****

**APLIKASI MANAJEMEN UNIT PROPERTI BERBASIS PHP MYSQL**

YANSA NOVAN

41512010111

**PROGRAM STUDI INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS MERCU BUANA  
JAKARTA  
2016**

****

**APLIKASI MANAJEMEN UNIT PROPERTI BERBASIS PHP MYSQL**

Laporan Tugas Akhir

Diajukan Untuk Melengkapi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer

Oleh:

YANSA NOVAN

41512010111

**PROGRAM STUDI INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS MERCU BUANA  
JAKARTA  
2016**

**LEMBAR PERNYATAAN**

Yang bertanda tangan dibawah ini:

NIM : 41512010111

Nama : Yansa Novan

Judul Tugas Akhir : Aplikasi Manajemen Unit Properti Berbasis PHP MySQL.

Menyatakan bahwa Tugas Akhir dengan judul yang tersebut diatas adalah hasil karya saya sendiri dan bukan plagiat kecuali kutipan-kutipan dan teori-teori yang digunakan dalam skripsi ini. Apabila ternyata ditemukan didalam Laporan Tugas Akhir saya terdapat unsur plagiat, maka saya siap untuk mendapatkan sanksi akademik yang terkait dengan hal tersebut.

Jakarta, 23 Desember 2016

Yansa Novan

**LEMBAR PENGESAHAN**

Nama : Yansa Novan

NIM : 41512010111

Jurusan : Informatika

Fakultas : Ilmu Komputer

Judul : Manajemen Unit Properti Berbasis Php MySQL.

Jakarta, 23 Desember 2016

Disetujui dan diterima oleh,

|  |  |
| --- | --- |
| Leonard Goeirmanto, ST, M.Sc  Dosen Pembimbing | |
| Dr. Yaya Sudarya Triana, M.Kom, Ph.D  Kaprodi Informatika | Desi Ramayanti, S.Kom, MT  Koordinator Tugas Akhir |

KATA PENGANTAR

Puji Syukur penulis panjatkan kepada Tuhan Yang Maha Esa, karena atas karunia yang telah diberikan kepada penulis sehingga dapat menyelesaikan Laporan Tugas Akhir tepat pada waktunya, dimana Laporan Tugas Akhir tersebut merupakan salah satu persyaratan untuk dapat menyelesaikan Program Studi Strata Satu (S1) pada Jurusan Teknik Informatika Universitas Mercu Buana.

Penulis menyadari bahwa Laporan Tugas Akhir ini masih belum dapat dikatakan sempurna. Karena itu, kritik dan saran akan diterima dengan senang hati. Penulis juga menyadari bahwa Laporan Tugas Akhir ini tidak akan selesai tepat pada waktunya tanpa bantuan, bimbingan, dan motivasi dari berbagai pihak. Maka dari itu, dengan segala kerendahan hati, Penulis ingin menyampaikan ucapan terima kasih kepada :

1. Leonard Goeirmanto, ST, M.Sc., selaku Pembimbing Tugas Akhir dan Pembimbing Akademik yang telah membimbing penulis dengan semua nasihat, semangat dan ilmunya dalam menyusun laporan tugas akhir ini.
2. Dr.Yaya Sudarya Triana, M.Kom, Ph.D., selaku Kaprodi Teknik Informatika Universitas Mercu Buana.
3. Desi Ramayanti, S.Kom, MT., selaku Koordinator Tugas Akhir Teknik Informatika Universitas Mercu Buana.
4. Kedua orang tua yang selama ini telah membesarkan penulis.
5. Umny Salamah, S.Kom, MMSI. selaku dosen pembimbing akademik dan kepada teman-teman yang telah memotivasi dan ikut serta memberikan bantuannya.

Semoga Tuhan Yang Maha Esa membalas kebaikan yang telah diberikan kepada penulis dan penulis berharap semoga laporan tugas akhir ini bermanfaat bagi kita semua. Amin.

Jakarta 30 Desember 2016

Yansa Novan

*ABSTRACT*

*Property usually show something to do with the ownership of a person or group of people on an exclusive right. The main form of property is real property (land), personal wealth(personal property), (ownership of physical goods more), and intellectual property. The business development in the field of property is increasing and many examples such as apartments, houses, shop houses, hotels, and more. It is the role of information technology to support such a business. In documenting the data unit property they have, many companies began to use the manual method that is documented in the book of or on a personal note, that there are still companies that do not use information technology to manage the property that they own.*

*The increasing number of property developers big and famous in Indonesia for example like PT.Agung Podomoro, Ciputra, Lippo Group, which the company has provided services such as rental and sale of apartments, houses, shop houses, hotels and more, through a variety of media to to serve its customers who are looking for housing that they want."Application Management Property Unit" was created using Bubble Sort algorithm, and the methods used in data collection is the method of observation and study the literature. From the problems expected with the "Application Management Property Unit based PHP Mysql" is useful to assist and support business processes in data management of property owned by the company.*

***Keyword: Aplication Managament Unit Property, Algorithm Buble Sort, Framework Laravel, Bootstrap.***

ABSTRAK

Properti biasanya menunjukkan kepada sesuatu kaitannya dengan kepemilikan seseorang atau sekelompok orang atas suatu hak eksklusif. Bentuk yang utama dari properti adalah real property (tanah), kekayaan pribadi *(personal property)*, (kepemilikan barang secara fisik lainnya), dan kekayaan intelektual. Perkembangan bisnis dibidang properti semakin meningkat dan banyak contohnya seperti apartemen, perumahan, ruko-ruko, hotel, dan lainnya. Hal ini tidak lepas dari peranan teknologi informasi untuk menunjang suatu bisnis tersebut. Dalam mendokumentasikan data unit properti yang mereka miliki, masih banyak perusahan menggunakan cara manual yaitu mendokumentasikannya dibuku atau pada catatan pribadi, sehingga masih ada perusahaan yang tidak memanfaatkan teknologi informasi untuk mengelola data properti yang mereka miliki.

Semakin banyaknya pengembang properti yang besar dan terkenal di Indonesia contohnya seperti PT.Agung Podomoro, Ciputra, Lippo Group, dimana perusahaan tersebut telah memberikan layanan berupa penyewaan dan penjualan unit apartemen, rumah, ruko-ruko, hotel dan lainnya, dengan melalui berbagai media untuk melayani para customer yang sedang mencari hunian yang mereka inginkan. “Aplikasi Manajemen Unit Properti” ini yang dibuat menggunakan Algoritma *Bubble Sort,* dan metode yang digunakan dalam pengumpulan data adalah metode observasi dan study pustaka. Dari permasalahan tersebut diharapkan dengan adanya “Aplikasi Manajemen Unit Properti berbasis PHP Mysql” berguna untuk membantu dan mendukung proses bisnis dalam pengelolaan data properti yang dimiliki perusahaan tersebut.

**Kata Kunci: Aplikasi Manajemen Unit Properti, Algortima Buble Sort, Framework Laravel, Bootstrap.**

DAFTAR ISI

[KATA PENGANTAR i](#_Toc471225063)

[*ABSTRACT* ii](#_Toc471225064)

[ABSTRAK iii](#_Toc471225065)

[DAFTAR ISI iv](#_Toc471225066)

[DAFTAR TABEL 1](#_Toc471225067)

[DAFTAR GAMBAR 2](#_Toc471225068)

[BAB I PENDAHULUAN 3](#_Toc471225069)

[1.1 Latar Belakang 3](#_Toc471225070)

[1.2 Rumusan Masalah 4](#_Toc471225071)

[1.3 Batasan Masalah 4](#_Toc471225072)

[1.4 Tujuan dan Manfaat 5](#_Toc471225073)

[1.5 Metode Penelitian 6](#_Toc471225074)

[1.5.1. Metode Analisis 6](#_Toc471225075)

[1.5.2 Model Proses Pengembangan Perangkat Lunak 6](#_Toc471225076)

[1.6 Sistematika Penulisan 8](#_Toc471225077)

[BAB II LANDASAN TEORI 10](#_Toc471225078)

[2.1 Pengertian Web 10](#_Toc471225079)

[2.2 HTML (Hypertext Markup Language) 10](#_Toc471225080)

[2.3 PHP (Hypertext Preprocessor) 10](#_Toc471225081)

[2.4 XAMPP 11](#_Toc471225082)

[2.5 Bootstrap 11](#_Toc471225083)

[2.6 Laravel 11](#_Toc471225084)

[2.7 Basis data (*Database)* 12](#_Toc471225085)

[2.8 MYSQL 13](#_Toc471225086)

[2.9 Algoritma Bubble Sort 14](#_Toc471225087)

[2.10 Model *Waterfall* 15](#_Toc471225088)

[2.11 Entity Relationship Diagram (ERD) 17](#_Toc471225089)

[2.12 UML (*Unified Modeling Language*) 19](#_Toc471225090)

[2.13 Use Case Diagram 20](#_Toc471225091)

[2.14 Activity Diagram 23](#_Toc471225092)

[2.15 Sequence Diagram 25](#_Toc471225093)

[2.16 Class Diagram 27](#_Toc471225094)

[2.17 Pengujian Black Box 29](#_Toc471225095)

[BAB III ANALISIS DAN PERANCANGAN SISTEM 30](#_Toc471225096)

[3.1 Analisis 30](#_Toc471225097)

[3.2 Perancangan Sistem 31](#_Toc471225098)

[3.2.1 Perancangan *Use Case Diagram* 31](#_Toc471225099)

[3.3 Activity Diagram 35](#_Toc471225100)

[3.3.1 Activity Diagram *Login* Admin, User, Manager pada Aplikasi Manajemen web unit properti 36](#_Toc471225101)

[3.3.2 Activity Diagram Menambahkan Data Unit Properti 38](#_Toc471225102)

[3.3.3 Activity Diagram Merubah Data Properti 39](#_Toc471225103)

[3.3.4 Activity Diagram Menghapus data properti 40](#_Toc471225104)

[3.3.5 Activity Diagram Membuat Album Photo 41](#_Toc471225105)

[3.3.6 Activity Diagram upload Photo 42](#_Toc471225106)

[3.3.7 Activity Diagram Merubah Album photo 43](#_Toc471225107)

[3.3.8 Activity Diagram Melihat photo 44](#_Toc471225108)

[3.3.9 Activity Diagram Menghapus Album 46](#_Toc471225109)

[3.4 Sequence Diagram 47](#_Toc471225110)

[3.4.1 Sequence Diagram *Login* Admin dan User, dan manager 47](#_Toc471225111)

[3.4.2 Sequence Diagram Membuat Data properti 48](#_Toc471225112)

[3.4.3 Sequence Diagram Merubah Data Properti 49](#_Toc471225113)

[3.4.4 Sequence Diagram Menghapus Data Properti 50](#_Toc471225114)

[3.4.5 Sequence Diagram Upload Photo 52](#_Toc471225115)

[3.4.6 Sequence Diagram Merubah Album Photo 54](#_Toc471225116)

[3.4.7 Sequence Diagram Menghapus Album Photo 55](#_Toc471225117)

[3.5 Class Diagram Manajemen Unit Properti 56](#_Toc471225118)

[3.6 ERD (*Entity Relation Diagram*) 57](#_Toc471225119)

[3.7 Perancangan Database Manajemen Unit Properti 58](#_Toc471225120)

[3.8 Perancangan Desain Sistem 61](#_Toc471225121)

[3.8.1 Perancangan Desain Sistem Login 61](#_Toc471225122)

[3.8.2 Perancangan Desain Sistem Register 62](#_Toc471225123)

[3.8.3 Perancangan Desain Sistem Halaman Utama 63](#_Toc471225124)

[3.8.4 Perancangan Desain Sistem Halaman Tampilan Data 64](#_Toc471225125)

[3.8.5 Perancangan Desain Sistem Halaman Create Apartemen 66](#_Toc471225126)

[3.8.6 Perancangan Desain Sistem Halaman Edit Apartemen 68](#_Toc471225127)

[3.8.7 Perancangan Desain Sistem Halaman Album 70](#_Toc471225128)

[3.8.8 Perancangan Desain Sistem Halaman Upload Photo 71](#_Toc471225129)

[3.9 PerancanganInterface 72](#_Toc471225130)

[3.9.1 Perancangan *interface login* 72](#_Toc471225131)

[3.9.2 Perancangan *interface register* 73](#_Toc471225132)

[3.9.3 Perancangan Tampilan Utama Web 74](#_Toc471225133)

[3.9.4 Perancangan Merubah Data (*edit*) 76](#_Toc471225134)

[3.9.5 Perancangan Tampilan Membuat Album Photo, Upload Dan List Album Photo. 77](#_Toc471225135)

[3.9.6 Perancangan Tampilan Upload Photo. 77](#_Toc471225136)

[3.9.7 Perancangan Tampilan List Album. 78](#_Toc471225137)

[BAB IV IMPLEMENTASI DAN PENGUJIAN 79](#_Toc471225138)

[4.1 Implementasi Sistem 79](#_Toc471225139)

[4.2 Spesifikasi kebutuhan perangkat lunak dan perangkat keras 79](#_Toc471225140)

[4.3 Implementasi Tampilan 80](#_Toc471225141)

[4.3.1 Implemen Tampilan Pada Halaman Login 80](#_Toc471225142)

[4.3.2 Implemen Tampilan Pada Halaman Register 80](#_Toc471225143)

[4.3.3 Implemen Tampilan Pada Halaman Utama 81](#_Toc471225144)

[4.3.4 Implemen Tampilan Pada *Create* Apartemen 82](#_Toc471225145)

[4.3.5 Implemen Tampilan Pada Edit Apartemen 83](#_Toc471225146)

[4.2.6 Implemen Tampilan Pada Create Album 83](#_Toc471225147)

[4.3.6 Implemen Tampilan Pada *List Album* 84](#_Toc471225148)

[4.4 Implementasi Basis Data 85](#_Toc471225149)

[4.4.1 Tabel  *Database Albums* 85](#_Toc471225150)

[4.4.2 Tabel *Database* Apartemen 86](#_Toc471225151)

[4.4.3 Table *Database* Photos 86](#_Toc471225152)

[4.4.4 Tabel *Database* User 87](#_Toc471225153)

[4.4 Pengujian Fungsionalitas 87](#_Toc471225154)

[4.4.1 Lingkungan Pengujian 87](#_Toc471225155)

[4.4.2 Skenario Pengujian 88](#_Toc471225156)

[4.5 Analisis Hasil Pengujian 91](#_Toc471225157)

[BAB V KESIMPULAN DAN SARAN 92](#_Toc471225158)

[5.1 Kesimpulan 92](#_Toc471225159)

[5.2 Saran 92](#_Toc471225160)

[DAFTAR PUSTAKA 93](#_Toc471225161)

DAFTAR TABEL

[Tabel 2. 1 Berikut tabel ERD 18](#_Toc470941623)

[Tabel 2. 2 adalah simbol-simbol yang ada pada diagram use case 20](#_Toc470941624)

[Tabel 2. 3 adalah tabel skenario use case. 22](#_Toc470941625)

[Tabel 2. 4 simbol-simbol yang ada pada diagram use case 23](#_Toc470941626)

[Tabel 2. 5 adalah simbol-simbol yang ada pada diagram sekuen : 25](#_Toc470941627)

[Tabel 2. 6 adalah tabel Class Diagram 27](#_Toc470941628)

[Tabel 3. 1 Identifikasi Aktor 32](#_Toc470941722)

[Tabel 3. 2 Membuat data properti 32](#_Toc470941723)

[Tabel 3. 3 Membuat data properti 33](#_Toc470941724)

[Tabel 3. 4 Menghapus Data 33](#_Toc470941725)

[Tabel 3. 5 Membuat Album Photo. 34](#_Toc470941726)

[Tabel 3. 6 Upload Photo 34](#_Toc470941727)

[Tabel 3. 7 Melihat Photo Properti 35](#_Toc470941728)

[Tabel 3. 8 Struktur Database Admin, Users, Manager 58](#_Toc470941729)

[Tabel 3. 9 Struktur Database Apartement 58](#_Toc470941730)

[Tabel 3. 10 Struktur Database Album Photo 59](#_Toc470941731)

[Tabel 3. 11 Struktur Database Upload Album Photo 59](#_Toc470941732)

[Tabel 4. 1 Skenario Pengujian 88](#_Toc470941736)

[Tabel 4. 2 89](#_Toc470941737)

DAFTAR GAMBAR

[Gambar 3. 1 Use case Aplikasi manajemen unit properti 31](#_Toc470941751)

[Gambar 3. 2 Activity Diagram Login Admin, users, dan manager. 37](#_Toc470941752)

[Gambar 3. 3 Activity Diagram Menambahkan Data Properti 38](#_Toc470941753)

[Gambar 3. 4 Activity diagram merubah data properti 39](#_Toc470941754)

[Gambar 3. 5 Activity Diagram Menambahkan Menu Makanan Baru 40](#_Toc470941755)

[Gambar 3. 6 Activity Diagram membuat album photo. 41](#_Toc470941756)

[Gambar 3. 7 activity Diagram Upload Photo 42](#_Toc470941757)

[Gambar 3. 8 activity Diagram Upload Photo 43](#_Toc470941758)

[Gambar 3. 9 Activity Diagram melihat photo unit properti 45](#_Toc470941759)

[Gambar 3. 10 Activity Diagram menghapus album unit properti 46](#_Toc470941760)

[Gambar 3. 11 Sequence Diagram Login Admin, User, dan Manager 47](#_Toc470941761)

[Gambar 3. 12 Sequence Diagram membuat data properti 48](#_Toc470941762)

[Gambar 3. 13 Sequence Diagram Menambahkan Daftar Pelanggan 49](#_Toc470941763)

[Gambar 3. 14 Sequence Diagram Menghapus Data properti 50](#_Toc470941764)

[Gambar 3. 15 Sequence Diagram Membuat Album Photo 51](#_Toc470941765)

[Gambar 3. 16 Sequence Diagram Upload Photo 52](#_Toc470941766)

[Gambar 3. 17 Sequence Diagram Menampilkan Photo 53](#_Toc470941767)

[Gambar 3. 18 Sequence Diagram Merubah Album Photo 54](#_Toc470941768)

[Gambar 3. 19 Sequence Diagram Menghapus Album Photo 55](#_Toc470941769)

[Gambar 3. 20 Class Diagram 56](#_Toc470941770)

[Gambar 3. 21 Entity Relation Diagram 57](#_Toc470941771)

[Gambar 4. 1 Tampilan login 80](#_Toc470941772)

[Gambar 4. 2 Tampilan register 81](#_Toc470941773)

[Gambar 4. 3 Tampilan Halaman utama 81](#_Toc470941774)

[Gambar 4. 4Tampilan Create apartemen 82](#_Toc470941775)

[Gambar 4. 5 Tampilan form edit apartemen 83](#_Toc470941776)

[Gambar 4. 6 Tampilan create album photo 83](#_Toc470941777)

[Gambar 4. 7 Tampilan list album photo 84](#_Toc470941778)

[Gambar 4. 8 Tampilan Database Unit Propery 85](#_Toc470941779)

[Gambar 4. 9 Tampilan Database Albums 85](#_Toc470941780)

[Gambar 4. 10 Tampilan Database Apartemen 86](#_Toc470941781)

[Gambar 4. 11 Tampilan Database Photos 86](#_Toc470941782)

[Gambar 4. 12 Tampilan Database Apartemen 87](#_Toc470941783)

BAB I PENDAHULUAN

1.1 Latar Belakang

Persaingan dan perkembangan bisnis properti di Indonesia mengalami kenaikan yang sangat meningkat dari tahun ke tahun, banyak hal yang dapat kita lihat contohnya seperti di kota-kota besar seperti di Jakarta, dimana pembangunan properti seperti rumah, apartemen, hotel, mall, ruko dan lain-lain semakin banyak dan bertambah. Karena semakin banyaknya pertumbuhan penduduk dari tahun ke tahun, maka semakin banyak juga kebutuhan masyarakat untuk mencari tempat hunian yang bagus untuk mereka tempati.

Semakin banyak pengusaha atau *developer* yang mendirikan bisnis dibidang properti, maka dibutuhkan banyak orang atau pekerja untuk membantu mereka dalam memajukan dan mengelola bisnis supaya berjalan dengan baik, hal tersebut tidak lepas juga dari peranan teknologi informasi seperti aplikasi website properti untuk memasarkan produk-produk yang telah mereka gunakan. Dengan adanya peranan teknologi informasi tersebut masyarakat bisa mengaksesnya melalui *gadged* seperti *smartphone*, *laptop*, mereka masing-masing untuk mencari informasi penting mengenai properti, contohnya masyarakat ingin tahu nama apartemen, lokasi atau alamat, luas, harga, jumlah kamar dan masih banyak lagi.

Banyak *develepoer* *properti* yang terkenal di Indonesia contohnya PT.Agung Podomoro, Ciputra, Lippo Group. Perusahaan ini memberikan layanan berupa penyewaan dan penjualan seperti apartemen, dan rumah, dengan melalui website untuk melayani para customer yang sedang mencari hunian yang mereka inginkan.

Untuk membantu memajukan dan mengembangkan bisnis dibidang properti, Data collector berperan penting untuk mengumpulkan serta mendokumentasikan informasi tentang apartemen, perumahan, ruko-ruko dengan cara survey atau terjun langsung dilapangan. Data-data yang dikumpulkan oleh data collector yaitu seperti nama apartemen, alamat, harga, ukuran, jumlah kamar, fasilitas, biaya perawatan, serta photo unit properti.

Perusahaan ini masih banyak kendala yang dialami untuk mengembangkan dan memajukan bisnis seperti halnya pengelolaan, pengumpulan informasi mengenai apartemen, perumahan, dimana masih dilakukan dengan manual atau di catat manual dibuku catatan maupun di notepad handphone masing-masing Data collector.

Dengan “Aplikasi Manajemen Unit Properti Berbasis Php Mysql” inilah yang menjadi ketertarikan saya untuk membantu Data collector dalam mengelola, pengumpulan dan mendokumentasi informasi-informasi penting mengenai properti.

1.2 Rumusan Masalah

Berdasarkan permasalahan diatas maka dapat kita identifikasikan sebagai berikut :

1. Bagaimana membangun sebuah aplikasi web manajemen properti untuk membantu Data collector dalam menerapkan pada pekerjaan?
2. Bagaimana pengaruh adanya aplikasi manajemen untuk perusahaan tersebut ?

1.3 Batasan Masalah

1. Aplikasi ini hanya berbasis *web base*, tidak berbasis *dekstop* dan *mobile*.

2. Aplikasi ini tidak memiliki fitur maps untuk memberikan lokasi apartemen.

3. Aplikasi ini tidak memiliki fitur *chating.*

1.4 Tujuan dan Manfaat

Tujuan dari penulisan dan pembuatan aplikasi ini diharapkan berguna dan menambah wawasan bagi saya sebagai mahasiswa Universitas Mercu Buana, dimana ilmu yang telah didapat dari para dosen Fakultas Ilmu Komputer, selama kuliah bisa diaplikasikan dalam dunia kerja.

Tujuan dari tugas akhir ini adalah :

1. Membuat aplikasi menajemen unit properti berbasis php mysql, supaya dapat diterapkan di perusahaan yang bergerak dibidang properti.

2. Dengan adanya aplikasi manajemen unit properti ini, dapat membantu bisnis dibidang properti supaya berjalan dengan baik.

Manfaat dari penelitian tugas akhir ini yaitu :

1. Untuk membantu dan mempermudah Data collector dalam pengumpulan maupun mendokumentasikan data-data penting mengenai unit properti.
2. Data unit properti yang didokumentasikan lebih terstruktur.
3. Meningkatkan kinerja karwayan, khususnya data collector lebih efektif dan efisien.

1.5 Metode Penelitian

## 1.5.1. Metode Analisis

Dalam penulisan dan pembuatan aplikasi tugas akhir ini, penulis menggunakan metode sebagai berikut :

1. Studi Pustaka ( Pengumpulan Data)

Suatu pembahasan yang berdasarkan pada bukubuku referensi yang bertujuan untuk memperkuat materi pembahasan maupun sebagai dasar untuk menggunakan rumus-rumus tertentu dalam menganalisa dan mendesain suatu struktur. Studi pustaka digunakan untuk memecahkan masalah yang ada, baik untuk menganalisa faktor - faktor dan data pendukung maupun untuk merencanakan konstruksi, maka pada bagian ini kami menguraikan secara global pemakaian rumus-rumus dan persamaan yang akan digunakan untuk memecahkan masalah yang ada.

1. Metode observasi

Yaitu suatu metode pengumpulan data dengan cara melakukan pengamatan secara langsung terhadap cara kerja Data collector yang diteliti pada perusahaan yang bergerak dibidang properti.

## 1.5.2 Model Proses Pengembangan Perangkat Lunak

Metode adalah suatu cara yang disarankan untuk melakukan suatu hal. System Development Life Cycle (SDLC) adalah metodologi klasik yang digunakan untuk mengembangkan, memelihara dan menggunakan sistem informasi. Siklus hidup sistem itu sendiri merupakan metodologi, tetapi polanya lebih dipengaruhi oleh kebutuhan untuk mengembangkan sistem yang lebih cepat. Pengembangan sistem yang lebih cepat dapat dicapai dengan peningkatan siklus hidup dan penggunaan peralatan pengembangan berbasis computer. Metode SDLC menggunakan pendekatan sistem yang disebut pendekatan air terjun *(waterfall approach)*, yang menggunakan beberapa tahapan dalam pengembangan sistem. Tahap-tahap dalam pengembangan sistem sesuai dengan SDLC meliputi tahapan sebagai berikut.

1. Perencanaan sistem *(system planning)*

Merupakan tahap awal dari pengembangan sistem ,tahap ini bertujuan untuk mengidentifikasi dan memprioritaskan sistem informasi apa yang akan dikembangkan, sasaran-sasaran yang ingin dicapai, jangka waktu pelaksanaan serta mempertimbangkan dana yang tersedia dan siapa yang melaksanakan. Tahap perencanaan menjadi penting karena Permasalahan yang sebenarnya didefinisikan dan diidentifikasikan secara rinci.

1. Analis sistem *(system analyst)*

*Analisis system* adalah penelitian atas sistem yang telah ada dengan tujuan untuk merancang system baru atau memperbaharui system yang sudah ada.

1. Desain atau perancangan sistem *(system design)*

Rancangan system adalah penentuan proses dan data yang diperlukan oleh system baru. Jika system ini berbasis komputer, rancangan dapat menyertakan spesifikasi jenis peralatan yang akan digunakan.

1. Penerapan Sistem *(system implementation)*

Penerapan merupakan kegiatan memperoleh dan mengintegrasikan sumber daya fisik dan konseptual yang menghasilkan suatu sistem yang bekerja. Pada tahapan ini dilakukan beberapa hal yaitu: Coding, Testing, Instalasi. Dan Output dari tahapan ini adalah : source code, prosedur, pelatihan.

1. Perawatan sistem *(system maintenance)*

Selama manajer menggunakan system, berbagai modifikasi dibuat sehingga system terus memberikan dukungan yang diperlukan. Modifikasi ini disebut dengan pemeliharaan system (*System maintenance)*

1.6 Sistematika Penulisan

Dalam sistematika penulisan ini yang berjudul “Aplikasi web manajemen unit properti berbasis php mysql” terdiri dari susunan sebanyak 5 bab yaitu :

BAB I PENDAHULUAN

Pada bab ini menjelaskan tentang latar belakang, rumusan masalah, batasan masalah, tujuan dan manfaat, metode penelitian serta sistematika penulisan.

BAB II LANDASAN TEORI

Pada bab ini berisikan teori-teori yang mendukung dibuatnya “Aplikasi Web Manajemen Unit Properti Php Mysql”.

BAB III ANALISA DAN PERANCANGAN SISTEM

Pada Bab ini membahas tentang analisis, yang bertujuan untuk mengidentifikasi permasalahan-permasalahan yang terdapat pada sistem dan perancangan aplikasi.

BAB IV IMPLEMENTASI DAN PENGUJIAN

Bab ini membahas tentang implementasi dan pengujian aplikasi, dan setelah di implementasi maka dilakukan pengujian terhadap aplikasi tersebut, apakah sudah berjalan dengan baik serta melihat kekurangannya.

BAB V KESIMPULAN DAN SARAN

Pada Bab ini berisikan tentang kesimpulan dan saran dari hasil laporan tugas akhir

BAB II LANDASAN TEORI

2.1 Pengertian Web

WEB merupakan kumpulan informasi pada server komputer yang terhubung satu sama lain dalam jaringan internet, sedangkan aplikasi berbasis web (*web base aplication)* secara prinsip menyerupai aplikasi dalam komputer biasa, yang membedakannya adalah aplikasi menggunakan tag-tag html sebagai dasar tampilan, sedangkan aplikasi program komputer menggunakan platform bahasa pemrograman.

Untuk mendukung terciptanya web aplikasi yang dinamis dan interaktif, diperlukan struktur navigasi yang tepat dalam pembuatannya. Navigasi pada sebuah web aplikasi yang ditampilkan pada menu dan link yang menjadi petunjuk bagi pengguna terhadap halaman-halaman yang terdapat dalam web aplikasi. Pengguna akan semakin mudah menemukan halaman-halaman dalam web aplikasi, jika menu-menu link yang ada tampil secara terstuktur.

2.2 HTML (Hypertext Markup Language)

Menurut Abdul Kadir (2009) Aplikasi web yang paling dasar ditulis dengan menggunakan HTML. Sebagai mana diketahui, HTML (*HyperText Markup Language*) adalah bahasa standar untuk membuat halaman-halaman web.

2.3 PHP (Hypertext Preprocessor)

PHP (PHP: *Hypertext Preprocessor*) adalah bahasa scripting server-side yang dirancang untuk pengembangan web tetapi juga digunakan sebagai bahasa pemrograman tujuan umum. PHP digunakan karena fitur pengolahan teks nya yang memungkinkan keluaran dari HTML, gambar, file PDF dan bahkan teks seperti XHTML dan XML. Ini juga bekerja dengan baik dengan database MySQL dan dapat digunakan pada kebanyakan server web dan sistem operasi.

2.4 XAMPP

Xampp adalah paket server yang telah dikonfigurasikan yang berisi web server HTTP apache, MySQL sebagai database relasional dan PHP sebagai bahasascripting.

2.5 Bootstrap

Menurut Nahado (2014:14), “Bootstrap adalah platform untuk membuat interface website dan aplikasi berbasis web”. Bootstrap berisi kode HTML dan CSS yang telah dilengkapi desain untuk tipografi, bentuk, tombol, navigasi dan sebagainya. Bootstrap bertujuan untuk meringankan pembuatan dan pengembangan web.

2.6 Laravel

Menurut (Nugraha, 2014) Laravel adalah aplikasi web dengan sintak yang ekspresif dan elegan. Dengan Laravel, tugas-tugas umum developer dapat dikurangi pada sebagian besar proyek-proyek web seperti *routing*, *session*dan *caching*. Disamping itu, laravel berusaha menggabungkan pengalaman-pengalaman developmentdalam bahasa lain, seperti Ruby on Rails, ASP.NET, MVC.



Gambar 2. 1 Laravel

2.7 Basis data (*Database)*

Menurut (Rosa A.S, M.Shalahudin, 2013) Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat.

Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan menyimpan basis data ataupun bentuknya, entah berupa fie teks ataupun *Database Managament System(DBMS)*.

Kebutuhan basis data dalam sistem informasi meliputi :

1. Memasukan, menyimpan, dan mengambil data
2. Membuat laporan berdasarkan data yang telah disimpan.

2.8 MYSQL

Menurut Nugroho (2005) MySQL adalah sebuah program *database server* yang mampu menerima mengirimkan datanya dengan sangat cepat, multi user serta menggunakan perintah standar SQL (Strutured Query Language).

MySQL merupakan sebuah *database server* yang *free,*artinya kita bebas menggunakan databse ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensi.

Kelebihan MySQL :

1. MySQL sebagai *Database Managament System* (RDBMS)
2. MySQL adalah sebuah software database yang opensource, artinya program ini bersifat free atau bebas digunakan oleh siapa saja tanpa harus membeli dan membayar lisensi kepada pembuatnya.
3. MySQL merupakan database server, jadi dengan menggunakan database ini dapat menghubungkannya ke media internet sehingga dapat diakses dari jauh.
4. MySQL merupakan sebuah *database* *client*. Selain menjadi server yang melayani permintaan, MySQL juga dapat melakukan query yang mengakses database pada server. Jadi Mysql dapat juga berperan sebagai client.
5. MySQL merupakan sebuah database yang mampu menyimpan data berkapasitas sangat besar hingga berukuran Gigabyte sekalipun.

2.9 Algoritma Bubble Sort

*Bubble sort* merupakan salah satu jenis sorting. Ide dari algoritma ini adalah mengulang proses pembandingan antara tiap-tiap elemen array dan menukarnya apabila urutannya salah. Pembandingan elemen-elemen ini akan terus diulang hingga tidak perlu dilakukan penukaran lagi. Algoritma ini termasuk dalam golongan algoritma comparison sort, karena menggunakan perbandingan dalam operasi antar elemennya. Langkah-langkah dalam pengurutan dalam bubble sort, sebagai berikut: misalkan kita mempunyai sebuah array dengan elemen-elemen “ 4 2 5 3 9”.

Proses yang akan terjadi apabila menggunakan algoritma bubble sort adalah sebagai berikut:

Pass pertama

(4 2 5 3 9) menjadi (2 4 5 3 9)

(2 4 5 3 9) menjadi (2 4 5 3 9)

(2 4 5 3 9) menjadi (2 4 3 5 9)

(2 4 3 5 9) menjadi (2 4 3 5 9)

Pass kedua

(2 4 3 5 9) menjadi (2 4 3 5 9)

(2 4 3 5 9) menjadi (2 3 4 5 9)

(2 3 4 5 9) menjadi (2 3 4 5 9)

(2 3 4 5 9) menjadi (2 3 4 5 9)

Pass ketiga

(2 3 4 5 9) menjadi (2 3 4 5 9)

(2 3 4 5 9) menjadi (2 3 4 5 9)

2.10 Model *Waterfall*

Menurut (Rosa A.S, M.Shalahudin, 2013) Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linier*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian dan tahap pendukung (*support*). Berikut adalah gambar model air terjun :

Analisis Kebutuhan perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan pperangkat lunak agar dapat dipahami perangkat unak seperti apa yang dibutuhkan oleh user. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu didokumentasikan.

Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengkodean. Tahap ini menstralansi kebutuhan perangkat lunak dari tahap analisis kebutuhan representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak pada tahap ini juga perlu didokumentasikan.

Pembuatan kode program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

Pengujian

Pengujian fokus pada perangkat lunak secara dari segi lojik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

Pendukung (*support*) atau pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirim ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tetapi tidak untuk membuat perangkat lunak baru.

2.11 Entity Relationship Diagram (ERD)

Menurut Sutanta (2011:91) Entity Relationship Diagram adalah sebagai berikut :

1. Entitas Entitas merupakan suatu objek yang dapat dibedakan dari lainnya yang dapat diwujudkan dalam basis data. Objek dasar dapat berupa orang, benda, atau hal yang keterangannya perlu disimpan didalam basis data. Untuk menggambarkan sebuah entitas digunakan aturan sebagai berikut :

* 1. Entitas dinyatakan dengan simbol persegi panjang.
  2. Nama entitas dituliskan didalam simbol persegi panjang.
  3. Nama entitas berupa kata benda, tunggal.
  4. Nama entitas sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas.

2. Atribut Atribut merupakan keterangan-keterangan yang terkait pada sebuah entitas yang perlu disimpan dalam basis data. Atribut berfungsi sebagai penjelas pada sebuah entitas. Untuk menggambarkan atribut digunakan aturan sebagai berikut:

* 1. Atribut digambarkan dengan simbol ellips.
  2. Nama atribut dituliskan didalam simbol ellips.
  3. Nama atribut merupakan kata benda, tunggal.
  4. Nama atribut sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas.

3. Relasi Relasi merupakan hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Aturan penggambaran relasi adalah sebagai berikut :

* 1. Relasi dinyatakan dengan simbol belah ketupat.
  2. Nama relasi dituliskan didalam simbol belah ketupat
  3. Nama relasi berupa kata kerja aktif.
  4. Nama relasi sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas

Tabel 2. 1 Berikut tabel ERD

|  |  |
| --- | --- |
| Notasi | Keterangan |
| Entitas | Entitas adalah suatu objek yang dapat diedintifikasi dalam lingkungan pemakai. |
| Relasi | Relasi menunjukan adanya hubungan diantara sejumlah entitas yang berbeda. |
|  | Atribut berfungsi mendeskripsikan karakter entias (atribut yang berfungsi sebagai *key* diberi garis bawah). |
| Garis | Garis sebagai penghubung antara relasi dan entitas atau relasi dan entitas dengan atribut. |

2.12 UML (*Unified Modeling Language*)

Menurut (Rosa A.S, M.Shalahudin, 2013) Pada perkembangan teknologi perangkat lunak, diperlukannya adanya bahasa yang digunakan untuk memodelkan perangkat lunak yang akan dibuat dan perlu adanya standarisasi agar orang diberbagai negara dapat mengerti pemodelan perangkat lunak. Seperti yang kita ketahui bahwa menyatukan banyak kepala untuk menceritakan sebuah ide dengan tujuan memahami hal yang sama tidaklah mudah, oleh karena itu diperlukan sebuah bahasa pemodelan perangkat lunak yang dapat dimengerti oleh banyak orang.

Banyak orang yang telah membuat bahasa pemodelan pembangunan perangkat lunak sesuai dengan teknologi pemrograman yang berkembang pada saat itu, misalnya yang sempat berkembang dan digunakan banyak orang adalah *Data Flow Diagram* (DFD) untuk memodelkan perangkat lunak yang menggunakan pemrograman prosedural atau struktural, kemudian juga ada *State Transition Diagram* (STD) yang digunakan untuk memodelkan sistem *real time* (waktu nyata).

Pada perkembangan teknik pemrograman berorientasi objek, munculah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language* (UML). UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun dan mendokumentasikan dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.

2.13 Use Case Diagram

Menurut (Rosa A.S, M.Shalahudin, 2013) *Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior)* sistem informasi yang akan dibuat. Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, use case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

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

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

Tabel 2. 2 adalah simbol-simbol yang ada pada diagram use case

|  |  |
| --- | --- |
| Simbol | Deskripsi |
| Use Case | Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama *Use case.* |
| Aktor/*actor* | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi itu sendiri, walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan dengan menggunakan kata benda di awal frase nama aktor. |
| Asosiasi/ *association* | Komunikasi antara aktor dan *use case* yang berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan aktor. |
| Ekstensi /*extend* | Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri tanpa *use case* tambahan itu; mirip dengan prinsip *inheritance* pada pemrograman berorientasi objek; biasanya *use case* tambahan memiliki nama depan yang sama dengan *use case yang* ditambahkanmisalnya |
| Generalisasi/*generalization* | Hubungan generalisasi dan spesialisasi (Umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya. |

|  |  |
| --- | --- |
| Menggunakan/*include*/*uses* | Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya atau sebagai syarat dijalankan *use case* ini |

*Use case* nantinya akan menjadi kelas proses pada diagram kelas sehingga perlu dipertimbangkan penamaan yang dilakukan apakah sudah layak menjadi kelas atau belum sesuai dengan aturan pendefinisian kelas yang baik.

Setiap *use case* dilengkapi dengan skenario. Skenario *use case* adalah sebagai alur jalannya proses *use case* dari sisi aktor dan sistem. Berikut adalah format tabel skenario *use case* :

Tabel 2. 3 adalah tabel skenario use case.

|  |  |
| --- | --- |
| Aksi Aktor | Reaksi Sistem |
| Skenario Normal |  |
| Skenario Alternatif |  |

Skenario *Use case* dibuat per *use case* terkecil, misalkan untuk generalisasi maka skenario yang dibuat adalah *use case* yang lebih khusus. Skenario normal adalah skenario bila sistem berjalan normal tanpa terjadi kesalahan atau *error.* Sedangkan skenario alternatif adalah skenario bila sistem tidak berjalan normal, atau mengalami *error.* Skenario normal dan skenario alternatif dapat lebih dari satu. Alur dari skenario inilah yang nantinya menjadi dasar pembuatan diagram sekuen.

2.14 Activity Diagram

Menurut Rosa.A.S M-Shalahudin (2013) *Activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi intinya adalah suatu aktivitas yang dapat dilakukan oleh sistem.

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

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
2. Urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
4. Rancangan menu yang ditampilkan pada perangkat lunak.

Tabel 2. 4 simbol-simbol yang ada pada diagram use case

|  |  |
| --- | --- |
| Simbol | Deskripsi |
| Status Awal | Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal. |
| Aktivitas | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja. |
| Percabangan / *decision* | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu. |
| Penggabungan / *Join* | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu |

Lanjutan Tabel 2.5 *Activity Diagram*.

|  |  |
| --- | --- |
| Simbol | Deskripsi |
| Status akhir | Status akhir yang dilakukan sistem, sebuah diagram memiliki sebuah status akhir |
| Swimlane | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi. |

2.15 Sequence Diagram

Menurut Rosa.A.S M-Shalahudin (2013) diagram sequence menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sequence maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

Tabel 2. 5 adalah simbol-simbol yang ada pada diagram sekuen :

|  |  |
| --- | --- |
| Simbol | Deskripsi |
| Aktor | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi itu sendiri, jadi walaupun aktor adalah gambar orang, tetapi aktor belum tentu merupakan orang; biasanya dinyatakan kata benda di awal frase nama aktor |
| Garis hidup / *lifeline* | Menyatakan kehidupan suatu objek |
| Objek  nama objek : nama kelas | Menyatakan objek yang berinteraksi pesan |
| Waktu aktif | Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.` |
| Pesan tipe *create* | Menyatakan suatu objek membuat objek lain, arah panah mengarah pada objek yang dibuat. |

Lanjutan Tabel 2.5. *Sequence Diagram*

|  |  |
| --- | --- |
| Simbol | Deskripsi |
| Pesan tipe call | Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri. |
| Pesan tipe *send* | Menyatakan suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim. |
| Pesan tipe *return* | Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian. |
| Pesan tipe *destroy* | Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada *create* makan ada *destroy.* |

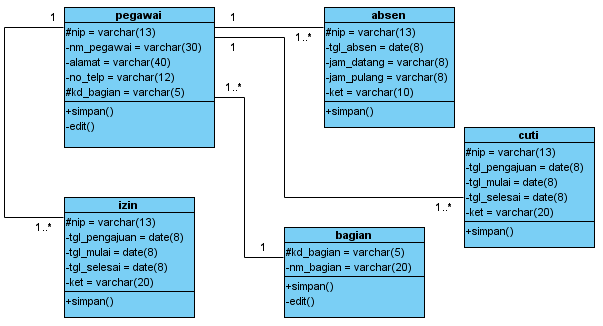
2.16 Class Diagram

Menurut(Roger S. Pressman , 2015) *Class Diagram* adalah diagram yang menunjukkan *class-class­* yang ada dari sebuah sistem dan saling berhubungan secara logika. Diagram ini menggambarkan struktur statis dari sebuah sistem. Karena itu, *class diagram* merupakan tulang punggung atau kekuatan dasar dari hamper setiap metode berorientasi objek termasuk *UML ( Unified Modeling Language ).*

Berikut komponen – komponen yang harus dipakai dalam pembuatan *class diagram :*

Tabel 2. 6 adalah tabel Class Diagram

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama Elemen** | **Keterangan** |
|  | Kelas | * Simbol untuk membangun sebuah pemrograman dengan objek * Terdiri 3 bagian, bagian **atas** adalah nama kelas, bagian **tengah** adalah atribut, dan bagian **bawah** adalah metode dari kelas tersebut |
|  | Garis Asosiasi | * Simbol yang menggambarkan adanya hubungan antara satu kelas dengan kelas lainnya |
|  | Generalisasi | * Simbol yang menandakan adanya generalisasi dari kelas input untuk menghasilkan data yang dibutuhkan |

****

Gambar 2. 2 Contoh Class diagram

2.17 Pengujian Black Box

**2.17.1**  **Metode Black Box**

Memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program.

**2.17.2** **Black Box**

Black box dapat menemukan kesalahan dalam kategori berikut :

a. Fungsi-fungsi yang tidak benar atau hilang

b. Kesalahan interface

c. Kesalahan dalam strutur data atau akses basisdata eksternal

d. Inisialisasi dan kesalahan terminasi

e. validitas fungsional

f. kesensitifan sistem terhadap nilai input tertentu

g. batasan dari suatu data`

BAB III ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis

Analisis bertujuan untuk memecahkan permasalahan-permasalahan yang telah ada, dengan adanya analisis kita dapat mencari solusi-solusi untuk permasalahan yang telah ditemukan.

**3.1.1 Analisis Sistem**

Kegiatan analisis Sistem adalah kegiatan untuk melihat sistem yang sedang berjalan, melihat bagian mana yang bagus dan tidak bagus, kemudian mendokumentasikan kebutuhan yang akan dipenuhi dalam sistem yang baru.

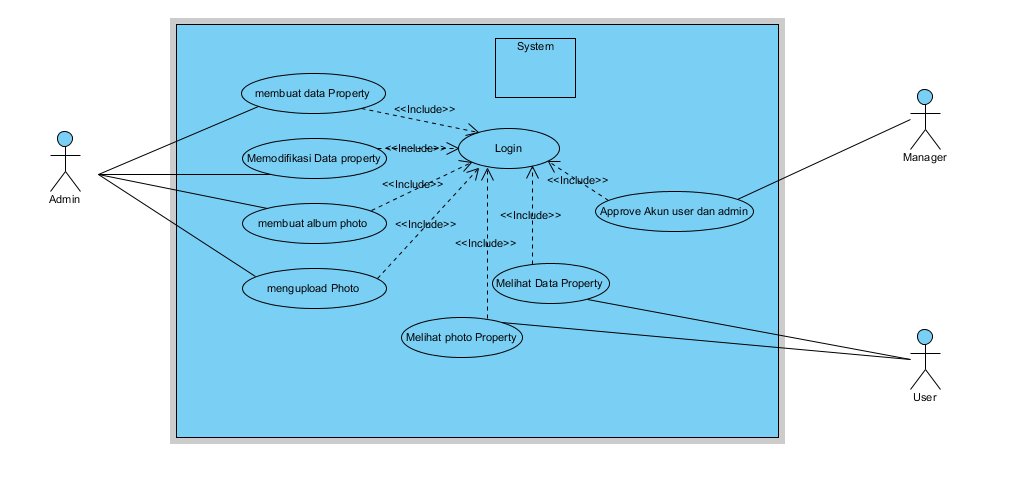
**3.1.2 Analisis Permasalahan**

Permasalahan pada penelitian ini yaitu masih banyaknya data-data properti yang tidak disimpan pada suatu sistem atau aplikasi, dan masih dicatat secara manual di buku catatan, dan photo-photo unit properti disimpan dikomputer pribadi masing-masing data collector. Dengan adanya aplikasi manajemen unit properti ini diharapkan data-data unit properti tersebut lebih mudah untuk dikelola dan diakses.

3.2 Perancangan Sistem

Padatahap perancangan sistem dan pembuatan aplikasi manajemen unit properti ini menggunakan UML (*Unfied Modeling Language),* yang terdiri dari perancangan *use case diagram,* perancangan *activity diagram,* perancangan *sequence diagram dan* perancangan *class diagram*.

## 3.2.1 Perancangan *Use Case Diagram*

Perancangan Use case diagram

Gambar 3. 1 Use case Aplikasi manajemen unit properti

Pada gambar 3.1 adalah *use case* aplikasi manajemen unit properti menjelaskan bahwa admin harus melakukan login untuk dapat melakukan aktifitas seperti membuat data properti, memodifikasi data, membuat album photo, mengupload photo. User juga harus login terlebih dahulu, tetapi perbedaan antara admin dan user yaitu admin dapat melakukan banyak aktivitas seperti membuat data, merubah data, menghapus data, membuat album photo, mengupload photo, atau sering juga disebut dengan CRUD (*create, read, update, delete*), sedangkan user hanya dapat melihat data properti dan melihat galery photo-photo properti.

Berikut tabel deskripsi use case diagram aplikasi manajemen unit properti:

Tabel 3. 1 Identifikasi Aktor

|  |  |
| --- | --- |
| **Aktor** | **Deskripsi** |
| Admin | Merupakan pengelola data properti |
| User | Merupakan pengguna aplikasi web manajemen unit properti |
| Manager | Manager merupakan aktor untuk *approve*/ menyutujui akun yang telah diregistrasi. |

Tabel 3. 2 Membuat data properti

|  |  |
| --- | --- |
| **Nama *Use Case*** | **Membuat data** |
| Aktor | Admin |
| Deskripsi | Admin dapat melakukan aktifitas seperti membuat data, mengubah data, serta menghapus data. |
| Pra Kondisi | Admin membuka aplikasi web dan melakukan login lalu pilih menu dasbord dan klik view details lalu pilih tambah data. |
| Skenario | Admin membuka aplikasi web dan melakukan login lalu klik dasbord dan klik view details. Setelah di klik views details maka sistem akan menampilkan data properti, dan juga terdapat button tambah data. |
| Tindakan Alternatif | Logout |

Tabel 3.2 terdapat nama use case yaitu Membuat data (*Create data*), Admin sebagai aktor. Jika Admin hendak membuat data properti baru, maka admin harus melakukan login terlebih dahulu.

Tabel 3. 3 Memodifikasi data properti

|  |  |
| --- | --- |
| **Nama *Use Case*** | **Memodifikasi data properti** |
| Actor | Admin |
| Deskripsi | Melakukan perubahan data properti |
| Pra Kondisi | Admin membuka aplikasi web, setelah itu harus melakukan login, lalu pilih menu Dasbord dan klik view details. |
| Skenario | Admin harus login terlebih dahulu sebelum melakukan perubahan data properti. |
| Tindakan Alternatif | Logout |

Tabel 3.3 berisi nama use case merubah data properti, admin sebagai aktor, admin dapat merubah data properti yang telah dibuat.

Tabel 3. 4 Menghapus Data

|  |  |
| --- | --- |
| Nama *Use Case* | Menghapus data properti |
| Actor | Admin |
| Deskripsi | Melakukan penghapusan data properti |
| Pra Kondisi | Admin membuka web dan melakukan login lalu supaya bisa melakukan aktivitas menghapus data. |
| Skenario | Admin dapat menghapus data properti jika berhasil login ke system, dengan memilih menu dasbord dan klik view details lalu pilih tombol hapus yang berwarna merah. |
| Tindakan Alternatif | Logout |

Tabel 3.4 berisi nama use case menghapus data properti, admin sebagai aktor, dimana admin dapat menekan tombol hapus untuk menghapus data properti.

Tabel 3. 5 Membuat Album Photo.

|  |  |
| --- | --- |
| Nama *Use Case* | Membuat Album photo |
| Actor | Admin |
| Deskripsi | Admin dapat membuat album photo |
| Pra Kondisi | Admin harus login terlebih dahulu, supaya bisa membuat album photo. |
| Skenario | Setelah admin berhasil login, maka admin dapat membuat album photo dengan pilih menu photo lalu pilih buat album photo /*create album.* |
| Tindakan Alternatif | Logout |

Tabel 3.5 berisi nama use case membuat album photo, admin sebagai aktor. Admin dapat membuat album photo, jika sudah berhasil login.

Tabel 3. 6 Upload Photo

|  |  |
| --- | --- |
| Nama *Use Case* | Upload Photo |
| Actor | Admin |
| Deskripsi | Admin dapat mengupload photo unit properti |
| Pra Kondisi | Admin harus login terlebih dahulu, supaya bisa upload photo unit properti |
| Skenario | Setelah admin selesai membuat album photo, langkah berikutnya yaitu upload photo, dengan memilih menu upload photo, dan terdapat beberapa pilihan seperti memilih apartemen, memilih album, memilih photo. |
| Tindakan Alternatif | Logout |

Tabel 3.6 berisi nama *use case* upload photo, admin sebagaiaktor. Admin dapat melakukan upload photo jika sudah membuat album sebelumnya.

Tabel 3. 7 Melihat Photo Properti

|  |  |
| --- | --- |
| Nama *Use Case* | Melihat Photo Properti |
| Actor | Admin/users/manager |
| Deskripsi | Untuk melihat galery photo properti |
| Pra Kondisi | Pilih menu daftar photo |
| Skenario | Admin/user/manager melakukan login terlebih dahulu untuk dapat melihat galery photo properti, setelah berhasil login pilih menu *list photo.* |
| Tindakan Alternatif | Pindah ke menu lain atau logout |

Tabel 3.7 berisi nama use case melihat photo, aktornya admin, users, manager. Admin, users, manager dapat melihat kumpulan photo-photo unit properti.

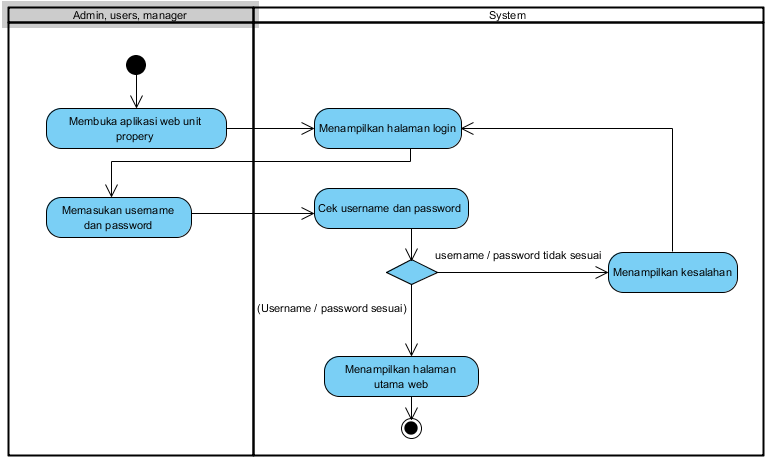
3.3 Activity Diagram

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

## 3.3.1 Activity Diagram *Login* Admin, User, Manager pada Aplikasi Manajemen web unit properti

Aktifitas *login* merupakan suatu aktifitas yang harus dilakukan admin, user maupun manager untuk melakukan proses seperti menambahkan data, merubah data, menghapus data, membuat album photo, mengunggah photo. Dalam suatu aktivitas ini terdapat beberapa aktivitas yaitu sebagai berikut :

1. Aktifitas yang pertama yaitu membuka halaman *login*, pada aktifitas ini admin, users, manager membuka aplikasi web manajemen unit properti untuk mengakses halaman login.
2. Aktifitas yang kedua yaitu menampilkan halaman *login*, pada aktifitas ini sistem menampilkan halaman login, dan terdapat *form login* yang harus di isi oleh admin, users, manager.
3. Aktifitas yang ketiga mengisi *form login*, pada aktifitasi ini, admin, users, managermengisi *form* yang berupa *username* dan *password*.
4. Aktifitas yang keempat yaitu mencari *username* dan *password*, pada aktifitas ini sistem akan melakukan proses mencocokan *username* dan *password* yang ada pada *database*.
5. Aktifitas yang kelima menampilkan pesan kesalahan, aktifitas ini terjadi jika *username* dan *password* yang admin dimasukkan tidak ada pada *database* atau terjadi kesalahan pada saat memasukan *username* dan *password*.
6. Aktifitas yang ke enam menampilkan halaman *admin*, aktifitas ini akan terjadi apabila *username* dan *password* yang dimasukan pada saat *login* sesuai dengan data yang ada pada database lalu sistem menampilkan halaman *admin*, setelah itu admin dapat melakukan banyak akfititas.

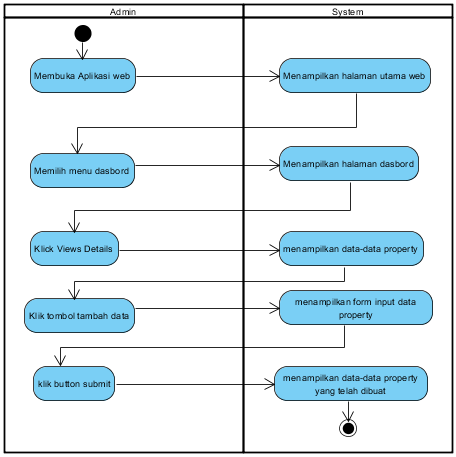


Gambar 3. 2 Activity Diagram Login Admin, users, dan manager.

## 3.3.2 Activity Diagram Menambahkan Data Unit Properti

Aktifitas menambahkan data unit properti merupakan aktifitas yang dilakukan oleh admin. Berikut adalah aktifitas yang dilakukan oleh admin :

1. Setelah berhasil login sebagai admin, system akan menampilkan halaman utama web.
2. Untuk memasukan data properti atau menambahkan data baru, admin memilih menu dasbord.
3. Pada tahap ini, system akan menampilkan menu dasbord, setelah itu pilih atau *click view details.*
4. Tahap selanjutnya yaitu system menampilkan data-data properti yang telah dibuat sebelumnya, dan terdapat *button* tambah data untuk menambah data baru, admin menekan tombol button tambah data, setelah itu sistem akan mengarahkan ke bagian halaman website tambah data.
5. Admin mengisi data-data properti, jika sudah diisi, maka klik button sumbit untuk menyimpan data kedalam database.
6. System akan menampilkan data-data properti yang telah kita buat atau kita simpan.

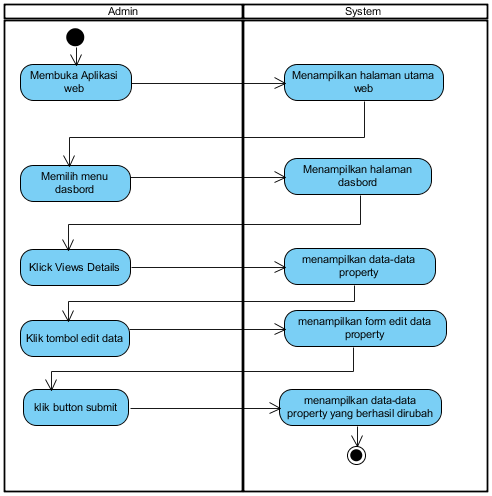


Gambar 3. 3 Activity Diagram Menambahkan Data Properti

## 3.3.3 Activity Diagram Merubah Data Properti

Aktifitas merubah data properti merupakan aktifitas yang dilakukan oleh admin jika terjadi kesalahan saat melakukan aktifitas memasukan data. Dalam aktifitas ini terdapat beberapa aktifitas yaitu :

1. Setelah berhasil login sebagai admin, system akan menampilkan halaman utama web.
2. Untuk merubah data properti, admin memilih menu dasbord.
3. Pada tahap ini, system akan menampilkan menu dasbord, setelah itu pilih atau *click view details.*
4. Tahap selanjutnya yaitu system menampilkan data-data properti yang telah dibuat sebelumnya, Admin harus memilih data yang hendak diubah / *edit.*
5. Jika sudah menentukan data yang hendak diubah, pilih *button* berwana hijau dengan icon pencil untuk merubah data, klik button tersebut, setelah itu sistem akan mengarahkan ke bagian halaman ubah data.
6. Pilih bagian data yang hendak diubah, jika sudah merubah, klik *button submit* untuk melakukan propses penyimpanan ke *database*.

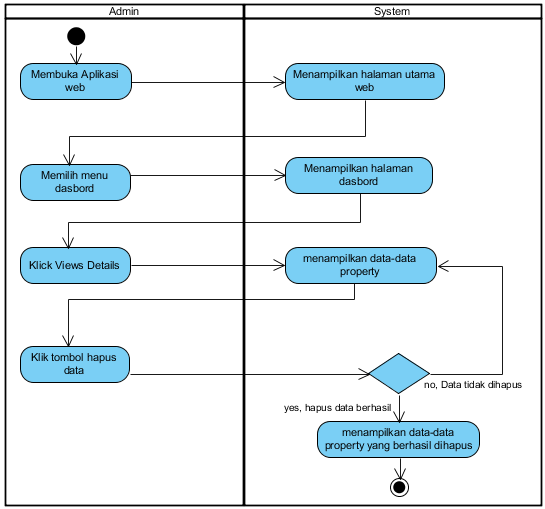


Gambar 3. 4 Activity diagram merubah data properti

## 3.3.4 Activity Diagram Menghapus data properti

aktivitas menghapus data properti merupakan aktifitas yang dilakukan oleh admin, jika terjadi kesalahan saat melakukan aktifitas memasukan data. Dalam aktifitas ini terdapat beberapa aktifitas yaitu :

1. Setelah berhasil login sebagai admin, system akan menampilkan halaman utama web.
2. Untuk merubah data properti, admin memilih menu dasbord.
3. Pada tahap ini, system akan menampilkan menu dasbord, setelah itu pilih atau *click view details.*
4. Tahap selanjutnya yaitu system menampilkan data-data properti yang telah dibuat sebelumnya, Admin harus memilih data yang hendak dihapus/*delete.*
5. Jika sudah menentukan data yang hendak dihapus, pilih atau klik *button* berwana merah dengan *icon* penghapus untuk menghapus data, setelah itu sistem system akan memunculkan notifikasi apakah kita hendak menghapus data atau tidak, jika kita melihi *yes* maka system akan menghapus data properti dari *database* namun jika memilih *no,* maka data tidak akan terhapus.

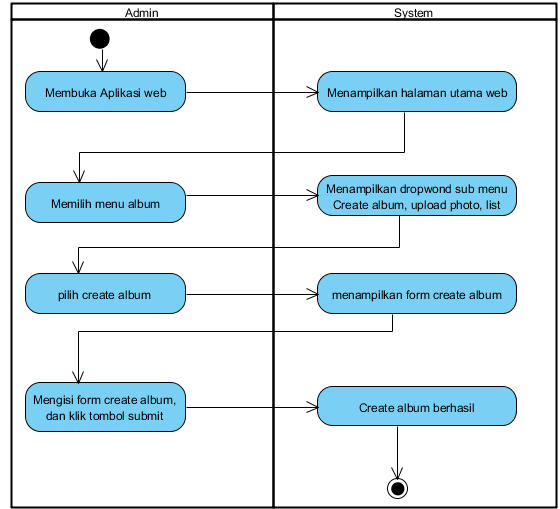


Gambar 3. 5 Activity Diagram Menambahkan Menu Makanan Baru

## 3.3.5 Activity Diagram Membuat Album Photo

Aktifitas membuat album photo merupakan aktivitas yang dilakukan oleh admin, berikut adalah aktivitas membuat album photo yang dilakukan admin :

1. Menjalankan menu utama, pada aktifitas ini admin membuka aplikasi web.
2. Admin memilih menu Album, dan kemudian pilih sub menu *create album.* System akan menampilkan halaman *create* album photo.
3. Admin mengisi *form* atau membuat album photo.
4. Jika sudah mengisi *form* *create* album photo, setelah itu klik tombol *submit* untuk membuat album photo, maka system akan memproses dan album photo berhasil dibuat.

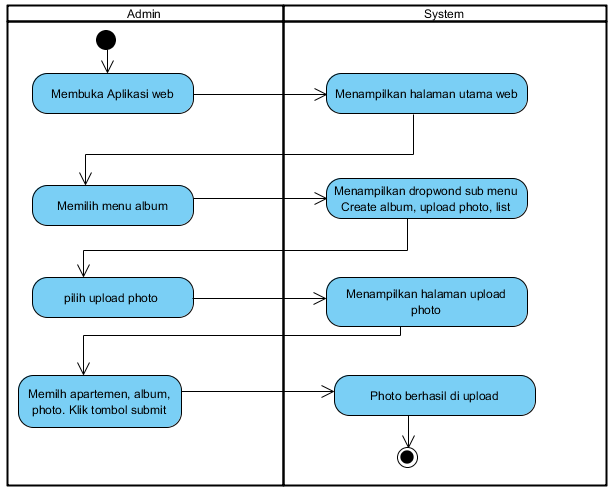


Gambar 3. 6 Activity Diagram membuat album photo.

3.3.6 Activity Diagram upload Photo

Aktifitas *upload photo* merupakan aktivitas yang dilakukan oleh admin, jika admin sudah selesai membuat album. berikut adalah aktivitas upload photo yang dilakukan admin :

1. Menjalankan menu utama, pada aktifitas ini admin membuka aplikasi web.
2. Admin memilih menu Album, dan kemudian pilih sub menu *upload photo.* System akan menampilkan halaman *upload photo*.
3. Admin memilih apartemen, album, dan photo-photo yang hendak di *upload*.
4. Jika sudah memilih apartemen, album, dan photo, klik tombol *submit* untuk melakukan proses upload photo ke *database*.

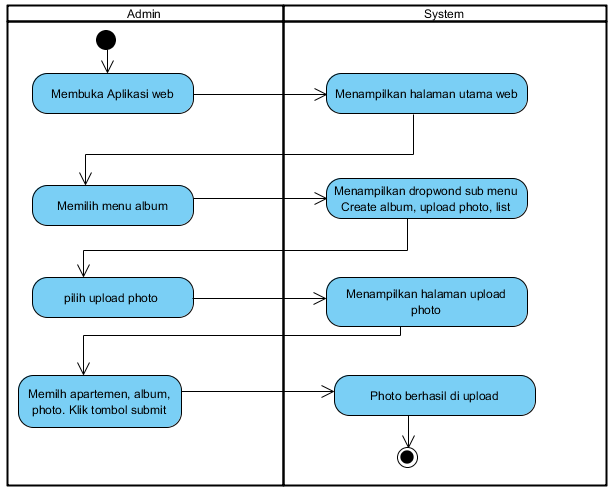


Gambar 3. 7 activity Diagram Upload Photo

## 3.3.7 Activity Diagram Merubah Album photo

Aktifitas merubah album photo merupakan aktivitas yang dilakukan oleh admin. berikut adalah aktivitas merubah album photo yang dilakukan admin :

1. Menjalankan menu utama, pada aktifitas ini admin membuka aplikasi web.
2. Admin, users dan manager memilih menu Album, dan kemudian pilih sub menu *List album,* Kemudian klik album photo, maka system akan menampilkan album photo seperti *bedroom, bathroom*, dll.
3. Setelah itu klik album yang hendak kita lihat photonya.
4. System akan menampilkan photo-photo unit properti yang telah kita *upload*.

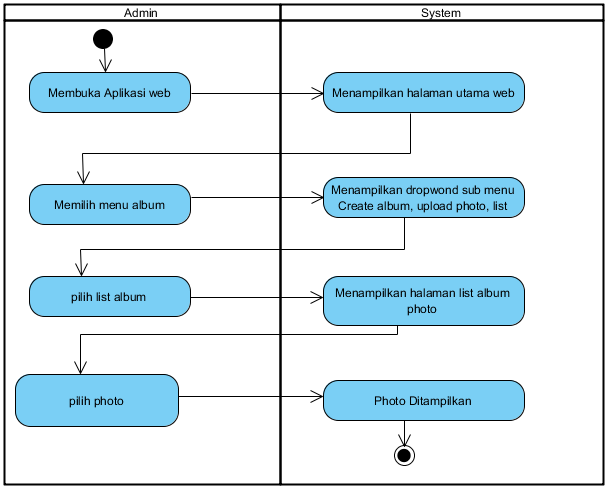


Gambar 3. 8 activity Diagram Upload Photo

## 3.3.8 Activity Diagram Melihat photo

Aktifitas melihat photo merupakan aktivitas yang dilakukan oleh admin, users dan manager. berikut adalah aktivitas melihat photo yang dilakukan admin, users dan manager :

1. Menjalankan menu utama, pada aktifitas ini admin membuka aplikasi web.
2. Admin, users dan manager memilih menu Album, dan kemudian pilih sub menu *List album,* Kemudian klik album photo, maka system akan menampilkan album photo seperti *bedroom, bathroom*, dll.
3. Setelah itu klik album yang hendak kita lihat photonya.
4. System akan menampilkan photo-photo unit properti yang telah di *upload*.

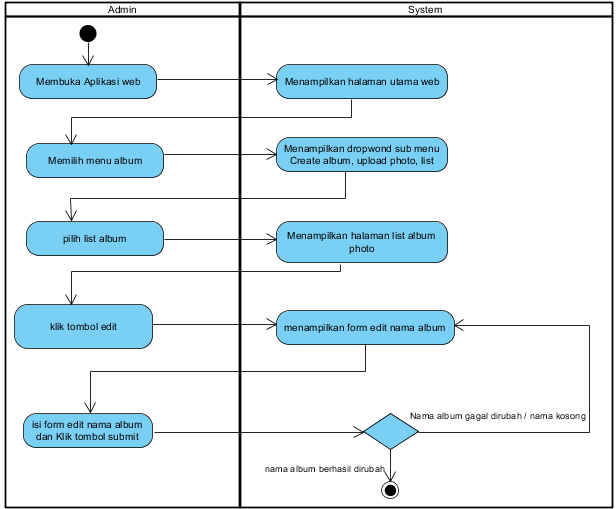


**Gambar 3.8** Activity Diagram melihat photo unit properti

**3.3.9 Activity Diagram Merubah Nama Album**

Aktifitas merubah album photo merupakan aktivitas yang dilakukan oleh admin. berikut adalah aktivitas melihat photo yang dilakukan admin :

1. Menjalankan menu utama, pada aktifitas ini admin membuka aplikasi web.
2. Admin memilih menu album, pilih sub menu *List album,* Kemudian system akan menampilkan album photo.
3. Setelah itu klik tombol edit, untuk melakukan perubahan pada nama album.
4. Isi nama album yang hendak dilakukan perubahan dan tekan tombol submit.
5. System akan memproses dan menampilkan nama *list album* yang telah dirubah. Jika terjadi kesalahan atau data kosong, maka system akan menampilkan kembali *form edit list album*

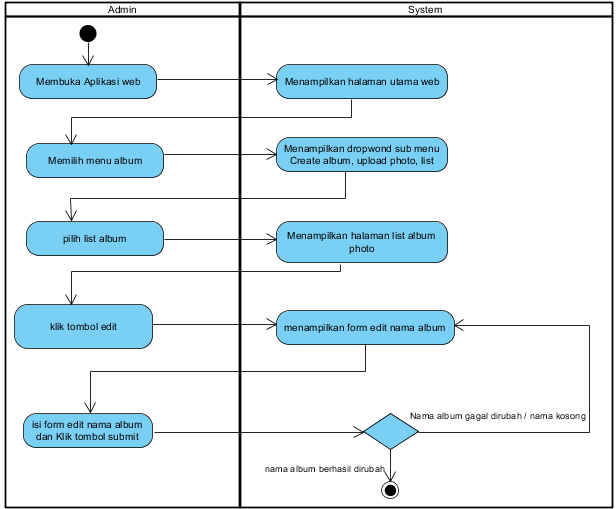


Gambar 3. 9 Activity Diagram melihat photo unit properti

## 3.3.9 Activity Diagram Menghapus Album

Aktifitas menghapus album photo merupakan aktivitas yang dilakukan oleh admin. berikut adalah aktivitas menghapus album photo yang dilakukan admin :

1. Menjalankan menu utama, pada aktifitas ini admin membuka aplikasi web.
2. Admin memilih menu album, pilih sub menu *List album,* Kemudian system akan menampilkan album photo.
3. Pilih album yang hendak dihapus, setelah itu klik tombol hapus, yang berfungsi untuk melakukan penghapusan pada nama album photo.
4. Setelah klik tombol hapus, system akan memberikan notifikasi apakah kita akan menghapus album, jika ingin menghapus klik *yes,* jika tidak ingin pilih atau klik no.
5. System akan memproses penghapusan data jika kita klik *yes.*



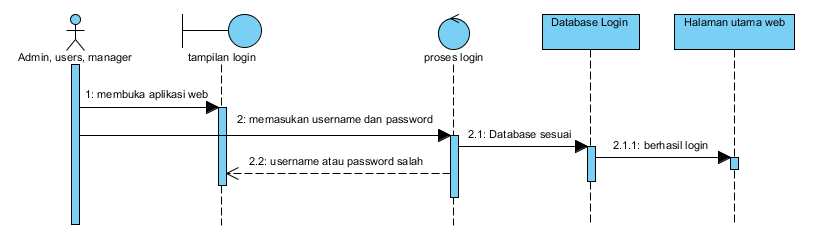
Gambar 3. 10 Activity Diagram menghapus album unit properti

3.4 Sequence Diagram

Diagram sequence digunakan untuk menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu, untuk menggambarkan diagram sequence maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

## 3.4.1 Sequence Diagram *Login* Admin dan User, dan manager

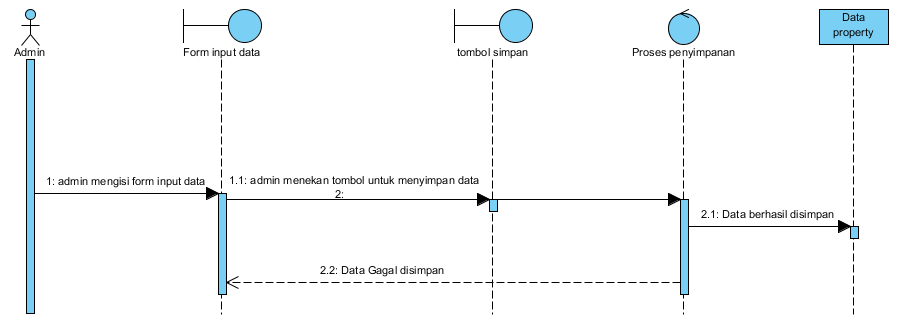
Pada gambar 3.11 diagram sekuen, menunjukan bahwa admin, users, manager harus melakukan login terlebih dahulu jika ingin masuk ke aplikasi. Admin, users, dan manager diminta untuk mengisi *username* dan *password,* jika sudah klik tombol masuk, maka sistem akan memproses atau memvalidasi *username* dan *password* apakah data tersebut ada pada database atau tidak.



Gambar 3. 11 Sequence Diagram Login Admin, User, dan Manager

## 3.4.2 Sequence Diagram Membuat Data properti

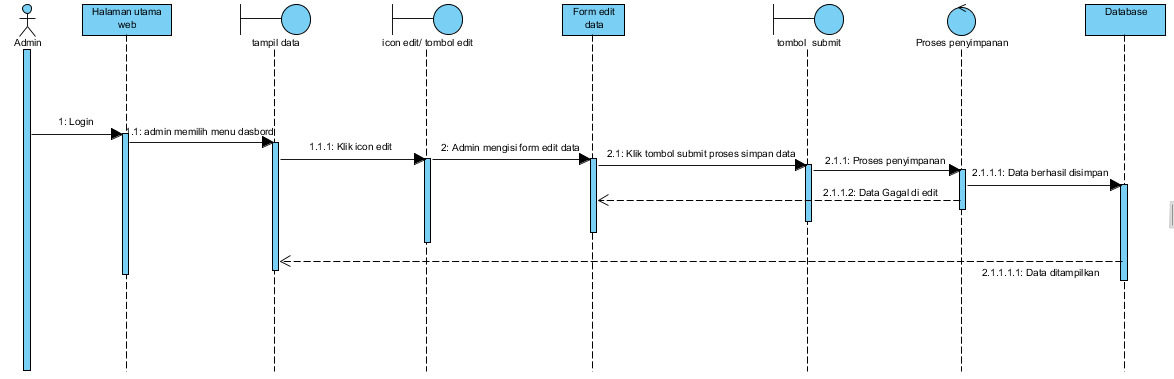
Pada gambar 3.12 dibawah menunjukkan bahwa Admin membuka aplikasi web, proses ini dimulai dari tahap login ke sistem, dan sistem akan melakukan proses validasi *username* dan *password*. Setelah berhasil login, maka sistem akan mengarahkan pada halaman utama web, kemudian admin pilih menu dasbord, dan klik *view details* dan sistem akan menampilkan data properti, dan terdapat juga tombol tambah data. Klik tombol tambah data untuk melakukan proses input data properti, sistem akan mengarahkan pada *form input* data.



Gambar 3. 12 Sequence Diagram membuat data properti

## 3.4.3 Sequence Diagram Merubah Data Properti

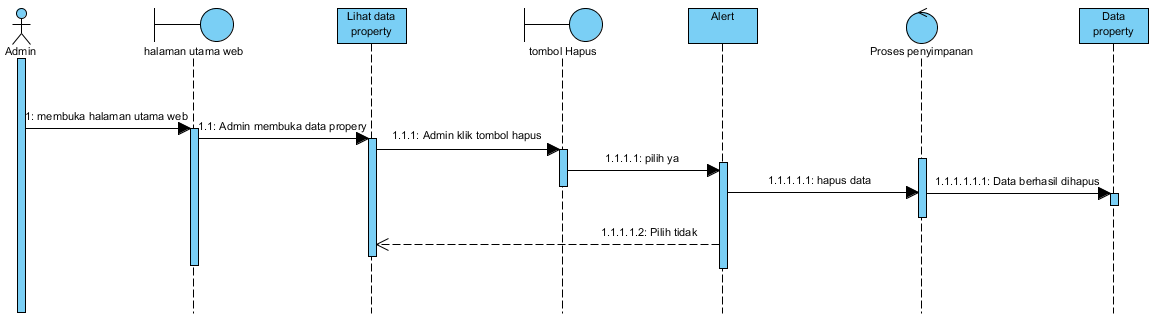
Pada gambar 3.13 dibawah ini menunjukkan bahwa Admin membuka aplikasi web, proses ini dimulai dari tahap login ke sistem, dan sistem akan melakukan proses validasi *username* dan *password*. Setelah berhasil login, maka sistem akan mengarahkan pada halaman utama web, kemudian admin pilih menu dasbord, dan klik *view details* dan sistem akan menampilkan data properti, dan terdapat juga *icon edit* data. Klik tombol *icon edit* data untuk melakukan proses merubah data, sistem akan mengarahkan pada *form edit* data.



Gambar 3. 13 Sequence Diagram Menambahkan Daftar Pelanggan

## 3.4.4 Sequence Diagram Menghapus Data Properti

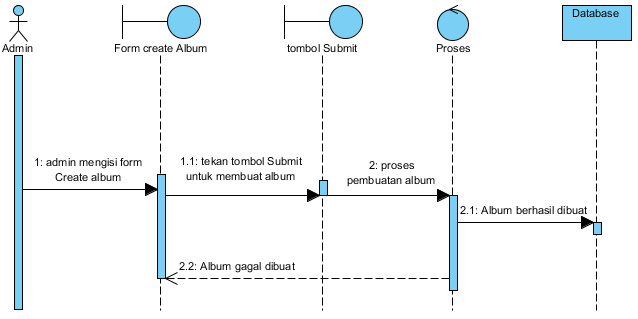
Pada gambar 3.14 dibawah menunjukkan bahwa Admin membuka aplikasi web, proses ini dimulai dari tahap login ke sistem, dan sistem akan melakukan proses validasi *username* dan *password*. Setelah berhasil login, maka sistem akan mengarahkan pada halaman utama web, kemudian admin pilih menu dasbord, dan klik *view details* dan sistem akan menampilkan data properti, dan terdapat juga *icon edit* data. Klik tombol *icon edit* data untuk melakukan proses merubah data, sistem akan mengarahkan pada *form edit* data.



Gambar 3. 14 Sequence Diagram Menghapus Data properti

**3.4.5 Sequence Diagram Membuat Album Photo**

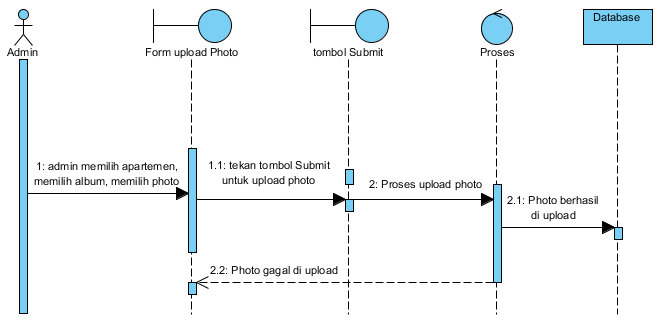
Pada gambar 3.15 dibawah menunjukkan bahwa Admin membuka aplikasi web, proses ini dimulai dari tahap login ke sistem, dan sistem akan melakukan proses validasi *username* dan *password*. Setelah berhasil login, maka sistem akan mengarahkan pada halaman utama web, kemudian admin pilih menu album, dan setelah itu pilih sub menu *create album*, setelah itu system akan menampilkan form *create album*. Isi form create album dan klik tombol submit untuk melakukan proses pembuatan album photo.



Gambar 3. 15 Sequence Diagram Membuat Album Photo

## 3.4.5 Sequence Diagram Upload Photo

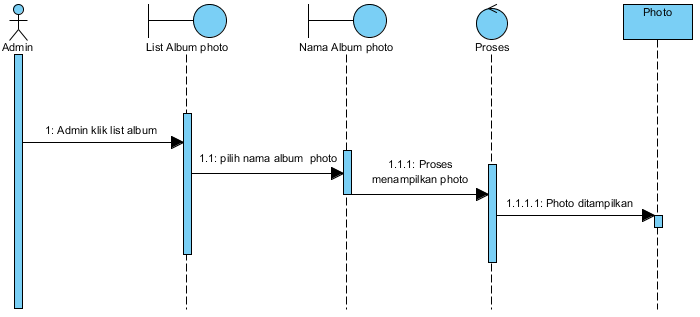
Pada gambar 3.16 dibawah menunjukkan bahwa Admin membuka aplikasi web, proses ini dimulai dari tahap login ke sistem, dan sistem akan melakukan proses validasi *username* dan *password*. Setelah berhasil login, maka sistem akan mengarahkan pada halaman utama web, kemudian admin pilih menu album, dan setelah itu pilih sub menu *upload photo*, setelah itu system akan menampilkan form *upload photo*. Sebelum melakukan proses upload, pertama pilih apartemen, pilih album dan pilih photo, jika telah dipilih klik tombol submit untuk melakukan proses upload photo.



Gambar 3. 16 Sequence Diagram Upload Photo

**3.4.6 Sequence Diagram Menampilkan Photo Unit Properti**

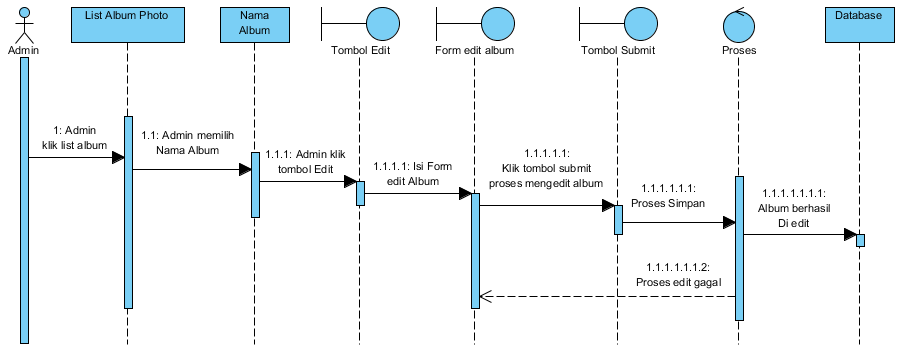
Pada gambar 3.17 dibawah menunjukkan bahwa Admin membuka aplikasi web, proses ini dimulai dari tahap login ke sistem, dan sistem akan melakukan proses validasi *username* dan *password*. Setelah berhasil login, maka sistem akan mengarahkan pada halaman utama web, kemudian admin pilih menu album, dan setelah itu pilih sub menu *list Album*, setelah itu system akan menampilkan *list album* photo.



Gambar 3. 17 Sequence Diagram Menampilkan Photo

## 3.4.6 Sequence Diagram Merubah Album Photo

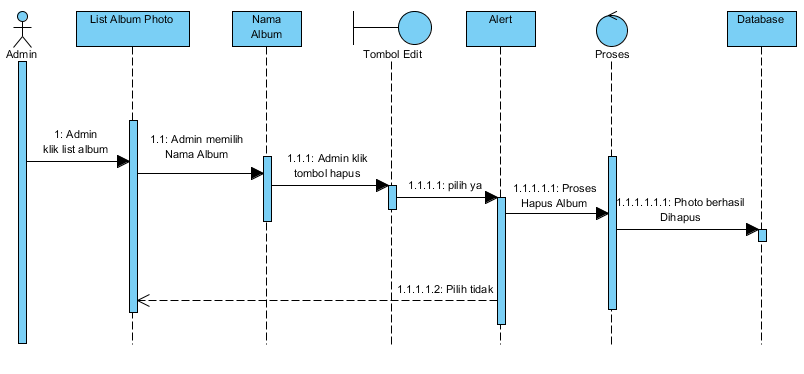
Pada gambar 3.18 dibawah menunjukkan bahwa Admin membuka aplikasi web, proses ini dimulai dari tahap login ke sistem, dan sistem akan melakukan proses validasi *username* dan *password*. Setelah berhasil login, maka sistem akan mengarahkan pada halaman utama web, kemudian admin pilih menu album, dan setelah itu pilih sub menu *list Album*, setelah itu system akan menampilkan *list album* photo.



Gambar 3. 18 Sequence Diagram Merubah Album Photo

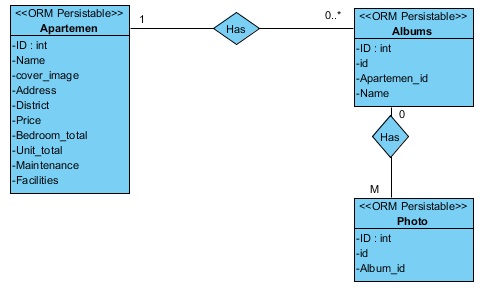
## 3.4.7 Sequence Diagram Menghapus Album Photo

Pada gambar 3.19dibawah menunjukkan bahwa Admin membuka aplikasi web, proses ini dimulai dari tahap login ke sistem, dan sistem akan melakukan proses validasi *username* dan *password*. Setelah berhasil login, maka sistem akan mengarahkan pada halaman utama web, kemudian admin pilih menu album, dan setelah itu pilih sub menu *list Album*, setelah itu system akan menampilkan *list album* photo. Terdapat tombol hapus album photo, klik untuk menghapus photo, maka system akan memproses penghapusan album.



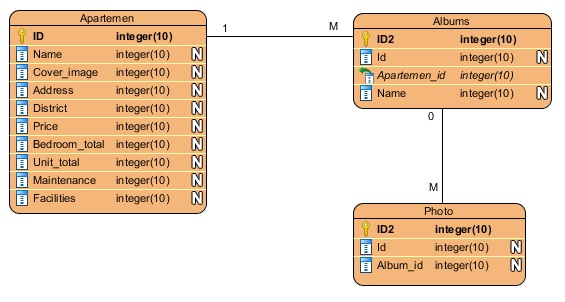
Gambar 3. 19 Sequence Diagram Menghapus Album Photo

3.5 Class Diagram Manajemen Unit Properti



Gambar 3. 20 Class Diagram

* 1. ERD (*Entity Relation Diagram*)



Gambar 3. 21 Entity Relation Diagram

3.7 Perancangan Database Manajemen Unit Properti

Berikut adalah tabel-tabel basis data yang digunakan untuk membangun aplikasi Manajemen Unit Properti.

Tabel 3. 8 Struktur Database Admin, Users, Manager

|  |  |  |
| --- | --- | --- |
| Nama Field | Type Data | Deskripsi |
| **Id** | int(10) | Id user |
| Name | varchar(25) | Nama user |
| **Email** | varchar(25) | Email user |
| Password | varchar(10) | Password user |
| Level | int(10) | Level user |
| Active | tinyint(1) | Status aktif dari admin, user, manager |
| remember\_token | varchar(25) | Jika browser di close, tidak perlu melakukan login lagi. |
| created\_at | Timestamp | Tanggal akun dibuat |

Tabel 3. 9 Struktur Database Apartement

|  |  |  |
| --- | --- | --- |
| Nama Field | Type Data | Deskripsi |
| **Id** | int(10) | Id Apartemen |
| Name | varchar(25) | Nama Apartemen |
| cover\_image | varchar(25) | Gambar Utama Apartemen |
| Address | varchar(15) | Alamat Apartemen |
| District | varchar(20) | District atau area apartemen |
| Price | int(10) | Harga Apartemen |
| bedroom\_total | int(11) | Jumlah Kamar |
| unit\_total | int(11) | Total unit apartemen |
| maintenance\_fee | int(10) | Biaya pemeliharaan |

Tabel 3. 10 Struktur Database Album Photo

|  |  |  |
| --- | --- | --- |
| Nama Field | Type Data | Deskripsi |
| **Id** | int(10) | Id album |
| Name | varchar(25) | Nama Album |
| created\_at | Timestamp | Tanggal album dibuat |

Tabel 3. 11 Struktur Database Upload Album Photo

|  |  |  |
| --- | --- | --- |
| Nama Field | Type Data | Deskripsi |
| **album\_id** | int(10) | Id album Photo |
| **apartment\_id** | int(10) | Id Apartemen |
| Images | Text | Photo properti |
| created\_at | Timestamp | Kapan photo dibuat |
| updated\_at | Timestamp | Kapan photo diupload |

Pada tabel-tabel diatas menjelaskan tentang struktur database aplikasi manajemen unit properti, dimana terdapat field sebagai nama database, type sebagai type dari nama database, sebagai contoh int, varchar, string dan masih banyak lagi tipe-tipe data, sedangkan keterangan adalah penjelasan tentang database.

Berikut adalah penjelasan tentang tipe-tipe data yang digunakan untuk membangun aplikasi manajemen unit properti :

* Varchar adalah tipe data untuk menampung data yang bertipe karakter atau alfabetic(a-z)  Tipe data VARCHAR ini dapat Mendefinisikan string sepanjang variabel n. Tipe data ini mirip dengan tipe data char, namun tipe data ini berguna bagi Anda yang tidak mengetahui secara pasti jumlah karakter yang akan dimasukkan oleh user. Tipe data ini juga bisa menerima nilai hingga 8000 karakter tapi biasanya jumlah karakter normal yang sering digunakan adalah 225.
* Datetime, Tipe data Datetime ini dapat menerima nilai tanggal dan waktu. Berfungsi untuk mendefinisikan tanggal, menyimpan tahun, bulan, hari, jam, menit, detik dan seperseribu detik (milliseconds). Tipe data Datetime ini menghabiskan 8 bytes untuk menyimpan data pada harddisk.
* Integer , Adalah sebuah bilangan bulat berukuran normal yang dapat ditanda tangani atau unsigned. Anda dapat menentukan lebar ukuran data sampai dengan 11 digit.
* Tinyint, Adalah sebuah bilangan yang sangat kecil yang dapat ditandatangani atau unsigned. Anda dapat menentukan lebar ukuran data sampai dengan 4 digit.

3.8 Perancangan Desain Sistem

## 3.8.1 Perancangan Desain Sistem Login

|  |
| --- |
| Btn\_2  Txt\_1  Username  Txt\_2  Password  Btn\_1 |

Deskripsi :

* Btn\_1 : Adalah *button* **/** tombol login
* Btn\_2 : Adalah button / tombol untuk register/pendaftaran.
* Txt\_1 : Tempat untuk mengisi username.
* Txt\_2 : Tempat untuk mengisi password.

## 3.8.2 Perancangan Desain Sistem Register

|  |
| --- |
| Btn\_4  Txt\_3  Txt\_4  Opt\_1  Txt\_5  Txt\_6  Txt\_7  Btn\_3 |

Deskripsi :

* Txt\_3 : Form tempat mengisi nama
* Txt\_4 : Form tempat mengisis email
* Opt\_1 : Form untuk memilih/*option* level
* Txt\_6 : Tempat untuk mengisi password.
* Txt\_7 : konfirmasi password.
* Btn\_3 : *Button /* tombol untuk register.
* Btn\_4 : *Button /* tombol untuk ke hamalan *login*.

## 3.8.3 Perancangan Desain Sistem Halaman Utama

|  |  |
| --- | --- |
| Btn\_5 | |
| UL\_1 | Li\_5 |
| UL\_2 |
| Li\_1 |
| Li\_2 |
| Li\_3 |
| UL\_3 |
|  |

Deskripsi :

* UL\_1 : *List* menu untuk dasbord / untuk menampilkan data
* UL\_2 : Dropdown untuk menu Galery Photo
* Li\_1 : Sub menu create album
* Li\_2 : Sub menu untuk upload photo.
* Li\_3 : Sub menu Untuk melihat *list* Album Photo.
* Ul\_3 : Menu untuk *approve* akun.
* Li\_5 : Sub Menu untuk lihat detail apartemen

## 3.8.4 Perancangan Desain Sistem Halaman Tampilan Data

|  |  |
| --- | --- |
|  | |
| UL\_1 | Btn\_6  **Apartment**  Nama Apartemen  NO  Nama  District  Address  Price  Total Bedroom  Total Unit  Maintenance Fee  Facilities  Txt\_8  IMG    Txt\_9  Txt\_10  Txt\_11  Txt\_12  Txt\_13  Txt\_14  Txt\_15  Txt\_16  Btn\_8  Btn\_7 |
| UL\_2 |
| Li\_1 |
| Li\_2 |
| Li\_3 |
| UL\_3 |
|  |

Deskripsi :

* UL\_1 : Menu untuk dasbord / untuk menampilkan data
* UL\_2 : Menu dropdown untuk menu Galery Photo
* Li\_1 : Sub menu create album
* Li\_2 : Sub menu untuk upload photo.
* Li\_3 : Sub menu Untuk melihat *list* Album Photo.
* Ul\_3 : Menu untuk *approve* akun.
* Li\_5 : Sub Menu untuk lihat detail apartemen
* IMG : Gambar utama apartemen.

Lanjutan deskripsi tampilan data:

* Btn\_6 : Tombol untuk membuat data apartemen (*Button create*).
* Btn\_7 : Tombol untuk mengubah data
* Btn\_8 : Tombol untuk menghapus data
* Txt\_8 : Tempat mengisi no apartemen
* Txt\_9 : Tempat mengisi nama apartemen
* Txt\_10 : Tempat mengisi district apartemen
* Txt\_11 : Tempat mengisi address apartemen
* Txt\_12 : Tempat mengisi price apartemen
* Txt\_13 : Tempat mengisi total bedroom apartemen
* Txt\_14 : Tempat mengisi total unit apartemen
* Txt\_15 : Tempat mengisi maintence fee apartemen
* Txt\_16 : Tempat mengisi facilities apartemen

## 3.8.5 Perancangan Desain Sistem Halaman Create Apartemen

|  |  |
| --- | --- |
|  | |
| UL\_1 | **Create Apartemen**  Name :  Txt\_8  Cover Image :  IMG  Txt\_9  Address :  Txt\_10  District :  Txt\_11  Price :  Txt\_12  Total Bedroom :  Txt\_13  Total unit :  Txt\_14  Maintenance fee :  Txt\_15  Facilities :  Txt\_16  Btn\_9 |
| UL\_2 |
| Li\_1 |
| Li\_2 |
| Li\_3 |
| UL\_3 |
|  |

Deskripsi :

1. UL\_1 : Menu untuk dasbord / untuk menampilkan data
2. UL\_2 : Menu dropdown untuk menu Galery Photo
3. Li\_1 : Sub menu create album
4. Li\_2 : Sub menu untuk upload photo.
5. Li\_3 : Sub menu Untuk melihat *list* Album Photo.
6. Ul\_3 : Menu untuk *approve* akun.
7. Li\_5 : Sub Menu untuk lihat detail apartemen
8. Btn\_6 : Tombol untuk membuat data apartemen (*Button create*).
9. CVR : Gambar utama apartemen.

## 3.8.6 Perancangan Desain Sistem Halaman Edit Apartemen

|  |  |
| --- | --- |
|  | |
| UL\_1 | **Edit Apartemen**  Name :  Txt\_8  Cover Image :  IMG  Txt\_9  Address :  Txt\_10  District :  Txt\_11  Price :  Txt\_12  Total Bedroom :  Txt\_13  Total unit :  Txt\_14  Maintenance fee :  Txt\_15  Facilities :  Txt\_16  Btn\_9 |
| UL\_2 |
| Li\_1 |
| Li\_2 |
| Li\_3 |
| UL\_3 |
|  |

Deskripsi :

UL\_1 : Menu untuk dasbord / untuk menampilkan data

UL\_2 : Menu dropdown untuk menu Galery Photo

Li\_1 : Sub menu create album

Li\_2 : Sub menu untuk upload photo.

Li\_3 : Sub menu Untuk melihat *list* Album Photo.

Ul\_3 : Menu untuk *approve* akun.

Li\_5 : Sub Menu untuk lihat detail apartemen

Btn\_6 : Tombol untuk membuat data apartemen (*Button create*).

CVR : Gambar utama apartemen.

## 3.8.7 Perancangan Desain Sistem Halaman Album

|  |  |
| --- | --- |
|  | |
| UL\_1 | **Create Album**  Apartement:  OPT\_2  Album Name :  Txt\_18  Btn\_10 |
| UL\_2 |
| Li\_1 |
| Li\_2 |
| Li\_3 |
| UL\_3 |
|  |

Deskripsi :

UL\_1 : Menu untuk dasbord / untuk menampilkan data

UL\_2 : Menu dropdown untuk menu Galery Photo

Li\_1 : Sub menu create album

Li\_2 : Sub menu untuk upload photo.

Li\_3 : Sub menu Untuk melihat *list* Album Photo.

Ul\_3 : Menu untuk *approve* akun.

Li\_5 : Sub Menu untuk lihat detail apartemen

OPT\_2: Memilih Apartemen

Txt\_18 : Tempat membuat/mengetik nama album

Btn\_10 : Tombol Submit

## 3.8.8 Perancangan Desain Sistem Halaman Upload Photo

|  |  |
| --- | --- |
|  | |
| UL\_1 | **Upload Photo**  Select Apartement:  OPT\_3  Select Album Name :  OPT\_4  Choose Image :  CHS  Btn\_11 |
| UL\_2 |
| Li\_1 |
| Li\_2 |
| Li\_3 |
| UL\_3 |
|  |

Deskripsi :

UL\_1 : Menu untuk dasbord / untuk menampilkan data

UL\_2 : Menu dropdown untuk menu Galery Photo

Li\_1 : Sub menu create album

Li\_2 : Sub menu untuk upload photo.

Li\_3 : Sub menu Untuk melihat *list* Album Photo.

Ul\_3 : Menu untuk *approve* akun.

Li\_5 : Sub Menu untuk lihat detail apartemen

OPT\_3: option memilih apartemen

OPT\_4: option memilih album

CHS : Tempat membuat/mengetik nama album

Btn\_11: Tombol Submit, untuk proses upload photo.

3.9 PerancanganInterface

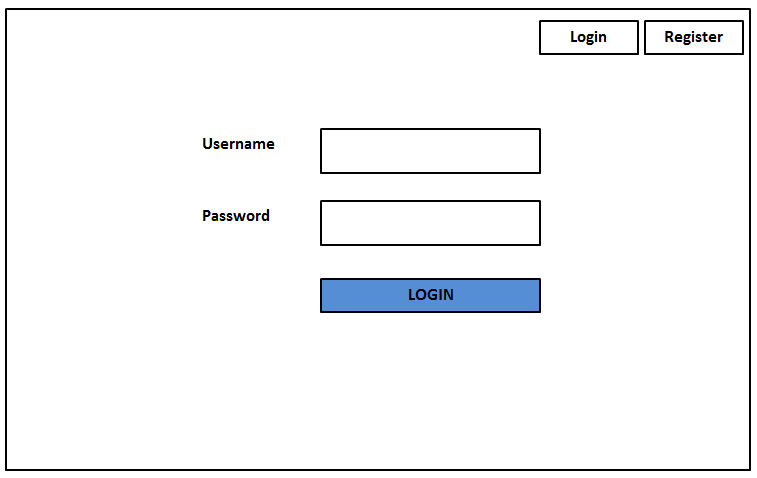
Perancangan interface aplikasi manajemen unit properti menjelaskan bagaimana suatu gambaran umum aplikasi yang hendak dibangun, tahap ini membantu programer untuk mendapat gambaran bagaimana aplikasi yang akan dibangun dan fitur-fitur apa saja yang terdapat didalamnya.

Dibawah ini beberapa perancangan aplikasi manajemen unit properti yang akan dibangun.

## 3.9.1 Perancangan *interface login*

Perancangan *interface login* yang menampilkan dua kolom, yang pertama kolom *username* dan kolom yang kedua *password* dan juga terdapat tombol login untuk melakukan proses masuk ke sistem jika di klik.

Berikut ini adalah *interface login* aplikasi manajemen unit properti

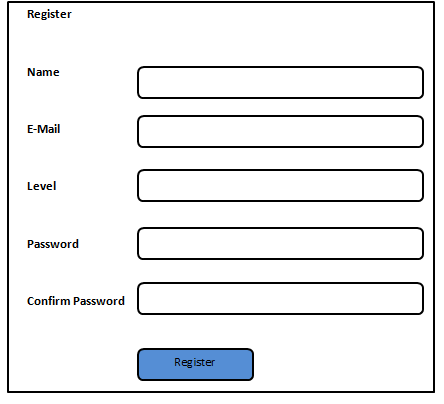


**Gambar 3.20** Rancangan Tampilan Login

## 3.9.2 Perancangan *interface register*

Perancangan *interface register* yang menampilkan lima kolom, yaitu kolom *name, e-mail, level, password, confirm password* dan juga terdapat tombol register untuk melakukan proses masuk ke sistem jika di klik.

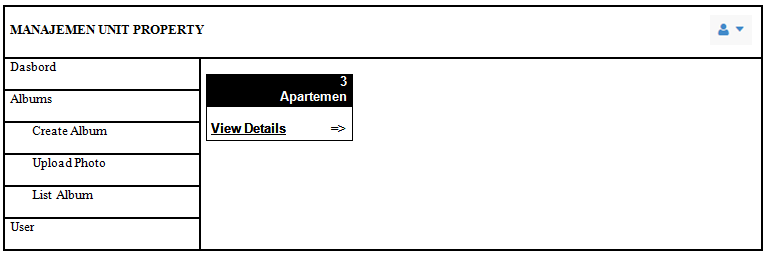
Berikut ini adalah *interface register* aplikasi manajemen unit properti



**Gambar 3.21** Rancangantampilanregister

## 3.9.3 Perancangan Tampilan Utama Web

Perancangan tampilan utama web adalah tampilan yang menampilkan beberapan menu yang terdiri dari menu dasbord, menu album, dan juga terdiri dari sub menu *create album, upload photo,* dan *list Album.* Gambar dibawah ini adalah tampilan halaman utama web.

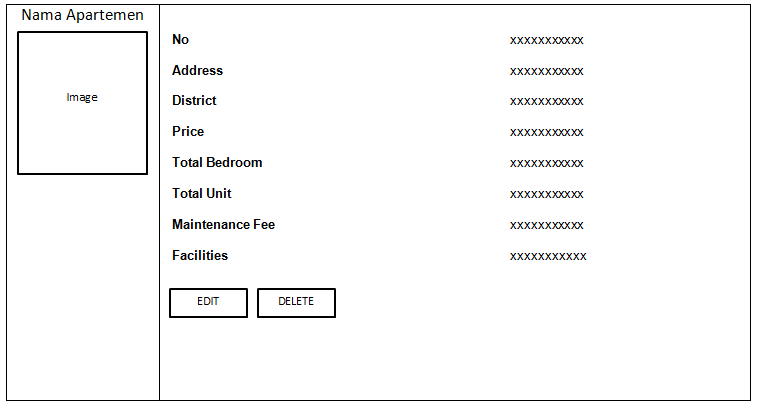


**Gambar 3.22** Rancangan Tampilan Menu Utama

**3.7.4 Perancangan Tampilan Data Unit Properti**

Perancangan menampilkan data adalah rancangan antar muka yang menampilkan data-data telah dibuat oleh admin, dengan menampilkan *name, address, district, price, total bedroom, total unit, maintenace fee, facilities.*

Berikut adalah perancangan menampilkan data properti.

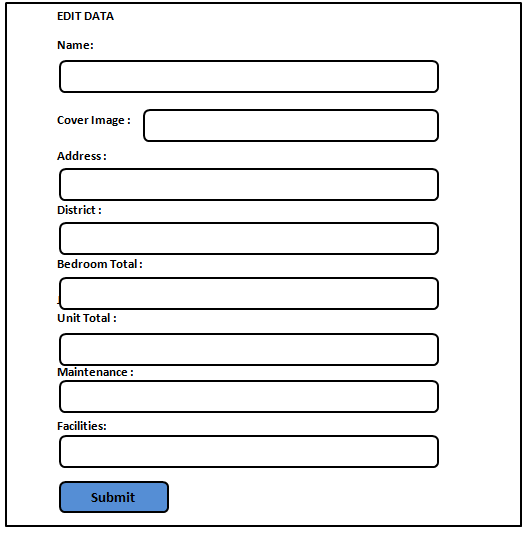


**Gambar 3.23** Rancangan Menampilkan Data

## 3.9.4 Perancangan Merubah Data (*edit*)

Perancangan tampilan merubah data yaitu *interface* yang menampilkan data-data properti yang akan dirubah.

Berikut tampilan antar muka halaman data penjual aplikasi pemesan makanan kantin kampus online.

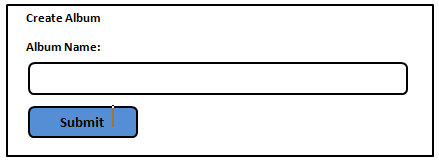


**Gambar 3.24** Rancangan Tampilan Data Penjual

## 3.9.5 Perancangan Tampilan Membuat Album Photo, Upload Dan List Album Photo.

Perancangan membuat album adalah rancangan untuk membuat *interface* aplikasi manajemen unit properti yaitu dengan menampilkan fitur seperti *create album, upload photo, list album phot .*

Berikut tampilan interface halaman *create album* aplikasi manajemen unit properti.

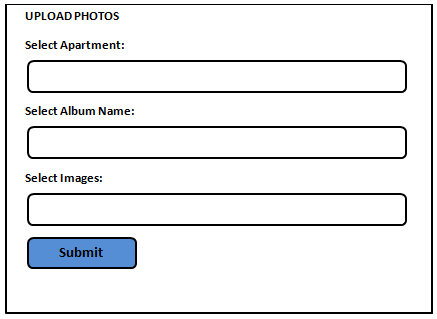
****

**Gambar 3.25** Rancangan Tampilan Membuat Album photo

## 3.9.6 Perancangan Tampilan Upload Photo.

Perancangan membuat album adalah rancangan untuk membuat *interface* aplikasi manajemen unit properti yaitu dengan menampilkan fitur seperti *create album, upload photo, list album phot .*

Berikut tampilan interface halaman *upload photo* aplikasi manajemen unit properti :



**Gambar 3.26** Rancangan Tampilan Upload Photo

## 3.9.7 Perancangan Tampilan List Album.

Perancangan membuat album adalah rancangan untuk membuat *interface* aplikasi manajemen unit properti yaitu dengan menampilkan fitur seperti *create album, upload photo, list album phot .*

Berikut tampilan interface halaman *List Album* aplikasi manajemen unit properti :



**Gambar 3.27** Rancangan Tampilan List Album

BAB IV IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi Sistem

Tahap implementasi dan pengujian sistem dilakukan setelah tahap analisis dan perancangan selesai dilakukan. Dalam menjalankan suatu sistem perlu diperhatikan sistem spesifikasi apa saja yang dapat menunjang supaya sebuah sistem dapat berjalan dengan optimal. Dimana spesifikasi tersebut mencakup perangkat lunak (*software*) dan perangkat keras (*hardware*).

4.2 Spesifikasi kebutuhan perangkat lunak dan perangkat keras

Berikut ini adalah spesifikasi dari perangkat lunak maupun perangkat keras yang dibutuhkan agar sistem dapat berjalan dengan baik.

Adapun spesifikasi perangkat keras sebagai berikut :

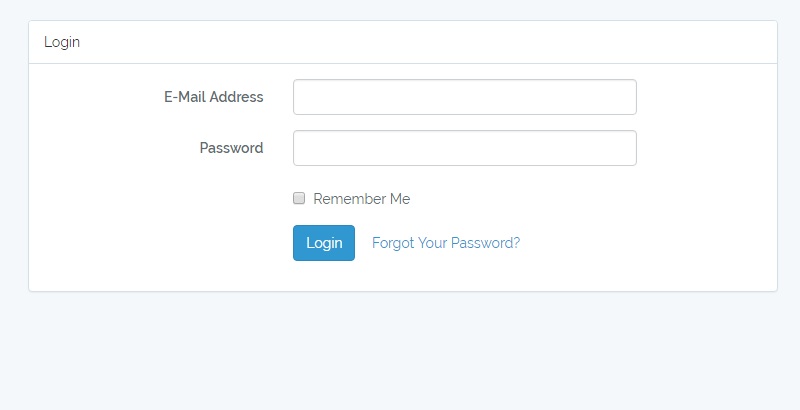
1. Perangkat Keras :
2. Laptop Dell dengan Prosesor Intel(R) Celeron(R) 3205U @1.50 GHz (2 CPUs), ~ 1.5 GHz.
3. Memory Ram 4096 MB (4GB)
4. Hardisk 500 GB
5. VGA 1696 MB
6. Perangkat Lunak :
7. Microsoft Windows 7 Ultimate 64 bit, sistem operasi yang digunakan pada *website* ini.
8. Sublime, yaitu text editor yang digunakan untuk melakukan penulisan kode-kode program.
9. MySQL, yaitu aplikasi database, yang digunakan untuk menyimpan berbagai data.
10. Google Chrome, digunakan sebagai browser untuk menjalankan aplikasi website ini.

4.3 Implementasi Tampilan

Tahap ini implementasi tampilan ini yaitu dengan menampilkan gambar-gambar dari sistem yang telah dibuat.

## 4.3.1 Implemen Tampilan Pada Halaman Login

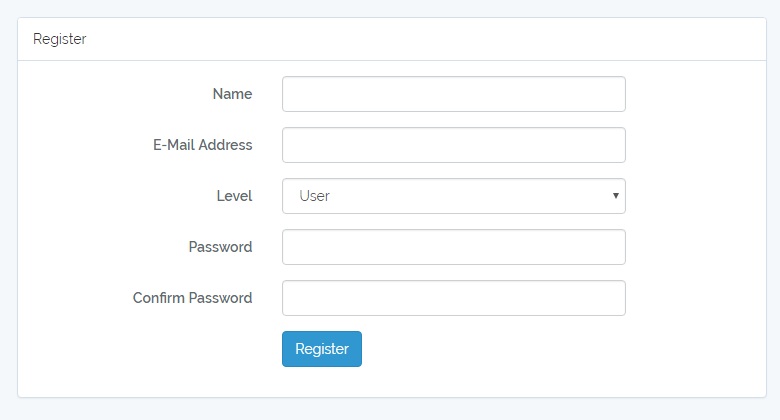
Berikut adalah tampilan halaman login pada aplikasi manajemen unit properti.



Gambar 4. 1 Tampilan login

4.3.2 Implemen Tampilan Pada Halaman Register

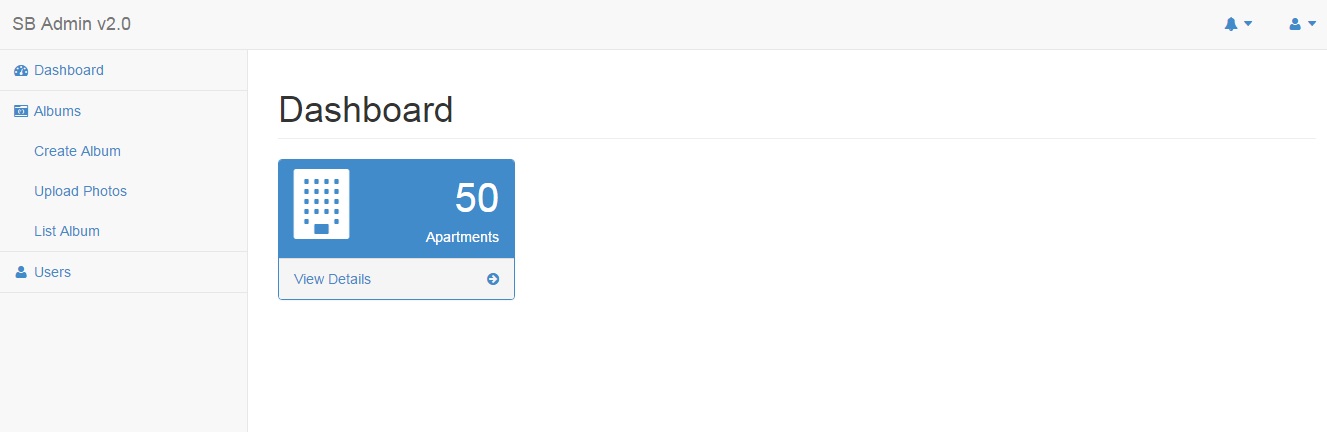
Berikut adalah tampilan halaman register pada aplikasi manajemen unit properti.



Gambar 4. 2 Tampilan register

## 4.3.3 Implemen Tampilan Pada Halaman Utama

Berikut adalah tampilan halaman utama pada aplikasi manajemen unit properti.



Gambar 4. 3 Tampilan Halaman utama

4.3.4 Implemen Tampilan Pada *Create* Apartemen

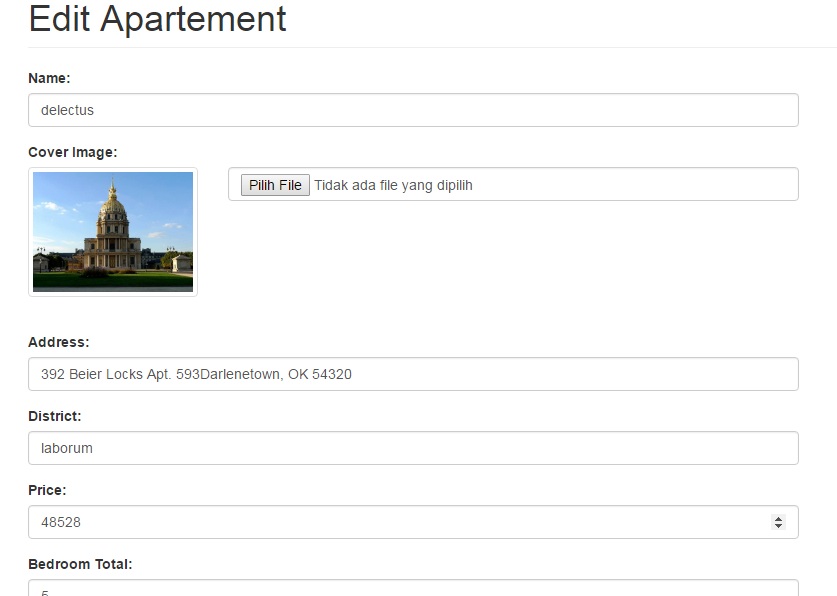
Berikut adalah tampilan create apartemen pada aplikasi manajemen unit poperty.



Gambar 4. 4Tampilan Create apartemen

## 4.3.5 Implemen Tampilan Pada Edit Apartemen

Berikut adalah tampilan edit apartemen pada aplikasi manajemen unit poperty.



Gambar 4. 5 Tampilan form edit apartemen

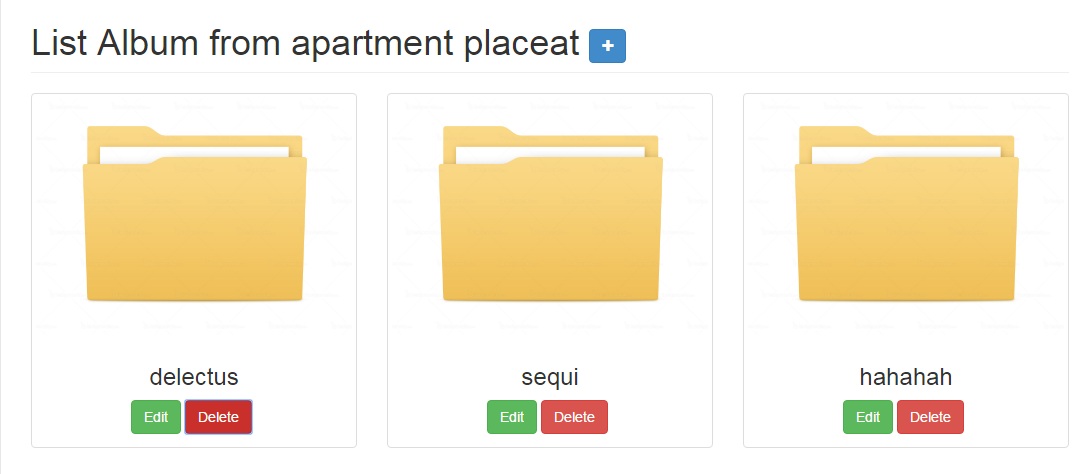
## 4.2.6 Implemen Tampilan Pada Create Album

Berikut adalah tampilan create album pada aplikasi manajemen unit poperty.



Gambar 4. 6 Tampilan create album photo

4.3.6 Implemen Tampilan Pada *List Album*

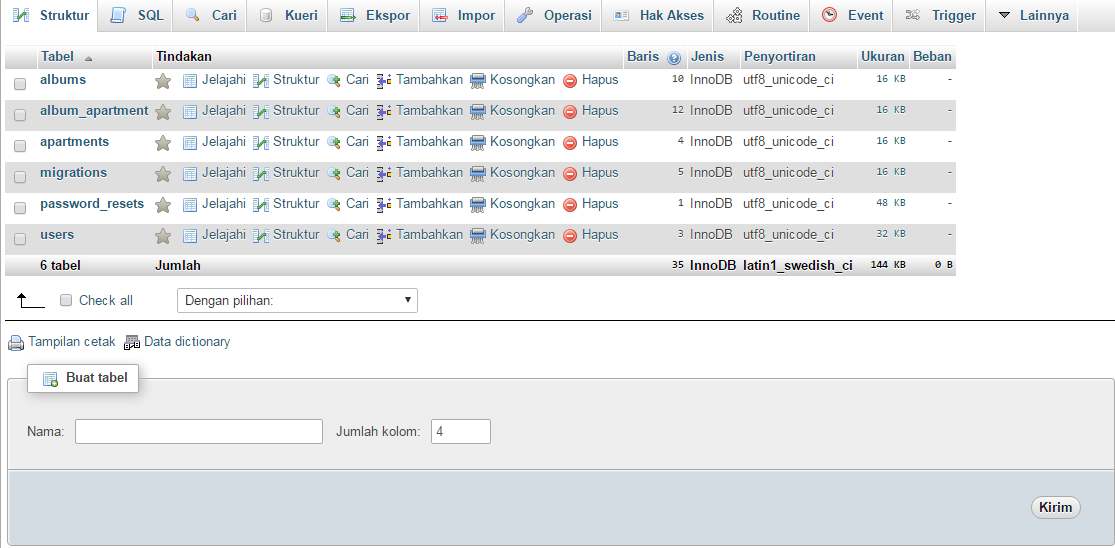
Berikut adalah gambar dari tampilan *list album photo* aplikasi manajemen unit properti

Gambar 4. 7 Tampilan list album photo

4.4 Implementasi Basis Data

Dalam pembuatan basis data untuk aplikasi manajemen unit properti ini menggunakan PhpMyAdmin versi 3.2.2 yang terdapat dalam aplikasi XAMPP.

Berikut adalah gambar database dari aplikasi manajemen unit properti :



Gambar 4. 8 Tampilan Database Unit Propery

4.4.1 Tabel  *Database Albums*

Tabel Album terdiri dari empat *field* yaitu id, name, create\_at, update\_at. Field inilah yang akan diisi dengan nama-nama album photo, ketika melakukan pembuatan album pada aplikasi manajemen unit properti.

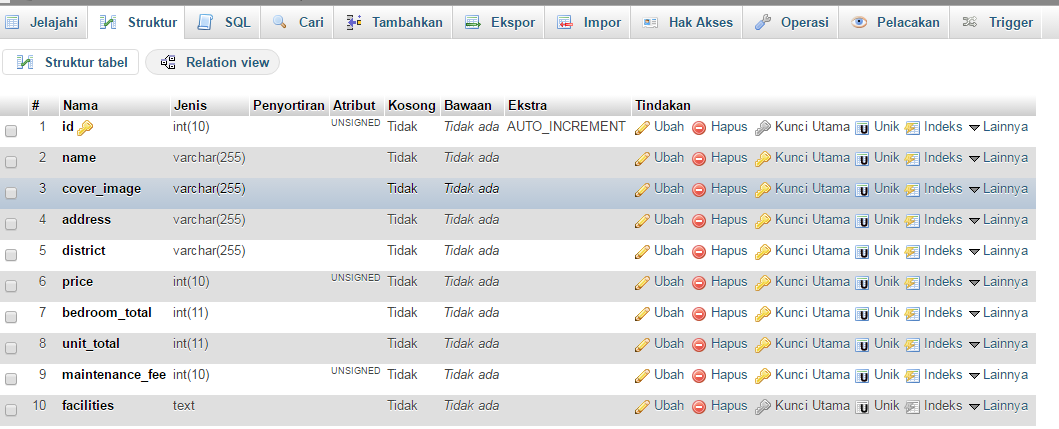


Gambar 4. 9 Tampilan Database Albums

4.4.2 Tabel *Database* Apartemen

Tabel apartemen terdiri dari sepuluh *field* yang terdiri dari id, name, cover\_image, address, district, price, bedroom\_total, unit\_total, maintenance\_fee, facilities.

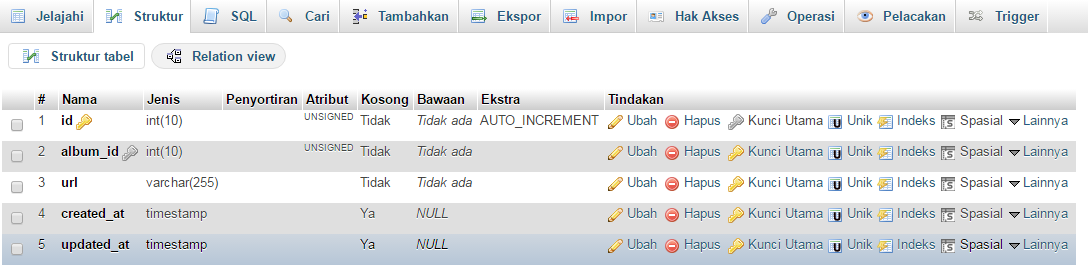
Berikut adalah gambar *database* apartemen



Gambar 4. 10 Tampilan Database Apartemen

## 4.4.3 Table *Database* Photos

Tabel database photos yang terdiri dari lima field yaitu id, Album\_id, url, create\_at, update\_at. Berikut adalah gambar database photos



Gambar 4. 11 Tampilan Database Photos

## 4.4.4 Tabel *Database* User

Tabel database photos yang terdiri dari sembilan field yaitu id, name, email, password, level, active, remeber\_token, create\_at, update\_at.



Gambar 4. 12 Tampilan Database Apartemen

4.4 Pengujian Fungsionalitas

Pada tahap ini akan dilakukan pengujian menggunakan metode aplikasi *blackbox.* Tujuan pengujian ini yaitu untuk melihat apakah setiap fungsi pada aplikasi berjalan dengan baik atau tidak, dan juga untuk mengetahui dibagian mana saja terdapat kesalahan, jika terjadi kesalahan maka segera diperbaiki.

## 4.4.1 Lingkungan Pengujian

Aplikasi manajemen unit properti ini dilakukan uji coba menggunakan laptop Dell.

Spesifikasi Laptop Dell:

1. Laptop Dell dengan Prosesor Intel(R) Celeron(R) 3205U @1.50 GHz (2 CPUs), ~ 1.5 GHz.
2. Memory Ram 4096 MB (4GB)
3. Hardisk 500 GB
4. VGA 1696 MB

4.4.2 Skenario Pengujian

Berikut adalah skenario pengujian yang dilakukan :

Tabel 4. 1 Skenario Pengujian

|  |  |  |  |
| --- | --- | --- | --- |
| No | Nama Pengujian | Sifat Kegiatan | Hasil Pengujian |
| 1. | Masuk ke halaman Login | Normal | Menampilkan halaman form login |
| 2. | Memilih menu register | Normal | Menampilkan halaman form register |
| 3. | Memilih Menu Dasbord | Normal | Menampilkan halaman data properti |
| 4. | Memilih Create Apartemen (tambah apartemen) | Normal | Menampilkan halaman input data apartemen |
| Tidak normal | Menampilkan alert |
| 5. | Memilih tombol hapus data | Normal | Menghapus data |
| Tidak normal | Menampilkan alert |
| 6. | Memilih tombol edit data | Normal | Menampilkan *form* edit data |
| Tidak normal | Menampilkan alert |

**Lanjutan Tabel 4.1**

|  |  |  |  |
| --- | --- | --- | --- |
| 7. | Memilih Menu Album | Normal | Menampilkan |
| 8. | Memilih Sub menu upload Photo | Normal | Menampilkan halaman upload photo. |
| 9. | Memilih Sub Menu List Album | Normal | Menampilkan List album photo-photo. |
| 10 | Memilih Tombol Hapus Album | Normal | Album dihapus |
| 11. | Memilih Tombol Edit | Normal | Menampilkan Form Edit Album |
| 12 | Melakukan Pencarian | Normal | Menampilkan data yang dicari |
| 13. | Melakukan Pengurutan (*Sorting* ) | Normal | Data diurutkan |
| 12. | Memilih keluar aplikasi (*logout*) | Normal | Keluar aplikasi, dan diarahkan ke halaman login. |

**4.4.3 Hasil Pengujian**

Berikut ini adalah hasil pengujian

Tabel 4. 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Nama Pengujian | Menu Yang diuji | Sifat Kegiatan | Hasil Pengujian yang diharapkan | Hasil Pengujian |
| 1. | Halaman Utama | Login | Normal | Menampilkan halaman form login | Sesuai |
| 2. | Memilih menu register | Register | Normal | Menampilkan halaman form register | Sesuai |
| 3. | Memilih Menu Dasbord | View Dasbord | Normal | Menampilkan halaman data properti | Sesuai |
| 4. | Memilih Create Apartemen (tambah apartemen) | Tombol tambah data  (*create apartemen)* | Normal | Menampilkan halaman input data apartemen | Sesuai |
|  | Tidak normal | Menampilkan alert | Sesuai |
| 5. | Memilih tombol hapus data | Tombol hapus data | Normal | Menghapus data | Sesuai |
|  | Tidak normal | Menampilkan alert | Sesuai |
| 6. | Memilih tombol edit data | Tombol edit data | Normal | Menampilkan *form* edit data | Sesuai |
|  | Tidak normal | Menampilkan alert | Sesuai |

**Lanjutan tabel 4.2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 7. | Memilih Menu Album | Normal | Menampilkan | Sesuai |
| 8. | Memilih Sub menu upload Photo | Normal | Menampilkan halaman upload photo. | Sesuai |
| 9. | Memilih Sub Menu List Album | Normal | Menampilkan List album photo-photo. | Sesuai |
| 10 | Memilih Tombol Hapus Album | Normal | Album dihapus | Sesuai |
| 11. | Memilih Tombol Edit | Normal | Menampilkan Form Edit Album | Sesuai |
| 12 | Melakukan Pencarian | Normal | Menampilkan data yang dicari | Sesuai |
| 13. | Melakukan Pengurutan (*Sorting* ) | Normal | Data diurutkan | Sesuai |
| 12. | Memilih keluar aplikasi (*logout*) | Normal | Keluar aplikasi, dan diarahkan ke halaman login. | Sesuai |

4.5 Analisis Hasil Pengujian

Analisa hasil pengujian dari sistem menujukan bahwa sistem beserta fungsi-fungsinya sudah berjalan dengan lancar, dengan rancangan dan spesifikasi awal sistem. Setelah tahap pengujian blackbox diatas terhadap semua fungsi pada Aplikasi mnajemen unit properti berbasis php mysql yang sudah dijalankan, dimana hasilnya sesuai dengan skenario pengujian yang direncanakan. berikut adalah kesimpulan yang di peroleh dari hasil pengujian tersebut:

1. Tombol-tombol pada setiap halaman menu sukses dan berhasil dijalankan pada skenario pengujian dan mendapatkan hasil pengujian pada tabel 4.2. Pada tabel tersebut terbukti tombol-tombol yang di uji secara normal dan tidak normal menghasilkan hasil yang sesuai.

2. Fungsi-fungsi pada aplikasi berjalan dengan baik seperti, melakukan penambahan data, menampilkan data, menghapus data, merubah data.

3. Proses dari pembuatan album, upload photo dan menampilkan *list* album berjalan dengan baik.

BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah melalui tahap-tahapan perancangan, implementasi dan pengujian aplikasi manajemen unit properti*,* dan berdasarkan teori dan pembahasan pada bab-bab sebelumnya yang telah dilakukan, maka dapat diambil kesimpulan sebagai berikut:

1. Telah dirancang dan dibangun aplikasi manajemen unit properti, yang dapat digunakan untuk pengelolaan unit-unit properti pada perusahan yang bergerak dibidang tersebut.
2. Aplikasi manajemen unit properti ini dapat mengelola data properti menjadi lebih terstruktur dan rapi, dan dapat diakses seperti penambahan data, penghapusan, dan perubahan, dimanapun kita berada.

5.2 Saran

Penelitian yang dilakukan tidak terlepas dari kekurangan dan kelemahan. Oleh karena itu, untuk kebaikan pengembangan aplikasi lebih lanjut, maka perlu diperhatikan hal, diantaranya :

1. Aplikasi ini dapat dikembangkan tidak hanya di aplikasi berbasis *web base*.
2. Aplikasi ini dapat dikembangkan dengan menambahkan fitur maps pada lokasi apartemen.
3. Aplikasi ini dikembangkan dengan menambahkan fitur chatting.

DAFTAR PUSTAKA

Anisya Sonita, Febrian Nurtaneo. *Analisis Perbandingan Algoritma Bubble Sort, Merge Sort, Dan Quick Sort Dalam Proses Pengurutan Kombinasi Angka Dan Huruf*. Jurnal Pseudocode, Volume II Nomor 2, September 2015, ISSN 2355 – 5920

<http://ejournal.unib.ac.id/index.php/pseudocode/article/view/887/767>

akses tanggal 23 Desember 2016

Journal of Tea Science Research. 2015, Vol. 5, No. 4, 1-11

<http://search.proquest.com/docview/1810128492/9DBA6F8E064F4964PQ/13?accountid=34643>

[Kammerer, Ferdinand Josef](http://search.proquest.com/indexinglinkhandler/sng/au/Kammerer,+Ferdinand+Josef/$N?accountid=34643); [Janka, Rolf](http://search.proquest.com/indexinglinkhandler/sng/au/Janka,+Rolf/$N?accountid=34643); [Kramer, Manuel](http://search.proquest.com/indexinglinkhandler/sng/au/Kramer,+Manuel/$N?accountid=34643); [Hammon, Matthias](http://search.proquest.com/indexinglinkhandler/sng/au/Hammon,+Matthias/$N?accountid=34643); [Uder](http://search.proquest.com/indexinglinkhandler/sng/au/Uder,+Michael/$N?accountid=34643); dkk. [**Journal of Digital Imaging**](http://search.proquest.com/pubidlinkhandler/sng/pubtitle/Journal+of+Digital+Imaging/$N/34218/PagePdf/1564093322/fulltextPDF/67320F711EDD475DPQ/6?accountid=34643)[[http://search.proquest.com/assets/r20161.9.1.366.866/core/spacer.gif](http://search.proquest.com/indexingvolumeissuelinkhandler/34218/Journal+of+Digital+Imaging/02014Y10Y01$23Oct+2014$3b++Vol.+27+$285$29/27/5?accountid=34643)27.5[http://search.proquest.com/assets/r20161.9.1.366.866/core/spacer.gif](http://search.proquest.com/indexingvolumeissuelinkhandler/34218/Journal+of+Digital+Imaging/02014Y10Y01$23Oct+2014$3b++Vol.+27+$285$29/27/5?accountid=34643)](http://search.proquest.com/indexingvolumeissuelinkhandler/34218/Journal+of+Digital+Imaging/02014Y10Y01$23Oct+2014$3b++Vol.+27+$285$29/27/5?accountid=34643)(Oct 2014):557-62.

<http://search.proquest.com/docview/1564093322/fulltextPDF/67320F711EDD475DPQ/6?accountid=34643>

Akses 29 Noverber 2016

Kadir, Abdul. 2009. *Membuat Aplikasi Web Dengan Php Dan Database Mysql.* Yogyakarta: Andi.

Nugroho, Bunafit. *Database Relational dengan MySQL*. Yogyakarta. Andi, 2005.

Nugraha, Teten. 2014. *Tutorial Dasar Laravel*. Published.

Nahado, Moh Royhan. 2014. *Bumbu Bumbu Membuat Website PHP, Apache & MySQL, Editor (Komodo Edit)*. Published.

HIMSYATECH-Jurnal Teknologi Informasi.

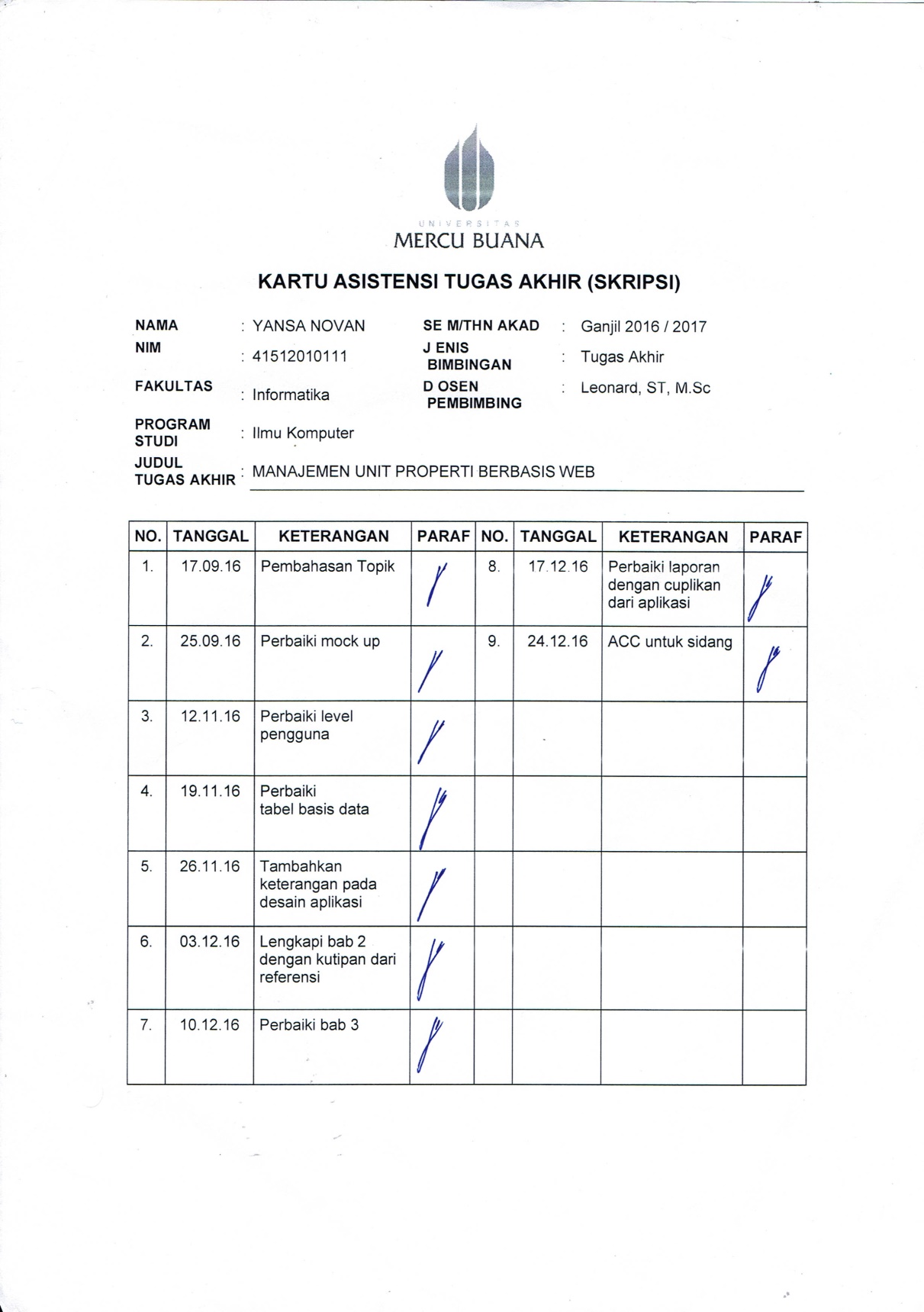
<http://www.ejournal.himsya.ac.id/index.php/HIMSYATECH/article/view/28/27>

Akses 3 Januari 2017

Sutanta, Edhy. 2011. Basis Data Dalam Tinjauan Konseptual. Yogyakarta : Andi Offset.

Rosa A.S, M.Shalahuddin. 2013. Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek. Bandung : Informatika Bandung.

**LAMPIRAN**



**CURICULUM VITAE**

****Name : Yansa Novan

Place And date of birth : Palangka Raya, 31 Januari 1994

Sex : Male

Nationality : Indonesia

Marital Status : Lajang

ID Number : 6271013101940002

Pasport Number : -

Email : yansanovan.yn@gmail.com

Religion : Kristen Protestan

Address : Jl. Keruing II No 05

Kec/Kel : Panarung/Pahandut

RT/RW : 003/001

Kota Palangka Raya

Provinsi Kalimantan Tengah

Phone Number/Mobile : 082213822826 & 081352878700

EDUCATION

Elementary School : SDN Menteng 7

Secondary School : SMP 6 Palangka Raya

Senior High School : SMKN 1 Palangka Raya

University : Universitas Mercu Buana

Degree Awarded : Strata 1 S.Kom

Faculty : Fakultas Ilmu Komputer (FASILKOM)

Title of Thesis : Aplikasi Manajemen Unit Properti Berbasis PHP MySQL

GPA :

SKILLS

Langauge : Indonesia & English Score TOEFL : ..............

ORGANIZATION EXPERIENCE

|  |  |  |
| --- | --- | --- |
| No of Years | Title | Name of Organization |
|  |  |  |
|  |  |  |

WORK EXPERIENCE

|  |  |  |
| --- | --- | --- |
| No of Years | Title | Name of Organization |
| 2015 | Data Collector | PT. 99.CO |
| 2016 | IT Support | PT. Avigra Rancang Prima  (Intership) |

ACTIVITIES

NATIONAL/INTERNATIONAL

|  |  |  |
| --- | --- | --- |
| No of Years | Title | Name of Organization |
|  |  |  |
|  |  |  |

ACHIEVMENT

|  |  |  |
| --- | --- | --- |
| No of Years | Title | Name of Organization |
|  |  |  |
|  |  |  |

HOBBIES

Sports : Futsal, Badminton

Various : Music