

## BAB II

### LANDASAN TEORI

#### 2.1 Perencanaan Produksi

Perencanaan produksi adalah proses perencanaan dan pengendalian aliran bahan masuk dan keluar dari sistem produksi. Perencanaan produksi bertujuan untuk menentukan apa, berapa banyak, dan kapan untuk diproduksi, dibeli, dan dikirimkan sehingga perusahaan dapat memenuhi permintaan pasar[3].

Dalam perencanaan produksi, aktivitas yang dilakukan yaitu mengevaluasi fakta lalu dan sekarang untuk kemudian ditentukan strategi agar penjadwalan dapat dilakukan secara efektif dan efisien. Perencanaan produksi dapat menjawab hal – hal sebagai berikut [4]:

- Mengapa (*Why*): Tujuan dibuatnya perencanaan produksi yaitu agar produk dapat memiliki nilai tambah, meningkatkan keuntungan dan memenuhi permintaan.
- Apa (*What*): Produk yang akan dibuat pada periode mendatang (rancangan, spesifikasi, kualitas, kuantitas)
- Siapa (*Who*): Yang diperlukan dalam membuat produk seperti tenaga kerja, mesin, peralatan dan alat bantu.
- Kapan (*When*): Mulai dan berakhirnya produksi tersebut dijadwalkan sehingga menjadi tepat waktu dengan memiliki biaya dan kualitas produk yang baik.
- Di mana (*Where*): Lokasi proses produksi dijalankan.
- Bagaimana (*How*): Metode kerja yang digunakan.

##### 2.1.1 Tujuan

Adapun tujuan dari perencanaan produksi yaitu [5] :

- Menentukan volume produksi berdasarkan perkiraan penjualan
- Mementukan keputusan pembelian

- Menentukan urutan pengoperasian dalam produksi suatu produk
- Menentukan setiap produk mengenai
  - o Jenis bahan yang digunakan
  - o Mesin yang akan dioperasikan
  - o Alat- alat yang digunakan
- Untuk menjalankan langkah – langkah untuk memenuhi target produksi yang sudah dijadwalkan dan sudah di *budget*-kan.

### 2.1.2 Jenis Perencanaan Produksi

Ada tiga level jenis perencanaan produksi berdasarkan jangka waktu, diantaranya [4]:

#### ➤ Perencanaan jangka pendek

Estimasi waktu perencanaan sekitar 1-30 hari, bertujuan untuk menghasilkan produk melalui penjadwalan

#### ➤ Perencanaan jangka menengah

Estimasi waktu perencanaan sekitar 1-24 bulan, bertujuan agar kapasitas maupun fasilitas yang dimiliki oleh suatu perusahaan dapat memenuhi permintaan.

#### ➤ Perencanaan panjang

Estimasi waktu perencanaan sekitar 2-10 tahun, bertujuan untuk membuat rencana strategi pengembangan perusahaan.

## 2.2 Pengendalian Stok

Pengendalian stok merupakan usaha penyediaan barang yang digunakan untuk menunjukkan seberapa banyak stok yang dimiliki pada satu waktu dan bagaimana melacaknya.

Stok memiliki berbagai tipe, diantaranya [6]:

- *Raw Material*: bahan utama yang digunakan dalam proses produksi suatu barang.
- *Semi-Finished Goods*: Produk hasil penggabungan beberapa *raw material*, namun produk belum selesai dalam produksi.
- *Finished Goods*: Produk jadi hasil proses produksi yang siap untuk dijual kepada pelanggan.

Pengendalian stok juga memiliki fungsi untuk[7]:

- Menghindari keterlambatan pengiriman bahan baku yang dibutuhkan.
- Menghindari pesanan barang diretur.
- Menghindari kenaikan harga secara mendadak.
- Memberikan pelayanan yang maksimal kepada pelanggan dengan kecukupan ketersediaan produk yang dibutuhkan oleh mereka.

Dari berbagai fungsi tersebut, manfaat yang dapat diperoleh adalah mendapatkan keseimbangan antara persediaan dan permintaan sehingga meminimalisir biaya operasional yang diakibatkan karena kelebihan produksi. Selain itu juga memiliki persediaan stok pengaman (*safety stock*) untuk mengantisipasi kekurangan stok baik *raw material*, *semi finished good*, maupun *finished goods*.

### 2.3 Metode Moora

*Moora* atau *Multi-Objective Optimization on the Basis of Ratio Analysis* merupakan salah satu metode pengambilan keputusan yang mengoptimalkan dua atau lebih atribut yang saling bertentangan secara bersamaan [8]. Adapun langkah proses dari metode *moora* yaitu [9], [10] :

#### 1. Membuat matriks keputusan

Matriks keputusan dipresentasikan sebagai matriks persamaan 2.1, dimana matriks keputusan berfungsi untuk melakukan pengukuran kinerja pada alternatif/baris ke-*i* pada atribut/kriteria ke-*j*, *m* sebagai jumlah alternatif/baris sedangkan *n* sebagai jumlah atribut/kriteria.

$$x_{ij} = \begin{bmatrix} x_{11} & x_{12} & \cdot & x_{1n} \\ x_{21} & x_{22} & \cdot & x_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ x_{m1} & x_{m2} & \cdot & x_{mn} \end{bmatrix} \quad (2.1)$$

**Keterangan:**

$x_{ij}$  = Matrik Keputusan alternatif i pada kriteria j

i = Alternatif (Baris)

j = attribute/kriteria (kolom)

m = jumlah alternatif/baris

n = Jumlah attribute/kriteria

2. Normalisasikan Matriks Keputusan

Adapun dilakukan normalisasi pada matriks keputusan adalah untuk menyeragamkan nilai yang ada pada matriks keputusan yang sudah ditentukan, yaitu dengan melakukan persamaan 2.2.

$$X^*_{ij} = \frac{x_{ij}}{\sqrt{[\sum_{j=1}^m x_{ij}^2]}} \quad (2.2)$$

**Keterangan:**

$x_{ij}$  = Matrik Keputusan alternatif i pada kriteria j

$x^*_{ij}$  = Matriks normalisasi pada alternatif i pada kriteria j

i = Alternatif (Baris)

j = attribute/kriteria (Kolom)

m = Jumlah alternatif/Baris

### 3. Optimalkan atribut

Pada tahapan ini, terdapat dua rumus yang menjadi acuan dalam perhitungan, jika atribut/kriteria mempunyai nilai bobot kepentingan, maka digunakan persamaan rumus persamaan 2.3.

$$y_i^* = \sum_{j=1}^g w_j x_{ij}^* - \sum_{j=g+1}^n w_j x_{ij}^* \quad (2.3)$$

**Keterangan:**

$x_{ij}^*$  = Matriks normalisasi pada alternatif i pada kriteria j

i = Alternatif (Baris)

j = attribute/kriteria (Kolom)

$w_j$  = bobot kriteria ke j

g = Jumlah atribut/kriteria/kolom dengan kriteria benefit

g+1 = Atribut/Kriteria (Kolom) dengan kriteria cost(biaya)

Sedangkan jika atribut/kriteria tidak diberikan nilai bobot kepentingan, maka digunakan rumus persamaan 2.4.

$$y_i^* = \sum_{j=1}^g x_{ij}^* - \sum_{j=g+1}^n x_{ij}^* \quad (2.4)$$

**Keterangan:**

$x_{ij}^*$  = Matriks normalisasi pada alternatif i pada kriteria j

j = attribute/kriteria (Kolom)

g = Jumlah atribut/kriteria/kolom dengan kriteria benefit

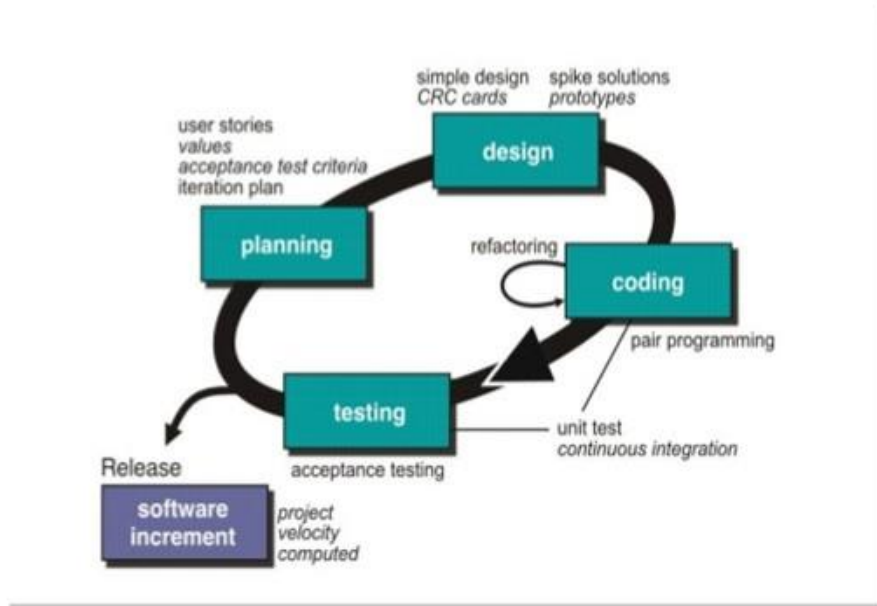
g+1 = Atribut/Kriteria (Kolom) dengan kriteria cost(biaya)

n = Jumlah attribute/kriteria/kolom

$y_i^*$  = Nilai Optimasi pada alternatif i

## 2.4 Metode Extreme Programming (XP)

# Extreme Programming (XP)



Gambar 2.41 Proses *Extreme Programming* (Sumber: [11])

*Extreme programming* menurut [11] adalah metodologi pengembangan perangkat lunak yang ditunjukkan untuk meningkatkan kualitas perangkat lunak. Ada beberapa tahapan dalam *Extreme Programming*, diantaranya [11] :

1. Perencanaan (*Planning*): Untuk memahami apa saja kriteria dari pengguna dan bagaimana rencana dalam pengembangannya.
2. Desain (*Design*): Merancang dan membangun *prototype*.
3. Pengkodean (*Coding*): Pengolahan dan pengintegrasian kode yang dituliskan dari bahasa pemrograman.
4. Pengujian (*Testing*): Pengetesan terhadap aplikasi yang dibuat.

Selain itu, proses dari Extreme Programming lainnya menurut [12] , yaitu:

#### 1. Exploration phase

Fase *exploration* merupakan fase pertama dari XP yang berhubungan dengan kebutuhan dan pemodelan arsitektur sistem. Kebutuhan pengguna, arsitektur, alat dan teknologi didefinisikan pada fase ini. Pelanggan menuliskan *user stories* yang berisikan persyaratan tentang *software* yang dibuat. *User Stories* diharuskan rinci untuk membantu pengembang memahami kebutuhan sistem dan juga dalam membuat perkiraan.

#### 2. Planning phase

Tahap *planning* bertujuan untuk menjelaskan mengenai apa yang dibangun dalam batas waktu yang memiliki beberapa nilai bisnis dan apa yang akan dilakukan pada iterasi berikutnya. Selama fase perencanaan, tugas yang dikerjakan untuk pembangunan sistem diambil dari *user stories*.

#### 3. Release Planning

Tujuan dari proses ini yaitu untuk mengetahui fitur-fitur apa yang dibutuhkan dalam sistem dan jadwal kapan dibutuhkannya fitur tersebut untuk dikirimkan atau diselesaikan. Baik dari pelanggan dan pengembang berpartisipasi dalam proses ini. Pelanggan menuliskan *stories* untuk mengidentifikasi fitur sistem yang diperlukan. Fitur-fitur ini kemudian diurutkan menurut kepentingannya.

#### 4. Iteration Planning

Di fase ini, pengembang menyiapkan rencana kegiatan untuk mengimplementasikan fitur yang diperlukan saat itu. Selama perencanaan iterasi, programmer memilih tugas untuk diimplementasikan dan memperkirakan biaya, waktu, dan upaya yang diperlukan untuk tugas yang dipilih.

#### 5. Iteration to release phase

Fase ini merupakan fase berulang dimana di dalamnya mencakup kegiatan perancangan, pengkodean, pengujian, dan integrasi. Setiap iterasi dimulai

dengan perencanaan iterasi. Pemrogram memilih tugas, membuat desain, dan pengkodean. Setelah pengkodean, dilakukan *functional testing* dan pengintegrasian kode.

#### 6. Productionizing phase

Fase produksi mendeploy *software* ke dalam rilis kecil. Rilis merupakan bagian kecil dari *software* terencana yang mengimplementasikan beberapa kebutuhan bisnis. Dalam XP, rilis memungkinkan untuk membangun sistem yang diperlukan secara bertahap. Untuk memeriksa apakah *software* siap untuk diproduksi, *acceptance testing*, *system testing*, dan *load testing* dilakukan.

#### 7. Maintenance phase

Di XP, *software* terus berkembang selama periode waktu tertentu. Dalam fase ini, fungsionalitas baru dibangun dengan menjaga yang lama tetap berjalan. Selain itu di fase ini dilakukan juga perbaikan jika *software* rusak.

#### 8. Death phase

Fase terakhir dari XP. Ada dua kemungkinan situasi dimana *software* berada dalam fase ini. Yang pertama jika *software* memiliki semua fungsionalitas yang dibutuhkan, pelanggan puas, dan tidak memiliki *user stories* lagi. Yang kedua, jika pelanggan memerlukan serangkaian fitur yang tidak dapat dikembangkan secara ekonomis.

### 2.5 Website

Website menurut [13] merupakan kumpulan halaman yang digunakan untuk menampilkan sekumpulan bentuk informasi yang konsisten seperti audio, teks, video atau gambar diam, audio, animasi, atau kombinasi dari keduanya dinamis dan statis.

Berdasarkan sifatnya, ada beberapa jenis website ketika akan membangun sebuah website, diantaranya:



## 1. Website Statis

Website yang kontennya tidak berubah atau konstan. Karena sifatnya yang statis, website ini tidak memerlukan database

## 2. Website Dinamis

Website dinamis adalah website yang kontennya selalu berubah dan di-update secara berkala.

### 2.6 Sistem Informasi

Menurut [14], Sistem informasi adalah seperangkat komponen terintegrasi untuk mengumpulkan, menyimpan, dan memproses data untuk menyediakan informasi, pengetahuan, dan produk digital. Perusahaan dan organisasi menggunakan sistem informasi untuk menjalankan dan mengelola operasi, berinteraksi dengan pelanggan, dan bersaing di pasar. Sebagai contoh perusahaan menggunakan sistem informasi untuk mengelola sumber daya manusia dan keuangan mereka serta menjangkau pelanggan potensial. Ada beberapa komponen utama pada sistem informasi, yaitu [14] :

- *Computer Hardware dan Software*
- *Telecommunication*
- *Database dan datawarehouse*
- *Human Resource*
- *Procedure*

### 2.7 Visual Studio Code (VS Code)

VS Code merupakan salah satu *code editor* gratis yang dikembangkan oleh *microsoft* untuk membantu dalam pembuatan sebuah program. Code Editor ini bekerja pada sistem operasi Windows, Linux, dan, macOS. Keunggulan yang dimiliki VS Code yaitu [15]:

- Dapat menyoroti kata kunci kode pemrograman dengan menampilkan warna untuk identifikasi pengkodean dengan mudah

- Mudah untuk berkolaborasi *realtime* secara jarak jauh menggunakan ekstensi yang ada pada VS Code.
- Adanya pemberian saran untuk perbaikan saat salah dalam menulis kode.
- Dapat melakukan *savepoint* dan melakukan komparasi kode terhadap *savepoint* sebelumnya atau lainnya.

## 2.8 PHP

*Hypertext Preprocessor* atau PHP merupakan bahasa pemrograman yang dipublikasikan secara umum yang biasanya digunakan untuk pengembangan website[16]. PHP merupakan *server-side programming*, yaitu jenis bahasa pemrograman yang nantinya program akan dijalankan di server. Hasil daripada pengolahan program tersebut kemudian dikirimkan ke browser/client.

Menurut [7], sistem kerja PHP dimulai dengan permintaan dari halaman web browser. Berdasarkan URL, browser akan menemukan alamat dari server web, mengidentifikasi halaman, dan mengirimkan semua informasi yang diperlukan ke server web. Browser yang mendapatkan kontennya kemudian mengonversi kode HTML dan menampilkannya. Untuk halaman yang mengandung PHP, cara kerja pemanggilannya sama dengan HTML, namun saat *request* tersebut dikirimkan, web server mengecek jenis file yang diminta user. Jika jenis file adalah PHP, maka, isi *script* akan diperiksa oleh PHP.

## 2.9 MySQL





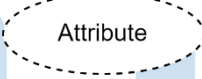
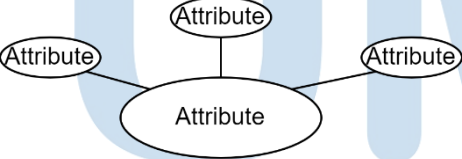

MySQL adalah salah satu database server atau sistem manajemen database *open-source* yang digunakan dalam membangun aplikasi web. Dalam fungsinya di web, MySQL dapat mengumpulkan data agar situs web dapat beroperasi dengan baik yang dimana ini dapat membantu dalam membuat atau mengubah data dengan memberikan opsi seperti menambah, menghapus, dan mengubah data di server web [18].

## 2.10 Entity Relationship Diagram (ERD)

ERD merupakan pemodelan untuk membuat struktur data dan hubungan data berbasis pada dunia nyata yaitu yang terdiri dari sekumpulan objek dan hubungan antar objek di konsepkan dalam bentuk simbol grafis diagram[19].

Adapun komponen dalam ERD antara lain [19] :

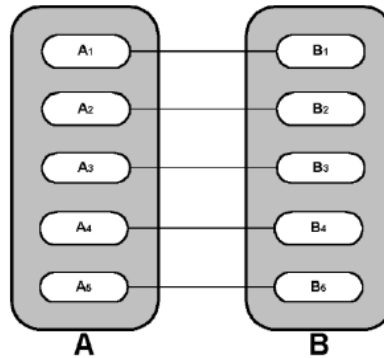
Tabel 2.10.1 Simbol ERD

Simbol	Keterangan
	Disimbolkan dengan persegi panjang yang menyatakan himpunan entitas.
	Disimbolkan dengan belah ketupat yang menyatakan himpunan relasi atau hubungan antar satu entitas dengan entitas lainnya.
	Disimbolkan dengan elipse yang menyatakan atribut
	Disimbolkan dengan <i>double elipse</i> yang menyatakan atribut <i>multi value</i> . Contohnya yaitu gender yang mempunyai dua <i>value</i> yaitu, laki-laki dan perempuan.
	Disimbolkan dengan elips putus-putus, merupakan atribut yang dihasilkan dari atribut lain atau dapat disebut sebagai <i>attribute derivative</i>
	Disimbolkan dengan elipse yang lebih kecil, yaitu turunan dari attribute biasa dimana merupakan atribut yang lebih spesifik untuk menjabarkan suatu atribut.
	Disimbolkan dengan garis yang menyatakan hubungan antara himpunan relasi dengan himpunan entitas dan atribut dengan himpunan entitas.

Dalam hubungan antar entitas, dibagi menjadi berbagai macam diantaranya[19]:

- *One to one*

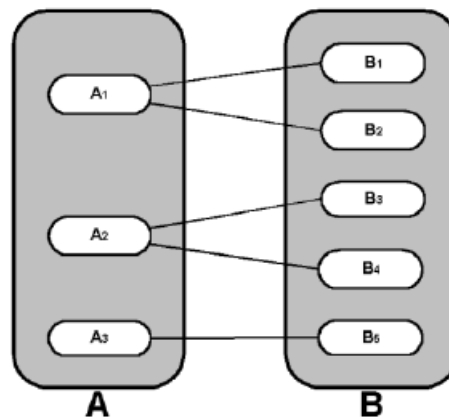
Tiap entitas berhubungan dengan entitas lainnya paling banyak satu atau sebaliknya.



Gambar 2.10.1 Relasi One to One (Sumber: [19])

- *One to Many*

Tiap entitas berhubungan pada banyak entitas, namun tidak berlaku sebaliknya.

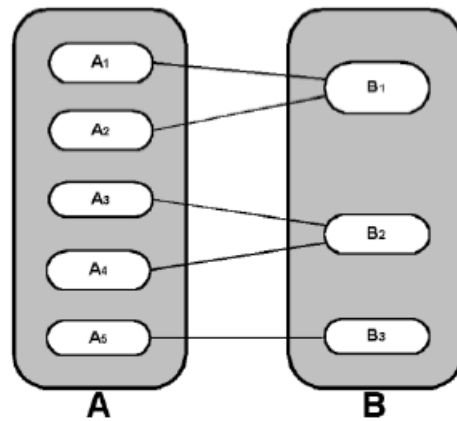


Gambar 2.10.2 Relasi one to many (Sumber: [19])

- *Many to One*

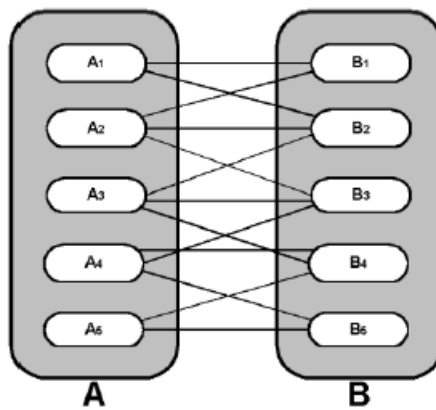
Kebalikan dari One to Many, tiap entitas yang banyak dapat berhubungan pada satu entitas, namun tidak berlaku sebaliknya.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 2.10.3 Relasi Many to One (Sumber: [19])

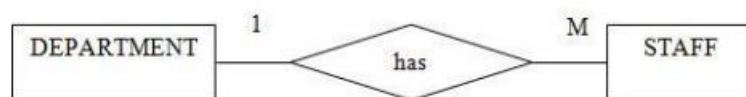
- *Many to Many*  
Banyak entitas dapat berhubungan pada banyak entitas lainnya, begitu pula sebaliknya.



Gambar 2.10.4 Relasi Many to Many (Sumber: [19])

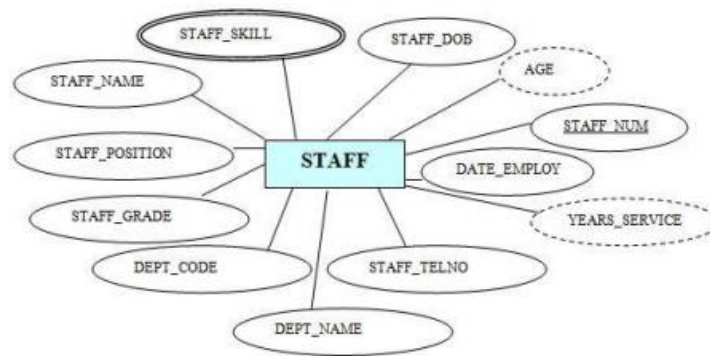
Beberapa contoh dari penggunaan ERD adalah sebagai berikut[20]:

- Pada setiap departement memiliki banyak staff dan tiap staff hanya dimiliki oleh satu departement.



Gambar 2.10.5 Relasi Antar Entitas (Sumber: [20])

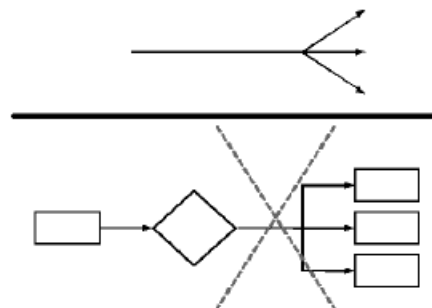
- Dalam satu entitas staff, memiliki banyak atribut seperti nama, umur, nomor telepon, nomor id staff, skill staff, nama tempat departemen staff bekerja, kode departemen, posisi, nilai staff, berapa lama staff bekerja, dan tanggal memulai kerja.



Gambar 2.10.6 Contoh Entitas yang memiliki atribut (Sumber: [20])

Atribut skill staff merupakan *multivalue attribute* dikarenakan pada satu staff dapat memiliki lebih dari satu skill. Kemudian pada atribut umur merupakan atribut *derivative* karena nilai dari atribut umur dapat dihasilkan dari attribute tanggal lahir. Sama halnya dengan attribute berapa lama staff bekerja yang merupakan attribute *derivative*, yaitu attribute yang dapat dihasilkan dari attribute tanggal memulai kerja.

Dalam pembuatan ERD, berdasarkan [19] ada pula hal yang dilarang, yaitu dari dan ke relasi ke entitas ataupun sebaliknya memiliki percabangan. Karena hal tersebut menimbulkan kerancuan pembacaan data.



Gambar 2.10.7 Pembuatan ERD yang tidak diperbolehkan (Sumber: [19])

## 2.11 Unified Modelling Language (UML)

UML merupakan metode pemodelan visual dan teknik diagram yang digunakan untuk perancangan sistem hingga implementasi yang didasarkan pada pendekatan pemodelan berorientasi objek. Dalam UML, terdapat beberapa jenis diagram, diantaranya[21]:

### 2.11.1 Use Case Diagram

Use Case adalah metodologi yang digunakan dalam analisa sistem untuk mengidentifikasi, mengklarifikasi, dan mengatur kebutuhan sistem. Selain itu, diagram use case menggambarkan berbagai jenis pengguna yang akan berinteraksi pada sistem tersebut. Use Case Diagram memiliki empat objek, yaitu:

- *Actor*: Setiap entitas (orang, organisasi, sistem) yang menjalankan peran dalam satu sistem tertentu.



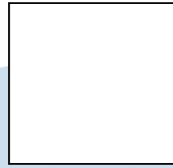
Gambar 2.11.1 Simbol *Actor*

- *Use Case*: Representasi visual dari fungsionalitas bisnis dalam suatu sistem.



Gambar 2.11.2 Simbol *Use Case*

- *System*: Direpresentasikan berbentuk persegi panjang yang mencakup semua kasus penggunaan dalam sistem yang mendefinisikan ruang lingkup sistem.



Gambar 2.11.3 Simbol *System*

- *Package*: Untuk mengelompokkan kelas atau *interfaces* yang terkait.



Gambar 2.11.4 Simbol *Package*

### 2.11.2 Activity Diagram

*Activity diagram* menjelaskan aliran kontrol dalam suatu sistem yang terdiri dari aktivitas dan berbagai link untuk memodelkan perilaku dalam proses bisnis. Jadi, *activity diagram* merupakan operasi sistem yang ditunjukkan dalam diagram untuk menunjukkan semua aliran kontrol dari satu operasi ke operasi lainnya.

Secara garis besar, terdapat beberapa element pada *activity diagram*, diantaranya adalah:

- *Action and Activities*

Menggambarkan aktivitas manual ataupun yang terkomputerisasi. Dinotasikan sebagai persegi panjang dan memiliki nama kata kerja

- *Object Nodes*

Diwakili sebagai aliran informasi dari aktivitas satu ke aktivitas lainnya.





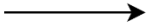
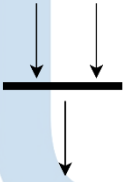
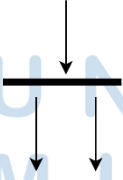
- *Control Flow and Object Flow*

Dalam jenis aliran di *diagram activity* terdapat dua macam, yaitu *control flow* dan *object flow*. Dimana *control flow* merupakan urutan aliran aktivitas sedangkan *object flow* menunjukkan aliran objek dari satu aktivitas ke aktivitas lainnya.

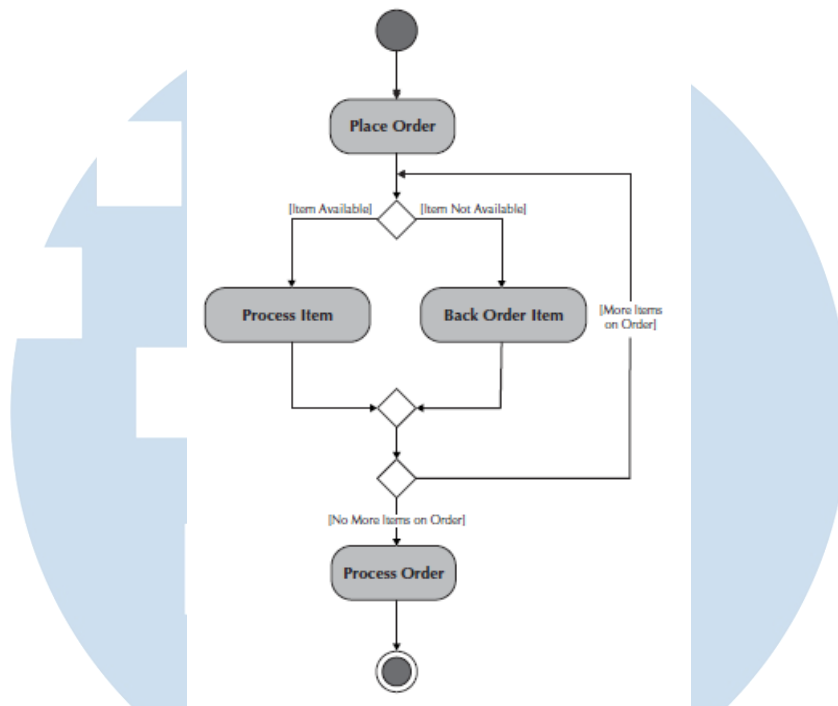


Adapun beberapa notasi dan fungsi dari activity diagram adalah sebagai berikut:

Tabel 2.11.1 Activity Diagram

Simbol	Nama	Fungsi
	<i>Start Node</i>	Notasi awal sebelum <i>activity</i> berlangsung
	<i>Final Activity Node</i>	Notasi berakhirnya rangkaian aktivitas
	<i>Activity</i>	Notasi yang digunakan untuk mewakili serangkaian tindakan aktivitas
	<i>Decision</i>	Notasi pengujian untuk memastikan ke arah mana lagi aliran aktivitas tertuju.
	<i>Control Flow</i>	Notasi yang merepresentasikan arah aliran aktivitas
	<i>Join Node</i>	Notasi untuk menyatukan kembali serangkaian aktivitas paralel
	<i>Fork Node</i>	Notasi untuk memisahkan aliran aktivitas yang tunggal menjadi aktivitas paralel.

Contoh penggunaan *activity diagram* yaitu sebagai berikut:

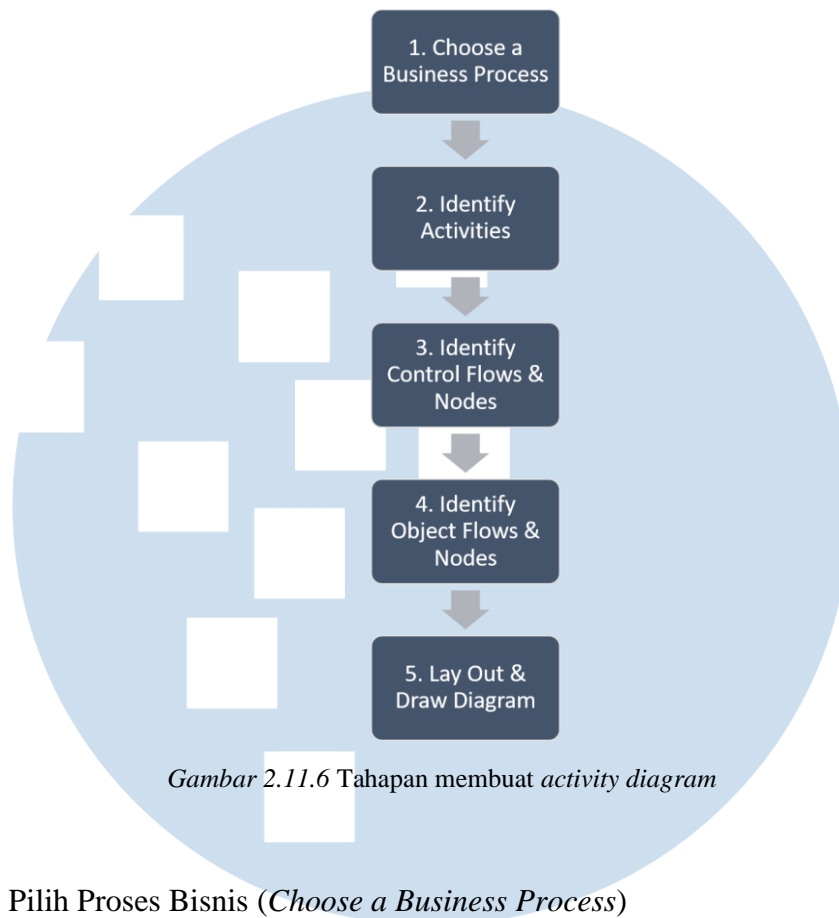


Gambar 2.11.5 Contoh Activity Diagram (Sumber: [21])

Dalam contoh *activity diagram* diatas merupakan aktivitas melakukan pemesanan barang. Dimulai dari *start node*, kemudian melakukan aktivitas pemesanan. Saat melakukan pemesanan didapatkan *decision*, yaitu apakah item yang ingin dipesan tersedia atau tidak. Dalam *decision* ini merupakan *fork node*, yaitu adanya pemisahan suatu aktivitas menjadi paralel. Jika tersedia maka barang akan diproses. Namun jika tidak, maka item akan di *back order*. Kedua aktivitas tersebut kemudian disatukan dalam *join node* dan dihadapi *decision* apakah masih ada barang yang ingin dipesan. Jika iya, maka akan *flow* akan kembali ke aktivitas awal, namun jika tidak maka semua pesanan akan langsung diproses. Lalu *activity diagram* tersebut diakhiri dengan *final activity node*.

Dalam pembuatan *activity diagram* ada 5 tahap yang harus dilakukan:

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 2.11.6 Tahapan membuat activity diagram

1. Pilih Proses Bisnis (*Choose a Business Process*)

Dalam memilih proses bisnis, dibutuhkan peninjauan requirement dan merujuk ke *use case diagram* yang sudah dibuat.

2. Mengidentifikasi kegiatan (*Identify Activities*)

Pada proses bisnis diidentifikasi keseluruhan aktivitas yang diperlukan.

3. Mengidentifikasi aliran dan node kontrol (*Identify Control Flows & Nodes*)

Pada tahap ini, diidentifikasi mengenai aliran kontrol dan node yang diperlukan dalam pendokumentasian alur logika dari proses bisnis.

4. Mengidentifikasi aliran dan node objek (*Identify Object Flows & Nodes*)

Pada tahap ini, diidentifikasi mengenai aliran objek dan node yang diperlukan dalam pendokumentasian alur logika dari proses bisnis.

## 5. Tata letak dan menggambar diagram (*Lay Out & Draw Diagram*)

Ketika menggambar diagram, diminimalkan untuk menggambar dengan garis yang saling silang karena akan menyebabkan kebiasan atau kerancuan dalam aliran aktivitas.

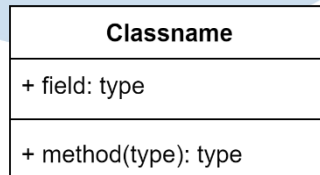
### 2.11.3 Class Diagram

Class Diagram merupakan model statis yang menggambarkan struktur dari sistem dengan menunjukkan keberadaan kelas dan hubungan antar kelas. *Class diagram* merupakan model yang statis dimana ditunjukkan hubungan antar kelas pada sistem.

Dalam *class diagram*, terdapat elemen- element didalamnya, yaitu:

#### ➤ Class

Class merupakan bagian utama dalam *class diagram*, yang bertujuan untuk melakukan penyimpanan dan pengelolaan informasi yang ada pada sistem.

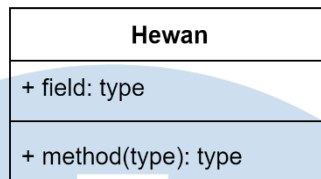


Gambar 2.11.7 Struktur Class pada Class Diagram

Pada class terdiri dari 3 bagian, yaitu di paling atas yaitu bagian “*classname*”, pada bagian kedua yaitu “*attribute/field*”, dan yang terakhir merupakan “*method/operation*”.

- *Classname* merupakan representasi dari suatu hal seperti orang, tempat, hewan, dan sebagainya.

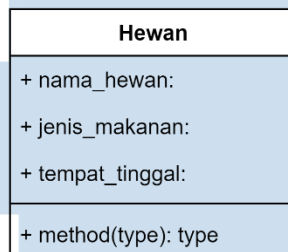
Contoh: *classname* bernama hewan



Gambar 2.11.8 Class Diagram dengan classname Hewan

- *Attribute/field* merupakan nilai dari tiap *classname* yang berguna untuk mengidentifikasi setiap *instance* pada *classname*.

Contoh: pada hewan terdapat attribute nama hewan, jenis makanan, dan tempat tinggal.



Gambar 2.11.9 Class diagram dengan attribute

- *Method/operation* merupakan representasi dari tindakan atau hal yang dapat dilakukan oleh *class*

Contoh: hewan dapat makan/minum



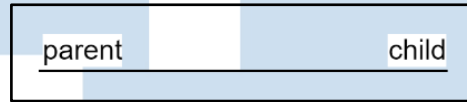
Gambar 2.11.10 class diagram dengan method

### ➤ Relationship

Relasi merupakan hubungan antar *class*. Ada berbagai macam jenis relasi antar *class*, yaitu:

- Asosiasi

Merupakan hubungan antar class yang dimana suatu class dapat menggunakan class lainnya.



Gambar 2.11.11 Simbol Relasi Asosiasi

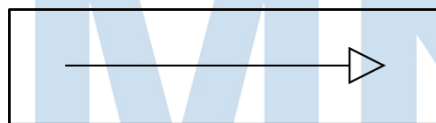
Contoh hubungan asosiasi yaitu antara pilot dengan *aircraft*, dimana pilot “menerbangkan” *aircraft*. “Menerbangkan” merupakan penghubung antara *class* pilot dengan *class aircraft* dan itu merupakan asosiasi.



Gambar 2.11.12 Contoh hubungan relasi asosiasi (Sumber: [21] )

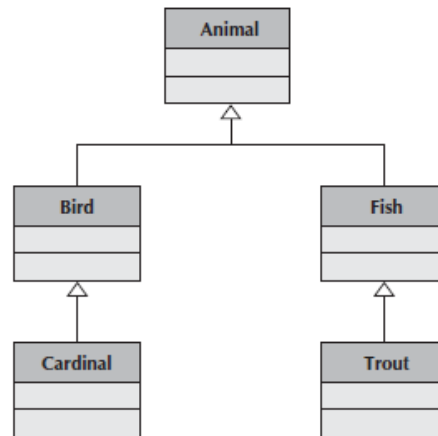
- Generalisasi / inheritance

Merupakan hubungan antara beberapa kelas dimana dalam suatu kelas mewariskan kelas lainnya.



Gambar 2.11.13 Simbol Relasi Generalisasi

Contoh hubungannya yaitu pada class hewan. Pada gambar dibawah, burung dan ikan merupakan satu spesies atau dapat digeneralisasikan sebagai hewan. Begitu pula dengan kardinal dan trout, yang dapat digeneralisasikan sebagai burung dan ikan.



Gambar 2.11.14 Contoh Hubungan generalisasi (Sumber: [21])

- Agregasi

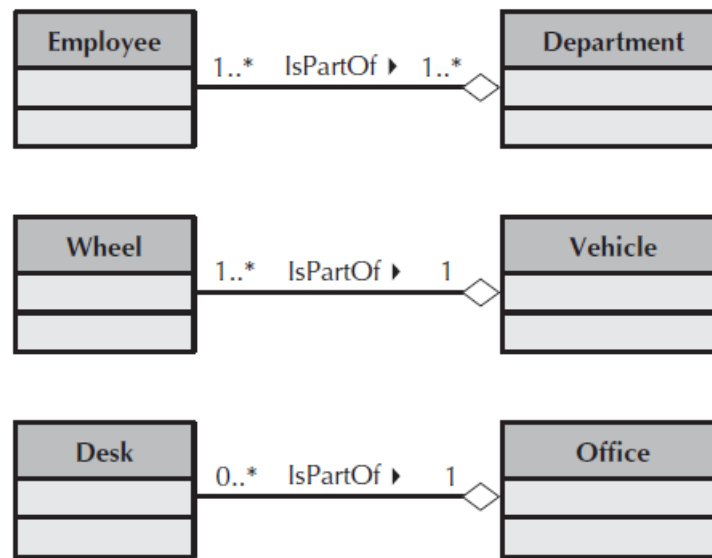
Merupakan hubungan antara *class* yang dimana satu *class* merupakan bagian dari *class* lainnya.



Gambar 2.11.15 Simbol Relasi Agregasi

Contoh hubungan dari relasi agregasi yaitu antara kendaraan dan roda. Roda merupakan bagian dari kendaraan, sedangkan kendaraan memiliki komponen yaitu roda. Selain itu hubungan antara meja dengan kantor. Meja merupakan bagian properti dari kantor, sedangkan kantor memiliki properti meja. Dan yang terakhir yaitu hubungan antara karyawan dengan departemen. Di perusahaan, karyawan merupakan bagian pada suatu departemen, sedangkan departemen mempunyai karyawan.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



Gambar 2.11.16 Hubungan agregasi (Sumber: [21])

- Komposit

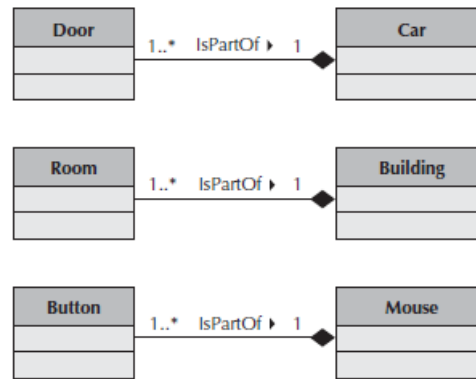
Merupakan hubungan antar *class* yang dimana suatu kelas memiliki ketergantungan terhadap kelas lainnya. Jika satu kelas yang berhubungan dengan kelas lainnya dihapus/tidak ada, maka kelas lainnya juga tidak ada namun tidak berlaku sebaliknya.



Gambar 2.11.17 Simbol Relasi Komposit

Contoh hubungan dari relasi komposit yaitu pada hubungan antara *class* pintu(mobil) dengan *class* mobil. Jika mobil tidak ada, maka pintu mobil pun juga tidak ada, namun sebaliknya jika pintu tidak ada, entitas *class* mobil akan tetap ada. Hubungan lainnya yaitu ruangan dengan bangunan. Jika tidak ada bangunan, maka tidak ada pula ruangan, namun sebaliknya jika ruangan tidak ada, maka *class* gedung akan tetap ada. Dan hubungan antara tombol (pada mouse) dengan mouse. Jika mouse tidak ada, maka button pun tidak ada.

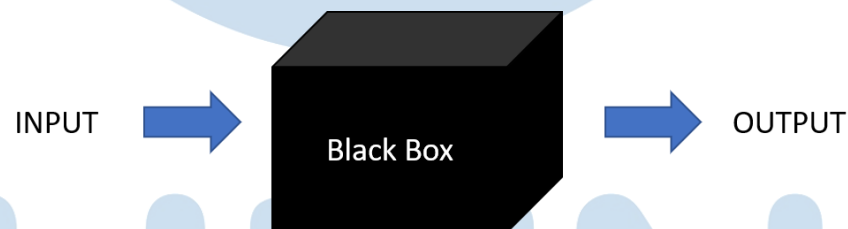




Gambar 2.11.18 Contoh hubungan relasi komposit (Sumber: [21])

## 2.12 User Acceptance Test: Black Box Testing

*Black Box Testing* atau pengujian fungsionalitas merupakan teknik pengujian fungsionalitas software yang merancang kasus uji berdasarkan informasi dan spesifikasi dengan mengharapkan suatu output yang diinginkan dari setiap nilai input yang dimasukkan.



Gambar 2.12.1 *Black Box Testing*

Adapun kelebihan dari *black box testing* diantaranya [22]:

- Pengujian dilakukan pada syarat – syarat dari sudut pandang user pelanggan.
- Dilakukan oleh pihak ketiga untuk menghindari ambiguitas atau inkonsistensi pada spesifikasi persyaratan developer.
- Karena pengujian ini tidak memerlukan keterampilan dalam pemrograman, penguji tidak harus menjadi seseorang yang ahli programming dan sebagainya.
- Pengujian yang efisien saat digunakan pada sistem yang besar.

Adapun kelemahan dari *black box testing*, diantaranya [22]:

- Dapat menimbulkan pengujian yang redundan jika developer sudah melakukan pengetesan terhadap suatu fungsionalitas atau kasus uji.
- Merancang kasus uji bisa sulit jika persyaratannya tidak jelas dan ringkas.
- Tidak efisien jika menguji kode yang kompleks.
- Hasil tes yang sering dilebih-lebihkan.

Contoh dari testing black box adalah pada gambar 2.12.2, dimana gambar tersebut merupakan pengecekan fungsionalitas pada aplikasi pembayaran sekolah. Dari tes tersebut menghasilkan semua fitur berhasil dilakukan pengetesan dan menghasilkan hasil yang positif.

Admin and User Testing Scenarios			
Feature Test	Testing Details	Types of Testing	Results Testing
Login	Verify login data	Black Blox	✓
Processing payment data	Input student identification number, Pay, Print Data, Delete data	Black Blox	✓
Processing department data	Add data, Save data, Edit data, Delete data	Black Blox	✓
Manage student data	Add data, Save data, Edit data, Delete data	Black Blox	✓
Manage class data	Add data, Save data, Edit data, Delete data	Black Blox	✓
Processing user data	Add data, Save data, Edit data, Delete data	Black Blox	✓
Manage lesson data	Add data, Save data, Edit data, Delete data	Black Blox	✓

Gambar 2.12.2 Contoh Pengujian Black Box



## 2.13 Penelitian Terdahulu

Tabel 2.13.1 Penelitian Terdahulu

Nama Penulis	Judul Penelitian	Tahun	Metodologi	Hasil
Imam Subhan	Pembangunan Sistem Informasi Perencanaan Produksi Menggunakan Metode Moving Average Berbasis Web (Studi Kasus: PT. Mutiara Mandiri Jaya)	2018	<b>Pengumpulan data:</b> <ul style="list-style-type: none"> <li>- Studi Lapangan</li> <li>- Studi Pustaka</li> </ul> <b>SDLC:</b> Extreme Proramming (XP) <b>Desain Sistem:</b> <ul style="list-style-type: none"> <li>- Use Case Diagram</li> <li>- Activity Diagram</li> <li>- Sequence Diagram</li> <li>- Class Diagram</li> </ul>	Peneliti membuat website dengan memberikan informasi mengenai jumlah kebutuhan bahan baku atau laporan <i>Material Production Planning (MRP)</i> dan kapan bahan baku tersebut dibutuhkan atau laporan <i>Master Production Schedule (MPS)</i> . Selain itu dalam website peneliti juga dapat didokumentasikan Laporan penjualan, laporan perkiraan, laporan pengadaan bahan baku
Achmad Sidik, Edy Tekat Bronto Waluyo, Siti Susilawati	Perancangan Sistem Informasi Manajemen Produksi di PT Aneka Paperindo Sejahtera	2018	Menggunakan use case diagram dan menggunakan Bahasa pemrograman PHP dengan server MySQL	<ol style="list-style-type: none"> <li>1. Karena persediaan bahan baku tidak terpantau, maka faktor yang mempengaruhi pengendalian produksi adalah proses permintaan bahan baku yang lama.</li> <li>2. Dalam sistem kontrol produksi ini, hanya bagian tertentu yang memasukkan data yang digunakan dalam kontrol produksi.</li> <li>3. Sistem kendali</li> </ol>

Nama Penulis	Judul Penelitian	Tahun	Metodologi	Hasil
				<p>produksi ini dapat memonitor status dari setiap proses didalamnya. Sistem dapat memantau tingkat persediaan dan jadwal produksi berdasarkan pesanan.</p> <p>4. Hasil dari sistem pengendalian produksi tersebut dapat digunakan sebagai bahan pelaporan persediaan bahan baku dan rencana produksi yang ada.</p>
Sri Wahyuningsih	Perancangan Sistem Informasi Perencanaan Produksi PT Katolec Indonesia Berbasis Web	2018	<p><b>Pengumpulan data:</b></p> <ol style="list-style-type: none"> <li>Melakukan survey perencanaan produksi terkait dengan masalah yang sedang dikerjakan.</li> <li>Pengamatan proses manufaktur PT. Katolec Indonesia</li> <li>Wawancara pada pihak yang terkait di line dan PPC</li> </ol> <p><b>SDLC:</b> <i>Waterfall</i></p> <p><b>Bahasa</b></p>	<ol style="list-style-type: none"> <li>Pembuatan sistem dilakukan di area perusahaan dan informasi yang ditampilkan hanya terkait dengan rencana produksi dan bahan yang diperlukan.</li> <li>Perancangan sistem informasi berupa data kebutuhan material dari PPC hingga produksi.</li> </ol>

Nama Penulis	Judul Penelitian	Tahun	Metodologi	Hasil
			<b>Pemrograman:</b> <i>PHP</i> <b>IDE:</b> <i>Dreamweaver CS6</i> <b>Database:</b> <i>Mysql</i>	
Wahyuni Yahyan, Muhammad Ilham A Siregar	Pemilihan Pupuk Pada Tamanam Padi Berbasis Web Untuk Meningkatkan Hasil Panen Dengan Menggunakan Metode Analitical Hierarcy Proses	2020	Model sistem pendukung keputusan yang digunakan menggunakan <i>Analitical Hierarcy Process (AHP)</i>	Adapun hasil dari penelitian ini yaitu sistem pendukung keputusan yang menunjukkan kisaran nilai untuk setiap jenis pupuk, sehingga memudahkan petani untuk memilih pupuk yang tepat untuk benih atau bibit padi terbaik berdasarkan lokasi persawahan di kabupaten Solok.
Siti Fathuroh, Titis Wulandari, Tri Wera Agrita	Implementasi Metode Moora Dalam Pengambilan Keputusan Penentuan Lokasi Terbaik Untuk Minyak Sawit Berkualitas Pada KUD Karya Mukti Berbasis Web	2020	<b>DSS: Multi-Objective Optimization on The Basic of Ratio Analysis (MOORA)</b>  <b>Pengumpulan data:</b> <ul style="list-style-type: none"> <li>- Observasi</li> <li>- Wawancara</li> </ul>	Metode MOORA pada peneliitan ini mengolah data tanah dan menghitung nilai optimum dari kriteria evaluasi masing-masing data tanah. Hasil pengujian mencapai nilai tertinggi yaitu 7,56

Diantara penelitian terdahulu, diambil metode SDLC *extreme programming* yang sama dari penelitian 1, yang membedakan yaitu di penelitian 1, objek penelitian yaitu bergerak di bidang industri minuman. Sedangkan untuk penelitian 2 dan 3 tidak menggunakan *extreme programming*, dan di penelitian 3 menggunakan *waterfall*. Bahasa pemrograman yang digunakan sama dengan penelitian lainnya yaitu *html*, *css*, dan *php*. Untuk strategi perencanaan produksi dari para penelitian terdahulu itu menggunakan *make to order* sedangkan pada objek penelitian penulis yaitu *make to stock*.

Untuk penelitian 4 diambil judul yaitu pemilihan jenis pupuk. Yang membedakan antara pemilihan jenis pupuk penelitian 4 dengan penelitian ini yaitu model pengembangan sistem pendukung keputusan yang dibuat menggunakan model AHP. Pada penelitian 5 diambil metode pengambilan keputusan yaitu MOORA. Yang membedakan antara penelitian 5 dengan penelitian ini yaitu pada tujuan penelitiannya, dimana MOORA pada penelitian tersebut digunakan untuk menentukan lokasi terbaik untuk minyak sawit.

