

BAB III

PERMASALAHAN PERUSAHAAN

3.1 Analisa Permasalahan Perusahaan

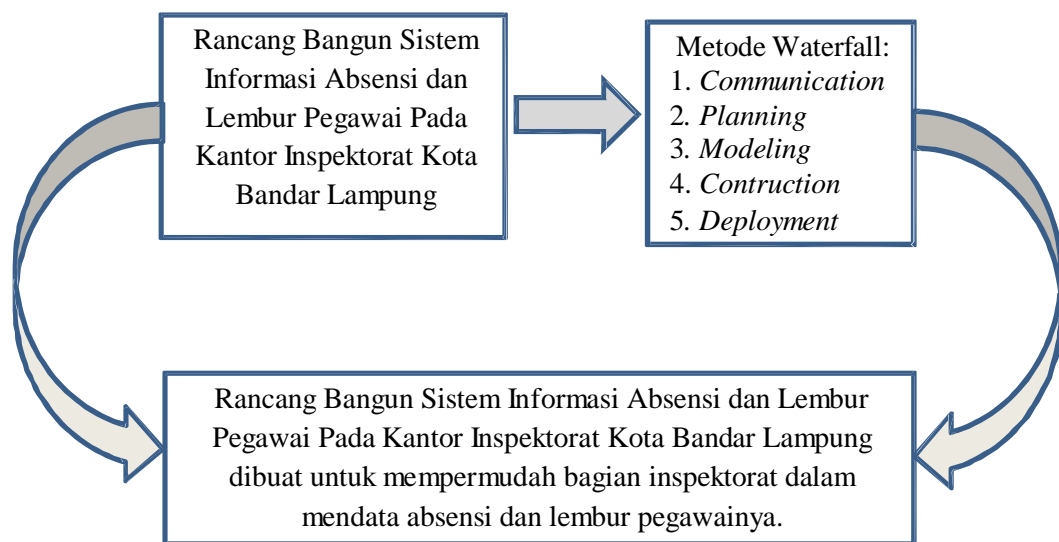
3.1.1 Temuan Masalah

Proses *entry* data absensi yang selama ini dilakukan secara manual tidak lagi dibutuhkan, karena semuanya dilakukan secara komputerisasi. Dengan demikian pengolahan data pegawai dilakukan secara cepat, akurat, dan mudah.

3.1.2 Perumusan Masalah

1. Bagaimana rancang bangun aplikasi absensi karyawan *outsourcing* yang melakukan lembur pada kantor Inspektorat kota Bandar Lampung?
2. Bagaimana membuat laporan absensi karyawan *outsourcing* yang melakukan lembur?

3.1.3 Kerangka Pemecahan Masalah



Gambar 3.1 Kerangka pemecahan masalah

3.2 Landasan Teori

3.2.1 Konsep Dasar Sistem Informasi

Konsep dasar dari sistem informasi terbagi atas dua pengertian. Pertama adalah sistem, dan yang kedua adalah sistem informasi itu sendiri.

3.2.2 Sistem

Menurut Herlambang (2016:116), definisi sistem dapat dibagi menjadi dua pendekatan, yaitu pendekatan secara prosedur dan pendekatan secara komponen. Berdasarkan pendekatan prosedur, sistem didefinisikan sebagai kumpulan dari beberapa prosedur yang mempunyai tujuan tertentu. Sedangkan berdasarkan pendekatan komponen, sistem merupakan kumpulan dari komponen-komponen yang saling berkaitan untuk mencapai tujuan tertentu.

Dalam perkembangan sistem yang ada, sistem dibedakan menjadi dua jenis, yaitu sistem terbuka dan sistem tertutup. Sistem terbuka merupakan sistem yang dihubungkan dengan arus sumber daya luar dan tidak mempunyai elemen pengendali. Sedangkan sistem tertutup tidak mempunyai elemen pengontrol dan dihubungkan pada lingkungan sekitarnya.

3.2.3 Sistem Informasi

Menurut Herlambang (2016:121), data adalah fakta-fakta atau kejadian-kejadian yang dapat berupa angka-angka atau kode-kode tertentu. Data masih belum mempunyai arti bagi penggunanya. Untuk dapat mempunyai arti data diolah sedemikian rupa sehingga dapat digunakan oleh penggunanya. Hasil pengolahan data inilah yang disebut sebagai informasi. Secara ringkas, informasi adalah data yang telah diolah dan mempunyai arti bagi penggunanya. Sehingga sistem informasi dapat didefinisikan sebagai prosedur-prosedur yang digunakan untuk mengolah data sehingga dapat digunakan oleh penggunanya.

3.3 Rancangan Program

Analisis sistem dilakukan dengan tujuan untuk dapat mengidentifikasi dan mengevaluasi permasalahan yang terjadi dan kebutuhan yang diharapkan, sehingga dapat diusulkan perbaikannya. Perancangan sistem merupakan

penguraian suatu sistem informasi yang utuh kedalam bagian komputerisasi yang dimaksud, mengidentifikasi dan mengevaluasi permasalahan, menentukan kriteria, menghitung konsistensi terhadap kriteria yang ada, serta mendapatkan hasil atau tujuan dari masalah tersebut serta mengimplementasikan seluruh kebutuhan operasional dalam membangun sebuah aplikasi Menurut Kendall (2017:7), analisa dan perancangan sistem dipergunakan untuk menganalisis, merancang, dan mengimplementasikan peningkatanpeningkatan fungsi bisnis yang dapat dicapai melalui penggunaan sisteminformasi komputerisasi. Berikut ini adalah proses dalam analisis dan perancangan sistem:

1. *Entity Relationship Diagram*

Entity Relationship Diagram (ERD) adalah gambaran pada sistem dimana didalamnya terdapat hubungan antara *entity* beserta relasinya. *Entity* merupakan sesuatu yang ada dan terdefiniskan di dalam suatu organisasi, dapat abstrak dan nyata. Untuk setiap *entity* biasanya mempunyai atribut yang merupakan ciri *entity* tersebut. Relasi adalah hubungan antara *entity* yang berfungsi sebagai hubungan yang mewujudkan pemetaan antar *entity*. Menurut Marlinda (2018:28), *atribute* adalah kolom disebuah relasi. Macammacam *atribute* yaitu:

a. *Simple attribute*

Atribute ini merupakan *atribute* yang unik dan tidak dimiliki oleh *atribute* lainnya, misalnya *entity* mahasiswa yang *atribute*-nya NIM.

b. *Composite attribute*

Composite attribute adalah *atribute* yang memiliki dua nilai harga, misalnya nama besar (nama keluarga) dan nama kecil (nama asli).

c. *Single value attribute*

Atribute yang hanya memiliki satu nilai harga, misalnya *entity* mahasiswa dengan *atribute*-nya Umur (tanggal lahir).

d. *Multi value attribute*

Multi value attribute adalah *atribute* yang banyak memiliki nilai harga, misalnya *entity* mahasiswa dengan *atribute*-nya pendidikan (SD, SMP, SMA).

e. *Null value attribute*

Null value attribute adalah *attribute* yang tidak memiliki nilai harga, misalnya *entity* tukang becak dengan *attribute*-nya pendidikan (tanpa memiliki ijazah). *Entity Relationship Diagram* ini diperlukan agar dapat menggambarkan hubungan antar *entity* dengan jelas, dapat menggambarkan batasan jumlah *entity* dan partisipasi antar *entity*, mudah dimengerti pemakai dan mudah disajikan oleh perancang database. Untuk itu *Entity Relationship Diagram* dibagi menjadi dua jenis model, yaitu:

- *Conceptual Data Model*

Conceptual Data Model (CDM) adalah jenis model data yang menggambarkan hubungan antar tabel secara konseptual.

- *Physical Data Model*

Physical Data Model (PDM) adalah jenis model data yang menggambarkan hubungan antar tabel secara fisikal.

2. *Data Flow Diagram*

Pada tahap ini, penggunaan notasi dapat dapat membantu komunikasi dengan pemakai/user sistem untuk memahami sistem tersebut secara logika. Diagram yang menggunakan notasi-notasi untuk menggambarkan arus dari data sistem ini dikenal dengan nama diagram arus data (*Data flow diagram*). DFD berfungsi untuk menggambarkan proses aliran data yang terjadi didalam sistem dari tingkat yang tertinggi sampai yang terendah, yang memungkinkan untuk melakukan dekomposisi, mempartisi atau membagi sistem kedalam bagian-bagian yang lebih kecil dan yang lebih sederhana. DFD fokus pada aliran data dari dan ke dalam sistem serta memproses data tersebut (Kendall, 2017 : 241).

Simbol-simbol dasar dalam DFD antara lain:

a. *Eksternal Entity*

Suatu *eksternal entity* atau entitas merupakan orang, kelompok, departemen, atau sistem lain di luar sistem yang dibuat dapat menerima atau memberikan informasi atau data ke dalam sistem

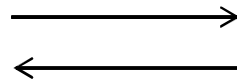
yang dibuat. Gambar 3.2 merupakan simbol entitas dalam DFD dalam model Gene dan Sarson.



Gambar 3.2 Simbol *Eksternal Entity*

b. *Data Flow*

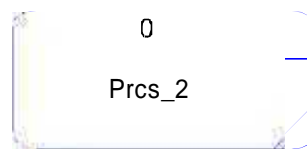
Data flow atau aliran data disimbolkan dengan tanda panah. *Data Flow* menunjukkan arus data atau aliran data yang menghubungkan dua proses atau entitas dengan proses. Gambar 3.2 merupakan simbol *Data Flow*.



Gambar 3.3 Simbol *Data Flow*

c. *Process*

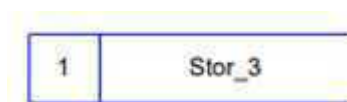
Suatu *Proses* dimana beberapa tindakan atau sekelompok tindakan dijalankan. Gambar 3.4 merupakan simbol *process*.



Gambar 3.4 Simbol *Process*

d. *Data Store*

Data store adalah simbol yang digunakan untuk melambangkan proses penyimpanan data. Gambar 3.5 merupakan simbol file penyimpanan/*data store*.



Gambar 3.5 Simbol *Data Store*

3.4 Konsep Dasar Basis Data

Konsep dasar basis data dibagi menjadi tiga bagian, yaitu sistem basis data, *database*, dan *database management system*.

3.4.1 Sistem Basis Data

Menurut Marlinda (2018:1), sistem basis data adalah suatu sistem menyusun dan mengelola record-record menggunakan komputer untuk menyimpan atau merekam serta memelihara dan operasional lengkap sebuah organisasi/perusahaan sehingga mampu menyediakan informasi optimal yang diperlukan pemakai untuk proses mengambil keputusan.

Pada sebuah sistem basis data terdapat komponen-komponen utama yaitu perangkat keras (*hardware*), sistem operasi (*operating system*), basis data (*database*), sistem (aplikasi atau perangkat lunak) pengelola basis data (DBMS), pemakai (*user*), dan aplikasi (perangkat lunak) lain (bersifat operasional).

Keuntungan sistem basis data adalah:

1. Mengurangi kerangka pandata, yaitu data yang sama disimpan dalam berkas data yang berbeda-beda sehingga update dilakukan berulang-ulang.
2. Mencegah ketidak konsistenan.
3. Keamanan data dapat terjaga, yaitu data dapat dilindungi dari pemakai yang tidak berwenang.
4. Integritas dapat dipertahankan.
5. Data dapat digunakan bersama-sama.
6. Menyediakan recovery.
7. Memudahkan penerapan standarisasi.
8. Data bersifat mandiri.
9. Keterpaduan data terjaga, memelihara keterpaduan data berarti data harus akurat.

Kerugian sistem basis data adalah:

1. Diperlukan tempat penyimpanan yang besar.
2. Diperlukan tenaga yang terampil dalam mengolah data.
3. Perangkat lunaknya mahal.

4. Kerusakan sistem basis data dapat mempengaruhi departemen yang terkait.

3.4.2 Database

Menurut Yuswanto (2015:2), *database* merupakan sekumpulan data yang berisi informasi yang saling berhubungan. Pengertian ini sangat berbeda antara *database* Relasional dan Non Relasional. Pada *database* Non Relasional, sebuah data base hanya merupakan sebuah file. Menurut Marlinda (2004:1), *database* adalah suatu susunan/kumpulan data operasional lengkap dari suatu organisasi/perusahaan yang diorganisir/dikelola dan disimpan secara terintegrasi dengan menggunakan metode tertentu menggunakan komputer sehingga mampu menyediakan informasi optimal yang diperlukan pemakainya.

Penyusunan satu *database* digunakan untuk mengatasi masalah-masalah pada penyusunan data yaitu redundansi dan inkosistensi data, kesulitan pengaksesan data, isolasi data untuk standartisasi, multiple user, masalah keamanan, masalah integrasi, dan masalah data independence.

3.4.3 Database Management System

Menurut Marlinda (2018:6), *Database Management System* (DBMS) merupakan kumpulan file yang saling berkaitan dan program untuk pengelolanya. Basis Data adalah kumpulan datanya, sedang program pengelolanya berdiri sendiri dalam suatu paket program yang komersial untuk membaca data, menghapus data, dan melaporkan data dalam basis data. Bahasa-bahasa yang terdapat dalam DBMS adalah:

1. Data Definition Language

Pola skema basis data dispesifikasikan dengan satu set definisi yang diekspresikan dengan satu bahasa khusus yang disebut DDL. Hasil kompilasi perintah DDL adalah satu set tabel yang disimpan di dalam file khusus yang disebut data dictionary/directory.

2. Data Manipulation Language

Bahasa yang memperbolehkan pemakai mengakses atau memanipulasi data sebagai yang diorganisasikan sebelumnya model data yang tepat.

3. Query

Pernyataan yang diajukan untuk mengambil informasi. Merupakan bagian DML yang digunakan untuk pengambilan informasi. DBMS memiliki fungsi sebagai berikut:

a. Data Definition

DBMS harus dapat mengolah pendefinisian data.

b. Data Manipulation

DBMS harus dapat menangani permintaan-permintaan dari pemakai untuk mengakses data.

c. Data Security dan Integrity

DBMS dapat memeriksa security dan integrity data yang didefinisikan oleh DBA.

d. Data Recovery dan Concurrency

- DBMS harus dapat menangani kegagalan-kegagalan pengaksesan basis data yang dapat disebabkan oleh kesalahan sistem, kerusakan disk, dan sebagainya.
- DBMS harus dapat mengontrol pengaksesan data yang konkuren yaitu bila satu data diakses secara bersama-sama oleh lebih dari satu pemakai pada saat yang bersamaan.

e. Data Dictionary

DBMS harus menyediakan data dictionary.

3.5 Interaksi Manusia dan Komputer

Menurut Rizky (2018:4), Interaksi Manusia dan Komputer (IMK) adalah sebuah disiplin ilmu yang mempelajari desain, evaluasi, implementasi dari sistem komputer interaktif untuk dipakai oleh manusia, beserta studi tentang faktor-faktor utama dalam lingkungan interaksinya.

Deskripsi lain dari IMK adalah suatu ilmu yang mempelajari perencanaan dan desain tentang cara manusia dan komputer saling bekerja sama, sehingga manusia dapat merasa puas dengan cara yang paling efektif. Dikatakan juga bahwa sebuah desain antar muka yang ideal adalah yang mampu memberikan kepuasan terhadap manusia sebagai pengguna dengan faktor kapabilitas serta keterbatasan yang

terdapat dalam sistem. Pada implementasiannya, IMK dipengaruhi berbagai macam faktor antara lain organisasi, lingkungan, kesehatan, pengguna, kenyamanan, antarmuka, kendala dan produktifitas.

3.6 ORACLE

Merupakan software database yang dibuat oleh *Oracle Corporation* dan banyak dipakai diperusahaan-perusahaan besar diseluruh dunia. Software ini juga banyak diminati oleh para konsultan pembuat aplikasi yang berkaitan dengan database, karena sistem keamanannya yang sangat handal. Database bukanlah sebuah file tetapi merupakan sebuah konsep logis yang berisi sekumpulan objek-objek yang berhubungan. Misalnya sebuah data base berisi data, struktur *database*, *index*, *sekuritas view*, dan *stored procedure*.

ORACLE memberikan bahasa dan antarmuka yang baik untuk pemrograman dan komunikasi pada server. ORACLE adalah bahasa pemrograman server yang merupakan superset dari ANSI. ANSI mendefinisikan empat perintah dasar untuk memanipulasi data yaitu SELECT, INSERT, UPDATE, DELETE dan sejumlah perintah untuk mendefinisikan struktur database. ORACLE menambahkan beberapa hal pada ANSI penambahan tersebut adalah konstruksi pemrograman yang memungkinkan pemakaian *stored procedure* untuk mengubah data dan trigger yang akan dijalankan, karena terjadi ivent tertentu.

Objek-objek dalam ORACLE adalah:

1. Tabel

Objek yang berisi tipe-tipe datadan data mentah.

2. Kolom

Sebuah table berisi kolom-kolom untuk menampung data. Kolom mempunyai sebuah tipe dan nama yang unik.

3. Tipe Data

Sebuah kolom mempunyai sebuah tipe data. Tipe-tipe yang dapat di pilih adalah karakter, numerik, tanggal, boolean.

4. *Stored Procedure*

Merupakan perintah ANSI yang membentuk makro. Dengan menjalankan *stored procedure* berarti pengembang menjalankan perintahperintah ANSI didalam sebuah *procedure*.

5. *Trigger*

Merupakan *stored procedure* yang diaktifkan pada saat ditambahkan, diubah, atau dihapus dari database. *Trigger* dipakai untuk menjamin atau integritas didalam database.

6. *Rule*

Rule diberlakukan pada kolom sehingga data yang dimasukkan harus sesuai dengan aturan.

7. *Primary Key*

Berfungsi menjamin setiap baris data unik,dapat dibedakan daridata lain.

8. *Foreign Key*

Foreign Key adalah kolom-kolom yang mengacu pada *Primary Key* atau konstrain unik pada tabel lain. *Primary Key* dan *Foreign Key* dipakai untuk menghubungkan sebuah tabel dengan tabel yang lain.

9. *Konstrain*

Konstrain adalah mekanisme integritas data yang berbasis server dan diimplementasikan oleh sistem.

10. *Default*

Default dinyatakan pada *field* (kolom) sehingga jika kolom tersebut tidak diisidata,maka diisi dengan nilai default.

11. *View*

View adalah query yang memakai beberapa tabel, dan disimpan di dalam database. *View* dapat memilih beberapa kolom dari sebuah tabel atau menghubungkan beberapa tabel. *View* juga dapat dipakai untuk menjaga keamanan data.

12. *Index*

Index berfungsi membantu mengorganisasi data sehingga query menjadi lebih cepat. ORACLE memakai sebuah tipe database yang dinamakan *database relation*. *Database relation* adalah database yang

mengorganisasikan data dalam bentuk table. Table dibentuk dengan mengelompokkan data yang mempunyai subjek yang sama. Table berisi baris dan kolom informasi. Table tabel dapat saling berhubungan jika diinginkan.

3.7 Testing dan Implementasi Sistem

Menurut standart ANSI/IEEE 1059, Testing adalah proses menganalisa suatu entitas software untuk mendeteksi perbedaan antara kondisi yang ada dengan kondisi yang diinginkan (defects/error/bugs) dan mengevaluasi fitur-fitur dari entitas software.

Menurut Romeo (2017:3), Testing software adalah proses mengopersikan software dalam suatu kondisi yang dikendalikan untuk:

1. Verifikasi Apakah telah berlaku sebagaimana yang ditetapkan (menurut spesifikasi)?
2. Mendeteksi Error
Apakah telah bebas dari kesalahan software?
3. Validasi Apakah spesifikasi yang ditetapkan telah memenuhi keinginan atau kebutuhan pengguna yang sebenarnya?

Menurut Romeo (2017:33), Test Case merupakan tes yang dilakukan berdasarkan pada suatu inisialisasi, masukan kondisi ataupun hasil yang telah ditentukan sebelumnya. Metode testing ini dibagi menjadi dua, yaitu:

3.7.1 White Box Testing

White Box Testing atau glass box testing atau clear box testing adalah suatu metode test case yang menggunakan struktur kendali dari desain prosedural. Metodedesain test case ini dapat menjamin:

1. Semua jalur (path) yang independen/terpisah dapat dites setidaknya sekali tes.
2. Semua logika keputusan dapat dites dengan jalur yang salah atau jalur yang benar.
3. Semua loop dapat dites terhadap batasannya dan ikatan operasional.
4. Semua struktur internal data dapat dites untuk memastikan validasinya.

3.7.2 *Black Box Testing*

Black box testing atau behavioral testing atau *specification-based testing*, *input/output testing* atau *functional testing* dilakukan tanpa sepengetahuan detail struktur internal dari sistem atau komponen yang dites. *Black box testing* berfokus pada kebutuhan fungsional pada software, berdasarkan spesifikasi kebutuhan dari software. Menggunakan *black box testing*, perancang software dapat menggunakan sekumpulan kondisi masukan yang dapat secara penuh memeriksa keseluruhan kebutuhan fungsional pada suatu program. Kategori error dapat diketahui melalui *black box testing*, antara lain:

1. Fungsi yang hilang atau tidak benar.
2. Error dari antar muka.
3. Error dari struktur data atau akses *eksternal database*.
4. Error dari kinerja atau tingkah laku.