BAB II

TINJAUAN PUSTAKA

2.1 Landasan Teori

Dalam sub bab ini akan dibahas tentang teori-teori yang berhubungan dengan Aplikasi optimalisasi produksi dan keuntungan pada pabrik garment handuk baik mengenai teori optimasi, sistem pendukung keputusan, database dan bahasa pemrograman pembangun aplikasi.

2.2 Teknik Optimasi

Teknik optimasi sangat aplikatif pada permasalahan-permaalahan yang menyangkut pengoptimalan, baik itu maksimasi atau minimasi. Teknik optimasi merupakan suatu teknik pengalokasian sumber daya, baik bahan baku, waktu, tenaga kerja maupun uang, tergantung dari kondisi yang diinginkan. Dengan menggunakan teknik ini, maka sumber daya terbatas yang dimiliki dapat terproses dengan baik dan mendapatkan hasil yang maksimal.

Terdapat banyak jenis teknik optimasi yang dapat digunakan mengikuti proyek yang akan di optimasi yaitu optimasi linier atau yang sering disebut program linier, kemudian optimasi linier dengan variabelnya bilangan Integer atau bilangan bulat atu yang sering disebut dengan *Integer Linier Programming*, ada juga optimasi linier yang variabelnya bersifat ya/tidak atau disimbolkan 1 jika ya, 0 jika tidak yang dikenal dengan *Binary Integer Linier Program.*

2.2.1 Linier Programming

Program linier (*Linier Programming* yang disingkat LP) merupakan metode matematik dalam mengalokasikan sumber daya yang langka atau terbatas untuk mencapai tujuan tunggal seperti memaksimumkan keuntungan dan meminimumkan biaya.

2.2.1.1 Formulasi Model LP

Terdapat tiga tahapan formulasi matematik dari model LP yaitu :

Contoh permasalahan :

Tabel 2.1 Contoh tabel masalah kombinasi produk handuk

Jenis Produk	**Kebutuhan Sumber Daya**		**Harga (Rp...ribu/Unit)**
	Waktu Mesin (jam / unit)	**Benang (Kg / unit)**	
Handuk 1	2	6	7
Handuk 2	3	5	6
Batasan	≤ 12	≤ 30	

1. Variabel keputusan

   Menentukan variabel yang tak diketahui (variabel keputusan) dan dinyatakan dalam simbol matematik.

   Contoh : X_1 = Jumlah handuk 1

   X_2 = Jumlah handuk 2

2. Fungsi tujuan

   Membentuk suatu fungsi tujuan yang ditunjukkan sebagai satu hubungan linier (bukan perkalian) dari variabel keputusan.

   Contoh : Maksimalkan $Z = 7X_1 + 6X_2$

3. Sistem kendala (*Constraint*)

   Menentukan semua kendala masalah tersebut (fungsi kendala) dan mengekspresikan dalam persamaan atau pertidaksamaan. Dalam masalah ini kendalanya adalah waktu mesin dan bahan baku yang terbatas. Masing-masing handuk membutuhkan baik mesin maupun bahan baku.

   Bagi handuk 1, benang yang dibutuhkan untuk menghasilkan tiap unit adalah 6 Kg/unit, sehingga untuk handuk 1 dibutuhkan benang sebanyak X_1 Kg. Dengan cara serupa, handuk 2 membutuhkan 5 Kg/unit, sehingga benang yang dibutuhkan untuk handuk 2 adalah X_2 Kg. Jumlah benang yang tersedia adalah 30Kg. Sehingga kendala benang dituliskan : $6X_1 + 5X_2 \leq 30$

   Kendala waktu mesin dirumuskan dengan cara yang sama. Handuk 1 membutuhkan 2 jam/unit, handuk 2 membutuhkan 3 jam/unit. Jumlah waktu yang tersedia adalah 12 jam, maka kendala ini dituliskan :

   $2X_1 + 3X_2 \leq 12$.

Kendala yang didapat berdasarkan pernyataan diatas adalah :

Fungsi kendala : $2X_1 + 3X_2 \leq 12.$

$$6X_1 + 5X_2 \leq 30$$

2.2.2 Metode Simpleks

Penentuan solusi optimal menggunakan simpleks didasarkan pada teknik eliminasi Gaus Jordan. Dalam metode simpleks, diadakan pengubahan pertidaksamaan menjadi persamaan dengan cara menambahkan "*slack variable*" untuk pertidaksamaan yang mengandung tanda ≤ dan mengurangkan variable surplus untuk pertidaksamaan yang mengandung tanda ≥. Untuk menyelesaikan masalah LP menggunakan metode simplekas, model LP harus diubah ke dalam suatu bentuk umum yang dinamakan " bentuk baku" (standard form). Keunggulan metode ini adalah bahwa ia dapat menyelesaikan masalah LP dengan berapapun jumlah variabel.

Adapun langkah- langkah penyelesaian dari menggunakan metode simpleks adalah sebagai berikut :

1. Langkah pertama

   Periksa apakah tabel layak atau tidak. Kelayakan tabel simpleks dilihat dari solusi (nilai kanan) Jika solusi ada yang bernilai negatif, maka tabel tidak layak. Tabel yang tidak layak tidak dapat diteruskan. Jika sebaliknya, maka langkah-langkah yang dilakukan berikutnya adalah :

a. Fungsi tujuan diubah menjadi fungsi implisit, yaitu diubah menjadi persamaan nol. Fungsi kendala diubah dari pertidaksamaan menjadi persamaan dengan menambahkan "*slack variable*". Variabel slack adalah variabel yang ditambahkan ke model matematik kendala untuk mengkonversikan pertidaksamaan $\leq$ menjadi persamaan (=). Penambahan variabel ini terjadi pada tahap inisialisasi. Pada solusi awal, variabel slack akan berfungsi sebagai variabel basis dan menyatakan jumlah sumber daya yang tak digunakan di akhir perhitungan.

b. Maka susunan formulasi menjadi :

Fungsi tujuan : $Z - 7X_1 - 6X_2 + S_1 + S_2 = 0$

Fungsi kendala : $2X_1 + 3X_2 + S_1 \leq 12.$

$6X_1 + 5X_2 + S_2 \leq 30$

$X_1, X_2, S_1, S_2 \geq 0$

2. Langkah pembentukan tabel

Dari susunan formulasi baru, kemudian susun dalam bnetuk tabel sbb:

2.2 Tabel persamaan awal

Z	**X_1**	**X_2**	**S_1**	**S_2**	**Solusi**
Z	-7	-6	0	0	0
S_1	2	3	1	0	12
S_2	6	5	0	1	30

a. Kolom variabel dasar memuat variabel z dan varabel tambahan.

b. Kolom z memuat data koefisien z dan koefisien variabel tambahan.

c. Kolom X_1, X_1,....., X_n memuat data koefisien yang bersesuaian dengan variable

d. Kolom solusi (kolom K) memuat data sebelah kanan persamaan dari fungsi kendala.

Untuk mendapatkan jawaban optimal, maka dilakukan proses iteratif dengan cara berikut :

3. Langkah menentukan kolom kunci (Kolom pivot)

Kolom kunci adalah suatu kolom yang mempunyai nilai pada baris fungsi tujuan yang bertanda negatif dan harga mutlak terbesar. Penentuan kolom kunci dilihat dari koefisien fungsi tujuan dan tergantung dari bentuk tujuan. Jika tujuan berupa maksimisasi, maka kolom pivot adalah kolom dengan koefisien negatif terbesar. Jika tujuan minimisasi, maka kolom pivot adalah kolom dengan koefisien terkecil. Jika nilai negatif terbesar atau positif terbesar lebih dari dua, pilih salah satu secara sembarang.

Lakukan iterasi sehingga jawaban optimal didapat . Untuk masalah maksimisasi jawaban optimal didapat jika tanda negatif pada suatu tabel pada baris fungsi tujuan sudah tidak ada lagi

Tabel 2.3 Simpleks iterasi pertama

Z	**X_1**	**X_2**	**S_1**	**S_2**	**Solusi**	**Rasio**
Z	**-7**	-6	0	0	0	
S_1	2	3	1	0	12 / 2	6
S_2	**(6)**	5	0	1	30 / 6	5

4. Langkah menentukan baris kunci (Baris pivot)

Baris kunci adalah suatu baris dimana rasio antara element pada kolom K dengan kolom kunci adalah terkecil.

S_1	6	5	0	1	30 / 6	5

5. Langkah mengubah element pada baris kunci

Semua element pada baris kunci dibagi dengan elemen kunci, dengan menerapkan metode Gauss Jordan jenis 1 :

$$\text{Elemen persamaan baru} = \frac{\text{Elemen persamaan pivot tabel lama}}{\text{Elemen pivot}}$$

$S_2 =$ (6 5 0 1 30) /6

$X_{1 =}$ (1 5/6 0 1/6 5)

6. Langkah mengubah element pada baris lainnya

Selain mengubah elemen pada baris kunci, elemen pada baris lain juga diubah dengan menerapkan metode Gauss Jordan jenis 2 :

Elemen baris baru = Elemen baris lama – (Elemen pada kolom kuncinya X Element baris kunci baru)

Setelah menerapkan metode Gauss Jordan diatas, sehingga dihasilkan tabel sebagai berikut :

Tabel 2.4 Simpleks iterasi kedua

Z	**X_1**	**X_2**	**S_1**	**S_2**	**Solusi**	**Rasio**
Z	0	**-1/6**	0	7/6	35	
S_1	0	(8/6)	1	-2/6	2 / (8/6) = 12/8	1.5
X_1	1	5/6	0	1/6	5 / (5/6) = 30/5	6

Kemudian didapat kembali baris kunci :

S_1	0	(8/6)	1	-2/6	2 / (8/6) = 12/8	1.5

Lakukan lagi langkah 5 untuk mengubah elemen pada baris kunci menggunakan persamaan Gauss Jordan jenis 1 :

S_1 = (0 8/6 1 -2/6 2) / (8/6)

X_2 = (0 1 8/6 -12/48 1.5)

Lakukan langkah 6 untuk mengubah elemen pada baris lainnya dengan menggunakan metode Gauss Jordan jenis 2, sehingga didapat tabel sebagai berikut :

Tabel 2.5 Simpleks optimum

Z	**X_1**	**X_2**	**S_1**	**S_2**	**Solusi**
Z	0	0	8/36	1.125	35.25
X_2	0	1	8/6	-0.25	1.5
X_1	1	0	-40/36	108/288	3.75

Solusi baru memberikan X_1 = 3.75 dan X_2 = 1.5 dan nilai Z = 35.25. Tabel diatas adalah optimal karena tak ada variabel nonbasis yang memiliki koefisien negatif pada persamaan Z. Berdasarkan kasus produksi handuk diatas, perusahaan tidak dapat memproduksi handuk dalam bentuk pecahan. Oleh karena itu permasalahan diatas diselesaikan menggunakan pemrograman bulat / *Integer linier Programming* menggunakan metode *Branch and Bound* untuk menghasilkan nilai bulat.

2.2.3 *Integer Linier Programming (ILP)* Metode *Barnch and Bound*

Pemrograman integer pada intinya adalah perhitungan program linier dimana semua variabel memiliki nilai-nilai *integer* (bulat). Untuk menyelesaikan permasalahan ILP, digunakanlah Metode Branch and Bound karena metode ini lebih efisien dibanding metode perhitungan nilai bulat lainnya dan telah menjadi kode komputer standar untuk ILP.

2.2.3.1 Penyelesaian dengan Menggunakan Metode Branch and Bound

Langkah – langkah metode Branch and Bound untuk masalah maksimisasi dapat dijelaskan sebagai berikut :

1. Selesaikan masalah LP dengan metode simpleks biasa tanpa pembatasan bilangan bulat.

2. Teliti solusi optimumnya. Jika variabel basis yang diharapkan bulta adalah bulat, solusi optimum bulat telah tercapai. Jika satu atau lebih variabel basis yang diharapkan bulat ternyata tidfak bulat, lanjutkan ke langkah 3.

3. Nilai solusi pecah yang layak dicabangkan ke dalam sub-sub masalah. Tujuannya adalah untuk menghilangkan solusi kontinu yang tidak memenuhi persyaratan bulat dari masalah itu. Pencabangan itu dilakukan melalui kendala-kendala mutually exclusive yang perlu untuk memenuhi persyratan bulat dengan jaminan tak ada solusi bulat layak yang tak diikutsertakan.

4. Untuk setiap submasalah, nilai solusi optimum kontinu fungsi tujuan ditetapkan sebagai batas atas. Solusi bulat terbaik menjadi batas bawah. Sub – sub masalah yang memiliki batas atas kurang dari batas bawah yang ada tak diikutsertakan pada analisis selanjutnya. Suatu solusi bulat layak adalah sama baik atau lebih baik dari batas atas untuk setiap submasalah yang dicari. Jika solusi demikian ada, suatu submasalah dengan batas atas terbaik dipilih untuk dicabangkan. Kembali kelangkah 3.

Berdasarkan kasus diatas, karena $X_1 = 3.75$ dan $X_2 = 1.5$ bukan bilangan bulat, maka solusi ini tidak valid, nilai keuntungan Z = 35.25 dijadikan batas atas awal. Dalam metode Branch and Bound, masalah tersebut dibagi ke dalam dua bagian untuk mencari nilai solusi bulat yang mungkin bagi X_1 dan X_2. Untuk melakukan ini, variabel dengan nilai solusi pecah yang memiliki bagian pecah terbesar dipilih.

Untuk menghilangkan bagian pecah dari nilai $X_1 = 3.75$, dua kendala baru diciptakan. Dalam hal ini, dua nilai bulat terdekat terhadap 3.75 didapat. Sehingga diperoleh dua masalah baru melalui dua kendala *mutually exclusive*, $X_1 \leq 3$ dan $X_2 \geq 4$ yang kemudian diuraikan sebagai berikut :

Iterasi 1

Permasalahan diatas kemudian dibagi menjadi 2 sub problem, A dan B. kita dapat melakukan pencabangan (branch) pada hasil dengan variabel tidak bulat (integer)

Bagian A

Maksimisasi : $7\ X_1 + 6X_2$

Ditujukan pada : $2\ X_1 + 3\ X_2 \leq 12$

$6\ X_1 + 5\ X_2 \leq 30$

$X_1 \geq 4$

Bagian B

Maksimisasi : $7\ X_1 + 6\ X_2$

Ditujukan pada : $2\ X_1 + 3\ X_2 \leq 12$

$6\ X_1 + 5\ X_2 \leq 30$

$X_1 \leq 3$

Dengan metode LP sederhana didapatkan solusi :

Solusi optimal subproblem A: $X_1 = 4$, $X_2 = 1.2$, profit = 35.2

Solusi optimal subproblem B: $X_1 = 3$, $X_2 = 2$, profit = 33.0

Iterasi 2

Sub problem A kita cabangkan menjadi 2, menjadi subproblem C dan D dengan batasan tambahan untuk subproblem C adalah $X_2 \geq 2$ dan untuk subproblem D adalah $X2 \leq 1$. Logika dari pengembangan subproblem ini adalah karena solusi optimal dari subproblem A X2 = 1.2 tidak feasible, maka solusi integer haruslah berada dalam wilayah $X2 \geq 2$ atau $X2 \leq 1$.

Bagian C

Maksimisasi : $7X_1 + 6X_2$

Ditujukan pada : $2\ X_1 + 3\ X_2 \leq 12$

$6\ X_1 + 5\ X_2 \leq 30$

$X_1 \geq 4$

$X_2 \geq 2$

Bagian D

Maksimisasi : $7\ X_1 + 6\ X_2$

Ditujukan pada : $2\ X_1 + 3\ X_2 \leq 12$

$6\ X_1 + 5\ X_2 \leq 30$

$X_1 \leq 3$

$X_2 \leq 1$

Subproblem C tidak mempunyai solusi karena dua batasan awal tidak terpenuhi bila ada batasan tambahan $X_1 \geq 4$ dan $X_2 \geq 2$, jadi cabang ini tidak digunakan. Solusi optimal dari cabang D adalah X_1 = 4 1/6 dan X_2 = 1, profit 35.16, jadi batas atas berubah menjadi 35.16.

Iterasi 3

Kita buat cabang baru E dengan batasan tambahan batasan $X_1 \leq 4$ dan F dengan batasan tambahan $X1 \geq 5$

Bagian E

Maksimisasi : $7 X_1 + 6 X_2$

Ditujukan pada : $2 X_1 + 3 X_2 \leq 12$

$6 X_1 + 5 X_2 \leq 30$

$X_1 \geq 4$

$X_1 \leq 4$

$X_2 \leq 1$

Solusi optimal E adalah $X_1 = 4$ dan $X_2 = 1$ dengan profit 34

Bagian F

Maksimisasi : $7X_1 + 6X_2$

Ditujukan pada : $2 X_1 + 3 X_2 \leq 12$

$6 X_1 + 5 X_2 \leq 30$

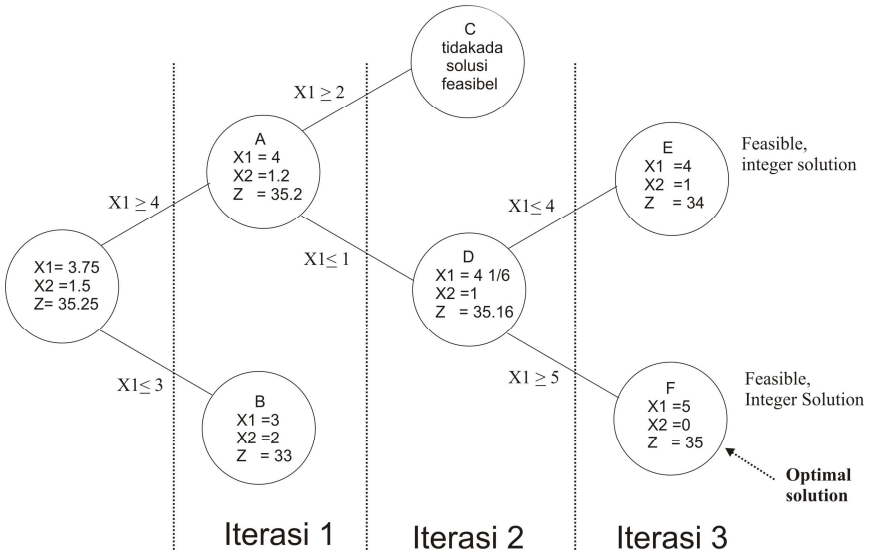
$X_1 \geq 4$

$X_1 \geq 5$

$X2 \leq 1$

Solusi optimal F adalah $X_1 = 5$ dan $X_2 = 0$ dengan profit 35

Jadi solusi optimal untuk pemrograman bulat ini adalah $X_1 = 5$ dan $X_2 = 0$ dengan profit 35



Gambar 2.5 Prosedur *Branch and Bound*

2.3 2.3 Analisis Sistem

3.3 Analisis sistem (*system analisis*) dapat didefinisikan sebagai berikut :

4.3 " *Pengurutan dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan, kesempatan, dan hambatan yang terjadi, serta kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya*."

Tahap analisa sistem dilakukan setelah tahap perencanaan (*system planning)*, dan sebelum tahap desain sistem (*system design*). Tahap analisis merupakan tahap yang sangat kritis dan sangat penting, karena kesalahan pada tahap ini akan menyebabkan kesalahan pada tahap selanjutnya.

Analisis yang dilakukan menggunakan metode analisis terstruktur, dimana metode analisis terstruktur ini banyak memakai *tools* dalam mempresentasikan suatu sistem, sehingga penguraian sistem dapat terstruktur dengan baik, jelas dan lebih mudah dipahami.

Penggambaran dan rancangan aplikasi dapat dibuat dalam bentuk Diagram Alir Data (DAD) atau Data *Flow Diagram* (DFD) dan Diagram Konteks dan

2.3.1 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) adalah representasi grafik dari sebuah sistem. DFD menggambarkan arus data dari suatu sistem informasi, baik sistem lama maupun sistem baru secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut berada.

Simbol-simbol yang digunakan pada diagram alir data atau *data flow diagram* dengan penjelasannya antara lain:

1. Entitas luar (*external entity*)

   Sesuatu yang berada di luar sistem, tetapi memberikan data ke dalam sistem atau memberikan data dari sistem, disimbolkan dengan suatu kotak notasi. External entity tidak termasuk bagian dari sistem bila sistem informasi dirancang untuk satu bagian maka bagian lain yang masih terkait menjadi external entity.

2. Arus data (*data flow*)

   Arus data merupakan tempat mengalirnya informasi dan digambarkan dengan garis yang menghubungkan komponen dari sistem. Arus data ditunjukan dengan arah panah dan garis diberi nama atau arus data yang mengalir. Arus data ini mengalir diantara proses, data store dan menunjukan arus data dari data yang berupa masukan untuk sistem atau hasil proses sistem.

3. Proses (*proccess*)

   Proses merupakan apa yang dikerjakan oleh sistem. Proses dapat mengolah data atau aliran data masuk menjadi aliran data keluar. Proses berfungsi mentransformasikan satu atau beberapa data keluaran sesuai dengan spesifikasi yang diinginkan. Setiap proses memiliki satu atau beberapa masukan serta menghasilkan satu atau beberapa data keluaran. Proses sering juga disebut *bubble*.

4. Simpanan data (*data store*)

   Simpanan data merupakan tempat penyimpanan data yang ada dalam sistem. Data *store* dapat disimbolkan dengan dua garis sejajar atau dua garis dengan salah satu sisi samping terbuka. Proses dapat mengambli data dari atau memberikan data ke *database*.

Tingkatan DFD atau diagram level *n* dimulai dari diagram konteks yang menjelaskan secara umum suatu sistem atau batasan sistem aplikasi yang akan dikembangkan. Kemudian DFD dikembangkan menjadi DFD tingkat 0 atau level 0 dan kemudian DFD level 0 dikembangkan lagi menjadi level 1 dan selanjutnya

sampai sistem tersebut tergambarkan secara rinci menjadi tingkatan-tingkatan lebih rendah lagi. Beberapa hal yang harus diperhatikan dalam membuat DFD ialah:

a. Pemberian Nomor pada diagram level *n*
b. Jangan menghubungkan langsung antara satu penyimpanan dengan penyimpanan lainnya (harus melalui proses).
c. Jangan menghubungkan langsung dengan tempat penyimpanan data dengan entitas eksternal / terminator (harus melalui proses), atau sebaliknya.
d. Jangan membuat suatu proses menerima input tetapi tidak pernah mengeluarkan output yang disebut dengan istilah “black hole”.
e. Jangan membuat suatu tempat penyimpanan menerima input tetapi tidak pernah digunakan untuk proses.
f. Jangan membuat suatu hasil proses yang lengkap dengan data yang terbatas yang disebut dengan istilah “magic process”.
g. Jika terdapat terminator yang mempunyai banyak masukan dan keluaran, diperbolehkan untuk digambarkan lebih dari satu sehingga mencegah penggambaran yang terlalu rumit, dengan memberikan tanda asterik (*) atau garis silang (#), begitu dengan bentuk penyimpanan.
h. Aliran data ke proses dan keluar sebagai output keterangan aliran data berbeda.

Dalam diagram n DFD dapat digunakan untuk menggambarkan diagram fisik maupun diagram logis.

1. Diagram Fisik

Diagram Fisik lebih tepat digunakan untuk menggambarkan sistem yang ada (sistem yang lama). Penekanan dari Diagram Fisik adalah bagaimana proses-proses dari sistem diterapkan (dengan cara apa, oleh siapa dan dimana), termasuk proses-proses manual.

2. Diagram Logis / Logika

Diagram Logis lebih tepat digunakan untuk menggambarkan sistem yang akan diusulkan (sistem yang baru). Untuk sistem komputerisasi, penggambaran Diagram Logis hanya menunjukkan kebutuhan proses dari sistem yang diusulkan secara logika, biasanya proses-proses yang digambarkan hanya merupakan proses-proses secara komputer saja.

2.3.2 Diagram Konteks

Diagram Konteks adalah diagram tingkat tinggi dari Diagram Aliran Data yang merupakan gambaran global dari sistem informasi yang menggambarkan aliran-aliran data ke dalam maupun keluar suatu sistem dan merupakan alat yang digunakan untuk melihat batasan antara sistem dengan *eksternal entity***.**

2.4 Perancangan Sistem

Perancangan sistem adalah suatu fase kreatif dalam siklus pengembangan sistem, dimana sistem yang dirancang harus memenuhi tujuan sistem secara teknis dana operasional. Pada tahap ini perlu dikembangkan spesifikasi untuk keluaran, database, operasi pemrosesan data, masukan data, pengendalian serta pengamanan.

Perancangan sistem informasi yang dilakukan bertujuan untuk meghilangkan beberapa kelemahan sistem dan meningkatkan kemampuan sistem dalam menyelesaikan permasalahan yang ada dengan acuan dari hasil analisa yang telah dilakukan.

2.4.1 Basis Data

Basis data terdiri dari kata basis dan data. Basis dapat diartikan gudang atau tempat bersarang dan data yang berarti representasi fakta dunia nyata yang mewakili suatu objek seperti manusia, hewan, peristiwa, konsep dan sebaginya yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi atau kombinasinya.

Jadi dapat disimpulkan bahwa basis data merupakan kumpulan data (arsip) yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan. Atau bisa diartikan sebagai kumpulan file, tabel, arsip yang saling berhubunngan yang disimpan dalam media penyimpanan elektronik.

2.4.2 Perancangan Basis Data

Perancangan basis data dibutuhkan agar diperoleh suatu sistem yang sesuai dengan apa yang diinginkan, dimana dapat melalui tahapan berikut

2.4.2.1 Entity Relationship Diagram (ERD)

Entity Relationship Diagram atau biasa dikenal dengan diagram E-R secara grafis menggambarkan isi sebuah database. Diagram ini memiliki dua komponen utama yaitu entity dan relasi. Untuk melambangkan fungsi diatas maka digunakan simbol-simbol yang bisa dilihat pada daftar simbol.

Elemen-elemen Entity Relationship Diagram adalah sebagai berikut:

1. Entity (Entitas)

   Pada E-R diagram, entity digambarkan dengan sebuah bentuk persegi panjang. Entity adalah sesuatu apa saja yang ada didalam sistem, nyata maupun abstrak dimana data tersimpan. Entitas diberi nama dengan kata benda dan dapat dikelompokan dalam empat jenis nama, yaitu : orang, benda, lokasi kejadian (terdapat unsur waktu didalamnya).

2. Relationship (Relasi)

   Pada E-R diagram, relationship dapat digambarkan dengan sebuah bentuk belah ketupat. Realationship adalah hubungan alamiah yang terjadi antara entitas. Pada umumnya relationship diberi nama dengan kata kerja dasar, sehinga memudahkan untuk melakukan pembacaan relasinya.

3. Atribut

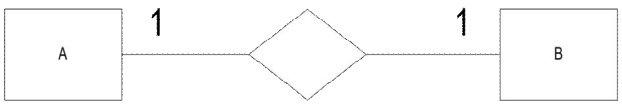
   Seacara umum atribut adalah sifat atau karakteristik dari tiap entitas maupun tiap relationship. Maksudnya adalah sesutau yang menjelaskan apa sebenarnya yang dimaksud entitas maupun relationship, sehingga sering dikatakan bahwa atribut adalah elemen dari setiap entitas dan relationship.

4. Kardinalitas

   Kardinalitas relasi menunjukan jumlah maksimum tupel yang dapat berelasi dengan entitas yang lainnya. Dari sejumlah kemungkinan banyaknya hubungan yang terjadi dari entitas, kardinalitas relasi merujuk kepada hubungan maksimum yang terjadi dari entitas yang satu ke entitas yang lainnya dan begitu juga sebaliknya. Terdapat tiga macam kardinalitas relasi, yaitu :

   a. One to one Relationship

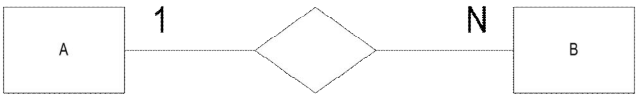
   Tingkat hubungann satu ke satu, dinyatakan dengan satu kejadian pada entitas pertama, hanya mempunyai satu hubungan dengan satu kejadian pada entitas yang kedua dan sebaliknya.



Gambar 2.6 *One to One Relationship*

   b. One to many Relationship

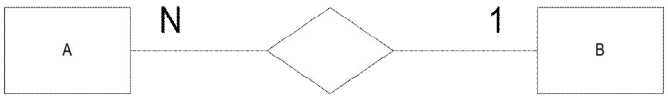
   Tingkat hubungan satu ke banyak adalah untuk satu kejadian pada entitas yang pertama dapat mempunyai banyak hubungan dengan kejadian pada entitas yang kedua.



Gambar 2.7 *One to Many Relationship*

c. Many To One Relationship

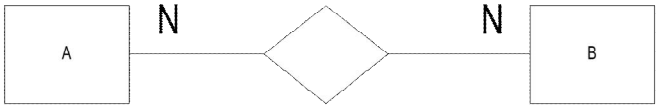
Untuk banyak kejadian pada entitas yang pertama hanya dapat mempunyai satu hubungan dengan kejadian pada entitas yang kedua.



Gambar 2.8 *Many to One Relationship*

d. Many to many Relationship

Tingkat hubungan banyak ke banyak terjadi jika tiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainnya, baik dilihat dari sisi entitas yang pertama maupun dilihat dari sisi yang kedua.



Gambar 2.9 *Many to Many Relationship*

5. *Key* (Kunci)

Sebuah atribut atau set atribut yang nilainya mengidentifikasikan entitas secara unik dalam suatu entitas. *Key* memiliki beberapa jenis sesuai dengan

kegunaannya masing-masing, yaitu *primary key* (kunci utama), *foreign key* (kunci tamu).

2.5 Pemodelan proses bisnis

2.5.1 BPMN (*Bisnis Proses Modeling Notation*)

BPMN adalah salah satu alat yang dapat digunakan untuk memodelkan proses bisnis. BPMN dikembangkan berbasiskan *flowchart dan* menyediakan sarana memodelkan aktivitas pihak luar yang terlibat dalam proses. sehingga alur proses bisnis yang digambarkan mudah untuk dipahami.

Diagram BPMN terdiri atas beberapa elemen. Elemen ini terbagi atas epat kategori yaitu *Flow object, Connecting object, Swimlanes*, dan *Atrifacts.* Berikut penjelasan dari masing-masing elemen BPMN.

1. *Flow object,* terdiri atas :

a. *Event* dipresentasikan dalam bentuk lingkaran dan menjelaskan apa yang terjadi saat itu. Ada tiga jenis *event* yaitu *start, intermediate,* dan *end*. Masing-masing mewakili kejadian dimulainya proses bisnis, dan akhir dari proses bisnis, interupsi proses bisnis, dan akhir dari proses bisnis. Untuk setiap jenis *event* tersebut sendiri terbagi atas beberapa jenis, misalnya *message start.*

2. *Activity*

Merepresentasikan *task* yang harus diselesaikan. Ada empat macam *activity* yaitu *task, looping task, sub process, dan looping subprocess.*

3. *Gateway*

Merepresentasikan pemecahan alur yang terdapat di dalam proses bisnis.

4. *Connecting object,* terdiri atas :

a. *Squence flow,* merepresentasikan pilihan *default* untuk menjalankan proses

b. *Message flow,* merepresentasikan aliran pesan antar proses

c. *Association*, digunakan untuk menghubungkan elemen dengan *artifact*

3. *Swimlanes,* digunakan untuk mengkategorikan secara visual seluruh elemen dengan diagram. Ada dua jenis *swimlanes, yaitu pool* dan *lane. Lane* terletak di bagian dalam *pool* untuk mengkategorisasi elemen-elemen di dalam *pool* menjadi lebih spesifik.

4. *Artifacts,* digunakan untuk memberi penjelasan pada diagram. Elemen ini terdiri atas tiga bagian :

a. *Data object,* digunakan untuk menjelaskan data yang ada apa yang dibutuhkan dalam proses.

b. *Group*, untuk mengelompokan sejumlah aktivitas di dalam proses tanpa mempengaruh proses yang sedang berjalan.

c. *Annotation,* digunakan untuk memberi catatan agar diagram menjadi lebih mudah dimengerti.

2.6 Perangkat Lunak Pendukung

Perangkat lunak merupakan suatu kebutuhan yang harus dipenuhi oleh suatu sistem. Kebutuhan ini diperlukan untuk mencapai suatu tujuan.

2.6.1 MySQL

MySQL adalah suatu sistem manajemen database. Suatu database adalah sebuah kumpulan data yang terstruktur. Untuk menambahkan, mengakses, dan memproses data yang tersimpan pada suatu database komputer anda memerlukan sistem manajemen database seperti MySQL. Karena komputer sangat unggul dalam menangani sejumlah besar data, sistem manajemen database memainkan suatu peranan yang penting dalam komputasi, baik sebagai utility stand-alone maupun bagian dari aplikasi lainnya.

Suatu database relasional menyimpan data dalam tabel-tabel terpisah. Hal ini memungkinkan kecepatan dan fleksibilitas. Tabel-tabel yang dihubungkan dengan relasi yang ditentukan membuatnya bisa mengkombinasikan data dari beberapa tabel pada suatu permintaan. Bagian SQL dari kata MYSQL berasal dari “Structured Query Language” bahasa paling umum yang dipergunakna untuk mengakses database.

Beberapa perintah dasar SQL yang sering dipergunakan pada MySQL adalah sebagai berikut :

a. Create Database

   Yaitu perintah yang digunakan untuk membuat database baru.

   Sintaks : Create database database_nama database

b. Drop Database

Yaitu perintah yang digunakan untuk menghapus database.

Sintaks : Drop Tabel Tabel_name

c. Create Tabel

Yaitu perintah yang digunakan untuk membuat tabel baru.

Sintaks Create Tabel tabel_name (create_definition)

d. Describe

Yaitu perintah yang digunakan untuk mendeskripsikan tabel atau logam

Sintaks Describe (Desc) tabel [colum]

e. Alter Tabel

Yaitu perintah yang digunakan untuk memodifikasi tabel

Sintaks Alter [Ignor] Tabel table_name

f. Drop Tabel

Yaitu perintah yang digunakan untuk menghapus tabel

Sintaks Drop Tabel tabel_name [tabel_name..]

g. Delete

Yaitu perintah yang digunakan untuk menghapus record dri tabel

Sintaks Delete From tabel_name Where Where_definiition

h. Select

Yaitu perintah yang digunakan untuk query ke database

Sintaks select*from tabel_name

Select Field from tabel_name

2.5.2 HTML

HTML (*Hypertext Markup Language*) dapat didefinisikan sebagai seperangkat kode khusus yang ditambahkan ke dalam text dengan tujuan melakukan pemformatan dan menghubungkan informasi-informasi yang terdapat didalamnya ***(Taylor**, 2004)*. HTML diciptakan berdasarkan pada *Standard Generalized Markup Language* (SGML). Untuk menuliskan dokumen HTML dapat digunakan teks editor apapun. Dokumen HTML disimpan dengan ekstensi *.html.

Semua informasi HTML diformat menggunakan perintah-perintah yang disebut ***tag*.** Sebuah ***tag*** adalah perintah yang ada diantara tanda kurung siku buka (<) dan tanda kurung siku tutup (>). Umumnya ***tag*** memiliki pasangan yaitu *closing tag* yang berfungsi memberitahukan komputer untuk menghentikan pemformatan. *Closing tag* ditulis dengan penambahan tanda (/) didepan nama ***tag*.** Misalnya **<html>**, jenis (***tag*)** ini akan memberitahu program penerjemah HTML (*browser*) bahwa dokumen tersebut ditulis dan dimanipulasi dengan menggunakan standar HTML.

2.5.3 PHP

PHP merupakan kependekan dari *Hypertext Preprocessing*, banyak dipakai untuk memrogram situs *web* dinamis, walaupun tidak tertutup kemungkinan digunakan untuk pemakaian lain. PHP dikatakan sebagai sebuah server-side embedded script language artinya sintaks-sintaks dan perintah yang kita berikan akan sepenuhnya dijalankan oleh server. Aplikasi-aplikasi yang

dibangun oleh PHP pada umumnya akan memberikan hasil pada web browser, tetapi prosesnya secara keseluruhan dijalankan di server.

Beberapa kelebihan PHP sebagai bahasa pemrograman web adalah sebagai berikut ***(Castagnetto**, 1999)* :

1. PHP berkerja cepat karena PHP adalah bahasa pemrograman yang berbasis bahasa C. Selain bekerja cepat PHP juga mudah digunakan.
2. PHP merupakan *web scripting open source*
3. PHP dapat berfungsi pada semua sistem operasi.
4. PHP dapat mengakses berbagai program (misalnya *mail server*, Oracle, DB2, dll). Hal ini karena PHP dilengkapi dengan berbagai fungsi yang dapat mengakses berbagai jenis program.
5. Kemudahan mencari informasi mengenai PHP. Banyak sekali pengembang dan pengguna PHP serta forum-forum diskusi web yang menyediakan informasi yang diperlukan mengenai PHP (baik mengenai funsi-fungsinya, maupun *error* yang terjadi pada PHP).