

RANCANGBANGUN APLIKASI ADMINISTRASI DESA BERBASIS WEBSITE PADA KANTOR DESA BAROS

SKRIPSI

Karya Tulis Sebagai Syarat untuk Memperoleh
Gelar Sarjana Komputer dari Fakultas Teknologi Informasi
Universitas Bale Bandung

Disusun oleh:

SYIFA FAUZIAH

NPM. 301170022



PROGRAM STRATA 1

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS BALE BANDUNG

BANDUNG

2021

LEMBAR PERSETUJUAN PEMBIMBING
RANCANG BANGUN APLIKASI ADMINISTRASI DESA
BERBASIS WEBSITE PADA KANTOR DESA BAROS

Disusun oleh:

SYIFA FAUZIAH

NPM. 301170022

Telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar

SARJANA KOMPUTER

Pada

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS BALE BANDUNG

Bandung, Juli 2021

Disetujui oleh:

Pembimbing 1



Yudi Herdiana, S.T., M.T.
NIK. 04104808008

Pembimbing 2



Rosmalina, S.T., M.Kom.
NIK. 04104808122

LEMBAR PENGESAHAN PENGUJI
PEMBANGUNAN APLIKASI KLASIFIKASI KODE SURAT
BERBASIS ANDROID MENGGUNAKAN ALGORITMA
BOYER-MOORE DI KANTOR KECAMATAN CIPARAY

Disusun oleh :
SYIFA FAUZIAH
NPM. 301170022

Telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar

SARJANA KOMPUTER

Pada

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG

Baleendah, Juli 2021

Disetujui oleh:

Penguji 1



Sutyono W.P, S.T,M.Kom
NIK. 01043180002

Penguji 2



Yaya Sufiaya., S.kom, M.T
NIK. 01043170007

**LEMBAR PENGESAHAN PROGRAM STUDI
RANCANG BANGUN APLIKASI ADMINISTRASI DESA
BERBASIS WEBSITE PADA KANTOR DESA BAROS**

Disusun Oleh:

SYIFA FAUZIAH

NPM. 301170022

SKRIPSI ini telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar

SARJANA KOMPUTER

Pada

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS BALE BANDUNG

Bandung, Juli 2021

Mengetahui,

Dekan,



Yudi Herdiana, S.T., M.T.

NIK. 04104808008

Mengesahkan,

Ketua Program Studi



Yusuf Muharam M.Kom.

NIK. 0410482000

LEMBAR PERNYATAAN KEASLIAN

Saya yang bertanda tangan dibawah ini:

Nama : SYIFA FAUZIAH

NPM : 301170022

Judul : **RANCANG BANGUN APLIKASI ADMINISTRASI DESA PADA KANTOR DESA BAROS**

Menyatakan dengan sebenarnya bahwa penulisan skripsi ini berdasarkan hasil penelitian, pemikiran dan pemaparan asli dari saya sendiri, baik untuk naskah laporan maupun kegiatan *programming* yang tercantum sebagai bagian dari skripsi ini. Jika terdapat karya orang lain, saya mencantumkan sumber yang jelas.

Pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini dan sanksi lain sesuai dengan peraturan yang berlaku di FAKULTAS TEKNOLOGI INFORMASI

Demikian surat pernyataan ini saya buat dalam keadaan sadar tanpa paksaan dari pihak manapun.

Baleendah, Juli 2021

Yang membuat Pernyataan



SYIFA FAUZIAH

NPM.301170022

ABSTRAK

Pengelolaan keuangan desa merupakan suatu sub sistem dari pengelolaan keuangan negara dan daerah dalam mendanai penyelenggaraan pemerintahan desa dan pemberdayaan masyarakat desa sehubungan dengan hal tersebut, maka dalam pengelolaan keuangan desa diperlukan mengevaluasi standar yang digunakan dimulai dari aspek perencanaan dan penganggaran maupun aspek pelaksanaan, penatausahaan keuangan desa dan aspek pertanggungjawaban keuangan desa. Proses perencanaan dan penganggaran maupun aspek pelaksanaan, penatausahaan keuangan desa dan aspek pertanggungjawaban merupakan kesempatan yang baik untuk melakukan evaluasi apakah pemerintahan desa melakukan tugasnya secara efektif dan efisien, dengan melakukan hal yang benar terhadap pencapaian tujuan dan sasaran untuk memenuhi kebutuhan masyarakat. Pembuatan aplikasi Administrasi Desa ini di bangun menggunakan HTML, CSS dan Bootstrap sebagai tampilannya. Bahasa pemograman PHP sebagai sistem/ back end dan MYSQL sebagai data base. Hasil dari penelitian ini adalah Aplikasi Administrasi Desa Berbasis *website*, bisa diakses langsung oleh Admin dan Bendahara Desa sebagai efisiensi mengelola data keuangan yang dibutuhkan oleh bendahara desa, fitur yang bisa di akses oleh Admin dan bendahara yaitu penganggaran desa, perencanaan desa, penatausahaan, dan laporan penatausahaan. permasalahan yang terjadi di kantor desa bros yaitu pengelolaan data keuangan desa masih menggunakan cara semi manual (*Microsoft Excel*), sehingga banyak kelemahannya yaitu besarnya human *error* dan ketidak seimbangan (*balance*), dikarenakan banyaknya format yang harus dibuat dalam beberapa lembar kerja. Hal ini menyebabkan pengerjaan laporan keuangan akan membutuhkan waktu lama. Metodologi pengembangan sistem yang digunakan yaitu metode air terjun yang dikenal juga dengan istilah metode *waterfall*. Metode *waterfall* adalah metode pengembangan perangkat lunak yang memungkinkan pembuatan sistem dilakukan secara terstruktur dan sistematis (berurutan) sesuai dengan situs pengembangan yang ada. Metode ini disebut *waterfall* atau air terjun karena dalam prosesnya, sistem akan dibuat berurutan setahap demi setahap. Soulsu dari permasalahan tersebut maka dalam penelitian ini penulis akan membangun Aplikasi Administrasi Desa yang dapat mempermudah Admin dan Bendahara Desa dalam mengelola data perencanaan, penganggaran, dan penatausahaan. Kesimpulan dari peenelitian ini yaitu penerapan Aplikasi Administrasi Desa bertujuan untuk memudahkan bendahara desa dalam mengelola keuangan desa dengan baik sehingga dapat berdampak pada kesejahteraan masyarakat desa baros. Mulai dari proses perencanaan, penganggaran, penatausahaan sampai dengan pelaporan keuangan desa bisa berjalan dengan baik dengan diterapkannya Aplikasi Administrasi Desa ini.

Kata Kunci: Pengelolaan keuangan, Metode *waterfall*, Aplikasi Administrasi Desa

ABSTRACT

Village financial management is a sub-system of state and regional financial management in funding the implementation of village governance and village community empowerment in connection with this, so in village financial management it is necessary to evaluate the standards used starting from planning and budgeting aspects as well as implementation aspects, village financial administration. and aspects of village financial accountability. The planning and budgeting process as well as aspects of implementation, village financial administration and accountability aspects are good opportunities to evaluate whether the village government is carrying out its duties effectively and efficiently, by doing the right things towards achieving goals and targets to meet community needs. The making of this Village Administration application was built using HTML, CSS and Bootstrap as its appearance. PHP programming language as system / back end and MYSQL as data base. The results of this study are a website-based Village Administration Application, which can be accessed directly by the Admin and Village Treasurer as an efficiency in managing the financial data needed by the village treasurer, features that can be accessed by the Admin and treasurer, namely village budgeting, village planning, administration, and administrative reports. . The problem that occurs in the Bros village office is that the management of village financial data still uses a semi-manual method (Microsoft Excel), so that there are many weaknesses, namely the amount of human error and balance, due to the many formats that must be made in several worksheets. This causes the processing of financial statements will take a long time. The system development methodology used is the waterfall method, also known as the waterfall method. The waterfall method is a software development method that allows system creation to be carried out in a structured and systematic (sequential) manner according to the existing development site. This method is called the waterfall or waterfall because in the process, the system will be made sequentially step by step. The solution to these problems, in this study the author will build a Village Administration Application that can facilitate the Village Admin and Treasurer in managing planning, budgeting, and administrative data. The conclusion of this research is that the Application of the Village Administration Application aims to facilitate the village treasurer in managing village finances properly so that it can have an impact on the Baros village community. Starting from the planning, budgeting, administration to village financial reporting processes can run well with the implementation of this Village Administration Application.

Keywords: financial management, waterfall method, village administration application

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT yang telah melimpahkan rahmat karunia-Nya sehingga penulis dapat menyelesaikan skripsi dengan baik dan tepat waktu. Skripsi ini dibuat berdasarkan hasil kegiatan penelitian penulis yang berjudul “ **RANCANG BANGUN APLIKASI ADMINISTRASI DESA BERBASIS WEBSITE PADA KANTOR DESA BAROS**”. Dalam poses pembuatan skripsi ini tidak sedikit kendala yang penulis alami. Namun berkat bantuan dan dukungan dari berbagai pihak, Alhamdulillah kendala tersebut dapat penulis lalui dengan baik. Oleh sebab itu, penulis ingin mengucapkan terimakasih kepada:

1. Allah Swt yang telah memberikan rahmat dan karunianya saat mengerjakan skripsi ini.
2. Kedua orang tua yang selalu memberikan dukungan dan do’a.
3. Bapak Yudi Herdiana, S.T., M.T. selaku Dekan Fakultas Teknologi Informasi dan juga selaku Dosen Pembimbing Utama.
4. Bapak Yusuf Muharam S.Kom., M.Kom. selaku Ketua Prodi Fakultas Teknologi Informasi.
5. Ibu Rosmalina, S.T., M.Kom. selaku Pembimbing Kedua
6. Semua staf pengajar di Fakultas Teknologi Informasi
7. Heru Kiatno, S.PD.,M.SI selaku Sekdes Desa Baros
8. Delna Nur Pranata *my boyfriend* yang sedang mennempuh *study* S2 di Seoul National University Korea Selatan yang selalu memotivasi dan menyemangati.
9. Boygrup Super Junior dan Exo yang lagu-lagu nya selalu menjadi penyemangat dan memberi inspirasi saat mengerjakan skripsi.
10. Dan rekan-rekan seperjuangan yang selalu mendukung dan memberi masukan untuk membantu dalam penyusunan laporan ini.

Bandung, Juli 2021

Syifa Fauziah

DAFTAR ISI

ABSTRAK	vi
ABSTRACK	vii
KATA PENGANTAR	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL.....	xiv
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.1.2 Rumusan Masalah	2
2.1.3 Batasan Masalah.....	3
3 1.4 Tujuan Penelitian.....	3
1.5 Metode Pengumpulan Data	3
1.5.1 Metode Pengembangan Sistem	4
1.5.2 Metode Perancangan	6
1.6 Sistematika Penulisan	7
BAB II TINJAUAN PUSTAKA.....	8
2.1.1 Landasan Teori.....	8
2.2 Dasar Teori.....	10
2.2.1 Keuangan Desa.....	10
2.2.2 Manajemen Keuangan.....	14
2.2.3 Aplikasi Keuangan	14
2.2.4 Aplikasi	16
2.2.5 Website.....	17
2.2.6 Mysql	18
2.2.7 HTML	19
2.2.8 Pengertian CSS	20
2.2.9 Java Script	20
2.2.10 J query	21
2.2.11 AJAX	21
2.2.12 Php	22

2.2.13 Morris Js.....	23
2.2.14 XAMPP.....	23
2.2.15 Boostp.....	24
2.2.16 Sublime Text.....	24
2.2.17 Web Browser.....	25
2.2.18 Unfiled Modeling Language (UML).....	25
2.2.19 Diagram Class.....	29
BAB III METODELOGI PENELITIAN	30
3.1 Deskripsi	31
3.2.1 Investigasi Pendahuluan.....	31
3.2.2 Analisis Masalah	31
3.2.3 Analisis Kebutuhan	32
3.2.4 Requeriment	32
3.2.5 Implementasi	34
3.2.6 Verivication.....	35
3.2.7 Maintenance	35
BAB IV ANALISIS DAN PERANCANGAN.....	36
1.1 Analisis.....	36
4.1.1 Analisis Masalah	36
4.1.2 Analisis Software	36
4.1.3 Web browser	37
4.1.4 Analisis User Interface.....	38
1.1.6 Analisis Data	39
4.2 Perancangan	40
4.2.1 Use Case Diagram.....	41
4.2.2 Activity Diagram Admin	43
4.2.3 Class Diagram Admin.....	48
4.2.4 Perancangan Tabel	48
4.2.5 Perancangan User Interface.....	60
BAB V IMPLEMENTASI DAN PENGUJIAN	47
1.6 Implementasi	47
1.5.1 Implementasi User Interface	47

1.5.2 Implementasi Bahasa Pemograman Php	48
1.5.3 IMplementasi XAMPP	48
1.5.4 Hasil Implementasi.....	90
1.5.5 Hasil Pengujian	90
BAB VI KESIMPULAN DAN SARAN	94
6.1 Kesimpulan	94
6.2 Saran.....	95
DAFTAR PUSTAKA	96
LAMPIRAN.....	98
RIWAYAT HIDUP.....	133

DAFTAR GAMBAR

Gambar 1.1 <i>Metode Waterfal</i>	5
Gambar 1.3 Kerangka Pikir.....	30
Gambar 4.5 <i>Use Case Diagram</i> Admin	42
Gambar 4.6 <i>Activity Diagram</i> Admin.....	43
Gambar 4.7 <i>Activity Diagram</i> Menu Perencanaan	44
Gambar 4.8 <i>Activity Diagram</i> Menu Penganggaran.....	45
Gambar 4.9 <i>Activity Diagram</i> Penyelenggaraan Desa	45
Gambar 4.10 <i>Activity Diagram</i> Laporan Penganggaran	47
Gambar 4.11 <i>Class Diagram</i> Admin	48
Gambar 4.12 From <i>Login</i> Aplikasi Administrasi Desa.....	61
Gambar 4.13 Dashboard Aplikasi Keuangan Desa.....	62
Gambar 4.14 Menu Penganggaran	63
Gambar 4.15 Menu RPJM	64
Gambar 4.16 Penganggaran Desa	65
Gambar 4.17 Menu RAP	66
Gambar 4.18 Menu RAB	67
Gambar 4.19 Menu Penatausahaan Desa	68
Gambar 4.20 Menu Penyelenggaraan Desa	79
Gambar 4.21 Menu Laporan	70
Gambar 4.22 Menu Administrator	71
Gambar 50 Tampilan Login	72
Gambar 5.1 <i>interface Dashboard</i> Aplikasi Administrasi Desa	73
Gambar 5.2 Menu Perencanaan Desa	73
Gambar 5.3 From Data Umum	74
Gambar 5.4 From Data Visi	74
Gambar 5.5 From Data Misi	75
Gambar 5.6 From Data Tujuan	77

Gambar 5.7 From Data Sasaran	76
Gambar 5.8 From Data Pelaksanaan Kegiatan	76
Gambar 5.9 Menu Penganggaran Desa	77
Gambar 5.10 From Menu Data Kelompok	77
Gambar 5.11 From Menu Data Jenis	78
Gambar 5.12 From Menu Data Objek.....	78
Gambar 5.13 Sub Menu RAP	79
Gambar 5.14 From Data RAP	80
Gambar 5.15 Sub Menu RAB	80
Gambar 5.17 From Data RAB	81
Gambar 5.18 Tampilan From Data RAB Rincian.....	81
Gambar 5.19 Tampilan Menu Penatausahaan.....	82
Gambar 5.20 Menu Penerimaan.....	82
Gambar 5.21 From Penerimaan Bank	83
Gambar 5.22 Tampilan From Rincian Penerimaan Bank	83
Gambar 5.23 Tampilan From Data Penerimaan Tunai	84
Gambar 5.24 Tampilan From Rincian Penerimaan Tunai	84
Gambar 5.25 Menu Penyelenggaraan	85
Gambar 5.26 Tampilan From Data SPP.....	85
Gambar 5.27 Tampilan From Data Rincian SPP	86
Gambar 5.28 Tampilan From Bukti Pengeluaran	86
Gambar 5.29 Tampilan From Potongan.....	87
Gambar 5.30 Tampilan Menu Laporan.....	87
Gambar 5.31 Tampilan Menu Laporan Penganggaran Desa	88
Gambar 5.32 Tampilan Menu Laporan Penatausahaan	88
Gambar 5.33 Menu Administrator	89
Gambar 5.34 Data Administrator	89

DAFTAR TABEL

Tabel 1 Simbol Use Case Diagram	26
Tabel 2 Simbol Activity Diagram	28
Tabel 4.0.1 Sefesifikasi Hardware	36
Tabel 4.2 Analisis Data	39
Tabel 4.3 <i>Actor</i>	41
Tabel 4.10 Data Umum	49
Tabel 4.11 Bukti Pengeluaran	50
Tabel 4.12 Data Potongan	51
Tabel 4.13 Visi	51
Tabel 4.14 Misi	52
Tabel 4.15 Tujuan	52
Tabel 4. 16 Sasaran	52
Tabel 4.17 Pelaksanaan Kegiatan	53
Tabel 4.18 SPP	53
Tabel 4.19 Rincian SPP	53
Tabel 4.20 Admin	54
Tabel 4.21 RKP Bidang	54
Tabel 4.22 RKP Sub Bidang	55
Tabel 4.23 Rincian Penerimaan Tunai	55
Tabel 4.24 Rincian Penerimaan Bank	56
Tabel 4.25 Objek	56
Tabel 4.26 Jenis	57
Tabel 4.27 Kelompok	57
Tabel 4.28 Kegiatan	58
Tabel 4.29 Data RAB	58
Tabel 4.30 Data Rincian RAB	59
Tabel 4.31 Data Data RAP Rinci	59
Tabel 4.32 Data RKP Dana Indikatif	60
Tabel 5.3 .6 Black Box Testing	92

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Pengelolaan keuangan desa merupakan suatu sub sistem dari pengelolaan keuangan negara dan daerah dalam mendanai penyelenggaraan pemerintahan desa dan pemberdayaan masyarakat desa sehubungan dengan hal tersebut, maka dalam pengelolaan keuangan desa diperlukan mengevaluasi standar yang digunakan dimulai dari aspek perencanaan dan penganggaran maupun aspek pelaksanaan, penatausahaan keuangan desa dan aspek pertanggungjawaban keuangan desa. Proses perencanaan dan penganggaran maupun aspek pelaksanaan, penatausahaan keuangan desa dan aspek pertanggungjawaban merupakan kesempatan yang baik untuk melakukan evaluasi apakah pemerintahan desa melakukan tugasnya secara efektif dan efisien, dengan melakukan hal yang benar terhadap pencapaian tujuan dan sasaran untuk memenuhi kebutuhan masyarakat (Moedarlis, 2016).

Pembuatan aplikasi Administrasi Desa ini di bangun menggunakan HTML, CSS dan Bootstrap sebagai tampilannya. Bahasa pemograman PHP sebagai sistem/ back end dan MYSQL sebagai data base. Hasil dari penelitian ini adalah Aplikasi Administrasi Desa Berbasis *website*, bisa diakses langsung oleh Admin dan Bendahara Desa sebagai efisiensi mengelola data keuangan yang dibutuhkan oleh bendahara desa, fitur yang bisa di akses oleh Admin dan bendahara yaitu penganggaran desa, perencanaan desa, penatausahaan, dan laporan penatausahaan.

Permasalahan yang terjadi di kantor desa bros yaitu pengelolaan data keuangan desa masih menggunakan cara semi manual (*Microsoft Excel*), sehingga banyak kelemahannya yaitu besarnya *human eror* dan ketidak seimbangan (*balance*), dikarenakan banyaknya format yang harus dibuat dalam beberapa lembar kerja. Hal ini menyebabkan pengerjaan laporan keuangan akan membutuhkan waktu lama.

Metodelogi pengembangan sistem yang digunakan yaitu metode air terjun yang dikenal juga dengan istilah metode *waterfall*. Metode *waterfall* adalah metode pengembangan perangkat lunak yang memungkinkan pembuatan sistem dilakukan secara terstruktur dan sistematis (berurutan) sesuai dengan situs pengembangan yang ada. Metode ini disebut *waterfall* atau air terjun karena dalam prosesnya, sistem akan dibuat berurutan setahap demi setahap

Berdasarkan permasalahan tersebut maka dalam penelitian ini penulis akan membangun Aplikasi Administrasi Desa yang dapat mempermudah Admin dan Bendahara Desa dalam mengelola data perencanaan, penganggaran, dan penatausahaan. Bendahara yang ingin mengelola data keuangan, tidak perlu menggunakan *Microsoft excel* lagi, cukup mengakses situs aplikasi administrasi desa, *login* lalu mengisi data pada form yang sudah tersedia pada fitur aplikasi administrasi desa. Kemudian setelah proses peng *input* tan data selesai, bendahara desa bisa menyimpan data pada aplikasi administrasi desa. Apabila bendahara membutuhkan data tersebut, bendahara bisa mencetaknya. Maka dengan ini penulis tertarik melakukan penelitian dengan mengambil judul “Rancang Bangun Aplikasi Administrasi Desa Berbasis *Website* Pada Kantor Desa Baros”.

Kesimpulan dari penelitian ini yaitu penerapan Aplikasi Administrasi Desa bertujuan untuk memudahkan bendahara desa dalam mengelola keuangan desa dengan baik sehingga dapat berdampak pada kesejahteraan masyarakat desa baros. Mulai dari proses perencanaan, penganggaran, penatausahaan sampai dengan pelaporan keuangan desa bisa berjalan dengan baik dengan diterapkannya Aplikasi Administrasi Desa ini.

1.2 Rumusan Masalah

Dari uraian di atas, rumusan masalah yang akan di bahas dalam penelitian ini yaitu:

1. Bagaimana membangun Aplikasi Administrasi Desa Di Kantor Desa Baros yang dapat membantu bendahara desa dalam mengelola keuangan desa?

2. Aplikasi Pengelolaan Keuangan Desa Berbasis *Website* ini dirancang menggunakan Php, MySQL, dan XAMPP.

3. Aplikasi Pengelolaan Keuangan Desa Berbasis *Website* ini dapat menyimpan data keuangan desa.

1.3 Batasan Masalah

Adapun batasan masalah dalam pelaksanaan penelitian dan pembuatan Aplikasi yaitu:

1. Aplikasi Administrasi Desa dapat digunakan untuk menginput data

Perencanaan, Penganggaran, dan Penatausahaan

2. Aplikasi Administrasi Desa Berbasis *Website* ini dirancang menggunakan PHP, MySQL, dan XAMPP.

3. Aplikasi Administrasi Desa Berbasis *Website* ini dapat mengedit dan menyimpan data keuangan desa.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini yaitu sebagai berikut:

1. Mengikuti perkembangan IPTEK untuk memajukan Kantor Desa Baros.

2. Mengubah sistem pencatatan dan pendataan semi manual menjadi berbasis *website*.

3. Mengimplementasikan Aplikasi Administrasi Desa agar dapat diakses oleh bendahara desa, dan kepala desa.

1.5 Metode Pengumpulan Data

Metode pengumpulan data merupakan langkah peneliti yang bertujuan untuk memperoleh data dan mengumpulkannya. Pengumpulan data dalam penelitian kualitatif berdasarkan ketrampilan dan pengetahuan peneliti. Sehingga kredibilitas peneliti sangat dibutuhkan. Ada tiga teknik pengumpulan data yang digunakan dalam penelitian ini yaitu Wawancara, Observasi dan Studi Pustaka.

1.Observasi

Dalam penelitian ini Penulis melakukan observasi langsung ke Kantor Desa Baros untuk mengamati secara langsung mengenai pengelolaan keuangan desa.

2.Wawancara

Dalam penelitian ini penulis menggunakan wawancara bertipe *open-ended*. Peneliti menggunakan wawancara ini dengan maksud agar informan bisa mengungkapkan informasi-informasi yang dia miliki dan dia pahami secara luas dan agar informasi juga bisa memberi saran tentang sumber-sumber bukti lain yang mendukung serta menciptakan akses terhadap sumber yang bersangkutan secara langsung maupun tidak langsung. Peneliti melakukan wawancara kepada Sekertaris Desa Baros untuk menanyakan pertanyaan mengenai pengelolaan keuangan desa.

3.Studi pustaka

Studi pustaka dilakukan dengan cara membaca jurnal penelitian terkait dengan objek penelitian sebagai penunjang dan juga mencari beberapa referensi dari *website*, buku, dll.

1.5.1 Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan yaitu metode air terjun yang dikenal juga dengan istilah metode *waterfall*. Metode waterfall adalah metode pengembangan perangkat lunak yang memungkinkan pembuatan sistem dilakukan secara terstruktur dan sistematis (berurutan) sesuai dengan situs pengembangan yang ada. Metode ini disebut *waterfall* atau air terjun karena dalam prosesnya, sistem akan dibuat berurutan setahap demi setahap mulai dari tahapan:

a.*Requirement*

b.*Design*

c.*Implementation*

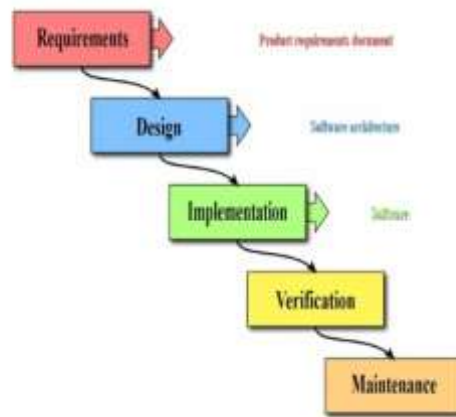
d.*Verification dan*

e.*Maintenance*

Dalam metode waterfall, Jika tahapan 1 belum selesai, maka tahapan 2 tidak bisa berjalan, begitupun seterusnya . Semua tahapan saling berkaitan dengan masing-masing harus di kerjakan secara detail dan terdokumentasai. Metode *waterfall* mengharuskan setiap spesifikasi, persyaratan dan tujuan sistem

terdefiniskan secara detail di tahap awal (*requirement & design*) sebelum masuk pada tahap pengerjaan (implementasi).

Tahapan dari metode *waterfall* yaitu sebagai berikut:



Gambar 1.1 Metode *Waterfall*

Sumber: <https://salamadian.com/metode-waterfall/>

a.Requirement

Pada tahap ini pengembang harus mengetahui seluruh informasi mengenai kebutuhan *software* seperti kegunaan *software* yang diinginkan oleh pengguna dan batasan *software*. Informasi tersebut biasanya di peroleh dari wawancara, survey, ataupun diskusi. Setelah itu informasi dianalisis sehingga mendapatkan data-data yang lengkap mengenai kebutuhan pengguna akan *software* yang akan di kembangkan.

b.Design

Tahap selanjutnya yaitu Desain. Desain dilakukan sebelum proses *coding* dimulai. Ini bertujuan untuk memberikan gambaran lengkap tentang apa yang harus di kerjakan dan bagaimana tampilan dari sebuah sistem yang diinginkan. Sehingga membantu menspesifikasikan kebutuhan *hardware* dan sistem, juga mendefinisikan arsitektur sistem yang akan dibuat secara keseluruhan.

c.Implementation

Proses penulisan *code* ada di tahap ini. Pembuatan software akan di pecah menjadi modul-modul kecil nantinya akan digabungkan pada tahap selanjutnya. Dalam tahap ini juga akan dilakukan pemeriksaan lebih dalam terhadap modul yang sudah dibuat, apakah sudah memenuhi fungsi yang sudah dibuat atau belum.

d.Verification

Tahapan verifikasi meliputi pengintegrasian sistem dan juga melakukan testing terhadap aplikasi yang telah dibuat. Sistem akan diverifikasi untuk diuji sejauh mana kelayakannya. Dalam tahapan ini semua modul yang dikerjakan oleh pogramer berbeda akan digabungkan kemudian diuji apakah telah sesuai dengan spesifikasi yang ditetapkan atau terdapat kesalahan / *error* dalam sistem sebelum kemudian diperbaiki ulang.

e.Maintenance

Maintenance adalah tahapan terakhir dari metode pengembangan waterfall. Disini *software* yang sudah jadi akan dijalankan atau di operasikan oleh pengguna nya. Disamping itu dilakukan pula pemeliharaan yang termasuk:

- 1.Perbaikan kesalahan
- 2.Perbaikan implementasi unit
- 3.Peningkatan jasa sistem sesuai kebutuhan baru

1.5.2 Metode Perancangan

Tahap perancangan merupakan pengembangan dari gambaran umum sistem. Dalam tahap perancangan dijelaskan lebih detail tentang isi dari aplikasi yang dibuat yaitu dengan membuat *Flow chart* dan diagram *Unified Modelling Language* (UML) yang meliputi *use case*, *activity diagram* dan *sequence diagram* serta membuat desain *input* dan *output*. Setelah tahap perancangan selesai maka dilakukan tahap implementasi yaitu menerjemahkan desain ke dalam *source code* berbasis web, yang akan digunakan untuk membuat aplikasi keuangan desa .

1.1 Sistematika Penulisan

1.BAB I PENDAHULUAN

Pada Bagian ini terdiri dari 5 bagian di atas, yaitu latar belakang meliputi penjelasan konseptual tentang apa dan mengapa judul riset tersebut menarik untuk diangkat dan digarap. Rumusan Masalah adalah pertanyaan hasil temuan permasalahan terikat penelitian yang sudah dilakukan. Batasan Masalah digunakan untuk menghindari adanya penyimpangan maupun pelebaran pokok masalah. Tujuan Penelitian dilakukan terkait mengapa riset tersebut dilakukan dan tujuan seperti apa yang ingin dicapai. Metodologi Penelitian adalah metode apa saja yang akan digunakan penulis.

2.BAB II TINJAUAN PUSTAKA

Pada bagian ini terdiri dari landasan teori yaitu penjelasan dalam bentuk literature review atau hasil penelitian yang telah dilakukan dan Dasar Teori yang digunakan untuk menjelaskan tentang definisi konseptual.

3.BAB III METODELOGI PENELITIAN

Pada bagian ini membahas mengenai komponen dari metode Penelitian kemudian mendeskripsikan secara spesifik.

4.BAB IV ANALISIS PERANCANGAN

Pada bagian ini membahas mengenai analisis dimulai dari instrument penelitian, analisis sistem, analisis kebutuhan dan analisis data. Kemudian perancangan aplikasi yang direncanakan untuk membangun aplikasi.

5.BAB V IMPLEMENTASI DAN PENGUJIAN

Pada bab ini dilakukan implementasi dan pengujian. Implementasi berisi uraian penggunaan sistem dari hasil pengembangan hasil penelitian yang merupakan jawaban tegas dan lugas atau rumusan masalah. Pengujian berisi tahapan pengujian yang dilakukan sesuai dengan aplikasi.

6.BAB VI KESIMPULAN DAN SARAN

Bagian ini membahas tentang kesimpulan yang berisi simpulan hasil penelitian yang merupakan jawaban dari rumusan masalah kemudian saran berisi usulan konkrit serta operasional yang merupakan tindak lanjut sumbangan penelitian.

BAB II

TINJAUAN PUSTAKA

2.1 Landasan Teori

Dalam penelitian ini akan digunakan tiga tinjauan studi yang nantinya mendukung penelitian yang akan dilakukan, dimana tinjauan studi yang di ambil sebagai berikut:

1. Menurut Jurnal yang berjudul Peran Sistem Keuangan Desa SISKUDES studi kasus di desa kaba-kaba, kecamatan kecamatan kediri, kabupaten tabanan (Gusti Ayu Trisha Sulina, IMade Arie Wahyuni, 2Putu Sukma Kurniawan) Jurnal : e-Journal S1 Ak Universitas Pendidikan Ganesha Jurusan Akuntansi Program S1 (Vol:8 No:2 Tahun 2017). Pada jurnal ini menjelaskan tentang pengembangan aplikasi sistem keuangan desa (SISKUDES) mengharapkan desa-desa di seluruh Indonesia untuk dapat menerapkan aplikasi (SISKUDES) guna mencegah terjadinya kecurangan-kecurangan yang tidak diinginkan. Desa kabs-kaba yang terletak di kecamatan kediri, kabupaten tambanan ini merupakan salah satu desa yang telah mampu menerapkan sistem keuangan desa (SISKUDES) yaitu jika dalam sistem manual masih terpisah-pisah dalam pelaporannya. Sedangkan pada sistem yang baru (SISKUDES) dapat dilakukan dengan sekali kerja sudah banyak laporan yang muncul, tetapi tetap dalam penginputannya juga dilakukan beberapa kali. Walaupun desa kaba-kaba sudah menggunakan sistem yang baru, sistem lama masih tetap digunakan. Karena dasar-dasar dalam memasukan data ke sistem perlu menggunakan cara manual, untuk menanggulangi kesalahan yang akan terjadi.

2. Menurut Jurnal yang berjudul Penerapan Sistem Keuangan Desa dalam Pengelolaan Keuangan Desa (Arif Rivan1 & Irfan Ridwan Maksun) Jurnal: *Jurnal Administrasi Publik* 9 (2) Desember 2019). Sistem keuangan desa (siskeudes) merupakan suatu aplikasi yang digunakan oleh desa dalam proses penganggaran, penatausahaan serta pelaporan keuangan

desa. Sistem keuangan desa (siskeudes) secara otomatis menghasilkan berbagai laporan yang diperlukan, sehingga dapat menghemat waktu dan biaya, mengurangi potensi kecurangan dan kesalahan, dan membantu agregasi data. Selain berbasis online, sistem keuangan desa juga disiapkan secara offline atau manual, mengingat kemampuan dari sumber daya yang ada di desa serta kondisi di masing-masing daerah yang berbeda-beda. Pelaksanaan sistem keuangan desa (siskeudes) mengacu pada Peraturan Menteri Dalam Negeri nomor 20 Tahun 2018 tentang pengelolaan keuangan desa. Fitur-fitur yang ada pada Sistem Keuangan Desa (Siskeudes) dibuat sederhana, *user friendly*, built-in internal control, didukung dengan petunjuk pelaksanaan, memudahkan tatakelola keuangan desa, kesesuaian dengan aturan yang berlaku, dan menatausahakan seluruh anggaran yang dikelola oleh desa. Kemudian output yang dihasilkan dari Sistem Keuangan Desa (Siskeudes) antara lain yaitu Rencana Pembangunan Jangka Menengah Desa (RPJMDesa) dan Rencana Kerja Pemerintah Desa (RKPDesa), Anggaran Pendapatan dan Belanja Desa (APBDesa), Dokumen penatausahaan Keuangan Desa, Laporan realisasi APBDesa, Laporan kekayaan milik Desa, Laporan realisasi per sumber Dana serta Laporan kompilasi ditingkat Pemerintah Daerah.

3. Menurut Jurnal yang berjudul Analisis Aplikasi Sistem Keuangan Desa (SISKEUDES) versi 2.0 Studi kasus di desa Lebaksari Kecamatan Parakansalak Kabupaten Sukabumi. (Febi Nugraha, Sifa, Deni Iskandar, Idang Nourdin). Jurnal: Jurnal Ilmiah Mahasiswa Ekonoi Volume 03 Nomor 02 Tahun 2020. Untuk membantu pemerintah desa dalam pelaporan keuangan maka pemerintah pusat melalui Badan Pengawas Keuangan dan Pembangunan (BPKP) meluncurkan Aplikasi Sistem Keuangan Desa (SISKEUDES), untuk digunakan oleh semua desa yang ada di Indonesia agar pelaporan keuangan desa mempunyai standar yang sama dan semua desa dapat mengelola keuangan dengan mandiri dan dapat meningkatkan akuntabilitas dan transparansi pengelolaan keuangan desa, dengan meningkatnya hal tersebut maka tata kelola keuangan desa

akan terlaksana dengan efektif dan efisien guna terciptanya pemerintahan desa yang baik (*good governance*).

2.2 Dasar Teori

2.2.1 Keuangan Desa

Menurut (“Widjaja,” 2005) dalam buku otonomi desa otonomi Desa 2005:133) keuangan Desa adalah pemerintah Desa menekankan pada prinsip-prinsip demokrasi , peran serta masyarakat, pemerataan dan keadilan serta memperhatikan potensi dan keanekaragaman daerah. Desa memiliki posisi yang sangat strategis, sehingga diperlukan adanya perhatian yang seimbang terhadap penyelenggaraan otonomi daerah. Menurut (Nurcholis, 2018) bahwa keuangan Desa adalah semua hak dan kewajiban dalam rangka penyelenggaraan pemerintah Desa yang dapat dinilai dengan uang, termasuk didalamnya segala bentuk kekayaan yang berhubungan dengan hak dan kewajiban Desa tersebut. (Rahum, 2015).

Menurut (Indriana Nasari 2017 (Nasari, 2017) Siklus pengelolaan keuangan Desa sesuai dengannilai-nilai pengeloaan keuangan Desa sebagai berikut:

a.Perencanaan

Perencanaan adalah sebagai perhitungan dan penentuan tentang apa yang dijalankan dalam rangka mencapai tujuan tertentu dimana menyangkut tempat, oleh siapa pelaku itu atau pelaksanaan tata cara mencapai tujuan tersebut. Dari pernyataan tersebut perencanaan dapat diartikan sebagai pemilihan sekumpulan kegiatan dan pemusatan selanjutnya apa yang harus dilakukan, kapan, bagaimana, dan oleh siapa.

b.Pelaksanaan dan penatausahaan

Pelaksanaan dan penatausahaan anggaran pendapatan dan belanja Desa terdiri dari:

- 1.Prinsip pelaksanaan keuangan Desa dalam pelaksanaan keuangan Desa, terdapat beberapa prinsip umum yang harus ditaati yang mencakup penerimaan dan pengeluaran. Prinsip itu diantaranya bahwa seluruh penerimaan dan pengeluaran Desa dilaksanakan

melalui rekening kas Desa. pencairan dana dalam rekening kas Desa. Pencairan dalam rekening kas Desa ditandatangani oleh kepala Desa dan bendahara Desa. namun khusus bagi Desa yang belum memiliki pelayanan perbankan.

2. Pelaksanaan penerimaan pendapatan yaitu proses menerima dan mencatat pendapatan Desa. pendapatan Desa yang bersifat pendapatan asli Desa berasal dari masyarakat dan lingkungan Desa, sedangkan pendapatan transfer berasal dari pemerintah supra Desa. pihak yang terkait dalam proses penerimaan pendapatan adalah pemberi dana (pemerintah pusat/Provinsi/Kabupaten/Kota, masyarakat, pihak ketiga), penerima dana (bendahara Desa/pelaksana kegiatan/kepala Dusun) dan bank.

3. Pelaksanaan pengeluaran/belanja

Belanja Desa diprioritaskan untuk memenuhi kebutuhan pembangunan yang disepakati dalam musyawarah desa dan sesuai dengan prioritas pemerintah baik pemerintah pusat maupun pemerintah Provinsi/Kabupaten/Kota. Hal tersebut seluruhnya tertuang dalam rencana kerja pemerintah Desa yang pelaksanaannya akan diwujudkan melalui anggaran pendapatan dan belanja desa. setelah anggaran pendapatan dan belanja Desa ditetapkan dalam bentuk peraturan Desa, program dan kegiatan sebagaimana yang telah direncanakan baru dapat dilaksanakan. Hal ini dikecualikan untuk belanja pegawai yang bersifat mengikat dan operasional perkantoran yang diatur dalam keputusan kepala Desa. dengan adanya ketentuan dari kepala Desa tersebut, maka belanja pegawai dan operasional dapat dilakukan tanpa perlu menunggu penetapan anggaran pendapatan dan belanja Desa. pelaksanaan anggaran pendapatan dan belanja Desa dilakukan sesuai dengan kewenangan yang dimiliki oleh Desa berdasarkan ketentuan yang berlaku.

4. Pelaksanaan pembiayaan

Pelaksanaan pembiayaan mencakup penerimaan pembiayaan dan pengeluaran pembiayaan.

5. Penerimaan pembiayaan

a. Penerimaan pembiayaan mencakup SILPA tahun sebelumnya, pencairan dana cadangan dan hasil penjualan kekayaan Desa yang dipisahkan. Sisa lebih perhitungan anggaran adalah penerimaan pembiayaan yang digunakan untuk mendanai pelaksanaan kegiatan tahun berjalan yang berasal dari pelampauan penerimaan pendapatan dan penghematan belanja tahun sebelumnya. Pencairan dana cadangan merupakan kegiatan pencairan dana dari rekening dan cadangan ke rekening Desa yang dilakukan sesuai peraturan Desa yang mengatur hal tersebut. Sedangkan penerimaan pembiayaan yang berasal dari hasil penjualan kekayaan Desa yang dipisahkan diperoleh dari realisasi penjualan aset/kekayaan Desa kepada pihak ketiga.

b. Pengeluaran pembiayaan

Pengeluaran pembiayaan diantaranya pembentukan dana cadangan dan penyertaan modal Desa. bendahara Desa wajib melakukan pencatatan terhadap seluruh transaksi yang ada berupa penerimaan dan pengeluaran. Bendahara Desa melakukan pencatatan secara sistematis dan kronologis atas transaksi-transaksi keuangan yang terjadi. Penatausahaan keuangan Desa yang dilakukan oleh bendahara Desa dilakukan dengan cara sederhana, yaitu berupa pembukuan dan kronologis atas transaksi-transaksi keuangan yang terjadi.

c. Penatausahaan keuangan

Desa yang dilakukan oleh bendahara Desa dilakukan dengan cara sederhana, yaitu berupa pembukuan belum menggunakan jurnal akuntansi. Penata usahaan baik penerimaan kas maupun pengeluaran kas, bendahara Desa menggunakan:

1. Buku kas umum
2. Buku kas pembantu pajak dan
3. Buku bank.
4. Pelaporan dan pertanggungjawaban

Dalam melaksanakan kewenangan dan kewajibannya dalam pengelolaan keuangan Desa, kepala Desa memiliki kewajiban untuk menyampaikan laporan. Laporan tersebut bersifat periodik semesteran dan tahunan, yang disampaikan ke bupati/walikota dan ada juga yang disampaikan ke BPD.

Rincian laporan sebagai berikut:

1. Laporan semesteran realisasi pelaksanaan anggaran pendapatan dan belanja Desa. laporan realisasi pelaksanaan APB Desa semester pertama menggambarkan realisasi pendapatan, belanja dan pembiayaan selama semester 1 dibandingkan dengan target dan anggarannya, sedangkan laporan realisasi pelaksanaan anggaran pendapatan dan belanja Desa semester akhir tahun menggambarkan realisasi pendapatan, belanja dan pembiayaan sampai dengan akhir tahun, jadi bersifat akumulasi hingga akhir tahun anggaran.
2. Laporan pertanggung jawaban realisasi pelaksanaan anggaran pendapatan dan belanja desa kepada bupati/walikota setiap akhir tahun anggaran. Laporan pertanggungjawaban realisasi pelaksanaan APB Desa setiap akhir tahun anggaran disampaikan kepada bupati/walikota melalui camat terdiri dari pendapatan, belanja, dan pembiayaan yang telah ditetapkan dengan peraturan Desa. setelah pemerintah Desa dan BPD telah sepakat terhadap laporan pertanggung jawaban realisasi pelaksanaan APB Desa dalam bentuk peraturan Desa, maka PERDES ini disampaikan kepada bupati/walikota sebagai bagian tidak terpisahkan dari laporan penyelenggaraan pemerintahan Desa. laporan pertanggung jawaban realisasi

pelaksanaan APB Desa sebagai mana tercantum dalam pada pasal 41 Perm(Pemendagri, 2014) disampaikan paling lambat 1 (satu) bulan setelah tahun anggaran berkenaan.

- a. Untuk semester 1 paling lambat minggu keempat bulan juli tahun anggaran berjalan
- b..Untuk semester II paling lambat minggu keempat bulan januari tahun anggaran berikutnya.

Menurut ahli tersebut, fungsi manajemen pada dasarnya mengacu kepada inti permasalahan dan tujuan yang sama, yaitu dimaksudkan agar mekanisme manajemen dalam melaksanakan berbagai kegiatan dapat mencapai tujuan yang telah ditetapkan dengan berdaya guna, berhasil guna tepat guna. Apabila manajemen atau pengelolaan dikaitkan dengan keuangan atau anggaran maka pengelolaan keuangan dapat didefinisikan sebagai proses atau cara mengendalikan, mengatur, menyelenggarakan, mengurus dan menjalankan keuangan dan anggaran.

2.2.2 Manajemen Keuangan

Menurt (Sartono, 2011) istilah manajemen keuangan dapat di artikan sebagai manajemen dana baik yang berkaitan dengan pengalokasian dana dalam berbagai bentuk investasi secara efektif maupun usaha pengumpulan dana untuk pembiayaan investasi atau pembelanjaan secara efisien. sedangkan Menurut (Sutrisno, 2003)Manajemen keuangan adalah sebagai semua aktifitas perusahaan dengan usaha-usaha mendapatkan dana perusahaan dengan biaya yang murah serta usaha untuk menggunakan dan mengalokasikan dana tersebut secara efisien. Menurut (Sonny, 2009) Manajemen keuangan adalah aktivitas perusahaan yang berhubungan dengan bagaimana memperoleh dana, menggunakan dana, dan mengelola asset sesuai dengan tujuan perusahaan secara menyeluruh.

2.2.3 Aplikasi Keuangan

Aplikasi keuangan adalah sebuah pogram computer yang digunakan untuk memanajemen keuangan, baik untuk keperluan pribadi, bisnis, intansi dan perusahaan. Dengan menggunakan aplikasi keuangan dapat memmbantu alur transaksi keuangan, sehingga apabila terjadi keuntungan atau kerugian akan

cepat diketahui.

Adapun beberapa fungsi aplikasi keuangan, diantaranya sebagai berikut:

1. Mempercepat Pekerjaan

Sebelum adanya aplikasi keuangan baik untuk PC maupun smartphone, proses pembuatan laporan keuangan dibuat secara manual dan bertahap, sehingga membutuhkan waktu yang cukup lama. Dengan adanya aplikasi keuangan proses pembuatan laporan keuangan hanya dilakukan dalam satu kali proses saja, sehingga waktu yang dibutuhkan jauh lebih cepat dan lebih akurat.

2. Akurasi perhitungan keuangan sangat diperlukan, supaya dalam membuat laporan tidak terjadi kesalahan. Dengan adanya aplikasi keuangan, pembuatan laporan keuangan dapat dikerjakan secara akurat dan lebih cepat.

3. Transparan

Fungsi aplikasi keuangan yang tidak kalah pentingnya adalah transparansi, dengan adanya aplikasi keuangan laporan keuangan dapat ditampilkan lebih detail sehingga meminimalisir adanya kecurangan.

4. Aman

Apabila anda menggunakan aplikasi keuangan untuk membuat laporan keuangan di perusahaan atau instansi anda, maka keamanan data akan lebih terjaga. Hal ini dikarenakan aplikasi keuangan dirancang untuk melindungi data anda yang disimpan di komputer.

5. *Flexibel*

Selain itu aplikasi keuangan juga dirancang untuk digunakan secara *flexible*, jadi anda dapat mengerjakan laporan keuangan kapan saja dan dimana saja terlebih apabila aplikasi tersebut berbasis *website*, anda tentunya akan lebih dipermudah dalam menggunakannya meskipun menggunakan tim khusus.

Adapun beberapa aplikasi yang sering digunakan dalam mengelola keuangan yaitu sebagai berikut:

1. MYOB adalah *software* akuntansi yang cukup umum dan banyak digunakan di Indonesia karena relative mudah diaplikasikan. MYOB bisa digunakan oleh perusahaan mana pun, termasuk perusahaan-perusahaan kecil seperti home industry yang membutuhkan pencatatan aliran keuangan (pemasukan dan pengeluaran). Sebagai *software* akuntansi, MYOB memiliki beberapa

kelebihan. Misalnya MYOB mampu melakukan Analisa bisnis , MYOB bisa digunakan untuk membuat laporan yang menyeluruh dan bisa diubah kedalam bentuk file MS.Excel , MYOB juga bisa mendeteksi kesalahan melalui company Data Auditor .

2.Accurate 5

Accurate accounting software saat ini telah banyak digunakan berkat keunggulannya dalam menyelesaikan berbagai macam tugas akuntansi.

3.Bee Accounting

Bee accounting adalah *software* yang saat ini telah banyak digunakan berkat keunggulannya dalam menyelesaikan berbagai macam tugas akuntansi.

2.2.4 Aplikasi

Aplikasi adalah sebuah perangkat lunak yang menjadi *front end* dalam sebuah sistem yang digunakan untuk mengolah data menjadi suatu informasi yang berguna orang-orang dan sistem yang bersangkutan (Basara, 2014) Basara 2014). Istilah aplikasi berasal dari bahasa Inggris *application* yang berarti penerapan, lamaran ataupun penggunaan. Sedangkan secara istilah, pengertian aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju. Menurut kamus komputer eksekutif, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan. Adapun definisi Aplikasi menurut para ahli:

a.Menurut Rachmad Hakim

Aplikasi merupakan perangkat lunak yang digunakan untuk tujuan tertentu, seperti mengolah dokumen, mengatur Windows &, permainan (*game*), dan sebagainya. Beberapa aplikasi yang digabung bersama menjadi suatu paket disebut sebagai suatu paket atau *application suite*. Aplikasi- aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan tiap aplikasi.

b. Menurut Ali Zaki & Smitdev Community

Aplikasi adalah komponen yang berguna melakukan pengolahan data maupun kegiatan-kegiatan seperti pembuatan dokumen atau pengolahan data. Aplikasi adalah bagian PC yang berinteraksi langsung dengan *user*. Aplikasi berjalan di atas sistem operasi, sehingga agar aplikasi bisa diaktifkan, kita perlu melakukan instalasi sistem operasi terlebih dahulu.

a. Menurut R. Eko. I & Djokopran

Aplikasi merupakan proses atau prosedur aliran data dalam infrastruktur teknologi informasi yang dapat dimanfaatkan oleh para pengambil keputusan yang sesuai dengan jenjang dan kebutuhan (relevan).

b. Menurut Hengky W. Pramana

Aplikasi adalah satu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas seperti sistem perniagaan, *game*, pelayanan masyarakat, periklanan, atau semua proses yang hampir dilakukan manusia.

2.2.5 Website

Website adalah kumpulan halaman web yang saling terhubung dan file-filenya saling terkait. Web terdiri dari page atau halaman, dan kumpulan halaman yang dinamakan *homepage*. *Homepage* berada pada posisi teratas, dengan halaman-halaman terkait berada di bawahnya. Biasanya setiap halaman di bawah *homepage* disebut *child page*, yang berisi *hyperlink* ke halaman lain dalam web. Penemu website adalah Sir Timothy John “ Tim ” Berners – Lee. pada tahun 1991. Maksud dari Tim membuat website adalah untuk mempermudah tukar menukar dan memperbarui informasi kepada sesama peneliti di tempatnya bekerja. Pada tanggal 30 April 1993, *CERN* (tempat dimana Tim bekerja) menginformasikan bahwa *WWW* dapat digunakan secara gratis oleh semua orang. Sebuah website bisa berupa hasil kerja dari perorangan atau individu, atau menunjukkan kepemilikan dari sebuah organisasi, perusahaan, dan biasanya website itu menunjukkan beberapa topik khusus dan kepentingan tertentu. Sebuah *website* dapat berisi *hyperlink* (pranala) yang menghubungkan ke website lain, jadi, terkadang perbedaan antar website yang dibuat oleh individu perseorangan dengan website yang dibuat oleh organisasi bisnis bisa saja tidak terlalu terlihat. *Website* ditulis, atau

secara dinamik di konversi menjadi *HTML* dan diakses melalui sebuah program *software* yang biasa disebut dengan *web browser*, yang dikenal juga dengan dengan server *web*, atau yang disebut *HTTP Server*, dan pengertian ini dapat menunjuk pada *software* yang dipakai untuk menjalankan sistem ini, yang kemudian menerima lalu mengirimkan halaman-halaman yang diperlukan untuk merespon permintaan dari pengguna. *Apache* adalah piranti lunak yang biasa digunakan dalam sebuah *webserver*, kemudian setelah itu adalah *Microsoft Internet InformationServices (IIS)*.

Secara umum, **Website** atau **Web** adalah sekumpulan halaman yang terdiri dari beberapa laman dan berisi tentang informasi dalam bentuk digital baik itu tulisan (teks), gambar animasi yang disediakan melalui internet sehingga dapat diakses oleh banyak orang diseluruh dunia yang memiliki koneksi internet. Maka disimpulkan, **website** adalah halaman web atau situs yang saling berhubungan oleh perorangan, kelompok atau organisasi. Sebuah situs web biasanya ditempatkan pada sebuah *server web* yang dapat diakses melalui jaringan seperti Internet atau jaringan area lokal (*LAN*) melalui alamat internet yang dikenal dengan *URL*. Kombinasi dari semua situs yang dapat diakses publik di internet juga dikenal dengan *World Wide Web* atau disingkat *www*.

2.2.6 Mysql

MySQL merupakan database yang pertama kali didukung oleh bahasa pemrograman *script* untuk internet (PHP dan Perl). MySQL dan PHP dianggap sebagai pasangan *software* pembangun aplikasi web yang ideal. Kepopuleran MySQL antara lain karena MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses databasenya sehingga mudah untuk digunakan, kinerja query cepat, dan mencukupi untuk kebutuhan database perusahaan-perusahaan yang berskala kecil sampai menengah, MySQL juga bersifat *open source* (tidak berbayar). MySQL dikembangkan oleh perusahaan swedia bernama MySQL AB yang pada saat ini bernama Tcx DataKonsult AB sekitar tahun 1994-1995, namun cikal bakal kodenya sudah ada sejak tahun 1979. Awalnya Tcx merupakan perusahaan pengembang software dan konsultan *database*, dan saat ini MySQL sudah diambil alih oleh *Oracle Corp*. MySQL didistribusikan dengan licensi open source GPL (*General Public License*)

namun tidak boleh menjadikan produk turunan yang bersifat komersial. (Arief, M.Rudianto. 2011.) SQL dapat digunakan secara berdiri sendiri maupun di lekatkan pada bahasa pemrograman seperti C, dan Delphi. Pernyataan SQL dapat dikelompokkan menjadi beberapa bagian yaitu :

1.DDL

DDL berfungsi untuk mendefinisikan atribut basis data, tabel, atribut (kolom), batasan-batasan terhadap suatu atribut, serta hubungan antar tabel. Perintah yang digunakan biasanya : *create*, *alter*, dan *drop*.

2.DML

DML berfungsi untuk memanipulasi data yang ada di dalam *database*, contohnya untuk pengambilan data, penyisipan data, pengubahan data dan penghapusan data. Adapun keunggulan dari mysql adalah sebagai berikut:

- a.Program yang *multi-threaded*, sehingga dapat dipasang pada *server* yang memiliki *mulit-CPU*
- b.Didukung bahasa pemrograman umum seperti *C*, *C++*, *Java*, *Perl*, *PHP*, *Python*, *TCL*, *APls* dls.
- c.Bekerja pada berbagai *platform*
- d.Memiliki jenis kolom yang cukup banyak sehingga memudahkan konfigurasi sistem *database*.
- e.Memiliki sistem keamanan yang cukup baik dengan verifikasi *host*.
- f.Mendukung ODBC untuk OS *Microsoft Windows*.
- g.Mendukung *record* yang memiliki kolom dengan panjang tetap. Saling terintegrasi dengan PHP.

2.2.7 HTML

Hypertext Markup language (HTML) merupakan suatu bahasa markah yang digunakan untuk membuat halaman web, menampilkan berbagai informasi di dalam sebuah *browser* web. HTML dibuat dengan menggunakan aplikasi pengolah kata dan disimpan pada format *SCII* sehingga menjadi halaman web dengan perintah-perintah HTML. HTML merupakan standar internet yang di definisikan dan dikendalikan penggunaannya oleh *World Wide Web Consortium* (W3C). HTML dibuat oleh kolaborasi cillau TIM yang diketuai oleh Breners lee

Robert ketika bekerja di CERN pada tahun 1989. Secara umum fungsi HTML adalah untuk mengelola serangkaian data dan informasi sehingga suatu dokumen dapat di akses dan ditampilkan di internet melalui layanan web. Secara umum fungsi HTML adalah untuk mengelola serangkaian data dan informasi sehingga suatu dokumen dapat di akses dan ditampilkan di internet melalui layanan web. Fungsi HTML yang lebih spesifik yaitu:

1. Membuat halaman web
2. Menampilkan berbagai informasi didalam sebuah browser internet
3. Membuat link menuju halaman web lain dengan kode tertentu (*hypertext*).

2.2.8 Pengertian CSS

CSS adalah bahasa *Cascading Style Sheet* dan biasanya digunakan untuk mengatur tampilan elemen yang tertulis dalam bahasa markup, seperti HTML. CSS berfungsi untuk memisahkan konten dari tampilan visualnya di situs. CSS dibuat dan dikembangkan oleh W3C (*World Wide Web Consortium*) pada tahun 1996 untuk alasan yang sederhana. Dulu HTML tidak dilengkapi dengan tags yang berfungsi untuk memformat halaman. Anda hanya perlu menulis markup untuk situs. HTML dan CSS memiliki keterkaitan yang erat. Karena HTML adalah bahasa markup fondasi situs) dan CSS memperbaiki style (untuk semua aspek) yang terkait dengan tampilan website, maka kedua bahasa pemrograman ini harus berjalan beriringan.

2.2.9 Java Script

Java script adalah bahasa pemrograman tingkat tinggi dan dinamis. Java script populer di internet dan dapat bekerja di sebagian besar penjelajah web populer seperti *Google Chrome*, *Internet Explorer*, *Netscape*, dan *Opera*. Kode java script dapat disisipkan dalam halaman web menggunakan tag script. Bahasa pemrograman ini merupakan salah satu yang paling utama bagi web developer di antaranya :

1. HTML memungkinkan anda menambahkan konten kedalam web
2. CSS : menentukan lay out, style, serta keselarasan *website*.
3. *Java Script*: menyempurnakan tampilan dan sistem halaman web.

Cara kerja java script adalah pengunjung situs akan diarahkan melalui file js yang terpisah atau embed langsung ke halaman *website*. Jadi bisa diartikan sebagai bahasa dari sisi klien. Nanti klien akan mengakses *website* lalu script tersebut akan terunduh di perangkat klien. *Script* tersebut juga di proses di perangkat yang di maksud. Jadi pemrosesannya bukan pada *web server* , tapi pada perangkat klien atau pengunjung *website*.

2.2.10 J query

J Query adalah library javascript yang cukup handal, praktis dan memiliki fitur-fitur yang lengkap. JQuery bisa digunakan untuk pemrosesan di HTML, mulai dari melakukan perubahan manipulasi dokumen, membuat animasi, membuat event handling dan dapat melakukan ajax dengan lebih sederhana. Penggunaan JQuery telah didukung dengan API, sehingga dapat bekerja dan dapat digunakan pada berbagai macam jenis browser. Menurut (Sibero, 2011) JQuery adalah salah satu javascript dikembangkan oleh John Resig pada tahun 2006 di Bar Camp NYC. Pada awal perkembangannya, JQuery pertama dibuat untuk meringkas penggunaan CSS Selector dalam suatu pustaka fungsi. JQuery memiliki ciri khas pada penggunaan perintahnya, prefix untuk JQuery dengan tanda \$ kemudian dilanjutkan dengan fungsi atau perintah.

Adapun beberapa keunggulan JQuery yaitu sebagai berikut:

1. JQuery bisa berinteraksi baik dengan berbagai macam tipe bahasa pemrograman
2. Didukung oleh berbagai macam plugin
3. JQuery dapat membuat animasi dasar dengan sangat mudah
4. Kompatibel dengan semua jenis CSS
5. Memudahkan untuk mengakses elemen-elemen HTML
6. Open source/free dengan lisensi dari GNU (General Public Licence) dan MIT licence.

2.2.11 AJAX

AJAX adalah kependekan dari Asynchronous JavaScript and XML, dalam bahasa indonesia asinkron antara Javascript dan XML. Pengertian mudahnya menggabungkan antara Java script dan XML untuk mengakses sumber data di server , Jadi server tidak diakses secara langsung, biarkan mesin AJAX yang mengaksesnya. Javascript sebagai pemrograman di sisi client (artinya program

yang dibuat dengan Javascript , bisa dijalankan tanpa menggunakan server) sekarang ini bisa digunakan untuk mengakses server secara asinkron (di belakang layar, artinya proses akses tidak terlihat oleh user). Secara sederhana, AJAX memungkinkan kita untuk berkomunikasi dengan kode program pada server melalui JavaScript.

Berikut adalah beberapa pengertian AJAX menurut para ahli:

1. Menurut (Desrizal, 2014) seorang yang telah melakukan implementasi membuat sebuah request dengan XML ke server, menggunakan AJAX adalah singkatan dari Asynchronous JavaScript and XML . AJAX, terdiri dari HTML, JavaScript, DHTML dan DOM yang kemudian digabungkan dengan bahasa pemrograman web di sisi server seperti PHP dan ASP, sehingga membentuk suatu aplikasi berbasis web yang interaktif.
2. Menurut Lukmanul Hakim (Hakim, 2011) ‘‘AJAX merupakan singkatan dari Asynchronous Javascript and XML. Secara sederhana, AJAX memungkinkan kita untuk berkomunikasi dengan kode program pada server melalui JavaScript.

2.2.12 Php

PHP Hypertext Processor adalah bahasa skrip yang dapat ditanamkan atau disisipkan kedalam HTML. Php banyak dipakai untuk memprogram itu web dinamis. PHP dapat digunakan untuk membangun sebuah CMS. Manfaat PHP pada sitem ERP ini adalah PHP mudah digunakan untuk membangun sitem ERP. Berikut adalah beberapa pengertian PHP menurut para ahli:

1. Menurut Oktavian (Oktavian, 2010) & Diar puji (Puji,2010) yang merupakan penulis buku tentang @ Menjadi Pogramer Jempolan Menggunakan PHP’’ menjelaskan PHP adalah akronim dari Hypertext Preproceso, yaitu suatu bahasa pemrograman berbasiskan kode-kode (script) yang digunakan untuk mengolah suatu data dan mengirimkannya kembali ke web broser menjadi kode HTML.
2. Menurut Kustiyahningsih (Kustiahningsih, 2011) dan Anamisa D.R (Anamisa,2011) merupakan penulis buku tentang ‘‘Pemograman Basis Data Berbasis Web menggunakan PHP & Mysql’’ menjelaskan bahwa PHP (atau resminya PHP :Hypertext Preproceso) adalah skrip bersifat server-side yang ditambahkan kedalam HTML.

3. Menurut Tri Pujadi (Pujadi, 2010) seorang yang telah membuat prototipe E-Klinik untuk Pelayanan Kesehatan Berbasis Pertanian, menjelaskan bahwa PHP adalah bahasa pemrograman script yang paling banyak dipakai saat ini. PHP banyak dipakai untuk memogram situs web dinamis, walaupun tidak tertutup kemungkinan digunakan untuk pemakaian lain. PHP dibuat oleh Rasmus Lerdorf dan Zend yang wujudnya berupa sekumpulan script yang digunakan untuk mengolah data form dari web.

Dibandingkan bahasa pemrograman lain Kelebihan PHP menurut (sehur,2012) seorang penulis buku ‘‘PHP 5 Konsep Implementasi’’ yang telah mengimplementasikan konsep PHP 5 pada pemogramannya, antara lain:

1. Bahasa pemrograman PHP adalah sebuah bahasa script yang tidak perlu melakukan sebuah kompilasi dalam penggunaannya.
2. Banyak Web Server yang mendukung PHP dengan konfigurasi yang relative mudah seperti IIS, Apache, dan yang lainnya.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis-milis dan pengembang-pengembang lain yang siap membantu dalam pengembangan.
4. Dalam sisi pemahaman, PHP adalah bahasa scripting yang paling mudah karena tersedia referensi yang cukup banyak.
5. PHP adalah bahasa open source yang dapat digunakan di berbagai mesin (linux, unix, windows) dan dapat dijalankan secara runtime melalui console serta juga dapat menjalankan perintah-perintah sistem.

2.2.13 Morris JS

Morris Js adalah Javascript library untuk membuat grafik (chart) dalam dokumen html untuk grafik jenis laini (seperti: garis, pie) dapat dicoba dengan cara yang sama.

2.2.14 XAMPP

XAMPP adalah sebuah paket perangkat lunak bebas yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri, yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Dalam prakteknya, XAMPP bisa digunakan untuk menguji kinerja fitur ataupun menampilkan konten yang ada didalam website

kepada oranglain tanpa harus terkoneksi dengan internet, cukup akses melauai xampp control panel, atau istilahnya *website offline*.

2.2.15 Bootstrap

Bootstrap adalah sebuah alat bantu untuk membuat sebuah tampilan halaman website yang dapat mempercepat pekerjaan seorang pengembang website atau pun pendesain halaman website. Sesuai namanya, *website* yang dibuat dengan alat bantu ini memiliki tampilan halaman yang sama tau mirip dengan tampilan halaman Twitter atau desainer juga mengubah tampilan halaman website sesuai dengan kebutuhan.

Berikut adalah beberapa pengertian bootstrap menurut para ahli:

a. Menurut Jubilee Enterprise penulis dari buku ''Pemograman Bootstrap untuk Pemula'' yang telah mencoba mendesain tampilan website, mengatakan Bootstrap adalah sebuah alat bantu untuk membuat sebuah tampilan halaman website yang dapat mempercepat pekerjaan seorang pengembang *website* yang dapat mempercepat pekerjaan seorang pengembang *website* ataupun pendesain halaman website. Sesuai namanya *website* yang dibuat dengan alat bantu ini memiliki tampilan halaman yang sama/mirip dengan tampilan halaman Twitter atau desainer juga dapat mengubah tampilan halaman *website* sesuai dengan kebutuhan. (Enterprise, 2015).

b. Menurut Handoko seorang yang telah melakukan pengolahan sebuah database dengan XML pada jurnalnya mengatakan bahasa markup yang selama ini telah mengalami beberapa perubahan. Dari SGML (*Standardized General Markup Laguange*) yang merupakan bahasa markup pertama untuk halaman web, kemudian berubah menjadi HTML (*Hyper Tekt Markup Language*) yang mendukung multimedia dan beberapa scrpt. (Handoko, 2015)

2.2.16 Sublime Text

Sublime Text adalah aplikasi editor untuk kode dan teks yang dapat berjalan diberbagai platform *operating system* dengan menggunakan teknologi Phyton APL terciptanya aplikasi ini terinspirasi dari aplikasi Vim. Aplikasi ini sangatlah leksibel dan *powerfull*. Fungsionalitas dari aplikasi ini dapat dikembangkan dengan menggunakan sublime packages. Sublime text bukanlah aplikasi *open source* dan juga aplikasi yang dapat digunakan dan didapatkan secara gratis, akan

tetapi beberapa fitur pengembangan fungsionalitas (packages) dari aplikasi ini merupakan hasil dari temuan dan mendapat dukungan penuh dari komunitas serta memiliki lisensi aplikasi gratis.

2.2.17 Web Browser

Web Browser merupakan aplikasi yang bisa menjelajahi, menyajikan, maupun mengambil konten yang ada di berbagai sumber informasi pada jaringan internet atau WWW. Pengertian dari web browser juga sering disebut dengan suatu perangkat lunak dengan fungsi yang dimilikinya sebagai penerima, mengakses, penyaji berbagai informasi dari internet.

Adapun pengertian web browser menurut para ahli yaitu sebagai berikut:

- a. Menurut Lia Kuwayatno, web browser adalah perangkat lunak yang fungsinya untuk membuka website dengan cara menuliskan alamat situs yang dituju pada kotak adres.
- b. Menurut Smitdev Pahala web browser merupakan laman kata dari server yang tugasnya untuk menampilkan halaman website.
- c. Menurut Ali zaki webbrowser adalah sebuah pogram yang dijalankan untuk mengakses website dengan komputer dan jaringan internet, guna menjadi sebuah sambungan yang bisa di operasikan.
- d. Menurut Rachmad hakim web browser merupakan aplikasi untuk menjelajah di internet yang fungsinya guna menampilkan halaman web dengan dokumen yang disediakan oleh server.
- e. Menurut Yuhevizar web browser adalah metode untuk menampilkan informasi di internet yang berupa gambar, teks, atau suara guna menghubungkan satu dokumen dengan dokumen yang lain.
- f. Menurut Gustav Pahala Frans pengertian web browser adalah aplikasi yang digunakan untuk mengoperasikan internet sebagai media browsing, serta menjelajahi dunia maya.

2.2.18 Unified Modeling Language (UML)




“*Unified Modeling Language (UML)* adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek” (Salahudin, 2013).



a. Use Case Diagram

“Use case diagram merupakan pemodelan untuk melakukan (behavior) sistem informasi yang akan dibuat. Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat secara kasar yang digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu” (Rosa dan Shalahuddin, 2018:155).

Simbol Simbol yang digunakan pada use case diagram bisa dilihat pada tabel 1.

Table 1 : Simbol Use Case Diagram

NO	Simbol	Nama	Keterangan
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan use case.
2		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
3		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
4		<i>Extend</i>	Menspesifikasikan bahwa use case target memperluas perilaku dari use



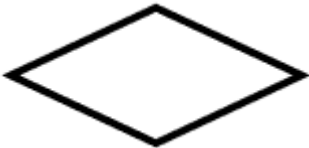

			case sumber pada suatu titik yang diberikan
5		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
6		<i>Include</i>	Menspesifikasikan bahwa use case sumber secara eksplisit.


a. Activity Diagram

“Activity Diagram menggambarkan workflow 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 aktivitas yang dapat dilakukan oleh sistem” (Rosa dan Shalahuddin, 2018:161).

Simbol Simbol yang digunakan pada Activty diagram bisa dilihat pada tabel 2.

Table 2 : Simbol *Activity Diagram*

NO	Simbol	Nama	Keterangan
1		Status Awal	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
2		Aktivitas	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
3		Percabangan	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
4		Penggabungan	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu

5		Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
---	---	--------------	---

b. Sequence Diagram

“*Sequence Diagram* menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada use case”. (Rosa dan Shalahuddin, 2018:165)

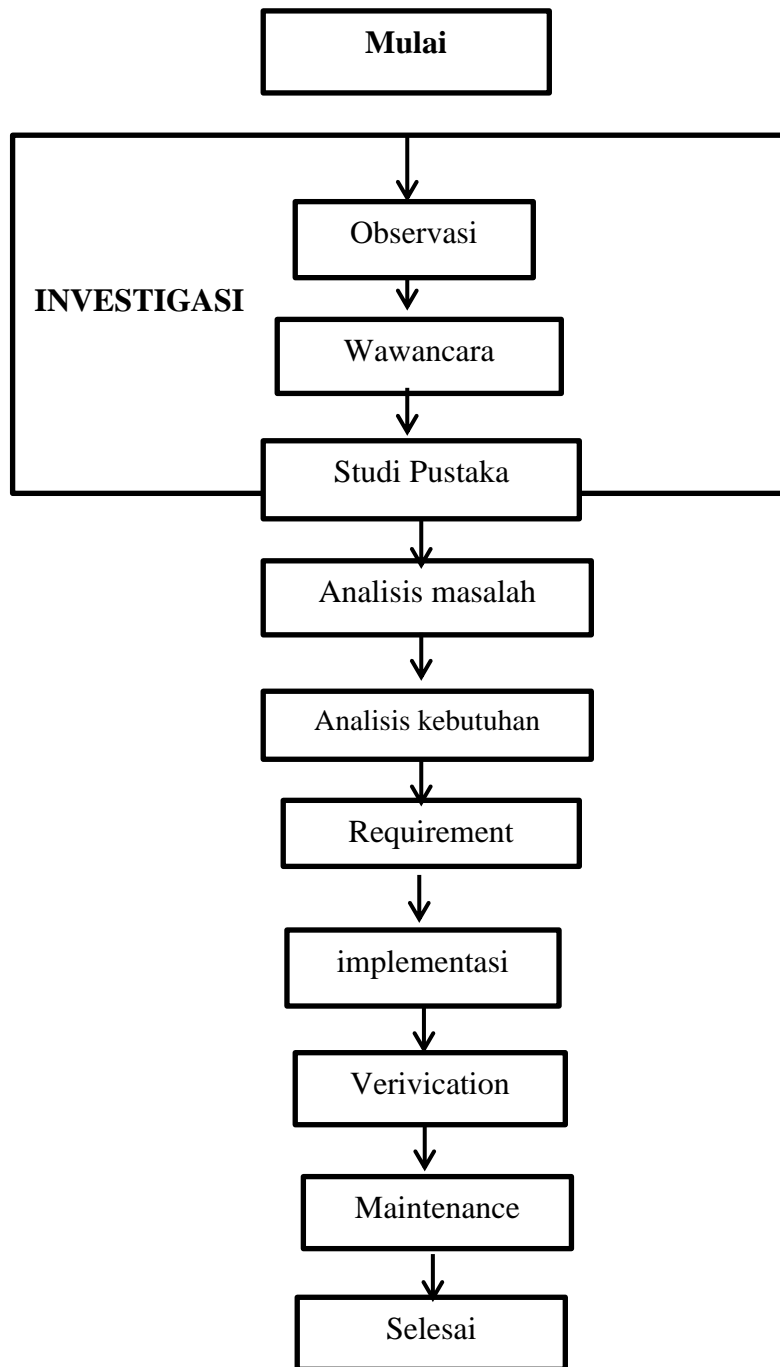
2.2.19 Diagram Class

Diagram class merupakan himpunan dari objek-objek yang sejenis, sebuah objek memiliki keadaan sesaat (state)dan perilaku (behavior). State sebuah objek adalah kondisi objek tersebut yang dinyatakan dalam attribute atau properties. (Munawar, 2005)

BAB III

METODELOGI PENELITIAN

1.3 KERANGKA PIKIR



Gambar 1.3 Kerangka Pikir

Berikut ini adalah langkah –langkah yang dilakukan untuk mencapai tujuan dari penilaian ini adalah sebagai berikut:

3.1 Deskripsi

3.2.1 Investagi Pendahuluan

Identifikasi dilakukan Pada Kantor Desa Baros yang merupakan kantor desa yang berada di desa baros kecamatan arjasari. Dalam hal ini investigasi penulis melakukan 3 metode yaitu sebagai berikut:

a.Observasi: Metode observasi merupakan cara pengumpulan data dengan cara mengamati langsung aktivitas dan kegiatan yang berlangsung di Kantor Desa Baros untuk mengetahui hal apa saja yang menjadi hambatan dan kendala di kantor desa baros.

b.Wawancara: wawancara dilakukan dengan Bapak Heru selaku sekdes desa baros. wawancara dilaksanakan di kantor desa baros yang beralamat di KP Cimentrik Rt 01 Rw 07 Desa Baros Kecamatan Arjasari Kabupaten Bandung dengan kode pos 40379. Adapun hasil wawancara yang telah dilaksanakan yaitu kendala dalam mengelola keuangan desa yang masih menggunakan cara semi manual.

c. Studi pustaka dilakukan dengan cara membaca jurnal penelitian terkait dengan objek penelitian sebagai penunjang dan juga mencari beberapa referensi dari website, buku, dll.

3.2.2 Analisis Masalah

Pada tahap ini penulis melakukan analisis terhadap kegiatan yang sedang berjalan. Dalam pendataan dan pencatatannya petugas desa masih menggunakan cara manual yaitu dengan menggunakan *Microsoft Excel* dalam pengelolaan keuangan desa, sehingga banyak kelemahan yaitu besarnya human *error* dan ketidak seimbangan (*balance*) dikarenakan banyaknya format yang harus dibuat dalam beberapa lembar kerja. Hal ini menyebabkan pengerjaan laporan keuangan akan membutuhkan waktu lama. Oleh karena itu dibutuhkan aplikasi pengelolaan keuangan yang seseuai dengan standar akuntansi pemerintahan untuk memproses pencatatan transaksi secara otomatis dan menghasilkan informasi yang cepat dan tepat.

3.2.3 Analisis Kebutuhan

Berikut ini adalah analisis kebutuhan yang di perlukan untuk menjawab permasalahan yang terdiri dari 2 yaitu:

1. Kebutuhan fungsional meliputi aplikasi yang berfungsi untuk mengelola data keuangan desa seperti: perencanaan, penganggaran, dan penatausahaan .

Menghasilkan aplikasi pengelolaan keuangan desa.

2. Kebutuhan non fungsional

Kebutuhan non fungsional meliputi Hardware dan Software pendukung untuk menjalankan aplikasi keuangan desa.

a. Hardware

Sesifikasi perangkat keras yang digunakan untuk membangun Aplikasi Keuangan Desa Berbasis Website tersebut adalah dibutuhkan hardware dengan spesifikasi sebagai berikut:

1. Processor Intel (R) Core(TM) i3-5005U CPU @ 2.00GHZ
2. RAM 4GB
3. VGA Nvida GeForce 920MX, Shared Memory 2048 MB
4. Harddisk 500 GB

b. Software

Perangkat lunak yang digunakan untuk membangun aplikasi Pengelolaan Keuangan Desa Berbasis Website dibutuhkan perangkat lunak sebagai berikut:

1. Microsoft Windows 10 Home Language 64-bit
2. XAMPP
3. Sublimetext
4. Web browser

3.2.4 Requirement

Pembuatan perancangan menggunakan metode *object oriented*, standarisasi bahasa pemodelan untuk menggunakan teknik pemograman berorientasi objek yaitu Unified Modeling Language (UML). Perancangan menggunakan UML dilakukan agar perancangan sistem yang dikerjakan akan lebih mudah untuk dikerjakan dan di kembangkan maka didapat perancangan sebagai berikut:

a. *Usecase Diagram* digunakan untuk menggambarkan keterhubungan actor dan use case yang terlibat dalam penelitian ini adalah:

1. *Actor* : Admin dan bendahara desa

a. *Use Case Diagram*:

Admin : Dapat menambah data keuangan desa, dapat menyimpan data keuangan, mengedit laporan keuangan, dapat mencetak data keuangan desa, menghapus data keuangan.

2. Bendahara: Dapat menginput data keuangan dan menyimpan data keuangan.

b. *Class Diagram* sistem yang akan dibuat:

1. Mengelola data perencanaan

2. Mengelola data penganggaran

3. Mengelola data penatausahaan

c. *Activity Diagram* digunakan untuk memodelkan alur kerja dari sistem dan aktivitas dari actor dalam sistem informasi. Adapun diagram alur yang akan dibuat diantaranya:

Menu *Start* :

1. Melihat *home* : menampilkan fitur pengelolaan keuangan

2. Melihat *about* : tentang aplikasi keuangan desa

3. Melihat *contact* : alamat dan kontak kantor desa baros

Menu *Login* :

4. Mengelola data user

5. Mengelola data laporan

d. *Sequence Diagram* digunakan untuk menunjukkan rangkaian pesan yang dikirim antara objek lain. Berikut *sequence diagram* yang akan dibuat:

1. *Sequence Diagram Login*

2. *Sequence Diagram Admin*

3. *Sequence Diagram Data keuangan desa*

1. Database

Dalam perancangan database, data yang dibutuhkan sebelumnya telah didapatkan pada proses pengumpulan data yang dibutuhkan sebelumnya telah

didapatkan pada proses pengumpulan data dan sudah di analisis. Perancangan database menggunakan *class diagram*. Beberapa data yang digunakan adalah:

- 1.Data Perencanaan
- 2.Data Penganggaran
- 3.Data Penatausahaan

2.Perancangan *User Interface*

Pada tahap perancangan *user interface* ini merupakan sebuah tampilan pada sistem yang akan dibangun dengan menggunakan *Bootstrap*. Maka diperoleh perancangan *user interface* adalah sebagai berikut:

- a.tampilan halaman home
- b.tampilan halaman about
- c.tampilan halaman contact
- d.tampilan halaman login
- e.tampilan halaman admin
- f.tampilan halaman perencanaan
- g.tampilan halaman penganggaran
- h.tampilan halaman penatausahaan

Jika dalam tahap ini masih terdapat kesalahan atau kekurangan maka akan kembali ke tahap analisis kebutuhan sampai mendapatkan hasil yang tepat dan sesuai, kemudian baru dilanjutkan ke tahap berikutnya.

3.2.5 Implementasi

Sebelum implementasi dilakukan maka di awali melakukan pengujian terlebih dahulu terhadap sistem yang sudah dibuat. Pengujian dilakukan dengan cara mengamati input dan output dari sistem informasi menggunakan teknik *Black box* yaitu pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Kriteria yang diukur dalam tahap ini yaitu:

- 1.Menguji apakah sistem berjalan dengan baik dan bisa digunakan oleh *user* seperti Login, data perencanaan, penganggaran, penatausahaan, dan data *user*.
- 2.Menghasilkan *database* yang mampu menyimpan, mengelola data keuangan desa.

3.2.6 Verivication

Pada tahapan verifikasi merupakan tahapan pengujian sistem yang dilakukan untuk mengetahui kelayakan sistem Aplikasi Administrasi Desa dengan cara menguji aplikasi dari segi sefesifikasi fungsional tanpa menguji desain dan kode pogram.

3.2.7 Maintenance

Maintenance adalah tahapan terakhir dari metode pengembangan *waterfall*. Disini *software* yang sudah jadi akan dijalankan atau di operasikan oleh pengguna nya. Disamping itu dilakukan pula pemeliharaan yang termasuk:

- 1.Perbaiki kesalahan
- 2.Perbaiki implementasi unit
- 3.Peningkatan jasa sistem sesuai kebutuhan baru

BAB IV

ANALISIS DAN PERANCANGAN

1.1 Analisis

4.1.1 Analisis Masalah

Berdasarkan analisis yang penulis temukan, pendataan, pencatatan, keuangan desa. masih menggunakan cara semi manual (*Microsoft Excel*) sehingga banyak kelemahannya yaitu besarnya human eror dan ketidak seimbangan (*balance*) dikarenakan banyaknya format yang harus dibuat dalam beberapa lembar kerja. Hal ini menyebabkan pengerjaan laporan keuangan membutuhkan waktu lama. Oleh karena itu, dalam penelitian ini penulis mengusulkan untuk merancang aplikasi administrasi desa berbasis *website*, agar bisa mempermudah pekerjaan bendahara desa. Berdasarkan paparan permasalahan diatas, penulis melakukan uji coba menggunakan Xampp sebagai solusi untuk permasalahan tersebut. Kemudian analisis kebutuhan, juga menjadi hal yang penting dalam penelitian ini.

4.1.2 Analisis Software

Pada pembuatan aplikasi administrasi desa berbasis *website* ini, penulis menggunakan *hardware* dan *software* dengan spesifikasi sebagai berikut:

1. Hardware

Perangkat keras yang digunakan untuk mengembangkan dan mengumpulkan data pada aplikasi, dengan spesifikasi sebagai berikut:

Tabel 4.0.1 Spesifikasi Hardware

Spesifikasi	Deskripsi
Tipe Laptop	Asus
Prosesor	Intel R Pentium (R) CPU NR 3540 @2.116GHZ
RAM	200 GB (1,89 GB useable)
HDD	500GB

2. *Software*

Adapun *Software* yang digunakan dalam mengembangkan dan mengumpulkan data pada aplikasi ini adalah sebagai berikut:

1. Microsoft Windows 10 Home Page Language 64-bit
2. XAMPP
3. Sublime text
4. Web browser

4.1.3 Analisis Pengguna

Analisis kebutuhan pengguna dilakukan agar dapat mengetahui siapa saja yang dapat mengakses aplikasi pengelolaan keuangan desa berbasis *website* ini. Pengguna dapat meng *input* data keuangan dan mencetak laporan keuangan. pengguna merupakan *user* general yang memiliki kemampuan dalam mengakses aplikasi keuangan desa berbasis desa, memiliki pengetahuan tentang aplikasi berbasis *website*. Pengguna yang bisa mengakses aplikasi keuangan desa ini yaitu admin dan bendahara desa, Adapun hal-hal yang bisa dilakukan oleh admin dalam sistem yaitu sebagai berikut:

1. Admin

a. *Login*

- b. Memverifikasi *upload* data keuangan desa
- c. Memverifikasi *edit* data keuangan
- d. Memverifikasi *delete* data keuangan
- e. Memverifikasi cetak laporan keuangan

f. *Logout*

2. Bendahara

Bendahara yang dimaksud adalah bendahara desa yang bertugas mengelola data keuangan desa. Adapun hal-hal yang dapat dilakukan bendahara dalam sistem yaitu sebagai berikut:

a. *Login*

- b. Meng *input* data perencanaan

- c. Meng *input* data penganggaran
- e. Meng *input* data penatausahaan
- g. Mencetak laporan penata usahaan
- h. *Logout*

4.1.4 Analisis User Interface

Desain yang baik tentu saja memberikan tampilan yang menarik dan mudah untuk digunakan. *User interface* adalah tampilan visual dalam sebuah aplikasi sebagai tempat (media) untuk menghubungkan antara sistem dengan *user*. Dari situlah terdapat antarmuka awal dari *website* agar *user* dapat berinteraksi dengan baik. Tampilan *interface* dari aplikasi administrasi desa ini diharapkan memudahkan *user* saat menggunakan aplikasi administrasi desa ini.

Berikut adalah rancangan pada aplikasi administrasi desa yaitu sebagai berikut:

- a. Tampilan halaman *login*
- b. Tampilan *dashboard* aplikasi adminisrasi keuangan desa
- c. Tampilan menu perencanaan
- d. Tampilan menu penganggaran
- e. Tampilan menu penatausahaan
- f. Tampilan menu laporan
- g. Tampilan menu administrator
- i. Fitur-Fitur

Fitur-fitur yang dibuat dalam aplikasi ini dimaksudkan untuk mempermudah pengguna dalam menjalankan aplikasi ini, dan juga mempermudah admin dan bendahara dalam melakukan pengelolaan aplikasi ini. Berikut ini adalah fitur-fitur yang ada dalam aplikasi ini yaitu sebagai berikut:

1. *Login* dapat dilakukan oleh admin dan bendahara apabila admin atau bendahara ingin melihat data keuangan desa yang sudah di input kedalam aplikasi administrasi desa.

2. Pada menu perencanaan desa terdapat sub fitur data umum, visi, misi, tujuan, sasaran dan pelaksanaan kegiatan. Bendahara dapat menginput data pada form yang ada didalam sub fitur.
3. Pada menu penganggaran desa terdapat sub fitur kelompok, jenis, dan objek. Bendahara dapat meng input data pada form yang ada di dalam sub fitur.
4. Pada menu RAP terdapat sub fitur RAP dan RAP rincian, bendahara dapat meng *input* data pada form yang ada didalam sub fitur.
5. Pada menu RAB terdapat sub fitur RAB dan RAB rincian, bendahara dapat meng *input* data pada form yang ada didalam sub fitur.
6. Pada menu penerimaan desa, terdapat sub fitur penerimaan bank, rincian penerimaan bank, penerimaan tunai, dan rincian penerimaan tunai. Bendahara dapat meng *input* data pada form yang ada di dalam sub fitur.
7. Pada menu penyelenggaraan desa terdapat sub fitur spp, rincian spp, bukti pengeluaran, dan potongan. Bendahara dapat meng *input* data pada form yang ada didalam sub fitur.
8. Pada menu daftar laporan terdapat sub fitur, laporan penganggaran desa, laporan penatausahaan desa dan potongan pajak. Bendahar dapat mencetak laporan yang ada di dalam sub fitur.

1.1.6 Analisis Data

Berikut ini merupakan analisis data berupa data masukan (*input*) proses, dan keluaran (*Output*) sebagai penunjang untuk aplikasi administrasi desa yang akan dibuat yaitu sebagai berikut:

Tabel 4.2 AnalisisData

No	Masukan	Proses	Keluaran
1.	Login	Memeriksa Status login validasi	Informasi Login
2.	Menginput data penganggaran	Memeriksa data penganggaran Validasi Mengurutkan data	Informasi data penganggaran

		penganggaran	
3.	Menginput data perencanaan	Memeriksa data perencanaan Validasi Mengurutkan data perencanaan	Informasi data perencanaan
	Menginput data penatausahaan	Memeriksa data penatausahaan Validasi Mengurutkan data penatausahaan	Informasi data penatausahaan
4.			
5.	Cetak Laporan	Memeriksa data laporan Mencetak laporan	Informasi Laporan perbulan

4.2 Perancangan

Pembuatan perancangan menggunakan metode *object oriented*, standarisasi bahasa pemodelan untuk menggunakan teknik pemograman berorientasi objek yaitu *Unified Modeling Language* (UML). Perancangan menggunakan UML dilakukan agar perancangan sistem yang dikerjakan akan lebih mudah untuk dikerjakan dan di kembangkan.

1. Aktor

Aktor merupakan penjelasan dari apa yang dilakukan aktor-aktor yang terlibat dalam aplikasi yang dibangun. Pada sistem ini terdapat dua aktor yaitu admin dan bendahara. Adapun deskripsi dari aktor-aktor yang terlibat dalam aplikasi keuangan desa yaitu sebagai berikut:

Tabel 4.3 *Actor*

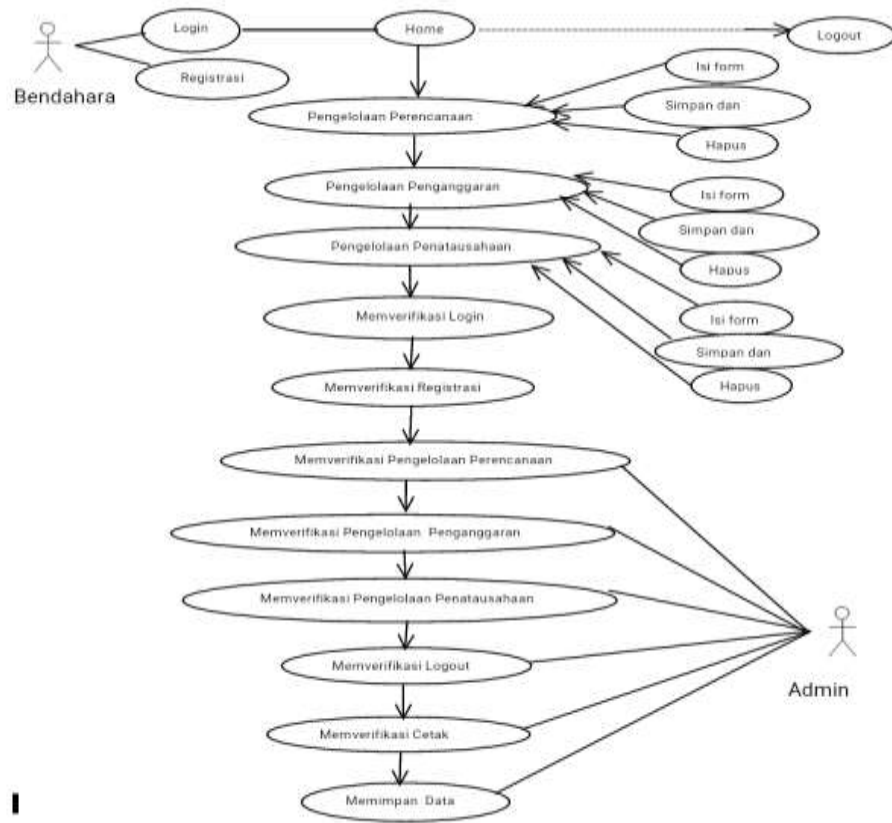
No	Aktor	Deskripsi
1.	Admin	1. <i>Login</i> 2. Lihat tampilan <i>dashboard</i> 3. Mencari, menambah, menghapus , mengedit data admin 4. Mencari, menambah, menghapus , mengedit data dan <i>upload</i> profile. 5. Mencari, menambah, melihat detail, menghapus, mengedit data penganggaran. Perencanaan, dan penatausahaan. 6. Mencari dan Mencetak Laporan
2.	Aktor	Deskripsi
	Bendahara	1. <i>Login</i> 2. Melihat tampilan <i>dasboard</i> 3. Meng <i>input</i> data perencanaan 4. <i>input</i> data penganggaran 5. <i>input</i> data pelaksanaan 6 <i>input</i> data penatausahaan 7. mencetak laporan keuangan 8. <i>Logout</i>

4.2.1 Use Case Diagram

Use case diagram merupakan pemodelan untuk melakukan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat secara kasar yang digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu” (Rosa dan Shalahuddin, 2018:155).

Berikut ini adalah *use case* aplikasi administrasi desa yaitu sebagai berikut:

1. Use Case Diagram Admin Dan Bendahara

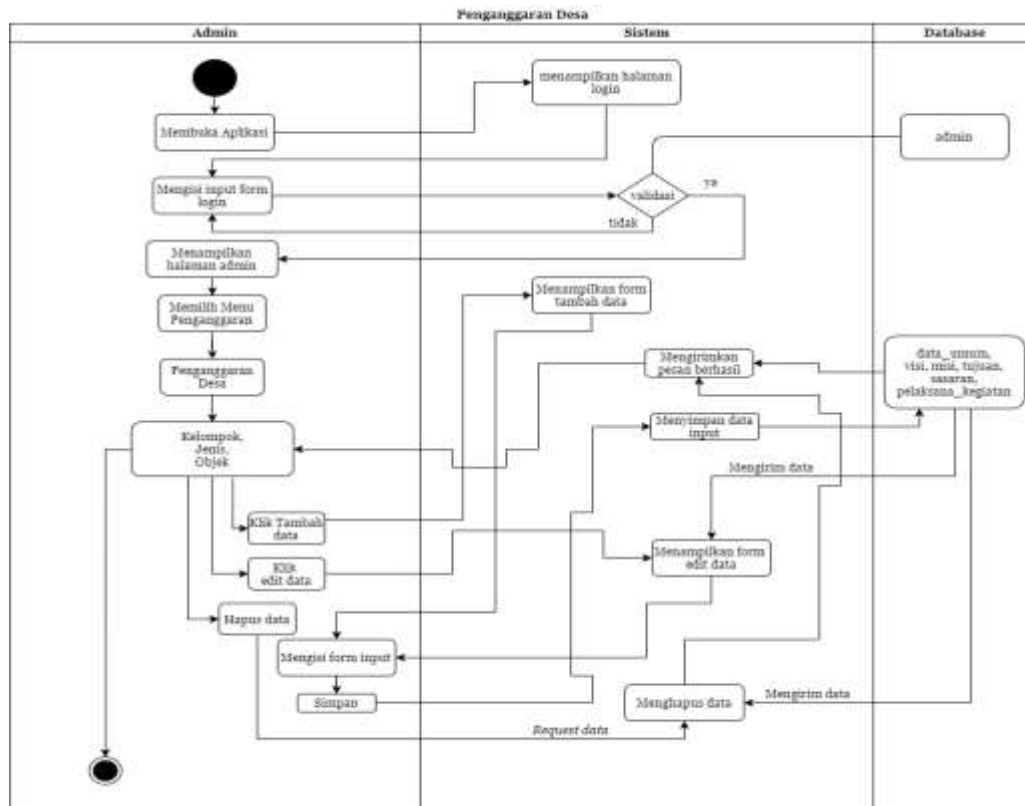


Gambar 4.5 Use Case Diagram Admin & Bendahara

Deskripsi Diagram *Usecase* Diusulkan diatas, yaitu Admin dapat melakukan *login*, registrasi, memverifikasi data perencanaan, penganggaran, penatausahaan ,menyimpan data dan memverifikasi cetak laporan. Sedangkan Bendahara dapat *login*, registrasi, meng *input* data pada from, dapat menyimpan, melihat data, menghapus data dan mencetak data.

8. Admin dapat menyompan data

- **Activity Diagram Menu Penganggaran Desa**



Gambar 4.8 Activity Diagram Menu Penganggaran Desa

Activity Diagram merupakan suatu flow diagram yang memodelkan alur kerja (*work flow*) sebuah proses informasi dan sebuah aktifitas sebuah proses. activity diagram diatas , merupakan Activity Diagram Menu Penganggaran Desa. Berikut ini adalah penjelasan dari Diagram Menu Penganggaran Desa

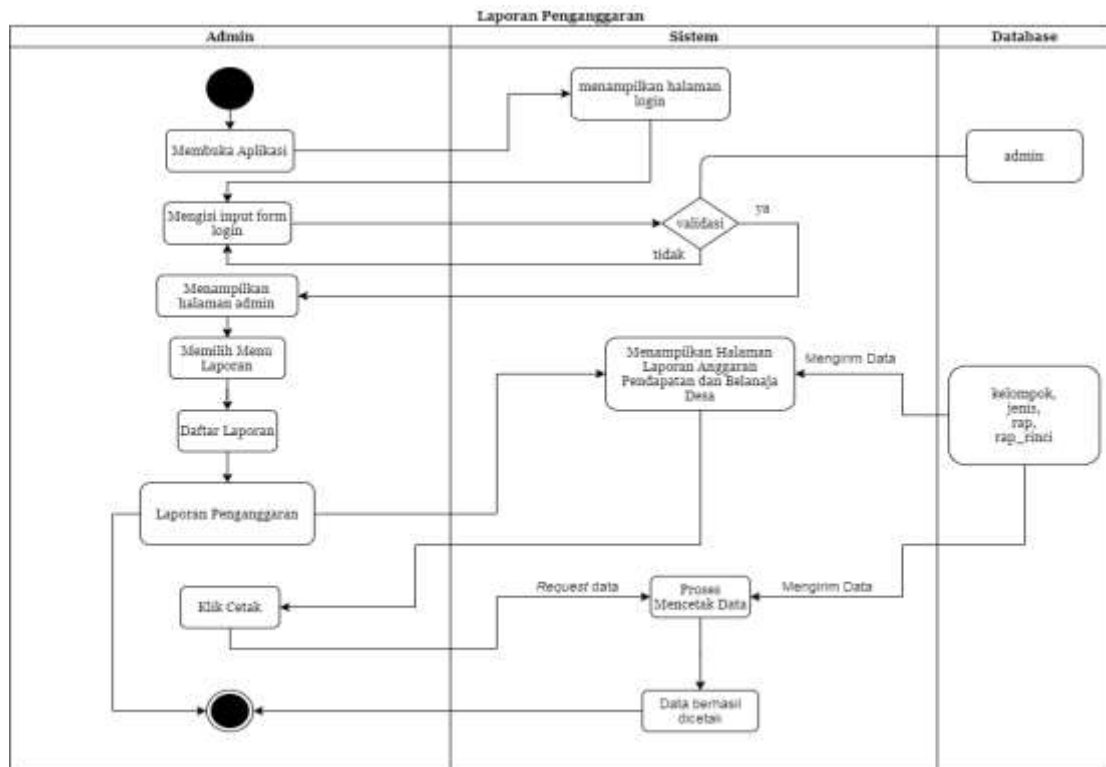
1. Menampilkan Halaman *Login*
2. Menginput *Form Login*
3. Menampilkan Halaman Admin
4. Memilih Menu Penganggaran
5. Menampilkan Form Tambah Data
6. Menambah Data
7. Mengedit Data
8. Menghapus Data
9. Mengirimkan Pesan Berhasil
10. Menyimpan Data *Input*

8.Mengirimkan Pesan Berhasil

9.Menyimpan Data *Input*

10.Menghapus Data

- **Activity Diagram Laporan Penganggaran**



Gambar 4.10 *Activity Diagram* Laporan Penganggaran

Activity Diagram merupakan *flow diagram* yang memodelkan alur kerja (*work flow*) sebuah proses informasi dan sebuah aktivitas sebuah proses. *Activity diagram* diatas merupakan *Activity Diagram* Laporan penganggaran.

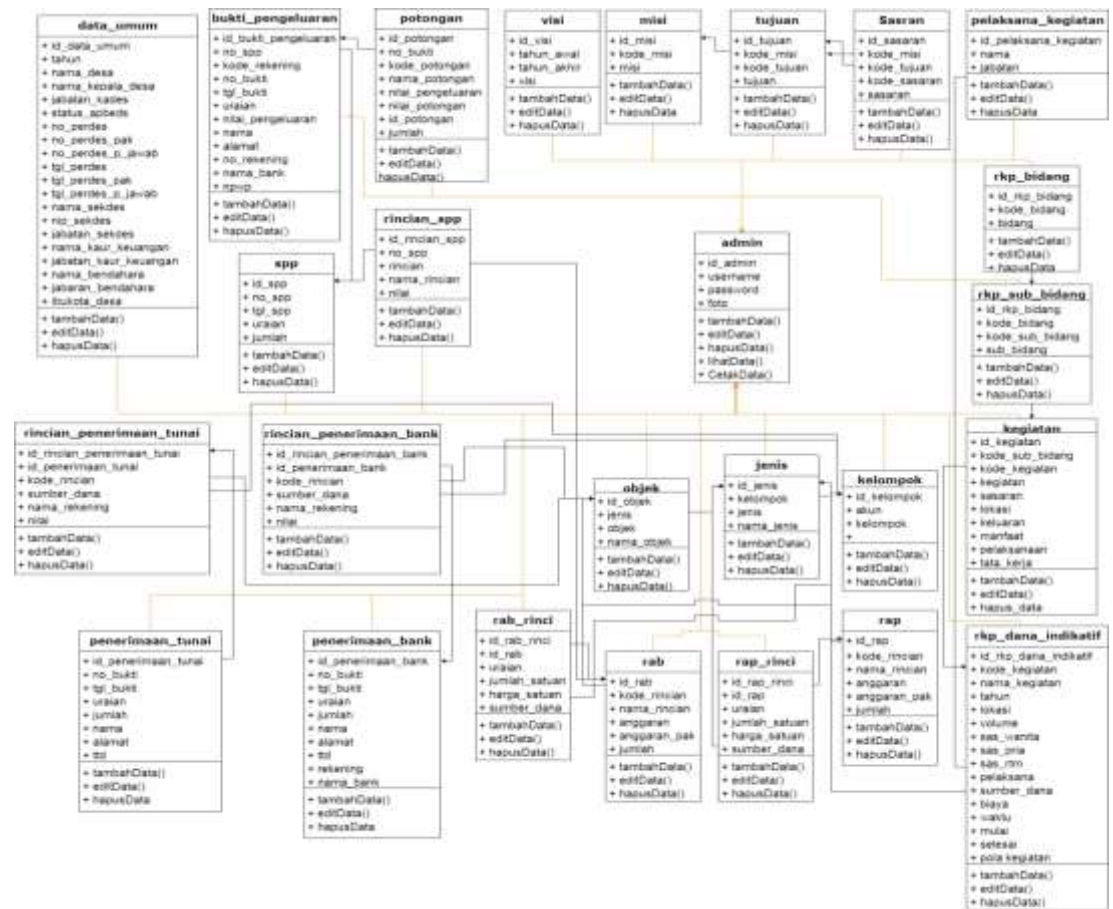
Berikut ini adalah penjelasan dari *Activity Diagram* Laporan Penganggaran

yaitu sebagai berikut:

- 1.Membuka Aplikasi
- 2.Menampilkan Halaman *Login*
- 3.Menginput *From Login*
- 4.Memilih Menu Laporan

5. Menampilkan Laporan APBD

4.2.3 Class Diagram Admin



Gambar 4.11 Class Diagram Admin

Deskripsi Diagram *Class*, *Class* adalah sebuah klasifikasi yang jika intansi akan menghasilkan sebuah objek dan merupakan inti-inti dari pengembangan dan desain beroreintasi objek. *Class* menggambarkan keadaan suatu sistem, sekaligus menawarkan layanan untuk memanipulasi layanan tersebut.

4.2.4 Perancangan Tabel

Berikut ini dijelaskan tabel- tabel yang digunakan dalam perancangan tabel dengan deskripsi sebagai berikut :

1. Tabel Data Umum

Nama Tabel : Admin

Keterangan : Menampung data umum

Tabel 4.10 Tabel Data Umum

NO	Nama field	Tipe Data	Keterangan
1	id_data umum	Int (12)	Untuk mengetahui data umum dan tahun.
	tahun	Int (12)	
	nama_desa	Int (12)	Untuk mengetahui nama desa.
	nama_kepala desa	Int (12)	Untuk mengetahui nama kades.
	jabatan_kades	Int (12)	
	status_kades	Int (12)	Untuk mengetahui jabatan kades.
	no_perdes	Int (12)	
	np _perdes pak	Int (12)	Untuk mengetahui nama perdes.
	no_perdes_p_ jawab	Int (12)	
	tgl_perdes	Int (12)	Untuk mengetahui nama perdes pak.
	perdes_pak	Int (12)	
	perdes_ p_jawab	Int (12)	Untuk mengetahui nama perdes p jawab.
	nama_sekdes	Int (12)	
	nip_sekdes	Int (12)	Untuk mengetahui nama sekdes.
	jabatan_sekdes	Int (12)	
	nama	Int (12)	Untuk mengetahui nip sekdes.
	kaur_keuangan	Int (12)	
	jabatan_kaur	Int (12)	Untuk mengetahui jabatan sekdes.
	keuangan	Int (12)	
	nama_bendahara	Int (12)	Untuk mengetahui nama kaur keuangan.
	jabatan_bendahara	Int (12)	
	ibu kota_desa	Int (12)	Untuk mengetahui jabatan kaur.
		Int (12)	
		Int (12)	Untuk mengetahui

		Int (12) Int (12) Int (12) Int (12) Int (12) Int (12) Int (12)	nama bendahara. Untuk mengetahui jabatan bendahara. Untuk mengetahui ibu kota desa.
2	Username	Varchar (255)	Nama pengguna
3	Password	Varchar (255)	Password pengguna

1. Tabel Bukti Pengeluaran

Nama Tabel : Bukti Pengeluaran

Keterangan : Menampung Data Pengeluaran

Tabel 4.11 Tabel Bukti Pengeluaran

No	Nama filed	Tipe Data	Keterangan
1.	id_bukti pengeluaran	Int (12)	Nomor identitas untuk
	no_spp	Int (12)	Pengurutan otomatis
	kode_rekening	Int (12)	Bukti pengeluaran
	no_bukti	Int (12)	Kode rekening
	nilai_pengeluaran	Int (12)	Nomor bukti
	nama	Int (12)	Nilai pengeluaran
	alamat	Int (12)	Nama
	no_rekening	Int (12)	Alamat
	nama_bank	Int (12)	No rekening
	npwp	Int (12)	Npwp

2. Tabel Potongan

Nama Tabel : Potongan

Keterangan : Menampung Data Potongan

Tabel 4. 12 Tabel Potongan

No	Nama filed	Tipe Data	Keterangan
1.	Id_potongan	Int (12)	Id potongan untuk
	no_bukti	Int(12)	mengetahui identitas
	kode_potongan	Int(12)	potongan
	nama_potongan	Int(12)	no bukti untuk
	nilai_pengeluaran	Int(12)	mengetahui nomor bukti
	nilai_potongan	Int(12)	nilai pengeluaran untuk
	jumlah	Int(12)	mengetahui nilai pengeluaran jumlah untuk mengetahui jumlah keseluruhan.

3. Tabel visi

Nama Tabel : visi

Keterangan : Menampung data visi

Tabel 4.13 Tabel Visi

No	Nama Filed	Tipe Data	Keterangan
1.	Id_vis	Int (12)	Id visi untutuk
	Tahun_awal	Int(12)	mengetahui id pengguna visi.
	Tahun_akhir	Int (12)	Tahun awal untuk mengetahui tahun awal visi. Tahun akhir untuk mengetahui tahun akhir visi.

4. Tabel Misi

Nama Tabel : Misi

Keterangan : Menampung Data Misi

Tabel 4.14 Data Misi

No	Nama Filed	Tipe Data	Keterangan
1.	Id_misi kode misi	Int (12) Int (12)	Id misi untuk mengetahui id pengguna misi. kode misi untuk mengetahui kode misi.

5. Tabel Data Tujuan

Nama Tabel : Tujuan

Keterangan : Menampung Data Tujuan

Tabel 4.15 Data Tujuan

No	Nama Filed	Tipe Data	Keterangan
1.	Id_tujuan kode_tujuan	Int (12) Int (12)	Id misi untuk mengetahui id pengguna tujuan. Untuk mengetahui kode tujuan.

6. Tabel Data Sasaran

Nama Tabel : Sasaran

Keterangan : Menampung Data Sasaran

Tabel 4.16 Data Sasaran

No	Nama Filed	Tipe Data	Keterangan
1.	Id_sasaran kode_sasaran	Int (12) Int (12)	Id misi untuk mengetahui id pengguna sasaran. Kode sasaran untuk mengetahui kode sasaran.

7. Tabel Data Pelaksanaan Kegiatan

Nama Tabel : Pelaksanaan Kegiatan

Keterangan : Menampung Data Pelaksanaan Kegiatan

Tabel 4.17 Data Pelaksanaan Kegiatan

No	Nama Filed	Tipe Data	Keterangan
1.	Id_pelaksanaan kegiatan Nama_jabatan	Int (12) Int (12)	Id misi untuk mengetahui id pengguna pelaksanaan kegiatan. Nama jabatan untuk mengetahui nama jabatan.

8. Tabel Data spp

Nama Tabel : Data SPP

Keterangan : Menampung Data Spp

Tabel 4.18 Data SPP

No	Nama Filed	Tipe Data	Keterangan
1.	Id_spp no_spp tanggal_spp	Int (12) Int (12) Int (12)	Id spp untuk mengetahui id pengguna spp. No spp untuk mengetahui nomor spp. Tanggal spp untuk mengetahui tanggal spp.

9. Tabel Data Rincian Spp

Nama Tabel : rincian spp

Keterangan : Menampung Data Rincian Spp

Tabel 4.19 Data Rincian Spp

No	Nama Filed	Tipe Data	Keterangan
1.	Id_rincian spp Nomor_spp Nama_rincian	Int (12) Int (12) Int (12)	Id rincian spp untuk mengetahui rincian spp. Nomor spp untuk mengetahui nomor spp.

			Nama rincian untuk mengetahui nama rincian.
--	--	--	---

10. Tabel Data Admin

Nama Tabel : Admin

Keterangan : Menampung Data Admin

Tabel 4.20 Data Admin

No	Nama Filed	Tipe Data	Keterangan
1.	Id_admin user_name password	Int (12) Int (12) Int (12)	Id admin untuk mengetahui id pengguna admin. User name, nama pengguna aplikasi. Password yaitu kata sandi yang digunakan saat masuk aplikasi.

11. Data RKP Bidang

Nama Tabel : RKP Bidang

Keterangan : Menampung Data RKP Bidang

Tabel 4.21 Data RKP Bidang

No	Nama Filed	Tipe Data	Keterangan
1.	Id_rkp Kode_bidang	Int (12) Int (12)	Id rkp mengetahui id pengguna rkp. Kode bidang untuk mengetahui kode bidang.

12. Tabel Data RKP Sub Bidang

Nama Tabel : RKP Sub Bidang

Keterangan : Menampung RKP Sub Bidang

Tabel 4.22 RKP Sub Bidang

No	Nama Filed	Tipe Data	Keterangan
1.	Id_rkp bidang kode_bidang kode_sub bidang	Int (12) Int (12)	Id rkp sub bidang untuk mengetahui id pengguna. Kode bidang untuk mengetahui kode bidang yang digunakan. Kode sub bidang digunakan untuk mengetahui kode sub bidang.

13. Tabel Data Rincian Penerimaan Tunai

Nama Tabel : Rincian Penerimaan Tuna

Keterangan : Menampung Data Rincian Penerimaan Tunai

Tabel 4.23 Data Rincian Penerimaan Tunai

No	Nama Filed	Tipe Data	Keterangan
1.	Id_rincian penerimaan tunai kode_rician sumber_dana nama_rekening	Int (12) Int (12)	Id rincian penerimaan tunai digunakan untuk mengetahui id pengguna. Kode rincian digunakan untuk mengetahui kode rincian. Sumber dana digunakan untuk mengetahui jumlah sumber dana. Nama rekening digunakan untuk mengetahui nama pemilik rekening.

14. Tabel Data Penerimaan Bank

Nama Tabel : Data Penerimaan Bank

Keterangan : Menampung Data Penerimaan Bank

Tabel 4.24 Data Penerimaan Bank

No	Nama Filed	Tipe Data	Keterangan
1.	Id_penerimaan bank	Int (12)	Id pengguna digunakan untuk mengetahui id pengguna penerima. Kode rincian untuk mengetahui kode rincian yang digunakan. Nama rekening digunakan untuk mengetahui nama pemilik rekening.
	Kode_rincian	Int (12)	
	Sumber_dana	Int (12)	
	Nama_rekening	Int (12)	

15. Tabel Objek

Nama Tabel : Data Objek

Keterangan : Menampung Data Objek

Tabel 4.25 Data Objek

No	Nama Filed	Tipe Data	Keterangan
1.	Id_objek	Int (12)	Id objek untuk mengetahui objek yang digunakan. Nama objek untuk mengetahui nama objek yang digunakan.
	nama _objek	Int (12)	

16. Tabel Jenis

Nama Tabel : Data Jenis

Keterangan : Menampung Data Jenis

Tabel 4.26 Data Jenis

No	Nama Filed	Tipe Data	Keterangan
1.	Id_jenis Nama_jenis	Int (12) Int (12)	Id jenis untuk mengetahui jenis yang digunakan. Nama jenis untuk mengetahui nama yang digunakan.

17. Tabel Kelompok

Nama Tabel : Data Kelompok

Keterangan : Menampung Data Kelompok

Tabel 4.27 Data Kelompok

No	Nama Filed	Tipe Data	Keterangan
1.	Id_kelompok Akun_kelompok	Int (12) Int (12)	Id kelompok untuk mengetahui kelompok mana yang digunakan. Akun kelompok untuk mengetahui akunkelompok mana yang digunakan.

18. Tabel Kegiatan

Nama Tabel : Data Kegiatan

Keterangan : Menampung Data Kegiatan

Tabel 4.28 Data Kegiatan

No	Nama Filed	Tipe Data	Keterangan
1.	Id_kegiatan Kode_kegiatan Tata_kerja	Int (12) Int (12) Int (12)	Id kegiatan untuk mengetahui id pengguna. Kode kegiatan untuk mengetahui kode kegiatan yang digunakan. Tata kerja untuk mengetahui tata kerja yang digunakan.

19. Tabel Data RAB

Nama Tabel : Data RAB

Keterangan : Menampung RAB

Tabel 4.29 Data Penerimaan RAB

No	Nama Filed	Tipe Data	Keterangan
1.	Id_RAB Kode_rincian Anggaran_pak	Int (12) Int (12) Int (12)	Untuk mengetahui id RAB. Untuk mengetahui kode rincian. Untuk mengetahui anggaran.

20. Tabel Penerimaan Rincian RAB

Nama Tabel : Data Rincian RAB

Keterangan : Menampung Data Rincian RAB

Tabel 4.31 Data Rincian RAB

No	Nama Filed	Tipe Data	Keterangan
1.	Id_ RAB rinci	Int (12)	Id RAB Rinci untuk
	jumlah _satuan	Int (12)	mengetahui pengguna id
	harga_satuan	Int (12)	yang digunakan.
	sumber _dana		Julah satuan untuk
			mengetahui jumlah
			satuan yang digunakan.
			Sumberdana untuk
			mengetahui sumber dana
			yang digunakan.

21Tabel Penerimaan RAP Rinci

Nama Tabel : Data RAP Rinci

Keterangan : Menampung Data RAP Rinci

Tabel 32 Data RAP Rinci

No	Nama Filed	Tipe Data	Keterangan
1.	Id_ RAP rinci	Int (12)	Id RAP rinci untuk
	Kode_rincian	Int (12)	mengetahui rinccian
	Nama_rincian	Int (12)	RAP.
	Anggran pak		Kode rincian untuk
			mengetahui nama
			rincian.
			Anggaran pak untuk
			mengetahui anggaran.

22. Tabel RKP Dana Indikatif

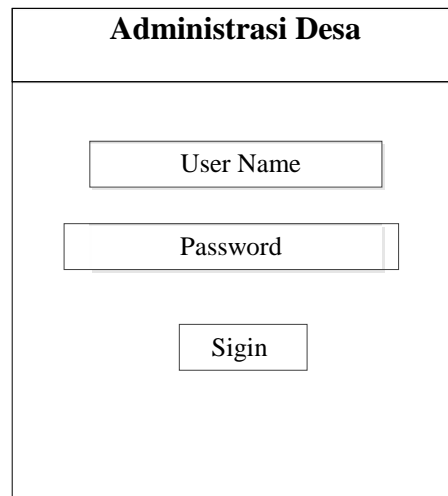
Nama Tabel : Data Dana Indikatif

Keterangan : Menampung Data RKP Dana Indikatif

Tabel 4.33 Data RKP Dana Indikatif

No	Nama Filed	Tipe Data	Keterangan
1.	Id_rkp_dana_indikatif	Int (12)	Id RKP dana indikatif
	Kode_kegiatan	Int (12)	untuk mengetahui dana
	Nama_kegiatan	Int (12)	indikatif yang
	Ss_wanita	Int (12)	digunakan.
	Ss_pria	Int (12)	Kode kegiatan untuk
	Sumber_dana	Int (12)	mengetahui kode yang
			digunakan.
			Ss wanita untuk
			mengetahui seksi wanita.
			Ss pria untuk mengetahui
			seksi pria.
			Sumber dana untuk
			mengetahui sumber
			dana.

4.2.5 Perancangan User Interface



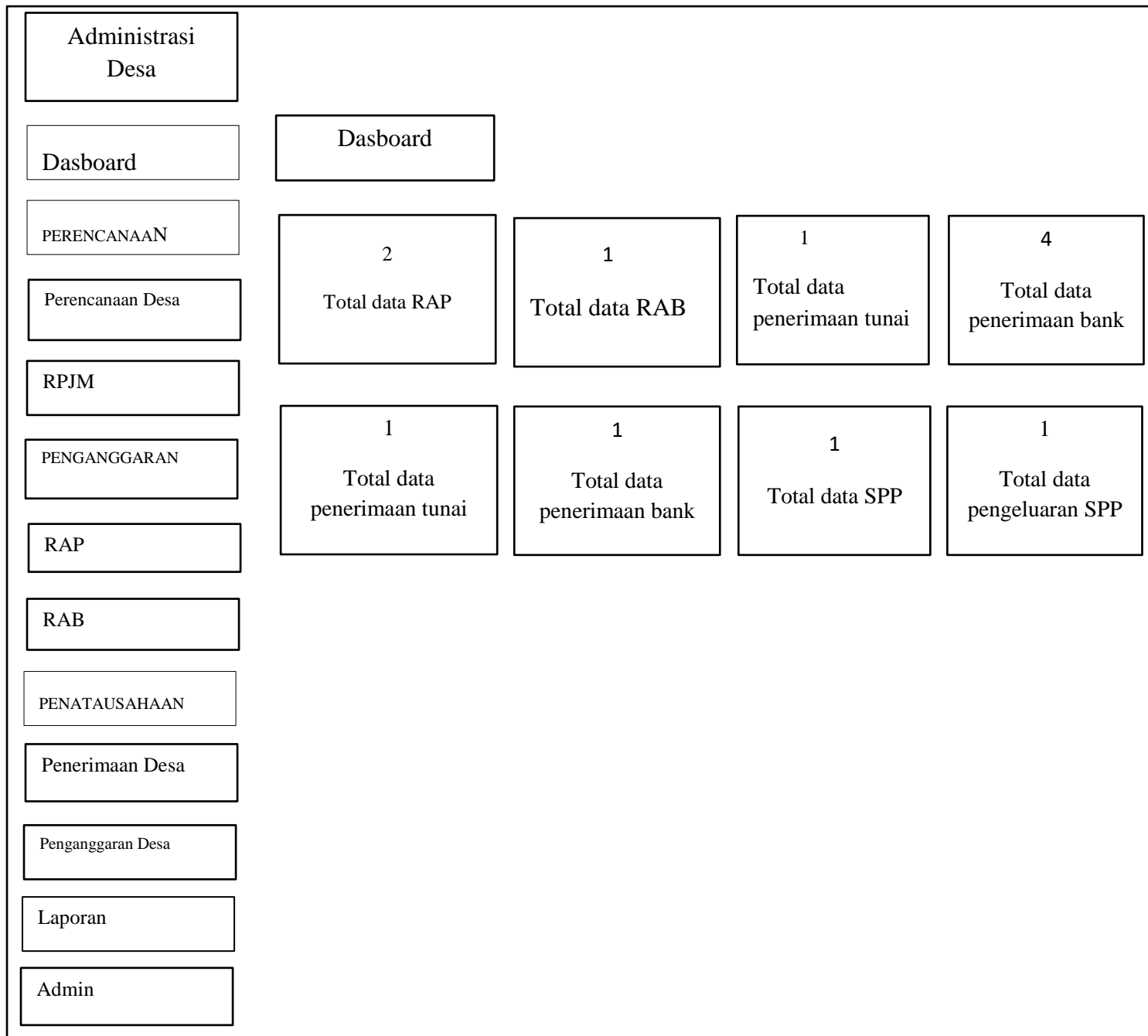
The image shows a login form titled "Administrasi Desa". It contains three input fields: "User Name", "Password", and a "Signin" button. The form is enclosed in a rectangular border.

Administrasi Desa
<div>User Name</div> <div>Password</div> <div>Signin</div>

Gambar 4.12 From *Login* Aplikasi Administrasi Desa

Gambar 4.12 diatas merupakan tampilan *form* pendaftaran *login*, rancangan *form* daftar *login* akan tampil setelah mengakses link *website* administrasi desa. *User* mengisi *user name* dan *password* terlebih dahulu, lalu mengklik *signin* setelah itu akan muncul halaman *dashboard* aplikasi administrasi desa.

• **Rancang Bangun Dashboard Aplikasi Keuangan Desa**



Gambar 4.13 Dashboard Aplikasi Keuangan Desa

Gambar 4.13 diatas merupakan tampilan web administrasi desa setelah berhasil melakukan *login*, akan muncul tampilan web administrasi desa baros.

- **Rancang Bangun Menu Perencanaan Desa**

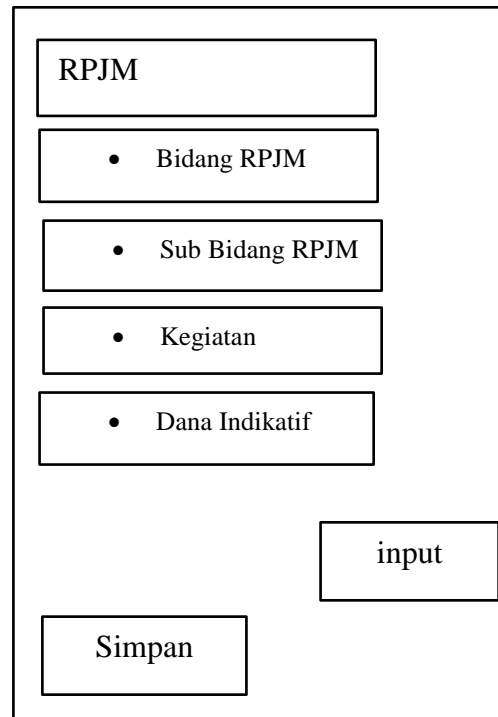
The image shows a software interface for village planning. It features a main container with a title bar 'Perencanaan Desa'. Below the title bar is a list of items: 'Visi', 'Misi', 'Tujuan', 'Sasaran', and 'Pelaksanaan Kegiatan'. Each item is preceded by a bullet point. At the bottom of the container, there are two buttons: 'Simpan' (Save) on the left and 'input' on the right.

Perencanaan Desa	
•	Visi
•	Misi
•	Tujuan
•	Sasaran
•	Pelaksanaan Kegiatan
input	
Simpan	

Gambar 4.14 Menu Penganggaran

Gambar 4.14 diatas merupakan tampilan menu perencanaan desa, yang menampilkan beberapa menu diantaranya: menu visi, menu misi, menu tujuan, menu sasaran dan menu pelaksanaan kegiatan.

- **Rancang Bangun Menu RPJM**

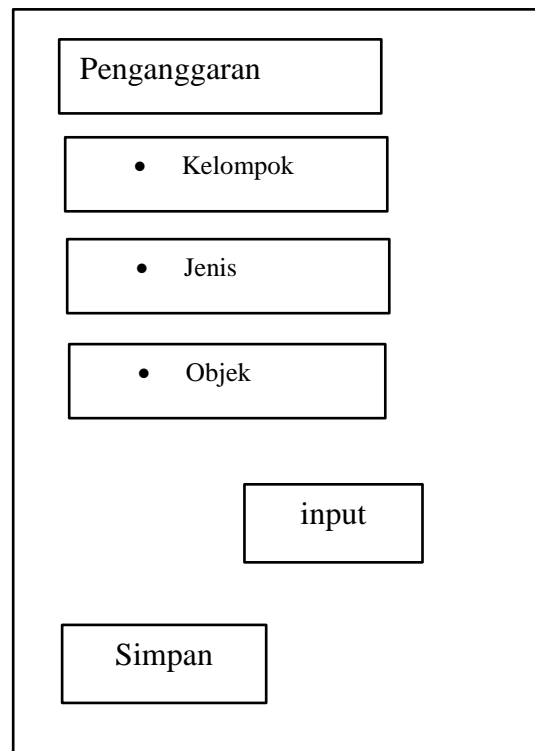


Gambar 4.15 M

Gambar 4.15 Rancang Bangun RPJM

Gambar 4.15 diatas merupakan tampilan menu RPJM yang menampilkan beberapa menu diantaranya: menu bidang RKP,menu sub bidang, menu kegiatan dan menu dana indikatif.

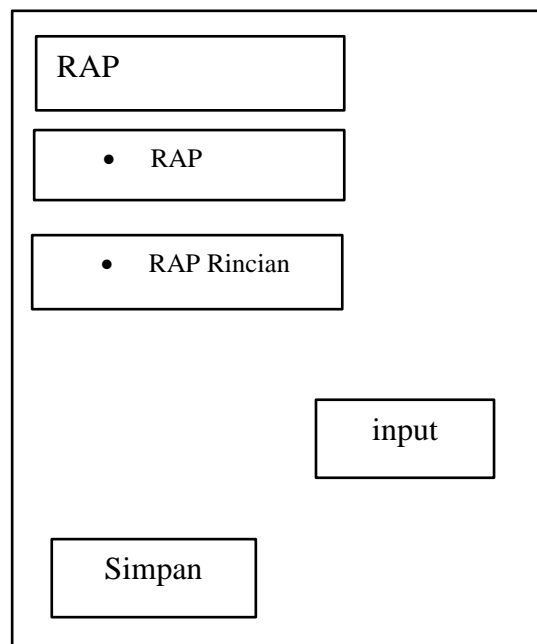
- **Rancang Bangun Menu Penganggaran Desa**



Gmbar 4.16 Penganggaran Desa

Gambar 4.16 diatas merupakan tampilan menu Penganggaran Desa yang menampilkan beberapa menu diantaranya: menu kelompok, menu jenis, dan menu objek.

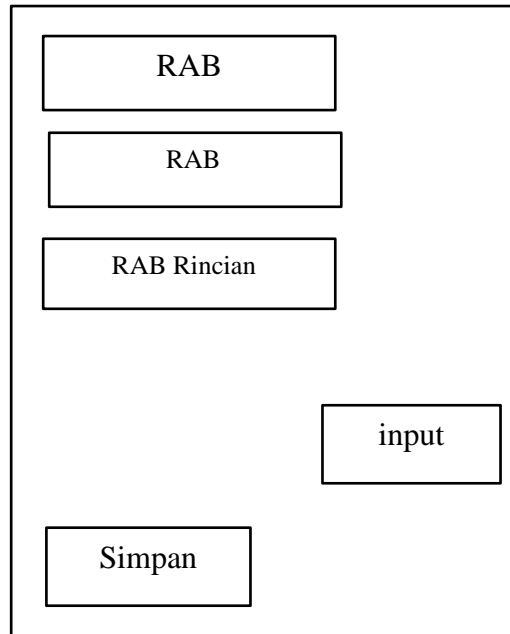
- **Rancang Bangun Menu RAP**



Gambar 4.17 Menu RAP

Gambar 4.17 diatas merupakan tampilan menu RAP yang menampilkan beberapa menu diantaranya: menu RAP, dan menu RAP rincian.

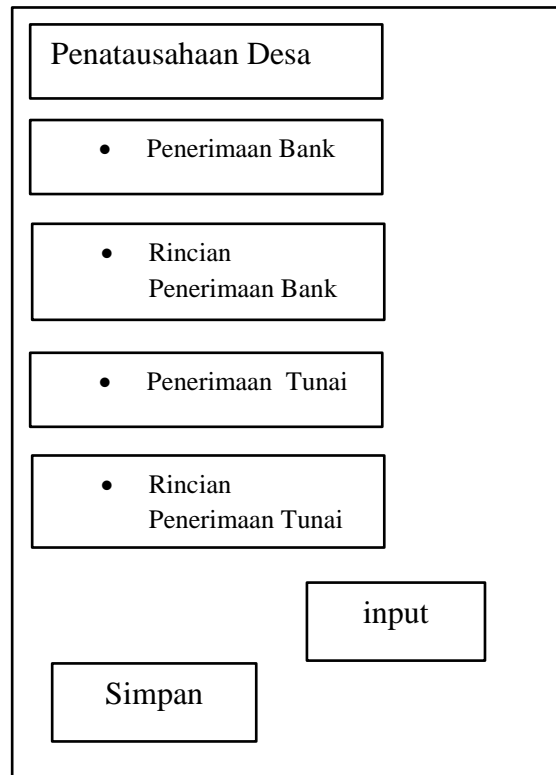
- **Rancang Bangun Menu RAB**



Gambar 4.18 Menu RAB

Gambar 4.18 diatas merupakan tampilan menu RAB yang menampilkan beberapa menu diantaranya: menu RAB, dan menu RAB rincian.

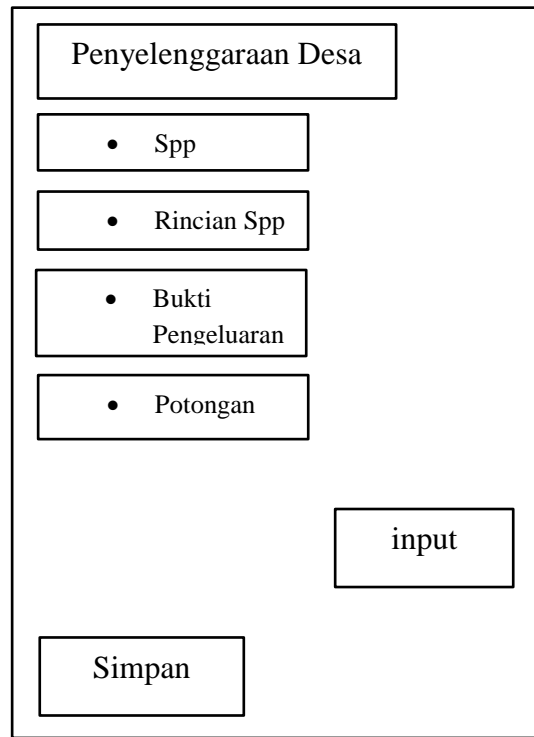
- **Rancang Bangun Menu Penatausahaan Desa**



Gambar 4.19 Menu Penatausahaan Desa

Gambar 4.19 diatas merupakan tampilan menu Penatausahaan Desa yang menampilkan beberapa menu diantaranya: menu penerimaan bank, rincian penerimaan bank, menu penerimaan tunai, dan menu rincian penerimaan tunai.

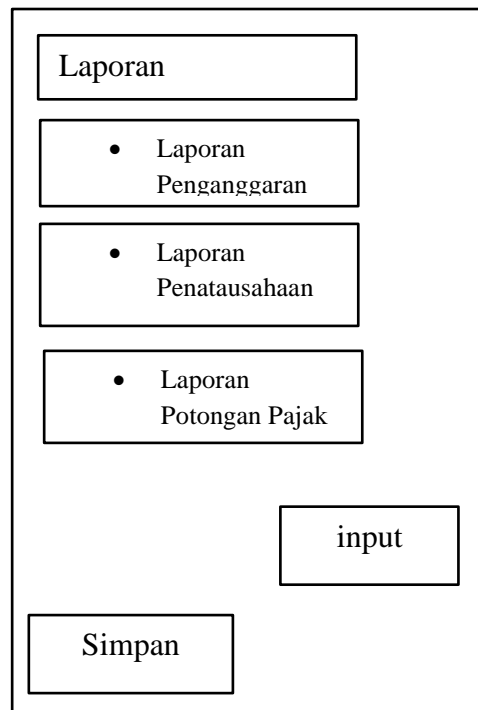
- **Rancang Bangun Menu Penyelenggaraan Desa**



Gambar 4.20 Menu Penyelenggaraan Desa

Gambar 4.20 diatas merupakan tampilan menu penyelenggaraan desa yang menampilkan beberapa menu diantaranya: menu spp, menu rincian spp, bukti pengeluaran, dan potongan.

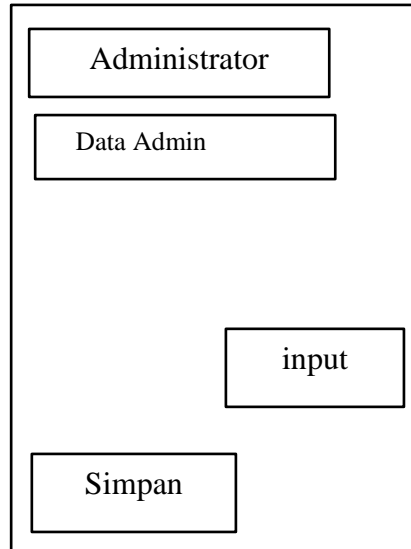
- **Rancang Menu Laporan**



Gambar 4.21 Menu Laporan

Gambar 4.21 diatas merupakan tampilan menu Laporan yang menampilkan beberapa menu diantaranya: menu laporan penganggaran, menu laporan penatausahaan, dan laporan potongan pajak.

- **Rancang Bangun Menu Administrator**



Gambar 4.22 Menu Administrator

Gambar 4.22 diatas merupakan tampilan menu Administrator yang menampilkan beberapa menu diantaranya: menu data admin.

BAB V

IMPLEMENTASI DAN PENGUJIAN

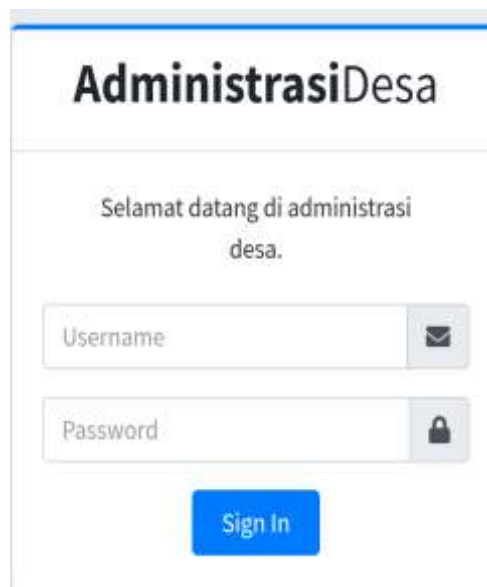
1.5 Implementasi

Implementasi adalah tahap penerapan sistem berdasarkan hasil analisis dan perancangan yang dilakukan pada bab IV. Pada bab V ini merupakan implementasi hasil rancangan menjadi Administrasi Desa Pada Kantor Desa Baros menggunakan PHP, MySQL, dan XAMPP untuk memudahkan bendahara desa dalam mengelola data keuangan desa.

5.1.1 Implementasi User Interface

Berikut adalah beberapa gambar hasil implementasi *user interface* :

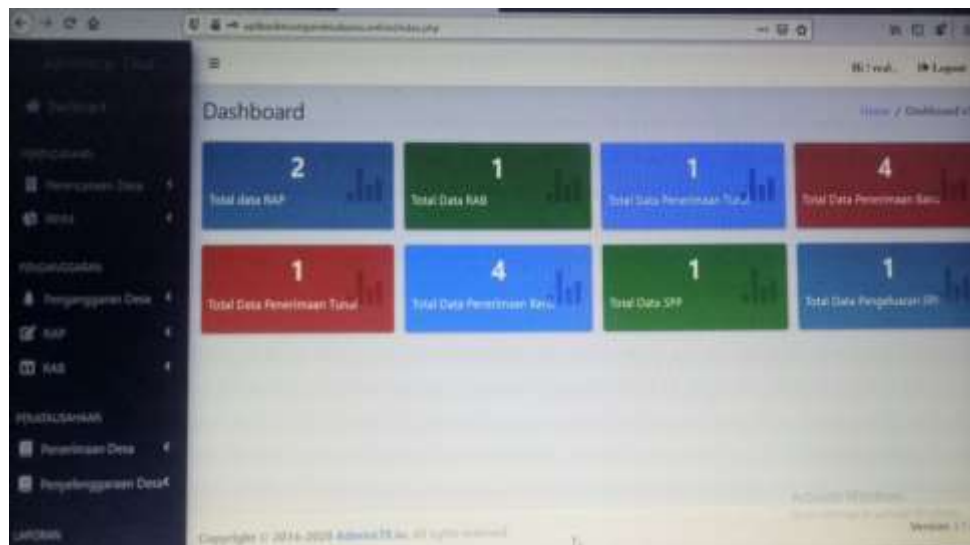
1. Tampilan halaman login



Gambar 5.0 Tampilan Login

Gambar 5.0 diatas merupakan tampilan menu *login* pada aplikasi keuangan desa. Terdapat dua kolom untuk mengisi user name, *password* dan terdapat menu *signin* untuk masuk kedalam aplikasi.

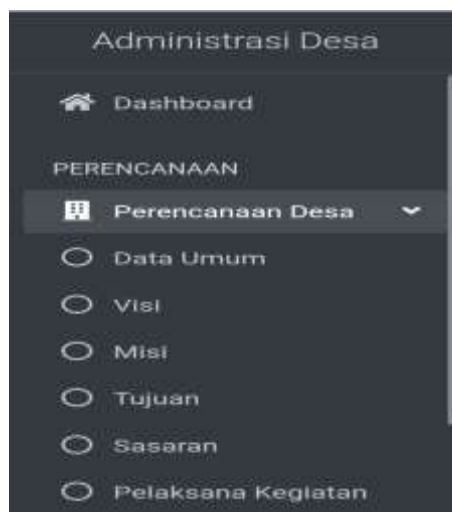
2. Tampilan *Interface Dashboard* Aplikasi Administrasi Desa



Gambar 5.1 *interface Dasboard* Aplikasi Administrasi Desa

Gambar 5.1 diatas merupakan gambar *interface dashboard* aplikasi keuangan desa, yang memuat beberapa menu seperti menu perencanaan, perencanaan desa, RPJM, Penganggaran, penganggaran desa, RAP, RAB, penatausahaan penerimaan desa, penyelenggaraan desa, laporan dan admin.

1. Tampilan Menu Perencanaan Desa

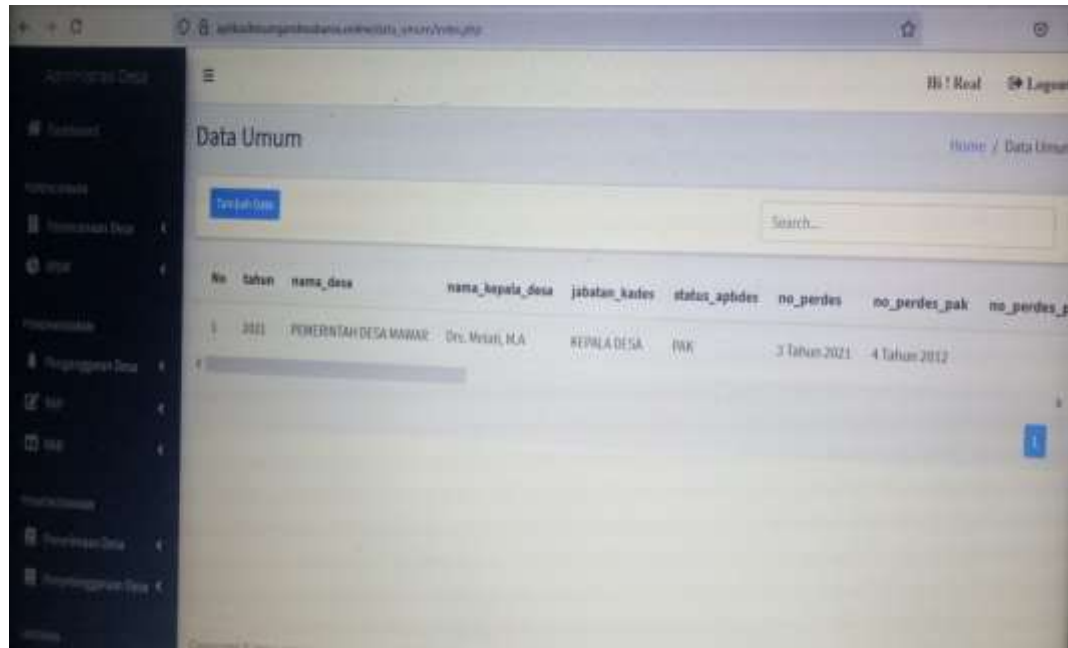


Gambar 5.2 Menu Perencanaan Desa

Pada Gambar 5.2 diatas merupakan menu perencanaan desa, yang terdapat beberapa sub menu yang terdiri dari: menu data umum, menu visi, menu misi, menu tujuan, menu sasaran dan menu pelaksanaan.

Berikut ini beberapa tampilan from dari sub menu data umum, menu visi, menu misi, menu tujuan, menu sasaran, dan menu pelaksanaan yaitu sebagai berikut:

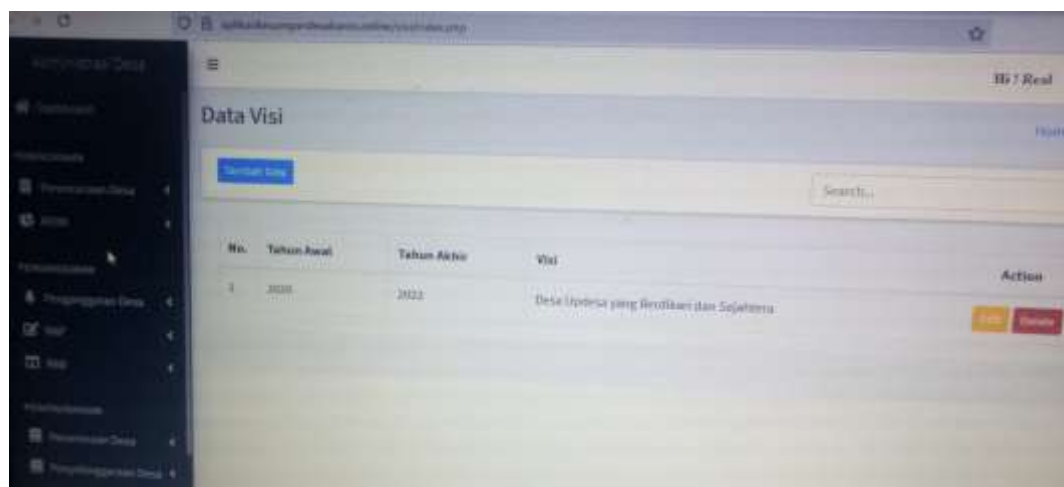
a. Tampilan From Data umum



Gambar 5.3 From Data Umum

Pada from data umum bendahara dapat meng *input* data nama desa, nama kepala desa, status apebedes, no apebedes, dan no perdes.

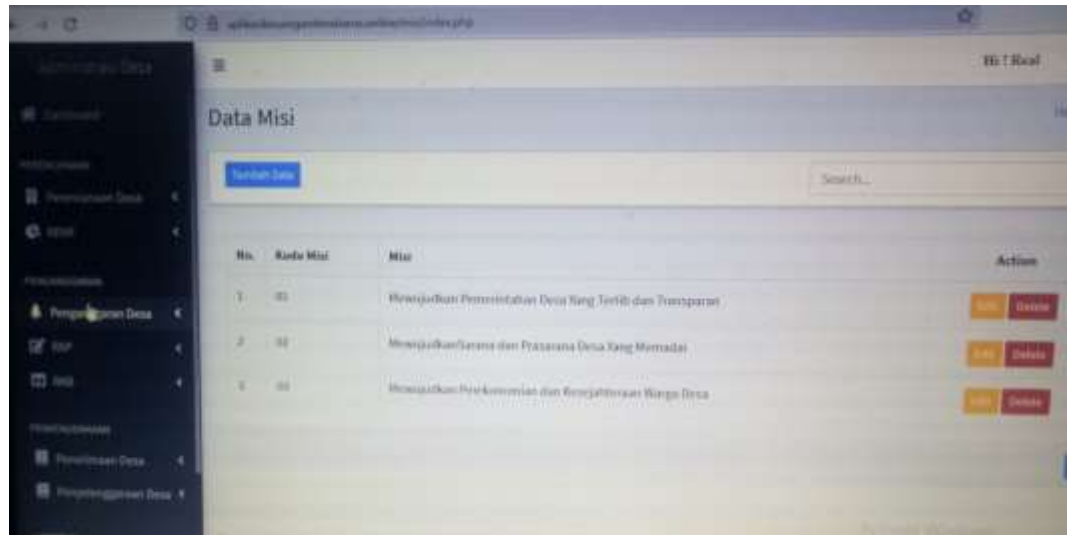
b. Tampilan From Visi



Gambar 5.4 From Data Visi

Pada from visi, bendahara dapat meng *input* data tahun, data tahun ahir, dan data visi.

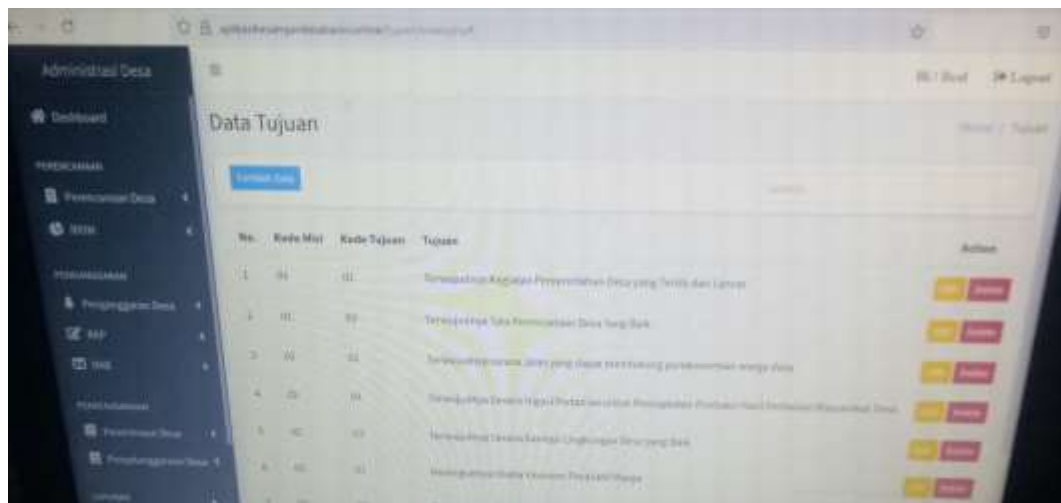
c. Tampilan From Data Misi



Gambar 5.5 From Data Misi

Pada from misi, bendahara dapat meng *input* data kode misi dan misi.

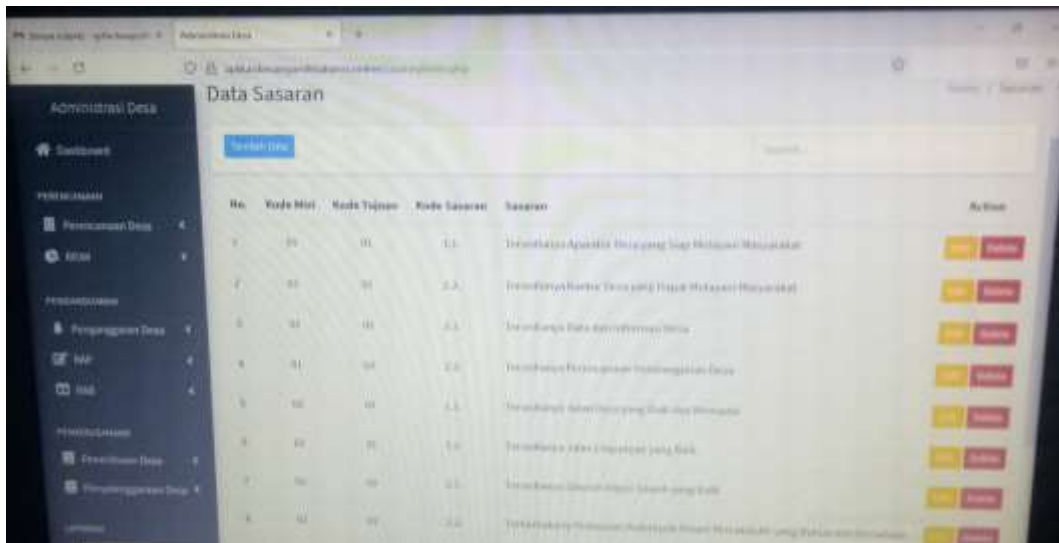
d. Tampilan From Data Tujuan



Gambar 5.6 From Data Tujuan

Pada from data tujuan, bendahara dapat meng *input* data kode misi, data kode tujuan, dan sasaran.

e.From Data Sasaran

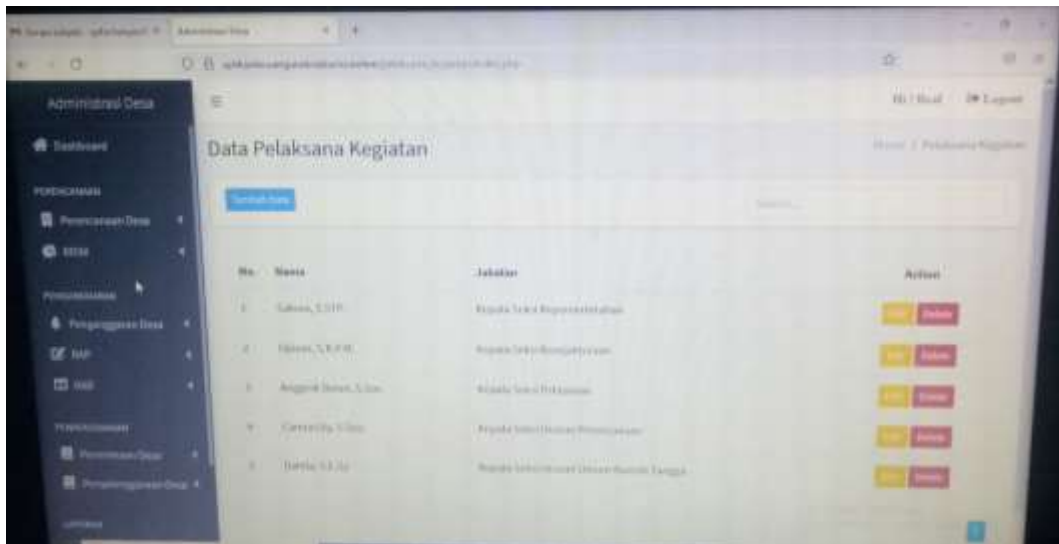


No.	Kode Misi	Kode Tujuan	Kode Sasaran	Sasaran	Action
1	01	01	1.1	Desain dan Aplikasi Perangkat Lunak Desa	Tambah Hapus
2	01	01	1.2	Desain dan Aplikasi Perangkat Lunak Desa	Tambah Hapus
3	01	01	1.3	Desain dan Aplikasi Perangkat Lunak Desa	Tambah Hapus
4	01	01	1.4	Desain dan Aplikasi Perangkat Lunak Desa	Tambah Hapus
5	01	01	1.5	Desain dan Aplikasi Perangkat Lunak Desa	Tambah Hapus
6	01	01	1.6	Desain dan Aplikasi Perangkat Lunak Desa	Tambah Hapus
7	01	01	1.7	Desain dan Aplikasi Perangkat Lunak Desa	Tambah Hapus
8	01	01	1.8	Desain dan Aplikasi Perangkat Lunak Desa	Tambah Hapus

Gambar 5.7 From Data Sasaran

Pada from data sasaran, bendahara dapat meng *input* data kode misi, kode tujuan, kode sasaran dan sasaran.

f.From Menu Pelaksanaan Kegiatan

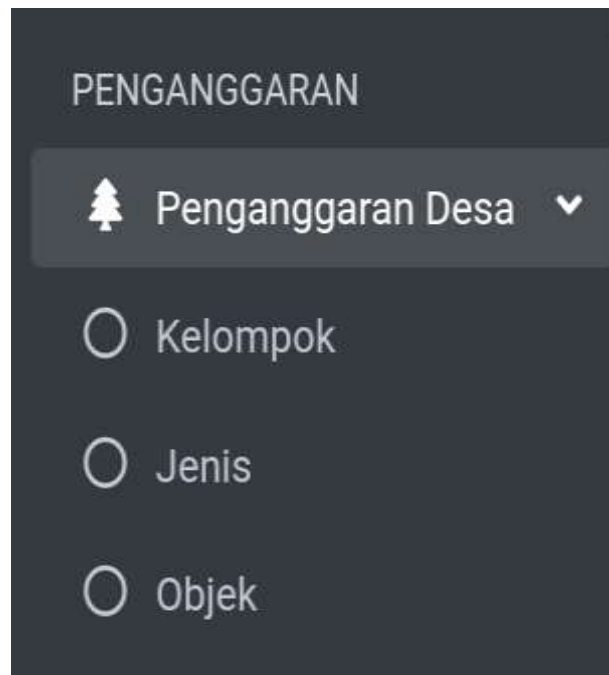


No.	Nama	Jabatan	Action
1	Salwa, S.S.P.	Kepala Seksi Administrasi	Tambah Hapus
2	Salwa, S.S.P.	Kepala Seksi Administrasi	Tambah Hapus
3	Salwa, S.S.P.	Kepala Seksi Administrasi	Tambah Hapus
4	Salwa, S.S.P.	Kepala Seksi Administrasi	Tambah Hapus
5	Salwa, S.S.P.	Kepala Seksi Administrasi	Tambah Hapus
6	Salwa, S.S.P.	Kepala Seksi Administrasi	Tambah Hapus

Gambar 5.8 From Data Pelaksanaan Kegiatan

Pada menu pelaksanaan kegiatan, bendahara dapat meng *input* data nama, dan jabatan.

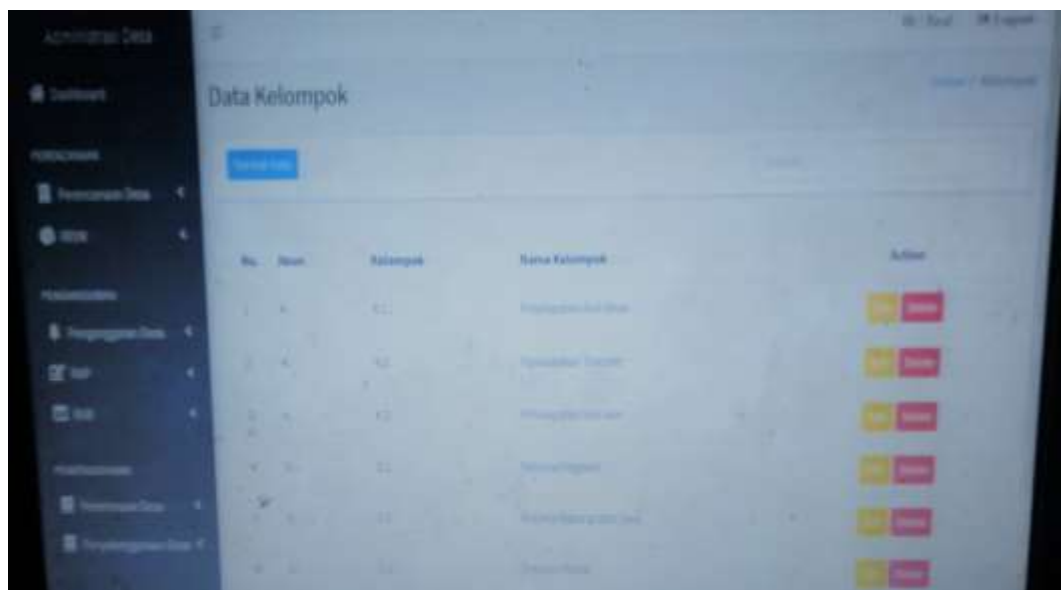
2. Tampilan Menu Penganggaran Desa



Gambar 5.9 Menu Penganggaran Desa

Pada menu penganggaran desa, terdapat sub menu kelompok, jenis, dan objek.

a. Tampilan From Menu Data Kelompok



Gambar 5.10 From Menu Data Kelompok

Pada from menu kelompok, dapat meng *input* data akun, data kelompok, dan

nama kelompok.

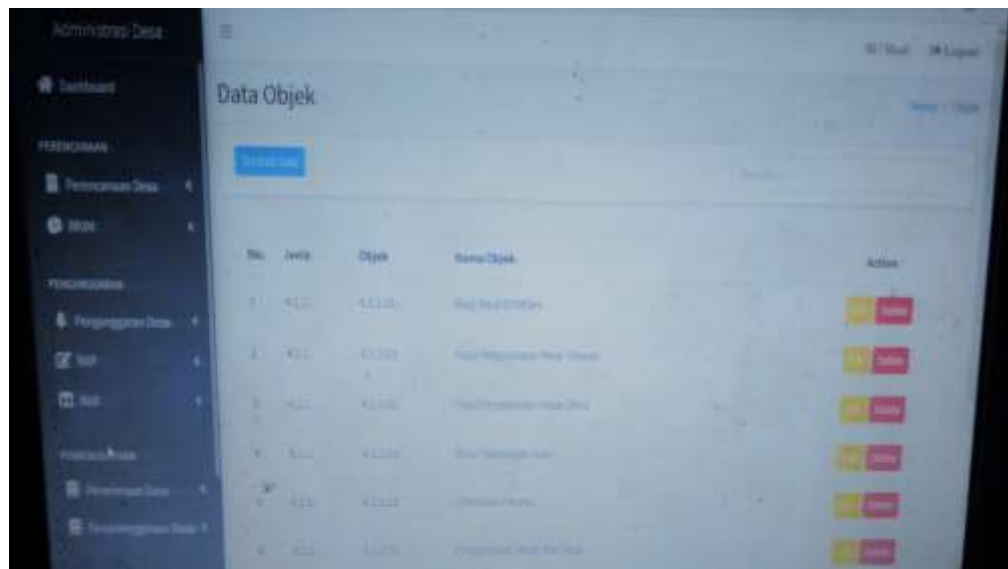
b. Tampilan Menu Data Jenis



Gambar 5.11 From Menu Data Jenis

Pada from menu data jenis, dapat meng *input* data kelompok, data jenis, dan data nama jenis.

c. From Data Objek

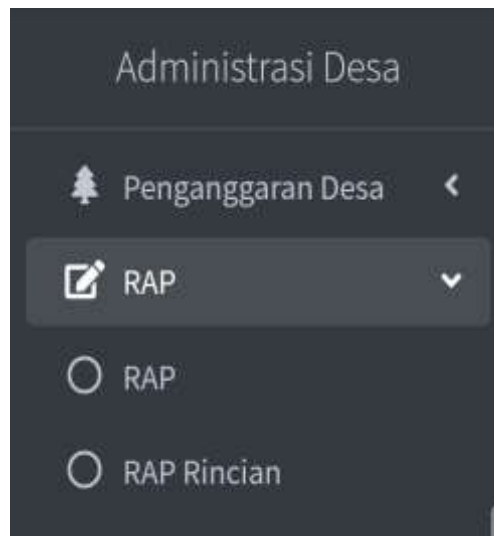


Gambar 5.12 From Menu Data Objek

Pada from menu data objek, dapat meng *input* data jenis, data objek, data nama

Objek.

d.Sub Menu RAP



Gambar 5.13 Sub Menu RAP

Pada Sub Menu RAP, terdapat menu Rap dan RAP rincian

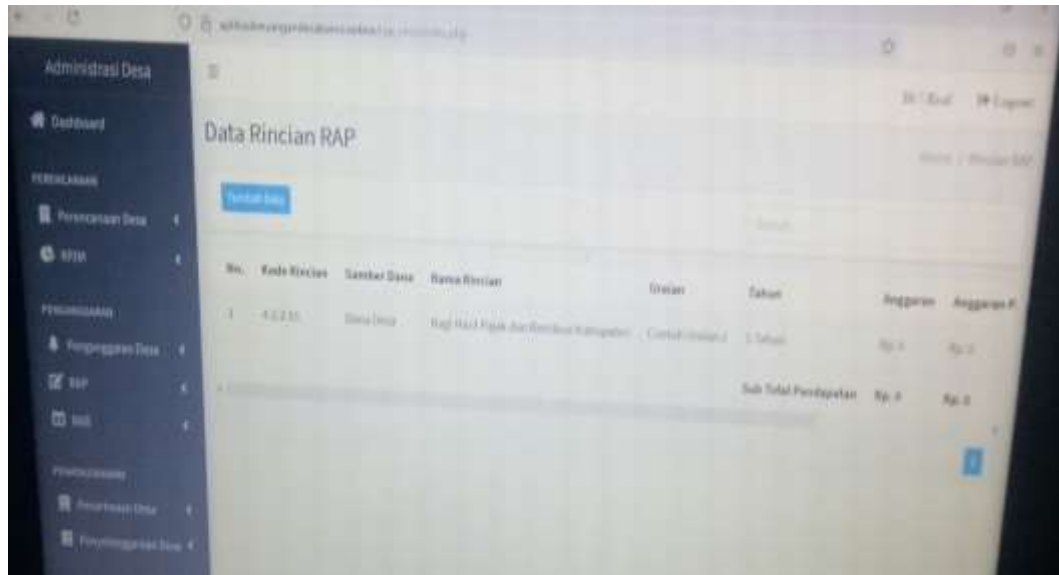
e.Tampilan From Data RAP



Gambar 5.14 From Data RAP

Pada sub menu data RAP, dapat meng *input* data kelompok data jenis, data objek, data anggran, data anggaran pkk, dan jumlah.

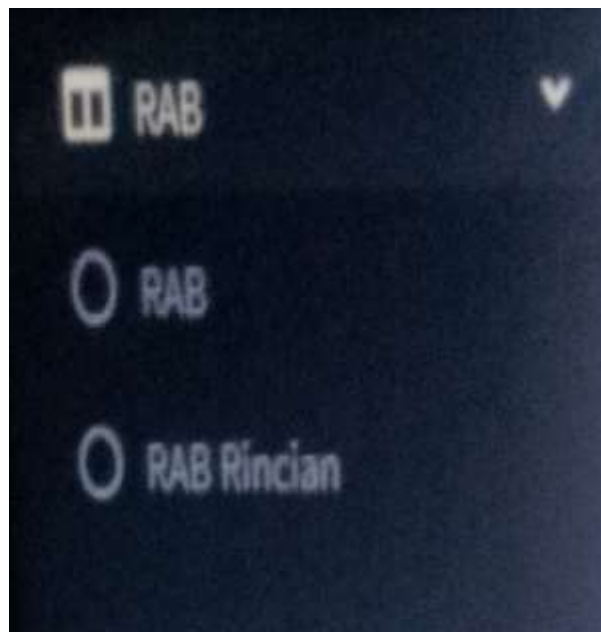
f. Tampilan From Data RAP Rincian



Gambar 5.15 From Data RAP Rincian

Pada sub menu from data RAP rincian, dapat meng *input* data kode rincian, sumber dana, uraian, tahun, dan anggaran.

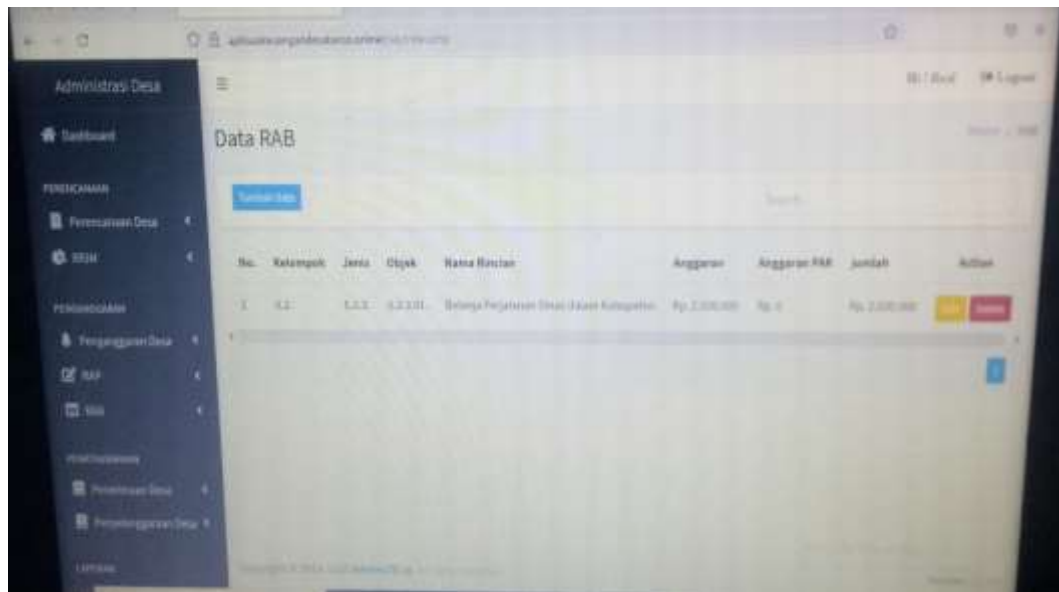
g. Gambar Sub Menu RAB



Gambar 5.16 Sub Menu RAB

Pada Menu RAB, terdapat sub menu RAB dan RAB rincian

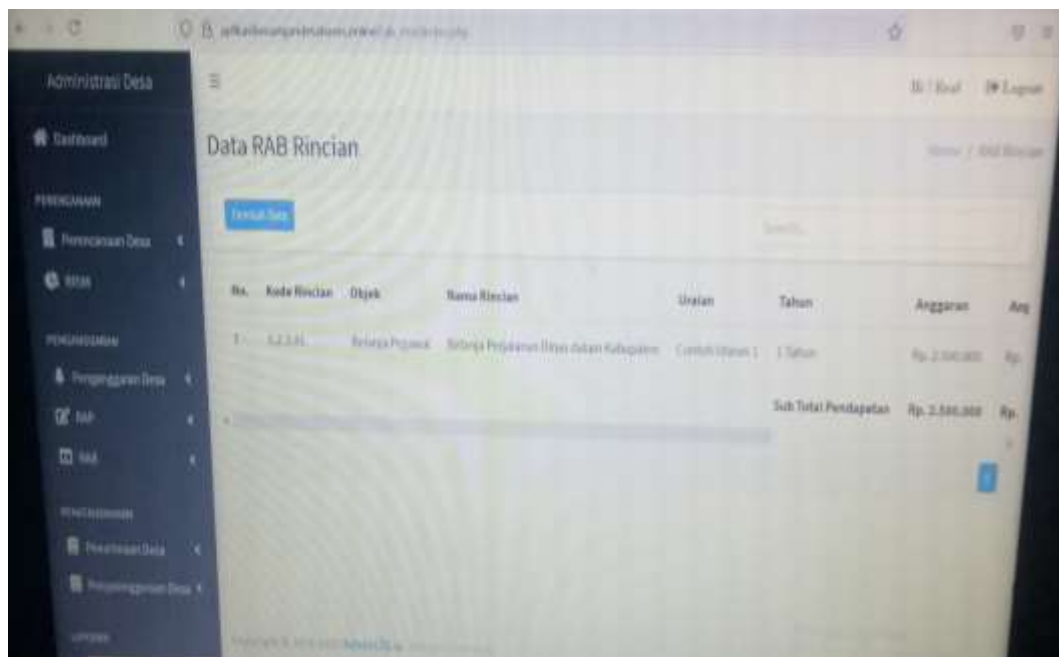
h. Tampilan From Data RAB



Gambar 5.17 From Data RAB

Pada sub menu from data RAB, dapat meng *input* data kelompok, data jenis, data objek, nama rincian, anggaran dan jumlah.

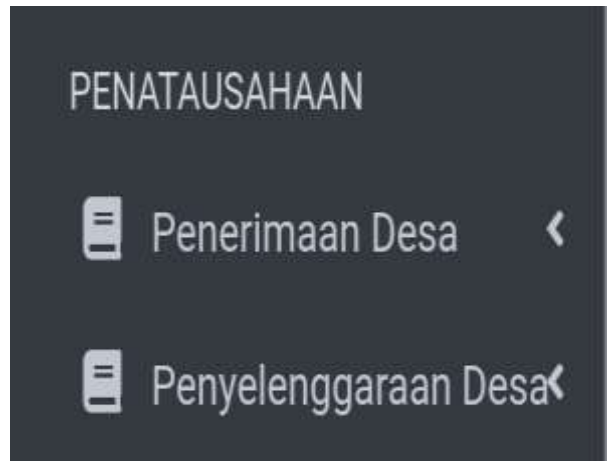
i. Tampilan From Data RAB Rincian



Gambar 5.18 Tampilan From Data RAB Rincian

Pada sub menu from data RAB rincian, dapat meng *input* data kode rincian, objek, nama rincian, uraian, tahun dan anggaran.

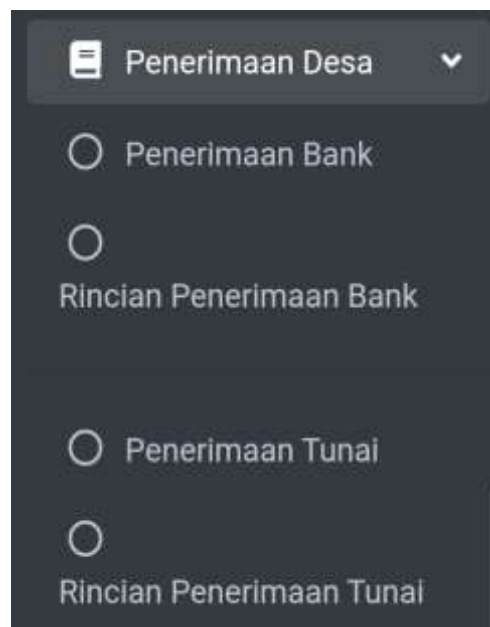
5. Tampilan Menu Penatausahaan



Gambar 5.19 Tampilan Menu Penatausahaan

Pada menu penatausahaan, terdapat sub menu penerimaan desa dan penyelenggaraan desa.

a. Tampilan Menu Penerimaan Desa



Gambar 5.20 Menu Penerimaan Desa

Pada menu penerimaan desa, terdapat sub menu penerimaan bank, rincian penerimaan bank, penerimaan tunai, dan rincian penerimaan tunai.

b. Tampilan From Penerimaan Bank

No.	Uraian	Nama	Alamat	TTS	Rekening	Nama Bank	Jumlah	Akt
1	Contoh Uraian 1	Contoh Nama 1	Jl. Sempoa, No. 123	001-001-00001	001-001-00001	Bank Mandiri	Rp. 10.000.000	
2	Contoh Uraian 2	Contoh Nama 2	Jl. Sempoa, No. 123	001-001-00002	001-001-00002	Bank Mandiri	Rp. 10.000.000	
3	Contoh Uraian 3	Contoh Nama 3	Jl. Sempoa, No. 123	001-001-00003	001-001-00003	Bank Mandiri	Rp. 10.000.000	
4	Contoh Uraian 4	Contoh Nama 4	Jl. Sempoa, No. 123	001-001-00004	001-001-00004	Bank Mandiri	Rp. 10.000.000	
Sub Total Penerimaan							Rp. 40.000.000	

Gambar 5.21 From Penerimaan Bank

Pada from penerimaan bank, dapat meng *input* uraian, alamat, no rekening, nama bank, dan jumlah.

c. Tampilan From Rincian Penerimaan Bank

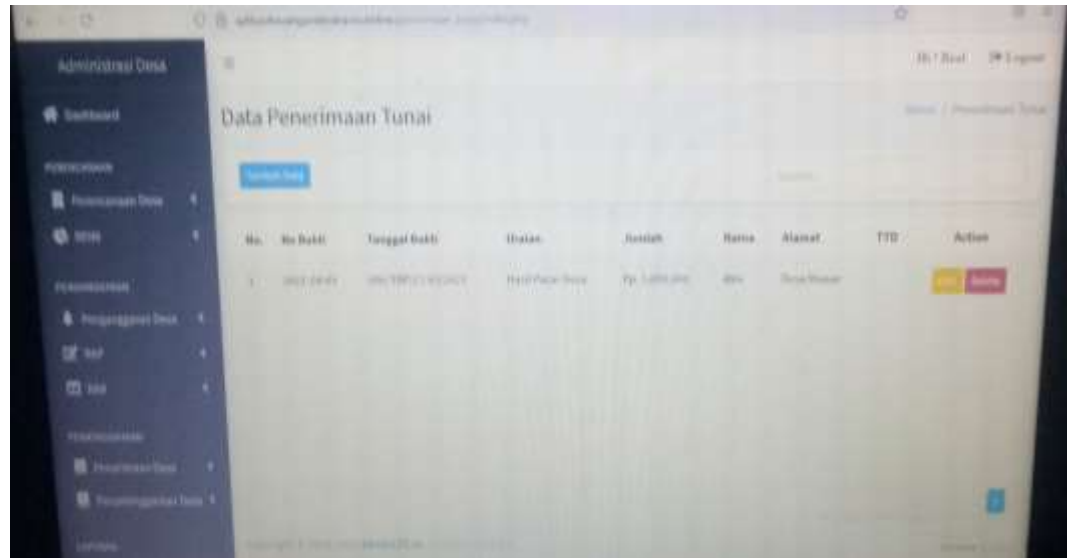
No.	Tanggal	No Bukti	Kode Rincian	Sumber Dana	Nama Rekening	Uraian	Nama	Alamat
1	2023-04-10	902/100/01.01.001	4.2.1.01	Pendapatan Transfer	Dana Desa	Contoh Uraian 1	Contoh Nama 1	Jl. Sempoa
2	2023-04-10	902/100/01.01.001	4.2.1.01	Pendapatan Asli Desa	Dana Desa	Contoh Uraian 2	Contoh Nama 2	Jl. Sempoa

Gambar 5.22 Tampilan From Rincian Penerimaan Bank

Pada from rincian penerimaan bank, dapat meng *input* tanggal, no bukti,

sumber dana, nama barang, uraian dan alamat.

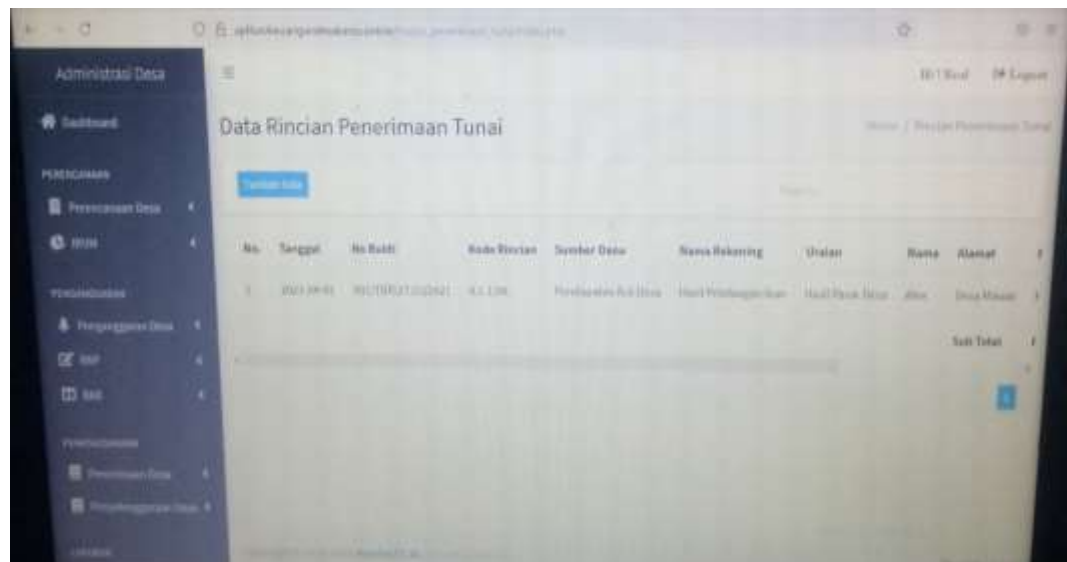
d. Tampilan From Data Penerimaan Tunai



Gambar 5.23 Tampilan From Data Penerimaan Tunai

Pada sub menu data penerimaan tunai, dapat meng *input* no bukti, tanggal bukti, uraian, jumlah, nama, alamat, dan ttd.

e. Tampilan From Rincian Penerimaan Tunai

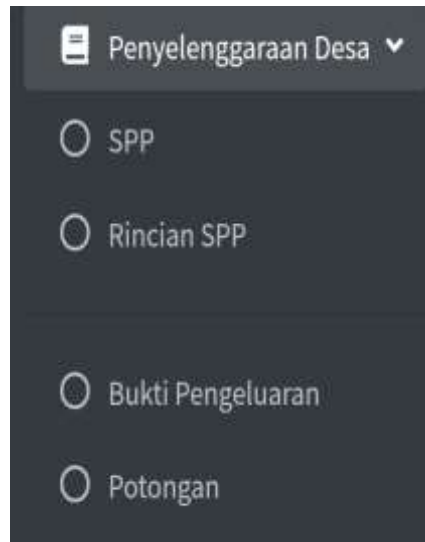


Gambar 5.24 Tampilan From Rincian Penerimaan Tunai

Pada sub menu from rincian penerimaan tunai, dapat meng *input* tanggal, no bukti, kode rincian, sumber dana, nama rekening, uraian, nama,

dan alamat.

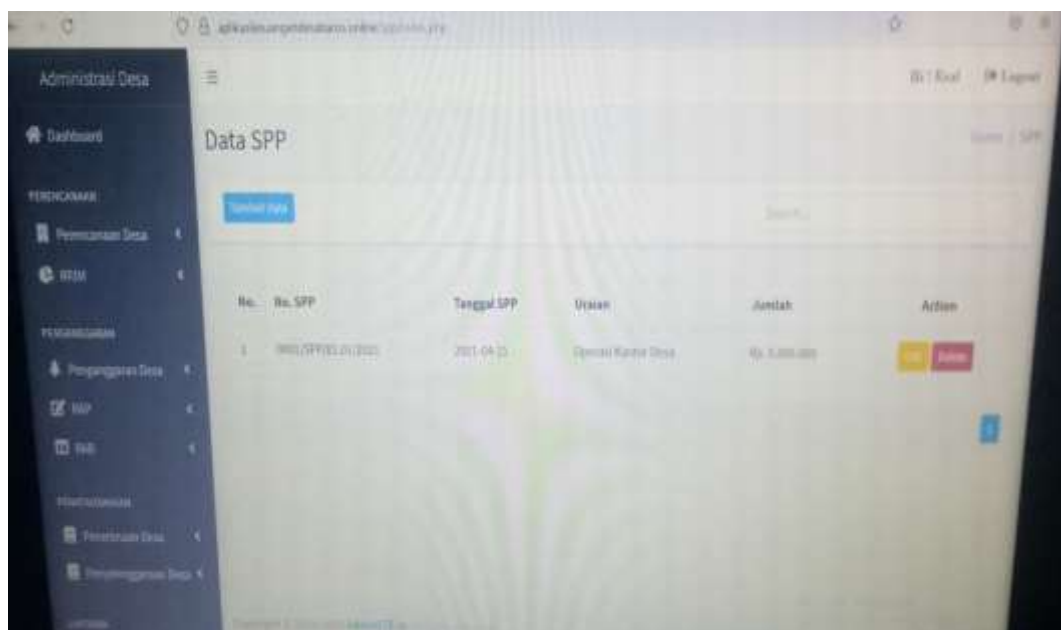
f. Tampilan Menu Penyelenggaraan Desa



Gambar 5.25 Menu Penyelenggaraan Desa

Pada menu penyelenggaraan desa, terdapat sub menu spp, rincian spp, bukti pengeluaran, dan potongan.

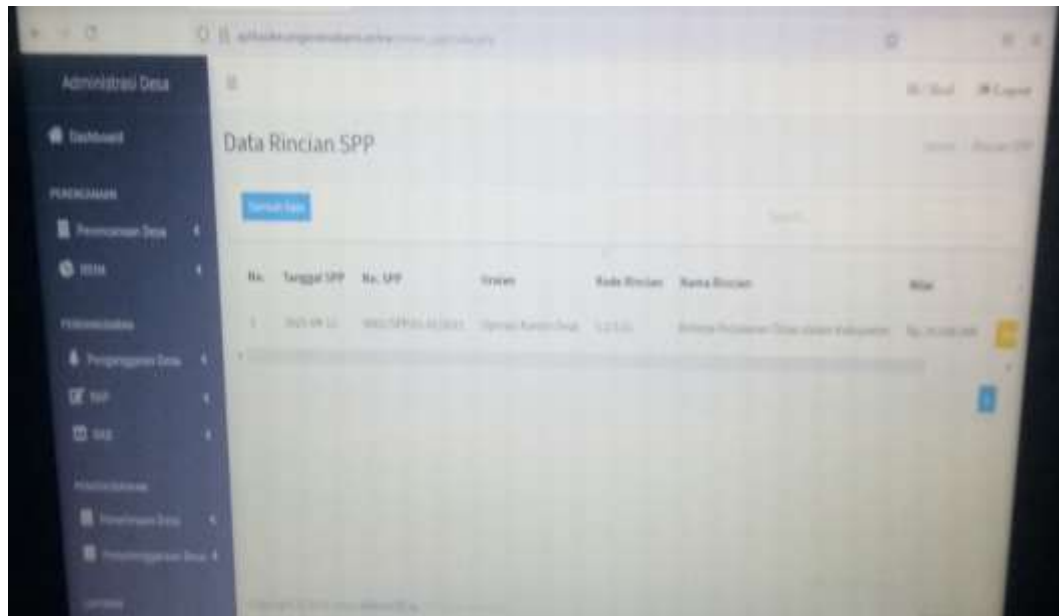
g. Tampilan From Data SPP



Gambar 5.26 Tampilan From Data SPP

Pada from data spp, dapat meng *input* no spp, tanggal spp, uraian, dan jumlah.

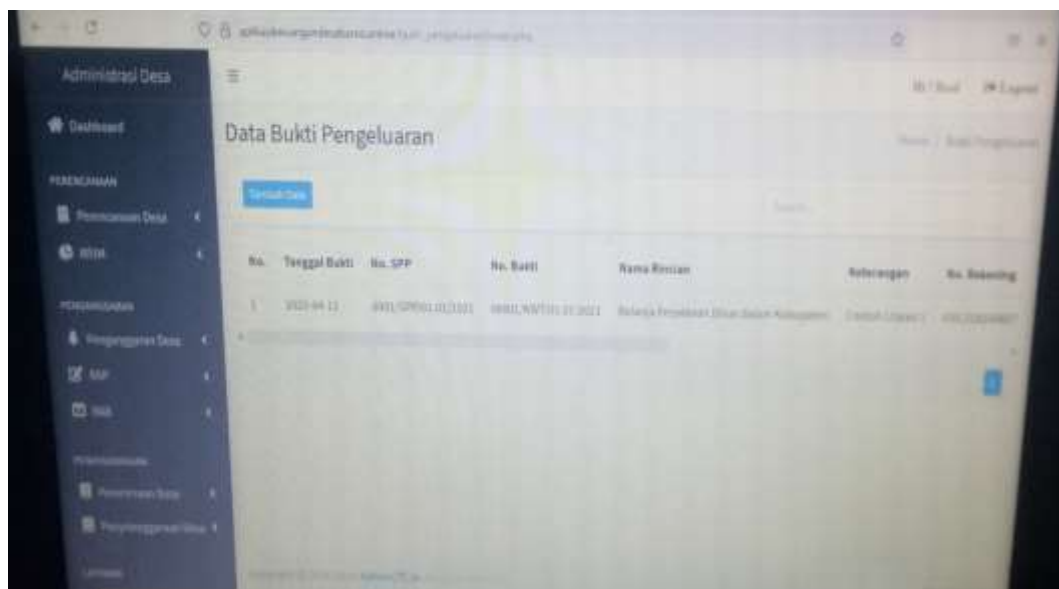
h. Tampilan From Data Rincian SPP



Gambar 5.27 Tampilan From Data Rincian SPP

Pada from rincian data spp, dapat meng *input* tanggal spp, no spp, uraian, kode rincian, nama rincian dan hasil.

i. Tampilan From Bukti Pengeluaran

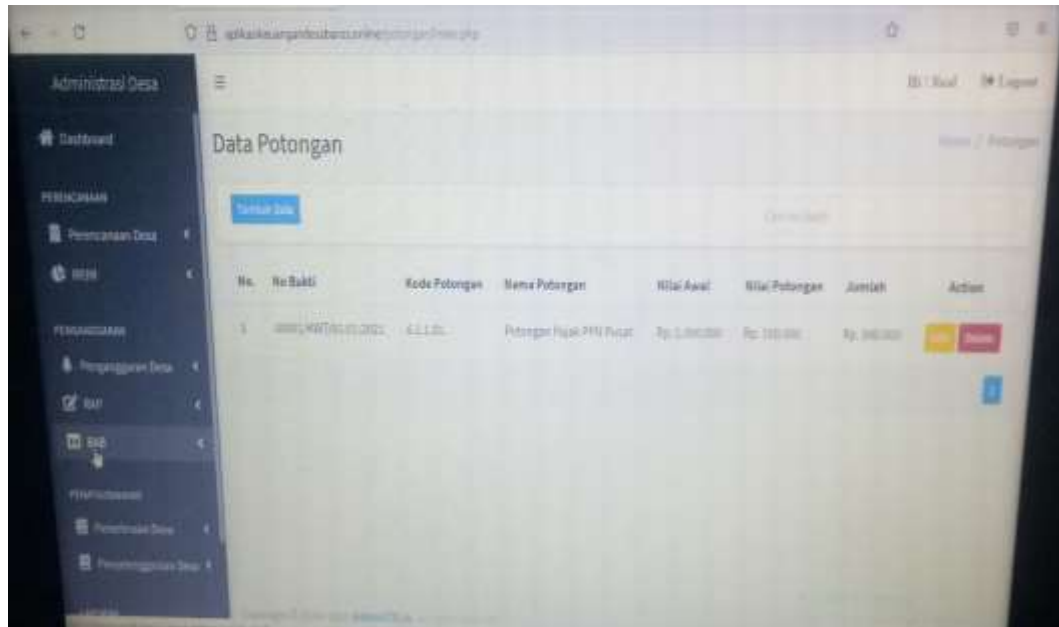


Gambar 5.28 Tampilan From Bukti Pengeluaran

Pada from bukti pengeluaran, dapat meng *input* tanggal bukti, no spp, no

bukti, nama rincian, keterangan dan no rekening.

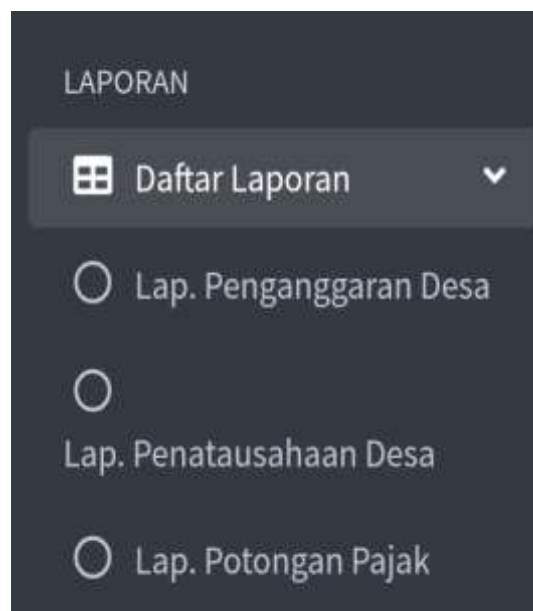
j. Tampilan From Potongan



Gambar 5.29 Tampilan From Potongan

Pada from potongn pajak, dapat meng *input* data no bukti, kode potongan, nilai awal, nilai potongan, dan jumlah.

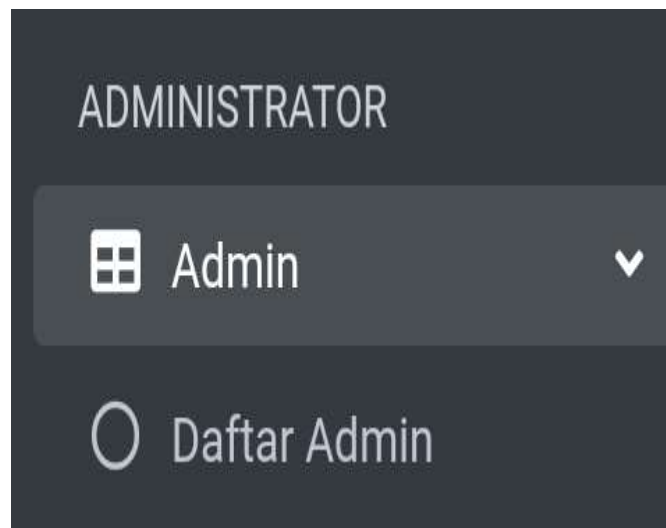
k. Tampilan Menu Laporan



Gambar 5.30 Tampilan Menu Laporan

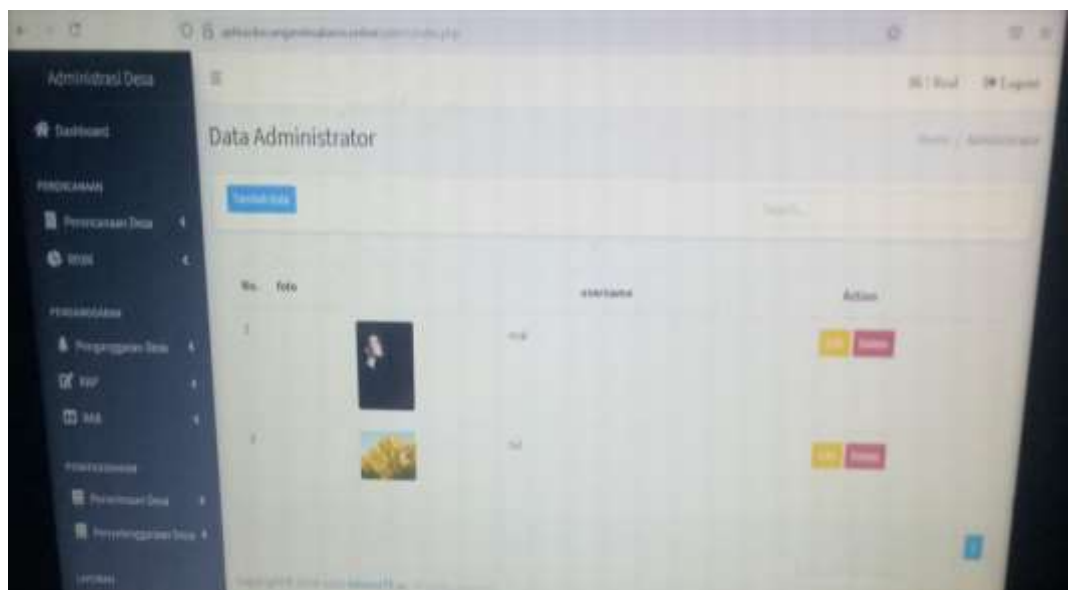
Pada menu laporan, terdapat sub menu laporan penganggaran desa, laporan

c. Tampilan Menu Administrator



Gambar 5.33 Menu Administrator

d. Tampilan Data Administrator



Gambar 5.34 Data Administrator

1.5.2 Implementasi Bahasa Pemograman Php

Pada Penelitian ini, peneliti menggunakan bahasa pemograman Php. Karena php mendukung berbagai macam database. Php mudah dipelajari dan berjalan secara efisien di server. Php mendukung banyak jumlah protocol besar seperti POP3,IMAP, dan LADP. PHP 4 menambahkan dukungan untuk bahasa pemograman Java dan arsitektur objek yang terdistribusi (COM dan CORBA).

1.5.3 Implementasi XAMPP

Pada penelitian ini, peneliti menggunakan software xampp. Penggunaan dari xampp sangat dibutuhkan untuk dapat mengembangkan software atau pun tampilan website dengan lebih mudah, cepat, dan terstruktur. Terdapat tiga komponen penyusun utama dari tool ini, yaitu HTDoc, Control Panel, dan PhpMy Admin.penggunaan aplikasi web server ini sebagai tool bantuan untuk mulai belajar tahapan pengembangan perangkat lunak sesuai kebutuhan atau proyek bisnis.

1.5.4 Hasil Implementasi

Hasil implementasi adalah untuk menerapkan dan mewujudkan rencana yang telah disusun menjadi bentuk nyata. Adapun keseluruhan implementasi meliputi: pengkodean terhadap basis data, antar muka, coding aplikasi, perancangan sistem dan lain-lain. Hasil dari tahapan tersebut adalah Aplikasi Administrasi Desa yang dapat diakses melalui website desa baros.

1.5.5 Pengujian

Pada penelitian ini, penulis menggunakan *Black Box Testing* untuk menguji aplikasi. *Black Box Testing* merupakan pengujian yang dapat dilakukan dengan melakukan pengamatan, pada hasil eksekusi melalui beberapa data uji dan memeriksa fungsional yang terdapat pada perangkat lunak. Jadi dapat kita dianalogikan seperti halnya kita melihat kedalam kotak hitam, sehingga kita hanya bisa melihat tampilan luarnya saja tanpa kita tau apa yang ada di dalam kotak hitam tersebut. Sehingga sama seperti halnya dengan *Black Box Testing* yang

hanya dapat mengevaluasi dari tampilan luarnya dan fungsionalitasnya. Tanpa harus mengetahui apa sesungguhnya yang terjadi dalam proses detailnya, yakni pada aplikasi yang seharusnya dilakukan. Menggunakan deskripsi eksternal dalam perangkat lunak, termasuk spesifikasi persyaratan, dan desain untuk digunakan dalam uji kasus, pengujian ini dapat menjadikan suatu fungsional atau non fungsional meskipun pengujian tersebut sering fungsional.

Pengujian *Black Box* akan dilakukan dengan cara yang relative bertentangan dengan kebutuhan yang ada dan memasukan sistem dapat menangani semua masukan yang tidak sesuai. Oleh karena itu, pengguna hanya bisa memasukan data yang benar ke dalam sistem. Pengujian ini berusaha menemukan kesalahan misalnya, seperti:

Fungsi-fungsi yang tidak benar atau hilang terdapat pada perangkat lunak.

1. Kesalahan dalam *interface* perangkat lunak
2. Kesalahan dalam struktur data atau akses database eksternal dalam perangkat lunak.
3. Permasalahan dalam kerja perangkat lunak.
4. Inisialisasi dan kesalahan terminasi perangkat lunak.

Berikut ini merupakan table Black Box Testing yang bisa digunakan dalam melakukan pengujian perangkat lunak, yaitu sebagai berikut:

Tabel 5.36 Black Box Testing

Skenario Pengujian	Kasus Pengujian	Hasil Yang diharapkan	Hasil Pengujian	Kesimpulan
Tambah data keuangan	Masukan data keuangan klik tombol tambah	Data keuangan berhasil di tambah	Sesuai	Normal
Edit data keuangan	Klik data yang ingin di edit klik tombol edit	Data keuangan berhasil di edit	Sesuai	Normal;
Delete data keuangan	Klik data yang ingin di hapus klik tombol hapus	Data keuangan berhasil di hapus	Sesuai	Normal

Berikut ini merupakan teknik yang sering digunakan *Black Box Testing* desain meliputi:

1. Decision Table

Decision Table merupakan cara yang tepat untuk memodelkan logika yang cukup rumit, seperti diagram alur, *if-then-else* dan *switch*

laporan kasus. Dalam kondisi ini mengaitkan dengan tindakan untuk melakukan, akan tetapi banyak kasus melakukannya dengan cara yang lebih elegan.

2. All-Pairs Testing

All-Pairs Testing atau disebut pairwise testing merupakan metode pengujian perangkat lunak kombinatorial yang digunakan untuk setiap pasangan parameter yang masuk kedalam sistem atau algoritma yang ada pada perangkat lunak.

3. State Transition Table

State Transition Table merupakan teori automata dan logika skuensial, pada table yang menunjukkan *state* dalam pengujian. Pada dasarnya sebuah table *state* merupakan table kebenaran yang digunakan untuk beberapa *input* dan *output* termasuk dengan *state* berikutnya dengan kondisi yang sebenarnya terjadi.

4. Equivalence Partitioning

Equivalence Partitioning merupakan teknik yang membagi data masukan dari beberapa unit perangkat lunak menjadi beberapa partisi data dari mana *test case* dapat diturunkan. Pada prinsipnya, uji kasus ini dirancang untuk menutupi setiap partisi minimal.

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Penerapan aplikasi Administrasi Desa dalam pengoperasiannya telah sesuai prosedur dari BPKP, namun belum optimal. Faktor utama yang menghambat penerapan Administrasi Desa adalah lemahnya sumber daya manusia dan terbatasnya fasilitas yang memadai. Pelatihan berkelanjutan terhadap perangkat desa sangat diperlukan untuk meningkatkan kemampuan dan daya ingatnya terkait pengoperasian aplikasi Administrasi Desa. Sebaiknya juga menghadirkan tenaga kerja (dari kota) yang kompeten di bidang akuntansi pemerintahan untuk menyusun laporan keuangan berbasis aplikasi Administrasi Desa sehingga tujuan penerapan aplikasi Administrasi Desa tersebut bisa tercapai. Kemudian, perlu dianggarkan dana untuk pengadaan komputer sebagai fasilitas desa, terlebih lagi pembangunan kantor desa. Hal ini bertujuan agar pelaksanaan pemerintahan desa dalam pengelolaan keuangan bisa berjalan secara efektif dan efisien.

Kemudian di Desa Baros menerapkan prinsip patuh dan taat dalam melaksanakan tugas dan tanggungjawab serta bersikap adil dalam menyampaikan informasi sehingga semua proses pengerjaan sudah sangat baik bahkan penerapan aplikasi siskeudes merupakan aplikasi yang sangat baik yang telah menghasilkan kualitas akuntabilitas keuangan desa dan mewujudkan tata kelola desa yang baik. Berdasarkan Kualitas Akuntabilitas Keuangan Desa Di Desa Baros setelah penerapan aplikasi Administrasi Desa sangat membawa perubahan yang baik di Desa Baros, mempermudah dalam proses pelaporan pertanggungjawaban dan memberikan hasil peningkatan kualitas desa yang baik, menghasilkan pelaporan keuangan yang sesuai dengan waktu yang ditentukan, dan dalam penerapan aplikasi Administrasi Desa di Desa Baros menghasilkan laporan keuangan

yang akuntabel.

Berdasarkan penelitian yang dilakukan penulis melalui beberapa tahapan yang dilakukan pada bab-bab sebelumnya, maka penulis dapat menyimpulkan bahwa:

1. Aplikasi Administrasi Desa dapat meng input data perencanaan, penganggaran dan penatausahaan.
2. Aplikasi Administrasi Desa dapat mencetak laporan penganggaran desa, dan laporan penatausahaan.
3. Aplikasi Administrasi Desa dapat digunakan oleh bendahara desa, untuk memudahkan dalam mengelola keuangan desa.

6.2 Saran

Berdasarkan hasil penelitian, maka peneliti merekomendasikan atau menyarankan beberapa hal, yaitu :

1. Pelatihan berkelanjutan terhadap bendahara desa sangat diperlukan untuk meningkatkan kemampuan dan daya ingatnya terkait pengoperasian aplikasi administrasi desa.
2. Perlu dianggarkan dana untuk pengadaan komputer dan wifi sebagai fasilitas desa.
3. Aplikasi administrasi desa ini, bisa dikembangkan menjadi aplikasi berbasis android.

DAFTAR PUSTAKA

Moedralis 2016. Analisis Pengelolaan Keuangan Dan Kekayaan Desa
jurnalkatalogisISSN 2302-2019. Volume 3. Univerisitas Muhammadiyah
Surakarta.

Gusti Ayu Trisha Sulina, ¹Made Arie Wahyuni, ²Putu Sukma Kurniawan
e-Journal Universitas Pendidikan Ganesha Jurusan Akuntansi Program S1
(Vol:8 No:2 Tahun 2017) PERANAN SISTEM KEUANGAN DESA
(SISKEUDES) TERHADAP KINERJA PEMERINTAH DESA

Arif Rivan¹) & Irfan Ridwan Maksum²) *Jurnal Administrasi Publik (Public
Administration Journal)*, (2) Desember Hadi, S. (2019).Penerapan
SistemKeuangan Desa.

Febi Nugraha Sifa C¹, Deni Iskandar Z², Idang Nurodin³Hadi, S. (2019). Analisis
Penerapan Aplikasi Sistem Keuangan Desa Jimek : Jurnal Ilmiah Mahasiswa
Ekonomi Volume 03 Nomor 02 Tahun 2020.

Basara. (2014). *Pengertian Aplikasi Pembukuan dan Pelaporan Informasi
Akuntansi Keuangan Berbasis PSAK 45: pada Panti Asuhan Al-Amanah,
Palembang. Aptekmas, 1* (2), 20-29.

Enterprise. (2015). *Pemograman bootstrap. Administrasi Publik Teori dan
Aplikasi Good Governance.*

Enterprise. (2015). *Pemograman bootstrap.*

Hadi, S. (2019). No Title. *JURNAL SAINS DAN INFORMATIKA (Aplikasi
Manajemen Keuangan Berbasis Web Dinas Komunikasi Dan Informatika
Kabupaten Tanah Laut),. 5.*

Handoko. (2015). *Pengertian HTML*.

Menurut Sony. (2009). No Title. *Implementasi Ilmu Manajemen Dalam Mewujudkan Pembangunan Masjid Raya Abdul Kadim, Yayasan Ar-Rohim, Kab. Musi Banyuasin, Propinsi Sumatra Selatan. Jurnal Abdi Masyarakat Humanis, 1(1)*.

Munawar. (2005). *Pengertian Diagram Class*.

Nasari, I. (2017). *Siklus Pengelolaan Keuangan Desa*.

Nurcholis. (2018). No Title. *Petunjuk Pelaksanaan Bimbingan Dan Konsultasi Pengelolaan Keuangan Desa. Tim Penyusun Deputy Bidang Pengawasan Penyelenggaraan Keuangan Daerah., 81*.

Pemendagri. (2006). *Pengertian administrasi desa. 32. Peraturan Menteri Dalam Negeri Tahun 2014 tentang Pengelolaan Keuangan Desa*.

Rahum. (2015). *Perspektif Pengelolaan Keuangan Dan Ekonomi Desa. Malang. Universitas Brawijaya., 21*(Prajudi Adismita (2014:21) Perspektif Pengelolaan Keuangan Dan Ekonomi Desa. Malang. Universitas Brawijaya.)

Rosa dan Shalahuddin. (2018). *No Title. 161*.

Sartono. (2011). No Title. *Pengaruh Personality Traits Terhadap Perencanaan Keuangan Keluarga. Jurnal Dinamika Manajemen. Vol 4, No 2, Univ Semarang, 50*.

Sibero, A. F. . (2011). *No Title. 2180*.

Sonny. (2009). *Manajemen Keuangan. Manajemen keuangan Journal of Business and Banking Vol. 3 No 1*

Sutrisno. (2003). *Manajemen Keuangan*.

Widjaja. (2005). Otonomi Desa,155. *Perencanaan Keuangan Komunitas Miskin Di Perkampungan Vatutela. Jurnal Academica. Fisip Untad. Vol 3 No 01 Februari, 155..*

LAMPIRAN

Codding Login

```
<?php
require_once 'function.php';

?>
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Administrasi Desa</title>

  <!-- Google Font: Source Sans Pro -->
  <link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700&disp
lay=fallback">
  <!-- Font Awesome -->
  <link rel="stylesheet" href="../assets/plugins/fontawesome-free/css/all.min.css">
  <!-- icheck bootstrap -->
  <link rel="stylesheet" href="../assets/plugins/icheck-bootstrap/icheck-
bootstrap.min.css">
  <!-- Theme style -->
  <link rel="stylesheet" href="../assets/css/adminlte.min.css">
</head>

<body class="hold-transition login-page">
  <div class="login-box">
    <!-- /.login-logo -->
    <div class="card card-outline card-primary">
      <div class="card-header text-center">
        <h2><b>Administrasi</b>Desa</h2>
```

```

</div>
<div class="card-body">
  <?php if (isset($_SESSION['failed'])) : ?>
    <div class='alert alert-danger'>
      <?= $_SESSION['failed']; ?>
    </div>
  <?php unset($_SESSION['failed']);
endif; ?>
  <p class="login-box-msg">Selamat datang di administrasi desa.</p>

  <form action="function.php" method="post">
    <div class="input-group mb-3">
      <input type="text" name="username" class="form-control"
placeholder="Username">
      <div class="input-group-append">
        <div class="input-group-text">
          <span class="fas fa-envelope"></span>
        </div>
      </div>
    </div>
    <div class="input-group mb-3">
      <input type="password" name="password" class="form-control"
placeholder="Password">
      <div class="input-group-append">
        <div class="input-group-text">
          <span class="fas fa-lock"></span>
        </div>
      </div>
    </div>
    <div class="row d-flex justify-content-center">

      <!-- /.col -->
      <div class="col-4">
        <button type="submit" name="login" class="btn btn-primary btn-block">Sign
In</button>

```

```

        </div>
        <!-- /.col -->
    </div>
</form>
</div>
<!-- /.card-body -->
</div>
<!-- /.card -->
</div>
<!-- /.login-box -->

<!-- jQuery -->
<script src="../assets/plugins/jquery/jquery.min.js"></script>
<!-- Bootstrap 4 -->
<script src="../assets/plugins/bootstrap/js/bootstrap.bundle.min.js"></script>
<!-- AdminLTE App -->
<script src="../assets/dist/js/adminlte.min.js"></script>
</body>

</html>

```

Function Aplikasi Administrasi Desa

```

<?php
    session_start();
    require_once '../config/get_connection.php';
    function getAll()
    {
        $exe = pagination()["exe"];
        while ($row = mysqli_fetch_assoc($exe)) {
            $data[] = [
                'id_admin' => $row['id_admin'],
                'username' => $row['username'],
                'password' => $row['password'],
                'foto' => $row['foto'],
            ];
        }
    }

```

```

];
}

return $data;
}
function pagination()
{
    $min_data = 10;
    $page = isset($_GET['page']) ? $_GET['page'] : 1;
    $start = ($page > 1) ? ($page * $min_data) - $min_data : 0;
    $result_page = mysqli_query(connect(), "SELECT * FROM
admin");

    $total = mysqli_num_rows($result_page);
    $total_page = ceil($total / $min_data);

    $query = "SELECT * FROM admin ORDER BY id_admin ASC
LIMIT $start, $min_data";
    $exe = mysqli_query(connect(), $query);

    $output = [
        'exe' => $exe,
        'total_page' => $total_page,
        'page' => $page
    ];
    return $output;
}

function GetById($id)
{
    $query = "SELECT * FROM `admin` WHERE `id_admin` =
'$id'";

    $exe = mysqli_query(connect(), $query);
    while ($data = mysqli_fetch_array($exe)) {
        $datas[] = array(
            'id_admin' => $data['id_admin'],

```

```

        'username' => $data['username'],
        'password' => $data['password'],
        'foto' => $data['foto'],

    );
}
return $datas;
}

function GetBySearch($search)
{
    $query = "SELECT * FROM `admin` WHERE username LIKE
'%$search%' OR password LIKE '%$search%' OR foto LIKE '%$search%' ";
    $exe = mysqli_query(connect(), $query);
    while ($data = mysqli_fetch_array($exe)) {
        $datas[] = array(
            'id_admin' => $data['id_admin'],
            'username' => $data['username'],
            'password' => $data['password'],
            'foto' => $data['foto'],

        );
    }
    return $datas;
}

function insert()
{
    $username = strtolower(stripslashes($_POST['username']));
    $password = mysqli_real_escape_string(connect(),
$_POST['password']);

    $nama_file = $_FILES['foto']['name'];
    $source = $_FILES['foto']['tmp_name'];
    $folder = '../assets/img/admin/';

```

```

        // Cek user
        $query = "SELECT * FROM `admin` WHERE `username` =
$username";

        $exe = mysqli_query(connect(), $query);
        if (mysqli_fetch_assoc($exe)) {
            echo "Username sudah terpakai";
            return false;
        }

        move_uploaded_file($source, $folder . $nama_file);

        $password = password_hash($password,
PASSWORD_DEFAULT);

        $query = "INSERT INTO `admin`
(id_admin`,`username`,`password`,`foto`)
VALUES (NULL,'$username','$password','$nama_file')";
        $exe = mysqli_query(connect(), $query);
        if ($exe) {
            // kalau berhasil
            $_SESSION['success'] = " Data added! ";
            header("Location: index.php");
        } else {
            $_SESSION['failed'] = " Data failed to add ";
            header("Location: index.php");
        }
    }

    function Update($id)
    {
        $username = $_POST['username'];

        $nama_file = $_FILES['foto']['name'];
        $source = $_FILES['foto']['tmp_name'];
        $folder = '../assets/img/admin/';

```

```

if ($nama_file != "") {

    move_uploaded_file($source, $folder . $nama_file);
    $query = "UPDATE `admin` SET `username` =
'$username', `foto` = '$nama_file' WHERE `id_admin` = '$id'";
    $exe = mysqli_query(connect(), $query);
    if ($exe) {
        // kalau berhasil
        $_SESSION['success'] = " Data added! ";
        header("Location: index.php");
    } else {
        $_SESSION['failed'] = " Data failed to add ";
        header("Location: index.php");
    }
}

$query = "UPDATE `admin` SET `username` = '$username'
WHERE `id_admin` = '$id'";
$exe = mysqli_query(connect(), $query);
if ($exe) {
    // kalau berhasil
    $_SESSION['success'] = " Data updated! ";
    header("Location: index.php");
} else {
    $_SESSION['failed'] = " Data failed to updated ";
    header("Location: index.php");
}
}

function Delete($id)
{
    $query = "DELETE FROM `admin` WHERE `id_admin` =
'$id'";

    $exe = mysqli_query(connect(), $query);

```

```

        if ($exe) {
            // kalau berhasil
            $_SESSION['success'] = " Data deleted! ";
            header("Location: index.php");
        } else {
            $_SESSION['failed'] = " Data failed to delete ";
            header("Location: index.php");
        }
    }
    if (isset($_POST['insert'])) {
        insert();
    } else if (isset($_POST['update'])) {
        update($_POST['id_admin']);
    } else if (isset($_POST['delete'])) {
        delete($_POST['id_admin']);
    } else if (isset($_POST['search'])) {
        GetBySearch($_POST['search']);
    }
?>

```

Codding Admin

<?php

```
require_once 'function.php';
```

```

if (!isset($_SESSION["login"])) {
    header("Location: ../login/index.php");
    exit;
}

```

```

require_once('../templates/header.html');
require_once('../templates/navbar.html');

```



```
require_once('../templates/sidebar.html');
```

```
?>
```

```
<!-- Content Wrapper. Contains page content -->
```

```
<div class="content-wrapper">
```

```
    <!-- Content Header (Page header) -->
```

```
    <section class="content-header">
```

```
        <div class="container-fluid">
```

```
            <div class="row mb-2">
```

```
                <div class="col-sm-6">
```

```
                    <h1>Data Administrator</h1>
```

```
                </div>
```

```
                <div class="col-sm-6">
```

```
                    <ol class="breadcrumb float-sm-right">
```

```
                        <li class="breadcrumb-item"><a
```

```
href="../index.php">Home</a></li>
```

```
                        <li class="breadcrumb-item
```

```
active">Create Admin</li>
```

```
                    </ol>
```

```
                </div>
```

```
            </div>
```

```
        </div><!-- /.container-fluid -->
```

```
    </section>
```

```
<!-- Main content -->
```

```
<section class="content">
```

```
    <div class="container-fluid">
```

```
        <div class="row">
```

```
            <!-- left column -->
```

```
            <div class="col-md-12">
```

```
                <!-- general form elements -->
```

```

<div class="card card-primary">
    <div class="card-header">
        <h3 class="card-title">Form
Admin</h3>

    </div>
    <!-- /.card-header -->
    <!-- form start -->
    <form action='function.php'
method='post' enctype="multipart/form-data">
        <div class=" card-body">
            <div class="form-
group">
                <label
for="">username</label>
                <input
type='text' name='username' class="form-control" id="" placeholder="Masukan
username">
            </div>
            <div class="form-
group">
                <label
for="">password</label>
                <input
type='text' name='password' class="form-control" id="" placeholder="Masukan
password">
            </div>
            <div class="form-
group">
                <label
for="">Foto</label>
                <input
type="file" name="foto" class="form-control-file"

```

```

id="exampleFormControlFile1" placeholder="saa">
</div>
</div>
<!-- /.card-body -->
<div class="card-footer">
<button type='submit'
name='insert' class="btn btn-primary">Submit</button>
</div>
</form>
</div>
<!-- /.card -->

</div>
<!--/.col (left) -->
</div>
<!-- /.row -->
</div><!-- /.container-fluid -->
</section>
<!-- /.content -->
</div>
<!-- /.content-wrapper -->

```

Coding Bukti Pengeluaran

```

<?php

require_once 'function.php';

if (!isset($_SESSION["login"])) {
    header("Location: ../login/index.php");
    exit;
}

```

```

require_once('../templates/header.html');
require_once('../templates/navbar.html');
require_once('../templates/sidebar.html');

$datas = getRincianSpp();
$query_pengeluaran = mysqli_query(connect(), "SELECT * FROM bukti_pengeluaran");
$jumlah = mysqli_num_rows($query_pengeluaran) + 1;
$no_pengeluaran = $jumlah."/KWT/".$date('d.m/Y');

?>

<!-- Content Wrapper. Contains page content -->
<div class="content-wrapper">
    <!-- Content Header (Page header) -->
    <section class="content-header">
        <div class="container-fluid">
            <div class="row mb-2">
                <div class="col-sm-6">
                    <h1>Data Bukti Pengeluaran</h1>
                </div>
                <div class="col-sm-6">
                    <ol class="breadcrumb float-sm-right">
                        <li class="breadcrumb-item"><a
href="../index.php">Home</a>
                        <li class="breadcrumb-item
active">Create Bukti Pengeluaran</li>
                    </ol>
                </div>
            </div>
        </div>
    </section>
    <!-- Main content -->
    <section class="content">
        <div class="container-fluid">

```

```

<div class="row">
    <!-- left column -->
    <div class="col-md-12">
        <!-- general form elements -->
        <div class="card card-primary">
            <div class="card-header">
                <h3 class="card-title">Form
Bukti Pengeluaran</h3>
            </div>
            <!-- /.card-header -->
            <!-- form start -->
            <form action='function.php'
method='post'>
                <div class=" card-body">
                    <div class="form-
group">
                        <label
for="exampleSelectRounded0">No. SPP</label>
                        <select
name="no_spp" class="custom-select rounded-0" id="exampleSelectRounded0">
                            <option>Pilih No. SPP</option>
                        <?php
foreach ($datas as $data) : ?>
                            <option value="<?= $data['no_spp']; ?>"><?= $data['no_spp']; ?> (<?=
$data['nama_rincian']; ?>)</option>
                        <?php
                    endforeach; ?>
                </select>
            </div>
            <div class="form-
group">

```

```

</label>
for="exampleSelectRounded0">Kode Rekening</label>

<select
name="kode_rekening" class="custom-select rounded-0" id="exampleSelectRounded0">

    <option>Pilih Kode Rekening</option>

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

    <option value="<?= $data['nama_rincian']; ?>"><?= $data['nama_rincian'];
?></option>

<?php
endforeach; ?>

</select>
</div>

<div class="form-
group">

    <label
for="">No Bukti</label>

    <input
type='text' name='no_bukti' class="form-control" id="" placeholder="Masukan no bukti"
value="<?= $no_pengeluaran; ?>" readonly>

</div>

<div class="form-
group">

    <label
for="">Tanggal Bukti</label>

    <input
type='date' name='tgl_bukti' class="form-control" id="" placeholder="Masukan tanggal
bukti">

</div>

<div class="form-

```

```

group">
<label
for="">Uraian</label>
<input
type='text' name='uraian' class="form-control" id="" placeholder="Masukan uraian">
</div>
<div class="form-
group">
<label
for="">Nilai Pengeluaran</label>
<input
type='text' name='nilai_pengeluaran' class="form-control" id="" placeholder="Masukan
nominal (Rp.)">
</div>
<p class="text-primary
text-bold">PENERIMA</p>
<div class="form-
group">
<label
for="">NPWP</label>
<input
type='text' name='npwp' class="form-control" id="" placeholder="Masukan npwp">
</div>
<div class="form-
group">
<label
for="">Nama</label>
<input
type='text' name='nama' class="form-control" id="" placeholder="Masukan nama">
</div>
<div class="form-

```

```

group">
<label
for="">Alamat</label>
<input
type='text' name='alamat' class="form-control" id="" placeholder="Masukan alamat">
</div>

<div class="form-
group">
<label
for="">No Rekening</label>
<input
type='text' name='no_rekening' class="form-control" id="" placeholder="Masukan no
rekening">
</div>

<div class="form-
group">
<label
for="">Nama Bank</label>
<input
type='text' name='nama_bank' class="form-control" id="" placeholder="Masukan nama
bank">
</div>

</div>
<!-- /.card-body -->
<div class="card-footer">
<button type='submit'
name='insert' class="btn btn-primary">Submit</button>
</div>
</form>
</div>
<!-- /.card -->

```



```

        </div>
        <!--/.col (left) -->

        </div>
        <!-- /.row -->

        </div><!-- /.container-fluid -->
    </section>
    <!-- /.content -->
</div>
<!-- /.content-wrapper -->

<?php require_once('..templates/footer.html'); ?>

```

Coding data umum

```

<?php

require_once 'function.php';

if (!isset($_SESSION["login"])) {
    header("Location: ../login/index.php");
    exit;
}

require_once('..templates/header.html');
require_once('..templates/navbar.html');
require_once('..templates/sidebar.html');

?>

<!-- Content Wrapper. Contains page content -->
<div class="content-wrapper">
    <!-- Content Header (Page header) -->
    <section class="content-header">
        <div class="container-fluid">
            <div class="row mb-2">

```

```

        <div class="col-sm-6">
            <h1>Form Data Umum</h1>
        </div>
        <div class="col-sm-6">
            <ol class="breadcrumb float-sm-right">
                <li class="breadcrumb-item"><a
href="..../index.php">Home</a></li>
                <li class="breadcrumb-item
active">Create Data Umum</li>
            </ol>
        </div>
    </div>
</div>
</div><!-- /.container-fluid -->
</section>

<!-- Main content -->
<section class="content">
    <div class="container-fluid">
        <div class="row">
            <!-- left column -->
            <div class="col-md-12">
                <!-- general form elements -->
                <div class="card card-primary">
                    <div class="card-header">
                        <h3 class="card-title">Data
Umum Desa</h3>
                    </div>
                    <!-- /.card-header -->
                    <!-- form start -->
                    <form action='function.php'
method='post'>

                        <div class=" card-body">
                            <div class="form-
group">
                                <label

```

```

for="">Tahun</label>
<input
type='text' name='tahun' class="form-control" id="" placeholder="Masukan Tahun">
</div>
<div class="form-
group">
<label
for="">Nama Desa</label>
<input
type='text' name='nama_desa' class="form-control" id="" placeholder="Masukan Nama
Desa">
</div>
<div class="form-
group">
<label
for="">Nama Kepala Desa</label>
<input
type='text' name='nama_kepala_desa' class="form-control" id="" placeholder="Masukan
Nama Kepala Desa">
</div>
<div class="form-
group">
<label
for="">Jabatan Kepala Desa</label>
<input
type='text' name='jabatan_kades' class="form-control" id="" placeholder="Masukan
Jabatan Kepala Desa">
</div>
<div class="form-
group">
<label
for="">Status APBdes</label>
<select
name="status_apbdes" class="custom-select rounded-0" id="exampleSelectRounded0">
<option

```

```

value="AWAL">AWAL</option>
</option>
value="PAK">PAK</option>
</select>
</div>
<div class="form-
group">
<label
for="">No Perdes</label>
<input
type='text' name='no_perdes' class="form-control" id="" placeholder="Masukan No
Perdes">
</div>
<div class="form-
group">
<label
for="">No Perdes PAK</label>
<input
type='text' name='no_perdes_pak' class="form-control" id="" placeholder="Masukan No
Perdes PAK">
</div>
<div class="form-
group">
<label
for="">No Perdes Pertanggung Jawaban</label>
<input
type='text' name='no_perdes_p_jawab' class="form-control" id=""
placeholder="Masukan No Perdes Pertanggung Jawaban">
</div>
<div class="form-
group">
<label
for="">Tanggal Perdes</label>
<input
type='text' name='tgl_perdes' class="form-control" id="" placeholder="Masukan Tanggal

```

```

Perdes">

</div>
<div class="form-
group">

<label
for="">Tanggal Perdes PAK</label>

<input
type='text' name='tgl_perdes_pak' class="form-control" id="" placeholder="Masukan
Tanggal Perdes PAK">

</div>
<div class="form-
group">

<label
for="">Tanggal Perdes Pertanggung Jawaban</label>

<input
type='text' name='tgl_perdes_p_jawab' class="form-control" id=""
placeholder="Masukan Tanggal Perdes Pertanggung Jawaban">

</div>
<div class="form-
group">

<label
for="">Nama Sekdes</label>

<input
type='text' name='nama_sekdes' class="form-control" id="" placeholder="Masukan
Sekertaris Desa">

</div>
<div class="form-
group">

<label
for="">NIP Sekdes</label>

<input
type='text' name='nip_sekdes' class="form-control" id="" placeholder="Masukan NIP
Sekertaris Desa">

</div>
<div class="form-

```

```

group">
<label
for="">Jabatan Sekdes</label>
<input
type='text' name='jabatan_sekdes' class="form-control" id="" placeholder="Masukan
Jabatan Sekertaris Desa">
</div>
<div class="form-
group">
<label
for="">Nama Kaur Keuangan</label>
<input
type='text' name='nama_kaur_keuangan' class="form-control" id=""
placeholder="Masukan Nama Kaur Keuangan">
</div>
<div class="form-
group">
<label
for="">Jabatan Kaur Keuangan</label>
<input
type='text' name='jabatan_kaur_keuangan' class="form-control" id=""
placeholder="Masukan Jabatan Kaur Keuangan">
</div>
<div class="form-
group">
<label
for="">Nama Bendahara</label>
<input
type='text' name='nama_bendahara' class="form-control" id="" placeholder="Masukan
Nama Bendahara">
</div>
<div class="form-
group">
<label
for="">Nama Jabatan Bendahara</label>

```

```

<input
type='text' name='jabatan_bendahara' class="form-control" id="" placeholder="Masukan
Nama Bendahara">

</div>
<div class="form-
group">

<label
for="">Ibu Kota Desa</label>

<input
type='text' name='ibukota_desa' class="form-control" id="" placeholder="Masukan Ibu
Kota Desa">

</div>
</div>
<!-- /.card-body -->
<div class="card-footer">
<button type='submit'
name='insert' class="btn btn-primary">Submit</button>
</div>
</form>
</div>
<!-- /.card -->

</div>
<!--/.col (left) -->
</div>
<!-- /.row -->
</div><!-- /.container-fluid -->
</section>
<!-- /.content -->
</div>
<!-- /.content-wrapper -->

<?php require_once('../templates/footer.html'); ?>

```

Codding kegiatan

```

<?php

require_once 'function.php';

if (!isset($_SESSION["login"])) {
    header("Location: ../login/index.php");
    exit;
}

require_once('../templates/header.html');
require_once('../templates/navbar.html');
require_once('../templates/sidebar.html');

$datas = GetSubBidang();
$datas2 = GetSasaran();
?>

<!-- Content Wrapper. Contains page content -->
<div class="content-wrapper">
    <!-- Content Header (Page header) -->
    <section class="content-header">
        <div class="container-fluid">
            <div class="row mb-2">
                <div class="col-sm-6">
                    <h1>Data Kegiatan</h1>
                </div>
                <div class="col-sm-6">
                    <ol class="breadcrumb float-sm-right">
                        <li class="breadcrumb-item"><a
href="../index.php">Home</a>
                        <li class="breadcrumb-item
active">Create Kegiatan</li>
                    </ol>
                </div>
            </div>
        </div>
    </section>

```



```

        </div><!-- /.container-fluid -->
    </section>

    <!-- Main content -->
    <section class="content">
        <div class="container-fluid">
            <div class="row">
                <!-- left column -->
                <div class="col-md-12">
                    <!-- general form elements -->
                    <div class="card card-primary">
                        <div class="card-header">
                            <h3 class="card-title">Quick
Example</h3>
                        </div>
                        <!-- /.card-header -->
                        <!-- form start -->
                        <form action='function.php'
method='post'>
                            <div class=" card-body">
                                <div class="form-
group">
                                    <label
for="exampleSelectRounded0">Kode Sub Bidang</label>
                                    <select
name="kode_sub_bidang" id="kode_sub_bidang" class="custom-select rounded-0"
id="exampleSelectRounded0">
                                        <option>Pilih Kode Sub Bidang </option>
                                </div>
                                <?php
foreach ($datas as $data) : ?>
                                    <option value="<?=$data['kode_sub_bidang']; ?>"><?=$data['kode_bidang'];
?>.<?=$data['kode_sub_bidang']; ?> <?=$data['sub_bidang']; ?></option>
                                </div>
                                <?php

```

```

endforeach; ?>

</select>
</div>
<div class="form-
group">

<label
for="">kode_kegiatan</label>

<input
type='text' name='kode_kegiatan' class="form-control" id="kode_kegiatan"
placeholder="Masukan Kegiatan">

</div>
<div class="form-
group">

<label
for="">Kegiatan</label>

<input
type='text' name='kegiatan' class="form-control" id="" placeholder="Masukan
Kegiatan">

</div>
<div class="form-
group">

<label
for="exampleSelectRounded0">Sasaran</label>

<select
name="sasaran" class="custom-select rounded-0" id="exampleSelectRounded0">

<option>Pilih Sasaran </option>

<?php
foreach ($datas2 as $data) : ?>

<option value="<?= $data['sasaran']; ?>"><?= $data['sasaran']; ?></option>

<?php
endforeach; ?>

</select>
</div>

```

```

<div class="form-
group">
    <label
for="">Lokasi</label>
    <input
type='text' name='lokasi' class="form-control" id="" placeholder="Masukan Lokasi">
</div>
<div class="form-
group">
    <label
for="">Keluaran</label>
    <input
type='text' name='keluaran' class="form-control" id="" placeholder="Masukan
Keluaran">
</div>
<div class="form-
group">
    <label
for="">Manfaat</label>
    <textarea
id="summernote" type='text' name='manfaat'></textarea>
</div>
<div class="form-
group">
    <label
for="">Pelaksanaan <small>(Hanya bernilai angka)</small></label>
    <select
name="pelaksanaan" class="custom-select rounded-0" id="exampleSelectRounded0">
        <option>Pilih Lama Pelaksanaan</option>
        <option
value="1">1 Tahun</option>
        <option
value="2">2 Tahun</option>
        <option

```

```

value="3">3 Tahun</option>
<option
value="4">4 Tahun</option>
<option
value="5">5 Tahun</option>
</select>
</div>
<div class="form-
group">
<label
for="">Tata Kerja</label>
<select
name="tata_kerja" class="custom-select rounded-0" id="exampleSelectRounded0">
<option>Pilih Tata Kerja</option>
<option
value="Swakelola">Swakelola</option>
<option
value="Kerjasama Antar Desa">Kerjasama Antar Desa</option>
<option
value="Pihak Ketiga">Pihak Ketiga</option>
</select>
</div>
</div>
<!-- /.card-body -->
<div class="card-footer">
<button type='submit'
name='insert' class="btn btn-primary">Submit</button>
</div>
</form>
</div>
<!-- /.card -->
</div>
<!--/.col (left) -->

```

```

        </div>
        <!-- /.row -->
    </div><!-- /.container-fluid -->
</section>
<!-- /.content -->
</div>
<!-- /.content-wrapper -->

<?php require_once('../templates/footer.html'); ?>
<script>
    $(document).ready(function() {
        $('#kode_sub_bidang').change(() => {
            let value = $('#kode_sub_bidang').val();
            $.ajax({
                url: 'function.php',
                method: 'post',
                data: {
                    getKodeSubBidangFromAjax: value
                },
                success: function(response) {
                    $('#kode_kegiatan').val($.trim(response));
                }
            });
        });
    });
</script>

```

Codding Kelompok

```
<?php
```

```

require_once 'function.php';

if (!isset($_SESSION["login"])) {
    header("Location: ../login/index.php");
    exit;
}

require_once('../templates/header.html');
require_once('../templates/navbar.html');
require_once('../templates/sidebar.html');

?>
<!-- Content Wrapper. Contains page content -->
<div class="content-wrapper">
    <!-- Content Header (Page header) -->
    <section class="content-header">
        <div class="container-fluid">
            <div class="row mb-2">
                <div class="col-sm-6">
                    <h1>Data Kelompok</h1>
                </div>
                <div class="col-sm-6">
                    <ol class="breadcrumb float-sm-right">
                        <li class="breadcrumb-item"><a
href="../index.php">Home</a>
                        <li class="breadcrumb-item
active">Create Kelompok</li>
                    </ol>
                </div>
            </div>
        </div>
    </section>

    <!-- Main content -->
    <section class="content">

```

```

<div class="container-fluid">
  <div class="row">
    <!-- left column -->
    <div class="col-md-12">
      <!-- general form elements -->
      <div class="card card-primary">
        <div class="card-header">
          <h3 class="card-title">Form
Input Kelompok</h3>

        </div>
        <!-- /.card-header -->
        <!-- form start -->
        <form action='function.php'
method='post'>

          <div class="card-body">
            <div class="form-
group">

              <label
for="">Akun</label>

              <!--
              <input
type='text' name='akun' class="form-control" id="" placeholder="Masukan Akun"> -->
              <select
name="akun" class="form-control" id="akun">

                <option
value="0">- Pilih Akun -</option>

                <option
value="4.">Pendapatan</option>

                <option
value="5.">Belanja</option>

              </select>
            </div>
            <div class="form-
group">

              <label
for="">Kelompok</label>

```

```

<input
type='text' name='kelompok' id="kelompok" class="form-control" id=""
placeholder="Masukan Kelompok" readonly>

</div>
<div class="form-
group">

<label
for="">Nama Kelompok</label>

<input
type='text' name='nama_kelompok' class="form-control" id="" placeholder="Masukan
Nama Kelompok">

</div>
</div>
<!-- /.card-body -->
<div class="card-footer">
<button type='submit'
name='insert' class="btn btn-primary">Submit</button>
</div>
</form>
</div>
<!-- /.card -->

</div>
<!--/.col (left) -->

</div>
<!-- /.row -->
</div><!-- /.container-fluid -->
</section>
<!-- /.content -->
</div>
<!-- /.content-wrapper -->

<?php require_once('../templates/footer.html'); ?>
<script>
$(document).ready(function() {

```



```
$('#akun').change() => {  
    let value = $('#akun').val();  
    $.ajax({  
        url: 'function.php',  
        method: 'post',  
        data: {  
            getAkunFromAjax: value  
        },  
        success: function(response) {  
            $('#kelompok').val($.trim(response));  
        }  
    });  
});  
});  
</script>
```

RIWAYAT HIDUP



Syifa Fauziah, lahir di Ciamis pada tanggal 16 September 1996. Anak ke dua dari tiga bersaudara, yang lahir dari pasangan Heris Bahrudin dan Nunung Nurjannah. Mulai mengenyam pendidikan tingkat dasar di SDN Baranangsiang kemudian melanjutkan pendidikan tingkat menengah pertama di MTSN CIPARAY(2012) serta melanjutkan pendidikan tingkat atas di SMK Wirakarya 2 Ciparay (2015). Untuk mendapatkan gelar sarjana, Syifa melanjutkan ke jenjang S1 di Universitas BaleBandung Fakultas Teknologi Informasi Jurusan Teknik Informatika.

*A never ending dream leands you to the great
Stage of imagination and to make
dream come true
You have to try an save to bring
The best of you life your dream!*