# APLIKASI *SCHEDULE* PENGERJAAN PROYEK *ONLINE* DINAS PU KAB. SIDENRENG RAPPANG

****

Oleh :

**Asmar Saputra**

**NIM : 216 280 098**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS MUHAMMADIYAH PAREPARE**

**2020**

# LEMBAR PERSETUJUAN

**APLIKASI *SCHEDULE* PENGERJAAN PROYEK *ONLINE* DINAS PU KAB. SIDENRENG RAPPANG**

****

**Asmar Saputra**

**216 280 098**

Setelah diperiksa/diteliti maka Skripsi ini dapat diajukan untuk

mengikuti ujian seminar Hasil

Menyetujui;

**Pembimbing I Pembimbing II**

**Syahirun Alam, S.T., M.T Marlina,S.Kom.,M.Kom**

**NBM. 883 314 NBM. 1162 680**

Mengetahui ;

**Ketua Program Studi**

|  |
| --- |
| **Syahirun Alam, S.T., M.T**  **NBM. 883 314** |

# ABSTRAK

ASMAR SAPUTRA (216 280 098)*.* Aplikasi *Schedule* Pengerjaan Proyek Online Dinas PU Kab. Sidenreng Rappang (Dibimbing oleh Syahirun Alam & Marlina).

Penelitian ini mengambil permasalahan mengenai Aplikasi *Schedule* Pengerjaan Proyek Online Dinas PU Kab. Sidenreng Rappang. Bertujuan untuk membuat program yang dapat mengoptimalkan proses pemantauan progress pengerjaan proyek. Metode penelitian yang digunakan adalah Penelitian Lapangan *(Field Research) dan Kajian* Kepustakaan *(Literature Study)*. Dibuatnya sebuah sistem pengolahan data *schedule* pengerjaan proyek yang akan memberikan kemudahan saat penginputan data dan proses pengolahannya untuk mengasilkan laporan-laporan data pengerjaan proyek. Dari hasil pengujian menggunakan *Black Box,* dengan menguji listing program diperoleh bahwa sistem berhasil berjalan setelah dilakukan pengujian penginputan data *schedul* proyek sesuai dengan fungsi aplikasi .Dari hasil pengujian *White Box* menguji algoritma program diperoleh bahwa jumlah flowgraph dengan jumlah path pada pengujian white box sama. Dengan Sistem Web Hosting sehingga dihasilkan sistem pengelolaan data *schedule* pengerjaan proyek yang diharapkan dapat membantu dalam mengelola aplikasi *schedule* pengerjaan proyek online.

Kata kunci : Sistem, *Schedule* Pengerjaan Proyek, WebHosting.

# *ABSTRACT*

*ASMAR SAPUTRA (216 280 098). Schedule of Online Project Work Application for Public Works Service Kab. Sidenreng Rappang (Supervised by Syahirun Alam & Marlina).*

*Research Schedule Retrieval of Online Project Work Application for Public Works Agency of Kab. Sidenreng Rappang. Aims to create programs that can optimize the process of monitoring the progress of project work. The research method used is Field Research (Field Studies) and Literature Review. Created a system Process data project work schedule that will provide convenience when inputting data and processing to produce reports on project work data. From the test results using the Black Box, by studying the list of programs obtained by the system which is done after inputting the project data scheduling testing in accordance with the application. With a Web Hosting System so that the resulting project management schedule management system is expected to help in managing the project work schedule online.*

*Keywords: System, Project Work Schedule, WebHosting.*

# KATA PENGANTAR

C:\Users\Samsung\Downloads\bismillah.jpg

*Assalamu Alaikum WarahMatullahi Wabarakatu*

Alhamdulillahi Rabbil Alamin,. Tiada kata paling indah diucapkan selagi puji syukur penulis panjatkan kehadirat Allah SWT atas segala Rahmat dan Taufiknya, sehingga skripsi ini dapat terselesaikan. Shalawat serta salam semoga tercurah kepada Rasulullah Muhammad SAW, keluarga dan sahabat serta pengikutnya yang setia hingga akhir zaman. Tugas akhir ini sebagai salah satu syarat yang harus dipenuhi dalam menyelesaikan studi pada Fakultas Teknik Jurusan Informatika Universitas Muhammadiyah Parepare dengan judul “Aplikasi *Schedule* Pengerjaan Proyek *Online* Dinas PU Kab. Sidenreng Rappang”.

Dengan segala kerendahan hati, penulis menyadari bahwa dalam penyelesaian tugas akhir ini tidak lepas dari bantuan dari berbagai pihak, dimana pada kesempatan ini penulis menghaturkan terima kasih yang sebesar-besarnya kepada :

1. Allah SWT, Pencipta alam semesta beserta segala isinya dan pemilik segala ilmu. Terima kasih yaa Allah atas segala skenario kehidupan yang telah engkau persiapkan untuk hamba, hanya kepadaMu hamba menyembah, hanya kepadaMu hamba meminta pertolongan dan hanya kepadaMu hamba berserah diri, segala daya dan upaya serta kemampuan datang dariMu yaa Allah, Allahu Akbar.
2. Yang terhormat Bapak Drs H. M. Nasir, M.Pd., selaku Rektor Universitas Muhammadiyah Parepare.
3. Bapak Dr. Muhammad Nasir T.S.T, M.T selaku Dekan Fakultas Teknik Universitas Muhammadiyah Parepare
4. Bapak Syahirun Alam, ST,.MT. Selaku Ketua Program Studi Teknik Informatika Fakultas Teknik Universitas Muhammadiyah Parepare.
5. Bapak Syahirun Alam, ST,.MT Selaku Dosen Pembimbing Skripsi I dan Ibu Marlina, S.Kom, M.Kom. Selaku Pembimbing II yang telah berkenan memberikan bimbingan, perhatian, pengarahan, serta saran sejak awal hingga akhir penyusunan tugas akhir ini.
6. Segenap dosen Jurusan Teknik Informatika Universitas Muhammadiyah Parepare.
7. Para dosen dan staf Fakultas Teknik, Terima kasih atas partisipasinya.
8. Kedua orang tua yang tak henti-hentinya memberikan dorongan dan motivasi sehingga tugas akhir ini dapat terselesaikan
9. Untuk teman – teman yang selalu memberikan motivasi dan dukungan langsung selama proses penyusunan skripsi di Fakultas Teknik Informatika
10. Terima kasih buat semua pihak yang terlibat dalam penyelesaian tugas akhir ini yang tidak bisa saya sebutkan namanya satu persatu.

Penulis menyadari bahwa dalam penyusunan tugas akhir ini masih terdapat berbagai kekurangan, maka segala kritikan dan saran untuk perbaikan dan pengembangan dimasa yang akan datang sangat penulis harapkan demi kesempurnaan tugas akhir ini.

Akhir kata, semoga tugas akhir ini dapat memberikan manfaat kepada para pembaca dan dapat membantu dalam penulisan dan pembuatan selanjutnya. Amien

Parepare,  Juni 2020

Asmar Saputra

# DAFTAR ISI

[SAMPUL](#_Toc44377115)

[LEMBAR PERSETUJUAN i](#_Toc44377116)

[ABSTRAK ii](#_Toc44377117)

[*ABSTRACT* iii](#_Toc44377118)

[KATA PENGANTAR iv](#_Toc44377119)

[DAFTAR ISI vii](#_Toc44377120)

[DAFTAR TABEL viii](#_Toc44377121)

[DAFTAR GAMBAR x](#_Toc44377122)

[BAB I PENDAHULUAN 1](#_Toc44377123)

[1. LATAR BELAKANG 1](#_Toc44377124)

[2. RUMUSAN MASALAH 3](#_Toc44377125)

[3. BATASAN MASALAH 3](#_Toc44377126)

[4. TUJUAN PENELITIAN 4](#_Toc44377127)

[5. MANFAAT PENELITIAN 4](#_Toc44377128)

[BAB II TINJAUAN PUSTAKA 5](#_Toc44377129)

[A. PENELITIAN TERDAHULU 5](#_Toc44377130)

[B. TINJAUAN TEORI 7](#_Toc44377131)

[1. Aplikasi 7](#_Toc44377132)

[2. Konsep Proyek 7](#_Toc44377133)

[3. Konsep Penyusunan Jadwal (*Schedule*) 12](#_Toc44377134)

[C. Tinjauan Perangkat Lunak 14](#_Toc44377135)

[1. *MySQL* 14](#_Toc44377136)

[*2.* *HTML* 15](#_Toc44377137)

[*3.* *CSS (Cascading Style Sheet)* 16](#_Toc44377138)

[*4.* *Javascript* 17](#_Toc44377139)

[*5.* *PHP* 19](#_Toc44377140)

[*6.* *Framework CodeIgniter* 20](#_Toc44377141)

[*7.* *XAMPP* 22](#_Toc44377142)

[*8.* *APACHE* 23](#_Toc44377143)

[*9.* *Unified Modeling Language (UML)* 23](#_Toc44377144)

[D. Metode Pengujian Perangkat Lunak 30](#_Toc44377145)

[E. Kerangka Fikir 32](#_Toc44377146)

[BAB III METODE PENELITIAN 33](#_Toc44377147)

[A. Tempat dan Waktu Penelitian 33](#_Toc44377148)

[B. Jenis penelitian 33](#_Toc44377149)

[C. Metode pengumpulan data 33](#_Toc44377150)

[D. Alat dan bahan penelitian 34](#_Toc44377151)

[E. Tahap penelitian 35](#_Toc44377152)

[F. Metode Pengujian 36](#_Toc44377153)

[BAB IV ANALISIS DAN PERANCANGAN SISTEM 37](#_Toc44377154)

[A. Analisis Sistem Yang Sedang Berjalan 37](#_Toc44377155)

[B. Rancangan Sistem Yang Diusulkan 37](#_Toc44377156)

[1. *Use Case* Diagram 37](#_Toc44377157)

[*2.* *Activity Diagram* 40](#_Toc44377158)

[*3.* *Sequence Diagram* 43](#_Toc44377159)

[C. Kebutuhan Pembuatan Program 46](#_Toc44377160)

[D. Rancangan *Input* Program 52](#_Toc44377161)

[E. Rancangan Output Program 60](#_Toc44377162)

[BAB V PENGUJIAN SISTEM 61](#_Toc44377163)

[A. Metode Pengujian 61](#_Toc44377164)

[B. Teknik Pengujian 61](#_Toc44377165)

[a. Pengujian *Black Box* 62](#_Toc44377166)

[b. Pengujian *White Box* 68](#_Toc44377167)

[BAB VI PENUTUP 78](#_Toc44377168)

[A. Kesimpulan 78](#_Toc44377169)

[B. Saran-saran 79](#_Toc44377170)

[DAFTAR PUSTAKA 80](#_Toc44377171)

# DAFTAR TABEL

[Tabel 2.1 Table Use Case Diagram 26](#_Toc44377043)

[Tabel 2.2 Symbol Aktivity Diagram 27](#_Toc44377044)

[Tabel 2.3 Symbol Sequence Diagram 28](#_Toc44377045)

[Tabel 4.1 Penjelasan *Use Case* Diagram 39](#_Toc44377046)

[Tabel 4.2 Tabel *Admin* 50](#_Toc44377047)

[Tabel 4.3 Tabel *User* 50](#_Toc44377048)

[Tabel 4.4 Tabel Proyek 51](#_Toc44377049)

[Tabel 4.5 Tabel *Schedule* 51](#_Toc44377050)

[Tabel 5.1 Pengujian pada *Form Login* 62](#_Toc44377051)

[Tabel 5.2 Pengujian Pada *Form Login* 62](#_Toc44377052)

[Tabel 5.3 Pengujian Pada *Form* Data Pengguna 64](#_Toc44377053)

[Tabel 5.4 Pengujian Pada *Form* Data Proyek 64](#_Toc44377054)

[Tabel 5.5 Pengujian Pada *Form* Data *Schedule Project* 65](#_Toc44377055)

[Tabel 5.6 Pengujian Pada *Form Upload* Laporan Berkala 66](#_Toc44377056)

[Tabel 5.7 Pengujian Pada *Form Upload* LPJ 66](#_Toc44377057)

[Tabel 5.8 Pengujian Pada *Form* Data Akun 67](#_Toc44377058)

[Tabel 5.9 Grafik Matriks dari *Flowgraph Login* 69](#_Toc44377059)

[Tabel 5.10 Grafik Matriks pada *Flowgraph Input* Data 71](#_Toc44377060)

[Tabel 5.11 Grafik Matriks pada *Flowgraph* Edit Data 73](#_Toc44377061)

[Tabel 5.12 Grafik Matriks pada *Flowgraph* Hapus Data 75](#_Toc44377062)

[Tabel 5.13 Grafik Matriks pada *Flowgraph* Cetak Data Laporan 77](#_Toc44377063)

# DAFTAR GAMBAR

[Gambar 2.1 CodeIgniter 22](#_Toc44377076)

[Gambar 2.2 Kerangka Pikir 32](file:///D:\TUGAS\KULIAH\SKRIPSI\ASMAR%20SAPUTRA\Skripsi.Asmar.docx#_Toc44377077)

[Gambar 4.1 *Use Case* Diagram Sistem yang Berjalan 37](#_Toc44377078)

[Gambar 4.2 *Use Case* Diagram Sistem yang diusulkan 38](#_Toc44377079)

[Gambar 4.3 *Activity Diagram Admin* 40](#_Toc44377080)

[Gambar 4.4 *Activity* Diagram Pelaksana 41](#_Toc44377081)

[Gambar 4.5 *Activity* Diagram Pengawas 42](#_Toc44377082)

[Gambar 4.6 *Sequence Diagram* Aplikasi *Admin* 43](#_Toc44377083)

[Gambar 4.7 Sequence Diagram Aplikasi Pelaksana 44](#_Toc44377084)

[Gambar 4.8 Sequence Diagram Aplikasi Pengawas 45](#_Toc44377085)

[Gambar 4.9 Aplikasi XAMPP 47](#_Toc44377086)

[Gambar 4.10 *Phpmyadmin* Membuat Database Baru 47](#_Toc44377087)

[Gambar 4.11 *Phpmyadmin* Membuat *Table* 48](#_Toc44377088)

[Gambar 4.12 Penentuan *Field* pada *tbl\_admin* 48](#_Toc44377089)

[Gambar 4.13 Penentuan *Field* pada *tbl\_laporan* 48](#_Toc44377090)

[Gambar 4.14 Penentuan *Field* pada *tbl\_proyek* 49](#_Toc44377091)

[Gambar 4.15 Penentuan *Field* pada *tbl\_schedule* 49](#_Toc44377092)

[Gambar 4.16 Penentuan Field pada tbl\_user 49](#_Toc44377093)

[Gambar 4.17 Tampilan *Login Admin* 53](#_Toc44377094)

[Gambar 4.18 Tampilan Dashboard Admin 53](#_Toc44377095)

[Gambar 4.19 Tampilan Form Data Pelaksana 54](#_Toc44377096)

[Gambar 4.20 Tampilan Form Data Pengawas 54](#_Toc44377097)

[Gambar 4.21 Tampilan Form Data Proyek Admin 55](#_Toc44377098)

[Gambar 4.22 Tampilan Form Data Schedule Proyek Admin 55](#_Toc44377099)

[Gambar 4.23 Tampilan Form Data Akun 56](#_Toc44377100)

[Gambar 4.24 Tampilan *Login* Pelaksana 56](#_Toc44377101)

[Gambar 4.25 Tampilan *Dashboard* Pelaksana 57](#_Toc44377102)

[Gambar 4.26 Tampilan *Form* Data *Schedule* Proyek 57](#_Toc44377103)

[Gambar 4.27 Tampilan *Form* Data Akun 58](#_Toc44377104)

[Gambar 4.28 Tampilan *Login* Pengawas 58](#_Toc44377105)

[Gambar 4.29 Tampilan *Dashboard* Pengawas 59](#_Toc44377106)

[Gambar 4.30 Tampilan *Form* Data *Schedule* Proyek Pengawas 59](#_Toc44377107)

[Gambar 4.31 Tampilan *Form* Data Akun Pengawas 60](#_Toc44377108)

[Gambar 4.32 Tampilan *Schedule* Pelaksanaan Proyek 60](#_Toc44377109)

[Gambar 5.1 Pengujian Login 68](#_Toc44377110)

[Gambar 5.2 Pengujian *Input* Data 70](#_Toc44377111)

[Gambar 5.3 Pengujian *Edit* Data 72](#_Toc44377112)

[Gambar 5.4 Pengujian Hapus Data 74](#_Toc44377113)

[Gambar 5.5 Pengujian *Download Schedule* Proyek 76](#_Toc44377114)

# BAB I PENDAHULUAN

1. **LATAR BELAKANG**

Dinas Pekerjaan Umum mempunyai tugas melaksanakan tugas umum pemerintahan dan pembangunan di bidang bina marga, cipta karya, pengairan dan tata ruang sesuai dengan Peraturan Perundang-undangan. Keberhasilan suatu proyek di Dinas PU (Pekerjaan Umum) Kabupaten Sidenreng Rappang, tidak hanya dapat dilihat dari jangka waktu penyelesaian dan hasil akhir dari proyek tersebut, tetapi salah satu faktor penting adalah laporan perkembangan proyek yang dapat dipantau secara harian (*up to date*) oleh pihak Bina Marga. Laporan perkembangan proyek merupakan sesuatu hal yang penting bagi Bina Marga sebagai bahan pertimbangan dalam mengambil keputusan. Apabila proyek tersebut mengalami keterlambatan dan hal itu terpantau maka pihak Bina Marga dapat menentukan tindakan lebih lanjut. Oleh karena itu, informasi perkembangan dalam proyek merupakan sesuatu hal yang sangat penting dan utama untuk diberikan. Bina Marga mempunyai proyek-proyek yang berhubungan dengan peningkatan dan pemeliharaan transportasi, khususnya jalan dan jembatan. Setiap proyek dalam Bina Marga mempunyai data rencana pekerjaan. Data awal tersebut digunakan sebagai tolak ukur perkembangan proyek sehingga informasi perkembangan proyek merupakan hal yang sangat dibutuhkan untuk mengetahui apakah proyek berjalan sesuai dengan rencana.

Sistem yang berjalan di Bina Marga saat ini, pihak Bina Marga mendapatkan informasi perkembangan proyek secara mingguan dari konsultan pengawas lapangan, termasuk informasi permasalahan yang terjadi dalam pengerjaan proyek. Permasalahan yang terjadi ada yang sudah terselesaikan dan ada yang menunggu tindakan turun tangan dari pihak Bina Marga yang diwakili oleh pejabat pembuat komitmen. Pejabat pembuat komitmen (PPKm) merupakan Kepala Divisi Bina Marga yang bertanggung jawab terhadap pelaksanaan proyek yang ada di divisinya. Sistem yang berjalan saat ini mempunyai beberapa kelemahan. Pertama, PPKm tidak bisa memantau perkembangan secara harian karena data yang diberikan bersifat mingguan dan sering terlambat dengan alasan belum selesai dibuat. Kedua, PPkm tidak mengetahui secara cepat permasalahan-permasalahan yang terjadi dalam pelaksanaan proyek sehingga terkadang penyelesaiannya terlambat. Ketiga, pihak konsultan pengawas kesulitan untuk melaporkan perkembangan proyek secara harian karena letaknya yang tersebar di wilayah Kabupaten Sidenreng Rappang dan proyek yang diawasi lebih dari satu sehingga tidak bisa setiap hari memberikan laporan ke kantor Bina Marga. Dengan adanya aplikasi *web* nantinya diharapkan PPKm mempunyai sistem yang dapat digunakan sebagai pertimbangan dalam mengambil keputusan dalam mengendalikan dan melakukan pengawasan terhadap proyek yang ada pada Bina Marga. Di samping itu juga dengan adanya aplikasi *Schedule* pengerjaan proyek ini maka pihak konsultan bisa memasukan data langsung di lapangan dan pihak Bina Marga dapat langsung memonitor perkembangan proyek. Berdasarkan masalah diatas, penulis mengusulkan untuk membangun suatu aplikasi yang menghasilkan *output* yang dapat membantu Dinas PU dalam melakukan pemantau secara *Online* maka penulis mengusulkan judul “**Aplikasi *Schedule* Pengerjaan Proyek *Online* Dinas PU Kabupaten Sidenreng Rappang**”.

1. **RUMUSAN MASALAH**

Berdasarkan latar belakang diatas maka dirumuskan masalah sebagai berikut :

1. Bagaimana Merancang Aplikasi *Schedule* Pengerjaan Proyek *Online* Dinas PU Kabupaten Sidenreng Rappang?
2. Bagaimana Pelaporan *Schedule* Pengerjaan Proyek *Online* Dinas PU Kabupaten Sidenreng Rappang ?
3. **BATASAN MASALAH**

Batasan masalah diperlukan agar penelitian yang dilakukan dapat lebih terfokus, maka Penulis membatasi masalah dengan pokok bahasan yaitu Masalah Pelaporan *Schedule* Pengerjaan Proyek *Online* Dinas PU Kabupaten Sidenreng Rappang dan lebih memfokuskan pada permasalahan, pada perancangan aplikasi yakni dengan penggunaan bahasa program *PHP* dengan *database DBMS MySQL*

1. **TUJUAN PENELITIAN**

Tujuan yang hendak dicapai dalam penelitian ini adalah :

1. Untuk mengetahui bagaimana Merancang Aplikasi *Schedule* Pengerjaan Proyek *Online* Dinas PU Kabupaten Sidenreng Rappang.
2. Untuk mengetahui bagaimana Pelaporan *Schedule* Pengerjaan Proyek *Online* Dinas PU Kabupaten Sidenreng Rappang.
3. **MANFAAT PENELITIAN**

Adapun manfaat penelitian dari perancangan Aplikasi *Schedule* Pengerjaan Proyek *Online* Dinas PU Kabupaten Sidenreng Rappangsebagai berikut :

1. Manfaat yang diperoleh dari penelitian ini dapat membantu Dinas Pekerjaan Umum (PU) Kabupaten Sidenreng Rappang dalam pengawasan pengerjaan proyek di Kabupaten Sidenreng Rappang.
2. Bagi penulis bermanfaat sebagai wadah untuk menerapkan dan mengimplementasikan ilmu yang diperoleh selama perkuliahan agar menjadi bekal didunia kerja.
3. Bagi akademik bermanfaat sebagai materi evaluasi bagi pengembangan peningkatan pendidikan dimasa yang akan datang .

# BAB II TINJAUAN PUSTAKA

1. **PENELITIAN TERDAHULU**

Berbagai penelitian Terdahulu merupakan hal yang sangat perlu dan dapat dijadikan sebagai data pendukung. Salah satu data pendukung yang menurut peneliti perlu dijadikan bagian tersendiri adalah penelitian terdahulu yang relevan dengan permasalahan yang sedang dibahas dalam penelitian ini. Dalam hal ini, fokus penelitian terdahulu yang dijadikan acuan adalah terkait dengan masalah teknologi informasi. Oleh karena itu, peneliti melakukan kajian terhadap beberapa hasil penelitian berupa tugas akhir/skripsi atau jurnal melalui *internet.*

1. Mudjahidin dan Nyoman Dita Pahang Putra (2010) yang berjudul **“ Rancang Bangun Sistem Informasi *Monitoring* Perkembangan Proyek Berbasis *Web* Studi Kasus DiDinas Bina Marga Dan Pemantusan “** Setiap proyek dalam Dinas Bimatus mempunyai data rencana pekerjaan. Data awal tersebut digunakan sebagai tolak ukur perkembangan proyek sehingga informasi perkembangan proyek merupakan hal yang sangat dibutuhkan untuk mengetahui apakah proyek berjalan sesuai dengan rencana. Dengan adanya kebutuhan ini kemudian dikembangkan sebuah sistem berbasis komputer yang digunakan untuk mengelola data realisasi dan data rencana sehingga pihak dinas bisa mengetahui apakah pelaksanaan proyek berjalan sesuai rencana atau tidak. Pengembangan sistem dimulai dengan identifikasi kebutuhan, selanjutnya dibuat desain sistem yang digambarkan dengan *UML (Unified Modeling Language)* dan *ERD (Entity Relationship Diagram)*. Desain sistem diterapkan dalam lingkungan *web* dan *WAP (Wireless Aplication Protocol)* yang berbasis *PHP (Hypertext Preprocessor)* dan menggunakan basis data *MySQL*. Terakhir, dilakukan uji coba terhadap beberapa fungsi untuk mengevaluasi kinerja sistem. Penelitian ini menghasilkan sebuah aplikasi *Web* dan *WAP* yang mampu mengelola data perkembangan proyek di Dinas Bina Marga dan Pematusan.
2. Bagas Wido Permana (2016) yang berjudul **“** **Perancangan Sistem Informasi Manajemen Proyek Berbasis *Website* Di Ph Masa *Creative* Yogyakarta”** Tujuan penelitian ini adalahBagaimana merancang sistem informasi manajemen proyek berbasis *Website* yang efektif dan efisien bagi manajer dan karyawan di PH Masa *Creative* Yogyakarta peneliti mencoba menganalisis beberapa masalah mendasar yang ada dengan metode analisis *PIECES* untuk membandingkan sistem manual dengan sistem yang akan dibangun. Selain itu, peneliti juga melakukan perancangan sistem dengan model proses *DFD,* desain basis data, desain hubungan antar tabel, desain antarmuka. Dari analisis dan desain yang telah dilakukan, dihasilkan sistem informasi berbasis *web* yang dapat diimplementasikan sementara di Rumah Produksi *Creative* Masa
3. **TINJAUAN TEORI**
4. **Aplikasi**

Aplikasi berasal dari kata *application* yaitu bentuk benda dari kata kerja *to apply* yang dalam Bahasa Indonesia berarti pengolah.Secara istilah, aplikasi komputer adalah suatu sub kelas perangkat lunak komputer yang menggunakan komputer langsung untuk melakukan suatu tugas yang diinginkan pemakai.

Menurut Kamus Besar Bahasa Indonesia (KBBI, 2015) Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna. Sementara menurut Hartono (1999) aplikasi adalah penggunaan dalam suatu komputer, insturksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses *input* menjadi *output.*

1. **Konsep Proyek**
2. Definisi Proyek

Proyek merupakan satu kegiatan sementara yang berlangsung dalam jangka waktu terbatas, dengan alokasi sumber daya tertentu dan dimaksudkan untuk menghasilkan produk atau *deliverable* yang kriteria mutunya telah digariskan dengan jelas. Menurut *Project Management Institute*, (*PMI*), dalam buku yang berjudul “ *A Guide to the Project Management Body of Knowledge”* (*PMBOK Guide*) mendefinisikan proyek sebagai berikut: “ Usaha temporer yang dilakukan untuk menciptakan proyek atau jasa (*service*) yang unik” (Schwalbe, 2004). Proyek juga diartikan sebagai sebuah usaha yang tidak rutin, yang dibatasi oleh waktu, anggaran, sumber daya, dan spesifikasi kinerja yang dirancang untuk memenuhi kebutuhan pelanggan. Proyek biasanya dilakukan untuk membangun atau meningkatkan daya guna sesuatu, misalnya proyek pembangunan jalan, pembangunan jembatan, pembangunan sistem informasi rumah sakit, dan lain-lain. Dari pengertian diatas maka ciri pokok proyek adalah sebagai berikut (Iman,1999):

1. Bertujuan menghasilkan lingkup (*scope*) tertentu berupa produk akhir atau hasil kerja akhir.
2. Dalam proses mewujudkan lingkup diatas, ditentukan sejumlah biaya, jadwal, serta kriteria mutu.
3. Bersifat sementara, dalam arti umurnya dibatasi oleh selesainya tugas. Titik awal dan akhir ditentukan dengan jelas.
4. Nonrutin, tidak berulang-ulang. Macam dan intensitas kegiatan berubah sepanjang proyek berlangsung. Secara umum, ada dua jenis pekerjaan jika dilihat berdasarkan waktu penyelesaian dan tempatnya, yaitu pekerjaan yang merupakan operasional dan pekerjaan yang merupakan proyek. Operasional merupakan jenis pekerjaan dimana aktifitas yang sama dilakukan berulang-ulang setiap jangka waktu tertentu ditempat yang sama dengan mendayagunakan sistem yang telah ada, misalnya kegiatan memproduksi semen di pabrik semen atau merakit mobil di bengkel (Iman, 1999).

Sedangkan proyek merupakan jenis pekerjaan dimana pekerjaan yang berbeda-beda dilakukan untuk waktu dan lokasi tertentu saja bermaksud untuk membangun sistem yang belum ada (Iman, 1999). Selain itu hasil proyek biasanya membawa perubahan yang besar, sedangkan operasional biasanya tidak membawa perubahan yang besar (Iman, 1999).

1. Macam - macam Proyek

Dilihat dari komponen kegiatan utamanya macam proyek dapat dikelompokkan sebagai berikut (Iman, 1999):

1. Proyek *Engineering* Konstruksi,komponen utama jenis proyek ini terdiri dari pengkajian kelayakan, desain *engineering*, pengadaan dan konstruksi. Contoh proyek macam ini adalah pembangunan jalan, jembatan gedung sekolah, fasilitas industri, dan lain-lain. Dalam Tugas Akhir ini aplikasi yang dibangun menangani pengelolaan proyek tahap perencanaan dan pengendalian jadwal untuk proyek kelompok *engineering-konstruksi.*
2. Proyek *Engineering* Manufaktur*,* merupakan proyek yang dilaksanakan untuk menghasilkan produk baru. Kegiatan utamanya meliputi desain *engineering*, pengembangan produk, pengadaan, manufaktur, perakitan, uji coba fungsi dan operasi produk yang dihasilkan. Contohnya adalah pembuatan ketel uap, generator listrik, mesin pabrik, dan lain-lain.
3. Proyek Penelitian dan Pengembangan, proyek ini bertujuan untuk melakukan penelitian dan pengembangan dalam rangka menghasilkan suatu produk tertentu.
4. Proyek Pelayanan Manajemen*,* contohnya merancang sistem informasi manajemen meliputi perangkat lunak atau perangkat keras, merancang program efisiensi dan penghematan. Proyek jenis ini tidak menghasilkan produk dalam bentuk fisik, tetapi laporan akhir
5. Proyek Kapital, berbagai badan usaha atau pemerintah memiliki kriteria tertentu untuk proyek kapital. Proyek kapital umumnya meliputi pembebasan tanah, pembelian material dan lain-lain.
6. Proyek Radio Telekomunikasi, tujuannya untuk membangun jaringan telekomunikasi yang dapat menjangkau area yang luas dengan biaya yang relatif tidak terlalu mahal.
7. Proyek Konservasi *Bio-Diversity,* proyek ini berkaitan dengan usaha
8. pelestarian lingkungan.
9. Pelaku Proyek

Suatu proyek melibatkan banyak pihak. Secara umum, berdasarkan tugas dan tanggung jawabnya, pelaku proyek dapat dibedakan atas (Fadla, 2005):

1. Manajer proyek

Orang yang bertanggung jawab atas pelaksanaan proyek dan hasil akhir dari proyek. Manajer proyek bertugas memimpin tim proyek untuk menyelesaikan proyek.

1. Anggota tim proyek

Terdiri dari beberapa orang yang bertugas melakukan tahapan pengeksekusian proyek.

1. Sponsor proyek

Sponsor proyek biasanya adalah seorang manajer dari organisasi atau perusahaan yang bertanggung jawab untuk memberikan dana dan sumber daya lainnya.

1. *Client* proyek

*Client* proyek adalah orang-orang yang nantinya akan menggunakan atau terkena dampak dari hasil akhir proyek.

1. *Group* pendukung

*Group* pendukung merupakan orang-orang yang mendukung lancarnya pelaksanaan proyek secara tidak langsung. Misalnya, bagian akomodasi, keuangan, dan lain sebagainya.

1. Tim proyek lainnya

Jika seandainya proyek yang akan dikerjakan ada kaitannya dengan proyek yang dikerjakan tim lain, maka tim lain tersebut merupakan salah satu dari pelaku proyek meskipun tidak secara langsung.

1. **Konsep Penyusunan Jadwal (*Schedule*)**

Macam-macam dari *schedule* dapat dibagi menjadi 2 yaitu Bagan Balok dan Jaringan Kerja (CPM). Dimana keduanya mempunyai kelebihan dan kekurangan masing-masing. Selain itu terdapat perpaduan antara *Gantt / Bar chart* dengan Jaringan Kerja yang disebut *Time-Based* Diagram. Evaluasi kemajuan proyek tergantung pada akurasi pengukuran dan pembuatan laporan di lapangan. Laporan kemajuan di lapangan adalah dokumen yang sangat penting dalam menganalisa kemajuan pada akhir penyelesaian proyek. Laporan-laporan yang diperlukan meliputi presentase penyelesaian proyek pada tiap-tiap aktivitasnya . Menganalisa atau mengevaluasi tidak hanya dilakukan pada akhir proyek saja, tapi bisa juga dilakukan sewaktu-waktu apabila proyek telah terlihat ketinggalan dari jadwalnya (Smith, 2000). Setelah menerima laporan kemajuan di lapangan, informasi yang didapat kemudian di bandingkan dengan penjadwalan proyek.

Tujuan dasar dari *updating* adalah meng-*schedule* ulang pekerjaan yang sudah dilakukan dengan menggunakan status proyek yang aktual sebagai awal mula penentuan ulang *schedule* proyek. Dalam kenyataan di lapangannya, pelaksanaan manajemen waktu proyek konstruksi banyak menemui kendala-kendala yang menyebabkan pelaksanaannya tidak optimal. Dari penelitian yang telah dilakukan beberapa ahli pada perusahaan kontraktor di Indonesia sebelumnya, disebutkan bahwa kendala-kendala yang sering dihadapi tersebut adalah :

1. Kesulitan untuk mendapatkan *supliyer* dan subkontraktor yang commit dengan *schedule* yang sudah dibuat bersama.
2. Desain yang sebelum selesai dan perubahan desain.
3. Kurangnya koordinasi dan komunikasi dengan pelaksana di lapangan.
4. Ketelambatan pembayaran dari *owner* kepada kontraktor.
5. Kekurangan material dan peralatan.
6. Perubahan cuaca yang tidak bisa diduga.
7. Kurangnya koordinasi atau pengawasan antara pengawas dengan kerja.
8. Ketidakakuratan informasi yang di dapat dari *monitoring*.
9. Kurangnya sumber daya (tenaga ahli) yang mampu menganalitis keadaan proyek.
10. Program komputer yang kurang baik.

Manajemen waktu itu dikatakan telah dilaksanakan dengan baik, bila setiap perusahaan kontraktor tersebut melaksanakan setiap aspek-aspek dari manajemen waktu. Dimana aspek-aspek manajemen waktu yaitu:

1. Menentukan penjadwalan proyek
2. *Monitoring* (Mengukur dan Membuat Laporan Kemajuan Proyek)
3. Membandingkan Jadwal dengan Kemajuan Proyek *(Analysis)*
4. Merencanakan dan Menerapkan Tindakan Pembetulan *(Corective Action)*
5. Memperbaharui Penjadwalan Proyek *(Update Operational Schedule)*
6. **Tinjauan Perangkat Lunak**

Berikut ini adalah tinjauan singkat perangkat lunak yang digunakan penulis untuk membantu penulisan dan pembuatan penelitian ini:

1. *MySQL*

*MySQL* adalah sebuah perangkat lunak sistem manajemen basis data *SQL* (bahasa Inggris: *database management system*) atau *DBMS* yang *multithread, multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. *MySQL* dikenal sebagai *DBMS* (*DataBase Management System*) perangkat lunak ini bermanfaat mengelolah data dengan cara sangat fleksibel dan cepat. *DBMS* adalah perangkat lunak yang digunakan untuk mengelola dan mengontrol pengaksesan *database*. Berikut ini adalah sejumlah aktivitas yang terkait dengan data yang didukung oleh perangkat lunak tersebut.

1. Menyimpan data ke dalam tabel
2. Menghapus data dalam tabel
3. Mengubah data dalam tabel
4. Mengambil data yang tersimpan dalam tabel
5. Memungkinkan untuk memilih data tertentu yang diambil
6. Memungkinkan untuk melakukan pengaturan hak akses terhadap data

*MySQL* banyak dipakai untuk kepentingan penanganan *database* karena selain handal juga bersifat *open source*.

1. *HTML*

Menurut Sibero (2013:19) “*HyperText Markup Language* atau *HTML* adalah bahasa yang digunakan pada dokumen *web* sebagai bahasa untuk pertukaran dokumen *web*”.

Salah satu format yang digunakan dalam pembuatan dokumen dan aplikasi yang berjalan di sebuah halaman web adalah pengertian dari *HTML*. (Arief, 2011)

Dari teori diatas dapat disimpulkan bahwa bahasa pemrograman *HTML* adalah sebuah format yang digunakan dalam pembuatan aplikasi dan pertukaran dokumen web.

Contoh dokumen *HTML* sederhana :

<!DOCTYPE HTML>

<html>

<head style="background-color: black; color: white;">

<title>Selamat Datang HTML</title>

</head>

<body>

<p>Halo dunia!</p>

</body>

</html>

1. *CSS (Cascading Style Sheet)*

Menurut Jayan (2010:2) mengemukakan bahwa “*CSS* merupakan singkatan dari*Cascading Style Sheet*. Kegunaannya adalah untuk mengatur tampilan dokumen *HTML*, contohnya seperti pengaturan jarak antar baris, teks, warna dan format border bahkan penampilan file gambar”.

Untuk saat ini terdapat tiga versi *CSS*, yaitu *CSS1, CSS2*, dan *CSS3. CSS1* dikembangkan berpusat pada pemformatan dokumen *HTML, CSS2* dikembangkan untuk memenuhi kebutuhan terhadap format dokumen agar bisa ditampilkan di *printer*, sedangkan *CSS3* adalah versi terbaru dari *CSS* yang mampu melakukan banyak hal dalam desain *website*.

*CSS3* juga dapat melakukan animasi pada halaman *website*, di antaranya animasi warna hingga animasi 3D. Dengan CSS3 desainer lebih dimudahkan dalam hal kompatibilitas *website*nya pada *smartphone* dengan dukungan fitur yakni media *query*. Selain itu, banyak fitur baru pada *CSS3* seperti : *multiple background, border-radius, drop-shadow, border-image, CSS Math, dan CSS Object Model.*

Ada dua sifat *CSS* yaitu *interna*l dan *eksternal*. Jika internal yang dipilih, maka *script* itu dimasukkan secara langsung ke halaman *website* yang akan didesain. Kalau halaman *web* yang lain didesain dengan model yang sama, maka *script CSS* itu harus dimasukkan lagi ke dalam halaman *web* yang lain itu.

Sifat yang kedua adalah *eksterna*l dimana *script CSS* dipisahkan dan diletakkan dalam berkas khusus. Nanti, cukup gunakan semacam tautan menuju berkas *CSS* itu jika halaman *web* yang didesain akan dibuat seperti model yang di *script* tersebut.

1. *Javascript*

*Javascript* merupakan bahasa *scripting* yang pertama kali dikembangkan *Netscape* pada tahun 1995. Penulisan *Javascript* berada di dalam dokumen *HTML* dan pemanggilan program tersebut tergantung pada *browser (navigator)* yang digunakan dalam memanggil halaman yang terdapat pada *script* tersebut. *Javascript* juga memerlukan kompilator atau penerjemah untuk menjalankannya. (Utomo, 2007).

Hal tersebut juga dijelaskan oleh Mac Bride (2007) dalam bukunya, *Javascript.* Menurut Bride (2007), *Javascript* adalah bahasa pemrograman berbasis *browser*. Kode-kodenya ditulis langsung ke dalam *HTML* dari halaman-halaman *web* dan diterjemahkan serta dieksekusi sebagai respon terhadapat aktivitas-aktivitas pada halaman *web.*

Karakteristik *Javascript*

Menurut Sidik (2007), terdapat beberapa hal khusus yang terdapat dalam *Javascipt* sebagai berikut :

1. Menggunakan *blok* awal “{“ dan blok akhir “}”.
2. *Automatic conversion* dalam pengoperasian tipe data yang berbeda.
3. *Case sensitive*, sehingga programmer harus berhati-hati dalam menggunakan variabel, fungsi, dan lain-lain.
4. *File extension* yang umumnya digunakan adalah “\*.js”.
5. Setiap *statement* dapat diakhiri dengan “;” tetapi dapat juga tidak.
6. Jika tidak didukung oleh *browser* tipe lama, scriptnya dapat disembunyikan di antara *tag* “<!-“ dan “->”.
7. Jika program dalam satu baris terlalu panjang, dapat disambung ke baris berikutnya dengan karakter “\” diawal baris tersebut.

Berlandaskan pada definisi *Javascript* yang disampaikan, Bride (2007) menjabarkan *Javascript* secara lebih lanjut sebagai berikut:

1. *Javascript* berinteraksi dengan *browser*. *Javascript* dapat membaca informasi dari halaman *web*, seperti data yang dimasukkan melalui *field-field* dalam sebuah *form*.
2. *Javascript* adalah bahasa *event-driven* berbasis objek. *Javascript* memberikan respon terhadap aktivitas yang dilakukan terhadap objek-objek pada halaman *web*.

*Javascript* bukanlah bahasa yang rumit dan memiliki batasan dalam penggunaannya. Sebagai contoh, dengan *Javascript* dapat digunakan tidak hanya untuk sekedar menampilkan kotak pesan atau notifikasi saat sebuah tombol diklik. *Javascript* juga digunakan untuk aplikasi *game interaktif* yang kompleks atau memeriksa pesanan *online* sebuah situs belanja. Batasannya hanyalah bahwa setiap *input* atau *ouput* dari aplikasi dengan *Javascript* hanya dapat dilakukan/disajikan melalui sebuah *web browser*.

Kode *JavaScript* biasanya dituliskan dalam bentuk [fungsi](https://id.wikipedia.org/wiki/Fungsi) yang ditaruh di *tag <head>* yang dibuka dengan[*tag*](https://id.wikipedia.org/wiki/Tag)*<script type="text/javascript">.*

<script type="text/javascript">

alert("Halo Dunia!");

</script>

1. *PHP*

*PHP* adalah sebuah bahasa pemrograman yang berbentuk *scripting*, sistem kerja dari program ini adalah sebagai *interpreter* bukan sebagai *compiler* (Nugroho, 2004).

*PHP* merupakan singkatan dari *Hypertext Preprocessor* adalah sebuah bahasa *scripting* yang terpasang pada *HTML*. Sebagian besar sintaks mirip dengan bahasa *C*, *Java*, *asp* dan *Perl*, ditambah beberapa fungsi *PHP* yang spesifik. Tujuan utama bahasa ini adalah untuk memungkinkan perancang web untuk menulis halaman web dinamik dengan cepat. (Madcoms madium, 2011)

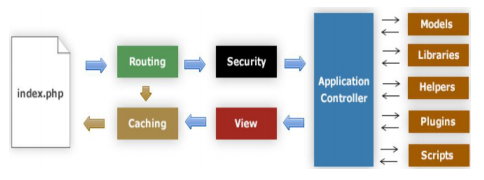
Kelebihan *PHP* dari bahasa pemrograman lain:

* + - * 1. Bahasa pemrograman *PHP* adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
        2. *Web Server* yang mendukung *PHP* dapat ditemukan dimana-mana dari *IIS* *(Internet Information Services)* dengan *Apache*, dengan *configurasi* yang relatif lebih mudah.
        3. Dalam sisi pengembangan lebih mudah, karena banyak *developer* yang siap membantu dalam pengembangan.
        4. Dalam sisi pemahamanan, *PHP* adalah bahasa scripting yang paling mudah karena referensi banyak.
        5. *PHP* adalah bahasa *open source* yang dapat digunakan di berbagai mesin (*linux, unix, windows*) dan dapat digunakan di berbagai mesin *console* serta juga dapat menjalankan perintah-perintah sistem.

1. *Framework CodeIgniter*

Menurut Hakim (2010:8) “*CodeIgniter* adalah sebuah *framework* *PHP* yang dapat membantu mempercepat *developer* dalam pengembangan aplikasi *web* berbasis *PHP* dibanding jika menulis semua kode program dari awal”. Didalam *CI* ini terdapat beberapa macam kelas yang berbentuk *library* dan *helper* yang berfungsi untuk membantu pemrograman dalam mengembangkan aplikasi. *CI* juga mempunyai file dokumentasi yang sangat memadai untuk menjelaskan setiap fungsi yang ada pada *library* dan *helper*. *File* dokumentasi ini disertakan secara langsung pada saat Pengunduhan paket *framework CI.*

*CodeIgniter* menggunakan konsep *MVC (Model View Controller)*. Menurut Akhmad Sofwan (2003), konsep *MVC* adalah konsep pemisahan antara logika dengan tampilan dan *database*. Manfaat dari konsep ini adalah membuat pengodean logika lebih *simple*, karena sudah dipisahkan dengan kode untuk tampilan dan membuat *programmer* dapat bekerja secara terpisah dengan desainer. *Programmer* mengerjakan logika, sedangkan desainer berkutat dengan desain dan tampilan. Model (M) berisi kode penghubung *database, View (V)* berisi kode desain tampilan, dan Controller (C) berisi kode logika.



Gambar 2.1 CodeIgniter

1. *XAMPP*

*XAMPP*  merupakan  salah  satu  paket installasi  *Apache*,  *PHP*  dan *MySQL*  instant yang dapat kita gunakan untuk membantu prosesi nstallasi ketiga produk tersebut. *XAMPP* adalah  perangkat  lunak bebas, yang  mendukung  banyak  sistem operasi, merupakan  kompilasi dari  beberapa program. Fungsinya adalah  sebagai  *server* yang berdiri  sendiri  (*localhost*),  yang  terdiri  atas program  *Apache* *HTTP*  *Server*, *MySQL* database, dan  penerjemah  bahasa  yang  ditulis  dengan  bahasa pemrograman *PHP*  dan *Perl*. Nama  *XAMPP* merupakan  singkatan dari  X (empat sistem  operasi apapun),  *Apache*, *MySQL*,  PHP  dan  Perl.  Program ini  tersedia  dalam  *GNU* *General Public License dan*  bebas, merupakan web *server* yang mudah digunakan  yang  dapat  melayani  tampilan  halaman *web* yang dinamis.

Adapun fungsi dari komponen *XAMPP* yaitu sebagai berikut :

1. *Apache* sebagai *web* *server* untuk menjalankan aplikasi *PHP*.
2. *MySQL* untuk membuat dan mengubah database.
3. *FileZilla* sebagai alat untuk mengunggah suatu *file* dari komputer lokal kita ke sebuah *hosting* atau *server* yang nantinya *file* tersebut bisa diakseskan oleh orang lain di seluruh dunia.
4. *Tomcat* sebagai *web server* untuk menjalankan *JSP (Java Server Page).*
5. *APACHE*

*Apache* merupakan web *server* yang digunakan untuk menampilkan *Website* di *internet* seperti menggunakan *Mozilla firefox, Google Crome, IE, Safari*, dll berdasarkan kode-kode yang di tulis di dalam *Website* tersebut baik menggunakan bahasa pemrograman *HTML* maupun *PHP* yang mengambil suatu *database* yang dibangun di *MySQL*, sehingga terbentuklah sebuah *Website* yang dapat di lihat di *Mozilla firefox* dan kawan-kawannya. *Apache* sendiri bersifat *open source* sehingga dapat digunakan oleh siapa saja dan dikembangkan oleh siapa saja tentunya bagi yang mampu mengembagkannya.

1. *Unified Modeling Language (UML)*

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

Untuk upaya tersebut *UML* menyediakan 9 jenis diagram yang dapat dikelompokkan berdasarkan sifatnya ­*statis* atau *dinamis*. Ke 9 jenis diagram dalam *UML* itu adalah :

1. Diagram kelas 🡪Bersifat statis Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas aktif.
2. Diagram Objek 🡪Besifat statis. Diagram ini memperlihatkan objek-objek serta relasi-relasi antarobjek. Diagram objek memperlihatkan instalasi statis dari segala sesuatu yang dijumpai pada diagram kelas.
3. *Use-Case* Diagram 🡪Bersifat statis. Diagram ini memperlihatkan himpunan *use case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan pengguna.
4. *Sequence* Diagram 🡪Bersifat dinamis. Diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.
5. *Collaboration* Diagram 🡪Bersifat dinamis. Diagram kolaborasi adalah diagram interaksi yang menekankan organisasi struktural dari objek-objek yang menerima serta mengirim pesan.
6. *Statechart* Diagram 🡪 Bersifat dinamis. Diagram *state* ini memperlihatkan *state-state* pada sistem memuat *state*, transisi, *event*, serta aktifitas. Diagram ini terutama penting untuk memperlihatkan sifat dinamis dari antarmuka, kelas, kolaborasi dan terutama penting pada pemodelan sistem-sistem yang reaktif.
7. *Activity* Diagram 🡪 Bersifat dinamis. Diagram aktivitas ini adalah tipe khusus dari diagram state yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi dalam suatu sistem dan memberi tekanan pada aliran kendali antar objek.
8. *Component* Diagram 🡪 Bersifat statis. Dengan kemponen ini memperlihatkan organisasi serta kebergantungan sistem atau perangkat lunak pada komponen-komponen yang telah ada sebelumnya. Diagram ini berhubungan dengan diagram kelas dimana komponen secara tipikal dipetakan kedalam satu atau lebih kelas-kelas, antarmuka-antarmuka serta kolaborasi-kolaborasi.
9. *Deployment* Diagram 🡪 Bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan. Diagram ini membuat simpul-simpul beserta komponen-komponen yang ada di dalamnya. Diagram ini sangat berguna pada banyak mesin.(Sholiq*,* 2006).
10. Daftar simbol *UML*

Tabel 2.1 Table Use Case Diagram

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

Tabel 2.2 Symbol Aktivity Diagram

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

Tabel 2.3 Symbol Sequence Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1 | Object | *Objekct*  *(Partisipan)* | Object atau biasa juga disebut partisipan merupakan instance dari sebuah class dan dituliskan tersusun secara horizontal. Digambarkan sebagai sebuah class (kotak) dengan nama objek di dalamnya yang diawali dengan sebuah titik koma. |
| 2 |  | *Actor* | Actor juga dapat berkomunikasi dengan object, maka actor juga dapat diurutkan sebagai kolom. |
| 3 |  | *Life line* | *Life line* mengidentifikasikan keberadaan sebuah object dalam basis waktu. Notasi untuk *life line* adalah garis putus-putus vertical yang di tarik oleh sebuah *object* |
| 4 |  | *Collaborations* | Collaborations dinotasikan sebagai kotak segi empat yang digambar pada sebuah life line action mengidentifikasikan sebuah object yang akan melakukan sebuah aksi |
| 5 |  | *Boundary* | Boundary terletak diantara system dengan dunia di sekelilingnya. Semua form, laporan-laporan, antara muka ke perangkat keras seperti printer atau scenner dan antara muka ke system lainnya adalah termasuk dalam kategori. |
| 6 |  | *Control* | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaharui elemen yang bergantun padanya elemen yang tidak mandiri |
| 7 |  | *Entity* | Entity digunakan menengani informasi yang mungkin akan disimpan secara permanen. Entity biasa juga merupakan sebuah table |

1. **Metode Pengujian Perangkat Lunak**

Beberapa *test-case* harus dilaksanakan dengan beberapa perbedaan strategi, *query*, atau jalur navigasi yang mewakili penggunaan sistem yang *tipical,* kritis atau abnormal. Isu kunci pada pengembangan sistem adalah pemilihan sekelompok *test-case* yang cocok, sekecil dan secepat mungkin, untuk meyakinkan perilaku sistem secara detail. Pengujian harus mencakup *unit testing*, yang mengecek validasi dari prosedur dan fungsi-fungsi secara independen dari komponen sistem yang lain. Kemudian modul *testing* harus menyusul dilakukan untuk mengetahui apakah penggabungan beberapa unit dalam satu modul sudah berjalan dengan baik, termasuk eksekusi dari beberapa modul yang saling berelasi, apakah sudah berjalan sesuai karakteristik sistem yang diinginkan.

Jika struktur kendali antar modul sudah terbukti bagus, maka pengujian yang tak kalah pentingnya adalah pengujian unit. Pengujian unit digunakan untuk menguji setiap modul untuk menjamin setiap modul menjalankan fungsinya dengan baik. Ada 2 metode untuk melakukan unit *testing*, yaitu :

1. *White Box Testing*

Uji coba *white box testing* merupakan metode perancangan *testcase* yang mengunakan struktural untuk mendapatkan *testcase*, *test* ini digunakan untuk meramal cara kerja perangkat lunak secara rinci kepada *logic path* ( jalur logika ), perangkat lunak di tes dengan kondisi dan perulangan secara fisik.

Contoh pengujian *white box testing* ini merupakan peringatan ketika *user* memasukkan *password user* yang salah, untuk kesalahan semacam ini akan memberikan suatu informasi kepada *user* mengenai kesalahan yang di lakukan.

1. *Black Box*

Berdasarkan hasil uji coba yang dilakukan, seluruh navigasi dan tombol fasilitas program lainnya serta proses yang di jalankan tidak terjadi kesalahan, tetapi aplikasi mempunyai aturan-aturan yang sudah di tetapkan dan harus di ikuti karena apabila di hiraukan maka sistem akan menolak perintah yang tidak sesuai seperti kesalahan ketika *user* belum memasukkan data yang harusnya di *input* sesuai ketentuan sistem yang di jalankan dan sistem memberikan informasi kepada *user* karena data yang ingin diproses belum lengkap atau tidak memenuhi ketentuan untuk proses selanjutnya.

1. **Kerangka Fikir**

Gambar 2.2 Kerangka Pikir

*Schedule* Pengerjaan Proyek adalah laporan waktu pengerjaan suatu proyek yang direncanakan dalam waktu tertentu sesuai jadwal yang telah ditentukan

Di usulkan sebuah Aplikasi *Schedule* Pengerjaan Proyek *Online* Dinas PU Kabupaten Sidenreng Rappang

Memasukkan data pekerja proyek dan laporan *Schedule* Pengerjaan Proyek Dinas PU Kabupaten Sidenreng Rappangyang dilakukan dengan komputersiasi dengan penginputan *database* melakukan *MySQL*

Pelaporan *Schedule* Pengerjaan Proyek yang dilakukan secara *online*  sehingga data yang disajikan dapat diakses Dinas PU Kabupaten Sidenreng Rappang sebagai *monitoring* jadwal waktu pengerjaan proyek yang dilakukan.

# BAB III METODE PENELITIAN

1. **Tempat dan Waktu Penelitian**

Kegiatan penelitian ini dilakukan di Kantor Dinas PU Kabupaten Sidenreng Rappang dengan waktu penelitian yang digunakan selama 2 bulan

1. **Jenis penelitian**

Dalam melakukan penelitian ini menggunakan penelitian deskriptif kualitatif yang bertujuan untuk memahami realitas sosial, yaitu melihat dunia dari apa adanya, bukan dunia yang seharusnya atau dengan kata lain memahami suatu masalah secara mendalam. Jenis penelitian deskriptif kualitatif yang digunakan adalah *Design and Creation* yang merupakan jenis penelitian untuk mengembangkan produk di bidang teknologi informasi dan komunikasi. Proses *Design and Creation* menggunakan prinsip *learning by making.* Metode pengembangan sistem dapat menggunakan model *waterfall* atau *prototyping.* (Ismail, 2015). Dipilihnya jenis penelitian ini dikarenakan konsep dari *Design and Creation* sesuai untuk mengelola penelitian ini dan juga mengembangkan produk berdasarkan penelitian yang dilakukan.

1. **Metode pengumpulan data**

Metode pengumpulan data yang dipakai pada penelitian untuk aplikasi ini adalah metode observasi, wawancara dan studi pustaka.

* 1. Observasi

Observasi yang dilakukan yaitu mengamati secara langsung proses pengolahan Laporan *Schedule* Pengerjaan Proyek *Online* Dinas PU Kabupaten Sidenreng Rappang untuk memperoleh gambaran yang jelas mengenai permasalahan yang diteliti.

* 1. Wawancara

Melakukan tanya jawab langsung antara pengumpul data terhadap narasumber yaitu Kepala Dinas PU Kabupaten Sidenreng Rappang maupun pegawai, pihak-pihak terkait untuk memperoleh informasi agar data yang diperoleh lebih akurat.

* 1. Studi Pustaka

Melakukan pengumpulan data dengan mempelajari referensi-referensi buku, artikel, dan internet yang berhubungan dengan *Schedule* pengerjaan proyek dan laporan Dinas PU Kabupaten Sidenreng Rappang.

1. **Alat dan bahan penelitian**
2. Perangkat keras

Perangkat keras yang digunakan untuk mengembangkan dan menguji coba adalah sebagai berikut :

1. Laptop ASUS
2. *Processor* : *Intel(R) Core* (*TM*) i3-4030U *CPU* @ 1.90*GHz*
3. *Memory* : 2048*MB* *RAM*
4. *Harddisk*  : 500 *GB*
5. *Monitor*  : 14 *inc*
6. Sistem Operasi : *Windows 7 Ultimate 64- bit*
7. Perangkat lunak

Dalam perancangan sistem ini diperlukan *software* berikut :

1. Bahasa pemrograman PHP, *Codeigneter*
2. Sistem Operasi *Windows* 7 (64 bit).
3. *Visual Studio Code*
4. *Notepad++*
5. *XAMPP*
6. *MySQL*
7. **Tahap penelitian**

Tahapan penelitian yang dilakukan adalah pengembangan sistem dan disertai dengan studi literatur. Tahapan penelitian secara terinci adalah sebagai berikut :

1. Penyiapan Penelitian

Tahap persiapan adalah tahap yang dilakukan sebelum melakukan penelitian. Pada tahapan ini dimulai dengan mengkaji permasalahan yang telah ada kemudian melakukan studi literatur tentang penelitian sejenis yang pernah dilakukan.

1. Tahap Pelaksanaan

Tahap pelaksanaan merupakan tahap penerapan penelitian. Pada tahap ini dilakukan beberapa kegiatan yang mengenai dengan penelitian yang akan dilakukan. Pada tahapan ini ada empat langkah yang harus dipenuhi guna untuk mencapai hasil maksimal dalam penelitian yaitu tahapan pengumpulan data dengan menggunakan beberapa metode pengumpulan data yang telah dijelaskan pada poin diatas, pengolahan data, analisis data dan selanjutnya penafsiran hasil analisis. Setelah kegiatan tersebut, proses selanjutnya yaitu dengan melakukan tugas lapangan dalam rangka mengumpulkan data, untuk kemudian akan diproses. Proses yang dimaksud meliputi penyuntingan, penerapan masalah dalam aplikasi program, serta analisis sebagai penarikan kesimpulan hasil akhir.

1. Tahap Penyelesaian

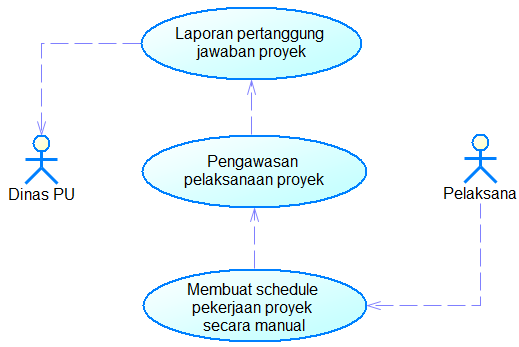
Tahap penyelesaian merupakan tahap akhir yang dilakukan pada penelitian yang dilakukan. Pada tahap ini penyusunan laporan penelitian disusun*.*

1. **Metode Pengujian**

Metode pengujian yang dilakukan pada peneltian ini menggunakan *blackbox* yang dipakai untuk mengetahui kinerja aplikasi pada saat melakukan proses pedaftaran hingga pemesanan dan mengetahui berfungsinya semua aplikasi yang telah dibuat, *blackbox* dalam metode pengujian perangkat lunak yang tes fungsionalitas dari aplikasi yang bertentangan dengan struktur *internal* atau kerja

# BAB IV ANALISIS DAN PERANCANGAN SISTEM

1. **Analisis Sistem Yang Sedang Berjalan**

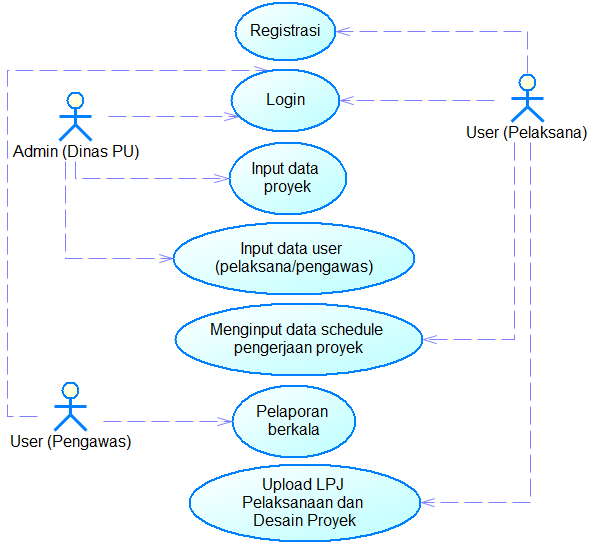


Gambar 4.1 *Use Case* Diagram Sistem yang Berjalan

Pelaksana membuat *Schedule* pengerjaan proyek secara manual dengan pengawasan oleh Dinas PU Kabupaten Sidenreng Rappang untuk dilaporkan sebagai laporan pertanggung jawaban proyek untuk di *monitoring* pelaksanaannya.

1. **Rancangan Sistem Yang Diusulkan**
   * 1. *Use Case* Diagram

Sistem yang diusulkan menggunakan *Use Case* Diagram. *Use case* diagram menggambarkan interaksi antara actor dengan proses atau sistem yang dibuat. *Use case* dan *actor* menggambarkan ruang lingkup sistem yang sedang dibangun. Use case meliputi semua hal yang ada pada sistem, sedangkan actor meliputi semua hal yang ada diluar sistem. *Actor* termasuk seseorang atau apa saja yang berhubungan dengan sistem yang dibangun.



Gambar 4.2 *Use Case* Diagram Sistem yang diusulkan

Penjelasan *Use Case* Diagram :

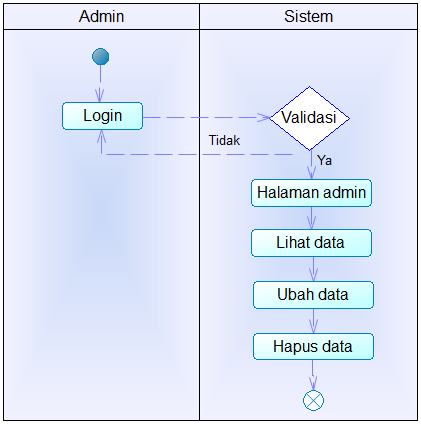
Tabel 4.1 Penjelasan *Use Case* Diagram

|  |  |
| --- | --- |
| **Nama *Use Case*** | **Deskripsi *Use Case*** |
| Admin | Aktor |
| Pelaksana (User) | Aktor |
| Pengawas (User) | Aktor |
| Login | Use case yang menjelaskan tentang proses login sebagai admin/user untuk masuk ke halaman beranda |
| *Input* data proyek | Use Case ini menjelaskan tentang proses memasukkan data proyek oleh admin |
| *Input* data user | User Case ini menjelaskan tentang proses memasukkan data user (Pelaksana/Pengawas) oleh admin |
| Menginput data *Schedule* pengerjaan proyek | Use case ini menjelaskan tentang proses memasukkan data Schedule pengerjaan proyek oleh pelaksana |
| Pelaporan berkala | Use case ini menampilkan data Laporan berkala pengerjaan proyek oleh pelaksana/user dan diawasi oleh pengawas proyek |
| Upload LPJ pelaksanaan dan Desain Proyek | Use case ini menjelaskan tentang LPJ pelaporan dan Desain 3d proyek yang dilakukan oleh pelaksana sebagai pertanggung jawaban proyek yang telah dilakukan. |

* + 1. *Activity Diagram*

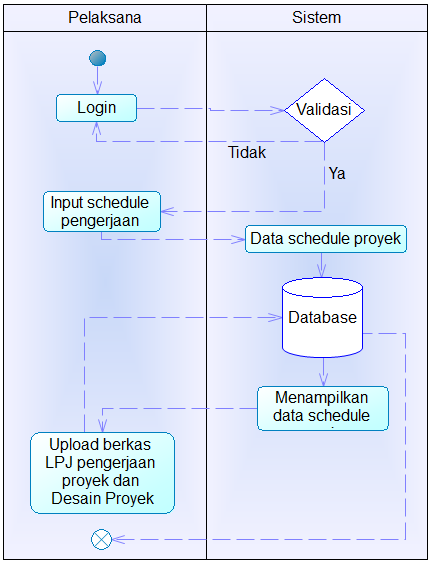
*Activity* diagram menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam satu operasi sehingga dapat juga untuk aktivitas lainnya.

* 1. *Activity Diagram* Admin



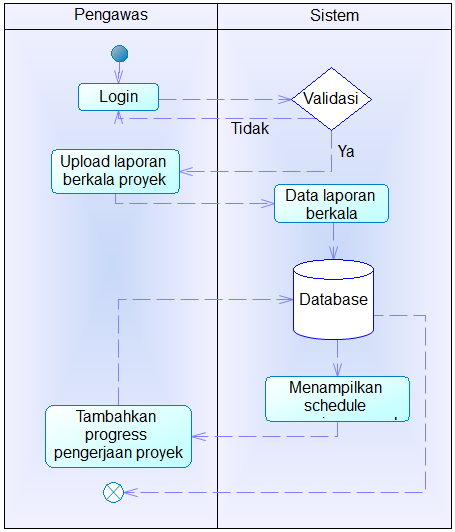
Gambar 4.3 *Activity Diagram Admin*

* 1. *Activity Diagram Pelaksana*



Gambar 4.4 *Activity* Diagram Pelaksana

* 1. *Activity Diagram Pengawas*

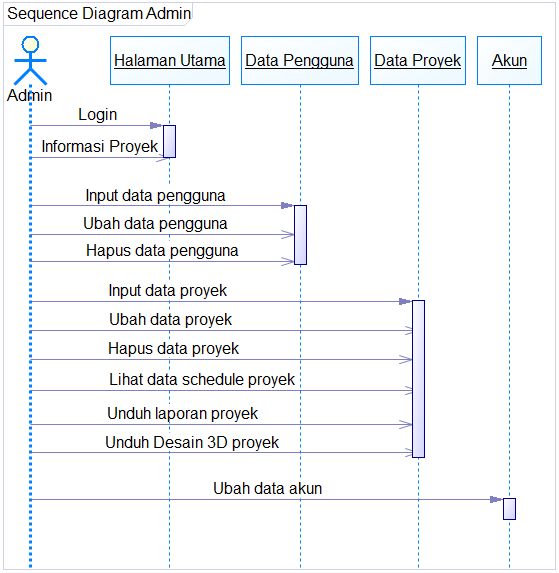


Gambar 4.5 *Activity* Diagram Pengawas

* + 1. *Sequence Diagram*

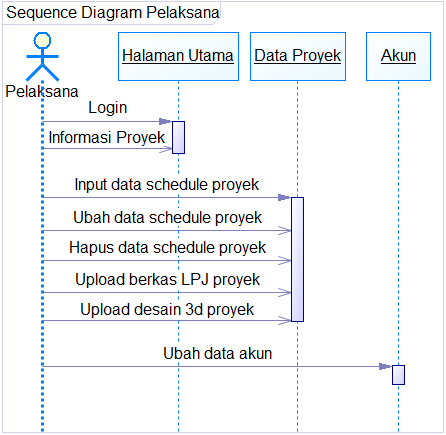
Diagram *sequence* merupakan salah satu diagram *Interaction* yang menjelaskan bagaimana suatu operasi itu dilakukan, *message* (pesan) apa yang dikirim dan kapan pelaksanaannya.

1. *Sequence Diagram* Aplikasi *Admin*



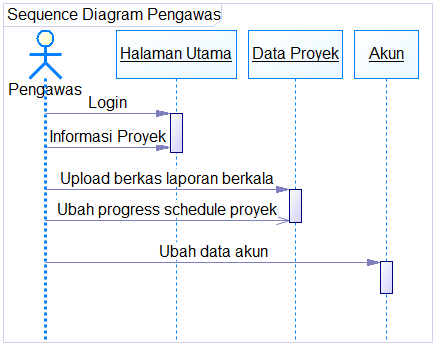
Gambar 4.6 *Sequence Diagram* Aplikasi *Admin*

1. *Sequence Diagram Aplikasi Pelaksana*



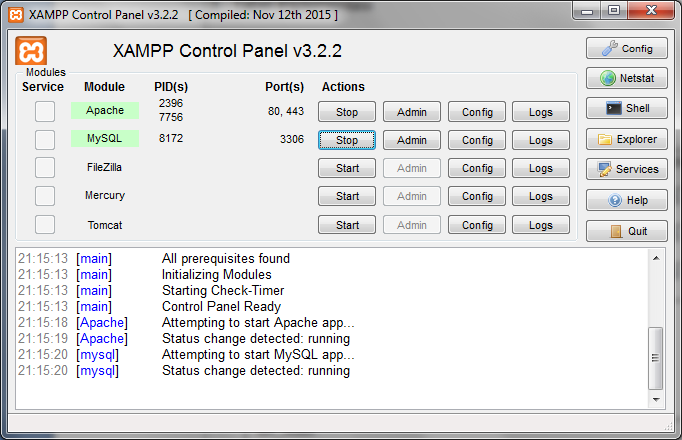
Gambar 4.7 Sequence Diagram Aplikasi Pelaksana

1. *Sequence Diagram Aplikasi Pengawas*



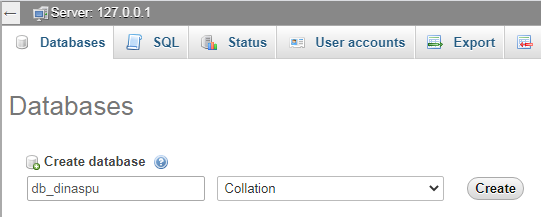
Gambar 4.8 Sequence Diagram Aplikasi Pengawas

1. **Kebutuhan Pembuatan Program**
2. Perangkat lunak yang dibutuhkan
3. *Xampp* sebagai server lokal untuk mengampu berbagai jenis data website yang sedang dalam proses pengembangan. Dalam prakteknya, *XAMPP* bisa digunakan untuk menguji kinerja fitur ataupun menampilkan konten yang ada didalam website kepada orang lain tanpa harus terkoneksi dengan internet, atau istilahnya website *offline*.
4. *HTML (Hypertext Markup Language),* untuk mengelola serangkaian data dan informasi sehingga suatu dokumen dapat diakses dan ditampilkan di Internet melalui layanan *web.*
5. *Javascript,* berfungsi membuat sebuah halaman *website* lebih interaktif dan dinamis. Penggunaan kode *javascript* sangat dibutuhkan untuk *website-website* yang berorientasi pada kenyamanan pengakses *(user-experience).*
6. *CSS,* berfungsi untuk mendesain, membentuk, serta mengubah tampilan halaman sebuah *website.*
7. *CodeIgniter (CI),* Mempercepat dan mempermudah kita dalam pembuatan *website*. Menghasilkan struktur pemrograman yang sangat rapi, baik dari segi kode maupun struktur file phpnya. Memberikan standar *coding* sehingga memudahkan kita atau orang lain untuk mempelajari kembali *system* aplikasi yang dibangun.
8. Bootstrap, fungsinya agar tampilan aplikasi *web* menjadi responsif.
9. *Web* Browser, Untuk pilihannya bisa menggunakan *Google Chrome* atau *Mozilla firefox*.
10. Teks *Editor, notepad* bawaan *windows* atau teks *editor* dasar lainnya seperti *notepad++,Sublime Text, Visual Studio Code*.
11. Pertama install dan jalan aplikasi XAMPP, kemudian mengklik tombol Start pada Module Apache dan MySQL.



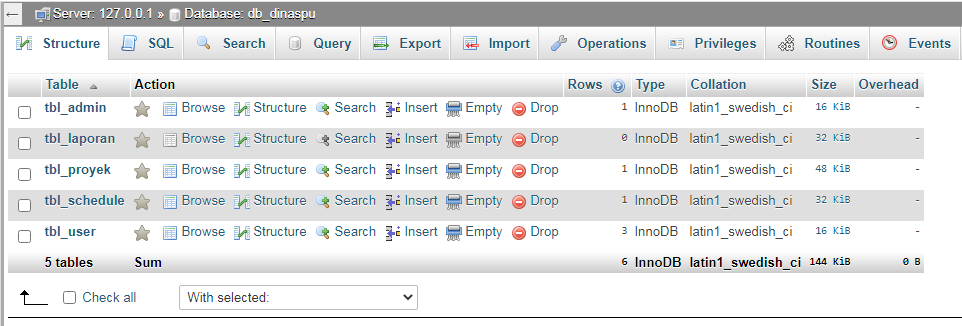
Gambar 4.9 Aplikasi XAMPP

1. Membuka *Web* *Browser*, lalu masuk ke halaman *phpmyadmin*. Kemudian buat sebuah *database* baru yang bernama *db\_dinaspu.*



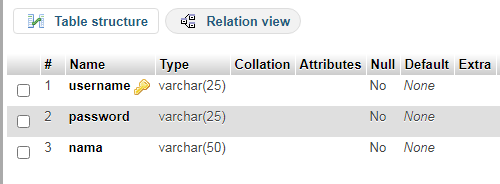
Gambar 4.10 *Phpmyadmin* Membuat Database Baru

1. Kemudian buatlah tabel yang terdiri dari tabel berikut.

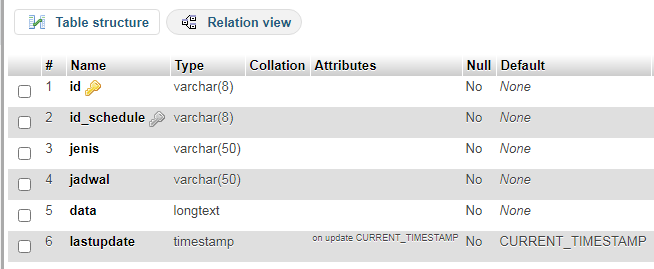


Gambar 4.11 *Phpmyadmin* Membuat *Table*

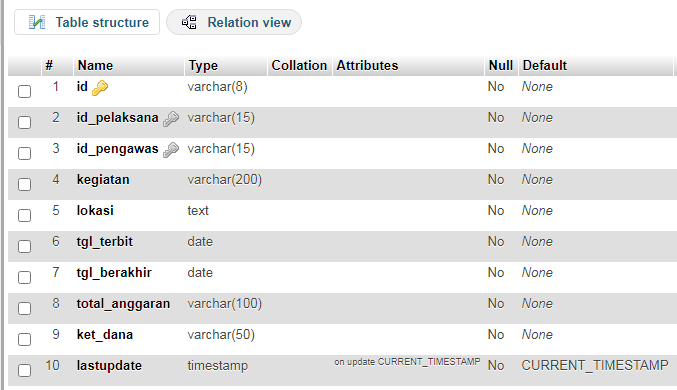
1. Setelah membuat *database* dan tabel yang dibutuhkan, selanjutnya menentukan *field* atau kolom dan *type* *field* dari masing-masing tabel yang telah dibuat.



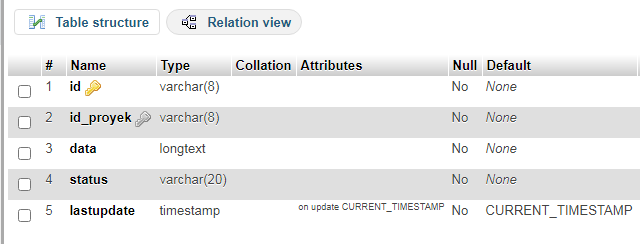
Gambar 4.12 Penentuan *Field* pada *tbl\_admin*



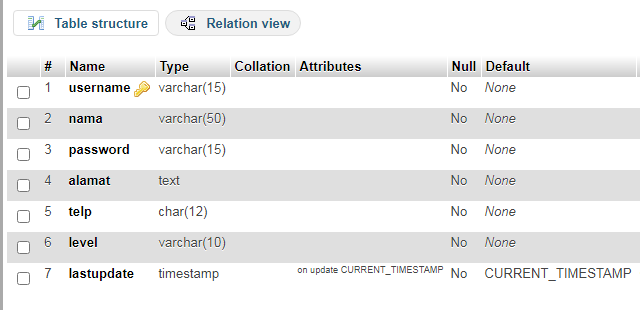
Gambar 4.13 Penentuan *Field* pada *tbl\_laporan*



Gambar 4.14 Penentuan *Field* pada *tbl\_proyek*



Gambar 4.15 Penentuan *Field* pada *tbl\_schedule*



Gambar 4.16 Penentuan Field pada tbl\_user

1. Aplikasi ini menggunakan *database* dengan nama *db\_dinaspu* dan 5 tabel yang dibuat menggunakan *PHPmyadmin*.

Berikut uraian tabel yang terdapat dalam *database :*

1. Tabel *Admin*

Tabel 4.2 Tabel *Admin*

|  |  |  |  |
| --- | --- | --- | --- |
| tbl\_admin | | | |
| No | *Name* | *Type* | *Size* |
| 1 | *Username* | *Varchar* | 25 |
| 2 | *Password* | *Varchar* | 25 |
| 3 | Nama | *Varchar* | 50 |

1. Tabel *User*

Tabel 4.3 Tabel *User*

|  |  |  |  |
| --- | --- | --- | --- |
| tbl\_user | | | |
| No | *Name* | *Type* | *Size* |
| 1 | *Username* | *Varchar* | 25 |
| 2 | Nama | *Varchar* | 50 |
| 3 | *Password* | *Varchar* | 15 |
| 4 | Alamat | *Text* | - |
| 5 | *Telp* | *Char* | 12 |
| 6 | *Level* | *Varchar* | 10 |
| 7 | *Lastupdate* | *Timestamp* | - |

1. Tabel Proyek

Tabel 4.4 Tabel Proyek

|  |  |  |  |
| --- | --- | --- | --- |
| tbl\_proyek | | | |
| No | *Name* | *Type* | *Size* |
| 1 | *Id* | *Varchar* | 8 |
| 2 | Id\_pelaksana | *Varchar* | 15 |
| 3 | Id\_pengawas | *Varchar* | 15 |
| 4 | Kegiatan | *Varchar* | 200 |
| 5 | Lokasi | *Text* | - |
| 6 | Tgl\_terbit | *Date* | - |
| 7 | Tgl\_berakhir | *Date* | - |
| 8 | Total\_anggaran | *Varchar* | 100 |
| 9 | Ket\_dana | *Varchar* | 50 |
| 10 | Lastupdate | *Timestamp* | - |

1. Tabel *Schedule*

Tabel 4.5 Tabel *Schedule*

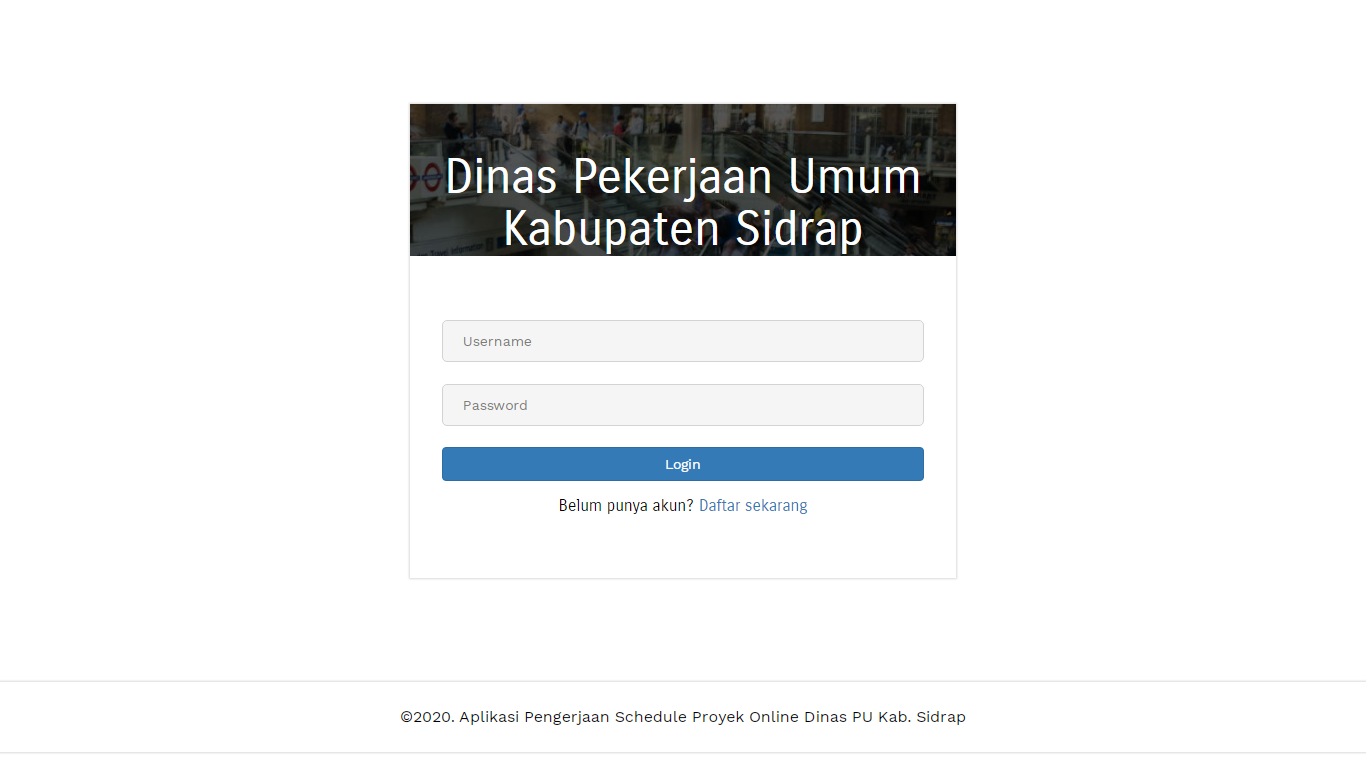
|  |  |  |  |
| --- | --- | --- | --- |
| tbl\_schedule | | | |
| No | *Name* | *Type* | *Size* |
| 1 | *Id* | *Varchar* | 8 |
| 2 | *Id\_proyek* | *Varchar* | 8 |
| 3 | Data | *Longtext* | - |
| 4 | Status | *Varchar* | 20 |
| 5 | *Lastupdate* | *Timestamp* | - |

1. Tabel Laporan

|  |  |  |  |
| --- | --- | --- | --- |
| tbl\_laporan | | | |
| No | *Name* | *Type* | *Size* |
| 1 | *Id* | *Varchar* | 8 |
| 2 | *Id\_schedule* | *Varchar* | 8 |
| 3 | Jenis | *Varchar* | 50 |
| 4 | Jadwal | *Varchar* | 50 |
| 5 | *Data* | *Longtext* | - |
| 6 | Lastupdate | *Timestamp* | - |

1. Setelah *database* selesai di buat, maka langkah selanjutnya adalah membuat *folder* atau *project* baru di dalam folder *XAMPP/htdocs.*
2. Kemudian tentukan *template* yang akan digunakan, dengan cara mendownload *library Bootstrap* dan juga *framework CodeIgniter*.
3. **Rancangan *Input* Program**
   * + 1. Admin
4. *Login*

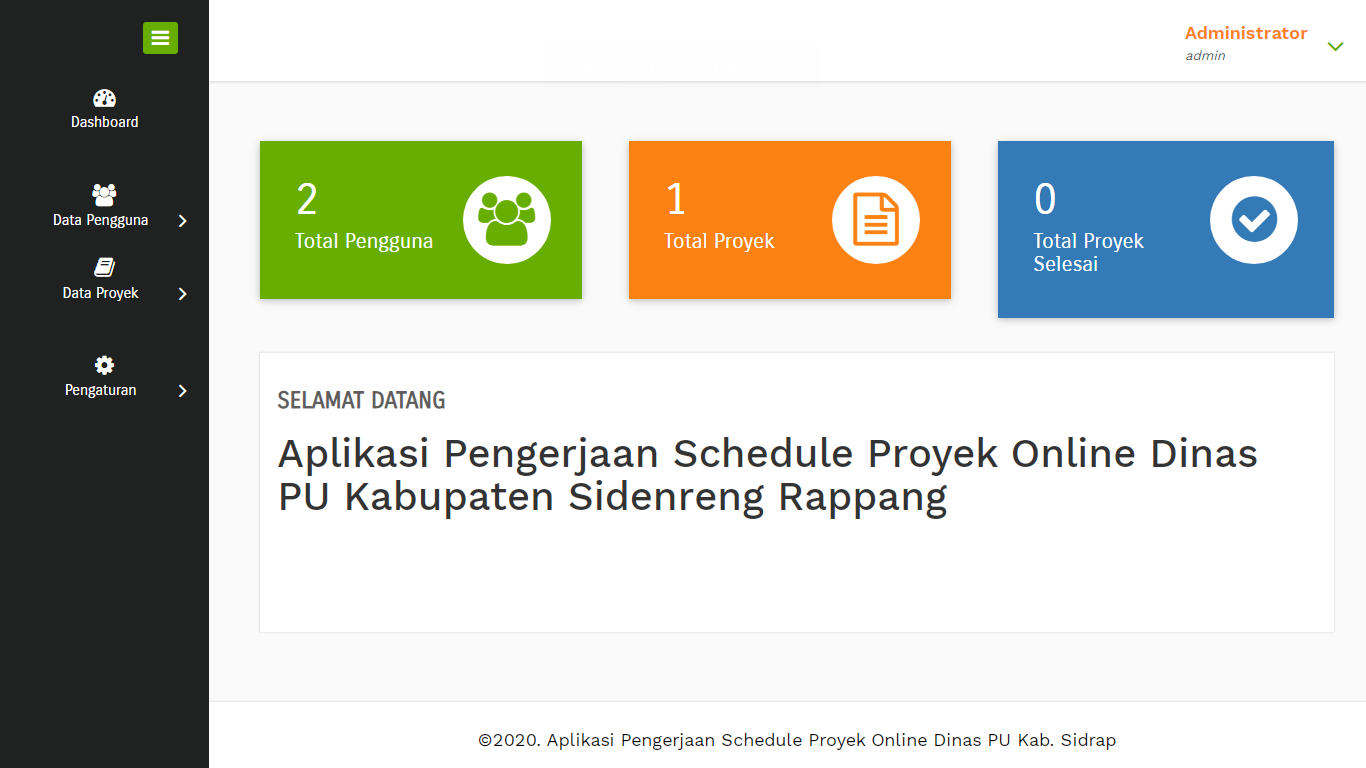
*Login* merupakan tampilan awal program yang digunakan untuk masuk ke akun *admin.*



Gambar 4.17 Tampilan *Login Admin*

1. *Dashboard*

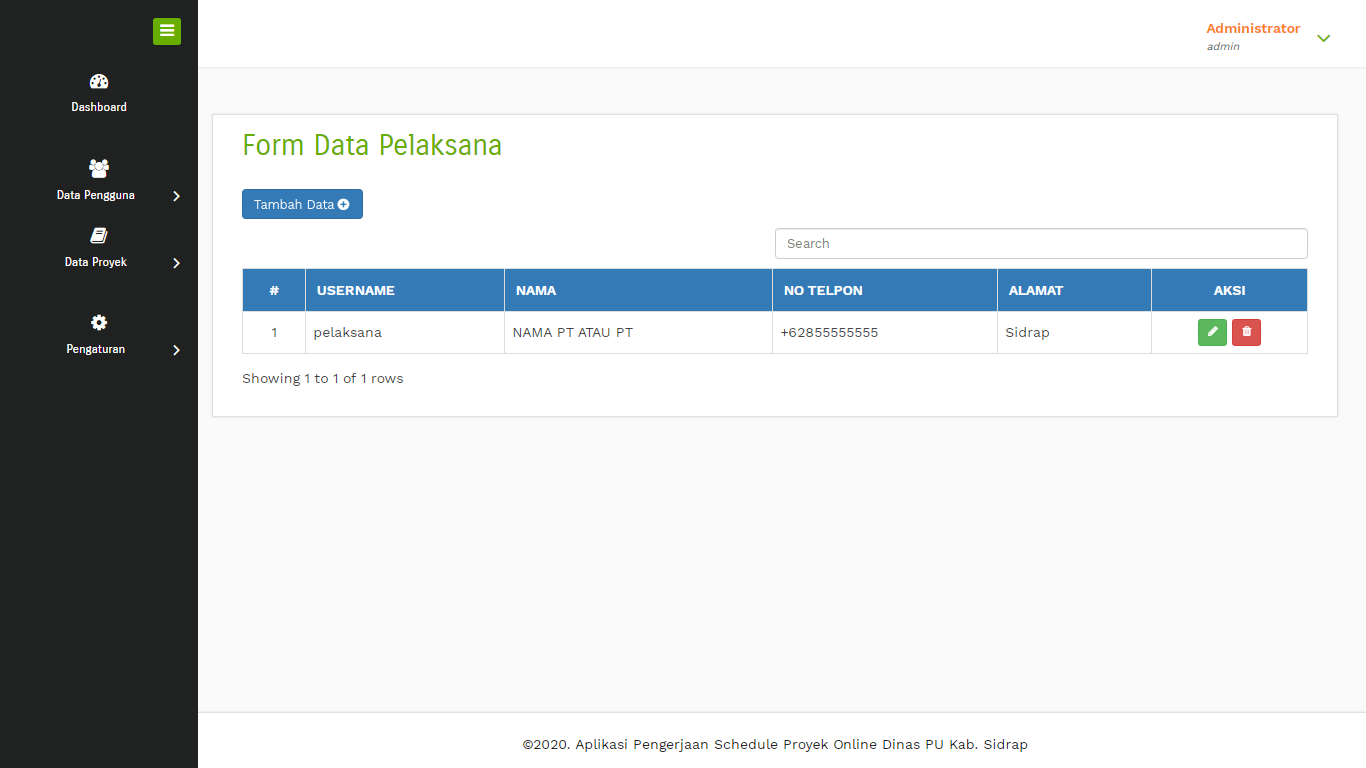
*Dashboard* yang berfungsi menampilkan informasi proyek, pengguna dan total proyek yang telah di *input.*



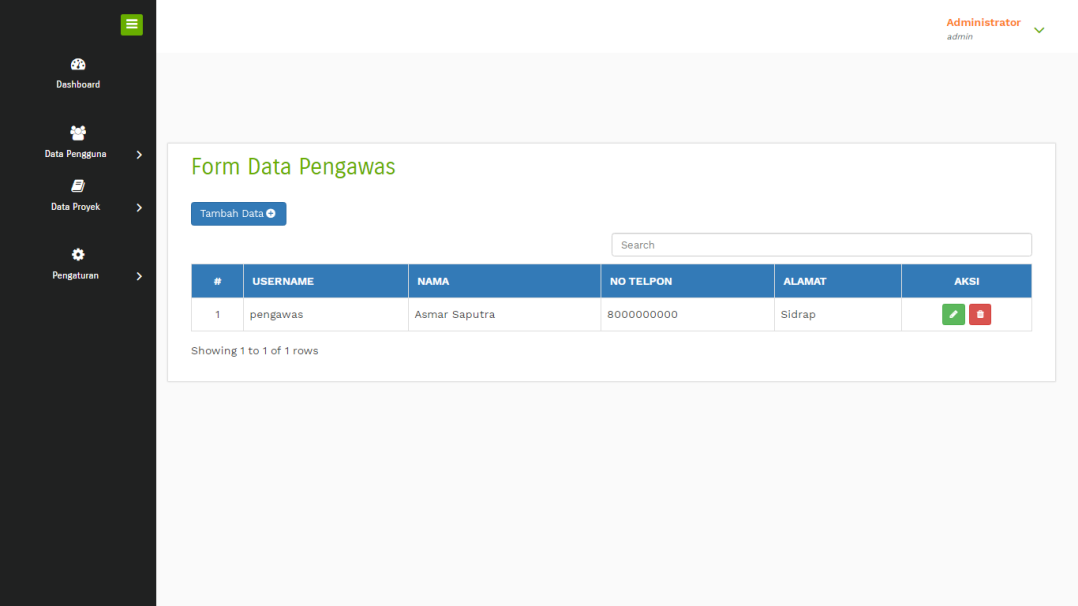
Gambar 4.18 Tampilan Dashboard Admin

1. Data Pengguna

Data pengguna yang berfungsi untuk mengatur data Pelaksanan dan Data Pengawas.



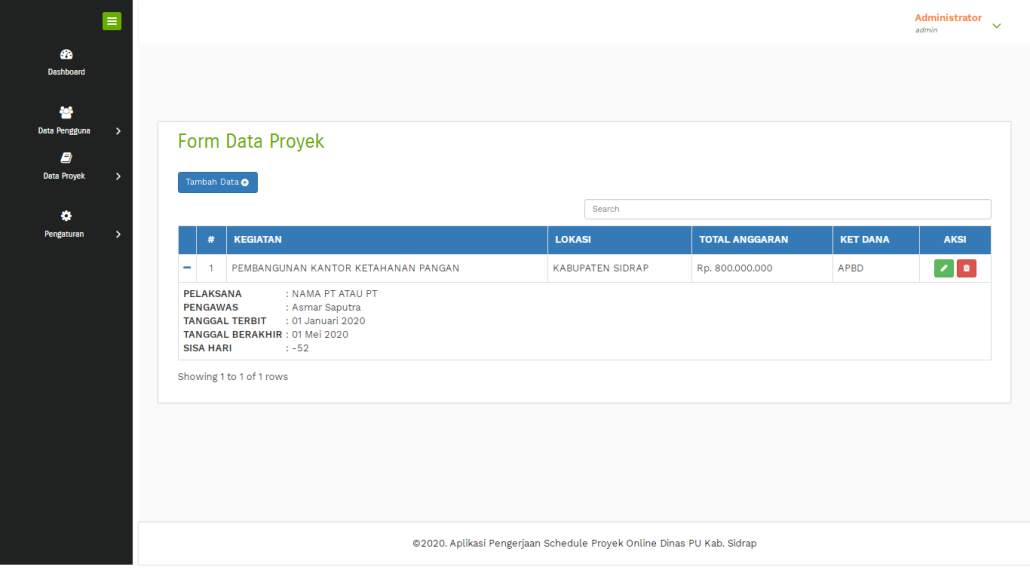
Gambar 4.19 Tampilan Form Data Pelaksana



Gambar 4.20 Tampilan Form Data Pengawas

1. Data Proyek

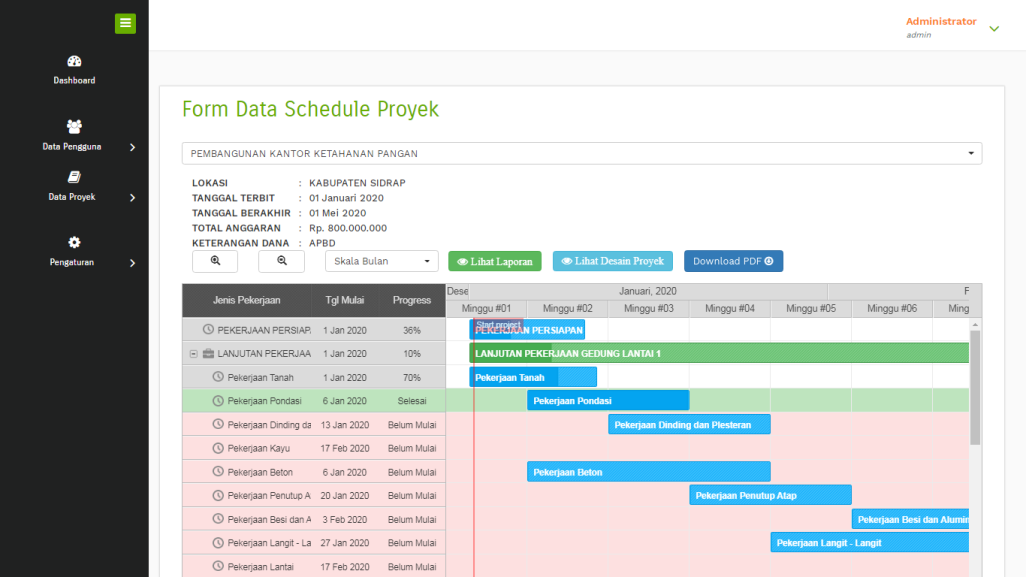
*Form* Data Proyek berfungi untuk mengatur data proyek.



Gambar 4.21 Tampilan Form Data Proyek Admin

1. Data *Schedule*

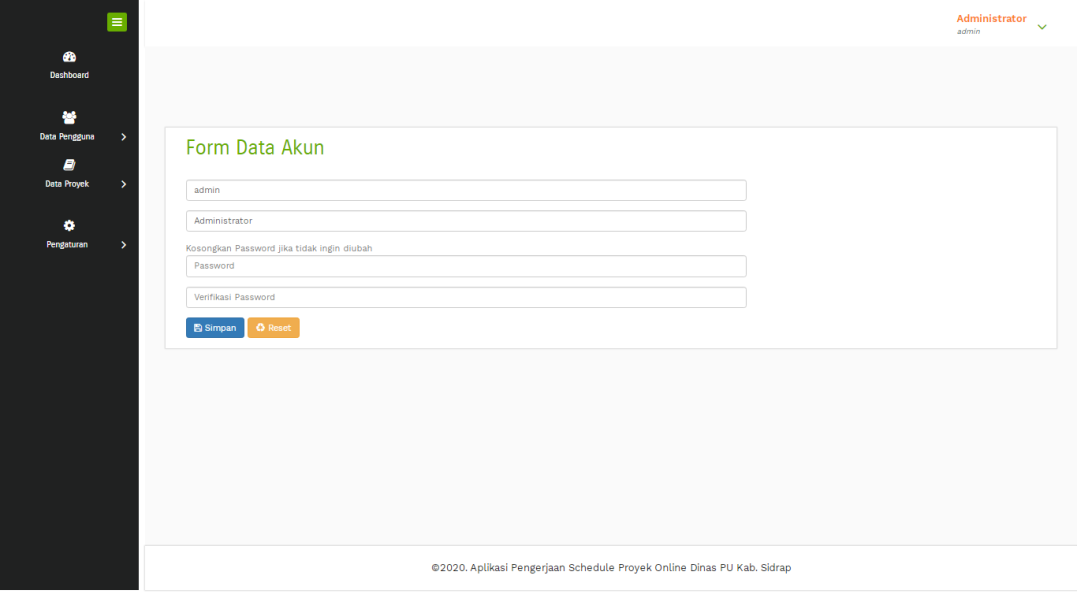
*Form* data *schedule* berfungi untuk memantau *schedule* dan pengembangan pelaksanan proyek, dan juga berfungsi untuk melihat daftar laporan proyek.



Gambar 4.22 Tampilan Form Data Schedule Proyek Admin

1. Form Data Akun

*Form* akun berfungsi untuk mengatur data akun *admin.*

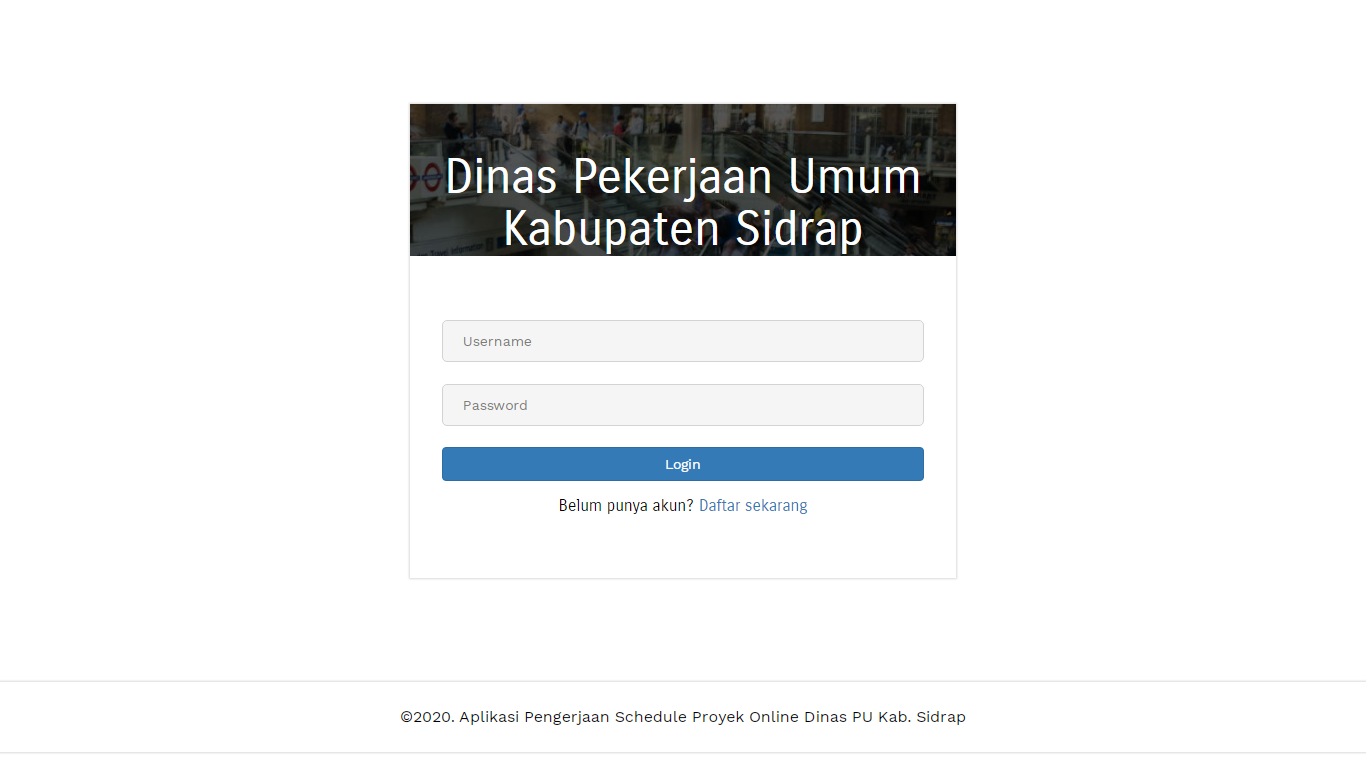


Gambar 4.23 Tampilan Form Data Akun

* + - 1. Pelaksana

1. *Login*

*Login* merupakan tampilan awal program yang digunakan untuk masuk ke akun pelaksana*.*



Gambar 4.24 Tampilan *Login* Pelaksana

1. *Dashboard*

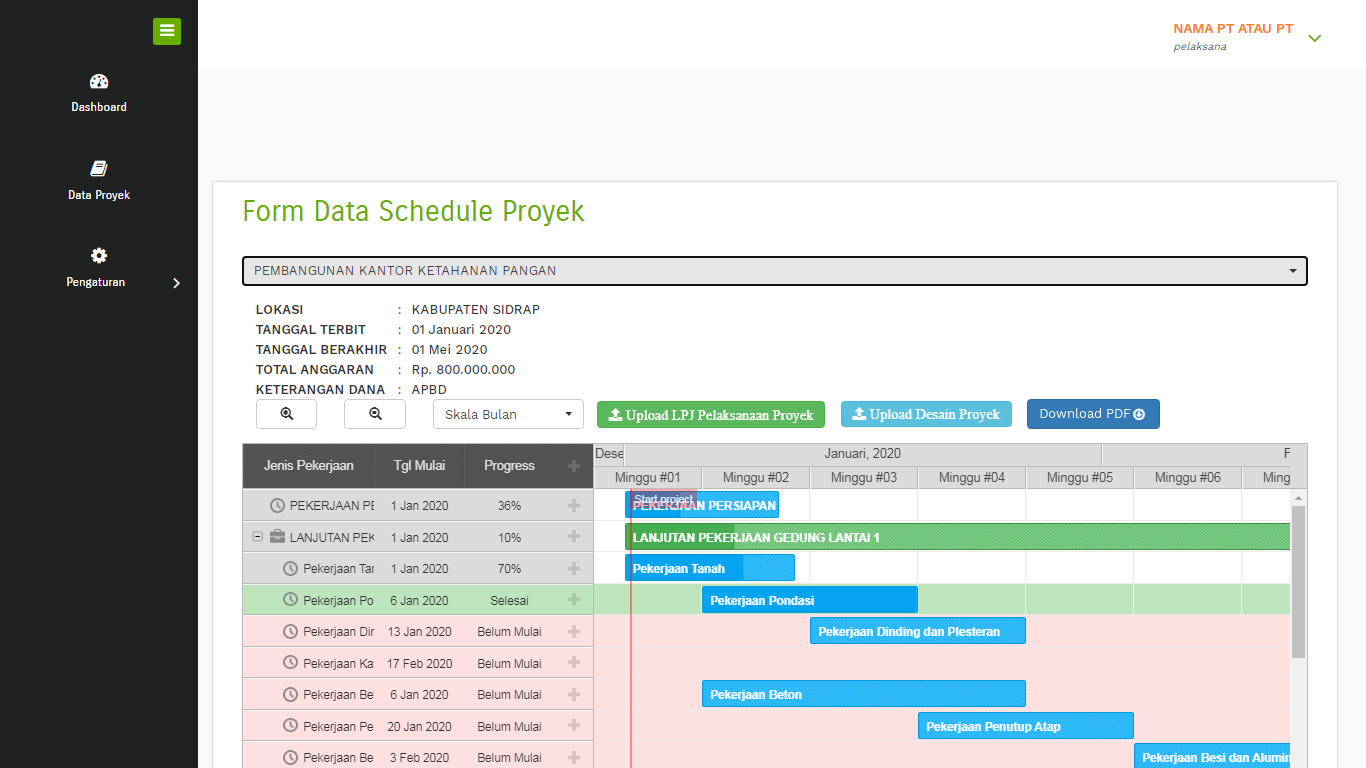
*Dashboard* yang berfungsi menampilkan informasi total proyek dan total proyek telah selesai yang dikerjakan oleh pelaksana*.*

**

Gambar 4.25 Tampilan *Dashboard* Pelaksana

1. Data Proyek

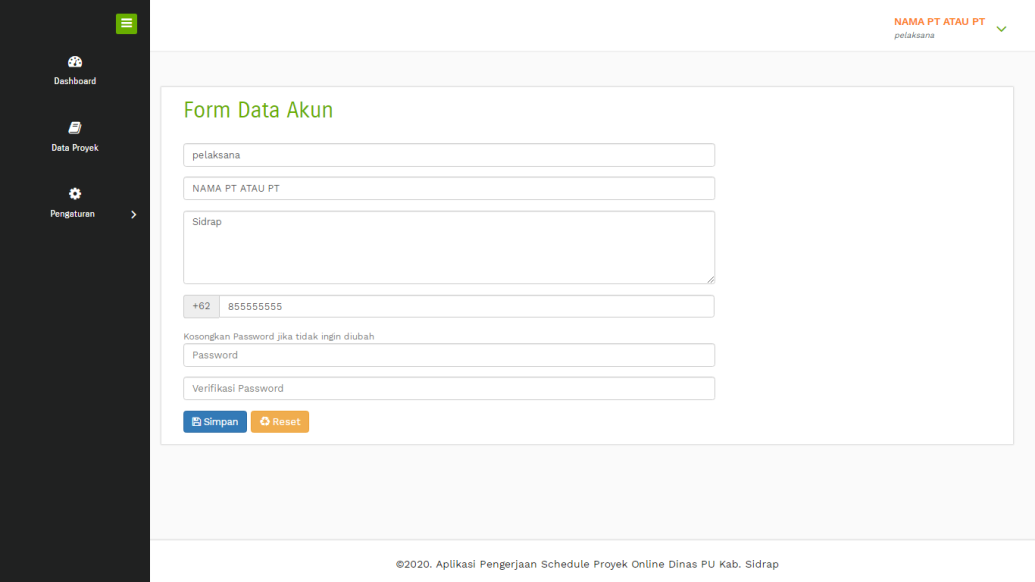
Data Proyek berfungsi untuk mengelola *schedule* proyek pelaksana dan meng-*upload* berkas laporan LPJ.



Gambar 4.26 Tampilan *Form* Data *Schedule* Proyek

1. *Form* Data Akun

*Form* data akun berfungsi untuk mengatur akun pelaksana.

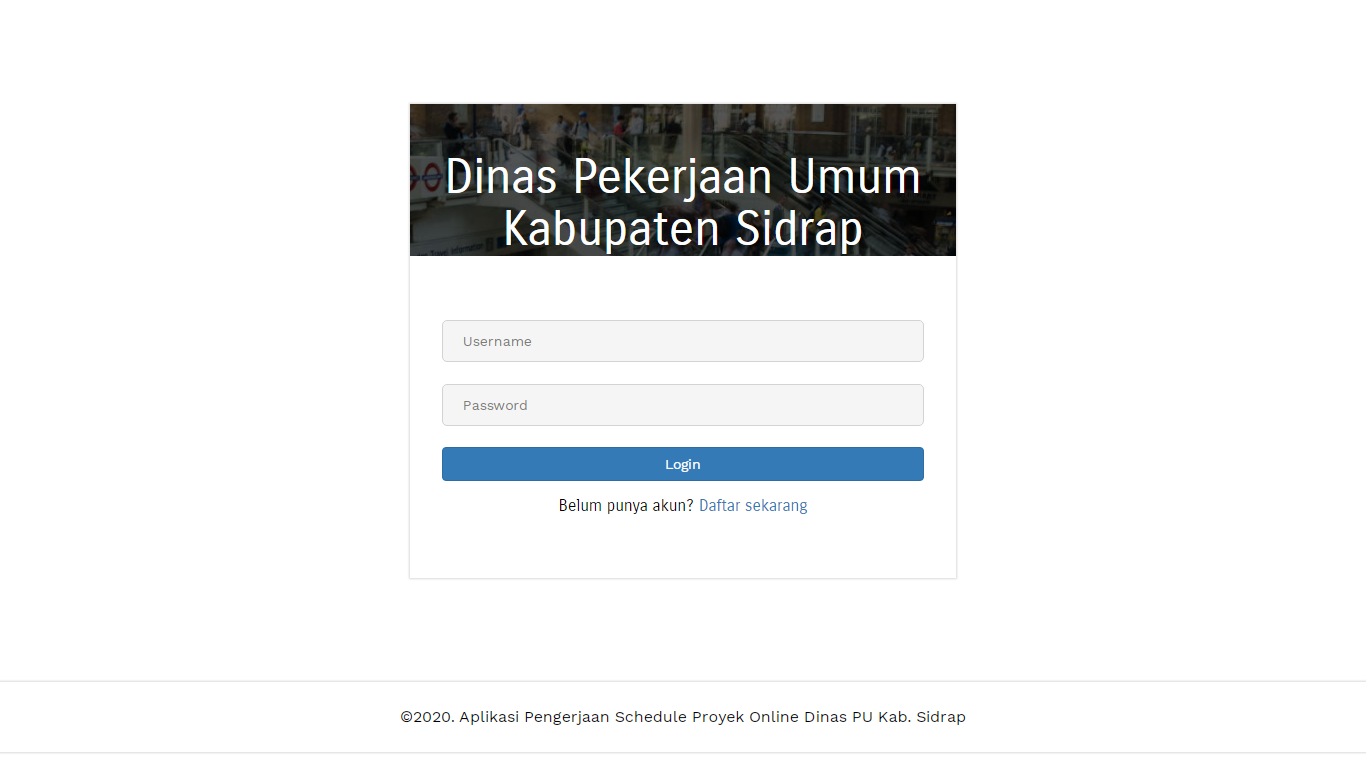


Gambar 4.27 Tampilan *Form* Data Akun

* + - 1. Pengawas

1. *Login*

*Login* merupakan tampilan awal program yang digunakan untuk masuk ke akun pengawas.

**

Gambar 4.28 Tampilan *Login* Pengawas

1. *Dashboard*

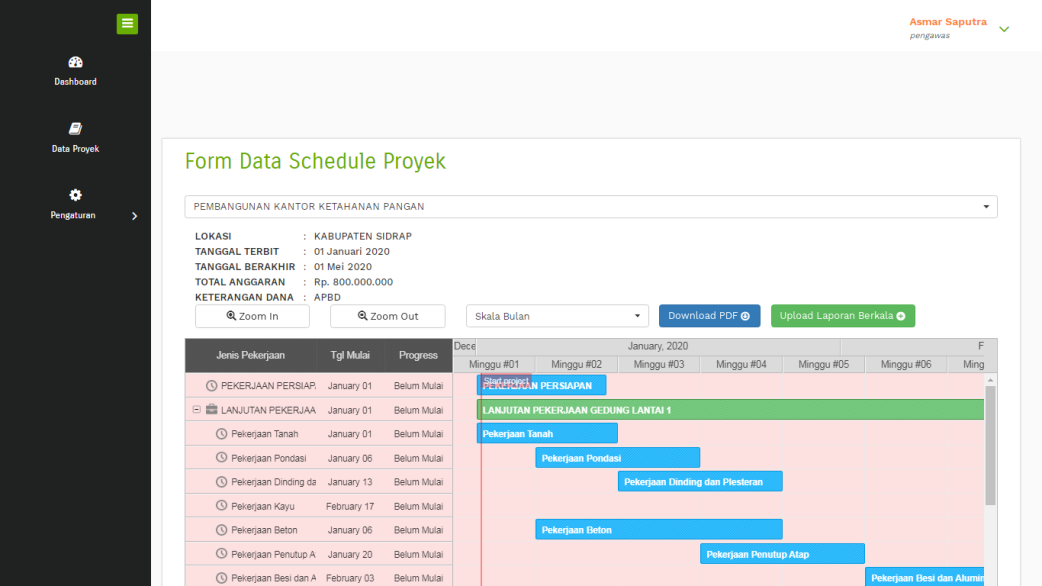
*Dashboard* yang berfungsi menampilkan informasi total proyek dan total proyek telah selesai yang diawasi oleh pengawas*.*

**

Gambar 4.29 Tampilan *Dashboard* Pengawas

1. Data Proyek

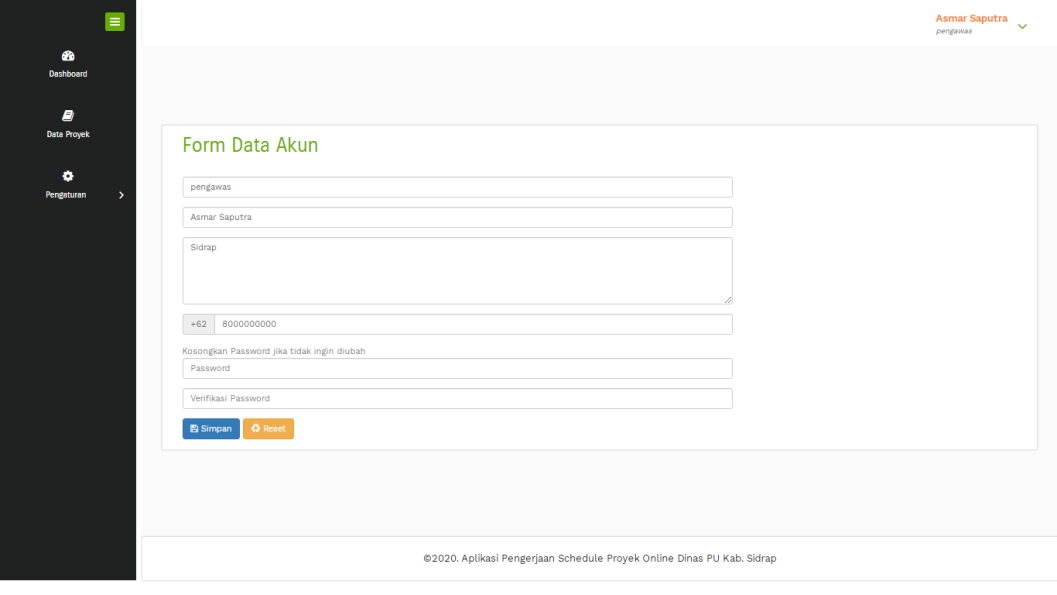
Data Proyek berfungsi untuk meng-*upload* laporan berkala dan mengubah progress *schedule* proyek.



Gambar 4.30 Tampilan *Form* Data *Schedule* Proyek Pengawas

1. *Form* Data Akun

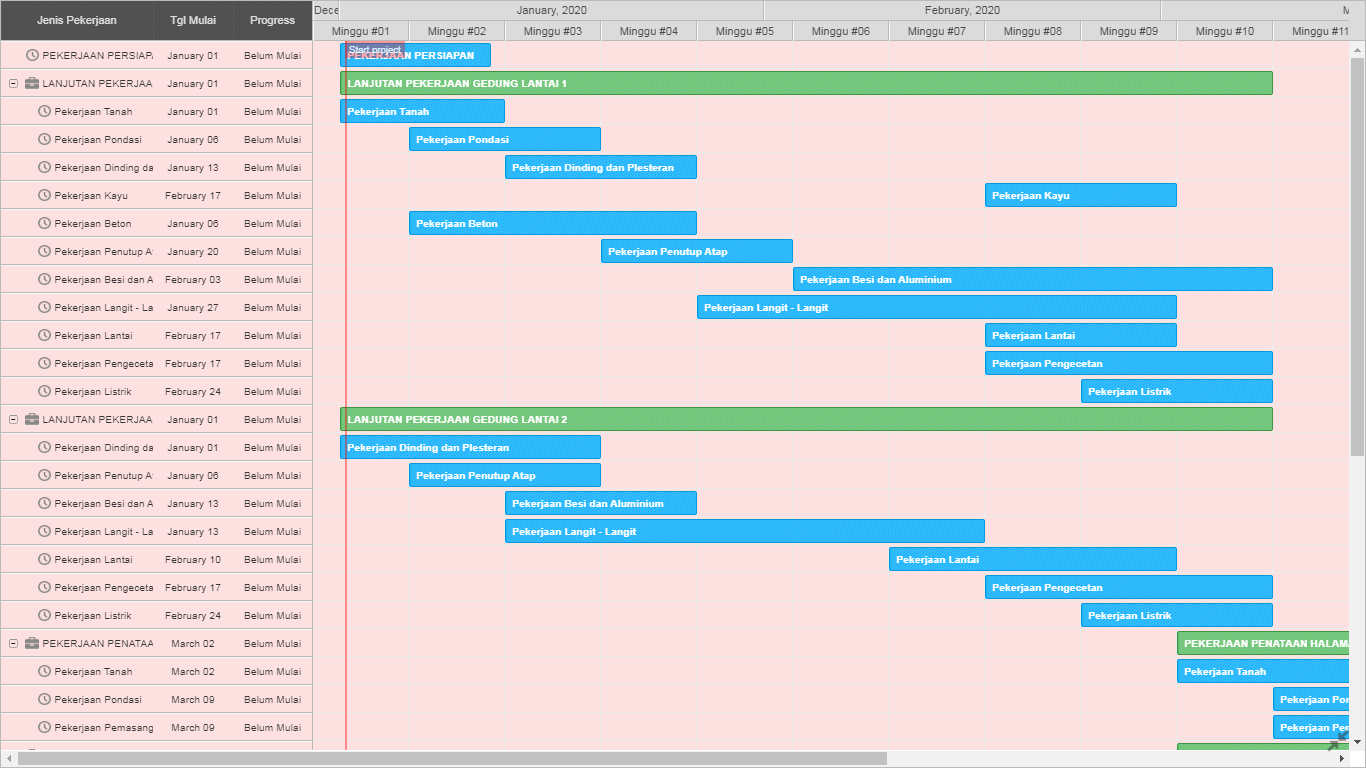
*Form* Data Akun berfungsi untuk mengatur akun pengawas.



Gambar 4.31 Tampilan *Form* Data Akun Pengawas

1. **Rancangan Output Program**

Perancangan *Output* Data Schedule pelaksanaan proyek.



Gambar 4.32 Tampilan *Schedule* Pelaksanaan Proyek

# BAB V PENGUJIAN SISTEM

1. **Metode Pengujian**

Pengujian sistem merupakan proses eksekusi suatu program atau sistem secara keseluruhan. Pengujian menunjukkan bahwa fungsi dari perangkat lunak bekerja sesuai dengan spesifikasi dan bahwa persyaratan kinerja telah dipenuhi.

Metode pengujian yang digunakan ada *black-box*. Pengujian *blackbox* perangkat lunak dilakukan untuk memperlihatkan bahwa masing-masing fungsi telah beroperasi sepenuhnya dan secara bersamaan pula mencari kesalahan pada setiap fungsi.

1. **Teknik Pengujian**

Pengujian *black-box* berkaitan dengan pengujian yang dilakukan pada *interface* perangkat linak. Meskipun didesain untuk mengungkapkan kesalahan, pengujian *black-box* digunakan untuk memperlihatkan bahwa fungsi-fungsi perangkat lunak adalah operasional, bahwa *input* diterima dengan baik dan *output* dihasilkan dengan tepat, dan integritas informasi *external* (seperti file data) dipelihara. Pengujian *black-box* menguji beberapa aspek dasar suatu sistem dengan sedikit memperhatikan struktur logika internal perangkat lunak tersebut.

1. Pengujian *Black Box*
   * + - 1. Pengujian pada *Form Login*

Tabel 5.1 Pengujian pada *Form Login*

|  |  |  |
| --- | --- | --- |
| *Test Factor* | Hasil | Keterangan |
| Jika salah memasukkan username dan password |  | Berhasil, karena username dan password tidak sesuai maka tidak dapat login. |
| *Screen Shoot* | | |
|  | | |

Tabel 5.2 Pengujian Pada *Form Login*

|  |  |  |
| --- | --- | --- |
| *Test Factor* | Hasil | Keterangan |
| Jika user dan password di masukan sesuai |  | Berhasil, karena dapat menampilkan halaman admin. |
| *Screen Shoot* | | |
|  | | |

* + - * 1. Pengujian Pada *Form* Data Pengguna

Tabel 5.3 Pengujian Pada *Form* Data Pengguna

|  |  |  |
| --- | --- | --- |
| *Test Factor* | Hasil | Keterangan |
| Memasukkan data Pengguna |  | Berhasil, karena data sukses di simpan. |
| *Screen Shoot* | | |
|  | | |

* + - * 1. Pengujian Pada *Form* Data Proyek

Tabel 5.4 Pengujian Pada *Form* Data Proyek

|  |  |  |
| --- | --- | --- |
| *Test Factor* | Hasil | Keterangan |
| Memasukkan data Proyek |  | Berhasil, karena data sukses di simpan. |
| *Screen Shoot* | | |
|  | | |

* + - * 1. Pengujian Pada *Form* Data *Schedule Project*

Tabel 5.5 Pengujian Pada *Form* Data *Schedule Project*

|  |  |  |
| --- | --- | --- |
| *Test Factor* | Hasil | Keterangan |
| Memasukkan data Schedule Proyek |  | Berhasil, karena data sukses di simpan. |
| *Screen Shoot* | | |
|  | | |

* + - * 1. Pengujian Pada *Form Upload* Laporan Berkala

Tabel 5.6 Pengujian Pada *Form Upload* Laporan Berkala

|  |  |  |
| --- | --- | --- |
| *Test Factor* | Hasil | Keterangan |
| Memasukkan data Laporan Berkala |  | Berhasil, karena berkas sukses di upload. |
| *Screen Shoot* | | |
|  | | |

* + - * 1. Pengujian Pada *Form Upload* LPJ

Tabel 5.7 Pengujian Pada *Form Upload* LPJ

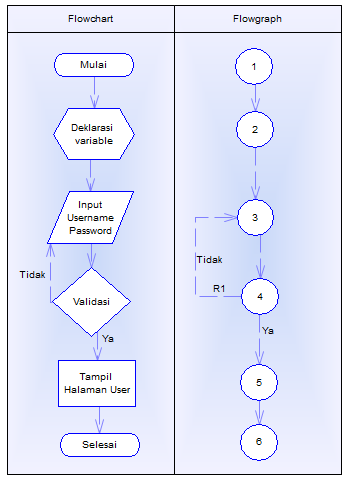
|  |  |  |
| --- | --- | --- |
| *Test Factor* | Hasil | Keterangan |
| Memasukkan data Laporan LPJ |  | Berhasil, karena berkas sukses di upload. |
| *Screen Shoot* | | |
|  | | |

* + - * 1. Pengujian Pada *Form* Data Akun

Tabel 5.8 Pengujian Pada *Form* Data Akun

|  |  |  |
| --- | --- | --- |
| *Test Factor* | Hasil | Keterangan |
| Memasukkan data akun |  | Berhasil, karena data sukses di simpan. |
| *Screen Shoot* | | |
|  | | |

1. Pengujian *White Box*
2. *Login*



Gambar 5.1 Pengujian Login

Dari gambar *flowgraph* Menu *Login* di atas dapat dilakukan proses perhitungan sebagai berikut:

1. Menghitung *Cyclomatic ComplexcityV(G)* dari *Egde* dan *Node*:

Dengan Rumus : V (G) = E – N + 2

Dengan E (*edge*) = 6

N (*node*) = 6

Predikat *Node* (P) = 1

Penyelesaian :V(G) = E – N + 2

= 6 – 6 + 2

= 2

Predikat (P) = P + 1

= 1 + 1

= 2

2. Berdasarkan perhitungan *Cyclomatic Complexcity* dari *Flowgraph* diatas memiliki Region = 2

3. *Independent Path* pada *flowgraph* diatas adalah :

*Path* 1 = 1 – 2 – 3 – 4 – 3

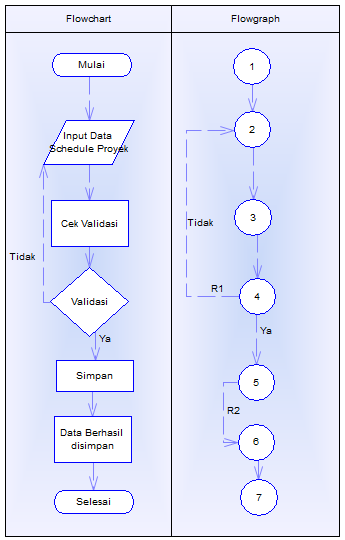
*Path* 2 = 1 – 2 – 3 – 4 – 5 – 6

1. Grafis Matriks

Tabel 5.9 Grafik Matriks dari *Flowgraph Login*

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

1. *Input* Data



Gambar 5.2 Pengujian *Input* Data

Dari gambar flowgraph Input Data di atas dapat dilakukan proses perhitungan sebagai berikut:

1. Menghitung *Cyclomatic ComplexcityV(G)* dari *Egde* dan *Node*:

Dengan Rumus : V (G) = E – N + 2

Dengan E (*edge*) = 7

N (*node*) = 7

Predikat *Node* (P) = 1

Penyelesaian :V(G) = E – N + 2

= 7 – 7 + 2 = 2

Predikat (P) = P + 1

= 1 + 1 = 2

2. Berdasarkan perhitungan *Cyclomatic Complexcity* dari *Flowgraph* diatas memiliki Region = 2

3. *Independent Path* pada *flowgraph* diatas adalah :

*Path* 1 = 1 – 2 – 3 – 4 – 2

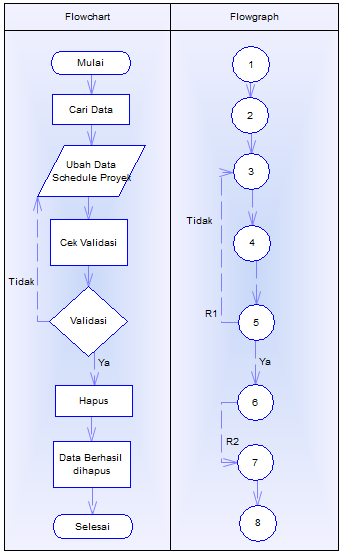
*Path* 2 = 1 – 2 – 3 – 4 – 5 – 6 – 7

1. Grafik Matriks

Tabel 5.10 Grafik Matriks pada *Flowgraph Input* Data

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | E – 1 = 0 |
| 1 |  | 1 |  |  |  |  |  | 1 – 1 = 0 |
| 2 |  |  | 1 |  |  |  |  | 1 – 1 = 0 |
| 3 |  |  |  | 1 |  |  |  | 1 – 1 = 0 |
| 4 |  | 1 |  |  | 1 |  |  | 2 – 1 = 1 |
| 5 |  |  |  |  |  | 1 |  | 1 – 1 = 0 |
| 6 |  |  |  |  |  |  | 1 | 1 – 1 = 0 |
| 7 |  |  |  |  |  |  |  | 0 |
| Sum (E) + 1 | | | | | | | | 1 + 1 =2 |

1. Edit Data



Gambar 5.3 Pengujian *Edit* Data

Dari gambar *flowgraph* *Edit* Data di atas dapat dilakukan proses perhitungan sebagai berikut:

1. Menghitung *Cyclomatic ComplexcityV(G)* dari *Egde* dan *Node*:

Dengan Rumus : V (G) = E – N + 2

Dengan E (*edge*) = 8

N (*node*) = 8

Predikat Node (P) = 1

Penyelesaian :V(G) = E – N + 2

= 8 – 8 + 2 = 2

Predikat (P) = P + 1

= 1 + 1 = 2

2. Berdasarkan perhitungan *Cyclomatic Complexcity* dari *Flowgraph* diatas memiliki Region = 2

3. *Independent Path* pada *flowgraph* diatas adalah :

*Path* 1 = 1 – 2 – 3 – 4 – 5 – 3

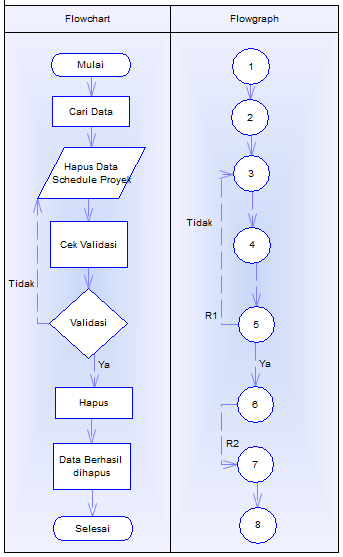
*Path* 2 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8

1. Grafik Matriks

Tabel 5.11 Grafik Matriks pada *Flowgraph* Edit Data

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | E – 1 = 0 |
| 1 |  | 1 |  |  |  |  |  |  | 1 – 1 = 0 |
| 2 |  |  | 1 |  |  |  |  |  | 1 – 1 = 0 |
| 3 |  |  |  | 1 |  |  |  |  | 1 – 1 = 0 |
| 4 |  |  |  |  | 1 |  |  |  | 1 – 1 = 0 |
| 5 |  |  | 1 |  |  | 1 |  |  | 2 – 1 = 1 |
| 6 |  |  |  |  |  |  | 1 |  | 1 – 1 = 0 |
| 7 |  |  |  |  |  |  |  | 1 | 1 – 1 = 0 |
| 8 |  |  |  |  |  |  |  |  | 0 |
| Sum (E) + 1 | | | | | | | |  | 1 + 1 =2 |

1. Hapus Data



Gambar 5.4 Pengujian Hapus Data

Dari gambar *flowgraph* Hapus Data di atas dapat dilakukan proses perhitungan sebagai berikut:

1. Menghitung *Cyclomatic ComplexcityV(G)* dari *Egde* dan *Node*:

Dengan Rumus : V (G) = E – N + 2

Dengan E (*edge*) = 8

N (*node*) = 8

Predikat *Node* (P) = 1

Penyelesaian :V(G) = E – N + 2

= 8 – 8 + 2 = 2

Predikat (P) = P + 1

= 1 + 1= 2

2. Berdasarkan perhitungan *Cyclomatic Complexcity* dari *Flowgraph* diatas memiliki Region = 2

3. *Independent Path* pada *flowgraph* diatas adalah :

*Path* 1 = 1 – 2 – 3 – 4 – 5 – 3

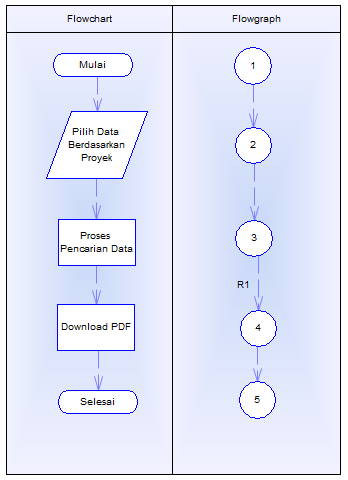
*Path* 2 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8

1. Grafik Matriks

Tabel 5.12 Grafik Matriks pada *Flowgraph* Hapus Data

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | E – 1 = 0 |
| 1 |  | 1 |  |  |  |  |  |  | 1 – 1 = 0 |
| 2 |  |  | 1 |  |  |  |  |  | 1 – 1 = 0 |
| 3 |  |  |  | 1 |  |  |  |  | 1 – 1 = 0 |
| 4 |  |  |  |  | 1 |  |  |  | 1 – 1 = 0 |
| 5 |  |  | 1 |  |  | 1 |  |  | 2 – 1 = 1 |
| 6 |  |  |  |  |  |  | 1 |  | 1 – 1 = 0 |
| 7 |  |  |  |  |  |  |  | 1 | 1 – 1 = 0 |
| 8 |  |  |  |  |  |  |  |  | 0 |
| Sum (E) + 1 | | | | | | | |  | 1 + 1 =2 |

1. *Download* *Schedule* Proyek



Gambar 5.5 Pengujian *Download Schedule* Proyek

Dari gambar *flowgraph* *Download Schedule* Proyek di atas dapat dilakukan proses perhitungan sebagai berikut:

1. Menghitung *Cyclomatic ComplexcityV(G)* dari *Egde* dan *Node*:

Dengan Rumus : V (G) = E – N + 2

Dengan E (*edge*) = 4

N (*node*) = 5

Predikat *Node* (P) = 0

Penyelesaian :V(G) = E – N + 2

= 4 – 5 + 2

= 1

Predikat (P) = P + 1

= 0 + 1

= 1

2. Berdasarkan perhitungan *Cyclomatic Complexcity* dari *Flowgraph* diatas memiliki Region = 1

3. *Independent Path* pada flowgraph diatas adalah :

*Path* 1 = 1 – 2 – 3 – 4 – 5

1. Grafik Matriks

Tabel 5.13 Grafik Matriks pada *Flowgraph* Cetak Data Laporan

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | E – 1 = 0 |
| 1 |  | 1 |  |  |  | 1 – 1 = 0 |
| 2 |  |  | 1 |  |  | 1 – 1 = 0 |
| 3 |  |  |  | 1 |  | 1 – 1 = 0 |
| 4 |  |  |  |  | 1 | 1 – 1 = 0 |
| 5 |  |  |  |  |  | 0 |
| Sum ( E ) + 1 | | | | | | 0 + 1 =1 |

# BAB VI PENUTUP

1. **Kesimpulan**

Dihasilkannya sebuah Aplikasi *schedule* pengerjaan proyek online Dinas PU Kab. Sidenreng Rappang dengan menggunakan bahasa pemrograman PHP untuk *Database*-nya menggunakan *Mysql*. Pengelolaan *Schedule* Proyek dengan cara *onlin*e akan lebih efektif dan efesien dan dapat di pantau secara berkala oleh operator Dinas PU Kab. Sidenreng Rappang. Aplikasi yang dibuat sudah bebas dari kesalahan karena diuji dengan menggunakan dua teknik pengujian yaitu pengujian *Black Box* dan pengujian *White Box*.

1. Untuk membuat Aplikasi *schedule* pengerjaan proyek online Dinas PU Kab. Sidenreng Rappang telah dibuktikan dengan menggunakan metode Pengujian *Black Box* menyatakan aplikasi yang dihasilkan sudah berjalan sesuai dengan kebutuhan dan bebas dari kesalahan dan pegujian *White Box Testing* untuk mendapatkan cara kerja program secara rinci. Pada pengujiannya, struktur logika dipetakan dengan *Flowchart* dan *Flowgraph* kemudian jumlah edge dan nodenya dijumlahkan sehingga memperoleh nilai *Cyclometic Complexcity.* Berdasarkan hasil pengujian pada *login, home,input* data,*edit* data,hapus data, *download* *schedule* proyek diperoleh nilai *Independent Path, Region, dan Cyclomatic Complexity* bernilai sama sehingga dapat dikatakan aplikasi yang dibuat sudah berjalan sesuai dengan kebutuhan.
2. Dengan meng-*upload* laporan *schedule* pengerjaan proyek secara *online* maka dapat dipantau secara *up to date* oleh pihak Dinas PU Kab. Sidenreng Rappang dan dapat meminimalisir permasalahan dalam melaksanakan proyek tersebut karena ketika terjadi keterlambatan pengerjaan maka dapat lebih cepat dilakukan tindakan lebih lanjut.
3. **Saran-saran**

Dalam Aplikasi *schedule* pengerjaan proyek online ini masih terdapat beberapa kekurangan, sehingga penulis memberikan saran terhadap bagian-bagian tertentu, yaitu :

1. Hendaknya sistem yang penulis rancang dapat dikembangkan sehingga dalam pengolahan datanya dapat lebih dijelaskan secara mendetail, agar pengolahan data dan pembuatan laporan dapat dilakukan lebih efektif dan efesien.
2. Sistem informasi dan aplikasi yang telah dikembangkan penulis sebaiknya dipelihara dengan sebaik-baiknya dan di perbaharui sesuai dengan kebutuhan yang ada serta melakukan *backup* data pada jangka waktu tertentu untuk menghindari kemungkinan data hilang atau rusak.

# DAFTAR PUSTAKA

Alan Nur Aditya, 2011, “*Jago PHP dan MySQL*”, Dunia Komputer.

Arief M Rudianto. 2011. Pemrograman *Web Dinamis menggunakan PHP dan MySQL*. Yogyakarta : Andi.

Bunafit Nugroho, 2009, “Panduan Lengkap menguasai Perintah SQL”, Mediakita.

Hakim, Lukmanul. 2010. *Membangun Web Berbasis PHP dengan Framework Codeigniter*. Yogyakarta : Lokomedia.

Hartono, Jogiyanto.Analisis dan Desain Sistem Informasi : Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis. Yogyakarta:Andi.1999.

Jayan. 2010. *CSS untuk Orang Awam*. Palembang: Maxikom.

Jogiyanto.Analisis dan Desain.Yogyakarta: Andi, 2005.

Nugroho, Bunafit. 2004. *PHP dan MySQL dengan editor Dreamweaver MX*. Yogyakarta : Andi.

Sholiq, 2006.Pemodelan Sistem Informasi Berorientasi Objek Dengan UML, Yogyakarta:GrahaIlmu.

Sidik, B. & Pohan, H.I. 2007. *Pemrograman Web dengan HTML*. Bandung: Informatika.

Sibero, Alexander F.K . 2013. *Web Programming Power Pack*. Yogyakarta : Mediakom.

Sofwan, Akhmad. 2010. Belajar PHP dengan Framework CodeIgniter. <http://ilmukomputer.org/wp-content/uploads/2010/05/belajar-php-dengan-framework-code-igniter.pdf>

Utomo, E.P. 2007. *55 Tips Mempercantik Website dengan Javascript*. Jakarta: Restu Agung.

Wahana Komputer. Android Programming With Eclipse. Yogyakarta : Andi. 2013.

**LAMPIRAN**