



Implementasi Algoritma Genetika pada Perancangan Aplikasi Penjadwalan Instalasi Antivirus Berbasis *Website* menggunakan Metode *Waterfall*

Yoga Septian Nugraha ^{1*}, Ucuk Darusalam ², Agus Iskandar ³

^{1,2,3} Program Studi Sistem Informasi, Fakultas Teknologi Komunikasi dan Informatika, Universitas Nasional.

article info

Article history:

Received 4 June 2021

Received in revised form

29 June 2021

Accepted 27 August 2021

Available online January 2022

DOI:

<https://doi.org/10.35870/jtik.v6i1.417>

Keywords:

Scheduling; Antivirus
Installation; Genetics;
Waterfall; Web.

Kata Kunci:

Penjadwalan; Instalasi
Antivirus; Genetika; Waterfall;
Web.

abstract

Scheduling is the activity of planning in conducting activity at a certain time. The main objective of the scheduling is to improve the effectiveness and the efficiency of the use of resources so that progress that has been determined previously achieved well. A genetic algorithm is one of the most often algorithms used in optimization scheduling. This algorithm works based on a natural process carried out using heuristic methods. One of the scheduling problems at company x in the step to increase security through system security updates with this antivirus installation program involves all employees as users of each division and handled by the IT team as the pic that handles the program. Scheduling is still compiled manually so that raises some of the problems starting from the preparation process to the division of the schedules to require a long time, it usually tends to clashes between the hour and the day when the PIC handles the user due to not automatic scheduling. Therefore, designing an application system by using web-based programming is conducted waterfall method to ease in the process of the preparation and distribution of a schedule which is more systematic with the implementation of the method of the genetic algorithm so that the scheduling that there will be more optimal.

abstrak

Penjadwalan merupakan sebuah aktivitas perencanaan dalam melakukan suatu kegiatan dalam waktu tertentu. Tujuan utama dari sebuah penjadwalan adalah untuk meningkatkan efektivitas dan efisiensi penggunaan sumber daya sehingga suatu progress yang telah ditentukan sebelumnya bisa tercapai dengan baik. Algoritma genetika merupakan salah satu algoritma yang paling sering digunakan pada proses optimasi penjadwalan dimana algoritma ini bekerja berdasarkan pada proses alamiah yang dilakukan menggunakan metode heuristic. Salah satu masalah penjadwalan pada perusahaan x dalam langkah meningkatkan keamanan melalui pembaruan security sistem dengan program instalasi antivirus melibatkan seluruh karyawan sebagai user dari setiap divisi dan ditangani oleh tim IT sebagai pic yang menangani program tersebut. Penjadwalan yang ada masih disusun secara manual sehingga menimbulkan beberapa masalah mulai dari proses penyusunan dan pembagian jadwal memerlukan waktu yang lama dan cenderung terjadi benturan jam saat pic menangani user dikarenakan penjadwalan yang ada belum otomatis. Penelitian menghasilkan sebuah sistem aplikasi dengan menggunakan pemrograman berbasis web dengan mengadopsi metode waterfall dimana hasil dari sistem yang dibuat dapat memberikan kemudahan dalam proses penyusunan dan pembagian jadwal yang lebih sistematis dengan mengimplementasikan metode algoritma genetika sehingga penjadwalan yang ada akan lebih optimal.

Corresponding author. Email: yoganugraha839@gmail.com ^{1}, ucuk.darusalam@gmail.com ^{2*}, agusiskandar1005@gmail.com ^{3*}.

© E-ISSN: 2580-1643.

Copyright © 2022. Published by Lembaga Otonom Lembaga Informasi dan Riset Indonesia (KITA INFO dan RISET) (<http://creativecommons.org/licenses/by/4.0/>).

1. Latar Belakang

Penjadwalan merupakan sebuah aktivitas perencanaan dalam melakukan suatu kegiatan dalam waktu tertentu. Pada dasarnya sebuah penjadwalan mencakup beberapa aspek mulai dari pengurutan aktivitas, pengalokasian aktivitas pada fasilitas dan pemetaan aktivitas menurut urutan waktu. Tujuan utama dari sebuah penjadwalan adalah untuk meningkatkan efektivitas dan efisiensi penggunaan sumber daya sehingga suatu *progress* yang telah ditentukan sebelumnya bisa tercapai dengan baik.

Penelitian ini menggunakan beberapa referensi dari penelitian terdahulu di mana penggunaan algoritma genetika dalam menyelesaikan masalah penjadwalan banyak dilakukan dan telah berhasil di terapkan dalam berbagai masalah kombinatorial, seperti optimasi penjadwalan. Penelitian yang pertama adalah Perancangan Sistem Penjadwalan Seminar Proposal dan Sidang Skripsi Dengan Metode Algoritma Genetika, Dwi Oktarina dan Alyauma Hajjah (2020), membahas tentang penerapan algoritma genetika dalam menghasilkan optimasi penjadwalan yang baik dengan tidak terjadinya tumbukan jadwal [1]. Penelitian kedua yaitu Aplikasi Penjadwalan Mata Pelajaran di SMAN 31 Menggunakan Algoritma Genetika Berbasis *Web*. Ivan, Stephanus Raphael, Halim Agung (2019). membahas tentang penerapan algoritma genetika dalam menghasilkan penjadwalan yang optimal secara otomatis dan tidak saling bertabrakan [2]. Penelitian ketiga adalah Implementasi Algoritma Genetika Pada Aplikasi Penjadwalan Perkuliahan Berbasis *Web* Dengan Mengadopsi Model *Waterfall* [3,4,5]. Ahmat Josi (2016), pada penelitian ini membahas mengenai optimasi penjadwalan pada studi kasus STMIK Prabumulih dimana proses yang dilakukan menghasilkan penjadwalan yang lebih efektif dan efisien dengan alur perancangan sistem memanfaatkan metode *waterfall* [6]. Berdasarkan referensi diatas masing-masing menggunakan satu buah metode yaitu algoritma genetika dalam proses penyelesaian masalah penjadwalan di bidang akademik. Pengembangan yang dilakukan pada penelitian ini mengacu dari beberapa referensi diatas adalah dengan menerapkan algoritma genetika pada studi kasus di perusahaan x dengan latar belakang yang berbeda mengenai permasalahan yang terjadi dengan membuat sistem yang lebih terkomputerisasi

agar penjadwalan yang dihasilkan bisa lebih optimal.

Permasalahan yang terjadi pada penelitian ini diambil dari salah satu penjadwalan yang sedang dilakukan di suatu perusahaan x dalam langkah meningkatkan keamanan melalui pembaruan *security* sistem dengan program instalasi antivirus. Program ini melibatkan seluruh karyawan bertindak sebagai *user* dari setiap divisi dan ditangani oleh tim IT yang bertugas sebagai pic. Penjadwalan yang ada masih menggunakan penyusunan dan pembagian jadwal secara manual mengacu pada data penjadwalan berupa *spreadsheet*, sehingga menimbulkan beberapa masalah mulai dari proses penyusunan dan pembagian jadwal memerlukan waktu yang lama dan cenderung terjadi benturan jam saat pic menangani *user* dalam program tersebut. Oleh karena itu, masalah penjadwalan ini sangatlah penting mengingat permasalahan yang terjadi sangatlah kompleks dengan tingkat efektivitas dan efisiensi waktu yang ada tidak tercapai.

Dari permasalahan ini dibutuhkan suatu algoritma yang dapat menyelesaikan masalah optimasi dalam penjadwalan. Peneliti memilih algoritma genetika dimana algoritma ini dapat digunakan untuk menyelesaikan masalah penjadwalan dengan serangkaian proses evolusi yang dilakukan seperti pembangkitan kromosom dari pembentukan individu secara acak dengan meliputi proses seleksi, *crossover* dan mutasi. Dari proses tersebut akan menghasilkan individu baru dengan nilai kebugaran (*fitness*) paling tinggi, di mana nilai tersebut yang akan digunakan pada penyelesaian masalah penjadwalan ini. Perancangan sistem aplikasi penjadwalan menggunakan pemrograman berbasis *web* dengan bahasa pemrograman PHP dan MySQL sebagai database penyimpanannya. Untuk metode yang digunakan menggunakan metode *waterfall* menggunakan pendekatan model sekuensial linier dengan pengembangan sistem dimulai dari tahap analisis, desain, koding dan pengujian. Proses pengujian sistem sendiri menggunakan metode *Black-box Testing* yang dimana pengujian ini didasarkan pada detail aplikasi seperti tampilan aplikasi, fungsi -fungsi yang ada pada aplikasi, dan kesesuaian alur fungsi yang diinginkan sebelumnya sesuai kebutuhan. teknik yang digunakan yaitu Equivalence Partitioning dimana teknik ini membagi domain input dari suatu program kedalam kelas-kelas data sehingga *test case* dapat diperoleh.

2. Metode Penelitian

Pengumpulan Data

Pengumpulan data yang digunakan pada penelitian ini terdiri dari dua jenis yaitu data primer dan sekunder. Data primer dihasilkan melalui observasi di lapangan dan wawancara dengan pihak-pihak terkait sebagai proses bagian dari analisis kebutuhan sistem serta melakukan studi literature dari sumber-sumber tertulis seperti halnya jurnal ilmiah dari penelitian terdahulu sebagai referensi dan acuan dalam hal pembuatan sistem yang akan dilakukan. Data sekunder didapatkan dari salah satu perusahaan x yang sedang menangani program instalasi antivirus ini berupa *spreadsheet*. Data ini akan digunakan dalam proses perancangan sistem penjadwalan agar menghasilkan output sesuai dengan harapan.

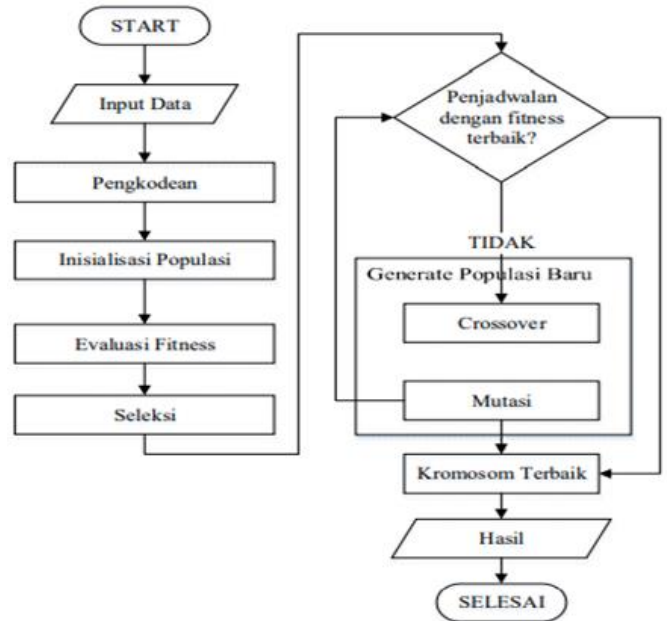
Algoritma Genetika

Algoritma Genetika merupakan suatu metode heuristik yang dikembangkan berdasarkan prinsip genetika dan proses seleksi alamiah Teori Evolusi Darwin. Metode optimasi dikembangkan oleh John Holland sekitar tahun 1960 dan dipopulerkan oleh salah seorang mahasiswanya [7,8], David Goldberg, pada tahun 1980-an [9]. Proses pencarian penyelesaian yang dilakukan dengan terpilihnya sebuah penyelesaian dalam algoritma ini adalah berlangsung sama seperti terpilihnya suatu individu untuk bertahan hidup dalam sebuah proses evolusi [10,11].

Pencarian dimulai dengan proses pembangkitan sejumlah “individu” secara acak yang disebut dengan kromosom [12,13]. Kromosom-kromosom inilah merupakan representasi calon penyelesaian yang akan diperiksa nilai yang sebenarnya seperti halnya proses evolusi alamiah, kromosom-kromosom akan dinilai tingkat kebugarannya, hanya kromosom dengan tingkat kebugaran yang tinggi saja yang akan terpilih untuk bertahan dalam populasi tersebut.

Kromosom-kromosom yang terpilih sebagian akan melakukan proses reproduksi melalui penyilangan (*crossover*) [14]. Didalam proses reproduksi ini akan melahirkan individu-individu baru. Gabungan dari individu-individu baru dengan kromosom-kromosom yang tidak melakukan proses reproduksi akan membentuk populasi baru pada generasi berikutnya sehingga proses tersebut akan terus

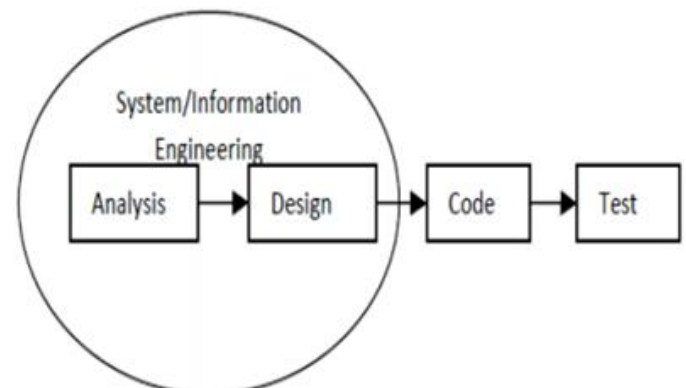
diulang [15]. Serangkaian proses yang dilakukan berlangsung sampai jumlah generasi tercapai. Penyelesaian yang ditemukan adalah kromosom yang mempunyai tingkat kebugaran (*fitness*) paling tinggi pada generasi terakhir yang akan digunakan pada penyelesaian ini. Berikut dibawah ini alur *Flowchart* algoritma genetika:



Gambar 1. *Flowchart* Algoritma Genetika.

Pengembangan Sistem

Metode pengembangan sistem yang digunakan pada penelitian ini adalah menggunakan metode *Waterfall* dengan model sekuensial linier. Metode ini menggambarkan pendekatan yang cukup sistematis dan berurutan pada pengembangan software, mulai dari analisis, design, coding, dan testing.



Gambar 2. Model Sekuensial Linear/*Waterfall* Model.

3. Hasil dan Pembahasan

Analisis

Analisis yang dilakukan menjadi dua bagian yaitu antara analisis permasalahan dan analisis sistem.

Pada analisis permasalahan bertujuan untuk mengetahui masalah sistem penjadwalan antivirus yang ada pada perusahaan x. Berdasarkan dari studi kasus yang terjadi bahwa penyusunan dan pembagian jadwal mengenai program instalasi antivirus pada perusahaan x masih dilakukan secara manual. Sehingga di dalam proses tersebut mengalami berbagai masalah mulai dari membutuhkan waktu yang lama saat proses penyusunan dan pembagian jadwal yang dilakukan dan cenderung terjadi benturan jam saat pic menangani *user*. Pada sistem penjadwalan instalasi antivirus ini masing – masing pic yang terdiri dari 6 orang diberikan waktu sekitar 45 menit untuk menangani proses instalasi antivirus terhadap *user* dari setiap divisi selama masa jam kerja dari senin sampai dengan jumat. Karena terbatasnya pic yang hanya 6 orang saat menangani *user* sering terganggu karena banyaknya jadwal yang berbenturan.

Pada analisis sistem ditujukan untuk mencari solusi dari analisis permasalahan yang terjadi pada studi kasus utama yaitu masalah penjadwalan mengenai program instalasi antivirus yang ada di perusahaan x. analisis sistem dilakukan dengan tujuan agar sistem yang dibuat dapat memberikan kemudahan terutama dalam proses penyusunan dan pembagian jadwal secara otomatis dan tidak terjadi benturan jam. Beberapa tahapan analisis sistem dilakukan dari proses input untuk membuat aplikasi penjadwalan yaitu data pic, data *user*, data divisi. Berikut dibawah ini adalah data yang digunakan untuk proses perancangan aplikasi.

Tabel 1. Data PIC

| Nama PIC | Kode PIC |
|----------|----------|
| Aldi | PC1 |
| Alif | PC2 |
| Adam | PC3 |
| Thoriq | PC4 |
| Yoga | PC5 |
| Wawan | PC6 |

Tabel 2. Data *User*

| No | Model | Divisi | Manufacturer | User |
|----|-------------|-------------------|--------------|----------------------|
| 1 | V130 | People Operations | Lenovo | Achmad Farih |
| 2 | Macbook Pro | Software Engineer | Apple | Achmad Fauzi |
| 3 | Macbook Pro | Software Engineer | Apple | Achmad Jafar |
| 4 | V130 | Finance | Lenovo | Achmad Reza Adriansa |
| 5 | V130 | HR | Lenovo | Adam Farid Alhadi |
| 6 | V130 | Tele Marketing | Lenovo | Ade Achmad |
| 7 | V130 | Sales | Lenovo | Ade Isni |
| 8 | Macbook Air | Software Engineer | Lenovo | Ade Saputra |
| 9 | V130 | Finance | Lenovo | Adeline |
| 10 | V130 | People Operations | Lenovo | Adetia Suhartini |
| 11 | V130 | People Operations | Lenovo | Andhika Nirmalasari |
| 12 | V130 | Sales | Lenovo | Adhy Wibawa |
| 13 | Macbook Pro | Software Engineer | Apple | Adillah Azmi |
| 14 | Inspiron | Tele Marketing | Dell | Adit Triyadi |
| 15 | V130 | People Operations | Lenovo | Dien Marbella |
| 16 | Macbook Pro | Software Engineer | Apple | Dimas Hendria |
| 17 | Thinkpad | Finance | Lenovo | Dinar Arindani |
| 18 | V130 | HR | Lenovo | Dini Sekar |
| 19 | Macbook Pro | Software Engineer | Apple | Dion Hiananto |
| 20 | V130 | Sales | Lenovo | Doly Feriandy |
| 21 | V130 | Sales | Lenovo | Donny Putranto |
| 22 | V130 | HR | Lenovo | Dwi Hariana Pane |
| 23 | V130 | Sales | Lenovo | Dwi Nurcahyo |
| 24 | V310 | Finance | Lenovo | Dwi Setyo |
| 25 | Ideapad | Sales | Lenovo | Dwimar Nurwanto |

Tabel 3. Data Divisi

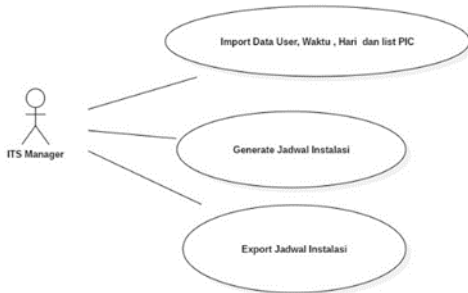
| No | Nama Divisi | Kode Divisi |
|----|-------------------|-------------|
| 1 | People Operations | D01 |
| 2 | Software Engineer | D02 |
| 3 | HR | D03 |
| 4 | Finance | D04 |
| 5 | Tele Marketing | D05 |
| 6 | Sales | D06 |

Desain

Selain analisis sistem digunakan untuk kebutuhan awal agar proses yang dilakukan dapat menggambarkan alur dari rancangan yang dibuat. Tahapan desain menggambarkan sistem yang akan dibuat dengan memperlihatkan urutan dan hubungan beserta instruksinya. Proses desain pada penelitian ini menggunakan model use case diagram dan *Flowchart* berikut seperti dibawah ini:

a) *Use Case diagram*

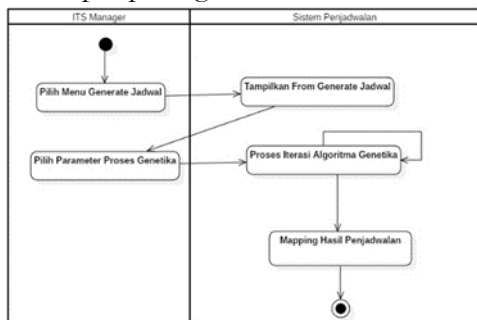
pada hal ini bertujuan untuk menggambarkan proses yang terdapat dalam suatu sistem secara garis besar, proses yang dikembangkan menggambarkan interaksi antara ITS Manager dengan sistem penjadwalan yang dibuat seperti terdapat pada gambar dibawah ini:



Gambar 3. Use Case Diagram

b) *Activity diagram*

pada hal ini digunakan untuk menggambarkan workflow atau aliran kerja pada suatu sistem yang dimana urutan aktivitas proses ini akan menggambarkan aktifitas pada sistem yang dijalankan seperti terdapat pada gambar dibawah ini:

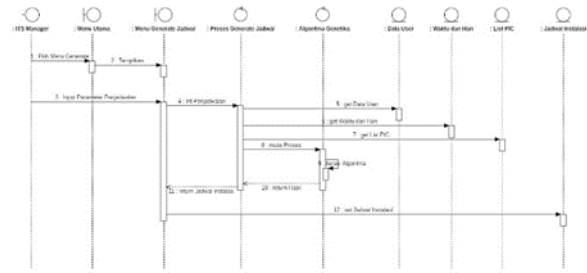


Gambar 4. Activity Diagram

c) *Sequence diagram*

pada hal ini digunakan untuk menggambarkan sebuah interaksi antar objek yang terlibat dalam sistem pada proses yang dilakukan dengan rangkaian skenario langkah – langkah yang dilakukan sebagai

respon dari sebuah event untuk menghasilkan output tertentu seperti terdapat pada gambar dibawah ini:

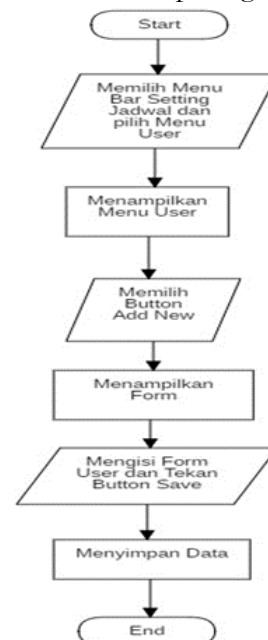


Gambar 5. Sequence Diagram

Flowchart adalah sebuah gambaran dari sistem yang akan dibuat dengan memperlihatkan urutan dengan hubungan proses berikut instruksinya. Dengan penggunaan *Flowchart* pada penelitian ini bertujuan agar urutan dari proses kegiatan dari suatu sistem menjadi lebih jelas, jika terjadi penambahan pada proses yang terjadi dapat dilakukan dengan mudah. Dan berikut dibawah ini alur dari *Flowchart* yang dibuat:

d) *Flowchart Menu User*

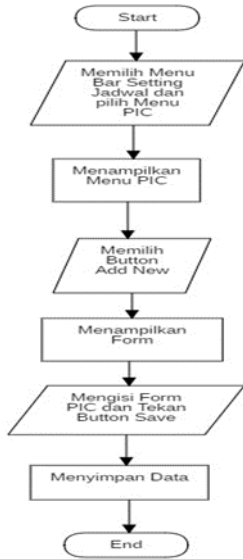
Pada proses ini langkah langkah yang dilakukan dalam menambahkan data *user* dimulai dari memilih menu bar setting jadwal kemudian pilih menu *user* dan akan menampilkan form data *user*, klik button add new setelah form ditampilkan kemudian isi sesuai kebutuhan dan klik button save maka otomatis akan menambahkan data *user*. Seperti gambar dibawah ini:



Gambar 6. *Flowchart* Menu User

e) *Flowchart* Menu PIC

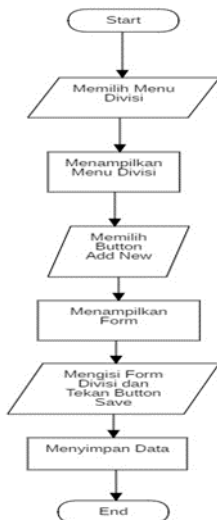
Pada proses ini langkah langkah yang dilakukan dalam menambahkan data pic dimulai dari memilih menu bar setting jadwal kemudian pilih menu pic dan akan menampilkan form data pic, klik button add new setelah form ditampilkan kemudian isi sesuai kebutuhan dan klik button save maka otomatis akan menambahkan data pic. Seperti gambar dibawah ini:



Gambar 7. *Flowchart* Menu PIC

f) *Flowchart* Menu Divisi

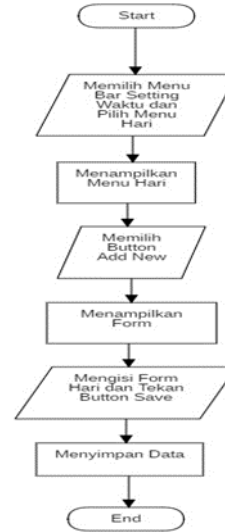
Pada proses ini langkah langkah yang dilakukan dalam menambahkan data divisi dimulai dari memilih menu bar setting kemudian pilih menu divisi dan akan menampilkan form input data divisi setelah form selesai di isi kemudian klik button save maka otomatis akan menambahkan data divisi. Seperti gambar dibawah ini:



Gambar 8. *Flowchart* Menu Divisi

g) *Flowchart* Menu Hari

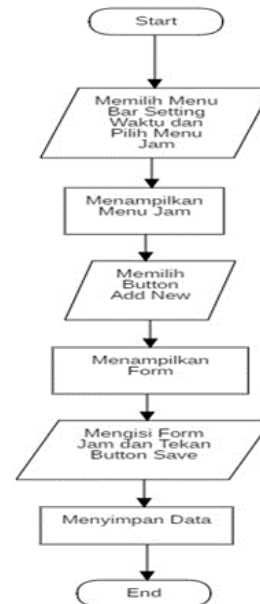
Pada proses ini langkah langkah yang dilakukan dalam menambahkan data hari dimulai dari memilih menu bar setting waktu kemudian pilih menu hari dan akan menampilkan form input data hari setelah form selesai di isi kemudian klik button save maka otomatis akan menambahkan data hari. Seperti gambar dibawah ini:



Gambar 9. *Flowchart* Menu Hari

h) *Flowchart* Menu Jam

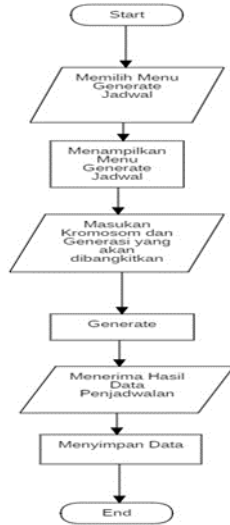
Pada proses ini langkah langkah yang dilakukan dalam menambahkan data jam dimulai dari memilih menu bar setting waktu kemudian pilih menu jam dan akan menampilkan form input data jam setelah form selesai di isi kemudian klik button save maka otomatis akan menambahkan data jam. Seperti gambar dibawah ini:



Gambar 10. *Flowchart* Menu Jam

i) *Flowchart* Menu Generate Penjadwalan

Pada proses ini langkah yang dilakukan yaitu pilih menu generate jadwal dan masukan kromosom dan generasi yang akan dibangkitkan secara acak kemudian tekan button generate maka hasil dari proses generate penjadwalan akan muncul dan menampilkan penjadwalan dengan mengambil nilai *fitness* terbaik seperti gambar dibawah ini:



Gambar 11. *Flowchart* Menu Generate Jadwal

Coding

Pada proses rancangan yang telah dibuat kemudian diterjemahkan ke dalam bentuk bahasa pemrograman pada penelitian ini menggunakan bahasa pemrograman PHP. Coding berfungsi sebagai proses dalam menjalankan aplikasi dengan logika algoritma yang telah di terjemahkan, agar aplikasi berjalan sesuai dengan harapan yang telah ditentukan sebelumnya, maka dari itu coding harus sesuai dengan rancangan yang telah dibuat. Berikut ini potongan coding aplikasi penjadwalan antivirus berbasis *web* dapat dilihat pada gambar berikut:

```

    <code>
    # penjadwalan.php
    1 <div class="page-header">
    2 | <h1>Penjadwalan</h1>
    3 </div>
    4 <?php
    5 $success = true;
    6 $a = 10;
    7 $b = 25;
    8
    9 if(isset($_GET['num_kromosom'])) {
    10 $num_kromosom = $_GET['num_kromosom'];
    11 if($num_kromosom < $a || $num_kromosom > 500) {
    12 print_msg("Masukkan jumlah kromosom dari $a sampai 500");
    13 $success = false;
    14 }
    15
    16 $max_generation = $_GET['max_generation'];
    17 if($max_generation < $b || $max_generation > 500) {
    18 print_msg("Masukkan maksimal generasi dari $b sampai 500");
    19 $success = false;
    20 }
    21 } else {
    22 $num_kromosom = $a;
    23 $max_generation = $b;
    24 }
    25 </code>
    </pre>

```

```

    <code>
    function generate(){
    //this->console-> "000Generasi Awal/has";
    $cro = $this->get_cromosom();

    while(($this->generation < $this->max_generation) && $this->success == false){
    $this->console-> "001Generasi ke-$this->generation/has";
    $this->show_cromosom();
    $this->show_fitness();
    $this->get_cro_pro();
    $this->selection();
    $this->show_cromosom();
    $this->show_fitness();
    $this->crossover();
    $this->show_cromosom();
    $this->generation++;
    }
    $this->save_result();

    if($this->debug)
    $this->console-> "";

    $this->console-> "\n\nFITNESS TERBAIK: $this->best_fitness";
    $this->console-> "\n\nGENERASI: $this->generation";
    $this->console-> "\n\nKROMOSOM TERBAIK: " . $this->print_cros($this->best_cromosom);
    $this->get_debug();
    }
    </code>
    </pre>

```

Gambar 12. Potongan Coding

Pada potongan coding diatas pembangkitan kromosom dan Generasi sama-sama dibatasi hanya sampai dengan 500 dikarenakan proses yang dijalankan agar tidak terjadi floating dan komputasi data terlalu banyak. Function generate berfungsi untuk memproses data penjadwalan meliputi serangkaian proses yang dilalui mulai dari pembentukan kromosom, generasi sampai ke proses seleksi, *crossover* dan mutasi hingga menghasilkan nilai *fitness* terbaik yang akan digunakan sebagai solusi masalah penjadwalan.

Implementasi Algoritma Genetika

Secara singkat, pemetaan proses alamiah ke dalam proses komputasi Algoritma Genetika dapat dilihat di Tabel. 4

Tabel 4. Pemetaan Proses Alamiah ke Proses Komputasi

| Proses Alamiah | Proses Komputasi |
|---------------------------|---|
| Individu | Penyelesaian Masalah |
| Populasi | Himpunan Penyelesaian |
| <i>Fitness</i> /Kebugaran | Kualitas Penyelesaian |
| Kromosom | Kode/representasi penyelesaian |
| Mutasi | Operator Genetika |
| Penyilangan | Operator Genetika |
| Gen | Bagian dari representasi penyelesaian |
| Seleksi | Menyelesaikan masalah berdasarkan kualitasnya |

Proses pertama dimulai dari pembangkitan kromosom dan pembangkitan generasi dari populasi secara acak. Pada implementasinya sendiri serangkaian proses yang dilalui akan meliputi proses seleksi, *crossover* dan mutasi hingga akan menghasilkan nilai *fitness* terbaik yang akan digunakan pada proses penjadwalan. Sebagai contoh kromosom yang dibangkitkan adalah 14 dan Generasi yang

dibangkitkan adalah 25 dan berikut hasil representasi kromosom:

Kromosom[0]: ([1,4,25],[2,5,2],[3,4,27],[4,4,6],[5,2,17],[6,4,21],[7,1,11],[8,1,5],[9,0,6],[10,3,7],[11,3,2],[12,5,23],[13,1,2],[14,5,16],[15,0,6],[16,3,15],[17,3,3],[18,4,0],[19,4,17],[20,3,28],[21,4,10],[22,3,18],[23,5,22],[24,0,9],[25,5,5])

Kromosom[1]: ([1,1,22],[2,5,10],[3,1,27],[4,4,7],[5,4,19],[6,5,22],[7,0,13],[8,4,10],[9,4,6],[10,5,24],[11,2,14],[12,2,12],[13,3,1],[14,1,14],[15,4,7],[16,2,10],[17,3,12],[18,4,17],[19,4,26],[20,0,3],[21,5,16],[22,3,6],[23,2,29],[24,3,23],[25,3,23])

Kromosom[2]: ([1,2,20],[2,2,1],[3,5,6],[4,2,2],[5,0,23],[6,3,29],[7,2,20],[8,3,19],[9,1,13],[10,4,9],[11,2,6],[12,0,15],[13,3,7],[14,4,5],[15,5,16],[16,2,13],[17,4,10],[18,0,3],[19,0,12],[20,3,11],[21,0,22],[22,2,0],[23,3,24],[24,1,27],[25,3,22])

Kromosom[3]: ([1,0,11],[2,0,5],[3,4,7],[4,4,7],[5,2,10],[6,5,23],[7,1,17],[8,4,21],[9,4,7],[10,5,24],[11,5,6],[12,3,28],[13,5,14],[14,4,19],[15,2,12],[16,0,9],[17,4,18],[18,5,3],[19,0,22],[20,4,25],[21,1,4],[22,3,29],[23,1,15],[24,4,2],[25,3,14])

Kromosom[4]: ([1,0,10],[2,0,10],[3,5,15],[4,3,6],[5,1,10],[6,2,22],[7,3,0],[8,4,8],[9,0,10],[10,5,2],[11,0,13],[12,3,2],[13,5,10],[14,1,23],[15,1,18],[16,5,17],[17,1,1],[18,1,24],[19,5,29],[20,5,23],[21,2,9],[22,5,23],[23,1,5],[24,1,13],[25,0,14])

Kromosom[5]: ([1,1,22],[2,0,6],[3,2,29],[4,1,19],[5,5,13],[6,2,1],[7,4,13],[8,4,10],[9,5,13],[10,0,25],[11,2,22],[12,1,6],[13,3,2],[14,5,9],[15,5,17],[16,2,0],[17,0,17],[18,3,13],[19,5,28],[20,0,21],[21,5,18],[22,1,11],[23,0,6],[24,5,7],[25,3,21])

Kromosom[6]: ([1,0,18],[2,2,9],[3,3,18],[4,2,21],[5,0,17],[6,0,11],[7,4,0],[8,0,18],[9,2,5],[10,1,24],[11,4,22],[12,4,21],[13,3,0],[14,4,7],[15,2,10],[16,1,10],[17,2,19],[18,4,5],[19,5,13],[20,4,3],[21,2,28],[22,3,29],[23,3,12],[24,0,5],[25,3,11])

Kromosom[7]: ([1,5,26],[2,4,3],[3,4,12],[4,1,26],[5,0,9],[6,1,5],[7,1,18],[8,5,27],[9,1,9],[10,5,11],[11,4,27],[12,1,1],[13,5,7],[14,0,15],[15,4,11],[16,0,22],[17,0,5],[18,0,23],[19,5,14],[20,5,18],[21,5,26],[22,5,21],[23,1,15],[24,4,17],[25,5,28])

Kromosom[8]: ([1,3,24],[2,5,13],[3,1,27],[4,4,9],[5,1,21],[6,4,23],[7,3,19],[8,5,7],[9,4,26],[10,0,23],[11,1,12],[12,5,16],[13,3,17],[14,5,25],[15,0,2],[16,0,5],[17,2,18],[18,0,12],[19,4,23],[20,4,7],[21,4,17],[22,3,18],[23,1,2],[24,2,27],[25,0,5])

Kromosom[9]: ([1,0,4],[2,0,13],[3,1,25],[4,2,22],[5,3,6],[6,5,4],[7,3,5],[8,1,8],[9,5,21],[10,2,1],[11,1,16],[12,3,1],[13,2,9],[14,0,22],[15,4,26],[16,1,29],[17,3,8],[18,3,26],[19,4,25],[20,2,20],[21,3,18],[22,1,29],[23,5,6],[24,0,16],[25,1,8])

Kromosom[10]: ([1,5,9],[2,2,4],[3,5,22],[4,1,27],[5,1,29],[6,2,3],[7,0,13],[8,0,4],[9,5,24],[10,1,6],[11,0,9],[12,5,23],[13,0,27],[14,3,9],[15,0,29],[16,3,6],[17,5,15],[18,0,0],[19,1,16],[20,5,9],[21,3,0],[22,4,22],[23,1,17],[24,3,21],[25,4,6])

Kromosom[11]: ([1,5,10],[2,4,8],[3,5,25],[4,4,4],[5,1,12],[6,2,22],[7,1,3],[8,4,8],[9,1,21],[10,3,4],[11,1,1],[12,0,25],[13,2,21],[14,2,26],[15,0,1],[16,4,0],[17,2,13],[18,1,10],[19,4,13],[20,5,20],[21,0,10],[22,1,20],[23,4,22],[24,3,24],[25,5,7])

Kromosom[12]: ([1,0,19],[2,3,11],[3,1,15],[4,2,21],[5,4,10],[6,2,6],[7,4,4],[8,5,19],[9,3,26],[10,3,29],[11,0,25],[12,4,25],[13,3,15],[14,0,29],[15,2,22],[16,0,28],[17,4,18],[18,5,7],[19,0,9],[20,4,17],[21,3,6],[22,1,20],[23,4,1],[24,2,8],[25,3,0])

Kromosom[13]: ([1,0,5],[2,2,15],[3,5,1],[4,0,12],[5,1,6],[6,3,14],[7,0,8],[8,5,5],[9,3,3],[10,5,17],[11,0,5],[12,0,6],[13,1,19],[14,2,18],[15,3,29],[16,3,15],[17,3,9],[18,5,27],[19,4,26],[20,5,23],[21,4,0],[22,4,1],[23,0,5],[24,2,17],[25,1,18])

Pada proses representasi kromosom hanya kromosom dengan nilai *fitness* tertinggi yang dapat bertahan pada populasi. Untuk mencari nilai *fitness* tertinggi dilakukan perhitungan dengan rumus $F = 1 / (1 + \text{Fungsi_Objektif})$ fungsi objektif sendiri diambil dari perhitungan kromosom yang telah dibangkitkan. Pada perhitungan ini ditambahkan 1 agar dapat menghindari proses kesalahan program saat dikurangi 0.

$Fitness [0] = 1/(1+0+1+4) = 0.1667$
 $Fitness [1] = 1/(1+0+1+4) = 0.1667$
 $Fitness [2] = 1/(1+0+1+1) = 0.3333$
 $Fitness [3] = 1/(1+0+3+1) = 0.2$
 $Fitness [4] = 1/(1+0+4+6) = 0.091$
 $Fitness [5] = 1/(1+0+2+8) = 0.091$
 $Fitness [6] = 1/(1+0+1+2) = 0.25$
 $Fitness [7] = 1/(1+0+1+6) = 0.125$
 $Fitness [8] = 1/(1+0+1+3) = 0.2$
 $Fitness [9] = 1/(1+0+1+5) = 0.142857$
 $Fitness [10] = 1/(1+0+1+5) = 0.142857$
 $Fitness [11] = 1/(1+0+1+7) = 0.1111$
 $Fitness [12] = 1/(1+0+0+3) = 0.25$
 $Fitness [13] = 1/(1+0+3+5) = 0.1111$
 Total *Fitness* = 2.381

Setelah proses perhitungan selesai maka akan menghasilkan nilai *fitness* dari generasi ke 1 yaitu $F = 2.3814213564214$. Kemudian dihitung nilai probabilitasnya dari setiap kromosom yang bertujuan agar dapat mengetahui kromosom mana yang berpeluang untuk terpilih pada generasi selanjutnya. Rumus mencari probabilitas ($P[i] = fitness[i] / total_fitness$)

$$\begin{aligned}
 P[0] &= 0.1667 / 2.381 = 0.700 \\
 P[1] &= 0.1667 / 2.381 = 0.700 \\
 P[2] &= 0.3333 / 2.381 = 0.139 \\
 P[3] &= 0.2 / 2.381 = 0.083 \\
 P[4] &= 0.091 / 2.381 = 0.038 \\
 P[5] &= 0.091 / 2.381 = 0.038 \\
 P[6] &= 0.25 / 2.381 = 0.104 \\
 P[7] &= 0.125 / 2.381 = 0.052 \\
 P[8] &= 0.2 / 2.381 = 0.083 \\
 P[9] &= 0.142857 / 2.381 = 0.059 \\
 P[10] &= 0.142857 / 2.381 = 0.059 \\
 P[11] &= 0.1111 / 2.381 = 0.046 \\
 P[12] &= 0.25 / 2.381 = 0.104 \\
 P[13] &= 0.1111 / 2.381 = 0.046
 \end{aligned}$$

Dari perhitungan diatas terlihat bahwa kromosom ke [0] dan [1] mempunyai nilai probabilitas paling tinggi dan berpeluang untuk terpilih pada generasi berikutnya. Untuk melakukan seleksi menggunakan metode Roda Roulette dapat dilakukan dengan menghitung nilai probabilitas dipilih dari setiap kromosom yang ada sebagai berikut:

$$\begin{aligned}
 P[0] &= 0.700 \\
 P[1] &= 0.700 \\
 P[2] &= 0.139 \\
 P[3] &= 0.083 \\
 P[4] &= 0.038 \\
 P[5] &= 0.038 \\
 P[6] &= 0.104 \\
 P[7] &= 0.052 \\
 P[8] &= 0.083 \\
 P[9] &= 0.059 \\
 P[10] &= 0.059 \\
 P[11] &= 0.046 \\
 P[12] &= 0.104 \\
 P[13] &= 0.046
 \end{aligned}$$

Setelah diketahui nilai probabilitas setiap kromosom dilanjutkan dengan menghitung nilai probabilitas kumulatif. Untuk menghitung nilai probabilitas kumulatif yang dipilih untuk setiap kromosom dapat ditentukan sebagai berikut:

$$\begin{aligned}
 C[0] &= 0.700 \\
 C[1] &= 0.700 + 0.700 = 1.4 \\
 C[2] &= 0.139 + 0.700 + 0.700 = 1.539 \\
 C[3] &= 0.083 + 0.139 + 0.700 + 0.700 = 1.622
 \end{aligned}$$

$$\begin{aligned}
 C[4] &= 0.038 + 0.083 + 0.139 + 0.700 + 0.700 = 1.66 \\
 C[5] &= 0.038 + 0.038 + 0.083 + 0.139 + 0.700 + 0.700 = 1.698 \\
 C[6] &= 0.104 + 0.038 + 0.038 + 0.083 + 0.139 + 0.700 + 0.700 = 1.802 \\
 C[7] &= 0.052 + 0.104 + 0.038 + 0.038 + 0.083 + 0.139 + 0.700 + 0.700 = 1.854 \\
 C[8] &= 0.083 + 0.052 + 0.104 + 0.038 + 0.038 + 0.083 + 0.139 + 0.700 + 0.700 = 1.937 \\
 C[9] &= 0.059 + 0.083 + 0.052 + 0.104 + 0.038 + 0.038 + 0.083 + 0.139 + 0.700 + 0.700 = 1.996 \\
 C[10] &= 0.059 + 0.059 + 0.083 + 0.052 + 0.104 + 0.038 + 0.038 + 0.083 + 0.139 + 0.700 + 0.700 = 2.055 \\
 C[11] &= 0.046 + 0.059 + 0.059 + 0.083 + 0.052 + 0.104 + 0.038 + 0.038 + 0.083 + 0.139 + 0.700 + 0.700 = 2.101 \\
 C[12] &= 0.104 + 0.046 + 0.059 + 0.059 + 0.083 + 0.052 + 0.104 + 0.038 + 0.038 + 0.083 + 0.139 + 0.700 + 0.700 = 2.205 \\
 C[13] &= 0.046 + 0.104 + 0.046 + 0.059 + 0.059 + 0.083 + 0.052 + 0.104 + 0.038 + 0.038 + 0.083 + 0.139 + 0.700 + 0.700 = 2.251
 \end{aligned}$$

Maka hasil Kumulatifnya adalah dari 14 kromosom yang dibangkitkan adalah:

$$\begin{aligned}
 \text{Kromosom}[0] &= K [0.7] \\
 \text{Kromosom}[1] &= K [1.4] \\
 \text{Kromosom}[2] &= K [1.5] \\
 \text{Kromosom}[3] &= K [1.6] \\
 \text{Kromosom}[4] &= K [1.6] \\
 \text{Kromosom}[5] &= K [1.6] \\
 \text{Kromosom}[6] &= K [1.8] \\
 \text{Kromosom}[7] &= K [1.8] \\
 \text{Kromosom}[8] &= K [1.9] \\
 \text{Kromosom}[9] &= K [1.9] \\
 \text{Kromosom}[10] &= K [2.0] \\
 \text{Kromosom}[11] &= K [2.1] \\
 \text{Kromosom}[12] &= K [2.2] \\
 \text{Kromosom}[13] &= K [2.2]
 \end{aligned}$$

Kemudian hasilkan nomor acak [0-1] untuk setiap kromosom, yaitu R_j , dengan $J = 1, 2, n$, dimana setiap kromosom akan dipilih jika bilangan acak diangkat dalam interval probabilitas kumulatif antara kromosom. Diketahui bahwa K_j adalah kromosom j th, 1. Jika $R_j < PK_i$, pilih K_1 sebagai induk kromosom j th. 2. Jika $PK_{(i-1)} < R_j < PK_i$, pilih K_1 sebagai kromosom induk j th. Hasilnya seperti dibawah ini kromosom baru hasil seleksi:

$$\begin{aligned}
 \text{Kromosom}[0]: & ([1,0,19], [2,3,11], [3,1,15], [4,2,21], [5,4,10], [6,2,6], [7,4,4], [8,5,19], [9,3,26], [10,3,29], [11,0,25], [12,4,25], [13,3,15], [14,0,29], [15,2,22], [16,0,28], [17,4,18], [18,5,7], [19,0,9], [20,4,17], [21,3,6], [22,1,20], [23,4,1], [24,2,8], [25,3,0]) \\
 \text{Kromosom}[1]: & ([1,3,24], [2,5,13], [3,1,27], [4,4,9])
 \end{aligned}$$

, [5, 1, 21], [6, 4, 23], [7, 3, 19], [8, 5, 7], [9, 4, 26], [10, 0, 23], [11, 1, 12], [12, 5, 16], [13, 3, 17], [14, 5, 25], [15, 0, 2], [16, 0, 5], [17, 2, 18], [18, 0, 12], [19, 4, 23], [20, 4, 7], [21, 4, 17], [22, 3, 18], [23, 1, 2], [24, 2, 27], [25, 0, 5])

Kromosom[2]: ([1, 3, 24], [2, 5, 13], [3, 1, 27], [4, 4, 9], [5, 1, 21], [6, 4, 23], [7, 3, 19], [8, 5, 7], [9, 4, 26], [10, 0, 23], [11, 1, 12], [12, 5, 16], [13, 3, 17], [14, 5, 25], [15, 0, 2], [16, 0, 5], [17, 2, 18], [18, 0, 12], [19, 4, 23], [20, 4, 7], [21, 4, 17], [22, 3, 18], [23, 1, 2], [24, 2, 27], [25, 0, 5])

Kromosom[3]: ([1, 0, 18], [2, 2, 9], [3, 3, 18], [4, 2, 21], [5, 0, 17], [6, 0, 11], [7, 4, 0], [8, 0, 18], [9, 2, 5], [10, 0, 1, 24], [11, 4, 22], [12, 4, 21], [13, 3, 0], [14, 4, 7], [15, 2, 10], [16, 1, 10], [17, 2, 19], [18, 4, 5], [19, 5, 13], [20, 4, 3], [21, 2, 28], [22, 3, 29], [23, 3, 12], [24, 0, 5], [25, 3, 11])

Kromosom[4]: ([1, 5, 26], [2, 4, 3], [3, 4, 12], [4, 1, 26], [5, 0, 9], [6, 1, 5], [7, 1, 18], [8, 5, 27], [9, 1, 9], [10, 5, 11], [11, 4, 27], [12, 1, 1], [13, 5, 7], [14, 0, 15], [15, 4, 11], [16, 0, 22], [17, 0, 5], [18, 0, 23], [19, 5, 14], [20, 5, 18], [21, 5, 26], [22, 5, 21], [23, 1, 15], [24, 4, 17], [25, 5, 28])

Kromosom[5]: ([1, 2, 20], [2, 2, 1], [3, 5, 6], [4, 2, 2], [5, 0, 23], [6, 3, 29], [7, 2, 20], [8, 3, 19], [9, 1, 13], [10, 4, 9], [11, 2, 6], [12, 0, 15], [13, 3, 7], [14, 4, 5], [15, 5, 16], [16, 2, 13], [17, 4, 10], [18, 0, 3], [19, 0, 12], [20, 3, 11], [21, 0, 22], [22, 2, 0], [23, 3, 24], [24, 1, 27], [25, 3, 22])

Kromosom[6]: ([1, 3, 24], [2, 5, 13], [3, 1, 27], [4, 4, 9], [5, 1, 21], [6, 4, 23], [7, 3, 19], [8, 5, 7], [9, 4, 26], [10, 0, 23], [11, 1, 12], [12, 5, 16], [13, 3, 17], [14, 5, 25], [15, 0, 2], [16, 0, 5], [17, 2, 18], [18, 0, 12], [19, 4, 23], [20, 4, 7], [21, 4, 17], [22, 3, 18], [23, 1, 2], [24, 2, 27], [25, 0, 5])

Kromosom[7]: ([1, 3, 24], [2, 5, 13], [3, 1, 27], [4, 4, 9], [5, 1, 21], [6, 4, 23], [7, 3, 19], [8, 5, 7], [9, 4, 26], [10, 0, 23], [11, 1, 12], [12, 5, 16], [13, 3, 17], [14, 5, 25], [15, 0, 2], [16, 0, 5], [17, 2, 18], [18, 0, 12], [19, 4, 23], [20, 4, 7], [21, 4, 17], [22, 3, 18], [23, 1, 2], [24, 2, 27], [25, 0, 5])

Kromosom[8]: ([1, 1, 22], [2, 0, 6], [3, 2, 29], [4, 1, 19], [5, 5, 13], [6, 2, 1], [7, 4, 13], [8, 4, 10], [9, 5, 13], [10, 0, 25], [11, 2, 22], [12, 1, 6], [13, 3, 2], [14, 5, 9], [15, 5, 17], [16, 2, 0], [17, 0, 17], [18, 3, 13], [19, 5, 28], [20, 0, 21], [21, 5, 18], [22, 1, 11], [23, 0, 6], [24, 5, 7], [25, 3, 21])

Kromosom[9]: ([1, 1, 22], [2, 5, 10], [3, 1, 27], [4, 4, 7], [5, 4, 19], [6, 5, 22], [7, 0, 13], [8, 4, 10], [9, 4, 6], [10, 5, 24], [11, 2, 14], [12, 2, 12], [13, 3, 1], [14, 1, 14], [15, 4, 7], [16, 2, 10], [17, 3, 12], [18, 4, 17], [19, 4, 26], [20, 0, 3], [21, 5, 16], [22, 3, 6], [23, 2, 29], [24, 3, 23], [25, 3, 23])

Kromosom[10]: ([1, 2, 20], [2, 2, 1], [3, 5, 6], [4, 2, 2], [5, 0, 23], [6, 3, 29], [7, 2, 20], [8, 3, 19], [9, 1, 13], [10, 4, 9], [11, 2, 6], [12, 0, 15], [13, 3, 7], [14, 4, 5], [15, 5, 16], [16, 2, 13], [17, 4, 10], [18, 0, 3], [19, 0, 12], [20, 3, 11], [21, 0, 22], [22, 2, 0], [23, 3, 24], [24, 1, 27]

, [25, 3, 22])

Kromosom[11]: ([1, 0, 18], [2, 2, 9], [3, 3, 18], [4, 2, 21], [5, 0, 17], [6, 0, 11], [7, 4, 0], [8, 0, 18], [9, 2, 5], [10, 1, 24], [11, 4, 22], [12, 4, 21], [13, 3, 0], [14, 4, 7], [15, 2, 10], [16, 1, 10], [17, 2, 19], [18, 4, 5], [19, 5, 13], [20, 4, 3], [21, 2, 28], [22, 3, 29], [23, 3, 12], [24, 0, 5], [25, 3, 11])

Kromosom[12]: ([1, 2, 20], [2, 2, 1], [3, 5, 6], [4, 2, 2], [5, 0, 23], [6, 3, 29], [7, 2, 20], [8, 3, 19], [9, 1, 13], [10, 4, 9], [11, 2, 6], [12, 0, 15], [13, 3, 7], [14, 4, 5], [15, 5, 16], [16, 2, 13], [17, 4, 10], [18, 0, 3], [19, 0, 12], [20, 3, 11], [21, 0, 22], [22, 2, 0], [23, 3, 24], [24, 1, 27], [25, 3, 22])

Kromosom[13]: ([1, 0, 4], [2, 0, 13], [3, 1, 25], [4, 2, 22], [5, 3, 6], [6, 5, 4], [7, 3, 5], [8, 1, 8], [9, 5, 21], [10, 2, 1], [11, 1, 16], [12, 3, 1], [13, 2, 9], [14, 0, 22], [15, 4, 26], [16, 1, 29], [17, 3, 8], [18, 3, 26], [19, 4, 25], [20, 2, 20], [21, 3, 18], [22, 1, 29], [23, 5, 6], [24, 0, 16], [25, 1, 8])

Kemudian k untuk kromosom akan dipilih sebagai induk jika $R[k] < pc$, dari nomor acak R di atas. Dan berikut dibawah ini induk (Parents) yang terpilih yaitu: kromosom [0], kromosom [1], kromosom [2], kromosome [3], kromosom [4] dan kromosom [5],

Parent[0] : 0
Parent[1] : 1
Parent[2] : 3
Parent[3] : 6
Parent[4] : 9
Parent[5] : 10

Selanjutnya adalah *Crossover* proses ini dilakukan dengan menghasilkan angka acak dengan batas 0 sampai (total kromosom induk [i]). Lalu kromosom akan dipotong dari Gen ke tertentu. Di bawah ini adalah posisi gen dipilih sebagai titik potong untuk proses *crossover*. Offspring sendiri merupakan kromosom – kromosom yang dibentuk dari kromosom generasi sebelumnya disebut sebagai anak.

Offspring: 0
Offspring: 5
Offspring: 14
Offspring: 5
Offspring: 12
Offspring: 18

Proses yang terakhir adalah mutasi. Pada proses ini rumusnya yang digunakan adalah sebagai berikut:

$$\text{Total Gen} = (\text{jumlah kromosom} * \text{jumlah gen per kromosom}) \text{ Kromosom } 14 * \text{Gen per kromosom } 25 = \text{maka total Generasi adalah } 350.$$

Dari serangkaian proses yang telah dilakukan diatas maka hasil yang didapat berupa penjadwalan dengan *Fitness* Terbaik 1 Pada Generasi Ke -7. Hasil ini diambil dari sistem penjadwalan dengan membangkitkan kromosom 14 dan Generasi 25 dan berikut dibawah ini:

FITNESS TERBAIK: 1

GENERASI: 7

CROMOSSOM TERBAIK: Kromosom [0]: ([,],[1,0,18],[2,2,9],[3,5,6],[4,2,2],[5,0,23],[6,3,29],[7,2,20],[8,3,19],[9,1,13],[10,4,9],[11,2,6],[12,0,15],[13,3,7],[14,4,5],[15,5,16],[16,2,13],[17,4,10],[18,0,3],[19,0,12],[20,3,11],[21,0,22],[22,2,0],[23,3,24],[24,0,5])

Implementasi algoritma genetika menghasilkan mapping penjadwalan antivirus seperti gambar dibawah:

| No | Hari | Jam | PIC | Manufaktur | Divisi | User |
|----|--------|---------------|--------|-------------------|-------------------|----------------------|
| 1 | Senin | 08:00 - 08:45 | Thoriq | Lenovo V130 | HR | Dwi Hariana Pane |
| 2 | Senin | 10:15 - 11:00 | Thoriq | Lenovo V130 | Finance | Achmad Reza Adriansa |
| 3 | Senin | 13:15 - 14:00 | Wawan | Lenovo V130 | HR | Dini Sekar |
| 4 | Senin | 15:15 - 16:00 | Wawan | Lenovo V310 | Finance | Dwi Setyo |
| 5 | Senin | 15:15 - 16:00 | Alif | Dell Inspiron | Tele Marketing | Adit Triyadi |
| 6 | Selasa | 08:00 - 08:45 | Adam | Apple Macbook Pro | Software Engineer | Achmad Jafar |
| 7 | Selasa | 08:00 - 08:45 | Yoga | Lenovo V130 | People Operations | Andhika Nirmalasari |
| 8 | Selasa | 09:15 - 10:00 | Aldi | Apple Macbook Pro | Software Engineer | Adillah Azmi |
| 9 | Selasa | 13:15 - 14:00 | Alif | Apple Macbook Pro | Software Engineer | Achmad Fauzi |
| 10 | Selasa | 13:15 - 14:00 | Thoriq | Lenovo V130 | People Operations | Adetia Suhartini |
| 11 | Selasa | 14:15 - 15:00 | Yoga | Lenovo Thinkpad | Finance | Dinar Arindani |
| 12 | Selasa | 15:15 - 16:00 | Alif | Lenovo V130 | Sales | Doly Feriandy |
| 13 | Rabu | 08:00 - 08:45 | Aldi | Apple Macbook Pro | Software Engineer | Dion Hiananto |
| 14 | Rabu | 09:15 - 10:00 | Adam | Lenovo V130 | Finance | Adeline |
| 15 | Rabu | 09:15 - 10:00 | Thoriq | Apple Macbook Pro | Software Engineer | Dimas Hendria |
| 16 | Rabu | 13:15 - 14:00 | Wawan | Lenovo V130 | Sales | Adhy Wibawa |
| 17 | Rabu | 14:15 - 15:00 | Adam | Lenovo V130 | People Operations | Dien Marbela |
| 18 | Kamis | 08:00 - 08:45 | Aldi | Lenovo V130 | People Operations | Achmad Farih |
| 19 | Kamis | 09:15 - 10:00 | Alif | Apple Macbook Air | Software Engineer | Ade Saputra |
| 20 | Kamis | 10:15 - 11:00 | Aldi | Lenovo V130 | Sales | Ade Isni |
| 21 | Kamis | 14:15 - 15:00 | Adam | Lenovo V130 | Sales | Dony Putranto |
| 22 | Kamis | 15:15 - 16:00 | Yoga | Lenovo V130 | HR | Adam Farid Alhadli |
| 23 | Jumat | 08:00 - 08:45 | Yoga | Lenovo V130 | Sales | Dwi Nurcahyo |
| 24 | Jumat | 15:15 - 16:00 | Wawan | Lenovo V130 | Tele Marketing | Ade Achmad |

Gambar 13. Hasil Implementasi Algoritma Genetika

6/17/2021

Cetak Laporan

| No | Hari | Jam | PIC | Manufaktur | Divisi | User |
|----|--------|---------------|--------|-------------------|-------------------|----------------------|
| 1 | Senin | 08:00 - 08:45 | Thoriq | Lenovo V130 | HR | Dwi Hariana Pane |
| 2 | Senin | 10:15 - 11:00 | Thoriq | Lenovo V130 | Finance | Achmad Reza Adriansa |
| 3 | Senin | 13:15 - 14:00 | Wawan | Lenovo V130 | HR | Dini Sekar |
| 4 | Senin | 15:15 - 16:00 | Wawan | Lenovo V310 | Finance | Dwi Setyo |
| 5 | Senin | 15:15 - 16:00 | Alif | Dell Inspiron | Tele Marketing | Adit Triyadi |
| 6 | Selasa | 08:00 - 08:45 | Adam | Apple Macbook Pro | Software Engineer | Achmad Jafar |
| 7 | Selasa | 08:00 - 08:45 | Yoga | Lenovo V130 | People Operations | Andhika Nirmalasari |
| 8 | Selasa | 09:15 - 10:00 | Aldi | Apple Macbook Pro | Software Engineer | Adillah Azmi |
| 9 | Selasa | 13:15 - 14:00 | Alif | Apple Macbook Pro | Software Engineer | Achmad Fauzi |
| 10 | Selasa | 13:15 - 14:00 | Thoriq | Lenovo V130 | People Operations | Adetia Suhartini |
| 11 | Selasa | 14:15 - 15:00 | Yoga | Lenovo Thinkpad | Finance | Dinar Arindani |
| 12 | Selasa | 15:15 - 16:00 | Alif | Lenovo V130 | Sales | Doly Feriandy |
| 13 | Rabu | 08:00 - 08:45 | Aldi | Apple Macbook Pro | Software Engineer | Dion Hiananto |
| 14 | Rabu | 09:15 - 10:00 | Adam | Lenovo V130 | Finance | Adeline |
| 15 | Rabu | 09:15 - 10:00 | Thoriq | Apple Macbook Pro | Software Engineer | Dimas Hendria |
| 16 | Rabu | 13:15 - 14:00 | Wawan | Lenovo V130 | Sales | Adhy Wibawa |
| 17 | Rabu | 14:15 - 15:00 | Adam | Lenovo V130 | People Operations | Dien Marbela |
| 18 | Kamis | 08:00 - 08:45 | Aldi | Lenovo V130 | People Operations | Achmad Farih |
| 19 | Kamis | 09:15 - 10:00 | Alif | Apple Macbook Air | Software Engineer | Ade Saputra |
| 20 | Kamis | 10:15 - 11:00 | Aldi | Lenovo V130 | Sales | Ade Isni |
| 21 | Kamis | 14:15 - 15:00 | Adam | Lenovo V130 | Sales | Dony Putranto |
| 22 | Kamis | 15:15 - 16:00 | Yoga | Lenovo V130 | HR | Adam Farid Alhadli |
| 23 | Jumat | 08:00 - 08:45 | Yoga | Lenovo V130 | Sales | Dwi Nurcahyo |
| 24 | Jumat | 15:15 - 16:00 | Wawan | Lenovo V130 | Tele Marketing | Ade Achmad |

Gambar 14. Gambar Hasil Penjadwalan





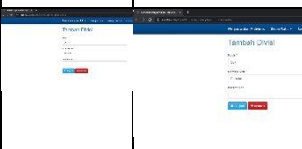
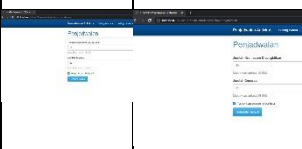

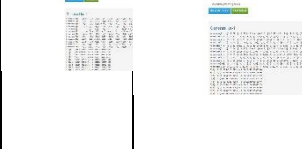
Berdasarkan proses yang telah dilakukan dari 24 sample data yang digunakan dalam proses penyusunan dan pembagian jadwal mengenai antivirus berhasil dilakukan tanpa adanya jadwal yang berbenturan, *fitness* terbaik didapatkan dari generasi ke-7 kromosom [0] dari 14 Kromosom dan 25 Generasi yang dibangkitkan.

Testing

Pengujian yang dilakukan menggunakan metode *Black-box Testing* yang dimana pengujian ini didasarkan pada detail aplikasi seperti tampilan aplikasi, fungsi - fungsi yang ada pada aplikasi, dan kesesuaian alur fungsi yang diinginkan sebelumnya sesuai kebutuhan. teknik yang digunakan yaitu *Equivalence Partitioning* dimana teknik ini membagi domain input dari suatu program kedalam kelas-kelas data sehingga *test case* dapat diperoleh. Berikut dibawah ini pengujian yang telah dilakukan:

Tabel 5. Hasil Pengujian

| No | Skenario Pengujian | Test case | Hasil Yang Diharapkan | Hasil |
|----|--|-----------|-----------------------|--------|
| 1 | Masuk Halaman Login | | | Sesuai |
| 2 | Masuk Dashboard Utama | | | Sesuai |
| 3 | Ke Menu Input Data Jam dan Proses Input Data | | | Sesuai |

| | | | |
|----|--|---|--------|
| 4 | Ke Menu Input Data Hari dan Proses Input Data |  | Sesuai |
| 5 | Masuk Ke Menu Input Data User dan Proses Input Data |  | Sesuai |
| 6 | Masuk Ke Menu Input Data PIC dan Proses Input Data |  | Sesuai |
| 7 | Masuk Ke Menu Input Data Manufaktur dan Proses Input Data |  | Sesuai |
| 8 | Masuk Ke Menu Input Data Divisi dan Proses Input Data |  | Sesuai |
| 9 | Masuk Ke Menu Generate Penjadwalan dan Proses Input Kromosom dan Generasi |  | Sesuai |
| 10 | Menampilkan Hasil Perhitungan Algoritma Genetika dan Penjadwalan Antivirus |  | Sesuai |
| | |  | Sesuai |

Analisa

Berdasarkan serangkaian proses yang telah dilakukan diatas dari mulai pembangkitan kromosom dan generasi yang dilakukan secara acak maka dapat dihasilkan penjadwalan yang lebih optimal daripada sebelumnya. Akan tetapi proses evolusi yang dilakukan oleh algoritma genetika tidak selamanya menghasilkan optimasi penjadwalan dengan hasil

optimum yang pasti secara global, dikarenakan proses yang dilakukan bergantung pada evolusi dari kromosom – kromosom yang dibangkitkan sampai mendapatkan hasil terbaik dari sebuah generasi melihat dari nilai *fitness* yang dihasilkan. Pada proses algoritma genetika terdapat fungsi objective dari sebuah kromosom dan dibatasi oleh parameter yang ditentukan seperti *clash* yang dibuat agar hasil penjadwalan tidak berbenturan. *Clash* sendiri merupakan representasi pembatas agar kesalahan yang terjadi pada penjadwalan seperti benturan jam bisa diminimalisir. Beberapa *clash* yang digunakan pada implementasi algoritma genetika penjadwalan antivirus ini antara lain *clash* divisi, *clash* user dan *clash* manufaktur. Dari ketiga *clash* ini menjadi awal proses operasi yang dilakukan agar hasil penjadwalan yang diinginkan sesuai dengan yang diharapkan..

4. Kesimpulan

Berdasarkan Penelitian yang telah dilakukan mengacu dari berbagai referensi penjadwalan kuliah pada masalah optimasi, bahwa hasil dari aplikasi yang dirancang dengan memanfaatkan Algoritma Genetika sebagai solusi dari masalah penjadwalan yang terjadi pada perusahaan x mengenai program instalasi antivirus ini dapat menghasilkan penjadwalan yang lebih optimal dengan tidak terjadinya benturan jam dan hari setelah beberapa kali percobaan yang dilakukan dengan cara pembangkitan kromosom dan generasi secara acak hingga mendapatkan hasil terbaik berdasarkan nilai *fitness* tertinggi yang digunakan pada solusi masalah penjadwalan ini. Diharapkan sistem penjadwalan yang baru dapat memberikan kemudahan terutama dalam meningkatkan efektivitas dan efisiensi terhadap waktu yang ada.

5. Daftar Pustaka

[1] Ferdyawan, F. and Hajjah, A., 2020. Penerapan Algoritma Genetika dalam Optimasi Penjadwalan Proyek. *Jurnal Mahasiswa Aplikasi Teknologi Komputer dan Informasi (JMApTeKsi)*, 2(1), pp.50-55.

[2] Oktarina, D. and Hajjah, A., 2019. Perancangan Sistem Penjadwalan Seminar Proposal dan Sidang Skripsi Dengan Metode Algoritma Genetika. *JOISIE (Journal Of Information Systems And Informatics Engineering)*, 3(1), pp.32-40.

- [3] Ginantra, N.L.W.S.R. and Anandita, I.B.G., 2019. Implementasi Algoritma Genetika Berbasis Web Pada Sistem Penjadwalan Mengajar Di Smk Dwijendra Denpasar. *Jurnal Teknologi Informasi dan Komputer*, 5(1).
- [4] Ivan, I., Raphael, S. and Agung, H., 2018. Aplikasi Penjadwalan Mata Pelajaran di SMAN 31 Menggunakan Algoritma Genetika Berbasis WEB. *Simetris: Jurnal Teknik Mesin, Elektro dan Ilmu Komputer*, 9(1), pp.641-656.
- [5] Suwarjono, S. and Susanto, S., 2018. Sistem Penjadwalan Perkuliahan Pada Universitas Musamus Menggunakan Algoritma Genetika Berbasis Web. *Musamus Journal Of Research Information and Communication Technology*, 1(1), pp.12-17.
- [6] Josi, A., 2017. Implementasi Algoritma Genetika Pada Aplikasi Penjadwalan Perkuliahan Berbasis Web Dengan Mengadopsi Model Waterfall (Studi Kasus: STMIK Prabumulih). *Jurnal Informatika: Jurnal Pengembangan IT*, 2(2), pp.77-83.
- [7] Damanik, R.R., 2020. APLIKASI Penjadwalan Perawat Rs Porsea Menggunakan Algoritma Genetika. *Journal Information System Development (ISD)*, 5(1), pp.12-18.
- [8] Susanto, S., Rachmat, R. and Hardiantono, D., 2018. Rancang Bangun Sistem Penjadwalan Kuliah Jurusan Teknik Informatika Fakultas Teknik Universitas Musamus Merauke Menggunakan Algoritma Genetika. *Musamus Journal of Technology & Information*, 1(01), pp.33-41.
- [9] Haupt, R.L. and Haupt, S.E., 2004. *Practical genetic algorithms*. John Wiley & Sons.
- [10] Suratno, T., Rarasati, N. and Gusmanely, Z., 2019. Optimization of Genetic Algorithm for Implementation Designing and Modeling in Academic Scheduling. *Eksakta: Berkala Ilmiah Bidang MIPA (E-ISSN: 2549-7464)*, 20(1), pp.17-24.
- [11] Christian, A., Sujaini, H. and Negara, A.B.P., 2017. Implementasi Sistem Penjadwalan Akademik Fakultas Teknik Universitas Tanjungpura Menggunakan Metode Algoritma Genetika. *Jurnal Sistem dan Teknologi Informasi (JUSTIN) Vol, 1(2)*, p.28.
- [12] Alfred, R. and Yu, H.F., 2020. Automated scheduling of hostel room allocation using genetic algorithm. In *Data Management, Analytics and Innovation* (pp. 151-160). Springer, Singapore.
- [13] M. H. Mubin, B. Arifitama, and Yaddarabullah, 2020. Implementation of genetic algorithms for scheduling the memorization of Al-Qur'an at the Markaz Al-Qur'an, *Int. J. Sci. Technol. Res.*, vol. 9, no. 4, pp. 3251–3258.
- [14] Lubis, N. and Ginting, G., 2016. Perancangan Aplikasi Penjadwalan Kereta Api dengan Menggunakan Algoritma Genetika (Studi Kasus: Stasiun Kereta Api Bandar Khalipah Medan). *JURIKOM (Jurnal Riset Komputer)*, 3(6).
- [15] Taoroni, M., 2018. *Travelling salesman problem di pt. Agung jaya perkasa dengan menggunakan algoritma genetika* (Doctoral dissertation, Universitas Widyatama).