# **Verifikasi Proyek**

Telah dipresentasikan pada proyek ini dihadapan tim verifikasi, dosen verifikator, dan program studi.

|  |  |  |
| --- | --- | --- |
| 1. | Ir. Hardi Jamhur | (…………………………….) |
| 2. | Irmayansyah M.Kom | (…………………………….) |

**[ HALAMAN INI SENGAJA DIKOSONGKAN ]**

# **Lembar Persetujuan**

**PENERAPAN CAKEPHP FRAMEWORK DAN MYSQL UNTUK AGENDARIS SURAT DAN NOTIFIKASI DI STIKOM BINANIAGA BOGOR**

|  |  |  |
| --- | --- | --- |
| **Oleh :** | | |
| **Pelaksana 1 :** |  | **Pelaksana 2 :** |
|  |  |  |
| **Ogiano Waskitajaya**  **1512027** |  | **Yanuar Nurcahyo**  **1512018** |
|  |  |  |
|  | **Disetujui Oleh :** |  |
|  |  |  |
| **Dosen Pembimbing** |  | **Pembimbing Proyek** |
|  |  |  |
| **Anggra Triawan, S.Kom**  **11.120.1003** |  | **Novi Nurrohmatunnisa**  **12.302.007** |
|  |  |  |
|  | **Dosen ITPM** |  |
|  |  |  |
|  | **Ir. Hardi Jamhur**  **11.119.9101** |  |
|  |  |  |
| **Penanggung Jawab Proyek** |  | **Ketua Program Studi** |
|  |  |  |
| **Rajib Ghany, S.Kom**  **11.220.1202** |  | **Irmayansyah, M.Kom**  **11.120.0404** |

**[ HALAMAN INI SENGAJA DIKOSONGKAN ]**

# **Kata Pengantar**

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa yang telah memberikan rahmat dan karunianya, sehingga penulis dapat menyelesaikan laporan proyek yang berjudul Penerapan CakePHP Framework dan MySQL untuk Agendaris Surat dan Notifikasi di STIKOM Binaniaga Bogor dengan baik.

Informasi Teknologi Proyek Minor ini merupakan salah satu matakuliah yang wajib ditempuh di STIKOM Binaniaga Bogor. Laporan proyek ini disusun untuk memenuhi tugas dan merupakan lanjutan dari pada seminar proposal Project Minor.

Dengan selesainya laporan proyek ini tidak terlepas dari bantuan banyak pihak yang telah memberikan masukan-masukan kepada penulis. Untuk itu penulis mengucapkan banyak terimakasih kepada :

1. Ir. Hardi Jamhur selaku dosen mata kuliah Information Technology Project Minor yang telah memberikan arahan materi dan wawasan pengetahuan.
2. Anggra Triawan, S.Kom selaku dosen pembimbing yang telah memberikan arahan materi dan memberikan masukan – masukan yang bermanfaat.
3. Novi Nurrohmatunnisa selaku pembimbing proyek yang telah menjelaskan mengenai teknikal proyek.
4. Kawan – kawan diskusi di STIKOM Binaniaga yang telah membantu memberikan konstribusinya dalam proyek ini.

Penulis menyadari bahwa masih banyak kekurangan dari laporan ini, baik dari materi maupun teknik penyajiannya, mengingat kurangnya pengetahuan dan pengalaman penulis. Oleh karena itu, kritik dan saran yang membangun sangat penulis harapkan.

Bogor, September 2015

Penyusun

**[HALAMAN INI SENGAJA DIKOSONGKAN]**

# **Daftar Isi**

[**Verifikasi Proyek** i](#_Toc433455197)

[**Lembar Persetujuan** iii](#_Toc433455198)

[**Kata Pengantar** v](#_Toc433455199)

[**Daftar Isi** vii](#_Toc433455200)

[**Daftar Tabel** ix](#_Toc433455201)

[**Daftar Gambar** xi](#_Toc433455202)

[**Abstrak** xiii](#_Toc433455203)

[**PENDAHULUAN** 1](#_Toc433455204)

[**A.** **Latar Belakang** 1](#_Toc433455205)

[**B.** **Masalah Proyek** 2](#_Toc433455206)

[**C.** **Tujuan Proyek** 2](#_Toc433455207)

[**D.** **Signifikasi Proyek** 3](#_Toc433455208)

[**PENDEKATAN PROYEK** 4](#_Toc433455209)

[**1.** **Feasibility (Kelayakan)** 4](#_Toc433455210)

[**A.** **Aspek Ekonomi** 4](#_Toc433455211)

[**B.** **Aspek Operasional** 5](#_Toc433455212)

[**C.** **Aspek Teknikal** 7](#_Toc433455213)

[**D.** **Aspek Hukum** 7](#_Toc433455214)

[**E.** **Aspek Kultural** 7](#_Toc433455215)

[**F.** **Kelayakan Jadwal** 8](#_Toc433455216)

[**2.** **Penjadwalan** 9](#_Toc433455217)

[**3.** **Pemikiran Teoritis** 9](#_Toc433455218)

[**4.** **Kerangka Pemecahan** 10](#_Toc433455219)

[**5.** **Konsepsi Pengembangan** 10](#_Toc433455220)

[**6.** **Analisis** 11](#_Toc433455221)

[**PEMODELAN DAN STRUKTUR** 12](#_Toc433455222)

[**1.** **Arsitektur** 12](#_Toc433455223)

[**a.** **Hardware** 12](#_Toc433455224)

[**b.** **Software** 12](#_Toc433455225)

[**2.** **Komponen Pengembangan** 13](#_Toc433455226)

[**A.** **Proses Bisnis** 13](#_Toc433455227)

[**B.** **Diagram Konteks** 15](#_Toc433455228)

[**C.** **Diagram Usecase** 16](#_Toc433455229)

[**D.** **Matriks Ranking** 19](#_Toc433455230)

[**E.** **Usecase Narative** 20](#_Toc433455231)

[**F.** **Aktifitas Proses Bisnis** 71](#_Toc433455232)

[**G.** **Pemetaan Objek** 81](#_Toc433455233)

[**H.** **Diagram Sequence** 84](#_Toc433455234)

[**I.** **ERD** 89](#_Toc433455235)

[**J.** **Diagram Kelas** 90](#_Toc433455236)

[**IMPLEMENTASI** 91](#_Toc433455237)

[**1.** **Konstruksi** 91](#_Toc433455238)

[**a.** **Diagram Deployment** 91](#_Toc433455239)

[**b.** **Topologi Sistem** 92](#_Toc433455240)

[**c.** **Pemrograman** 92](#_Toc433455241)

[**d.** **Source Code** 93](#_Toc433455242)

[**e.** **Dukungan User** 121](#_Toc433455243)

[**PENGUJIAN SISTEM** 129](#_Toc433455244)

[**1.** **Pengujian Blackbox** 129](#_Toc433455245)

[**2.** **Pengujian Whitebox** 130](#_Toc433455246)

[**KINERJA SISTEM** 132](#_Toc433455247)

[**IMPLIKASI** 133](#_Toc433455248)

[**RUJUKAN** 135](#_Toc433455249)

[**PROFIL PELAKSANA** 137](#_Toc433455250)

[**LAMPIRAN** 139](#_Toc433455251)

# **Daftar Tabel**

[*Tabel 1. Biaya dan Manfaat Proyek* 4](#_Toc433452838)

[*Table 2. Data Aktifitas Proyek* 8](#_Toc433452839)

[*Table 3. Waktu yang diharapkan dari proyek (Expected Time) ET* 8](#_Toc433452840)

[*Tabel 4. Spesifikasi Hardware* 12](#_Toc433452841)

[*Tabel 5. SPesifikasi Software* 12](#_Toc433452842)

[*Tabel 6. Identifikasi Aktor* 17](#_Toc433452843)

[*Tabel 7. Identifikasi Usecase* 17](#_Toc433452844)

[*Tabel 8. Matriks Ranking* 19](#_Toc433452845)

[*Table 9. Use Case Naratif Persyaratan Bisnis Login* 21](#_Toc433452846)

[*Table 10. Use Case Naratif Analisis SIstem Login* 23](#_Toc433452847)

[*Table 11. Use Case Naratif Design Sistem Login* 25](#_Toc433452848)

[*Table 12. Use Case Naratif Persyaratan Bisnis Logout* 26](#_Toc433452849)

[*Table 13. Use Case Naratif Analisis Sistem Logout* 27](#_Toc433452850)

[*Table 14. Use Case Naratif Design Sistem Logout* 29](#_Toc433452851)

[*Table 15. Use Case Naratif Persyaratan Bisnis Menerima Notifikasi* 30](#_Toc433452852)

[*Table 16. Use Case Naratif Analisis Sistem Menerima Notifikasi* 31](#_Toc433452853)

[*Table 17. Use Case Naratif Design Sistem Menerima Notifikasi* 32](#_Toc433452854)

[*Table 18. Use Case Naratif Persyaratan Bisnis Input Data Surat Masuk* 33](#_Toc433452855)

[*Table 19. Use Case Naratif Analisis Sistem Input Data Surat Masuk* 35](#_Toc433452856)

[*Table 20. Use Case Naratif Design Sistem Input Data Surat Masuk* 37](#_Toc433452857)

[*Table 21. Use Case Naratif Persyaratan Bisnis Membuat Surat Keluar* 38](#_Toc433452858)

[*Table 22. Use Case Naratif Analisis Sistem Membuat Surat Keluar* 40](#_Toc433452859)

[*Table 23. Use Case Naratif Design Sistem Membuat Surat Keluar* 42](#_Toc433452860)

[*Table 24. Use Case Naratif Persyaratan Bisnis Mengisi Disposisi* 43](#_Toc433452861)

[*Table 25. Use Case Naratif Analisis Sistem Mengisi Disposisi* 45](#_Toc433452862)

[*Table 26. Use Case Naratif Design Sistem Mengisi Disposisi* 47](#_Toc433452863)

[*Table 27. Use Case Naratif Persyaratan Bisnis Memberi Perintah ke Kepala Unit* 48](#_Toc433452864)

[*Table 28. Use Case Naratif Analisis Sistem Memberi Perintah ke Kepala Unit* 50](#_Toc433452865)

[*Table 29. Use Case Naratif Design Sistem Memberi Perintah ke Kepala Unit* 52](#_Toc433452866)

[*Table 30. Use Case Naratif Persyaratan Bisnis Melakukan Verifikasi* 53](#_Toc433452867)

[*Table 31. Use Case Naratif Analisis Sistem Melakukan Verifikasi* 55](#_Toc433452868)

[*Table 32. Use Case Naratif Design Sistem Melakukan Verifikasi* 57](#_Toc433452869)

[*Table 33. Use Case Naratif Persyaratan Bisnis Memberi Paraf Koreksi* 58](#_Toc433452870)

[*Table 34. Use Case Naratif Analisis SIstem Memberi Paraf Koreksi* 60](#_Toc433452871)

[*Table 35. Use Case Naratif Design Sistem Memberi Paraf Koreksi* 62](#_Toc433452872)

[*Table 36. Use Case Naratif Persyaratan Bisnis Memberi Tanda Tangan* 63](#_Toc433452873)

[*Table 37. Use Case Naratif Analisis Sistem Memberi Tanda Tangan* 65](#_Toc433452874)

[*Table 38. Use Case Naratif Design Sistem Memberi Tanda Tangan* 67](#_Toc433452875)

[*Table 39. Use Case Naratif Persyaratan Bisnis Melihat Disposisi* 68](#_Toc433452876)

[*Table 40. Use Case Naratif Analisis SIstem Melihat Disposisi* 69](#_Toc433452877)

[*Table 41. Use Case Naratif Design SIstem Melihat Disposisi* 70](#_Toc433452878)

[*Tabel 42. Pengujian Black box pada validasi data login administrasi* 129](#_Toc433452879)

# **Daftar Gambar**

[*Gambar 1. Chart Gant penjadwalan proyek* 9](#_Toc433463623)

[*Gambar 2. Bagan cara kerja Sistem* 12](#_Toc433463624)

[*Gambar 3. Proses Bisnis Surat Masuk* 13](#_Toc433463625)

[*Gambar 4. Proses Bisnis Surat Keluar Dalam Instansi* 14](#_Toc433463626)

[*Gambar 5. Proses Bisnis Surat Keluar Luar Instansi* 14](#_Toc433463627)

[*Gambar 6. Diagram Konteks* 15](#_Toc433463628)

[*Gambar 7. Diagram Usecase* 16](#_Toc433463629)

[*Gambar 8. Diagram Activity Login* 71](#_Toc433463630)

[*Gambar 9. Diagram Activity Logout* 72](#_Toc433463631)

[*Gambar 10. Diagram Activity Menerima Notifikasi* 72](#_Toc433463632)

[*Gambar 11. Diagram Activity Input Data Surat Masuk* 73](#_Toc433463633)

[*Gambar 12. Diagram Activity Membuat Surat Keluar* 74](#_Toc433463634)

[*Gambar 13. Diagram Activity Memberi Perintah ke Kepala Unit* 75](#_Toc433463635)

[*Gambar 14. Diagram Activity Melakukan Verifikasi* 76](#_Toc433463636)

[*Gambar 15. Diagram Activity Mengisi Disposisi* 77](#_Toc433463637)

[*Gambar 16. Diagram Activity Memberi Paraf Koreksi* 78](#_Toc433463638)

[*Gambar 17. Diagram Activity Memberi Tanda Tangan* 79](#_Toc433463639)

[*Gambar 18. Diagram Activity Melihat Disposisi* 80](#_Toc433463640)

[*Gambar 19. Pemetaan Objek Login* 81](#_Toc433463641)

[*Gambar 20. Pemetaan Objek Input Data Surat Masuk* 81](#_Toc433463642)

[*Gambar 21. Pemetaan Objek Mengisi Disposisi* 81](#_Toc433463643)

[*Gambar 22. Pemetaan Objek Memberi Perintah ke Kepala Unit* 81](#_Toc433463644)

[*Gambar 23. Pemetaan Objek Menerima Notifikasi* 82](#_Toc433463645)

[*Gambar 24. Pemetaan Objek Membuat Surat Keluar* 82](#_Toc433463646)

[*Gambar 25. Pemetaan Objek Melakukan Verifikasi* 82](#_Toc433463647)

[*Gambar 26. Pemetaan Objek Logout* 82](#_Toc433463648)

[*Gambar 27. Pemetaan Objek Paraf Koreksi* 83](#_Toc433463649)

[*Gambar 28. Pemetaan Objek Tanda Tangan* 83](#_Toc433463650)

[*Gambar 29. Pemetaan Objek Melihat Disposisi* 83](#_Toc433463651)

[*Gambar 30. Diagram Sequence Login* 84](#_Toc433463652)

[*Gambar 31. Diagram Sequence Input Data Surat Masuk* 84](#_Toc433463653)

[*Gambar 32. Diagram Sequence Mengisi Disposis* 85](#_Toc433463654)

[*Gambar 33. Diagram Sequence Memberi Perintah ke Kepala Unit* 85](#_Toc433463655)

[*Gambar 34. Diagram Sequence Menerima Notifikasi* 86](#_Toc433463656)

[*Gambar 35. Diagram Sequence Membuat Surat Keluar* 86](#_Toc433463657)

[*Gambar 36. Diagram Sequence Melakukan Verifikasi* 86](#_Toc433463658)

[*Gambar 37. Diagram Sequence Logout* 87](#_Toc433463659)

[*Gambar 38. Diagram Sequence Memberi Paraf Koreksi* 87](#_Toc433463660)

[*Gambar 39. Diagram Sequence Memberi Tanda Tangan* 87](#_Toc433463661)

[*Gambar 40. Diagram Sequence Melihat Disposisi* 88](#_Toc433463662)

[*Gambar 41. ERD* 89](#_Toc433463663)

[*Gambar 42. Class Diagram* 90](#_Toc433463664)

[*Gambar 43. Deployment Diagram* 91](#_Toc433463665)

[*Gambar 44. Topologi Sistem* 92](#_Toc433463666)

[*Gambar 45. Component Diagram* 92](#_Toc433463667)

[*Gambar 46. Interface Login* 121](#_Toc433463668)

[*Gambar 47. Halaman Utama Administrasi* 122](#_Toc433463669)

[*Gambar 48. Form Input Data Surat Masuk* 122](#_Toc433463670)

[*Gambar 49. Form Input Data Surat Keluar* 123](#_Toc433463671)

[*Gambar 50. Menu Utama Ketua* 123](#_Toc433463672)

[*Gambar 51. Form Ketua Mengisi Disposisi* 124](#_Toc433463673)

[*Gambar 52. Menu Utama Assistant* 124](#_Toc433463674)

[*Gambar 53. Form Assistant Mengisi Disposisi Surat Masuk* 125](#_Toc433463675)

[*Gambar 54. Pembantu Ketua melakukan verifikasi surat keluar* 125](#_Toc433463676)

[*Gambar 55. Menu Utama Kepala Unit* 126](#_Toc433463677)

[*Gambar 56. Unit Melihat Disposisi* 126](#_Toc433463678)

[*Gambar 57. Unit Melakukan Verifikasi Surat Keluar* 127](#_Toc433463679)

[*Gambar 58. Flow graph – White Box Testing Login* 130](#_Toc433463680)

[*Gambar 59. Flow Grap – White Box Testing Input Data Surat Masuk* 131](#_Toc433463681)

# **Abstrak**

Sebagai sebuah instansi atau lembaga tentunya STIKOM Binaniaga melakukan proses surat menyurat, baik itu surat masuk maupun surat keluar. Surat yang disampaikan dapat berupa pemberitahuan, pernyataan, permintaan, laporan, dan sebagainya. Setup surat masuk yang diterima dan surat keluar yang dikirim baik itu oleh instansi swasta maupun pemerintah mempunyai nilai yang penting, baik itu sebagai alat komunikasi, dan dapat menunjukkan kegiatan suatu kantor atau instansi.

Didalam proses surat menyurat, terdapat proses pengarsipan yang juga menjadi sesuatu yang penting untuk diperhatikan. Dalam proses pengarsipan di STIKOM Binaniaga yang masih dilakukan secara konvensional, dapat mengakibatkan arsip – arsip yang ada akan menumpuk dan akan memakan tempat untuk menyimpannya. Terlebih lagi resiko yang akan diterima jika arsip tersebut rusak atau hilang.

Oleh karena itu perlu adanya Sistem yang dapat mengelola perihal tersebut yang menyangkut agendaris dan surat menyurat. Sistem informasi agendaris surat dan notifikasi menjadi Sistem yang diharapkan dapat menangani masalah tersebut. Dengan adanya Sistem tersebut diharapkan dapat membantu meningkatkan kinerja staff dari STIKOM Binaniaga khususnya di bidang administrasi yang menangani mengenai agendaris surat dan notifikasi.

**[HALAMAN INI SENGAJA DIKOSONGKAN]**

# **PENDAHULUAN**

# **Latar Belakang**

Pengelolaan surat pada STIKOM Binaniaga Bogor yaitu ada pada bagian BAUM yang mengurusi kegiatan administrasi termasuk mengurusi jalannya surat masuk dan surat keluar. Proses pengelolaan surat masuk pada penelitian kali yang dilakukan di STIKOM Binaniaga yaitu dari mulai penerimaan surat masuk yang diterima oleh bagian Administrasi, setelah itu dilakukan disposisi surat oleh staff bersangkutan, dan dikirimkan oleh pengirim ke Ketua Stikom lalu dikirim kembali ke bagian administrasi untuk memerintahkan surat itu ke bagian – bagian yang terlibat dalam perintah surat tersebut. Dan di dikirimkan kepada pihak pembantu ketua (puket) yang bersangkutan sesuai dengan sifat suratnya untuk dilakukan persetujuan, setelah di setujui oleh bagian puket maka surat. Jika surat telah di setujui oleh semua bagian yang terlibat maka surat akan di arsipkan secara konvensional di bagian administrasi.

Seiring dengan berkembangnya zaman, teknologi komunikasi berkembang begitu pesat dengan banyak bermunculnya berbagai alat telekomunikasi atau penghubung yang canggih seperti; internet, telepon, handphone, televisi, radio, telegram, faxsimile dan sebagainya, namun masih ada komunikasi tertulis yang tidak dapat dilupakan keberadaannya, bahkan masih tetap kokoh terpakai seolah tak bisa tergantikan oleh berbagai peralatan komunikasi yang canggih itu, komunikasi tertulis tersebut adalah surat.

CakePHP adalah sebuah php framework yang berfungsi sebagai kerangka kerja untuk mendevelopment sebuah Sistem sehingga Sistem dapat bekerja secara efektif dan lebih efisien. Dengan di kombinasikan dengah MySQL yang berperan sebagai tempat penyimpanan data ke dalam suatu database Sistem yang teratur untuk mengelola pengarsipan surat menyurat.

# **Masalah Proyek**

Dalam STIKOM Binaniaga proses surat menyurat dikelola oleh bagian Administrasi. Proses surat masuk di STIKOM Binaniaga dilakukan oleh bagian Administrasi, surat yang masuk akan diterima oleh staff di bagian Administrasi dan langsung di disposisi. Surat yang telah di disposisi akan dikirmkan kepada ketua STIKOM untuk meminta persetujuan melalui seorang pengirim surat, setelah meminta persetujuan dari ketua STIKOM surat akan dikirimkan melalui seorang pengirim surat kepada bagian Administrasi, yang nantinya surat itu akan dikirimkan lagi kepada bagian pembantu ketua yang bersangkutan dengan perihal surat tersebut melalui seorang pengirim surat. Setelah dari pembantu ketua surat akan dikirimkan lagi ke bagian administrasi agar perintah surat egera dilaksanakan dan dilakukan pengarsipan.

Dari proses – proses yang terjadi, proses pendistribusian surat masih dilakukan secara konvensional dengan menggunakan seorang untuk mengirim surat tersebut. Dan proses pengarsipan yang dilakukan pun masih konvensional sehingga proses tersebut jadi kurang efektif dan efisien.

Identifikasi Masalah   
- Lalu lintas pendistribusian surat kurang efektif dan efisien

- Penyimpanan data yang dilakukan masih kurang efektif

Pokok Masalah   
Mekanisme pendistribusian surat masih kurang efektif dan efisien, serta pengelolaan data yang masih kurang efektif

Rumusan Masalah   
- Bagaimana penerapan Framework CakePHP untuk agendaris surat dan notifikasi

* Bagaimana menerapkan MySQL untuk mengelola data arsip agar lebih terorganisir

Keterbatasan

Dalam sistem agendaris surat dan notifikasi ini proses pendistribusian surat untuk surat keluar masih dilakukan dengan cara konvensional.

# **Tujuan Proyek**

Tujuan pembuatan proyek agendaris surat dan notifikasi adalah:

* + 1. Menciptakan pelayanan untuk mengoptimalkan distribusi surat menyurat
    2. Menciptakan penglohan data ke dalam database agar lebih terorganisir

# **Signifikasi Proyek**

Mengalihkan proses pengelolaan data yang lama dimana proses tersebut masih dilakukan dengan cara konvensional menjadi proses pengelolaan data ke dalam database, agar nantinya data yang dikelola menjadi lebih terorganisir.

Sistem ini nantinya akan memberikan hasil yang bermanfaat dan membantu mengoptimalkan kinerja dari bagian akademik dalam pengarsipan.

Mempercepat proses surat menyurat masuk dan keluar di bagian administrasi

Kemudahan akses menggunakan tablet, smartphone, PC, Laptop.

Memudahkan pencarian data yang telah terarsip

Proses distribusi surat lebih efektif dan efisien

Pengelolaan data lebih terorganisir

# **PENDEKATAN PROYEK**

# **Feasibility (Kelayakan)**

## **Aspek Ekonomi**

Pengembangan suatu sistem informasi membutuhkan sumber-sumber daya yang ada pada lingkungan sebuah organisasi/institusi. Sebagai hasilnya, sistem informasi yang baru diharapkan akan memberikan manfaat-manfaat dan keuntungan-keuntungan yang dapat digunakakn oleh organisasi tersebut untuk menjalankan proses bisnis. Jika manfaat yang diharapkan lebih kecil dari sumber-sumber daya yang dikeluarkan, maka sistem informasi yang baru ini dikatakan tidak bernilai atau tidak layak. Oleh karena itu, sebelum sebuah sistem informasi dikembangkan, maka perlu dihitung kelayakan ekonomisnya, dengan teknik analisis biaya. Adapun tabel rincian dan manfaat untuk pengembangan Sistem Informasi Agendaris Surat dan

Notifikasi adalah sebagai berikut:

*Tabel 1. Biaya dan Manfaat Proyek*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| I | Biaya-biaya | Tahun 0 | Tahun 1 | Tahun 2 | Tahun 3 |
|  | **1. Biaya Pengadaan** |  |  |  |  |
|  | a) Biaya pembelian perangkat keras | 3.500.000 |  |  |  |
|  | b) Biaya instalasi perangkat keras | 300.000 |  |  |  |
|  | **total biaya pengembangan** | **3.800.000** |  |  |  |
|  | **2. Biaya Operasional** |  |  |  |  |
|  | a) Biaya perlengkapan perangkat lunak system | 1.000.000 |  |  |  |
|  | b) Biaya instalasi perangkat lunak | 200.000 |  |  |  |
|  | **total biaya operasional** | **1.200.000** |  |  |  |
|  | **3. Biaya Proyek** |  |  |  |  |
|  | a) Biaya Konsultan |  |  |  |  |
|  | 1. 1 orang analis system | 6.000.000 |  |  |  |
|  | 2. 1 orang programmer | 4.000.000 |  |  |  |
|  | 3. akomodasi dan transportasi | 500.000 |  |  |  |
|  | **total biaya persiapan operasional** | **10.500.000** |  |  |  |
|  | b) biaya analisis system |  |  |  |  |
|  | 1. biaya pengumpulan data | 300.000 |  |  |  |
|  | 2. biaya dokumentasi (kertas, fotocopy) | 500.000 |  |  |  |
|  | **total biaya analisis sistem** | 800.000 |  |  |  |
|  | c) biaya penerapan system |  |  |  |  |
|  | 1. biaya konversi data | 400.000 |  |  |  |
|  | 2. biaya pelatihan personil | 300.000 |  |  |  |
|  | **total biaya penerapan sistem** | 700.000 |  |  |  |
|  | **total biaya pembangunan sistem** | **12.000.000** |  |  |  |
|  | **4. biaya operasi dan perawatan** |  |  |  |  |
|  | a. biaya perawatan software |  | 1.500.000 | 1.700.000 | 1.900.000 |
|  | **total biaya operasi dan perawatan** |  | 1.500.000 | 1.700.000 | 1.900.000 |
|  | **total biaya-biaya** | **13.200.000** | 1.500.000 | 1.700.000 | 1.900.000 |

**Return of Investment**

ROI digunakan untuk mengukur presentase manfaat yang dihasilkan oleh proyek dibandingkan dengan biaya yang dikeluarkan. Atau besarnya keuntungan yang bisa diperoleh dalam bentuk persen (%) selama periode yang telah ditentukan.

Suatu proyek investasi yang mempunyai ROI > 0 adalah proyek yang dapat diterima. Pada nilai ROI proyek ini adalah 9.33% maka dapat dikatakan proyek ini dapat diterima, karena akan memberikan keuntungan sebesar 9.33%.

## **Aspek Operasional**

Permasalahan yang ada pada saat ini menyangkut aspek fungsi control dan pengawasan sangat berharga untuk diselesaikan.  
Berikut ini adalah beberapa identifikasi masalah yang berkaitan dengan operasional yang dapat diselesaikan oleh Sistem ini.

* Performance : Sistem ini dapat memberikan throughput dan waktu respons yang cukup.
* Information : Sistem ini menyediakan informasi terhormat yang tepat waktu, saling terkait, akurat dan berguna bagi staff administrasi.
* Economy : Sistem ini menawarkan tingkat dan kapasitas pelayanan yang memadai untuk mengurangi biaya operasional.
* Control : Sistem ini dapat mengcontrol dan mengkategorikan type surat.
* Efficiency : Sistem ini menggunakan teknologi yang dapat dijalankan secara mobile sehingga dapat menghemat waktu dan tempat pengerjaan.
* Service : Sistem ini menyediakan layanan yang diinginkan dan andal pada siapa saja.

## **Aspek Teknikal**

Kebutuhan Perangkat Keras

|  |  |  |
| --- | --- | --- |
| Processor | : | Pentium Core 2 Duo atau lebih |
| Hardisk | : | Min 80 GB |
| RAM | : | Min 1 GB |

Kebutuhan Perangkat Lunak

|  |  |  |
| --- | --- | --- |
| Operating Sistem | : | Windows 7 / Linux / Mac |
| Aplikasi | : | Mozilla Firefox / Chrome / IE |

## **Aspek Hukum**

Dalam hal pembuatan dan pengoperasian, semua kegiatan menggunakan software original yang tidak melanggar HAKI. Hal ini turut membantu program pemerintah dalam menekan angka pembajakan yang cukup tinggi dinegara ini.

## **Aspek Kultural**

Dengan adanya sebuah Sistem terkomputerisasi maka pekerjaan membuat Surat Pertanggun Jawaban menjadi lebih mudah. Ini bias meningkatkan semangat kerja terutama kepada petugas yang bertanggung jawab terhadap pembuatan Surat Pertanggung Jawaban.

## **Kelayakan Jadwal**

Dalam melakukan proyek memiliki tahapan – tahapan kegiatan untuk menyelesaikan sebuah proyek. Seperti yang dijelaskan pada Tabel 2 berikut.

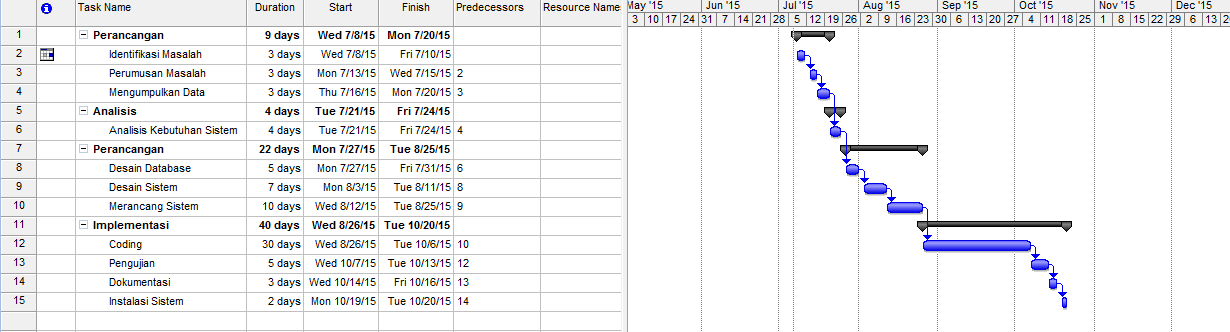
*Table 2. Data Aktifitas Proyek*

|  |  |  |
| --- | --- | --- |
| **No** | **Kegiatan** | **Id Kegiatan** |
| 1 | Perencanaan | 1 |
|  | Identifikasi Masalah | 1.1 |
|  | Perumusan Masalah | 1.2 |
|  | Mengumpulkan Masalah | 1.3 |
| 2 | Analisis | 2 |
|  | Analisis Kebutuhan Sistem | 2.1 |
| 3 | Perancangan | 3 |
|  | Desain Database | 3.1 |
|  | Desain Sistem | 3.2 |
|  | Merancang Sistem | 3.3 |
| 4 | Implementasi | 4 |
|  | Coding | 4.1 |
|  | Pengujian | 4.2 |
|  | Dokumentasi | 4.3 |
|  | Instalasi Sistem | 4.4 |

*Table 3. Waktu yang diharapkan dari proyek (Expected Time) ET*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Id Kegiatan** | **Wasktu Optimis (a)** | **Waktu Realistis (m)** | **Waktu Pesimis (b)** | **Waktu yang Diharapkan (ET)** |
| 1 | 1.1 | 2 | 3 | 4 | 3.00 |
| 2 | 1.2 | 2 | 3 | 4 | 3.00 |
| 3 | 1.3 | 2 | 3 | 4 | 3.00 |
| 4 | 2.1 | 3 | 4 | 6 | 4.17 |
| 5 | 3.1 | 4 | 5 | 6 | 5.00 |
| 6 | 3.2 | 6 | 7 | 9 | 7.17 |
| 7 | 3.3 | 7 | 10 | 15 | 10.33 |
| 8 | 4.1 | 27 | 30 | 35 | 30.33 |
| 9 | 4.2 | 4 | 5 | 6 | 5.00 |
| 10 | 4.3 | 2 | 3 | 4 | 3.00 |
| 11 | 4.4 | 1 | 2 | 3 | 2.00 |
|  | Total | 60 | 75 | 96 | 76.00 |

# **Penjadwalan**



*Gambar 1. Chart Gant penjadwalan proyek*

# **Pemikiran Teoritis**

CakePHP Framework sebagai sebuah kerangka kerja antara lain karena CakePHP bersifat open source yang berlisensi MIT dimana framework ini sesuai untuk aplikasi komersil. CakePHP juga memiliki keamanan aplikasi yang cukup aman. Karena CakePHP memiliki fasilitas CSRF Protection dimana dapat menghindari atau menolak SQL Injection dan XSS Attack.  
Dan alas an memilih MySQL sebagai sebuah DBMS (Data Base Management Sistem ) antara lain karena MySQL dapat bekerja platform manapun selain itu MySQL juga bersifat open source.   
  
Dalam merancang Sistem ini akan menggunakan konsep OOP atau Object Oriented Programming.

Kelebihan dari penggunaan OOP ini adalah :

* Konsep ini menyederhanakan kompleksitas dengan memungkinkan kita untuk mendefinisikan sebuah sistem besar dan kompleks menggunakan set yang lebih kecil dari objek yang saling terkait.
* Dengan merancang aplikasi menggunakan objek, berarti kita telah menerapkan permodelan yang mendekati kehidupan nyata. Hal ini memungkinkan desain program kita menjadi lebih alamiah , yang memungkinkan kita untuk bekerja lebih intuitif.

Menyangkut masalah keamanan, setiap password user akan di enkripsi menggunakan teknik Security Salt dimana sebuah kata random dalam methode Security Hashing. Hal ini untuk mencegah terjadi penyalahgunaan ID User yang dilakukan oleh Administrator Database.

**Definisi:** Framework dapat didefinisikan dalam berbagai cara:  
- Suatu struktur konseptual dasar yang digunakan untuk memecahkan atau menangani suatu masalah kompleks

- Sebuah software untuk memudahkan para programmer membuat aplikasi atau web yang isinya adalah berbagai fungsi, plugin, dan konsep sehingga membentuk suatu sistem tertentu.

- Framework (kerangka kerja) PHP berbasis-komponen, berkinerja tinggi untuk pengembangan aplikasi Web berskala-besar.

# **Kerangka Pemecahan**

Pada metode pendekatan sistem menggunakan metode *Object Oriented* (OO) karena sistem yang sebelumnya digunakan masih belum jelas, sehingga dengan menggunakan metode ini dapat memfokuskan kepada sistem informasi yang akan dibangun

# **Konsepsi Pengembangan**

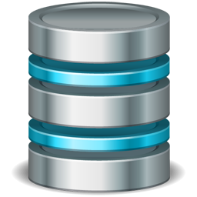
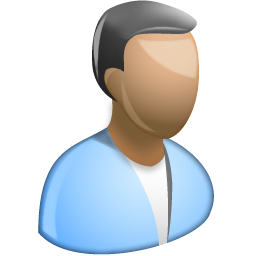
Platform yang dipilih untuk membangun Sistem ini yaitu platform website. Dengan menggunakan framework CakePHP sebagai bahasa pemrograman yang berbasis PHP dan Bootstrap v3.2.0 sebagai Framework CSS untuk tampilan website nya. Selain itu digunakan juga jquery , dan AJAX untuk membantu membangun tmapilan website.

# **Analisis**

1. Pengumpulan Data
   1. Observasi, melakukan pengamatan mengenai bagaimana proses surat masuk dan surat keluar yang berlangsung di STIKOM Binaniaga Bogor mulai dari membuat surat keluar dan mengisi disposisi untuk surat mausk.
   2. Wawancara, bertanya kepada pihak yang bersangkutan mengenai agendaris dan surat dalam hal ini yaitu bagian administrasi umum atau BAUM mengenai apa saja data yang dibutuhkan selama proses surat menyurat di STIKOM Binaniaga Bogor.
2. Analisis Data
   1. Menganalisa data yang digunakan dalam proses surat menyurat
   2. Menganalisa bentuk laporan yang tersedia

# **PEMODELAN DAN STRUKTUR**

# **Arsitektur**



Database

User



Sistem pengajuan proyek minor

*Gambar 2. Bagan cara kerja Sistem*

## **Hardware**

Berdasarkan dengan kebutuhan sistem maka melalui tabel 6 berikut akan dijelaskan mengenai kebutuhan *hardware*

|  |  |
| --- | --- |
| **Item Hardware** | **Spesifikasi** |
| PC Server | Intel Core 2 Duo, RAM 2 GB, 500GB HDD |
| PC Client | Intel Core 2 Duo, RAM 1 GB, 500GB HDD |

*Tabel 4. Spesifikasi Hardware*

## **Software**

Kebutuhan software yang diperlukan untuk menjalankan Sistem ini dapat dilihat pada table berikut.

*Tabel 5. SPesifikasi Software*

|  |  |
| --- | --- |
| **Item Software** | **Spesifikasi** |
| Operating Sistem | Windows 7 – 32 Bit |
| Aplikasi | Mozilla Firefox, Chrome, IE |

# **Komponen Pengembangan**

## **Proses Bisnis**



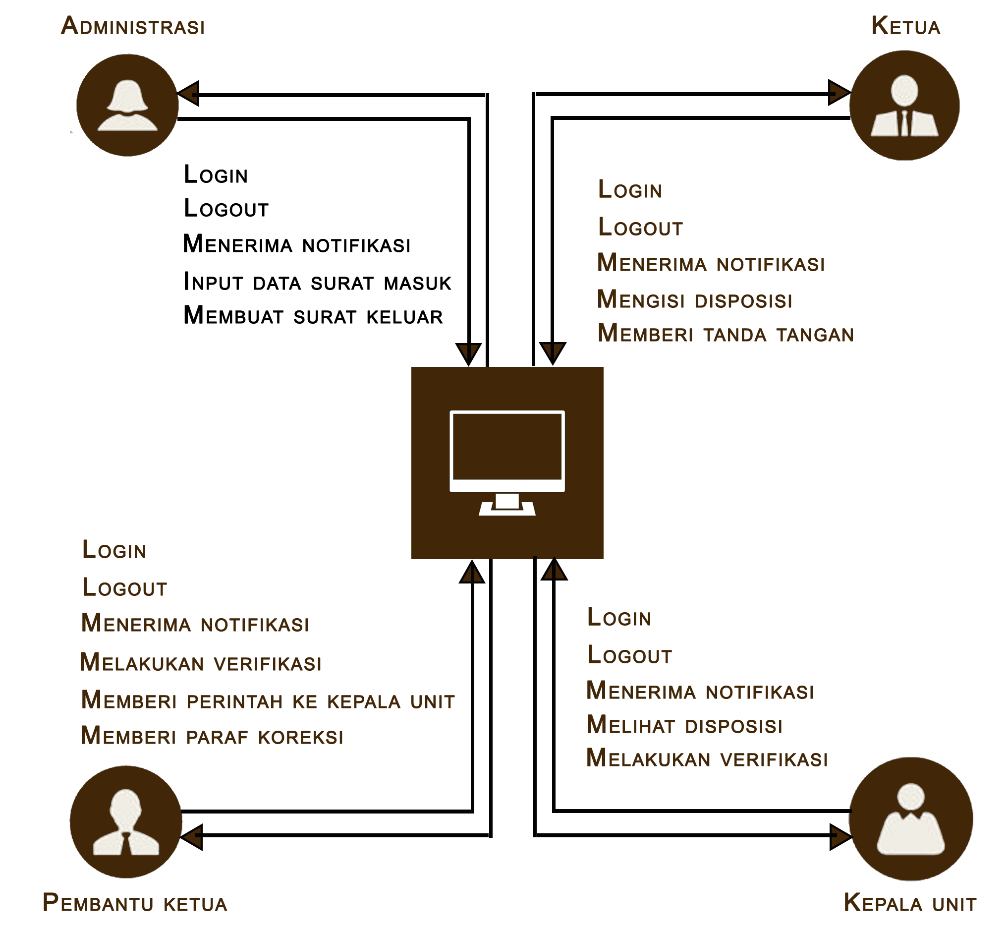
*Gambar 3. Proses Bisnis Surat Masuk*

**

*Gambar 4. Proses Bisnis Surat Keluar Dalam Instansi*

*Gambar 5. Proses Bisnis Surat Keluar Luar Instansi*

## **Diagram Konteks**



*Gambar 6. Diagram Konteks*

Diagram Konteks diatas dapat dijelaskan sebagai berikut :

* + - 1. Administrasi, menginput data surat masuk ke dalam Sistem dan mengirim nya kepada ketua untuk dilakukan disposisi, selain itu administrasi juga bertugas untuk membuat surat keluar dan mengirimnya kepada pembantu ketua atau kepala unit untuk di verifikasi.
      2. Ketua menerima notifikasi bahwa ada surat masuk dari administrasi yang harus di disposisi, Ketua mengisi disposisi surat masuk, Ketua memberi notifikasi balik ke admin, Ketua memberi tanda tangan untuk surat keluar dengan melakukan verifikasi di dalam sistem.
      3. Pembantu Ketua menerima notifikasi bahwa ada surat masuk yang harus di disposisi, Pembantu Ketua memberi perintah ke kepala unit dengan mengisi disposisi, Pembantu Ketua memberi notifikasi balik ke admin, Pembantu Ketua melakukan verifikasi untuk surat keluar dalam instansi, Pembantu Ketua memberi paraf koreksi untuk surat keluar luar instansi dengan melakukan verifikasi di dalam sistem.
      4. Kepala Unit menerima notifikasi bahwa ada surat masuk dan memberikan notifikasi balik ke admin, selain itu juga kepala unit dapat melakukan verifikasi untuk surat keluar dalam instansi.

## **Diagram Usecase**



*Gambar 7. Diagram Usecase*

* **Identifikasi Aktor**

Berdasarkan use case yang digambarkan pada gambar 4, maka dapat di identifikasi deskripsi dari tiap actor pada gambar tersebut dengan table dibawah ini.

*Tabel 6. Identifikasi Aktor*

|  |  |
| --- | --- |
| **Aktor** | **Deskripsi** |
| Administrasi | Aktor yang bertugas input data surat masuk dan membuat surat keluar. |
| Ketua | Aktor yang bertugas mengisi disposisi surat masuk, dan memberikan tanda tangan untuk surat keluar luar instansi. |
| Pembantu Ketua | Aktor yang bertugas mengisi disposisi surat masuk dan memverifikasi surat keluar serta memberikan paraf koreksi. |
| Kepala Unit | Aktor yang bertugas memberi notifikasi balik kepada administrasi dan melakukan verifikasi surat keluar. |

* **Identifikasi Usecase**

*Tabel 7. Identifikasi Usecase*

|  |  |  |
| --- | --- | --- |
| **Nama Usecase** | **Deskripsi** | **Pelaku Utama** |
| Login | Use case ini mendeskripsikan tentang kegiatan yang dilakukan user dalam menginputkan userid dan password yang sudah ditentukan sebelumnya. | User |
| Input Data Surat Masuk | Use case ini mendeskripsikan tentang kegiatan yang dilakukan oleh Administrasi untuk menginput data surat masuk yang diterimanya. | Administrasi |
| Membuat Surat Keluar | Use case ini mendeskripsikan tentang kegiatan yang dilakukan bagian Administrasi untuk membuat surat keluar. | Administrasi |
| Mengisi Disposisi | Use case ini mendeskripsikan tentang kegiatan yang dilakukan oleh Ketua untuk mengisi disposisi surat. | Ketua |
| Memberi Perintah ke Kepala Unit | Use case ini mendeskripsikan tentang kegiatan yang dilakukan oleh Pembantu Ketua untuk memberikan perintah kepada kepala unit. | Pembantu Ketua |
| Melihat Disposisi | Use case ini mendeskripsikan tentang kegiatan Kepala Unit untuk melihat lembar disposisi yang dikirimkan kepada nya. | Kepala Unit |
| Melakukan Verifikasi | Use case ini mendiskripsikan tentang kegiatan yang dilakukan oleh Pembantu Ketua dan Kepala Unit untuk melakukan verifikasi terhadap surat keluar yang akan di kirim. | Pembantu Ketua, Kepala Unit |
| Memberi Paraf Koreksi | Use case ini mendeskripsikan tentang kegiatan yang dilakukan oleh Pembantu Ketua untuk memberikan paraf koreksi atau notifikasi dalam sistem terhadap surat keluar. | Pembantu Ketua |
| Memberi Tanda Tangan | Use case ini mendeskripsikan tentang kegiatan yang dilakukan oleh Ketua untuk memberikan tanda tangan atau notifikasi dalam sistem terhadap surat keluar | Ketua |
| Menerima Notifikasi | Use case ini mendeskripsikan tentang kegiatan yang dilakukan oleh User yang menerima notifikasi. | User |
| Logout | Use case ini mendeskripsikan tentang kegiatan yang dilakukan oleh User untuk keluar dari Sistem. | User |

## **Matriks Ranking**

*Tabel 8. Matriks Ranking*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Nama Usecase** | **Kriteria rangking** | | | | | | **Skor total** | **prioritas** | **Siklus pembangunan** |
| **1 s.d 5** | | | | | |
|  | **1** | **2** | **3** | **4** | **5** | **6** |  |  |  |
| Login | 5 | 5 | 5 | 1 | 5 | 1 | 22 | Tinggi | 1 |
| Input Data Surat Masuk | 5 | 5 | 5 | 1 | 5 | 5 | 26 | Tinggi | 2 |
| Membuat Surat Keluar | 5 | 5 | 5 | 1 | 5 | 5 | 26 | Tinggi | 2 |
| Mengisi Disposisi | 5 | 5 | 5 | 1 | 5 | 5 | 26 | Tinggi | 2 |
| Memberi Perintah ke Kepala Unit | 3 | 4 | 4 | 1 | 5 | 4 | 21 | Tinggi | 2 |
| Melakukan Verifikasi | 4 | 4 | 4 | 1 | 1 | 1 | 15 | Tinggi | 4 |
| Memberi Paraf Koreksi | 3 | 4 | 4 | 1 | 5 | 5 | 22 | Tinggi | 2 |
| Melihat Disposisi | 4 | 3 | 3 | 1 | 1 | 3 | 16 | Tinggi | 2 |
| Memberi Tanda Tangan | 3 | 4 | 4 | 1 | 5 | 5 | 22 | Tinggi | 2 |
| Menerima Notifikasi | 5 | 5 | 4 | 1 | 5 | 5 | 25 | Tinggi | 2 |
| Logout | 4 | 4 | 4 | 1 | 1 | 1 | 15 | Tinggi | 4 |
| Kriteria Ranking (Craig Larman) :   1. Dampak signifikasi pada design arsitektur 2. Mudah diimplementasikan, tetapi berisi fungsi signifikan 3. Mencantumkan resiko, waktu kritis, atau fungsi kompleks 4. Melibatkan penyelidikan yang signifikan atau baru, atau teknologi yang beresiko 5. Menyertakan fungsi bisnis utama 6. Akan meningkatkan pendapatan atau mengurangi biaya | | | | | | | | | |

## **Usecase Narative**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pengarang | : | Ogiano Waskitajaya | Tanggal | : | 14 Agustus 2015 |
|  |  | Yanuar Nurcahyo | Versi | : | 1.0 |

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Login | Type Use Case |
| ID Use Case | ASN01 | Persyaratan Bisnis Description: Description: Description: BD21301_ |
| Prioritas | Tinggi |
| Sumber | - |
| Pelaku Bisnis Utama | User |
| Pelaku Partisipan Lain | Sistem | |
| Stakeholder yang berminat lain | - | |
| Deskripsi | Use case ini mendeskripsikan tentang kegiatan yang dilakukan user dalam menginputkan userid dan password yang sudah ditentukan sebelumnya. | |
| Prakondisi | User menginputkan Userid dan password ke dalam sistem. | |
| Sasaran | Use Case ini diawali pada saat sistem akan melakukan validasi. | |
| Bidang khas suatu event | **Kegiatan Pelaku** | **Respons Sistem** |
|  | **Langkah 1:**  User menyiapkan userid dan password untuk masuk ke dalam sistem. | **Langkah 2:**  Sistem memvalidasi userid dan password apabila sesuai maka sistem akan melanjutkan ke langkah berikutnya. |
| Bidang Alternatif | **Alt langkah 2: Jika userid dan password tidak sesuai maka User kembali melakukan login sampai sesuai** | |
| Kesimpulan | Use Case ini menyimpulkan tentang tahap-tahap validasi yang harus dilakukan oleh user sebelum diterima oleh sistem. | |
| Aturan Bisnis | User login dengan menggunakan userid dan password yang disesuaikan ID masing-masing. User dapat merubah password jika dihendaki. Apabila user id dan password tidak sesuai maka user tidak dapat mengakses kedalam sistem | |
| Batasan dan spesifikasi implementasi | Validasi harus terpenuhi, jika salah satu tidak terpenuhi, maka sistem akan menolaknya. | |
| Asumsi | User harus tepat dalam menginputkan userid dan password. | |
| Pasca Kondisi | Sistem akan menampilkan menu-menu pilihan bagi setiap user yang login. Menu pilihan yang ditampilkan tergantung userid dan password yang dimasukkan. | |
| Masalah Terbuka | User lupa password dan cara penulisan password. | |

*Table 9. Use Case Naratif Persyaratan Bisnis Login*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pengarang | : | Ogiano Waskitajaya | Tanggal | : | 14 Agustus 2015 |
|  |  | Yanuar Nurcahyo | Versi | : | 1.0 |

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Login | Type Use Case |
| ID Use Case | ASN01 | Analisis Sistem Description: Description: Description: BD21301_ |
| Prioritas | Tinggi |
| Sumber | - |
| Pelaku Bisnis Utama | User |
| Pelaku Partisipan Lain | Sistem | |
| Stakeholder yang berminat lain | - | |
| Deskripsi | Use case ini mendeskripsikan tentang kegiatan yang dilakukan user dalam menginputkan userid dan password yang sudah ditentukan sebelumnya. | |
| Prakondisi | User menginputkan Userid dan password ke dalam sistem. | |
| Pemicu | Use Case ini dilakukan pertama kali saat user masuk ke dalam sistem. Agar tidak terjadinya kesalahan pengecekan akses ke dalam sistem. | |
| Bidang khas suatu event | **Kegiatan Pelaku** | **Respons Sistem** |
|  | **Langkah 1:**  User masuk form login.  **Langkah 2:**  User menginput userid dan password | **Langkah 3:**  Sistem memvalidasi userid dan password jika userid dan password sesuai sistem akan melanjutkan ke langkah berikutnya  **Langkah 4:**  Sistem menampilkan menu utama |
| Bidang Alternatif | **Alt langkah 3: Jika sistem menolak validasi login maka User kembali menginputkan userid dan password.** | |
| Kesimpulan | Use Case ini menyimpulkan tentang tahap-tahap validasi yang harus dilakukan oleh user sebelum diterima oleh sistem. | |
| Pasca kondisi | Sistem akan menampilkan menu-menu pilihan bagi setiap user yang login. Menu pilihan yang ditampilkan adalah menu yang akan dijadikan pemasukkan. | |
| Aturan Bisnis | User login dengan menggunakan userid dan password yang disesuaikan ID masing-masing. User dapat merubah password jika dihendaki. Apabila user id dan password tidak sesuai maka user tidak dapat mengakses kedalam sistem | |
| Batasan dan spesifikasi implementasi | - Validasi harus terpenuhi  - Jika salah satu ada yang tidak terpenuhi, sistem akan  menolaknya | |
| Asumsi | User harus tepat dalam menginputkan userid dan password. | |
| Masalah Terbuka | User lupa password dan cara penulisan password. | |

*Table 10. Use Case Naratif Analisis SIstem Login*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pengarang | : | Ogiano Waskitajaya | Tanggal | : | 14 Agustus 2015 |
|  |  | Yanuar Nurcahyo | Versi | : | 1.0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nama Use Case | Login | | Type Use Case | |
| ID Use Case | ASN01 | | Design sistem Description: Description: Description: BD21301_ | |
| Prioritas | Tinggi | |
| Sumber | - | |
| Pelaku Bisnis Utama | User | |
| Pelaku Partisipan Lain | Sistem | | | |
| Stakeholder yang berminat lain | - | | | |
| Deskripsi | Use case ini mendeskripsikan tentang kegiatan yang dilakukan user dalam menginputkan userid dan password yang sudah ditentukan sebelumnya. | | | |
| Prakondisi | User menginputkan User Id dan password ke dalam sistem. | | | |
| Pemicu | Use Case ini dilakukan pada saat user masuk pertama kali ke sistem agar tidak adanya kesalahan pengecekan akses ke dalam sistem. | | | |
| Bidang khas suatu event | **Kegiatan Pelaku** | | | **Respons Sistem** |
|  | **Langkah 1:**  User masuk form login  **Langkah 2:**  User menginputkan userid dan password pada textbox.  **Langkah 4:**  User menekan tombol login. | | | **Langkah 3:**  Sistem memvalidasi userid dan password jika userid dan password sesuai sistem akan melanjutkan ke langkah berikutnya.  **Langkah 5:**  Sistem menampilkan Form menu utama. |
| Bidang Alternatif | | **Alt langkah 3:**  **Jika sistem menolak validasi login maka akan tampil kotak dialog pesan lalu user kembali menginputkan userid dan password sampai userid dan password sesuai** | | |
| Kesimpulan | | Use Case ini menyimpulkan tentang tahap-tahap validasi yang harus dilakukan sebelum diterima dan masuk ke dalam sistem. | | |
| Pasca kondisi | | Sistem akan menampilkan menu-menu pilihan bagi setiap user yang login. Menu pilihan yang ditampilkan tergantung user id dan password yang dimasukkan. | | |
| Aturan Bisnis | | User login dengan menggunakan user id dan password yang disesuaikan ID masing-masing. User dapat merubah password jika dihendaki. Apabila user id dan password tidak sesuai maka user tidak dapat mengakses kedalam sistem | | |
| Batasan dan spesifikasi implementasi | | - Validasi harus terpenuhi  - Jika salah satu ada yang tidak terpenuhi, sistem akan menolaknya | | |
| Asumsi | | User harus tepat dalam menginputkan userid dan password | | |
| Masalah Terbuka | | User lupa password dan cara penulisan password | | |

*Table 11. Use Case Naratif Design Sistem Login*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pengarang | : | Ogiano Waskitajaya | Tanggal | : | 14 Agustus 2015 |
|  |  | Yanuar Nurcahyo | Versi | : | 1.0 |

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Logout | Type Use Case |
| ID Use Case | ASN02 | Persyaratan Bisnis Description: Description: Description: BD21301_ |
| Prioritas | Tinggi |
| Sumber | - |
| Pelaku Bisnis Utama | User |
| Pelaku Partisipan Lain | Sistem | |
| Stakeholder yang berminat lain | - | |
| Deskripsi | Use case ini mendeskripsikan tentang kegiatan yang dilakukan oleh User untuk keluar dari Sistem. | |
| Prakondisi | User telah selesai melakukan segala keperluannya dengan sistem. | |
| Sasaran | User dapat keluar dari sistem | |
| Bidang khas suatu event | **Kegiatan Pelaku** | **Respons Sistem** |
|  | **Langkah 1:**  User memilih tombol logout. | **Langkah 2:**  Sistem mengarahkan user kembali ke halaman login. |
| Bidang Alternatif | **-** | |
| Kesimpulan | Use Case ini menyimpulkan tentang tahap-tahap yang perlu dilakukan oleh user untuk keluar dari sistem. | |
| Aturan Bisnis | Memastikan kegiatan terakhir user sebelum melakukan logout sudah tersimpan ke dalam database. | |
| Batasan dan spesifikasi implementasi | Menu logout diperuntukan bagi user yang memiliki username dan password valid yang telah melakukan login ke dalam sistem. | |
| Asumsi | User dapat melakukan logout. | |
| Pasca Kondisi | Sistem akan menampilkan kembali menu login setelah proses logout dilakukan | |
| Masalah Terbuka | User lupa untuk melakukan logout setelah melakukan kegiatannya | |

*Table 12. Use Case Naratif Persyaratan Bisnis Logout*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pengarang | : | Ogiano Waskitajaya | Tanggal | : | 14 Agustus 2015 |
|  |  | Yanuar Nurcahyo | Versi | : | 1.0 |

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Logout | Type Use Case |
| ID Use Case | ASN02 | Analisis Sistem Description: Description: Description: BD21301_ |
| Prioritas | Tinggi |
| Sumber | - |
| Pelaku Bisnis Utama | User |
| Pelaku Partisipan Lain | Sistem | |
| Stakeholder yang berminat lain | - | |
| Deskripsi | Use case ini mendeskripsikan tentang kegiatan yang dilakukan oleh User untuk keluar dari Sistem. | |
| Prakondisi | User telah selesai melakukan segala keperluannya dengan sistem. | |
| Pemicu | User menekan logout | |
| Bidang khas suatu event | **Kegiatan Pelaku** | **Respons Sistem** |
|  | **Langkah 1:**  User memilih logout | **Langkah 2:**  Sistem merespon dengan menghapus session login user  **Langkah 3:**  Sistem mengarahkan user ke halaman login |
| Bidang Alternatif | **-** | |
| Kesimpulan | Use Case ini menyimpulkan tentang tahap-tahap yang perlu dilakukan oleh user untuk keluar dari sistem. | |
| Pasca kondisi | Memastikan kegiatan terakhir user sebelum melakukan logout sudah tersimpan ke dalam database. | |
| Aturan Bisnis | Menu logout diperuntukan bagi user yang memiliki username dan password valid yang telah melakukan login ke dalam sistem. | |
| Batasan dan spesifikasi implementasi | User dapat melakukan logout. | |
| Asumsi | Sistem akan menampilkan kembali menu login setelah proses logout dilakukan | |
| Masalah Terbuka | User lupa untuk melakukan logout setelah melakukan kegiatannya | |

*Table 13. Use Case Naratif Analisis Sistem Logout*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pengarang | : | Ogiano Waskitajaya | Tanggal | : | 14 Agustus 2015 |
|  |  | Yanuar Nurcahyo | Versi | : | 1.0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nama Use Case | Logout | | Type Use Case | |
| ID Use Case | ASN02 | | Design sistem Description: Description: Description: BD21301_ | |
| Prioritas | Tinggi | |
| Sumber | - | |
| Pelaku Bisnis Utama | User | |
| Pelaku Partisipan Lain | Sistem | | | |
| Stakeholder yang berminat lain | - | | | |
| Deskripsi | Use case ini mendeskripsikan tentang kegiatan yang dilakukan oleh User untuk keluar dari Sistem. | | | |
| Prakondisi | User telah selesai melakukan segala keperluannya dengan sistem. | | | |
| Pemicu | User menekan tombol logout | | | |
| Bidang khas suatu event | **Kegiatan Pelaku** | | | **Respons Sistem** |
|  | **Langkah 1:**  User memilih logout dengan menekan tombol logout | | | **Langkah 2:**  Sistem merespon dengan menghapus session login user  **Langkah 3:**  Sistem mengarahkan user ke halaman login |
| Bidang Alternatif | | **-** | | |
| Kesimpulan | | Use Case ini menyimpulkan tentang tahap-tahap yang perlu dilakukan oleh user untuk keluar dari sistem. | | |
| Pasca kondisi | | Memastikan kegiatan terakhir user sebelum melakukan logout sudah tersimpan ke dalam database. | | |
| Aturan Bisnis | | Menu logout diperuntukan bagi user yang memiliki username dan password valid yang telah melakukan login ke dalam sistem. | | |
| Batasan dan spesifikasi implementasi | | User dapat melakukan logout. | | |
| Asumsi | | Sistem akan menampilkan kembali menu login setelah proses logout dilakukan | | |
| Masalah Terbuka | | User lupa untuk melakukan logout setelah melakukan kegiatannya | | |

*Table 14. Use Case Naratif Design Sistem Logout*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Pengarang | : | Ogiano Waskitajaya | | Tanggal | | : | 14 Agustus 2015 |
|  |  | Yanuar Nurcahyo | | Versi | | : | 1.0 |
|  |  |  | |  | |  |  |
| Nama Use Case | | | | Menerima Notifikasi | | Type Use Case | | |
| ID Use Case | | | | ASN03 | | Persyaratan Bisnis Description: Description: Description: BD21301_ | | |
| Prioritas | | | | Tinggi | |
| Sumber | | | | - | |
| Pelaku Bisnis Utama | | | | User | |
| Pelaku Partisipan Lain | | | | Sistem | | | | |
| Stakeholder yang berminat lain | | | | - | | | | |
| Deskripsi | | | | Use case ini mendeskripsikan tentang kegiatan yang dilakukan oleh User yang menerima notifikasi. | | | | |
| Prakondisi | | | | User harus sudah melakukan login ke menu utama. | | | | |
| Sasaran | | | | Use case ini dilakukan agar user dapat mengetahui informasi terbaru yang masuk. | | | | |
| Bidang khas suatu event | | | | **Kegiatan Pelaku** | | **Respon Sistem** | | |
|  | | | | **Langkah 1 :**  User melakukan login kedalam Sistem. | | **Langkah 2 :**  Sistem memberikan notifikasi apabila ada informasi baru yang masuk. | | |
| Bidang Alternatif | | | | - | | | | |
| Kesimpulan | | | | Use case ini menyimpulkan kegiatan user dalam menerima notifikasi dalam Sistem. | | | | |
| Pasca Kondisi | | | | Sistem akan mengirimkan notifikasi kepada user telah melakukan login. | | | | |
| Aturan Bisnis | | | | User harus sudah melakukan login terlebih dahulu, dan ada informasi baru yang masuk. | | | | |
| Batasan dan spesifikasi implementasi | | | | Sistem akan memberikan respon apabila ada informasi baru yang masuk. | | | | |
| Asumsi | | | | Sistem akan mengirimkan notifikasi bila ada pesan baru yang masuk. | | | | |
| Masalah terbuka | | | | - | | | | |

*Table 15. Use Case Naratif Persyaratan Bisnis Menerima Notifikasi*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pengarang | : | Ogiano Waskitajaya | Tanggal | : | 14 Agustus 2015 |
|  |  | Yanuar Nurcahyo | Versi | : | 1.0 |

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Menerima Notifikasi | Type Use Case |
| ID Use Case | ASN03 | Analisis Sistem Description: Description: Description: BD21301_ |
| Prioritas | Tinggi |
| Sumber | - |
| Pelaku Bisnis Utama | User |
| Pelaku Partisipan Lain | Sistem | |
| Stakeholder yang berminat lain | - | |
| Deskripsi | Use case ini mendeskripsikan tentang kegiatan yang dilakukan oleh User yang menerima notifikasi. | |
| Prakondisi | User harus sudah melakukan login ke menu utama. | |
| Pemicu | User masuk ke menu utama | |
| Bidang khas suatu event | **Kegiatan Pelaku** | **Respon Sistem** |
|  | **Langkah 1 :**  User masuk halaman utama.  **Langkah 3 :**  User membuka notifikasi. | **Langkah 2:**  Sistem memberikan notifikasi apabila ada pesan baru yang masuk. |
| Bidang Alternatif | - | |
| Kesimpulan | Use case ini menyimpulkankegiatan user dalam menerima notifikasi dalam Sistem. | |
| Pasca Kondisi | Sistem akan mengirimkan notifikasi kepada user telah melakukan login. | |
| Aturan Bisnis | User harus sudah melakukan login terlebih dahulu, dan ada informasi baru yang masuk. | |
| Batasan dan spesifikasi implementasi | Sistem akan memberikan respon apabila ada informasi baru yang masuk. | |
| Asumsi | Sistem akan mengirimkan notifikasi bila ada pesan baru yang masuk. | |
| Masalah terbuka | - | |

*Table 16. Use Case Naratif Analisis Sistem Menerima Notifikasi*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pengarang | : | Ogiano Waskitajaya | Tanggal | : | 14 Agustus 2015 |
|  |  | Yanuar Nurcahyo | Versi | : | 1.0 |

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Menerima Notifikasi | Type Use Case |
| ID Use Case | ASN03 | Design Sistem Description: Description: Description: BD21301_ |
| Prioritas | Tinggi |
| Sumber | - |
| Pelaku Bisnis Utama | User |
| Pelaku Partisipan Lain | SIstem | |
| Stakeholder yang berminat lain | - | |
| Deskripsi | Use case ini mendeskripsikan tentang kegiatan yang dilakukan oleh User yang menerima notifikasi. | |
| Prakondisi | User harus sudah melakukan login ke menu utama. | |
| Pemicu | User masuk ke menu utama | |
| Bidang khas suatu event | **Kegiatan Pelaku** | **Respon Sistem** |
|  | **Langkah 1 :**  User masuk halaman utama.  **Langkah 3 :**  User membuka notifikasi dengan menekan icon notifikasi. | **Langkah 2:**  Sistem memberikan notifikasi apabila ada pesan baru yang masuk. |
| Bidang Alternatif | - | |
| Kesimpulan | Use case ini menyimpulkankegiatan user dalam menerima notifikasi dalam Sistem. | |
| Pasca Kondisi | Sistem akan mengirimkan notifikasi kepada user telah melakukan login. | |
| Aturan Bisnis | User harus sudah melakukan login terlebih dahulu, dan ada informasi baru yang masuk. | |
| Batasan dan spesifikasi implementasi | Sistem akan memberikan respon apabila ada informasi baru yang masuk. | |
| Asumsi | Sistem akan mengirimkan notifikasi bila ada pesan baru yang masuk. | |
| Masalah terbuka | - | |

*Table 17. Use Case Naratif Design Sistem Menerima Notifikasi*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pengarang | : | Ogiano Waskitajaya | Tanggal | : | 14 Agustus 2015 |
|  |  | Yanuar Nurcahyo | Versi | : | 1.0 |

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Input Data Surat Masuk | Type Use Case |
| ID Use Case | ASN04 | Persyaratan Bisnis Description: Description: Description: BD21301_ |
| Prioritas | Tinggi |
| Sumber | - |
| Pelaku Bisnis Utama | Administrasi |
| Pelaku Partisipan Lain | SIstem | |
| Stakeholder yang berminat lain | - | |
| Deskripsi | Use case ini mendeskripsikan tentang kegiatan yang dilakukan oleh Administrasi untuk menginput data surat masuk yang diterimanya. | |
| Prakondisi | Bagian Administrasi harus sudah menerima surat masuk terlebih dahulu. | |
| Sasaran | Administrasi dapat menginput data surat masuk. | |
| Bidang khas suatu event | **Kegiatan Pelaku** | **Respon Sistem** |
|  | **Langkah 1 :**  Bagian Administrasi menyiapkan data yang akan dimasukkan ke dalam sistem. | **Langkah 2 :**  Sistem merespon dengan menyimpan data apabila data telah lengkap. |
| Bidang Alternatif | - | |
| Kesimpulan | Use case ini menyimpulkan kegiatan bagian Administrasi pada saat menginputkan data surat masuk. | |
| Pasca Kondisi | Sistem akan menyimpan data surat masuk yang telah dimasukkan ke dalam database. | |
| Aturan Bisnis | Data surat masuk yang dimasukkan sesuai dengan data yang sebenarnya. | |
| Batasan dan spesifikasi implementasi | Sistem akan memberikan respon terhadap data yang sudah terisi dengan lengkap. | |
| Asumsi | Bagian Administrasi sudah mempunyai data surat masuk yang akan dimasukkan ke dalam sistem. | |
| Masalah terbuka | - | |

*Table 18. Use Case Naratif Persyaratan Bisnis Input Data Surat Masuk*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pengarang | : | Ogiano Waskitajaya | Tanggal | : | 14 Agustus 2015 |
|  |  | Yanuar Nurcahyo | Versi | : | 1.0 |

|  |  |  |  |
| --- | --- | --- | --- |
| Nama Use Case | Input Data Surat Masuk | Type Use Case | |
| ID Use Case | ASN04 | Analisis Sistem Description: Description: Description: BD21301_ | |
| Prioritas | Tinggi |
| Sumber | - |
| Pelaku Bisnis Utama | Administrasi |
| Pelaku Partisipan Lain | Sistem | | |
| Stakeholder yang berminat lain | - | | |
| Deskripsi | Use case ini mendeskripsikan tentang kegiatan yang dilakukan oleh Administrasi untuk menginput data surat masuk yang diterimanya. | | |
| Prakondisi | Bagian Administrasi harus sudah menerima surat masuk terlebih dahulu. | | |
| Pemicu | Administrasi membuka menu utama. | | |
| Bidang khas suatu event | **Kegiatan Pelaku** | **Respons Sistem** | |
|  | **Langkah 1 :**  Administrasi masuk ke menu utama  **Langkah 2 :**  Bagian Administrasi membuka menu surat masuk.  **Langkah 4 :**  Membuat item baru.  **Langkah 6 :**  Mengisi form disposisi surat masuk, setelah itu mengirim disposisi. | **Langkah 3 :**  Sistem merespon dengan menampilkan menu surat masuk.  **Langkah 5 :**  Sistem merespon dengan menampilkan form surat masuk.  **Langkah 7:**  Sistem melakukan validasi data.  **Langkah 8 :**  Setelah melakukan validasi, sistem akan menyimpan ke dalam database dan mengirimkannya ke Ketua. | |
| Bidang Alternatif | **Alt langkah 7 : Apabila data tidak valid maka administrasi perlu menginputkan kembali data dengan benar.** | |
| Kesimpulan | Use case ini menyimpulkan kegiatan bagian Administrasi pada saat menginputkan data surat masuk. | |
| Pasca kondisi | Sistem akan menyimpan data surat masuk yang telah dimasukkan ke dalam database. | |
| Aturan Bisnis | Data surat masuk yang dimasukkan sesuai dengan data yang sebenarnya. | |
| Batasan dan spesifikasi implementasi | Sistem akan memberikan respon terhadap data yang sudah terisi dengan lengkap. | | |
| Asumsi | Bagian Administrasi sudah mempunyai data surat masuk yang akan dimasukkan ke dalam sistem. | | |
| Masalah terbuka | - | | |

*Table 19. Use Case Naratif Analisis Sistem Input Data Surat Masuk*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pengarang | : | Ogiano Waskitajaya | Tanggal | : | 14 Agustus 2015 |
|  |  | Yanuar Nurcahyo | Versi | : | 1.0 |

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Input Data Surat Masuk | Type Use Case |
| ID Use Case | ASN04 | Design Sistem Description: Description: Description: BD21301_ |
| Prioritas | Tingggi |
| Sumber | - |
| Pelaku Bisnis Utama | Administrasi |
| Pelaku Partisipan Lain | Sistem | |
| Stakeholder yang berminat lain | - | |
| Deskripsi | Use case ini mendeskripsikan tentang kegiatan yang dilakukan oleh Administrasi untuk menginput data surat masuk yang diterimanya. | |
| Prakondisi | Bagian Administrasi harus sudah menerima surat masuk terlebih dahulu. | |
| Pemicu | Administrasi membuka menu surat masuk. | |
| Bidang khas suatu event | **Kegiatan Pelaku** | **Respons Sistem** |
|  | **Langkah 1 :**  Administrasi masuk ke menu utama  **Langkah 2 :**  Bagian Administrasi membuka menu surat masuk.  **Langkah 4 :**  Membuat item baru dengan menekan tombol buat item baru.  **Langkah 6 :**  Mengisi form disposisi surat masuk, setelah itu mengirim disposisi. | **Langkah 3 :**  Sistem merespon dengan menampilkan menu surat masuk.  **Langkah 5 :**  Sistem merespon dengan menampilkan form surat masuk.  **Langkah 7:**  Sistem melakukan validasi data.  **Langkah 8 :**  Setelah melakukan validasi, sistem akan menyimpan ke dalam database dan mengirimkannya ke Ketua. |
| Bidang Alternatif | **Alt langkah 7 : Apabila data tidak valid maka administrasi perlu menginputkan kembali data dengan benar.** | |
| Kesimpulan | Use case ini menyimpulkan kegiatan bagian Administrasi pada saat menginputkan data surat masuk. | |
| Pasca Kondisi | Sistem akan menyimpan data surat masuk yang telah dimasukkan ke dalam database. | |
| Aturan Bisnis | Data surat masuk yang dimasukkan sesuai dengan data yang sebenarnya. | |
| Batasan dan spesifikasi implementasi | Sistem akan memberikan respon terhadap data yang sudah terisi dengan lengkap. | |
| Asumsi | Bagian Administrasi sudah mempunyai data surat masuk yang akan dimasukkan ke dalam sistem. | |
| Masalah Terbuka | - | |

*Table 20. Use Case Naratif Design Sistem Input Data Surat Masuk*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pengarang | : | Ogiano Waskitajaya | Tanggal | : | 14 Agustus 2015 |
|  |  | Yanuar Nurcahyo | Versi | : | 1.0 |

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Membuat Surat Keluar | Type Use Case |
| ID Use Case | ASN05 | Persyaratan Bisnis Description: Description: Description: BD21301_ |
| Prioritas | Tinggi |
| Sumber | - |
| Pelaku bisnis utama | Administrasi |
| Pelaku pertisipan lain | Sistem | |
| Stakeholder yang berminat lain | - | |
| Deskripsi | Use case ini mendeskripsikan tentang kegiatan yang dilakukan bagian Administrasi untuk membuat surat keluar. | |
| Prakondisi | Administrasi sudah membuat surat keluar sebelum di inputkan. | |
| Sasaran | Administrasi dapat memasukkan data surat keluar. | |
| Bdiang khas suatu event | **Kegiatan Pelaku** | **Respons Sistem** |
|  | **Langkah 1 :**  Administrasi menyiapkan data yang akan dimasukkan dalam Sistem. | **Langkah 2 :**  Sistem merespon dengan menyimpan data. |
| Bidang alternative | - | |
| Kesimpulan | Use case ini menyimpulkan kegiatan bagian Administrasi pada saat menginputkan data surat keluar. | |
| Pasca kondisi | Sistem akan menyimpan data yang telah dimasukkan kedalam database. | |
| Aturan Bisnis | Data surat keluar yang dimasukkan sesuai dengan data yang sebenarnya. | |
| Batasan dan spesifikasi implementasi | Sistem akan memberikan respon terhadap data yang sudah terisi dengan lengkap. | |
| Asumsi | Bagian Administrasi mengisi data dengan lengkap kedalam sistem sesuai ketentuan yang berlaku. | |
| Masalah terbuka | - | |

*Table 21. Use Case Naratif Persyaratan Bisnis Membuat Surat Keluar*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pengarang | : | Ogiano Waskitajaya | Tanggal | : | 14 Agustus 2015 |
|  |  | Yanuar Nurcahyo | Versi | : | 1.0 |

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Membuat Surat Keluar | Type Use Case |
| ID Use Case | ASN05 | Analisis Sistem Description: Description: Description: BD21301_ |
| Prioritas | Tinggi |
| Sumber | - |
| Pelaku Bisnis Utama | Administrasi |
| Pelaku Partisipan Lain | Sistem | |
| Stakeholder yang berminat lain | - | |
| Deskripsi | Use case ini mendeskripsikan tentang kegiatan yang dilakukan bagian Administrasi untuk membuat surat keluar. | |
| Prakondisi | Administrasi sudah membuat surat keluar sebelum di inputkan. | |
| Pemicu | Administrasi membuka menu utama. | |
| Bidang khas suatu event | **Kegiatan Pelaku** | **Respons Sistem** |
|  | **Langkah 1 :**  Administrasi masuk ke menu utama.  **Langkah 2 :**  Bagian Administrasi membuka menu surat keluar.  **Langkah 4 :**  Administrasi membuat item baru.  **Langkah 6 :**  Mengisi form surat keluar lalu menyimpan surat keluar. | **Langkah 3 :**  Sistem menampilkan menu surat keluar.  **Langkah 5 :**  Sistem merespon dengan menampilkan form surat keluar.  **Langkah 7 :**  Sistem merespon dengan melakukan validasi terhadap data yang di inputkan.  **Langkah 8 :**  Setelah melakukan validasi Sistem akan menyimpan data yang di inputkan. |
| Bidang Alternatif | **Alternatif langkah 7 :**  **Apabila data yang di inputkan tidak valid maka administrasi perlu menginput ulang data tersebut.** | |
| Kesimpulan | Use case ini menyimpulkan kegiatan bagian Administrasi pada saat menginputkan data surat keluar. | |
| Pasca kondisi | Sistem akan menyimpan data surat keluar yang telah dimasukkan ke dalam database. | |
| Aturan Bisnis | Data surat keluar yang dimasukkan sesuai dengan data yang sebenarnya. | |
| Batasan dan spesifikasi implementasi | Sistem akan memberikan respon terhadap data yang sudah terisi dengan lengkap. | |
| Asumsi | Bagian Administrasi mengisi data dengan lengkap kedalam sistem sesuai ketentuan yang berlaku. | |
| Masalah terbuka | - | |

*Table 22. Use Case Naratif Analisis Sistem Membuat Surat Keluar*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pengarang | : | Ogiano Waskitajaya | Tanggal | : | 14 Agustus 2015 |
|  |  | Yanuar Nurcahyo | Versi | : | 1.0 |

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Membuat Surat Keluar | Type Use Case |
| ID Use Case | ASN05 | Design Sistem Description: Description: Description: BD21301_ |
| Prioritas | Tingggi |
| Sumber | - |
| Pelaku Bisnis Utama | Administrasi |
| Pelaku Partisipan Lain | Sistem | |
| Stakeholder yang berminat lain | - | |
| Deskripsi | Use case ini mendeskripsikan tentang kegiatan yang dilakukan bagian Administrasi untuk membuat surat keluar. | |
| Prakondisi | Administrasi sudah membuat surat keluar sebelum di inputkan. | |
| Pemicu | Administrasi membuka menu surat keluar. | |
| Bidang khas suatu event | **Kegiatan Pelaku** | **Respons Sistem** |
|  | **Langkah 1 :**  Administrasi masuk ke menu utama.  **Langkah 2 :**  Bagian Administrasi membuka menu surat keluar.  **Langkah 4 :**  Administrasi membuat item baru dengan menekan tombol tambah item baru.  **Langkah 6 :**  Mengisi form surat keluar lalu menyimpan surat keluar. | **Langkah 3 :**  Sistem menampilkan menu surat keluar.  **Langkah 5 :**  Sistem merespon dengan menampilkan form surat keluar.  **Langkah 7 :**  Sistem merespon dengan melakukan validasi terhadap data yang di inputkan.  **Langkah 8 :**  Setelah melakukan validasi Sistem akan menyimpan data yang di inputkan. |
| Bidang Alternatif | **Alternatif langkah 7 :**  **Apabila data yang di inputkan tidak valid maka administrasi perlu menginput ulang data tersebut.** | |
| Kesimpulan | Use case ini menyimpulkan kegiatan bagian Administrasi pada saat menginputkan data surat keluar. | |
| Pasca Kondisi | Sistem akan menyimpan data surat keluar yang telah dimasukkan ke dalam database. | |
| Aturan Bisnis | Data surat masuk yang dimasukkan sesuai dengan data yang sebenarnya. | |
| Batasan dan spesifikasi implementasi | Sistem akan memberikan respon terhadap data yang sudah terisi dengan lengkap. | |
| Asumsi | Bagian Administrasi mengisi data dengan lengkap kedalam sistem sesuai ketentuan yang berlaku. | |
| Masalah Terbuka | - | |

*Table 23. Use Case Naratif Design Sistem Membuat Surat Keluar*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pengarang | : | Ogiano Waskitajaya | Tanggal | : | 14 Agustus 2015 |
|  |  | Yanuar Nurcahyo | Versi | : | 1.0 |

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Mengisi Disposisi | Type Use Case |
| ID Use Case | ASN06 | Persyaratan Bisnis Description: Description: Description: BD21301_ |
| Prioritas | Tinggi |
| Sumber | - |
| Pelaku bisnis utama | Ketua |
| Pelaku pertisipan lain | Sistem | |
| Stakeholder yang berminat lain | - | |
| Deskripsi | Use case ini mendeskripsikan tentang kegiatan yang dilakukan oleh Ketua untuk mengisi disposisi surat. | |
| Prakondisi | Ketua menerima notifikasi surat masuk tersebut. | |
| Sasaran | Ketua dapat mengisi lembar disposisi. | |
| Bidang khas suatu event | **Kegiatan Pelaku** | **Respons Sistem** |
|  | **Langkah 1 :**  Ketua menyiapkan data yang akan dimasukkan dalam Sistem. | **Langkah 2 :**  Sistem merespon dengan menyimpan data. |
| Bidang alternative | **-** | |
| Kesimpulan | Use Case ini menyimpulkan kegiatan ketua pada saat mengisi disposisi. | |
| Pasca kondisi | Sistem akan menyimpan data yang telah dimasukkan kedalam database. | |
| Aturan Bisnis | Data disposisi surat yang dimasukkan sesuai dengan data yang sebenarnya. | |
| Batasan dan spesifikasi implementasi | Sistem akan memberikan respon terhadap data yang sudah terisi dengan lengkap. | |
| Asumsi | Ketua mengisi disposisi dengan lengkap kedalam sistem sesuai ketentuan yang berlaku. | |
| Masalah terbuka | - | |

*Table 24. Use Case Naratif Persyaratan Bisnis Mengisi Disposisi*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pengarang | : | Ogiano Waskitajaya | Tanggal | : | 14 Agustus 2015 |
|  |  | Yanuar Nurcahyo | Versi | : | 1.0 |

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Mengisi Disposisi | Type Use Case |
| ID Use Case | ASN06 | Analisis Sistem Description: Description: Description: BD21301_ |
| Prioritas | Tinggi |
| Sumber | - |
| Pelaku Bisnis Utama | Ketua |
| Pelaku Partisipan Lain | Sistem | |
| Stakeholder yang berminat lain | - | |
| Deskripsi | Use case ini mendeskripsikan tentang kegiatan yang dilakukan oleh Ketua untuk mengisi disposisi surat. | |
| Prakondisi | Ketua menerima notifikasi surat masuk tersebut. | |
| Pemicu | Ketua membuka menu surat masuk. | |
| Bidang khas suatu event | **Kegiatan Pelaku** | **Respons Sistem** |
|  | **Langkah 1 :**  Ketua masuk ke menu utama.  **Langkah 3 :**  Ketua membuka notifikasi surat masuk tersebut.  **Langkah 5:**  Ketua menginputkan data disposisi surat. | **Langkah 2 :**  Sistem menampilkan notifikasi bahwa ada surat masuk kepada ketua.  **Langkah 4 :**  Sistem menampilkan form dengan data surat masuk tersebut.  **Langkah 6 :**  Sistem melakukan validasi terhadap data yang telah di inputkan.  **Langkah 7 :**  Setelah melakukan validasi Sistem akan menyimpan data tersebut dan mengirimnya kepada pembantu ketua. |
| Bidang Alternatif | **Alt langkah 6 : Apabilah data tidak valid maka ketua perlu menginputkan kembali data dengan benar.** | |
| Kesimpulan | Use Case ini menyimpulkan kegiatan ketua pada saat ingin mengisi disposisi. | |
| Pasca kondisi | Sistem akan menyimpan data yang telah dimasukkan kedalam database. | |
| Aturan Bisnis | Data disposisi surat yang dimasukkan sesuai dengan data yang sebenarnya. | |
| Batasan dan spesifikasi implementasi | Sistem akan memberikan respon terhadap data yang sudah terisi dengan lengkap. | |
| Asumsi | Ketua mengisi disposisi dengan lengkap kedalam sistem sesuai ketentuan yang berlaku. | |
| Masalah terbuka | - | |

*Table 25. Use Case Naratif Analisis Sistem Mengisi Disposisi*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pengarang | : | Ogiano Waskitajaya | Tanggal | : | 14 Agustus 2015 |
|  |  | Yanuar Nurcahyo | Versi | : | 1.0 |

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Mengisi Disposisi | Type Use Case |
| ID Use Case | ASN06 | Design Sistem Description: Description: Description: BD21301_ |
| Prioritas | Tinggi |
| Sumber | - |
| Pelaku bisnis utama | Ketua |
| Pelaku pertisipan lain | Sistem | |
| Stakeholder yang berminat lain | - | |
| Deskripsi | Use case ini mendeskripsikan tentang kegiatan yang dilakukan oleh Ketua untuk mengisi disposisi surat. | |
| Prakondisi | Ketua menerima notifikasi surat masuk tersebut. | |
| Pemicu | Ketua membuka lembar disposisi. | |
| Bidang khas suatu event | **Kegiatan Pelaku** | **Respons Sistem** |
|  | **Langkah 1 :**  Ketua masuk ke menu utama.  **Langkah 3 :**  Ketua membuka notifikasi surat masuk tersebut dengan menekan icon notifikasi surat masuk.  **Langkah 5:**  Ketua menginputkan data disposisi surat dan mengirimnya dengan menekan tombol kirim disposisi . | **Langkah 2 :**  Sistem menampilkan notifikasi bahwa ada surat masuk kepada ketua.  **Langkah 4 :**  Sistem menampilkan form dengan data surat masuk tersebut.  **Langkah 6 :**  Sistem melakukan validasi terhadap data yang telah di inputkan.  **Langkah 7 :**  Setelah melakukan validasi Sistem akan menyimpan data tersebut dan mengirimnya kepada pembantu ketua. |
| Bidang alternative | **Alt langkah 6 : Apabila data tidak valid maka ketua perlu menginputkan kembali data dengan benar.** | |
| Kesimpulan | Use Case ini menyimpulkan kegiatan ketua pada saat mengisi disposisi. | |
| Pasca kondisi | Sistem akan menyimpan data yang telah dimasukkan kedalam database. | |
| Aturan Bisnis | Data disposisi surat yang dimasukkan sesuai dengan data yang sebenarnya. | |
| Batasan dan spesifikasi implementasi | Sistem akan memberikan respon terhadap data yang sudah terisi dengan lengkap. | |
| Asumsi | Ketua mengisi disposisi dengan lengkap kedalam sistem sesuai ketentuan yang berlaku. | |
| Masalah terbuka | - | |

*Table 26. Use Case Naratif Design Sistem Mengisi Disposisi*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pengarang | : | Ogiano Waskitajaya | Tanggal | : | 14 Agustus 2015 |
|  |  | Yanuar Nurcahyo | Versi | : | 1.0 |

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Memberi Perintah ke Kepala Unit | Type Use Case |
| ID Use Case | ASN07 | Persyaratan Bisnis Description: Description: Description: BD21301_ |
| Prioritas | Tinggi |
| Sumber | - |
| Pelaku bisnis utama | Pembantu Ketua |
| Pelaku pertisipan lain | Sistem | |
| Stakeholder yang berminat lain | - | |
| Deskripsi | Use case ini mendeskripsikan tentang kegiatan yang dilakukan oleh Pembantu Ketua untuk memberikan perintah kepada kepala unit. | |
| Prakondisi | Pembantu Ketua menerima notifikasi surat masuk. | |
| Sasaran | Pembantu Ketua memberi perintah ke kepala unit. | |
| Bidang khas suatu event | **Kegiatan Pelaku** | **Respons Sistem** |
|  | **Langkah 1 :**  Pembantu Ketua menyiapkan data yang akan dimasukkan dalam Sistem. | **Langkah 2 :**  Sistem merespon dengan menyimpan data. |
| Bidang alternative | - | |
| Kesimpulan | Use Case ini menyimpulkan kegiatan Pembantu Ketua pada saat memberikan perintah ke kepala unit. | |
| Pasca kondisi | Sistem akan menyimpan data yang telah dimasukkan kedalam database. | |
| Aturan Bisnis | Data yang dimasukkan sesuai dengan data yang sebenarnya. | |
| Batasan dan spesifikasi implementasi | Sistem akan memberikan respon terhadap data yang sudah terisi dengan lengkap. | |
| Asumsi | Pembantu Ketua mengisi data dengan lengkap kedalam sistem sesuai ketentuan yang berlaku. | |
| Masalah terbuka | - | |

*Table 27. Use Case Naratif Persyaratan Bisnis Memberi Perintah ke Kepala Unit*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pengarang | : | Ogiano Waskitajaya | Tanggal | : | 14 Agustus 2015 |
|  |  | Yanuar Nurcahyo | Versi | : | 1.0 |

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Memberi Perintah ke Kepala Unit | Type Use Case |
| ID Use Case | ASN07 | Analisis Sistem Description: Description: Description: BD21301_ |
| Prioritas | Tinggi |
| Sumber | - |
| Pelaku bisnis utama | Pembantu Ketua |
| Pelaku pertisipan lain | Sistem | |
| Stakeholder yang berminat lain | - | |
| Deskripsi | Use case ini mendeskripsikan tentang kegiatan yang dilakukan oleh Pembantu Ketua untuk memberikan perintah kepada kepala unit. | |
| Prakondisi | Pembantu Ketua menerima notifikasi surat masuk. | |
| Pemicu | Pembantu Ketua membuka menu surat masuk. | |
| Bidang khas suatu event | **Kegiatan Pelaku** | **Respons Sistem** |
|  | **Langkah 1 :**  Pembantu Ketua masuk ke menu utama.  **Langkah 3 :**  Pembantu Ketua membuka notifikasi surat masuk tersebut.  **Langkah 5:**  Pembantu Ketua menginputkan data disposisi surat dan mengirimnya kepada kepala unit. | **Langkah 2 :**  Sistem menampilkan notifikasi bahwa ada surat masuk kepada pembantu ketua.  **Langkah 4 :**  Sistem menampilkan form dengan data surat masuk tersebut.  **Langkah 6 :**  Sistem melakukan validasi terhadap data yang telah di inputkan.  **Langkah 7 :**  Setelah melakukan validasi Sistem akan menyimpan data tersebut dan mengirimnya kepada kepala unit. |
| Bidang alternative | **Alternatif langkah 6 :**  **Apabila data yang di inputkan tidak valid maka user perlu menginputkan kembali data tersebut.** | |
| Kesimpulan | Use Case ini menyimpulkan kegiatan Pembantu Ketua pada saat memberikan perintah ke kepala unit. | |
| Pasca kondisi | Sistem akan menyimpan data yang telah dimasukkan kedalam database. | |
| Aturan Bisnis | Data yang dimasukkan sesuai dengan data yang sebenarnya. | |
| Batasan dan spesifikasi implementasi | Sistem akan memberikan respon terhadap data yang sudah terisi dengan lengkap. | |
| Asumsi | Pembantu Ketua mengisi data dengan lengkap kedalam sistem sesuai ketentuan yang berlaku. | |
| Masalah terbuka | - | |

*Table 28. Use Case Naratif Analisis Sistem Memberi Perintah ke Kepala Unit*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pengarang | : | Ogiano Waskitajaya | Tanggal | : | 14 Agustus 2015 |
|  |  | Yanuar Nurcahyo | Versi | : | 1.0 |

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Memberi Perintah ke Kepala Unit | Type Use Case |
| ID Use Case | ASN07 | Design Sistem Description: Description: Description: BD21301_ |
| Prioritas | Tinggi |
| Sumber | - |
| Pelaku bisnis utama | Pembantu Ketua |
| Pelaku pertisipan lain | Sistem | |
| Stakeholder yang berminat lain | - | |
| Deskripsi | Use case ini mendeskripsikan tentang kegiatan yang dilakukan oleh Pembantu Ketua untuk memberikan perintah kepada kepala unit. | |
| Prakondisi | Pembantu Ketua menerima notifikasi surat masuk. | |
| Pemicu | Pembantu Ketua membuka lembar disposisi. | |
| Bidang khas suatu event | **Kegiatan Pelaku** | **Respons Sistem** |
|  | **Langkah 1 :**  Pembantu Ketua masuk ke menu utama.  **Langkah 3 :**  Pembantu Ketua membuka notifikasi surat masuk tersebut dengan menekan icon notifikasi surat masuk.  **Langkah 5:**  Pembantu Ketua menginputkan data disposisi surat dan mengirimnya kepada kepala unit dengan menekan tombol kirim disposisi. | **Langkah 2 :**  Sistem menampilkan notifikasi bahwa ada surat masuk kepada pembantu ketua.  **Langkah 4 :**  Sistem menampilkan form dengan data surat masuk tersebut.  **Langkah 6 :**  Sistem melakukan validasi terhadap data yang telah di inputkan.  **Langkah 7 :**  Setelah melakukan validasi Sistem akan menyimpan data tersebut dan mengirimnya kepada kepala unit. |
| Bidang alternative | **Alternatif langkah 6 :**  **Apabila data yang di inputkan tidak valid maka user perlu menginputkan kembali data tersebut.** | |
| Kesimpulan | Use Case ini menyimpulkan kegiatan Pembantu Ketua pada saat memberikan perintah ke kepala unit. | |
| Pasca kondisi | Sistem akan menyimpan data yang telah dimasukkan kedalam database. | |
| Aturan Bisnis | Data yang dimasukkan sesuai dengan data yang sebenarnya. | |
| Batasan dan spesifikasi implementasi | Sistem akan memberikan respon terhadap data yang sudah terisi dengan lengkap. | |
| Asumsi | Pembantu Ketua mengisi data dengan lengkap kedalam sistem sesuai ketentuan yang berlaku. | |
| Masalah terbuka | - | |

*Table 29. Use Case Naratif Design Sistem Memberi Perintah ke Kepala Unit*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pengarang | : | Ogiano Waskitajaya | Tanggal | : | 14 Agustus 2015 |
|  |  | Yanuar Nurcahyo | Versi | : | 1.0 |

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Melakukan Verifikasi | Type Use Case |
| ID Use Case | ASN08 | Persyaratan Bisnis Description: Description: Description: BD21301_ |
| Prioritas | Tinggi |
| Sumber | - |
| Pelaku bisnis utama | Pembantu Ketua |
| Pelaku pertisipan lain | Sistem | |
| Stakeholder yang berminat lain | Kepala Unit | |
| Deskripsi | Use case ini mendiskripsikan tentang kegiatan yang dilakukan oleh Pembantu Ketua dan Kepala Unit untuk melakukan verifikasi terhadap surat keluar yang akan di kirim. | |
| Prakondisi | Surat sudah disampaikan oleh bagian administrasi kepada Pembantu Ketua dan Kepala Unit | |
| Sasaran | Pembantu ketua dapat memverifikasi surat keluar. | |
| Bidang khas suatu event | **Kegiatan Pelaku** | **Respons Sistem** |
|  | **Langkah 1 :**  User melakukan login dan memverifikasi surat. | **Langkah 2 :**  Sistem merespon dengan menyimpan data. |
| Bidang alternative | - | |
| Kesimpulan | Use Case ini menyimpulkan kegiatan Pembantu Ketua dan Kepala Unit pada saat melakukan verifikasi untuk surat keluar. | |
| Pasca kondisi | Sistem akan mengirim hasil verifikasi kepada bagian administrasi. | |
| Aturan Bisnis | Pembantu Ketua dan Kepala unit telah membaca surat keluar. | |
| Batasan dan spesifikasi implementasi | Sistem akan memberikan respon terhadap data yang sudah terisi dengan lengkap. | |
| Asumsi | Pembantu Ketua dan Kepala unit mengisi data dengan lengkap kedalam sistem sesuai ketentuan yang berlaku. | |
| Masalah terbuka | - | |

*Table 30. Use Case Naratif Persyaratan Bisnis Melakukan Verifikasi*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pengarang | : | Ogiano Waskitajaya | Tanggal | : | 14 Agustus 2015 |
|  |  | Yanuar Nurcahyo | Versi | : | 1.0 |

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Melakukan Verifikasi | Type Use Case |
| ID Use Case | ASN08 | Analisis Sistem Description: Description: Description: BD21301_ |
| Prioritas | Tinggi |
| Sumber | - |
| Pelaku bisnis utama | Pembantu Ketua, Kepala Unit |
| Pelaku pertisipan lain | Sistem | |
| Stakeholder yang berminat lain | Kepala Unit | |
| Deskripsi | Use case ini mendiskripsikan tentang kegiatan yang dilakukan oleh Pembantu Ketua dan Kepala Unit untuk melakukan verifikasi terhadap surat keluar yang akan di kirim. | |
| Prakondisi | Surat sudah disampaikan oleh bagian administrasi kepada Pembantu Ketua | |
| Pemicu | Pembantu Ketua membuka menu surat keluar. | |
| Bidang khas suatu event | **Kegiatan Pelaku** | **Respons Sistem** |
|  | **Langkah 1:**  User masuk ke menu utama  **Langkah 3 :**  User membuka notifikasi tersebut.  **Langkah 5 :**  User melakukan verifikasi. | **Langkah 2 :**  Sistem akan menampilkan notifikasi apabila ada pesan baru.  **Langkah 4 :**  Sistem merespon dengan menampilkan bahwa terdapat pesan baru yang belum di verifikasi.  **Langkah 6 :**  Sistem merespon dengan mengirimkan kembali kepada bagian administrasi. |
| Bidang alternative | - | |
| Kesimpulan | Use Case ini menyimpulkan kegiatan Pembantu Ketua dan Kepala Unit pada saat melakukan verifikasi untuk surat keluar. | |
| Pasca kondisi | Sistem akan mengirim hasil verifikasi kepada bagian administrasi. | |
| Aturan Bisnis | Pembantu Ketua dan Kepala unit telah membaca surat keluar. | |
| Batasan dan spesifikasi implementasi | Sistem akan memberikan respon terhadap data yang sudah terisi dengan lengkap. | |
| Asumsi | Pembantu Ketua dan Kepala unit mengisi data dengan lengkap kedalam sistem sesuai ketentuan yang berlaku. | |
| Masalah terbuka | - | |

*Table 31. Use Case Naratif Analisis Sistem Melakukan Verifikasi*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pengarang | : | Ogiano Waskitajaya | Tanggal | : | 14 Agustus 2015 |
|  |  | Yanuar Nurcahyo | Versi | : | 1.0 |

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Melakukan Verifikasi | Type Use Case |
| ID Use Case | ASN08 | Design Sistem Description: Description: Description: BD21301_ |
| Prioritas | Tinggi |
| Sumber | - |
| Pelaku bisnis utama | Pembantu Ketua |
| Pelaku pertisipan lain | Sistem | |
| Stakeholder yang berminat lain | Kepala Unit | |
| Deskripsi | Use case ini mendiskripsikan tentang kegiatan yang dilakukan oleh Pembantu Ketua dan Kepala Unit untuk melakukan verifikasi terhadap surat keluar yang akan di kirim. | |
| Prakondisi | Surat sudah disampaikan oleh bagian administrasi kepada Pembantu Ketua | |
| Pemicu | Pembantu Ketua membaca surat keluar. | |
| Bidang khas suatu event | **Kegiatan Pelaku** | **Respons Sistem** |
|  | **Langkah 1:**  User masuk ke menu utama  **Langkah 3 :**  User membuka notifikasi tersebut.  **Langkah 5 :**  User melakukan verifikasi dengan menekan tombol approve. | **Langkah 2 :**  Sistem akan menampilkan notifikasi apabila ada pesan baru.  **Langkah 4 :**  Sistem merespon dengan menampilkan bahwa terdapat pesan baru yang belum di verifikasi.  **Langkah 6 :**  Sistem merespon dengan mengirimkan kembali kepada bagian administrasi. |
| Bidang alternative | - | |
| Kesimpulan | Use Case ini menyimpulkan kegiatan Pembantu Ketua dan Kepala Unit pada saat melakukan verifikasi untuk surat keluar. | |
| Pasca kondisi | Sistem akan mengirim hasil verifikasi kepada bagian administrasi. | |
| Aturan Bisnis | Pembantu Ketua dan Kepala unit telah membaca surat keluar. | |
| Batasan dan spesifikasi implementasi | Sistem akan memberikan respon terhadap data yang sudah terisi dengan lengkap. | |
| Asumsi | Pembantu Ketua dan Kepala unit mengisi data dengan lengkap kedalam sistem sesuai ketentuan yang berlaku. | |
| Masalah terbuka | - | |

*Table 32. Use Case Naratif Design Sistem Melakukan Verifikasi*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pengarang | : | Ogiano Waskitajaya | Tanggal | : | 14 Agustus 2015 |
|  |  | Yanuar Nurcahyo | Versi | : | 1.0 |

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Memberi Paraf Koreksi | Type Use Case |
| ID Use Case | ASN10 | Persyaratan Bisnis Description: Description: Description: BD21301_ |
| Prioritas | Tinggi |
| Sumber | - |
| Pelaku bisnis utama | Pembantu Ketua |
| Pelaku pertisipan lain | Sistem | |
| Stakeholder yang berminat lain | - | |
| Deskripsi | Use case ini mendeskripsikan tentang kegiatan yang dilakukan oleh Pembantu Ketua untuk memberikan paraf koreksi atau notifikasi dalam sistem terhadap surat keluar. | |
| Prakondisi | Pembantu Ketua sudah menerima notifikasi surat keluar. | |
| Sasaran | Pembantu Ketua dapat memberi paraf koreksi. | |
| Bidang khas suatu event | **Kegiatan Pelaku** | **Respons Sistem** |
|  | **Langkah 1 :**  Pembantu Ketua melakukan login dan memverifikasi surat. | **Langkah 2 :**  Sistem merespon dengan menyimpan data. |
| Bidang alternative | - | |
| Kesimpulan | Use Case ini menyimpulkan kegiatan Pembantu Ketua pada saat memberikan paraf koreksi. | |
| Pasca kondisi | Sistem akan menyimpan data yang telah dimasukkan kedalam database. | |
| Aturan Bisnis | Pembantu Ketua telah menerima surat keluar tersebut. | |
| Batasan dan spesifikasi implementasi | Sistem akan memberikan respon terhadap data yang sudah terisi dengan lengkap. | |
| Asumsi | Pembantu Ketua mengisi data kedalam sistem sesuai ketentuan yang berlaku. | |
| Masalah terbuka | - | |

*Table 33. Use Case Naratif Persyaratan Bisnis Memberi Paraf Koreksi*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pengarang | : | Ogiano Waskitajaya | Tanggal | : | 14 Agustus 2015 |
|  |  | Yanuar Nurcahyo | Versi | : | 1.0 |

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Memberi Paraf Koreksi | Type Use Case |
| ID Use Case | ASN10 | Analisis Sistem Description: Description: Description: BD21301_ |
| Prioritas | Tinggi |
| Sumber | - |
| Pelaku bisnis utama | Pembantu Ketua |
| Pelaku pertisipan lain | Sistem | |
| Stakeholder yang berminat lain | Pembantu Ketua, Kepala Unit | |
| Deskripsi | Use case ini mendeskripsikan tentang kegiatan yang dilakukan oleh Pembantu Ketua untuk memberikan paraf koreksi atau notifikasi dalam sistem terhadap surat keluar. | |
| Prakondisi | Pembantu Ketua sudah menerima notifikasi surat keluar. | |
| Pemicu | User membuka menu surat keluar. | |
| Bidang khas suatu event | **Kegiatan Pelaku** | **Respons Sistem** |
|  | **Langkah 1:**  Pembantu Ketua masuk ke menu utama  **Langkah 3 :**  Pembantu Ketua membuka notifikasi tersebut.  **Langkah 5 :**  User memberikan paraf koreksi. | **Langkah 2 :**  Sistem akan menampilkan notifikasi apabila ada pesan baru.  **Langkah 4 :**  Sistem merespon dengan menampilkan bahwa terdapat pesan baru yang belum di beri paraf koreksi.  **Langkah 6 :**  Sistem merespon dengan mengirim ke administrasi dan ketua. |
| Bidang alternative | - | |
| Kesimpulan | Use Case ini menyimpulkan kegiatan Pembantu Ketua pada saat memberikan paraf koreksi. | |
| Pasca kondisi | Sistem akan menyimpan data yang telah dimasukkan kedalam database. | |
| Aturan Bisnis | Pembantu Ketua telah menerima surat keluar tersebut. | |
| Batasan dan spesifikasi implementasi | Sistem akan memberikan respon terhadap data yang sudah terisi dengan lengkap. | |
| Asumsi | Pembantu Ketua mengisi data kedalam sistem sesuai ketentuan yang berlaku. | |
| Masalah terbuka | - | |

*Table 34. Use Case Naratif Analisis SIstem Memberi Paraf Koreksi*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pengarang | : | Ogiano Waskitajaya | Tanggal | : | 14 Agustus 2015 |
|  |  | Yanuar Nurcahyo | Versi | : | 1.0 |

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Memberi Paraf Koreksi | Type Use Case |
| ID Use Case | ASN10 | Design Sistem Description: Description: Description: BD21301_ |
| Prioritas | Tinggi |
| Sumber | - |
| Pelaku bisnis utama | Pembantu Ketua |
| Pelaku pertisipan lain | Sistem | |
| Stakeholder yang berminat lain | Pembantu Ketua, Kepala Unit | |
| Deskripsi | Use case ini mendeskripsikan tentang kegiatan yang dilakukan oleh Pembantu Ketua untuk memberikan paraf koreksi atau notifikasi dalam sistem terhadap surat keluar. | |
| Prakondisi | Pembantu Ketua sudah menerima notifikasi surat keluar. | |
| Pemicu | User membuka menu surat keluar. | |
| Bidang khas suatu event | **Kegiatan Pelaku** | **Respons Sistem** |
|  | **Langkah 1:**  Pembantu Ketua masuk ke menu utama  **Langkah 3 :**  Pembantu Ketua membuka notifikasi tersebut.  **Langkah 5 :**  User memberikan paraf koreksi dengan menekan tombol approve. | **Langkah 2 :**  Sistem akan menampilkan notifikasi apabila ada pesan baru.  **Langkah 4 :**  Sistem merespon dengan menampilkan bahwa terdapat pesan baru yang belum di beri paraf koreksi.  **Langkah 6 :**  Sistem merespon dengan mengirim ke administrasi dan ketua. |
| Bidang alternative | - | |
| Kesimpulan | Use Case ini menyimpulkan kegiatan Pembantu Ketua pada saat memberikan paraf koreksi. | |
| Pasca kondisi | Sistem akan menyimpan data yang telah dimasukkan kedalam database. | |
| Aturan Bisnis | Pembantu Ketua telah menerima surat keluar tersebut. | |
| Batasan dan spesifikasi implementasi | Sistem akan memberikan respon terhadap data yang sudah terisi dengan lengkap. | |
| Asumsi | Pembantu Ketua mengisi data kedalam sistem sesuai ketentuan yang berlaku. | |
| Masalah terbuka | - | |

*Table 35. Use Case Naratif Design Sistem Memberi Paraf Koreksi*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pengarang | : | Ogiano Waskitajaya | Tanggal | : | 14 Agustus 2015 |
|  |  | Yanuar Nurcahyo | Versi | : | 1.0 |

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Memberi Tanda Tangan | Type Use Case |
| ID Use Case | ASN11 | Persyaratan Bisnis Description: Description: Description: BD21301_ |
| Prioritas | Tinggi |
| Sumber | - |
| Pelaku bisnis utama | Ketua |
| Pelaku pertisipan lain | Sistem | |
| Stakeholder yang berminat lain | - | |
| Deskripsi | Use case ini mendeskripsikan tentang kegiatan yang dilakukan oleh Ketua untuk memberikan tanda tangan atau notifikasi dalam sistem terhadap surat keluar | |
| Prakondisi | Ketua sudah menerima notifikasi surat keluar. | |
| Sasaran | Ketua dapat memberi tanda tangan. | |
| Bidang khas suatu event | **Kegiatan Pelaku** | **Respons Sistem** |
|  | **Langkah 1 :**  Ketua melakukan login dan memverifikasi surat. | **Langkah 2 :**  Sistem merespon dengan menyimpan data. |
| Bidang alternative | - | |
| Kesimpulan | Use Case ini menyimpulkan kegiatan Ketua pada saat memberikan tanda tangan. | |
| Pasca kondisi | Sistem akan menyimpan data yang telah dimasukkan kedalam database. | |
| Aturan Bisnis | Ketua telah menerima surat keluar tersebut. | |
| Batasan dan spesifikasi implementasi | Sistem akan memberikan respon terhadap data yang sudah terisi dengan lengkap. | |
| Asumsi | Ketua mengisi data kedalam sistem sesuai ketentuan yang berlaku. | |
| Masalah terbuka | - | |

*Table 36. Use Case Naratif Persyaratan Bisnis Memberi Tanda Tangan*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pengarang | : | Ogiano Waskitajaya | Tanggal | : | 14 Agustus 2015 |
|  |  | Yanuar Nurcahyo | Versi | : | 1.0 |

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Memberi Tanda Tangan | Type Use Case |
| ID Use Case | ASN11 | Analisis Sistem Description: Description: Description: BD21301_ |
| Prioritas | Tinggi |
| Sumber | - |
| Pelaku bisnis utama | Ketua |
| Pelaku pertisipan lain | Sistem | |
| Stakeholder yang berminat lain | - | |
| Deskripsi | Use case ini mendeskripsikan tentang kegiatan yang dilakukan oleh Ketua untuk memberikan tanda tangan atau notifikasi dalam sistem terhadap surat keluar | |
| Prakondisi | Ketua sudah menerima notifikasi surat keluar. | |
| Pemicu | Ketua membuka surat keluar | |
| Bidang khas suatu event | **Kegiatan Pelaku** | **Respons Sistem** |
|  | **Langkah 1:**  Ketua masuk ke menu utama  **Langkah 3 :**  Ketua membuka notifikasi tersebut.  **Langkah 5 :**  User memberikan tanda tangan. | **Langkah 2 :**  Sistem akan menampilkan notifikasi apabila ada pesan baru.  **Langkah 4 :**  Sistem merespon dengan menampilkan bahwa terdapat pesan baru yang belum di tanda tangan.  **Langkah 6 :**  Sistem merespon dengan mengirim ke administrasi. |
| Bidang alternative | - | |
| Kesimpulan | Use Case ini menyimpulkan kegiatan Ketua pada saat memberikan tanda tangan. | |
| Pasca kondisi | Sistem akan menyimpan data yang telah dimasukkan kedalam database. | |
| Aturan Bisnis | Ketua telah menerima surat keluar tersebut. | |
| Batasan dan spesifikasi implementasi | Sistem akan memberikan respon terhadap data yang sudah terisi dengan lengkap. | |
| Asumsi | Ketua mengisi data kedalam sistem sesuai ketentuan yang berlaku. | |
| Masalah terbuka | - | |

*Table 37. Use Case Naratif Analisis Sistem Memberi Tanda Tangan*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pengarang | : | Ogiano Waskitajaya | Tanggal | : | 14 Agustus 2015 |
|  |  | Yanuar Nurcahyo | Versi | : | 1.0 |

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Memberi Tanda Tangan | Type Use Case |
| ID Use Case | ASN11 | Design SIstem Description: Description: Description: BD21301_ |
| Prioritas | Tinggi |
| Sumber | - |
| Pelaku bisnis utama | Ketua |
| Pelaku pertisipan lain | Sistem | |
| Stakeholder yang berminat lain | - | |
| Deskripsi | Use case ini mendeskripsikan tentang kegiatan yang dilakukan oleh Ketua untuk memberikan tanda tangan atau notifikasi dalam sistem terhadap surat keluar | |
| Prakondisi | Ketua sudah menerima notifikasi surat keluar. | |
| Pemicu | Ketua membuka surat keluar | |
| Bidang khas suatu event | **Kegiatan Pelaku** | **Respons Sistem** |
|  | **Langkah 1:**  Ketua masuk ke menu utama  **Langkah 3 :**  Ketua membuka notifikasi tersebut.  **Langkah 5 :**  User memberikan tanda tangan dengan menekan tombol approve. | **Langkah 2 :**  Sistem akan menampilkan notifikasi apabila ada pesan baru.  **Langkah 4 :**  Sistem merespon dengan menampilkan bahwa terdapat pesan baru yang belum di tanda tangan.  **Langkah 6 :**  Sistem merespon dengan mengirim ke administrasi. |
| Bidang alternative | - | |
| Kesimpulan | Use Case ini menyimpulkan kegiatan Ketua pada saat memberikan tanda tangan. | |
| Pasca kondisi | Sistem akan menyimpan data yang telah dimasukkan kedalam database. | |
| Aturan Bisnis | Ketua telah menerima surat keluar tersebut. | |
| Batasan dan spesifikasi implementasi | Sistem akan memberikan respon terhadap data yang sudah terisi dengan lengkap. | |
| Asumsi | Ketua mengisi data kedalam sistem sesuai ketentuan yang berlaku. | |
| Masalah terbuka | - | |

*Table 38. Use Case Naratif Design Sistem Memberi Tanda Tangan*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pengarang | : | Ogiano Waskitajaya | Tanggal | : | 14 Agustus 2015 |
|  |  | Yanuar Nurcahyo | Versi | : | 1.0 |

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Melihat Disposisi | Type Use Case |
| ID Use Case | ASN12 | Persyaratan Bisnis Description: Description: Description: BD21301_ |
| Prioritas | Tinggi |
| Sumber | - |
| Pelaku bisnis utama | Kepala Unit |
| Pelaku pertisipan lain | Sistem | |
| Stakeholder yang berminat lain | - | |
| Deskripsi | Use case ini mendeskripsikan tentang kegiatan Kepala Unit untuk melihat lembar disposisi yang dikirimkan kepada nya. | |
| Prakondisi | Kepala Unit menerima notifikasi disposisi tersebut. | |
| Sasaran | Use case ini dilakukan agar kepala unit dapat melihat disposisi surat masuk. | |
| Bidang khas suatu event | **Kegiatan Pelaku** | **Respons Sistem** |
|  | **Langkah 1 :**  Kepala Unit mengaktifkan menu surat masuk. | **Langkah 2 :**  Sistem merespon dengan menampilkan disposisi surat masuk. |
| Bidang alternative | - | |
| Kesimpulan | Use Case ini menyimpulkan kegiatan Kepala Unit melihat lembar disposisi. | |
| Pasca kondisi | Kepala Unit dapat melihat lembar disposisi. | |
| Aturan Bisnis | Ketua telah menerima surat keluar tersebut. | |
| Batasan dan spesifikasi implementasi | Kepala Unit melihat lembar disposisi sesuai dengan yang telah di masukkan oleh Ketua dan Pembantu Ketua. | |
| Asumsi | Administrasi telah membuat lembar disposisi sebelumnya. | |
| Masalah terbuka | - | |

*Table 39. Use Case Naratif Persyaratan Bisnis Melihat Disposisi*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pengarang | : | Ogiano Waskitajaya | Tanggal | : | 14 Agustus 2015 |
|  |  | Yanuar Nurcahyo | Versi | : | 1.0 |

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Melihat Disposisi | Type Use Case |
| ID Use Case | ASN12 | Analisis Sistem Description: Description: Description: BD21301_ |
| Prioritas | Tinggi |
| Sumber | - |
| Pelaku bisnis utama | Kepala Unit |
| Pelaku pertisipan lain | Sistem | |
| Stakeholder yang berminat lain | - | |
| Deskripsi | Use case ini mendeskripsikan tentang kegiatan Kepala Unit untuk melihat lembar disposisi yang dikirimkan kepada nya. | |
| Prakondisi | Kepala Unit menerima notifikasi disposisi tersebut. | |
| Pemicu | Kepala Unit membuka menu surat masuk. | |
| Bidang khas suatu event | **Kegiatan Pelaku** | **Respons Sistem** |
|  | **Langkah 1 :**  Kepala Unit masuk ke menu utama.  **Langkah 2 :**  Kepala Unit membuka menu surat masuk  **Langkah 4 :**  Kepala Unit membuka disposisi surat masuk. | **Langkah 3 :**  Sistem merespon dengan menampilkan menu surat masuk.  **Langkah 5 :**  Sistem merespon dengan menampilkan lembar disposisi surat masuk. |
| Bidang alternative | - | |
| Kesimpulan | Use Case ini menyimpulkan kegiatan Kepala Unit melihat lembar disposisi. | |
| Pasca kondisi | Kepala Unit dapat melihat lembar disposisi. | |
| Aturan Bisnis | Ketua telah menerima surat keluar tersebut. | |
| Batasan dan spesifikasi implementasi | Kepala Unit melihat lembar disposisi sesuai dengan yang telah di masukkan oleh Ketua dan Pembantu Ketua. | |
| Asumsi | Administrasi telah membuat lembar disposisi sebelumnya. | |
| Masalah terbuka | - | |

*Table 40. Use Case Naratif Analisis SIstem Melihat Disposisi*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pengarang | : | Ogiano Waskitajaya | Tanggal | : | 14 Agustus 2015 |
|  |  | Yanuar Nurcahyo | Versi | : | 1.0 |

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Melihat Disposisi | Type Use Case |
| ID Use Case | ASN12 | Design Sistem Description: Description: Description: BD21301_ |
| Prioritas | Tinggi |
| Sumber | - |
| Pelaku bisnis utama | Kepala Unit |
| Pelaku pertisipan lain | Sistem | |
| Stakeholder yang berminat lain | - | |
| Deskripsi | Use case ini mendeskripsikan tentang kegiatan Kepala Unit untuk melihat lembar disposisi yang dikirimkan kepada nya. | |
| Prakondisi | Kepala Unit menerima notifikasi disposisi tersebut. | |
| Pemicu | Kepala Unit membuka menu surat masuk. | |
| Bidang khas suatu event | **Kegiatan Pelaku** | **Kegiatan Pelaku** |
|  | **Langkah 1 :**  Kepala Unit masuk ke menu utama.  **Langkah 2 :**  Kepala Unit membuka menu surat masuk  **Langkah 4 :**  Kepala Unit membuka disposisi surat masuk. | **Langkah 3 :**  Sistem merespon dengan menampilkan menu surat masuk.  **Langkah 5 :**  Sistem merespon dengan menampilkan lembar disposisi surat masuk. |
| Bidang alternative | - | |
| Kesimpulan | Use Case ini menyimpulkan kegiatan Kepala Unit melihat lembar disposisi. | |
| Pasca kondisi | Kepala Unit dapat melihat lembar disposisi. | |
| Aturan Bisnis | Ketua telah menerima surat keluar tersebut. | |
| Batasan dan spesifikasi implementasi | Kepala Unit melihat lembar disposisi sesuai dengan yang telah di masukkan oleh Ketua dan Pembantu Ketua. | |
| Asumsi | Administrasi telah membuat lembar disposisi sebelumnya. | |
| Masalah terbuka | - | |

*Table 41. Use Case Naratif Design SIstem Melihat Disposisi*

## **Aktifitas Proses Bisnis**



*Gambar 8. Diagram Activity Login*



*Gambar 9. Diagram Activity Logout*



*Gambar 10. Diagram Activity Menerima Notifikasi*



*Gambar 11. Diagram Activity Input Data Surat Masuk*



*Gambar 12. Diagram Activity Membuat Surat Keluar*



*Gambar 13. Diagram Activity Memberi Perintah ke Kepala Unit*



*Gambar 14. Diagram Activity Melakukan Verifikasi*



*Gambar 15. Diagram Activity Mengisi Disposisi*



*Gambar 16. Diagram Activity Memberi Paraf Koreksi*



*Gambar 17. Diagram Activity Memberi Tanda Tangan*



*Gambar 18. Diagram Activity Melihat Disposisi*

## **Pemetaan Objek**



*Gambar 19. Pemetaan Objek Login*



*Gambar 20. Pemetaan Objek Input Data Surat Masuk*



*Gambar 21. Pemetaan Objek Mengisi Disposisi*



*Gambar 22. Pemetaan Objek Memberi Perintah ke Kepala Unit*



*Gambar 23. Pemetaan Objek Menerima Notifikasi*



*Gambar 24. Pemetaan Objek Membuat Surat Keluar*



*Gambar 25. Pemetaan Objek Melakukan Verifikasi*



*Gambar 26. Pemetaan Objek Logout*



*Gambar 27. Pemetaan Objek Paraf Koreksi*



*Gambar 28. Pemetaan Objek Tanda Tangan*



*Gambar 29. Pemetaan Objek Melihat Disposisi*

## **Diagram Sequence**



*Gambar 30. Diagram Sequence Login*

*Gambar 31. Diagram Sequence Input Data Surat Masuk*

**

*Gambar 32. Diagram Sequence Mengisi Disposis*



*Gambar 33. Diagram Sequence Memberi Perintah ke Kepala Unit*



*Gambar 34. Diagram Sequence Menerima Notifikasi*



*Gambar 35. Diagram Sequence Membuat Surat Keluar*



*Gambar 36. Diagram Sequence Melakukan Verifikasi*



*Gambar 37. Diagram Sequence Logout*



*Gambar 38. Diagram Sequence Memberi Paraf Koreksi*

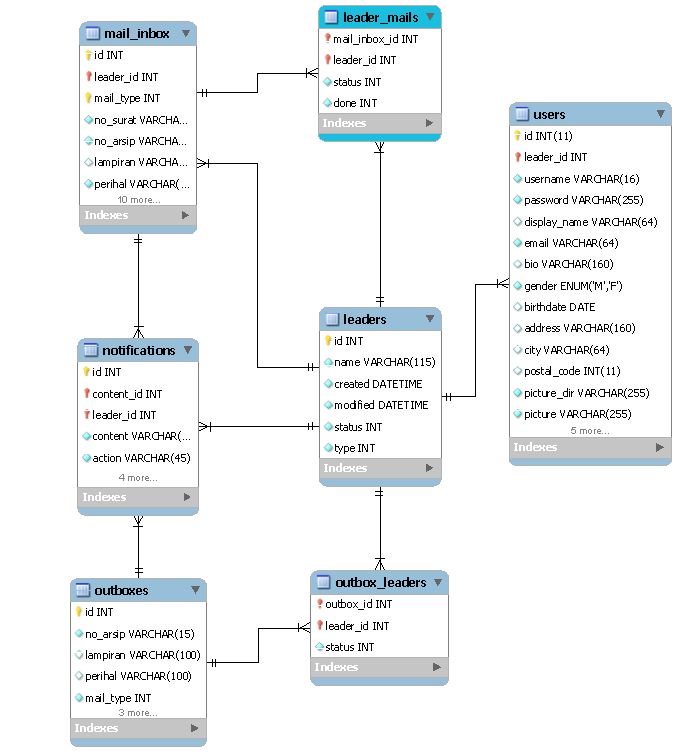


*Gambar 39. Diagram Sequence Memberi Tanda Tangan*



*Gambar 40. Diagram Sequence Melihat Disposisi*

## **ERD**



*Gambar 41. ERD*

## **Diagram Kelas**



*Gambar 42. Class Diagram*

# **IMPLEMENTASI**

# **Konstruksi**

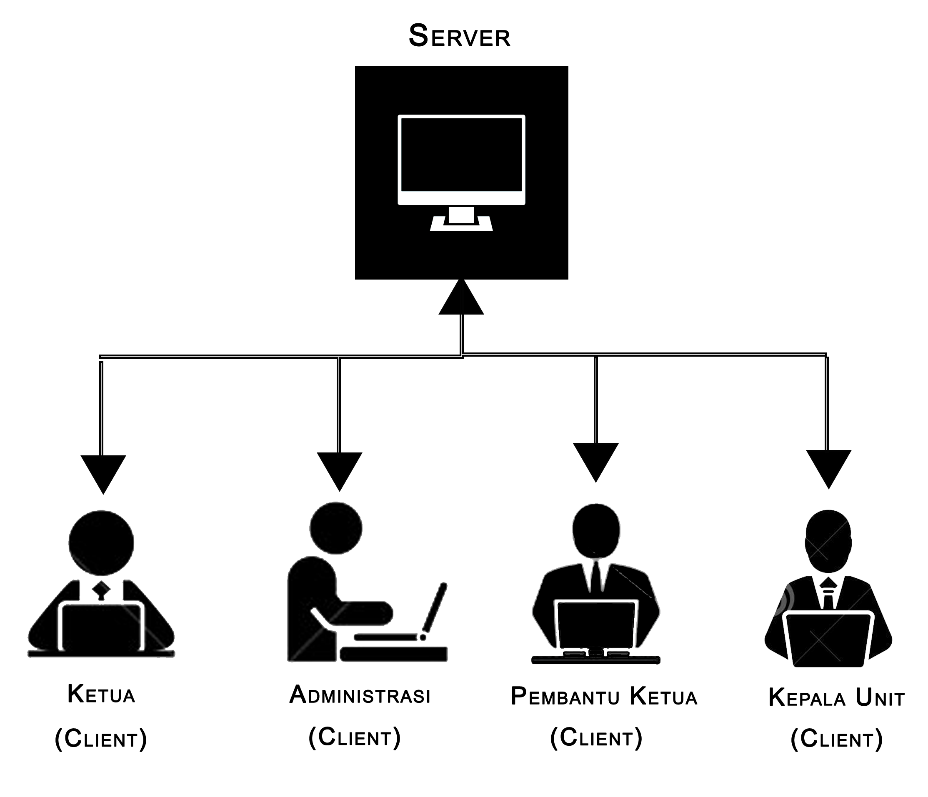
## **Diagram Deployment**

Diagram Deployment berisi tentang letak secara fisik dari sebuah Sistem. Dalam sebuah Sistem yang dibangun terdapat sebuah deployment diagram. Untuk penjelasan lebih lanjut dapat dilihat gambar dibawah :

*Gambar 43. Deployment Diagram*

Berdasarkan gambar diatas dapat diketahui bahwa Sistem ini dibangun dengan jaringan client-server. Dimana server sebagai tempat penyimpan data yang dibutuhkan oleh client. Pada gambar diatas dapat dijelaskan dimana dalam searver terdapat operating Sistem dan software database yang terinstal, pada client terdapat Sistem operasi dan Sistem agendaris surat berbasis web yang dapat dibuka dengan browser.

## **Topologi Sistem**



*Gambar 44. Topologi Sistem*

## **Pemrograman**

 Dalam membangun Sistem Agendaris Surat dan Notifikasi terdapat beberapa komponen yang dibutuhkan dapat dilihat pada gambar dibawah ini :

*Gambar 45. Component Diagram*

## **Source Code**

**Controller AppController.php**

<?php App::uses('Controller', 'Controller');

class AppController extends Controller {

const DEFAULT\_LANGUAGE = 'eng'; public $components = array(

'DebugKit.Toolbar', 'Paginator', 'Session', 'Cookie', 'Auth' => array(

'loginRedirect' => array( 'controller' => 'home', 'action' => 'index',

'leader' => false, 'assistant' => false, 'admin' => false, 'unit' => false

), 'logoutRedirect' => array( 'controller' => 'users', 'action' =>

'login', 'leader' => false, 'assistant' => false, 'admin' => false, 'unit'

=> false

), 'authorize' => array( 'Controller' ),

'authenticate' => array( 'Form' => array(

'scope' => array( 'User.status' => 0) ) )

) );

public $page\_types = array( 'About Me' => 'About Us', 'FAQ' => 'FAQ',

'Privacy Policy' => 'Privacy Policy', 'Career' => 'Career', 'Terms and

Conditions' => 'Terms and Condition', 'Health Info' => 'Health Info',

'Copyright Statement' => 'Copyright Statement', 'Site Map' => 'Site Map'

);

public $auth\_role = null; public $auth\_leader\_id = null; public

$auth\_id = null;

public $helpers = array( 'Html', 'Form', 'Text', 'Session', 'Paginator',

'Js' );

public function isAuthorized( $user ) {

if( isset( $this->request->params[ 'admin' ] ) ) { return (bool) (

$user[ 'role' ] == 'admin' );

} elseif( isset( $this->request->params[ 'leader' ] ) ) {

return (bool) ( $user[ 'role' ] == 'leader' );

} elseif( isset( $this->request->params[ 'assistant' ] ) ) {

return (bool) ( $user[ 'role' ] == 'assistant' );

} elseif( isset( $this->request->params[ 'unit' ] ) ) {

return (bool) ( $user[ 'role' ] == 'unit' ); }

// Default deny return false;

public function beforeFilter() { //$this->data = null; // Set Language

$this->\_setLanguage(); $this->\_checkRoute();

$this->\_checkLanguageForFields();

$this->set( 'lang', $this->Session->read( 'Config.language' ) );

);

if( isset( $this->request->params[ 'admin' ] ) ) {

$this->\_notificationDatas(); $this->theme = 'Admin'; } elseif(

isset( $this->request->params[ 'leader' ] ) ){ $this->theme =

'Leader'; } elseif( isset( $this->request->params[ 'assistant' ] ) ){

$this->theme = 'Assistant'; } elseif( isset( $this->request->params[

'unit' ] ) ){ $this->theme = 'Unit'; } $this->layout = 'default';

/\* Authentication \*/ /\*\*/$this->Auth->allow(

'login' );

if( $this->Auth->loggedIn() ) { $auth\_logged\_in =

$this->Auth->loggedIn(); $auth\_data = $this->Auth->user();

$auth\_id = $this->auth\_id = $auth\_data[ 'id' ]; $auth\_role =

$this->auth\_role = $auth\_data[ 'role' ]; $auth\_display\_name =

$auth\_data[ 'display\_name' ]; $auth\_username = $auth\_data[

'email' ]; $auth\_picture\_dir = '/files/user/picture/' . $auth\_id

. '/'; //$auth\_picture\_dir = '/files/user/picture/' .

$auth\_data[ 'picture\_dir' ] . '/'; $auth\_picture =

$auth\_picture\_dir . 'thumb\_' . $auth\_data[ 'picture' ]; //.

$auth\_data[ 'picture' ];

$this->auth\_leader\_id = $auth\_data[ 'leader\_id' ];

//$this->auth\_leader\_id = $auth\_data[ 'leader\_id' ];

$this->set( compact(

'auth\_data', 'auth\_id', 'auth\_role',

'auth\_display\_name', 'auth\_username',

'auth\_picture' ) ); }

/\* Global Variables \*/ $var\_controller =

$this->request->controller; $var\_action = $this->action;

//$statuses = array( TEXT\_ENABLED, TEXT\_DISABLED );

$statuses = array( \_\_( 'Enable' ), \_\_( 'Disable' ) );

$confirm\_statuses = array( \_\_( 'Yes' ), \_\_( 'No' ) ); gender

$= array( 'M' => \_\_( 'Male' ), 'F' => \_\_( 'Female' ) );

$confirmed\_status = array( \_\_( 'Unconfirmed' ), \_\_( 'Confirmed' ) );

$this->set( compact( 'var\_controller',

'var\_action', 'statuses', 'confirm\_statuses',

'types', 'gender' ) );

public function \_setLanguage() {

if ($this->Cookie->read('lang') &&

!$this->Session->check('Config.language')) {

$this->Session->write('Config.language', $this->Cookie->read('lang'));

} else if (isset($this->params['language']) &&

($this->params['language'] !=

$this->Session->read('Config.language'))) {

$this->Session->write('Config.language',

$this->params['language']); $this->Cookie->write('lang',

$this->params['language'], false, '20 days'); } } public

function \_checkRoute() { $params = $this->params['pass'];

$url = $this->here;

if (strpos($url, 'language:ind')) {

$this->Session->write('Config.language', 'ind');

Configure::write('Config.language', 'ind'); }

elseif (strpos($url, 'language:eng')) {

Configure::write('Config.language', 'eng');

$this->Session->write('Config.language', 'eng'); }

public function \_checkLanguageForFields() {

if( $this->Session->read( 'Config.language' ) == 'eng' ) {

$this->field\_description = 'description'; $this->field\_title

= 'title'; } else { $this->field\_description =

'description\_in\_indonesian as description'; $this->field\_title

= 'title\_in\_indonesian as title'; }

public function \_notificationDatas( $leader\_id = null ) {

$this->loadModel( 'Notification' );

$conditions = array(); options[ 'order' ] = 'Notification.id DESC';

if ( $leader\_id != null ) $conditions = array(

'Notification.leader\_id' => $this->auth\_leader\_id );

$options[ 'conditions' ] = array\_merge( $conditions, array(

$'Notification.status' => 0 ) ); notifications =

$$this->Notification->find( 'all', $options );

$text\_notification = array( 'not\_approved' => 'tidak

menyetujui', 'approved' => 'menyetujui', 'outboxes' =>

'surat keluar', 'MailInbox' => 'surat masuk/disposisi',

'add' => 'mendapatkan' ); $this->set( compact(

'notifications', 'text\_notification' ) ); } }

**Model AppModel.php**

<?php

App::uses('Model', 'Model');

/\*\* \* Application model for Cake. \* \* Add your application-wide methods in

the class below, your models \* will inherit them. \* \* @package

app.Model \*/ class AppModel extends Model { public $field\_title =

'title'; public $field\_name = 'name'; public $field\_description =

'description';

public function beforeFilter() {

if( $this->Session->read( 'Config.language' ) == 'ind' ) {

$this->field\_description = 'description\_in\_indonesian as description';

$this->field\_title = 'title\_in\_indonesian as title';

$this->field\_name = 'name\_in\_indonesian as name'; }

return true; } }

**Admin MailInbox admin\_index.ctp**

<div class="btn-toolbar">

<div class="btn-group">

<a href="javascript:history.go(-1);" class="btn btn-sm btn-

white tooltips" data-toggle="tooltip" data-original-title="<?php echo \_\_(

TOOLTIP\_PREVIOUS ); ?>"> <i class="fa fa-chevron-

left"></i> &nbsp; <?php echo \_\_( BTN\_PREVIOUS ); ?> </a>

<a href="<?php echo $this->here; ?>" class="btn btn-sm btn-

white tooltips" data-toggle="tooltip" data-original-title="<?php echo \_\_(

TOOLTIP\_REFRESH ); ?>"> <i class="fa fa-refresh"></i>

&nbsp; <?php echo \_\_( BTN\_REFRESH ); ?> </a>

</div><!--/ .btn-group -->

<div class="btn-group">

<?php echo $this->Html->link(

'<i class="fa fa-plus"></i> &nbsp; ' . \_\_( BTN\_ADD\_NEW ),

array( 'controller' => $var\_controller,

'action' => ACTION\_ADD, ),

array( 'class' => 'btn btn-sm btn-white tooltips',

'data-toggle' => 'tooltip', 'data-original-title'

=> \_\_( TOOLTIP\_ADD\_NEW ), 'tabindex' => 1,

'escape' => false ) );

?>

</div>

<div class="btn-group">

<?php echo $this->Html->link(

'<i class="fa fa-sort"></i> &nbsp; ' . \_\_( BTN\_SORT\_BY ),

array( 'controller' => $var\_controller,

'action' => ACTION\_ADD, ),

array( 'class' => 'btn btn-sm btn-white tooltips',

'data-toggle' => 'tooltip', 'data-original-title'

=> \_\_( TOOLTIP\_SORT\_BY ), 'tabindex' => 1,

'escape' => false ) );

?>

<button type="button" class="btn btn-sm btn-white dropdown-

toggle" data-toggle="dropdown"> <span

class="caret"></span> </button>

<ul class="dropdown-menu" role="menu">

<li><?php echo $this->Paginator->sort( $var\_model . '.id', 'Id' ); ?></li>

<li><?php echo $this->Paginator->sort( $var\_model . '.perihal', \_\_( 'Perihal'

) ); ?></li> <li><?php echo $this->Paginator->sort(

$var\_model . '.leader\_status', \_\_( 'Status' ) ); ?></li>

<li><?php echo $this->Paginator->sort( $var\_model . '.received\_date', \_\_(

'Received Date' ) ); ?></li> </ul>

</div><!--/ .btn-group -->

</div><!--/ .btn-toolbar -->

<br />

<table class="table table-bordered table-hover">

<tbody>

<?php foreach( $datas as $data ):

$row = $data[ $var\_model ]; id = $row[ 'id'

$];

$pesan\_ketua = substr( $row[ 'message\_leader' ],

0, 120 ); $created = date( 'D, d F Y H:i:s',

strtotime( $row[ 'created' ] ) ); $published =

date( 'D, d F Y', strtotime( $row[ 'received\_date' ] ) );

$limit\_date = date( 'D, d F Y', strtotime( $row[ 'limit\_date' ] ) );

$status = $row[ 'leader\_status' ]; ?>

<tr class="<?= bootstrap\_row\_status\_type2( $status ) ?>">

<td> <h4> <?php

echo $this->Html->link( $row[

'perihal' ], array(

'controller' => $var\_controller,

'action' => 'read', $id

) ); ?>

/ <small><?= $mail\_types[ $row[ 'mail\_type' ]

] ?></small> </h4>

<p>Lampiran: <br><small><?= $row[ 'lampiran' ] ?></small></p>

<small> <i

class="fa fa-calendar"></i> Dibuat: &nbsp; <?= $created ?> &nbsp;

<i class="fa fa-calendar"></i> Tanggal akhir: &nbsp; <?= $limit\_date ?> &nbsp;

<i class="fa fa-user"></i> Dari: &nbsp; <?= $row[ 'asal\_surat' ]; ?> &nbsp;

<i class="fa fa-bullhorn"></i> Status: &nbsp; <?= text\_approved( $status ); ?>

&nbsp; </small> </td>

</tr>

<?php endforeach; ?> </tbody> </table>

</div><!--/ .table-responsive -->

<?php echo $this->Element( 'Administrator/pagination' ); ?>

**Admin Outboxes admin\_index.ctp**

<div class="panel-body">

<div class="btn-toolbar">

<div class="btn-group">

<a href="javascript:history.go(-1);" class="btn btn-sm btn-

white tooltips" data-toggle="tooltip" data-original-title="<?php echo \_\_(

TOOLTIP\_PREVIOUS ); ?>"> <i class="fa fa-chevron-

left"></i> &nbsp; <?php echo \_\_( BTN\_PREVIOUS ); ?> </a>

<a href="<?php echo $this->here; ?>" class="btn btn-sm btn-

white tooltips" data-toggle="tooltip" data-original-title="<?php echo \_\_(

TOOLTIP\_REFRESH ); ?>"> <i class="fa fa-refresh"></i>

&nbsp; <?php echo \_\_( BTN\_REFRESH ); ?> </a>

</div><!--/ .btn-group -->

<div class="btn-group">

<?php echo $this->Html->link(

'<i class="fa fa-plus"></i> &nbsp; ' . \_\_( BTN\_ADD\_NEW ),

array( 'controller' => $var\_controller,

'action' => ACTION\_ADD, ),

array( 'class' => 'btn btn-sm btn-white tooltips',

'data-toggle' => 'tooltip', 'data-original-title'

=> \_\_( TOOLTIP\_ADD\_NEW ), 'tabindex' => 1,

'escape' => false ) );

?>

</div>

<div class="btn-group">

<?php echo $this->Html->link(

'<i class="fa fa-sort"></i> &nbsp; ' . \_\_( BTN\_SORT\_BY ),

array( 'controller' => $var\_controller,

'action' => ACTION\_ADD, ),

array( 'class' => 'btn btn-sm btn-white tooltips',

'data-toggle' => 'tooltip', 'data-original-title'

=> \_\_( TOOLTIP\_SORT\_BY ), 'tabindex' => 1,

'escape' => false ) );

?>

<button type="button" class="btn btn-sm btn-white dropdown-

toggle" data-toggle="dropdown"> <span

class="caret"></span> </button>

<ul class="dropdown-menu" role="menu">

<li><?php echo $this->Paginator->sort( $var\_model . '.perihal', \_\_( 'Perihal'

) ); ?></li> <li><?php echo $this->Paginator->sort(

$var\_model . '.created', 'Tanggal' ); ?></li> <li><?php

echo $this->Paginator->sort( $var\_model . '.purpose', \_\_( 'Tujuan Surat' ) );

?></li> </ul>

</div><!--/ .btn-group -->

</div><!--/ .btn-toolbar -->

<br />

<table class="table table-bordered table-hover">

<tbody>

<?php foreach( $datas as $data ):

$row = $data[ $var\_model ]; id = $row[ 'id'

$];

$created = date( 'D, d F Y', strtotime( $row[

'created' ] ) ); ?>

<tr class=""> <td>

<h4> <?php

echo $this->Html->link( $row[

'perihal' ], array(

'controller' => $var\_controller,

'action' => 'read', $id

) ); ?> /

<small><?= $mail\_types[ $row[ 'mail\_type' ] ] ?></small>

</h4> <p>Tujuan: <small><?= $getPurposes[ $row[

'purpose' ] ] ?></small>; &nbsp; Lampiran: <small><?= $row[ 'lampiran' ]

?></small></p> <p>Untuk: <small>

<?php foreach( $data[ 'Leader' ] as $leader ):

$status = $leader[ 'OutboxLeader' ][

'status' ]; echo $leader[ 'name' ] . ' ( '

.text\_approved( $status ) . ' ) ' . " &nbsp; / &nbsp; ";

endforeach; ?>

</small></p> <small>

<i class="fa fa-calendar"> &nbsp; <?= $created ?></i>

</small> </td> </tr>

<?php endforeach; ?>

</tbody> </table>

<?php echo $this->Element( 'Administrator/pagination' ); ?>

</div><!--/ .panel-body -->

**Assistant MailInboxes assistant\_index.ctp**

<div class="btn-toolbar">

<div class="btn-group">

<a href="javascript:history.go(-1);" class="btn btn-sm btn-

white tooltips" data-toggle="tooltip" data-original-title="<?php echo \_\_(

TOOLTIP\_PREVIOUS ); ?>"> <i class="fa fa-chevron-

left"></i> &nbsp; <?php echo \_\_( BTN\_PREVIOUS ); ?> </a>

<a href="<?php echo $this->here; ?>" class="btn btn-sm btn-

white tooltips" data-toggle="tooltip" data-original-title="<?php echo \_\_(

TOOLTIP\_REFRESH ); ?>"> <i class="fa fa-refresh"></i>

&nbsp; <?php echo \_\_( BTN\_REFRESH ); ?> </a>

</div><!--/ .btn-group -->

<div class="btn-group">

<?php echo $this->Html->link(

'<i class="fa fa-sort"></i> &nbsp; ' . \_\_( BTN\_SORT\_BY ),

array( 'controller' => $var\_controller,

'action' => ACTION\_ADD, ),

array( 'class' => 'btn btn-sm btn-white tooltips',

'data-toggle' => 'tooltip', 'data-original-title'

=> \_\_( TOOLTIP\_SORT\_BY ), 'tabindex' => 1,

'escape' => false ) );

?>

<button type="button" class="btn btn-sm btn-white dropdown-

toggle" data-toggle="dropdown"> <span

class="caret"></span> </button>

<ul class="dropdown-menu" role="menu">

<li><?php echo $this->Paginator->sort( $var\_model . '.id', 'Id' ); ?></li>

<li><?php echo $this->Paginator->sort( $var\_model . '.perihal', \_\_( 'Perihal'

) ); ?></li> <li><?php echo $this->Paginator->sort(

'LeaderMail.status', \_\_( 'Status' ) ); ?></li> <li><?php

echo $this->Paginator->sort( $var\_model . '.received\_date', \_\_( 'Received

Date' ) ); ?></li> </ul>

</div><!--/ .btn-group -->

</div><!--/ .btn-toolbar -->

<br />

<table class="table table-bordered table-hover">

<tbody>

<?php foreach( $datas as $data ):

$row = $data[ $var\_model ]; id = $row[ 'id'

$];

$pesan\_ketua = substr( $row[ 'message\_leader' ],

0, 120 ); $created = date( 'D, d F Y H:i:s',

strtotime( $row[ 'created' ] ) ); $published =

date( 'D, d F Y', strtotime( $row[ 'received\_date' ] ) );

$limit\_date = date( 'D, d F Y', strtotime( $row[ 'limit\_date' ] ) );

$status = $data[ 'LeaderMail' ][ 0 ][ 'status' ]; ?>

<tr class="<?= bootstrap\_row\_status\_type2( $status ) ?>">

<td> <h4> <?php

echo $this->Html->link( $row[

'perihal' ], array(

'controller' => $var\_controller,

'action' => 'read', $id

) ); ?>

/ <small><?= $mail\_types[ $row[ 'mail\_type' ]

] ?></small> </h4>

<p>Pesan ketua: <br><small><?= $pesan\_ketua ?></small></p>

<small> <i

class="fa fa-calendar"></i> Tanggal akhir: &nbsp; <?= $limit\_date ?> &nbsp;

<i class="fa fa-calendar"></i> Dibuat: &nbsp; <?= $created ?> &nbsp;

<i class="fa fa-user"></i> Dari: &nbsp; <?= $row[ 'asal\_surat' ]; ?> &nbsp;

<i class="fa fa-bullhorn"></i> Dari: &nbsp; <?= text\_approved( $status ); ?>

&nbsp;

</small> </td>

</tr>

<?php endforeach; ?>

</tbody> </table>

<?php echo $this->Element( 'Administrator/pagination' ); ?>

**Assistant Outboxes assistant\_index.ctp**

<div class="btn-toolbar">

<div class="btn-group">

<a href="javascript:history.go(-1);" class="btn btn-sm btn-

white tooltips" data-toggle="tooltip" data-original-title="<?php echo \_\_(

TOOLTIP\_PREVIOUS ); ?>"> <i class="fa fa-chevron-

left"></i> &nbsp; <?php echo \_\_( BTN\_PREVIOUS ); ?> </a>

<a href="<?php echo $this->here; ?>" class="btn btn-sm btn-

white tooltips" data-toggle="tooltip" data-original-title="<?php echo \_\_(

TOOLTIP\_REFRESH ); ?>"> <i class="fa fa-refresh"></i>

&nbsp; <?php echo \_\_( BTN\_REFRESH ); ?> </a>

</div><!--/ .btn-group -->

<div class="btn-group">

<?php echo $this->Html->link(

'<i class="fa fa-sort"></i> &nbsp; ' . \_\_( BTN\_SORT\_BY ),

array( 'controller' => $var\_controller,

'action' => ACTION\_ADD, ),

array( 'class' => 'btn btn-sm btn-white tooltips',

'data-toggle' => 'tooltip', 'data-original-title'

=> \_\_( TOOLTIP\_SORT\_BY ), 'tabindex' => 1,

'escape' => false ) );

?>

<button type="button" class="btn btn-sm btn-white dropdown-

toggle" data-toggle="dropdown"> <span

class="caret"></span> </button>

<ul class="dropdown-menu" role="menu">

<li><?php echo $this->Paginator->sort( $var\_model . '.perihal', \_\_( 'Perihal'

) ); ?></li> <li><?php echo $this->Paginator->sort(

$var\_model . '.created', \_\_( 'Tanggal' ) ); ?></li> </ul>

</div><!--/ .btn-group -->

</div><!--/ .btn-toolbar -->

<br />

<table class="table table-bordered table-hover">

<thead>

</thead>

<tbody>

<?php foreach( $datas as $data ):

$row = $data[ $var\_model ]; id = $row[ 'id'

$];

$created = date( 'D, d F Y H:i:s', strtotime(

$row[ 'created' ] ) ); foreach( $data[

'OutboxLeader' ] as $outbox ) {

if( $outbox[ 'leader\_id' ] == $auth\_data[ 'leader\_id' ] )

{ $status = $outbox[ 'status' ];

} }

?>

<tr class="<?= bootstrap\_row\_status\_type2( $status ) ?>">

<td> <h4> <?php

echo $this->Html->link( \_\_( $row[

'perihal' ] ), array(

'controller' => $var\_controller,

'action' => 'read', $id

) ); ?>

/ <small>

<?php echo text\_approved( $status ) ;

?>

</small> </h4>

<p>Lampiran: <small><?= $row[ 'lampiran' ] ?></small>; &nbsp; Tujuan:

<small><?= $getPurposes[ $row[ 'purpose' ] ]; ?></small>; &nbsp; Tipe Surat :

<small><?= $mail\_types[ $row [ 'mail\_type' ] ]; ?></small> </p>

<p> <?php

foreach( $data[ 'Leader' ] as $data ):

$leader = $data[ 'OutboxLeader' ]; $status

= $leader[ 'status' ]; echo $data[ 'name'

] . " ( " . text\_approved( $status ) . " ), &nbsp ";

endforeach; ?>

</p> <o><?php echo $created; ?></o>

</td> </tr>

<?php endforeach; ?>

</tbody> </table>

<?php echo $this->Element( 'Administrator/pagination' ); ?>

**Leader MailInboxes leader\_index.ctp**

<div class="panel-body">

<div class="btn-toolbar">

<div class="btn-group">

<a href="javascript:history.go(-1);" class="btn btn-sm btn-

white tooltips" data-toggle="tooltip" data-original-title="<?php echo \_\_(

TOOLTIP\_PREVIOUS ); ?>"> <i class="fa fa-chevron-

left"></i> &nbsp; <?php echo \_\_( BTN\_PREVIOUS ); ?> </a>

<a href="<?php echo $this->here; ?>" class="btn btn-sm btn-

white tooltips" data-toggle="tooltip" data-original-title="<?php echo \_\_(

TOOLTIP\_REFRESH ); ?>"> <i class="fa fa-refresh"></i>

&nbsp; <?php echo \_\_( BTN\_REFRESH ); ?> </a>

</div><!--/ .btn-group -->

<div class="btn-group">

<?php echo $this->Html->link(

'<i class="fa fa-sort"></i> &nbsp; ' . \_\_( BTN\_SORT\_BY ),

array( 'controller' => $var\_controller,

'action' => ACTION\_ADD, ),

array( 'class' => 'btn btn-sm btn-white tooltips',

'data-toggle' => 'tooltip', 'data-original-title'

=> \_\_( TOOLTIP\_SORT\_BY ), 'tabindex' => 1,

'escape' => false ) );

?>

<button type="button" class="btn btn-sm btn-white dropdown-

toggle" data-toggle="dropdown"> <span

class="caret"></span> </button>

<ul class="dropdown-menu" role="menu">

<li><?php echo $this->Paginator->sort( $var\_model . '.id', 'Id' ); ?></li>

<li><?php echo $this->Paginator->sort( $var\_model . '.perihal', \_\_( 'Perihal'

) ); ?></li> <li><?php echo $this->Paginator->sort(

$var\_model . '.leader\_status', \_\_( 'Status' ) ); ?></li>

<li><?php echo $this->Paginator->sort( $var\_model . '.received\_date', \_\_(

'Received Date' ) ); ?></li> </ul>

</div><!--/ .btn-group -->

</div><!--/ .btn-toolbar -->

<br />

<div class="table-responsive">

<table class="table table-bordered table-hover">

<tbody>

<?php foreach( $datas as $data ):

$row = $data[ $var\_model ]; id = $row[ 'id'

$];

$pesan\_ketua = substr( $row[ 'message\_leader' ],

0, 120 ); $created = date( 'D, d F Y',

strtotime( $row[ 'created' ] ) ); $published =

date( 'D, d F Y', strtotime( $row[ 'received\_date' ] ) );

$limit\_date = date( 'D, d F Y', strtotime( $row[ 'limit\_date' ] ) );

$status = $row[ 'leader\_status' ]; ?>

<tr class="<?= bootstrap\_row\_status\_type2( $status ) ?>">

<td> <h4> <?php

echo $this->Html->link( $row[

'perihal' ], array(

'controller' => $var\_controller,

'action' => ACTION\_EDIT, $id )

);

?> /

<small><?= $mail\_types[ $row[ 'mail\_type' ] ] ?></small>

</h4> <p>Lampiran: <br><small><?= $row[ 'lampiran'

] ?></small></p>

<small> <i

class="fa fa-calendar"> Tanggal Diterima : &nbsp; <?= $created ?></i> &nbsp;

<i class="fa fa-calendar"></i> Tanggal akhir : &nbsp; <?= $limit\_date ?>

&nbsp; <i class="fa fa-user"></i> Dari: &nbsp;

<?= $row[ 'asal\_surat' ]; ?> &nbsp; Status :

&nbsp;<?= text\_approved( $status ); ?> </small>

</td> </tr>

<?php endforeach; ?>

</tbody> </table>

</div><!--/ .table-responsive -->

<?php echo $this->Element( 'Administrator/pagination' ); ?>

</div><!--/ .panel-body -->

**Unit MailInboxes unit\_index.ctp**

<div class="btn-toolbar">

<div class="btn-group">

<a href="javascript:history.go(-1);" class="btn btn-sm btn-

white tooltips" data-toggle="tooltip" data-original-title="<?php echo \_\_(

TOOLTIP\_PREVIOUS ); ?>"> <i class="fa fa-chevron-

left"></i> &nbsp; <?php echo \_\_( BTN\_PREVIOUS ); ?> </a>

<a href="<?php echo $this->here; ?>" class="btn btn-sm btn-

white tooltips" data-toggle="tooltip" data-original-title="<?php echo \_\_(

TOOLTIP\_REFRESH ); ?>"> <i class="fa fa-refresh"></i>

&nbsp; <?php echo \_\_( BTN\_REFRESH ); ?> </a>

</div><!--/ .btn-group -->

<div class="btn-group">

<?php echo $this->Html->link(

'<i class="fa fa-sort"></i> &nbsp; ' . \_\_( BTN\_SORT\_BY ),

array( 'controller' => $var\_controller,

'action' => ACTION\_ADD, ),

array( 'class' => 'btn btn-sm btn-white tooltips',

'data-toggle' => 'tooltip', 'data-original-title'

=> \_\_( TOOLTIP\_SORT\_BY ), 'tabindex' => 1,

'escape' => false ) );

?>

<button type="button" class="btn btn-sm btn-white dropdown-

toggle" data-toggle="dropdown"> <span

class="caret"></span> </button>

<ul class="dropdown-menu" role="menu">

<li><?php echo $this->Paginator->sort( $var\_model . '.id', 'Id' ); ?></li>

<li><?php echo $this->Paginator->sort( $var\_model . '.perihal', \_\_( 'Perihal'

) ); ?></li> <li><?php echo $this->Paginator->sort(

'LeaderMail.status', \_\_( 'Status' ) ); ?></li> <li><?php

echo $this->Paginator->sort( $var\_model . '.received\_date', \_\_( 'Received

Date' ) ); ?></li> </ul>

</div><!--/ .btn-group -->

</div><!--/ .btn-toolbar -->

<br />

<table class="table table-bordered table-hover">

<tbody>

<?php foreach( $datas as $data ):

$row = $data[ $var\_model ]; id = $row[ 'id'

$];

$pesan\_ketua = substr( $row[ 'message\_leader' ],

0, 120 ); $created = date( 'D, d F Y H:i:s',

strtotime( $row[ 'created' ] ) ); $published =

date( 'D, d F Y', strtotime( $row[ 'received\_date' ] ) );

$limit\_date = date( 'D, d F Y', strtotime( $row[ 'limit\_date' ] ) );

$status = $data[ 'LeaderMail' ][ 0 ][ 'status' ]; ?>

<tr class="<?= bootstrap\_row\_status\_type2( $status ) ?>">

<td> <h4> <?php

echo $this->Html->link( $row[

'perihal' ], array(

'controller' => $var\_controller,

'action' => 'read', $id

) ); ?>

/ <small><?= $mail\_types[ $row[ 'mail\_type' ]

] ?></small> </h4>

<p>Pesan ketua: <br><small><?= $pesan\_ketua ?></small></p>

<small> <i

class="fa fa-calendar"></i> Tanggal akhir: &nbsp; <?= $limit\_date ?> &nbsp;<i class="fa fa-calendar"></i> Dibuat: &nbsp; <?= $created ?> &nbsp;<i class="fa fa-user"></i> Dari: &nbsp; <?= $row[ 'asal\_surat' ]; ?> &nbsp;

<i class="fa fa-bullhorn"></i> Dari: &nbsp; <?= text\_approved( $status ); ?>

&nbsp;

</small> </td>

</tr>

<?php endforeach; ?>

</tbody> </table>

<?php echo $this->Element( 'Administrator/pagination' ); ?>

**Unit Outboxes unit\_index.ctp**

<div class="btn-toolbar">

<div class="btn-group">

<a href="javascript:history.go(-1);" class="btn btn-sm btn-

white tooltips" data-toggle="tooltip" data-original-title="<?php echo \_\_(

TOOLTIP\_PREVIOUS ); ?>"> <i class="fa fa-chevron-

left"></i> &nbsp; <?php echo \_\_( BTN\_PREVIOUS ); ?> </a>

<a href="<?php echo $this->here; ?>" class="btn btn-sm btn-

white tooltips" data-toggle="tooltip" data-original-title="<?php echo \_\_(

TOOLTIP\_REFRESH ); ?>"> <i class="fa fa-refresh"></i>

&nbsp; <?php echo \_\_( BTN\_REFRESH ); ?> </a>

</div><!--/ .btn-group -->

<div class="btn-group">

<?php echo $this->Html->link(

'<i class="fa fa-plus"></i> &nbsp; ' . \_\_( BTN\_ADD\_NEW ),

array( 'controller' => $var\_controller,

'action' => ACTION\_ADD, ),

array( 'class' => 'btn btn-sm btn-white tooltips',

'data-toggle' => 'tooltip', 'data-original-title'

=> \_\_( TOOLTIP\_ADD\_NEW ), 'tabindex' => 1,

'escape' => false ) );

?>

</div>

<div class="btn-group">

<?php echo $this->Html->link(

'<i class="fa fa-sort"></i> &nbsp; ' . \_\_( BTN\_SORT\_BY ),

array( 'controller' => $var\_controller,

'action' => ACTION\_ADD, ),

array( 'class' => 'btn btn-sm btn-white tooltips',

'data-toggle' => 'tooltip', 'data-original-title'

=> \_\_( TOOLTIP\_SORT\_BY ), 'tabindex' => 1,

'escape' => false ) );

?>

<button type="button" class="btn btn-sm btn-white dropdown-

toggle" data-toggle="dropdown"> <span

class="caret"></span> </button>

<ul class="dropdown-menu" role="menu">

<li><?php echo $this->Paginator->sort( $var\_model . '.id', \_\_( 'Tanggal' ) );

?></li> <li><?php echo $this->Paginator->sort( $var\_model

. '.perihal', \_\_( 'Perihal' ) ); ?></li> </ul>

</div><!--/ .btn-group -->

</div><!--/ .btn-toolbar -->

<br />

<table class="table table-bordered table-hover">

<tbody>

<?php foreach( $datas as $data ):

$row = $data[ $var\_model ]; id = $row[ 'id'

$];

$created = date( 'D, d F Y', strtotime( $row[

'created' ] ) ); ?>

<tr class=""> <td>

<h4> <?php

echo $this->Html->link( \_\_( $row[

'perihal' ] ), array(

'controller' => $var\_controller,

'action' => 'read', $id

) ); ?>

</h4> <p>Lampiran: <br><small><?= $row[ 'lampiran'

] ?></small></p> <p> Tipe Surat : <small><?=

$mail\_types[ $row [ 'mail\_type' ] ]; ?></small> </p>

<small> <i class="fa fa-calendar"></i>

&nbsp;Tanggal dibuat : <?= $created ?> &nbsp;

<i class="fa fa-bullhorn"></i> &nbsp;Status :

<?php foreach( $data[ 'OutboxLeader' ]

as $outbox ) {

if( $outbox[ 'leader\_id' ] == $auth\_data[ 'leader\_id' ] )

{ $status = $outbox[ 'status'

]; echo text\_approved( $status

) ; }

} ?> </small>

</td> </tr>

<?php endforeach; ?>

</tbody> </table>

<?php echo $this->Element( 'Administrator/pagination' ); ?>

**Controller MailInboxesController.php**

<?php

class MailInboxesController extends AppController {

public $model\_name = 'MailInbox'; public $module\_title =

'Disposisi'; public $module\_desc = ''; public $module\_icon =

'fa fa-envelope-o';

public function beforeFilter() { parent::beforeFilter();

//$this->Auth->allow( 'add', 'logout' );

$var\_model = $this->model\_name; module\_title =

$$this->module\_title; module\_desc = $this->module\_title;

$module\_icon = $this->module\_icon; readed\_status = array( \_\_(

$TEXT\_READ ), \_\_( TEXT\_UNREAD ) ); mail\_types =

$$this->MailInbox->getMailTypes(); this->loadModel( 'Leader' );

$this->loadModel( 'LeaderMail' ); this->loadModel( 'Notification'

$);

$this->set( compact( 'var\_model', 'module\_title', 'module\_desc',

$'module\_icon', 'readed\_status', 'mail\_types' ) );

$leader\_assistants = $this->Leader->find( 'list', array(

'conditions' => array( 'Leader.type' => 2 ), 'fields' =>

array( 'Leader.id', 'Leader.name' ) ) );

$leader\_units = $this->Leader->find( 'list', array(

'conditions' => array( 'Leader.type' => 3 ), 'fields' =>

array( 'Leader.id', 'Leader.name' ) ) );

$this->set( compact( 'leader\_units', 'leader\_assistants',

'mail\_types' ) ); }

public function admin\_add() { if(

$this->request->is( 'post' ) ) {

$this->MailInbox->create(); $this->request->data[ 'MailInbox'

][ 'status' ] = 1; if( $this->MailInbox->save(

$this->request->data ) ) {

$this->Session->setFlash( \_\_( MSG\_DATA\_UPDATE\_SUCCESS ), 'Bootstrap/flash-

success' ); $datas = array( array(

'sender\_id' => $this->auth\_leader\_id, 'receiver\_id'

=> 2, // ID KETUA STIKOM 'content\_id' =>

$this->MailInbox->getInsertID(), 'content' =>

'mail\_inboxes', 'action' => 'add',

'redirect' => 'edit', 'mail\_status' => 2

) ); $this->Notification->addNotif( $datas );

return $this->redirect( array( 'action' => ACTION\_INDEX ) ); }

else { $this->Session->setFLash( \_\_(

MSG\_DATA\_SAVE\_FAILED ) ); } } $created

= date( 'Y-m-d h:i:s' ); /\* $data = $this->MailInbox->find(

'first', array( 'fields' => array( 'no\_arsip' ), 'order' => 'MailInbox.id

DESC' ) ); $value = isset( $data[ 'MailInbox' ][ 'no\_arsip'] ) ?

$data[ 'MailInbox' ][ 'no\_arsip'] : 0; $len = strlen( $value );

$max = (int) substr($value, 3, $len); $max++; $newid

= 'AS-' . sprintf("%04s", $max); \*/ $data =

$this->MailInbox->find( 'first', array( 'fields' => array( 'id' ), 'order' =>

'MailInbox.id DESC' ) ); $value = isset( $data[ 'MailInbox' ][

'id'] ) ? $data[ 'MailInbox' ][ 'id'] + 1 : 1; $no\_surat =

sprintf( "%03s", $value ) . '/STIKOM-M/' . number2roman( date( 'm' ) ) . '/'

. date( 'Y' );

$newid = 'AS-' . sprintf("%04s", $value);

$leaders = $this->MailInbox->Leader->find( 'list', array(

'conditions' => array( 'type' => 1 ) ) ); $this->request->data[

'MailInbox' ][ 'no\_surat' ] = $no\_surat; $this->request->data[

'MailInbox' ][ 'no\_arsip' ] = $newid; $this->set( compact(

'leaders' ) ); }

public function admin\_index() {

//$this->layout = LAYOUT\_ADMIN;

$options[ 'contain' ] = array( 'LeaderMail' ); options[ 'order'

$] = array( 'MailInbox.id' => 'DESC' );

$this->Paginator->settings = $options; datas =

$$this->Paginator->paginate( $this->model\_name );

//$datas = $this->MailInbox->find( 'all', $options );

$this->set( compact( 'datas' ) ); }

public function admin\_edit( $id = null ) {

if( !$id ) { throw new NotFoundException( \_\_(

MSG\_DATA\_NOT\_FOUND ) ); }

$this->MailInbox->id = $id; if(

!$this->MailInbox->exists() ) { throw new

NotFoundException( \_\_( MSG\_DATA\_NOT\_FOUND ) ); }

if( $this->request->is( 'post' ) || $this->request->is( 'put' ) )

{

//$this->set( 'x', $x ); if(

$this->MailInbox->saveAssociated( $this->request->data, array( 'deep' => true

) ) ) { $this->Session->setFlash( \_\_(

MSG\_DATA\_EDIT\_SUCCESS ), 'Bootstrap/flash-success' );

return $this->redirect( array( 'action' => ACTION\_INDEX ) ); }

} else { $leaders =

$this->MailInbox->Leader->find( 'list', array( 'conditions' => array( 'type'

=> 1 ) ) ); $this->set( compact( 'leaders' ) );

$this->request->data = $this->MailInbox->read( null, $id ); }

public function admin\_read( $id = null ) {

if( !$id ) { throw new NotFoundException( \_\_(

MSG\_DATA\_NOT\_FOUND ) ); }

$this->MailInbox->id = $id; if(

!$this->MailInbox->exists() ) { throw new

NotFoundException( \_\_( MSG\_DATA\_NOT\_FOUND ) ); }

if( !$this->Notification->updateAll( array( 'status' => 1 ),

array( 'Notification.content\_id' => $id, 'Notification.content' =>

'mail\_inboxes', 'Notification.receiver\_id' =>

$this->auth\_leader\_id ) ) ) $this->Session->setFLash( \_\_(

MSG\_DATA\_SAVE\_FAILED ), 'Bootstrap/flash-error' );

$options[ 'conditions' ] = array( 'MailInbox.id'

=> $id ); $options[ 'contain' ] = array(

'LeaderMail', 'Assistant' => array(

'conditions' => array( 'Assistant.type' => 2 ) ),

'Unit' => array( 'conditions' => array( 'Unit.type' => 3 )

) ); $data = $this->MailInbox->find( 'first', $options

); $this->set( 'data', $data ); //return

$this->redirect( array( 'action' => ACTION\_INDEX ) );

public function admin\_unread( $id = null ) {

if( !$id ) { throw new NotFoundException( \_\_(

MSG\_DATA\_NOT\_FOUND ) ); }

$this->MailInbox->id = $id; if(

!$this->MailInbox->exists() ) { throw new

NotFoundException( \_\_( MSG\_DATA\_NOT\_FOUND ) ); }

if( $this->MailInbox->updateAll( array( 'status' => 1 ), array(

'id' => $id ) ) ) $this->Session->setFlash( \_\_(

MSG\_DATA\_UPDATE\_SUCCESS ), 'Bootstrap/flash-success' ); else

$this->Session->setFLash( \_\_( MSG\_DATA\_SAVE\_FAILED ) );

return $this->redirect( array( 'action' => ACTION\_INDEX ) );

public function admin\_delete( $id = null ) {

if( !$id ) { throw new NotFoundException( \_\_(

MSG\_DATA\_NOT\_FOUND ) ); }

$this->request->onlyAllow( 'post' );

$this->MailInbox->id = $id; if(

!$this->MailInbox->exists() ) { throw new

NotFoundException( \_\_( MSG\_DATA\_NOT\_FOUND ) ); }

if( $this->MailInbox->delete() ) { $this->Session->setFlash(

\_\_( MSG\_DATA\_DELETE\_SUCCESS ), 'Bootstrap/flash-success' );

return $this->redirect( array( 'action' => ACTION\_INDEX ) ); }

$this->Session->setFlash( \_\_( MSG\_DATA\_DELETE\_FAILED ), 'Bootstrap

/flash-error' ); return $this->redirect( array( 'action' =>

ACTION\_INDEX ) ); }

public function index() { $options[ 'joins' ] = array( array(

'table' => 'leader\_mails', 'alias' => 'LeaderMail', 'type' =>

'INNER', 'conditions' => array( 'LeaderMail.mail\_inbox\_id =

MailInbox.id' ) ) ); $options[ 'conditions' ] = array(

'LeaderMail.leader\_id' => $this->auth\_leader\_id ); $options[

'contain' ] = array( 'LeaderMail' => array( 'conditions' => array(

'LeaderMail.leader\_id' => $this->auth\_leader\_id ) ) ); $options[

'order' ] = array( 'MailInbox.id' => 'DESC' );

$this->Paginator->settings = $options; $datas =

$this->Paginator->paginate( $this->model\_name );

//$datas = $this->MailInbox->find( 'all', $options );

$this->set( compact( 'datas' ) ); }

public function approved( $id = null ) {

if( !$id ) { throw new NotFoundException( \_\_(

MSG\_DATA\_NOT\_FOUND ) ); }

$this->MailInbox->id = $id; if(

!$this->MailInbox->exists() ) { throw new

NotFoundException( \_\_( MSG\_DATA\_NOT\_FOUND ) ); } if(

$this->MailInbox->LeaderMail->updateAll( array( 'LeaderMail.status' => 1 ),

array( 'mail\_inbox\_id' => $id, 'LeaderMail.leader\_id' => $this->auth\_leader\_id

) ) ) $this->Session->setFlash( \_\_( MSG\_DATA\_UPDATE\_SUCCESS

), 'Bootstrap/flash-success' ); else $this->Session->setFLash( \_\_(

MSG\_DATA\_SAVE\_FAILED ) );

$this->Notification->addNotif( $id, 'mail\_inboxes', 'approved',

$'read', 1 );

//$this->set( 'data', $this->MailInbox->read( null, $id ) );

return $this->redirect( array( 'action' => ACTION\_INDEX ) );

public function not\_approved( $id = null ) {

if( !$id ) { throw new NotFoundException( \_\_(

MSG\_DATA\_NOT\_FOUND ) ); }

$this->MailInbox->id = $id; if(

!$this->MailInbox->exists() ) { throw new

NotFoundException( \_\_( MSG\_DATA\_NOT\_FOUND ) ); } if(

$this->MailInbox->LeaderMail->updateAll( array( 'LeaderMail.status' => 0 ),

array( 'mail\_inbox\_id' => $id, 'LeaderMail.leader\_id' => $this->auth\_leader\_id

) ) ) $this->Session->setFlash( \_\_( MSG\_DATA\_UPDATE\_SUCCESS

), 'Bootstrap/flash-success' ); else $this->Session->setFLash( \_\_(

MSG\_DATA\_SAVE\_FAILED ) );

$this->Notification->addNotif( $id, 'mail\_inboxes',

'not\_approved', 'read', 0 ); //$this->set( 'data',

$this->MailInbox->read( null, $id ) ); return $this->redirect(

array( 'action' => ACTION\_INDEX ) );

public function add() { if( $this->request->is( 'post' ) ) {

$this->MailInbox->create(); $this->request->data[ 'MailInbox' ][

'status' ] = 1; if( $this->MailInbox->save( $this->request->data ) ) {

$this->Session->setFlash( \_\_( MSG\_DATA\_UPDATE\_SUCCESS ), 'Bootstrap

/flash-success' ); return $this->redirect( array( 'action' =>

ACTION\_INDEX ) ); } else { $this->Session->setFLash( \_\_(

MSG\_DATA\_SAVE\_FAILED ) ); } } }

public function leader\_index() { $options[ 'contain' ] = array(

'LeaderMail' ); $options[ 'order' ] = array( 'MailInbox.id'

=> 'DESC' ); $this->Paginator->settings = $options; $datas =

$this->Paginator->paginate( $this->model\_name );

//$datas = $this->MailInbox->find( 'all', $options );

$this->set( compact( 'datas' ) ); }

public function leader\_edit( $id = null ) {

if( !$id ) { throw new NotFoundException( \_\_(

MSG\_DATA\_NOT\_FOUND ) ); }

$this->MailInbox->id = $id; if(

!$this->MailInbox->exists() ) { throw new

NotFoundException( \_\_( MSG\_DATA\_NOT\_FOUND ) ); }

if( !$this->Notification->updateAll( array( 'status' => 1 ),

array( 'Notification.content\_id' => $id, 'Notification.content' =>

'mail\_inboxes', 'Notification.receiver\_id' => '%'.

$this->auth\_role .'%' ) ) ) $this->Session->setFLash( \_\_(

MSG\_DATA\_SAVE\_FAILED ), 'Bootstrap/flash-error' );

if( $this->request->is( 'post' ) || $this->request->is( 'put' ) )

{ /\*foreach( $this->request->data[ 'LeaderMail' ] as $key =>

$data ) { //if ( $data[ 'leader\_id' ] == 0 ) unset(

$this->request->data[ 'LeaderMail' ][ $key ] ); }\*/

//$x = $this->MailInbox->Unit->find( 'all', array(

'conditions' => array( 'Unit.type' => 2, 'LeaderMail.mail\_inbox' =>2 ) ) );

$options[ 'joins' ] = array( array(

'table' => 'leader\_mails', 'alias' => 'LeaderMail',

'type' => 'INNER', 'conditions' => array(

'LeaderMail.leader\_id = Leader.id' )

) ); $options[ 'conditions' ] = array(

'LeaderMail.mail\_inbox\_id' => $id, 'Leader.type' => 2 );

$options[ 'fields' ] = array( 'Leader.id' ); $x =

$this->Leader->find( 'list', $options ); $z = array();

foreach( $x as $id ) { $z[]= $id; }

$this->LeaderMail->deleteAll( array( 'Leader.id' => $z ),

false ); //$this->set( 'x', $x );

$datas = $leaders = array();

$admin = array( array(

'sender\_id' => $this->auth\_leader\_id,

'receiver\_id' => 1, // ID ADMIN 'content\_id' =>

$id, 'content' => 'mail\_inboxes',

'action' => 'approved', 'redirect' =>

'read', 'mail\_status' => 1

) );

if( $this->MailInbox->saveAssociated( $this->request->data,

array( 'deep' => true ) ) ) { $this->Session->setFlash(

\_\_( MSG\_DATA\_EDIT\_SUCCESS ), 'Bootstrap/flash-success' );

if( isset( $this->request->data[ $this->model\_name ][

'Assistant' ] ) ) { foreach( $this->request->data[

$this->model\_name ][ 'Assistant' ] as $key ):

$leaders[] = array( 'sender\_id' =>

$this->auth\_leader\_id, 'receiver\_id' => $key,

'content\_id' => $this->request->data[ $this->model\_name ][

'id' ], 'content' => 'mail\_inboxes', 'action'

=> 'add', 'redirect' => 'read', 'mail\_status' => 2

); endforeach; } $datas =

array\_merge( $admin, $leaders );

$this->Notification->addNotif( $datas );

return $this->redirect( array( 'action' => ACTION\_INDEX )

); } } else {

$this->request->data = $this->MailInbox->read( null, $id ); }

public function leader\_approved( $id = null ) {

if( !$id ) { throw new NotFoundException( \_\_(

MSG\_DATA\_NOT\_FOUND ) ); }

$this->MailInbox->id = $id; if(

!$this->MailInbox->exists() ) { throw new

NotFoundException( \_\_( MSG\_DATA\_NOT\_FOUND ) ); } if(

$this->MailInbox->updateAll( array( 'MailInbox.leader\_status' => 1 ), array(

'MailInbox.id' => $id ) ) ) $this->Session->setFlash( \_\_(

MSG\_DATA\_UPDATE\_SUCCESS ), 'Bootstrap/flash-success' ); else

$this->Session->setFLash( \_\_( MSG\_DATA\_SAVE\_FAILED ) ); $admin =

array( array( 'sender\_id' => $this->auth\_leader\_id,

'receiver\_id' => 1, // ID ADMIN 'content\_id' => $id,

'content' => 'mail\_inboxes', 'action' =>

'approved', 'redirect' => 'read',

'mail\_status' => 1 ) );

$this->Notification->addNotif( $admin ); //$this->set( 'data',

$this->MailInbox->read( null, $id ) ); return $this->redirect(

array( 'action' => ACTION\_INDEX ) ); }

public function leader\_not\_approved( $id = null ) { if( !$id )

{ throw new NotFoundException( \_\_( MSG\_DATA\_NOT\_FOUND ) ); }

$this->MailInbox->id = $id; if(

!$this->MailInbox->exists() ) { throw new

NotFoundException( \_\_( MSG\_DATA\_NOT\_FOUND ) ); } if(

$this->MailInbox->updateAll( array( 'MailInbox.leader\_status' => 0 ), array(

'MailInbox.id' => $id ) ) ) $this->Session->setFlash( \_\_(

MSG\_DATA\_UPDATE\_SUCCESS ), 'Bootstrap/flash-success' ); else

$this->Session->setFLash( \_\_( MSG\_DATA\_SAVE\_FAILED ) );

$admin = array( array( 'sender\_id' =>

$this->auth\_leader\_id, 'receiver\_id' => 1, // ID

ADMIN 'content\_id' => $id,

'content' => 'mail\_inboxes', 'action' =>

'not\_approved', 'redirect' => 'read',

'mail\_status' => 0 ) );

$this->Notification->addNotif( $admin ); //$this->set( 'data',

$this->MailInbox->read( null, $id ) ); return $this->redirect(

array( 'action' => ACTION\_INDEX ) );

public function leader\_inbox\_notif() { $this->autoRender = false;

$count = $this->MailInbox->find( 'count', array( 'conditions' =>

array( 'MailInbox.leader\_id' => $this->auth\_leader\_id,

'MailInbox.leader\_status' => 0 ) ) ); if ( $count > 0 )

return $count; }

public function assistant\_approved( $id = null ) {

$this->approved( $id ); }

public function assistant\_not\_approved( $id = null ) {

$this->not\_approved( $id ); }

public function assistant\_index() { $this->index(); }

public function assistant\_read( $id = null ) {

if( !$id ) { throw new NotFoundException( \_\_(

MSG\_DATA\_NOT\_FOUND ) ); }

$this->MailInbox->id = $id; if(

!$this->MailInbox->exists() ) { throw new

NotFoundException( \_\_( MSG\_DATA\_NOT\_FOUND ) ); }

if( !$this->Notification->updateAll( array( 'status' => 1 ),

array( 'Notification.content\_id' => $id, 'Notification.content' =>

'mail\_inboxes', 'Notification.receiver\_id' => '%'.

$this->auth\_role .'%' ) ) ) $this->Session->setFLash( \_\_(

MSG\_DATA\_SAVE\_FAILED ), 'Bootstrap/flash-error' );

if( $this->request->is( 'post' ) || $this->request->is( 'put' ) )

{ /\* foreach( $this->request->data[

'LeaderMail' ] as $key => $data ) { // if (

$data[ 'leader\_id' ] == 0 ) unset( $this->request->data[ 'LeaderMail' ][ $key

] ); } $options[ 'joins' ] = array(

array( 'table' => 'leader\_mails',

'alias' => 'LeaderMail', 'type' => 'INNER',

'conditions' => array( 'LeaderMail.leader\_id =

Leader.id' ) ) );

$options[ 'conditions' ] = array( 'LeaderMail.mail\_inbox\_id' => $id,

'LeaderMail.status' => 0, 'Leader.type' => 3 ); $options[

'fields' ] = array( 'Leader.id' ); $x = $this->Leader->find(

'list', $options ); \*/ $options[ 'joins' ] = array(

array( 'table' => 'leader\_mails',

'alias' => 'LeaderMail', 'type' => 'INNER',

'conditions' => array( 'LeaderMail.leader\_id =

Leader.id' ) ) );

$options[ 'conditions' ] = array( 'LeaderMail.mail\_inbox\_id' => $id,

'Leader.type' => 3 ); $options[ 'fields' ] = array(

'Leader.id' ); $x = $this->Leader->find( 'list', $options );

$z = array();

foreach( $x as $id ) { $z[]= $id; }

$this->LeaderMail->deleteAll( array( 'Leader.id' => $z ),

false ); if( $this->MailInbox->saveAssociated(

$this->request->data, array( 'deep' => true ) ) ) {

$this->Session->setFlash( \_\_( MSG\_DATA\_EDIT\_SUCCESS ), 'Bootstrap/flash-

success' ); if( isset( $this->request->data[

$this->model\_name ][ 'Unit' ] ) ) {

foreach( $this->request->data[ $this->model\_name ][ 'Unit' ] as $key ):

$this->Notification->addNotif( $this->request->data[ $this->model\_name ][ 'id'

], 'mail\_inboxes', 'add', 'read', 2, 'unit' );

endforeach; }

return $this->redirect( array( 'action' => ACTION\_INDEX ) ); }

} else { $this->request->data =

$this->MailInbox->read( null, $id ); $options[ 'joins' ] =

array( array( 'table' =>

'leader\_mails', 'alias' => 'LeaderMail',

'type' => 'INNER', 'conditions' => array(

'LeaderMail.leader\_id = Leader.id' )

) ); $options[ 'conditions' ] = array(

'LeaderMail.mail\_inbox\_id' => 2, 'LeaderMail.status' => 0, 'Leader.type' => 3

); $options[ 'fields' ] = array( 'Leader.id' );

$x = $this->Leader->find( 'list', $options ); $z = array();

foreach( $x as $id ) { $z[][ 'id' ] = $id;

} $this->set( 'x', $z ); } }

public function unit\_index() { $this->index();

//$this->render( '/MailInboxes/assistant\_index' );

}

public function unit\_read($id = null) {

if( !$id ) { throw new NotFoundException( \_\_(

MSG\_DATA\_NOT\_FOUND ) ); }

$this->MailInbox->id = $id; if(

!$this->MailInbox->exists() ) { throw new

NotFoundException( \_\_( MSG\_DATA\_NOT\_FOUND ) ); }

if( !$this->Notification->updateAll( array( 'status' => 1 ),

array( 'Notification.content\_id' => $id, 'Notification.content' =>

'mail\_inboxes', 'Notification.receiver\_id' => '%'.

$this->auth\_role .'%' ) ) ) $this->Session->setFLash( \_\_(

MSG\_DATA\_SAVE\_FAILED ), 'Bootstrap/flash-error' );

$this->request->data = $this->MailInbox->read( null, $id );

$options[ 'joins' ] = array( array(

'table' => 'leader\_mails', 'alias' => 'LeaderMail',

'type' => 'INNER', 'conditions' => array(

'LeaderMail.leader\_id = Leader.id' ) )

); $options[ 'conditions' ] = array( 'LeaderMail.mail\_inbox\_id' =>

2, 'LeaderMail.status' => 0, 'Leader.type' => 3 ); $options[

'fields' ] = array( 'Leader.id' ); $x = $this->Leader->find(

'list', $options ); $z = array(); foreach( $x as $id )

{ $z[][ 'id' ] = $id; } $this->set(

'x', $z );

public function unit\_approved( $id = null ) { $this->approved( $id

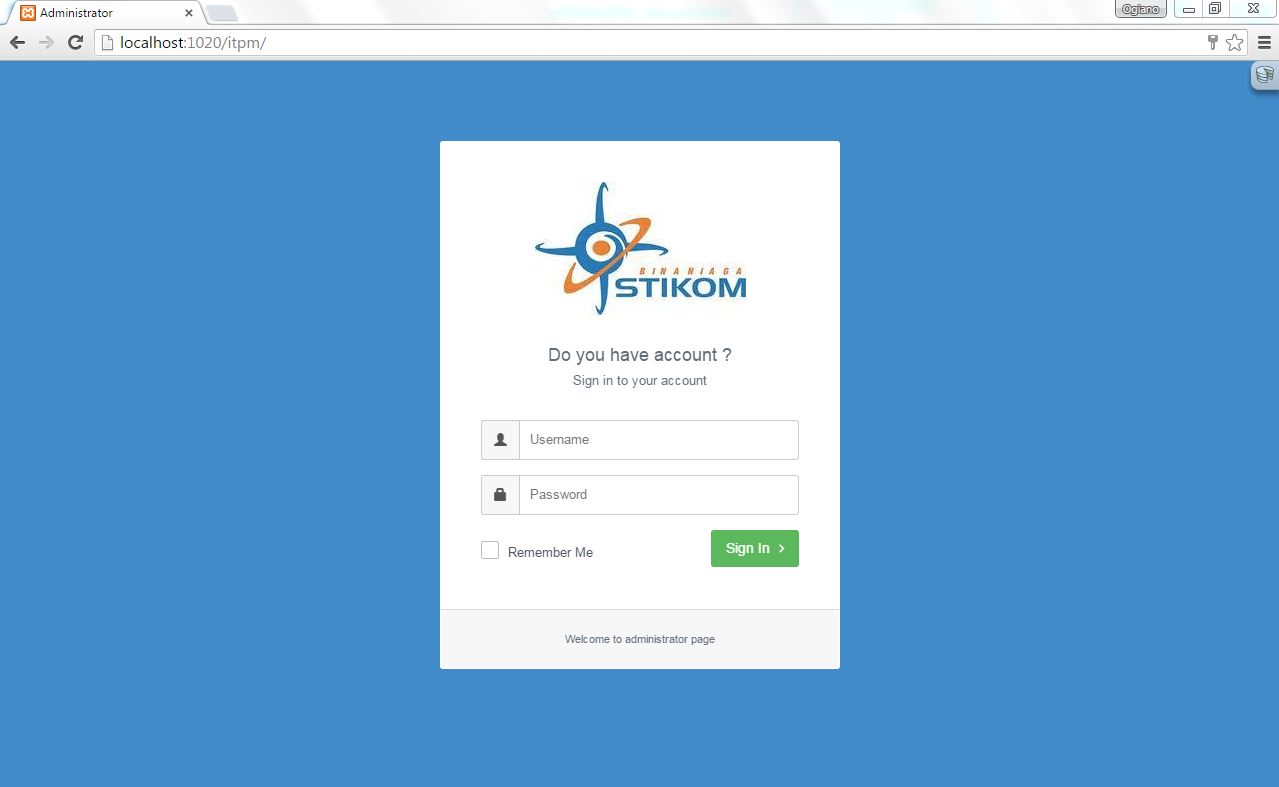
); }

public function unit\_not\_approved( $id = null ) {

$this->not\_approved( $id ); } }

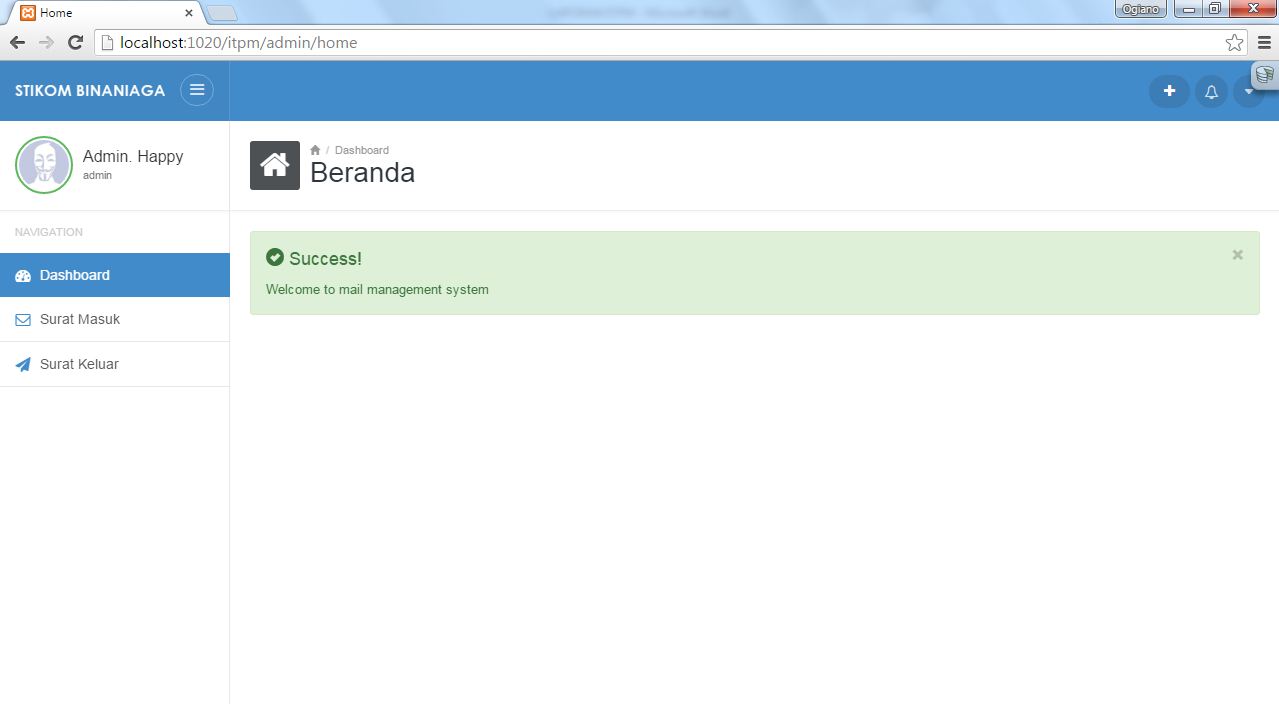
## **Dukungan User**

**User Guide**

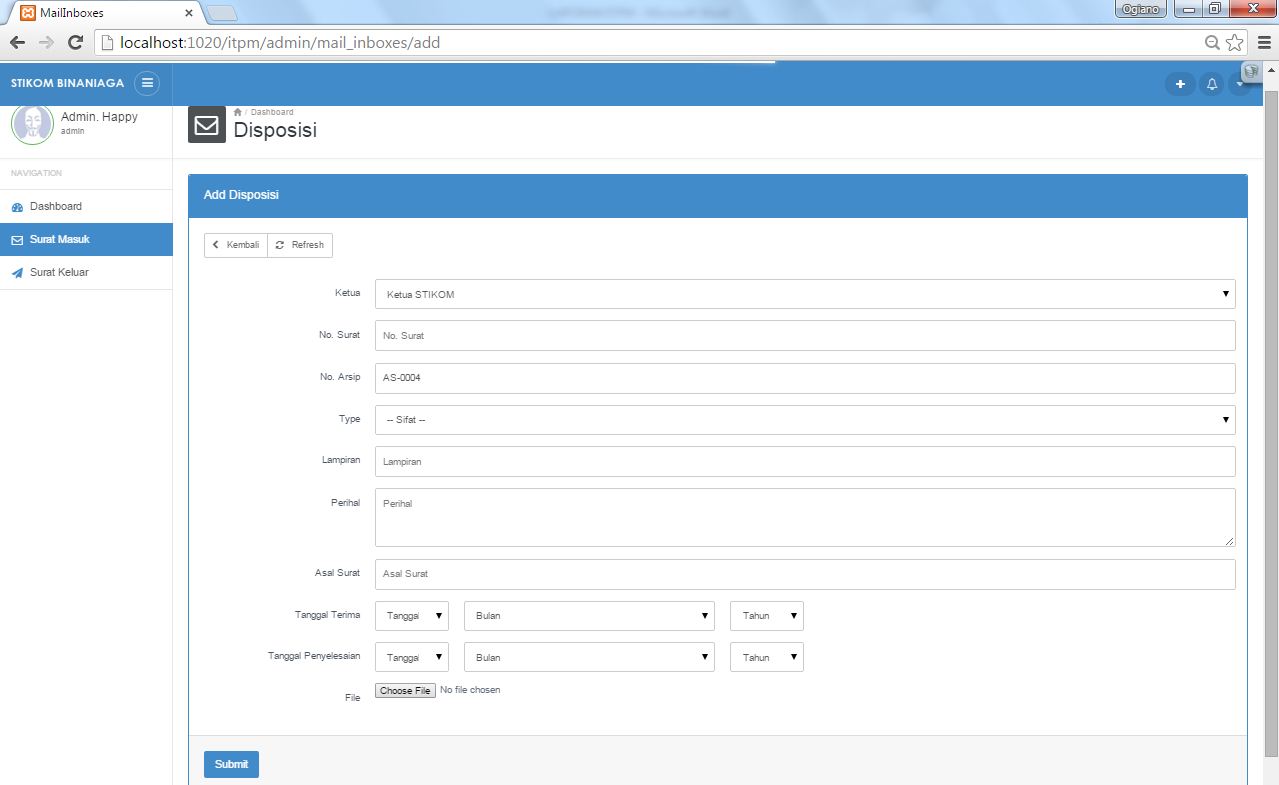
*Gambar 46. Interface Login*

Interface ini merupakan interface awal pada saat sistem ini dijalankan, adapun langkah – langkah dalam penggunaan form ini yaitu :

* + - 1. Masukkan URL pada web browser.
      2. Masukkan Username dan Password, kemudian klik tombol Sign In atau tekan enter.
      3. Jika username atau password yang dimasukkan salah, maka sistem akan merespon dengan menampilkan peringatan bahwa username atau password yang dimasukkan salah. Jika username dan password yang di inputkan benar maka akan diarahkan ke menu utama.

*Gambar 47. Halaman Utama Administrasi*

Interface ini merupakan halaman utama untuk bagian administrasi yang disajikan setelah melakukan login, jika sesuai maka akan masuk ke dalam menu utama yang menampilkan menu yang ada dalam aplikasi untuk kemudian menjalankan fungsi dari aplikasi tersebut.



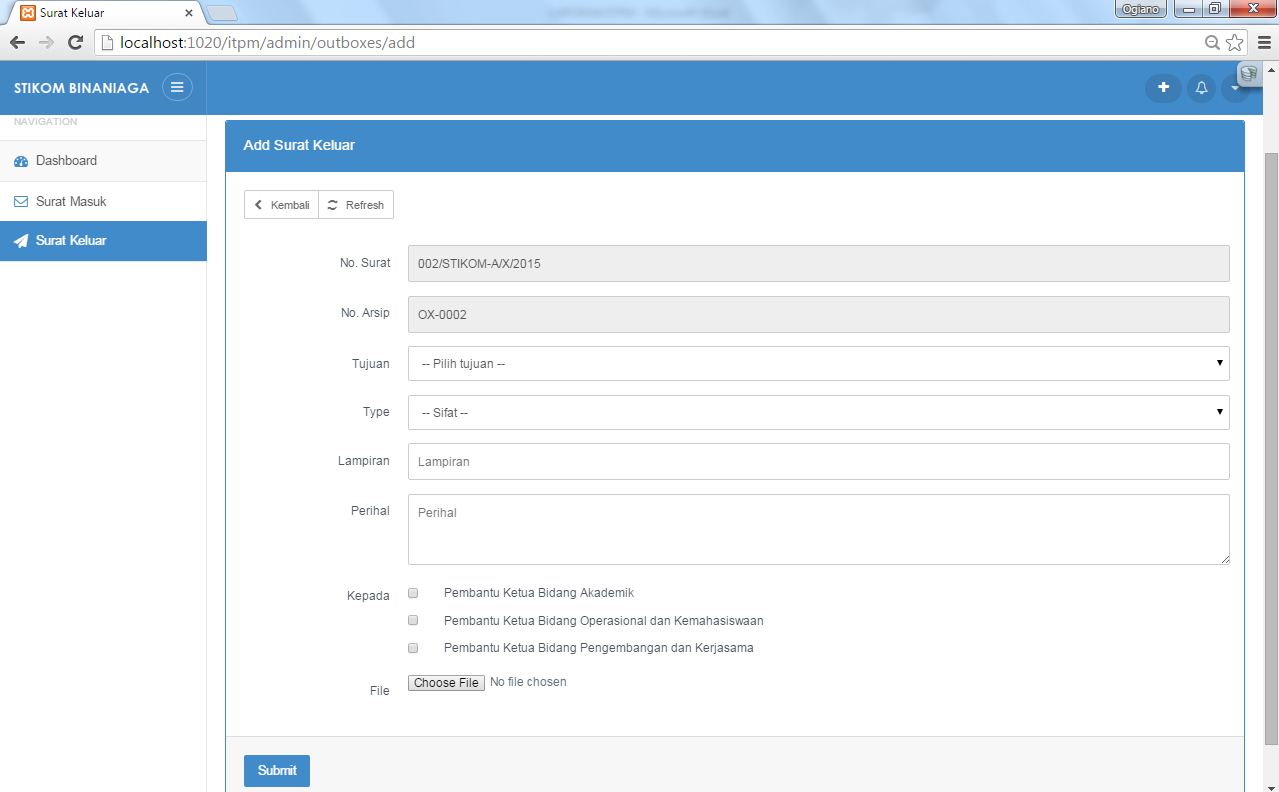
*Gambar 48. Form Input Data Surat Masuk*

Untuk masuk ke dalam form input data surat masuk, user memilih menu surat masuk kemudian menekan tombol tambah item baru.

Kemudian user menginputkan data yang dibutuhkan.

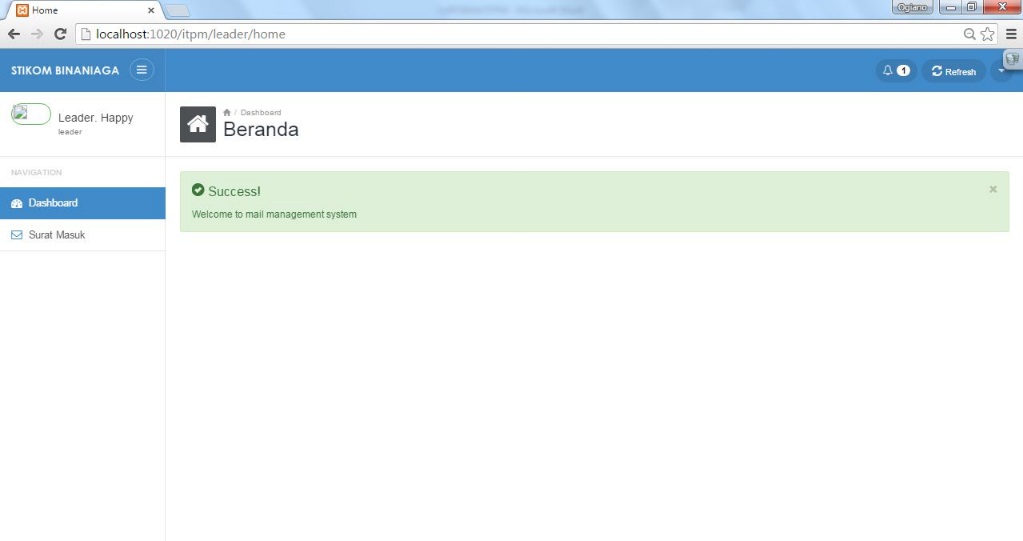
Setelah mengisi form tersebut user memilih tombol submit.

Form pengajuan ini dapat diakses oleh user dengan hak akses administrasi.



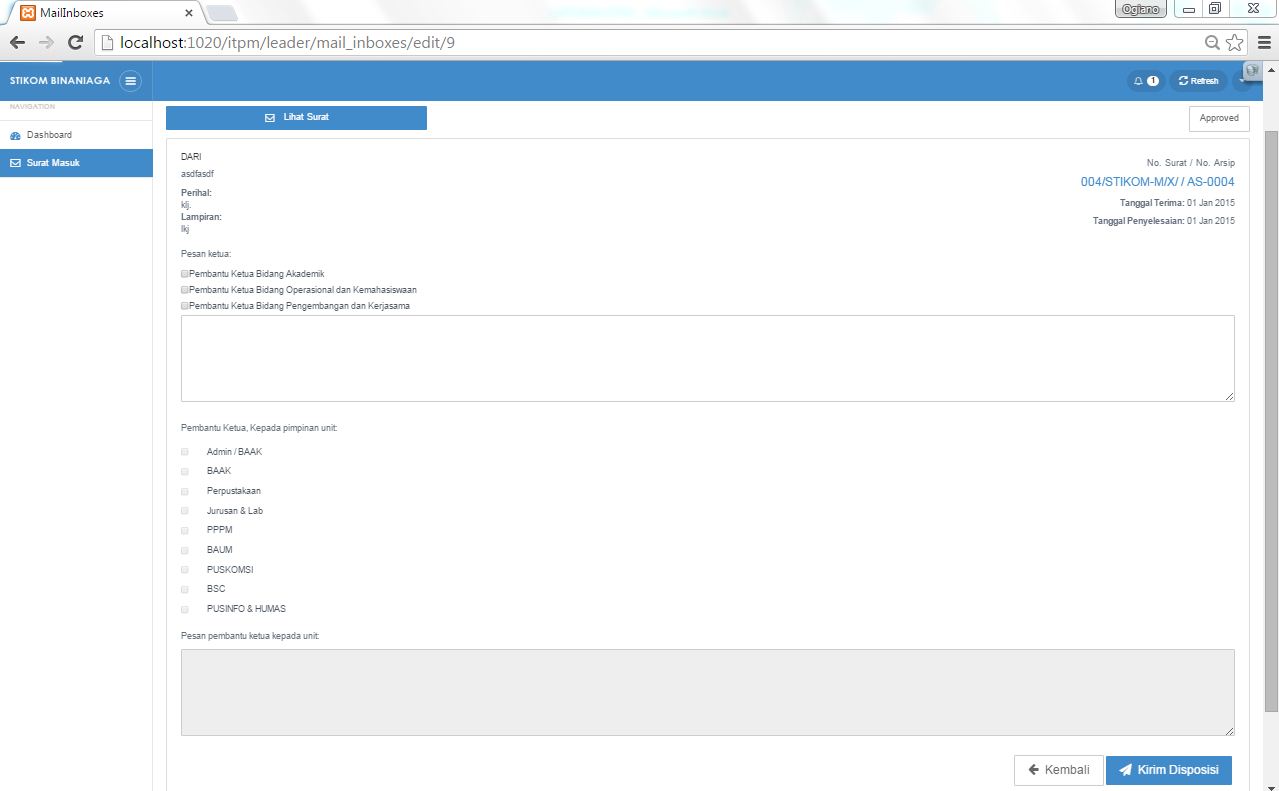
*Gambar 49. Form Input Data Surat Keluar*

1. Untuk masuk ke dalam form input data surat keluar, user memilih menu surat keluar kemudian menekan tombol tambah item baru.
2. Kemudian user menginputkan data yang dibutuhkan.
3. Setelah mengisi form tersebut user memilih tombol submit.
4. Form pengajuan ini dapat diakses oleh user dengan hak akses administrasi



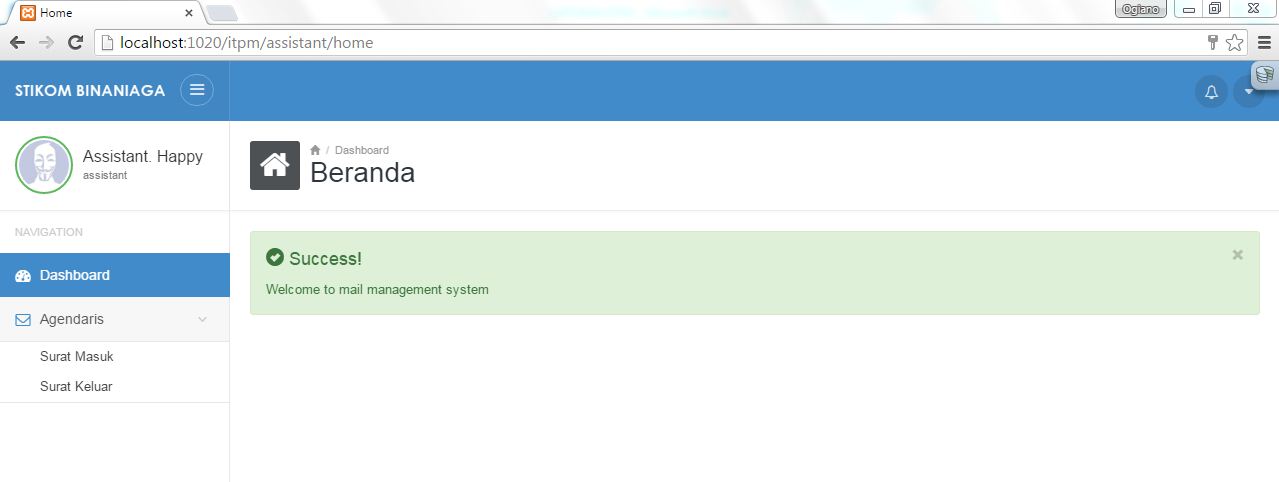
*Gambar 50. Menu Utama Ketua*

Interface ini merupakan halaman utama untuk Ketua yang disajikan setelah melakukan login, jika sesuai maka akan masuk ke dalam menu utama yang menampilkan menu yang ada dalam aplikasi untuk kemudian menjalankan fungsi dari aplikasi tersebut.

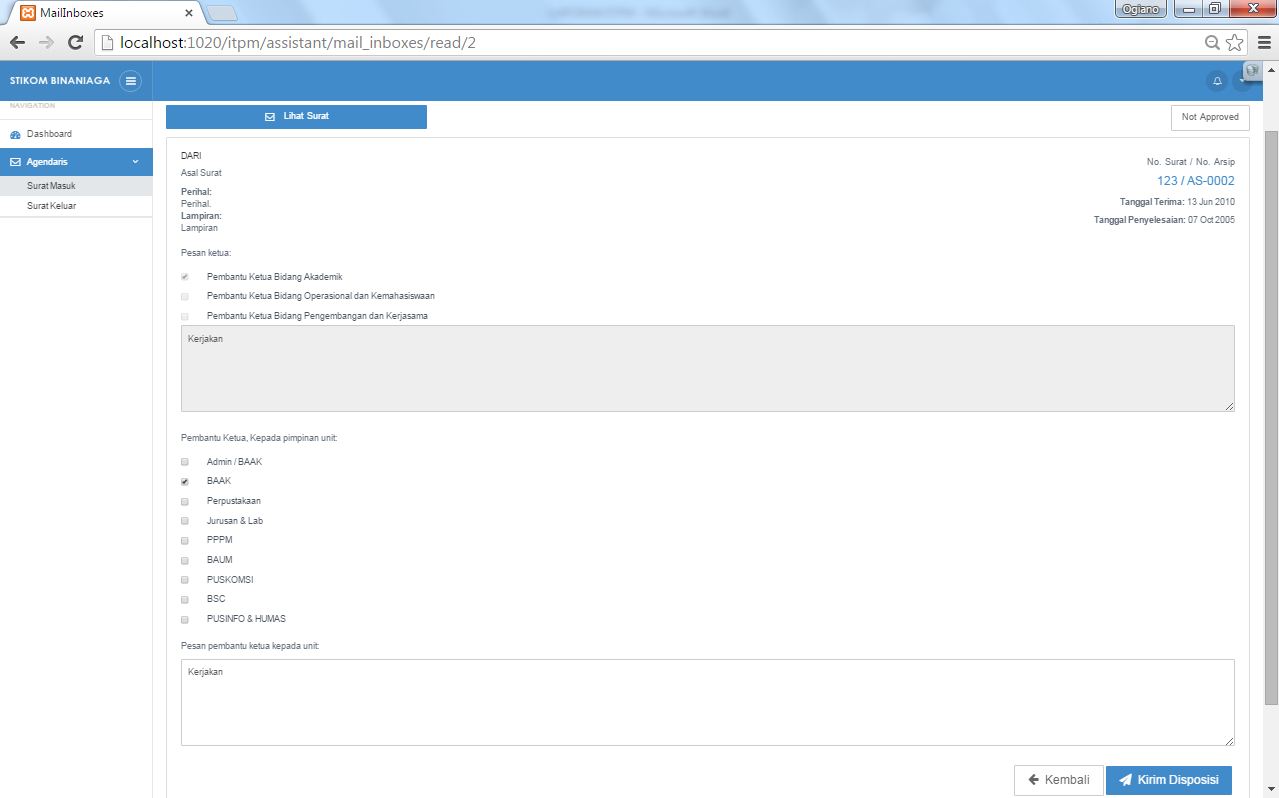


*Gambar 51. Form Ketua Mengisi Disposisi*

1. Untuk masuk ke form ketua mengisi disposisi, user memilih surat masuk kemudian membuka pesan baru yang belum di disposisi. Selain itu bisa juga dengan membuka notifikasi pesan baru yang masuk pada icon notifikasi di bagian pojok kanan atas.
2. Kemudian user menginputkan data yang di perlukan.
3. Setelah mengisi form tersebut tekan tombol kirim disposisi untuk mengirim disposisi.
4. Form Ketua mengisi disposisi ini dapat diakses oleh user yang memiliki hak akses Ketua.

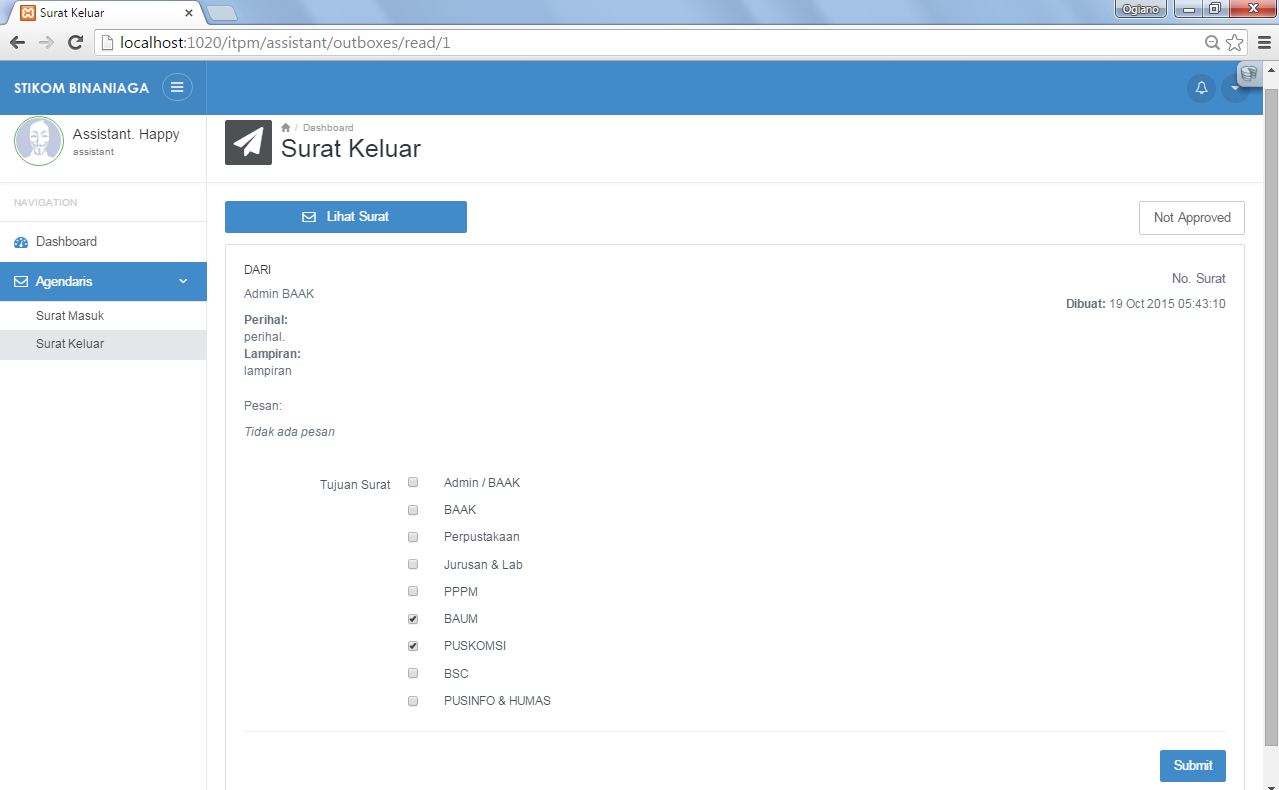
*Gambar 52. Menu Utama Assistant*

Interface ini merupakan halaman utama untuk Pembantu Ketua yang disajikan setelah melakukan login, jika sesuai maka akan masuk ke dalam menu utama yang menampilkan menu yang ada dalam aplikasi untuk kemudian menjalankan fungsi dari aplikasi tersebut.



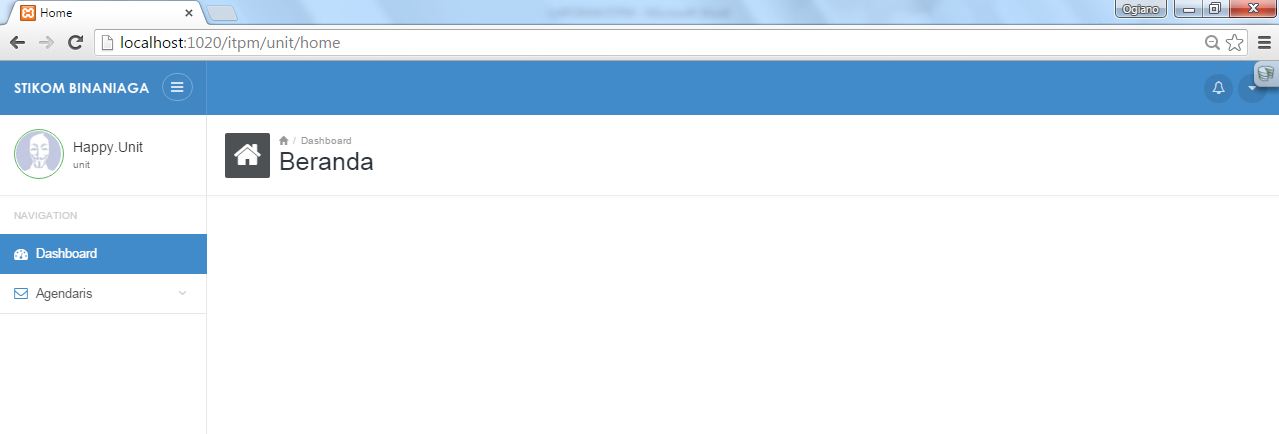
*Gambar 53. Form Assistant Mengisi Disposisi Surat Masuk*

1. Untuk dapat masuk ke form assistant mengisi disposisi, user memilih menu agendaris kemudian pilih sub menu surat masuk kemudian membuka surat masuk yang baru. Selain itu bisa juga dengan cara menekan tombol notifikasi pada bagian kanan atas sistem lalu membuka pesan baru yang masuk.
2. Kemudian user menginputkan data yang di perlukan.
3. Setelah form terisi semua tekan tombol kirim disposisi untuk mengirim disposisi.
4. User yang dapat mengakses form ini yaitu user yang memiliki hak akses assistant atau pembantu ketua.



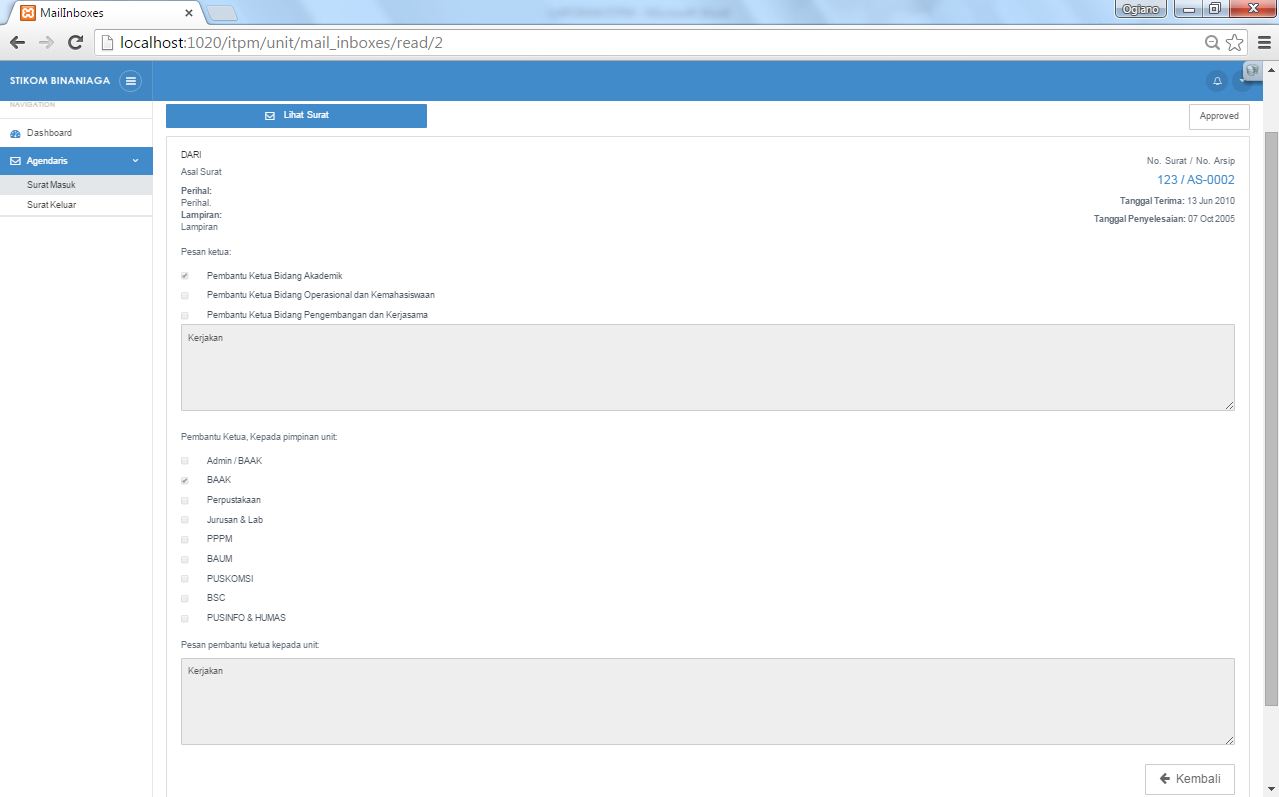
*Gambar 54. Pembantu Ketua melakukan verifikasi surat keluar*

1. Untuk dapat masuk ke verifikasi surat keluar, user memilih menu agendari kemudian pilih sub menu surat keluar kemudian buka surat keluar yang memiliki tujuan dalam untuk dalam instansi. Selain itu bisa juga membuka surat melalui icon notifikasi apabila surat keluar tersebut baru.
2. Kemudian user menginputkan data yang diperlukan.
3. Setelah itu klik submit untuk mengirim surat dan melakukan verifikasi.
4. User yang dapat mengakses form ini yaitu user yang memiliki hak akses assistant atau pembantu ketua.



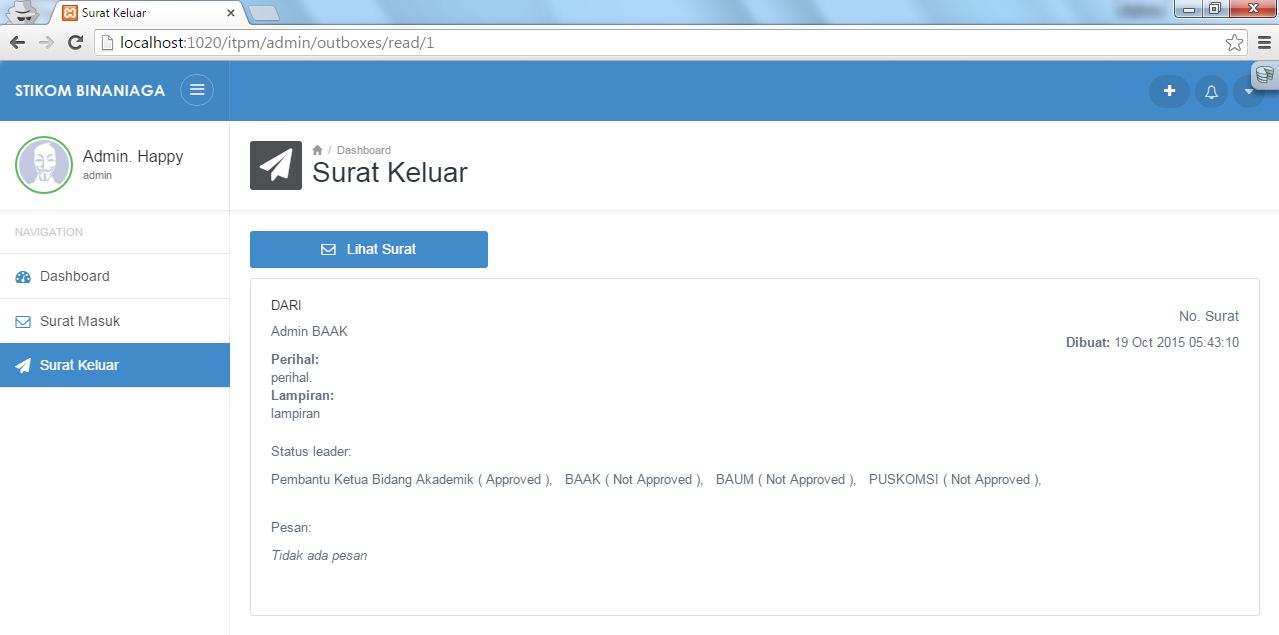
*Gambar 55. Menu Utama Kepala Unit*

Interface ini merupakan halaman utama untuk Kepala Unit yang disajikan setelah melakukan login, jika sesuai maka akan masuk ke dalam menu utama yang menampilkan menu yang ada dalam aplikasi untuk kemudian menjalankan fungsi dari aplikasi tersebut.



*Gambar 56. Unit Melihat Disposisi*

1. Untuk dapat melihat lembar disposisi, user memilih menu agendaris kemudian pilih submenu surat masuk kemudian buka surat masuk. Atau bisa juga dengan membuka notifikasi apabila ada pesan baru yang masuk.
2. User memberi notifikasi kembali ke admnistrasi dengan menekan tombol approve pada bagian kanan atas disposisi.
3. User yang dapat mengakses ini yaitu user yang memiliki hak akses kepala unit.



*Gambar 57. Unit Melakukan Verifikasi Surat Keluar*

1. Untuk dapat melakukan verifikasi surat keluar, user memilih menu agendaris kemudian pilih submenu surat keluar kemudian buka surat keluar. Atau bisa juga dengan menekan tombol notifikasi apabila ada pesan surat keluar baru yang masuk.
2. User melakukan verifikasi dengan menekan tombol approve pada bagian kanan atas sistem.
3. User yang dapat mengakses ini yaitu user yang memiliki hak akses unit.

**[HALAMAN INI SENGAJA DIKOSONGKAN]**

# **PENGUJIAN SISTEM**

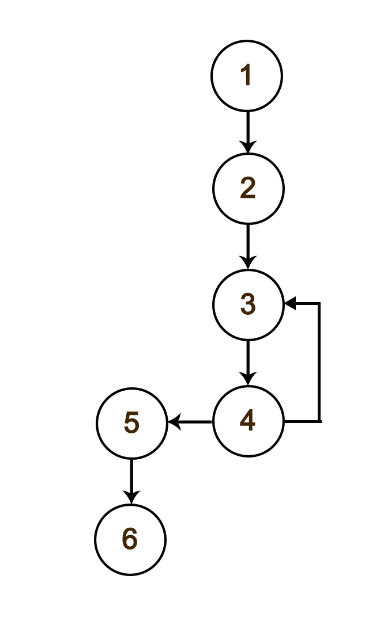
## **Pengujian Blackbox**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Skenario Pengujian | Test Case | Hasil yang diharapkan | Hasil pengujian | Kesimpulan |
| 1 | Mengosongkan semua isian data login, lalu langsung menekan tombol “Sign In” | Username : -  Password : - | Sistem akan menolak akses login dan menampilkan pesan, “Please fill out this field” | Sesuai harapan | Valid |
| 2 | Hanya mengisi data nama user dan mengosongkan data password, lalu langsung menekan tombol “Sign In” | Username : admin  Password : - | Sistem akan menolak akses login dan menampilkan pesan “Please fill out this field” | Sesuai harapan | Valid |
| 3 | Hanya mengisi password dan mengosongkan data username lalu menenkan tombol “Sign In” | Username : -  Password : 123 | Sistem akan menolak akses login dan menampilkan pesan “Please fill out this field” | Sesuai harapan | Valid |
| 4 | Menginputkan dengan kondisi salah satu data benar dan salah satu lagi salah, lalu langsung menekan tombol “Sign In” | Username : user (salah)  Password : user (benar) | Sistem akan menolak dan menampilkan pesan “Invalid username or password” | Sesuai harapan | Valid |
| 5 | Menginputkan sintaks SQL injection oada kedua data login lalu menekan tombol “Sign In” | Username : ‘OR ‘1’=’1  Password : ‘OR ‘1’=’1 | Sistem akan menolak dan menampilkan pesan “Invalid username or password” | Sesuai Harapan | Valid |
| 6 | Menginputkan data login yang benar, lalu menekan tombol “Sign In” | Username : admin  Password : user | Sistem akan menerima akses login dan langsung menampilkan halaman utama sistem | Sesuai harapan | Valid |

*Tabel 42. Pengujian Black box pada validasi data login administrasi*

**Keterangan :** - = kosong (tidak di isi)

## **Pengujian Whitebox**



*Gambar 58. Flow graph – White Box Testing Login*

Berdasarkan flow graph gambar diatas maka dapat dihitung kompleksitas cyclomatic nya sebagai berikut :

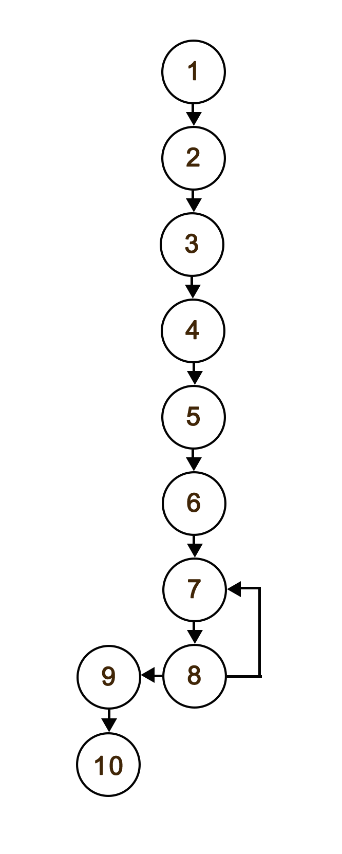
V(G) = 6 edges – 6 nodes + 2 = 2

Hasil kompleksitas cyclomatic menggambarkan banyaknya path dan batas atas sejumlah ujicoba yang harus dirancang dan dieksekusi untuk seluruh perintah dalam program. Maka jalur independent yang harus didapat yaitu diantaranya:

Jalur 1 : 1 – 2 – 3 – 4 – 3

Jalur 2 : 1 - 2 – 3 – 4 – 5 – 6

Hasil pengujian dari pengujian basis path yang dilakukan menunjukkan bahwa secara perhitungan yang dilakukan telah diemukan jalur (rute) dari titik awal ke titik tujuan dengan perkiraan aliran eksekusi sebanyak 2 jalur.



*Gambar 59. Flow Grap – White Box Testing Input Data Surat Masuk*

Berdasarkan flow graph gambar diatas maka dapat dihitung kompleksitas cyclomatic nya sebagai berikut :

V(G) = 10 edges – 10 nodes + 2 =2

Hasil kompleksitas cyclomatic menggambarkan banyaknya path dan batas atas sejumlah ujicoba yang harus dirancang dan dieksekusi untuk seluruh perintah dalam program. Maka jalur independent yang harus didapat yaitu diantaranya:

Jalur 1 : 1 – 2 – 3 – 4 – 5 – 6 – 5 – 6 – 7 – 8 - 7

Jalur 2 : 1 – 2 – 3 – 4 – 5 – 6 – 5 – 6 – 7 – 8 – 9 - 10

Hasil pengujian dari pengujian basis path yang dilakukan menunjukkan bahwa secara perhitungan yang dilakukan telah diemukan jalur (rute) dari titik awal ke titik tujuan dengan perkiraan aliran eksekusi sebanyak 2 jalur.

# **KINERJA SISTEM**

|  |  |
| --- | --- |
| **Sistem Lama** | **Sistem Baru** |
| Proses disposisi surat surat masih menggunakan kertas | Proses disposisi surat sudah menggunakan system sehingga memudahkan dalam proses pengiriman disposisi |
| Proses pencarian data arsip masih tergolong sulit karena menumpuknya berkas | Proses pencarian data arsip lebih mudah karena tersimpan dalam database |
| Proses distribusi surat masih dilakukan dengan cara konvensional sehingga belum efektif dan efisien | Proses distribusi surat telah menggunakan sistem sehingga lebih efektif dan efisien |
| Proses pengolahan data belum terorganisir | Proses pengolahan data lebih terorganisir dengan adanya sistem ini |

# **IMPLIKASI**

Implikasi yang terdapat pada Penerapan CakePHP Framework dan MySQL untuk Agendaris Surat dan Notifikasi di STIKOM Binaniaga Bogor yaitu :

Meningkatkan distribusi surat menyurat menjadi lebih efektif dan efisien dengan menggunakan teknologi komputer

Proses pengolahan data lebih terorganisir dengan sistem penyimpanan sudah menggunakan database

Proses pencarian data arsip lebih cepat dengan menggunakan sistem ini

**[HALAMAN INI SENGAJA DIKOSONGKAN]**

# **RUJUKAN**

1. Memorandum
2. CakePHP Cookbook (http://cakephp.org)
3. PHP Manual (<http://php.net>)

**[HALAMAN INI SENGAJA DIKOSONGKAN]**

# **PROFIL PELAKSANA**

|  |  |
| --- | --- |
|  | Ogiano Waskitajaya, mahasiswa Sekolah Tinggi Ilmu Komputer (STIKOM) Binaniaga Bogor semester 6. Program studi Teknik Informatika jenjang strata 1, berminat menjadi Aplication Developer di salah satu perusahaan startup. |
|  | Yanuar Nurcahyo, mahasiswa Sekolah Tinggi Ilmu Komputer (STIKOM) Binaniaga Bogor semester 6. Program studi Teknik Informatika jenjang strata 1, berminat menjadi CEO perusahaan yang bergerak di bidang IT. |

**[HALAMAN INI SENGAJA DIKOSONGKAN]**

# **LAMPIRAN**

**Dokumentasi**

Foto kegiatan presentasi

