

Sistem Pengolahan Data dan Seleksi Penerima Bantuan Beasiswa Siswa Miskin (BSM) Berbasis Web dan Android, (Studi Kasus : Madrasah Ibtidaiyah Miftahul Ulum, Jember)

M. Habibullah Arief¹, Ir. Irawan Thamrin, M.T.², Reza Budiawan, S.T., M.T.³

¹M. Habibullah Arief, ²Ir. Irawan Thamrin, M.T., ³Reza Budiawan, S.T., M.T.
¹abiarief@yahoo.com, ²irawanthamrin@gmail.com, ³rezabudiawan@gmail.com

Abstrak

(5) Bantuan Beasiswa Siswa Miskin (BSM) adalah salah satu program pemberian beasiswa kepada siswa miskin yang bertujuan untuk meningkatkan mutu pendidikan. Yang diberikan oleh pemerintah melalui Kementerian Agama. Salah satu sekolah penerima adalah Madrasah Ibtidaiyah Miftahul Ulum Jember. Namun Madrasah Ibtidaiyah Miftahul Ulum Jember saat ini mengalami kesulitan pada saat menentukan siswa dan kuota yang akan mendapatkan bantuan tersebut.

Berdasarkan masalah diatas, dibutuhkan sebuah aplikasi yang mempermudah pihak Madrasah Ibtidaiyah Miftahul Ulum Jember untuk menentukan siswa dan kuota penerima Bantuan Beasiswa Siswa Miskin. Konten yang dibuat pada aplikasi ini menggunakan metode SDLC (*Software Development Life Cycle*).

Aplikasi ini dikembangkan dengan metode *waterfall*. Aplikasi Pengolahan Data dan Seleksi Penerima Bantuan Beasiswa Siswa Miskin (BSM) Berbasis Web dan Android ini memiliki fungsionalitas utama yang memudahkan pihak Madrasah Ibtidaiyah Miftahul Ulum Jember untuk menyeleksi siswa dan menentukan kuota penerima Bantuan Beasiswa Siswa Miskin (BSM).

Kata kunci: BSM, SDLC, *waterfall*, *web*, *android*

Abstract

Bantuan Beasiswa Siswa Miskin (BSM) is a program which provide scholarships for poor students. The purpose of this program is to improve the quality of education. Provided by the government through the Ministry of Religious Affairs. One Madrasah Ibtidaiyah Miftahul Ulum is one school which receive this program. However Madrasah Ibtidaiyah Miftahul Ulum Jember currently experiencing difficulties when determining quota students and those who will get help.

An application is needed to facilitates the Madrasah Ibtidaiyah Miftahul Ulum Jember to determine students and quotas of Bantuan Beasiswa Siswa Miskin (BSM) recipient. Content created on this application using SDLC (Software Development Life Cycle).

This application was developed by the waterfall method. This application has the main functionality that facilitates the Madrasah Ibtidaiyah Miftahul Ulum Jember for selecting students and determine quotas Bantuan Beasiswa Siswa Miskin (BSM) recipient.

Keywords: BSM, SDLC, *waterfall*, *web*, *android*

1. Pendahuluan

1.1 Latar Belakang

Madrasah Ibtidaiyah Miftahul Ulum adalah sebuah lembaga pendidikan islam yang setara dengan tingkatan Sekolah Dasar (SD), yang berada di naungan Kementerian Agama. Sebagaimana diketahui, madrasah pada umumnya berada di lingkungan masyarakat yang tidak mampu dan siswanya adalah siswa yang secara ekonomis berasal dari keluarga tidak mampu atau miskin. Permasalahan besar yang dihadapi oleh madrasah adalah realitas rendahnya mutu dan kualitas pendidikan madrasah. Berbagai usaha tentu telah dilakukan untuk meningkatkan mutu dan kualitas pendidikan madrasah. Namun demikian, berbagai indikator peningkatan mutu dan kualitas pendidikan madrasah belum menunjukkan peningkatan signifikan. Sebagian madrasah, terutama di kota-kota mungkin menunjukkan adanya peningkatan mutu dan kualitas pendidikan, namun sebagian madrasah lain terutama di pelosok daerah keadaannya masih memprihatinkan.

Oleh karena itu pemerintah melalui Kementerian Agama RI melaksanakan program Bantuan Beasiswa Siswa Miskin (BSM). Salah satu tujuan utama bantuan beasiswa siswa miskin adalah untuk menurunkan angka *drop out* siswa madrasah. Di samping itu, untuk memberikan motivasi siswa agar dapat meningkatkan prestasi pendidikannya, sehingga dapat menghilangkan kesenjangan sosial antara yang kaya dan yang miskin. Perlu diketahui bahwa secara psikologis anak miskin merasa dirinya "tak berdaya" dan kurang percaya diri, apabila tidak memiliki prestasi atau putus sekolah.

Namun Madrasah Ibtidaiyah Miftahul Ulum saat ini masih mengalami kesulitan pada saat menentukan siswa dan kuota yang akan mendapatkan bantuan tersebut. Dan hasilnya ada beberapa penyaluran bantuan tersebut yang tidak tepat sasaran. Dikarenakan selama ini tahap penyeleksian hanya dilakukan dengan pendataan yang sederhana, persyaratan yang diberikan kepada siswa hanya mempunyai surat keterangan miskin dari

Kantor Desa. Dan pemberian surat keterangan miskin dari Kantor Desa tidak semua tepat sasaran. Selain itu, terdapat kecenderungan dari pihak masyarakat untuk berpura-pura miskin agar mendapatkan bantuan tersebut.

1.2 Definisi Operasional

Sistem pengolahan data dan seleksi penerima Bantuan Beasiswa Siswa Miskin (BSM) berbasis *web* dan *android* merupakan sebuah sistem informasi yang membantu Madrasah Ibtidaiyah Miftahul Ulum dalam proses pengolahan data dan seleksi penerima bantuan beasiswa siswa miskin. Sistem ini menjembatani antara pihak madrasah dengan siswa, agar dalam proses penyaluran bantuan beasiswa siswa miskin dapat tepat sasaran. Sesuai yang diharapkan oleh pemerintah agar terjadi peningkatan kualitas pendidikan di kalangan madrasah.

Sistem informasi ini akan digunakan oleh 3 user, yaitu : kepala sekolah, petugas (guru yang terpilih) dan admin. Kepala sekolah mempunyai hak akses untuk *approve* data calon penerima BSM menjadi penerima BSM, menentukan kuota penerima bantuan beasiswa siswa miskin, menyeleksi data calon penerima bantuan beasiswa siswa miskin, mengelola data penerima bantuan beasiswa siswa miskin dan melihat lokasi penerima bantuan beasiswa siswa miskin. Petugas mempunyai hak akses melihat data siswa, melihat data persyaratan, mengelola data calon penerima bantuan beasiswa siswa miskin beserta lokasi dan persyaratan yang terpenuhi dan melihat data penerima bantuan beasiswa siswa miskin. Sedangkan admin mempunyai hak akses untuk mengolah data petugas, mengelola data persyaratan penerima bantuan beasiswa siswa miskin, mengelola data kepala sekolah dan mengelola data siswa. Admin juga bisa melihat data calon penerima bantuan beasiswa siswa miskin dan penerima bantuan beasiswa siswa miskin beserta lokasinya.

Sistem informasi ini dapat diakses setiap saat selama 24 jam dan selama *server* tidak dalam keadaan *down* atau *maintenance*. Sistem informasi ini berbasis *web* dan *android*, dibangun pada jaringan internet. Jadi sistem informasi ini dapat

diakses dimana saja selama user terhubung dengan jaringan internet.

1.3 Metode Pengerjaan

Metode pengerjaan dari aplikasi ini adalah sebagai berikut :

a. Observasi

Observasi adalah suatu cara untuk mengumpulkan data dengan melakukan penelitian secara langsung datang ke pihak Komite Sekolah Madrasah Ibtidaiyah Miftahul Ulum Jember dan Kementerian Agama wilayah Jawa Barat. Hal ini bertujuan untuk mengamati dan pencatatan terhadap peristiwa yang sedang diselidiki pada objek penelitian.

b. Study Literatur

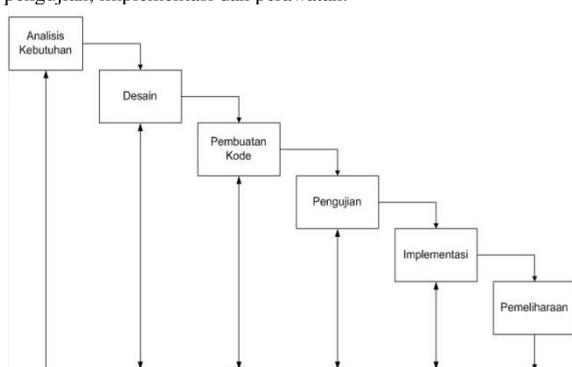
Pada tahap ini yang dilakukan adalah mencari dan membaca literatur yang yang diberikan oleh pihak Komite Sekolah Madrasah Ibtidaiyah Miftahul Ulum Jember. Dan mencari literatur tambahan yang dibutuhkan dalam pendalaman materi terhadap konsep dan teori penelitian.

c. Wawancara

Suatu cara pengumpulan data melalui tanya jawab secara langsung antara peneliti (pengumpul data) dengan responden (sumber data), dalam hal ini wawancara dilakukan dengan responden yang berhubungan langsung dengan pengendali kebijakan. Yaitu dari pihak Komite Sekolah Madrasah Ibtidaiyah Miftahul Ulum Jember dan Kementerian Agama Wilayah Jawa Barat

d. Metode Pengembangan Sistem

Pengembangan sistem secara terstruktur dengan menggunakan metode *Waterfall* pada tahapan *Software Development Life Cycle (SDLC)* meliputi : analisis, perancangan, pembuatan kode, pengujian, implementasi dan perawatan.



Gambar 1- { SEQ Gambar_1-* ARABIC }
SDLC (System Development Life Cycle)

2. Tinjauan Pustaka

2.1 Madrasah

Jika diterjemahkan ke dalam bahasa Indonesia, menurut [1] kata "madrasah" memiliki arti "sekolah" kendati pada mulanya kata "sekolah" itu sendiri bukan berasal dari bahasa Indonesia, melainkan dari bahasa asing, yaitu *school* atau *scola*. Sungguhpun secara teknis, yakni dalam proses belajar-mengajarnya secara formal, madrasah tidak berbeda dengan sekolah. Namun di Indonesia madrasah tidak lantas dipahami sebagai sekolah, melainkan diberi konotasi yang lebih spesifik lagi, yakni "sekolah agama", tempat di mana anak-anak didik memperoleh pembelajaran hal-ihwal atau seluk-beluk agama dan keagamaan (dalam hal ini agama Islam) [1].

2.2 BSM

BSM adalah singkatan dari Bantuan Beasiswa Siswa Miskin yang dikeluarkan oleh pemerintah menggunakan dana dari APBN. Dengan program Bantuan Beasiswa Siswa Miskin ini, diharapkan dapat membantu sebagian siswa yang berasal dari keluarga kurang mampu/miskin di lingkungan madrasah dalam membiayai sebagian kebutuhan pendidikannya, sehingga dapat menyelesaikan pendidikannya, bahkan dapat melanjutkan pendidikan ke jenjang berikutnya [2].

2.3 Sistem

Sistem adalah sekelompok elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan.

Menurut [3] syarat-syarat sistem adalah :

1. Sistem harus dibentuk untuk menyelesaikan suatu tujuan.
2. Untuk mendukung fungsi kepengurusan manajemen.
3. Untuk mendukung pengambilan keputusan manajemen.
4. Adanya hubungan di antara elemen sistem.

2.4 Data

Data adalah fakta atau apapun yang dapat digunakan sebagai *input* dalam menghasilkan informasi. Data terdiri dari fakta-fakta dan angka-angka yang relatif tidak berarti bagi pemakai. Contoh, data dapat berupa jumlah jam kerja pegawai dalam perusahaan. Setelah data diproses, data dapat berubah menjadi informasi [3].

2.5 CodeIgniter (CI)

CodeIgniter (CI) adalah sebuah *framework* yang digunakan untuk membuat sebuah aplikasi berbasis *web* yang disusun dengan menggunakan bahasa PHP. Di dalam CI ini, terdapat beberapa kelas yang berbentuk *library* dan *helper* yang berfungsi untuk membantu pemrograman dan mengembangkan aplikasinya [9]. Seperti halnya PHP, CI juga bersifat *free open source software* setiap orang berhak menggunakan tanpa harus dikenai lisensi. CI sangat mudah dipelajari oleh seorang *programmer web* pemula sekalipun, karena CI mempunyai file dokumentasi yang sangat memadai untuk menjelaskan setiap fungsi yang ada pada *library* dan *helper*. File dokumentasi ini disertakan secara langsung pada saat mengunduh paket *framework* CI.

Framework ini dapat digunakan untuk membuat sistem aplikasi *web* yang kompleks. CI dapat mempercepat proses pembuatan *web* karena terdapat *class* yang siap untuk dipakai dan dimodifikasi. CI juga merupakan *framework* yang berbasis *Model-View-Controller(MVC)* yaitu adanya pemisahan antara aplikasi logika dengan presentasi pada halaman *web*.

Beberapa kelebihan menggunakan *framework CodeIgniter* antara lain :

a. Performa sangat cepat

CodeIgniter sangat cepat bahkan mungkin bisa dibiling *CodeIgniter* merupakan *framework* yang paling cepat dibanding *framework* yang lain, sangat cepat dalam hal eksekusi *file*. Kecepatan ini memungkinkan CI digunakan sebagai *framework* untuk membangun aplikasi berskala *enterprise*.

b. Konfigurasi yang sangat minim (*nearly zero configuration*)

Tentu saja untuk menyesuaikan dengan *database* dan keluasaan *routing* tetap diizinkan melakukan konfigurasi dengan mengubah beberapa *file* konfigurasi, seperti "database.php" atau "autoload.php", namun untuk menggunakan *CodeIgniter* dengan *standard*, hanya perlu merubah sedikit saja *file* pada *folder config*.

c. CodeIgniter memiliki banyak komunitas

Dengan banyaknya komunitas CI ini, memudahkan untuk berinteraksi dengan yang lain, baik itu bertanya atau menunjukan teknologi terbaru.

d. Dokumentasi yang sangat lengkap

Setiap paket instalasi *CodeIgniter* sudah disertai *user guide* yang sangat bagus dan lengkap untuk dijadikan permulaan, bahasanya pun mudah dipahami.

e. Berukuran kecil

Hanya dengan sekitar 4MB, seorang pengembang sudah bisa menggunakan CI yang di dalamnya terdapat berbagai fitur handal yang bisa digunakan untuk membangun sebuah aplikasi *web* yang kompleks.

f. Sederhana

Sederhana dalam sintaks yang digunakan dan dari sisi penyusunan program.

g. Bisa langsung digunakan

Setiap orang akan bisa menggunakan CI hanya tinggal mengunduh, mengekstrak *file* dan meletakkan pada *folder* kerja *web server* (htdocs atau public html).

h. Bisa dikombinasikan dengan beberapa aplikasi dengan cara yang sangat mudah.

i. Kode program rapi

Kode program akan sangat menjadi rapi dikarenakan penggunaan sistem MVC yang mengelompokkan kode sesuai dengan fungsinya man pencariian *error* ketika muncul *bug*.

2.6 Model View Controller (MVC)

MVC (Model-View-Controller) adalah sebuah pola rancangan yang memungkinkan para pengembang memisahkan kode program mereka ke dalam tiga kelompok, yaitu :

a. *Model*

Model merupakan bagian yang didesain untuk bekerja dengan *database*. *Model* bisa juga diartikan sebagai bagian yang digunakan untuk membuat semacam *prototype* data. Bagian model ini biasanya berisi *query* yang dialamatkan pada sebuah *database*. Berbagai *query* tersebut disesuaikan dengan kebutuhan data yang akan diolah melalui *controller*, sebelum akhirnya ditampilkan menggunakan *view*. *File-file* model diletakkan di dalam *folder application/models*.

b. *View*

View adalah bagian presentasi yang berfungsi sebagai antarmuka program aplikasi dengan pengguna. *View* akan menampilkan data yang telah diolah sebelumnya melalui *controller*. Penggunaan *view* tergantung kepada *controller*, *view* yang akan digunakan dan data yang akan ditampilkan. Pada bagian *view* biasanya tidak terdapat fungsi logika sama sekali, yang ada hanyalah kode HTML dan sebagian kecil kode PHP yang digunakan untuk melakukan *parsing* data. *Fileview* dalam CI diletakkan di dalam *folder application/views*.

c. *Controller*

Controller merupakan bagian yang digunakan untuk menempatkan fungsi-fungsi logika pengolahan data yang telah diperoleh dari model (atau sumber lainnya), kemudian ditampilkan melalui (atau tanpa) *view*, inti dari program yang dibuat CI terletak pada *controller*. Melalui *controller* inilah *request* HTTP akan diolah. Nama kelas dan *method* beserta parameternya yang terdapat dalam *controller* akan digunakan untuk pengasosiasian Url sehingga pemahaman tentang *controller* akan dibutuhkan. *File controller* diletakkan dalam *folder application/controllers*.

2.7 Hypertext Preprocessor (PHP)

PHP merupakan singkatan dari *Hypertext Preprocessor*, merupakan bahasa berbentuk skript yang ditempatkan dalam *server* dan diproses di *server*. Hasilnya yang dikirimkan ke klien, tempat memakai menggunakan *browser*.

Secara khusus, PHP dirancang untuk membentuk aplikasi *web* dinamis. Artinya, PHP dapat membentuk suatu tampilan berdasarkan permintaan terkini. Pada prinsipnya PHP mempunyai fungsi yang sama dengan skrip-skrip seperti ASP, Cold Fusion, ataupun Perl.

Pada saat ini PHP cukup populer sebagai peranti pemrograman *web*, terutama di lingkungan Linux. Walaupun demikian, PHP sebenarnya juga dapat berfungsi pada *server-server* yang berbasis UNIX, Windows, dan Macintos [10].

2.8 Hypertext Markup Language (HTML)

Hypertext Markup Language (HTML) adalah bahasa yang digunakan untuk membuat suatu situs *web* atau *homepage*. Setiap dokumen dalam *web* ditulis dalam format HTML. Semua format dokumen, *hyperlink* yang dapat diklik, gambar, dokumen multimedia, *form* yang dapat diisi dan sebagainya didasarkan atas HTML [5].

2.9 MyStructure Query Language (MySQL)

MySQL merupakan *software* sistem manajemen *database (Database Management System - DBMS)* yang sangat populer di kalangan pemrogram *web*, terutama di lingkungan Linux dengan menggunakan *script* PHP dan Perl. *Software database* ini kini telah tersedia juga pada platform sistem operasi Windows (98/Me ataupun NT/2000/XP).

MySQL merupakan *database* yang paling populer digunakan untuk membangun aplikasi *web* yang menggunakan *database* sebagai sumber dan pengelola datanya. MySQL merupakan *database* yang menjanjikan sebagai alternatif pilihan *database* yang dapat digunakan untuk sistem *database* personal atau organisasi. [5]

Keistimewaan dari MySQL, yaitu:

1. *Portability*, dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac OS X Server, Solaris, Amiga, dan lain sebagainya
2. *Multuser*, dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
3. *Security*, memiliki beberapa lapisan sekuritas seperti level *subnetmask*, nama *host*, dan izin akses *user* dengan sistem perizinan yang mendetail serta *password* terenkripsi.
4. *Scalability* dan *limits*, mampu menangani *database* dalam skala besar dengan jumlah *records* lebih dari 50 juta dan 60 ribu table serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya[6].

2.10 Android

Android adalah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi *middleware* dan aplikasi. *Android* menyediakan *platform* yang terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, *Google Inc* membeli *Android Inc*, yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel atau *smartphone*, kemudian untuk mengembangkan *Android*, dibentuklah *Open Handset Alliance*, *Konsorsium* dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel Motorola, Qualcomm, T-Mobile, dan Nvidia [11].

2.11 Google Maps

Google Maps adalah layanan mapping *online* yang disediakan google. Layanan ini dapat diakses melalui situs {[HYPERLINK "http://maps.google.com"](http://maps.google.com)}. Pada situs tersebut dapat melihat informasi geografis pada hampir semua wilayah di bumi. Layanan ini interaktif, karena di dalamnya peta dapat digeser sesuai keinginan pengguna, mengubah tingkat *zoom*, serta mengubah tampilan peta. Fasilitas yang terdapat pada *Google Maps* antara lain adalah menjelajah peta, mencari lokasi tertentu dan menghitung rute dalam berkendara [12].

2.12 Browser

Browser merupakan *tool* yang berada di sisi *client*. Untuk mengakses *web*, diperlukan suatu program yang disebut *Web Browser*. *Browser* merupakan suatu program yang dirancang untuk mengambil informasi-informasi dari suatu *server computer* pada jaringan internet. Informasi ini biasanya dikemas dalam *page-page*, dimana setiap *page* bisa memiliki beberapa *link* yang menghubungkan *web page* tersebut ke sumber informasi lainnya [5].

2.13 Cascading Style Sheet (CSS)

Cascading Style Sheet (CSS) merupakan konsep sederhana yang berfungsi untuk membuat *style* atau gaya yang lebih diprioritaskan kepada bagaimana data yang ada pada HTML ditampilkan *browser*. Metode ini diperkenalkan oleh W3C sejak tahun 1996 untuk mempermudah pemisahan antara data dan *style* [7].

2.14 Entity Relationship Diagram

Entity Relationship Diagram (ERD) adalah diagram yang menggambarkan keterhubungan antar data secara konseptual. Penggambaran keterhubungan antar data ini didasarkan pada anggapan bahwa dunia nyata terdiri dari kumpulan objek yang

disebut entitas (*entity*), dan hubungan yang terjadi diantaranya disebut relasi (*relationship*) [4].

Pada dasarnya ada tiga simbol yang digunakan, yaitu:

1. *Entity* adalah obyek yang dapat dibedakan dalam dunia nyata.
2. *Relationship* merupakan relasi yang terjadi diantara dua himpunan entitas (misalnya A dan B) dalam satu *database*
3. Atribut adalah karakteristik dari *entity* atau *relationship*, yang menyediakan penjelasan detail tentang *entity* atau *relationship* tersebut

2.15 Use Case Diagram

Fungsionalitas *use case* direpresentasikan dengan aliran-aliran peristiwa. *Use case* menggambarkan bagaimana realisasi dari setiap *use case* yang ada pada model [13].

Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah orang, tapi aktor belum tentu merupakan orang.
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

2.16 Diagram Aktivitas (Activity Diagram)

Diagram aktivitas atau *activity diagram* memodelkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan oleh aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut :

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
2. Urutan atau pengelompokan tampilan dari sistem/*userinterface* di mana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan pengujian di mana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya [13].

2.17 Diagram Kelas (Class Diagram)

Diagram kelas menggambarkan struktur sistem segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan *variable-variable* yang dimiliki oleh suatu kelas. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:

1. kelas main,
2. kelas yang menangani tampilan sistem,
3. kelas yang diambil dari pendefinisian *use case*,
4. kelas yang diambil dari pendefinisian data.

Dalam mendefinisikan metode yang ada di dalam kelas perlu memperhatikan apa yang disebut dengan *cohesi* dan *coupling*. *Cohesion* adalah ukuran seberapa dekat keterkaitan instruksi di dalam sebuah metode terkait satu sama lain, sedangkan *coupling* adalah ukuran seberapa dekat keterkaitan instruksi antara metode yang satu dengan metode yang lain dalam sebuah kelas [13].

2.18 Diagram Sequence

Sequence diagram menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi sistem.

Banyaknya *sequence diagram* yang harus digambar adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada *sequence diagram* sehingga semakin banyak *use case* yang didefinisikan maka *sequence diagram* yang harus dibuat juga semakin banyak.

Penggambaran letak pesan harus berurutan, pesan yang lebih atas dari lainnya adalah pesan yang berjalan terlebih dahulu [13].

2.19 Black Box Testing

Pengujian Black Box Testing berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian *black box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian *black box* bukan merupakan alternatif dari teknik *white-box*, tetapi merupakan pendekatan komplementer yang kemungkinan besar mampu mengungkap kelas kesalahan daripada metode *white box*.

Pengujian black box berusaha menemukan kesalahan dalam kategori sebagai berikut [9] :

1. Fungsi-fungsi yang tidak benar atau salah.
2. kesalahan *interface*.
3. kesalahan dalam struktur data atau *database* eksternal.
4. kesalahan kinerja
5. Instalasi dan kesalahan terminasi.

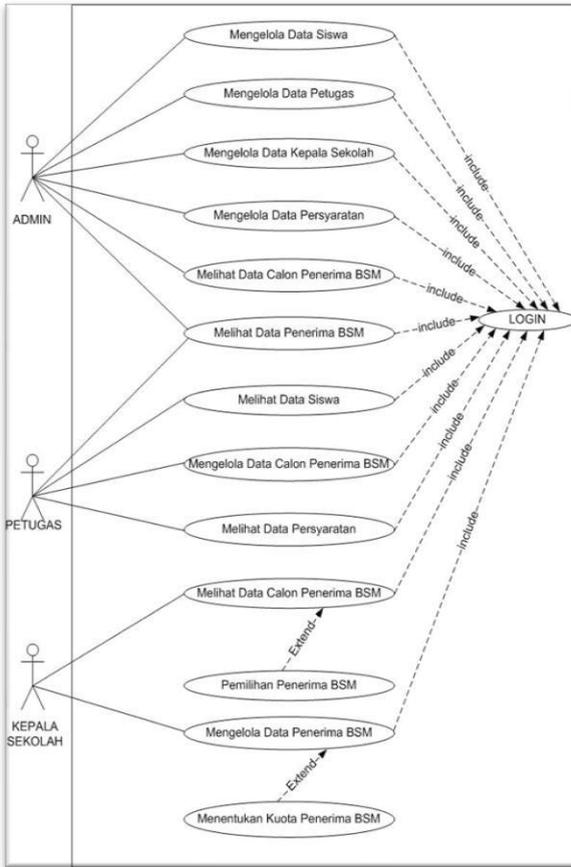
3. Analisis dan Perancangan Sistem

Sistem Pengolahan Data dan Seleksi Penerima Bantuan Beasiswa Siswa Miskin (BSM) berbasis *web* dan *android* ini mengambil studi kasus pada Madrasah Ibtidaiyah Miftahul Ulum, Jember. Sistem ini menjembatani antara pihak madrasah dengan siswa, agar dalam proses penyaluran bantuan beasiswa siswa miskin dapat tepat sasaran. Sesuai yang diharapkan oleh pemerintah agar terjadi peningkatan kualitas pendidikan di kalangan madrasah.

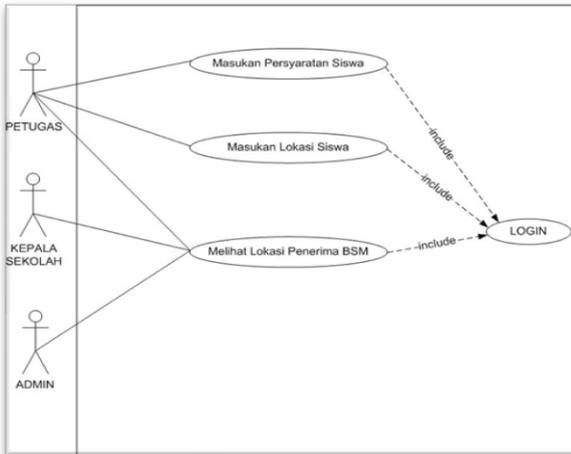
Sistem informasi ini akan digunakan oleh 3 user, yaitu : kepala sekolah, petugas (guru yang terpilih) dan admin. Kepala sekolah mempunyai hak akses untuk *approve* data calon penerima BSM menjadi penerima BSM, menentukan kuota penerima bantuan beasiswa siswa miskin, menyeleksi data calon penerima bantuan beasiswa siswa miskin, mengelola data penerima bantuan beasiswa siswa miskin dan melihat lokasi penerima bantuan beasiswa siswa miskin. Petugas mempunyai hak akses melihat data siswa, melihat data persyaratan, mengelola data calon penerima bantuan beasiswa siswa miskin beserta lokasi dan persyaratan yang terpenuhi dan melihat data penerima bantuan beasiswa siswa miskin. Sedangkan admin mempunyai hak akses untuk mengolah data petugas, mengelola data persyaratan penerima bantuan beasiswa siswa miskin, mengelola data kepala sekolah dan mengelola data siswa. Admin juga bisa melihat data calon penerima bantuan beasiswa siswa miskin dan penerima bantuan beasiswa siswa miskin beserta lokasinya.

3.1 Use Case

Adapun diagram use case dari aplikasi tersebut adalah sebagai berikut :



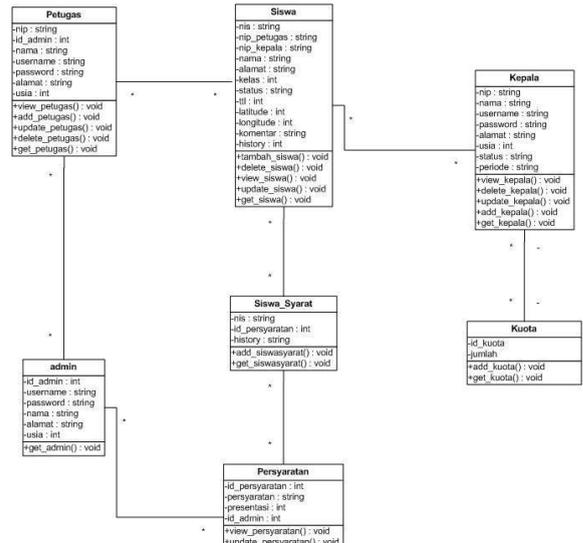
Gambar 1- { SEQ Gambar 1- * ARABIC } Diagram Use Case Untuk Aplikasi Web



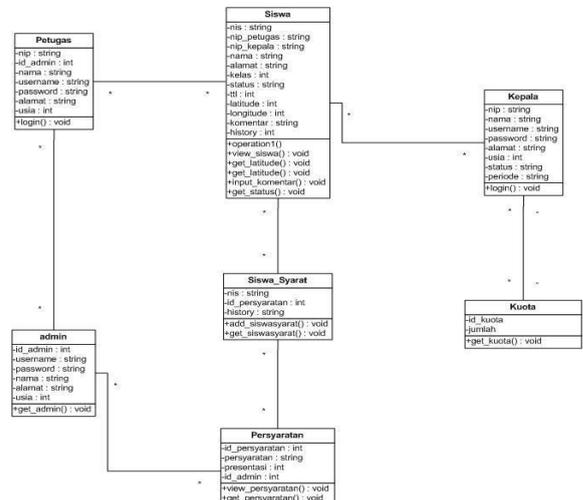
Gambar 1- { SEQ Gambar 1- * ARABIC } Diagram Use Case Untuk Aplikasi Android

3.2 Diagram Kelas

Adapun model diagram kelas dari aplikasi tersebut sebagai berikut :



Gambar 1- { SEQ Gambar 1- * ARABIC } Diagram Kelas Berbasis Web



Gambar 1- { SEQ Gambar 1- * ARABIC } Diagram Kelas Berbasis Android

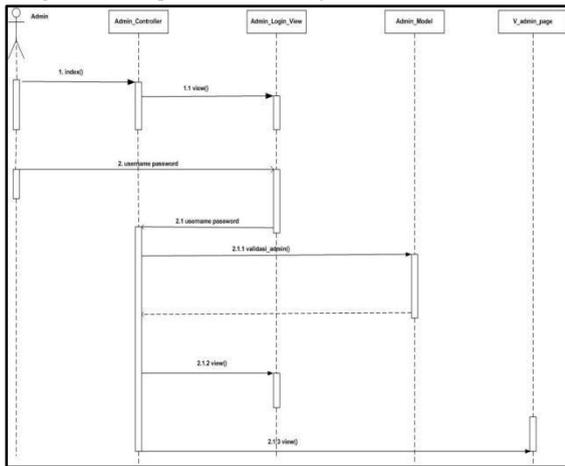
3.3 Analisis Kebutuhan Fungsionalitas

Kebutuhan fungsionalitas adalah deskripsi tentang aktifitas atau layanan yang harus disediakan oleh aplikasi yang akan dibuat. Sistem dapat melakukan fungsi sebagai berikut :

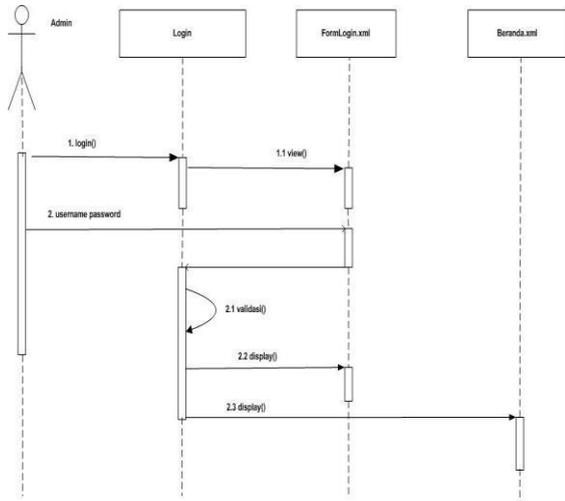
- Sistem mampu mengelola data petugas.
- Sistem mampu mengelola data siswa.
- Sistem mampu mengelola data persyaratan.
- Sistem mampu mengelola data kepala sekolah.
- Sistem mampu mengelola data calon penerima Bantuan Beasiswa Siswa Miskin (BSM).
- Sistem mampu mengelola data penerima Bantuan Beasiswa Siswa Miskin (BSM).
- Sistem mampu menyediakan *form input* lokasi siswa.
- Sistem mampu menyediakan *form input* persyaratan yang dianggap layak dipenuhi oleh calon penerima Bantuan Beasiswa Siswa Miskin (BSM).
- Sistem mampu menampilkan lokasi siswa.
- Sistem mampu menyediakan *form* kuota untuk penerima Bantuan Beasiswa Siswa Miskin (BSM).
- Sistem dapat menyediakan menu untuk *approve* data calon penerima Bantuan Beasiswa Siswa Miskin (BSM) menjadi penerima Bantuan Beasiswa Siswa Miskin (BSM).

3.4 Diagram Sequence

Berikut adalah salah satu contoh diagram sequence yang mengambil kasus pada saat admin login



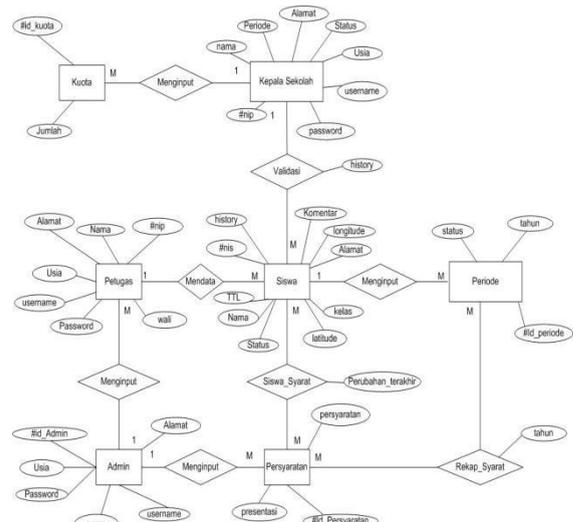
Gambar 1- { SEQ Gambar_1- * ARABIC } diagram sequence login admin web



Gambar 1- { SEQ Gambar_1- * ARABIC } Diagram sequence login admin android

3.5 Entity Relationship Diagram (ERD)

Gambar di bawah ini merupakan Entity Relationship Diagram pada aplikasi Pengolahan Data dan Seleksi Penerima Bantuan Beasiswa Siswa Miskin (BSM) Berbasis Web dan Android. ERD digunakan untuk mengembangkan inisial dari desain basis data, menyediakan suatu konsep yang bermanfaat yang dapat mengubah deskripsi informasi dari apa yang diinginkan oleh user menjadi hal yang lebih detail.



Gambar 1- { SEQ Gambar_1- * ARABIC } Entity Relationship Diagram (ERD)

3.6 Skema Relasi

1. Admin (#id_admin, nama, username, password, alamat, usia)
2. Penerimaan (#id_penerimaan, persyaratan, presentasi)
3. Siswa (#nis, nama, ttl, kelas, alamat, status, latitude, longitude, history, komentar, @nip_petugas, @nip_kepala)
4. Siswa_syarat (#id_penerimaan, #nis, history)
5. Petugas (#nip, nama, alamat, usia, username, password, @id_admin)
6. Kepala sekolah (#nip, nama, alamat, usia, username, password, status, periode)
7. kuota (#id_kuota, jumlah, @nip)
8. Rekap_syarat (#id_periode, #id_penerimaan, tahun)
9. Periode (#id_periode, @nis, tahun, status)

4. Implementasi dan Pengujian Aplikasi

4.1 Implementasi

Dalam tahap ini, hasil analisa dan perancangan mulai diimplementasikan untuk mencapai tujuan pembuatan Aplikasi Pengolahan Data dan Seleksi Penerima Bantuan Beasiswa Siswa Miskin (BSM) Berbasis Web dan Android dengan menjalankan studi kasus di Madrasah Ibtidaiyah Miftahul Ulum Jember. Implementasi yang akan dijelaskan merupakan implementasi tampilan antar muka web dan android.

Berikut adalah contoh tampilan antarmuka aplikasi berbasis web dan android :

- a. Tampilan antar muka login petugas web



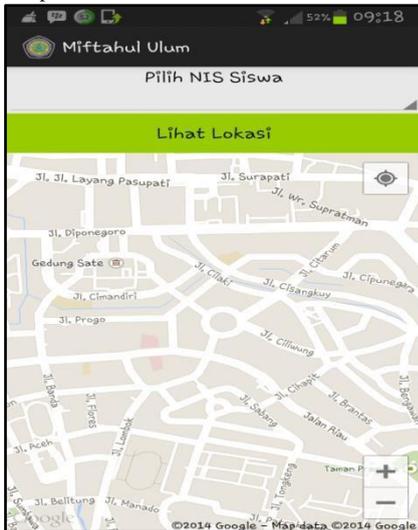
Gambar 1- { SEQ Gambar_1- * ARABIC } Antar muka login petugas web

- b. Tampilan antar muka login petugas android



Gambar 1- { SEQ Gambar_1- * ARABIC }
antar muka login petugas android

c. Tampilan antar muka petugas melihat lokasi penerima BSM pada android



Gambar 1- { SEQ Gambar_1- * ARABIC }
Antar muka Lihat Lokasi Penerima BSM petugas pada Android

d. Tampilan antar muka menu utama admin pada android



Gambar 1- { SEQ Gambar_1- * ARABIC }
Antar Muka Menu Utama Admin pada Android

4.2 Pengujian Aplikasi

Pengujian dilakukan dengan menggunakan teknik *black box testing*. Pengujian ini memungkinkan pemrogram untuk mengambil salah satu kasus uji. Kasus uji tersebut menjadi sebuah kondisi yang akan dijalankan oleh program. Dalam kasus ini diambil salah satu kasus uji yaitu *login petugas*.

Nama butir uji	Menguji form login Petugas
Tujuan	Memastikan sistem memvalidasi masukan dari pengguna secara tepat
Kondisi awal	Menampilkan form login
Skenario	[1] <i>Field username</i> dan <i>password</i> dalam keadaan kosong [2] <i>Field username</i> diisi dengan data yang salah [3] <i>Field Password</i> diisi dengan data yang salah [4] <i>Field username</i> atau <i>field password</i> diisi dengan <i>username</i> dan <i>password</i> yang sesuai dengan <i>database</i>

Hasil			
Perlakuan yang diberikan	Test Data (input)	Hasil yang diharapkan	Kesimpulan
Kedua <i>field</i> dalam keadaan kosong, kemudian klik tombol "Login"	<i>username</i> dan <i>password</i> (kosong)	Tidak terjadi apa-apa. Pengguna tidak dapat masuk ke dalam aplikasi.	Sesuai
<i>Field username</i> diisi dengan data yang salah, kemudian klik tombol "login"	<i>username</i> : 12345 <i>password</i> : habib	Menampilkan halaman <i>login</i> kembali	Sesuai
<i>Field password</i> diisi dengan data yang salah, kemudian klik tombol "login"	<i>username</i> : habib <i>password</i> : bibah	Menampilkan halaman <i>login</i> kembali	Sesuai
<i>Field username</i> atau <i>field password</i> diisi dengan <i>username</i> dan <i>password</i> yang sesuai dengan <i>database</i> , kemudian klik tombol "login"	<i>username</i> : habib <i>password</i> : 12345	kemudian petugas masuk ke halaman menu utama petugas pada aplikasi	Sesuai

Daftar Pustaka

[1] Direktorat Pendidikan Madrasah. (2012, januari). Retrieved Mei 2, 2013, from <http://madrasah.kemenag.go.id/>

[2] Kementerian Agama Republik Indonesia. (2012, Februari 28). Retrieved Mei 2, 2013, from <http://madrasah.kemenag.go.id/detail38.html>

[3] Asti Widayati, dkk. (2009). *Sistem Informasi Manajemen*. Bandung: Politeknik Telkom.

- [4] Eddy Prasetyo Nugroho, dkk. (2009). *Rekayasa Perangkat Lunak*. Bandung: Politeknik Telkom.
- [5] Suryatiningsih, dkk. (2009). *Web Programing*. Bandung: Politeknik Telkom.
- [6] Miftakhul Huda, dkk. (2010). *Membuat Aplikasi Data Base Dengan Java, MySQL, dan NetBeans*. Jakarta: Elex Media Komputindo.
- [7] Sakur. (2004). *Aplikasi Web Database dengan Dreamweaver MX*. Yogyakarta: Andi.
- [8] Pressman. (2002). *Rekayasa Perangkat Lunak*. Yogyakarta: Andi.
- [9] Riyanto. (2011). *Membuat Sendiri Aplikasi E-Commerce dengan PHP dan MySQL Menggunakan CodeIgniter dan JQuery*. Yogyakarta: C.V ANDI OFFSET.
- [10] Larman. (2005). *Applying UML and Patterns : an introduction to object-oriented analsisi and design and iterative development*.
- [11] Safaat, N. H. (2011). *Pemogaman Aplikasi Mobile Smartphone dan PC Berbasis Android*. Bandung: Informatika.
- [12] Ictiara. (2008). *Pemetaan, SIG, dan Google Maps*.
- [13] Ariani, R. &. (2011). *Rekayasa Perangkat Lunak*. Bandung: MODULA