# SKRIPSI

**APLIKASI ADMINISTRASI DAN PELAPORAN KOPERASI PADA KANTOR DINAS PENDIDIKAN KABUPATEN ENREKANG**



**ZULKARNAIN RAHMAN**

**NIM. 216280006**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS MUHAMMADIYAH PAREPARE**

**2021**

# HALAMAN PERSETUJUAN

**APLIKASI ADMINISTRASI DAN PELAPORAN KOPERASI PADA KANTOR DINAS PENDIDIKAN KABUPATEN ENREKANG**

**ZULKARNAIN RAHMAN**

**NIM. 26280006**

Telah diperiksa dan disetujui untuk mengikuti Seminar Hasil

Parepare, 24 Mei 2021

Komisi Pembimbing

Pembimbing I

Pembimbing I

Mengetahui

Ketua Program Studi

# PRAKATA

Assalamu Alaikum Warahmatullahi Wabarakatuh

Puji syukur penulis panjatkan kehadirat Allah SWT, karena berkat Rahmat-Nya lah penulis dapat menyelesaikan Penyusunan Proposal dengan judul **“APLIKASI ADMINISTRASI DAN PELAPORAN KOPERASI PADA KANTOR DINAS PENDIDIKAN KABUPATEN ENREKANG”**.

Penulis mengucapkan terima kasih kepada sleuruh pihak yang telah membantu memberikan pengarahan kepada penulis dalam pembuatan proposal ini, sehingga proposal ini dapat selesai.

Penulis menyadari bahwa proposal ini masih belum sempurna, Oleh karena itu, penulis sangat mengharapkan kritik ataupun saran yang sifatnya membangun demi terwujudnya suatu hasil proposal yang sempurna, Semoga proposal ini dapat bermanfaat bagi siapapun yang membacanya.

Billahi Fii Sabilil Haq, Fastabiqul Khaerat

Wassalamu Alaikum Warahmatullahi Wabarakatuh

Parepare, Mei 2021

Penulis

**Zulkarnain Rahman**

NIM. 216280006

# DAFTAR ISI

[SAMPUL i](#_Toc72832843)

[HALAMAN PERSETUJUAN ii](#_Toc72832844)

[PRAKATA iii](#_Toc72832845)

[DAFTAR ISI iv](#_Toc72832846)

[DAFTAR TABEL vii](#_Toc72832847)

[DAFTAR GAMBAR viii](#_Toc72832848)

[BAB I PENDAHULUAN 1](#_Toc72832849)

[A. Latar Belakang Masalah 1](#_Toc72832850)

[B. Rumusan Masalah 2](#_Toc72832851)

[C. Tujuan Penelitian 2](#_Toc72832852)

[D. Batasan Masalah 3](#_Toc72832853)

[E. Manfaat Penelitian 4](#_Toc72832854)

[F. Definisi Dan Istilah 4](#_Toc72832855)

[G. Sistematika Penulisan 5](#_Toc72832856)

[BAB II TINJAUAN PUSTAKA 7](#_Toc72832857)

[A. Penelitian Terdahulu 7](#_Toc72832858)

[B. Pengertian Komputer 9](#_Toc72832859)

[C. Data 10](#_Toc72832860)

[D. Algoritma 10](#_Toc72832861)

[E. Program 11](#_Toc72832862)

[F. Pengertian Aplikasi 11](#_Toc72832863)

[G. Pengertian Administrasi 13](#_Toc72832864)

[H. Pengertian Pelaporan 13](#_Toc72832865)

[*I.* *HTML (Hyper Text Markup Language)* 14](#_Toc72832866)

[*J.* *PHP (Hypertext Preprocessor)* 15](#_Toc72832867)

[*K.* *DATABASE* 16](#_Toc72832868)

[*L.* *PHPMYADMIN* 18](#_Toc72832869)

[*M.* *MYSQL* 21](#_Toc72832870)

[N. Metode Pengujian 30](#_Toc72832871)

[*O.* *UML (Unified Modelling Language)* 31](#_Toc72832872)

[*P.* *Kerangka Pikir* 41](#_Toc72832873)

[BAB III METODE PENELITIAN 42](#_Toc72832874)

[A. Jenis Penelitian 42](#_Toc72832875)

[B. Lokasi Dan Waktu 42](#_Toc72832876)

[C. Alat Dan Bahan 43](#_Toc72832877)

[D. Rancangan Sistem 44](#_Toc72832878)

[E. Diagram Alir 47](#_Toc72832879)

[BAB IV HASIL DAN PEMBAHASAN 54](#_Toc72832880)

[A. Metode Pengujian 54](#_Toc72832881)

[B. Pengujian *White Box* 55](#_Toc72832882)

[*C.* Pengujian *Black Box* 62](#_Toc72832883)

[BAB V PENUTUP 76](#_Toc72832884)

[A. Kesimpulan 76](#_Toc72832885)

[B. Saran 76](#_Toc72832886)

[DAFTAR PUSTAKA 77](#_Toc72832887)

[LAMPIRAN 79](#_Toc72832888)

# DAFTAR TABEL

[Tabel 3.1 Spesifikasi Laptop 49](#_Toc72832415)

[Tabel 3.2 Perangkat Lunak 49](#_Toc72832416)

[Tabel 4.1 *White Box Login* 61](#_Toc72832417)

[Tabel 4.2 Grafik Matriks L*ogin* 62](#_Toc72832418)

[Tabel 4.3 White Box Admin 63](#_Toc72832419)

[Tabel 4.4 Grafik Matriks Admin 65](#_Toc72832420)

[Tabel 4.5 White Box User 66](#_Toc72832421)

[Tabel 4.6 Grafik Matriks L*ogin* 67](#_Toc72832422)

[Tabel 4.7 Black Box Login 68](#_Toc72832423)

[Tabel 4.8 *Black Box* Admin (Halaman *User*) 70](#_Toc72832424)

[Tabel 4.9 *Black* *Box* *Form* Admin (Halaman Simpanan Wajib) 72](#_Toc72832425)

[Tabel 4.10 *Black* *Box* *Form* Admin (Halaman Simpanan Wajib) 74](#_Toc72832426)

[Tabel 4.11 *Black* *Box* *Admin* ( Halaman Simpanan Sukarela ) 76](#_Toc72832427)

[Tabel 4.12 *Black Box* Admin ( Halaman Pinjaman ) 78](#_Toc72832428)

[Tabel 4.13 Black Box Admin ( Halaman Pengembalian ) 80](#_Toc72832429)

# DAFTAR GAMBAR

[Gambar 2.1 Kerangka Pikir 47](#_Toc72832441)

[Gambar 3.1 *Use Case Diagram* 50](#_Toc72832442)

[Gambar 3.2 *Activity Diagram Login* 53](#_Toc72832443)

[Gambar 3.3 *Activity Diagram Admin* 54](#_Toc72832444)

[Gambar 3.4 *Activity Diagram User* 56](#_Toc72832445)

[Gambar 3.5 *Sequence Diagram Login* 57](#_Toc72832446)

[Gambar 3.6 *Sequence Diagram Admin* 58](#_Toc72832447)

[Gambar 3.7 Sequence Diagram User 59](#_Toc72832448)

# BAB I

**PENDAHULUAN**

## Latar Belakang Masalah

Koperasi adalah badan hukum yang didirikan oleh orang perseorangan atau badan hukum koperasi, dengan pemisahan kekayaan para anggotanya sebagai modal untuk menjalankan usaha yang memenuhi aspirasi dan kebutuhan bersama di bidang ekonomi, sosial, dan budaya sesuai nilai dan prinsip koperasi.

Koperasi berperan serta secara aktif dalam upaya meningkatkan kualitas kehidupan manusia dan masyarakat serta membangun dan mengembangkan potensi dan kemampuan ekonomi anggota pada khususnya dan masyarakat pada umumnya untuk meningkatkan kesejahteraan ekonomi dan sosialnya.

Koperasi Dinas Pendidikan merupakan koperasi yang begerak di bidang usaha simpan pinjam dan pertokoan. Namun didalam pengelolaan administrasi dan pelaporan masih menggunakan sistem yang sangat sederhana. Artinya administrasi dan laporan koperasi masih menggunakan pembukuan. Proses ini masih melakukan pencatatan satu persatu, sehingga menimbulkan masalah-masalah yaitu adanya kesalahan-kesalahan yang terjadi baik dari segi keakuratan data, efesiensi waktu serta manipulasi data yang kemungkinan bisa terjadi. Seiring bertambahnya nasabah pada koperasi Dinas Pendidikan yang memerlukan proses administrasi yang tepat, teratur dan efesien, maka diperlukan adanya sistem administrasi yang terkomputerisasi yang dapat mengolah data pada koperasi.

Untuk itu penulis mengusulkan rancangan aplikasi yang berbasis komputer yang nantinya dapat menangani segala permasalahan yang ada dan diharapkan dapat membantu kelancaran tugas dalam hal pelayanan informasi dan penemuan kembali data pembayaran kredit yang dibutuhkan dengan lebih mudah dan cepat dibandingkan cara yang sebelumnya.

Atas beberapa dasar pemikiran tersebut maka penulis merumuskan judul: **“Aplikasi Administrasi Dan Pelaporan Koperasi Pada Dinas Pendidikan Kabupaten Enrekang”**.

## Rumusan Masalah

Berdasarkan latar belakang diatas maka dapat dirumuskan masalah sebagai berikut :

1. Bagaimana pengelolaan administrasi dan pelaporan koperasi ?
2. Bagaimana membuat program untuk mengelola data dan laporan koperasi?

## Tujuan Penelitian

1. Tujuan yang hendak dicapai dalam penelitian ini adalah :
2. Untuk mengetahui pengelolaan administrasi dan pelaporan koperasi.
3. Untuk membuat aplikasi program yang mengelola data dan laporan koperasi.
4. Manfaat penelitianya itu sebagai berikut :
5. Manfaat Bagi Penulis

Menambah wawasan, penguasaan dan penerapan teknologi komputer dalam perancangan suatu program aplikasi administrasi koperasi bagi penulis khususnya dan bagi para mahasiswa lain pada umumnya.

1. Manfaat bagi dunia akademik

Dapat manjadi bahan rujukan atau bahan studi bagi penelitian lain dibidang yang sama.

1. Manfaat Bagi Koperasi

Memberikan konstribusi pemikiran dalam pengambilan keputusan, pemberian kerjasama serta meningkatkan efesiensi kerja karyawan.

## Batasan Masalah

Agar pembahasan menjadi lebih terarah pada pokok permasalahan maka penulis membatasi masalah yaitu pada pendataan anggota koperasi, data simpan pinjam dan data pembelian dan penjualan barang koperasi dengan menggunakan bahasa pemrograman *Javascript*, *PHP* dan basis data menggunakan *MYSQL*.

## Manfaat Penelitian

Adapun manfaat penelitian sebagai berikut :

* + 1. Aspek Akademis

Sebagai langkah awal bagi peneliti untuk dapat mengembangkan teori-teori yang didapat selama ini di bangku perkuliahan khususnya dalam mengenai aplikasi administrasi pelaporan berbasis *web.*

1. Aspek Praktis
2. Sebagai kemudahan untuk administrasi pelaporan yang berlaku di Dinas Pendidikan Kabupaten Enrekang
3. Bagi peneliti diharapkan dapat memberikan pengalaman dan menerapkan ilmu pengetahuan yang diterima selama mengikuti perkuliahan
4. Sebagai tugas akhir untuk mendapatkan gelar sarjana strata 1 (S1) pada Fakultas Teknik Informatika

## Definisi Dan Istilah

Definisi yang akan selalu digunakan oleh penulis adalah “administrasi koperasi” dan “pelaporan” yang mana “administrasi koperasi” merupakan sebuah administrasi berkaitan dengan koperasi yang nantinya akan dijelaskan oleh penulis pada bab 2 (dua) dan “pelaporan” merupakan data-data administrasi koperasi yang nantinya akan menjadi laporan. Diharapkan ini dapat sedikit menjelaskan definisi dan istilah yang akan selalu digunakan agar tidak menimbulkan salah tafsir, dan pengertian kepada pembaca.

## Sistematika Penulisan

Demi terwujudnya suatu hasil yang baik dalam penyusunan proposal ini penulis menggunakan pembahasan yang sesuai dengan ketentuan yang diberikan, sistematika penulisan meliputi antara lain :

**BAB I : PENDAHULUAN**

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

**BAB II: TINJAUAN PUSTAKA**

Dalam bab ini penulis menjelaskan teori-teori yang bersangkutan dengan masalah yang akan dibahas. Teori-teori tersebut yaitu membahas definisi Implementasi, *Web*, *Pemrograman PHP* (Personal *Home Page*), *Javascript* dan *Database MySQL*.

**BAB III: METODOLOGI PENELITIAN**

Dalam bab ini berisi tentang perancangan sistem yang meliputi lokasi dan waktu penelitian, jenis penelitian, metode pengumpulan, alat dan bahan penelitian, tahapan penelitian, metode pengujian, jenis data serta desain sistem.

**BAB IV: HASIL DAN PEMBAHASAN**

Bab ini berisi jenis perancangan, materi perancangan, alat perancangan, ,rancangan aplikasi implementasi dan pengujian sistem yang telah dibuat..

**BAB V : PENUTUP**

Bab ini berisikan tentang kesimpulan dan saran sebagai tindak   
lanjut yang diperlukan untuk melakukan perbaikan di masa   
mendatang.

# BAB II

**LANDASAN TEORI**

## Penelitian Terdahulu

1. **“Perancangan Sistem Simpan Pinjam Pada Koperasi PT.Muliapack”** oleh (Apriliani, 2018) .Koperasi simpan pinjam PT.Muliapack merupakan salah satu organisasi yang seharusnya menjadi wadah untuk pertumbuhan ekonomi atau kesejahteraan karyawan . ini masih memiliki beberapa kekurangan sehingga kinerjanya belum optimal. Sistem informasi yang berjalan disini masih belum efektif karena masih di kombinasikan dengan proses manual untuk proses pinjamannya, hal tersebut memakan waktu lama untuk setiap transakasinya, namun seiring dengan pesatnya perkembangan teknologi, maka sudah seharusnya masalah perkoperasian tersebut diberi sentuhan teknologi sebagai solusi contohnya seperti dibangun sistem informasi untuk memudahkan setiap operasional yang ada. Dengan permasalahan yang ada, maka peneliti mengusulkan sistem koperasi berbasis *web*, menggunakan metode analisis PIECES. Perancangan sistem menggunakan metode analisa berorientasi objek dengan *Unified Modelling Language* (UML), bahasa pemrograman menggunakan *PHP* dengan *MySQL* sebagai *database* nya. Sistem yang diusulkan dapat memudahkan user mencari informasi dan menyusun laporan koperasi dengan baik, format laporan lebih rapi, keamanan data koperasi terjamin karena adanya menu login. Perbedaan penelitian ini penelitian penulis adalah *user* yang terdaftar pada koperasi dapat mengakses untuk mengetahui penyimpanan serta pinjaman yang telah dilakukan sebelumnya.
2. **“Peneliti Meli Agustina, Perancangan Sistem Informasi Simpan Pinjam Di Koperasi Karyawan Mt Haryono Bebasis Java”** oleh (Agustina, Rismawati, & Acep, 2020)**,** Penelitian ini bertujuan untuk membuat suatu aplikasi simpan pinjam koperasi karyawan, serta mendeskripsikan hasil uji coba sistem aplikasi. Metode penelitian yang digunakan adalah metode *Grounded* *Research.* Peneliti mengumpulkan data dan observasi di koperasi karyawan Indomobil dengan metode wawancara langsung kepada penanggung jawab dan petugas koperasi. Peneliti menemukan berbagai masalah yang dihadapi di koperasi tersebut yaitu kendala sistem yang masih sangat manual sehingga pekerjaan tidak efekti dan efisien. Hasil penelitian dapat disimpulkan dengan adanya aplikasi yang sudah terkomputerisasi, transaksi di koperasi semakin mudah dan cepat. Perbedaan penelitian ini dengan penelitian penulis adalah penelitian sebelumnya fokus pada aplikasi *desktop* sedangkan aplikasi yang dibangun penulis merupakan aplikasi *web* yang mana bisa diakses secara *online.*
3. **“Perancangan Aplikasi Sistem Informasi Pinjaman Keuangan Anggota Koperasi Syariah Usaha Mandiri”** oleh (Suciyanto, 2014)**,** Dengan perkembangan teknologi yang semakin maju, dimana semua hal mengandalkan proses dengan menggunakan teknologi yang semakin canggih serta volume pekerjaan yang semakin tinggi. Maka sangat perlu dibuat sebuah program atau sistem yang dapat mendukung proses pekerjaan dengan cepat dan tepat.Teknologi informasi dalam bidang pinjaman keuangan digunakan untuk menyimpan serta mengolah data, dan mempermudah pihak administrasi dalam mengoreksi data nasabah yang sudah ada maupun yang baru masuk menjadi nasabah. Oleh karena itu dibutuhkan suatu sistem yang dapat menyimpan dan mengolah data-data tersebut bagi pihak yang memerlukan, sehingga kegiatan usaha ini dapat berkembang lebih baik lagi. Sistem Informasi Keuangan ini merupakan suatu sistem aplikasi terkomputerisasi yang mampu melakukan pengolahan data transaksi secara optimal. Sehingga apabila membutuhkan laporan ataupun data nasabah dapat terealisasikan secara cepat dan akurat. Perbedaan penelitian ini dengan penelitian penulis adalah penelitian sebelumnya fokus pada aplikasi *desktop* sedangkan aplikasi yang dibangun penulis merupakan aplikasi *web* yang mana bisa diakses secara *online.*

## Pengertian Komputer

Istilah komputer sudah tidak asing lagi bagi masyarakat terutama masyarakat dalam bidang pendidikan. Selama ini komputer digunakan sebagai alat untuk membantu manusia memudahkan berbagai pekerjaannya. komputer adalah sistem elektronik memanipulasi data yang cepat dan tepat serta dirancang dan diorganisaikan agar menerima dan menyimpan data input secara otomatis, memproses, dan menghasilkan output melalui langkah – langkah instruksi program yang tersimpan di memori.

## Koperasi

Menurut Pasal 1 angka 1 Undang-Undang No .17 Tahun 2012 tentang Perkoperasian menyatakan bahwa: “Koperasi adalah badan hukum yang didirikan oleh orang perseorangan atau badan hukum koperasi, dengan pemisahan kekayaan para anggotanya sebagai modal untuk menjalankan usaha yang memenuhi aspirasi dan kebutuhan bersama di bidang ekonomi, sosial, dan budaya sesuai nilai dan prinsip koperasi” .

Undang No. 25 Tahun 1992 Pasal 1 angka 1 menyatakan bahwa: “Koperasi adalah badan usaha yang beranggotakan orang-seorang atau badan hukum koperasi dengan melandaskan kegiatannya berdasarkan prinsip koperasi sekaligus sebagai gerakan ekonomi rakyat yang berdasar atas asa kekeluargaan.” .

Dari pengertian diatas, perbedaan UU No 25 Tahun 1992 dan UU . No 17 Tahun 2012 tentang perkoperasian adalah sebagai berikut :

1. Dalam UU No 25 Tahun 1992 menjabarkan pengertian koperasi sebagai badan usaha dan badan hukum yang beranggotakan orang-perseorangan. Sedangkan UU No 17 Tahun 2012 menjabarkan pengertian koperasi sebagai badan hukum yang didirikan oleh orang-perseorangan. Perbedaan tersebut, terlihat dari pemilihan kata yang digunakan untuk mendeskripsikan koperasi yakni badan usaha dan badan hukum yang jelas memiliki makna yang berbeda. Di ana badan usaha merupakan badan yang menguraikan falsafah, prinsip, dan landasan-landasan yang digunakan sebagai acuan dalam melakukan usaha, sedangkan badan hukum merupakan bagian dari badan usaha yang bersifat lebih mengikat dan ada sanksi yang tegas terhadap setiap pelanggaran. Dalam badan hukum juga terdapat persetujuan pemerintah atas penyelenggaraan suatu usaha.
2. Dilihat dari segi konsistensi kata (diksi kalimat / pilihan kata) dalam pengertian koperasi menurut UU No. 25 Tahun 1992 , terjadi ketidak konsistensi kata, di mana dalam UU No 25 Tahun 1992 tidak hanya menguraikan pengertian koperasi sebagai badan usaha tetapi pula sebagai badan hukum. Sedangkan UU No 17 Tahun 2012 terjadi hal yang berlawanan yakni: adanya konsistensi kata yang digunakan untuk mendeskripsikan pengertian koperasi yakni penggunaan kata badan hukum.

## Data

Data adalah nilai yang dibutuhkan dalam pembuatan aplikasi.Dalam pembuatan aplikasi programmer dapat menentukan jenis data sesuai dengan kebutuhan yang diinginkan. Pemilhan jenis data yang tepat akan berguna untuk menghemat memori, meningkatkan kecepatan proses, ketelitian dalam perhitungan dan lain-lain. Ada beberapa tipe data yang sering digunakan dalam pembuatan aplikasi, antara lain :

1. String adalah tipe data untuk teks ( huruf, angka, dan tanda baca )
2. Integer adalah tipe data untuk bilangan bulat
3. Single adalah tipe data untuk angka pecahan
4. Currency adalah tipe data untuk angka mata uang
5. Date adalah tipe data untuk tanggal dan jam
6. Boolean adalah tipe data yang bernilai true atau false .

## Algoritma

Algoritma adalah urutan langkah-langkah logis pengambilan keputusan untuk memecahkan masalah, alur pemikiran dalam menyelesaikan suatu pekerjaan yang dituangkan secara tertulis yang artinya dapat berupa kalimat, gambar, atau table tertentu.Dalam bidang komputer.Algoritma adalah teknik penyusunan langkah penyelesain masalah dalam bentuk kalimat singkat dengan jumlah kata terbatas, namun tesusun secara logis dan sistematis.

## Program

Program adalah kumpulan instruksi atau perintah yangdisusun sedemikian rupa sehingga mempunyai urutan nalar yang tepat untuk menyelesaikan suatu persoal.pemrograman berarti memberikan instruksi kepada komputer agar dapat bekerja seperti yang kita kehendaki. Instruksi (statement) yang dimaksud adaalh syntax (cara penulisan) sesuai dengan bahasa pemrograman yang digunakan yang mempunyai komponen–komponen Input, Output, Proses, Pecabangan dan Perulangan.

## Pengertian Aplikasi

Aplikasi adalah penerapan dari rancang system untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu dan selain itu merupakan program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna.

Program atau perangkat lunak yang dijalankan dikomputer yang dimana digunakan oleh siapa saja untuk membantu pekerjaan yang dengan mudah diinstal.Contoh utama perangkat lunak aplikasi adalah pengolah kata, lembar kerja, dan pemutar media.

Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket (*application suite*).

Aplikasi dapat dikelompokkan menjadi dua macam yaitu :

1. Program Aplikasi merupakan program yang langsung dibuat oleh seorang *programmer* yang disesuaikan dengan kebutuhan seseorang ataupun kebutuhan perusahaan biasanya menggunakan bantuan suatu bahasa pemrograman, misalnya visual basic, php, java ataupun bahasa pemrograman yang mendukung.
2. Program Paket

Program paket merupakan program khusus dalam paket-paket tertentu yang dibuat oleh *software* atau pun langsung bawaan dari system operasi tersebut seperti Program pengolah kata contohnya Microsoft Word, Open office.org, Writter, ChiWritter, Word Perfect, WordStar, K Writter, Ampiro, dan lain-lain.

1. Program pengolah angka contohnya Microsoft Excel, Open Office.org Calc, Quattro Pro, Lotus 123, dan lain-lain.

Program presentasicontohnya Microsoft Power point, Open Office.org Impres, Magic Point, Corel Presentation, Apple Work, dan lain-lain.

1. Program desain grafis contohnya adalah Adobe Photoshop, Corel Draw, Free Hand, Auto Cad, dan lain-lain. Program *browser* contohnya adalah Internet Explorer, Mozilla Firefox, Opera, Netscape Communcator.
2. Program Database contohnya adalah Microsoft Access, Open Office.org Base, Visual Foxpro, Fox Base, Dbase I-IV, dan lain-lain
3. Program animasi contohnya adalah Macromedia Flash, Swish, dll.
4. Program multimedia contohnya Windows Media Player, Winamp, Cyberlink, RealPlayer, DVD player. (Kamus Besar Bahasa Indonesia,1998).

## Pengertian Administrasi

Administrasi adalah keseluruhan proses kerja sama antara 2 orang atau lebih yang didasarkan atas rasionalitas tertentu untuk mencapai tujuan yang telah ditentukan sebelumnya .Berdasarkan uraian dan definisi tersebut maka dapat diambil kesimpulan bahwa administrasi adalah seluruh kegiatan yang dilakukan melalui kerjasama dalam suatu organisasi berdasarkan rencana yang telah ditetapkan untuk mencapai tujuan.(Sondang P. Siagian, 1994).

## Pengertian Pelaporan

Pelaporan merupakan suatu kegiatan yang dilakukan bawahan untuk menyampaikan hal-hal yang berhubungan dengan hasil pekerjaan yang telah dilakukan selama satu periode tertentu. Pelaporan dilakukan kepada atasan kepada siapa bawahan tersebut bertanggung jawab .Pelaporan adalah aktivitas yang berlawanan arah dari pengawasan, Jika pengawasan dilakukan oleh pihak atasan untuk mengetahui semua hal yang menyangkut pelaksanaan kerja bawahan, maka pelaporan merupakan jawaban dari kegiatan pengawasan tersebut.

Laporan dibuat oleh semua personal yang mendapat tugas dari atasan. Laporan bukan merupakan monopoli para atasan saja. Karena atasan harus membuat laporan kepada atasannya lagi.Laporan yang disampaikan kepada atasan tidak harus berupa uraian lengkap seperti memorandum akhir jabatan, atau tidak juga seperti laporan penelitian yang wujudnya tebal dengan sistematika baku, tetapi dapat disusun mulai dari bentuk yang paling sederhana sampai yang paling lengkap.

## *HTML (Hyper Text Markup Language)*

*HTML* (*Hyper Text Markup Language*) adalah bahasa yang digunakan untuk menulis halaman *WEB* dan menampilkan berbagai informasi di dalam sebuah *browser* internet (Jubilee, 2011). *HTML* ini merupakan pengembangan dari standar pemformatan dokumen teks yaitu *Standard Generalized Markup Language* (*SGML*). *HTML* saat ini merupakan standar Internet yang didefinisikan dan dikendalikan penggunaannya oleh *World Wide WEB Consortium* (*W3C*).

[*HTML*](http://www.blogofnurjaya.com/2010/01/definisi-dan-sejarah-html.html) dibuat oleh Tim Berners-Lee ketika masih bekerja untuk CERN (*Conseil Europeene Pour La Recherche Nuclaire*) dan dipopulerkan pertama kali oleh *Browser Mosaic* pada tahun 1989 sebagai cara sederhana namun efektif untuk mengkodekan dokumen elektronik. Sebetulnya, tujuan awal dari *WEB browser* adalah untuk melayani sebagai pembaca untuk dokumen berformat *HTML*. Namun, dua dekade kemudian *browser* sendiri telah menjadi sebuah portal ke dunia media online. Itulah sebabnya versi 2 *HTML* yang selanjutnya tidak hanya sebagai revisi *HTML*, tetapi juga sebagai standar yang *komprehensif* untuk bagaimana halaman *WEB* bekerja.

## *PHP (Hypertext Preprocessor)*

Menurut(Arief, 2011) *PHP* adalah sebuah peranti yang biasa dipakai untuk membuat aplikasi *WEB*. Aplikasi *WEB* adalah aplikasi yang dijalankan melalui *browser*.

*PHP* adalah bahasa *server*-side –*script*ing yang menyatu dengan *HTML* untuk membuat halaman *WEB* yang dinamis. Karena *PHP* merupakan *server*-side-*script*ing maka sintaks dan perintah-perintah *PHP* akan dieksekusi di *server* kemudian hasilnya akan dikirimkan ke *browser* dengan format *HTML*. Dengan demikian kode program yang ditulis dalam *PHP* tidak akan terlihat oleh *user* sehingga keamanan halaman *WEB* lebih terjamin. *PHP* dirancang untuk membuat halaman *WEB* yang dinamis.

Kelebihan-kelebihan *PHP* adalah sebagai berikut:

* + - 1. *Script*(kode program) terintegrasi dengan file *HTML*, sehingga *developer* (pengembang) bisa berkonsentrasi *Langsung* pada penampilan *WEB* nya.
      2. Tidak ada proses *compiling* dan *linking.*
      3. Berorientasi obyek (*object oriented*).
      4. *Sintaksis* program mudah dipelajari.
      5. Integrasi yang sangat luas ke berbagai *server Database*.

## *DATABASE*

*Database* (Nugroho, 2008) adalah suatu pengelolaan data dalam pengingat eksternal (misalnya *harddisk*) yang memungkinkan seseorang dengan mudah menyimpan data dan sekaligus menggunakannya ketika memerlukannya. *Software* yang digunakan untuk mengelola *database* antara lain yaitu *MySQL*, *Access*, dan *Oracle*.

Basis data (*Database*) adalah kumpu*lan* data yang disimpan secara sistematis di dalam komputer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi. Pendefinisian basis data meliputi spesifikasi berupa tipe data, struktur data dan juga batasan-batasan pada data yang kemudian disimpan.

Basis data merupakan aspek yang sangat penting dalam sistem informasi karena berfungsi sebagai gudang penyimpanan data yang akan diolah lebih *lan*jut. Basis data menjadi penting karena dapat mengorganisasi data, menghindari duplikasi data, menghindari hubungan antar data yang tidak jelas dan juga update yang rumit.

Proses memasukkan dan mengambil data ke dan dari media penyimpanan data memerlukan perangkat lunak yang disebut dengan sistem manajemen basis data (*Database management system*). *DBMS* merupakan sistem perangkat lunak yang memungkinkan pengguna basis data (*Database user*) untuk memelihara, mengontrol dan mengakses data secara praktis dan efisien.

Menururut jenisnya Basis data (*Database*) dapat dibagi menjadi :

1. *Basis data flat-file*.

Basis data ini ideal untuk data berukuran kecil dan dapat dirubah dengan mudah. Pada dasarnya, *Basis data flat-file*tersusun dari sekumpu*LAN* string dalam satu atau lebih file yang dapat diurai untuk mendapatkan informasi yang disimpan. *Basis data flat-file* cocok untuk menyimpan daftar atau data yang sederhana dan dalam jumlah kecil. *Basis data flat-file* akan menjadi sangat rumit apabila digunakan untuk menyimpan data dengan struktur kompleks walaupun dimungkinkan pula untuk itu.

Beberapa kendala dalam menggunakan basis data jenis ini adalah rentan pada korupsi data karena tidak adanya penguncian yang melekat ketika data digunakan atau dimodifikasi dan juga adanya duplikasi data yang mungkin sulit dihindari. Salah satu tipe *Basis data flat-file* adalah file CSV yang menggunakan pemisah koma untuk setiap nilainya.

1. *Basis Data Relasional*

Basis data ini mempunyai struktur yang lebih logis terkait cara penyimpanan. Kata "*relasional*" berasal dari kenyataan bahwa tabel-tabel yang ada di basis data *relasional* dihubungkan satu dengan lainnya. Basis data *relasional* menggunakan sekumpu*LAN* tabel dua dimensi yang masing-masing tabel tersusun atas baris (*tupel*) dan kolom (atribut).

Untuk membuat hubungan antara dua atau lebih tabel, digunakan key (atribut kunci) yaitu *primary key* di salah satu tabel dan foreign key di tabel yang lain. Saat ini, basis data *relasional* menjadi pilihan utama karena keunggu*LAN*nya. Program aplikasi untuk mengakses basis data *relasional* menjadi lebih mudah dibuat dan dikembangkan dibandingkan dengan penggunaan *Basis data flat-file*.

Beberapa kekurangan yang mungkin dirasakan di basis data jenis ini adalah implementasi yang lebih sulit untuk data dalam jumlah besar dengan tingkat kompleksitasnya yang tinggi. Selain itu, proses pencarian informasi juga menjadi lebih lambat karena perlu menghubungkan tabel-tabel terlebih dahulu apabila datanya tersebar di beberapa tabel.Namun, terlepas dari beberapa kekurangannya, basis data *relasional* telah digunakan secara luas saat ini.

## *PHPMYADMIN*

*PHPMYADMIN* adalah perangkat lunak bebas yang ditulis dalam bahasa pemrograman *PHP* yang digunakan untuk menangani administrasi *MySQL* melalui *website* Jejaring Jagat Jembar (*World* *Wide* *Web*) . *phpMyAdmin* mendukung berbagai operasi *MySQL*, diantaranya (mengelola basis data, tabel-tabel, bidang (*fields*), relasi (*relations*), indeks, pengguna (*users*), perizinan (*permissions*), dan lain-lain).

Pada dasarnya, mengelola basis data dengan *MySQL* harus dilakukan dengan cara mengetikkan baris-baris perintah yang sesuai (*command* *line*) untuk setiap maksud tertentu. Jika seseorang ingin membuat basis data (*database*), ketikkan baris perintah yang sesuai untuk membuat basis data. Jika seseorang menghapus tabel, ketikkan baris perintah yang sesuai untuk menghapus tabel. Hal tersebut tentu saja sangat menyulitkan karena seseorang harus hafal dan mengetikkan perintahnya satu per satu.

Saat ini banyak sekali perangkat lunak yang dapat dimanfaatkan untuk mengelola basis data dalam *MySQL*, salah satunya adalah *phpMyAdmin*. Dengan *phpMyAdmin*, seseorang dapat membuat *database*, membuat tabel, mengisi data, dan lain-lain dengan mudah, tanpa harus menghafal baris perintahnya.

*phpMyAdmin* merupakan bagian untuk mengelola basis data *MySQL* yang ada di komputer. Untuk membukanya, buka *browser* lalu ketikkan alamat *http://localhost/phpmyadmin,* maka akan muncul halaman *phpMyAdmin*. Di situ nantinya seseorang bisa membuat (*create*) basis data baru, dan mengelolanya.

Pengembangan *phpMyAdmin* dimulai pada tahun 1998 oleh *Tobias* *Ratschiller* seorang konsultan *IT*. *Ratschiller* mengerjakan sebuah program bernama *MySQL*-*Webadmin* dengan bebas, yang merupakan produk dari *Petrus* *Kuppelwieser*, yang telah berhenti mengembangkannnya pada saat itu. *Ratschiller* menulis kode baru untuk *phpmyadmin*, dan ditingkatkan pada konsep dari proyek *Kuppelwiesser*. *Ratschiller* meninggalkan proyek *phpMyAdmin* pada tahun 2001. Sekarang, Sebuah tim dari delapan pengembang yang dipimpin oleh *Oliver* *Muller* meneruskan pengembangan *phpMyAdmin* di *SourceForge*.*net*.

Beberapa fitur dalam phpMyAdmin:

1. Antarmuka berbasis web
2. Dukungan banyak fitur MySQL :
3. Menelusuri dan drop basisdata (database), tabel, pandangan (view), bidang (fields) dan indeks.
4. Membuat, menyalin, drop, dan mengubah nama basis data, tabel, kolom dan indeks.
5. Pemeliharaan server, basis data dan tabel, dengan server konfigurasi.
6. Melaksanakan, mengedit dan penunjuk pernyataan-SQL, bahkan batch-queries
7. Mengelola pengguna MySQL dan hak istimewa.
8. Mengelola prosedur penyimpanan.
9. Impor data dari CSV dan SQL.
10. Ekspor data ke berbagai format: CSV, SQL, XML, PDF, ISO / IEC 26300 - OpenDocument Text dan Spreadsheet, Word, Excel, LATEKS dan lain-lain.
11. Membuat grafik PDF dari tampilan basis data anda.
12. Membuat kompleks query menggunakan Query-by-example. (QBE).
13. Pencarian global dalam basis data.
14. Transformasi data disimpan ke dalam format yang menggunakan satu set fungsi yang telah ditetapkan, seperti menampilkan data *blob-data* atau *download-link*.

## *MYSQL*

*MySQL* menurut (Arief, 2011) adalah salah satu jenis *Databaseserver* yang menggunakan *SQL* sebagai bahasa dasar untuk mengakses *Database*nya. Dengan menggunakan *scriptPHP* dan *PERLSoftwareDatabase* ini dapat berfungsi atau berjalan pada semua platform sistem operasi yang biasa digunakan (*Windows, Linux, OS/2, berbagai varian Unix*).

Secara garis besar *MySQL* digunakan untuk membuat dan mengelola suatu *Database* secara terstruktur dan otomatis menggunakan suatu Bahasa khusus. Namun lebih jelasnya lagi *MySQL* ini memberikan kemudahan bagi para pengguna yang ingin mengelola suatu data yang berisi informasi secara String (*text based*) dan dapat diakses secara pribadi maupun untuk umum dalam suatu *WEB*. Hampir semua host atau penyedia *serverWEB* memberikan fasilitas *MySQL* untuk para developer *WEB* yang menginginkan pengelolaan *Database* di *website*nya.

Dalam dunia *website*nya bentuk *interface* atau tatap muka untuk *MySQL* ini sering disebut dengan *PHPMyadmin*. Inilah yang menjadi salah satu faktor adanya keterkaitan antara Bahasa *PHP* dengan *MySQL* terutama dalam hal pengelolaan *WEB*.

Secara umum SQL hanya memiliki 3 komponen penting, yang digunakan untuk mendefinisikan, memanipulasi serta sebagai pengontrol data yang ada pada sebuah database.

Komponen – komponen tersebut diantaranya :

1. *Data Definisi Language (DDL)*

DDL merupakan bahasa atau perintah pada SQL yang digunakan untuk mendefinisikan data seperti menulis, menghapus dan mengatur atau mengubah data.

Perintah – perintah yang ada pada komponen DDL yaitu CREATE, DROP, dan ALTER.

1. CREATE

Digunakan untuk membuat database, membuat tabel, membuat view dan membuat index.

Contoh sintak

Create table namatabel

(namakolom1 typedata1,

namakolom2 typedata2,

namakolom3 typedata3)Type data dapat berupa char(), int, numerik dan sebagainya sesuai denga nama kolom.

1. DROP

Drop pada SQL adalah perintah yang di gunakan untuk menghapus data – data yang telah di simpan pada database

Contoh sintak

Drop table

1. ALTER

kegunaan dari perintah alter adalah untuk merubah atribut pada suatu tabel

Contoh sintak

Alter table namatabel

Modify (namakolom typekolom) untuk mengganti type kolom dan nama kolom.alter table namatabel

add (namakolom type kolom[[before, namakolom]]) menambah kolom.alter table namatabel

drop (namakolom typekolom) untuk menghapus kolom pada sebuah tabel.

1. *Data Manipulation Language (DML)*
2. *INSERT*

insert berfungsi untuk menambah niali pada suatu tabel.

Contoh sintak

insert into namatabel values (‘value1’, ‘value2’,’ …’)keterangan : value adalah nilai yang ingin dimasukkan ke dalam kolom

1. *UPDATE*

Berfungsi untuk merubah record atau nilai didalam suatu kolom

Contoh sintak

Update namatabel SET namakolom = recordbaru where kondisi keterangan record baru di ubah sesuai dengan nama nilai baru yang akan di ubah

1. *DELETE*

Menghapus record atau nilai didalam suatu kolom pada tabel tertentu

Contoh sintak

*DELETE* nama kolom from namatabel where kondisi

1. *SELECT*

select adalah perintah yang di gunakan untuk menampilkan nilai atau recor yang di pilih

Contoh sintak

select nama kolom from namatabel

1. *Data Control Language (DCL)*
2. *GRANT*

grant berfungsi untuk memberikan izin akses kepada pengguna

Contoh sintak

Grant privilages on namatabel to nama pengguna artinya memberikan izin akses kepada nama pengguna

1. *REVOKE*

yaitu untuk mencabut izin akses kepada pengguna yang telah di beri izin akses

Contoh sintak

*REVOKE* *privileges* ON namatabel from nama pengguna

1. Macam – macam fungsi pada SQL
2. Tanda \*

digunakan untuk memilih semua isi tabel

Contoh sintak

select \* from nama tabel

1. Fungsi WHERE

berfungsi untuk menentukan suatu pilihan atau kondisi tertentu

Contoh sintak

select \* from nama tabel where nama kolom = “kondisi”

1. Fungsi DISTINC

digunakan untuk menghilangakan nilai ganda pada suatu kolom atau tabel

Contoh sintak

select distinct nama kolom from nama tabel

1. Operator AND

digunakan untuk memilih nilai yang memiliki kondisi sama antara kondisi1 dan kondisi2

contoh sintak

select \* from nama tabel where kondisi1=’..’ AND kondisi2 = ‘…’

1. Operator OR

digunakan untuk memilih suatu tabel atau kolom dan menampilkan nilai jika salah satu kondisi dari kondisi yang di tentukan memenuhi

Contoh sintak

select \* from nama tabel where kondisi 1=’..’ OR kondisi 2 = ‘…’

1. Fungsi NOT

digunakan untuk memilih suatu nilai dari tabel atau kolom dan mengurutkan data tersebut

Contoh sintak

select nama kolom from nama tabel

where namakolom = ‘kondisi’

order by nama kolom pengurut desc/asc keterangan : nama kolom pengurut adalah dasar pengurutan

1. Fungsi GROUP BY

berfungsi untuk menelompokkan suatu data, namun pada pada fungsi ini juga biasanya paling banyak melibatkan operator pembanding dan fungsi perhitungan

Contoh sintak

select nama kolom from nama tabel

group by nama kolom yang akan dikelompokkan

having count nama kolom > 4 keterangan having count adalah fungsi yang digunakan untuk mendukung fungsi group by.

1. Fungsi BETWEEN

fungsi between digunakan untuk memilih nilai di antara kondisi yang di tentukan

Contoh sintak

select \* from nama tabel where ‘kondisi’ between ‘kondisi’ – ‘kondisi’

1. Fungsi PERHITUNGAN

* Count : untuk mengitung jumlah baris atau kolom
* Sum : di gunakan untuk menjumlahkan suatu nilai yag memiliki type data numerik
* Avg : berfungsi untuk membuat nilai rata – rata
* Max : berfungsi untuk mencari nilai terbesar dan
* Min : untuk mencri nilai terkecil

Contoh sintak

select avg nama kolom from namatabel

1. Fungsi Join

* INNER JOIN (SIMPLE JOIN)

yaitu bentuk kondisi join dimana di antara 2 atau lebih tabel yang ingin di join memiliki hubungan sehingga semua isi tabel yang di pilih bisa di tampilkan

Contoh sintak

select \* from nama tabel1 inner join nama tabel2

where nama tabel1.kondisi sama = namatabel1. Kondisi sama

* NON-EQUIJOIN

adalah kondisi join yang tidak menagndung opertor sama dengan (=)

Contoh sintak

select \* from nama tabel1 inner join nama tabel2

where nama tabel1.kondisi sama and nama tabel1. Kondisi sama

* OUTER JOIN

yaitu bentuk nilai join yang memenuhi daari kedua tabel, Outer join terdiri dari 2 yaitu,

* LEFT OUTER JOIN

yaitu di tandai dengan operator (+), tabel asal atau yang memiliki tanda (+) akan dilengkapi dengan tabel yang lainnya dan berada pada sebelah kiri.

Contoh sintak

select tabel1.kolom, tabel2.kolom

from tabel1, tabel2

where tabel1.kolom (+) = tabel2.kolom atau select tabel1.kolom, tabel2.kolom

from tabel1 left outer join tabel2

on tabel1.kolom = tabel2.kolom

* RIGHT OUTER JOIN

Sama dengan left outer join namun peletakan data atai nilai dari tabel lain berada pada sebelah kanan tabel asal.

Contoh sintak

select tabel1.kolom, tabel2.kolom

from tabel1, tabel2

where tabel1.kolom = tabel2.kolom(+)atauselect tabel1.kolom, tabel2.kolom

from tabel1 right outer join tabel2

on tabel1.kolom = tabel2.kolom

* SELF JOIN

Self join adalah suatu bentuk kondisi join atau penggabungan yang terjadi pada dua tabel yang sama kondisinya dari kedua tabel tersebut dan menambahkan suatu kata dengan kondisi yang sama antara kedua tabel

Contoh sintak

Select tabel1.nama kolom ||’ works for’ || tabel2.namakolom

from nama tabel1, nama tabel2

where nama tabel1.kolom persamaan tabel1 = nama kolom2.kolom persamaan tabel

## Metode Pengujian

* + 1. Pengujian Black Box

Metode pengujian adalah salah satu cara atau metode untuk menguji perangkat lunak dan data kemungkinan terjadi kesalahan. Pengujian *Black Box* merupakan pendekatan komplementer dari teknik *White Box testing*, karena pengujian *Black Box* mampu mengungkap kesalahan yang lebih luas. Pengujian *Black Box* berfokus pada pengujian persyaratan fungsional perangkat lunak, karena untuk mendapat serangkaian kondisi *input* yang sesuai dengan persyaratan fungsional suatu program.(Yakub, 2012).

* + 1. Pegujian White Box

Pengujian dilakukan untuk menjamin operasi-operasi internal sesuai dengan spesifikasi yang telah ditetapkan dengan menggunakan struktur kendali dari prosedur yang dirancang. Metode white Box Digunakan untuk mengetahui cara kerja internal suatu perangkat lunak. Metode perancangan test case yang menggunakan struktur control dari perancangan procedural untuk mendapatkan test case.

**Pengujian white box adalah** pengujian yang didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur kontrol dari desain program secara procedural untuk membagi pengujian ke dalam beberapa kasus pengujian. Secara sekilas dapat diambil kesimpulan white box testing merupakan petunjuk untuk mendapatkan program yang benar secara 100%.

Penggunaan metode pengujian white box dilakukan untuk memberikan jaminan bahwa semua jalur independen suatu modul digunakan minimal satu kali, menggunakan semua keputusan logis untuk semua kondisi true atau false, mengeksekusi semua perulangan pada batasan nilai dan operasional pada setiap kondisi dan menggunakan struktur data internal untuk menjamin validitas jalur keputusan.

Beberapa persyaratan dalam menjalankan strategi White Box Testing yaitu Mendefinisikan semua alur logika, membangun kasus untuk digunakan dalam pengujian, mengevaluasi semua hasil pengujian dan elakukan pengujian secara menyeluruh.

## *UML (Unified Modelling Language)*

*UML (Unified Modelling Language*) menurut menurut Adi nugroho (2010:6), “*Unified Modelling Language* adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berpradigma berorientasi objek”. Pemodelan (*modeling*) sesungguhnya  digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami.

Metode *Unified Modelling Language* (UML) menggunakan tiga bangunan dasar untuk mendeskripsikan sistem atau perangkat lunak yang akan dikembangkan, yaitu:

1. Sesuatu (*things*)

Ada empat *things* dalam *Unified Modelling Language* (UML):

1. *Structural things*, bagian yang relatif statis dapat berupa elemen-elemen yang bersifat fisik maupun konseptual.
2. *Behavorial things,* bagian dinamis biasanya merupakan kata kerja dari model UML yang mencerminkan perilaku sepanjang waktu
3. *Grouping things,* bagian pengorganisasian dalam UML. Dalam penggambaran model UML yang rumit diperlukan penggambaran paket yang menyederhanakan model. Paket-paket ini kemudian dapat didekomposisi lebih lanjut. Paket berguna bagi pengelompokan sesuatu, misalnya model-model serta subsitem-subsistem.
4. *An notational things,* merupakan bagian yang meperjelas model UML. Dapat berisi komentar yang menjelaskan fungsi serta ciri-ciri tiap element dalam model UML
5. Relasi (*relationship*)

Ada empat *relationship* (hubungan) dalam *Unified Modelling Language* (UML):

1. Ketergantungan (d*ependency*) adalah hubungan dimana perubahan yang terjadi pada suatu elemn independent akan mempengaruhi elemen yang bergantung padanya.
2. Asosiasi adalah apa dan bagaimana yang mengubungkan antara objek satu dengan yang lainnya. Suatu bentuk asosiasi adalah agregasi yang menampilkan hubungan suatu objek dengan bagian-bagiannya.
3. Generalisasi adalah hubungan dimana objek anak berbagai perilaku dan struktur data dari objek yang ada diatasnya (objek induk). Arah dari objek induk ke objek anak dinamakan spesialisasi sedangkan arah sebaliknya dinamakan generalisasi..
4. Realisasi adalah operasi yang benar-benar dilakukan oleh suatu objek.
5. Diagram, *Unified Modelling Language* (UML) menyediakan Sembilan jenis diagram yang dapat dikelompokkan berdasarkan sifatnya (statis dan dinamis).

diagram *use case*

Diagram *use case* menyajikan interaksi antara *use case* dan aktor, dimana aktor dapat berupa orang, peralatan, atau sistem lain yang berinteraksi dengan sistem yang sedang di bangun. *Use case* menggambarkan fungsionalitas sistem atau persyaratan-persyaratan yang harus dipenuhi sistem dari pandangan **No table of figures entries found.**Adapun simbol-simbol Use Case Diangram antara lain :

Table 2.1 Simbol *Use Case Diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1 |  | *Actor* | Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*. |
| 2 |  | *Dependency* | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*). |
| 3 |  | *Generalization* | Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*). |
| 4 |  | *Include* | Menspesifikasikan bahwa *use case* sumber secara *eksplisit*. |
| 5 |  | *Extend* | Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan. |
| 6 |  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya. |
| 7 |  | *System* | Menspesifikasikan paket yang menampilkan sistem secara terbatas. |
| 8 |  | *Use Case* | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor |
| 9 |  | *Collaboration* | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi). |
| 10 |  | *Note* | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi |

Table 2.2 Simbol *Class Diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1 |  | *Generalization* | Hubungan dimana objek anak *(descendent)* berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*). |
| 2 |  | *Nary Association* | Upaya untuk menghindari asosiasi dengan lebih dari 2 objek. |
| 3 |  | *Class* | Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama. |
| 4 |  | *Collaboration* | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor |
| 5 |  | *Realization* | Operasi yang benar-benar dilakukan oleh suatu objek. |
| 6 |  | *Dependency* | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri |
| 7 |  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya |

Table 2.3 Simbol *Sequence Diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1 |  | *LifeLine* | Objek *entity*, antarmuka yang saling berinteraksi. |
| 2 |  | *Message* | Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi |
| 3 |  | *Message* | Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi |

Tabel 4. Simbol StateChart Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1 |  | *State* | Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek. |
| 2 |  | *Initial Pseudo State* | Bagaimana objek dibentuk atau diawali |
| 3 |  | *Final State* | Bagaimana objek dibentuk dan dihancurkan |
| 4 |  | *Transition* | Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya |
| 5 |  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya. |
| 6 |  | *Node* | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi. |

Tabel 5. Simbol Actifity Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| **1** |  | *Actifity* | Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain |
| **2** |  | *Action* | State dari sistem yang mencerminkan eksekusi dari suatu aksi |
| **3** |  | *Initial Node* | Bagaimana objek dibentuk atau diawali. |
| **4** |  | *Actifity Final Node* | Bagaimana objek dibentuk dan dihancurkan |
| **5** |  | *Fork Node* | Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran |

## *Kerangka Pikir*

Untuk lebih memperjelas kerangka berpikir yang akan disajikan maka berikut ini digunakan dalam bentuk diagram.

Koperasi SMKN 2 Enrekang merupakan salah satu koperasi yang berperan serta secara aktif dalam upaya meningkatkan kesejahteraan ekonomi dan sosial anggotanya.

Pengolahan data administrasi dan laporan koperasi masih menggunakan sistem yang sederhana dengan melakukan pencatatan satu persatu sehingga tidak efektif.

Merancang sebuah aplikasi pengololaan administrasi dan laporan koperasi yang lebih akurat, dengan menggunakan program aplikasi dengan bahasa pemrograman *Javascript*, *PHP* dengan menggunakan database *MYSQL*.

Diharapkan dengan adanya aplikasi ini dapat membantu pihak Koperasi SMKN 2 Enrekang yang akan memudahkan dalam mengolah data pembayaran kredit dan data tunggakan anggotanya.

Gambar 2.1 Kerangka Pikir

# BAB III

**METODE PENELITIAN**

## Jenis Penelitian

Adapun jenis dan sumber data yang dibutuhkan dalam penelitian ini adalah sebagai berikut:

1. Data Primer yaitu data yang diperoleh melalui wawancara yang dilakukan langsung dengan *responden* yang dapat mewakili beberapa sumber dalam hal ini adalah nama *user* ataupun guru-guru yang terdaftar di koperasi ini serta bentuk pelaporan dan data lain yang dibutuhkan oleh sistem.
2. Data sekunder yaitu merupakan data yang diperoleh melalui studi kepustakaan atau dari berbagai literatur dengan menelaah buku-buku dan tulisan-tulisan atau internet, jurnal , yang relevan dengan permasalahan yang diteliti.

## Lokasi Dan Waktu

Lokasi penelitian yang dilakukan adalah Dinas Pendidikan Kabupaten Enrekang dan waktu yang dipergunakan untuk melakukan penelitian adalah kurang lebih dari 2 (dua) bulan.

## Alat Dan Bahan

Dalam melakukan penelitian, penulis memerlukan alat dan bahan penelitian yang mendukung kegiatan penelitian tersebuat.

1. Alat yang digunakan selama proses penelitian yaitu :
2. Perangkat Keras

* *Laptop Asus X452C* dengan spesifikasi :

Tabel 3.1 Spesifikasi Laptop

|  |  |
| --- | --- |
| Prosesor | *Intel Core i3-4005U (1.7GHz)* |
| *RAM* | *4 GB* |
| *Hard Disk Drive* | *500 GB* |
| *Monitor* | 14.0 *Inches* |

1. Perangkat Lunak

Tabel 3.2 Perangkat Lunak

|  |  |
| --- | --- |
| Sistem Operasi | *Windows 10* |
| *Text Editor* | *Sublime Text* |
| *Web Server* | *XAMPP* |
| *Database* | *MYSQL* |
| *Web Browser* | *Google Chrome* |

## Rancangan Sistem

Dibawah merupakan rancangan sistem yang diusulkan penulis. Dimana user maupun guru dapat melihat info maupun laporan berkaitan dengan simpanan,dan pinjaman yang dilakukan manakala admin pula mengelola segala proses yang berlaku di sistem administrasi koperasi ini. Untuk lebih jelas dapat dilihat pada *Use Case Diagram* dibawah ini.



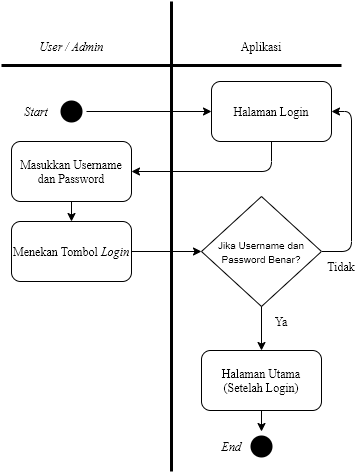
Gambar 3.1 *Use Case Diagram*

Penjelasan Use Case Diagram Sistem Yang Diusulkan

|  |  |
| --- | --- |
| Nama *Use Case* | Deskripsi *Use Case* |
| Login | Use Case ini menjelaskan bahwa admin dan *user* harus *login* untuk mengakses sistem. |
| Menampilkan *List User* | Use Case ini menjelaskan bahwa admin dapat menampilkan *list user* yang terdaftar dalam aplikasi |
| Tambah *user* | Use Case ini menjelaskan admin dapat menambah *user*/ guru. |
| Menampilkan Log Simpanan Pokok, Simpanan Wajib dan Simpanan Sukarela | Use Case ini menjelaskan bahwa admin maupun *user* dapat menampilkan log simpanan pokok , simpanan wajib dan simpanan sukarela, namun *user* khusus untuk laporan *user* tersebut |
| Menambah Simpanan Wajib dan Sukarela | Use Case ini menjelaskan bahwa admin dapat menambah simpanan wajib dan sukarela untuk *user.* |

## Diagram Alir

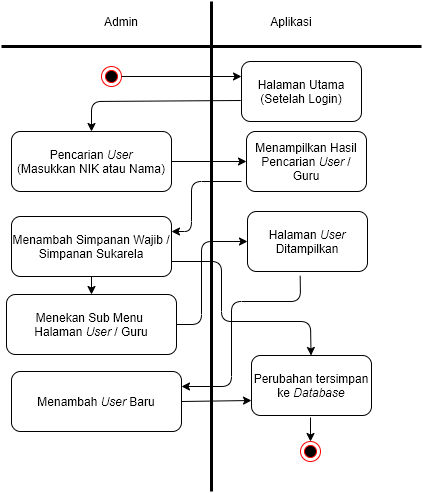
1. *Activity Diagram*
2. *Activity Diagram Login*

**

Gambar 3.2 *Activity Diagram Login*

*Activity* *Diagram* diatas menjelaskan bahwa *user* maupun admin akan dihadapkan ke halaman *login.*  Kemudian disini *user* maupun admin akan mengisi *username* dan *password* dan kemudian menekan tombol *login.* Aplikasi akan memeriksa *username* dan *password* yang dimasukkan jika benar maka akan dihadapkan ke halaman utama *user* maupun admin setelah sukses *login.* Namun jika *username* dan *password* salah maka *user* maupun admin akan dihadapkan kembali ke tampilan login

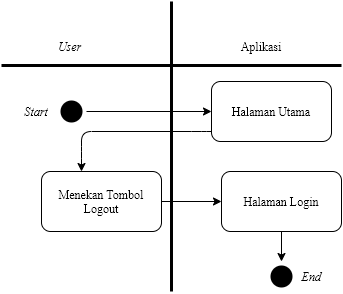
1. *Activity Diagram Admin*



Gambar 3.3 *Activity Diagram Admin*

*Activity* *Diagram* diatas menjelaskan bahwa admin setelah sukses login akan dihadapkan ke tampilan awal aplikasi. Kemudian admin dapat mencari *user* yang telah ditambahkan sebelumnya di kolum pencarian dengan memasukkan NIK atau nama *user.* Kemudian aplikasi akan menampilkan hasil pencarian *user* yang dicari oleh admin. Disini admin dapat menambah simpanan wajib ataupun simpanan sukarela *user* yang dicari sebelumnya. Admin juga dapat menekan sub menu Halaman *User* / Guru , kemudian aplikasi akan menampilkan halaman *user* dimana di halaman ini admin dapat menambah *user* baru. Kemudian segala perubahan akan tersimpan ke *database* aplikasi.

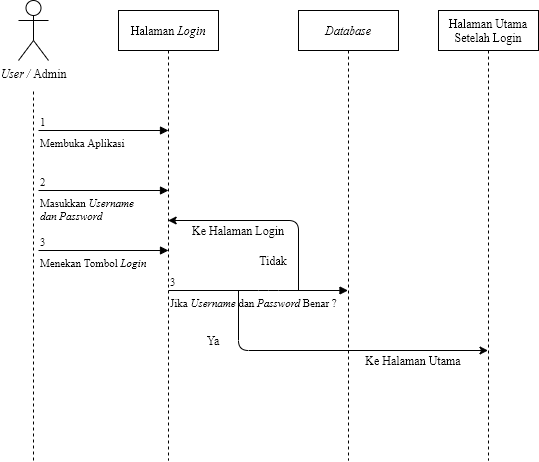
1. *Activity Diagram User*



Gambar 3.4 *Activity Diagram User*

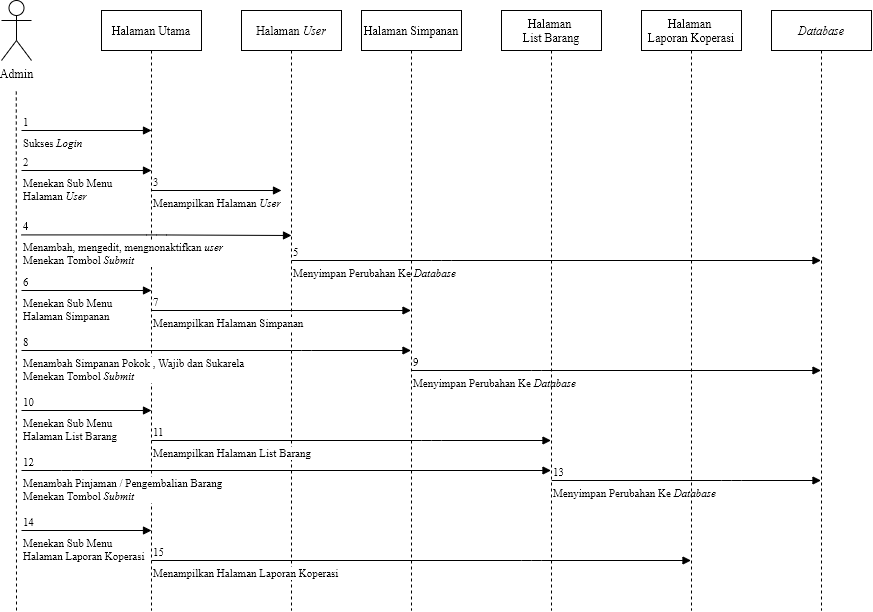
*Activity* *Diagram* diatas menjelaskan bahwa admin setelah sukses login akan dihadapkan ke halaman utama *user* dimana halaman ini akan menampilkan detail *user* , simpanan pokok, simpanan wajib, simpanan sukarela *user.*

1. *Sequence Diagram*
2. *Sequence Diagram Login*

**

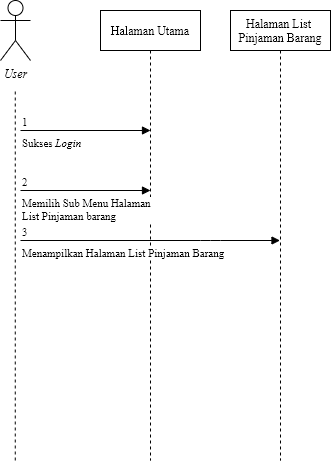
Gambar 3.5 *Sequence Diagram Login*

1. *Sequence Diagram Admin*

**

Gambar 3.6 *Sequence Diagram Admin*

1. *Sequence Diagram User*



Gambar 3.7 Sequence Diagram User

# BAB IV

**HASIL DAN PEMBAHASAN**

## Metode Pengujian

1. Metode Pengujian

Pengujian aplikasi dilakukan dengan menggunakan dua metode pengujian yaitu pengujian *Black Box*  dan pengujian *White Box*.

1. Teknik Pengujian

Pengujian *black-box* berkaitan dengan pengujian yang dilakukan pada antarmuka perangkat lunak.

Meski didesain untuk mengungkap kesalahan, pengujian *black-box*  digunakan untuk memperlihatkan bahwa fungsi-fungsi perangkat lunak adalah operasional, bahwa input diterima dengan baik dan ouput dihasilkan dengan tepat, dan integritas informasi external dipelihara. Pengujian *black-box*  menguji beberapa aspek dasar suatu aplikasi dengan sedikit memperhatikan struktur logika internal perangkat lunak tersebut.

Sedangkan pengujian *White-box*  berfokus pada efektifitas aplikasi yang dirancang.

## Pengujian *White Box*

1. W*hite Box Login*

Tabel 4.1 *White Box Login*

|  |  |
| --- | --- |
| *Flowchart* | *Flowgraph* |
|  |  |

Menghitung jumlah *Region, Cyclomatic Complexity* dan *Independent Path.*

*Independent Path.*

1. *Path* 1 = 1-2-3-4-5-6
2. *Path* 2 = 1-2-3-4-2

*Cyclomatic Complexity.*

V(G) = E - N + 2

= 6 - 6 + 2 = 2

V(G) = P + 1

= 1 + 1

= 2

Region = 2

Grafik Matriks

Tabel 4.2 Grafik Matriks L*ogin*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | E-1 |
| 1 |  | 1 |  |  |  |  | 1 - 1 = 0 |
| 2 |  |  | 1 |  |  |  | 1 - 1 = 0 |
| 3 |  |  |  | 1 |  |  | 1 - 1 = 0 |
| 4 |  | 1 |  |  | 1 |  | 2 - 1 = 1 |
| 5 |  |  |  |  |  | 1 | 1 - 1 = 0 |
| 6 |  |  |  |  |  |  |  |
| SUM ( E + 1) | | | | | | | 1 + 1 = 2 |

Berdasarkan perhitungan diatas, Region = 2, Independent Path = 2, Cyclomatic Complexity = 2, maka form Login dinyatakan bebas dari kesalahan.

1. *White Box Admin*

Tabel 4.3 White Box Admin

|  |  |
| --- | --- |
| *Flowchart* | *Flowgraph* |
| C:\Users\Kicap Karan\Documents\kerja\koperasi\rancangan sistem\flowchart admin.png | C:\Users\Kicap Karan\Documents\kerja\koperasi\rancangan sistem\Flowgraph Admin.png |

Menghitung jumlah *Region, Cyclomatic Complexity* dan *Independent Path.*

*Independent Path.*

1. *Path* 1 = 1-2-3-4-7-10-13-14-15-16
2. *Path* 2 = 1-2-3-4-5-6-15-16
3. *Path* 3 = 1-2-3-4-7-8-9-15-16
4. *Path* 4 = 1-2-3-4-7-10-11-12-15-16

*Cyclomatic Complexity.*

V(G) = E - N + 2

= 18 - 16 + 2 = 4

V(G) = P + 1

= 3 + 1

= 4

Region = 4

Tabel 4.4 Grafik Matriks Admin

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | E - 1 |
| 1 |  | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 - 1 = 0 |
| 2 |  |  | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 - 1 = 0 |
| 3 |  |  |  | 1 |  |  |  |  |  |  |  |  |  |  |  |  | 1 - 1 = 0 |
| 4 |  |  |  |  | 1 |  | 1 |  |  |  |  |  |  |  |  |  | 2 - 1 = 1 |
| 5 |  |  |  |  |  | 1 |  |  |  |  |  |  |  |  |  |  | 1 - 1 = 0 |
| 6 |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 |  | 1 - 1 = 0 |
| 7 |  |  |  |  |  |  |  | 1 |  | 1 |  |  |  |  |  |  | 2 - 1 = 1 |
| 8 |  |  |  |  |  |  |  |  | 1 |  |  |  |  |  |  |  | 1 - 1 = 0 |
| 9 |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 |  | 1 - 1 = 0 |
| 10 |  |  |  |  |  |  |  |  |  |  | 1 |  | 1 |  |  |  | 2 - 1 = 1 |
| 11 |  |  |  |  |  |  |  |  |  |  |  | 1 |  |  |  |  | 1 - 1 = 0 |
| 12 |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 |  | 1 - 1 = 0 |
| 13 |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 |  |  | 1 - 1 = 0 |
| 14 |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 |  | 1 - 1 = 0 |
| 15 |  |  | 1 |  |  |  |  |  |  |  |  |  |  |  |  | 1 | 2 - 1 = 1 |
| 16 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0 - 1 = -1 |
| SUM ( E + 1 ) | | | | | | | | | | | | | | | | | 3 + 1 = 4 |

Berdasarkan perhitungan diatas, Region = 4, Independent Path = 4 Cyclomatic Complexity = 4, maka form Login dinyatakan bebas dari kesalahan.

1. *White Box User*

Tabel 4.5 White Box User

|  |  |
| --- | --- |
| *Flowchart* | *Flowgraph* |
| C:\Users\Kicap Karan\Documents\kerja\koperasi\rancangan sistem\Flowchart User.png | C:\Users\Kicap Karan\Documents\kerja\koperasi\rancangan sistem\Flowgraph User.png |

Menghitung jumlah *Region, Cyclomatic Complexity* dan *Independent Path.*

*Independent Path.*

1. *Path* 1 = 1-2-3-4-5-6
2. *Path* 2 = 1-2-3-4-2

*Cyclomatic Complexity.*

V(G) = E - N + 2

= 6 - 6 + 2 = 2

V(G) = P + 1

= 1 + 1

= 2

Region = 2

Grafik Matriks

Tabel 4.6 Grafik Matriks L*ogin*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | E-1 |
| 1 |  | 1 |  |  |  |  | 1 - 1 = 0 |
| 2 |  |  | 1 |  |  |  | 1 - 1 = 0 |
| 3 |  |  |  | 1 |  |  | 1 - 1 = 0 |
| 4 |  |  |  |  | 1 |  | 1 - 1 = 0 |
| 5 |  |  | 1 |  |  | 1 | 2 - 1 = 1 |
| 6 |  |  |  |  |  |  |  |
| SUM ( E + 1) | | | | | | | 1 + 1 = 2 |

Berdasarkan perhitungan diatas, Region = 2, Independent Path = 2, Cyclomatic Complexity = 2, maka form Login dinyatakan bebas dari kesalahan.

## Pengujian *Black Box*

1. Pengujian Form *Login*

Tabel 4.7 Black Box Login

|  |  |  |
| --- | --- | --- |
| *Test Factor* | Hasil | Keterangan |
| Login dengan memasukkan *username* dan *Password* | Berhasil | Jika *Username* dan *Password* benar, maka akan login ke halaman utama |
|  | | |

1. Pengujian *Form* *Admin* ( Halaman *User* )

Tabel 4.8 *Black Box* Admin (Halaman *User*)

|  |  |  |
| --- | --- | --- |
| *Test Factor* | Hasil | Keterangan |
| *User /* Guru Baru ditambahkan ke aplikasi | Berhasil | Jika *User /* Guru Baru ditambahkan ke aplikasi, maka aplikasi akan menyimpan *user* yang barku ditambahkan |
|  | | |

1. Pengujian *Form* *Admin* ( Halaman Utama)

Tabel 4.8 *Black Box* Admin (Halaman Utama)

|  |  |  |
| --- | --- | --- |
| *Test Factor* | Hasil | Keterangan |
| Pencarian *User* / Guru pada kolum pencarian. | Berhasil | Jika *User /* Guru Baru ditambahkan ke aplikasi, maka aplikasi akan menyimpan *user* yang barku ditambahkan |
|  | | |

# BAB V

**PENUTUP**

## Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah diuraikan pada bab-bab sebelumnya, maka dapat ditarik kesimpulan sebagai berikut

Telah dihasilkan sebuah Aplikasi Administrasi Dan Pelaporan Koperasi Pada Kantor Dinas Pendidikan Kabupaten Enrekang, menggunakan bahasa pemrograman PHP, Javascript, HTML dan CSS serta *database* menggunakan MYSQL. Dimana admin pada aplikasi ini dapat mengelola semua proses yang berkaitan dengan administrasi koperasi Dinas Pendidikan Kabupaten Enrekang mulai dari penambahan *user /* guru , penambahan maupun *update*  simpanan serta menampilkan laporan berkaitan operasi koperasi administrasi, manakala *user* pula dapat melihat laporan simpanan yang mereka lakukan pada koperasi ini.

Aplikasi yang dibuat sudah bebas dari kesalahan dengan menggunakan dua pengujian yaitu pengujian  *Black Box*  dan pengujian  *White Box.*

## Saran

Setelah melakukan penelitian di koperasi Dinas Pendidikan Kabupaten Enrekang, maka penulis mengajukan saran. Untuk pengembangan lebih lanjut sebaiknya aplikasi ini dikembangkan menjadi mobile bagi *user* dan admin agar dapat diakses bagi admin dan *user* di mana saja.

# DAFTAR PUSTAKA

Agung, L. (2010). *Aplikasi Pemrograman Javascript untuk Halaman Web.* Yogyakarta: Andi Offset.

Agustina, M., Rismawati, N., & Acep. (2020). Perancangan Sistem Informasi Simpan Pinjam Di Koperasi Karyawan MT Haryono Berbasis Java. *Perancangan Sistem Informasi Simpan Pinjam Di Koperasi Karyawan MT Haryono Berbasis Java* .

Apriliani, I. N. (2018). PERANCANGAN SISTEM SIMPAN PINJAM PADA KOPERASI. *PERANCANGAN SISTEM SIMPAN PINJAM PADA KOPERASI* .

Arief, M. R. (2011). *Pemrograman Web Dinamis menggunakan PHP dan MySQL.* Makassar: Penerbit ANDI.

Jubilee. (2011). *HTML 5 Cara cepat dan mudah menguasai script HTML 5.* Yogyakarta: Jubilee Enterprise.

K. .. (2001). Koeprasi Indonesia yang Berdasarkan Pancasila dan UUD 1945. In K. .., *Koeprasi Indonesia yang Berdasarkan Pancasila dan UUD 1945* (p. 21). Jakarta: Rineka Cipta.

Nugroho, A. (2008). *Menjadi Administrator Basis Data ORACLE 10 g.* Bandung: Infromatika Bandung.

Solichin, A. (2010). *MYSQL 5 Dari Pemula Hingga Mahir.* Jakarta.

Suciyanto, M. (2014). Perancangan Aplikasi Sistem Informasi Pinjaman Keuangan Anggota Koperasi Syariah Usaha Mandiri. *Perancangan Aplikasi Sistem Informasi Pinjaman Keuangan Anggota Koperasi Syariah Usaha Mandiri* .

# LAMPIRAN