

**SISTEM PENJADWALAN SIDANG MENGGUNAKAN  
ALGORITMA GENETIKA PADA PENGADILAN TATA  
USAHA NEGARA**

**TUGAS AKHIR**

**Karya tulis sebagai salah satu syarat  
Untuk memperoleh gelar Sarjana dari  
Universitas Fajar Makassar**

**Oleh**

**DZULDAR**

**1620221024**



**PROGRAM STUDI TEKNIK ELEKTRO**

**FAKULTAS TEKNIK**

**UNIVERSITAS FAJAR**

**2023**

HALAMAN PENGESAHAN

SISTEM PENJADWALAN SIDANG MENGGUNAKAN ALGORITMA  
GENETIKA PADA PENGADILAN TATA USAHA NEGARA

Oleh  
**Dzuldar**  
1620221024

Menyetujui  
Tim Pembimbing  
Makassar, 10 April 2023

Pembimbing I



Andita Dani Achmad, S.T., M.T.  
NIDN. 0913029001

Pembimbing II



Febriansyah, S.Kom., M.T.  
NIDN. 0921029003

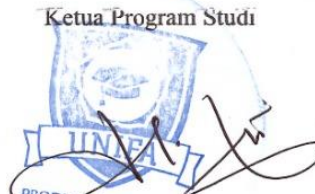
Mengetahui,

Dekan Fakultas Teknik



Prof. Dr. Ir. Erniati, S.T., M.T.  
NIDN. 0906107701

Ketua Program Studi



Dr. Safaruddin, S.Si., M.T.  
NIDN. 0909106901

## PERNYATAAN ORISINALITAS

Penulis dengan ini menyatakan bahwa Tugas akhir.

“Sistem Penjadwalan Sidang Menggunakan Algoritma Genetika Pada Pengadilan Tata Usaha Negara” adalah karya orisinal saya dan setiap serta dan seluruh sumber acuan telah di tulis sesuai dengan Panduan Penulis Ilmiah yang berlaku di Fakultas Universitas Fajar

Makassar, 10 April 2023

Yang Menyatakan



Dzuldar

## ABSTRAK

**Sistem Penjadwalan Sidang Menggunakan Algoritma Genetika Pada Pengadilan Tata Usaha Negara Berbasis Website, Dzuldar.** Penjadwalan sidang merupakan salah satu kegiatan penting yang dimiliki Pengadilan Tinggi Tata Usaha Negara Makassar. Proses penyusunan jadwal sidang yang masih semi terkomputerisasi, sehingga membutuhkan waktu serta harus memperhatikan banyak komponen yaitu: hari, waktu, ruangan, dan jumlah hakim yang tersedia. Sehingga diperlukan metode yang dapat membantu proses penjadwalan sidang. Algoritma genetika merupakan salah satu algoritma heuristik yang dapat digunakan untuk menyelesaikan masalah penjadwalan. Sistem penjadwalan sidang pada Pengadilan Tinggi Tata Usaha Negara Makassar menggunakan algoritma genetika berbasis web dirancang menggunakan pemodelan visual UML *use case*, menggunakan bahasa pemrograman PHP, dan *database* MySQL. Metode pengujian yang digunakan dalam sistem penjadwalan ini adalah metode *black box*. Hasil yang diperoleh menunjukkan bahwa tidak terdapat jadwal yang bentrok satu sama lain dan proses penjadwalan menjadi lebih cepat dan efektif.

Kata Kunci: Penjadwalan, Sidang, Pengadilan Tata Usaha Negara, Algoritma Genetika, Black Box.

## ABSTRACT

*Trial Scheduling System Using Genetic Algorithms in Website-Based State Administrative High Court, Dzuldar. Cheduling trials is one of the important activities of the Makassar State Administrative High Court. The process of preparing a trial schedule is still semi-computerized, so it takes time and has to pay attention to many components, namely: day, time, room, and the number of available judge. So we need a method that can help the trial scheduling process. The genetic algorithm is one of the heuristic algorithms that can be used to solve scheduling problems. The trial scheduling system at the Makassar State Administrative High Court uses a web-based genetic algorithm designed using UML use case visual modeling, using the PHP programming language, and MySQL database. The testing method used in this scheduling system is the black box method. The results obtained show that there are no schedules that clash with each other and the scheduling process becomes faster and more effective.*

*Keyword: Scheduling, Trial, State Administrative High Court, Genetic Algorithm, Black Box*

## KATA PENGANTAR

Segalah puji dan syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa, yang telah memberi berkat dan pertolongan-Nya, sehingga penulis dapat menyelesaikan penulisan tugas akhir ini yang berjudul “Sistem Penjadwalan Sidang Menggunakan Algoritma Genetika Pada Pengadilan Tata Usaha Negara”.

Dalam penyusunan tugas akhir ini tidak akan selesai tanpa bantuan dari berbagai pihak karena itu pada kesempatan ini penulis ingin mengucapkan terima kasih kepada:

1. Orang tua dan keluarga besar yang senantiasa mendoakan agar penulis selalu diberi kekuatan dan kesabaran
2. Dekan Fakultas Teknik Universitas Fajar Makassar.
3. Ketua Prodi Teknik Elektro Universitas Fajar Makassar, Bapak Dr. Safaruddin S.Si., M.T.
4. Dosen Pembimbing ibu Andita Dani Achmad, S.T., M.T. selaku pembimbing I dan Bapak Febriansyah, S.Kom., M.T. selaku pembimbing II di Fakultas Teknik Program Studi Teknik Elektro
5. Teman-teman serta semua pihak dengan segala kerendahan hati membantu dalam penyelesaian tugas akhir ini.

Diharapkan, tugas akhir ini dapat bermanfaat untuk semua pihak. Selain itu, kritik dan saran yang membangun sangat penulis harapkan dari para pembaca sekalian agar ini bisa lebih baik lagi.

Makassar, 6 Maret 2023

Dzuldar

## DAFTAR ISI

	Halaman
<b>HALAMAN JUDUL</b> .....	<b>i</b>
<b>HALAMAN PENGESAHAN</b> .....	<b>ii</b>
<b>PERNYATAAN ORISINALITAS</b> .....	<b>iii</b>
<b>ABSTRAK</b> .....	<b>iv</b>
<b>ABSTRACT</b> .....	<b>v</b>
<b>KATA PENGANTAR</b> .....	<b>vi</b>
<b>DAFTAR ISI</b> .....	<b>vii</b>
<b>DAFTAR TABEL</b> .....	<b>x</b>
<b>DAFTAR GAMBAR</b> .....	<b>xi</b>
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
I.1 Latar Belakang .....	1
I.2 Rumusan Masalah.....	3
I.3 Tujuan Penelitian .....	3
I.4 Batasan Masalah .....	3
<b>BAB II TINJAUAN PUSTAKA</b> .....	<b>4</b>
II.1 Tinjauan Teori.....	4
II.1.1 Sistem Penjadwalan .....	4
II.1.2 Pengadilan Tata Usaha Negara.....	4
II.1.3 Jenis-jenis Kekuasaan Peradilan.....	6
II.1.4 Algoritma Genetika .....	8
II.1.5 <i>Flowchart</i> .....	9
II.1.6 Metode <i>Waterfall</i> .....	10
II.1.7 UML ( <i>Unified Modelling Language</i> ) .....	11
II.1.8 Apache .....	13
II.1.9 PHP MyAdmin .....	14
II.1.10 MySQL .....	14
II.1.11 PHP ( <i>Hypertext Preprocessor</i> ).....	14
II.1.12 Pengujian Sistem .....	15
II.2 Penelitian Terdahulu ( <i>State of The Art</i> ) .....	18

II.3 Kerangka Fikir .....	20
<b>BAB III METODOLOGI PENELITIAN.....</b>	<b>21</b>
III.1 Bagan Alur Penelitian/Tahapan Penelitian .....	21
III.2 Perancangan Sistem.....	22
III.2.1 Analisis Sistem yang Berjalan.....	22
III.2.2 Analisis Sistem yang Diusulkan .....	23
III.3 Waktu dan Tempat Penelitian.....	41
III.4 Alat dan Bahan Penelitian .....	41
III.5 Metode Pengumpulan Data .....	42
III.6 Metode Pengujian Sistem .....	42
<b>BAB IV HASIL DAN PEMBAHASAN.....</b>	<b>43</b>
IV.1 Hasil Penelitian.....	43
IV.2 Pembahasan .....	47
<b>BAB V PENUTUP .....</b>	<b>59</b>
V.1 Kesimpulan.....	59
V.2 Saran.....	59
<b>DAFTAR PUSTAKA .....</b>	<b>60</b>



## DAFTAR TABEL

	Halaman
Tabel II.1 Penelitian Terdahulu/ <i>State of The Art</i> .....	18
Tabel III.1 Perangkat Keras.....	41
Tabel III.2 Perangkat Lunak.....	41
Tabel IV.1 Hasil Pengujian <i>Black Box</i> .....	47
Tabel IV.2 <i>Text Case Login</i> .....	52
Tabel IV.3 Nilai Skor Maksimum.....	53
Tabel IV.4 Kriteria Skor.....	54
Tabel IV.5. Hasil Kuesioner Pertanyaan Pertama .....	54
Tabel IV.6 Hasil Kuesioner Pertanyaan Kedua.....	55
Tabel IV.7. Hasil Kuesioner Pertanyaan Ketiga .....	56
Tabel IV. 8 Hasil Kuesioner Pertanyaan Keempat.....	57
Tabel IV.9 Hasil Kuesioner Pertanyaan Kelima .....	57
Tabel IV.10 Pengolahan Skala .....	58

## DAFTAR GAMBAR

	Halaman
Gambar II.1 <i>Flowchat</i> .....	10
Gambar II.2 Model <i>Waterfall</i> .....	11
Gambar II.3 Simbol <i>Use Case</i> .....	12
Gambar II.4 Simbol <i>Activity Diagram</i> .....	13
Gambar II.5 Metode <i>Black Box Testing</i> .....	15
Gambar II.6 Metode <i>White Box Testing</i> .....	17
Gambar II.7 Kerangka Fikir .....	20
Gambar III.1 Tahapan Penelitian .....	21
Gambar III.2 Sistem yang Sedang Berjalan .....	22
Gambar III.3 Sistem yang Diusulkan .....	23
Gambar III.4 <i>Use Case Diagram</i> .....	24
Gambar III.5 <i>Activity Diagram Login Akun Pengguna</i> .....	25
Gambar III.6 <i>Activity Diagram Registrasi Pengajuan Pada Penggugat</i> .....	25
Gambar III.7 <i>Activity Diagram Lihat Data Pengajuan Pada Admin</i> .....	26
Gambar III.8 <i>Activity Diagram Verifikasi Berkas Pada Admin</i> .....	26
Gambar III.9 <i>Activity Diagram Generate Jadwal Pada Admin</i> .....	27
Gambar III.10 <i>Activity Diagram Cetak Jadwal Sidang Pada Penggugat</i> .....	27
Gambar III.11 <i>Activity Diagram Cetak Jadwal Sidang Pada Hakim</i> .....	28
Gambar III.12 <i>Activity Diagram Cetak Jadwal Sidang Pada Admin</i> .....	28
Gambar III.13 <i>Sequence Diagram Mengakses Sistem Penjadwalan Persidangan</i> 29	
Gambar III.14 <i>Sequence Diagram Login Akun Pengguna</i> .....	29
Gambar III.15 <i>Sequence Diagram Registrasi Pengajuan Pada Penggugat</i> .....	30
Gambar III.16 <i>Sequence Diagram Lihat Data Pengajuan Pada admin</i> .....	30
Gambar III.17 <i>Sequence Diagram Verifikasi Berkas Pada Admin</i> .....	31
Gambar III.18 <i>Sequence Diagram Generate Jadwal Pada Admin</i> .....	31
Gambar III.19 <i>Sequence Diagram Lihat Jadwal Sidang Pada Penggugat</i> .....	32
Gambar III.20 <i>Sequence Diagram Cetak Jadwal Sidang Pada Hakim</i> .....	32
Gambar III.21 <i>Class Diagram</i> .....	33
Gambar III.22 Halaman <i>Login</i> .....	33

Gambar III.23 <i>Form Login</i> .....	34
Gambar III.24 <i>Form Daftar</i> .....	34
Gambar III.25 Halaman Utama Sistem .....	34
Gambar III.26 Halaman Data Hakim .....	35
Gambar III.27 Halaman Data Penggugat .....	35
Gambar III.28 Halaman Jadwal Sidang .....	35
Gambar III.29 Diagram Proses Algoritma Genetika .....	36
Gambar IV.1 Tampilan Halaman <i>Login</i> .....	43
Gambar IV.2 Tampilan Halaman Daftar .....	44
Gambar IV.3 Tampilan Halaman Utama Sistem.....	44
Gambar IV.4 Tampilan Halaman Utama Admin.....	45
Gambar IV.5 Halaman Daftar Daftar Penggugat .....	45
Gambar IV.6 Tampilan Halaman <i>Generate</i> Jadwal .....	46
Gambar IV.7 Tampilan Halaman Jadwal Sidang.....	46
Gambar IV.8 <i>Flowchart Login</i> .....	50
Gambar IV.9 <i>Flowgraph Login</i> .....	50

# **BAB I**

## **PENDAHULUAN**

### **I.1 Latar Belakang**

Penjadwalan merupakan suatu kegiatan yang dimiliki oleh setiap orang untuk dapat membantu dalam melakukan aktivitasnya sehari-hari. Terlebih lagi untuk sebuah instansi atau lembaga yang memiliki kegiatan yang penting dan rumit yang dapat diselesaikan dengan secara teratur dan rapi, begitu pentingnya penjadwalan ini dibuat agar kegiatan dapat berjalan sesuai dengan yang telah direncanakan (Assagaf, 2018).

Pengadilan Tata Usaha Negara Makassar sebagai salah satu instansi pemerintah yang bertugas memberikan penjadwalan dan pelayanan persidangan pada masyarakat dibidang hukum tentu harus memenuhi tuntutan masyarakat yang mengalami masalah atau konflik yang tidak dapat terselesaikan secara damai dan harus diselesaikan secara hukum. Sementara untuk membuat suatu jadwal sidang di Pengadilan Tata Usaha Negara (PTUN) Makassar yang cukup ideal membutuhkan waktu yang lama dikarenakan harus melalui beberapa tahapan yang saling terkait sehingga membutuhkan waktu untuk penanganan yang signifikan. Dalam pelaksanaan penjadwalan sidang Pengadilan Tata Usaha Negara (PTUN) Makassar ada hal yang harus dipertimbangkan seperti *availability* hakim, penggugat, tergugat, jam, tanggal, hari, ruangan.

Isu yang selalu mengemuka di masyarakat mengenai pelayanan pada Pengadilan Tata Usaha Negara (PTUN) Makassar adalah terlambatnya penyelesaian proses perkara. Tingkat penyelesaian perkara tidak hanya disebabkan oleh faktor kemampuan para hakim dalam memeriksa dan memutus perkara, namun juga masalah minimnya informasi yang diperoleh para pihak yang berperkara, sulitnya dalam mengakses informasi layanan persidangan yang masih terkesan tertutup, lambatnya pihak pengadilan dalam memberikan informasi karena informasi yang diberikan masih mengandalkan surat panggilan yang terkadang tidak tersampaikan kepada para pihak

sehingga membuat persidangan kembali tertunda dan mengakibatkan terlambatnya dalam proses penyelesaian perkara.

Dengan adanya perkembangan kemajuan teknologi informasi yang sangat pesat di era modern seperti saat ini, di mana setiap orang dapat menggunakan teknologi informasi diberbagai bidang kehidupan manusia.

Algoritma genetika adalah suatu algoritma pencarian yang meniru mekanisme dari genetika alam. Algoritma genetika pertama kali dikemukakan oleh John Holland awal tahun 1975. Algoritma Genetika banyak dipakai pada aplikasi bisnis, teknik maupun pada bidang keilmuan lainnya. Algoritma ini dimulai dengan kumpulan solusi yang disebut dengan populasi. Algoritma genetika cukup baik untuk digunakan dalam penjadwalan mata kuliah di sebuah perguruan tinggi negeri karena algoritma ini dapat menyelesaikan masalah multi-kriteria dan multi-objektif untuk menyelesaikan masalah yang dimodelkan dengan proses biologi dan evolusi (Josi, 2017).

Beberapa penelitian juga menggunakan algoritma genetika untuk membuat proses penjadwalan secara otomatis, antara lain Sari (2019) yang menggunakan parameter waktu untuk menggambarkan algoritma genetika mampu melakukan otomatisasi penjadwalan. Begitu juga penelitian yang dilakukan oleh Permadi (2010), melakukan perbandingan parameter *crossover* pada algoritma genetika. Penelitian Fiarni (2015) menggunakan *forward chaining method* pada penjadwalan sidang yaitu algoritma pencarian solusi secara berantai, penelitian ini menghasilkan alokasi waktu yang cocok (tidak disertai dengan ruangan) dari sebuah data mahasiswa dengan kelompok dosen yang menjadi penguji mahasiswa tersebut.

Berdasarkan latar belakang di atas, maka perlu kiranya dibuat sistem informasi yang dapat mempermudah aktivitas persidangan pada Pengadilan Tata Usaha Negara Makassar. Oleh karena itu, dilakukan penelitian dengan menganalisa dan merancang sebuah sistem informasi dengan judul “**Sistem Penjadwalan Sidang menggunakan Algoritma Genetika pada Pengadilan Tata Usaha Negara Kota Makassar**”.

## **I.2 Rumusan Masalah**

Rumusan masalah pada perancangan ini yang diharapkan bisa diselesaikan dengan penelitian ini sebagai berikut:

1. Bagaimana merancang sistem penjadwalan sidang untuk mempermudah masyarakat?
2. Bagaimana menerapkan jadwal sidang menggunakan algoritma genetika?

## **I.3 Tujuan Penelitian**

Adapun tujuan dari penelitian ini sebagai berikut:

1. Merancang sistem penjadwalan sidang yang dapat mempermudah masyarakat.
2. Menerapkan jadwal sidang menggunakan algoritma genetika.

## **I.4 Batasan Masalah**

Adapun batasan masalah pada perancangan ini sebagai berikut:

1. Aktor pada aplikasi ini hanya terdapat 3 aktor, yaitu: penggugat, hakim, dan admin.
2. Parameter penentuan jadwal yaitu hakim, hari, ruangan, dan jam.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1 Tinjauan Teori**

##### **II.1.1 Sistem Penjadwalan**

Menurut Gerald (dalam Mulyanto, 2009), sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu. Dalam mendefinisikan pengertian sistem, Gerald lebih menekankan pada urutan-urutan operasi di dalam sistem. Sistem memiliki sifat-sifat atau karakter untuk dapat menjalankan suatu fungsi tertentu.

Menurut Pinedo (dalam Bayu, 2021) penjadwalan merupakan sebuah fungsi pengambilan keputusan yang biasa digunakan banyak perusahaan manufaktur atau jasa, yang berhubungan dengan alokasi sumber daya untuk mengerjakan tugas selama waktu tertentu dan tujuannya adalah untuk mengoptimalkan satu atau lebih tujuan pengoptimalan. Penjadwalan merupakan kegiatan yang harus dimiliki oleh setiap orang untuk dapat membantu dalam melakukan aktivitasnya sehari-hari. Terlebih lagi sebuah instansi atau lembaga yang memiliki agenda-agenda penting dan rumit yang diselesaikan secara teratur dan rapi. Begitu pentingnya penjadwalan ini agar kegiatan dapat berjalan sesuai dengan yang telah direncanakan.

##### **II.1.2 Pengadilan Tata Usaha Negara**

Tata Usaha Negara adalah administrasi negara yang melaksanakan fungsi untuk menyelenggarakan urusan pemerintah baik di pusat maupun di daerah. Menurut Atmosudirjo (2014), pengertian Peradilan Tata Usaha Negara (PTUN) dapat dibagi menjadi dua, yaitu dalam artian luas dan dalam artian sempit, pengertian PTUN dalam artian luas adalah peradilan yang menyangkut pejabat-pejabat, dan instansi-instansi administrasi negara, baik yang bersifat perkara pidana, perkara perdata, perkara agama, perkara adat, dan perkara administrasi negara. Sedangkan pengertian PTUN dalam arti

sempit adalah peradilan yang menyelesaikan perkara- perkara administrasi negara.

Berdasarkan Pancasila dan UUD 1945 oleh karenanya hak dan kepentingan perseorangan dijunjung tinggi disamping juga hak masyarakatnya. Kepentingan perseorangan adalah seimbang dengan kepentingan masyarakat atau kepentingan umum. Tujuan pembentukan peradilan tata usaha negara (PTUN) adalah untuk memberikan perlindungan terhadap hak-hak perseorangan dan hak-hak masyarakat, sehingga tercapai keserasian, keseimbangan, dan keselarasan antara kepentingan perseorangan dengan kepentingan masyarakat atau kepentingan umum. Selain itu, menurut Atmosudirdjo tujuan dibentuknya peradilan tata usaha negara (PTUN) adalah untuk mengembangkan dan memelihara administrasi negara yang tepat menurut hukum (*rechtmatic*) atau tepat. Dengan demikian lembaga pengadilan tata usaha negara (PTUN) adalah sebagai salah satu badan peradilan yang melaksanakan kekuasaan kehakiman, merupakan kekuasaan yang merdeka yang berada di bawah Mahkamah Agung dalam rangka menyelenggarakan peradilan guna menegakkan hukum dan keadilan. Berdasarkan hal tersebut, maka Peradilan Tata Usaha Negara (PTUN) diadakan dalam rangka memberikan perlindungan (berdasarkan keadilan, kebenaran dan ketertiban dan kepastian hukum) kepada rakyat pencari keadilan (*justiciabelen*) yang merasa dirinya dirugikan akibat suatu perbuatan hukum publik oleh pejabat administrasi negara, melalui pemeriksaan, pemutusan dan penyelesaian sengketa dalam bidang administrasi negara (Hendrik, 2010).

Jadi pengadilan bukanlah merupakan satu satunya wadah yang menyelenggarakan peradilan. Pengadilan adalah badan atau organisasi yang diadakan oleh negara untuk mengurus atau mengadili perselisihan-perselisihan hukum. Pengadilan tata usaha negara merupakan salah satu sarana dalam mencapai tujuan keadilan dalam hukum, oleh karena itu organisasi menurut hukum merupakan wahana dalam kegiatan dan kerjasama untuk mencapai tujuan utama. Wadah kegiatan setiap orang atau



badan hukum harus jelas mempunyai tugas dan wewenang serta hubungan tata kerja, demikian pula dalam peradilan tata usaha Nnegara ini dan dalam pengertian demikian ini biasanya tidak hanya melihat struktur organisasi saja akan tetapi dalam pengertian organisasi bersifat dinamis dilihat dari sisi kegiatan aktivitas tindakan tata hubungan yang terjadi organisasi bersifat formil dan non formil beralaskan hukum hal ini dapat dilihat dari adanya hubungan antara atasan yaitu mahkamah agung dengan pengadilan tinggi peradilan tata usaha negara dan peradilan tata usaha negara di daerah-daerah yang bersifat satu tujuan dengan sistem kekeluargaan dan permusyawaratan demi untuk menciptakan suatu peradilan yang baik dan benar. Berhasil atau tidaknya suatu tujuan akan dicapai dan dicita-citakan pembentuk Undang-undang atau organisasi itu tergantung sepenuhnya dari faktor manusia dan individu pendukungnya.

Peradilan Tata Usaha Negara (PERATUN) merupakan lingkungan peradilan yang terakhir dibentuk, yang ditandai dengan disahkannya Undang-Undang Nomor 5 Tahun 1986 pada tanggal 29 Desember 1986, adapun tujuan dibentuknya Peradilan Tata Usaha Negara (PERATUN) adalah untuk mewujudkan tata kehidupan negara dan bangsa yang sejahtera, aman, tenteram serta tertib yang dapat menjamin kedudukan warga masyarakat dalam hukum dan menjamin terpeliharanya hubungan yang serasi, seimbang, serta selaras antara aparatur di bidang tata usaha negara dengan para warga masyarakat. Dengan terbentuknya Peradilan Tata Usaha Negara (PERATUN) menjadi bukti bahwa Indonesia adalah negara hukum yang menjunjung tinggi nilai-nilai keadilan, kepastian hukum dan Hak Asasi Manusia.

### **II.1.3 Jenis-jenis Kekuasaan Peradilan**

Miriam Budiarjo (2008), mendefinisikan kekuasaan sebagai kemampuan seseorang atau suatu kelompok untuk memengaruhi tingkah laku orang atau kelompok lain sesuai dengan keinginan pelaku, kekuasaan negara menjadi tiga, yaitu:

### 1. Kekuasaan Eksekutif

Kekuasaan eksekutif adalah kekuasaan untuk melaksanakan undang-undang. Cabang kekuasaan ini yang memegang kewenangan administrasi pemerintahan negara yang tertinggi. Kekuasaan ini berkaitan dengan sistem pemerintahan negara yang dianut masing-masing. Misalnya saja, Indonesia menganut sistem pemerintahan presidensial. Sehingga secara sempit, kekuasaan eksekutif berada ditangan presiden sebagai kepala negara dan kepala pemerintahan.

### 2. Kekuasaan Legislatif

Berbeda dari kekuasaan eksekutif yang melaksanakan undang-undang, kekuasaan legislatif adalah kekuasaan untuk membuat atau merumuskan undang-undang yang diperlukan negara. Cabang kekuasaan legislatif adalah cabang kekuasaan yang mencerminkan kedaulatan rakyat karena untuk menetapkan peraturan adalah wewenang dari lembaga perwakilan rakyat atau parlemen. Singkatnya, kekuasaan legislatif menjalankan fungsi pengaturan.

### 3. Kekuasaan Yudikatif

Kekuasaan yudikatif mengatur perihal kekuasaan yang merdeka untuk menyelenggarakan peradilan guna menegakkan hukum dan keadilan yang dilakukan oleh mahkamah agung dan badan peradilan yang berada di bawahnya dalam lingkungan peradilan umum, lingkungan peradilan agama, lingkungan peradilan militer, lingkungan peradilan tata usaha negara dan oleh mahkamah konstitusi. Jadi, singkatnya, menurut undang-undang dasar 1945 kekuasaan yudikatif di Indonesia dijalankan oleh lembaga mahkamah agung.

## II.1.4 Algoritma Genetika

Algoritma genetika didefinisikan sebagai algoritma yang berusaha menerapkan pemahaman mengenai evolusi alamiah pada tugas-tugas pemecahan masalah (*problem solving*). Pendekatan yang diambil oleh algoritma ini adalah dengan menggabungkan secara acak berbagai pilihan

solusi terbaik di dalam suatu kumpulan untuk mendapatkan generasi solusi terbaik berikutnya yaitu pada suatu kondisi yang memaksimalkan kecocokannya atau lazim disebut *fitness*. Generasi ini akan merepresentasikan perbaikan-perbaikan pada populasi awalnya. Dengan melakukan proses ini secara berulang, algoritma ini diharapkan dapat mensimulasikan proses *evolusioner* (Lumban, 2010).

Algoritma genetika sangat tepat digunakan untuk penyelesaian masalah optimasi yang kompleks dan sukar diselesaikan dengan menggunakan metode yang konvensional. Sebagaimana halnya proses evolusi di alam, suatu algoritma genetika yang sederhana umumnya terdiri dari tiga operator yaitu: operator seleksi, operator *crossover* (persilangan), dan operator mutasi. Untuk menggunakan algoritma genetika, solusi permasalahan direpresentasikan sebagai kromosom. Tiga aspek yang penting untuk penggunaan algoritma genetik (Angela, 2006) yaitu:

1. Definisi fungsi *fitness*.
2. Definisi dan implementasi representasi genetika.
3. Definisi dan implementasi operasi genetika.

Algoritma genetika dikembangkan pertama kali oleh John Holland dari New York, Amerika Serikat yang dipublikasikan dalam bukunya yang berjudul "*Adaption in Natural and Artificial Systems*" tahun 1975. Algoritma genetika merupakan teknik untuk menemukan solusi optimal dari permasalahan yang mempunyai banyak solusi. Teknik ini akan melakukan pencarian dari beberapa solusi yang diperoleh sampai mendapatkan solusi terbaik sesuai dengan kriteria yang telah ditentukan atau yang disebut sebagai fungsi *fitness*. Algoritma ini masuk dalam kelompok algoritma evolusioner dengan menggunakan pendekatan evolusi Darwin di bidang biologi seperti pewarisan sifat, seleksi alam, mutasi gen, dan kombinasi (*crossover*) karena merupakan teknik pencarian optimal dalam bidang ilmu komputer, maka algoritma ini juga termasuk dalam kelompok algoritma *metaheuristik*.



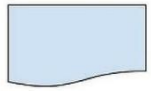
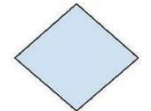
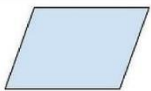
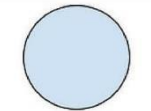
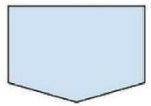

Aplikasi algoritma genetika dapat ditemukan diberbagai bidang terutama bidang-bidang yang memerlukan solusi kombinatorik seperti penjadwalan, peramalan, jarak terpendek, dan kombinasi ransum atau bahan. Algoritma genetika sering dipakai untuk melakukan simulasi dengan komputer untuk mendapatkan solusi terbaik berdasarkan calon-calon solusi yang *visible*. Proses pencarian solusi terbaik dimulai dengan merepresentasikan solusi-solusi yang mungkin terjadi berdasarkan domain yang biasanya dalam bentuk string biner (0 dan 1). Dari representasi ini dibentuk populasi individual secara acak yang membentuk suatu generasi. Kemudian dari setiap populasi yang terbentuk dievaluasi dengan menggunakan fungsi fitness untuk dapat memilih populasi terbaik. Kemudian populasi dimodifikasi dengan mutasi dan kombinasi untuk mendapatkan populasi baru. Proses ini diulang sampai mendapatkan individu dari populasi yang mencapai nilai *fitness*. Adapun langkah-langkah algoritma genetika dapat diuraikan sebagai berikut:

1. Penentuan representasi individu dari populasi.
2. Pembentukan populasi individu secara acak.
3. Evaluasi kecocokan setiap individu dalam populasi berdasarkan fungsi *fitness*.
4. Memilih individu dengan nilai kecocokan paling tinggi.
5. Kombinasi antar individu terpilih dalam populasi dan mutasi individu dengan tingkat tertentu untuk membentuk populasi baru.
6. Proses 3 diulang sampai mendapatkan solusi terbaik.

### **II.1.5 Flowchart**

Menurut Setyo (2013), *flowchart* merupakan penyajian yang sistematis tentang proses dan logika dari kegiatan penanganan informasi atau penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program. Bagan alir (*flowchart*) adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara

logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi.

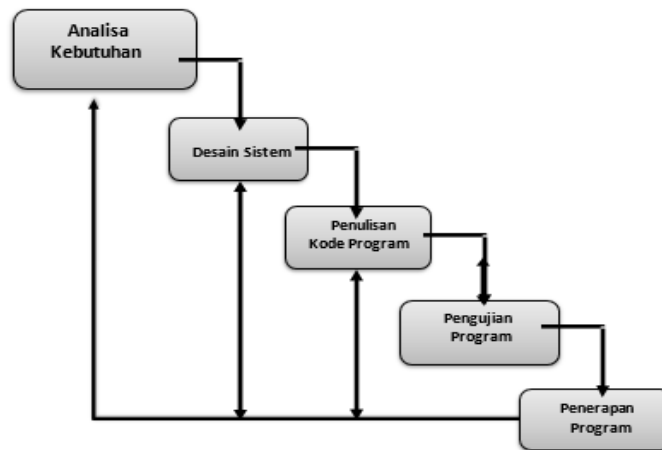
No.	Simbol Flowchart	Nama	Arti Simbol Flowchart
1		<i>Terminator</i>	Awal atau akhir konsep (prosedur)
2		<i>Process</i>	Proses operasional
3		<i>Document</i>	Dokumen atau laporan berupa <i>print out</i>
4		<i>Decision</i>	Keputusan atau sub-point. Garis yang terhubung dengan bentuk <i>decision</i> merujuk pada situasi-situasi yang berbeda sesuai dengan keputusan yang digambarkan
5		Data	Input dan Output (Contohnya, Input: feedback dari pelanggan. Output: desain produk baru)
6		<i>On-Page Reference/Connector</i>	Penghubung alur dalam halaman yang sama
7		<i>Off-Page Reference/Off-Page Connector</i>	Penghubung alur dalam halaman yang berbeda
8		<i>Flow</i>	Arah alur dalam konsep (prosedur)

Gambar II.1 *Flowchart*

Sumber: <https://www.uml-diagrams.org/activity-diagrams-controls.html>

### II.1.6 Metode *Waterfall*

*Waterfall* merupakan salah satu metode dalam SDLC (*System Development Life Cycle*) yang mempunyai ciri khas pengerjaan setiap fase dalam *waterfall* harus diselesaikan terlebih dahulu sebelum dilanjutkan ke fase berikutnya. Metode *waterfall* adalah pengerjaan dari suatu sistem dilakukan secara berurutan atau secara linier.



Gambar II.2 Model *Waterfall*

Sumber: <https://socs.binus.ac.id/2020/07/02/teknik-dalam-white-box-dan-black-box-testing/>

Secara garis besar metode *waterfall* mempunyai langkah-langkah sebagai berikut: analisa, desain, penulisan, pengujian, penerapan serta pemeliharaan (Kadir, 2003).

### II.1.7 UML (*Unified Modelling Language*)

Menurut Sukamto (2015), UML (*Unified Modelling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis, dan desain serta menggambarkan arsitektur dalam pemrograman berorientasikan objek. UML ini terdiri dari 13 macam diagram namun hanya beberapa diagram yang digunakan, diantaranya:

#### 1. *Use Case Diagram*




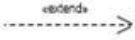


Menurut Sukamto (2015: 155), *use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

- a. Aktor merupakan orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi

yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.

- b. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Berikut adalah simbol-simbol yang ada pada diagram *use case* (Sukamto, 2014:156)




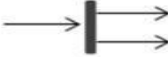


	<b>ACTOR</b> Orang proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari actor adalah gambar orang, biasanya dinyatakan menggunakan kata benda di awal frase nama <i>actor</i> .
	<b>USE CASE</b> Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau actor biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama use case.
	<b>ASOSIASI/ASSOCIATION</b> Komunikasi antara <i>actor</i> dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan <i>actor</i> .
	<b>EKSTENSI/EXTEND</b> Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan memiliki nama depan yang sama dengan use case yang di tambahkan.
	<b>GENERALISASI/GENERALIZATION</b> Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
	<b>MENGGUNAKAN/INCLUDE</b> Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsional atau sebagai syarat dijalankan use case ini.

Gambar II.3 Simbol *Use Case*

Sumber: [www.uml-diagrams.org/use-case-subject.html](http://www.uml-diagrams.org/use-case-subject.html)

## 2. Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak (Sukamto, 2015). Berikut adalah simbol-simbol yang ada pada diagram aktivitas:

	<b>STATUS AWAL/INITIAL</b> Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
	<b>AKTIVITAS/ACTIVITY</b> Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	<b>PERCABANGAN/DECISION</b> Asosiasi percabangan dimana lebih dari satu aktivitas digabungkan menjadi satu.
	<b>PENGGABUNGAN/JOIN</b> Asosiasi penggabungan dimana lebih dari satu aktivitas lebih dari satu.
	<b>STATUS AKHIR/FINAL</b> Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status satu.
	<b>SWIMLINE</b> Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Gambar II.4 Simbol Activity Diagram

Sumber: <https://www.uml-diagrams.org/activity-diagrams-controls.html>

### II.1.8 Apache

Menurut Artanto (2013), server HTTP apache atau server web/www Apache adalah server web yang dapat dijalankan dibanyak sistem operasi (Unix, BSD, Linux, Microsoft Windows, dan Novel Netware serta platform lainnya) yang berguna untuk melayani dan memfungsikan situs web. Protokol yang digunakan untuk melayani fasilitas web/www ini menggunakan HTTP.

Apache memiliki fitur-fitur canggih seperti pesan kesalahan yang dapat dikonfigurasi, dan autentikasi yang berbasis basis data. Apache juga didukung oleh sejumlah antar muka pengguna berbasis grafik *Grafical User Interface (GUI)* yang memungkinkan penanganan server menjadi mudah. Apache merupakan perangkat lunak sumber terbuka dikembangkan oleh



komunitas terbuka yang terdiri dari pengembang- pengembang dibawah naungan Apache Software Foundation.

### **II.1.9 PHP MyAdmin**

Menurut Anhar (2010), PHP MyAdmin adalah aplikasi berbasis web yang ditulis dalam bahasa PHP yang fungsi utamanya melakukan administrasi MySQL. Administrasi *user* MySQL inilah yang akan digunakan untuk mengakses *database* MySQL dengan PHP.

### **II.1.10 MySQL**

Menurut Artanto (2013), MySQL (*My Structure Query Language*) adalah sebuah program berbasis DOS yang bersifat *open source*. MySQL adalah produk yang berjalan pada platform baik Windows maupun Linux. Selain itu, MySQL merupakan program pengakses *database* yang bersifat jaringan sehingga dapat digunakan untuk *multi user* (banyak pengguna).

Kelebihan lain dari MySQL adalah menggunakan bahasa *query* standar yang dimiliki SQL (*Structure Query Language*). SQL adalah suatu bahasa permintaan yang terstruktur yang telah distandarkan untuk semua program pengakses *database* seperti Oracle, Postgres SQL, dan SQL Server. Sebagai program penghasil *database*, MySQL tidak dapat berjalan sendiri tanpa adanya sebuah aplikasi lain (*interface*) MySQL dapat didukung oleh hampir semua program aplikasi baik yang *open source* maupun yang tidak, yang ada pada *platform* Windows.

### **II.1.11 PHP (*Hypertext Preprocessor*)**

Menurut Sutarman (2003), PHP merupakan akronim dari PHP: *Hypertext Preprocessor* adalah bahasa pemrograman *script* berbasis web yang paling banyak dipakai saat ini. PHP banyak dipakai untuk memrogram situs web dinamis, walaupun tidak tertutup kemungkinan digunakan untuk pemakaian lain. Untuk menjalankan sistem PHP dibutuhkan tiga komponen:

1. *Web server*, karena PHP termasuk bahasa pemrograman *server side*.

2. Program PHP, program yang memproses *script* PHP.
3. *Database server*, yang berfungsi untuk mengelola *database*.

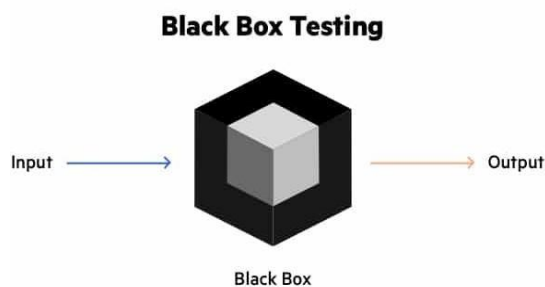
Kelebihan PHP dari bahasa pemrograman lain adalah:

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 mulai IIS sampai dengan Apache, dengan konfigurasi yang relatif mudah.
3. PHP adalah bahasa *open source* yang dapat digunakan diberbagai mesin (Linux, Unix, Windows) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem.

### II.1.12 Pengujian Sistem

1. *Black Box*

*Black box testing* merupakan cara pengujian perangkat lunak yang dipergunakan buat menguji perangkat lunak tanpa mengenali struktur internal kode ataupun program. *Black box testing* melibatkan pengujian berdasarkan detail aplikasi seperti tampilan aplikasi, fungsi-fungsi yang ada pada aplikasi, serta kesesuaian aliran fungsi untuk proses bisnis yang diinginkan oleh pengguna.



Gambar II.5 Metode *Black Box Testing*

Sumber: <https://socs.binus.ac.id/2020/07/02/teknik-dalam-white-box-dan-black-box-testing/>

Pengujian pada *black box testing* dapat menemukan permasalahan (Flinsetyadiz, 2020) seperti:

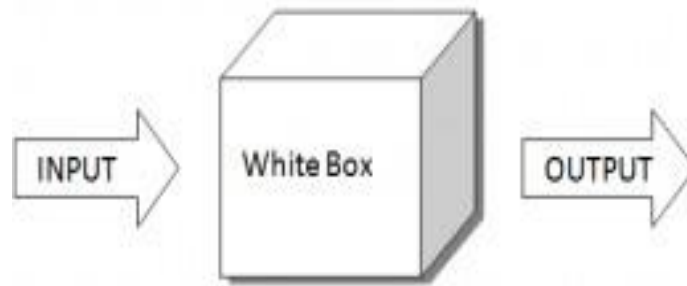
- a. Fungsi-fungsi tidak benar atau *error*.
- b. Masalah *interface*.
- c. Masalah terhadap kinerja atau *bug*.
- d. Masalah terhadap struktur data atau akses *database eksternal*.

Adapun kelebihan dan kekurangan pada *black box testing* (Flinsetyadiz, 2020) yaitu:

- a. Kelebihan *black box testing*
  - 1) Efisien untuk menguji *software* atau kode besar.
  - 2) Tidak perlu melihat *source code* secara mendetail.
  - 3) Mendeteksi kesalahan dalam pengetikan.
- b. Kekurangan *black box testing*
  - 1) Peninjau memiliki keterbatasan dalam melakukan pengujian atau hanya sebagian kecil dari skenario pengujian yang dilakukan.
  - 2) Pengujian tidak efisien dikarenakan keberuntungan *tester* dari pengetahuan tentang perangkat lunak internal.
  - 3) Jika spesifikasi program kurang jelas dan ringkas, maka akan sulit untuk mendokumentasikan seakurat mungkin.

## 2. *White Box*

Pengujian *white box* adalah cara untuk menguji suatu aplikasi atau perangkat lunak dengan melihat modul untuk melihat apakah kode program salah atau benar. Jika modul ini dibuat dalam *ouput* yang tidak memenuhi persyaratan, kode akan dikompilasi ulang dan diperiksa ulang hingga mencapai nilai yang diharapkan. Singkatnya pengujian *white box* ini menguji menggunakan kode murni aplikasi yang akan diuji, terlepas dari tampilan aplikasi pengguna (Wairooy, 2020).



Gambar II.6 Metode *White Box Testing*

Sumber: <https://www.erfaansetyadi.com/2019/01/08/pengertian-white-box-testing/>

Adapun kelebihan dan kekurangan pada *white box testing* (Wairooy, 2020) yaitu:

a. Beberapa kelebihan *white box testing*

- 1) Penguji dapat menguji semua bentuk, data, serta struktur aplikasi secara sekaligus.
- 2) Lebih detail dalam menemukan permasalahan kode yang *error*, bahkan *bug* dan *error* yang tersembunyi sebelumnya.
- 3) Meningkatkan perhatian *programmer* saat akan mengimplementasikan aplikasi untuk meminimalkan *error*.

b. Beberapa kekurangan *white box testing*

- 1) Setiap kode perlu diubah. Kalau penguji mengubah kode atau menambahkan kode baru, maka *programmer* harus mengulangi pengujian dari awal.
- 2) Tidak dapat digunakan untuk menguji fungsi dari perangkat lunak dikarenakan penekanannya pada pengujian *black box testing*. Metode pengujiannya cukup kompleks, jadi penguji benar-benar perlu menguasai tiap langkah-langkahnya.

## II.2 Penelitian Terdahulu (*State of The Art*)

Tabel II.1 Penelitian Terdahulu/*State of The Art*

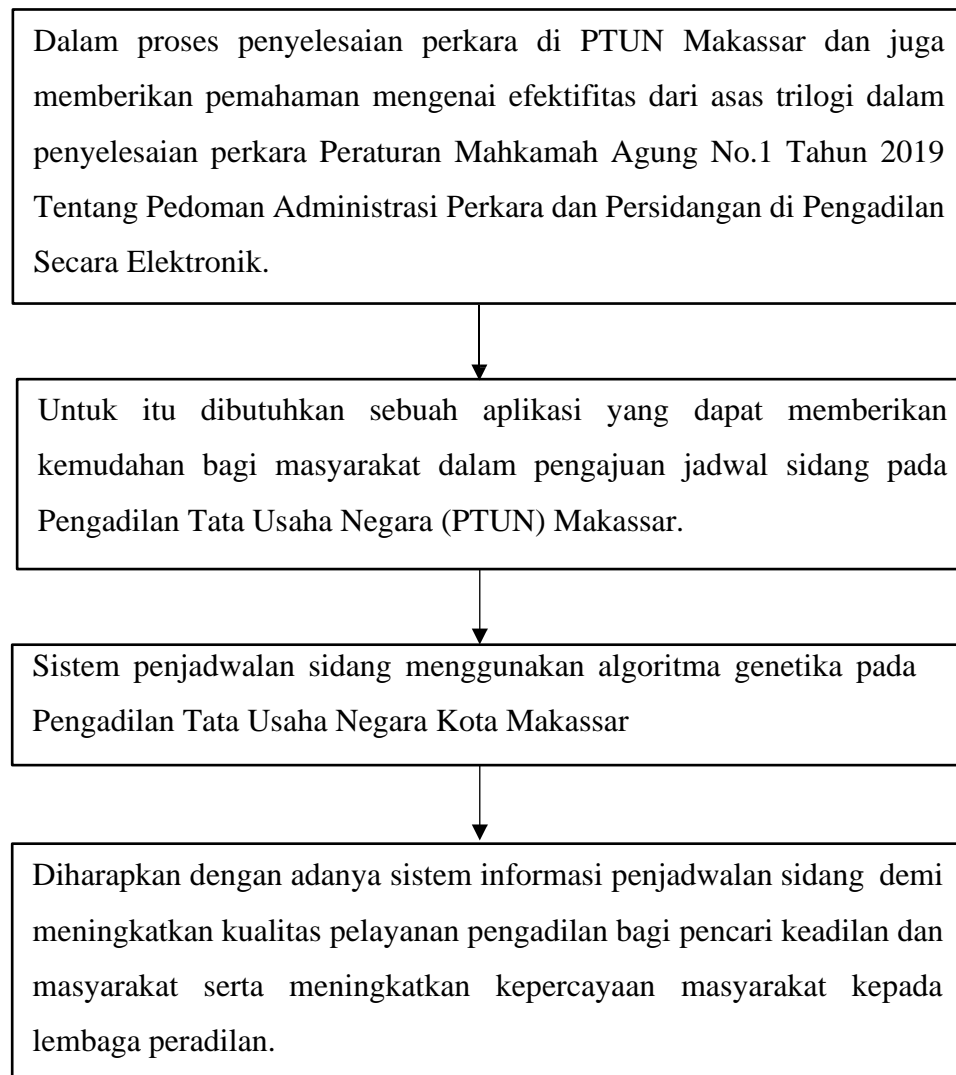
No	Peneliti	Tahun	Judul	Metode	Hasil
1.	I Made Budi Adnyana dan I Komang Wijayana	2017	Rancang Bangun Sistem Penjadwalan Sidang Skripsi Dengan Algoritma Genetika	Algoritma Genetika	Menyelesaikan penjadwalan sidang skripsi di STIKOM Bali secara otomatisasi sehingga dapat menghasilkan jadwal sidang skripsi yang optimal
2	Amin Nugroho	2021	Rancang Bangun Sistem Informasi Pejadwalan Pelaksanaan Sidang Skripsi Dengan Metode Algoritma Genetika Berbasis Web	Algoritma Genetika	Mempercepat proses kegiatan penjadwalan dan dengan adanya informasi yang didapatkan dari web mempermudah mahasiswa dan dosen untuk mengetahui jadwal yang telah diajukan
3	Inggrid Adelia	2021	Implementasi <i>E-Court</i>	Algoritma Genetika	Memberikan tingkat kepuasan

No	Peneliti	Tahun	Judul	Metode	Hasil
			Dalam Proses Penyelesaian Perkara di Pengadilan Tata Usaha Negara		dari segi efisiensi waktu dan biaya yang dikeluarkan oleh pihak penggugat
4.	Rubiyanto Maku	2021	Perancangan Sistem Penjadwalan Kuliah dan Ujian Skripsi Program Studi Sistem Informasi Berbasis Web	Algoritma Genetika	Sistem jadwal dan ruangan ujian menjadi efisien dan efektif.
5.	Dwi Oktarina	2019	Perancangan Sistem Penjadwalan Seminar Proposal dan Sidang Skripsi Dengan Metode Algoritma Genetika	Algoritma Genetika	Mempercepat proses kegiatan penjadwalan dan dengan adanya informasi yang didapatkan dari web mempermudah mahasiswa dan dosen untuk mengetahui jadwal yang telah diajukan, dan daftar

No	Peneliti	Tahun	Judul	Metode	Hasil
					jadwal menguji bagi dosen.

### II.3 Kerangka Pikir

Untuk memudahkan kegiatan penelitian yang dilakukan serta bagi memperjelas akar pemikiran dalam penelitian ini, berikut merupakan kerangka pemikiran penelitian yang akan dilakukan, sebagai berikut:

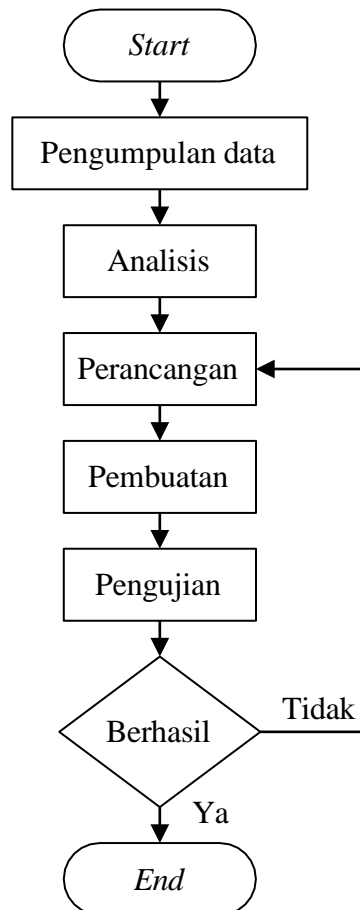


Gambar II.7 Kerangka Pikir

## BAB III METODOLOGI PENELITIAN

### III.1 Bagan Alur Penelitian/Tahapan Penelitian

Tahapan penelitian yang dilakukan dalam penelitian ini dapat dilihat pada gambar III.1.



Gambar III.1 Tahapan Penelitian

Penjelasan dari tahapan penelitian pada gambar III.1, yaitu:

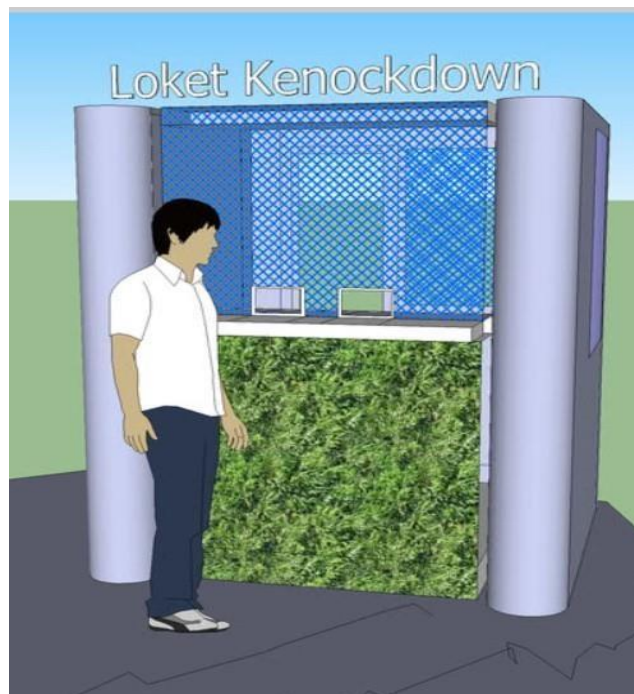
1. *Start*.
2. Pengumpulan data, pengumpulan data dilakukan dengan studi pustaka dan observasi. Studi pustaka dilakukan dengan mempelajari algoritma genetika. Observasi dilakukan dengan pengamatan langsung di Pengadilan Tata Usaha Negara (PTUN) Makassar.



3. Analisis, mengolah data untuk menemukan data mana yang dibutuhkan oleh sistem yang akan dibangun.
4. Perancangan, setelah pengumpulan data dan menganalisis maka dilakukan perancangan sistem, perancangan sistem menggunakan UML meliputi *use case diagram*, *activity diagram*, *sequence diagram*, dan *class diagram*.
5. Pembuatan, tahap ini mengimplementasikan hasil perancangan ke dalam sistem menggunakan bahasa pemrograman PHP dan mengimplementasikan algoritma genetika untuk penjadwalan sidang.
6. Pengujian sistem, pengujian dilakukan dengan teknik *black box testing* dan *white box testing* untuk memastikan bahwa sistem sesuai dengan rancangan sistem yang dibangun sebelumnya. Jika tidak sesuai maka dilakukan pengecekan kembali pada perancangan.
7. *End.*

## III.2 Perancangan Sistem

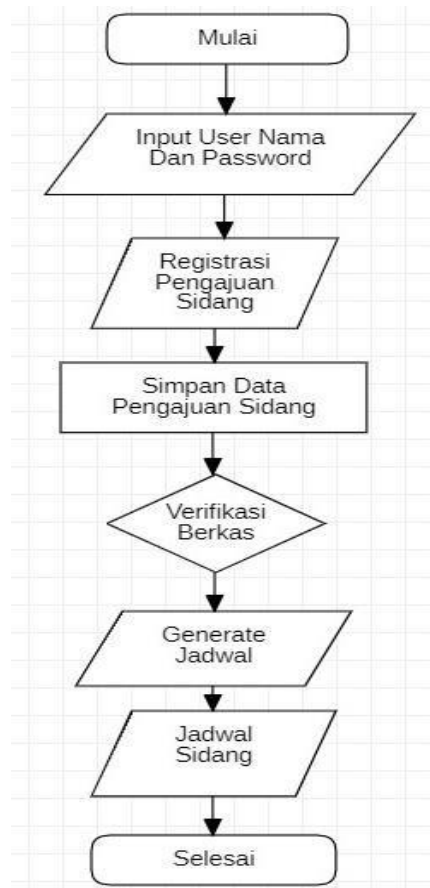
### III.2.1 Analisis Sistem yang Berjalan



Gambar III.2 Sistem yang Sedang Berjalan

Sistem yang berjalan saat ini untuk penjadwalan sidang Pengadilan Tata Usaha Negara Kota Makassar masih dilakukan dengan secara manual, yaitu penggugat melakukan pengajuan gugatan perkara di administrator PTUN, setelah melangkapi semua berkas dan menerima perkara. Administrator memverifikasi dan memberikan surat lampiran pengaduan. Jika penggugat ingin melanjutkan ke persidangan maka kembali ke staf PTUN Makassar untuk melakukan pengajuan sidang, selanjutnya staf menyimpan pengajuan dan selanjutnya dirapatkan dengan hakim perihal jadwal sidang.

### III.2.2 Analisis Sistem yang Diusulkan

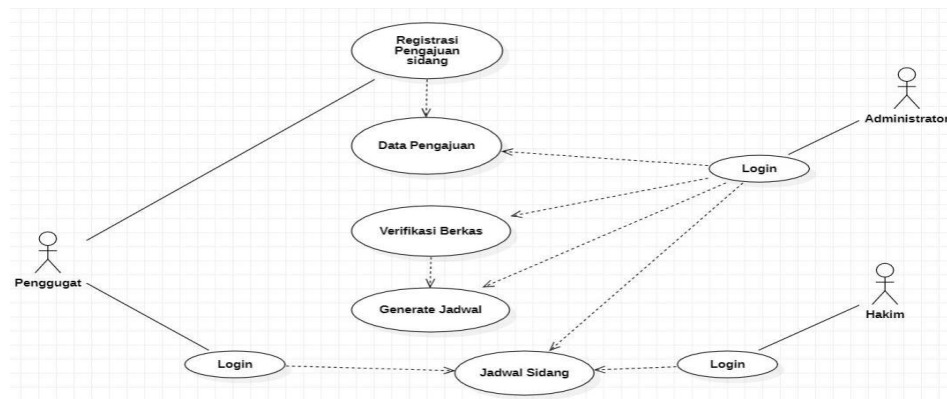


Gambar III.3 Sistem yang Diusulkan

Sistem yang diusulkan adalah sebuah *website* penjadwalan sidang di mana user melakukan pengajuan sidang secara *online*.

## 1. Use Case Diagram

*Use case* diagram merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara 1 atau lebih aktor dengan sistem informasi yang akan dibuat secara kasar, digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.



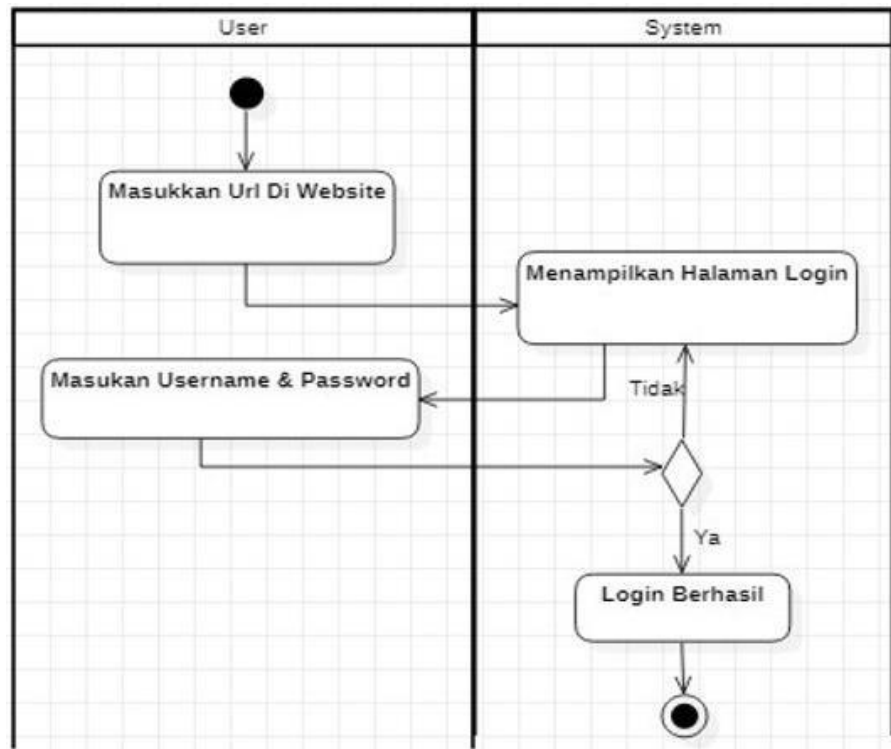
Gambar III.4 Use Case Diagram

Sistem ini memiliki 3 aktor yaitu: penggugat, hakim, dan administrator.

## 2. Activity Diagram

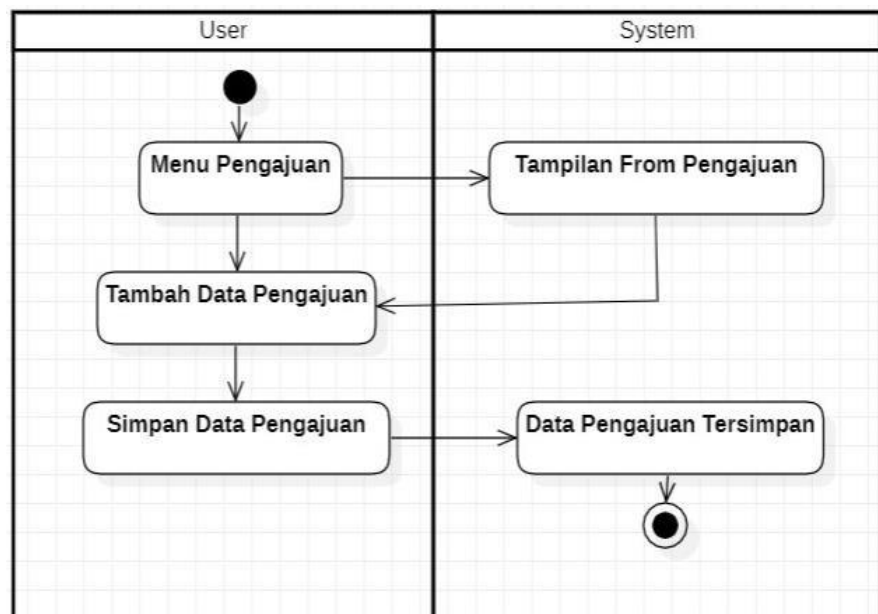
*Activity diagram* adalah *representasi grafis* dari alur kerja dari kegiatan, aktivitas atau tindakan serta pelaku aktivitas secara berurutan. *Activity diagram* digunakan untuk menunjukkan aliran pesan dari satu aktivitas ke aktivitas lainnya.

a. *Activity Diagram Login Akun Pengguna*



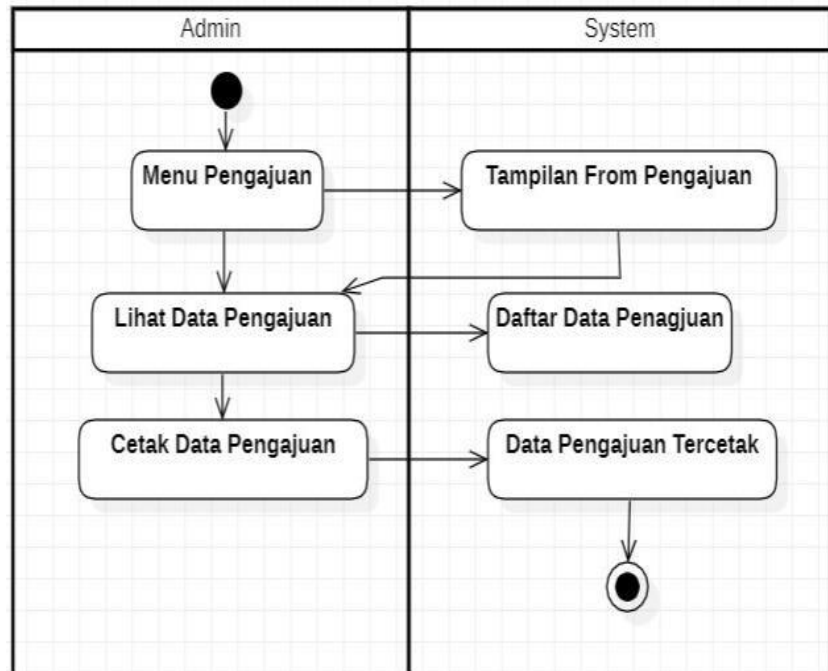
Gambar III.5 *Activity Diagram Login Akun Pengguna*

b. *Activity Diagram Registrasi Pengajuan Pada Penggugat*



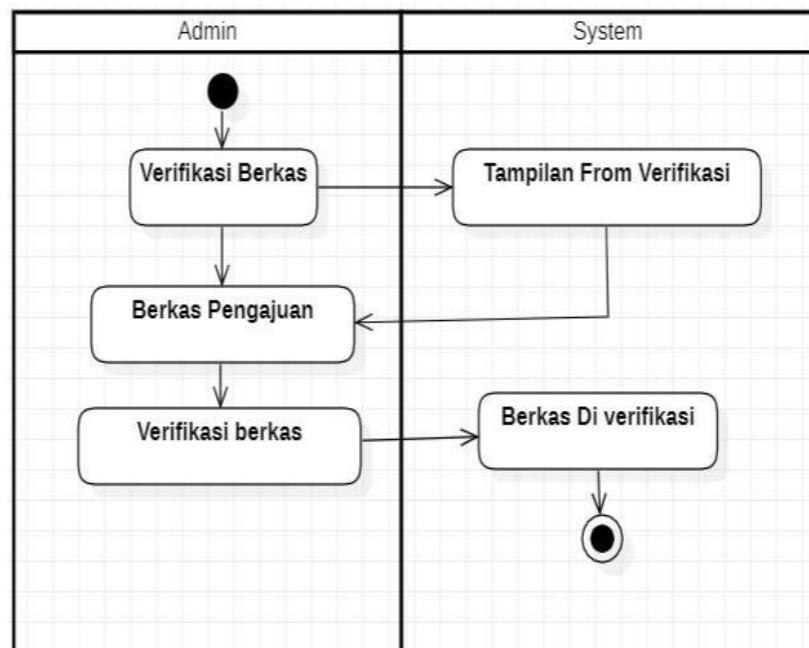
Gambar III.6 *Activity Diagram Registrasi Pengajuan Pada Penggugat*

c. *Activity Diagram* Lihat Data Pengajuan Pada Admin



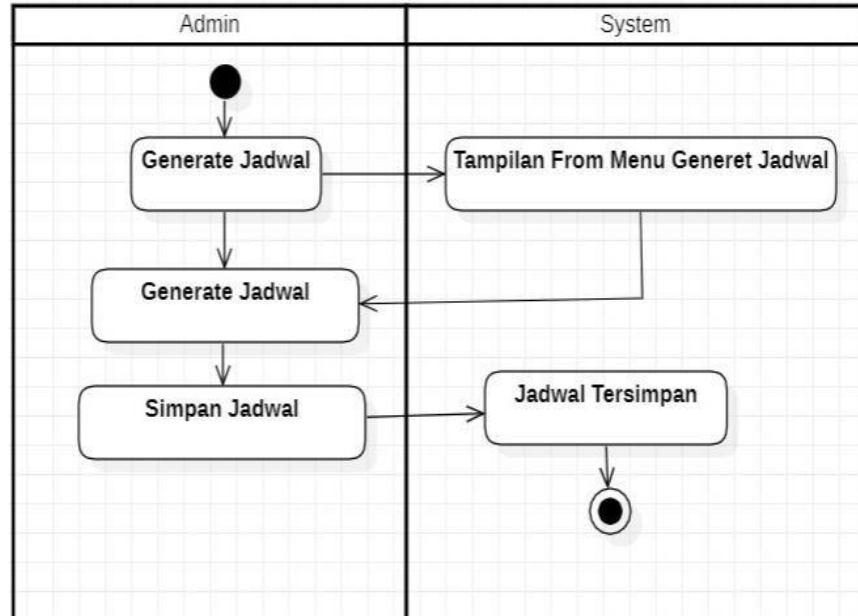
Gambar III.7 *Activity Diagram* Lihat Data Pengajuan Pada Admin

d. *Activity Diagram* Verifikasi Berkas Pada Admin



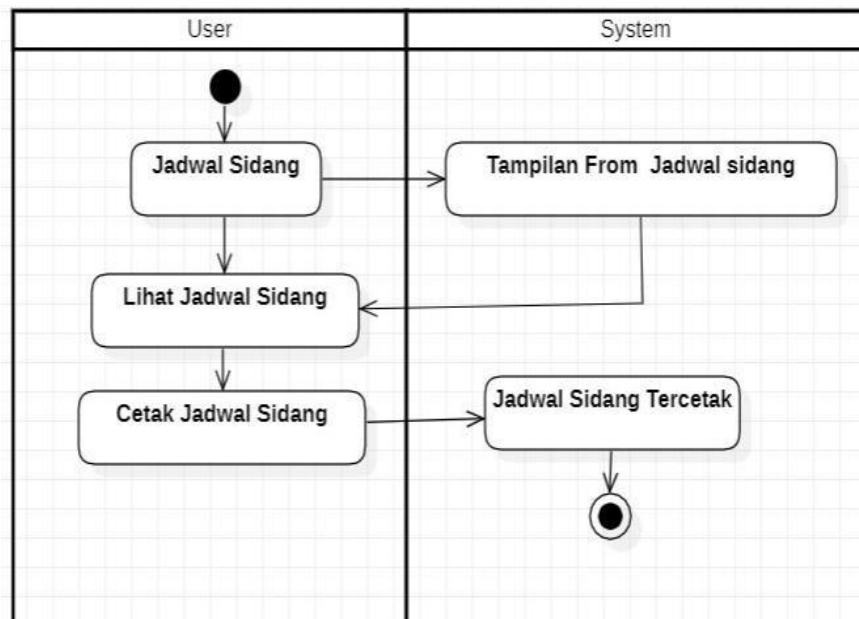
Gambar III.8 *Activity Diagram* Verifikasi Berkas Pada Admin

e. *Activity Diagram Generate Jadwal Pada Admin*



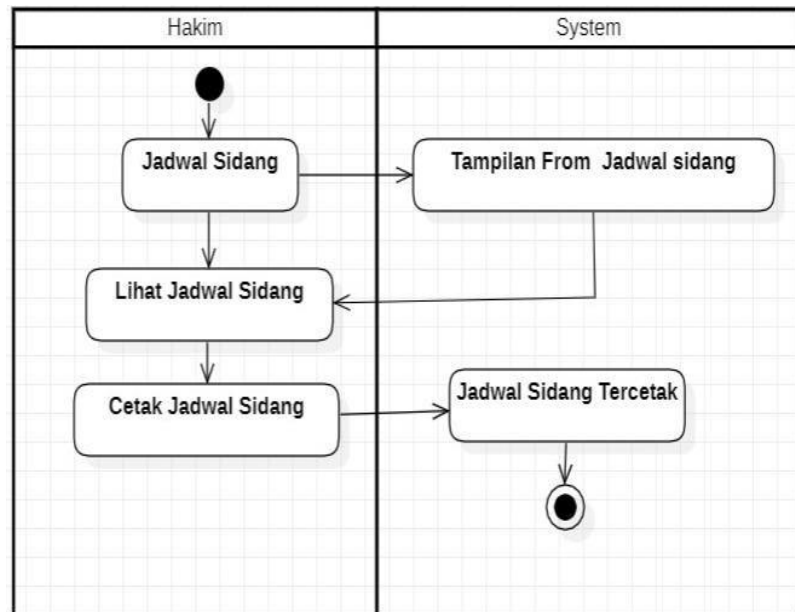
Gambar III.9 *Activity Diagram Generate Jadwal Pada Admin*

f. *Activity Diagram Cetak Jadwal Sidang Pada Penggugat*



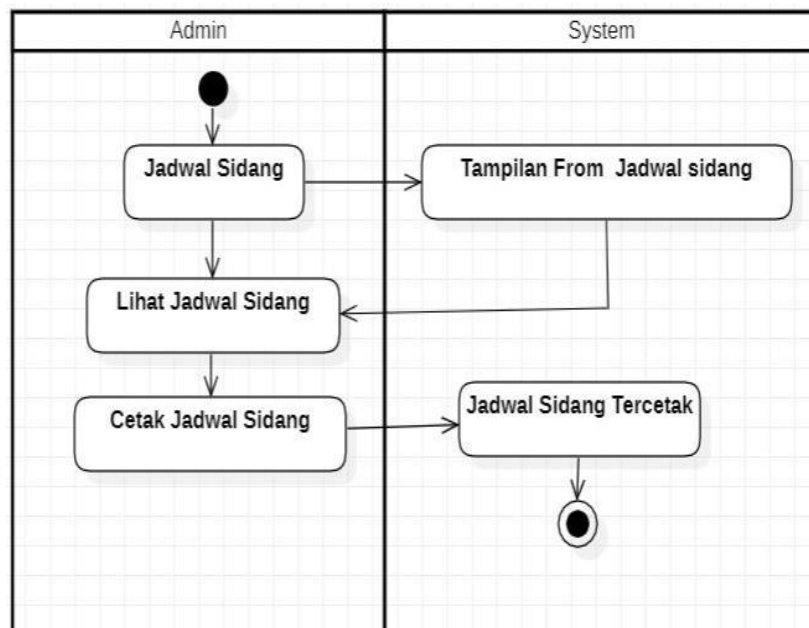
Gambar III.10 *Activity Diagram Cetak Jadwal Sidang Pada Penggugat*

g. *Activity Diagram* Cetak Jadwal Sidang Pada Hakim



Gambar III.11 *Activity Diagram* Cetak Jadwal Sidang Pada Hakim

h. *Activity Diagram* Cetak Jadwal Sidang Pada Admin

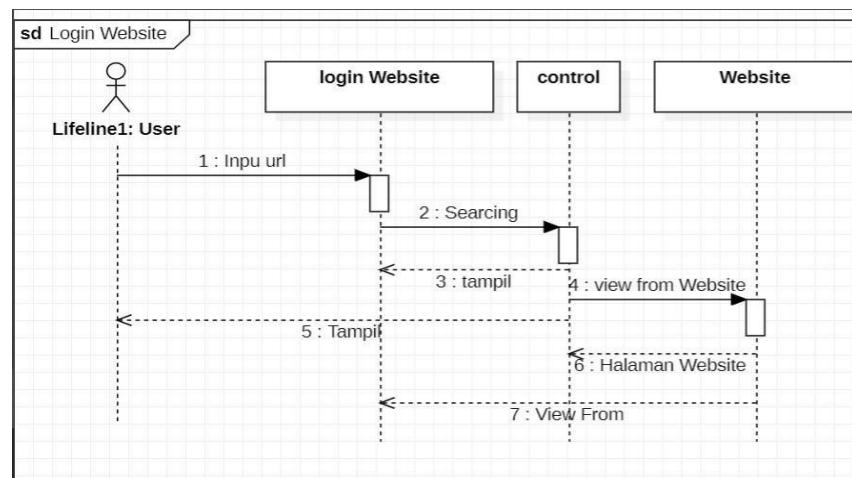


Gambar III.12 *Activity Diagram* Cetak Jadwal Sidang Pada Admin

### 3. Sequence Diagram

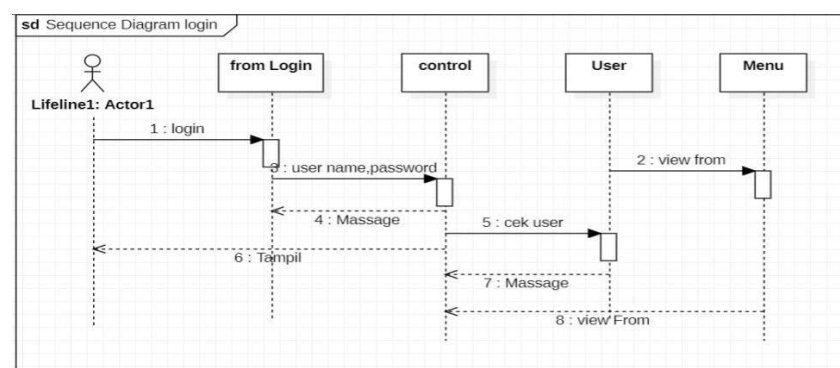
*Sequence Diagram* adalah suatu diagram yang menjelaskan interaksi objek dan menunjukkan (memberi tanda atau petunjuk) komunikasi diantara objek-objek tersebut. *Sequence* diagram digunakan untuk menjelaskan perilaku pada sebuah skenario dan menggambarkan bagaimana entitas dan sistem berinteraksi, termasuk pesan yang dipakai saat interaksi. Semua pesan digambarkan dalam urutan pada eksekusi.

#### a. Sequence Diagram Mengakses Sistem Penjadwalan Persidangan



Gambar III.13 *Sequence Diagram* Mengakses Sistem Penjadwalan Persidangan

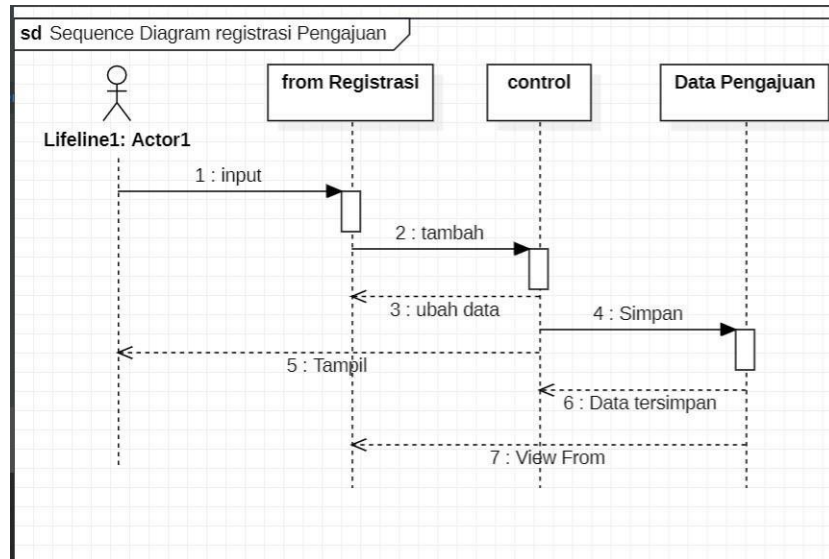
#### b. Sequence Diagram Login Akun Pengguna



Gambar III.14 *Sequence Diagram* Login Akun Pengguna

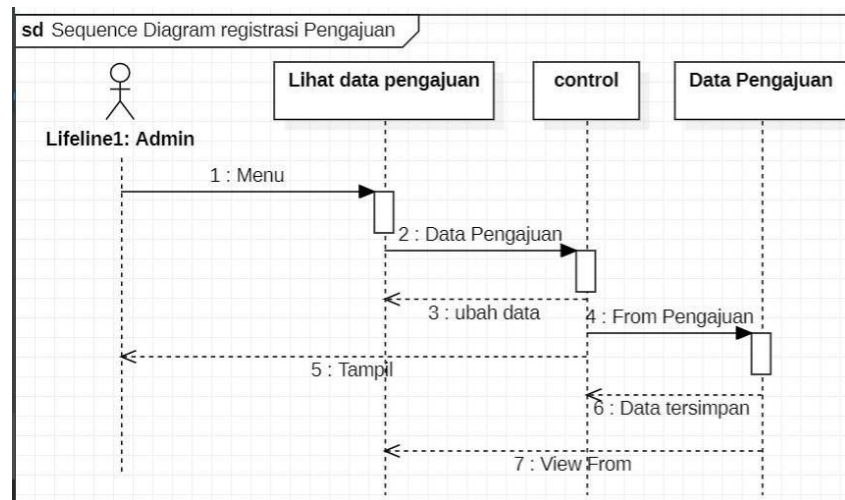


c. *Sequence Diagram* Registrasi Pengajuan Pada Penggugat



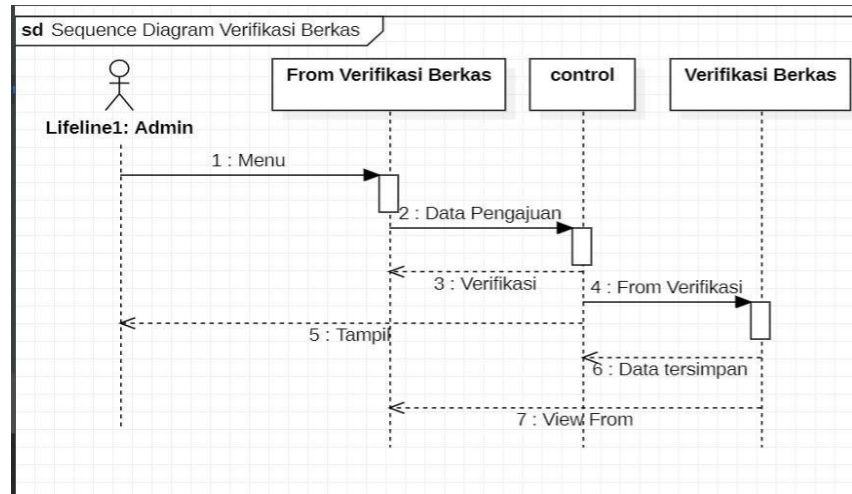
Gambar III.15 *Sequence Diagram* Registrasi Pengajuan Pada Penggugat

d. *Sequence Diagram* Lihat Data Pengajuan Pada Admin



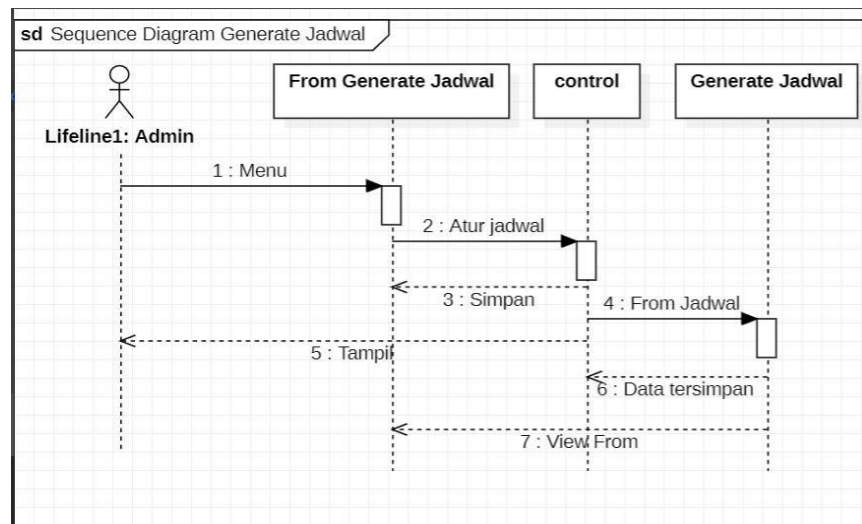
Gambar III.16 *Sequence Diagram* Lihat Data Pengajuan Pada Admin

e. *Sequence Diagram Verifikasi Berkas Pada Admin*



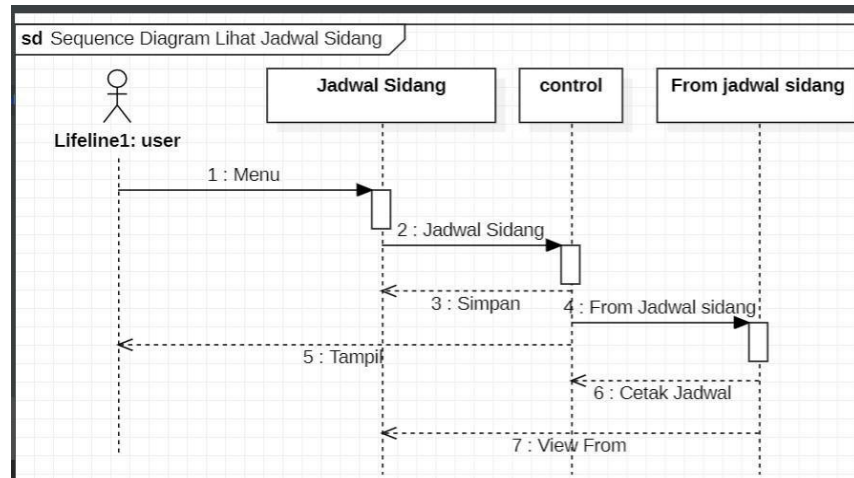
Gambar III.17 *Sequence Diagram* Verifikasi Berkas Pada Admin

f. *Sequence Diagram Generate Jadwal Pada Admin*



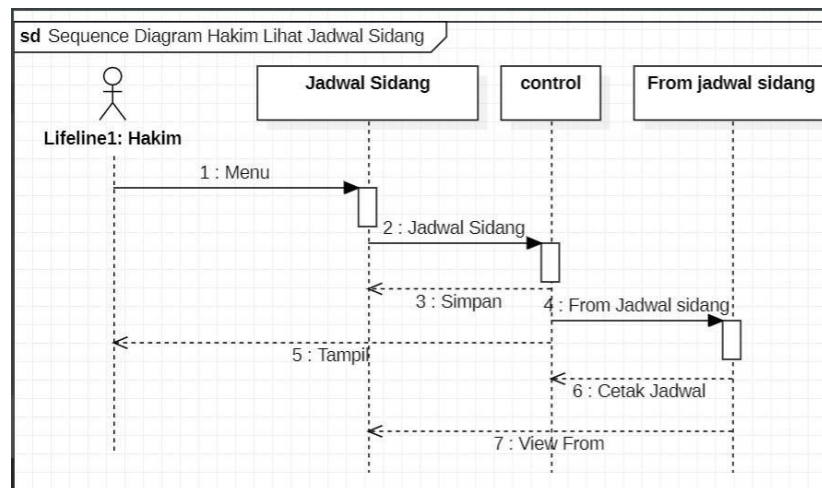
Gambar III.18 *Sequence Diagram* Generate Jadwal Pada Admin

g. *Sequence Diagram* Cetak Jadwal Sidang Pada Penggugat



Gambar III.19 *Sequence Diagram* Lihat Jadwal Sidang Pada Penggugat

h. *Sequence Diagram* Cetak Jadwal Sidang Pada Hakim



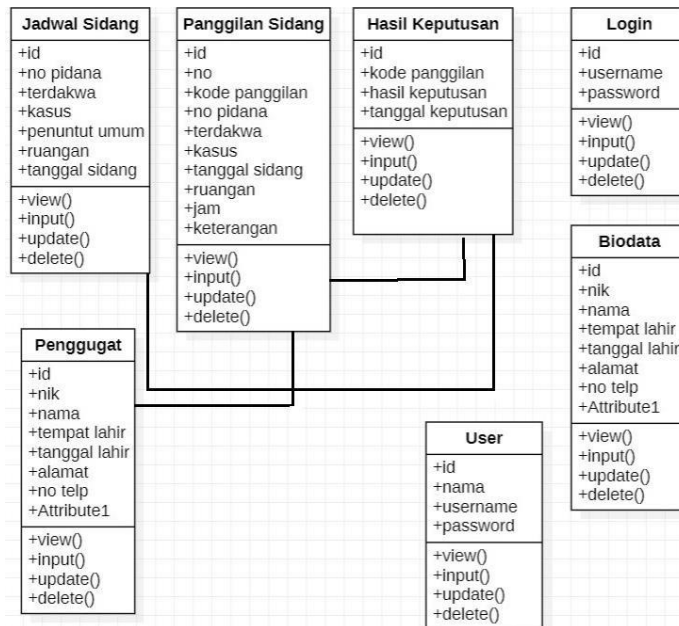
Gambar III.20 *Sequence Diagram* Cetak Jadwal Sidang Pada Hakim

4. *Class Diagram*

*Class diagram* adalah sebuah *class* yang menggambarkan struktur dan penjelasan *class*, paket, dan objek serta hubungan satu sama lain.

- Komponen atas: komponen yang satu ini memuat nama *class*. Tiap *class* sudah pasti akan memiliki nama yang berbeda. Istilah lain untuk penamaan ini adalah *simple name* atau nama sederhana.

- b. Komponen tengah: komponen ini memuat atribut *class* yang berguna untuk menjabarkan kualitas dari *class*. Atribut bisa menjelaskan lebih detail dengan cara menginput tipe nilainya.
- c. Komponen bawah: komponen ini menyematkan operasi yang terlihat dalam bentuk *list* atau daftar. Operasi ini memvisualisasikan bagaimana *class* berinteraksi dengan data

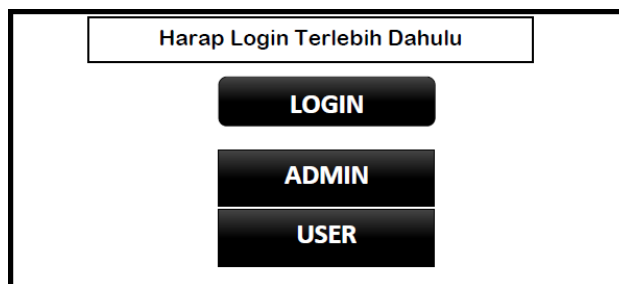


Gambar III.21 *Class Diagram*

## 5. Desain Interface

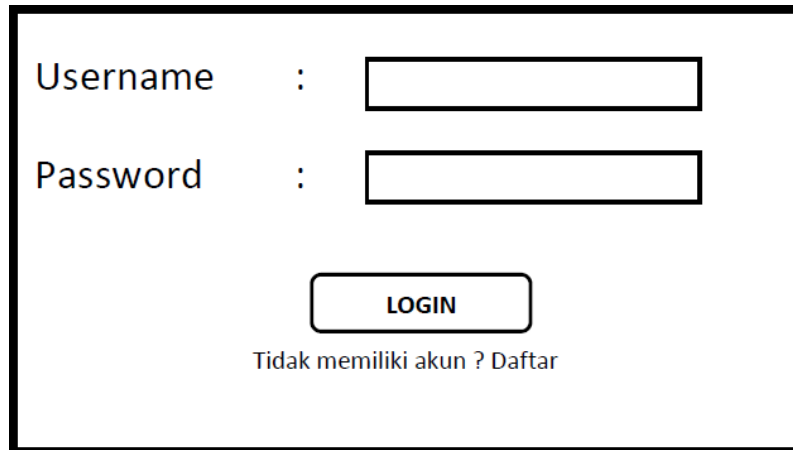
Pada pembuatan program ini memiliki rancangan desain konseptual sebagai berikut:

- a. Halaman *Login*



Gambar III.22 Halaman *Login*

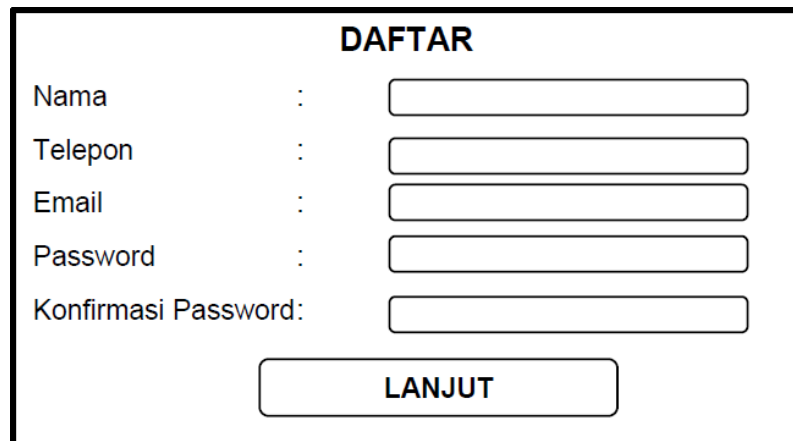
b. *Form Login*



A login form with a black border. It contains two input fields: 'Username' and 'Password', each followed by a colon and a rectangular input box. Below these fields is a rounded rectangular button labeled 'LOGIN'. At the bottom, there is a link that says 'Tidak memiliki akun ? Daftar'.

Gambar III.23 *Form Login*

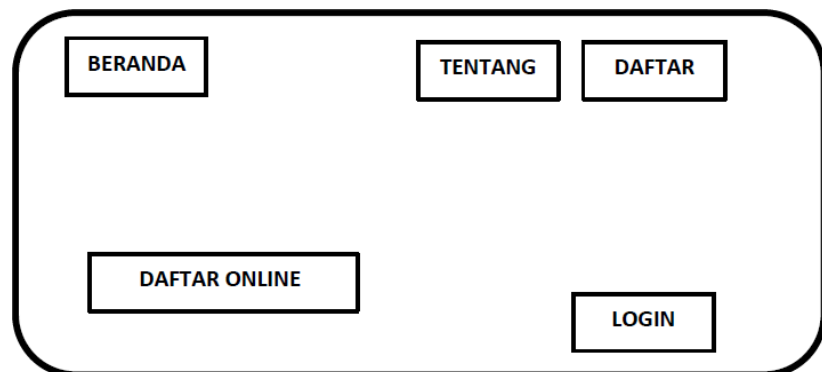
c. *Form Daftar*



A registration form titled 'DAFTAR' in bold. It has a black border and contains five input fields: 'Nama', 'Telepon', 'Email', 'Password', and 'Konfirmasi Password', each followed by a colon and a rectangular input box. Below these fields is a rounded rectangular button labeled 'LANJUT'.

Gambar III.24 *Form Daftar*

d. *Halaman Utama Sistem*



A main page layout with a rounded rectangular border. It features five buttons: 'BERANDA' (top left), 'TENTANG' (top middle), 'DAFTAR' (top right), 'DAFTAR ONLINE' (bottom left), and 'LOGIN' (bottom right).

Gambar III.25 *Halaman Utama Sistem*

e. Halaman Data Hakim

**DATA HAKIM**

No	Foto	NIP	Nama	Jabatan	Gol/Pangkat	Action

Gambar III.26 Halaman Data Hakim

f. Halaman Data Penggugat

**DATA PENGGUGAT**

No	Nama Penggugat	Nama Tergugat	Jenis Sidang	Search	Action

Gambar III.27 Halaman Data Penggugat

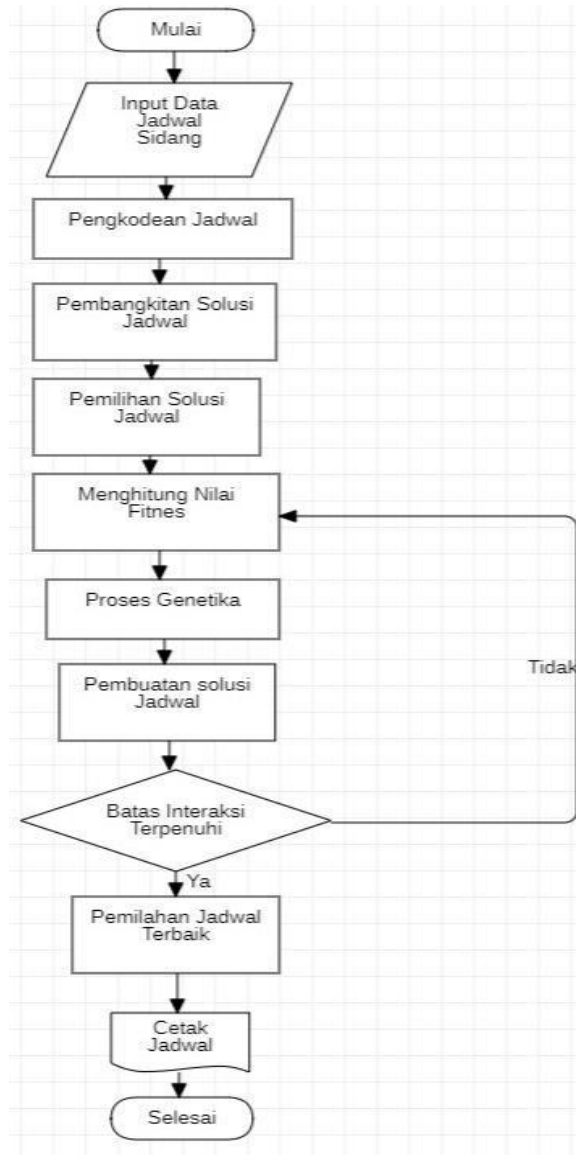
g. Halaman Jadwal Sidang

**DATA JADWAL**

No	Waktu Sidang	Nama	Keterangan Hakim sidang	Ruangan	Action

Gambar III.28 Halaman Jadwal Sidang

## 6. Diagram Proses Algoritma Genetika Pada Sistem



Gambar III.29 Diagram Proses Algoritma Genetika

Batasan penjadwalan yang digunakan dalam penelitian ini diantaranya, yaitu:

- Penjadwalan sidang disediakan 4 hari, yaitu: Senin, Selasa, Rabu, dan Kamis.
- Waktu sidang disediakan 4 slot jam dalam sehari, yaitu: 08.30–10.30, 10.30–13.0, 13.00–15.30, dan 15.30–17.30.
- Ruang persidangan 3, yaitu: ruangan 1, ruangan 2, dan ruangan 3.
- Setiap sidang terdapat 3 orang yang bertugas.

e. Hanya ada 13 hakim yang dapat bertugas.

## 7. Algoritma Genetika dalam Memecahkan Masalah

a. Kasus dalam penjadwalan sidang, algoritma genetika berfungsi untuk menentukan waktu (hari dan jam) serta ruang yang sesuai untuk setiap penggugat sehingga mempermudah jalannya persidangan berikut data yang dibutuhkan:

- 1) Data waktu, mencatat hari dan jam yang tersedia dalam 1 minggu.
- 2) Data ruang, berisi daftar ruang yang bisa digunakan.

b. Algoritma genetika penjadwalan penentuan nilai *fitness*. Dalam kasus penjadwalan, *fitness* ditentukan oleh:

- 1) *Clash Waktu* (CW), jumlah jadwal yang ruang yang sama di waktu yang sama.
- 2) *Clash Ruang* (CR), jumlah jadwal yang ruang yang sama di waktu yang sama.

Sehingga perhitungan *fitness* adalah:

$$F = 1 / (1 + CR + CW)$$

*Fitness* terbaik adalah *fitness* yang bernilai paling besar sehingga nilainya harusnya 1. Semakin banyak ruang maka *fitness* akan semakin kecil. Dalam beberapa kasus Setelah penentuan aturan nilai *fitness*, maka dilanjutkan dengan langkah algoritma genetika:

### Pembangkitan Generasi Awal

Sebelum pembangkitan generasi awal, ada beberapa pengaturan yang harus dibuat ketika merancang aplikasi diantaranya: jumlah kromosom. Kromosom adalah Kumpulan gen-gen (kombinasi antara kuliah, ruang, dan waktu) sejumlah kuliah yang ada. Di awal ditentukan jumlah kromosom yang dibangkitkan. Satu kromosom ada satu solusi, sehingga semakin banyak kromosom yang dibangkitkan semakin banyak pilihan solusi (jadwal) yang dihasilkan, yang nantinya akan dipilih satu yang terbaik.



Kromosom akan disimpan ke dalam sebuah *array* dengan format seperti berikut:

$$\text{Kromosom 1} = \{[0, 3, 5], [1, 5, 13], \dots [10, 4, 0]\}$$

Dapat dilihat kromosom 1 memiliki 30 gen yang artinya terdapat 10 ruang setiap gen adalah kombinasi [ruang, waktu]. Dalam gen data dimasukkan dalam angka (indeks dimulai dari 0). Gen [0, 3, 5] artinya kuliah ke-1, ruang ke-3, dan waktu ke-5. Tapi harus diperhatikan semakin banyak kromosom dibangkitkan maka semakin berat dan lambat proses *generate* jadwal.

1) **Maksimal Generasi**

Proses algoritma akan berhenti jika sudah mencapai nilai *fitness* 1 (tidak ada bentrok). Jika masih bentrok akan diulang lagi generasi berikutnya. Terkadang dalam *generate* jadwal bisa saja tidak menemukan *fitness* 1, hal ini bisa terjadi karena jumlah kuliah banyak sedangkan ruang dan waktu sedikit dan tidak memungkinkan untuk membuat jadwal yang tidak bentrok. Untuk mengatasi hal inilah diatur maksimal generasi sehingga program tidak akan berhenti jika sudah di generasi terakhir walaupun *fitness* belum mencapai 1.

2) **Crossover Rate**

Mengatur persentase kromosom yang akan dipindah silang (rentang 0–100).

3) **MutationRate**

Mengatur persentase gen yang akan diganti (mutasi).

c. **Seleksi**, proses seleksi adalah pemilihan kromosom yang mana yang akan digunakan untuk proses algoritma berikutnya. Seleksi ditentukan berdasarkan nilai *fitness* kromosom. Semakin besar *fitness* semakin besar kesempatan kromosom untuk terpilih.

Metode yang digunakan untuk seleksi adalah Roulette Wheel. Cara kerjanya adalah:

- 1) Hitung nilai *fitness* dari masing-masing individu. Misal terdapat 4 kromosom yang dibangkitkan dengan masing-masing *fitness*:  
 Kromosom1=0.8  
 Kromosom2=0.3  
 Kromosom3=0.5  
 Kromosom 4 = 0.4
- 2) Hitung total *fitness* dari semua individu.  
 Total *fitness* adalah  $0.8 + 0.3 + 0.5 + 0.4 = 2$
- 3) Hitung probabilitas masing-masing individu, probabilitas didapat dari nilai *fitness* dibagi dengan total *fitness*, hasilnya:  
 $P 1 = 0.8 / 2 = 0.4$   
 $P 2 = 0.3 / 2 = 0.15$   
 $P 3 = 0.5 / 2 = 0.25$   
 $P 4 = 0.4 / 2 = 0.2$
- 4) Dari probabilitas tersebut, hitung jatah masing-masing individu pada angka 1 sampai 100. Penentuan jatah dilakukan dengan mencari kumulatif dari probabilitas:  
 $PK 1 = 0 + 0.4 = 0.4$   
 $PK 2 = 0.4 + 0.15 = 0.55$   
 $PK 3 = 0.55 + 0.25 = 0.80$   
 $PK 4 = 0.80 + 0.2 = 1$
- 5) Bangkitkan bilangan acak antara 0–1 sejumlah kromosom.
- 6) Dari bilangan acak yang dihasilkan, tentukan individu mana yang terpilih dalam proses seleksi.
- 7) Misal bilangan pertama adalah 0.2 maka kromosom 1 terpilih (kromosom 1 antara 0–0.4), jika bilangan berikutnya adalah 0.7 maka kromosom 3 yang terpilih (kromosom 3 antara 0.55–0.80). Dapat dilihat bahwa kemungkinan besar kromosom 1 untuk terpilih lebih besar karena rentangannya nilainya paling panjang. Kromosom yang dipilih sebanyak jumlah kromosom

awal, tapi bisa saja ada kromosom yang sama terpilih dan ada kromosom yang tidak terpilih.

- d. Pindah silang (*Crossover*), pindah silang merupakan pertukaran gen antara dua buah kromosom. Kromosom yang menjadi induk dipilih secara acak sebanyak *crossover rate* yang sudah diatur diawal. Misal ada 10 kromosom dengan dengan CR 70%, maka induk yang digunakan adalah 7. Dari 7 induk ini akan dipasangkan dua dua yang menghasilkan 1 individu baru. Pasangan induk yang terjadi adalah 7 pasang yaitu, induk1 dan induk2, induk 2 dan induk 3, dan seterusnya sampe induk 7 dengan induk 1. Setiap induk akan berpasangan 2 kali.

Metode pindah silang yang digunakan adalah *one point crossover* di mana setiap pasangan akan dipilih 1 buah bilangan acak antara 1 sampe jumlah gen dalam kromosom. Misal dalam kromosom terdapat 50 gen, dan bilangan acak dibangkitkan nilainya 20, maka kromosom baru yang diciptakan adalah gen 0–19 dari induk1 dan 20–49 dari induk2, begitu juga untuk pasangan yang lain.

- e. Mutasi adalah penggantian gen. Gen yang dimutasi hanya diganti ruang, dan waktunya saja, untuk kuliahnya tetap. Jumlah gen yang diganti tergantung *Mutation Rate* (MR). Misal ada 5 kromosom dengan gen 10 masing-masing kromosom, dengan MR 50%. Maka:

$$\begin{aligned}\text{Total gen} &= (\text{jumlah kromosom} * \text{jumlah gen per kromosom}) \\ &= 5 * 10 \\ &= 50 \text{ gen}\end{aligned}$$

$$\text{Jumlah Mutasi} = 50\% * 50 = 25 \text{ gen.}$$

Cara mutasi adalah membangkitkan bilangan acak antara 1 sampai total gen (50) sebanyak 25 kali. Misal bilangan acak pertama adalah 12 maka akan diambil kromosom kedua gen ke dua (karena kromosom pertama hanya sampe 10, kurang lagi 2 diambil di kromosom ke 2). Sehingga ruang dan waktu gen kedua

dikromosom ke dua akan diganti dengan mengambil data ruang dan waktu secara acak.

### III.3 Waktu dan Tempat Penelitian

Waktu penelitian dilakukan selama 3 bulan, dimulai bulan November 2022 sampai bulan Maret 2023. Tempat penelitian dilakukan di Pengadilan Tata Usaha Negara.

### III.4 Alat dan Bahan Penelitian

Dalam penelitian ini, diperlukan baik perangkat lunak (*Software*) maupun perangkat keras (*Hardware*) yang mendukung jalannya penelitian.

#### 1. Perangkat Keras

Tabel III.1 Perangkat Keras

No	Nama	Spesifikasi
1.	Laptop	a. Menggunakan OS <i>Microsoft® Windows®</i> 7/8/10 (32- or 64-bit) b. Ram minimal 4 GB, yang direkomendasikan 8 GB RAM c. Ruang disk yang tersedia minimal 2 GB, namun yang direkomendasikan yaitu 4 d. Resolusi minimal yang digunakan adalah 1280 x 800

#### 2. Perangkat Lunak

Tabel III.2 Perangkat Lunak

No	Nama Perangkat Lunak	Versi
1	Sistem Operasi	Windows 10 Profesional
2	MySQL	8.0.31.0 64-Bit
3	PHP	7.1
4	XAMPP	1.8.1

No	Nama Perangkat Lunak	Versi
5	Javascript	4.8

### III.5 Metode Pengumpulan Data

Metode pengumpulan data yang digunakan dalam penulisan tugas akhir ini dengan mengadakan penelitian dan pengamatan langsung terhadap objek penelitian yaitu dengan cara:

1. Observasi

Observasi yang dilakukan dengan cara observasi langsung di lokasi penelitian dengan melakukan metode pengamatan langsung dan mengambil data yang dibutuhkan.

2. Studi Literatur

Dengan metode ini, dilakukan pengolahan data dari dokumen-dokumen yang sudah ada sebelumnya dan mendukung data penelitian seperti jurnal maupun buku, pengumpulan data ini digunakan sebagai mendukung kelengkapan data yang lain.

3. Wawancara

Wawancara adalah pengumpulan data yang dilakukan untuk mengajukan pertanyaan-pertanyaan tentang segala sesuatu kepada informan untuk memperoleh informasi yang diharapkan. Teknik wawancara ini dilakukan untuk melengkapi data yang terkait dalam penelitian.

### III.6 Metode Pengujian Sistem

Pengujian sistem dilakukan bertujuan untuk menguji kelayakan sistem yang dibuat. Pengujian dilakukan menggunakan metode *black box* dan menggunakan metode *white box*.

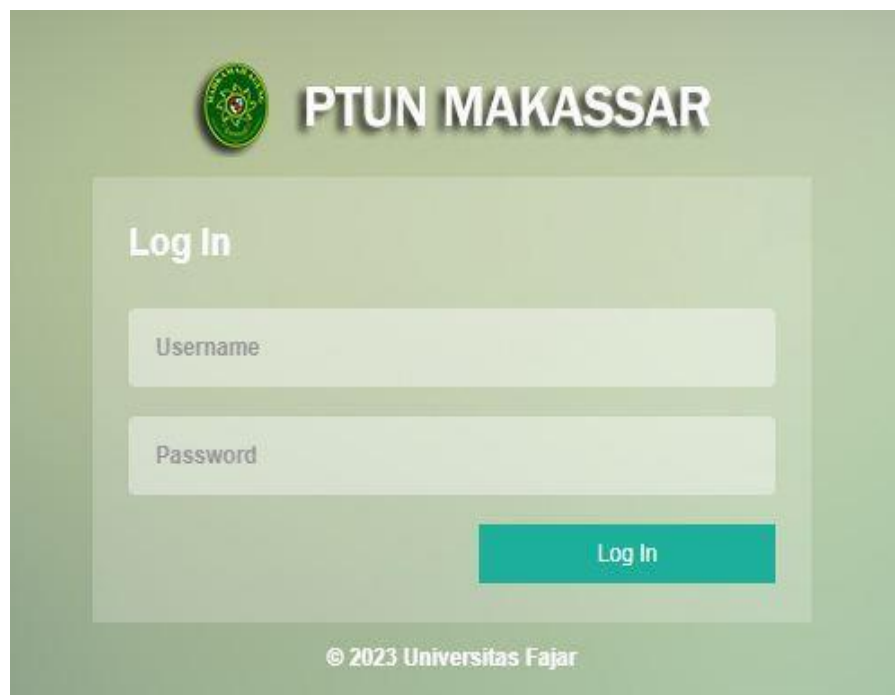
## BAB IV HASIL DAN PEMBAHASAN

### IV.1 Hasil Penelitian

Berdasarkan penelitian yang telah dilakukan, maka berikut adalah hasil yang telah diperoleh dari sistem yang telah dirancang untuk ditampilkan. Dalam mengakses sistem tersebut, pengguna yang berhak dibagi menjadi tiga yaitu admin, hakim, dan penggugat yang telah mempunyai akun. Berikut ini tampilan halaman-halaman yaitu:

1. Tampilan Halaman Login

*User yang login ke sistem ini, terlebih dahulu harus login dengan memasukkan username dan password. Setelah dinyatakan sukses melakukan login maka selanjutnya akan diarahkan menuju halaman login.*



Gambar IV.1 Tampilan Halaman *Login*

## 2. Tampilan Halaman Daftar

The screenshot shows a registration form with two columns: 'Data Penggugat' and 'Data Terlapor'. The 'Data Penggugat' column contains fields for 'No Pengaduan' (with a dropdown menu showing 'PG0016'), 'Nama Lengkap \*', 'Email \*', 'Teleponi/Hp \*', 'Alamat Domisili \*', 'Jenis Perkara \*' (with a dropdown menu showing 'Pilih'), 'Lampiran Pengaduan \*' (with a 'Pilih file' button and text 'Ukuran File max:50 mb' and 'Tipe file harus .jpg/.png'), and 'Keterangan \*'. The 'Data Terlapor' column contains fields for 'Nama Lengkap \*' and 'Alamat Domisili \*'. Each field has a small icon of a pencil and a red asterisk indicating it is required.

Gambar IV.2 Tampilan Halaman Daftar

Gambar IV.2 menunjukkan halaman daftar untuk *user* yang belum membuat akun, dalam *form* tersebut terdapat beberapa data yang harus diisi, antara lain nama lengkap, telpon hp, *email*, dan jenis perkara. Setelah data di *input*, tombol “Tambah” harus diklik agar akun tersimpan di *database* sehingga *user* dapat *login* ke web tersebut.

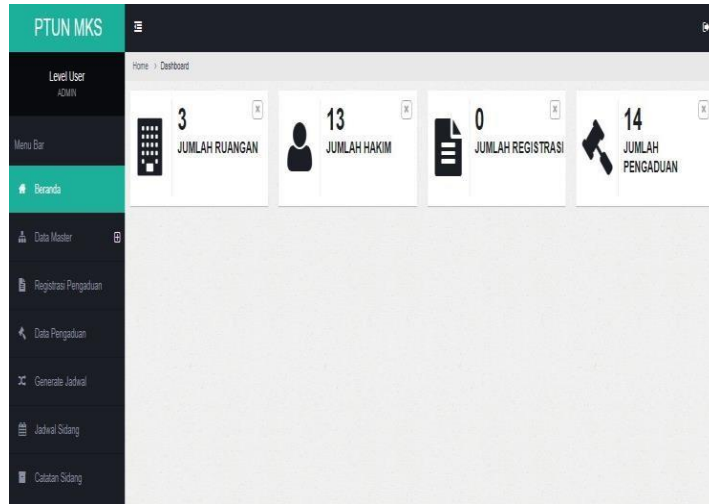
## 3. Tampilan Halaman Utama Sistem



Gambar IV.3 Tampilan Halaman Utama Sistem

Gambar IV.3 menunjukkan halaman utama (beranda) untuk *user* melakukan *registrasi* atau *login* untuk melihat jadwal sidang.

#### 4. Tampilan Halaman Utama Admin



Gambar IV.4 Tampilan Halaman Utama Admin

Gambar IV.4 menunjukkan halaman utama (beranda) untuk admin setelah berhasil melakukan *login*. Tampilan beranda admin berisi data master, registrasi pengajuan data pengajuan, *generate* jadwal, dan jadwal sidang. Pada menu juga disediakan tombol aksi untuk “tambah data pengajuan *user*”. Tombol tambah berfungsi untuk menambah data pengajuan.

#### 5. Tampilan Halaman Daftar Penggugat

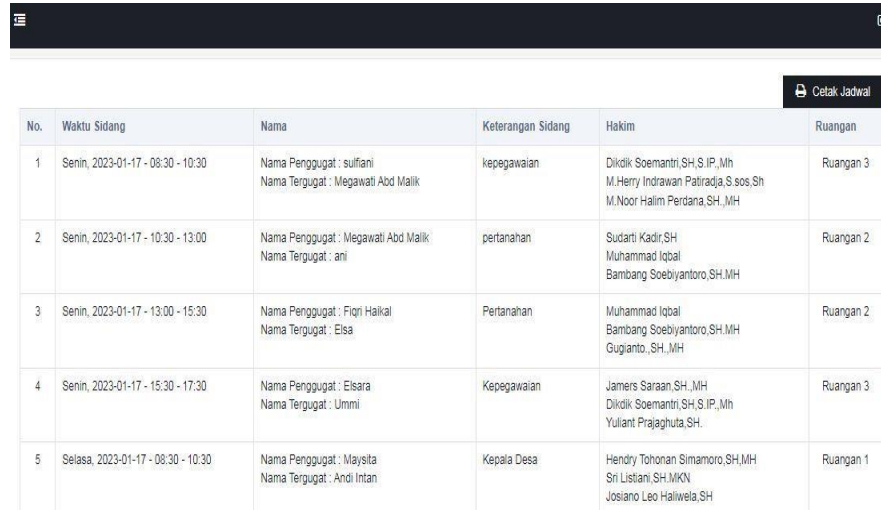
No.	No. Pengaduan	Nama Penggugat	Nama Tergugat	Jenis Sidang	Aksi
1	PGD015	abcd	sdhghf	Lelang	[+]
2	PGD014	dzuldar	linapriyanti	Pertanahan	[+]
3	PGD012	Muhammad Taufik	Ridho	Pertanahan	[+]
4	PGD011	suharfina	harfiana	Kepegawaian	[+]
5	PGD010	Fitri	Andika	Kepegawaian	[+]
6	PGD009	Andi Intan	Elisavera	Perzinan	[+]
7	PGD008	Muhammad Atfal	Muh Taufik	Kepegawaian	[+]
8	PGD007	Mayista	Andi Intan	Kepala Desa	[+]

Gambar IV.5 Tampilan Halaman Daftar Penggugat



Gambar IV.5 menunjukkan halaman daftar data penggugat yang telah melakukan *registrasi*, halama ini menyediakan tombol “Aksi” untuk mengubah dan menghapus data pengajuan.

## 6. Tampilan Halaman *Generate Jadwal*

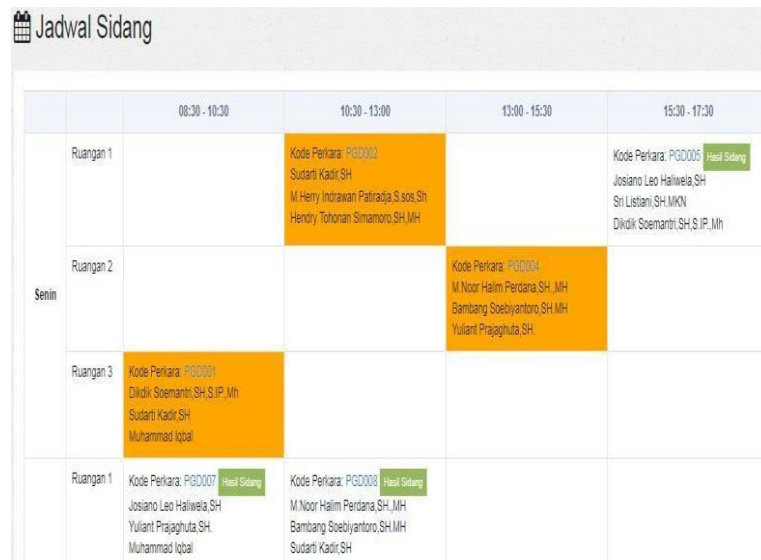


No.	Waktu Sidang	Nama	Keterangan Sidang	Hakim	Ruangan
1	Senin, 2023-01-17 - 08:30 - 10:30	Nama Penggugat : sullivan Nama Tergugat : Megawati Abd Malik	kepegawaian	Dikdik Soemantri,SH,S.IP.,Mh M.Herry Indrawan Patradja,S.sos,Sh M.Noor Halim Perdana,SH.,MH	Ruangan 3
2	Senin, 2023-01-17 - 10:30 - 13:00	Nama Penggugat : Megawati Abd Malik Nama Tergugat : anji	pertanahan	Sudarti Kadir,SH Muhammad Iqbal Bambang Soebiyantoro,SH,MH	Ruangan 2
3	Senin, 2023-01-17 - 13:00 - 15:30	Nama Penggugat : Fitri Haikal Nama Tergugat : Elsa	Pertanahan	Muhammad Iqbal Bambang Soebiyantoro,SH,MH Gugianto, SH, MH	Ruangan 2
4	Senin, 2023-01-17 - 15:30 - 17:30	Nama Penggugat : Elisara Nama Tergugat : Ummii	Kepegawaian	Jamers Saraan,SH.,MH Dikdik Soemantri,SH,S.IP.,Mh Yuliant Prajaghuta,SH.	Ruangan 3
5	Selasa, 2023-01-17 - 08:30 - 10:30	Nama Penggugat : Maysita Nama Tergugat : Andi Intan	Kepala Desa	Hendry Tohonan Simamoro,SH,MH Sri Listiani,SH,MKN Josiano Leo Halliwela,SH	Ruangan 1

Gambar IV.6 Tampilan Halaman *Generate Jadwal*

Gambar IV.6 menunjukkan halaman *generate* jadwal, halaman ini menyediakan tombol “Aksi” untuk *generate* jadwal dan cetak jadwal sidang.

## 7. Tampilan Halaman Jadwal Sidang



		08:30 - 10:30	10:30 - 13:00	13:00 - 15:30	15:30 - 17:30
Senin	Ruangan 1		Kode Perkara: PG0002 Sudarti Kadir,SH M.Herry Indrawan Patradja,S.sos,Sh Hendry Tohonan Simamoro,SH,MH		Kode Perkara: PG0005 Hasil Sidang Josiano Leo Halliwela,SH Sri Listiani,SH,MKN Dikdik Soemantri,SH,S.IP.,Mh
	Ruangan 2			Kode Perkara: PG0004 M.Noor Halim Perdana,SH.,MH Bambang Soebiyantoro,SH,MH Yuliant Prajaghuta,SH	
	Ruangan 3	Kode Perkara: PG0001 Dikdik Soemantri,SH,S.IP.,Mh Sudarti Kadir,SH Muhammad Iqbal			
	Ruangan 1	Kode Perkara: PG0007 Hasil Sidang Josiano Leo Halliwela,SH Yuliant Prajaghuta,SH Muhammad Iqbal	Kode Perkara: PG0003 Hasil Sidang M.Noor Halim Perdana,SH.,MH Bambang Soebiyantoro,SH,MH Sudarti Kadir,SH		

Gambar IV.7 Tampilan Halaman Jadwal Sidang

## IV.2 Pembahasan

Adapun pembahasan yaitu membahas mengenai pengujian sistem pada penelitian yang dilakukan menggunakan metode 2 pengujian sistem yaitu *black box* dan *white box*.

### 1. Pengujian *Black Box*

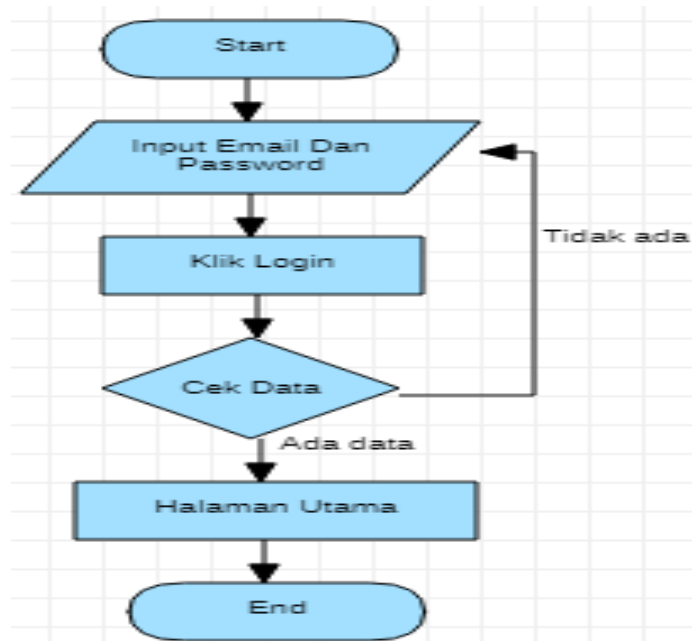
Tabel IV.1 Hasil Pengujian *Black Box*

Kasus dan Hasil Uji				
No	Komponen yang Diujikan	Skenario	Hasil yang Diharapkan	Hasil Pengujian
1	Login admin	Username, dan password benar	Sistem menerima akses <i>login</i> kemudian menampilkan halaman utama (beranda)	[ <input checked="" type="checkbox"/> ] Berfungsi [ ] Tidak Berfungsi
		Username, dan password salah	Sistem menolak akses <i>login</i> , maka tampilan tetap berada di halaman <i>login</i>	[ <input checked="" type="checkbox"/> ] Berfungsi [ ] Tidak Berfungsi
2	Login hakim	Username, dan password benar	Sistem menerima akses <i>login</i> dan kemudian menampilkan halaman utama (beranda)	[ <input checked="" type="checkbox"/> ] Berfungsi [ ] Tidak Berfungsi
		Username, dan	Sistem menolak akses <i>login</i> , tampilan tetap	[ <input checked="" type="checkbox"/> ] Berfungsi [ ] Tidak Berfungsi

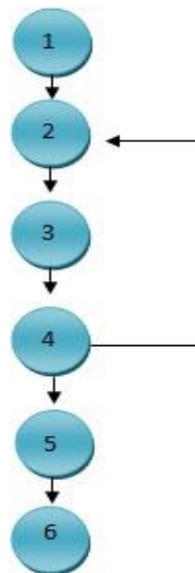
Kasus dan Hasil Uji				
No	Komponen yang Diujikan	Skenario	Hasil yang Diharapkan	Hasil Pengujian
		<i>password</i> salah	berada di halaman <i>login</i>	
3	<i>Login user</i>	<i>Username,</i> dan <i>password</i> salah	Sistem menolak akses <i>login</i> , maka tampilan tetap berada di halaman <i>login</i>	<input checked="" type="checkbox"/> Berfungsi <input type="checkbox"/> Tidak Berfungsi
		<i>Username,</i> dan <i>password</i> salah	Sistem menolak akses <i>login</i> , tampilan tetap berada di halaman <i>login</i>	<input checked="" type="checkbox"/> Berfungsi <input type="checkbox"/> Tidak Berfungsi
4	Menu beranda pada admin	Klik menu daftar pengajuan	Sistem akan menampilkan <i>form</i> daftar pengajuan yang berisi tombol aksi konfirmasi pengajuan	<input checked="" type="checkbox"/> Berfungsi <input type="checkbox"/> Tidak Berfungsi
		Klik tombol tolak yang tampil pada daftar pengajuan di <i>form</i> berkas	Sistem akan menerima dan langsung merubah data menjadi berkas di tolak	<input checked="" type="checkbox"/> Berfungsi <input type="checkbox"/> Tidak Berfungsi

Kasus dan Hasil Uji				
No	Komponen yang Diujikan	Skenario	Hasil yang Diharapkan	Hasil Pengujian
		Klik tombol tambah yang tampil pada menu	Sistem akan menerima dan langsung membuka <i>from registrasi</i>	[√] Berfungsi [ ] Tidak Berfungsi
		Klik simpan	Sistem akan menerima dan menyimpan data pengajuan	[√] Berfungsi [ ] Tidak Berfungsi
5	Menu beranda pada hakim	Klik menu daftar ruangan	Sistem akan menampilkan ruangan dan jadwal sidang	[√] Berfungsi [ ] Tidak Berfungsi
		Klik menu <i>logout</i>	Sistem akan menerima akun keluar dan kembali ke halaman <i>login</i>	[√] Berfungsi [ ] Tidak Berfungsi
6	Menu beranda pada penggugat	Klik menu jadwal sidang	Sistem akan menampilkan <i>form</i> jadwal dan ruangan sidang	[√] Berfungsi [ ] Tidak Berfungsi
		Klik menu <i>Logout</i>	Sistem akan menerima akun keluar dan kembali ke halaman <i>login</i>	[√] Berfungsi [ ] Tidak Berfungsi

## 2. Pengujian *White Box*



Gambar IV.8 *Flowchart Login*



Gambar IV.9 *Flowgraph Login*

Dari *flowgraph login* pada gambar IV.9 terdapat 6 *edge* dan 6 *node*. Berikut adalah potongan *script* untuk masing-masing *node*:

a. Script pada node 1: start

```
1 <?php
2 // mengaktifkan session pada php
3 require_once "../config/config.php";
4
5 // menangkap data yang dikirim dari form login
6 $username = $_POST['username'];
7 $password = $_POST['password'];
```

b. Script pada node 2: input username dan password

```
5 if(isset($_POST['add'])){
6     $nama = trim(mysqli_real_escape_string($koneksi, $_POST['nama']));
7     $username = trim(mysqli_real_escape_string($koneksi, $_POST['username']));
8     $password = trim(mysqli_real_escape_string($koneksi, $_POST['password']));
9     $level = trim(mysqli_real_escape_string($koneksi, $_POST['level']));
10    mysqli_query($koneksi, "INSERT INTO user (nama, username, password, level) VALUES ('$nama', '$username'
```

c. Script pada node 3: klik login

```
10 // menyeleksi data user dengan username dan password yang sesuai
11 $login = mysqli_query($koneksi, "select * from user where username='$username' and password='$password'");
12 // menghitung jumlah data yang ditemukan
13 $cek = mysqli_num_rows($login);
14
15 // cek apakah username dan password di temukan pada database
16 if($cek > 0){
17
```

d. Script pada node 4: data benar?

```
}else{
    // alihkan ke halaman login kembali
    header("location:index.php?pesan=Gagal !!!");
}
```

e. *Script pada node 5: menampilkan halaman User*

```

mysql_query($koneksi, "INSERT INTO user (nama, username, password, level ) VALUES ('$nama', '$username',
// echo "<script>window.location='".base_url('add.php?sukses=Sukses !!!')."';</script>";
header("location:index.php?sukses=Sukses !!!");
}
?>

```

f. *Script pada node 6: end*

```

mysql_query($koneksi, "INSERT INTO user (nama, username, password, level ) VALUES ('$nama', '$username',
// echo "<script>window.location='".base_url('add.php?sukses=Sukses !!!')."';</script>";
header("location:index.php?sukses=Sukses !!!");
}
?>

```

*Flowgraph login kompleksitas siklomatis* dihitung menggunakan 3 (tiga) cara, yaitu:

- a. Grafik alir mempunyai 2 region.
- b.  $V(G) = 6 \text{ edge} - 6 \text{ node} + 2 = 2$
- c.  $V(G) = 1 \text{ simpul yang diperkirakan} + 1 = 2$

Dengan demikian kompleksitas *siklomatis* dari *flowgraph* yang dijelaskan pada gambar IV.9 adalah 2. Dengan jalur independennya adalah:

Jalur 1 : 1-2-3-4-5-6

Jalur 2 : 1-2-3-4-2-3-4-5-6

Tabel IV.2 *Test Case Login*

Path	1
Jalur	1-2-3-4-5-6

Skenario	<ol style="list-style-type: none"> <li>1. <i>Start</i></li> <li>2. <i>Input username dan password</i></li> <li>3. <i>Klik login</i></li> <li>4. <i>Memvalidasi data benar</i></li> <li>5. <i>Sistem menampilkan halaman beranda user</i></li> <li>6. <i>End</i></li> </ol>
Hasil Pengujian	Berhasil
Path	2
Jalur	1-2-3-4-2-3-4-5-6
Skenario	<ol style="list-style-type: none"> <li>1. <i>Start</i></li> <li>2. <i>Input username dan password</i></li> <li>3. <i>Klik login</i></li> <li>4. <i>Muncul error</i></li> <li>5. <i>Mengisi ulang kembali username dan password</i></li> <li>6. <i>Klik login</i></li> <li>7. <i>Sistem menampilkan beranda user</i></li> <li>8. <i>End</i></li> </ol>
Hasil Pengujian	Berhasil

### 3. Pengujian Efektivitas Perangkat Lunak

Pengujian efektivitas perangkat lunak dilakukan dengan *kuesioner*. Yang diuji coba di sini adalah kualitas sistem secara langsung ke beberapa *user*. Pengujian berupa *kuesioner* terdiri dari lima pertanyaan yang dibagikan kepada 8 responden.

Tabel IV.3 Nilai Skor Maksimum

Jawaban	Skor	Skor Maksimum (skor * jumlah responden)
Sangat setuju	5	40
Setuju	4	32
Cukup setuju	3	24



Jawaban	Skor	Skor Maksimum (skor * jumlah responden)
Kurang setuju	2	16
Tidak setuju	1	8

Untuk persentase masing-masing jawaban dengan Menggunakan rumus:

Dimana:

$$Y = \frac{TS}{\text{Skor Ideal}} \times 100\%$$

Y = Nilai persentase

TS= Total skor responden =  $\sum$  skor x responden

Skor ideal = skor x jumlah responden =  $5 \times 8 = 40$

Tabel IV.4 Kriteria Skor

Kategori	Keterangan
0% - 20%	Tidak setuju
21% - 40%	Kurang setuju
41% - 60%	Cukup setuju
61% - 80%	Setuju
80% - 100%	Sangat setuju

Berikut hasil *persentase* setiap jawaban yang diujikan pada 8 responden (hasil kuisisioner terlampir).

a. Pertanyaan 1

Apakah anda setuju *website* ini cukup mudah digunakan? Hasil kuisisioner pertanyaan pertama dapat dilihat pada tabel IV.5

Tabel IV.5 Hasil Kuisisioner Pertanyaan Pertama

Pertanyaan	Jawaban	Skor	Responden	Jumlah Skor	Nilai Persentase
1	Sangat setuju	5	3	15	$(35/40) \times 100\%$ =87,5%
	Setuju	4	5	20	

Pertanyaan	Jawaban	Skor	Responden	Jumlah Skor	Nilai Persentase
	Cukup setuju	3	0	0	
	Kurang setuju	2	0	0	
	Tidak setuju	1	0	0	
Jumlah			8	35	

Berdasarkan nilai penyajian pertanyaan pertama dapat disimpulkan bahwa 87,5 % responden menilai *website* cukup mudah digunakan.

b. Pertanyaan 2

Apakah anda setuju mengenai tampilan desain dan warna pada *website* ini? Hasil kuesioner pertanyaan kedua dapat dilihat pada Tabel IV.6.

Tabel IV. 6 Hasil Kuesioner Pertanyaan Kedua

Pertanyaan	Jawaban	Skor	Responden	Jumlah Skor	Nilai Persentase
2	Sangat setuju	5	4	20	$(36/40) \cdot 100\% = 90\%$
	Setuju	4	4	16	
	Cukup setuju	3	0	0	
	Kurang setuju	2	0	0	
	Tidak setuju	1	0	0	
Jumlah			8	36	

Berdasarkan nilai penyajian pertanyaan kedua, dapat disimpulkan bahwa 90% responden menilai setuju mengenai tampilan desain dan warna pada *website* ini.

c. Pertanyaan 3

Apakah anda setuju mengenai tampilan menu-menu pada *website* ini?

Hasil kuesioner pertanyaan pertama dapat dilihat pada tabel IV.7.

Tabel IV.7 Hasil Kuesioner Pertanyaan Ketiga

Pertanyaan	Jawaban	Skor	Responden	Jumlah Skor	Nilai Persentase
3	Sangat setuju	5	0	0	$(32/40) \cdot 100\% = 80\%$
	Setuju	4	8	32	
	Cukup setuju	3	0	0	
	Kurang setuju	2	0	0	
	Tidak setuju	1	0	0	
Jumlah			8	32	

Berdasarkan nilai penyajian pertanyaan ketiga, dapat disimpulkan bahwa 80% responden menilai setuju mengenai tampilan menu-menu pada *website* ini.

d. Pertanyaan 4

Apakah anda setuju fitur-fitur dalam *website* ini sudah berfungsi dengan baik? Hasil kuesioner pertanyaan keempat dapat dilihat pada tabel IV.8.

Tabel IV.8 Hasil Kuesioner Pertanyaan Keempat

Pertanyaan	Jawaban	Skor	Responden	Jumlah Skor	Nilai Persentase
4	Sangat setuju	5	3	15	(35/40)*100% =87,5%
	Setuju	4	5	20	
	Cukup setuju	3	0	0	
	Kurang setuju	2	0	0	
	Tidak setuju	1	0	0	
Jumlah			8	35	

Berdasarkan nilai penyajian pertanyaan keempat, dapat disimpulkan bahwa 87,5% responden menilai setuju fitur-fitur dalam *website* ini sudah berfungsi dengan baik.

e. Pertanyaan 5

Apakah anda setuju mengenai tampilan informasi pada aplikasi ini?

Hasil kuisisioner pertanyaan kelima dapat dilihat pada tabel IV.9.

Tabel IV.9 Hasil Kuesioner Pertanyaan Kelima

Pertanyaan	Jawaban	Skor	Responden	Jumlah Skor	Nilai Persentase
5	Sangat setuju	5	1	5	(33/40)*100% =82,5%
	Setuju	4	7	28	
	Cukup setuju	3	0	0	
	Kurang setuju	2	0	0	

Pertanyaan	Jawaban	Skor	Responden	Jumlah Skor	Nilai Persentase
	Tidak setuju	1	0	0	
Jumlah			8	33	

Berdasarkan nilai penyajian pertanyaan kelima, dapat disimpulkan bahwa 82,5% responden menilai setuju mengenai tampilan informasi pada aplikasi ini.

Tabel IV.10 Pengolahan Skala

No Pertanyaan	Nilai Persentase	Keterangan
1	87,5%	Sangat setuju
2	90%	Sangat setuju
3	80%	Setuju
4	87,5%	Sangat setuju
5	82,5%	Sangat setuju
<b>Total persentase</b>	$87,5\% + 90\% + 80\% + 87,5\% + 82,5\% = 427,5\%$	Sangat setuju
<b>Rata-rata</b>	$427,5\% / 5 = 85,5\%$	

Hasil setiap pertanyaan dihitung pada rata-rata keseluruhan. Kemudian dibandingkan untuk menarik kesimpulan. Dari tabel IV.10 dapat ditarik kesimpulan bahwa 85,5% responden sangat setuju dengan kualitas sistem.

## **BAB V**

### **PENUTUP**

#### **V.1 Kesimpulan**

Berdasarkan penelitian yang telah dilakukan, maka dapat disimpulkan sebagai berikut:

1. Sistem penjadwalan sidang dirancang dengan menggunakan pemodelan visual UML *use case diagram*, *activity diagram*, *sequence diagram*, dan *class diagram*. Selain perancangan pemodelan visual dilakukan perancangan *interface* untuk membuat pengguna merasa mudah dan nyaman dalam menggunakan aplikasi.
2. Penerapan jadwal sidang dengan algoritma genetika menyeleksi jadwal, waktu, ruang, jumlah orang yang bertugas, dan hakim. Di mana tahapan algoritma dimulai dari membangkitkan populasi awal, mengevaluasi nilai awal, seleksi, *crossover*, mutasi sampai kriteria optimal sehingga mendapatkan jadwal yang terbaik.

#### **V.2 Saran**

Terlepas dari kemampuan aplikasi yang diulas diatas sebagai aplikasi yang baru berkembang dan masih banyak kekurangan. Saran yang dapat diusulkan adalah aplikasi dapat dikembangkan menggunakan sistem Android sehingga dapat lebih mudah lagi dalam mengakses informasi yang ada.

## DAFTAR PUSTAKA

- Adelia, I. (2021). “Implementasi *E-Court* dalam Proses Penyelesaian Perkara di Pengadilan Tata Usaha Negara”. Skripsi thesis, UIN Sulthan Thaha Saifuddin Jambi.
- Adnyana, IMB, & Wijayana, IK. (2017). “Rancang Bangun Sistem Penjadwalan Sidang Skripsi Menggunakan Algoritma Genetika”. *Jurnal Sistem Dan Informatika (JSI)*, 12 (1), 38-47. Diambil dari <https://jsi.stikom-bali.ac.id/index.php/jsi/article/view/145>.
- Angela Pardede, A. dkk. (2006). “Penerapan Algoritma Genetika pada Permainan Rubik’s Cube”. *Jurnal Departemen Teknik Informatika, Institut Teknologi Bandung*.
- Anhar. (2010). *Panduan Menguasai PHPMyAdmin & MySQL Secara Otodidak*. Jakarta: Mediakita.
- Artanto. (2013). *Penggunaan Apache dan Mysql*. Sekolah Tinggi Ilmu Komputer. Surabaya.
- Assagaf, A. et al. (2018) “Membangun Sistem Informasi Penjadwalan Dengan Metode Algoritma Genetika Pada Laboratorium Teknik Informatika Universitas Muhammadiyah Maluku Utara”, 1(2), pp. 95–105.
- Atmosudirjo, P. (2014). *Administrasi negara*. Jakarta: Ghalia Indonesia.
- Bayu Aji, B. dkk. (2021) “Sistem Penjadwalan Sidang Tugas Akhir Menggunakan Algoritma Genetika”. *Jurnal Teknologi Terapan*, 7(1).
- Budiardjo, Miriam (2008). *Dasar-dasar Ilmu Politik*. Jakarta: Gramedia Pustaka Utama.
- Fiarni, C., Gunawan, A. S., Ricky, Maharani, H., & Kurniawan, H. (2015). “*Automated Scheduling System for Thesis and Project Presentation Using Forward Chaining Method with Dynamic Allocation Resources*”. *Procedia Computer Science*, 72, 209–216. <https://doi.org/10.1016/j.procs.2015.12.133>.
- Flinsetyadz. (2020). *Cara Pengujian Black Box Testing*. Diakses dari <https://flinsetyadi.com/pengujian-black-box-testing/>.

- Hendrik, S. (2010). “Pentingnya Pengakuan Hukum Bagi Provinsi Kepulauan Menuju Keseimbangan Kewenangan Dalam Pengelolaan Sumber Daya Alam Di Wilayah Maritim Daerah Otonomi”. *Jurnal Administrasi Publik Dan Pemerintahan*, ISSN 2161-7104, Vol.3, No.4.
- Josi, A. (2017) “Implementasi Algoritma Genetika Pada Aplikasi Penjadwalan Perkuliahan Berbasis Web Dengan Mengadopsi Model Waterfall”, 02(02), pp. 77–83.
- Kadir, A. (2003). “Konsep dan Tuntunan Praktis Basis Data”. Andi, Yogyakarta.
- Lumban Tobing, R. (2010). “Sistem Simulasi Penjadwalan Kuliah Dengan Menggunakan Algoritma Genetik”. Universitas Sumatera Utara.
- Maku, R. (2021). “Perancangan Sistem Penjadwalan Kuliah dan Ujian Skripsi Program Studi Sistem Informasi Berbasis Web”. *Jurnal Ilmu Komputer*, Universitas Muhammadiyah Gorontalo.
- Mulyanto, Agus. (2009). *Sistem Informasi Konsep dan Aplikasi*. Yogyakarta: Pustaka Pelajar.
- Nugroho, A. (2021). “Rancang Bangun Sistem Informasi Pejadwalan Pelaksanaan Sidang Skripsi dengan Metode Algoritma Genetika Berbasis Web”. Politeknik Negeri Jakarta.
- Oktarina, D. (2019). “Perancangan Sistem Penjadwalan Seminar Proposal dan Sidang Skripsi dengan Metode Algoritma Genetika”. *Jurnal Sistem Informasi dan Informatika Erekeyasa*. ISSN 2503-5304, Vol. 3, No.1.
- Permadi, I. (2010). “Penerapan Algoritma Genetika untuk Optimasi Penjadwalan Tebangan Hutan (*Applying of Genetic Algorithm for Scheduling Optimization Cuts Away Forest*)”. *Juita*, 1, 19–27.
- Sari, Y., Alkaff, M., Wijaya, E. S., Soraya, S., & Kartikasari, D. P. (2019). “Optimasi Penjadwalan Mata Kuliah Menggunakan Metode Algoritma Genetika dengan Teknik *Tournament Selection*”. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 6(1), 85. <https://doi.org/10.25126/jtiik.2019611262>.
- Setyo Rejeki, M., & Tarmuji, A. (2013). “Membangun Aplikasi Autogenerate Script Ke Flowchart Untuk Mendukung Business Process



Reengineering”. Jurnal Sarjana Teknik Informatika, ISSN 2338-5197, Vol.1, No.2.

Sukamto, R. A., & Shalahuddin, M. (2015). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek (Edisi Revisi)*. Bandung: Informatika.

Sutarman. (2003). *Membangun Aplikasi WEB dengan PHP dan MySQL*. Jakarta: Graha Ilmu.

Wairooy. (2020). “Teknik Dalam *White-Box* dan *Black-Box Testing*”. <https://Socs.Binus.Ac.Id/2020/07/02/Teknik-Dalam-White-Box-Dan-Black-Boxtesting>

## LAMPIRAN