

BAB III

LANDASAN TEORI

3.1 Event Management

Event Management dapat didefinisikan sebagai pengorganisasian sebuah kegiatan yang dikelola secara profesional, sistematis, efisien dan efektif. Kegiatannya meliputi konsep (perencanaan) sampai dengan pelaksanaan hingga pengawasan. Dalam *event management*, semua orang harus bekerja keras dengan visi yang sama untuk menghasilkan kegiatan yang sesuai dengan harapan. Sangatlah diperlukan kekompakan pada setiap orang yang terlibat dalam tim.

Menurut Goldblatt (Goldblatt, 2013) *Event Management* adalah kegiatan profesional mengumpulkan dan mempertemukan sekelompok orang untuk tujuan perayaan, pendidikan, pemasaran dan reuni, serta bertanggung jawab mengadakan penelitian, membuat desain kegiatan, melakukan perencanaan dan melaksanakan koordinasi serta pengawasan untuk merealisasikan kehadiran sebuah kegiatan.

Menyelenggarakan sebuah kegiatan merupakan sebuah kebutuhan yang tidak dapat dipungkiri bagi sebuah perusahaan komersial maupun non komersial. kegiatan-kegiatan tersebut bertujuan untuk menunjang kegiatan *Public Relations* (PR) dalam menciptakan citra positif perusahaan di mata *internal stakeholder* maupun *external stakeholder*. Selain itu kegiatan-kegiatan tersebut tak dapat dilepaskan dari kegiatan *marketing communication* dari perusahaan, terutama dalam kegiatan-kegiatan *below the line*.

Pada mulanya, semua kegiatan perusahaan terutama yang berkaitan dengan kegiatan-kegiatan baik internal (seperti: *training* untuk karyawan, *family/employee*

gathering, pertemuan, dll) maupun *external*(seperti: pameran, kegiatan dengan *sponsorship*, seminar, *product launching*, dll) secara langsung dilaksanakan oleh PR atau *Marketing Communication Division*. Sementara itu di masyarakat, kegiatan-kegiatan semacam itu dilakukan dengan membentuk semacam kepanitiaan yang ditugaskan menyelenggarakan sebuah acara atau kegiatan tersebut. Prinsip dalam membuat kegiatan (5 W + 1 H):

1. *Why* : Alasan kegiatan itu dibuat
2. *What* : Apa bentuk kegiatan itu
3. *Where* : Dimana akan dilaksanakan
4. *When* : Kapan dan berapa lama kegiatan tersebut dilaksanakan
5. *Who* : Siapa saja yang akan terlibat dalam kegiatan tersebut
6. *How* : Bagaimana kegiatan itu dapat berjalan

3.2 Program Aplikasi

Program merupakan ekspresi, pernyataan kombinasi yang disusun dan dirangkai menjadi satu kesatuan prosedur yang berupa urutan langkah untuk menyelesaikan masalah yang diimplementasikan dengan menggunakan bahasa pemrograman, sehingga dapat dieksekusi oleh komputer. Aplikasi merupakan penerapan, menyimpan sesuatu hal, data, permasalahan pekerjaan kedalam suatu sarana atau media yang digunakan untuk menerapkan atau mengimplementasikan hal atau permasalahan tersebut sehingga berubah menjadi suatu bentuk yang baru tanpa menghilangkan nilai-nilai dasar dari hal, data, permasalahan atau pekerjaan. Jadi dalam hal ini hanya bentuk dari tampilan data yang berubah, sedangkan isi yang memuat dalam data tersebut tidak mengalami perubahan. Berdasarkan pengertian dari program dan aplikasi diatas maka dapat disimpulkan, program

aplikasi adalah sederetan kode yang digunakan untuk mengatur komputer agar dapat melakukan pekerjaan sesuai dengan keinginan dari permasalahan pengguna (Jogiyanto, 2000).

3.3 *Transaction Processing System*

Transaction Processing System (TPS) merupakan sistem informasi yang pertama kali diimplementasikan. Fokus utama TPS adalah pada data transaksi. Sesuai dengan namanya, sistem informasi ini digunakan untuk menghimpun, menyimpan, dan memproses data transaksi serta kadangkala mengendalikan keputusan yang merupakan bagian dari transaksi. TPS merupakan sistem yang mengendalikan keputusan. Maksudnya adalah TPS dapat memproses transaksi dan sekaligus dapat memvalidasi keabsahan data sesuai dengan kebutuhan pelanggan.

TPS mengumpulkan data secara kontinyu, biasanya *real time* – data dapat segera dihasilkan – dan sebagai data masukan untuk database perusahaan. TPS merupakan proses yang kritical untuk suksesnya organisasi. Fungsinya untuk memproses data transaksi dari kejadian bisnis. TPS menghapus rasa bosan saat melakukan transaksi operasional sekaligus mengurangi waktu, meskipun orang masih harus memasukkan data ke sistem komputer secara manual. TPS merupakan sistem tanpa batas yang memungkinkan organisasi berinteraksi dengan lingkungan eksternal. Karena manajer melihat data-data yang dihasilkan oleh TPS untuk memperbaharui informasi setiap menit mengenai apa yang terjadi di perusahaan mereka. Dimana hal ini sangat penting bagi operasi bisnis dari hari ke hari agar sistem-sistem ini dapat berfungsi dengan lancar dan tanpa interupsi sama sekali. TPS berkembang dari sistem informasi manual untuk sistem proses data dengan bantuan mesin menjadi sistem proses data elektronik (*electronic data*

processing systems). TPS mencatat dan memproses data hasil dari transaksi bisnis, seperti penjualan, pembelian, dan perubahan persediaan/inventori. TPS menghasilkan berbagai informasi produk untuk penggunaan internal maupun eksternal. Contoh: *POS-Terminal* untuk penjualan. Pada dunia usaha, proses-proses yang mengacu pada transaksi pertukaran barang atau uang atau jasa disebut dengan *Transaction Processing System (TPS)*.

Beberapa jenis subsistem yang ada pada TPS antara lain:

1. *Payroll*: pembayaran upah atau gaji karyawan.
2. *Order Entry* atau *Order Processing*: mencatat pembelian untuk konsumen.
3. *Invoicing*: menghasilkan faktur.
4. *Inventory*: mengelola barang supaya selalu tersedia.
5. *Shipping*: menyerahkan barang dari perusahaan sampai diterima oleh konsumen.
6. *Accounts receivable*: mengelola file konsumen & menyerahkan tagihan ke konsumen.
7. *Purchasing*: mengkoordinasi pembelian barang kepada konsumen.
8. *Receiving*: menerima barang dari pemasok/supplier pengembalian barang (retur) dari konsumen.
9. *Account Payable*: mengelola pembayaran tagihan kepada pemasok.
10. *General Ledger*: mengikat subsistem diatas menjadi satu & menghasilkan satu laporan.

Penginputan data ke dalam sistem informasi dapat melalui beragam cara, antara lain:

1. Dengan merekam data ke dalam sebuah formulir.

2. Dengan menginputkan data langsung ke dalam komputer.
3. Dengan *short message services* (SMS).
4. Dengan menginputkan data di internet.
5. Dengan *barcode-scanner*.
6. Dengan *scanner* yang lain.

Karakteristik TPS

Menurut Turban (Turban, 1999), karakteristik dari TPS adalah sebagai berikut:

7. Jumlah data yang diproses sangat besar.
8. Sumber data umumnya internal dan keluaran terutama dimaksudkan untuk pihak internal (meskipun bisa juga diperuntukkan bagi mitra kerja).
9. Pemrosesan informasi dilakukan secara teratur: harian, mingguan, dan sebagainya.
10. Kapasitas penyimpan (basis data) besar.
11. Kecepatan pemrosesan yang diperlukan tinggi karena volume yang besar.
12. Umumnya memantau dan mengumpulkan data masa lalu.
13. Masukan dan keluaran terstruktur. Mengingat data yang diproses cukup stabil, data diformat dalam suatu standar.
14. Tingkat kerincian yang tinggi mudah terlihat terutama pada masukan tetapi seringkali juga pada keluaran.
15. Komputasi tidak rumit (menggunakan matematika sederhana atau operasi statistik).
16. Memerlukan kehandalan yang tinggi.
17. Pemrosesan terhadap permintaan merupakan suatu keharusan. Pemakai dapat melakukan permintaan terhadap basis data.

Cara pemrosesan pada TPS

18. *Batch processing*

Batch processing adalah transaksi penghematan sumber daya untuk pengelolaan data dalam jumlah besar. *Batch processing* berguna bagi perusahaan yang perlu untuk memproses data dalam jumlah besar menggunakan sumber daya yang terbatas. Contoh *batch processing* meliputi transaksi kartu kredit, untuk transaksi yang diproses bulanan secara *real time*. Transaksi kartu kredit hanya perlu diproses sekali dalam sebulan untuk menghasilkan pernyataan bagi pelanggan, sehingga *batch processing* menghemat sumber daya TI dari keharusan untuk memproses setiap transaksi secara individual.

19. *Real-time processing*

Dalam banyak keadaan faktor utama adalah kecepatan. Sebagai contoh, ketika seorang nasabah bank menarik sejumlah uang dari rekeningnya, sangat penting bahwa transaksi akan diproses dan saldo akun diperbaharui sesegera mungkin, sehingga baik bank dan pelanggan untuk melacak dana. (Mahendra, 2013).

3.4 *System Development Life Cycle*

Daur hidup pengembangan sistem, yang dalam bahasa Inggris dikenal sebagai *software development life cycle* (SDLC), berfungsi untuk menggambarkan tahapan-tahapan utama dan langkah-langkah untuk pengembangan sistem.

3.4.1 Analisis

Tahap analisis adalah sebuah proses investigasi terhadap sistem yang sedang berjalan dengan tujuan untuk mendapatkan jawaban mengenai pengguna sistem, cara kerja sistem dan waktu penggunaan sistem. Dari proses analisa ini akan didapatkan cara untuk membangun sistem baru (Melengo, 2013).

3.4.2 Desain

Tahap desain memiliki tujuan untuk mendesain sistem baru yang dapat menyelesaikan masalah-masalah yang dihadapi perusahaan yang diperoleh dari pemilihan alternatif sistem yang terbaik. Kegiatan yang dilakukan dalam tahap desain ini meliputi desain proses, desain data, desain antar muka dan desain fisik.

A. Desain Proses

Desain proses menggambarkan bagaimana mengubah hasil analisis kedalam bentuk *Context Diagram* dan *Data Flow Diagram*.

A.1. *Context Diagram*

Context Diagram merupakan langkah pertama dalam desain proses. Pada *Context Diagram* dijelaskan sistem apa yang dibuat dan *eksternal entity* apa saja yang terlibat. Dalam *Context Diagram* harus ada arus data yang masuk dan arus data yang keluar (Kendall, 2003).

A.2. *Data Flow Diagram*

DFD level 0 adalah langkah selanjutnya setelah *Context Diagram*. Pada langkah ini, digambarkan proses-proses yang terjadi dalam sistem informasi. Setelah selesai maka akan di *decompose* menjadi DFD Level 1 yang merupakan penjelasan

dari DFD level 0. Pada proses ini dijelaskan proses apa saja yang dilakukan pada setiap proses yang terdapat di DFD level 0 (Kendall, 2003).

B. Desain Data

Desain data adalah aktivitas terpenting dalam pembuatan rekayasa perangkat lunak. Bagian desain data terdiri dari *ER-Model* dan Normalisasi.

B.1. *ER-Model*

ER-Model adalah model data konseptual tingkat tinggi untuk perancangan basis data. Model data konseptual adalah himpunan konsep yang mendeskripsikan struktur basis data, transaksi pengambilan dan pembaruan basis data.

Konsep paling dasar di *ER-Model* adalah entitas, relasi dan atribut. Komponen-komponen utama model ER adalah :

1. Entitas (*entity*) yang dipakai untuk memodelkan objek-objek yang berada di perusahaan/lingkungan.
2. Relasi (*relationship*) yang dipakai untuk memodelkan koneksi/hubungan di antara entitas-entitas.
3. Atribut yang dipakai untuk memodelkan properti yang dimiliki oleh entitas dan relasi.
4. Batasan yang terdiri dari batasan integritas serta ketentuan validasi.

Entitas merupakan individu yang mewakili sesuatu yang nyata (eksistensinya) dan dapat dibedakan dari sesuatu yang lain. Sebuah kursi yang kita duduki, seseorang yang menjadi pegawai di sebuah perusahaan dan sebuah mobil yang melintas di depan kita adalah entitas.

Sekelompok entitas yang sejenis dan berada dalam lingkup yang sama membentuk sebuah himpunan entitas (*entity sets*). Sederhananya, entitas menunjuk pada individualitas suatu objek, sedang himpunan entitas menunjuk pada rumpun (*family*) dari individu tersebut.

Seorang pasien, misalnya akan dimasukkan dalam himpunan entitas pasien. Sedang seorang dokter akan ditempatkan dalam himpunan entitas dokter. Dalam berbagai pembahasan/literatur, penyebutan himpunan entitas (yang kurang praktis) ini seringkali digantikan dengan sebutan entitas saja. Karena itu sering ditemui, penggunaan istilah entitas (*entity*) di sebuah literatur sebenarnya menunjuk pada himpunan entitas (Komputer, 2013).

B.2. Normalisasi

Normalisasi merupakan teknik analisis data yang mengorganisasikan atribut-atribut data dengan cara mengelompokkan sehingga terbentuk entitas yang *non-redundant*, stabil, dan fleksibel. Normalisasi dilakukan sebagai uji coba pada suatu tabel secara berkelanjutan untuk menentukan apakah tabel itu sudah baik, yaitu dapat dilakukan proses *insert*, *update*, *delete*, dan modifikasi pada satu atau beberapa atribut tanpa mempengaruhi integritas data dalam relasi tersebut.

Pada proses normalisasi terhadap tabel pada database dapat dilakukan dengan tiga tahap normalisasi antara lain:

1. Bentuk Tidak Normal

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti format tertentu, dapat saja data tidak lengkap atau terduplikasi. Data dikumpulkan apa adanya sesuai dengan saat menginput.

2. Bentuk Normal Pertama (1NF)

Pada tahap ini dilakukan penghilangan beberapa group elemen yang berulang agar menjadi satu harga tunggal yang berinteraksi di antara setiap baris pada suatu tabel, dan setiap atribut harus mempunyai nilai data yang atomik (bersifat *atomic-value*). Atom adalah zat terkecil yang masih memiliki sifat induknya, bila terpecah lagi maka ia tidak memiliki sifat induknya.

Syarat normal ke satu (1NF) antara lain:

- a. Setiap data dibentuk dalam *flat-file*, data dibentuk dalam satu *record* demi satu *record* nilai dari atribut berupa "*atomic value*".
- b. Tidak ada seperangkat atribut yang berulang atau bernilai ganda.
- c. Telah ditentukannya *primary key* untuk tabel / relasi tersebut.
- d. Setiap atribut hanya memiliki satu pengertian.

3. Bentuk Normal Kedua (2NF)

Bentuk normal kedua didasari atas konsep *full functional dependency* (ketergantungan fungsional sepenuhnya) yang dapat didefinisikan sebagai berikut. Jika A adalah atribut dari suatu relasi, B dikatakan *full functional dependency* (memiliki ketergantungan fungsional) terhadap A, tetapi tidak secara tepat memiliki ketergantungan fungsional dari subset (himpunan bagian) A.

Syarat normal kedua (2NF) sebagai berikut:

- e. Bentuk data telah memenuhi kriteria bentuk normal kesatu.
- f. Atribut bukan kunci (*non-key*) harus memiliki ketergantungan fungsional sepenuhnya (*fully functional dependency*) pada kunci utama atau *primary key*.

4. Bentuk Normal Ketiga (3NF)

Walaupun relasi 2NF memiliki redundansi yang lebih sedikit dari pada relasi 1NF, namun relasi tersebut masih mungkin mengalami kendala bila terjadi anomali peremajaan (*update*) terhadap relasi tersebut. Jika kita hanya mengupdate satu baris saja, sementara baris yang lainnya tidak, maka data didalam database tersebut menjadi tidak konsisten atau tidak teratur. Anomali terhadap *update* ini disebabkan oleh suatu ketergantungan transitif (*transitive dependency*). Kita harus menghilangkan ketergantungan tersebut dengan melakukan normalisasi ketiga (3NF).

Syarat normal ketiga (3NF) adalah sebagai berikut:

- g. Bentuk data telah memenuhi kriteria bentuk normal kedua.
- h. Atribut bukan kunci (*non-key*) harus tidak memiliki ketergantungan transitif, dengan kata lain suatu atribut bukan kunci (*non-key*) tidak boleh memiliki ketergantungan fungsional (*functional dependency*) terhadap atribut bukan kunci lainnya, seluruh atribut bukan kunci pada suatu relasi hanya memiliki ketergantungan fungsional terhadap *primary key* di relasi itu saja (Jwyner, 2013).

C. Desain Antar Muka

C.1. Perangkat Keras

Perangkat keras komputer atau *hardware* adalah perangkat pada komputer yang berbentuk fisik (dapat disentuh). Perangkat komputer sendiri dibedakan menjadi dua yaitu perangkat keras dan perangkat lunak istilah asingnya yaitu *hardware* (perangkat keras) dan *software* (Perangkat lunak). *Hardware* sendiri

berfungsi dengan baik dikarenakan adanya *software* sebagai sistem yang menjalankannya.

Agar proses pembelajaran mengenai komputer lebih terarah adapun pengelompokan *hardware* agar lebih mudah sesuai dengan fungsinya pada komputer. Berikut ini pengelompokan perangkat keras komputer :

1. *Input Device*: Perangkat input / masukan
2. *Process Device*: Perangkat yang menjalankan proses sistem komputer
3. *Output Device*: Perangkat output / keluaran, menghubungkan sistem keluar
4. *Storage Device*: Perangkat untuk menyimpan
5. (Affandi, 2013)

C.2. Perangkat Lunak

Software adalah perangkat lunak. Karena disebut juga sebagai perangkat lunak, maka sifatnya pun berbeda dengan *hardware* atau perangkat keras, jika perangkat keras adalah komponen yang nyata yang dapat dilihat dan disentuh oleh secara langsung manusia, maka *software* atau Perangkat lunak tidak dapat disentuh dan dilihat secara fisik, *software* memang tidak tampak secara fisik dan tidak berwujud benda namun bisa untuk dioperasikan.

Pengertian perangkat lunak komputer adalah sekumpulan data elektronik yang disimpan dan diatur oleh komputer, data elektronik yang disimpan oleh komputer itu dapat berupa program atau instruksi yang akan menjalankan suatu perintah. Melalui **perangkat lunak** inilah suatu komputer dapat menjalankan suatu perintah. (Vicky, 2013)

C.3. Jaringan Komputer

Jaringan komputer adalah sebuah sistem yang terdiri atas beberapa komputer yang didesain sedemikian rupa sebagaimana tujuan utamanya yakni untuk dapat berbagi sumber daya (CPU, *pzrinter*), berkomunikasi (pesan instan, surel), dan dapat mengakses informasi (situs web). Sementara menurut pembagiannya, jaringan komputer dapat dibedakan menjadi dua jenis, yakni jaringan terdistribusi dan jaringan tersentralisasi.

Manfaat jaringan komputer:

1. Menjadikan sistem komputer menjadi lebih mudah dioperasikan dan fleksibel
2. Banyak aplikasi pendukung yang dapat dijalankan di berbagai macam jenis komputer
3. Tidak menjadikan ketergantungan terhadap satu jenis komputer (sehingga jenis komputer lain bisa dikoordinasikan secara bersamaan)
4. Reabilitas tinggi dan dapat membagi sumber daya
5. Memperluas pendayagunaan sistem operasi
6. Memperluas kemudahan berkomunikasi
7. Memudahkan kecepatan mengakses informasi
8. (Prastya, 2013)
- 9.

D. Desain Fisik

D.1. *Physical Data Model*

Physical data model merupakan model yang menggunakan sejumlah tabel untuk menggambarkan data serta hubungan antara data-data tersebut. Setiap tabel mempunyai sejumlah kolom di mana setiap kolom memiliki nama yang unik.

Sebuah *physical data model* adalah representasi dari desain data yang memperhitungkan fasilitas dan kendala sistem database yang diberikan manajemen. Dalam siklus hidup proyek itu biasanya berasal dari model data logis, meskipun mungkin *reverse-engineered* dari implementasi database yang diberikan. Sebuah *physical data model* lengkap akan mencakup semua artefak database yang diperlukan untuk membuat hubungan antara tabel atau mencapai tujuan kinerja seperti indeks, definisi kendala, menghubungkan tabel, tabel dipartisi atau cluster. *Physical data model* biasanya dapat digunakan untuk menghitung perkiraan penyimpanan dan mungkin termasuk rincian alokasi penyimpanan khusus untuk sistem basis data tertentu.

Saat ini, ada enam basis data utama dalam pasar bisnis, Oracle, Postgres, SQL Server, Sybase, DB2 dan MySQL. Ada banyak sistem RDBMS besar lainnya di luar sana, tetapi ini cenderung baik untuk menjadi warisan database atau digunakan dalam akademis seperti universitas atau kolese pendidikan lanjutan. Sebuah *physical data model* pada setiap pelaksanaan akan secara signifikan berbeda, paling tidak karena kebutuhan OS yang mendasari yang duduk di bawah mereka. Contohnya SQLServer yang hanya berjalan pada sistem operasi Microsoft Windows, sedangkan Oracle dan MySQL dapat berjalan di Solaris, Linux dan sistem operasi berbasis UNIX serta Windows. Ini berarti bahwa persyaratan *disk*, persyaratan keamanan dan banyak aspek lain dari *physicaldatamodel* akan dipengaruhi sepenuhnya oleh RDBMS yang *database administrator* (atau organisasi mereka) memilih untuk menggunakan. (Alamsyah, 2013)

D.2. Kamus Data

Kamus data atau *data dictionary* atau disebut juga dengan istilah *systems data dictionary* adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Dengan menggunakan kamus data analisis sistem dapat mendefinisikan data yang mengalir di sistem secara lengkap. Kamus data digunakan untuk merancang input, laporan-laporan dan database, Kamus data dibuat berdasarkan arus data yang ada di DAD.

Kamus data harus memuat hal-hal berikut (Florentina, 2013), diantaranya:

1. Nama Arus Data

Karena kamus data dibuat berdasarkan arus data yang mengalir di DAD, maka nama arus data juga harus dicatat di kamus data.

2. Alias

Alias atau nama lain dari data dapat dituliskan bila nama lain ini ada. Alias perlu ditulis karena data yang sama mempunyai nama berbeda untuk orang atau departemen satu dengan lainnya.

3. Bentuk Data

Bentuk dari data yang mengalir dapat berupa:

- i. dokumen dasar atau formulir
- j. dokumen hasil cetakan komputer
- k. laporan tercetak
- l. tampilan layar monitor
- m. variabel
- n. *parameter*
- o. *field*

4. Arus Data

Arus data menunjukkan darimana data mengalir dan kemana data akan menuju.

Keterangan arus data ini perlu dicatat di kamus data supaya memudahkan mencari arus data ini di DAD.

5. Penjelasan

Bagian penjelasan dapat diisi dengan keterangan tentang arus data tersebut.

6. Periode

Periode ini menunjukkan kapan terjadinya arus data ini. Periode perlu dicatat di kamus data karena dapat digunakan untuk mengidentifikasi kapan input data harus dimasukkan ke sistem, kapan di proses dan kapan laporan harus dihasilkan.

7. Volume

Volume yang perlu dicatat dalam kamus data adalah tentang volume rata-rata dan volume puncak dari arus data. Volume ini digunakan untuk mengidentifikasi besarnya simpanan luar yang akan digunakan, kapasitas dan jumlah dari alat input, alat pemroses, dan alat output.

8. Struktur Data

Struktur data menunjukkan arus data yang dicatat di kamus data terdiri dari item-item data apa saja.

3.4.3 Pengujian

Pengujian perangkat lunak adalah elemen kritis dari jaminan kualitas perangkat lunak dan merepresentasikan spesifikasi, desain dan pengkodean. Meningkatnya visibilitas (kemampuan) perangkat lunak sebagai suatu elemen

sistem dan “biaya” yang muncul akibat kegagalan perangkat lunak, memotivasi dilakukannya perencanaan yang baik melalui pengujian yang teliti. Pada dasarnya, pengujian merupakan satu langkah dalam proses rekayasa perangkat lunak yang dapat dianggap sebagai hal yang merusak daripada membangun.

A. *Black Box Testing*

Test case ini bertujuan untuk menunjukkan fungsi perangkat lunak tentang cara beroperasinya, apakah pemasukan data keluaran telah berjalan sebagaimana yang diharapkan dan apakah informasi yang disimpan secara eksternal selalu dijaga kemutakhirannya.

Teknik pengujian *black-box* berfokus pada domain informasi dari perangkat lunak, dengan melakukan *test case* dengan menpartisi domain input dari suatu program dengan cara yang memberikan cakupan pengujian yang mendalam.

Metode pengujian *graph-based* mengeksplorasi hubungan antara dan tingkah laku objek-objek program. Partisi ekivalensi membagi domain input ke dalam kelas data yang mungkin untuk melakukan fungsi perangkat lunak tertentu. Analisis nilai batas memeriksa kemampuan program untuk menangani data pada batas yang dapat diterima.

Metode pengujian yang terspesialisasi meliputi sejumlah luas kemampuan perangkat lunak dan area aplikasi. Antar muka pengguna, arsitektur *client/server*, dokumentasi dan fasilitas *help* dan sistem *real time* masing-masing membutuhkan pedoman dan tehnik khusus untuk pengujian perangkat lunak (Alfiyanto, 2013).