

## **PROPOSAL**

### **IMPLEMENTASI METODE PEWARNAAN GRAF (*GRAPH COLORING*) MENGUNAKAN ALGORITMA *GREEDY* UNTUK MENENTUKAN JADWAL MATA PELAJARAN PADA SMKN 4 KENDARI**

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Sarjana Teknik



**MUHAMMAD FADEL**

**E1E117018**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS HALU OLEO**

**KENDARI**

**2021**

## HALAMAN PENGESAHAN

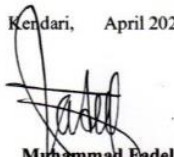
### Seminar Proposal

#### IMPLEMENTASI METODE PEWARNAAN GRAF (*GRAPH COLORING*) MENGUNAKAN ALGORITMA *GREEDY* UNTUK MENENTUKAN JADWAL MATA PELAJARAN PADA SMKN 4 KENDARI

(Studi Kasus : Sekolah Menengah Kejuruan Negeri 4 Kendari)

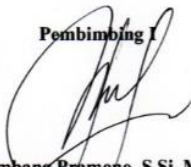
Adalah benar dibuat oleh saya sendiri dan belum pernah dibuat dan diserahkan sebelumnya baik sebagian ataupun seluruhnya, baik oleh saya ataupun orang lain, baik di Universitas Halu Oleo ataupun institusi pendidikan lainnya.

Kendari, April 2021




Muhammad Fadel  
NIM. E1E1 17 018

Pembimbing I



Bambang Pramono, S.Si., M.T.  
NIP. 19710425 200801 1 010

Pembimbing II



Statiswaty, ST., MMSI  
NIP. 19811107 200812 2 003

## DAFTAR ISI

<b>HALAMAN SAMPUL.....</b>	<b>i</b>
<b>HALAMAN PENGESAHAN.....</b>	<b>ii</b>
<b>DAFTAR ISI.....</b>	<b>iii</b>
<b>DAFTAR TABEL.....</b>	<b>v</b>
<b>DAFTAR GAMBAR.....</b>	<b>vii</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah .....	2
1.3. Batasan Masalah.....	3
1.4. Tujuan Penelitian.....	3
1.5. Manfaat Penelitian.....	3
1.6. Sistematika Penulisan.....	3
1.7. Tinjauan pustaka.....	4
<b>BAB II LANDASAN TEORI .....</b>	<b>6</b>
2.1. Penjadwalan Mata Pelajaran.....	6
2.1.1. Penyusunan Jadwal Pelajaran.....	6
2.1.2. Komponen-Komponen Dalam Penyusunan Jadwal Pelajaran .....	7
2.1.3. Penyusunan Program Mengajar .....	7
2.2. Metode <i>Graph colouring</i> .....	8
2.2.1. Jenis <i>graph</i> .....	10
2.2.2. Terminologi <i>graph</i> .....	12
2.3. Algoritma <i>Greedy</i> .....	13
2.4. <i>Hypertext preprocessor</i> (PHP) .....	18

2.5.	<i>Framework Codeigniter</i> .....	19
2.6.	<i>Database Management System</i> .....	20
2.7.	MySQL .....	21
2.8.	XAMPP .....	22
2.9.	Metode Perancangan Unified Modeling Language (UML) .....	23
2.10.	<i>Rational Unified Process</i> (RUP).....	28
<b>BAB III METODOLOGI PENELITIAN .....</b>		<b>30</b>
3.1	Waktu dan Tempat Penelitian .....	30
3.3.1	Waktu .....	30
3.3.2	Tempat.....	30
3.2	Metode Pengumpulan Data .....	30
3.3	Metode Pengembangan Sistem.....	31
3.4	Analisis Sistem .....	32
3.4.1	Analisis Kebutuh Fungsional.....	32
3.4.2	Analisis Kebutuhan Non Fungsional.....	32
3.5	Analisis Perancangan Sistem.....	33
3.5.1	Gambaran Umum Sistem yang Sedang Berjalan .....	34
3.5.2	Gambaran Umum Sistem yang Diusulkan .....	35
3.5.3	<i>Unified Modeling Language</i> (UML).....	35
<b>DAFTAR PUSTAKA .....</b>		<b>53</b>

## DAFTAR TABEL

Tabel 2. 1 Pengurutan tidak menurun untuk simpul berdasarkan jumlah derajat .	15
Tabel 2. 2 Simbol-Simbol <i>Use Case</i> .....	24
Tabel 2. 3 Simbol-Simbol <i>Activity Diagram</i> .....	25
Tabel 2. 4 Simbol-Simbol <i>Class Diagram</i> .....	26
Tabel 2. 5 Simbol-Simbol <i>Sequence Diagram</i> .....	27
Tabel 3. 1 <i>Gannt Chart</i> Waktu Penelitian.....	30
Tabel 3. 2 Spesifikasi perangkat keras .....	33
Tabel 3. 3 Spesifikasi perangkat lunak .....	33
Tabel 3. 4 Keterangan <i>Usecase</i> Sistem .....	37

## DAFTAR GAMBAR

Gambar 2. 1 pewarnaan graf pada titik (simpul).....	9
Gambar 2. 2 Pewarnaan graf pada sisi .....	10
Gambar 2. 3 Pewarnaan graf bidang (daerah).....	10
Gambar 2. 4 Contoh a) graf sederhana, b) graf ganda, dan c) graf semu .....	11
Gambar 2. 5 Contoh a) graf berarah dan b) graf ganda berarah .....	11
Gambar 2. 6 Contoh kasus .....	14
Gambar 2. 7 Tahap 1a .....	15
Gambar 2. 8 Tahap 1b.....	16
Gambar 2. 9 Tahap 1c .....	16
Gambar 2. 10 Tahap 2a .....	17
Gambar 2. 11 Tahap 2c .....	17
Gambar 2. 12 Tahap 2c .....	18
Gambar 3. 1 Sistem yang sedang berjalan .....	34
Gambar 3. 2 Sistem yang diusulkan.....	35
Gambar 3. 3 <i>Usecase Diagram</i> Sistem .....	36
Gambar 3. 4 <i>Activity Diagram Login</i> .....	38
Gambar 3. 5 <i>Activity Diagram</i> Data Guru .....	38
Gambar 3. 6 <i>Activity diagram</i> mata pelajaran.....	39
Gambar 3. 7 <i>Activity diagram</i> data kelas .....	40
Gambar 3. 8 <i>Acitivity diagram</i> ketersediaan mengajar .....	40
Gambar 3. 9 <i>Activity diagram</i> data waktu.....	41
Gambar 3. 10 Data jadwal.....	42
Gambar 3. 11 <i>Activity diagram</i> data user.....	43
Gambar 3. 12 <i>Class Diagram</i> .....	43
Gambar 3. 13 <i>Sequence Diagram user</i> .....	44
Gambar 3. 14 <i>Sequence diagram login</i> .....	45
Gambar 3. 15 <i>Sequence Diagram</i> data Guru .....	45
Gambar 3. 16 <i>Sequence diagram</i> mata pelajaran.....	46
Gambar 3. 17 <i>Sequence Diagram</i> data kelas .....	46

Gambar 3. 18 <i>Sequence Diagram</i> data waktu.....	47
Gambar 3. 19 <i>Sequence Diagram</i> jadwal.....	47
Gambar 3. 20 <i>Login</i> .....	48
Gambar 3. 21 Beranda.....	49
Gambar 3. 22 Menu data guru.....	49
Gambar 3. 23 Menu data pelajaran .....	50
Gambar 3. 24 Menu data waktu .....	50
Gambar 3. 25 Menu data kelas.....	51
Gambar 3. 26 Menu kesediaan mengajar .....	51
Gambar 3. 27 Menu jadwal.....	52

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Teknologi informasi yang berkembang semakin pesat dalam segala bidang mendorong manusia dari berbagai Lembaga untuk memanfaatkannya, teknologi informasi yang di maksud adalah untuk mengolah data, seperti memproses, mendapatkan, Menyusun, menyimpan atau memanipulasi data dalam berbagai cara untuk menghasilkan informasi yang berkualitas.

Wujud nyata dari teknologi itu sendiri adalah adanya penerapan sistem komputerisasi bahwa sistem tersebut diharapkan mampu menyelesaikan suatu pekerjaan secara efektif dan efisien yang digunakan untuk keperluan pribadi, bisnis, Pendidikan, Kesehatan, atau pemeritahan dalam mengambil keputusan dengan meminimalisir kesalahan-kesalahan yang terjadi. Pemanfaatan teknologi tentunya disesuaikan dengan kebutuhan dari desain sistem tersebut. Teknologi tidak lagi dipandang hanya pelengkap, tetapi sudah menjadi salah satu penentu atau terlaksananya sasaran dan strategi. Pengolahan data yang dikerjakan secara manual sudah tidak lagi diperlukan.

Pada Lembaga Pendidikan khususnya lingkup akademik SMK (Sekolah Menengah Kejuruan), penjadwalan mata pelajaran merupakan pekerjaan rutin yang dilakukan oleh bagian kurikulum setiap menyabut tahun ajaran baru. Peran bagian kurikulum dalam pembuatan jadwal kegiatan belajar mengajar sangatlah penting dan tidak mudah karena jawdal yang akan diatur ulang terdiri dari data yang sangat banyak.

Berkaitan dengan pembuatan jadwal kegiatan belajar mengajar , saat ini SMK Negeri 4 Kendari dalam pembuatan jadwal mata pelajaran masih dikerjakan secara manual, yaitu menggunakan *Microsoft Office Excel*. Sistem penjadwalan yang diterapkan saat ini adalah berdasarkan hasil rapat guru. Selain itu, masih terdapat kesalahan-kesalahan yang terjadi, seperti adanya mata pelajaran dan guru yang akan mengajar di waktu yang sama tetapi di kelas yang berbeda.



Untuk membantu pengolahan jadwal mata pelajaran di SMK negeri 4 Kendari, diperlukan adanya suatu sistem informasi penjadwalan agar penjadwalan kegiatan belajar mengajar dapat terselesaikan dengan cepat dan akurat serta mengurangi tingkat kesalahan yang biasa terjadi. Penjadwalan mata pelajaran terkait erat dengan masalah optimasi oleh karena itu, pengembangan sistem penjadwalan mata pelajaran dilakukan dengan melalui beberapa iterasi perbaikan. Fungsi tujuannya adalah memenuhi sejumlah aturan penjadwalan, seperti menghindari terjadinya bentrok jadwal. Dalam kajian ilmu di Matematika Diskrit, teori graf memberi solusi permasalahan ini melalui bahasanya tentang pewarnaan graf.

Salah satu algoritma yang dapat digunakan untuk menentukan jumlah warna minimum pada pewarnaan graf adalah algoritma *Greedy*. Algoritma *Greedy* merupakan salah satu jenis algoritma *Largest first*, artinya simpul berderajat tertinggi diwarnai terlebih dahulu. Algoritma *Greedy* ini juga mengambil penyelesaian sementara/lokal yang terbaik dalam setiap langkahnya untuk menyelesaikan suatu permasalahan. Pilihan terbaik akan diambil pada setiap langkahnya tanpa perlu memikirkan bagaimana pengaruhnya terhadap penyelesaian secara keseluruhan.

Berdasarkan latar belakang tersebut penelitian yang akan dilakukan ini berjudul “Implementasi metode pewarnaan graf (*Graph Colouring*) menggunakan Algoritma *Greedy* untuk menentukan jadwal mata pelajaran pada SMKN 4 Kendari”.

## **1.2. Rumusan Masalah**

Rumusan masalah dalam penelitian ini, yaitu :

1. Bagaimana merancang sistem informasi penjadwalan mata pelajaran pada SMK Negeri 4 Kendari
2. Bagaimana implementasi Metode *Graph Colouring* pada sistem informasi penjadwalan mata pelajaran dengan algoritma *Greedy*

### 1.3. Batasan Masalah

Batasan masalah dalam penelitian ini, yaitu :

1. Sistem informasi yang dijadikan simulasi hanya sebatas sistem penjadwalan mata pelajaran, bukan sistem pengolahan data akademik secara keseluruhan.
2. Sistem dibangun berbasis *Web*.
3. Sistem yang dibangun menggunakan metode *graph colouring* (pewarnaan graf) dengan algoritma *Greedy*.
4. Penjadwalan ini diluar dari peraturan pembelajaran dalam masa pandemi covid.

### 1.4. Tujuan Penelitian

Tujuan penelitian ini, yaitu :

1. Merancang sistem informasi untuk penjadwalan mata pelajaran di SMK Negeri 4 Kendari
2. Menerapkan metode *graph colouring* menggunakan algoritma *Greedy* pada sistem informasi penjadwalan mata pelajaran.

### 1.5. Manfaat Penelitian

Manfaat penelitian ini, yaitu :

1. Mengurangi kesalahan-kesalahan dalam pembuatan jadwal mata pelajaran yang sering terjadi, seperti kesamaan jadwal mengajar guru.
2. Pembuatan jadwal mata pelajaran yang dilakukan oleh wakil kepala sekolah bidang kurikulum SMK Negeri 4 Kendari dapat menghemat waktu dan tenaga.

### 1.6. Sistematika Penulisan

Penyusunan laporan penelitian ini terdiri dari beberapa bab dan masing masing bab tersebut berisi uraian singkat yang memperjelas selama mengadakan penelitian tugas akhir ini. Hal ini dimaksudkan agar pembahas lebih sistematis dan spesifik sesuai dengan topik permasalahan. Adapun sistematika penulisan laporan tugas akhir ini terdiri dari 6 bab yaitu :

## BAB I PENDAHULUAN

Bab ini menjelaskan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, sistematika penulisan dan tinjauan pustaka.

## **BAB II LANDASAN TEORI**

Pada bab ini berisi tentang pengertian dan teori yang berkaitan dengan isi dan judul penelitian tugas akhir.

## **BAB III METODOLOGI PENELITIAN**

Bab ini menyajikan Langkah-langkah pengumpulan data, prosedur pengembangan perangkat lunak dan perangkat keras yang dilakukan dalam penelitian dan waktu serta tempat pelaksanaan penelitian.

## **BAB IV ANALISIS DAN PERANCANGAN SISTEM**

Bab ini diuraikan analisis sistem yang akan dibuat dan kebutuhan sistem yang meliputi kebutuhan fungsional, kebutuhan non fungsional sistem. Rancangan sistem meliputi rancangan arsitektur sistem, rancangan proses, rancangan procedural, rancangan data, dan rancangan *user interface*

### **1.7. Tinjauan pustaka**

Terdapat banyak literatur yang ditemukan untuk penelitian mengenai penjadwalan menggunakan metode *graph colouring*. Salah satunya yaitu penelitian dengan judul "Analisis Penjadwalan Mata Pelajaran menggunakan Algoritma *Welch Powell*" oleh (Wicaksono & Kartono, 2020) menjelaskan Proses penyusunan jadwal dapat dilakukan dengan menggunakan Algoritma *Welch-Powell* yaitu teknik pada pewarnaan simpul pada graf. Dengan graf yang digunakan simpul sebagai mata pelajaran dan garis sebagai kelas. Garis yang menghubungkan tiap simpul merupakan mata pelajaran yang tidak dapat dilaksanakan pada jam dan kelas yang sama dan selanjutnya diimplementasikan menjadi tabel jadwal pelajaran dan diperoleh jadwal mata pelajaran yang tidak ada bentrok antar guru, mata pelajaran, dan jam mengajar.

Penelitian lainnya yang berkaitan dengan penelitian ini, yaitu oleh (Yusnita et al., 2019) "Penerapan metode pewarnaan graf untuk penjadwalan mata kuliah". Dalam penelitian ini Disimpulkan bahwa dengan menerapkan metode pewarnaan graf untuk penjadwalan matakuliah dapat mempermudah program studi dalam menyusun jadwal perkuliahan tiap semesternya, sehingga menghindari terjadinya tumpang tindih antara mata kuliah, dosen, ruangan dan waktu yang bersamaan. Hasil akhirnya sebelum digunakan, dilakukan pengujian *black box*, dengan

menguji apakah fungsi-fungsi dalam aplikasi tersebut berjalan dengan baik. Diharapkan kedepannya aplikasi dapat dikembangkan dengan melihat kapasitas ruangan yang tersedia dan tidak berfokus pada satu waktu.

Penelitian lain oleh (Rahadi, 2019) yaitu “penjadwalan mata kuliah menggunakan pewarnaan graf dengan algoritma *largest first*” membahas tentang perbedaan algoritma *Greedy* dan *welch powel*, dimana kedua algoritma ini sama sama menggunakan skema *largest first* yaitu simpul berderajat tinggi diwarnai terlebih dahulu. Dan dimana kedua algoritma tersebut baik algoritma *greed* maupun *welch powell* memberikan hasil akhir yang identik, tetapi berbeda urutan kerjanya.

Penelitian selanjutnya dari (Juniar, 2015) dengan judul “ Penjadwalan matakuliah menggunakan algoritma *Greedy*” maka dapat disimpulkan bahwa dari penelitian ini adalah sistem pendukung keputusan penjadwalan mata kuliah dengan model pemrograman *heuristic* menggunakan algoritma *Greedy* ini menghasilkan solusi jadwal mata kuliah yang dapat menjadi pertimbangan pengambilan keputusan (*decision maker*) dalam membangun jadwal kuliah.

Penelitian lain yaitu oleh (Handayani et al., 2016) dengan judul “ Penerapan Algoritma *Welch Powell* dengan Pewarnaan *graph* pada penjadwalan mata pelajaran SMA” Berdasarkan hasil dari analisis dan implementasi penjadwalan mata pelajaran di SMA Negeri 8 Bandung, maka dapat disimpulkan bahwa algoritma *welch powell* ini dapat menyusun jadwal mata pelajaran dengan proses pengolahan data dari form kesediaan mengajar yang diisi oleh guru dan dapat mengampu mata pelajaran ke sebanyak bilangan kromatik yang dihasilkan yang direpresentasikan dalam kelas.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Penjadwalan Mata Pelajaran**

Penjadwalan adalah alokasi dari sumber daya terhadap waktu untuk menghasilkan sebuah kumpulan pekerjaan (Masyoyo, 2014) . (Lesmana, 2017) menyatakan bahwa penjadwalan adalah alat ukur terhadap suatu perencanaan. Penjadwalan memberikan informasi mengenai kapan dan berapa lama suatu kegiatan dijalankan. Penjadwalan mata pelajaran adalah proses pengorganisasian untuk mengalokasikan waktu kapan dan dimana suatu kegiatan pembelajaran akan dilaksanakan (Handayani et al., 2016) . Ada hal-hal yang menjadi pertimbangan dalam menyusun jadwal mata pelajaran, misalnya dalam proses penjadwalan sering terdapat jadwal bentrok satu sama lain atau guru mengajar di tempat yang berbeda dengan waktu yang sama.

Penjadwalan juga adalah sebuah proses dari penempatan tugas untuk sebuah kumpulan data jadwal. Penjadwalan mata pelajaran sekolah adalah suatu informasi yang menentukan mata pelajaran, hari dan jam dimulainya, ruang yang akan digunakannya dimana akan membentuk suatu jadwal yang terstruktur dan tertata rapi sehingga kegiatan belajar mengajar dapat terselenggara dengan baik

##### **2.1.1. Penyusunan Jadwal Pelajaran**

Jadwal pelajaran berguna untuk mengetahui apa yang akan diajarkan pada suatu waktu dalam suatu kelas. Dari sudut guru, jadwal pelajaran merupakan pedoman di kelas mana dan berapa lama dia harus mengajar pada waktu itu. Jadwal dapat dibedakan menjadi 2 yaitu :

1. Jadwal umum. Memuat pengaturan pemberian mata pelajaran pada seluruh kelas dan menunjukkan pembagian waktu mengajar bagi seluruh guru di sekolah tersebut.
2. Jadwal khusus. Merupakan kegiatan pemberian mata pelajaran yang hanya berlaku bagi suatu kelas tertentu dan hari tertentu. Jadwal pelajaran dibuat untuk

3. satu minggu, sedangkan pada minggu-minggu berikutnya jadwal sama persis dengan lainnya.

### **2.1.2 Komponen-Komponen Dalam Penyusunan Jadwal Pelajaran**

kegiatan penyusunan jadwal pelajaran akan terasa mudah dan cepat apabila mengikuti Langkah-langkah sistematis penyusunan jadwal pelajaran. Komponen-komponen penyusunan jadwal pelajaran dalam manajemen kurikulum, yaitu :

1. Antara mata pelajaran satu dengan yang lainnya harus ada selingan agar tidak menjemukan.
2. Pelajaran jangan terlalu lama, 30 menit untuk satu jam pelajaran kelas I dan II SD, 40 menit untuk kelas III-VI dan 45 menit untuk sekolah lanjutan.
3. Masing-masing mata pelajaran dicarikan waktu yang sesuai, biasanya pelajaran yang banyak membutuhkan daya pikir dijadwalkan pada jam permulaan.
4. Harus disediakan waktu istirahat agar murid tidak terlalu lelah.
5. Jangan sampai kegiatan di suatu kelas dapat mengganggu kegiatan di kelas sebelahnya.
6. Untuk sekolah-sekolah yang kecil (murid sedikit) dapat diberikan kegiatan sama dalam waktu yang sama, misalnya olahraga, kesenian dan sebagainya.

### **2.1.3 Penyusunan Program Mengajar**

Berdasarkan peraturan pemerintah NO.74 tahun 2008, dalam rangka penyusunan program ini yang harus dilihat adalah isi kurikulum sekolah bersangkutan. Yang dimaksud isi di sini terutama adalah jumlah atau semacam pokok bahasan dan sub pokok bahasan dari setiap bidang studi masing-masing. Program mengajar harus disesuaikan dengan jangka waktu yang tertentu yang berlaku di sekolah, mungkin satu semester atau mungkin pada satu catur wulan. Menurut kurikulum 1975, untuk sekolah dasar digunakan catur wulan, sedangkan Sekolah lanjutan menggunakan sistem semester. Secara garis besar, penyusunan program mengajar ini dapat ditempuh dengan Langkah-langkah sebagai berikut :

1. Menghitung jumlah pokok bahasan yang tercantum dalam jangka waktu tertentu (semester atau catur wulan).
2. Menghitung jumlah sub pokok bahasan untuk masing-masing pokok bahasan tersebut.
3. Menghitung jumlah jam pelajaran (alokasi waktu) yang tersedia menurut kurikulum yang berlaku.
4. Menghitung jumlah hari belajar efektif pada semester atau catur wulan yang bersangkutan dengan melihat kalender akademi (kalender sekolah).
5. Membagi (menempatkan) pokok-pokok bahasan dan pokok sub bahasan disesuaikan dengan waktu yang tersedia.
6. Menyusun (merumuskan) Tujuan Instruksional khusus (TIK) untuk setiap pokok bahasan dan sub pokok bahasan.
7. Menentukan buku sumber-sumber bahasan yang diperlukan.
8. Menyusun satuan pelajaran, penyusunan satuan pelajaran ini dalam kurikulum 1975 disebutkan harus menggunakan pendekatan prosedur pengembangan sistem instruksional.

Penyusunan satuan pelajaran hendaklah dilakukan secara lengkap untuk jangka waktu tertentu (satu semester atau satu catur wulan). Diharapkan dengan perencanaan yang matang, guru akan memperoleh hasil Pendidikan yang lebih baik. Guru akan selalu dapat menyelesaikan tugasnya sesuai dengan program yang digariskan oleh kurikulum yang berlaku.

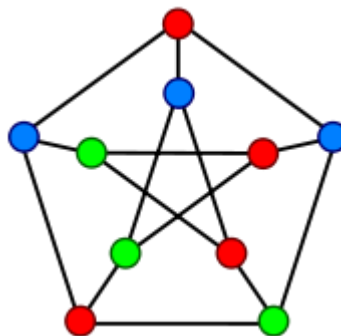
## **2.2 Metode *Graph colouring***

Graf adalah subjek yang dipelajari didalam teori graf. Teori Graf mulai dikenal saat matematikawan kebangsaan Swiss bernama Leonhard Euler, yang berhasil mengungkapkan Misteri Jembatan Koningsberg tahun 1736 (Arni, 2018). Menurut (Bender & Williamson, 2010), graf adalah pasangan terurut  $G = (V, E)$  dengan  $V$  adalah himpunan Verteks dan  $E$  adalah himpunan garis. Dalam teori graf, terdapat istilah pewarnaan graf (*graph coloring*) yaitu adalah pemberian warna terhadap simpul atau garis sedemikian sehingga dua simpul atau garis yang berdampingan mempunyai warna yang berbeda (Syakur, 2004). Menurut (Ramlah,

Hasmawati, & Lawi, 2013) terdapat tiga macam pewarnaan graf yaitu pewarnaan simpul (*vertex colouring*), pewarnaan sisi (*edge colouring*) dan pewarnaan wilayah (*face colouring*).

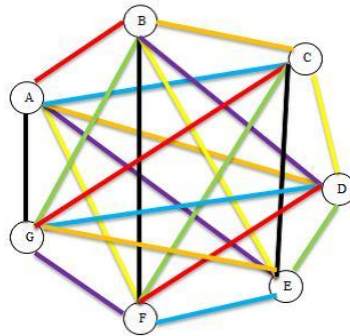
Pewarnaan yang digunakan pada graf penjadwalan ini adalah pewarnaan simpul. Menurut (Niarma, Bambang Pramono, 2018) , pewarnaan simpul pada graf adalah sebuah proses pemberian warna-warna ke simpul-simpul suatu graf sedemikian sehingga tidak ada dua buah simpul yang bertetangga memiliki warna yang sama. Menurut (Puspasari, 2015), pewarnaan simpul graf (*edge coloring*) pertama kali muncul sebagai masalah pewarnaan dalam peta yang setiap daerah yang saling berbatasan pada peta dibuat berlainan sehingga mudah untuk dibedakan. Salah satu metode untuk menyelesaikan permasalahan pewarnaan simpul graf adalah Algoritma *Greedy*

Ada tiga jenis pewarnaan pada graf, yakni pewarnaan simpul (*vertex coloring*), pewarnaan sisi (*edge coloring*), atau pewarnaan daerah (*region coloring*). Dalam tulisan ini akan dipaparkan pewarnaan simpul saja, mengingat graf yang dikenai pewarnaan sisi maupun daerah dapat direkayasa secara dual sehingga dapat dipandang sebagai suatu pewarnaan simpul. Sebagai contoh pewarnaan pada peta internasional merupakan pewarnaan daerah (*region*), yaitu setiap negara mendapatkan suatu warna yang berbeda dengan negara tetangganya. Negara-negara tersebut dapat dimodelkan menjadi simpul-simpul dan batas wilayah setiap 2 negara dapat dipandang sebagai sisi. Dengan demikian pewarnaan pada peta dapat dianggap sebagai pewarnaan simpul.

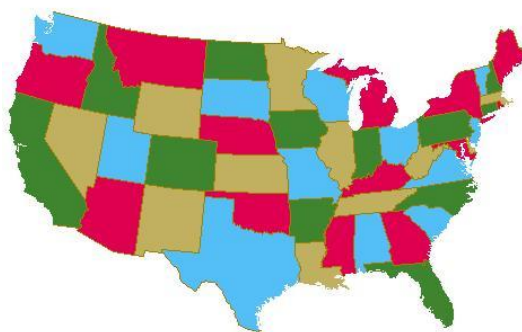


**Gambar 2. 1** pewarnaan graf pada titik (simpul)





**Gambar 2. 2 Pewarnaan graf pada sisi**



**Gambar 2. 3 Pewarnaan graf bidang (daerah)**

Suatu graf  $G$  dikatakan terwarnai bila setiap simpulnya diasosiasikan dengan suatu warna. Pewarnaan tersebut dikatakan teratur (*properly colored*) bila tidak ada dua simpul yang adjacent mendapatkan warna yang sama. Dengan kata lain setiap dua simpul yang *adjacent* haruslah diberi warna yang berbeda.

### 2.2.1. Jenis *graph*

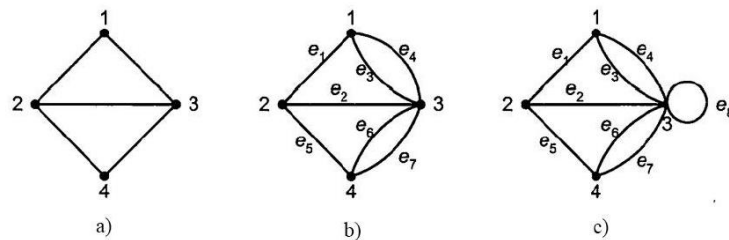
Berdasarkan ada atau tidaknya gelang atau sisi ganda pada suatu graf maka secara umum graf dapat digolongkan menjadi graf sederhana dan graf tidak sederhana (Munir, 2010).

#### a. Graf Sederhana (*Simple Graph*)

Graf sederhana adalah graf yang tidak mengandung gelang (*loop*) dan/atau sisi ganda. Gelang adalah sisi yang awal dan akhirnya adalah simpul yang sama, sedangkan sisi ganda adalah dua buah sisi yang menghubungkan dua simpul yang sama. Pada graf ini, sisi  $(u, v)$  sama saja dengan sisi  $(v, u)$ .

b. Graf Tidak Sederhana (*Unsimple Graph*)

Graf tidak sederhana adalah graf yang mengandung gelang (loop) dan/atau mengandung sisi ganda. Graf ini terbagi menjadi dua jenis yaitu graf ganda (multigraph) dan graf semu (*pseudograph*). Graf ganda adalah graf yang memiliki sisi ganda, sedangkan graf semu adalah graf yang memiliki gelang.



**Gambar 2. 4 Contoh a) graf sederhana, b) graf ganda, dan c) graf semu**

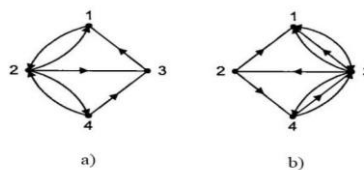
Sedangkan, berdasarkan orientasi arah pada sisi maka secara umum graf dapat dibedakan menjadi graf berarah dan graf tidak berarah.

a. Graf Berarah (*Directed Graph*)

Graf berarah adalah graf yang setiap sisinya memiliki orientasi arah. Sisi yang berarah disebut busur (arc). Pada graf ini, sisi  $(u, v)$  berbeda dengan sisi  $(v, u)$  karena suatu busur terdiri dari dua komponen yaitu simpul asal dan simpul terminal. Graf berarah dibagi menjadi dua jenis yaitu graf berarah biasa yang tidak memiliki sisi ganda, dan graf ganda berarah yang memiliki gelang atau sisi ganda.

b. Graf Tidak Berarah (*Undirected Graph*)

Graf tidak berarah adalah graf yang sisinya tidak memiliki orientasi arah. Pada graf ini, urutan pasangan simpul yang dihubungkan oleh sisi tidak diperhatikan sehingga sisi  $(u, v)$  sama saja dengan sisi  $(v, u)$ .



**Gambar 2. 5 Contoh a) graf berarah dan b) graf ganda berarah**

### 2.2.2. Terminologi *graph*

Terdapat beberapa istilah yang sering digunakan dalam teori graf, antara lain ketetanggaan dua simpul, bersisian, derajat, maupun lintasan (Munir, 2010).

a. Bertetangga (*Adjacent*)

Dua simpul dikatakan bertetangga apabila kedua simpul tersebut terhubung secara langsung oleh suatu sisi. Dengan kata lain, simpul  $u$  bertetangga dengan  $v$  jika  $(u, v)$  adalah sebuah sisi pada graf  $G$ .

b. Bersisian (*Incidency*)

Suatu sisi dikatakan bersisian dengan dua simpul apabila menghubungkan kedua simpul tersebut secara langsung.

c. Simpul Terpencil (*Isolated Vertex*)

Suatu simpul disebut simpul terpencil apabila tidak ditemukan sisi yang bersisian dengannya.

d. Graf Kosong (*Null Graph*)

Graf kosong adalah graf yang himpunan sisinya merupakan himpunan kosong. Dinotasikan dengan  $N_n$  dengan  $n$  adalah banyaknya simpul

e. Derajat (*Degree*)

Pada graf tidak berarah, derajat suatu simpul adalah banyaknya sisi yang bersisian dengan simpul tersebut. Sedangkan dalam graf berarah, derajat suatu simpul  $(v)$  merupakan jumlah dari banyaknya busur yang masuk ke simpul ( $\text{din}(v)$ ) dengan banyaknya busur yang keluar dari simpul ( $\text{dout}(v)$ ).

f. Lintasan (*Path*)

Lintasan dari sebuah graf adalah barisan satu atau lebih sisi yang dilalui dari suatu simpul awal ke simpul tujuan. Panjang suatu lintasan adalah banyaknya sisi yang dilewati dari simpul awal ke simpul tujuan

g. Sirkuit (*Circuit*)

Sirkuit atau siklus (*cycle*) adalah suatu lintasan yang berawal dan berakhir pada simpul yang sama.

#### h. Terhubung (*Connected*)

Graf tidak berarah  $G$  disebut graf terhubung jika untuk setiap pasang simpul  $u$  dan  $v$  di dalam himpunan  $V$  terdapat lintasan dari  $u$  ke  $v$ . Sedangkan pada graf berarah, suatu graf dikatakan terhubung jika graf tidak berarahnya juga terhubung.

#### i. Graf Bagian (Subgraph) dan Komplemen Subgraph

Jika  $G = (V, E)$  adalah sebuah graf maka  $G_1 = (V_1, E_1)$  adalah graf bagian atau subgraph dari  $G$  jika  $V_1 \subseteq V$  dan  $E_1 \subseteq E$ . Bagian yang bukan merupakan subgraph disebut dengan komplemen.

### 2.3. Algoritma *Greedy*

Algoritma *Greedy* merupakan metode yang paling populer untuk memecahkan persoalan mencari solusi optimum (optimasi). Terdapat dua jenis persoalan optimasi: maksimasi dan minimasi. Algoritma *Greedy* membentuk solusi Langkah perlangkah. Pada setiap langkah, dibuat pilihan optimum local untuk kemudian dicari solusi optimum global dari pilihan yang telah diambil. Secara garis besar, algoritma *Greedy* terbagi menjadi dua langkah besar :

1. Pilihan yang diambil merupakan pilihan yang terbaik yang dapat diperoleh pada saat itu tanpa menganalisis dampak yang akan terjadi dari pemilihan solusi terbaik saat ini.
2. Berharap bahwa dengan memilih pilihan yang terbaik saat itu (solusi optimum local) solusi terbaik global dapat dicapai (solusi optimum global).

Elemen-elemen pada algoritma *Greedy* :

1. Himpunan kandidat,  $C$
2. Himpunan solusi,  $S$
3. Fungsi seleksi (*Selection function*)
4. Fungsi kelayakan (*feasible*)
5. Fungsi objektif.

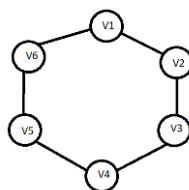
Algoritma *Greedy* melibatkan pencarian sebuah himpunan bagian  $S$  dari himpunan kandidat  $C$ .  $S$  harus memenuhi beberapa kriteria yang ditentukan, yaitu menyatakan suatu solusi dan  $S$  dioptimasi oleh fungsi objektif. Dengan cara yang sama, algoritma tersebut digunakan menyelesaikan persoalan pewarnaan graf.

Pewarnaan *Greedy* merupakan pewarnaan simpul pada graf yang dibentuk berdasarkan algoritma *Greedy*. Algoritma *Greedy* mempertimbangkan simpul simpul pada graf sebagai sebuah urutan dan mengisi setiap simpul dengan warna pertama yang tersedia.

Algoritma *Greedy* untuk pewarnaan graf :

1. Pada graf  $G$ , dari derajat setiap simpul pada  $G$
2. Inisialisasi himpunan simpul takberwarna dengan semua simpul pada graf  $G$  dengan urutan derajat tak menaik. Elemen pertama pada himpunan adalah simpul dengan derajat tertinggi.
3. Inisialisasi sebuah himpunan solusi dengan himpunan kosong.
4. Melakukan pemilihan simpul yang akan diisi warnanya dengan fungsi seleksi simpul pada himpunan tak berwarna.
5. Menghapus simpul yang terpilih dari daftar simpul tak berwarna dan menggeset warna simpul terpilih dengan warna yang sekarang aktif.
6. Masukkan simpul dalam himpunan solusi
7. Memeriksa seluruh simpul yang ada di himpunan simpul tak berwarna :
  - a. Simpul yang layak akan dimasukkan dalam himpunan solusi (tidak bertetangga dengan simpul yang telah ada di himpunan solusi)
  - b. Simpul yang telah dinyatakan layak dihapuskan dari himpunan simpul tak berwarna
  - c. Simpul yang telah dinyatakan layak diberi warna dengan warna yang sedang aktif.
8. Naikkan indeks warna aktif
9. Jika simpul sudah diwarnai, proses berakhir. Jika belum semua simpul terwarnai, Kembali ke Langkah 3.

Contoh kasus :



**Gambar 2. 6 Contoh kasus**

**Tabel 2. 1 Pengurutan tidak menurun untuk simpul berdasarkan jumlah derajat**

No	Simpul	Jumlah derajat
1	V1	2
2	V2	2
3	V3	2
4	V4	2
5	V5	2
6	V6	2

Tahap 0 :

Tak berwarna : {v1,v2,v3,v4,v5,v6}

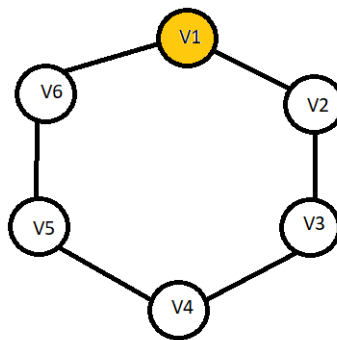
Tahap 1 :

Warna saat ini : 0

Himpunan solusi tahap 1 : { }

$V = v1$

Tak berwarna : { v2,v3,v4,v5,v6}

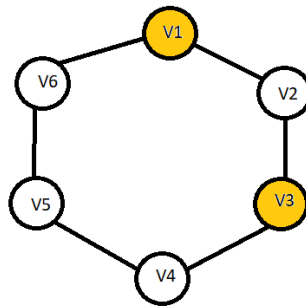
**Gambar 2. 7 Tahap 1a**

Himpunan solusi tahap 1 : {v1}

$Warna(v1) = 0$

$V = v3$

Tak berwarna {v2,v4,v5,v6}



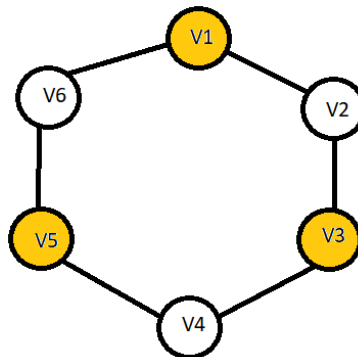
**Gambar 2. 8 Tahap 1b**

Himpunan solusi tahap 1:  $\{v1, v3\}$

$Warna(v3)=0$

$V = v5$

Tak berwarna  $\{v2, v4, v6\}$



**Gambar 2. 9 Tahap 1c**

Himpunan solusi tahap 1  $\{v1, v3, v5\}$

$Warna(v5) = 0$

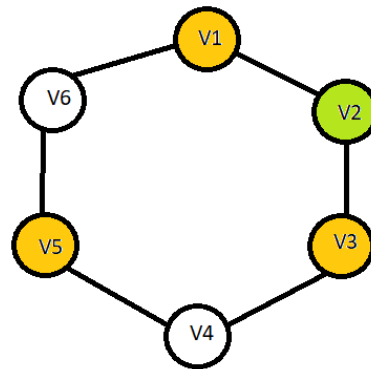
Tahap 2 :

Warna saat ini = 1

Himpunan solusi tahap 2 :  $\{\}$

$V = v2$

Tak berwarna :  $\{v4, v6\}$



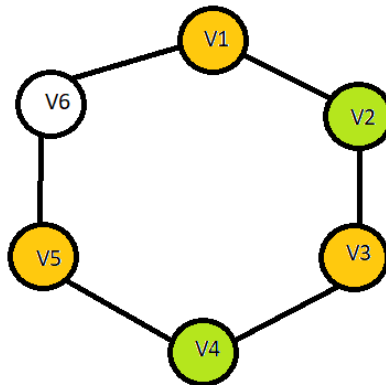
**Gambar 2. 10 Tahap 2a**

Himpunan solusi tahap 2 :  $\{v2\}$

Warna( $v2$ ) : 1

$V = v4$

Tak berwarna :  $\{v6\}$



**Gambar 2. 11 Tahap 2c**

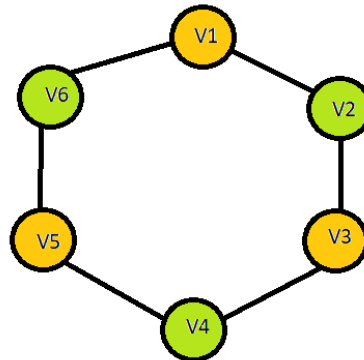
Himpunan solusi tahap 2 :  $\{v2, v4\}$

Warna ( $v4$ ) : 1

$V = v6$

Tak berwarna  $\{ \}$





**Gambar 2. 12 Tahap 2c**

#### **2.4. *Hypertext preprocessor (PHP)***

*Hypertext Preprocessor (PHP)* adalah bahasa *scripting web HTML-embedded*. Ini berarti kode PHP dapat disisipkan ke dalam HTML halaman Web. Ketika sebuah halaman PHP diakses, kode PHP dibaca atau "diurai" oleh *server*. *Output* dari fungsi PHP pada halaman biasanya dikembalikan sebagai kode HTML, yang dapat dibaca oleh *browser*. Karena kode PHP diubah menjadi HTML sebelum halaman dibuka, pengguna tidak dapat melihat kode PHP pada halaman. Ini membuat halaman PHP cukup aman untuk mengakses *database* dan informasi aman lainnya. (Ferdianto, 2013).

Banyak sintaks PHP yang hasil adaptasi dari bahasa lain seperti bahasa C, Java dan Perl. Namun, PHP memiliki sejumlah fitur unik dan fungsi tertentu juga. Tujuan dari bahasa pemrograman PHP adalah untuk memungkinkan pengembangan *web* untuk menulis halaman yang dihasilkan secara dinamis dengan cepat dan mudah. PHP juga bagus untuk menciptakan situs *Web database-driven*. Jika Anda ingin mempelajari lebih lanjut tentang PHP, situs resminya yaitu PHP.net. (Ferdianto, 2013).

Beberapa kelebihan PHP dari bahasa pemrograman *web*, antara 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 *mulaiapache*, IIS, *Lighttpd*, hingga Xitami dengan konfigurasi yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis-milis dan *developer* yang siap membantu dalam pengembangan.
4. Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang paling mudah karena memiliki referensi yang banyak.
5. PHP adalah bahasa *open source* yang dapat digunakan diberbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem. (Ferdianto, 2013).

## 2.5. *Framework Codeigniter*

*Framework* merupakan perangkat lunak yang mulai menjadi pilihan untuk membuat suatu aplikasi. Kemudahan-kemudahan yang diberikan menarik orang-orang untuk menggunakannya. Hal ini tidak terlepas dari tingkat efektifitas dan efisiensinya yang lebih baik dalam proses pengembangan suatu perangkat lunak. *Framework* adalah suatu kumpulan kode berupa pustaka (*library*) dan alat (*tool*) yang dipadukan sedemikian rupa menjadi satu kerangka kerja (*framework*) guna memudahkan dan mempercepat proses pengembangan aplikasi *web*.

Proses pengembangan *web* dapat dilakukan dengan beragam bahasa pemrograman seperti *PHP*, *Python*, *Ruby*, *Perl*, *C++*, *JAVA* dan sebagainya. Saat ini, banyak bermuculan *framework* web yang dirancang untuk bahasa pemrograman tersebut. Salah satunya adalah *CodeIgniter*. *CodeIgniter* adalah kerangka kerja PHP yang kuat dengan tapak yang sangat kecil, dibangun untuk pengembang yang membutuhkan *toolkit* sederhana dan elegan untuk membuat aplikasi *web* berfitur lengkap.

Beberapa keunggulan yang kemudian ditawarkan oleh *CodeIgniter* yaitu sebagai berikut:

1. *CodeIgniter* adalah *framework* yang bersifat *free* dan *open source*.
2. *CodeIgniter* memiliki ukuran yang kecil dibandingkan dengan *framework* lain. Setelah proses instalasi, *framework CodeIgniter* hanya berukuran kurang lebih 2 MB. Dokumentasi *CodeIgniter* memiliki ukuran sekitar 6 MB.
3. Aplikasi yang dibuat menggunakan *CodeIgniter* bisa berjalan cepat.
4. *CodeIgniter* menggunakan pola desain *Model-View-Controller* (MVC) sehingga satu file tidak terlalu berisi banyak kode. Hal ini menjadikan kode lebih mudah dibaca, dipahami, dan dipelihara dikemudian hari.
5. *CodeIgniter* dapat diperluas sesuai dengan kebutuhan.
6. *CodeIgniter* terdokumentasi dengan baik. Informasi tentang pustaka kelas dan fungsi yang disediakan oleh *CodeIgniter* dapat diperoleh melalui dokumentasi yang disertakan di dalam paket distribusinya.

## **2.6. Database Management System**

*Database Management System* (DBMS) merupakan perangkat lunak untuk mengendalikan pembuatan, pemeliharaan, pengolahan, dan penggunaan data yang berskala besar. Penggunaan DBMS saat ini merupakan hal yang sangat penting dalam segala aspek, baik itu dalam skala yang besar atau kecil. Sebagai contoh media sosial facebook menggunakan DBMS untuk menyimpan data-data pengguna facebook yang sangat banyak kedalam DBMS MySQL. Beberapa DBMS yang digunakan adalah MySQL dan MariaDB. Berdasarkan survey yang dilakukan, MySQL dan MariaDB merupakan DBMS yang banyak digunakan sebagai contoh survey yang terdapat pada db-engines.com *DB-Engines Ranking* menempatkan MySQL pada posisi ke-2 sedangkan MariaDB pada posisi ke-20 namun pada survey yang terdapat di serverwatch.com *Top 10 Enterprise Database Sistem Of 2016*, MariaDB menempati posisi ke-6 dan MySQL menempati posisi ke-7 (Wulan Nafesa Septine, S.T., 2019).

## 2.7. MySQL

MySQL merupakan sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS *Multiithread* dan *multi user*. MySQL sebenarnya merupakan turunan dari salah satu konsep utama dalam *database* untuk pemilihan atau seleksi dan pemasukan data yang memungkinkan pengoperasian data dikerjakan secara mudah dan otomatis. MySQL diciptakan oleh Michael "Monty" Widenius pada tahun 1979, seorang *programmer* komputer asal Swedia yang mengembangkan sebuah sistem *database* sederhana yang dinamakan UNIREG yang menggunakan koneksi *low-level ISAM database engine* dengan *indexing* 1.

### 1. Kelebihan Dan Kekurangan MYSQL

Adapun kelebihan dari MySQL dalam penggunaanya dalam *database* yakni sebagai berikut:

- a) Gratis sehingga MySQL dapat dengan mudah untuk mendapatkannya.
- b) MySQL stabil dalam pengoprasiannya.
- c) MySQL mempunyai sistem keamanan yang cukup baik.
- d) Sangat mendukung transaksi dan mempunyai banyak dukungan dari komunitas.
- e) Sangat *fleksibel* dengan berbagai macam *program* Perkembangan dari MySQL sangat cepat.

Adapun kelemahan dari MySQL dalam penggunaanya dalam *database* yakni sebagai berikut:

- a) Sulit untuk diaplikasikan pada intansi atau perusahaan dengan *database* yang besar.
- b) *Support* yang kurang.
- c) Tidak populer untuk aplikasi *game* dan *mobile*.

## 2.8. XAMPP

XAMPP adalah sebuah paket perangkat lunak (*software*) komputer yang sistem penamaannya diambil dari akronim kata Apache, MySQL (dulu) / MariaDB (sekarang), PHP, dan *Perl*. Sementara imbuhan huruf “X” yang terdapat pada awal kata berasal dari istilah *cross platform* sebagai simbol bahwa aplikasi ini bisa dijalankan di empat sistem operasi berbeda, seperti OS Linux, OS Windows, Mac OS, dan juga Solaris. Sejarah mencatat, *software* XAMPP pertama kali dikembangkan oleh tim proyek bernama *Apache Friends* dan sampai saat ini sudah masuk dalam rilis versi 7.3.9 yang bisa didapatkan secara gratis dengan label GNU (*General Public License*).

Jika dijabarkan secara gamblang, masing-masing huruf yang ada di dalam nama XAMPP memiliki arti sebagai berikut ini:

- a. X artinya *Cross Platform*  
Merupakan kode penanda untuk *software cross platform* atau yang bisa berjalan di banyak sistem operasi.
- b. A artinya *Apache*  
Apache adalah aplikasi web server yang bersifat gratis dan bisa dikembangkan oleh banyak orang (*open source*).
- c. M artinya MySQL / MariaDB  
MySQL atau MariaDB merupakan aplikasi *database* server yang dikembangkan oleh orang yang sama. MySQL berperan dalam mengolah, mengedit, dan menghapus daftar melalui *database*.
- d. PHP  
Huruf “P” yang pertama dari akronim kata XAMPP adalah inisial untuk menunjukkan eksistensi bahasa pemrograman PHP. Bahasa pemrograman ini biasanya digunakan untuk membuat *website* dinamis, contohnya dalam *website* berbasis CMS WordPress.
- e. P artinya Perl  
Sementara itu, untuk huruf P selanjutnya merupakan singkatan dari bahasa pemrograman Perl yang kerap digunakan untuk memenuhi berbagai macam

kebutuhan. Perl ini bisa berjalan di dalam banyak sistem operasi sehingga sangat fleksibel dan banyak digunakan.

Program aplikasi XAMPP berfungsi 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*. XAMPP bekerja secara *offline* layaknya *web hosting* biasa namun tidak bisa diakses oleh banyak orang. Maka dari itu, XAMPP biasanya banyak digunakan oleh para Mahasiswa maupun pelajar untuk melihat hasil desain *website* sebelum akhirnya dibuat *online* menggunakan *web hosting* yang biasa dijual dipasaran.

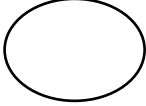


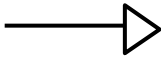

## 2.9. Metode Perancangan Unified Modeling Language (UML)

*Unified Modeling Language* (UML) adalah bahasa permodelan untuk merancang sistem berbasis *object-oriented*. UML digunakan untuk pengembangan sebuah sistem yang dapat menyampaikan alur kerja sistem dan menjelaskan tugas setiap *user* dalam sebuah sistem. UML memiliki tiga kategori *diagram*, yaitu *structure diagrams*, *behavior diagram*, dan *interaction diagram*. Ada beberapa jenis *diagram* dalam UML:

### 1. *Use Case Diagram*

*Use case diagram* atau *diagram use case* merupakan permodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih *actor* dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi- fungsi itu (Kiswanto, 2016).

Tabel 2. 2 Simbol-Simbol *Use Case*



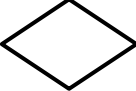



No	Simbol	Nama	Keterangan
1.		<i>Use Case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau <i>actor</i> ; biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i> .
2.		Aktor / <i>actor</i>	Orang proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri.
3.		Asosiasi / <i>association</i>	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan <i>actor</i> .
4.	<< extend >>	Ekstensi / <i>extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>usecase</i> yang Ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan.
5.		Generalisasi / <i>generalization</i>	Hubungan generalisasi dan spesialisasi (umum - khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6.	<<Include>> 	<i>Include</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.

Sumber: (Kiswanto, 2016).

## 2. Activity Diagram

*Activity diagram* atau *diagram* aktivitas menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau *menu* yang ada pada perangkat lunak (Kiswanto, 2016). Yang perlu diperhatikan adalah bahwa *diagram* aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor.

**Tabel 2. 3 Simbol-Simbol Activity Diagram**

No	Simbol	Nama	Keterangan
1.		<i>Start / Status awal</i>	Status awal aktivitas sistem, sebuah <i>diagram</i> aktivitas memiliki sebuah status awal
2.		Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
3.		Percabangan / <i>desicion</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
4.		Status akhir	Status akhir yang dilakukan sistem, sebuah <i>diagram</i> aktivitas memiliki sebuah status akhir.
5.		<i>swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.
6.		Penggabungan / <i>join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.

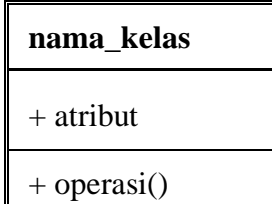



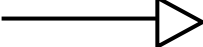
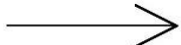
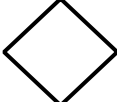
Sumber: (Kiswanto, 2016).



### 3. Class Diagram

*Class diagram* atau *diagram kelas* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. *Diagram kelas* dibuat agar *programmer* dapat membuat kelas-kelas didalam *program* perangkat lunak sesuai dengan perancangan *diagram kelas* (Kiswanto, 2016). Ada beberapa simbol-simbol yang terdapat dalam *diagram kelas*, yaitu:

**Tabel 2. 4 Simbol-Simbol Class Diagram**

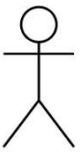


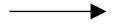


No	Simbol	Nama	Keterangan
1.		Kelas	Kelas pada struktur sistem
2.		Antarmuka / <i>interface</i>	Semua dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
3.		Asosiasi / <i>association</i>	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
4.		Asosiasi berarah / <i>directed asosiasi</i>	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
5.		Generalisasi	Relasi antarmuka dengan makna generalisasi spesialisasi (umum khusus)
6.		Kebergantungan / <i>dependency</i>	Relasi antarmuka dengan makna kebergantungan antarkelas
7.		Agregasi / <i>aggregation</i>	Relasi antarkelas dengan makna semua bagian ( <i>whole-part</i> )



Sumber: (Kiswanto, 2016).

#### 4. *Sequence Diagram*

*Sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan terima antar objek. Oleh karena itu untuk menggambar *diagram sequence* maka harus diketahui objek-objek yang dimiliki kelas yang diinstansiasi menjadi objek itu. Menggambar *diagram sequence* juga dibutuhkan untuk melihat *scenario* yang ada pada *use case* (Kiswanto, 2016).

**Tabel 2. 5 Simbol-Simbol *Sequence Diagram***

No	Simbol	Nama	Keterangan
1.		Aktor	Orang proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri.
2.		Garis hidup / <i>lifeline</i>	Menyatakan kehidupan suatu objek
3.	Nama objek : nama kelas	Objek	Menyatakan objek yang berinteraksi pesan
4.		Waktu aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.
5.	<<create>> 	Pesan tipe <i>create</i>	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
6.		Pesan tipe <i>call</i>	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri
7.		Pesan tipe <i>send</i>	Menyatakan bahwa suatu objek mengirim data / masukkan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim

No	Simbol	Nama	Keterangan
8.		Pesan tipe <i>return</i>	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
9.		Pesan tipe <i>destroy</i>	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada <i>destroy</i>

Sumber: (Kiswanto, 2016)

## 2.10. Rational Unified Process (RUP)

*Rational Unified Process* (RUP) merupakan suatu metode rekayasa perangkat lunak yang dikembangkan dengan mengumpulkan berbagai *best practises* yang terdapat dalam industri pengembangan perangkat lunak. Ciri utama metode ini adalah menggunakan *use case driven* dan pendekatan iteratif untuk siklus pengembangan perangkat lunak. Gambar dibawah menunjukkan secara keseluruhan arsitektur yang dimiliki RUP (Rahardjo, 2018).

Dalam metode ini, terdapat empat tahap pengembangan perangkat lunak yaitu:

### 1. *Inception*

Pada tahap ini pengembang mendefinisikan batasan kegiatan, melakukan analisis kebutuhan user, dan melakukan perancangan awal perangkat lunak (perancangan arsitektural dan *use case*). Pada akhir fase ini, prototipe perangkat lunak versi *Alpha* harus sudah dirilis.

### 2. *Elaboration*

Pada tahap ini dilakukan perancangan perangkat lunak mulai dari menspesifikasikan fitur perangkat lunak hingga perilsan prototipe versi *Betha* dari perangkat lunak.

### 3. *Construction*

Pengimplementasian rancangan perangkat lunak yang telah dibuat dilakukan pada tahap ini. Pada akhir tahap ini, perangkat lunak versi akhir yang sudah disetujui administrator dirilis beserta dokumentasi perangkat lunak.

### 4. *Transition*

Instalasi *deployment* dan sosialisasi perangkat lunak dilakukan pada tahap ini. Pada tahapan ini dilakukan tahap evaluasi dari sistem yang telah direncanakan, dianalisa dan dibangun. Dari tahap ini bisa dilihat kekurangan dari sistem yang telah dibangun.

## BAB III

### METODOLOGI PENELITIAN

#### 3.1 Waktu dan Tempat Penelitian

##### 3.3.1 Waktu

Waktu pelaksanaan penelitian tugas akhir dilaksanakan mulai dari Bulan Februari 2021 sampai dengan Juni 2021. Rincian kegiatan dapat dilihat pada tabel 3.1 berikut :

**Tabel 3. 1 *Gannt Chart* Waktu Penelitian**

No	Uraian	Waktu (2021)																			
		Maret			April				Mei				Juni				Juli				
		2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	
1	Inception																				
2	Elaboration																				
3	Construction																				
4	Transition																				

##### 3.3.2 Tempat

Adapun tempat penelitian tugas akhir yaitu akan dilakukan di Sekolah Menengah Kejuruan Negeri 4 Kendari, Sulawesi Tenggara.

#### 3.2 Metode Pengumpulan Data

Dalam penelitian ini metode pengumpulan data yang digunakan pada perancangan aplikasi tersebut adalah wawancara dan studi literatur.

##### 1. Wawancara

Wawancara adalah proses memperoleh keterangan dengan cara tanya jawab sambil bertatap muka antara si penanya atau pewawancara dengan si penjawab atau responden. Di sini peneliti melakukan wawancara dengan bagian kurikulum sekolah yang bertujuan untuk mengetahui informasi tentang data kelas, guru, mata pelajaran, waktu, ruang dan kesedian guru.

## 2. Studi literatur.

Metode pengumpulan data ini dengan cara mencari dan membaca referensi baik berasal dari literatur dokumen-dokumen serta arsip maupun sumber bacaan lain, seperti skripsi atau penelitian sebelumnya yang dapat membantu dalam hal penyusunan penelitian tugas akhir ini yang berkaitan dengan penjadwalan.

### 3.3 Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan dalam sistem ini adalah metode *Rational Unified Process* (RUP). Dalam metode ini, terdapat empat tahap pengembangan perangkat lunak, yaitu:

#### 1. Permulaan (*Inception*)

Pada *fase* ini dilakukan proses pengidentifikasian sistem, dilakukan dengan analisis kebutuhan akan aplikasi, melakukan kajian terhadap penelitian yang berkaitan dengan metode *Graph colouring*

#### 2. Perluasan/perencanaan (*Elaboration*)

Setelah menentukan ruang lingkup penelitian, tahap ini akan dilakukan perancangan dan analisis sistem menggunakan *flowchart*, yaitu *flowchart* sistem. Pada perancangan ini, digunakan juga *UML (Unified Modelling Language)* yang meliputi *use case diagram*, *activity diagram*, *class diagram* dan *sequence diagram* dari perangkat lunak yang akan dibuat, serta desain antarmuka aplikasi yang akan dibuat

#### 3. Konstruksi (*Construction*)

Proses yang dilakukan pada tahap ini yaitu membangun aplikasi dengan perancangan yang telah dilakukan sebelumnya, mulai dari tampilan *interface* sampai implementasi rancangan UML. Proses yang juga dilakukan pada tahap ini yaitu penerapan *coding* metode *graph colouring* sebagai metode penjadwalan.

#### 4. Transisi (*Transition*)

Pada tahap ini difokuskan untuk melakukan pengujian terhadap sistem penjadwalan. Memperbaiki masalah yang muncul saat pembuatan dan setelah pengujian aplikasi.

### 3.4 Analisis Sistem

Analisis sistem merupakan suatu tahapan yang bertujuan untuk mengetahui dan mengamati semua yang terlibat dalam suatu sistem. Pembahasan yang ada pada analisis sistem ini yaitu analisis kebutuhan fungsional meliputi perancangan sistem menggunakan bahasa pemodelan *UML (Unified Modeling Language)*, perancangan tampilan *interface* serta analisis kebutuhan fungsional dan non fungsional yang meliputi kebutuhan perangkat keras dan perangkat lunak yang akan digunakan.

#### 3.4.1 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional adalah segala bentuk data yang dibutuhkan oleh sistem agar sistem dapat berjalan sesuai dengan prosedur yang dibangun melalui perancangan sistem. Adapun kebutuhan fungsional dari sistem yang akan dibangun yaitu Perancangan *diagram* sistem menggunakan bahasa pemodelan *UML (Unified Modeling Language)* yang meliputi *flowchart* sistem, *flowchart* metode, *use case diagram*, *activity diagram*, *class diagram*, serta *sequence diagram* dan Desain *Interface* sistem.

#### 3.4.2 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional adalah sebuah langkah dimana pembangun aplikasi menganalisis sumber daya kebutuhan untuk membangun aplikasi yang akan dibangun. Analisis kebutuhan nonfungsional yang dilakukan dibagi dalam dua tahap, yaitu analisis kebutuhan perangkat keras dan analisis kebutuhan perangkat lunak. Kebutuhan perangkat keras yaitu kebutuhan perangkat atau komponen yang dibutuhkan pada sistem dan perangkat lunak yaitu kebutuhan perangkat lunak untuk membantu agar komponen perangkat keras dapat berfungsi dan dapat dijalankan pada sistem.

##### 1. Kebutuhan perangkat keras

Untuk menerapkan rancangan yang telah dijelaskan sebelumnya, dibutuhkan beberapa perangkat keras sebagai sarana untuk mengimplementasikan aplikasi yang dibangun. Berikut ini spesifikasi perangkat keras yang dibutuhkan.

**Tabel 3. 2 Spesifikasi perangkat keras**

No	Nama Perangkat	Spesifikasi
1.	<i>PC</i>	HP 14s-dk1005AU
2.	<i>Processor</i>	AMD Ryzen 3
3.	<i>RAM</i>	8 GB
4.	<i>Harddisk</i>	256 GB SSD
5.	<i>Monitor</i>	Monitor 14 Inch

## 2. Kebutuhan perangkat lunak

Perangkat lunak yang digunakan pada pembangunan sistem penjadwalan mata pelajaran. Adapun rincian kebutuhan perangkat lunak dapat dilihat pada Tabel 3.3 berikut ini.

**Tabel 3. 3 Spesifikasi perangkat lunak**

No.	Nama Perangkat	Fungsi	Spesifikasi
1.	<i>Windows</i>	<i>Operating System</i>	<i>Windows 10 Home</i>
2.	<i>Xampp</i>	<i>Universal development environment</i>	<i>Xampp v3.2.2</i>
3.	<i>Mysql</i>	<i>Database Management</i>	<i>Mysql 5.7</i>
4.	<i>PHP</i>	<i>Web development</i>	<i>PHP 7.3</i>
5.	<i>Visual Studio Code</i>	<i>Code Editor</i>	<i>VS Code v1.38</i>
6.	<i>Bootstrap</i>	<i>Framework CSS</i>	<i>Version 1.41</i>
7.	<i>CodeIgniter</i>	<i>Framework PHP</i>	<i>Version 3.1.11</i>

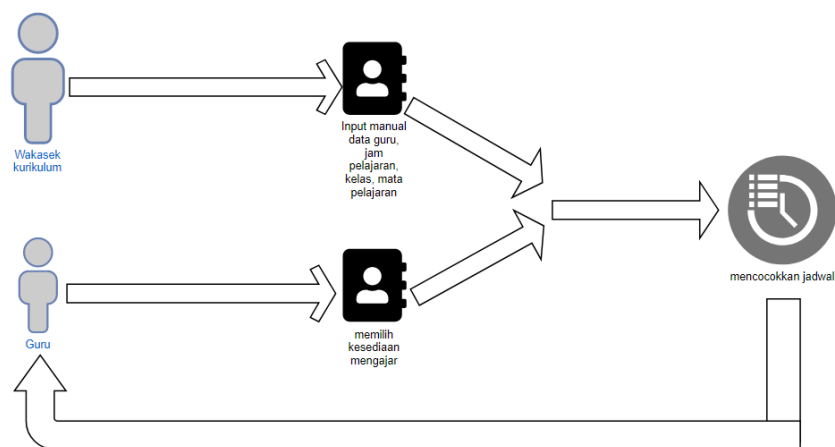
## 3.5 Analisis Perancangan Sistem

Perancangan sistem yang akan dibangun terdiri atas perancangan UML serta perancangan *user interface*.



### 3.5.1 Gambaran Umum Sistem yang Sedang Berjalan

Proses penjadwalan mata pelajaran pad SMKN 4 Kendari masih menggunakan sistem manual, yaitu Microsoft office excel sebagai aplikasi untuk mengolah data dalam pembuatan jadwal mata pelajaran. Pelajaran diatur kurang lebih lima hari dikarenakan inputan data dilakukan oleh satu pegawai, yaitu wakil kepala sekolah bidang kurikulum. Selain itu waktu yang terlalu lama, permasalahan yang ada dalam pengambilan keputusan adalah banyaknya kelas yang tidak sebanding dengan jumlah guru.



**Gambar 3. 1 Sistem yang sedang berjalan**

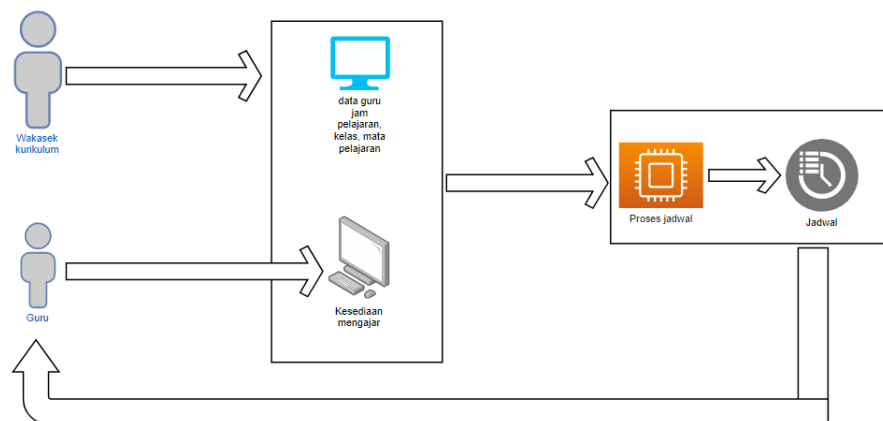
Adapun sistem yang saat ini sedang berjalan di SMKN 4 Kendari adalah sebagai berikut :

1. Wakil kepala sekolah bidang kurikulum mengisi data guru, jam pelajaran, kelas, dan mata pelajaran. Baik siswa yang baru masuk ataupun naik kelas
2. Kemudian tiap guru menentukan kesediaan mengajarnya dan menyetorkannya kepada WK kurikulum.
3. Setelah itu WK kurikulum Menyusun jadwal pelajaran sebanyak 3 rangkap, rangkap pertama diberikan kepada siswa , rangkap kedua diberikan kepada guru, dan rangkap ketiga sebagai laporan.
4. Setelah jadwal tersusun WK kurikulum membuat laporan sebanyak 2 rangkap, rangkap pertama di arsipkan dan rangkap kedua diberikan kepada kepala sekolah.

### 3.5.2 Gambaran Umum Sistem yang Diusulkan

Berdasarkan permasalahan pada sistem lama, maka peneliti membangun sistem baru untuk mempermudah pembuatan jadwal mata pelajaran pada SMK 4 Kendari. Sistem yang diusulkan berbasis website menggunakan Bahasa pemrograman *PHP* dan *framework CodeIgniter* (CI). Adapun masukan untuk aplikasi yang dibangun adalah data kelas, data guru, data pelajaran, data tahun ajaran, data waktu, data tahun ajaran.

Data tersebut akan diproses menggunakan algoritma *Greedy* dengan bantuan pewarnaan graf dalam pembuatan jadwal mata pelajaran.



**Gambar 3. 2 Sistem yang diusulkan**

Pada gambar 3.2 menampilkan sistem yang diusulkan Adapun penjelasan lebih lanjut adalah sebagai berikut :

1. Wakil kepala sekolah menginputkan data guru, data jam pelajaran, data kelas, data mata pelajaran ke dalam sistem
2. Kemudian guru memilih kesediaan mengajar yang tersedia disistem,
3. Lalu sistem akan memproses jadwal dengan melihat ketersediaan mengajar guru dan di cocokkan dengan mata pelajaran dan kelas yang tersedia.
4. Setelah jadwal di proses, kemudian jadwal akan di tampilkan disistem.

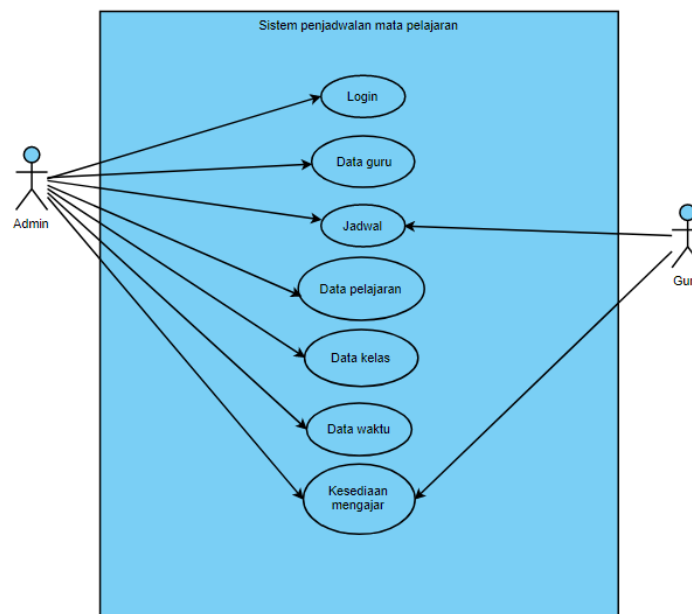
### 3.5.3 *Unified Modeling Language* (UML)

Aplikasi dibangun dengan menggunakan *Unified Modeling Language* (UML). UML merupakan bahasa visual untuk pemodelan dan komunikasi

mengenai sebuah sistem dengan menggunakan *diagram* yang terdiri dari *Use Case Diagram*, *Activity Diagram*, *Class Diagram* dan *Sequence Diagram*.

### 1. *Use Case Diagram*

*Use Case Diagram* adalah sebuah *diagram* yang dapat merepresentasikan interaksi yang terjadi antara *user* dengan sistem. *Use Case Diagram* ini mendeskripsikan siapa saja yang menggunakan sistem dan bagaimana cara mereka berinteraksi dengan sistem. *Use Case Diagram* dari sistem yang akan dibangun adalah sebagai berikut.



**Gambar 3. 3 Usecase Diagram Sistem**

Adapun penjelasan terkait *actor* dan sistem pada *usecase diagram* diatas adalah sebagai berikut.

**Tabel 3. 4 Keterangan *Usecase* Sistem**

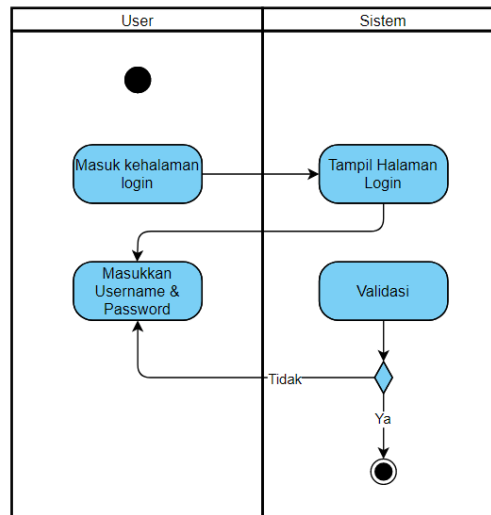
Aktor	Sistem
Admin memilih <i>login</i>	Sistem akan memvalidasi kecocokan <i>username</i> dan <i>password</i> jika cocok maka akan masuk ke halaman admin
Admin & guru memilih menu data guru	Sistem akan menampilkan data guru yang telah diinputkan dan admin juga dapat melakukan Tindakan terhadap data tersebut.
Admin dan guru memilih menu jadwal	Sistem akan menampilkan menu jadwal yang telah di atur oleh sistem yang ditampilkan kepada guru dan admin, guru dan admin bisa mencetak jadwal tersebut.
Admin memilih menu data pelajaran	Sistem menampilkan menu data pelajaran kemudian admin dapat melakukan penginputan data mata pelajaran yang ada.
Admin memilih menu data kelas	Sistem menampilkan menu data kelas, dan admin dapat mengisi data tiap kelas.
Admin memilih menu data waktu	Sistem menampilkan data waktu kemudian admin dapat melakukan pengisian data waktu mulai dari hari dan jam tiap mata pelajaran
Admin memilih menu user	Sistem akan menampilkan data user yang ada, dan admin dapat menambah dan menghapus data user (guru)
Guru memilih kesediaan mengajar	Guru dapat memilih kesediaan mengajar yang tersedia.

## 2. *Activity Diagram*

*Activity diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Berikut ini adalah *activity diagram* yang akan menggambarkan alir aktivitas sistem.

a. *Activity Diagram Login*

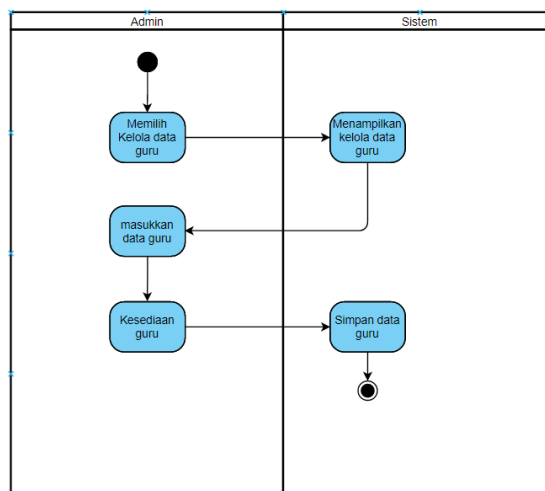
Pada gambar 3.4 merupakan *diagram* aktivitas yang menunjukkan aktivitas pengguna (admin, siswa, guru) Ketika memasukkan *username* dan *password* , lalu sistem akan melakukan validasi.



**Gambar 3. 4 Activity Diagram Login**

b. *Activity Diagram Data Guru (Admin)*

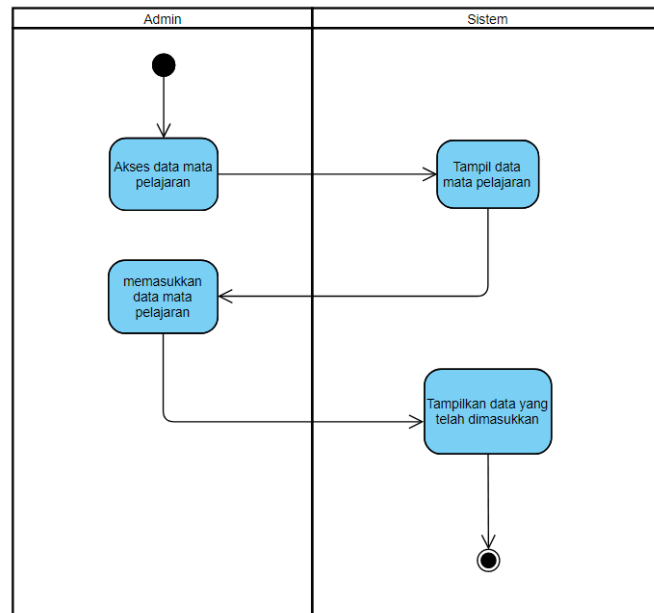
Gambar 3.5 menjelaskan tentang aktivitas yang terjadi pada halaman data guru. Admin mengakses halaman data guru dan kesediaan mengajar. Selanjutnya admin memasukkan data guru lalu sistem menampilkan data yang telah dimasukkan sebelumnya.



**Gambar 3. 5 Activity Diagram Data Guru**

c. *Activity Diagram* Data Mata Pelajaran (Admin)

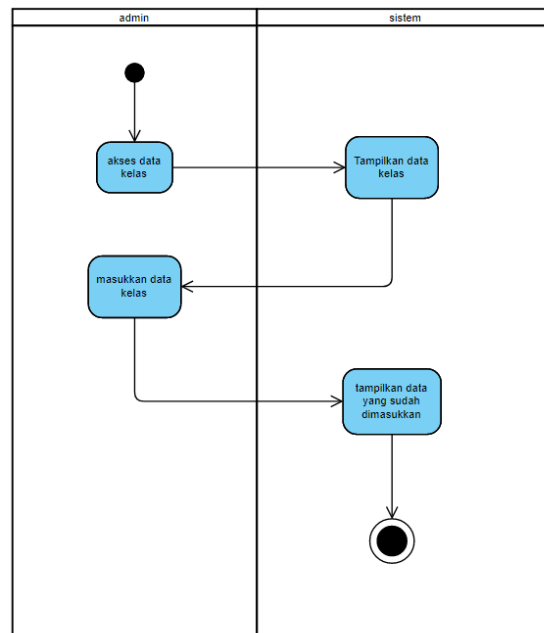
Gambar 3.6 menjelaskan tentang aktifitas yang terjadi pada halaman data mata pelajaran. Awalnya admin mengakses halaman data mata pelajaran kemudian sistem menampilkan data pelajaran. Selanjutnya admin memasukkan data dan sistem menampilkan data yang telah dimasukkan sebelumnya.



**Gambar 3. 6 Activity diagram mata pelajaran**

d. *Activity Diagram* Data Kelas

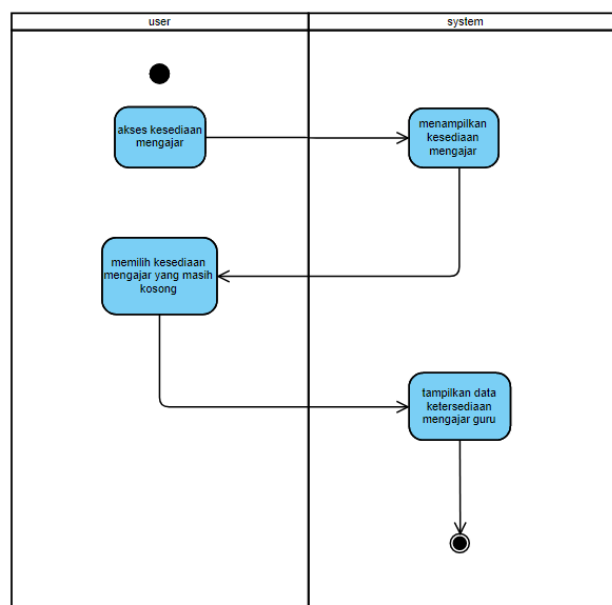
Gambar 3.7 menjelaskan tentang aktivitas yang terjadi pada halaman data kelas. Awalnya admin mengakses halaman data kelas kemudian sistem menampilkan data kelas. Selanjutnya admin memasukkan data dan sistem menampilkan data yang telah dimasukkan sebelumnya.



**Gambar 3. 7 Activity diagram data kelas**

e. *Activity Diagram Ketersediaan Mengajar*

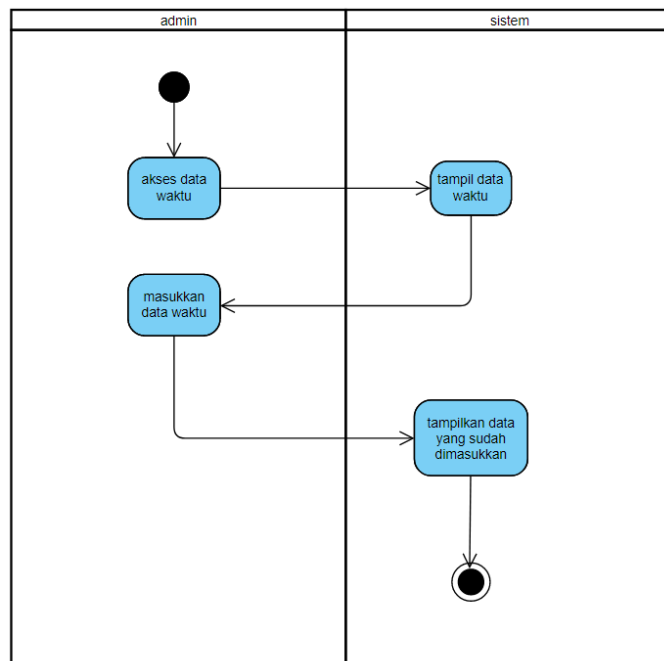
Gambar 3.8 menjelaskan tentang aktivitas yang terjadi pada halaman ketersediaan mengajar. Awalnya guru mengakses halaman ketersediaan mengajar kemudian sistem menampilkan data. Selanjutnya guru memilih data yang tersedia dan dan sistem menampilkan data yang telah dipilih oleh guru.



**Gambar 3. 8 Acitivity diagram ketersediaan mengajar**

f. *Activity Diagram Data Waktu*

Gambar 3.9 menjelaskan tentang aktivitas yang terjadi pada halaman data waktu. Awalnya admin mengakses halaman data waktu kemudian sistem menampilkan data. Selanjutnya admin memasukkan data dan sistem menampilkan data yang telah dimasukkan sebelumnya.

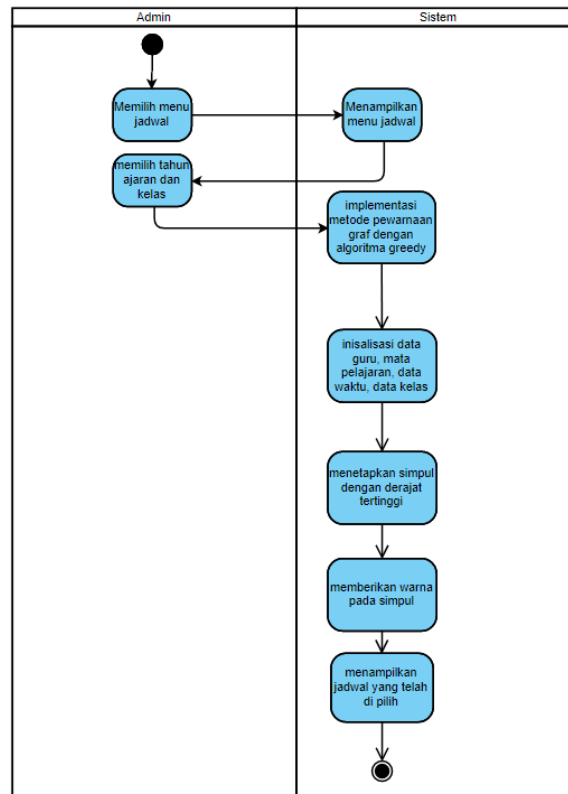


**Gambar 3. 9 Activity diagram data waktu**

g. *Activity Diagram Jadwal*

Gambar 3.10 menjelaskan tentang aktivitas yang terjadi pada halaman jadwal. Awalnya admin mengakses halaman jadwal dan kemudian sistem menampilkan pilihan tahun ajaran, kemudian masukkan data jadwal lalu sistem akan memproses jadwal menggunakan metode pewarnaan graf dengan algoritma *Greedy*.

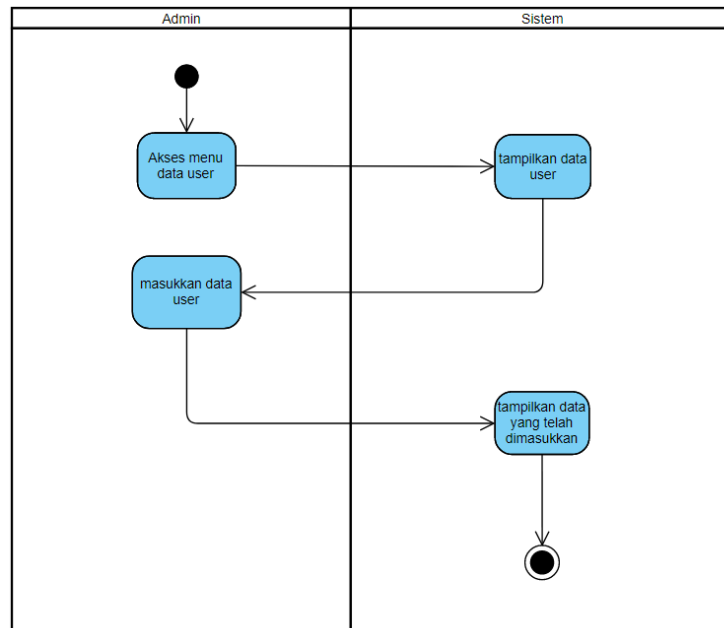




**Gambar 3. 10 Data jadwal.**

#### h. *Activity Diagram Data User*

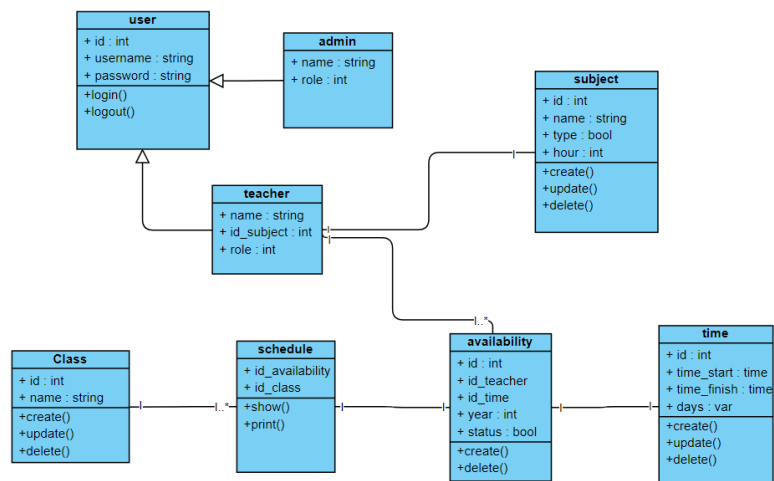
Gambar 3.11 menjelaskan tentang aktivitas yang terjadi pada menu data pengguna. Pengguna mengakses menu data pengguna kemudian sistem menampilkan data pengguna. Selanjutnya pengguna memasukkan data lalu sistem menampilkan data yang telah dimasukkan.



**Gambar 3. 11 Activity diagram data user**

### 3. Class Diagram

*Class diagram* menjelaskan hubungan antara class dalam sistem yang dapat dilihat pada gambar 3.12 berikut ini.

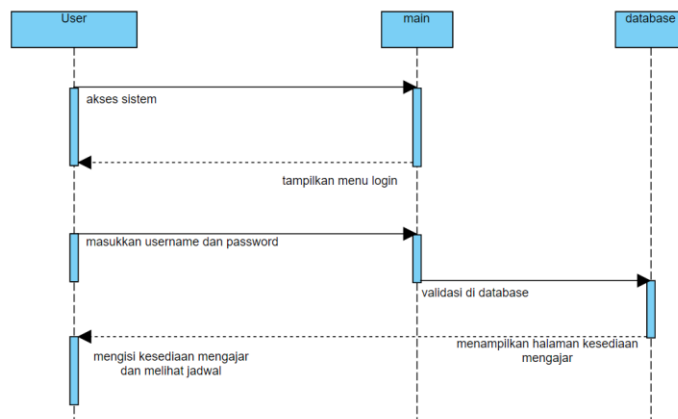


**Gambar 3. 12 Class Diagram**

#### 4. Sequence Diagram

##### a. Sequence Diagram User

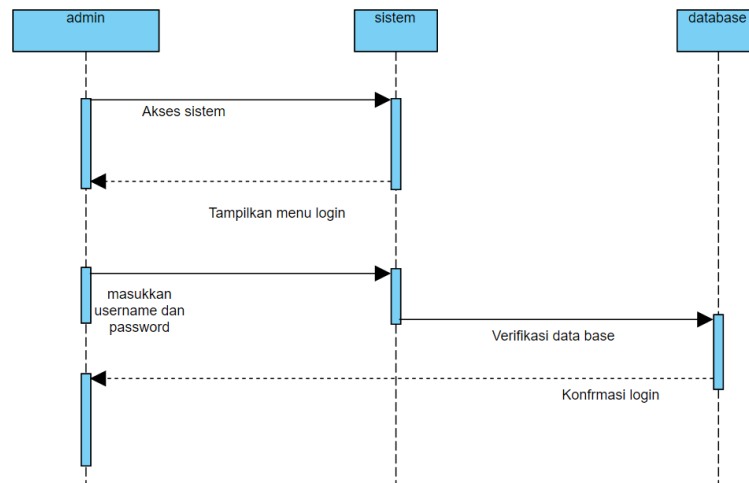
Setelah menganalisis sistem didapatkan rancangan sequence diagram user yang di tunjukkan oleh gambar 3.13 menjelaskan tentang pengguna mengakses menu *user*, lalu form user tampil kemudian pengguna dapat memilih kesediaan mengajar. Secara otomatis data akan tersimpan dan diperbaharui pada database, kemudian *user* akan meliat jadwal dan mencetak jadwal



**Gambar 3. 13 Sequence Diagram user**

##### b. Sequence Diagram Login (Admin)

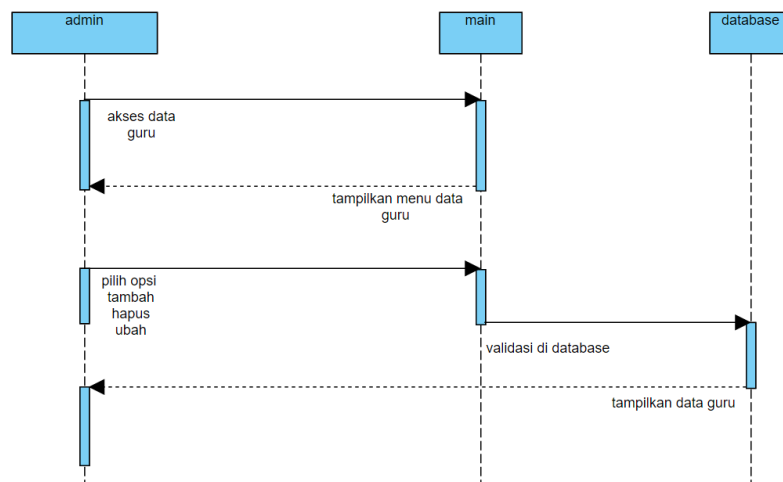
Pada gambar 3.14 menunjukkan alur dari *sequence diagram login*. Pada *sequence diagram login* dijelaskan admin memasukkan *username* dan *password* pada halaman *login* saat mulai penggunaan sistem. Selanjutnya, database akan memeriksa kebenaran data lalu memnkonfirmasiya , jika *username* dan *password* yang diisi benar, admin akan masuk ke beranda.



**Gambar 3. 14 Sequence diagram login**

c. *Sequence Diagram Data Guru (Admin)*

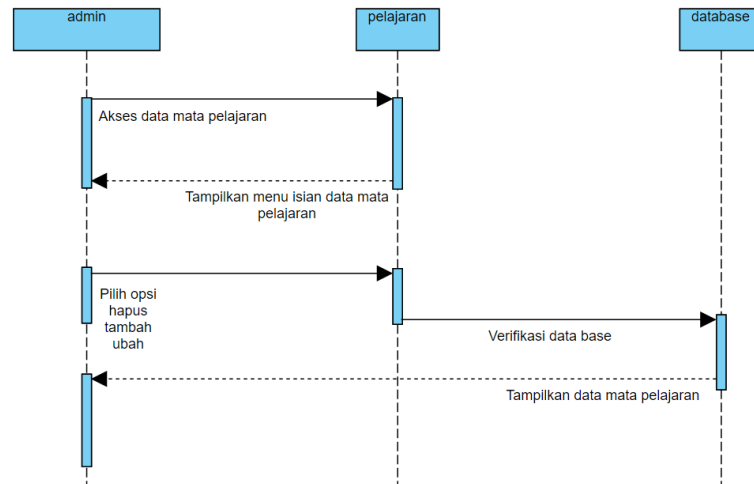
Diagram ini menjelaskan bagaimana admin mengakses halaman data guru. Pada halaman data guru, admin dapat menambah data , mengubah data, dan menghapus data. Setelah mengolah otomatis data akan diperbaharui dan tersimpan di *database*



**Gambar 3. 15 Sequence Diagram data Guru**

d. *Sequence Diagram Mata Pelajaran (Admin)*

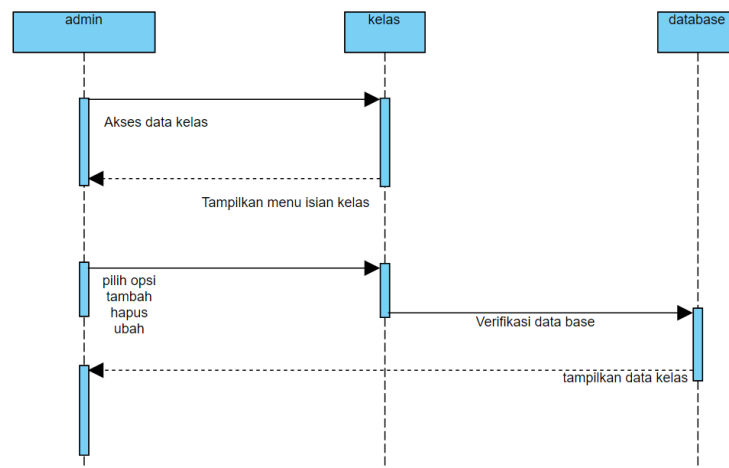
Diagram ini menjelaskan bagaimana admin mengakses halaman data pelajaran, pada halaman data pelajaran admin dapat memasukkan data, mengubah data, dan menghapus data atau melihat mata pelajaran yang telah diinputkan. Secara otomatis akan tersimpan di *database*



**Gambar 3. 16 Sequence diagram mata pelajaran**

e. *Sequence Diagram Data Kelas (Admin)*

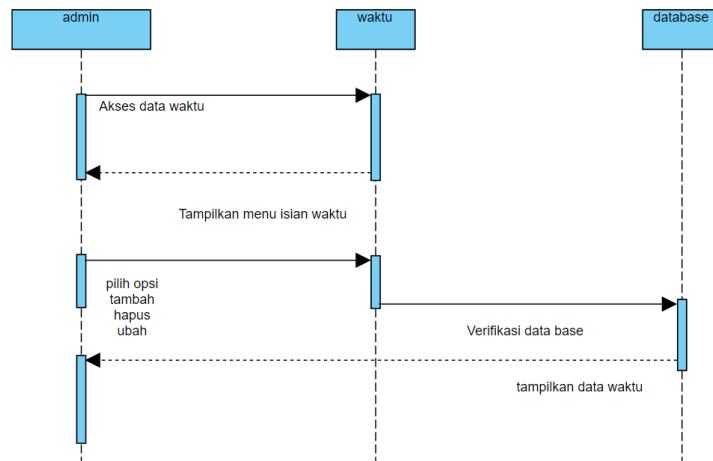
Diagram ini menjelaskan admin mengakses halaman data kelas. Pada halaman ini admin dapat menambah, mengubah dan menghapus data kelas, serta melihat data data kelas yang sudah ada, dan akan otomatis tersimpan dan diperbarui di *database*.



**Gambar 3. 17 Sequence Diagram data kelas**

f. *Sequence Diagram Data Waktu (Admin)*

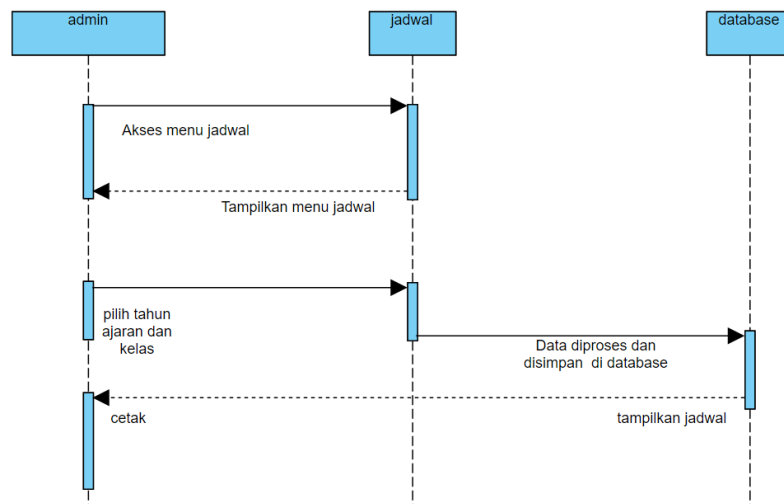
Diagram ini menjelaskan admin mengakses halaman data waktu. Pada halaman ini admin dapat menambahkan, mengubah dan menghapus data waktu sesuai ketentuan ketentuan yang ada, setelah mengolah data otomatis akan tersimpan di *database*.



**Gambar 3. 18 Sequence Diagram data waktu**

g. *Sequence Diagram Jadwal*

Diagram ini menjelaskan aktivitas yang terjadi pada halaman jadwal, awalnya admin mengakses halaman jadwal dan memilih tahun ajaran beserta kelas yang ada, kemudian sistem akan menampilkan jadwal. Admin juga dapat mencetak jadwal mata pelajaran.



**Gambar 3. 19 Sequence Diagram jadwal**

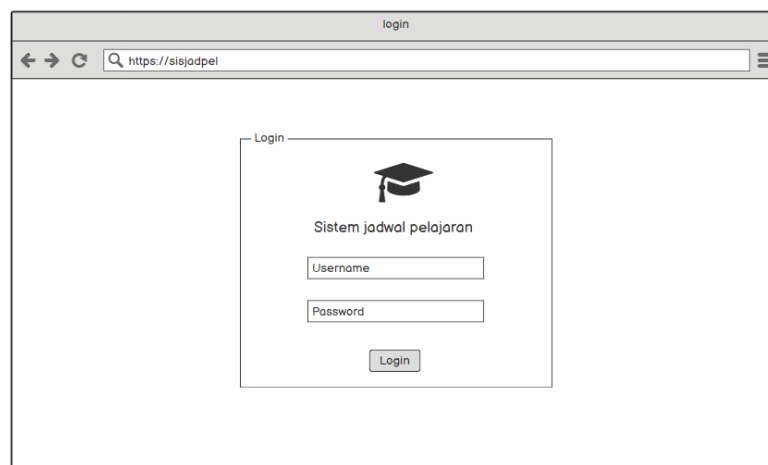
## 5. Perancangan antar muka (*design interface*)

Rancangan antarmuka pengguna atau *design user interface* merupakan penggambaran tampilan yang digunakan secara langsung oleh pengguna terhadap sistem. Rancangan *user interface* ini dibuat sederhana agar mudah dimengerti

pengguna dan tidak ada kerumitan dalam menjalankannya sehingga mencapai tujuan perangkat lunak yang *user friendly*.

a. Menu *Login*

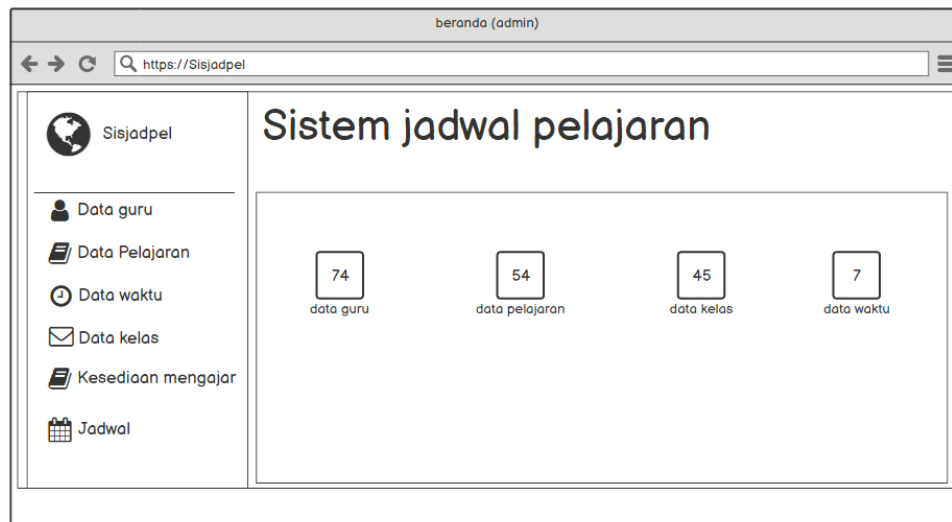
Halaman *login* pada *website* menampilkan form *login* yang terdiri dari *username* dan *password*. Dan setiap field harus diisi agar bisa *login*, dan setelah mengisi akan divalidasi dan datanya akan disamakan dengan data yang ada dalam *database*. Adapun bentuk tampilan halaman *login* dapat dilihat dari gambar berikut.



**Gambar 3. 20 Login**

b. Menu Beranda

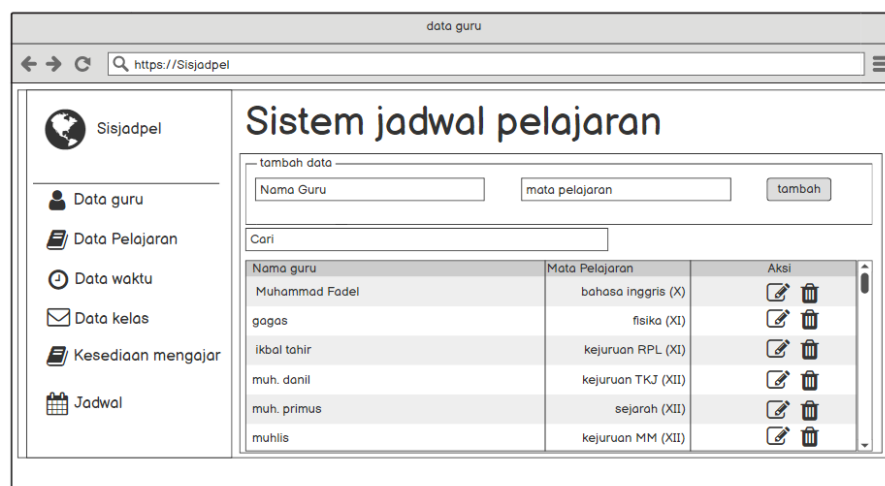
Setelah admin melakukan *login* otomatis akan masuk dimenu beranda, dimana menu beranda ini berisi logo dan menu menu pada sistem, serta jumlah data yang telah diinputkan pada tiap menu menu tersebut.



**Gambar 3. 21 Beranda**

c. Menu Data Guru

Pada menu ini menampilkan nama guru dan mata pelajaran yang diajarkan dan juga pilihan untuk menambah data guru, dan juga opsi mengubah dan menghapus data guru.

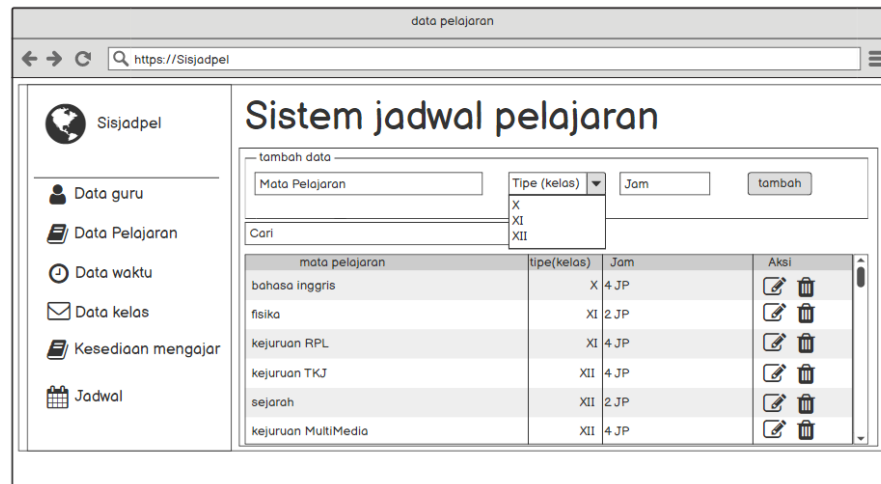


**Gambar 3. 22 Menu data guru**

d. Menu Data Pelajaran

Pada menu ini berisi mata pelajaran yang ada di sekolah ini, berserta mata pelajaran itu tersedia untuk kelas berapa, dan jumlah jam pada mata pelajaran tersebut.

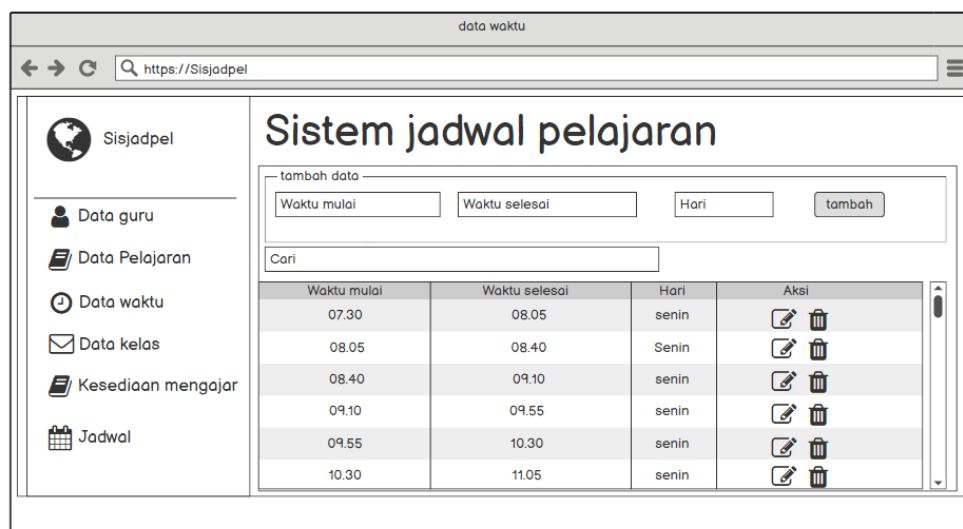




**Gambar 3. 23 Menu data pelajaran**

e. Menu Data Waktu

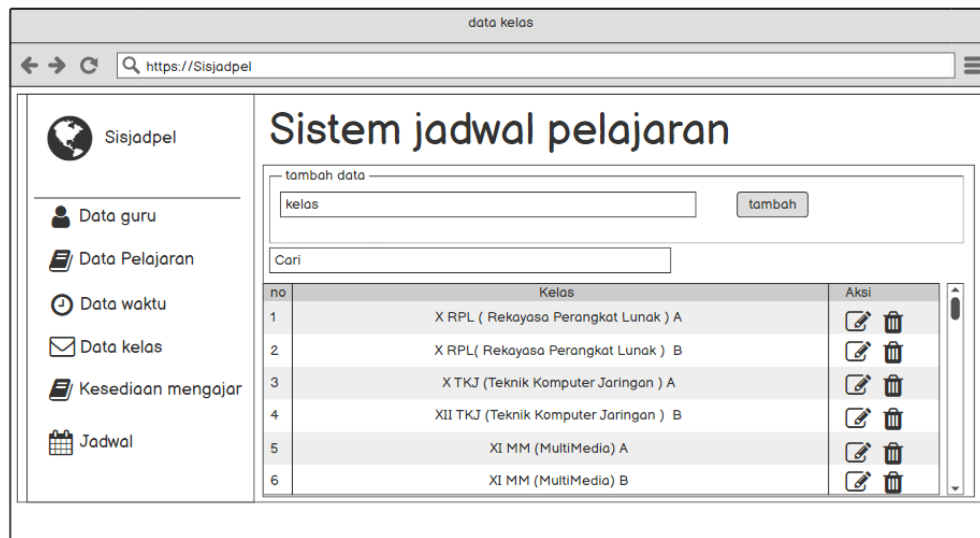
Menu data waktu berisi waktu mulai dan waktu berakhir pada sebuah mata pelajaran per jam pelajaran beserta hari harinya, dan admin dapat menambah data waktu tersebut, dan juga mengubah serta menghapus data waktu.



**Gambar 3. 24 Menu data waktu**

f. Menu Data Kelas

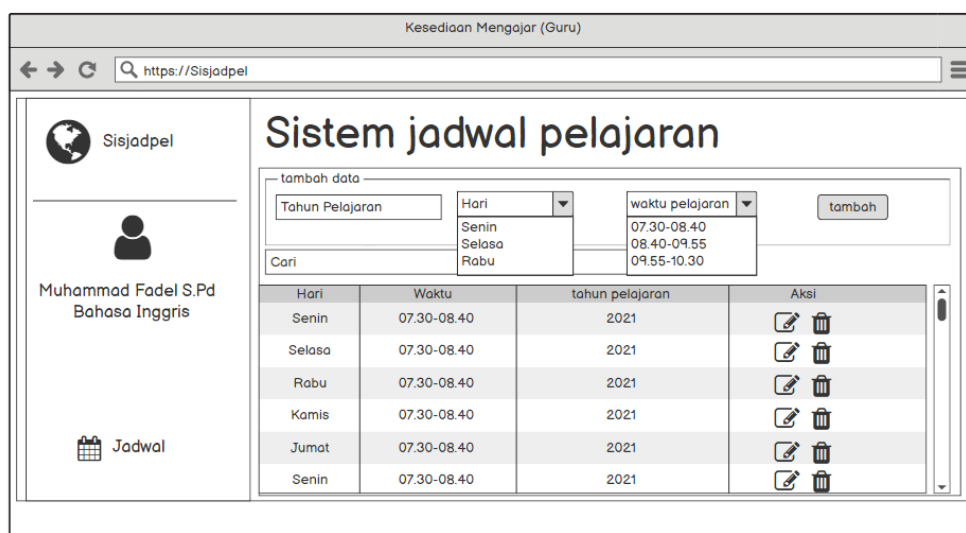
Pada menu ini, terdapat beberapa kelas dan jurusan yang ada pada SMKN 4 Kendari, dan admin dapat mengolah data tersebut.



**Gambar 3. 25 Menu data kelas**

g. Menu Ketersediaan Mengajar (Guru)

Pada menu ini, setelah user (guru) *login* menggunakan akunnya, maka guru akan masuk ke halaman ketersediaan mengajar, dimana guru dapat memilih ketersediaan mengajar yang masih kosong dan dengan ketentuan ketentuan yang ada.




**Gambar 3. 26 Menu ketersediaan mengajar**







h. Menu Jadwal

Pada menu ini admin dan guru dapat melihat jadwal yang telah diolah sebelumnya, dan pengguna dapat memilih tahun ajaran serta kelas yang ingin dilihat jadwal mata pelajarannya.

Jadwal

← → ↻

 Sisjadpel

 Data guru  
 Data Pelajaran  
 Data waktu  
 Data kelas  
 Ketersediaan mengajar  
 Jadwal

## Sistem jadwal pelajaran

Cari Kelas

Tahun ajaran  Tahun ajaran

Kelas

XI RPL A

Hari	Jam	mata pelajaran	Guru
Senin	07.30-08.40	Bahasa Indonesia	muh. danil S.pd
	08.40 - 09.55	sejarah	primus
	09.55 - 11.05	Kewirausahaan	Ice
	12.30 - 13.50	Fisika	Marlia S.Si
Selasa	07.30 - 08.40	Kejuruan RPL	Ikbal S.T
	10.30 - 11.05	Penjasokes	Elko

**Gambar 3. 27 Menu jadwal**

## DAFTAR PUSTAKA

- Arni, U. D. (2018). Logika dan Algoritma Backpropagation. *Garuda Cyber, tahun 1736*, 1–6. <https://garudacyber.co.id/artikel/635-logika-dan-algoritma->
- Bender, E. A., & Williamson, S. G. (2010). *Lists , Decisions and Graphs With an Introduction to Probability Unit DT : Decision Trees and Recursion Edward A . Bender*.
- Ferdianto, Y. (2013). *Pengertian PHP*.
- Handayani, D., Rosely, E., & Mayadewi, P. (2016). ALGORITMA WELCH POWELL STUDI KASUS : KELAS XMIPA SMA NEGERI 8 BANDUNG Pengumpulan & Pemilihan data Implementasi Model. *E-Proceeding of Applied Science*, 2(3), 933–935.
- Juniar, A. (2015). Penerapan Algoritma Greedy pada Penjadwalan Produksi Single-Stage dengan Parallel Machine di Industri Konveksi. *Jurnal Sifo Mikroskil*, 16(2), 175–184.
- Kiswanto, C. F. (2016). “*Sistem Informasi Akademik Sub-Sistem : Utility Dan Epsbed*.”
- Lesmana, N. I. (2017). Penjadwalan Produksi Untuk Meminimalkan Waktu Produksi Dengan Menggunakan Metode Branch And Bound. *Jurnal Teknik Industri*, 17(1), 42. <https://doi.org/10.22219/jtiumm.vol17.no1.42-50>
- Masyoyo, D. (2014). Analisa dan Implementasi Algoritma Priority Dispatching dalam Penjadwalan Pembagian Ruang Ujian. *Informasi Dan Teknologi Ilmiah*, II(1), 2014.
- Niarma, Bambang Pramono, L. T. (2018). Aplikasi penjadwalan menggunakan algoritma welch powell (studi kasus : sma muhammadiyah kendari). *SemanTIK*, 4(1), 1–6.
- Rahadi, A. P. (2019). Graf Dengan Algoritma Largest First. *Jurnal Padeagogik*, 2, 1–13.
- Rahardjo, A. (2018). *Pengertian RUP (Rational Unified Process)*.
- Wicaksono, P. S., & Kartono, K. (2020). Analisis Penjadwalan Mata Pelajaran

- Menggunakan Algoritma Welch-Powell. *Prismatika: Jurnal Pendidikan Dan Riset Matematika*, 3(1), 1–21. <https://doi.org/10.33503/prismatika.v3i1.1008>
- Wulan Nafesa Septine, S.T., M. (2019). *DATABASE MANAGEMENT SYSTEM (DBMS) MARIA DB*. 06 Nov 2019.
- Yusnita, A., Ekawati, H., & Wati, N. R. (2019). Penerapan Metode Pewarnaan Graf Untuk Penjadwalan Mata Kuliah. *Jurnal Media Informatika Budidarma*, 3(3), 153. <https://doi.org/10.30865/mib.v3i3.1102>