



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

LANDASAN TEORI

2.1 Teori Umum

2.1.1 Data

Menurut Hoffer (2009:46) data disebut sebagai fakta tentang benda-benda dan peristiwa yang dapat direkam dan disimpan di dalam media komputer. Contohnya: di dalam *database* seorang *seller*, terdapat data yang mencakup fakta-fakta seperti nama pelanggan, alamat, dan nomor telepon. Jenis data tersebut disebut *Structured Data*. Tipe *Structured Data* yang paling penting adalah numerik, *array*, dan tanggal. *Structured Data* disimpan dalam bentuk tabel (tabel, hubungan, *array*, *spreadsheet*, dll) paling sering ditemukan dalam *database* dan gudang yang masih bersifat tradisional.

Menurut Connolly (2010:70), Data merupakan bagian terpenting dari komponen suatu *database*. Connolly mengamati bahwa data bertindak sebagai jembatan antara komponen mesin dan komponen manusia. *Database* berisi kedua data *operasional* dan *metadata*, "Data tentang data." Struktur *database* disebut skema.

Menurut Stair (2006:05) data terdiri dari fakta-fakta mentah, seperti nama dan jumlah jam kerja dalam seminggu karyawan, nomor bagian persediaan, atau pesanan penjualan. Beberapa jenis data dapat digunakan untuk mewakili fakta. Ketika fakta diorganisir atau diatur dengan cara yang bermanfaat, mereka menjadi informasi.

Menurut Turban (2009:197) Data adalah sebuah kumpulan fakta yang bisa diperoleh dari pengalaman, pengamatan, atau percobaan. Data bisa terdiri dari angka, kata, gambar, dan sebagainya sebagai pengukuran dari satu set variabel. Data sering dipandang sebagai tingkat terendah abstraksi dari mana informasi dan pengetahuan berasal. Pada tingkat tertinggi dari abstraksi, data dapat diklasifikasikan sebagai *categorical* data atau *numeric* data. *Categorical* data dapat dibagi menjadi *Nominal* data atau *Ordinal* data, sedangkan *numeric* data dapat dibagi lagi menjadi *Interval* Data atau *Rasio* Data.

2.1.2 Database

Database adalah sebuah tempat penyimpanan yang besar dimana terdapat kumpulan data yang tidak hanya berisi data operasional tetapi juga deskripsi data. Seperti yang disampaikan oleh Connolly (2010:65), bahwa *database* adalah kumpulan data yang saling terhubung secara logis dan deskripsi dari data tersebut, dirancang untuk menemukan informasi yang dibutuhkan oleh sebuah organisasi. Dalam merancang *database*, salah satu hal yang perlu diperhatikan adalah efisiensi. Banyaknya data yang redundansi dapat mengurangi efisiensi pada *database* sehingga perlu dilakukan normalisasi. *Database* ini digunakan tidak hanya oleh satu orang maupun satu departemen, *database* dapat digunakan oleh seluruh departemen dalam perusahaan. *Database* ini akan menjadi sumber data yang digunakan secara bersama dalam perusahaan.

Menurut Hoffer (2009:46) mendefinisikan *database* sebagai koleksi yang terorganisir dari data secara logis. Sebuah *database* terdapat berbagai macam ukuran dan kompleksitas. Sebagai contoh, seorang penjual dapat menjaga *database*

kecil kontak pelanggan pada komputer laptop-nya yang terdiri dari beberapa *megabyte* data. Sebuah perusahaan besar mungkin membangun sebuah *database* besar yang terdiri dari beberapa *terabyte* data pada komputer *mainframe* besar yang digunakan untuk aplikasi pendukung keputusan.

Menurut Turban (2009:90-91) *Database* adalah kumpulan data yang saling berhubungan yang dibuat untuk memenuhi kebutuhan dan struktur organisasi yang dapat digunakan oleh lebih dari satu orang untuk lebih dari satu aplikasi. Dalam banyak kasus DSS, data yang diimpor dari data *warehouse* atau warisan sistem *database mainframe* melalui *web server database*. Untuk aplikasi DSS lain, *database* khusus terkonstruksi, sebagai *database* yang diperlukan. Beberapa *database* dapat digunakan dalam satu aplikasi DSS, tergantung pada sumber data.

2.1.3 Informasi

Menurut Hoffer (2009:47) Informasi adalah data yang telah diproses sedemikian rupa sehingga meningkatkan pengetahuan dari orang yang menggunakan tersebut dinamakan informasi.

Menurut Stair (2006:05) Informasi Adalah kumpulan fakta diatur sedemikian rupa sehingga mereka memiliki nilai tambahan di luar nilai fakta-fakta itu sendiri. Sebagai contoh, seorang manajer tertentu mungkin menemukan pengetahuan dari total penjualan bulanan untuk lebih cocok nya untuk tujuan dari jumlah penjualan untuk perwakilan penjualan individu. Memberikan informasi kepada pelanggan juga dapat membantu perusahaan meningkatkan pendapatan dan keuntungan.

Menurut O'Brien (2007:05) Informasi merupakan data yang telah diproses, terorganisir, dan terintegrasi untuk memberikan pemahaman. Sistem informasi yang dirancang untuk membantu manajer menganalisis data dan membuat keputusan. Dari sudut pandang seorang pembuat keputusan, tantangannya adalah bahwa Anda mungkin tidak tahu di depan waktu informasi yang Anda butuhkan, sehingga sulit untuk menentukan data apa yang Anda butuhkan untuk mengumpulkan.

2.1.4 Sistem

Pengertian sistem menurut O'Brien (2005:30) adalah sekelompok komponen yang saling berhubungan, bekerja bersama untuk mencapai tujuan bersama dengan menerima masukan (input) serta menghasilkan suatu keluaran (output) yang dikelola dalam proses transformasi yang teratur. Oleh karena itu, sistem mempunyai 3 komponen dasar yang saling berinteraksi atau fungsi dasar, yaitu:

- Masukan, yaitu bagian yang meliputi pengambilan elemen yang masuk ke dalam system untuk diproses. Contoh: bahan mentah, energi, data, dan sumber daya manusia harus bisa mengatur prosesnya
- Proses, yaitu bagian yang meliputi perubahan dari input menjadi output. Contoh: proses manufaktur, kalkulasi matematika.
- Keluaran, yaitu bagian yang meliputi elemen yang dihasilkan dari proses transformasi untuk mencapai tujuan yang diinginkan. Contoh: produk jadi.

Pengertian sistem menurut O'Brien (2009:26) yaitu seperangkat komponen yang saling terkait, dengan batas yang jelas, bekerja sama untuk mencapai

seperangkat tujuan dengan menerima *input* dan menghasilkan *output* dalam proses transformasi yang terorganisir. Banyak contoh dari sistem dapat ditemukan dalam ilmu fisika dan biologi, teknologi modern, dan dalam masyarakat manusia. Dengan demikian, kita bisa bicara dari sistem fisik kilang minyak, dan sistem sosial ekonomi dari sebuah organisasi bisnis.

Menurut Stair (2006:08-09) Adalah seperangkat elemen atau komponen yang berinteraksi untuk mencapai tujuan. Unsur-unsur itu sendiri dan hubungan di antara mereka menentukan bagaimana sistem bekerja. Sistem memiliki input, mekanisme pengolahan, keluaran, dan umpan balik. Sebagai contoh, pertimbangkan sebuah cuci mobil terkomputerisasi. Jelas, masukan nyata bagi proses ini adalah mobil yang kotor, air, dan berbagai bahan pembersih yang digunakan. Waktu, energi, keterampilan, dan pengetahuan juga dibutuhkan sebagai masukan ke sistem. Waktu dan energi yang dibutuhkan untuk mengoperasikan sistem. Keterampilan adalah kemampuan untuk berhasil mengoperasikan sprayer cair, berbusa kuas, dan perangkat pengering udara. Pengetahuan ini digunakan untuk menentukan langkah-langkah dalam operasi cuci mobil dan urutan langkah-langkah yang dijalankan. Mekanisme pengolahan terdiri memilih mana pilihan pembersih yang Anda inginkan (hanya mencuci, mencuci dengan lilin, cuci dengan lilin, dan tangan dll kering) dan mengkomunikasikan bahwa untuk operator cuci mobil. Perhatikan bahwa ada mekanisme umpan balik (penilaian Anda tentang bagaimana bersih mobil). *Liquid sprayer* menembak air jernih, sabun cair, atau lilin mobil tergantung di mana mobil Anda sedang dalam proses dan pilihan yang Anda pilih. Outputnya adalah mobil bersih. Penting untuk dicatat bahwa unsur-

unsur atau komponen dari sistem (*sprayer* cair, berbusa kuas, dan pengering udara) independen berinteraksi untuk menciptakan mobil yang bersih.

2.1.5 Sistem Informasi

Menurut Stair (2006:04) Sistem Informasi merupakan satu set komponen yang saling terkait yang mengumpulkan, memanipulasi, menyimpan, dan menyebarkan data dan informasi serta menyediakan mekanisme umpan balik untuk memenuhi tujuan. Bantuan *feedback* dapat membantu organisasi mencapai tujuan mereka, seperti meningkatkan keuntungan atau meningkatkan layanan pelanggan. Contoh pelanggan di pompa bensin menggesekkan tag *keychain* pada pembaca yang mengirimkan informasi ke jaringan untuk memverifikasi profil dan kredit informasi pelanggan. Terminal memproses transaksi mencetak tanda terima, dan kartu kredit / cek *bill* pelanggan secara terkomputerisasi.

Menurut Hoffer (2009:506) Sistem informasi dirancang untuk mendukung pengambilan keputusan didasarkan pada titik *Historical Point-in-time* dan *prediction data*. Sistem informasi juga dirancang untuk query kompleks atau aplikasi data mining. Contoh sistem informasi adalah analisis trend penjualan, segmentasi pelanggan, dan perencanaan sumber daya manusia.

Menurut O'Brien (2009:04) Sistem Informasi Dapat berupa kombinasi terorganisir orang, hardware, software, jaringan komunikasi, sumber daya data, dan kebijakan dan prosedur yang menyimpan, mengambil, mengubah, dan menyebarkan informasi dalam sebuah organisasi. Orang bergantung pada sistem informasi modern untuk berkomunikasi dengan satu sama lain menggunakan berbagai perangkat fisik (*hardware*), instruksi pemrosesan informasi dan prosedur

(*software*), saluran komunikasi (jaringan), dan data yang disimpan (sumber data). Meskipun sistem informasi saat ini dianggap memiliki sesuatu yang berkaitan dengan komputer, kami telah menggunakan sistem informasi sejak awal peradaban. Bahkan hari ini kita menggunakan rutin sistem informasi yang tidak ada hubungannya dengan komputer.

Menurut O.Brien (2007:582) Sistem Informasi adalah Sebuah koleksi dari *hardware*, *software*, data, dan orang-orang yang dirancang untuk mengumpulkan, memproses, dan mendistribusikan data seluruh tambahan organisasi. ini untuk mendukung pengambilan keputusan, koordinasi dan kontrol sistem informasi juga dapat membantu manajer dan pekerja menganalisis masalah, memvisualisasikan subjek yang kompleks, dan menciptakan produk baru.

2.2 Teori Khusus

2.2.1 Produksi

Produksi menurut Drs. Bambang Prishardoyo, M.Si (2005) merupakan kegiatan yang dilakukan untuk memenuhi kebutuhan manusia dengan menghasilkan barang atau meningkatkan nilai guna suatu barang dan jasa.

2.2.2 Analisis dan Perancangan Sistem Informasi

Menurut Jogiyanto (2001:129), analisis sistem adalah penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan, dan hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat di usulkan perbaikan perbaikannya.

Dari definisi diatas, maka analisa sistem adalah suatu perencanaan sekaligus pembuatan sketsa dari element yang terpisah namun saling berintegrasi dalam satu kesatuan.

2.2.2.1 Perancangan Sistem

Perancangan sistem adalah Menentukan bagaimana mencapai sasaran yang ditetapkan yang melibatkan pembentukan (*configuring*) perangkat lunak dan komponen perangkat keras sistem dimana setelah pemasangan sistem akan memenuhi spesifikasi yang dibuat pada akhir fase analisis sistem.”George M.Scott,(2001.534).

2.2.2.2 Data Flow Diagram (DFD)

Data flow diagram adalah *diagram* yang menggunakan notasi untuk menggambarkan arus dari data sistem yang penggunaanya sangat membantu untuk memahami sistem secara logika, terstruktur dan jelas. DFD merupakan alat bantu dalam menggambarkan atau menjelaskan sistem yang sedang berjalan.

Menurut Jogiyanto Hartono (2005) ada beberapa simbol digunakan pada DFD untuk mewakili :

1. Kesatuan Luar (*External Entity*)

Kesatuan luar (*external entity*) merupakan kesatuan (*entity*) di lingkungan luar sistem yang dapat berupa orang, organisasi, atau sistem lain yang berada pada lingkungan luarnya yang memberikan *input* atau menerima *output* dari sistem.

2. Arus Data (*Data Flow*)

Arus Data (*data flow*) di DFD diberi simbol suatu panah. Arus data ini mengalir di antara proses, simpan data dan kesatuan luar. Arus data ini





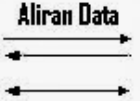
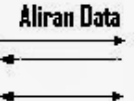


menunjukkan arus dari data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem.

3. Proses (*Process*)

Proses (*process*) menunjukkan pada bagian yang mengubah input menjadi output, yaitu menunjukkan bagaimana satu atau lebih input diubah menjadi beberapa output. Setiap proses mempunyai nama, nama dari proses ini menunjukkan apa yang dikerjakan proses.

4. Simpanan Data (*Data Store*)

Data Store merupakan simpanan dari data yang dapat berupa suatu file atau database pada sistem komputer.

Gane/Sarson	Yourdon/De Marco	Keterangan
		Entitas eksternal, dapat berupa orang/unit terkait yang berinteraksi dengan sistem tetapi diluar sistem
		Orang, unit yang mempergunakan atau melakukan transformasi data. Komponen fisik tidak diidentifikasi.
		Aliran data dengan arah khusus dari sumber ke tujuan
		Penyimpanan data atau tempat data direfer oleh proses.

Gambar 2.1. Simbol-simbol dalam DFD (Jogiyanto, 2005)

2.2.2.3 Entity Relationship Modeling

Menurut Connolly (2010:371) *entity-relationship modeling* merupakan permodelan yang berguna untuk memberikan pemahaman yang tepat terhadap data dan penggunaannya di dalam suatu perusahaan. Model ini menggunakan pendekatan *top-down* dalam perancangan basis data yang di mulai dengan mengidentifikasi data penting yang disebut *entity* dan relasi antar data yang akan direpresentasikan ke dalam model. Kemudian ditambahkan detail-detail lebih seperti informasi yang akan dicari mengenai *entities* dan *relationship* yang disebut dengan atribut dan *constraints* pada *entity*, atribut, dan *relationship*.

a. *Entity Type*

Menurut Connolly (2010:372) *entity type* merupakan sekumpulan objek di dunia yang memiliki *property* yang sama. *Entity* direpresentasikan dalam bentuk diagram berupa persegi panjang berlabel nama dari *entity*.

b. *Relationship Types*

Menurut Connolly (2010:372) *relationship types* merupakan suatu hubungan antar *entity types*. *Relationship Types* direpresentasikan dalam bentuk diagram berupa garis lurus yang menghubungkan dua buah *entity types*, ditandai dengan nama dari relasi tersebut. Pada umumnya, relasi dinamai dengan kata kerja.

Menurut Connolly (2010:376) dalam *relationship types* terdapat *degree of relationship type*. *Degree of relationship type* merupakan jumlah tipe entitas yang terkait dalam *relationship*. Entitas yang terkait dalam *relationship* disebut dengan *participants*. Jadi, *degree* dari suatu *relationship* menunjukkan banyaknya entitas

yang tergabung dalam suatu *relationship*. Terdapat 3 jenis *degree of relationship*, yaitu:

- *Binary Relationship*

Binary Relationship merupakan *relationship* yang mempunyai dua *degree*.

- *Ternary Relationship*

Ternary Relationship merupakan *relationship* yang mempunyai tiga *degree*.

- *Quarternary Relationship*

Quarternary Relationship merupakan *relationship* yang mempunyai empat *degree*.

c. *Attributes*

Menurut Connolly (2010:379) *atribute* adalah *property* suatu entitas atau jenis relasi. atribut domain adalah himpunan nilai yang diperbolehkan untuk satu atau lebih atribut. Atribut dapat diklasifikasikan menjadi lima, yaitu:

1) *Simple Attribute*

Simple attribute adalah sebuah atribut yang terdiri dari komponen tunggal yang mempunyai keberadaan bebas dan tidak dapat dibagi menjadi bagian yang lebih kecil dikenal dengan nama *atomic attribute*.

2) *Composite Attribute*

Composite attribute adalah atribut yang terdiri dari beberapa komponen, dimana masing-masing komponen mempunyai keberadaan yang bebas.

3) *Single-valued Attribute*

Single-valued attribute adalah atribut yang mempunyai nilai tunggal untuk setiap kejadian dari tipe *entity*.

4) *Multi-valued Attribute*

Multi-valued attribute adalah atribut yang mempunyai beberapa nilai untuk setiap kejadian dari tipe *entity*.

5) *Derived Attribute*

Derived attribute adalah atribut yang memiliki nilai yang dihasilkan dari satu atau sekelompok atribut yang berhubungan, dan tidak harus berasal dari satu entitas.

d. Key

Menurut Connolly (2010:381) *key* adalah sebuah *field* yang digunakan untuk mengidentifikasi satu atribut atau lebih secara unik mengidentifikasi setiap *record*.

Key yang sering digunakan yaitu:

- 1) *Candidate key*, merupakan kumpulan minimal dari atribut-atribut yang secara unik mengidentifikasikan suatu *entity*.
- 2) *Primary Key* merupakan *candidate key* yang dipilih untuk secara unik mengidentifikasikan suatu *entity*.
- 3) *Composite Key* merupakan *candidate key* yang terdiri atas dua atau lebih atribut.
- 4) *Alternate Key* merupakan *candidate key* yang tidak terpilih menjadi *primary key*, atau biasa disebut dengan *secondary key*.
- 5) *Foreign Key* merupakan sebuah *primary key* pada sebuah entitas yang digunakan pada entitas lainnya untuk mengidentifikasi sebuah *entity*.

e. Structural Constraints

Menurut Connolly (2010:385) memeriksa batasan tipe entitas yang mempunyai kesamaan dalam *relationship*. *Multiplicity* adalah jumlah atau *range* dari terjadinya yang mungkin dari suatu *entity* yang mungkin berhubungan dengan kejadian tunggal dari jenis entitas yang terkait melalui suatu hubungan tertentu.

Hubungan *Structural Constraints* dibagi menjadi 3 jenis, yaitu:

- 1) Hubungan *One-to-One* (1:1) adalah hubungan antara entitas yang satu dengan entitas lain mempunyai relasi hubungan satu entitas.
- 2) Hubungan *One-to-Many* (1:*) adalah hubungan antara entitas pertama yang mempunyai banyak relasi dengan entitas kedua yang mempunyai relasi satu entitas.
- 3) Hubungan *Many-to-Many* (*:*) adalah hubungan antara entitas pertama yang mempunyai relasi banyak dengan entitas kedua.

Cardinality adalah menjelaskan jumlah maksimum hubungan terjadinya yang mungkin untuk suatu entitas yang berpartisipasi dalam jenis hubungan tertentu.

Participation adalah menentukan semua atau hanya beberapa terjadinya entitas berpartisipasi dalam hubungan.

2.2.3 Normalisasi

Menurut Abdul Kadir (2009:116), “Normalisasi adalah suatu proses yang digunakan untuk menentukan pengelompokan atribut-atribut dalam sebuah relasi sehingga diperoleh relasi yang berstruktur baik”.

Normalisasi merupakan suatu pendekatan sistematis untuk meminimalkan redundansi data pada suatu database agar database tersebut dapat berkerja dengan optimal

2.2.3.1 Normalisasi database

Menurut Marlinda (2004:115) normalisasi merupakan "proses pengorganisasian file untuk menghilangkan grup elemen yang berulang menjadi tabel-tabel yang menunjukkan entity dan relasinya". Pada proses ini selalu dituju pada beberapa kondisi apakah ada kesulitan pada saat menambah (*insert*), menghapus (*delete*), mengubah (*update*), membaca (*retrieve*) pada suatu database. Normalisasi database terdiri dari banyak bentuk. Dalam ilmu basis data ada setidaknya 9 bentuk normalisasi yaitu 1NF, 2NF, 3NF, EKNF, BCNF, 4NF, 5NF, DKNF, dan 6NF. Bentuk bentuk normalisasi yang sering digunakan

1. Normal form

Data yang direkam dan dimasukan secara mentah dalam suatu table pada bentuk ini sangat mungkin terjadi inkonsistensi dan anomaly data

2. First normal form(1NF)

Bentuk normal yang pertama atau 1NF mensyaratkan beberapa kondisi dalam sebuah database. Berikut adalah fungsi dari bentuk normal pertama

1. Menghilangkan duplikasi kolom dari tabel yang sama.
2. Buat table terpisah untuk masing-masing kelompok data terkait dan mengidentifikasi setiap baris dengan kolom yang unik

3. Second normal form(2NF)

Syarat untuk menerapkan normalisasi bentuk kedua ini adalah data telah dibentuk dalam 1NF. Berikut adalah beberapa fungsi normalisasi 2NF.

- Menghapus beberapa subset data yang ada pada table dan menempatkan mereka pada table terpisah.
- Menciptakan hubungan antara tabel baru dan table lama dengan menciptakan foreign key.
- Tidak ada atribut dalam antara table yang secara fungsional bergantung pada candidate key table tersebut.
- *Third normal form (3NF)*

Normalisasi database dalam bentuk 3NF bertujuan untuk menghilangkan seluruh atribut atau field yang tidak berhubungan dengan primary key. Dengan demikian tidak ada ketergantungan transitif pada setiap kandidat key. Syarat dari bentuk normal ketiga atau 3NF adalah:

- ✓ Memenuhi semua persyaratan dari bentuk normal kedua.
- ✓ Menghapus kolom yang tidak tergantung pada primary key.

2.2.4 System Development Life Cycle (SDLC)

Definisi *System Development Life Cycle (SDLC)* menurut O'brein (2000, p383) adalah aplikasi penerapan dari penemuan permasalahan yang didapat dari pendekatan sistem menjadi pengembangan dari solusi sistem informasi terhadap masalah bisnis.

Tahap tahap dalam SDLC adalah

1. Investigasi sistem

Berisi studi kelayakan. Studi kelayakan digunakan untuk menentukan kemungkinan suksesnya proyek pengembangan sistem yang diajukan dan menentukan kelayakan ekonomi dan perilaku proyek.

2. Analisa sistem

Analisa sistem adalah penentuan permasalahan bisnis yang ingin di selesaikan oleh organisasi dengan sistem informasi. Tahap ini menentukan permasalahan bisnis

3. Perancangan sistem

Perancangan sistem menggambarkan bagaimana sistem dapat memenuhi tugasnya.

4. Pemrograman

Pemrograman meliputi translasi atau terjemahan dari perancangan spesifikasi ke dalam kode computer.

5. Testing

Testing bertujuan untuk melihat apakah kode computer akan memberikan hasil yang di inginkan dan di harapkan dalam kondisi tertentu. Testing di rancang untuk mendeteksi kesalahan kesalahan di dalam kode computer.

6. Implementasi

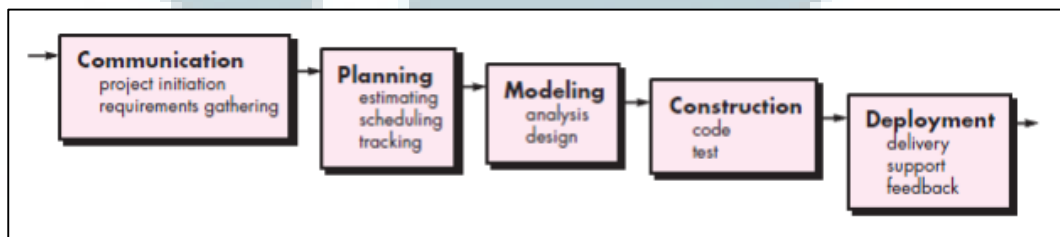
Implementasi dilakukan setelah sistem yang di buat berjalan dengan baik pada sesi testing.

7. Operasi

Setelah konversi, sistem baru di jalankan dalam periode waktu tertentu sampai sistem ini tidak lagi sesuai dengan kondisi tertentu. Setelah sistem baru di stabilkan akan dilakukan audit selama sistem di jalankan untuk memperlihatkan kemampuan sistem dan menentukan apakah sistem telah digunakan secara benar.

2.2.5 Model Waterfall

Metode rekayasa peranti lunak yang digunakan peneliti adalah Metode *waterfall*. Menurut Pressman (2010, p.39) *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun software. Berikut ini ada dua gambaran dari *waterfall* model. Fase-fase dalam model waterfall menurut referensi Pressman:



Gambar 2.2 Waterfall Model (Pressman)

1. *Communication*

Langkah ini merupakan analisis terhadap kebutuhan software, dan tahap untuk mengadakan pengumpulan data dengan melakukan pertemuan dengan *customer*, maupun mengumpulkan data-data tambahan baik yang ada di jurnal, artikel, maupun dari internet.

2. *Planning Proses*

Planning merupakan lanjutan dari proses *communication (analysis requirement)*. Tahapan ini akan menghasilkan dokumen *user requirement* atau bisa dikatakan sebagai data yang berhubungan dengan keinginan *user* dalam pembuatan software, termasuk rencana yang akan dilakukan.

3. *Modeling Proses*

Modeling ini akan menerjemahkan syarat kebutuhan ke sebuah perancangan software yang dapat diperkirakan sebelum dibuat coding. Proses ini berfokus pada rancangan struktur data, arsitektur software, representasi interface, dan detail (algoritma) prosedural. Tahapan ini akan menghasilkan dokumen yang disebut *software requirement*.

4. *Construction*

Construction merupakan proses membuat kode. Coding atau pengkodean merupakan penerjemahan desain dalam bahasa yang bisa dikenali oleh komputer. Programmer akan menerjemahkan transaksi yang diminta oleh user. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu software, artinya penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan testing terhadap sistem yang telah dibuat tadi. Tujuan testing adalah menemukan kesalahan-kesalahan terhadap sistem tersebut untuk kemudian bisa diperbaiki.

5. *Deployment*

Tahapan ini bisa dikatakan final dalam pembuatan sebuah software atau sistem. Setelah melakukan analisis, desain dan pengkodean maka sistem yang

sudah jadi akan digunakan oleh user. Kemudian software yang telah dibuat harus dilakukan pemeliharaan secara berkala.

Kelebihan sistem Waterfall menurut Pressman :

1. Merupakan model pengembangan paling handal dan paling lama digunakan
2. Cocok untuk system software berskala besar.
3. Cocok untuk software bersifat generic.
4. Pengerjaan project system akan terjadwal dengan baik, dan mudah di control.

Kekurangan sistem Waterfall menurut Pressman :

1. Persyaratan system harus di gambarkan dengan jelas.
2. Rincian proses harus benar-benar jelas dan tidak boleh berubah rubah.
3. Sulit untuk mengadaptasikan jika terjadi perubahan spesifikasi pada suatu tahapan pengembangan.

2.2.6 User Acceptance Testing

Menurut Perry (2006:70), *User Acceptance Testing* merupakan pengujian yang dilakukan oleh *end-user* dimana *user* tersebut adalah *staff/karyawan* perusahaan yang langsung berinteraksi dengan sistem dan dilakukan verifikasi apakah fungsi yang ada telah berjalan sesuai dengan kebutuhan/fungsinya.

Menurut Lewis (2009:134), setelah dilakukan *system testing*, *acceptance testing* menyatakan bahwa sistem *software* memenuhi persyaratan. *Acceptance testing* merupakan pengujian yang dilakukan oleh pengguna yang menggunakan teknik pengujian *black box* untuk menguji sistem terhadap spesifikasinya.

Pengguna akhir bertanggung jawab untuk memastikan semua fungsionalitas yang relevan telah diuji.

Menurut Black (2002:7), *acceptance testing* biasanya berusaha menunjukkan bahwa sistem telah memenuhi persyaratan-persyaratan tertentu. Pada pengembangan *software* dan *hardware* komersial, *acceptance test* biasanya disebut juga "*alpha tests*" (yang dilakukan oleh pengguna *in-house*) dan "*beta tests*" (yang dilakukan oleh pengguna yang sedang menggunakan atau akan menggunakan sistem tersebut). *Alpha* dan *beta test* biasanya juga menunjukkan bahwa produk sudah siap untuk dijual atau dipasarkan. *Acceptance testing* mencakup data, *environment* dan skenario yang sama atau hampir sama pada saat *live* yang biasanya berfokus pada skenario penggunaan produk tertentu.

Dari definisi di atas, *user acceptance testing* adalah pengujian yang dilakukan oleh pengguna dari sistem tersebut untuk memastikan fungsi-fungsi yang ada pada sistem tersebut telah berjalan dengan baik dan sesuai dengan kebutuhan pengguna.

U
M
N