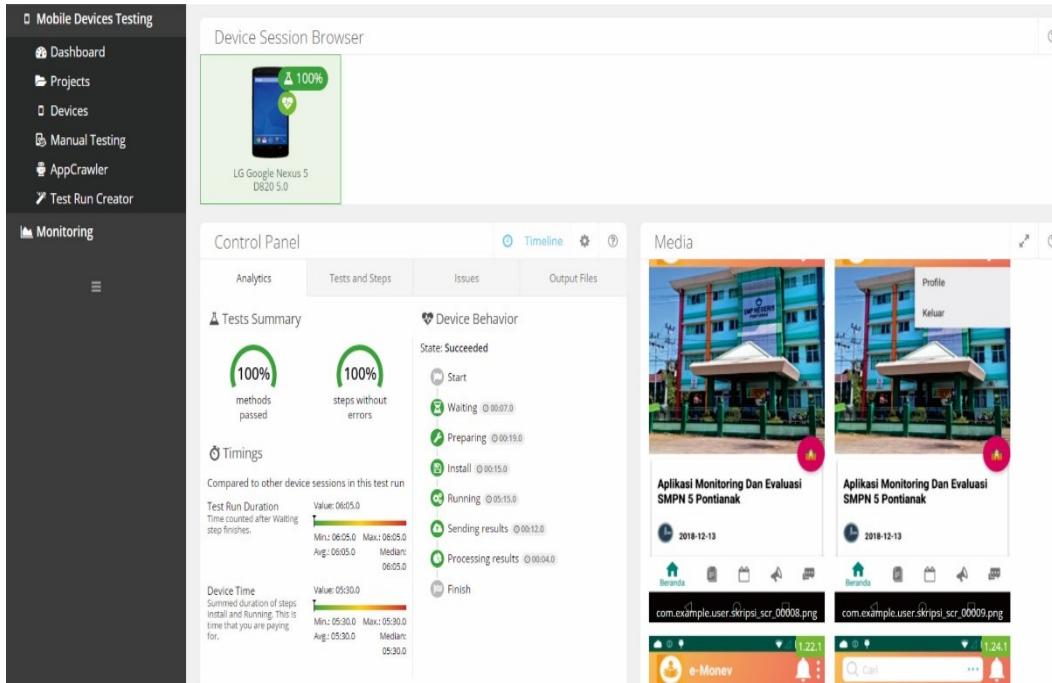
Gambar 4.21 Pengujian BlackBox Berhasil

Pengujian dilakukan dalam dua device yaitu LG Google Nexus 5 6.0, dan LG Google Nexus 5 D820. Bagian yang diujikan adalah fungsional, performa, dan memori, berikut adalah penjabaran hasil pengujinya.

4.2.1.1 Pengujian pada LG Google Nexus 5 D820

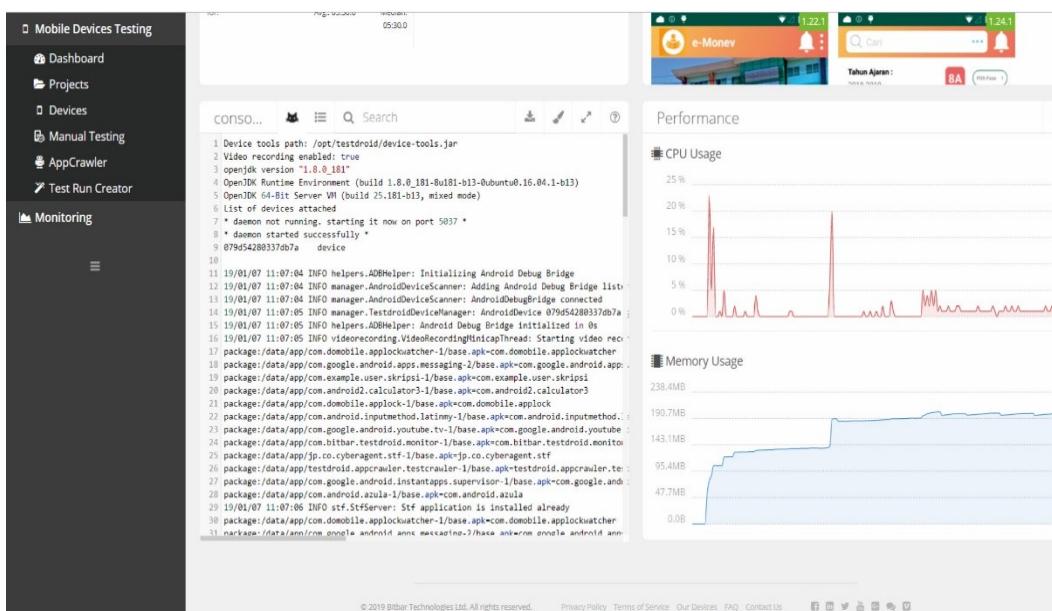
Pengujian telah dilakukan pada perangkat yaitu LG Google Nexus 5 D820, berikut adalah hasil dari pengujian yang telah dilakukan melalui Bitbar Testing. Didapatkan hasil pengujian di halaman awal ialah analisis dari pengujian metode yaitu 100%, menyatakan seluruh metode di dalam aplikasi dapat berjalan semua. Selanjutnya adalah pengujian tahapan yang mengalami *errors* yaitu 100%,

menyatakan seluruh tahapan yang dilakukan aplikasi berjalan tanpa gangguan. Berikut adalah hasil Gambar 4.22.



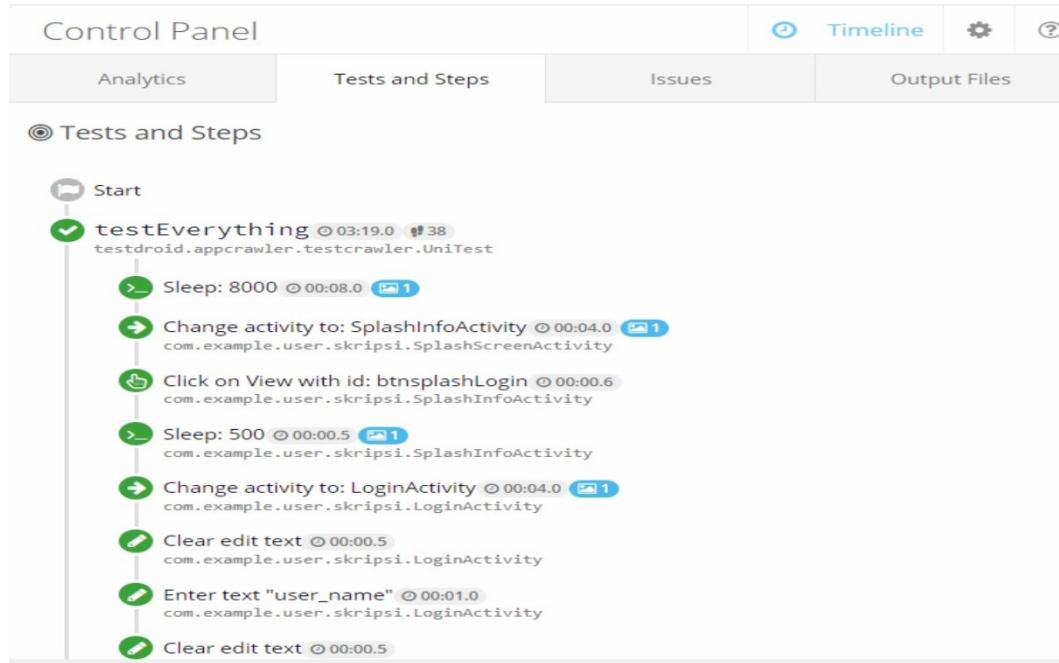
Gambar 4.22 Hasil pengujian analisis LG Google Nexus 5 D820

Pengujian performa atau penggunaan CPU yaitu dalam kisaran 3.2% dan penggunaan memori yaitu dalam kisaran 188MB. Penggunaan memori yang dipakai pada aplikasi dalam perangkat, dapat dilihat dalam Gambar 4.23.



Gambar 4.23 Hasil pengujian performa LG Google Nexus 5 D820

Langkah pengujian dan dokumentasi dari pengujian dapat dilihat dalam gambar 4.24.



Gambar 4.24 Hasil pengujian langkah LG Google Nexus 5 D820

Hasil pengujian memperlihatkan tidak ada permasalahan besar, 2 hal yang bisa dipertimbangkan untuk dirubah dalam perangkat LG Google Nexus 5 D820. Untuk melihat permasalahan yang terdapat dalam aplikasi dapat dilihat dalam Gambar 4.25.

Control Panel

Timeline | Settings | Help

- Analytics**
- Tests and Steps**
- Issues**
- Output Files**

Major Issues
No items

Worth Checking
No items

Checked Stuff that are OK

- No methods have failed
- No methods have been skipped
- No methods have been not available
- No method steps have errors
- No device session steps have failed
- No device session steps have been excluded
- Test Run Duration is 1 times as median
- Device Time is 1 times as median
- CPU usage check for step Install has been skipped
- Average CPU usage during Running step was 1.1%

Gambar 4.25 Hasil pengujian permasalahan LG Google Nexus 5 D820

4.2.1.2 Pengujian pada LG Google Nexus 5 6.0

Pengujian telah dilakukan pada perangkat yaitu LG Google Nexus 5 6.0, berikut adalah hasil dari pengujian yang telah dilakukan melalui Bitbar Testing. Didapatkan hasil pengujian di halaman awal ialah analisis dari pengujian metode yaitu 100%, menyatakan seluruh metode di dalam aplikasi dapat berjalan semua. Selanjutnya adalah pengujian tahapan yang mengalami *errors* yaitu 100%, menyatakan seluruh tahapan berjalan tanpa gangguan. Berikut adalah hasil Gambar 4.26.

Mobile Devices Testing

- Dashboard
- Projects
- Devices
- Manual Testing
- AppCrawler
- Test Run Creator
- Monitoring

Device Session Browser

Control Panel

- Analytics
- Tests and Steps
- Issues
- Output Files

Tests Summary

- 100% methods passed
- 100% steps without errors

Timings

Compared to other device sessions in this test run

Test Run Duration	Value: 05:46.0	Time counted after Waiting step finished.	
Min: 05:46.0	Max: 05:46.0	Avg: 05:46.0	Median: 05:46.0

Device Time

Summary of duration of steps Install and Running. This is time that you are paying for.

Value: 05:07.0			
Min: 05:07.0	Max: 05:07.0	Avg: 05:07.0	Median: 05:07.0

Device Behavior

State: Succeeded

- Start
- Waiting (0:00:08)
- Preparing (0:00:24.0)
- Install (0:00:12.0)
- Running (0:00:55.0)
- Sending results (0:00:11.0)
- Processing results (0:00:04.0)
- Finish

Media

Aplikasi Monitoring Dan Evaluasi SMPN 5 Pontianak

Tahun Ajaran : 2017-2018

2018-12-13

com.example.user.skripsi.scr_00010.png

com.example.user.skripsi.scr_00011.png

PESAN

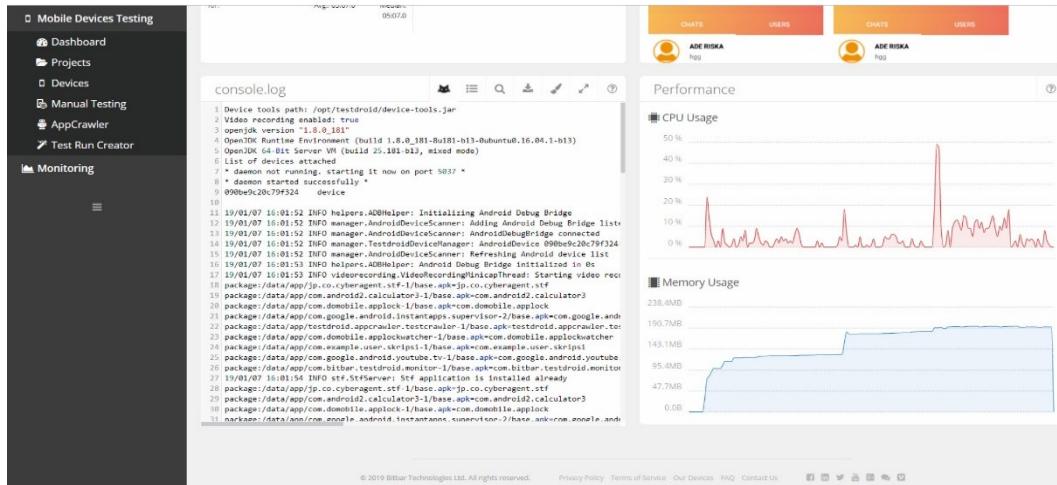
CHAT

USERS

ADE RISKA

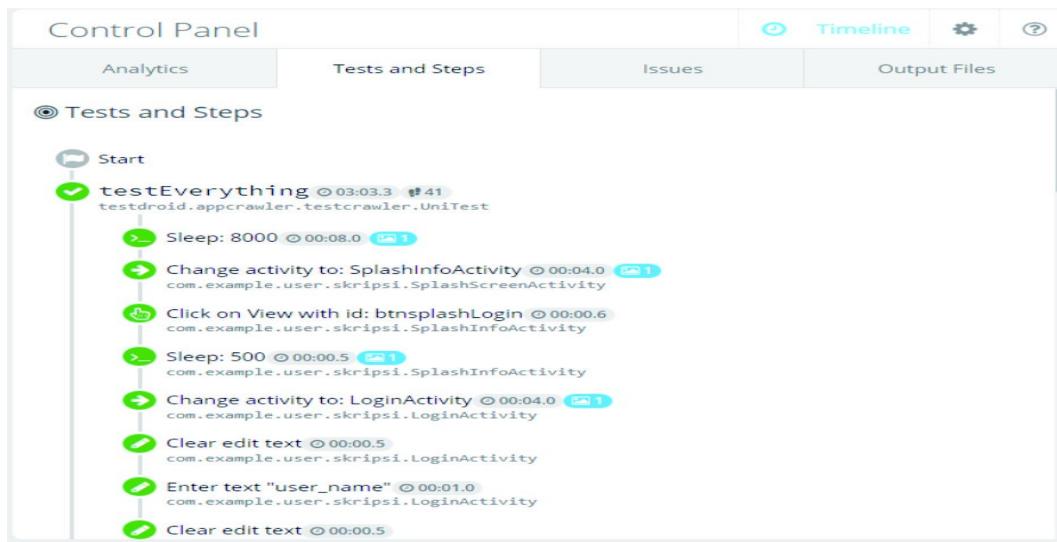
Gambar 4.26 Hasil pengujian analisis LG Google Nexus 5 6.0

Pengujian performa atau penggunaan CPU yaitu dalam kisaran 14.6% dan penggunaan memori yaitu dalam kisaran 189MB. Penggunaan memori yang dipakai lumayan banyak dalam perangkat, dapat dilihat dalam Gambar 4.27.



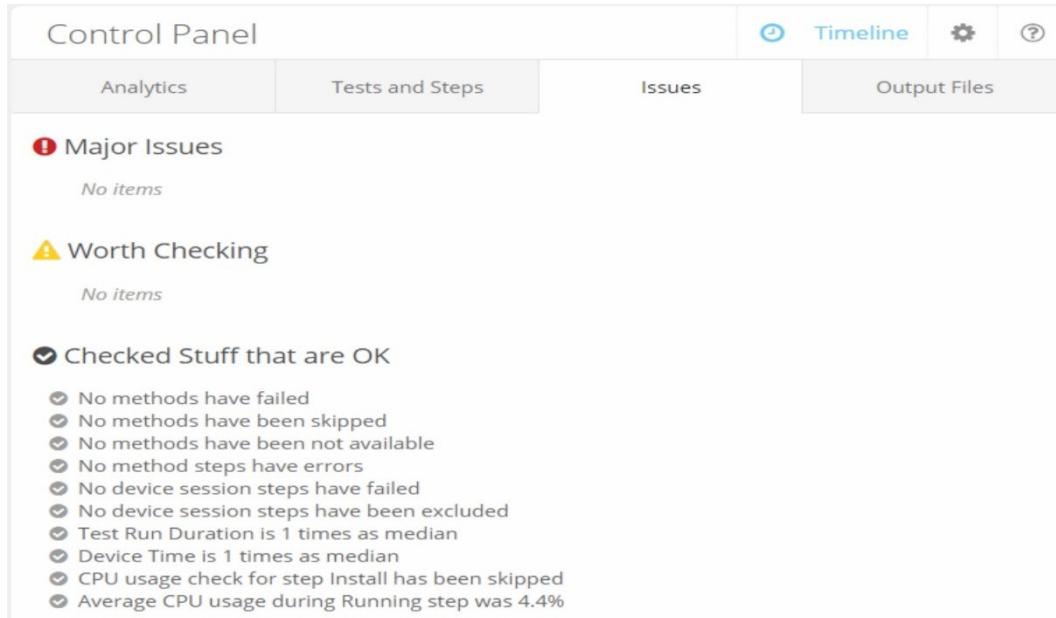
Gambar 4.27 Hasil pengujian performa LG Google Nexus 5 6.0

Langkah pengujian dan dokumentasi dari pengujian dapat dilihat dalam gambar 4.28.



Gambar 4.28 Hasil pengujian langkah LG Google Nexus 5 6.0

Hasil pengujian memperlihatkan tidak ada permasalahan besar dalam perangkat LG Google Nexus 5 6.0. Untuk melihat permasalahan yang terdapat dalam aplikasi dapat dilihat dalam Gambar 4.29.



Gambar 4.29 Hasil pengujian permasalahan LG Google Nexus 5 6.0

IV.2.2 Hasil Pengujian Kuesioner

IV.2.2.1 Latar Belakang Peserta Kuesioner

Dalam pengujian beta, dilakukan pengujian dengan menggunakan 30 responden dari tiga pihak yang akan berpartisipasi dalam pengisian kuesioner aplikasi e-monev. orang yang memenuhi kriteria sebagai berikut:

1. Dinas pendidikan diantaranya kepala bidang umum, staff IT di lingkungan dinas pendidikan.
2. Pihak sekolah diantaranya kepala sekolah, guru, admin sekolah dan staff TU.
3. Orang Tua Siswa.

Berikut adalah penjelasan tentang tiga pihak yang akan oleh peserta kuesioner yaitu dari dinas pendidikan berjumlah 6 orang, pihak sekolah 12 orang dan dari orang tua 12 orang, total responden yang berpartisipasi dalam pengisian kuesioner berjumlah 30 orang.

IV.2.2.2 Perhitungan Skala Likert

Hasil kuesioner perhitungan skala *likert* sesuai dengan aspek/pertanyaan masing-masing responden ditunjukkan pada Tabel 4.2.

Tabel 4.1 Hasil kuesioner aspek *likert*

No.	Aspek / Pertanyaan	Nilai Pengujian					Percentase Likert
		TB	KB	CB	B	SB	
Aspek Rekayasa Perangkat Lunak							
1.	Kemudahan menjalankan aplikasi	0	0	0	13	17	91,33
2.	Kelancaran menjalankan aplikasi	0	0	0	16	14	89,33
3.	Kenyamanan dalam menggunakan aplikasi secara keseluruhan	0	0	1	12	17	90,67
Aspek Fungsionalitas							
4.	Kemudahan menjalankan tombol menu pada aplikasi	0	0	1	10	19	92
5.	Tingkat kesesuaian menampilkan nilai siswa, absen perhari dan pengumuman yang berkaitan dengan sekolah, pada aplikasi android	0	0	2	15	13	87,33
6.	Tingkat kesesuaian aplikasi dalam menampilkan dan mengirim chat antara orang tua dan guru	0	0	0	18	12	88
7.	Kemudahan dalam mendapatkan informasi siswa seperti nilai siswa, absen dan pengumuman di aplikasi android	0	0	0	15	15	90
Aspek Komunikasi Visual							
8.	Kemudahan memahami informasi yang ditampilkan pada aplikasi	0	0	0	12	18	92
9.	Tampilan antarmuka aplikasi	0	0	0	13	17	91,33

No.	Aspek / Pertanyaan	Nilai Pengujian					Percentase Likert
		TB	KB	CB	B	SB	
10.	Tingkat kemudahan melihat hasil yang ditampilkan aplikasi	0	0	0	12	18	92
Total Percentase (%)						90,4 %	

Nilai persentase skala likert dihitung dengan rumus berikut ini:

$$L = \frac{(SB \times 5) + (B \times 4) + (CB \times 3) + (KB \times 2) + (TB \times 1)}{NB} \times 100$$

Hasil persentase menggunakan skala likert sebagai berikut:

Pertanyaan 1 :

$$L = \frac{(17 \times 5) + (13 \times 4) + (0 \times 3) + (0 \times 2) + (0 \times 1)}{5 \times 30} \times 100 = 91,33\%$$

Pertanyaan 2 :

$$L = \frac{(14 \times 5) + (16 \times 4) + (0 \times 3) + (0 \times 2) + (0 \times 1)}{5 \times 30} \times 100 = 89,33\%$$

Pertanyaan 3 :

$$L = \frac{(17 \times 5) + (12 \times 4) + (1 \times 3) + (0 \times 2) + (0 \times 1)}{5 \times 30} \times 100 = 90,67\%$$

Pertanyaan 4:

$$L = \frac{(19 \times 5) + (10 \times 4) + (1 \times 3) + (0 \times 2) + (0 \times 1)}{5 \times 30} \times 100 = 92\%$$

Pertanyaan 5 :

$$L = \frac{(13 \times 5) + (15 \times 4) + (2 \times 3) + (0 \times 2) + (0 \times 1)}{5 \times 30} \times 100 = 87,33\%$$

Pertanyaan 6 :

$$L = \frac{(12 \times 5) + (18 \times 4) + (0 \times 3) + (0 \times 2) + (0 \times 1)}{5 \times 30} \times 100 = 88\%$$

Pertanyaan 7 :

$$L = \frac{(15 \times 5) + (15 \times 4) + (0 \times 3) + (0 \times 2) + (0 \times 1)}{5 \times 30} \times 100 = 90\%$$

Pertanyaan 8 :

$$L = \frac{(18 \times 5) + (12 \times 4) + (0 \times 3) + (0 \times 2) + (0 \times 1)}{5 \times 30} \times 100 = 92\%$$

Pertanyaan 9 :

$$L = \frac{(17 \times 5) + (13 \times 4) + (0 \times 3) + (0 \times 2) + (0 \times 1)}{5 \times 30} \times 100 = 91,33\%$$

Pertanyaan 10 :

$$L = \frac{(18 \times 5) + (12 \times 4) + (0 \times 3) + (0 \times 2) + (0 \times 1)}{5 \times 30} \times 100 = 92\%$$

$$\begin{aligned} \text{Total Persentase} &= \underline{\underline{91,33 + 89,33 + 90,67 + 92 + 87,33 + 88 + 90 + 92 + 91,33}} \\ &\quad \underline{\underline{+ 92}} \\ &= 90,4\% \end{aligned}$$

Dari hasil kuesioner yang telah dilaksanakan terhadap 30 responden didapatkan nilai dari perhitungan skala likert dari total 10 pertanyaan yang diberikan kepada pihak dinas pendidikan, pihak sekolah maupun orang tua adalah 90,4%.

Nilai interval persentase pada kuesioner ini yaitu:

- $X \leq 19,99\%$ dikategorikan tidak baik
- $20\% \geq X \leq 39,99\%$ dikategorikan kurang baik
- $40\% \geq X \leq 59,99\%$ dikategorikan cukup baik
- $60\% \geq X \leq 79,99\%$ dikategorikan baik
- $X \geq 80\%$ dikategorikan sangat baik

Kesimpulan hasil akhir dari aplikasi e-monev, dari nilai persentase interval mendapatkan hasil aplikasi e-monev dikategorikan sangat baik.

IV.3 Analisis Hasil Perancangan dan Pengujian

Berikut ini adalah hasil analisis hasil perancangan dan pengujian perangkat lunak aplikasi permainan dalam penelitian:

1. Hasil pengujian fungsional yang dilakukan oleh sistem oleh perusahaan Bitbar menunjukkan semua metode yang ada dalam aplikasi dapat berjalan semua dengan lancar.
2. Hasil pengujian memori dan *cpu usage* yang dilakukan oleh sistem dari perusahaan Bitbar menunjukkan aplikasi e-monev menggunakan banyak memori untuk menjalankan aplikasi.
3. Berdasarkan hasil kuesioner, secara umum responden menerima baik aplikasi e-monev dalam penelitian yang dilakukan.
4. Berdasarkan hasil kuesioner, secara umum responden dapat menjalankan aplikasi e-monev dengan lancar.

5. Berdasarkan hasil kuesioner, secara umum responden menyukai tampilan antarmuka, aplikasi e-monev.
6. Berdasarkan hasil kuesioner, mendapatkan kesimpulan bahwa e-monev dapat membantu memonitor aktivitas anak disekolah dan dinilai berhasil.

BAB V

KESIMPULAN DAN SARAN

V.1 Kesimpulan

Berdasarkan hasil analisis dan pengujian terhadap penerapan aplikasi e-Monev sebagai media monitoring dan evaluasi antara orang tua dan guru di SMPN 5 Pontianak, maka dapat ditarik kesimpulan sebagai berikut:

1. Aplikasi e-Monev telah dirancang dan dibuat dengan baik serta berjalan dengan lancar dalam perangkat Android dengan menggunakan Bitbar Testing. Melalui kuesioner aplikasi juga telah diterima dengan baik oleh Dinas Pendidikan Kota Pontianak, SMPN 5 Pontianak, dan orang tua siswa dengan menggunakan Skala *Likert*.
2. Aplikasi e-Monev yang dibangun sebagai media komunikasi antara orang tua dan guru berjalan dengan baik berdasarkan pengujian automasi oleh sistem Bitbar Testing yaitu 100% metode berjalan dengan baik tanpa error dalam perangkat Android sistem operasi 4.2 (Jelly Bean) hingga versi 8.0 (Oreo) dengan berbagai ukuran layar handphone.
3. Penerapan aplikasi e-Monev dinilai berhasil dalam proses monitoring, evaluasi dan sebagai media komunikasi antara orang tua dan guru, ditunjukkan dengan berdsarkan pada pengisian menggunakan Skala Likert dimana diberikan kepada 30 responden untuk mereview aplikasi. Total persentase yang didapat dari 10 pertanyaan adalah 90,6% yang artinya diterima baik oleh pihak dinas pendidikan, pihak sekolah maupun orang tua.

V.2 Saran

Adapun beberapa hal yang perlu ditambahkan dalam aplikasi e-Monev sebagai media monitoring, evaluasi dan komunikasi antara oranag tua dan guru SMPN 5 Pontianak, adalah sebagai berikut:

1. Mengurangi atau meminimalisir penggunaan cpu dan memori pada aplikasi e-monev, agar aplikasi menjadi lebih ringan dan optimal di perangkat Android smartphone.
2. Penambahan fitur scan pengumuman yang memudahkan pihak sekolah dalam memberikan informasi atau pengumuman terbaru.