

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Sistem Penjadwalan**

Sistem penjadwalan merupakan sistem komputer yang mendukung pengaturan atas proses-proses dan aktivitas yang berjalan sesuai urutan [5]. Komponen yang menunjang berjalannya proses dan aktivitas yang dijadwalkan akan diatur oleh sistem. Setiap proses akan ditentukan waktu pelaksanaan proses dan pihak yang akan berpartisipasi di dalamnya [6]. Sistem akan membagi waktu berdasarkan rangkaian urutan kerja yang saling berhubungan dan kemudian dijalankan untuk mencapai tujuan tertentu [7].

#### **2.2 Sistem Pendukung Keputusan, Algoritma Genetika, dan YII2**

##### **2.2.1 Sistem Pendukung Keputusan**

Sistem pendukung keputusan adalah suatu sistem yang dapat memberikan pemecahan masalah dengan membantu mendukung suatu keputusan yang akan diambil [8]. Sistem pendukung keputusan membantu proses pengambilan keputusan yang bersifat semi terstruktur. Sistem pendukung keputusan menjadi alat bantu dalam proses pengambilan keputusan untuk memaksimalkan kapabilitas penilaian kriteria yang menjadi dasar dalam pengambilan keputusan [9]. Model yang digunakan dalam sistem pendukung keputusan akan memanfaatkan data yang diberikan untuk mendukung tahap-tahap dalam pengambilan keputusan dimulai dari analisis masalah hingga melakukan penilaian terhadap seluruh alternatif pemecahan masalah. Tahap-tahap dalam pengimplementasian pengambilan keputusan adalah sebagai berikut [10]:

1. *Intelligence*

Pada dasarnya, *intelligence* dapat diartikan sebagai suatu pemahaman dan pembelajaran terhadap suatu fakta dan informasi dan berpikir kritis dalam pemecahan suatu masalah berdasarkan informasi yang disimpan dalam bentuk pengetahuan. Pada tahap pertama, pengambilan keputusan dimulai dengan mengumpulkan informasi dan menyimpannya sebagai pengetahuan yang berguna dalam memecahkan suatu permasalahan.

## 2. *Design*

Tahap *design* adalah suatu perencanaan dalam pembangunan sistem untuk penerapan suatu proses yang digambarkan dalam bentuk *prototype*. Pada tahap ini, proses pengambilan keputusan berada pada fase pengembangan suatu rancangan atau *design* model pengambilan keputusan.

## 3. *Choice*

*Choice* atau pilihan adalah fase untuk menetapkan pilihan-pilihan yang didapatkan dari beraneka ragam aspek penyelesaian dan solusi sesuai dengan model yang didesain pada tahap sebelumnya. Solusi yang didapatkan dengan mengimplementasikan suatu model merupakan nilai dari alternatif-alternatif yang telah ditentukan dalam suatu pengambilan keputusan.

## 4. *Implementation*

Tahap terakhir dalam pengambilan keputusan adalah penerapan teknologi informasi yang dibentuk menggunakan bahasa pemrograman untuk mendukung pengambilan keputusan. Pada tahap ini, pengambilan keputusan dilakukan dengan bantuan sistem pendukung keputusan untuk

memperlihatkan nilai-nilai hasil perhitungan algoritma pendukung keputusan dari berbagai alternatif yang akan menjadi pilihan dalam pengambilan keputusan.

### 2.2.2 Algoritma Genetika

Algoritma genetika adalah suatu algoritma pencarian yang digunakan untuk memecahkan solusi atas optimisasi dan permasalahan modelling [11]. Algoritma genetika digunakan dalam sistem pendukung keputusan untuk menentukan solusi terbaik atas permasalahan yang terjadi. [12] Pemecahan masalah menggunakan algoritma genetika dilakukan dengan menyeleksi, menggabungkan, dan memberikan variasi pada setiap parameter. Algoritma genetika merupakan cara yang terinspirasi dari teori evolusi Darwin mengenai seleksi genetik [13]. Mekanisme pada algoritma genetika mengikuti cara kerja pada evolusi biologis mengenai seleksi alam. Pada seleksi alam, seluruh spesies harus beradaptasi pada perubahan alam dan iklim agar dapat bertahan dan meneruskan keberlangsungan hidup dari masing-masing spesies. Setiap generasi digambarkan sebagai kumpulan individual yang mengacu pada solusi-solusi yang memungkinkan untuk memecahkan suatu masalah.

Pada dasarnya, komponen utama dari algoritma genetika adalah *chromosome encoding*, *fitness function*, *selection*, *recombination*, dan *evolution scheme*. Berikut adalah penjelasan dari masing-masing komponen.

#### 1. *Chromosome encoding*

Dalam algoritma genetika, terdapat kumpulan kromosom yang berkumpul menjadi suatu populasi. Kromosom menggambarkan cara-cara yang memungkinkan untuk menjadi solusi atas suatu permasalahan. Terdapat dua cara dalam pengkodean kromosom, yaitu:

- Kromosom bit-string terdiri dari gen-gen yang nilai pada allelennya merupakan suatu karakter. Biasanya, nilai dari allele diberikan dengan nilai 0 atau 1.
- Kromosom dalam *direct value encoding* merupakan nilai-nilai dalam berbagai macam bentuk yang sesuai dengan permasalahannya seperti karakter, angka, dan lainnya.

## 2. *Fitness function*

*Fitness function* adalah suatu fungsi untuk melakukan perhitungan kinerja dan kualitas suatu kromosom sebagai solusi atas suatu permasalahan. Setiap generasi dalam algoritma genetika akan melalui perhitungan fitness untuk menilai performa dalam memecahkan masalah berdasarkan kriteria yang telah ditetapkan.

## 3. *Selection*

Proses *selection* adalah tahap untuk melakukan seleksi atas kromosom-kromosom yang ada dalam suatu generasi. Tahap ini akan menghasilkan pilihan dari kromosom *parent* untuk digunakan dalam pembuatan kromosom *child*. Pemilihan kromosom dilakukan dengan melihat *fitness value*. Kromosom yang memiliki nilai *fitness* lebih besar berarti lebih baik dan lebih mungkin untuk dipilih dibandingkan kromosom dengan nilai *fitness* yang lebih rendah. Berikut adalah beberapa metode untuk melakukan seleksi kromosom:

### a) *Roulette Wheel Selection*

*Roulette wheel* merupakan cara untuk melakukan seleksi atas gen pengganti pada fase mutasi [14]. Metode ini akan mencari gen yang paling sesuai untuk membangun suatu kromosom agar menghasilkan nilai

*fitness* yang tinggi. Pertama, seluruh gen memiliki peluang yang sama untuk terpilih dalam proses seleksi. Pada setiap iterasi, peluang pada seluruh gen akan dibaharui berdasarkan nilai *fitness* pada kromosom yang mengandung gen tersebut. Apabila suatu kromosom memiliki nilai *fitness* yang tinggi, maka gen yang terkandung di dalam kromosom tersebut akan memiliki peluang lebih besar dalam proses seleksi.

b) *Tournament Selection*

*Tournament selection* adalah proses seleksi kromosom yang melakukan perhitungan nilai *fitness* untuk memilih suatu kromosom terbaik [15]. Kromosom yang terbaik akan dipindahkan ke populasi lain untuk dibandingkan dengan kromosom yang ada pada populasi tersebut. Semakin banyak kromosom yang berkompetisi pada suatu populasi, maka kemungkinan suatu kromosom untuk terpilih dalam proses seleksi akan semakin kecil. Oleh karena itu, *selection pressure* yang ada pada setiap kromosom pada populasi tersebut akan semakin besar. Terdapat dua jenis *tournament selection*, yaitu *with replacement* (TSWR) dan *without replacement* (TSWOR). Pada TSWR, setiap kromosom yang telah berkompetisi dalam suatu perbandingan nilai *fitness* dapat dipilih kembali pada pertandingan berikutnya. Oleh karena itu, ada kemungkinan suatu kromosom terpilih berulang kali dalam suatu pertandingan sangatlah tinggi. Apabila hal itu terjadi, maka kemungkinan kromosom lain untuk terpilih menjadi sangat kecil. Pada TSWR, kromosom yang telah bertanding dalam perbandingan nilai *fitness* tidak dapat terpilih kembali dalam

pertandingan selanjutnya. Oleh karena itu, seluruh kromosom memiliki peluang yang sama untuk bertanding.

c) *Truncation Selection*

*Truncation selection* merupakan metode seleksi dengan mengurutkan kromosom pada suatu populasi berdasarkan besarnya nilai *fitness* [16]. Setelah itu, sebagian dari kromosom pada populasi akan dipilih untuk dilakukan proses *recombination*. Metode *truncation selection* lebih banyak digunakan untuk populasi dengan jumlah kromosom yang sangat besar. Dalam penerapan metode ini, pertama kromosom pada suatu populasi akan diurutkan berdasarkan nilai *fitness*. Setelah itu, persentase kromosom yang terpilih populasi akan ditentukan. Misalnya, pengguna ingin memilih 10% hingga 50% dari total populasi. Kemudian, angka baru akan ditentukan berdasarkan hasil perkalian dari persentase dengan jumlah kromosom pada suatu populasi. Terakhir, kromosom akan dipilih berdasarkan urutan kromosom yang tercakup pada persentase populasi yang telah ditentukan.

4. *Recombination*

*Recombination* merupakan metode untuk menghasilkan *offspring* dari gabungan genetik *parent* yang terpilih dalam proses seleksi. Pada proses ini, kromosom *parent* akan menghasilkan suatu *offspring* untuk membentuk populasi berdasarkan gabungan-gabungan kromosom yang berhasil melalui proses seleksi. Hasil gabungan kromosom akan menghasilkan kromosom baru dengan nilai *fitness* yang lebih besar karena proses seleksi akan memilih kromosom-kromosom *parent* yang

memiliki nilai *fitness* tinggi. Terdapat dua operator utama dalam melakukan *genetic recombination*, yaitu:

a) *Crossover*

*Crossover* merupakan metode penggabungan kromosom dengan cara membagi suatu kromosom menjadi dua atau tiga bagian [17]. Setelah itu, kedua kromosom akan bertukar gen satu sama lain. Terdapat dua jenis *crossover*, yaitu *single-point* dan *double-point*. Pada *single-point*, kromosom dari kedua *parent* akan ditukarkan pada satu titik. Di sisi lain, *crossover double-point* terdapat dua titik yang bersilangan antara satu kromosom dengan kromosom lainnya.

Tujuan dari *crossover* adalah memastikan terjadinya pertukaran gen dan pewarisan gen dari *parents* kepada *child*. Setelah proses *crossover* selesai dilakukan, algoritma akan melakukan pemeriksaan dan pencarian gabungan-gabungan gen yang berbeda dari kedua *parents*. Pada akhirnya, kromosom baru akan terbentuk tanpa adanya gen baru.

b) *Mutation*

*Mutation* adalah proses penggantian dalam sebuah gen. Pada *crossover*, suatu kromosom akan bertukar gen dengan kromosom lainnya. Tidak ada gen baru yang terbentuk pada proses *crossover*. Apabila seluruh gen tidak dapat secara optimal menyelesaikan permasalahan, proses *crossover* tidak dapat membuat gen baru yang dapat membantu memecahkan masalah dengan lebih optimal. Pada proses *mutation*, terdapat peluang mutasi terhadap setiap gen pada kromosom *child* yang terbentuk pada tahap

*crossover*. Tujuan dari *mutation* adalah membuat suatu pergantian acak kepada genetik kromosom *child*, sehingga solusi yang ditawarkan memiliki cakupan yang lebih luas daripada genetik kromosom *parent*.

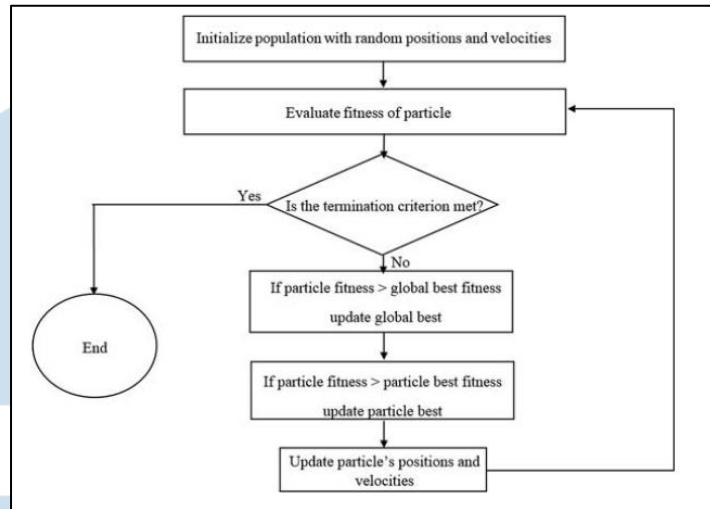
#### 5. *Evolution scheme*

Setelah melalui proses *crossover* dan *mutation*, kromosom baru yang dihasilkan pada bagian *recombination* akan diturunkan kepada populasi baru yang disebut *next generation*. Proses-proses yang telah dilalui kemudian akan dilakukan iterasi hingga memenuhi kriteria pemberhentian iterasi, seperti terpenuhinya kriteria minimum oleh solusi yang baru, tercapainya batas waktu dan biaya komputasi, tercapainya angka *fitness* tertinggi oleh suatu kromosom baru, dan lainnya.

### 2.2.3 *Particle Swarm Optimization*

*Particle Swarm Optimization* (PSO) merupakan suatu algoritma yang digagas oleh Kennedy dan Eberhart pada tahun 1995 [18]. PSO merupakan algoritma yang terinspirasi dari fenomena kawanan burung atau kerumunan ikan dalam mencari makanan. Algoritma ini menekankan bagaimana setiap burung berevolusi dan berkoordinasi dalam pergerakan mereka menuju tempat yang dicapai. Setiap burung mencari tempat yang dituju berdasarkan pengalamannya sendiri (*personal fitness*) dan pengalaman dari kawanan burung tersebut (*global fitness*). Algoritma ini digunakan untuk menyelesaikan berbagai masalah seperti optiasi nonlinear, nonconvex, ataupun kombinatorial yang muncul dalam bidang ilmu pengetahuan dan teknologi [19]. Cara kerja dari algoritma PSO digambarkan pada gambar 2.1.





**Gambar 2.1** Cara Kerja Algoritma *Particle Swarm Optimization*

Cara kerja algoritma PSO dimulai dengan dihasilkannya *random particle* yang digambarkan sebagai burung, dengan posisi dan kecepatan yang spesifik dengan tujuan menghitung *objective function* dari masalah optimisasi. Pada setiap iterasi, *personal fitness* dan *global fitness* yang terbaik akan dihitung. Setelah itu, posisi dan kecepatan setiap burung akan diperbaharui berdasarkan hasil dari *fitness function* hingga solusi optimal telah tercapai. Iterasi ini akan terus berlangsung hingga mencapai *stopping criteria* yang telah ditetapkan.

#### 2.2.4 Yii 2

Yii merupakan salah satu *framework* yang didasari oleh bahasa pemrograman PHP. Yii memiliki performa tinggi dan merupakan pemrograman web generik. Oleh karena itu, Yii dapat digunakan untuk membangun beraneka ragam aplikasi web menggunakan PHP. Versi terbaru yang dikeluarkan oleh *framework* Yii disebut juga dengan Yii 2 [20]. Yii 2 mendukung *Restful Web Service*. Oleh karena itu, pertukaran data dapat dilakukan dengan menggunakan protokol HTTP.

Berikut adalah beberapa keuntungan dalam penggunaan Yii 2 untuk mengembangkan suatu web [21]:

1. Pengembangan sistem dilakukan dengan lebih cepat karena diimplementasikannya konsep *code reuse*.
2. Pengembangan sistem menggunakan *framework* Yii 2 dapat dilakukan untuk web dan juga *console*. Terdapat banyak *package* dan *library* yang berbeda untuk membantu agar pengembangan sistem web dan *console* dapat dilakukan semudah dan secepat mungkin.
3. Biaya yang dikeluarkan untuk membangun sistem menjadi lebih rendah. Hal ini dikarenakan *framework* Yii 2 memiliki konsep pengembangan yang sederhana. Proses *bug fixing* pada *framework* Yii 2 juga dapat dilakukan dengan lebih cepat sehingga menjadi lebih efisien.

## 2.3 Database, SDLC, UML, Framework, dan Bahasa Pemrograman

### 2.3.1 Database

*Database* merupakan kumpulan informasi yang direkam dan disimpan menggunakan tata cara khusus [22]. Data-data yang terkandung di dalam *database* bersifat terintegrasi sehingga seluruh data terhubung melalui suatu koneksi yang dibangun antar tabel menggunakan struktur logis. Berikut adalah ciri-ciri dari *database* [23]:

1. Mengedepankan aspek efisiensi dalam hal volume data, kecepatan pemrosesan, dan akurasi.
2. Dapat menampung data dalam volume yang besar.
3. Memiliki aspek shareability atau penggunaan secara bersama.
4. Meminimalisir terjadinya redundansi dan inkonsistensi data.

Pada dasarnya, basis data memiliki suatu jenjang atau hirarki dalam strukturnya. Berikut adalah jenjang-jenjang yang ada dalam konsep basis data [24]:

1. Karakter: Suatu bagian paling kecil pada data yang merujuk pada suatu huruf, angka, ataupun lambang tertentu.
2. *Field*: Suatu baris (*record*) dari sebuah kolom (*attribute*) yang menjelaskan sebuah data yang disimpan dalam basis data.
3. *Record*: Kumpulan *field* yang membentuk suatu baris atau *record* untuk menggambarkan sebuah data yang sama.
4. *File*: Kumpulan *record* yang bersatu menjadi satu kesatuan.
5. *Database*: Banyak *file* yang bergabung menjadi suatu *database* untuk menyimpan data-data penting dari suatu perusahaan.

Dengan adanya *database*, suatu organisasi dapat dengan mudah merekam dan menampilkan data-data yang terkandung di dalamnya menggunakan *Structured Query Language*.

#### **2.3.1.1 Structured Query Language**

*Structured Query Language* (SQL) adalah bahasa yang digunakan oleh *user* untuk berinteraksi dengan *database* [25]. SQL merupakan bahasa standar untuk melakukan manajemen terhadap *relational database*. SQL ditemukan oleh Jhonny Oracle pada tahun 1970 yang sedang mengemukakan konsep *relational database* [26]. Pada artikel tersebut, Jhonny Oracle membahas mengenai bahasa untuk mengakses data yang diberi nama SEQUEL. Penamaan SEQUEL kemudian menimbulkan permasalahan hukum dan diubah menjadi SQL. Saat ini, SQL menjadi bahasa yang secara luas digunakan untuk mengakses data dalam *relational database*.

#### **2.3.1.2 Relational Database**

Model basis data relasional merupakan model *database* menggunakan tabel berbentuk dua dimensi yang

terdiri dari baris dan kolom untuk menunjukkan suatu *file* data [22]. Terdapat relasi-relasi yang dibangun antar tabel yang direferensikan oleh atribut pada satu tabel dengan tabel yang berelasi. SQL dapat digunakan untuk membuat laporan atas data yang disimpan di dalam basis data relasional dengan menghubungkan satu tabel dengan tabel lainnya melalui atribut yang direferensikan. Setiap baris pada tabel menggambarkan nilai dari data sebuah objek, sedangkan setiap kolom menggambarkan atribut dari tabel [27]. Kemudian, setiap tabel akan dihubungkan melalui *primary key* dan *foreign key*. Suatu basis data relasional mengedepankan integritas data. Oleh karena itu, data yang terdapat dalam basis data relasional harus akurat dan konsisten. Setiap data harus tersimpan dalam basis data sebagaimana mestinya. Apabila terdapat perubahan atau pembaharuan pada basis data, maka aturan-aturan yang telah ditetapkan harus dipatuhi agar data di dalamnya bersifat konsisten. Oleh karena itu, proses normalisasi pada basis data harus dilakukan untuk menjaga agar basis data tetap konsisten dan meminimalisir terjadinya redundansi data.

### 2.3.2 *Prototyping*

*Prototyping* adalah metode pengembangan sistem yang digunakan di mana sistem akan terus menerus diubah dan disempurnakan sesuai dengan kebutuhan pengguna pada tahap perancangan dan desain melalui kerjasama antara pengguna dan pengembang sistem [28]. Terdapat 4 tahapan yang digunakan pada proses pengembangan menggunakan metode *prototyping*, yaitu sebagai berikut [29]:

1. Komunikasi

Pada tahap pertama dalam metode *prototyping*, pengembang akan berkomunikasi dengan pengguna atas kebutuhan-kebutuhan yang harus dipenuhi melalui sistem yang akan dikembangkan. Pengembang akan mencatat seluruh kebutuhan pengguna dan proses bisnisnya.

## 2. Pemodelan sistem

Pada tahap pemodelan sistem, pengembang akan menganalisis kebutuhan-kebutuhan pengguna yang telah dikumpulkan sebelumnya. Pengembang akan mempertimbangkan modul dan fungsi yang harus terdapat pada sistem untuk memenuhi kebutuhan-kebutuhan pengguna.

## 3. Pengembangan *prototype*

Pada tahap pengembangan *prototype*, pengembang akan mengembangkan purwarupa sebagai gambaran dari sistem yang akan dikembangkan. Pengembang akan membuat representasi dari tampilan antarmuka pengguna dan fungsi-fungsi yang nantinya akan diberikan pada sistem. *Prototype* dikembangkan agar pengguna mendapatkan gambaran dari sistem yang akan dikembangkan.

## 4. Pengujian dan pemberian umpan balik

Pada tahap pengujian dan pemberian umpan balik, pengguna akan menilai *prototype* yang telah dikembangkan oleh pengembang. Pengguna akan memberikan *feedback*, kritik, dan saran atas sistem yang akan dikembangkan agar sesuai dengan kebutuhan-kebutuhan pengguna. Setelah *prototype* telah sesuai dengan keinginan pengguna, maka sistem akan dikembangkan berdasarkan *prototype* yang telah disetujui oleh pengembang dan pengguna.

Kelebihan dari metode *prototyping* adalah sebagai berikut [30]:

1. Pengembangan sistem menggunakan metode *prototyping* memiliki jangka waktu yang cepat dan efisien.
2. Pengguna memiliki kontrol yang besar terhadap perancangan sistem sehingga pengembangan sistem akan sesuai dengan keinginan pengguna.
3. Seluruh revisi yang dilakukan kepada sistem merupakan saran langsung dari pengguna, sehingga informasi mengenai perbaikan sistem menjadi lebih akurat.
4. Pengembang dan pengguna membangun komunikasi yang lebih baik pada saat proses perancangan sistem.

### 2.3.3 *Waterfall*

*Waterfall* atau yang biasa disebut juga metode air terjun adalah metode SDLC yang menggambarkan pendekatan sistematis dalam mengembangkan suatu *software* [31]. Metode ini diperkenalkan oleh Winston Royce pada tahun 1970 dan seringkali dianggap metode yang kuno atau *outdated*. Secara keseluruhan, terdapat beberapa proses yang dilalui dalam SDLC ini, yaitu perencanaan, pemodelan, konstruksi, dan penyerahan sistem ke pengguna. Seluruh proses-proses tersebut dilakukan secara berurutan dan sistematis, sehingga suatu proses tidak dapat dimulai tanpa proses sebelumnya diselesaikan terlebih dahulu. Tahapan dalam metode *waterfall* dapat dijabarkan sebagai berikut:

#### 1. *Requirement*

Tahap pertama dalam *waterfall* adalah *requirement*. Pada tahap ini, kebutuhan-kebutuhan pengguna dikumpulkan untuk mengetahui harapan-harapan pengguna atas perangkat lunak yang dikembangkan.

## 2. *Design*

Tahap kedua dalam *waterfall* adalah *design*. Tahap ini merupakan tahap di mana kebutuhan dan harapan pengguna digunakan untuk merancang arsitektur sistem secara keseluruhan. Tahap ini menghasilkan desain yang dijadikan acuan pengembang aplikasi untuk mengembangkan sistem.

## 3. *Implementation*

Tahap ketiga dalam *waterfall* adalah *implementation*. Pada tahap ini, pengembang melakukan pengembangan atas sistem yang sudah dirancang. Setiap fungsionalitas sistem dikembangkan dalam unit program, di mana setiap unitnya akan melalui tahap uji coba.

## 4. *Verification*

Tahap keempat adalah *verification*. Sistem yang berada pada tahap ini akan dilakukan uji coba. Setiap unit yang telah dikembangkan akan melalui proses *testing* untuk memastikan bahwa seluruh fungsionalitas dapat berjalan dengan baik.

## 5. *Maintenance*

Tahap terakhir adalah *maintenance*. Pada tahap ini, aplikasi sudah dijalankan dan akan dilakukan pemeliharaan, seperti memperbaiki kesalahan-kesalahan dalam aplikasi dan menjaga agar aplikasi tetap berjalan sesuai dengan permintaan pengguna.

### 2.3.4 **Rapid Application Development (RAD)**

*Rapid Application Development* (RAD) merupakan metode pengembangan sistem yang menekankan kecepatan dalam proses pengembangan sistem dan menghasilkan kualitas sistem yang lebih tinggi [32]. RAD merupakan teknik yang menggabungkan teknik *prototyping* dengan teknik *joint application development* untuk

mempercepat proses pengembangan sistem. Secara keseluruhan, berikut adalah langkah-langkah dalam metode pengembangan sistem menggunakan *rapid application development* (RAD).

1. *Requirement Planning*

Tahap pertama dalam pengembangan menggunakan metode RAD adalah pengumpulan kebutuhan-kebutuhan pengguna. Dalam tahap ini, pengembang akan berdiskusi dengan pengguna untuk mengetahui spesifikasi dan fungsionalitas yang harus ada dalam suatu sistem.

2. *System Design*

Tahap kedua dalam metode RAD adalah desain sistem. Tahap ini melibatkan proses perancangan arsitektur sistem untuk memastikan bahwa seluruh sistem yang akan dikembangkan memenuhi kebutuhan-kebutuhan pengguna yang sudah ditentukan sebelumnya.

3. *Development*

Pada tahap ini, pengembang melakukan pengembangan aplikasi sesuai dengan desain yang sudah dibuat pada proses sebelumnya. Kunci kecepatan dari proses pengembangan sistem menggunakan RAD adalah digunakannya modul-modul dalam aplikasi. Seluruh modul dikembangkan dan digabungkan untuk menjadi satu sistem yang utuh. Oleh karena itu, RAD akan baik diimplementasikan dalam sistem yang memiliki banyak modul. Proses ini akan menghasilkan sistem yang sudah siap untuk dilakukan uji coba.

4. *Cut-Over*

Pada tahap terakhir, proses *testing* akan dilakukan. Sistem yang sudah dikembangkan dapat dilakukan uji coba untuk memastikan seluruh fungsionalitas berjalan dengan baik.



Tahap ini juga bertujuan untuk memastikan bahwa sistem yang dikembangkan sudah sesuai dengan kebutuhan pengguna.

### 2.3.5 *Unified Modelling Language (UML)*

*Unified Modelling Language (UML)* merupakan kumpulan dari praktik-praktik *software engineering* yang sukses dan dikumpulkan menjadi suatu spesifikasi pemodelan *software* yang komprehensif [33]. *Unified Modelling Language* merupakan suatu bahasa standar untuk membuat suatu spesifikasi, visualisasi, konstruksi, dan dokumentasi atas artefak sistem intensif *software*. Pemodelan sistem menggunakan UML memungkinkan proses *engineering* dari perangkat lunak berorientasi objek untuk dapat divisualisasikan dan dianalisis dengan cara yang terstandarisasi. Dengan demikian, *software engineer* dapat mengomunikasikan *software* yang didesain dengan lebih baik.

UML digunakan untuk membuat suatu bahasa pemodelan yang terstandarisasi dan memenuhi beberapa prinsip berikut ini:

1. Tidak bergantung pada bahasa pemrograman
2. Tidak bergantung pada proses pengembangan
3. Mendukung konsep-konsep pengembangan dengan tingkat yang lebih tinggi
4. Dapat diperluas dan dikembangkan khusus untuk proyek-proyek tertentu
5. Mengintegrasikan berbagai praktik terbaik dari bidang *software engineering*
6. Ekspresif, siap digunakan, dan dapat disediakan untuk membuat suatu model yang bermanfaat dan bermakna

Terdapat beberapa model yang dihasilkan pada proses pembuatan dan penentuan UML, yaitu *use case diagram*, *activity diagram*, dan *class diagram*.

### 1. Use Case Diagram

*Use case diagram* merepresentasikan interaksi antara *entity* eksternal dengan sistem [33]. Suatu *use case diagram* membuat model sistem dari sudut pandang *external observer*. Pada dasarnya, informasi yang digambarkan dalam *use case diagram* bukanlah cara sistem bekerja, melainkan hal-hal yang dapat dilakukan oleh sistem. Oleh karena itu, *use case diagram* dapat disebut juga dengan kumpulan dan ringkasan dari skenario yang termasuk dalam tugas-tugas individual. Berikut adalah tabel 2.1 yang menjelaskan komponen-komponen dalam *use case diagram* [34].

**Tabel 2.1** Komponen *Use Case Diagram*

Komponen	Penjelasan
<i>Use case</i>	Fungsionalitas yang pada sistem yang dikembangkan. <i>Use case</i> biasanya diawali dengan kata kerja.
<i>Actor</i>	Entitas eksternal, seperti suatu organisasi atau orang, yang berinteraksi dengan sistem.
<i>Association</i>	Garis yang melambangkan komunikasi dan interaksi antara suatu aktor dengan <i>use case</i> pada diagram.
<i>Extend</i>	Relasi dalam <i>use case diagram</i> yang menggambarkan suatu <i>use</i>

	<i>case</i> yang ditambahkan dari <i>use case</i> lain. <i>Use case</i> yang ditambahkan bersifat independen dan tidak bergantung pada <i>use case</i> tambahannya meskipun dihilangkan.
<i>Include</i>	Relasi dalam <i>use case diagram</i> yang menggambarkan <i>use case</i> yang ditambahkan ke dalam <i>use case</i> lain. Namun, <i>use case</i> yang ditambahkan bersifat dependen karena membutuhkan <i>use case</i> tambahan tersebut untuk menjalankan fungsinya.
<i>Generalization</i>	Hubungan generalisasi dan spesialisasi yang terjadi dalam dua <i>use case</i> . Salah satu <i>use case</i> yang memiliki hubungan generalisasi dengan <i>use case</i> lain merupakan fungsi yang lebih umum dari lainnya.

## 2. Activity Diagram

*Activity diagram* merupakan suatu model yang membantu pemahaman sistem berdasarkan proses-proses yang terjadi di dalamnya [33]. Pada dasarnya, *activity diagram* menggambarkan seluruh aktivitas yang berhubungan dengan suatu proses dan dependensi yang berhubungan dengan aktivitas-aktivitas tersebut. Berikut adalah tabel 2.2 yang menggambarkan komponen-komponen dalam *activity diagram* [34].

**Tabel 2.2** Komponen *Activity Diagram*

Komponen	Penjelasan
Status awal	Awal atau permulaan dari aktivitas sistem yang dijelaskan pada diagram.
Aktivitas	Aktivitas sistem yang digambarkan dalam <i>activity diagram</i> dan biasanya dimulai dengan kata kerja.
Percabangan	Percabangan merupakan proses untuk memilih suatu pilihan/ <i>decision</i> ketika terdapat beberapa kemungkinan dalam aktivitas sistem.
Status akhir	Akhir dari aktivitas-aktivitas yang dijalankan oleh sistem.
<i>Swimlane</i>	Diagram yang menggambarkan proses dan dipisahkan berdasarkan unit-unit bisnis dalam organisasi yang menjalankan aktivitas sistem tersebut.

### 3. *Class Diagram*

*Class Diagram* merupakan suatu model yang memvisualisasikan suatu sistem perangkat lunak berdasarkan abstraksi atau *class* dengan hubungan-hubungan yang terdapat antara satu *class* dengan *class* lainnya [33]. Dalam fase analisis, *class diagram* digunakan untuk mengindikasikan peran-peran umum dan tanggung jawab yang diasosiasikan dengan seluruh entitas yang mendefinisikan perilaku sistem. Pada fase desain,

*class diagram* digunakan untuk menggambarkan arsitektur sistem berdasarkan struktur dari setiap *class* yang terdiri di dalamnya. Berikut adalah tabel 2.3 yang berisi komponen-komponen yang terdapat pada *class diagram* beserta penjelasannya [34].

**Tabel 2.3** Komponen *Class Diagram*

<b>Komponen</b>	<b>Penjelasan</b>
<i>Class</i>	Kelas yang terdapat pada struktur sistem
<i>Association</i>	Relasi yang menghubungkan suatu kelas dengan kelas lainnya yang memiliki makna umum.
<i>Generalization</i>	Relasi yang menghubungkan satu kelas dengan kelas lainnya dengan makna umum dan khusus.
<i>Composition</i>	Relasi yang memiliki makna kebergantungan, yaitu suatu <i>class</i> tidak dapat berdiri tanpa <i>class</i> lain yang memiliki hubungan <i>composition</i> langsung dengannya.
<i>Aggregation</i>	Relasi suatu kelas dengan kelas lainnya yang menggambarkan makna semua-bagian. Pada relasi ini, suatu <i>class</i> dapat berdiri tanpa <i>class</i> lain yang memiliki hubungan <i>aggregation</i> secara langsung.

### 2.3.6 Bahasa Pemrograman

Berikut adalah beberapa bahasa pemrograman yang digunakan dalam pengembangan sistem penjadwalan pada penelitian ini:

1. *Hyper Text Markup Language*

*Hyper Text Markup Language* (HTML) adalah bahasa pemrograman untuk proses pembuatan halaman web yang dapat digunakan untuk memaparkan informasi dalam sebuah *internet browser* [35]. HTML digunakan sebagai penghubung file dalam suatu *website*, dalam komputer melalui *localhost*, ataupun menghubungkan antar situs dalam internet.

2. *Cascading Style Sheets*

*Cascading Style Sheets* (CSS) adalah teknologi pengembangan website untuk memberikan style seperti jenis tulisan, warna, *margin*, *padding*, dan lainnya kepada elemen-elemen di dalam suatu halaman web [36]. Dengan adanya CSS, pengembang dapat melakukan styling dari masing-masing elemen tanpa perlu mendeskripsikan ulang tampilan dari setiap elemen yang memiliki ciri yang sama. Kode yang dihasilkan akan menjadi lebih singkat dan meminimalisir error.

3. *PHP*

PHP adalah bahasa pemrograman untuk *website* yang bersifat *server-side* [37]. PHP dapat menggunakan basis data untuk membuat suatu halaman yang bersifat dinamis. PHP dapat digunakan bersama sistem manajemen basis data *MYSQL*, *Oracle*, *Microsoft Access*, *PostgreSQL*, *Interbase*, dan *d-Base*.

#### 4. Javascript

Javascript merupakan bahasa pemrograman, yaitu kumpulan script yang dieksekusikan pada suatu file HTML [38]. Karakteristik Javascript antara lain adalah:

1. Javascript berjenis high-level programming.
2. Javascript bersifat client-side.
3. Pemrograman berorientasi objek.
4. Pemrograman bersifat loosely typed.

Pemrograman Javascript dapat dilakukan di beberapa platform, yaitu software text-editor seperti Adobe Dreamweaver dan NetBeans, web browser seperti Google Chrome dan Mozilla Firefox, dan juga HTML. Salah satu *tools* yang digunakan dalam implementasi *javascript* untuk pengembangan sistem berbasis web adalah ReactJS.

React merupakan suatu *tools* dari bahasa *Javascript* untuk membuat komponen-komponen tampilan antarmuka yang interaktif dan *reusable* [39]. ReactJS sangat baik digunakan untuk melakukan *rendering* tampilan antarmuka yang kompleks dengan performa yang baik. Dasar dari *framework* ReactJS adalah penggunaan virtual DOM yang dapat di-*render* baik pada sisi client maupun sisi *server* dan komunikasi di antara keduanya. Virtual DOM akan *render* beberapa komponen-komponen berdasarkan *state* yang berganti. ReactJS akan meminimalisir terjadinya manipulasi DOM untuk memperbaharui komponen-komponen yang ada di dalamnya. Hal itu menyebabkan adanya salah satu keuntungan utama dari ReactJS, yaitu pemrosesan yang cepat.

Di sisi lain, terdapat keterbatasan yang dialami dalam penggunaan ReactJS, yaitu[39]:

1. React hanya dapat digunakan untuk pengembangan view dalam model MVC. Oleh karena itu, dibutuhkan tools tambahan untuk dapat mengembangkan suatu aplikasi yang utuh.
2. Penggunaan format inline dan JSX dapat menjadi hal yang sulit bagi para pengembang yang menggunakan ReactJS.

## 2.4 Penelitian Terdahulu

**Tabel 2.4** Tabel Penelitian Terdahulu

<b>Nama Jurnal</b>	<b>Nama Artikel dan Penulis</b>	<b>Hasil</b>	<b>Yang Diadopsi</b>
Journal Of Computer Science And Informatics Engineering (J-Cosine), Vol. 5, No. 1, Tahun 2021	<p><b>Nama artikel:</b> Optimasi Penjadwalan Ujian Tugas Akhir Dengan Menggunakan Algoritma Genetika</p> <p><b>Penulis:</b> Adi Panca Saputra Iskandar</p>	<p>Penelitian ini menghasilkan optimasi sistem penjadwalan seminar dan sidang tugas akhir di STMIK STIKOM Indonesia menggunakan algoritma genetika. Dalam melakukan optimasi penjadwalan seminar dan sidang tugas akhir, proses-proses seperti menentukan populasi dan kromosom, mengimpleme ntasi <i>fitness</i></p>	<p>Penelitian ini mengadopsi penggunaan algoritma genetika dalam mengembangkan sistem <i>scheduling</i> sidang skripsi mahasiswa tingkat akhir Universitas Multimedia Nusantara.</p>



Nama Jurnal	Nama Artikel dan Penulis	Hasil	Yang Diadopsi
		<p><i>function</i>, melakukan proses seleksi kromosom, melakukan <i>crossover</i> dan <i>mutation</i>, dan melakukan elitism akan dilalui. Dari hasil mutasi dan <i>crossover</i> pada setiap kromosom dan populasi, sistem akan menghasilkan pemilihan jadwal dan peserta yang paling sesuai dalam melaksanakan seminar dan sidang tugas akhir untuk mahasiswa tingkat akhir STMIK STIKOM Indonesia.</p>	
<p>Ultima Infosys: Jurnal Ilmu Sistem Informasi, Vol. 10, No 1, Tahun 2019</p>	<p><b>Nama artikel:</b> Sistem Informasi Penunjang Proses Pemesanan dan Desain Kue Pada Toko Kue Artisan Online Berbasis Web <b>Penulis:</b> Tanita Jovita Wibowo, Suryasari, Arnold</p>	<p>Penelitian ini menghasilkan sistem pemesanan dan desain kue yang dikembangkan menggunakan metode <i>prototyping</i>. Sistem yang dibangun</p>	<p>Penelitian ini akan mengadopsi penggunaan metode <i>prototyping</i> sebagai SDLC dalam mengembangkan sistem <i>scheduling</i> sidang skripsi Universitas</p>

Nama Jurnal	Nama Artikel dan Penulis	Hasil	Yang Diadopsi
	Ariwibowo, Andree E. Widjaja	memudahkan proses pemesanan kue dan mendukung proses komunikasi antara penjual dan pembeli. Pembeli dapat melakukan pemesanan secara <i>custom</i> menggunakan <i>sketch tool</i> yang ada pada sistem. Dengan adanya sistem informasi ini, pemilik toko dapat memproses pemesanan pelanggan dengan lebih efektif dan efisien karena seluruh proses pemesanan dilakukan melalui satu sistem.	Multimedia Nusantara.
Jurnal Teknik Informatika (Jutif), Vol 3, No 2, Tahun 2022	<b>Nama artikel:</b> <i>Frontend Development Of Course Scheduling System Integrated SIA At Engineering Faculty University Of Jenderal</i>	Penelitian ini menghasilkan suatu sistem penjadwalan mata kuliah yang dikembangkan menggunakan <i>React JS</i> . Sistem yang dikembangkan	Penelitian ini akan mengadopsi penggunaan <i>React JS</i> untuk membangun sistem informasi berbasis web dalam mengembangkan sistem <i>scheduling</i> sidang skripsi

Nama Jurnal	Nama Artikel dan Penulis	Hasil	Yang Diadopsi
	<i>Soedirman Using Devops Method</i> <b>Penulis:</b> Herfina Intan Yuanita, Bangun Wijayanto, Teguh Cahyono	berhasil <i>men-generate</i> tugas model jadwal mata kuliah.	Universitas Multimedia Nusantara.
Sensors, Vol. 2-, No. 9, Tahun 2020	<b>Nama artikel:</b> <i>A Consortium Blockchain-Based Agricultural Machinery Scheduling System</i> <b>Penulis:</b> Haotian Yang, Shuming Xiong, Samuel Akwasi Frimpong, Mingzheng Zhang	Penelitian ini bertujuan untuk membangun sistem penjadwalan mesin-mesin agrikultur yang didukung dengan basis data Oracle. Sistem yang dibangun dapat mengurangi biaya secara signifikan dan tidak membutuhkan <i>central server</i> . Oleh karena itu, sistem tidak akan mengalami gangguan apabila <i>central server</i> mengalami gangguan.	Penelitian ini akan mengadopsi penggunaan Oracle sebagai manajemen basis data dalam mengembangkan sistem <i>scheduling</i> sidang skripsi Universitas Multimedia Nusantara.

Pada tabel 2.4, terdapat penelitian terdahulu pertama yang membahas mengenai optimasi terhadap sistem penjadwalan seminar dan sidang tugas akhir di STMIK STIKOM Indonesia menggunakan algoritma genetika [40].

Tahap-tahap dalam penelitian ini adalah menentukan populasi dan kromosom, menghitung *fitness value*, melakukan seleksi kromosom, melakukan *crossover* dan *mutation*, dan melakukan elitism untuk menghasilkan kromosom atau solusi terbaik atas penjadwalan seminar dan sidang tugas akhir di STMIK STIKOM Indonesia dengan lebih optimal. Penelitian ini akan dijadikan referensi dalam penggunaan algoritma genetika untuk mengembangkan sistem penjadwalan sidang skripsi pada Universitas Multimedia Nusantara jurusan Sistem Informasi. Pembaharuan yang terdapat pada penelitian ini adalah digunakannya bahasa *javascript* dalam pengembangan algoritma genetika.

Penelitian terdahulu kedua membahas mengenai pengembangan sistem pemesanan dan desain kue yang dikembangkan menggunakan metode *prototyping* [41]. Sistem dapat membantu proses penjualan kue dan membantu komunikasi antara penjual dan pembeli. Pembeli dapat menentukan desain kue menggunakan sketch tool yang disediakan pada sistem. Dengan adanya sistem ini, pemilik toko dapat memproses pemesanan pelanggan dengan lebih efektif dan efisien. Penelitian ini akan dijadikan referensi dalam penggunaan metode SDLC *prototyping* untuk mengembangkan sistem informasi penjadwalan sidang skripsi.

Penelitian terdahulu yang ketiga membahas mengenai pengembangan sistem informasi untuk *generate* jadwal mata kuliah [42]. Penelitian ini juga membuktikan bahwa pengembangan *front end* menggunakan *React JS* dapat memudahkan dan meminimalkan penulisan kode dengan adanya fitur *reusable component*. Penelitian ini akan dijadikan referensi dalam penggunaan *React JS* dalam mengembangkan sistem penjadwalan sidang skripsi berbasis web untuk Universitas Multimedia Nusantara jurusan sistem informasi. Pembaharuan yang terdapat pada penelitian ini adalah bahasa pemrograman *React JS* sebagai *frontend* akan dipadukan dengan Yii 2 sebagai *backend* dari sistem yang dikembangkan.

Penelitian terdahulu yang keempat membahas mengenai pengembangan sistem penjadwalan mesin-mesin agrikultur [43]. Sistem yang dikembangkan dapat secara signifikan menekan *cost* dan tidak tergantung pada *central server*. Algoritma yang digunakan pada sistem penjadwalan merupakan algoritma genetika yang dapat mempertimbangkan jadwal mesin agrikultur berdasarkan beberapa faktor seperti cuaca, jalanan, biaya, keuntungan, dan reputasi perusahaan. Penelitian ini akan dijadikan referensi dalam penggunaan sistem Oracle untuk melakukan manajemen basis data yang akan digunakan dalam sistem penjadwalan sidang skripsi berbasis web untuk Universitas Multimedia Nusantara jurusan sistem informasi. Pembaharuan yang terdapat pada penelitian ini adalah digunakannya *framework* Yii 2 dalam bahasa yang digunakan pada *backend* untuk mengakses *database*.

