

**PENGEMBANGAN APLIKASI PENERIMAAN
SISWA BARU PADA YAYASAN PERGURUAN
ISLAM AS SA'ADAH PONDOK KELAPA
JAKARTA TIMUR**



Diajukan oleh:

MAHFUD RAMDHANI. A

5235107392

**PROGRAM STUDI PENDIDIKAN TEKNIK INFORMATIKA DAN
KOMPUTER**

JURUSAN TEKNIK ELEKTRO - FAKULTAS TEKNIK

UNIVERSITAS NEGERI JAKARTA

2016

NASKAH PUBLIKASI JURNAL

**PENGEMBANGAN APLIKASI PENERIMAAN SISWA
BARU PADA YAYASAN PERGURUAN ISLAM AS
SA'ADAH PONDOK KELAPA JAKARTA TIMUR**

yang diajukan oleh :

MAHFUD RAMDHANI, A

5235107392

Telah disetujui oleh :

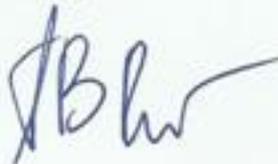
Pembimbing 1



Drs. Bachren Zaini, M.Pd
NIP. 195501101986021001

Tanggal 14-01-2016

Pembimbing 2



Bambang Prasetya Adhi, S.Pd, M.Kom
NIP.198302252014041001

Tanggal 14-01-2016

PENGEMBANGAN APLIKASI PENERIMAAN SISWA BARU PADA YAYASAN PERGURUAN ISLAM AS SA'ADAH PONDOK KELAPA JAKARTA TIMUR

M. Immadudin. A¹, Bachren Zaini², Bambang Prasetya Adhi³

¹Mahasiswa Prodi Pendidikan Teknik Informatika dan Komputer, Teknik Elektro, FT – UNJ

^{2,3}Dosen Prodi Pendidikan Teknik Informatika dan Komputer, Teknik Elektro, FT – UNJ

¹dudinerz@gmail.com, ²bachrenzaini@unj.ac.id, ³bambangpadhi@unj.ac.id

Abstrak

Penerimaan Siswa Baru (PSB) merupakan kegiatan yang amat penting karena proses tersebut mempunyai nilai strategis guna menjaring calon siswa yang berkualitas. Sistem yang sedang berjalan dalam pendaftaran siswa baru di Yayasan Perguruan Islam (YPI) As Sa'adah masih manual dan pencarian data siswa sewaktu di perlukan membutuhkan waktu cukup lama.. Hal ini sangat diperlukan komputer sebagai alat bantu dalam suatu sistem informasi tersebut. Proses pendaftaran, penyeleksian dan pengelompokan kelas merupakan langkah-langkah yang ditempuh untuk proses penerimaan siswa baru di YPI Assa'adah. Kegiatan pendaftaran siswa baru bertujuan untuk proses pendaftaran yang terstruktur dan informatif, sehingga dapat dihasilkan proses kerja yang lebih baik dan maksimal dengan menggunakan penyimpanan data yang tersentralisasi yang dapat memudahkan kegiatan perubahan data, pencarian dan penyimpanan data dengan menggunakan teknologi komputer. Dalam penelitian ini menggunakan metode eksperimen. Aplikasi penerimaan siswa baru dikembangkan dengan Bahasa pemrograman PHP dan basis data MySQL dengan XAMPP sebagai local server. Metode yang digunakan dalam pengembangan sistem ini menggunakan metode pengembangan perangkat lunak waterfall. Pengujian aplikasi penerimaan siswa baru ini menggunakan metode black box testing untuk mengetahui sistem penerimaan siswa baru berjalan dengan baik. Hasil penelitian menyimpulkan bahwa aplikasi ini dapat di dimanfaatkan dalam proses penerimaan siswa baru di YPI As Sa'adah.

Kata kunci : aplikasi penerimaan siswa baru, waterfall, black box testing

1. Pendahuluan

Perkembangan informasi teknologi saat ini semakin pesat, kebutuhan informasi dan pengolahan data dalam banyak aspek kehidupan manusia sangat penting. Dari perkembangan teknologi yang demikian pesat berdampak bagi seluruh kehidupan khususnya penyediaan informasi bagi suatu instansi atau perusahaan yang membutuhkan sistem pengelolaan data secara cepat, tepat dan akurat. Maka perlu menyusun suatu aplikasi pengelolaan data

terkomputerisasi yang baik dan berdaya guna. Aplikasi tersebut diharapkan dapat menyelesaikan tugas-tugas rutin yang berguna dalam pengambilan keputusan oleh organisasi/instansi atau perusahaan tersebut.

Sistem informasi banyak digunakan dalam suatu lembaga. Misalnya dalam perusahaan, perkantoran dan pendidikan. Dalam lembaga pendidikan sistem informasi ini bertujuan untuk promosi atau memberikan gambaran umum mengenai profil

lembaga pendidikan yang bersangkutan, administrasi, dan fasilitas yang lain. Sekolah merupakan lembaga pendidikan, yang menampung peserta didik dan dibina agar mereka memiliki kemampuan, kecerdasan dan keterampilan. Dalam proses pendidikan diperlukan pembinaan secara berkoordinasi dan terarah. Dengan demikian siswa diharapkan dapat mencapai prestasi belajar yang maksimal sehingga tercapainya tujuan pendidikan. Maka lembaga pendidikan di Yayasan Perguruan Islam (YPI) Assa'adah Pondok Kelapa sampai saat ini masih menggunakan sistem informasi yang belum terkomputerisasi atau manual. Setiap calon siswa harus mengisi formulir pendaftaran diselembur kertas, dan panitia mencatat setiap data calon siswa kedalam buku besar dan menyimpan data dalam sebuah ruangan yang berisi dokumen – dokumen lain sehingga sering menimbulkan lambatnya suatu informasi yang dihasilkan atau harus ditingkatkan lagi terutama dalam bidang pendaftaran, penyeleksian dan pengelompokan kelas siswa baru di YPI As Sa'adah Pondok Kelapa. Penerimaan Siswa Baru (PSB) merupakan kegiatan yang amat penting karena proses tersebut mempunyai nilai strategis guna menjaring calon siswa yang berkualitas. Hal ini sangat diperlukan komputer sebagai alat bantu dalam suatu sistem informasi tersebut. Proses pendaftaran, penyeleksian dan pengelompokan kelas merupakan langkah-langkah yang ditempuh guna untuk mengumpulkan data pendaftaran secara sistematis.

Pendaftaran, penyeleksian dan pengelompokan kelas siswa baru di YPI As Sa'adah Pondok Kelapa masih tergolong menggunakan sistem penulisan, sehingga segala prosesnya dari mulai menginput data masih secara tertulis. Adapun sistem yang sedang berjalan dalam pendaftaran siswa baru di YPI As Sa'adah Pondok Kelapa masih menggunakan cara manual yaitu dengan mengisi formulir pendaftaran dan itu pun akan memperlambat pencarian siswa sewaktu data tersebut di perlukan. Sistem penerimaan siswa baru di YPI As Sa'adah Pondok Kelapa Pondok Kelapayang ditujukan untuk penyediaan informasi tentang data siswa baru yang akan di terima. Dengan komputer sebagai alat bantu dalam menyelesaikan masalah yang dihadapi oleh sekolah dan sudah waktunya memakai fasilitas komputer yang meringankan sedikit beban pekerjaan dan diharapkan dapat membantu YPI As Sa'adah Pondok Kelapa Pondok Kelapa untuk mencapai tujuan dalam waktu yang singkat. Dengan adanya aplikasi ini memudahkan *admin* atau panitia penerimaan siswa baru untuk proses seleksi dan mengolah data. Data pendaftaran nantinya akan berlanjut digunakan untuk pembayaran siswa, keperluan laporan siswa ke dinas pendidikan, dan

laporan untuk Bantuan Operasional Sekolah (BOS) atau bantuan – bantuan lain.

2. Dasar Teori

2.1 Perangkat Lunak

Perangkat lunak adalah program computer yang berfungsi sebagai sarana interaksi antara pengguna dan perangkat keras. Perangkat lunak juga dapat dikatakan sebagai penerjemah perintah – perintah yang dijalankan pengguna computer untuk diteruskan atau diproses oleh perangkat keras. Perangkat lunak dibagi menjadi tiga tingkatan, yaitu tingkatan program aplikasi (*application program*) misalnya Microsoft Office, tingkatan sistem operasi (*operating system*) misalnya Microsoft Windows, tingkatan Bahasa pemrograman yang dibagi atas Bahasa pemrograman tingkat tinggi seperti pascal dan Bahasa pemrograman tingkat rendah, yaitu Bahasa rakitan (Janner Simartama, 2010). Perangkat lunak pada umumnya digunakan untuk mengontrol perangkat keras yang sering disebut *device driver*, melakukan proses perhitungan, berinteraksi dengan perangkat lunak yang lebih mendasar lainnya seperti sistem operasi, Bahasa pemrograman dan lain-lain.

2.2 Aplikasi

Aplikasi adalah satu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas seperti game, pelayanan masyarakat, periklanan, atau semua proses yang dilakukan manusia (Hengky W. Pranama, 2012). Aplikasi adalah kumpulan perintah program yang dibuat untuk melakukan pekerjaan tertentu (Hendrayudi, 2008). Aplikasi merupakan program yang dikembangkan untuk memenuhi kebutuhan pengguna dalam menjalankan pekerjaan tertentu (Yuhefizar, 2012). Aplikasi dapat digolongkan menjadi beberapa kelas yaitu perangkat lunak perusahaan (*enterprise*), perangkat lunak infrastruktur perusahaan, perangkat lunak informasi kerja, perangkat lunak media dan hiburan, perangkat lunak pendidikan, perangkat lunak pengembangan media, dan perangkat lunak rekayasa produk. Berdasarkan pengertian diatas, dapat disimpulkan bahwa aplikasi merupakan sebuah program yang dibuat dalam sebuah perangkat lunak untuk memudahkan pekerjaan atau tugas-tugas seperti penerapan, penggunaan, dan penambahan data yang dibutuhkan.

2.3 Framework

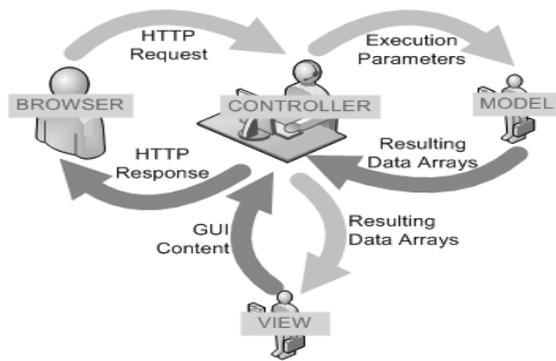
Framework adalah aplikasi *semi-complete* yang dapat digunakan untuk membuat aplikasi lain yang lebih spesifik (Husted 2003). *Framework* yang ideal merupakan intisari dari pendekatan terbaik dalam pengembangan perangkat lunak. Alasan

pengembangan sebuah *framework* adalah penggunaan kembali perangkat lunak, jadi suatu perangkat lunak dikembangkan dari perangkat lunak yang telah ada.

Saat ini banyak sekali *framework* berbasis PHP, diantaranya yang paling populer adalah *Zend Framework*, *Symphony*, *Prado*, *CakePHP*, *CodeIgniter* dan lain-lain. Menurut beberapa survei, *CodeIgniter* adalah *framework* yang paling mudah dipelajari, karena kesederhanaan strukturnya dan kelengkapan dokumentasinya yang baik. Selain itu, kerapian kode (*clean and healthy code*) dan ukurannya yang ramping membuatnya lebih cepat diakses dibandingkan dengan *framework* lainnya. Perlu diketahui nyaris (bahkan mungkin semua) *framework* menggunakan konsep yang sama dalam membangun suatu aplikasi, yaitu konsep MVC (*Model View Controller*). Kesimpulannya, kalau kita memahami MVC, maka secara garis besar kita sudah menguasai *framework*, apapun jenis *framework*-nya. Dan satu lagi kuncinya, pemahaman terhadap OOP (*Object Oriented Programming*) akan sangat membantu dalam menguasai *framework*.

2.4 Model View Controller (MVC)

Seperti kebanyakan *Software engineering*, konsep *Model-View-Controller* diciptakan oleh programmer *Smalltalk*. Lebih tepatnya diciptakan oleh salah seorang programmer *Smalltalk*, yaitu Trygve Reenskaug. Beliau memperkenalkan konsep ini pada paper yang beliau rilis tanggal 10 November 1979. Konsep ini sekarang populer dalam pemrograman lanjutan karena kemudahan dalam perawatan dan dapat digunakan oleh banyak programmer. MVC adalah sebuah metode dalam membuat aplikasi dengan memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, antarmuka pengguna, dan bagian yang menjadi kontrol utama aplikasi (Nur widiyanto. 2010).



Gambar 2.1. Cara Kerja MVC

2.5 Object Relational Mapping(ORM)

Object Relational Mapping merupakan teknik otomatisasi dan transparansi dari *object persistence* ke dalam tabel pada *Database*, menggunakan metadata yang mendeskripsikan pemetaan antara objek dan *Database* (Bauer dan King 2007). ORM berperan dalam lapisan model dalam konsep MVC. Model adalah sebuah lapisan yang paling dekat dengan sumber data, baik itu berasal dari *Database*, *webservice*, maupun *file system*. *Object Relational Mapping* (ORM) adalah sebuah *framework* yang mengatasi perbedaan sistem *Database* yang bersifat relational dengan paradigma pengembangan aplikasi yang berorientasi objek. Selain itu, ORM juga menjembatani dialek SQL yang digunakan, sehingga apapun produk RDBMS yang digunakan tidak berpengaruh terhadap kode program. ORM merupakan solusi yang mengatasi perbedaan aspek-aspek ketidaksesuaian antara konsep pemrograman berorientasi objek dengan konsep *Database* relasional.

3. Metodologi

Proses pengembangan aplikasi penerimaan siswa baru menggunakan metode pengembangan sistem waterfall untuk pengembangan perangkat lunak dan eksperimen untuk metode penelitian. Tahapan penelitian diawali dengan melakukan proses observasi di YPI As Sa'adah Pondok Kelapa mengenai penerimaan siswa baru dan wawancara kepada ketua panitia PSB YPI As Sa'adah. Tahapan dari *Waterfall* adalah sebagai berikut:

1. Analysis

Analisis bertujuan untuk menentukan hal-hal rinci yang akan dikerjakan oleh perangkat lunak. Analisis mencakup studi kelayakan dan analisis kebutuhan. Pengembangan aplikasi PSB bertujuan untuk melakukan registrasi atau pendaftaran siswa baru. Pada tahap penyeleksian PSB ini untuk jenjang TK dan SD menggunakan batas umur sesuai dengan peraturan yang berlaku di kementerian pendidikan dan SMP dan SMK menggunakan NEM atau nilai SKHUN sesuai dengan standar sekolah. Tidak semua data siswa Yayasan Perguruan Islam Assa'adah yang akan digunakan dalam proses pendaftaran siswa baru, hanya data siswa baru tahun ajaran 2015-2016 dari jenjang TK, SD, SMP, dan SMK yang ada di YPI As Sa'adah Pondok Kelapa. Berikut adalah beberapa analisis kebutuhan yang akan membentuk spesifikasi kebutuhan, dan beberapa diantaranya akan direalisasikan pada penelitian ini:

- a. Admin dapat membuat akun dengan mengisi data diri

- b. Admin dapat melihat, mengupdate dan menghapus data calon siswa
- c. Admin dapat mengatur kuota dan nilai minimal untuk seleksi
- d. Admin dapat mengatur halaman depan berupa informasi terbaru
- e. Admin dapat mengexport data siswa yang diterima kedalam bentuk csv atau excel.
- f. Admin dapat mengatur pembukaan dan penutupan pendaftaran.
- g. Admin dapat melihat grafik pendaftaran yang sudah masuk.
- h. *User* dapat melihat informasi mengenai pendaftaran
- i. *User* membuat akun sebelum mendaftar
- j. *User login* dengan akun yang dibuat
- k. *User* dapat mengisi form pendaftaran dengan data diri yang benar
- l. *User* dapat mencetak formulir pembayaran PSB
- m. *User* dapat mencetak biodata atau bukti pendaftaran
- n. *User* dapat melihat pengumuman hasil seleksi

Berdasarkan kebutuhan diatas, maka dibuatlah pengembangan aplikasi PSB berdasarkan kebutuhan yang ada. Tugas utama dari aplikasi PSB ini adalah seleksi calon siswa jenjang TK dan SD menggunakan batas umur sesuai dengan peraturan yang berlaku di kementerian pendidikan dan SMP dan SMK menggunakan NEM atau nilai SKHUN sesuai dengan standar sekolah.

2. Desain

Tahap desain pada penelitian ini adalah membuat desain rancangan halaman – halaman yang diperlukan dalam aplikasi PSB ini. Setelah itu membuat rancangan dokumentasi sistem dengan menggunakan *Unified Modeling Language* (UML) yaitu dengan dibuatnya *Usecase* admin dan *user*. *Usecase* mendeskripsikan sebuah interaksi satu atau lebih *actor* dengan sistem informasi yang akan dibuat. Pada pengembangan aplikasi PSB ada dua *Usecase* yaitu *Usecase* admin dan *Usecase* *user*. Selanjutnya perancangan *sequence diagram* yang menggambarkan kelakuan objek pada *Usecase* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Membuat diagram *sequence* juga dibutuhkan untuk melihat *scenario* yang ada pada *Usecase*. Pada sistem ini ada 13 *sequence*

diagram Dan yang terakhir adalah class diagram yang menggambarkan struktur sistem dari segi pendefinisian kelas – kelas yang akan dibuat untuk membangun sistem. *Class diagram* dibuat agar pembuat program membuat kelas – kelas sesuai rancangan didalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron. Pada pengembangan aplikasi PSB ini terdiri dari empat belas *sequence diagram* sebagai berikut : delapan *sequence diagram* admin, dan enam *sequence diagram* *user*.

3. Implementation

Setelah desain dilakukan, tahap selanjutnya adalah membuat kode program. Coding yang dilakukan penulis dalam *management* data yang diolah dalam satu paket aplikasi yang tergabung, yaitu XAMPP versi 1.8.3, dimana didalamnya menggunakan PHP.5.5.15 sebagai bahasa pemrograman dan mysql versi 5.0.41 sebagai data basenya, sedangkan webservernya menggunakan Apache/2.2.4

4. Pengujian

Pada aplikasi ini menggunakan pengujian blackbox dengan tipe *functional testing*. Perangkat lunak diuji untuk persyaratan fungsional. Pengujian dilakukan dalam bentuk tertulis untuk memeriksa apakah aplikasi berjalan seperti yang diharapkan. Walaupun pengujian fungsional sering dilakukan dibagian akhir dari siklus pengembangan, masing – masing komponen dan proses dapat diuji pada awal pengembangan, bahkan sebelum sistem berfungsi, pengujian ini sudah dapat dilakukan pada seluruh sistem. Pengujian fungsional meliputi seberapa baik sistem melaksanakan fungsinya, termasuk perintah – perintah pengguna, manipulasi data, pencarian dan proses bisnis, pengguna layar, dan integrasi. Pengujian fungsional juga meliputi permukaan yang jelas dari jenis fungsi – fungsi, serta operasi *back-end* seperti, keamanan dan bagaimana meningkatkan sistem (Janner Simartama, 2010).

Dengan menggunakan tipe *Black Box* ini kebenaran sistem yang diuji dilihat berdasarkan *output* yang ada dari data atau kondisi masukkan yang diberikan untuk fungsi yang ada. Tanpa melihat proses, dari *output* tersebut dapat diukur dan diketahui kesalahan dari program dalam memenuhi kebutuhan *user*.

5. Pemeliharaan

Pemeliharaan dianggap perlu dilakukan untuk memperbaiki jika ada kesalahan sistem yang tidak berjalan sesuai dengan

fungsinya yang mungkin disebabkan karena perubahan tampilan atau perubahan kode program. Walaupun sudah diuji sebelumnya

4. Hasil Penelitian

4.1 Hasil dan Analisis

Pendaftaran, penyeleksian dan pengelompokan kelas siswa baru di YPI As Sa'adah Pondok Kelapa menjadi keunggulan aplikasi penerimaan siswa baru, sehingga memudahkan admin atau panitia dalam hal mengelola data calon siswa. Pada tahap seleksi ada dua aturan yaitu dengan standar umur untuk jenjang TK dan SD, dan nilai SKHUN untuk jenjang SMP dan SMK. Penerimaan Siswa Baru (PSB) merupakan kegiatan yang amat penting karena proses tersebut mempunyai nilai strategis guna menjaring calon siswa yang berkualitas. Dengan alasan tersebut, menjadikan aplikasi penerimaan siswa baru layak untuk dikembangkan. Berikut adalah beberapa analisis masalah pada BAB III yang harus dijawab oleh *software*:

1. Cara mengotomasi proses akses penerimaan siswa baru.
2. Memberikan akses pada admin untuk memasukan ,merubah, dan menghapus data pendaftar. Dan memberikan akses kepada siswa baru untuk mendaftar sesuai dengan jenjang.
3. Cara melihat alur pendaftar
4. Cara menginput data pribadi calon siswa.
5. Cara melihat pengumuman seleksi
6. Cara menampilkan resi pembayaran.
7. Cara mencetak bukti pendaftaran

Berikut adalah hasil dari pengujian aplikasias permasalah tersebut:

1. Masalah mengotomasi proses akses aplikasi penerimaan siswa baru dijawab dengan adanya sistem *login* dan hak akses pada proses masuk. Siswa mengakses aplikasi sebagai *user* dan pihak yayasan mengakses sebagai *admin*.
2. Pemberian hak akses pada admin untuk memasukan, merubah, dan menghapus setiap data yang dibutuhkan calon siswa
3. Cara kerja penginputan data pribadi pendaftar dan data yang dibutuhkan calon siswa pada *software* adalah dengan cara memilih menu pendaftaran dan menu biodata untuk merubah yang salah pada *form* yang sudah disediakan. Data yang sudah terinput pada *form* akan tersimpan pada *database*.
4. Untuk bisa mendaftar calon siswa harus melakukan create akun. Setelah mempunyai akun pendaftar melakukan pendaftaran

dengan mengisi biodata terlebih dahulu. Jika ada kesalahan maka pendaftar harus login terlebih dahulu, setelah itu pendaftar bisa merubah kesalahan pada data pribadi.

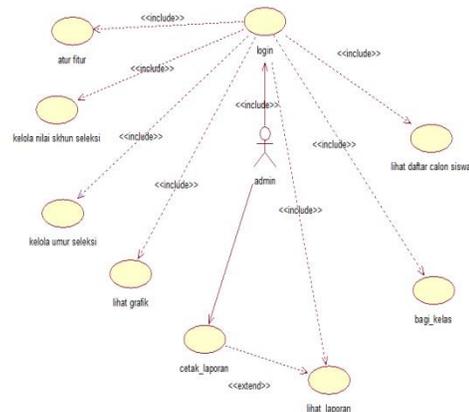
5. Setelah mengisi biodata calon siswa harus mencetak bukti pendaftaran atau formulir sebagai bukti pendaftaran dan untuk keperluan validasi berkas.
6. Calon siswa mencetak resi pembayaran setelah mengisi formulir yang berisi data pribadi. Cetak
7. Calon siswa melihat pengumuman dengan melakukan login terlebih dahulu dan calon siswa harus melakukan validasi berkas dan membayar pendaftaran. Jika belum melakukan validasi berkas maka tidak ada dalam pengumuman.

4.2 Perancangan

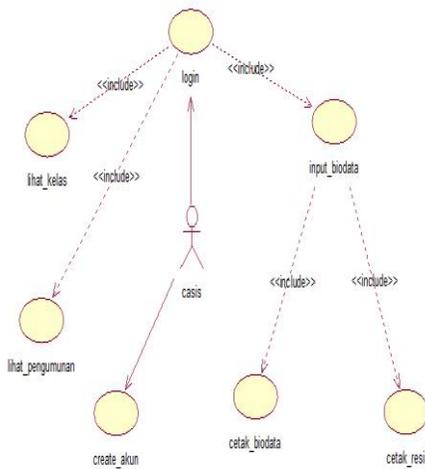
Tahap desain pada penelitian ini adalah membuat desain rancangan halaman – halaman yang diperlukan dalam aplikasi penerimaan siswa baru ini. Seperti yang telah disampaikan pada bab III. Setelah itu membuat rancangan dokumentasi sistem dengan menggunakan *Unified Modeling Language (UML)* yaitu dengan dibuatnya usecase, sequen diagram dan class diagram.

4.2.1 Usecase Diagram

Usecase Diagram secara grafis menggambarkan interaksi antara sistem, sistem eksternal, dan pengguna. Dengan kata lain *Usecase diagram* secara grafis mendeskripsikan siapa yang akan menggunakan sistem dan dalam cara apa pengguna (*user*) mengharapkan interaksi dengan sistem itu. *Usecase* secara naratif digunakan untuk secara tekstual menggambarkan sekuensi langkah-langkah dari setiap interaksi. Berikut ini adalah *Usecase* dari sistem PSB YPI As Sa'adah Pondok Kelapa :



Gambar 4.1 Usecase Admin

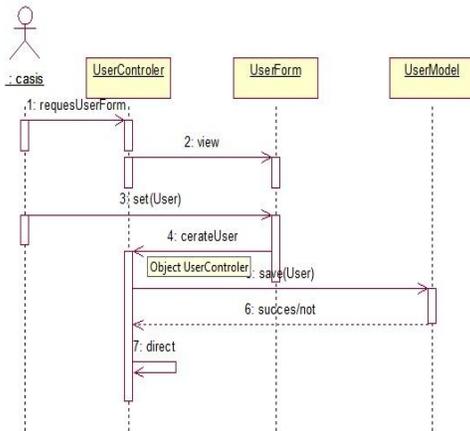


Gambar 4.2 Usecase Calon Siswa

4.2.2 Sequence Diagram

Sequence diagram menggambarkan bagaimana objek berinteraksi dengan satu sama lain melalui pesan pada sekuensi sebuah usecase atau operasi. Diagram ini mengilustrasikan bagaimana pesan terkirim dan diterima di antara objek dan dalam sekuensi. Berikut ini adalah sequence diagram dari Aplikasi PSB YPI As Sa'adah Pondok Kelapa :

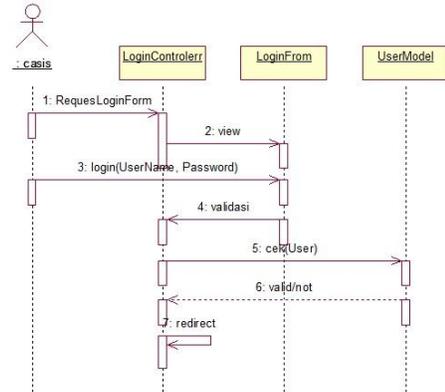
1. Sequence Diagram Create Akun Calon Siswa



Gambar 4.3 Sequence Diagram Create Akun Calon Siswa

Dari gambar diatas proses sequence diagram dimulai dari calon siswa meminta form user dari user controller lalu user controller mengirim pesan kepada user form untuk menampilkan. Calon siswa mengisi biodata user dan mensubmit biodata lalu user model menyimpan biodata kedalam database. Setelah itu User controller mengarahkan ke success page atau error

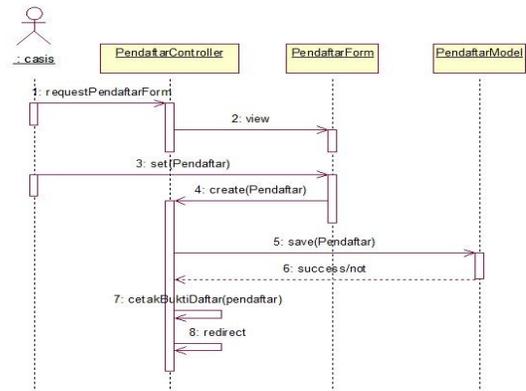
2. Sequence Diagram Login Siswa



Gambar 4.4. Sequence Diagram Login Siswa

proses sequence diagram dimulai dari calon siswa meminta form login dari login controller. Lalu login controller mengirim pesan kepada login form untuk menampilkan form login. Calon siswa mengisi form login dengan username dan password yang sudah dibuat sebelumnya dan mensubmit. Login controller mengecek kepada user model dan mengarahkan ke success atau error

3. Sequence Diagram Daftar Diri Siswa

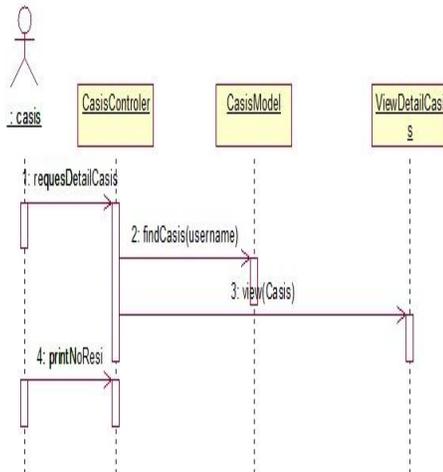


Gambar 4.6 Sequence Diagram Daftar Diri Siswa

pada gambar sequence diagram diatas calon siswa meminta form pendaftaran dari pendaftar controller. Lalu pendaftar controller mengirim pesan kepada pendaftar form untuk menampilkan. Calon siswa

mengisi *form* pendaftaran dan *mensubmit*. Pendaftar *model* menyimpan biodata pendaftar dan pendaftar *controller* mengarahkan ke halaman *success* atau *error*. Pendaftar *controller* juga mengarahkan ke halaman cetak bukti pendaftaran

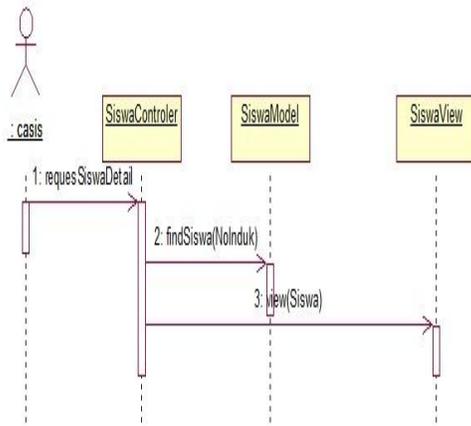
4. Sequence Diagram Cetak Resi



Gambar 4.7 Sequence Diagram Cetak Resi

pada gambar *sequence* diagram diatas proses dimulai dari calon siswa meminta detail casis dari casis *controller*. Lalu casis *controller* mengirim pesan ke casis *model* untuk mencari dengan username tersebut. Casis *controller* mengirim pesan ke *view detail* casis untuk menampilkan *form* cetak resi. Sebelum mencetak resi sistem mencari detail *user* dan menampilkan detail *user*. Setelah itu calon siswa *printout* resi pembayaran

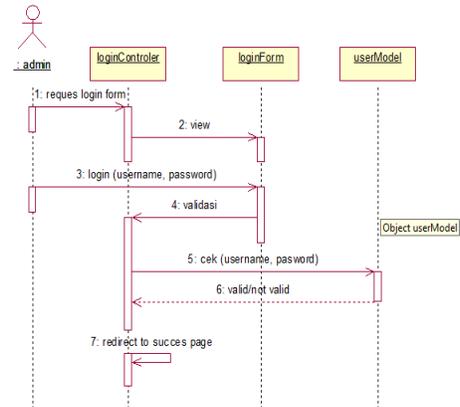
5. Sequence Diagram Lihat Kelas



Gambar 4.8 Sequence Diagram Lihat Kelas

pada gambar proses *sequence diagram* diatas casis meminta detail siswa dari siswa *controller*, lalu siswa *controller* mengirim pesan kepada siswa *model* untuk mencari detail siswa dengan nomor induk tersebut. Setelah selesai siswa *controller* mengirim pesan kepada siswa *view* untuk menampilkan.

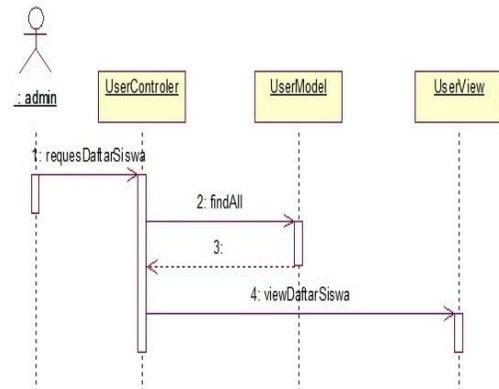
6. Sequence Diagram Login Admin



Gambar 4.9 Sequence Diagram Login Admin

Pada *sequence diagram* dimulai dari admin meminta *form login* dari login *controller*. Lalu login *controller* mengirim pesan kepada login *form* untuk menampilkan halaman login. Admin mengisi *form login* dengan *username*, *password* dan *mensubmit*. Setelah itu login *controller* mengirim pesan kepada *user model* untuk mengecek *username*, *password* apakah valid atau tidak. Login *controller* mengarahkan ke halaman *success* atau *error*.

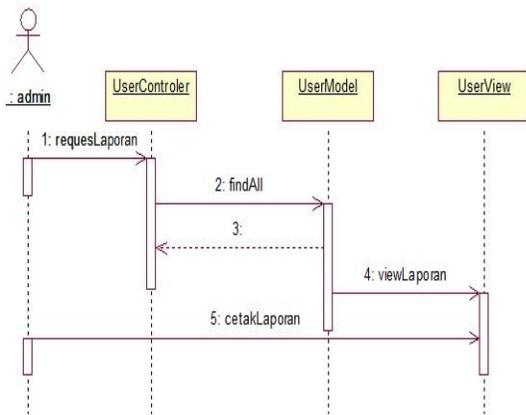
7. Sequence Diagram Daftar Siswa Admin



Gambar 4.10 Sequence Diagram Daftar Siswa Admin

Pada sequence diagram diatas Admin meminta data siswa dari *user controller*. Lalu *user controller* mengirim pesan kepada *user model* untuk mencari data siswa yang telah masuk di sistem. *User controller* mengirim pesan kepada *user view* untuk menampilkan dseluruh data siwa yang sudah mendaftar. Admin bisa melihat, mengubah, dan menghapus sesuai dengan ketentuan yang berlaku, jika ada masalah dalam proses pendaftaran.

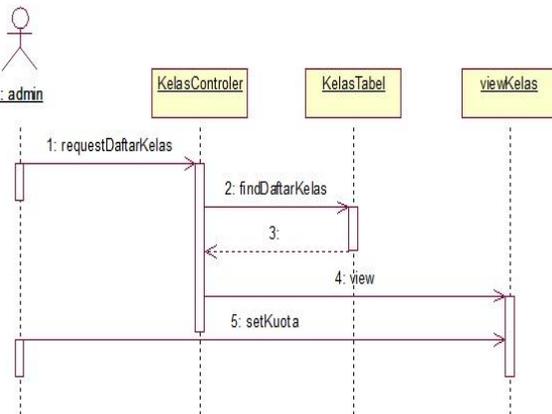
8. Sequence Diagram Cetak Laporan



Gambar 4.11 Sequence Diagram Cetak Laporan

Pada proses cetak laporan ini admin berhak untuk mengetahui semua pendaftar yang sudah masuk di sistem. Admin meminta laporan dari *user controller*. Lalu *user controller* mengirim pesan kepada *user tabel* untuk mencari laporan. Setelah itu *user tabel* mengirim pesan kepada *user view* untuk menampilkan laoporan. Admin melihat laporan dan mencetak laporan.

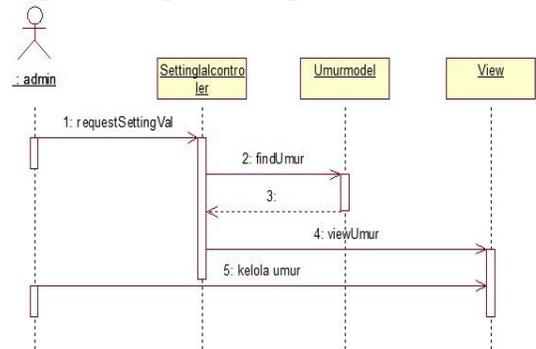
9. Sequence Diagram Bagi Kelas



Gambar 4.12 Sequence Diagram Bagi Kelas

Pada gambar *sequence diagram* bagi kelas diatas admin meminta daftar kelas dari kelas controller. Lalu kelas controller mengirim pesan kepada kelas tabel untuk mencari. Selanjutnya kelas controller mengirim pesan kepada view untuk menampilkan. Admin mengelola kuota susuai yang dibutuhkan oleh sekolah. Pada gambar *sequence diagram* bagi kelas diatas admin meminta daftar kelas dari kelas controller. Lalu kelas controller mengirim pesan kepada kelas tabel untuk mencari. Selanjutnya kelas controller mengirim pesan kepada view untuk menampilkan. Admin mengelola kuota susuai yang dibutuhkan oleh sekolah

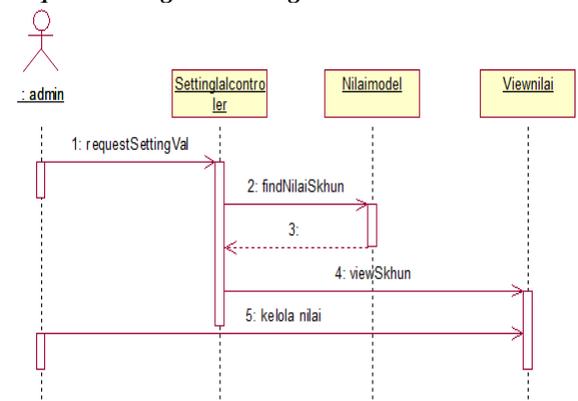
10. Sequence Diagram Setting Value Umur



Gambar 4.13 Sequence Diagram Set Val Umur

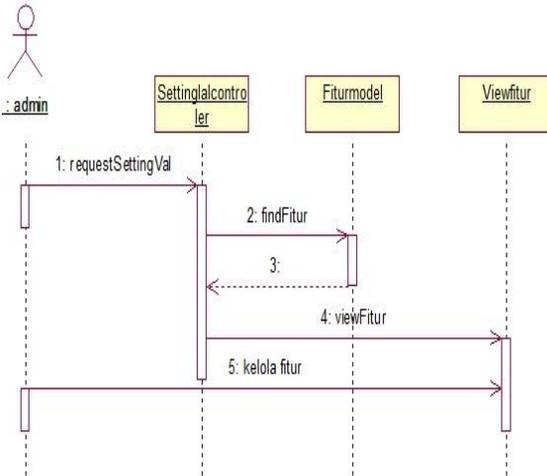
pada proses *setting value* umur ini berfungsi untuk melakukan seleksi di jenjang TK dan SD. Admin meminta *setting value* dari *setting controller*. Lalu *setting controller* mengirim pesan kepada umur *model* untuk mencari. Setelah itu *setting controller* mengirim pesan kepada *view* untuk menampilkan halaman *setting value* dan admin mengelola umur untuk menentukan seleksi

11. Sequence Diagram Setting Value Nilai



Gambar 4.14 Sequence Diagram Set Val Nilai

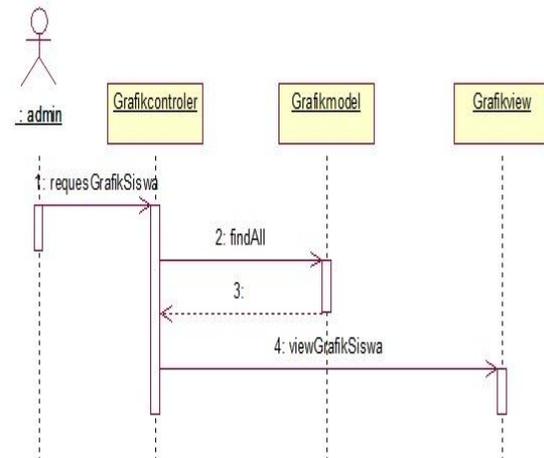
12. Sequence Diagram Atur Fitur



Gambar 4.15 Sequence Diagram Atur Fitur

pada proses *setting value* fitur ini berfungsi untuk mengatur proses pendaftaran dibuka dan ditutup. Admin meminta atur fitur dari setting controller. Lalu setting controller mengirim pesan kepada fitur model untuk mencari. Setelah itu setting controller mengirim pesan kepada view untuk menampilkan halaman kelola fitur dan admin mengelola fitur sesuai dengan kebutuhan.

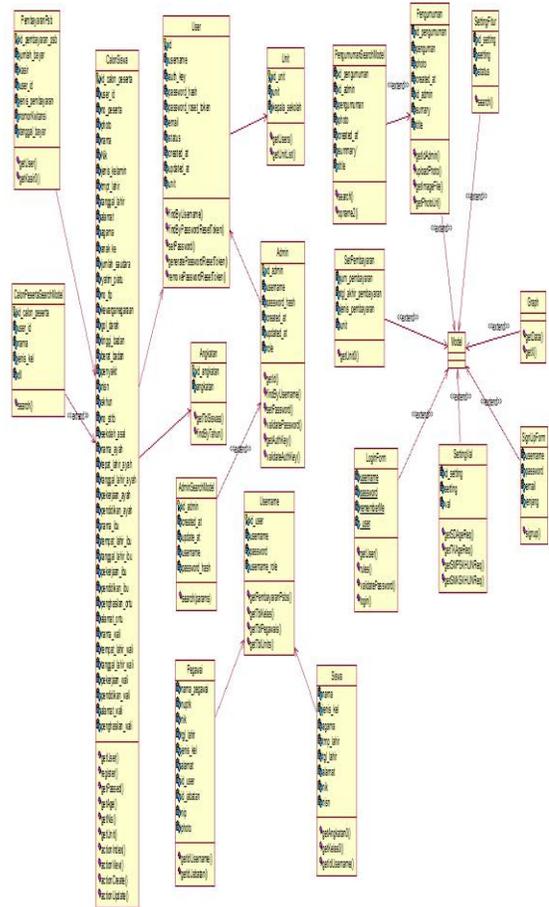
13. Sequence Diagram Lihat Grafik Admin



Gambar 4.16 Sequence Diagram Lihat Grafik

4.2.3 Class Diagram

Mengambarkan struktur object sistem. Diagram ini menunjukkan *class object* yang menyusun sistem dan juga hubungan antara *class object* tersebut. Berikut ini adalah *class diagram* dari sistem penerimaan siswa baru YPI As Sa'adah Pondok Kelapa :



Gambar 4.17 Class Diagram

5. Kesimpulan dan Saran

5.1 Kesimpulan

Pengembangan Aplikasi penerimaan siswa baru berbasis web merupakan salah satu solusi yang tepat bagi Yayasan Pendidikan Islam As Sa'adah. Dari proses dan hasil penelitian, peneliti dapat mengambil kesimpulan :

1. Berdasarkan pengujian aplikasi penerimaan siswa baru pada YPI Assa'adah sudah sesuai dengan tujuan penelitian.
2. Aplikasi penerimaan siswa baru yang dikembangkan mampu membantu tugas pokok dalam melakukan proses penyeleksian, penyimpanan data dan pengolahan data sesuai dengan kebutuhan.

5.2 Saran

Sistem informasi penerimaan siswa baru yang telah dikembangkan ini memiliki beberapa kekurangan, sehingga perlu adanya pengembangan dalam menyempurnakan aplikasi ini. Berikut ini beberapa kekurangan yang harus dikembangkan :

1. Dalam sistem penerimaan siswa baru ini terdapat banyak kelemahan terutama keamanan (*security*) sistem. Peneliti berharap pada penelitian selanjutnya masalah keamanan sistem harus diperbaiki.
2. Adanya pembayaran melalui banking atau ATM agar memudahkan siswa dalam melakukan pembayaran pendaftaran agar dapat memudahkan calon siswa dalam melakukan pembayaran.
3. Untuk penelitian selanjutnya diharapkan dapat menambahkan fasilitas pendukung.

Nur widiyanto. 2010. *Membangun Aplikasi Java Enterprise dengan Arsitektur Model View Controller*. Andi publisher

Roger Pressman.2007. *Rekayasa Perangkat Lunak (buku satu)*. Yogyakarta andi

Rosa,& M. Shalaludin. 2014. *Rekayasa Perangkat Lunak*. Bandung : Informatika.

Yuhefizar.2012. *10 Jam Menguasai Teknologi dan Aplikasinya*.Jakarta: Alex Media Komputerindo.

YM Kusuma. 2012. *PHP Menyelesaikan Website 30 juta*. Jakarta: jasakom.

Daftar Pustaka:

Abdul Kadir. 2009. *Dasar dan Perancangan dan Implementasi Database Relasional*. Yogyakarta : Andi.

Bauer C, King G. 2007. *Java Persistence with Hibernate*. United States :Manning.

Bonczek, R. H.; Holsapple, C. W.; & Whinston, A. B. 1980, *The Evolving Roles of Models in Decision Support Systems*, Decision Science, Vol. 11, No. 2.

Fakultas Teknik. 2012. *Buku Pedoman Skripsi / Komprehensif / Karya Inovatif (S1)*. Jakarta, UNJ cetakan II

Hanna, "Farewall to waterfalls", software magazine, Mei 1995

Hendrayudi, VB. 2008. *Untuk Berbagai Keperluan Pemrograman*. Jakarta: PT. Elex Media Komputindo

Hengky W. Pranama.2012. *Kunci Sukses Aplikasi Penjualan Berbasis Access*.Jakarta: AlexMedia Komputerindo.

Husted T. 2003. *Struts in Action : Building Web Application with the Leading Java Framework*. United States : Manning.

Indrajani.2014. *Pengantar Sistem Basis Data*.Jakarta: PT Elex Media Komputindo

Janner Simartama. 2010. *Rekayasa Perangkat Lunak*. Yogyakarta andi

Kamus Besar Bahasa Indonesia , Edisi keempat. 2008. Jakarta: Gramedia Pusaka Utama