

BAB II

TINJAUAN PUSTAKA

2.1 Penjadwalan

Penjadwalan adalah proses strategis yang melibatkan pengorganisasian urutan atau jadwal kegiatan dengan cara yang terencana atau peristiwa dalam waktu yang telah ditentukan. Hal ini melibatkan pengaturan waktu, sumber daya, dan prioritas guna mencapai tujuan tertentu secara efisien. Penjadwalan digunakan dalam berbagai bidang, seperti manajemen proyek, transportasi, produksi, pelayanan pelanggan, pendidikan, dan bidang lainnya. Dalam konteks manajemen proyek, penjadwalan mencakup proses penetapan waktu mulai dan selesai untuk setiap tugas, serta pengalokasian sumber daya yang diperlukan dan pengaturan urutan kegiatan untuk memastikan proyek selesai tepat waktu sesuai dengan batasan yang telah ditetapkan (Fajrianto et al., 2022).

Dalam konteks pendidikan, penjadwalan berkaitan dengan penentuan jadwal pelajaran, pengaturan waktu dan ruang kelas, serta pengelolaan jadwal kegiatan siswa dan guru agar efektif dan efisien. Penjadwalan juga dapat melibatkan faktor-faktor seperti prioritas, ketergantungan antara kegiatan, batasan sumber daya, waktu optimal, dan pencapaian tujuan yang diinginkan. Tujuan utama dari penjadwalan adalah meminimalkan waktu tunggu, mengoptimalkan penggunaan sumber daya, menghindari tabrakan jadwal, dan mencapai efisiensi secara keseluruhan (Afira & Wijaya, 2021).

SUMATERA UTARA MEDAN

2.2 Mata Pelajaran

Mata pelajaran adalah unit pembelajaran atau bidang studi yang terfokus pada pengetahuan, konsep, dan keterampilan tertentu. Mata pelajaran mencakup berbagai topik yang saling terkait dalam suatu disiplin ilmu atau area pengetahuan tertentu. Setiap mata pelajaran memiliki tujuan pembelajaran yang khas dan metode pengajaran yang sesuai untuk

memfasilitasi pemahaman dan perkembangan siswa dalam bidang tersebut (Djamaluddin & Wardana, 2019).

Mata pelajaran dapat mencakup berbagai disiplin ilmu, seperti matematika, bahasa Inggris, sains, sejarah, geografi, seni, musik, olahraga, bahasa asing, dan lain sebagainya. Setiap mata pelajaran biasanya memiliki kurikulum dan standar yang ditetapkan untuk memandu proses pembelajaran dan evaluasi prestasi siswa.

Tujuan dari pengajaran mata pelajaran adalah untuk memperluas pengetahuan siswa, meningkatkan pemahaman mereka terhadap konsep dan prinsip yang relevan dengan mata pelajaran tersebut, dan membangun keterampilan yang relevan. Mata pelajaran memberikan struktur dan arah dalam pendidikan formal, memungkinkan siswa untuk belajar dan mengembangkan kompetensi dalam bidang yang berbeda sesuai minat dan kemampuan mereka (Djamaluddin & Wardana, 2019).

Setiap mata pelajaran biasanya diajarkan oleh guru yang memiliki pemahaman mendalam tentang subjek tersebut dan menggunakan metode pengajaran yang sesuai untuk memfasilitasi pembelajaran efektif. Pengajaran mata pelajaran dapat melibatkan kegiatan seperti ceramah, diskusi, tugas, praktikum, penelitian, dan evaluasi untuk mengukur kemajuan dan pencapaian siswa dalam bidang tersebut.

2.3 Algoritma Genetika

Algoritma Genetika adalah metode optimasi yang efisien dan fleksibel, yang dapat diterapkan dalam berbagai jenis studi kasus. Dengan memanfaatkan prinsip-prinsip seleksi alam dan mekanisme evolusi. Metode ini sering dimanfaatkan untuk menemukan solusi terbaik dalam berbagai masalah yang kompleks. Dengan pendekatan sistematis, algoritma ini dapat mengeksplorasi berbagai kemungkinan dan mengevaluasi kinerja masing-masing solusi, sehingga memungkinkan identifikasi solusi optimal yang memenuhi kriteria tertentu. Algoritma genetika berfungsi dengan membentuk populasi yang terdiri dari individu-individu yang diwakili dalam bentuk

kromosom. Setiap individu dalam populasi dievaluasi untuk mengukur tingkat keoptimalannya (Elva, 2019).

Algoritma genetika telah banyak diterapkan dalam menyelesaikan berbagai masalah dan pemodelan di berbagai sektor, termasuk teknologi, bisnis, dan industri hiburan. Salah satu contoh dalam hal optimasi penjadwalan, algoritma ini berfungsi untuk menyusun jadwal yang lebih efisien dengan mempertimbangkan berbagai batasan yang ada, selain itu ada juga otomatisasi pemrograman, pembelajaran mesin (machine learning), pemodelan ekonomi, dan simulasi sistem kekebalan tubuh. Contoh penerapan algoritma genetika juga mencakup model ekologi serta interaksi antara evolusi dan pembelajaran. Dalam model ekologi, algoritma ini digunakan untuk mensimulasikan dinamika populasi, mengidentifikasi pola interaksi antar spesies, dan memprediksi dampak perubahan lingkungan terhadap ekosistem. Berdasarkan uraian yang telah disampaikan, sangat penting untuk mengembangkan sebuah aplikasi yang dapat memfasilitasi operator di sekolah dalam melakukan penjadwalan secara optimal. Tujuan dari aplikasi ini adalah untuk memaksimalkan keuntungan melalui proses seleksi jadwal yang lebih efisien. Dengan memanfaatkan teknologi yang telah ada, diharapkan akan tercapai perbaikan yang signifikan dalam efektivitas dan kualitas layanan yang diberikan oleh penyedia jasa (Pranatawijaya & Putra, 2019).

2.3.1 Istilah dalam Algoritma Genetika

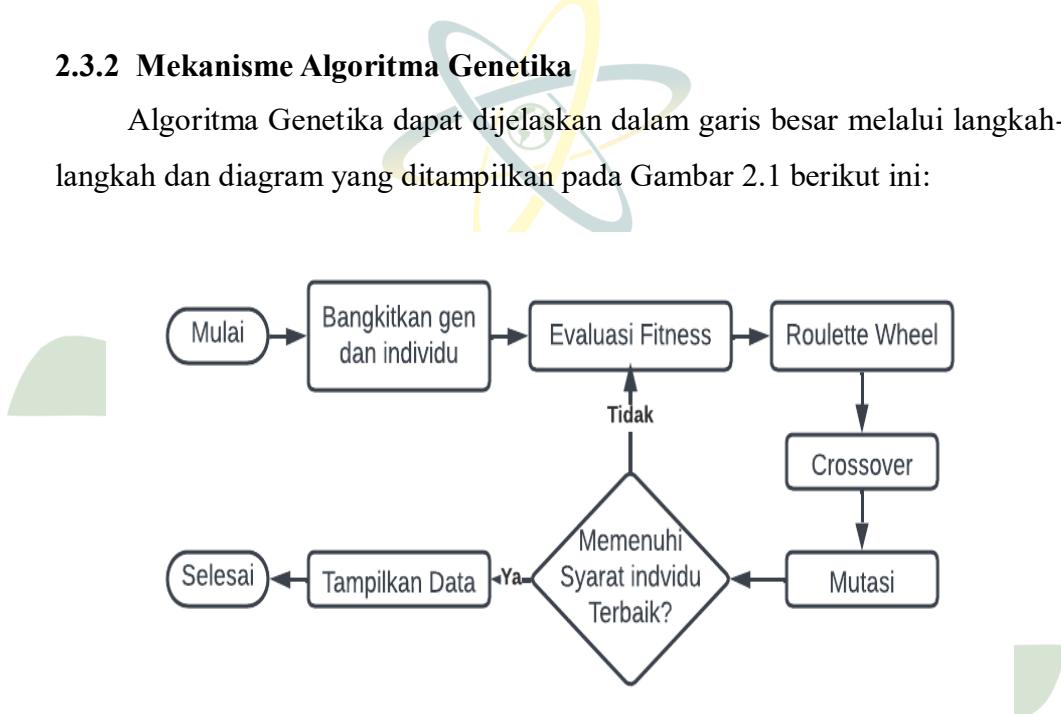
Beberapa istilah yang digunakan dalam Algoritma Genetika juga mengadopsi konsep-konsep serupa yang terinspirasi oleh prinsip-prinsip genetika dalam bidang biologi, seperti yang berikut ini:

1. Dalam Algoritma Genetika, populasi merujuk pada sekumpulan individu yang menjadi fokus utama dalam pencarian solusi.
2. Kromosom, yang juga dikenal sebagai Individu, merujuk pada representasi dari satu masalah atau solusi yang berperan sebagai komponen dalam proses literasi Algoritma Genetika.

3. Bagian dari kromosom yang disebut gen memiliki nilai tertentu dan berfungsi untuk membentuk keseluruhan individu.
4. Nilai *fitness* menggambarkan tingkat kebaikan atau kualitas suatu individu, yang digunakan untuk menilai sejauh mana individu tersebut cocok dengan tujuan yang ditentukan.
5. Generasi merujuk pada tingkatan perulangan dalam suatu populasi yang melibatkan reproduksi, persilangan, dan mutasi. Setiap generasi bertujuan untuk menghasilkan populasi yang lebih baik dan optimal.

2.3.2 Mekanisme Algoritma Genetika

Algoritma Genetika dapat dijelaskan dalam garis besar melalui langkah-langkah dan diagram yang ditampilkan pada Gambar 2.1 berikut ini:



Gambar 2. 1 Bagan perulangan algoritma genetika (Mataram, 2018)

Sebelum Algoritma Genetika diimplementasikan, sangat krusial untuk menetapkan fungsi fitness, yang berfungsi sebagai ukuran untuk menilai seberapa baik solusi yang dihasilkan dapat memenuhi kriteria optimasi dari masalah yang dihadapi. Semakin tinggi nilai fitness yang diperoleh, semakin baik kualitas sistem yang dihasilkan. Penentuan fungsi fitness ini dilakukan melalui metode heuristik. Algoritma Genetika merupakan pilihan yang ideal untuk menyelesaikan masalah optimasi yang rumit dan tidak dapat dipecahkan dengan metode tradisional. Menggambarkan proses evolusi alami, Algoritma

Genetika biasanya mencakup tiga operasi inti: reproduksi, persilangan (crossover), dan mutase. Serangkaian langkah-langkah berikut ini dapat digunakan untuk menjelaskan kerangka umum algoritma genetika:

- a. Mengumpulkan populasi awal melalui metode acak.
- b. Menghasilkan solusi baru melalui percobaan genetik berulang, dengan cara memilih, menggabungkan, dan memodifikasi sifat-sifat genetik.
- c. Melaksanakan pengembangan solusi dengan menganalisis setiap populasi melalui perhitungan nilai fitness dari masing-masing kromosom hingga memenuhi kriteria untuk berhenti.

Apabila kriteria penghentian belum terpenuhi, proses regenerasi akan dilakukan kembali untuk menghasilkan generasi yang baru. Berbagai kriteria penghentian yang sering diterapkan mencakup:

- a. Menghentikan proses setelah mencapai jumlah generasi yang ditentukan.
- b. Mengakhiri proses ketika nilai fitness tertinggi atau terendah (bergantung pada konteks masalah) tidak menunjukkan perubahan dalam beberapa generasi yang berurutan.
- c. Menghentikan proses jika tidak terdapat peningkatan pada nilai fitness yang lebih tinggi atau lebih rendah (sesuai konteks masalah) dalam sejumlah generasi berikutnya.

2.3.3 Fungsi Fitness

Fungsi fitness berperan dalam menilai kromosom untuk mendapatkan kromosom yang diharapkan. Fungsi ini berfungsi untuk mengidentifikasi perbedaan kualitas antara kromosom, sehingga dapat mengevaluasi seberapa efektif kromosom yang dihasilkan, yang dinyatakan melalui fungsi fitness berikut :

$$Fitness = \frac{1}{1+Penalty} \quad (1)$$

Dari persamaan tersebut, nilai fitness ditentukan oleh faktor penalti. Penalty tersebut mencerminkan jumlah pelanggaran terhadap kendala dalam

suatu kromosom; semakin kecil nilai *penalty* (jumlah pelanggaran), semakin tinggi nilai *fitness* yang diperoleh.

2.3.4 Seleksi Roulette

Seleksi roda *roulette* adalah pendekatan pemilihan yang intuitif dalam konteks algoritma genetik. Pada pendekatan ini, setiap individu dalam populasi di distribusikan pada roda *roulette* berdasarkan nilai *fitness* yang mereka peroleh. Proporsi segmen di roda roulette berhubungan langsung dengan nilai *fitness* dari setiap individu. Semakin besar nilai *fitness*, semakin luas segmen yang terwakili pada roda *roulette*. Setelah itu, roda *roulette* diputar untuk menentukan individu yang akan dipertahankan dalam proses seleksi (Juanita, 2018)

2.3.5 Crossover

Crossover merupakan proses percampuran antara dua kromosom yang menghasilkan keturunan kromosom baru. Mutasi adalah metode untuk memodifikasi sejumlah gen yang terdapat dalam kromosom. Elitisme merupakan strategi yang diterapkan untuk melestarikan individu-individu dengan nilai *fitness* tertinggi, sehingga mereka tidak lenyap selama proses evolusi. Penggantian populasi merujuk pada proses iteratif yang dilakukan untuk memperoleh populasi yang paling optimal dalam penerapan algoritma genetik, di mana individu-individu baru menggantikan yang lama berdasarkan kriteria seleksi tertentu (Supriana et al., 2021).

2.3.6 Mutasi

Dalam kehidupan nyata, mutasi bisa muncul sebagai hasil dari berbagai proses yang mempengaruhi genetik suatu organisme. Algoritma genetika juga memberlakukan hal yang sama (Hikmawan, 2021). Secara umum, proses mutasi dilaksanakan dengan cara menghasilkan bilangan acak yang berada di bawah nilai probabilitas mutasi yang telah ditentukan (*mutation rate*). Selanjutnya, nilai-nilai gen akan dimodifikasi menjadi kebalikan dari nilai

sebelumnya; contohnya, angka 0 akan diubah menjadi 1, sedangkan angka 1 akan berubah menjadi 0.

2.4 Penerapan Algoritma Genetika

Dalam proses perancangan sistem penjadwalan berbasis algoritma genetik, langkah awal yang harus dilakukan adalah menentukan populasi awal, yang terdiri dari berbagai solusi penjadwalan yang mungkin. Selanjutnya, nilai-nilai kromosom akan direpresentasikan, dan populasi awal akan dikonfigurasikan untuk memulai proses. emudian, rangkaian proses seleksi, crossover, dan mutasi diterapkan untuk menciptakan generasi baru yang lebih optimal.. Langkah-langkah ini terus diulang hingga mencapai kondisi berhenti yang ditentukan mencapai kondisi selesai.

Sebelum menetapkan populasi, tahap awal yang krusial adalah merancang bentuk kromosom yang akan diimplementasikan dalam sistem penjadwalan. Berikut adalah desain khas kromosom yang diterapkan dalam proses perancangan ini:

- a. Kode mata pelajaran (M)"
- b. Kode kelas (K)
- c. Kode jam pelajaran (T)"

Dengan demikian, urutan objek pada kromosom yang baru ini adalah: <M, G, K, T>. Panjang sebuah kromosom merupakan kombinasi gen yang didasarkan pada total jumlah seluruh mata pelajaran dan kelas yang tersedia pada semester yang sedang berjalan. Setiap gen dalam kromosom menyimpan informasi spesifik yang berkaitan dengan waktu dan jam untuk satu mata pelajaran serta kelas yang terkait. Sebagai ilustrasi untuk inisialisasi pembentukan kromosom, misalkan terdapat distribusi mata pelajaran yang tercantum dalam Tabel 2.1, yang menunjukkan mata pelajaran yang ditawarkan, dan sebaran jam pelajaran yang ditampilkan dalam Tabel 2.2, yang merinci jadwal waktu yang tersedia untuk setiap mata pelajaran dalam semester tersebut.

Tabel 2. 1 Sebaran Mata Pelajaran (Elva, 2019).

No	Id mata pelajaran	Nama mapel	Id kelas	Nama kelas
1	M01	Mengelola Induk Ikan	K01	X Agribisnis Perikaan
2	M02	Uji Coba Pakan Ikan	K01	X Agribisnis Perikaan
3	M03	Memijah Ikan	K01	X Agribisnis Perikaan
4	M04	Matematika	K01	X Agribisnis Perikaan

Tabel 2. 2 Sebaran Waktu (Elva, 2019).

Index waktu	Hari	Waktu
T01	Senin	07.30 – 08.05
T02	Selasa	07.30 – 08.05
T03	Rabu	08.10 – 08.50
T04	Kamis	11.05 – 11.45
T05	Jum'at	12.25 – 12.45
T06	Sabtu	13.20 – 13.55

Terdapat slot waktu kosong pada hari Senin pagi dari pukul 07.30 hingga 08.10 yang dialokasikan untuk pelaksanaan upacara pagi. Diasumsikan bahwa dalam suatu populasi yang terdiri dari tiga kromosom, setiap kromosom mewakili jumlah mata pelajaran yang tersedia. Setiap kromosom tersebut, pada gilirannya, memiliki tiga gen yang membentuknya. Untuk membentuk populasi awal, pemilihan dilakukan berdasarkan data dari tabel sebaran mata pelajaran dan tabel sebaran waktu. Proses pemilihan ini dilakukan secara acak, dengan setiap mata pelajaran dan waktu yang tersedia dipilih secara probabilistik. Dalam penyusunan populasi awal ini, setiap kromosom terdiri

dari gabungan kode mata pelajaran, kelas, dan waktu yang dipilih secara acak dari tabel sebaran. Misalnya:

- a. **M01** merupakan kode untuk *Mata Pelajaran 1*.
- b. **K01** adalah kode untuk *Kelas 1*.
- c. **T01** adalah kode untuk *Waktu 1*

K01M01T01 K01M04T04 K01M03T02

K01M02T03

K01M02T03 K01M02T01 K01M03T04

K01M04T01

K01M01T02 K01M03T03

K01M04T04 K01M01T02

Setiap gen adalah sebuah triplet yang berisi informasi tentang satu mata pelajaran yang dijadwalkan untuk kelas tertentu pada waktu tertentu.. setiap gen di dalam kromosom merupakan hasil dari pemilihan acak dari jumlah total mata pelajaran yang ditawarkan dan waktu yang tersedia untuk jadwal.

Fungsi fitness ini adalah untuk menentukan individu mana yang paling cocok atau "terbaik" dalam populasi, sehingga mereka dapat dipilih sebagai induk untuk menghasilkan generasi berikutnya.. Hasil dari fungsi fitness menunjukkan tingkat optimalisasi solusi yang dihasilkan. Kromosom yang memiliki nilai fitness tertinggi merepresentasikan solusi yang paling mendekati tujuan atau kriteria yang diinginkan, sehingga mereka dianggap sebagai solusi yang lebih optimal. semakin banyak pelanggaran yang terjadi, semakin tinggi nilai fitness (karena kita sedang meminimalkan pelanggaran), dan sebaliknya, semakin sedikit pelanggaran, semakin optimal solusi tersebut.

Beberapa batasan yang diutamakan dalam penyusunan jadwal ini :

- a. Satu kelas tidak diizinkan untuk dijadwalkan lebih dari satu sesi pada waktu yang sama.

K01G01M01T01 K01G02M04T04 K01G02M03T02

K01G02M02T03

K01G02M02T03 K01G02M02T01 K01G02M03T04

K01G01M04T01

K01G01M01T02 K01G02M03T03

K01G02M04T04 K01G01M01T02

Berdasarkan susunan populasi yang telah disebutkan sebelumnya, terlihat bahwa pada kromosom 1 tidak terdapat pelanggaran terhadap batasan yang ditetapkan, Selain itu, pada kromosom 3 dan 4 teridentifikasi adanya pelanggaran batasan. Pada kromosom 3, terdapat dua gen, yaitu gen 2 dan gen 4, yang menunjukkan kesamaan dalam hal kelas (K01) dan waktu (T01), pada kromosom ke 4 juga terdapat pelanggaran yaitu pada gen 1 dan 4 yang memiliki kesamaan kelas (K01), mata pelajaran (M01) dan waktu (T02) . Dari pelanggaran yang ada akan menghasilkan nilai *fitness* sebagai berikut :

Contoh Rumus :

$$\text{Fitness kromosom} = \frac{1}{1+(x^1+x^2+x^3)} \quad (2)$$

Diketahui : x^1 = Id Kelas

x^2 = Id Mata Pelajaran

x^3 = Index Waktu

Perhitungan :

$$\text{Fitness kromosom} 1 = \frac{1}{1+(0+0+0)} = 1$$

$$\text{Fitness Kromosom} 2 = \frac{1}{1+(1+0+0)} = 0,33$$

$$\text{Fitness Kromosom} 3 = \frac{1}{1+(1+1+1)} = 0,2$$

Pada tahap seleksi untuk pembentukan populasi baru, metode yang digunakan adalah metode seleksi *roulette-wheel*, masing-masing kromosom menempati potongan-potongan lingkaran secara proporsional sesuai dengan nilai *fitness* nya . Langkah awal dari metode ini adalah melakukan perhitungan total nilai fitness untuk semua kromosom, sebagaimana tercantum dalam Tabel 2.3 di bawah ini:

Tabel 2. 3 Tabel Nilai *Fitness* (Elva, 2019).

Kromosom	Nilai <i>Fitness</i>
1	1
2	0,33
3	0,2
Total nilai <i>fitness</i>	1,53

Langkah selanjutnya adalah menghitung probabilitas untuk setiap kromosom dengan cara membagi nilai *fitness* individu dengan total nilai *fitness* keseluruhan. Hasil dari proses ini disajikan dalam Tabel 2.4 berikut:

Tabel 2. 4 Tabel Probabilitas Nilai *Fitness* (Elva, 2019).

Kromosom	Probabilitas
1	$1 / 2,53 = 0,39$
2	$0,33 / 2,53 = 0,14$
3	$0,2 / 2,53 = 0,08$

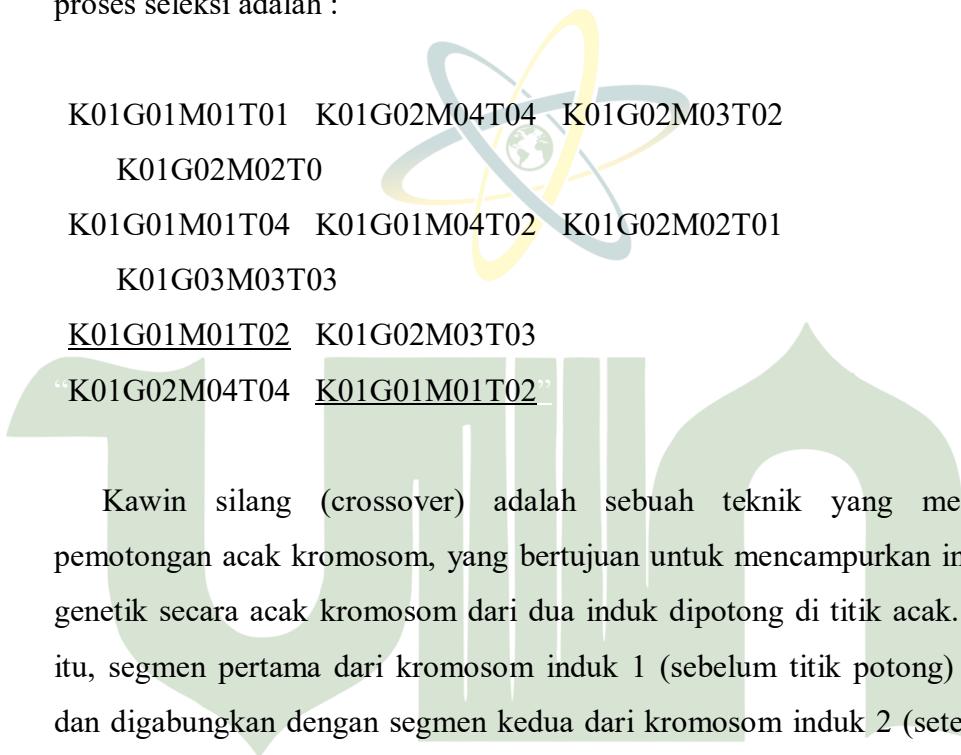
Ketiga adalah menempatkan masing-masing kromosom pada interval nilai [0-1]. Dapat dilihat pada tabel 2.5.”

“Tabel 2. 5 Tabel Interval Nilai Probabilitas (Elva, 2019).”

Kromosom	Interval Nilai
1	0 – 0,39
2	0,79 – 0,92
3	0,93 – 1

Untuk menentukan susunan populasi baru yang dihasilkan dari proses seleksi, dibuat bilangan acak dalam rentang [0 – 1], dimisalkan bilangan yang dibangkitkan adalah [0,2 ; 0,8 ; 0,5 ; 0, 95]. Dari bilangan acak yang dihasilkan, terlihat bahwa kromosom yang memiliki nilai 0,2 adalah kromosom 1, yang berada dalam interval nilai 0 hingga 0,39, Dengan demikian, kromosom 1 tidak terpilih dalam proses seleksi. Selanjutnya,

kromosom yang memiliki nilai 0.8 adalah kromosom ke-3, yang terletak dalam interval 0.79 hingga 0.92. Dengan demikian, kromosom ke-3 terpilih sebagai pengganti untuk kromosom ke-2, sedangkan kromosom keempat, yang memiliki nilai 0.95, berada dalam interval nilai 0.93 hingga 1. Oleh karena itu, kromosom ke-4 juga tidak terpilih dalam proses seleksi, karena bilangan acak yang dihasilkan berada dalam interval nilai yang ditetapkan untuk kromosom tersebut. Maka susunan kromosom populasi baru hasil proses seleksi adalah :



Kawin silang (crossover) adalah sebuah teknik yang melibatkan pemotongan acak kromosom, yang bertujuan untuk mencampurkan informasi genetik secara acak kromosom dari dua induk dipotong di titik acak. Setelah itu, segmen pertama dari kromosom induk 1 (sebelum titik potong) diambil dan digabungkan dengan segmen kedua dari kromosom induk 2 (setelah titik potong). Kawin silang (crossover) dilakukan dengan menggabungkan gen yang memiliki jenis serupa, di mana baris-baris gen tersebut dipilih secara acak. Kawin silang dilakukan ketika nilai bilangan acak yang dihasilkan oleh sebuah kromosom lebih kecil dari nilai probabilitas yang telah ditentukan sebelumnya, dimana nilai bilangan random dimisalkan adalah $[0, 2 ; 0,8 ; 0,5 ; 0, 95]$. Umumnya, nilai probabilitas ditetapkan minimal pada 0,5 (mendekati nilai 1). Metode kawin silang yang biasa diterapkan adalah teknik satu titik potong, di mana hanya satu titik pada kromosom yang dipilih secara acak untuk melakukan pemotongan. Sebuah titik potong ditentukan secara acak,

setelah itu segmen pertama dari kromosom induk 1 digabungkan dengan segmen kedua dari kromosom induk 2.

Bilangan acak yang dihasilkan untuk menentukan posisi titik potong berada dalam rentang $[1 - N]$, di mana N merupakan jumlah total gen dalam satu kromosom. Jadi, titik potong akan dipilih secara acak dari bilangan tersebut, dengan angka yang menunjukkan posisi gen di mana kromosom akan dibagi. Dari contoh nilai bilangan acak yang dihasilkan sebelumnya, kromosom 1 dan kromosom 3 mengalami proses crossover karena masing-masing memiliki nilai yang lebih rendah daripada probabilitas yang telah ditentukan dalam hal ini, kromosom 1 memiliki nilai 0,2 dan kromosom 3 bernilai 0,5, keduanya berada di bawah ambang probabilitas yang telah diatur sebelumnya. Jika posisi potong yang dipilih adalah gen ke-2, maka proses kawin silang dimulai dengan memotong kedua kromosom induk di posisi tersebut:

Kromosom 1 = K01G01M01T01 K01G02M04T04”

K01G02M03T02 K01G02M02T03”

Kromosom 3 = K01G01M01T04 K01G01M04T02

K01G02M02T01 K01G03M03T03”

Hasil kawin silang kedua kromosom tersebut seperti berikut :

Kromosom 1 = K01G01M01T01 K01G01M04T02

K01G02M02T01 K01G03M03T03

Kromosom 3 = K01G01M01T04 K01G02M04T04

K01G02M03T02 K01G02M02T03

Fitness kromosom 1 sesudah pindah silang

$$= \frac{1}{1+(1+0+0+1)} = 0,33$$

Fitness kromosom 3 sesudah pindah silang

$$= \frac{1}{1+(1+0+0+1)} = 0,33$$

Setelah proses kawin silang dilakukan, langkah selanjutnya adalah melaksanakan mutase, proses mutasi ini sangat penting untuk meningkatkan keragaman genetik dan memberikan peluang untuk menemukan solusi yang lebih optimal dalam proses evolusi. Hal ini menunjukkan bahwa proses perubahan dapat memberikan hasil yang bervariasi, menciptakan peluang untuk menemukan solusi yang lebih optimal dalam konteks tertentu. Jika ditemukan solusi yang memiliki nilai fitness yang lebih tinggi, maka hal tersebut merupakan hasil yang diinginkan. Jika ditemukan solusi dengan nilai fitness yang lebih rendah, ada kemungkinan bahwa pada iterasi berikutnya, solusi yang dihasilkan dari proses mutasi akan memiliki nilai fitness yang lebih baik dibandingkan dengan solusi induk.

Dengan melakukan modifikasi ini, diharapkan dapat menciptakan variasi baru yang dapat meningkatkan nilai fitness dari solusi yang dihasilkan. Dengan menggunakan pendekatan ini, kita bisa mengidentifikasi dan memperbaiki penempatan mata pelajaran yang kurang optimal, yang pada akhirnya akan meningkatkan kinerja keseluruhan dari sistem yang sedang dianalisis. Dengan mengetahui jumlah total gen, kita bisa melanjutkan ke proses pemilihan posisi gen yang spesifik untuk dilakukan mutasi.

Total gen = Jumlah gen dalam satu kromosom x Jumlah kromosom yang ada.

Berdasarkan contoh yang ada maka total gen adalah $4 \times 4 = 16$. Probabilitas mutasi ditetapkan 0,1 maka diharapkan mutasi yang terjadi adalah : $0,1 \times 16 = 1,6 = 2$. Maka ada 2 gen yang akan mengalami mutasi. Selanjutnya dilakukan iterasi sebanyak jumlah total gen [0-16] dan membangkitkan bilangan acak untuk tiap iterasi antara [0-1]. Diasumsikan gen yang mendapatkan bilangan di bawah probabilitas mutasi adalah gen 2 dan 3 pada kromosom 1. Informasi dalam gen yang akan diubah adalah waktu pelajaran, maka hasil mutasi pada kromosom tersebut adalah :

Kromosom sebelum mutasi = K01G01M01T01

K01G01M04T02 K01G02M02T01 K01G03M03T03

Kromosom setelah mutasi = K01G01M01T01

K01G02M04T02 K01G02M02T04 K01G03M03T03

Dari proses mutasi yang telah dilakukan, akan dihasilkan susunan kromosom baru yang dapat ditampilkan sebagai berikut:

K01G01M01T01 K01G02M04T02 K01G02M02T04

K01G03M03T03

K01G01M01T04 K01G02M04T04 K01G02M03T02

K01G02M02T03

K01G01M01T02 K01G02M03T03

K01G02M04T04 K01G01M01T02

Nilai *fitness* masing-masing kromosom dari hasil proses mutasi di atas adalah :

$$\text{Fitness kromosom 1 sesudah Mutasi} = \frac{1}{1+(0+0+0+0)} = 1$$

Fitness kromosom 2 sesudah Mutasi

$$= \frac{1}{1+(1+0+0+1)} = 0,33$$

Fitness kromosom 3 sesudah Mutasi

$$= \frac{1}{1+(1+1+1+1)} = 0,25$$

Dari hasil nilai *fitness* di atas dipilih kromosom 1 sebagai kromosom yang memiliki nilai *fitness* terbaik karena tidak terdapat pelanggaran yang ditetapkan dan merupakan solusi yang diinginkan. Hasil dari seluruh proses algoritma sudah sesuai dengan data asli yang diharapkan, bahwa guru yang sama tidak dijadwalkan mengajar lebih dari satu kali pada satu waktu yang bersamaan. Satu kelas tidak dijadwalkan lebih dari satu kali pada waktu yang bersamaan. Hasil akhir dari proses dapat dilihat pada tabel 2.6.

Kromosom 1 = K01G01M01T01 K01G02M04T02

K01G02M02T04 K01G02M03T03

Tabel 2. 6 Tabel hasil proses (Elva, 2019).

Id Mapel	Nama Mapel	Id Kelas	Kelas	Id Waktu	Waktu
M01	Mengelola Induk Ikan	K01	X Agribisnis Perikanan	T01	07.30 – 08.05
M04	Matematika	K01	X Agribisnis Perikanan	T02	07.30 – 08.05
M02	Uji Coba Pakan Alami	K01	X Agribisnis Perikanan	T04	08.40 – 09.15
M03	Memijah Ikan	K01	X Agribisnis Perikanan	T03	08.05 – 08.40

Kondisi selesai yang dapat menghentikan proses algoritma genetika ini adalah jika jumlah generasi atau iterasi telah maksimum.

2.5 Bahasa Pemrograman

Bahasa pemrograman adalah bentuk komunikasi antara manusia dan komputer melalui penulisan instruksi atau kode-kode yang akan dieksekusi oleh komputer. Fungsi bahasa pemrograman adalah menyediakan cara yang terstruktur dan standarisasi bagi manusia untuk berinteraksi dengan komputer. Dalam bahasa pemrograman, terdapat sintaks dan aturan yang harus diikuti agar komputer dapat memahami dan menjalankan perintah yang diberikan. Hal ini memastikan bahwa instruksi yang ditulis memiliki format yang selaras dengan tata bahasa yang disesuaikan (Hermiati et al., 2021).

Bahasa pemrograman adalah sarana untuk mengungkapkan ide dan konsep manusia yang dapat diinterpretasikan dan dieksekusi oleh mesin komputer, sehingga memiliki nilai dan kegunaan yang praktis. Setiap bahasa pemrograman memiliki aturan dan paradigma yang mengikatnya. Terdapat

berbagai macam paradigma bahasa pemrograman yang telah ditentukan Hermiati et al., 2021).

Penggunaan bahasa pemrograman tidak terbatas pada pembuatan program komputer saja. Bahasa pemrograman juga digunakan untuk mengembangkan aplikasi web, aplikasi mobile, perangkat lunak, dan berbagai sistem komputer lainnya. Dengan memanfaatkan bahasa pemrograman, manusia mampu merancang serta menerapkan solusi perangkat lunak yang disesuaikan dengan kebutuhan dan sasaran yang ingin dicapai. Beberapa konsep dasar dalam pemrograman meliputi paradigma prosedural, fungsional, deklaratif, berorientasi objek, dan konkuren. Konsep-konsep tersebut menjelaskan pendekatan dan cara kerja yang berbeda dalam mengembangkan program di bawah ini :

- a. Dalam kelas, simulasi sensitivitas terhadap masalah dan solusi potensial dilakukan baik secara manual maupun menggunakan komputer sebagai alat bantu. Sebagai contoh, aktivitas tersebut dapat berupa pengurutan tinggi badan siswa, dari yang tertinggi hingga terpendek, atau sebaliknya. Tujuannya adalah untuk mengembangkan pemahaman tentang masalah dan kemungkinan solusi.
- b. Pendekatan formal dalam menganalisis masalah dan merumuskan spesifikasi dilakukan melalui penulisan algoritma menggunakan notasi yang telah ditetapkan sebelumnya. Para siswa diwajibkan merumuskan solusi algoritma dengan menggunakan notasi standar yang telah ditetapkan selama proses pembelajaran di kelas, sehingga pemahaman tentang algoritma pemrograman dapat seragam dan terlepas dari aturan penulisan bahasa pemrograman.
- c. Tahap selanjutnya adalah menuliskan program dengan menerjemahkan notasi algoritma ke dalam sintaks bahasa pemrograman.
- d. Proses dan pengujian program dilakukan untuk memastikan program yang benar. Suatu program dianggap benar apabila tidak mengandung kesalahan logika maupun kesalahan sintaksis dalam bahasa pemrograman yang digunakan. Secara ideal, siswa diberi kesempatan

untuk menjalankan program sebanyak dua kali: yang pertama untuk memperbaiki kesalahan sintaksis, dan yang kedua untuk memastikan bahwa program berfungsi dengan benar. Asalkan analisis yang dilakukan benar, maka diharapkan tidak akan ada kesalahan logika yang terjadi..

- e. Penting untuk melakukan pengamatan terhadap eksekusi program guna meningkatkan pemahaman mengenai perubahan nilai dalam struktur data dan fenomena pelaksanaan. Hal ini menjadi krusial dalam pemrograman prosedural untuk memahami peristiwa yang terjadi selama proses.
- f. Membaca program penting untuk mengembangkan kemampuan menulis program yang baik. Siswa dapat berbagi teks algoritma dan saling membantu. Berlatih mandiri dan bekerja sama dalam kelompok adalah hal yang penting dalam membuat program.

2.5.1 ***PHP (Hypertext Preprocessor)***

PHP (Hypertext Preprocessor), yang dikenal sebagai *PHP*, adalah bahasa *scripting* yang digunakan khususnya untuk pengembangan *web*. *PHP* digunakan pada sisi server sehingga memerlukan *web server* untuk dijalankan. *PHP* juga dapat terintegrasi dengan *HTML*, *JavaScript*, *JQuery*, dan *Ajax*. Umumnya, *PHP* digunakan bersama dengan *file HTML*. Dengan *PHP*, dapat membuat website yang *powerful* dan dinamis, serta melakukan manajemen *database* (Mubarak, 2019)."

Rasmus Lerdorf adalah pencipta pertama *PHP*, yang awalnya digunakan untuk menghitung jumlah pengunjung pada halaman utamanya. Pada saat itu, *PHP* merupakan kumpulan skrip yang digunakan untuk memproses data formulir *web*. Kemudian, Rasmus melepaskan kode sumbernya dan memberi nama *PHP/FI*, yang merupakan singkatan dari *Personal Home Page/Form Interpreter*. Pelepasan kode sumber ini membuat *PHP* menjadi *open source*, sehingga banyak *programmer* tertarik untuk berkontribusi dalam pengembangan *PHP* (Mubarak, 2019).

2.5.2 MySQL

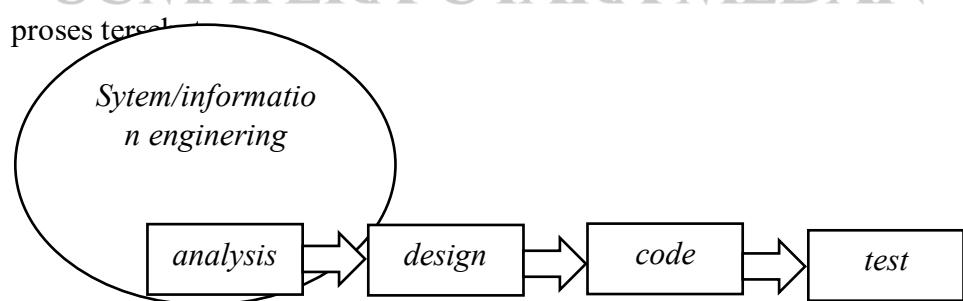
MySQL Server 2000 merupakan sistem manajemen basis data relasional (*RDBMS*) yang handal. *MySQL Server*, yang dirancang untuk mendukung transaksi dalam skala besar seperti pengolahan pesanan secara daring, manajemen inventaris, dan proses alur kerja atau manufaktur, secara otomatis menginstal enam basis data utama, yaitu *master*, *model*, *tempdb*, *pubs*, *northwind*, dan *msdb* (Christudas, 2019).

2.5.3 Web Server XAMPP

XAMPP merupakan sebuah perangkat lunak yang menyajikan lingkungan pengembangan web yang lengkap, meliputi server *web Apache*, *database MySQL*, dan *bahasa pemrograman PHP*. Dengan menggunakan *XAMPP*, para pengembang web dapat membuat, menguji, dan menjalankan aplikasi web secara lokal di komputer mereka sebelum akhirnya mempublikasikannya ke server web yang sesungguhnya. *Web Server* adalah lokasi tempat penyimpanan aplikasi web Anda, di mana Anda dapat mengaksesnya melalui internet (Fitri, 2021).

2.6 Perancangan Perangkat Lunak

Kerangka kerja yang dikenal sebagai proses perangkat lunak mencakup berbagai tugas yang esensial untuk menghasilkan perangkat lunak berkualitas tinggi. Terdapat sejumlah teori yang berhubungan dengan proses perancangan perangkat lunak, di antaranya adalah metode *Waterfall* atau yang dikenal sebagai *Linear Sequential Process*. Proses air jatuh merupakan pendekatan klasik dalam pengembangan perangkat lunak yang mengikuti urutan sistematis, termasuk proses analisis, desain, pengkodean, pengujian, dan dukungan. Model *Linear Sequential* menunjukkan tahapan-tahapan dalam proses tersebut.

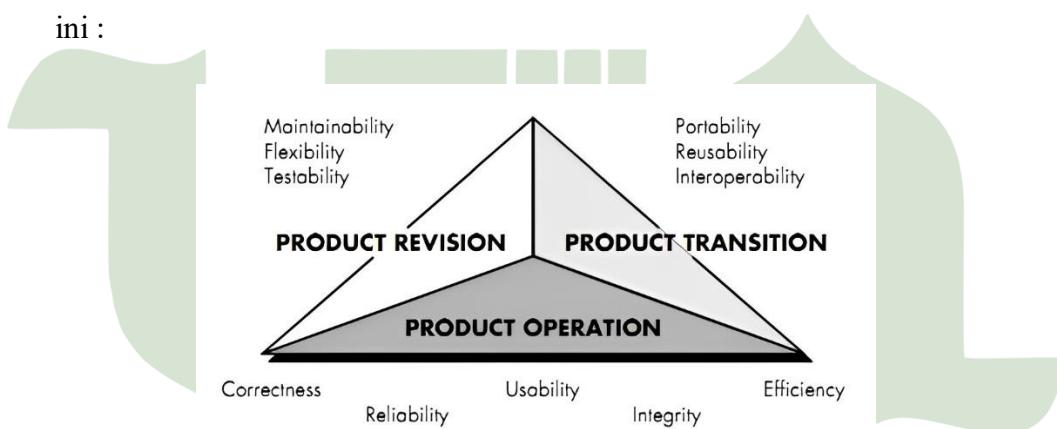


Gambar 2. 2 *Linear Sequential Software Model* (Hikmawan & Gata, 2021).

2.7 Kualitas Perangkat Lunak (Software Quality)

Software Quality, atau kualitas perangkat lunak, dihasilkan dari beragam faktor yang memengaruhi interaksi antara aplikasi dan pengguna yang membutuhkannya. Dalam ranah ini, mutu perangkat lunak diartikan sebagai representasi dari usaha manusia dalam menghormati dan mengapresiasi suatu perangkat lunak (Saputra et al., 2021).

Terdapat beberapa teori yang membahas tentang kualitas perangkat lunak, salah satunya adalah *McCall Quality Factor*. *McCall, Richards, dan Walter* mengembangkan serangkaian faktor yang menggambarkan kualitas perangkat lunak. Faktor-faktor kualitas tersebut dibagi menjadi tiga aspek penting dari sebuah perangkat lunak, yaitu karakteristik operasional, kemampuan untuk mengatasi perubahan, dan kemampuan beradaptasi dengan lingkungan baru (Saputra et al., 2021). Kategori tersebut digambarkan dalam gambar di bawah ini :



Gambar 2.3 Kualitas Perangkat Lunak (Saputra et al., 2021).

Faktor-faktor yang menunjukkan kualitas perangkat lunak tersebut antara lain :

- Ketepatan (*Correctness*): Menyajikan sejauh mana program dapat memberikan hasil yang tepat sesuai dengan kebutuhan dan spesifikasi yang ditetapkan.
- Keandalan (*Reliability*): Berkaitan dengan kemampuan program untuk beroperasi secara akurat dan konsisten, baik dalam situasi yang

membutuhkan presisi maupun dalam memenuhi tujuan yang diharapkan oleh pengguna.

- c. Kebergunaan (*Usability*): Menggambarkan tingkat kemudahan pengguna dalam mengoperasikan program, mempersiapkan input, dan memahami output yang dihasilkan.
- d. Integritas (*Integrity*): Terkait dengan tingkat kontrol yang diberikan kepada pengguna dalam mengakses program, baik melalui otorisasi maupun dalam menjaga keamanan program tersebut.
- e. Efisiensi (*Efficiency*): Menunjukkan sejauh mana program dapat menggunakan sumber daya komputer dengan efisien, serta meminimalkan jumlah kode yang diperlukan untuk menjalankan setiap fungsi program.
- f. Kemudahan pemeliharaan (*Maintainability*): Berkaitan dengan usaha yang diperlukan untuk menemukan dan mengatasi kesalahan yang terjadi dalam program.
- g. Fleksibilitas (*Flexibility*): Menyajikan tingkat kemudahan dalam mengubah program yang sedang berjalan sesuai dengan kebutuhan dan perubahan yang terjadi.
- h. Kemudahan pengujian (*Testability*): Terkait dengan upaya yang diperlukan untuk menguji program guna memastikan bahwa program berfungsi dengan baik sesuai yang diharapkan.
- i. Portabilitas (*Portability*): Menggambarkan tingkat kemudahan dalam mentransfer program dari satu lingkungan perangkat keras atau perangkat lunak ke lingkungan yang lain.
- j. Dapat digunakan kembali (*Reusability*): berhubungan dengan arah yang menjadi bagian program dapat digunakan kembali dalam program lain untuk menghemat waktu dan upaya pengembangan.
- k. Interoperabilitas (*Interoperability*): Terkait dengan upaya yang penting untuk menyambungkan antara sistem dengan sistem lain secara lancar dan kompatibel.

2.7.1 Kualitas Correctness

Sebuah program harus berfungsi dengan benar sesuai dengan parameter yang ditetapkan, menunjukkan sejauh mana perangkat lunak dapat melakukan fungsi yang diperlukan. Salah satu metrik umum yang digunakan untuk mengevaluasi faktor kebenaran atau *correctness* adalah jumlah error per *KLOC* (*kilo lines of code*), di mana kesalahan dalam program didefinisikan sebagai ketidaksesuaian antara kode dengan persyaratan perangkat lunak. Ketika menilai kualitas keseluruhan dari suatu produk perangkat lunak, kesalahan didefinisikan sebagai masalah yang dilaporkan oleh pengguna program setelah program tersebut dirilis untuk digunakan secara umum (Juliane et al., 2018).

Suatu program dianggap benar jika mampu memenuhi kebutuhan yang telah ditentukan dan mencapai tujuan atau misi pelanggan dengan tepat. Dalam hal ini, perangkat lunak dianggap benar jika mampu memastikan bahwa sistem berfungsi secara efektif dan dapat diandalkan dalam memenuhi kebutuhan pengguna serta menghasilkan hasil yang konsisten dan tepat (tanpa kekurangan atau kelebihan), dan dapat dibuktikan secara matematis secara formal (Juliane et al., 2018).

Beberapa metode pengukuran kebenaran per *KLOC* memiliki variasi, salah satunya adalah teori yang dikembangkan oleh McConnell yang dijelaskan dalam tabel berikut ini :

Tabel 2. 7 Perkiraan Jumlah Error KLOC Teori (Juliane et al., 2018).

Ukuran Project (Line of code /LOC)	Perkiraan Jumlah error
Lebih lecil dari 2k	0-25 error / KLOC
2K – 16K	0-40 error / KLOC
16K – 64K	0.5 – 50 error / KLOC
64K – 512K	2 – 70 error / KLOC
Lebih dari 512K	4 – 100 error / KLOC

2.7.2 Kualitas Functionally

Faktor kualitas *Functionally* adalah indicator yang mencakup kemampuan perangkat lunak untuk menjalankan tugas-tugas yang telah ditentukan serta kesesuaian dengan kebutuhan dan harapan pengguna, sehingga berkontribusi pada pengalaman pengguna yang positif a. Untuk menguji faktor kualitas fungsionalitas, dapat digunakan analisis fungsionalitas pada setiap komponen perangkat lunak. Salah satu metode yang dapat diterapkan untuk menguji fungsionalitas perangkat lunak adalah black-box testing, metode ini mengevaluasi sistem berdasarkan output yang dihasilkan dari input yang diberikan, tanpa mempertimbangkan bagaimana proses internalnya berjalan. Metode pengujian ini memberikan kesempatan bagi analis sistem untuk mendeteksi serangkaian kondisi input yang mencakup seluruh kebutuhan fungsional dari program. Pengujian black-box juga dikenal dengan istilah pengujian perilaku (Supriyono, 2019).

James Bach dalam tulisannya “*General Functionality and Stability Test Procedure for Certified for Microsoft Windows Logo Desktop Applications Edition*” membagi fungsi dalam perangkat lunak menjadi dua kategori, yaitu fungsi *primer (primary function)* dan fungsi pendukung (*contributing function*). Fungsi primer adalah fungsi utama yang sangat penting dalam perangkat lunak, dan kegagalan dalam fungsi ini akan membuat perangkat lunak menjadi tidak layak atau tidak dapat digunakan. Sementara itu, fungsi pendukung adalah fungsi yang memberikan kontribusi pada perangkat lunak, tetapi tidak termasuk dalam fungsi utama.

Dalam tulisannya yang berjudul “*General Functionality and Stability Test Procedure for Certified for Microsoft Windows Logo*,” James Bach memberikan gambaran tentang standar yang digunakan untuk menentukan apakah suatu perangkat lunak berhasil melewati pengujian faktor kualitas fungsionalitas (*functionality*) dalam program *Windows Logo*. Tulisannya memberikan panduan tentang bagaimana suatu perangkat lunak dapat memenuhi kriteria kualitas fungsionalitas yang ditetapkan dalam sertifikasi *Windows Logo*.

2.7.3 Kualitas Usability

Usability merupakan upaya untuk mengukur tingkat kebergunaan suatu program dan dapat dinilai berdasarkan empat karakteristik, yaitu: (1) tingkat keterampilan dan kecerdasan yang diperlukan untuk mempelajari sistem, (2) waktu yang diperlukan untuk menggunakan sistem secara efisien, (3) peningkatan produktivitas yang terjadi ketika seseorang menggunakan sistem dengan efisiensi yang memadai, dan (4) penilaian subyektif atau sikap pengguna terhadap sistem (dapat diperoleh melalui kuesioner). Faktor ini mengacu pada kemudahan penggunaan dan pembelajaran perangkat lunak. Usability melibatkan aspek akademis seperti psikologi, ergonomi, dan faktor-faktor manusia (Sembodo et al., 2021)

2.7.4 Kualitas Portability

Suatu perangkat lunak dapat dikategorikan sebagai portable jika biaya yang diperlukan untuk memindahkannya (baik dalam hal transportasi maupun adaptasi) ke lingkungan yang baru lebih rendah dibandingkan dengan biaya yang harus dikeluarkan untuk membangun perangkat lunak tersebut dari awal (Pamuji et al., 2019).

Faktor kualitas portability mencerminkan kemampuan suatu perangkat lunak untuk dapat dipindahkan dan dijalankan di lingkungan yang berbeda, baik itu terkait dengan perbedaan spesifikasi perangkat keras maupun sistem operasi.

2.8 Flowchart

UNIVERSITAS ISLAM NEGERI

SUMATERA UTARA MEDAN

Flowchart atau Diagram alir adalah suatu representasi visual yang disusun dalam bentuk grafik, yang menggambarkan berbagai langkah serta urutan prosedur dari suatu program. Umumnya, diagram alir digunakan untuk membantu dalam pemahaman dan evaluasi terhadap penyelesaian masalah tertentu yang memerlukan analisis yang lebih mendalam, sehingga memudahkan identifikasi langkah-langkah yang perlu diambil untuk mencapai solusi yang diinginkan. *Flowchart* dapat digunakan untuk menggambarkan aktivitas manual, proses pemrosesan data, atau kombinasi

keduanya. *Flowchart* terdiri dari serangkaian simbol-simbol yang digunakan untuk membangun representasi grafis dari proses atau alur kerja yang dijelaskan (Fajrianto et al., 2022).

Diagram alir untuk merepresentasikan **logika** dari program yang telah dikembangkan oleh seorang programmer, sehingga memudahkan pemahaman mengenai alur pemrosesan data dan keputusan yang diambil dalam program tersebut. Diagram alir adalah suatu representasi visual yang disajikan dalam bentuk diagram aliran, memanfaatkan simbol-simbol atau grafik yang memiliki fungsi dan makna tertentu. Terdapat sejumlah pedoman yang penting untuk diperhatikan saat menyusun diagram alir tersebut, agar hasilnya dapat memberikan klaritas dan efektivitas dalam menggambarkan proses yang dimaksud, yaitu sebagai berikut :

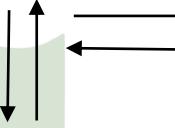
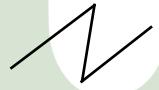
- a. **Diagram alir** disajikan dalam format vertikal yang mengalir dari bagian atas ke bagian bawah, serta dalam format horizontal yang bergerak dari sisi kiri ke sisi kanan
- b. Kegiatan yang diuraikan dalam diagram alir harus dijelaskan dengan ketelitian yang tinggi, dan penjelasannya harus disusun sedemikian rupa sehingga dapat dengan mudah dipahami oleh para pembaca..
- c. Titik awal dan titik akhir dari setiap kegiatan perlu diidentifikasi secara jelas, sehingga pemahaman mengenai pergeseran antara berbagai tahapan dalam aktivitas dapat terjaga dengan baik.
- d. Setiap tahap dalam suatu kegiatan harus diuraikan menggunakan deskripsi kata kerja yang sesuai dan spesifik, contohnya seperti "melaksanakan duplikasi," agar memberikan pemahaman yang lebih jelas mengenai tindakan yang harus diambil. ".
- e. Setiap langkah dalam aktivitas harus diurutkan secara benar sesuai urutan logisnya.
- f. Lingkup dan rentang aktivitas yang digambarkan dalam flowchart harus diperiksa dengan cermat. Percabangan yang memotong aktivitas yang sama dalam flowchart harus menggunakan simbol penghubung, atau jika

tidak terkait dengan sistem, cabang tersebut dapat dihilangkan atau ditempatkan pada halaman terpisah.

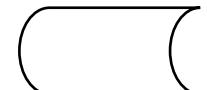
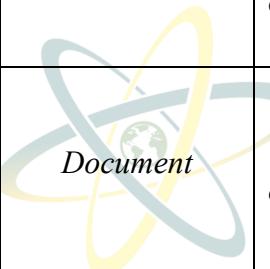
g. Gunakan simbol-simbol flowchart yang standar untuk menggambarkan aktivitas dan hubungannya dengan jelas.

Di bawah ini disajikan sejumlah simbol standar yang umum diterapkan dalam penyajian flowchart. Tabel 2.8 menyajikan simbol-simbol tersebut secara rinci:

Tabel 2. 8 Simbol-simbol *flowchart* (Fajrianto et al., 2022).

No	Symbol	Nama	Keterangan
1.		Arus / Flow	Untuk menyatakan jalannya arus suatu proses
2.		Comunication link	“Untuk menyatakan bahwa adanya transisi suatu data atau informasi dari suatu lokasi ke lokasi lainnya”
3.		Connector	“Untuk menyatakan sambungan dari satu proses ke proses lainnya dalam halaman / lembaran sama”
4.		Offline Connector	Untuk menyatakan sambungan dari satu proses lainnya dalam halaman atau lembaran yang berbeda

No	Symbol	Nama	Keterangan
5.		Proses	Sebuah fungsi pemrosesan yang dilaksanakan oleh komputer biasanya menghasilkan perubahan terhadap data atau informasi
6.		<i>Symbol manual</i>	Untuk menyatakan suatu tindakan (proses) yang tidak dilakukan oleh <i>computer</i> (manual)
7.		<i>Decision/logika</i>	Untuk menunjukkan suatu kondisi tertentu, dengan dua kemungkinan ya / tidak
8.		<i>Predefined Process</i>	Untuk menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal
9.		<i>Terminal</i>	Untuk menyatakan permulaan atau akhir suatu program
10.		<i>Offline storage</i>	Untuk menunjukkan bahwa data dalam <i>symbol</i> ini akan disimpan ke suatu media

11.		<i>Input / output</i>	Untuk menyatakan proses <i>input</i> dan <i>output</i> tanpa tergantung dengan jenis peralatannya
12.		<i>Dist storage</i>	Untuk menyatakan <i>input</i> berasal dari <i>disk</i> atau <i>output</i> disimpan ke <i>disk</i>
13.			Untuk menyatakan dokumen

2.9 Penelitian Terdahulu

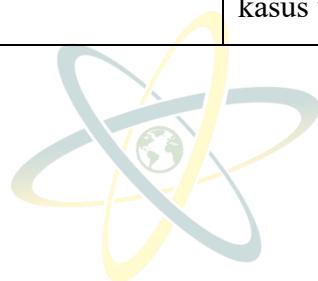
No	Judul	Ringkasan
	Penerbit dan tahun terbit	
1.	“Sistem Penjadwalan Sidang Tugas Akhir Menggunakan Algoritma Gentika (Aji et al., 2021). JTT (Jurnal Teknologi Terapan) Volume 7 Nomor 1 Maret 2021.”	“Algoritma genetika dapat menjadi solusi untuk masalah penjadwalan sidang tugas akhir dengan cara membangkitkan populasi atau kumpulan jadwal secara acak, kemudian jadwal tersebut akan saling membandingkan dan memperbaiki satu sama lain menggunakan operasi genetika. Kemudian dipilihlah hasil jadwal yang paling optimal, yaitu tidak memiliki pelanggaran batasan.
2.	Penjadwalan Mata Kuliah Menggunakan Algoritma Genetika	Aplikasi dapat berjalan dengan baik di web browser dengan

	di Jurusan Sistem Informasi ITS (Puspaningrum et al., 2018). JURNAL TEKNIK POMITS Vol. 7 No. 1 Maret 2018.	memberikan keluaran yang sesuai dan aplikasi dapat dijalankan dengan baik. Aplikasi dapat menghasilkan keluaran sesuai dengan batasan kaku yang harus dipenuhi oleh aplikasi. Aplikasi dapat berjalan meskipun data yang ada pada database pendukung aplikasi diganti dengan jumlah data yang berbeda namun harus dengan jenis data dan format tabel yang sama.
3.	Sistem Pakar Perencanaan Penjadwalan Bangunan Rumah Minimalis Berdasarkan Tipe dengan Metode Algoritma Genetika (Harijanto et al, 2021) JIP (Jurnal Informatika Polinema) Vol. 7 No. 2 Februari 2021.”	Sistem mampu memberikan informasi berupa gambaran kebutuhan pembangunan rumah sesuai dengan persiapan dana yang dimiliki dan tipe yang dinginkan. Karakteristik algoritma genetika dapat dilihat dari pengujian parameter algoritma genetika dimana jika jumlah kromosom yang dibangkitkan dengan ukuran besar maka akan cenderung mengalami penurunan nilai <i>fitness</i> , lalu pada jumlah generasi hampir sama seperti jumlah kromosom jika jumlah generasi besar maka nilai <i>fitness</i> juga akan mengalami penurunan nilai <i>fitness</i> dan pengkombinasian parameter probabilitas <i>crossover</i> . Hasil

		<p>pengujian terhadap probabilitas mutasi menunjukkan bahwa nilai fitness meningkat seiring dengan bertambahnya probabilitas crossover dan menurunnya probabilitas mutasi. Di sisi lain, kombinasi probabilitas crossover yang rendah dengan probabilitas mutasi yang tinggi dapat mengurangi efektivitas algoritma genetik dalam melakukan eksplorasi..</p>
4.	<p>“Aplikasi Algoritma Genetika Dalam Penjadwalan Mata Kuliah. (Mone & Simarmata, 2021). Jurnal Ilmu Matematika dan Terapan Vol. 15 No. 4 December 2021.”</p>	<p>Hasil running aplikasi diperoleh rata-rata waktu eksekusi 30 jadwal adalah 25.86 menit, standar deviasi 11,88 menit dengan jumlah ruang kuliah sebanyak 3 ruang kuliah, dan 1 ruang aula untuk mata kuliah umum, 51 pengampu mata kuliah, 18 dosen, 5 hari kerja dan 14 jam efektif per hari. Selain dari hasil tersebut, diperoleh bahwa dari 30 jadwal yang diproses secara berturut-turut memberikan kesimpulan yakni, (1) tidak terjadi bentrok dosen, (2) tidak terjadi bentrok ruang dan waktu, (3) tidak terjadi bentrok dengan waktu dosen yang berhalangan, (4) tidak terjadi bentrok dengan waktu sholat jumat,</p>

		(5) ruang Aula FIP digunakan untuk mata kuliah yang sudah ditetapkan sebelum proses jadwal. Berdasarkan hasil pengujian sistem disimpulkan bahwa aplikasi penjadwalan yang dibuat efektif dan efisien.
5.	“Algoritma Genetika dengan Mutasi Terbatas untuk Penjadwalan Perkuliahan (Hikmawan & Gata, 2021). Jurnal Kajian Ilmiah Vol.21 No.2 Mei 2021.”	Berdasarkan hasil penelitian yang telah dilakukan, dapat diambil kesimpulan bahwa Algoritma Genetika dengan mutasi individu terbatas dapat membantu dalam pembuatan jadwal perkuliahan dengan hard constraint dan soft constraint secara proporsional. Pentingnya pembatasan terhadap mutasi tidak bisa diabaikan, terutama dalam menghadapi tantangan constraint dalam penjadwalan perkuliahan, yang terikat oleh pilihan ruang, waktu, dan hari yang terbatas.. Pengujian yang dilakukan dalam percobaan ini berfokus pada data kurikulum untuk Semester Ganjil Tahun Akademik 2020/2021 dengan tujuan untuk mengevaluasi dan menganalisis efektivitas serta kesesuaian materi yang diajarkan menggunakan Algoritma Genetika dengan batasan pada proses mutasi, penelitian ini berhasil mencapai beban minimum

		sebesar 0 setelah melakukan 10 iterasi dengan total populasi sebanyak 500 individu. Namun, metode ini memiliki kelemahan, yaitu memerlukan pengklasifikasian terhadap kelas-kelas yang ada, yang mungkin tidak ideal untuk beberapa kasus yang diinginkan.
--	--	--



UNIVERSITAS ISLAM NEGERI
SUMATERA UTARA MEDAN