

Implementasi Algoritma “Particle Swarm Optimization” (PSO) Penjadwalan Belajar Mengajar

Alfina Febiani¹, Agung Mulyo Widodo², Nizirwan Anwar³, Binastya Anggara Sekti⁴,
Alivia Yulfitri⁵

^{1,2,3,4}Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Esa Unggul

E-mail: alfinafebiani@student.esaunggul.ac.id, agung.mulyo@esaunggul.ac.id,
nizirwan.anwar@esaunggul.ac.id, anggara@esaunggul.ac.id, alivia@esaunggul.ac.id

ABSTRAK

Penjadwalan belajar mengajar adalah pengaturan perencanaan belajar mengajar yang meliputi mata pelajaran, guru, waktu dan tempat pada sekolah. Pada umumnya penjadwalan belajar mengajar disediakan dalam sebuah tabel hari dalam seminggu yang terdiri dari slot waktu yang terdapat mata pelajaran, hari, jam, serta pengajar yang sesuai dengan mata pelajaran yang diajarkan. Dalam penyusunan jadwal belajar mengajar disebuah sekolah dibutuhkan alokasi dan penentuan guru sebagai salah satu elemen penting didalamnya. Pada proses penjadwalan secara manual, sangatlah sering terjadi bentrok jadwal antara guru mengajar pada kelas yang berbeda dalam waktu yang sama sehingga harus memiliki waktu luang untuk menyesuaikan jadwal yang baru agar didapatkan hasil yang maksimal. Maka dari itu diperlukan solusi untuk menyelesaikan permasalahan penjadwalan yaitu dengan aplikasi penjadwalan. Optimasi penjadwalan belajar mengajar adalah mempermudah dan memaksimalkan penyusunan penjadwalan dan Algoritma *Particle Swarm Optimization* (PSO) dapat mengoptimalkan masalah sehingga didapatkan solusi berupa jadwal pelajaran. Algoritma *Particle Swarm Optimization* (PSO) adalah algoritma pencarian yang menggunakan banyak individu, atau partikel, dan dikelompokkan ke dalam segerombolan. Masing-masing partikel ini akan mewakili solusi kandidat untuk optimasi masalah. Pengembangan sistem yang dibuat ini menggunakan metode *waterfall*, *framework laravel* dengan bahasa pemrograman PHP dan MySQL sebagai tempat untuk menyimpan seluruh data yang di kelola pada sistem. Pengujian sistem ini menggunakan pengujian *Black box*, dimana jenis pengujian ini dianggap paling tepat untuk mewakili penggunaan menu yang dibuat.

Kata kunci : Penjadwalan, *Particle Swarm Optimization*, *Fitness*

ABSTRACT

Teaching and learning scheduling is the arrangement of teaching and learning planning which includes subjects, teachers, time and place at school. In general, teaching and learning scheduling is provided in a table of days of the week consisting of time slots containing subjects, days, hours, and teachers according to the subjects being taught. In preparing a teaching and learning schedule in a school, the allocation and determination of teachers is needed as one of the important elements in it. In the manual scheduling process, there are often schedule clashes between teachers teaching different classes at the same time, so you have to have free time to adjust the new schedule to get maximum results. Therefore a solution is needed to solve the scheduling problem, namely with a scheduling application. Optimization of teaching and learning scheduling is to simplify and maximize the preparation of scheduling and the Particle Swarm Optimization (PSO) Algorithm can optimize the problem so that a solution is obtained in the form of a lesson schedule. The Particle Swarm Optimization (PSO) algorithm is a search algorithm that uses many individuals, or particles, and groups them into swarms. Each of these particles will represent a candidate solution for the optimization problem. The development of the system created uses the waterfall method, the Laravel framework with the PHP and MySQL programming languages as a place to store all data that is managed on the system. Testing this system uses Black box testing, where this type of test is considered the most appropriate to represent the use of the menus that are made.

Keyword : Scheduling, *Particle Swarm Optimization*, *Fitness*

1. PENDAHULUAN

Salah satu pemanfaatan kecerdasan buatan yaitu berupa pemecahan dalam penjadwalan. Penjadwalan berasal dari kata jadwal yang mendapat imbuhan pen yang memiliki arti pembagian waktu berdasarkan rencana pengaturan urutan kerja atau daftar atau tabel kegiatan atau rencana kegiatan dengan pembagian waktu pelaksanaan yang mendetail [1]. Dalam membuat jadwal yang baik, ada beberapa batasan yang harus diperhatikan, agar dapat sesuai kebutuhan, agar jadwal yang dibuat dapat sesuai dalam penggunaannya, agar proses pengolahan data penjadwalan cepat, efektif, efisien dan akurat untuk meningkatkan standar mutu pendidikan [2].

Penjadwalan hampir di semua sekolah yang meliputi jadwal mata pelajaran dan pembagian guru di setiap sekolah yang ada masih menggunakan cara manual, yaitu pihak Tata Usaha (TU) atau Wakil Kurikulum yang menuliskan nama guru dan mata pelajarannya yang kemudian akan ditentukan waktu mengajar dan kelasnya. Pada proses penjadwalan secara manual, sangatlah sering terjadi bentrok jadwal antara guru mengajar pada kelas yang berbeda dalam waktu yang sama sehingga harus memiliki waktu luang untuk menyesuaikan jadwal yang baru agar didapatkan hasil yang maksimal. Pekerjaan seperti ini tidak mudah dan diperlukan pengaturan yang cukup rumit. Dalam penyusunan jadwal kegiatan belajar mengajar disebuah sekolah dibutuhkan alokasi dan penentuan guru sebagai salah satu elemen penting didalamnya. Selain itu juga yang menjadi permasalahan umum yang selalu menjadi beban pihak kurikulum disekolah tersebut dengan ketebatasan guru yang ada, pihak kurikulum dituntut agar tetap memberikan pelayanan yang maksimal kepada siswa khususnya kelas-kelas yang ada [3].

Pada saat ini penyusunan jadwal di SMP Negeri 264 Jakarta sudah

tekomputerisasi yaitu dengan menggunakan *microsoft excel*, akan tetapi penggunaan cara tersebut juga dapat menyebabkan bentroknnya jadwal mata pelajaran dan ketidaksesuaian jadwal mata pelajaran. Hal ini akan sangat mengganggu efektifitas waktu belajar dan efektifitas guru dalam mengajar. Penyusunan jadwal yang baik perlu memperhatikan aspek seperti ketersediaan ruang kelas, jumlah guru, jumlah mata pelajaran dan jadwal guru yang bersangkutan [4].

Berdasarkan permasalahan tersebut diperlukan suatu aplikasi yang mempermudah dalam penyusunan jadwal dengan menggunakan algoritma penjadwalan yang tepat. Algoritma *Particle Swarm Optimization* (PSO) merupakan salah satu algoritma yang dapat menyelesaikan masalah penjadwalan dibeberapa peneliti sebelumnya. Algoritma *Particle Swarm Optimization* (PSO) adalah algoritma pencarian yang menggunakan banyak individu, atau partikel, dan dikelompokkan ke dalam segerombolan. Masing masing partikel ini akan mewakili solusi kandidat untuk optimasi masalah [5].

2. LANDASAN TEORI

Penjadwalan

Penjadwalan merupakan pengurutan pembuatan atau pengerjaan produk secara menyeluruh yang dikerjakan pada beberapa buah mesin.dengan demikian masalah *sequencing* senantiasa melibatkan pengerjaan sejumlah komponen yang sering disebut dengan istilah job. Job sendiri masih merupakan komposisi dari sejumlah elemen-elemen dasar yang disebut aktivitas atau operasi. Tiap aktivitas atau operasi ini membutuhkan alokasi sumber daya tertentu selama periode waktu tertentu

yang sering disebut dengan waktu proses [6].

Algoritma *Particle Swarm Optimization* (PSO)

Algoritma *Particle Swarm Optimization* (PSO) diperkenalkan oleh Dr. Eberhart dan Dr. Kennedy pada tahun 1995, yang merupakan algoritma optimasi yang meniru proses yang terjadi dalam kehidupan populasi burung dan ikan dalam bertahan hidup. Sejak diperkenalkan pertama kali, algoritma *Particle Swarm Optimization* (PSO) berkembang cukup pesat, baik dari sisi aplikasi maupun dari sisi pengembangan metode yang digunakan pada algoritma tersebut. Hal ini disebabkan, karena algoritma *Particle Swarm Optimization* (PSO) adalah algoritma optimasi yang mudah dipahami, cukup sederhana, dan memiliki untuk kerja yang sudah terbukti handal [7].

Algoritma *Particle Swarm Optimization* (PSO) adalah salah satu algoritma kecerdasan buatan (*Artificial Intelligence*), termasuk ke dalam kategori *swarm intelligence* yang dimana algoritma ini terinspirasi oleh perilaku sosial kolektif koloni burung atau binatang. Pergerakan burung yang direpresentasikan sebagai pergerakan partikel ditentukan oleh nilai posisi sebelumnya dan nilai kecepatan saat ini. Nilai posisi dari partikel ini kemudian merepresentasikan solusi yang mungkin pada kasus optimasi dan nilai kecepatan digunakan untuk update posisi partikel [8].

Berikut 3 tahapan Algoritma *Particle Swarm Optimization* (PSO) [9]:

1. Pembangkitan posisi dan *velocity* awal partikel.

Dalam proses pembangkitan posisi X dan kecepatan V menggunakan fungsi *rnd*. Dimana posisi X_k^i dan kecepatan V_k^i dari kumpulan partikel dibangkitkan secara random menggunakan batas atas (X_{max}) dan batas bawah (X_{min}), seperti persamaan berikut:

$$X_0^i = X_{min} + rnd (X_{max} - X_{min}) \quad (2.1)$$

$$V_0^i = X_{min} + rnd (X_{max} - X_{min}) \quad (2.2)$$

Dimana:

X_0^i = Posisi awal

V_0^i = Kecepatan awal

X_{min} = Batas bawah

X_{max} = Batas atas

rnd = Nilai random antara rentang nilai 0 dan 1

Pada penjadwalan ini posisi partikel diwakili oleh slot-slot untuk menampung posisi random dan kecepatan random. Dengan proses inialisasi ini maka kumpulan partikel dapat terdistribusi secara random. Vektor dapat ditunjukkan sebagai berikut:

$$X_k^i = (X_k^{i1}, X_k^{i1}, \dots, X_k^{in})^T$$

$$V_k^i = (V_k^{i1}, V_k^{i1}, \dots, V_k^{in})^T$$

2. Menentukan nilai *fitness* masing masing partikel

Dalam nilai *fitness* penjadwalan ini menentukan banyaknya pelanggaran *constraint* yang harus dioptimasi.

Batasan *constraint* yang digunakan dalam pengoptimasian aplikasi penjadwalan ini yaitu:

- a. Tidak adanya bentok antar guru
- b. Tidak adanya bentrok mata pelajaran
- c. Tidak adanya bentrok kelas

Jika pada masing-masing partikel terjadi pelanggaran terhadap *constraint-constraint* diatas maka nilai *fitness* masing-masing partikel akan di increment sebanyak satu untuk setiap pelanggaran yang terjadi.

3. *Update* kecepatan dan posisi

Perumusan update kecepatan mencakup beberapa parameter random, berikut ada tiga parameter yang memengaruhi pencarian, yaitu *inertia factor* (w), *self confidence* ($c1$), *swarm confidence* ($c2$) akan digabungkan dalam satu penyajian yang ditunjukkan seagai berikut:

$$V_{k+1}^i = w * V_k^i + c_1 * rnd * (p^i - X_k^i) + c_2 * rnd * (p_k^g - X_k^i) \quad (2.3)$$

w = *Inertia factor*, digunakan untuk mengontrol pengaruh kecepatan sebelumnya dikecepatan sekarang, mempengaruhi kemampuan *exploration*

(menjajah) *local* dan *global* selama proses pencarian. Nilai w memiliki rentang $0.4 - 0.9$

V_k^i = Kecepatan sekarang
 X_k^i = Posisi sekarang
 $c_1 c_2$ = Rentang nilai c_1 dan c_2 berkisar antara $0 - 4$ (*swarm intelligent*)
 rnd = Nilai random antara rentang nilai 0 dan 1
 p^i = *Local best*, posisi terbaik dari semua partikel

p_k^g = *Global best* pada *swarm* saat ini
 Tahap terakhir dalam *update* posisi tiap partikel dengan nilai *update* kecepatan, seperti persamaan berikut:

$$X_{k+1}^i = X_k^i + V_{k+1}^i \quad (2.4)$$

Dimana:

X_{k+1}^i = Posisi pencarian
 X_k^i = Arah Pencarian
 V_{k+1}^i = Posisi sekarang

Dari ketiga tahapan diatas akan diulang samapi kriteria kekonvergenan terpenuhi, kriteria kekonvergenan sangat penting dalam menghindari penambahan fungsi evaluasi setelah *optimum* didapatkan, namun kriteria kekonvergenan tidak selalu mutlak diperlukan, penetapan jumlah iterasi maksimal juga dapat digunakan sebagai *stopping condition* dari algoritma.

```

for setiap partikel
    Inialisasi partikel menggunakan persamaan (2.1) dan (2.2)
end
repeat
    for setiap partikel
        Hitung nilai fitness
        if nilai fitness baru lebih baik daripada nilai fitness lama
            Update nilai fitness dari partikel tersebut
        end
    end
    Pilih partikel dengan nilai fitness terbaik diantara semua partikel tetangganya dan simpan nilai fitness terbaik tersebut
for setiap partikel
    
```

```

        Hitung velocity partikel menggunakan persamaan (2.3)
        Update posisi partikel menggunakan persamaan (2.4)
    end
until (KriteriaBerhenti == true)
    
```

Tabel 1. Pseudo Code Algoritma PSO

Fitness

Fungsi dari *fitness* digunakan untuk mengukur tingkat kebaikan atau kesesuaian (*fitness*) suatu solusi dengan solusi yang dicari. Fungsi *fitness* sendiri bisa berhubungan langsung dengan fungsi tujuan, atau bisa juga sedikit modifikasi terhadap fungsi tujuan. Sejumlah solusi yang dibangkitkan salam populasi akan dievaluasi menggunakan fungsi *fitness*.

Jika tujuannya adalah memaksimalkan sebuah fungsi, maka fungsi *fitness* yang digunakan adalah fungsi itu sendiri, jadi $f = h$ (yang dimana f adalah fungsi *fitness*). Sedangkan jika tujuannya adalah meminimalkan fungsi h , maka fungsi h tidak bisa digunakan secara langsung. Fungsi *fitness* yang digunakan untuk masalah minimasi adalah $f = \frac{1}{h}$. Artinya, semakin kecil nilai h semakin besar nilai f . Tetapi fungsi *fitness* ini akan bermasalah jika h bisa bernilai 0 , yang mengakibatkan f bisa bernilai tak hingga. Untuk mengatasi masalah tersebut, fungsi *fitness* perlu dimodifikasi sedikit menjadi,

$$f = \frac{1}{h+a} \quad (2.5)$$

Dimana:

h = Hasil dari penjumlahan *constraint* yang dilanggar
 a = Bilangan yang dianggap sangat kecil untuk menghindari pembagian dengan 0 . Nilai a biasanya didefinisikan sebagai 0.001 atau disesuaikan dengan masalah yang akan diselesaikan.

Testing (Black Box)

Metode ini dilakukan dengan cara menjalankan atau mengeksekusi program yang dihasilkan. Dengan kemudian dapat

diamati apakah hasil yang diinginkan. Jika masih terdapat kesalahan atau terdapat hasil yang tidak sesuai dengan yang diinginkan, maka kesalahan ataupun ketidaksesuaian tersebut dicatat untuk selanjutnya dicek satu per satu dan diperbaiki. Tujuan dari metode *Black Box testing* adalah untuk mendapatkan kesalahan *output* yang dihasilkan oleh program [3].

3. METODOLOGI

Rencana Penelitian

Untuk memecahkan permasalahan yang telah diuraikan pada bab 1 sebelumnya, maka didapatkan solusi yang dapat dilakukan yaitu dengan membuat sebuah sistem aplikasi untuk mempermudah penyusunan jadwal lebih cepat dan efisien dengan memanfaatkan algoritma yang tepat. Salah satunya dengan menggunakan Algoritma *Particle Swarm Optimization* (PSO) untuk menyelesaikan masalah penjadwalan.

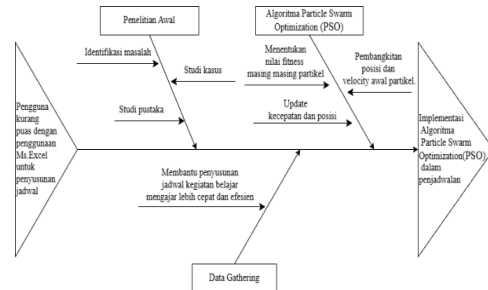
Objek Penelitian

Objek penelitian adalah salah satu hal yang menjadi sasaran penelitian. Objek penelitian dapat berupa orang, organisasi, ataupun barang yang dapat diteliti. Adapun objek penelitian ini adalah SMP Negeri 264 Jakarta.

Metode Analisis Masalah

Analisis yang digunakan dalam menjelaskan kinerja dari sistem aplikasi yang akan dibuat adalah dengan menggunakan Diagram *Fishbone*. Diagram *fishbone* biasa disebut *Cause and Effect* diagram yang merupakan sebuah diagram yang menyerupai tulang ikan yang dapat menunjukkan sebab akibat dari suatu permasalahan. Diagram *fishbone* adalah salah satu *tool* dari *2 basic quality tools*. Diagram *fishbone* digunakan saat kita ingin mengidentifikasi kemungkinan penyebab masalah dan terutama ketika sebuah *team* cenderung

jatuh berpikir pada rutinitas [10]. Berikut adalah gambar diagram *fishbone*:

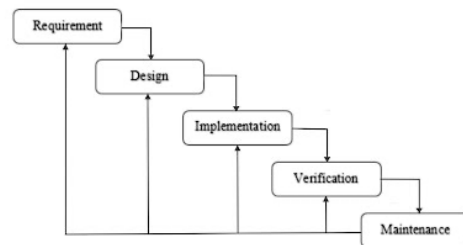


Gambar 1. Diagram *Fishbone*

Metode Pengembangan Data

Dalam metode pengembangan ini penulis menggunakan metode *Waterfall* untuk membuat dan mengembangkan sistem penjadwalan kegiatan belajar mengajar ini. Ada 5 tahapan dalam metode *Waterfall*, yaitu:

Dalam metode pengembangan ini penulis menggunakan metode *Waterfall* untuk membuat dan mengembangkan sistem penjadwalan belajar mengajar ini. Ada 5 tahapan dalam metode *Waterfall*, yaitu:



Gambar 2. *Waterfall* [11]

1. Requirement

Tahapan yang pertama dapat dilakukan proses formalitas permasalahan dengan cara mengidentifikasi masalah berdasarkan data yang telah didapatkan dari wawancara dan observasi lapangan. Identifikasi masalah bahwa proses penjadwalan kegiatan belajar mengajar belum berjalan dengan baik, dikarenakan masih ditemukan bentrok antar jadwal, sehingga mengganggu aktifitas belajar mengajar antara guru dan murid. Maka dari itu perlu adanya sistem penjadwalan

untuk mempermudah penyusunan jadwal dan menghindari bentrok antar jadwal, agar proses kegiatan belajar mengajar dapat berjalan dengan baik sesuai jadwal yang telah ditentukan.

2. Perancangan

Pada tahap ini rencana dari peneliti adalah merencanakan segala sesuatu yang berhubungan dengan sistem, mulai dari sistem yang akan dibentuk, sampai dengan cara kerja dari aplikasi sistem. Pada tahapan ini akan dihasilkan dokumen *User Requirement* atau bisa dikatakan sebagai data yang berhubungan dengan keinginan pengguna dalam pembuatan sistem.

3. Desain

Dalam tahap ini peneliti merancang dan melakukan pembuatan desain yang telah direncanakan sebelumnya. Perancangan sistem ini meliputi perancangan antarmuka pemakai (*User Interface*).

4. Pengujian

Pada tahap pengujian terdapat 2 fase, yaitu fase konstruksi dan implementasi.

- a. Fase konstruksi, adalah fase pembuatan kode program. Pada tahap ini akan dilakukan tahap perhitungan menggunakan Algoritma *Particle Swarm Optimization* (PSO) dan pengkodean metode Algoritma *Particle Swarm Optimization* (PSO) dengan menggunakan bahasa pemrograman PHP dan menggunakan MySQL sebagai *database*.
 - b. Fase implementasi, adalah fase pengujian sistem yang telah dibuat. Dalam fase ini akan dilakukan pengujian sistem dengan menggunakan teknik *Black Box Testing* atau pengujian *fungsiional* dimana pengujiannya dapat mempelajari *input* dan *output* pada aplikasi dengan hasil yang ingin dicapai.
- ## 5. Maintenance

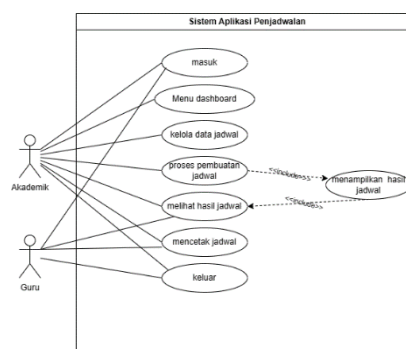
Semakin berkembangnya sistem penjadwalan, maka dibutuhkan pemeliharaan (pengembangan) sistem agar tidak terlalu lawas dan sistem yang dibuat tidak sia-sia. Pemeliharaan sistem yang dilakukan meliputi *upgrade* sistem secara berkala serta evaluasi sistem.

Usulan Rancangan Sistem

Berikut adalah usulan rancangan sistem aplikasi untuk mempermudah penyusunan jadwal kegiatan belajar mengajar menggunakan UML (*Unified Modeling Language*).

a. Use Case Diagram

Berikut merupakan gambaran *use case* diagram mengenai apa saja yang dilakukan oleh admin dan guru. Terdapat 2 aktor yaitu akademik dan guru.



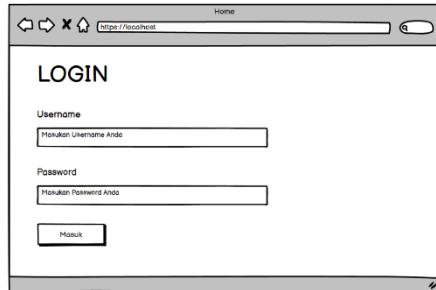
Gambar 3. Use Case Diagram

b. Activity Diagram

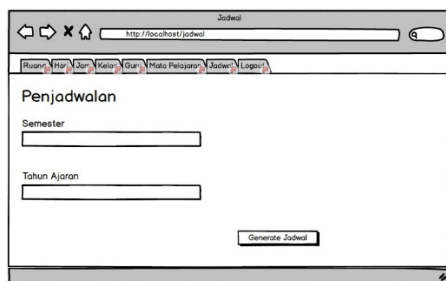
Activity diagram merupakan gambaran dari aktivitas-aktivitas yang terjadi didalam sistem. Berikut adalah gambaran *activity* diagram penjadwalan kegiatan belajar mengajar:

menjadi tiga yaitu *design* tampilan *interface* dan *design* algoritma

a. Desain Tampilan *Interface*

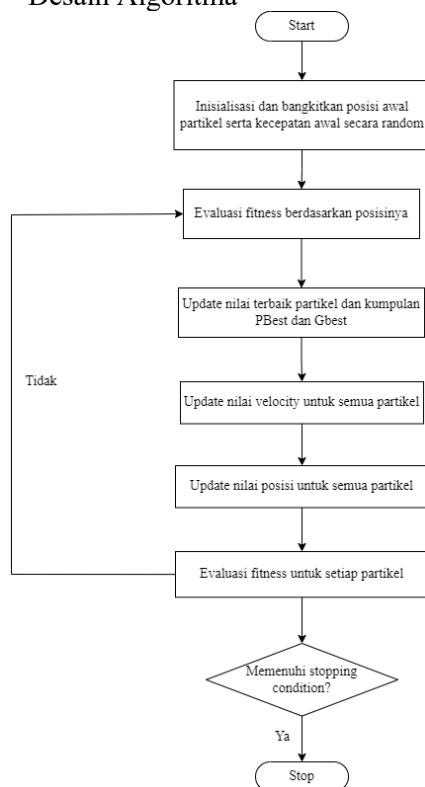


Gambar 9. Tampilan Login



Gambar 8. Tampilan Generate Jadwal

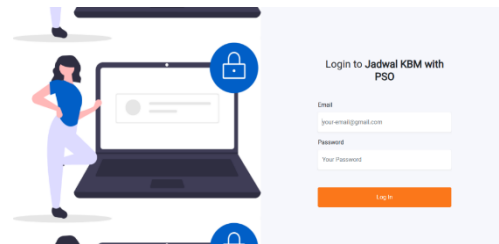
b. Desain Algoritma



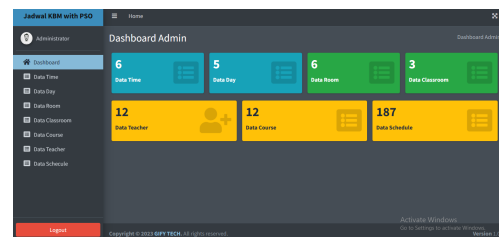
Gambar 10. Flowchart Algoritma PSO

Implementasi

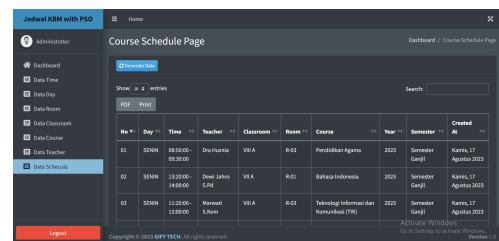
Implementasi merupakan tahap penerapan sekaligus pengujian bagi sistem baru dan juga tahap yang dimana aplikasi agar siap dioperasikan pada keadaan yang sebenarnya, pada implementasi ini ada beberapa tahapan yang harus dilakukan yaitu:



Gambar 11. Login



Gambar 12. Menu Utama



Gambar 13. Jadwal

Pengujian

Pengujian ini dilakukan dengan menggunakan pengujian *Black-Box*.

5. KESIMPULAN DAN SARAN

Kesimpulan

Kesimpulan dari penelitian implementasi algoritma *Particle Swarm Optimization* (PSO) untuk penjadwalan mata pelajaran:

1. Merancang suatu aplikasi penjadwalan belajar mengajar di SMPN 264 Jakarta dengan

menggunakan algoritma *Particle Swarm Optimization* (PSO) dilakukan dengan menggunakan metode *waterfall* dan UML sebagai perancangan sistem.

2. Aplikasi penjadwalan ini mampu menangani proses *input* data, melakukan proses pembuatan jadwal secara otomatis dan menghasilkan jadwal yang dapat ditampilkan dan dapat pula dicetak.
3. Sistem aplikasi *generate* jadwal mampu menangani pengelolaan komponen-komponen jadwal, pengelolaan pemetaan pelajaran, pembuatan jadwal dan penyimpanan hasil jadwal.

Saran

Dari hasil pembahasan yang telah dilakukan, ada beberapa saran untuk penelitian selanjutnya dengan harapan akan menjadi saran yang bermanfaat, yaitu :

1. Pada aplikasi ini digunakannya algoritma *Particle Swarm Optimization* (PSO) dalam proses pembuatan penjadwalan perkuliahan, diharapkan pada penelitian selanjutnya bisa menggunakan algoritma yang lain yang mempunyai kemampuan lebih baik dari pada algoritma *Particle Swarm Optimization* (PSO).
2. *Constraint* yang digunakan dalam penjadwalan dapat diubah sesuai dengan kebutuhan dan persyaratan pembuatan jadwal yang berlaku, sehingga program aplikasi dapat digunakan di tempat lain tanpa perlu mengubah *source code* program.
3. Algoritma *Particle Swarm Optimization* (PSO) dapat digunakan untuk memecahkan permasalahan penjadwalan lainnya, seperti

penjadwalan ruangan, penjadwalan kerja dan lain sebagainya.

Maintenance

Pada tahap terakhir ini adalah mengimplementasikan hasil *design* yang telah dibuat ke dalam aplikasi penjadwalan kegiatan belajar mengajar secara menyeluruh. Berikut merupakan penjelasan mengenai implementasi sistem:

a. Implementasi *Tools*

Implementasi sistem ini dilakukan pada perangkat keras dan perangkat lunak. Adapun perangkat keras dan perangkat lunak yang digunakan untuk membangun aplikasi ini adalah sebagai berikut:

1. Perangkat keras
 - a. Intel Core i3, 2.20 GHz
 - b. Memory (RAM) 4GB
2. Perangkat lunak
 - a. Windows 10 pro
 - b. Xampp
 - c. Microsoft Edge
 - d. Visual Studio Code

b. Implementasi Antarmuka

Pada tahapan ini, aplikasi penjadwalan sudah siap untuk digunakan atau dioperasikan pada SMP Negeri 264 Jakarta, agar dapat diketahui aplikasi penjadwalan yang telah dibuat sudah sesuai dengan rancangan antarmuka.

c. Instalasi Sistem

Instalasi sistem adalah proses untuk meng*instal* sistem operasi ke dalam suatu komputer, sehingga dapat melakukan kontrol dan manajemen perangkat keras dan juga operasi-operasi dasar sistem.

DAFTAR PUSTAKA

- [1] A. Josi, "Implementasi Algoritma Genetika pada Aplikasi Penjadwalan Perkuliahan Berbasis Web dengan Mengadopsi Model Waterfall," *J. Inform. J. Pengemb. IT*, vol. 02, no. 02, pp. 77–83, 2017, [Online]. Available:

- <http://ejournal.poltektegal.ac.id/index.php/informatika/article/view/517/554>.
- [2] M. S. A. Jemakmun, "IMPLEMENTASI ALGORITMA GENETIKA DALAM SISTEM PENJADWALAN," 2022.
- [3] E. Prasetyawan, "APLIKASI PENJADWALAN MATA PELAJARAN PADA SMK NEGERI 1 TENGGARONG MENGGUNAKAN ALGORITMA," 2015.
- [4] H. Ardiansyah and M. B. S. Junianto, "Penerapan Algoritma Genetika untuk Penjadwalan Mata Pelajaran," *J. Media Inform. Budidarma*, vol. 6, no. 1, p. 329, 2022, doi: 10.30865/mib.v6i1.3418.
- [5] A. Setiawan, L. W. Santoso, and R. Adipranata, "Penerapan Algoritma Particle Swarm Optimization (PSO) untuk Optimisasi Pembangunan Negara dalam Turn Based Strategy Game," *J. Infra*, vol. 7, no. 1, pp. 249–255, 2019.
- [6] A. T. Laksono, M. C. Utami, and Y. Sugiarti, "Pengembangan Website Program Studi sistem informasi Universitas Widyatama Menggunakan PHP dan MySQL," *Stud. Inform. J. Sist. Inf.*, vol. 9, no. 2, pp. 177–188, 2016.
- [7] D. A. R. Wati and Y. A. Rochman, "Model Penjadwalan Matakuliah Secara Otomatis Berbasis Algoritma Particle Swarm Optimization (PSO)," *J. Rekayasa Sist. Ind.*, vol. 2, no. 1, pp. 22–31, 2013.
- [8] N. Wicaksono, R. S. Wibowo, and H. Suryatmojo, "Economic Dispatch Untuk Sistem Kelistrikan Microgrid Dengan Energy Storage Berbasis Adaptive Particle Swarm Optimization," *J. Tek. ITS*, vol. 5, no. 2, 2016, doi: 10.12962/j23373539.v5i2.16096.
- [9] D. Ariani, A. Fahriza, and I. Prasetyaningrum, "Optimasi Penjadwalan Mata Kuliah di Jurusan Teknik Informatika PENS dengan Menggunakan Algoritma Particle Swarm Optimization (PSO)," *EPPIS Repos.*, no. January, pp. 1–11, 2011.
- [10] A. K. Julianto and A. Nugroho, "Prosiding SNST ke-11 Tahun 2021 Fakultas Teknik Universitas Wahid Hasyim 115," *J. Fak. Tek. Univ. Wahid Hasyim*, vol. 1, no. 1, pp. 115–121, 2021.
- [11] A. A. Wahid, "Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi," *J. Ilmu-ilmu Inform. dan Manaj. STMIK*, no. November, pp. 1–5, 2020.