

BAB II

TINJAUAN PUSTAKA

2.1 Pengertian Hotel

Menurut Dinas Pariwisata dan Budaya (2008) hotel adalah suatu bidang usaha yang menggunakan suatu bangunan atau sebagian bangunan yang disediakan khusus, untuk setiap orang menginap, makan, memperoleh pelayanan dan menggunakan fasilitas lainnya dengan pembayaran. Ciri khusus dari hotel adalah mempunyai restoran yang dikelola langsung di bawah manajemen hotel tersebut. Kelas hotel ditentukan oleh Dinas Pariwisata Daerah.

Hotel Berbintang adalah suatu bidang usaha yang menggunakan suatu bangunan atau sebagian yang penting disediakan secara khusus, untuk setiap orang yang menginap memperoleh pelayanan dan menggunakan fasilitas lainnya dengan pembayaran, dan telah memenuhi persyaratan yang ditentukan oleh Dinas Pariwisata Daerah. Persyaratan tersebut adalah:

1. Persyaratan Fisik, seperti lokasi hotel, kondisi bangunan.
2. Bentuk pelayanan yang diberikan.
3. Kualifikasi tenaga kerja, seperti pendidikan, dan kesejahteraan karyawan.
4. Fasilitas olahraga dan rekreasi lainnya yang tersedia, seperti lapangan tenis, kolam renang, dan diskotik.
5. Jumlah kamar yang tersedia.

Akomodasi lainnya yaitu suatu usaha yang menggunakan suatu bangunan yang disediakan secara khusus, di mana setiap orang dapat menginap dengan atau tanpa makan dan memperoleh pelayanan serta menggunakan fasilitas lainnya dengan pembayaran. Akomodasi lainnya merupakan hotel melati yaitu hotel yang belum memenuhi persyaratan sebagai hotel berbintang seperti yang ditentukan oleh Dinas Pariwisata Daerah, penginapan remaja, pondok wisata, dan jasa akomodasi lainnya.

1. Hotel Melati

Usaha penyediaan jasa pelayanan penginapan umum yang dikelola secara komersial dengan menggunakan sebagian atau seluruh bagian bangunan.

2. Penginapan Remaja

Usaha penyediaan jasa pelayanan penginapan yang ditujukan bagi remaja sebagai akomodasi dalam rangka kegiatan pariwisata dengan tujuan rekreasi, dan memperluas pengetahuan.

3. Villa

Usaha penyediaan jasa pelayanan penginapan bai umum dengan pembayaran yang dilakukan secara individual dengan menggunakan sebagian dari tempat tinggalnya.

2.2 *Analytic Network Process* (ANP)

Analytic Network Process atau ANP adalah teori umum pengukuran relatif yang digunakan untuk menurunkan rasio prioritas komposit dari skala rasio individu yang mencerminkan pengukuran relatif dari pengaruh elemen-elemen yang saling berinteraksi berkenaan dengan kriteria kontrol [SAA-03]. ANP merupakan teori matematika yang memungkinkan seseorang untuk memperlakukan *dependence* dan *feedback* secara sistematis yang dapat menangkap dan mengkombinasi faktor-faktor *tangible* dan *intangible* [AZI-03].

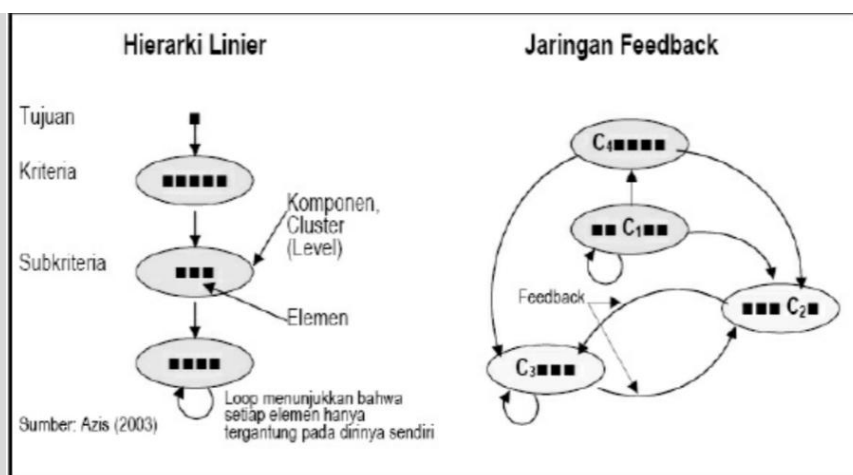
2.2.1 Kaidah Dasar ANP

Analytic Network Process atau ANP merupakan pendekatan baru metode kualitatif. Diperkenalkan Profesor Thomas Saaty pakar riset dari Pittsburgh University, dimaksudkan untuk “menggantikan” metode *Analytic Hierarchy Process* (AHP). Kelebihan ANP dari metodologi yang lain adalah kemampuannya melakukan pengukuran dan sintesis sejumlah faktor-faktor dalam hierarki atau jaringan. Tidak ada metodologi lain yang mempunyai fasilitas sintesis seperti metodologi ANP [SAA-96].

ANP digunakan untuk menurunkan rasio prioritas komposit dari skala rasio individu yang mencerminkan pengukuran relatif dari pengaruh elemen-elemen yang saling berinteraksi berkenaan dengan kriteria kontrol [SAA-01]. ANP merupakan teori matematika yang memungkinkan seseorang untuk memperlakukan *dependence* dan *feedback* secara sistematis yang dapat menangkap dan mengkombinasi faktor-faktor *tangible* dan *intangible*.

ANP merupakan pendekatan baru dalam proses pengambilan keputusan yang memberikan kerangka kerja umum dalam memperlakukan keputusan-keputusan tanpa membuat asumsi-asumsi tentang independensi elemen-elemen pada level yang lebih tinggi dari elemen-elemen pada level yang lebih rendah dan tentang independensi elemen-elemen dalam suatu level. Berbeda dengan *Analytic Hierarchy Process* (AHP), ANP dapat menggunakan jaringan tanpa harus menetapkan level seperti pada hierarki yang digunakan dalam AHP. Konsep utama dalam ANP adalah *influence* ‘pengaruh’, sementara konsep utama dalam AHP adalah *preference* ‘preferensi’. AHP dengan asumsi-asumsi dependensinya tentang *cluster* dan elemen merupakan kasus khusus dari ANP [ASC-05].

Pada jaringan AHP terdapat level tujuan, kriteria, subkriteria, dan alternatif, dimana masing-masing level memiliki elemen. Sementara itu, pada jaringan ANP, level dalam AHP disebut *cluster* yang dapat memiliki kriteria dan alternatif di dalamnya, yang sekarang disebut *node*.



Gambar 2.1. Perbandingan Hierarki Linier dan Jaringan Feedback [ASC-05].

Dengan *feedback*, alternatif-alternatif dapat bergantung/terikat pada kriteria seperti pada hierarki tetapi dapat juga bergantung/terikat pada sesama alternatif. Lebih jauh lagi, kriteria-kriteria itu sendiri dapat tergantung pada alternatif-alternatif dan pada sesama kriteria. Sementara itu, *feedback* meningkatkan prioritas yang diturunkan dari *judgements* dan membuat prediksi menjadi lebih akurat. Oleh karena itu, hasil dari ANP diperkirakan akan lebih stabil. Dari jaringan *feedback* pada gambar 2.2 dapat dilihat bahwa simpul atau elemen utama

dan simpul-simpul yang akan dibandingkan dapat berada pada *cluster-cluster* yang berbeda. Sebagai contoh, ada hubungan langsung dari simpul utama C4 ke cluster lain (C2 dan C3), yang merupakan *outer dependence*. Sementara itu, ada simpul utama dan simpul-simpul yang akan dibandingkan berada pada cluster yang sama, sehingga *cluster* ini terhubung dengan dirinya sendiri dan membentuk hubungan *loop*. Hal ini disebut *inner dependence* [ASC-05].

Dalam suatu jaringan, elemen dalam suatu komponen/*cluster* bisa saja berupa orang dan elemen dalam komponen/*cluster* yang lain bisa saja juga berupa orang. Elemen dalam suatu komponen/*cluster* dapat mempengaruhi elemen lain dalam komponen/*cluster* yang sama (*inner dependence*), dan dapat pula mempengaruhi elemen pada *cluster* yang lain (*outer dependence*) dengan memperhatikan setiap kriteria. Yang diinginkan dalam ANP adalah mengetahui keseluruhan pengaruh dari semua elemen. Oleh karena itu, semua kriteria harus diatur dan dibuat prioritas dalam suatu kerangka kerja hierarki kontrol atau jaringan, melakukan perbandingan dan sintesis untuk memperoleh urutan prioritas dari sekumpulan kriteria ini. Kemudian kita turunkan pengaruh dari elemen dalam *feedback* dengan memperhatikan masing-masing kriteria. Akhirnya, hasil dari pengaruh ini dibobotkan dari prioritas dari kriteria, dan ditambahkan untuk memperoleh pengaruh keseluruhan dari masing-masing elemen [ASC-05].

2.2.2 Landasan ANP

ANP memiliki tiga aksioma yang menjadi landasan teorinya [ASC-05]:

1. **Resiprokal.** Aksioma ini menyatakan bahwa jika PC (EA,EB) adalah nilai perbandingan pasangan dari elemen A dan B, dilihat dari elemen induknya C, yang menunjukkan berapa kali lebih banyak elemen A memiliki apa yang dimiliki elemen B, maka $PC(EB,EA) = 1/PC(EA,EB)$. Misalkan, jika A lima kali lebih besar dari B, maka B besarnya 1/5 dari besar A.
2. **Homogenitas.** Aksioma ini menyatakan bahwa elemen-elemen yang dibandingkan sebaiknya tidak memiliki perbedaan terlalu besar, yang dapat menyebabkan kesalahan *judgements* yang lebih besar.

3. **Ekspektasi.** Aksioma ini menyatakan bahwa mereka yang mempunyai alasan terhadap keyakinannya harus memastikan bahwa ide-ide mereka cukup terwakili dalam hasil agar sesuai dengan ekspektasinya.

2.2.3 Prinsip Dasar ANP

Prinsip-prinsip dasar ANP ada tiga, yaitu dekomposisi, penilaian komparasi (*comparative judgements*), dan komposisi hirarki atau sintesis dari prioritas [SAA-96].

Prinsip dekomposisi diterapkan untuk menstrukturkan masalah yang kompleks menjadi kerangka hierarki atau jaringan *cluster*, *sub-cluster*, *sus-sub cluster*, dan seterusnya. Dengan kata lain dekomposisi adalah memodelkan masalah ke dalam kerangka ANP.

Prinsip penilaian komparasi diterapkan untuk membangun perbandingan pasangan (*pairwise comparison*) dari semua kombinasi elemen-elemen dalam *cluster* dilihat dari *cluster* induknya. Perbandingan pasangan ini digunakan untuk mendapatkan prioritas lokal dari elemen-elemen dalam suatu *cluster* dilihat dari *cluster* induknya.

Prinsip komposisi hierarkis atau sintesis diterapkan untuk mengalikan prioritas lokal dari elemen-elemen dalam *cluster* dengan prioritas ‘global’ dari elemen induk, yang akan menghasilkan prioritas global seluruh hierarki dan menjumlahkannya untuk menghasilkan prioritas global untuk elemen level terendah (biasanya merupakan alternatif).

2.2.4 Penentuan Bobot Setiap Elemen

Setelah model dibuat, maka dilakukan pentabelan dari hasil data *pairwise comparison* dengan menggunakan tabel supermatriks. Setiap komponen pada jaringan harus ditentukan bobot relatifnya untuk mengetahui tingkat prioritas pengambilan keputusan terhadap setiap komponen tersebut. Langkah pertama dalam menentukan bobot prioritas masing-masing komponen adalah melakukan perbandingan berpasangan, yaitu membandingkan prioritas antara dua komponen dalam setiap sub-sistem. Perbandingan tersebut kemudian ditransformasikan ke dalam bentuk matrik untuk dilakukan analisis numerik. Perbandingan setiap

elemen menggunakan angka/nilai prioritas yang dijabarkan pada tabel 2.1, sebagai berikut:

Tabel 2.1 Skala Fundamental [BAY-06].

Skala	Definisi	Penjelasan
1	Sama pentingnya dibandingkan yang lain	Kedua elemen mempunyai pengaruh yang sama
3	Moderat pentingnya di banding yang lain	Pengalaman dan penelitian sedikit memihak satu elemen dibandingkan dengan pasangannya
5	Kuat pentingnya di banding yang lain	Pengalaman dan penilaian sangat memihak satu elemen dibandingkan dengan pasangannya
7	Sangat kuat pentingnya di banding yang lain	Pengalaman dan penilaian lebih kuat memihak satu elemen dibandingkan dengan elemen yang lain, dan dominannya terlihat dalam praktik
9	Ekstrim pentingnya di banding yang lain	Satu elemen terbukti mutlak lebih disukai dibandingkan dengan pasangannya pada tingkat keyakinan dengan pasangannya pada tingkat keyakinan yang tertinggi
2,4,6,8,9	Nilai diantara dua penilaian yang berdekatan	Diberikan bila terdapat penilaian antara dua penilaian yang berdekatan
Kebalikan	$a_{ij} = \frac{1}{a_{ji}}$	Jika elemen i memiliki salah satu angka di atas ketika dibandingkan elemen j, maka j memiliki nilai kebalikannya ketika dibandingkan elemen i.

Nilai prioritas dari perbandingan berpasangan didapat dari penilaian responden (k) yang diperoleh melalui penyebaran quisioner, ditransformasikan dalam matrik berukuran $n \times n$, dimana n yaitu jumlah kriteria yang dibandingkan.

Nilai Rasio Konsistensi (RK) merupakan tolak ukur konsistensi hasil komparasi berpasangan. Rasio Konsistensi merupakan hasil bagi dari Indeks Konsistensi (IK) dan Indeks Acak (IA). Nilai RK dapat dirumuskan pada persamaan 2.2 berikut [HIK-11]:

$$RK = \frac{IK}{IA} \quad \dots (2.1)$$

Menentukan Indeks Konsistensi (IK) dengan rumus yang ditunjukkan pada persamaan 2.3 berikut [HIK-11]:

$$IK = \frac{\lambda_{maks} - n}{n-1} \dots (2.2)$$

Dengan n merupakan jumlah kriteria.

Menentukan nilai eigen vektor (λ maks) ditunjukkan pada persamaan 2.4 berikut (Hikmah, 2011).

$$\lambda_{maks} = \sum(\text{jumlah kolom ke } - j \times V_{pi} \text{ untuk } i = j) \dots(2.3)$$

Dengan kriteria kompetensi (VP) merupakan nilai normalisasi seluruh kriteria [HIK-11].

Pengujian dilakukan dengan mencari nilai rasio konsistensi untuk setiap responden. Pengujian ini dilakukan terhadap kriteria kompetensi yang telah ditentukan. Jika nilai rasio konsistensi kurang atau sama dengan 0,1 berarti penilaian dapat dipertanggungjawabkan [SAA-80].

Dari 500 buah sampel matrik acak dengan skala perbandingan 1–9, untuk beberapa orde matrik, [SAA-80] mendapatkan nilai rata-rata indeks acak (IA) yang ditampilkan pada tabel berikut:

Tabel 2.2 Nilai IA Menurut Saaty

Orde	1	2	3	4	5	6	7	8	9
IA	0	0	0,56	0,9	1,12	1,24	1,32	1,41	1,45
Orde	10	11	12	13	14	15			
IA	1,49	1,51	1,48	1,56	1,57	1,59			

Elemen-elemen matrik perbandingan berpasangan (g_{ij}) dinormalisasi untuk mendapatkan nilai matrik tak berbobot, yaitu dengan menjumlahkan terlebih dahulu kolom pada matrik perbandingan berpasangan, kemudian membagi tiap elemen matrik dengan jumlah kolom tersebut. Formulasi dalam melakukan normalisasi matrik perbandingan berpasangan adalah pada persamaan 2.4 sebagai berikut [HIK-11]:

$$\text{Nilai normalisasi} = \frac{g_{ij}}{\sum_{i=1}^n g_{ij}} \dots (2.4)$$

Kemudian akan dilakukan proses pembobotan untuk setiap *cluster* yang telah ditentukan berdasarkan kriteria calon pegawai. Algoritma perhitungan pembobotan yang dilakukan dimulai dari data dengan bentuk *pairwise comparison* sampai dihasilkan bobot tiap indikator kinerjanya. Kriteria dibuat berdasarkan kebutuhan dan tujuan dari pemilihan [SAP-08].

Untuk menunjukkan hasil akhir dari perhitungan perbandingan maka supermatriks akan dipangkatkan secara terus-menerus hingga angka setiap kolom dalam satu baris sama besar. Rumus perhitungannya, dapat dilihat pada persamaan 2.5 [SAP-08].

$$\lim_{M \rightarrow \infty} \frac{1}{M} \sum_{k=1}^M \frac{\sum_{j=1}^n a_{ij}^k}{\sum_{i=1}^n \sum_{j=1}^n a_{ij}^k} \dots (2.5)$$

Bobot kriteria kompetensi (VP) diperoleh dengan membagi total nilai normalisasi tiap kriteria dengan total nilai normalisasi seluruh kriteria [HIK-11].

$$\text{Bobot kriteria kompetensi : } V_{pi} = \frac{1}{j_{total}} \begin{bmatrix} j_i \\ \dots \\ j_n \end{bmatrix} \dots (2.6)$$

2.2.5 Akurasi

Akurasi merupakan seberapa dekat suatu angka hasil pengukuran terhadap angka sebenarnya (*true value atau reference value*). Dalam penelitian ini akurasi diagnosis dihitung dari jumlah diagnosis yang tepat dibagi dengan jumlah data. Tingkat akurasi diperoleh dengan perhitungan sesuai dengan persamaan 2.7 [NUG-06].

$$\text{Tingkat akurasi} = \frac{\sum \text{data uji benar}}{\sum \text{total data uji}} \dots (2.7)$$

$$\text{Akurasi (\%)} = \frac{\sum \text{data uji benar}}{\sum \text{total data uji}} \times 100\% \dots (2.8)$$



2.3 Context Diagram (CD)

Context diagram atau diagram konteks adalah khusus DFD (bagian dari DFD yang berfungsi memetakan model lingkungan), yang direpresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem. Diagram konteks menyoroti sejumlah karakteristik penting sistem, yaitu:

1. Kelompok pemakai merupakan organisasi atau sistem lain dimana sistem kita melakukan komunikasi yang disebut juga sebagai terminator.
2. Data masuk merupakan data yang diterima sistem dari lingkungan dan harus diproses dengan cara tertentu.
3. Data keluar merupakan data yang dihasilkan sistem kita dan diberikan ke dunia luar.
4. Penyimpanan data (*data store*) yang digunakan secara bersama antara sistem kita dengan terminator. Data ini dapat dibuat oleh sistem dan digunakan oleh lingkungan atau sebaliknya, dibuat oleh lingkungan dan digunakan oleh sistem kita. Hal ini berarti pembuatan simbol *data store* dalam diagram konteks dibenarkan, dengan syarat simbol tersebut merupakan bagian dari dunia luar sistem.
5. Batasan antara sistem kita dan lingkungan (*rest of the world*).

Diagram konteks dimulai dengan penggambaran terminator, aliran data, aliran kontrol, penyimpanan dan proses tunggal yang merepresentasikan keseluruhan sistem. Bagian termudah adalah menetapkan proses yang hanya terdiri dari satu lingkaran dan diberi nama yang mewakili sistem. Nama dalam hal ini menjelaskan proses atau pekerjaan atau dalam kasus ekstrim berupa nama perusahaan yang mewakili proses yang dilakukan keseluruhan organisasi. Terminator direpresentasikan dalam bentuk persegi panjang dan berkomunikasi langsung dengan sistem melalui aliran data atau tidak langsung sehingga harus melalui penyimpanan eksternal. Antara terminator tidak diperbolehkan komunikasi langsung [WHI-04].

2.4 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) adalah suatu gambaran grafis dari suatu sistem yang menggunakan sejumlah bentuk-bentuk simbol untuk menggambarkan bagaimana data mengalir melalui suatu proses yang saling berkaitan [MCL-04].

Dalam analisa data, penggunaan *data flow diagram* sangat penting sekali dalam pemakaian sistem untuk memahami sistem secara logika. Penggunaan *data flow diagram* ini untuk mengetahui sistem yang sudah ada dalam suatu organisasi atau bisa untuk membuat sistem baru yang dikembangkan tanpa mempertimbangkan keadaan dimana data tersebut mengalir atau keadaan dimana data tersebut akan disimpan. DFD juga dapat disebut sebagai dokumentasi dari sistem yang berjalan dalam suatu organisasi.

DFD terdiri dari diagram konteks dan diagram rinci (*DFD Levelled*). Diagram konteks berfungsi memetakan model lingkungan (menggambarkan hubungan antara entitas luar, masukan dan keluaran sistem), yang direpresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem. *DFD levelled* menggambarkan sistem sebagai jaringan kerja antara fungsi yang berhubungan satu sama lain dengan aliran dan penyimpanan data, model ini hanya memodelkan sistem dari sudut png fungsi.

2.5 Entity Relational Diagram (ERD)

E-R Model adalah sebuah bagan yang merupakan gambaran data dari seluruh proyek. E-R Model yang berisi komponen-komponen himpunan *entity* dan himpunan *relationship* yang masing-masing dilengkapi dengan atribut-atribut yang mempresentasikan seluruh fakta dari dunia nyata dapat digambarkan dengan lebih sistematis dengan menggunakan *Entity-Relationship Model* [FAT-02].

2.6 Basis Data

Penerapan basis data dalam sistem informasi disebut dengan sistem basis data (database sistem). Sistem basis data ini adalah suatu sistem informasi yang mengintegrasikan kumpulan dari data yang saling berhubungan satu dengan yang lain dan tersedia untuk beberapa aplikasi yang bermacam-macam di dalam suatu organisasi.

Tujuan dari desain basis data ini adalah untuk menentukan data-data yang dibutuhkan dalam sistem, sehingga informasi yang dihasilkan dapat terpenuhi dengan baik. Perancangan database yang digunakan adalah untuk memudahkan dalam mengetahui file-file database yang digunakan dalam perancangan sistem, sekaligus untuk mengetahui hubungan antara file dari database tersebut.

Beberapa kriteria yang harus dipenuhi dalam database adalah sebagai

1. Melakukan perubahan-perubahan data untuk menyelesaikan dan untuk
2. pengembangan yang akan datang
3. Mengurangi redudansi data atau duplikasi data.
4. Menyimpan seluruh data dan informasi secara terpusat.
5. Menjamin keamanan data.

2.6.1 Perintah untuk mengelola basis data

Pengelolaan database dapat dilakukan dengan menggunakan perintah-perintah SQL (Structured Query Language), bahasa yang khusus digunakan untuk melakukan akses database relasional. Perintah SQL dikelompokkan sebagai berikut:

- **DDL (Data Definition Language)**

Merupakan kelompok perintah yang digunakan untuk melakukan pendefinisian database dan pendefinisian tabel. Dengan kelompok perintah dalam DDL ini maka kita dapat membuat table (create table), mengubah strukturnya, menghapus table (drop table), membuat indeks (create index) untuk tabel, dan lain-lain yang bermuara pada pembentukan struktur database.

- **DML (Data Manipulation Language)**

Perintah (statement) SQL digunakan untuk melakukan manipulasi data dalam database, menambahkan (insert), mengubah (update), menghapus (delete), mengambil dan mencari data (query). Perintah SQL standar seperti select, insert, update, delete, create, view dan drop dapat digunakan untuk menyelesaikan tugas yang diberikan berhubungan dengan data suatu database.

- **DCL (Data Control Language)**

Termasuk dalam DCL adalah perintah untuk pendefinisian pemakai yang boleh mengakses database dan apa saja privileginya. Fasilitas ini tersedia pada sistem manajemen database yang memiliki fasilitas keamanan dengan membatasi pemakai dan kewenangannya

2.7 Perangkat Lunak Pendukung

PHP merupakan script untuk pemrograman script web server-side, script yang membuat dokumen HTML secara on the fly, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML.

Dengan menggunakan PHP maka maintenance suatu situs web menjadi lebih mudah. Proses update data dapat dilakukan dengan menggunakan script PHP.

PHP/FI merupakan nama awal dari PHP. PHP – Personal Home Page, FI adalah Form Interface. Dibuat pertama kali oleh Rasmus Lerdoff. PHP, awalnya merupakan program CGI yang dikhususkan untuk menerima input melalui form yang ditampilkan dalam browser web. Berikut ini adalah contoh penggunaan tag PHP.

```
<html>
<head>
<title>
Contoh Sederhana
</title>
</head>
<body>
<?php
echo("Hello apa kabar?");
?>
</body>
</html>
```

Gambar 2.2 Contoh code dalam PHP [SID-06]

2.7.1 Pengenalan Javascript

Javascript diperkenalkan pertama kali oleh Netscape pada tahun 1995. Pada awalnya bahasa ini dinamakan “*LiveScript*” yang berfungsi sebagai bahasa sederhana untuk browser Netscape Navigator 2. Pada masa itu bahasa ini banyak di kritik karena kurang aman, pengembangannya yang terkesan buru buru dan tidak ada pesan kesalahan yang ditampilkan setiap kali kita membuat kesalahan pada saat menyusun suatu program.

Kemudian sejalan dengan sedang giatnya kerjasama antara Netscape dan Sun (pengembang bahasa pemrograman “Java”) pada masa itu, maka Netscape

memberikan nama “JavaScript” kepada bahasa tersebut pada tanggal 4 desember 1995. Pada saat yang bersamaan Microsoft sendiri mencoba untuk mengadaptasikan teknologi ini yang mereka sebut sebagai “Jscript” di browser Internet Explorer 3 [SUP-06].

Javascript adalah bahasa yang berbentuk kumpulan skrip yang pada fungsinya berjalan pada suatu dokumen HTML, sepanjang sejarah internet bahasa ini adalah bahasa skrip pertama untuk web. Bahasa ini adalah bahasa pemrograman untuk memberikan kemampuan tambahan terhadap bahasa HTML dengan mengijinkan pengekseskuan perintah perintah di sisi user, yang artinya di sisi browser bukan di sisi server web. Javascript bergantung kepada browser (navigator) yang memanggil halaman web yang berisi skrip-skrip dari Javascript dan tentu saja terselip di dalam dokumen HTML. Javascript juga tidak memerlukan kompilator atau penterjemah khusus untuk menjalankannya (pada kenyataannya kompilator Javascript sendiri sudah termasuk di dalam browser tersebut). Lain halnya dengan bahasa “Java” (dengan mana JavaScript selalu di banding bandingkan) yang memerlukan kompilator khusus untuk menterjemahkannya di sisi user/klien. Skrip dari JavaScript terletak di dalam dokumen HTML. Kode tersebut tidak akan terlihat dari dalam jendela navigator anda, karena diantara tag (kalau anda mengerti HTML pasti tahu dengan istilah ini) tertentu yang memerintahkan navigator untuk memperlakukan bahwa skrip tersebut adalah skrip dari JavaScript [SUP-06].

Contoh dari skrip yang menunjukkan bahwa skrip tersebut adalah skrip dari JavaScript adalah sebagai berikut :

```
<SCRIPT language="Javascript">  
  letakkan script anda disini  
</SCRIPT>
```

Gambar 2.3 Contoh Javascript [SUP-06].

2.7.2 Power Designer 12.5

Power Designer merupakan tool pemodelan yang dikeluarkan oleh Sybase untuk membangun sebuah sistem informasi yang cepat, terstruktur dan efektif. Sybase Power Designer mendukung beberapa pemodelan yaitu *Requirement*

Management, Business Process, Data Modelling, XML Modelling, Application Modelling dengan UML, Information Liquidity Modelling dan Integrated Modelling

Secara sederhana, untuk melakukan pemodelan data pada Power Designer, kita harus memulainya pada level *Conceptual Data Model*, dimana pemodelan data dilakukan dengan menggunakan metode *Entity Relationship Diagram*. Pada CDM, tipe data yang dipergunakan bersifat general, dan tidak spesifik terhadap suatu database tertentu.

Tahap kedua adalah membuat *Physical Data Model* (PDM), PDM merupakan bentuk spesifik dari CDM yang telah kita bangun. Power Designer memiliki banyak dukungan target database, sehingga kita tidak perlu bingung mengenai tipe – tipe data yang dipergunakan, karena Power Designer akan menyesuaikan seperti pada tipe data yang kita definisikan sebelumnya pada tahap CDM.

Tahap terakhir adalah *generate script Data Definition Language* (DDL) dari PDM yang telah dibuat. Melalui DDL inilah kita dapat *generate* objek – objek database (*table, trigger, view, procedure*) sehingga kemudian DDL *script* ini dapat kita eksekusi ke software database lain seperti Oracle atau MySQL, atau dapat juga kita buat koneksi dan mengeksekusinya langsung via Power Designer.

2.7.3 MySQL 2.3.4

MySQL merupakan sebuah *database developer* yang juga bersifat *free*, MySQL banyak digunakan sebagai *database* karena mudah digunakan dan juga sangat banyak tersedia. MySQL menggunakan bahasa SQL yang sudah banyak digunakan saat ini.

MySQL merupakan *software database* yang termasuk paling populer di lingkungan Linux atau Unix, kepopuleran ini ditunjang karena performansi *query* dari *database*-nya yang saat itu bisa dikatakan paling cepat, dan juga memiliki sedikit permasalahan.

Beberapa keunggulan MySQL dibandingkan database lain adalah:

- Kemudahan dalam penggunaan: MySQL adalah *simple database* system dengan performa tinggi dan tidak kompleks untuk proses instalasi dan administrasinya dibanding dengan sistem yang lebih besar.
- Mendukung bahasa *query*: MySQL dapat menggunakan SQL, juga dapat diakses dengan menggunakan aplikasi ODBC.
- Kemampuan: banyak client dapat berhubungan dengan server pada saat bersamaan. *Clients* dapat menggunakan *multiple database* secara bersamaan. MySQL adalah multiuser database yang menggunakan bahasa *Structured Query Language* (SQL). MySQL dalam operasi *client-server* melibatkan server daemon MySQL di sisi server dan berbagai macam program serta *library* yang berjalan disisi *client*. MySQL mampu menangani data yang cukup besar, lebih dari 40 *database*, 10.000 tabel dan sekitar 7 juta baris, totalnya kurang lebih 100 *Gigabyte* data [CUS-99].

2.8 User Acceptance Test

User Acceptance Test seringkali merupakan langkah terakhir sebelum meluncurkan aplikasi. Biasanya user yang akan menggunakan aplikasi, menguji aplikasi sebelum mendapatkan aplikasi. Jenis pengujian ini memberikan users keyakinan bahwa aplikasi yang disampaikan kepada mereka memenuhi persyaratan mereka. Testing ini juga membantu *nail bugs* yang berhubungan dengan kegunaan dari aplikasi.

User Acceptance Test biasanya merupakan *black box* jenis pengujian. Dengan kata lain, fokusnya adalah pada fungsi dan kegunaan aplikasi daripada aspek teknis. Secara umum diasumsikan bahwa aplikasi akan sudah mengalami Unit, Integrasi dan Pengujian Sistem.

Seperti biasa proses perencanaan adalah proses paling penting dari semua langkah. Hal ini mempengaruhi efektivitas proses pengujian. Proses perencanaan menguraikan strategi UAT. Ini juga menggambarkan bidang fokus utama, masuk dan keluar kriteria.

UAT membantu untuk menguji aplikasi secara menyeluruh. Hal ini juga akan membantu memastikan bahwa UAT menyediakan cakupan yang cukup dari semua skenario. Setiap kasus UAT menggambarkan dalam bahasa yang

sederhana langkah-langkah yang tepat yang akan diambil untuk menguji sesuatu.

2.9 Teknik Sampling

Untuk menentukan sampel dari populasi digunakan perhitungan maupun acuan tabel yang dikembangkan para ahli. Secara umum, untuk penelitian korelasional jumlah sampel minimal untuk memperoleh hasil yang baik adalah 30, sedangkan dalam penelitian eksperimen jumlah sampel minimum 15 dari masing-masing kelompok dan untuk penelitian survey jumlah sampel minimum adalah 100. Untuk menentukan Sample didapatkan rumus dari [RID-05] yaitu :

$$N = \frac{n}{n(d^2)+1} \dots (2.9)$$

Keterangan : N = sampel

n = populasi

d = nilai presisi

