

BAB II

LANDASAN TEORI

2.1 Kajian Terhadap Penelitian Yang Terkait Sebelumnya

Berikut ulasan beberapa penelitian terkait yang menjadi referensi yaitu:

1. Pada tahun 2018 penelitian yang disusun oleh Freddy Nur Afandi dan Mila Yulianis dalam jurnal yang berjudul: “Implementasi Genetic Algorithms Untuk Penjadwalan Mata Kuliah Berbasis Website” pada judul ini digunakan metode Algoritma Genetika untuk membuat sebuah jadwal perkuliahan yang dapat membantu Ka Prodi.
2. Pada tahun 2021 penelitian yang disusun Rendy Christian dan Dwi Sukma Donoriyanto dalam jurnal yang berjudul: “Penerapan Algoritma Genetika Dalam Penjadwalan Mata Kuliah Program Studi Teknik Industri UPN “Veteran” Jawa Timur”, pada judul ini digunakan metode Algoritma Genetika untuk membuat sebuah jadwal perkuliahan tanpa bentrok berbasis website.
3. Pada tahun 2018 penelitian yang disusun oleh Yasmin Math Khader, Youllia Indrawaty Nurhasanah dan Afriyanti Dwi Kartika dalam jurnal yang berjudul: “Penjadwalan Mata Kuliah Menggunakan Algoritma *Greedy* (Studi Kasus Penjadwalan Semester Ganjil 2017-2018 Informatika ITENAS”, pada judul ini digunakan metode Algoritma *Greedy* untuk membuat sebuah jadwal perkuliahan semester ganjil 2017-2018.

2.2 Pengertian Sistem

Menurut Rendy Christian, dkk (2021) menuturkan pengertian sistem artinya sekelompok komponen serta elemen yang digabungkan menjadi satu untuk mencapai tujuan eksklusif. Sistem berasal bahasa Latin (*systema*) serta bahasa Yunani (*systema*) merupakan suatu kesatuan yang terdiri komponen atau elemen yang dihubungkan beserta memudahkan aliran informasi, materi atau energi untuk mencapai suatu tujuan.

2.3 Pengertian Sistem Informasi

Menurut Rendy Christian, dkk (2021) menjelaskan bahwa sistem informasi merupakan gugusan asal subsistem apapun baik fisik ataupun non fisik yang saling berafiliasi satu sama lain dan bekerja sama secara serasi untuk mencapai satu tujuan yaitu mengolah data sebagai isu yang berarti serta berguna.

2.4 Pengertian Penjadwalan

Menurut Yasmin Math Khader, dkk (2018) menuturkan pengertian penjadwalan merupakan proses, cara, perbuatan menjadwalkan atau memasukkan dalam jadwal. Definisi yang lebih umum adalah menugaskan satu kumpulan peristiwa (kuliah, kendaraan, acara-acara public, dll) dengan kumpulan terbatas sumber dari waktu ke waktu sedemikian rupa untuk memenuhi kendala yang telah ditetapkan, kendala ini dapat dikategorikan sebagai *hard constraint* dan *soft constraint*, dimana *hard constraint* memiliki prioritas yang lebih tinggi daripada *soft constraint*.

2.5 PHP

Menurut Freddy Nur Affandi dan Mila Yulianis (2018) menuturkan penjelasan PHP (*Hypertext Preprocessor*) yaitu suatu bahasa pemrograman berbasis kode-kode (*script*) yang digunakan untuk mengolah suatu data dan mengirimkannya kembali ke *web browser* menjadi kode HTML.

2.6 XAMPP

Menurut Freddy Nur Affandi dan Mila Yulianis (2018) menjelaskan bahwa XAMPP adalah sebuah *software* yang berfungsi untuk menjalankan *website* berbasis PHP dan menggunakan pengolahan data MySQL di komputer lokal. XAMPP berperan sebagai *server web* pada komputer. XAMPP juga dapat disebut sebuah *Cpanel server* virtual, yang dapat membantu anda melakukan *preview* sehingga dapat memodifikasi *website* tanpa harus *online* atau terakses dengan *internet*.

2.7 MySQL

Menurut Freddy Nur Affandi dan Mila Yulianis (2018) menjelaskan bahwa MySQL adalah program *database* yang mampu mengirim dan menerima data dengan sangat cepat dan *multi user*. MySQL memiliki dua bentuk lisensi, yaitu *free software* dan *shareware*. MySQL *database server* adalah RDBMS (*Relational Database Management System*) dapat menangani data yang bervolume besar.

2.8 UML

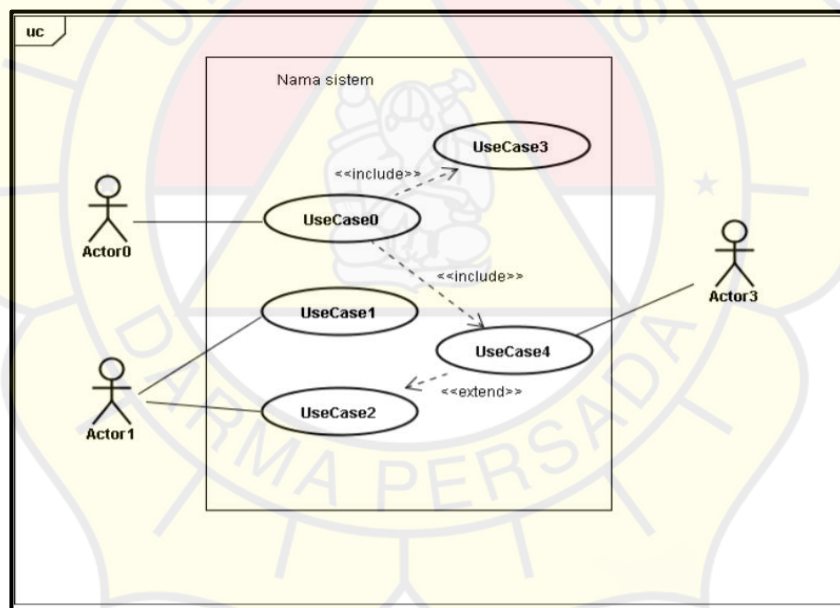
Menurut Ibnu Akil (2018) dikatakan bahwa UML (*Unified Modeling Language*) adalah bahasa pemodelan visual yang digunakan untuk menspesifikasikan, memvisualisasikan, membangun, dan mendokumentasikan rancangan dari suatu sistem perangkat lunak. Pemodelan memberikan gambaran

yang jelas mengenai sistem yang akan dibangun baik dari sisi struktural ataupun fungsional.

2.8.1 Use Case Diagram

Menurut Hammim Tohari (2014) mengatakan bahwa *Use Case* adalah rangkaian atau uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah actor. *Use Case* digunakan untuk membentuk tingkah laku benda dalam sebuah model serta direalisasikan oleh sebuah kolaborasi.

Elemen yang digunakan pada diagram use case:



Gambar 2.1 Elemen-Elemen Diagram Use Case

A. Sistem

Sistem menyatakan batasan sistem dalam relasi dengan aktor-aktor yang menggunakannya (di luar sistem) dan fitur-fitur yang harus disediakan (dalam sistem).

B. Aktor

Aktor atau aktor dapat berupa manusia, sistem, atau *device* yang memiliki peranan dalam keberhasilan operasi dari sistem. Digambarkan dengan *icon* yang mungkin bervariasi namun konsepnya sama.

C. Use Case

Use case mengidentifikasi fitur kunci dari sistem. Tanpa fitur ini, sistem tidak akan memenuhi permintaan *user/actor*. Setiap *use case* mengekspresikan *goal* dari sistem yang harus dicapai.

D. Assosiation

Mengidentifikasi interaksi antara setiap aktor tertentu dengan setiap use case tertentu. Digambarkan dengan garis antara aktor terhadap use case yang bersangkutan.

2.8.2 Activity Diagram

Menurut Hammim Tohari (2014) menjelaskan *Activity Diagram* memodelkan *workflow* proses bisnis dan urutan aktivitas dalam sebuah proses. Diagram ini sangat mirip dengan *flowchart* karena memodelkan *workflow* dari satu aktivitas ke aktivitas lainnya atau dari aktivitas ke status.

Elemen-elemen dari *Activity Diagram* meliputi ;

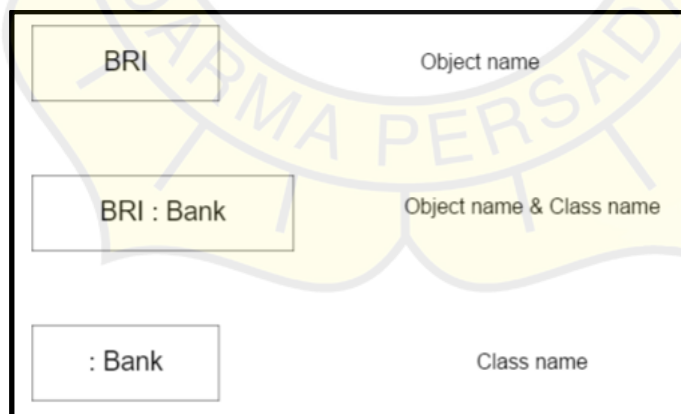
1. Status Start (mulai) dan end (akhir).
2. Aktivitas yang merepresentasikan sebuah langkah dalam *workflow*.
3. *Transition* menunjukkan terjadinya perubahan status aktivitas (*transition show what State follows another*).
4. Keputusan yang menunjukkan *alternative* dalam *workflow*.

5. *Synchronization bars* yang menunjukkan *subflow parallel*. *Synchronization bars* dapat digunakan untuk menunjukkan *concurrent threads* pada *workflow* proses bisnis.
6. *Swimlanes* yang merepresentasikan *role* bisnis yang bertanggung jawab pada aktivitas yang berjalan.

2.8.3 *Sequence Diagram*

Menurut Hammim Tohari (2014) menjelaskan *Sequence Diagram* menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi sistem.

Dalam UML, objek pada diagram sequence digambarkan dengan segi empat, yang berisi nama dari objek yang digarisbawahi. Terdapat tiga (3) cara untuk menamai objek yaitu, nama objek, nama objek dan *class* serta nama *class*. Misalnya:

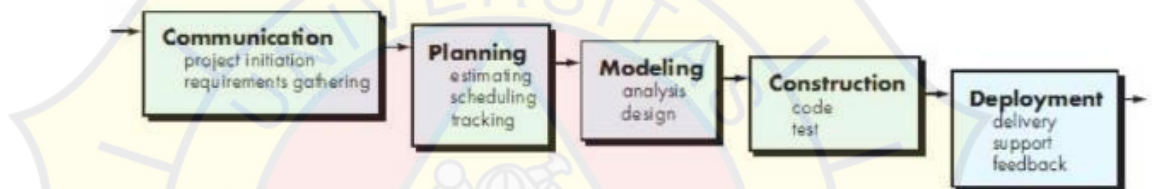


Gambar 2.2 Penamaan Objek

Pada *diagram sequence*, setiap objek hanya memiliki garis yang digambarkan garis putus-putus ke bawah. Pesan antar objek digambarkan dengan anak panah dari objek yang mengirimkan pesan ke objek yang menerima pesan.

2.9 Metode Pengembangan Sistem

Menurut Pressman (2012) Model waterfall adalah model klasik yang bersifat sistematis, berurutan dalam membangun software. Karena untuk melanjutkan ke tahap berikutnya, harus menyelesaikan tahap sebelumnya seperti berjalan berurutan.



Gambar 2.3 Model Waterfall

Tahap-tahap dari model waterfall:

1. *Communication*

Menganalisa permasalahan dan mengumpulkan data-data yang harus disiapkan terhadap kebutuhan sistem yang akan dibuat. Hasil yang didapat berupa masalah yang ada dan membantu mendefinisikan fitur dan fungsi perangkat lunak yang dibutuhkan.

2. *Planning*

Perencanaan menjelaskan estimasi tugas-tugas atau kebutuhan dari perangkat lunak, resiko-resiko yang dapat terjadi, sumber daya yang diperlukan

dalam membuat sistem, sehingga mendapatkan hasil agar dapat diimplementasikan menjadi program pada tahap selanjutnya.

3. *Modeling*

Tahap ini mengerjakan desain berdasarkan hasil pengumpulan informasi dan analisa kebutuhan. Proses yang ada dalam tahap ini berfokus pada desain perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka dan algoritma program.

4. *Construction*

Melakukan pengkodean dari hasil desain yang dilakukan pada tahap sebelumnya ke dalam program perangkat lunak. Hasil pada tahap ini berupa program komputer sesuai desain dan kebutuhan.

5. *Deployment*

Setelah tahap-tahap sebelumnya sudah selesai dijalankan, dilakukan uji coba kepada user. Kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirim ke user. Akan adanya perubahan dikarenakan bisa terjadinya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau karena lingkungan baru tempat perangkat lunak diterapkan.

2.10 Algoritma Genetika

Menurut Zainudin Zuhri (2014, h.10) mengatakan bahwa Algoritma Genetika merupakan suatu metode heuristik yang dikembangkan berdasarkan prinsip genetika dan proses seleksi alamiah Teori Evolusi Darwin. Proses pencarian penyelesaian atau proses terpilihnya sebuah penyelesaian dalam algoritma ini

berlangsung sama seperti terpilihnya suatu individu untuk bertahan hidup dalam proses evolusi.

Proses pencarian dalam Algoritma Genetika dimulai dengan pembangkitan sejumlah “individu” secara acak yang disebut dengan kromosom. Kromosom-kromosom ini merupakan representasi calon penyelesaian yang akan diperiksa nilai yang sebenarnya. Seperti halnya proses evolusi alamiah, kromosom-kromosom akan dinilai tingkat kebugarannya. Hanya kromosom dengan tingkat kebugaran yang tinggi saja yang terpilih untuk bertahan dalam populasi.

Kromosom-kromosom yang terpilih sebagian akan melakukan proses reproduksi melalui penyilangan (crossover). Proses reproduksi ini mirip dengan perkawinan individu dalam proses evolusi. Proses reproduksi ini akan melahirkan individu-individu baru.

Gabungan dari individu-individu baru dengan kromosom-kromosom yang tidak melakukan proses reproduksi akan membentuk populasi baru pada generasi berikutnya. Serangkaian proses seperti ini berlangsung sampai sejumlah generasi tercapai. Penyelesaian yang ditemukan adalah kromosom yang mempunyai tingkat kebugaran yang paling tinggi pada generasi terakhir.

2.11 Algoritma Greedy

Menurut Khader dkk (2018) menjelaskan Algoritma *Greedy* merupakan salah satu jenis algoritma *Best First Search* yang paling sederhana dengan hanya memerhitungkan biaya perkiraan (*estimated cost*) saja, yakni $f(n) = h(n)$. *Greedy Algorithm* adalah kelompok algoritma yang selalu mengambil penyelesaian sementara/lokal yang terbaik dalam setiap langkahnya untuk menyelesaikan suatu

permasalahan. Pilihan terbaik akan diambil pada setiap langkah tanpa perlu memikirkan bagaimana pengaruhnya terhadap penyelesaian secara keseluruhan.





TEKNOLOGI INFORMASI

UNIVERSITAS DARMA PERSADA