

BAB II

LANDASAN TEORI

2.1 Pesawat

Pesawat terbang atau pesawat udara atau kapal terbang atau cukup pesawat saja adalah kendaraan yang mampu terbang di atmosfer atau udara. Pesawat terbang yang lebih berat dari udara disebut aerodin, yang masuk dalam kategori ini adalah autogiro, helikopter, girokopter dan pesawat bersayap tetap.

Pesawat bersayap tetap umumnya menggunakan mesin pembakaran dalam yang berupa mesin piston (dengan baling-baling) atau mesin turbin (jet atau turboprop) untuk menghasilkan dorongan yang menggerakkan pesawat, lalu pergerakan udara di sayap menghasilkan gaya dorong ke atas, yang membuat pesawat ini bisa terbang. Sebagai pengecualian, pesawat bersayap tetap juga ada yang tidak menggunakan mesin, misalnya glider, yang hanya menggunakan gaya gravitasi dan arus udara panas. Helikopter dan autogiro menggunakan mesin dan sayap berputar untuk menghasilkan gaya dorong ke atas, dan helikopter juga menggunakan mesin untuk menghasilkan dorongan ke depan.

2.2 Perawatan Pesawat

Setiap pesawat udara selama beroperasi pasti mempunyai jadwal untuk perawatan. Perawatan ini harus dilakukan karena setiap komponen mempunyai batas usia tertentu sehingga komponen tersebut harus diganti. Selain itu, komponen juga harus diperbaiki bila ditemukan telah mengalami kerusakan. Secara garis besar, program perawatan dapat dibagi menjadi dua kelompok besar,

yaitu perawatan preventif dan korektif. Perawatan preventif adalah perawatan yang mencegah terjadinya kegagalan komponen sebelum komponen tersebut rusak. Sedangkan perawatan korektif adalah perawatan yang memperbaiki komponen yang rusak agar kembali ke kondisi awal.

Perawatan preventif dapat dibagi menjadi 2 jenis yaitu:

- i. Perawatan periodik atau *hard time*, merupakan perawatan yang dilakukan berdasarkan batas waktu dari umur maksimum suatu komponen pesawat. Dengan kata lain, perawatan ini merupakan perawatan pencegahan dengan cara mengganti komponen pesawat meskipun komponen tersebut belum mengalami kerusakan.
- ii. Perawatan *on-condition*, merupakan perawatan yang memerlukan inspeksi untuk menentukan kondisi suatu komponen pesawat. Setelah itu ditentukan tindakan selanjutnya berdasarkan hasil inspeksi tersebut. Bila ada gejala kerusakan, komponen tersebut dapat diganti bila alasan-alasan teknik dan ekonominya memenuhi.

Perawatan korektif dikenal pula dengan nama *condition monitoring* yaitu perawatan yang dilakukan setelah ditemukan kerusakan pada suatu komponen, dengan cara memperbaiki komponen tersebut. Bila cara perbaikan tidak dapat dilakukan dengan alasan teknik maupun ekonomi, maka harus dilakukan penggantian.

Perawatan pesawat biasanya dikelompokkan berdasarkan interval yang sepadan dalam paket-paket kerja atau disebut dengan clustering. Hal ini dilakukan agar tugas perawatan lebih mudah, efektif dan efisien. Interval yang dijadikan pedoman untuk melaksanakan paket-paket tersebut adalah sebagai berikut:

i. Flight Hours

Merupakan interval inspeksi yang didasarkan pada jumlah jam operasional suatu pesawat terbang.

ii. Flight Cycle

Merupakan interval inspeksi yang didasarkan pada jumlah *takeoff-landing* yang dilakukan suatu pesawat terbang. Satu kali *takeoff-landing* dihitung satu *cycle*.

iii. Calendar Time

Merupakan interval inspeksi yang dilakukan sesuai dengan jadwal tertentu. Dari jumlah tugas perawatan atau inspeksi yang dilaksanakan, *maintenance* dapat dibagi dalam *minor maintenance* seperti *transit check*, *before departure check*, *daily check*, *weekly check* dan *heavy maintenance* seperti A-Check, B-Check, C-Check dan D-Check.

Minor maintenance:

i. Transit Check

Inspeksi ini harus dilaksanakan setiap kali setelah melakukan penerbangan saat transit di *station* mana pun. Operator biasanya memeriksa pesawat untuk memastikan bahwa pada pesawat tidak terdapat satu pun kerusakan struktur, semua sistem berfungsi dengan sebagaimana mestinya, dan servis yang diharuskan telah dilakukan.

ii. Before Departure Check

Inspeksi ini harus dilakukan sedekat mungkin sebelum tiap kali pesawat berangkat beroperasi, maksimal dua jam sebelumnya.

iii. Daily Check (Overnight Check)

Pemeriksaan ini harus dilakukan satu kali dalam jangka waktu 24 jam setelah *daily check* sebelumnya dilakukan. Setiap hari pesawat telah diprediksi akan *ground stop* minimal selama empat jam. Inspeksi ini mencakup pemeriksaan komponen, pemeriksaan keliling pesawat secara visual untuk mendeteksi ada atau tidaknya ketidaksesuaian, melakukan pengamanan lebih lanjut, dan pemeriksaan sistem operasional.

iv. Weekly Check

Pemeriksaan ini harus telah dilakukan dalam tujuh hari penanggalan. Termasuk dalam inspeksi ini adalah *before departure check*.

Aircraft maintenance checks adalah periode pemeriksaan yang harus dilakukan pada pesawat setelah penggunaan pesawat untuk jangka waktu tertentu, digunakan sebagai parameter interval untuk *heavy maintenance* yang meliputi A-Check, B-Check, C-Check, dan D-Check.

A-Check dilakukan kira-kira setiap satu bulan. Pemeriksaan ini biasanya dilakukan hingga 10 jam. Pemeriksaan ini bervariasi, bergantung pada tipe pesawat, jumlah siklus (*takeoff* dan *landing* dianggap sebagai siklus pesawat, atau jam terbang sejak pemeriksaan terakhir. Perawatan pesawat jenis ini hanya melakukan pemeriksaan pada pesawat terbang untuk memastikan kelaikan mesin, sistem-sistem, komponen-komponen, dan struktur pesawat untuk beroperasi. Untuk Boeing 737 Classic A-check dilakukan setelah 300 jam terbang, Airbus A340 setelah 450 jam terbang, Boeing 747-200 setelah 650 jam.

B-Check bergantung pada masing-masing jenis pesawat, pemeriksaan berkisar antara 9 hingga 28 jam *ground time* dan biasanya dilakukan kira-kira

setiap lima bulan. Perawatan pesawat dalam skala kecil ini hanya meliputi proses pembersihan, pelumasan, penggantian ban apabila sudah aus, penggantian baterai, dan inspeksi struktur bagian dalam.

C-Check harus dilakukan setelah 15-18 bulan. Bergantung pada tipe pesawat, pemeriksaan ini bisa memakan waktu 10 hari. Perawatan pesawat tipe ini merupakan inspeksi komprehensif termasuk bagian-bagian yang tersembunyi, sehingga kerusakan dan keretakan di bagian dalam dapat ditemukan. Untuk Boeing 737-300 dan 737-500, inspeksi ini dilakukan setiap 4.000 FH. Untuk Boeing 737-400 dilakukan setiap 4.500 FH. Sedangkan untuk Boeing 747-400 dilakukan setiap 6.400 FH dan Airbus A-330-341 dilakukan setiap 21 bulan.

D-Check disebut *overhaul*. Pemeriksaan jenis ini adalah perawatan yang paling detail, untuk pesawat Boeing 737-300, 737-400 dan 737-500, inspeksi ini dilakukan setiap 24.000 FH. Sedangkan untuk Boeing 747-400 dilakukan setiap 28.000 FH dan untuk Airbus A-330-341 dilakukan setiap 6 tahun. Pada pengecekan jenis ini pesawat diinspeksi secara keseluruhan, biasanya memakan waktu 1 bulan.

2.3 Entitas dalam Perawatan Pesawat

Pada proses perawatan pesawat, ada beberapa entitas yang terlibat didalamnya baik yang secara langsung akan berinteraksi dengan objek, ataupun entitas yang terlibat secara tidak langsung.

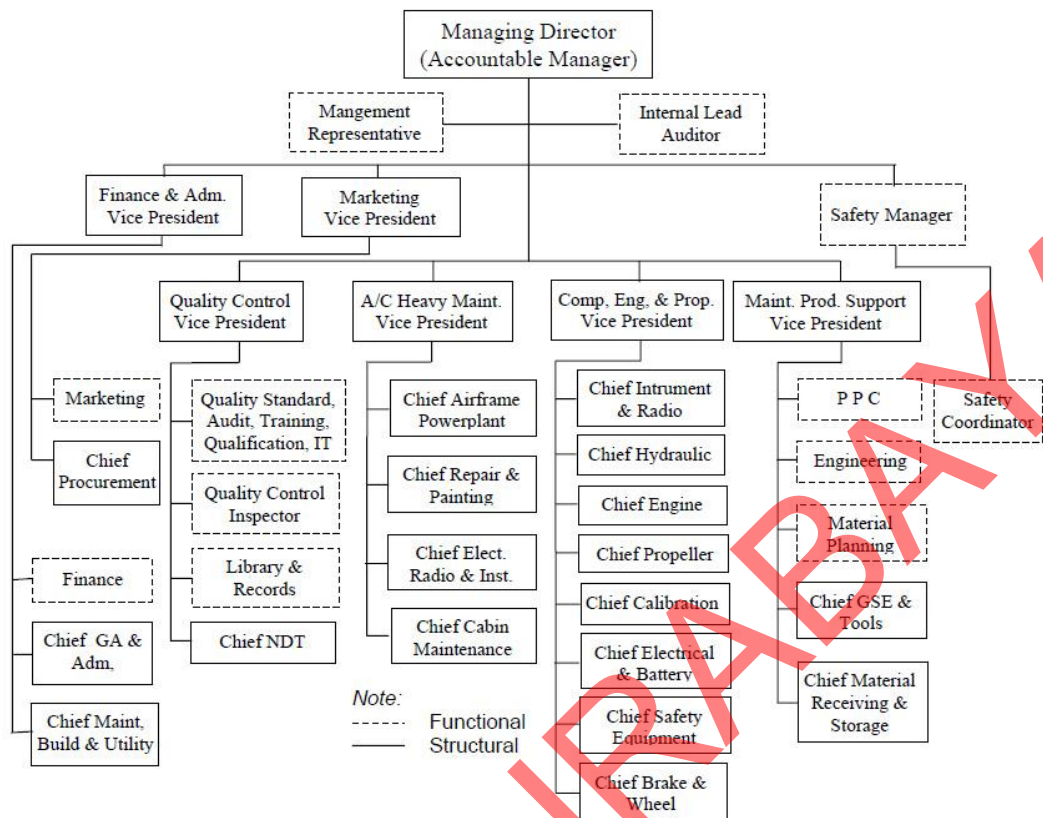
1. *Line Maintenance (LM)*. Bagian ini biasanya adalah divisi dari perusahaan *customer* yang memiliki pesawat yang akan dilakukan proses perawatan. Divisi LM harus ada di semua perusahaan aviasi, dan harus

bertanggungjawab dalam pengelolaan kebutuhan perawatan semua pesawat yang dimiliki perusahaan.

2. *Customer*. Sering disebut juga sebagai *techrep* oleh pihak yang mengadakan perawatan pesawat (*maintenance facility*). *Customer* adalah perusahaan, atau perwakilan perusahaan, atau perorangan yang memiliki pesawat dan menyerahkan proses perawatannya ke *maintenance facility*.
3. *Marketing*. Bagian ini yang akan berhadapan langsung dengan *customer*. Komunikasi antara *customer* dan *maintenance facility* akan dilakukan melalui divisi *marketing*.
4. *Production planner*. Bagian ini adalah penentu dan pengawas terhadap jalannya proses perawatan pesawat. *Planner* yang menentukan pekerjaan apa saja yang harus dilakukan oleh *engineer* di bagian *production* sesuai dengan komplain dan permintaan dari *customer* melalui *marketing*.
5. *Production*. Bagian ini berisi sekumpulan *engineer* dengan berbagai bidang keahlian yang akan mengerjakan detail-detail pekerjaan perawatan pesawat sesuai dengan lingkup yang telah ditentukan oleh *planner*. *Engineer* sendiri terdiri dari 3 tingkatan, yaitu *supporting staff*, *inspector*, dan *certified staff*.
6. *Material Store*. Bagian ini merupakan divisi yang bertugas untuk mengelola sirkulasi *material* pesawat yang terlibat. Mulai dari *material* yang turun dari pesawat, *material* yang akan dipasang di pesawat, hingga *material* yang dibutuhkan untuk penggantian. *Material Store* akan berkomunikasi intensif dengan bagian *purchasing* sehingga setiap ada *material* yang diminta/dibutuhkan maka bagian *purchasing* akan mendapat instruksi untuk melakukan pengadaan terhadap *material* tersebut.

7. *Tool Store*. Merupakan divisi yg khusus mengelola semua kebutuhan *tool* atau peralatan untuk melakukan perawatan pesawat.
8. *Purchasing*. Bagian ini merupakan divisi yang harus melakukan pengadaan terhadap seluruh benda/alat/*spare part* yang diperlukan dalam kegiatan perawatan pesawat.
9. *Finance*. Bagian keuangan dalam organisasi perawatan pesawat, divisi ini yang mengelola semua kebutuhan biaya langsung dan tidak langsung yang ada dalam proses perawatan pesawat.
10. *Quality Assurance*. Bagian memiliki 2 tugas pokok, yaitu menjamin kualitas hasil produksi yang dilakukan oleh *engineer* dan melakukan kontrol terhadap jalannya peraturan yang telah ditetapkan pada proses bisnis perawatan pesawat. *Quality Assurance* juga mengelola otorisasi *engineer* yang bekerja di bagian *production* sehingga setiap detil pekerjaan yang dilakukan oleh *engineer* sesuai dengan bidang keahliannya masing-masing, *quality assurance* harus menjadi *filter* untuk mencegah orang yang tidak berwenang untuk mengerjakan sesuatu yang seharusnya tidak boleh dikerjakan.

Struktur organisasi perawatan pesawat dan komponen pesawat yang ada pada Merpati Maintenance Facility dapat dilihat pada gambar 2.1.



Gambar 2.1. Struktur Organisasi pada Merpati Maintenance Facility

2.4 Administrasi Perawatan Pesawat

Administrasi perawatan pesawat (AZ) terdiri dari berbagai pekerjaan dan tugas yang harus dilakukan untuk menjaga kegiatan perawatan pesawat berjalan aman dan efisien. Setiap bagian yang terlibat dengan fungsinya masing-masing saling berkomunikasi dengan intensif dalam pekerjaan administrasi perawatan pesawat.

Beberapa tugas dan kewajiban yang dilakukan oleh AZ adalah:

- Penjadwalan pemeriksaan pesawat udara.
- Menjaga tren grafik kehandalan sistem kerja pesawat.
- Mengatur dan menjalankan daftar pustaka, laporan, dan data yang berhubungan dengan perawatan pesawat.

- d. Mengeluarkan perintah kerja dan merilis sertifikat inspeksi.
- e. Menjalankan tugas administrasi seperti dokumentasi dan pencetakan.
- f. Pembuatan laporan hasil perawatan pesawat dan korespondensi.
- g. Mempertahankan *logbooks* mesin pesawat dan catatan yang terkait.

Lingkungan kerja untuk pekerjaan administrasi pesawat biasanya merupakan lingkungan kantor yang bersih dan nyaman. Tempat kerja bervariasi tergantung dimana mereka ditugaskan, di darat atau di pantai maupun laut. Tugas mereka memerlukan kerjasama yang erat antar sesama pekerja administrasi perawatan pesawat di tiap bagian yang berbeda-beda fungsinya.

Dokumen dan *form* yang ada pada kegiatan administrasi perawatan pesawat antara lain:

- a. *Quotation*, berisi *work order* dari *customer*, di dalamnya terdiri dari beberapa *repair order* atau biasa disebut *subject*.
- b. *Work pack*, adalah paket/kumpulan pekerjaan yang harus dilakukan oleh *engineer* dalam proyek perawatan pesawat. *Work pack* terdiri dari pekerjaan *routine* dan *non-routine*. *Routine* berisi *basic task card*, *SIP (Structure Inspection Programme)*, dan *CPCP (Corrosion Prevention and Control Programme)*. *Non-routine* berisi *EO (Engineering Order)*, *AD-SB (Airworthiness Directive-Service Bulletin)*, *HT-CRR (Hard Time-Component Replacement)*, dan *SI (Special Instruction)*.
- c. *Task card*, adalah standar prosedur atau pekerjaan yang harus dilakukan oleh *engineer* dalam perawatan pesawat. Setiap jenis pesawat disertai *Basic Task Card* yang dibuat oleh *vendor*.

- d. *Job Order*, menunjukkan suatu *task card* yang harus dikerjakan oleh *engineer* dalam perawatan pesawat.
- e. *CRS (Certificate of Release to Service)*, merupakan sertifikat yang dikeluarkan oleh *Maintenance Facility* sebagai tanda bahwa suatu proyek perawatan pesawat telah selesai. *CRS* dibuat dan dipertanggungjawabkan secara penuh oleh *engineer* dengan tingkat *certifying staff*. *CRS* diberikan kepada *customer* saat proyek perawatan telah selesai.

Administrasi perawatan pesawat juga memiliki batasan-batasan terhadap *engineer* maupun semua orang yang terlibat pada setiap prosesnya, otorisasi dalam perawatan pesawat adalah:

- a. *Supporting Staff, engineer* dengan tingkat ini hanya dapat melakukan pekerjaan berdasar *job order* yang telah ditentukan.
- b. *Inspector, engineer* dengan tingkat ini selain dapat melakukan pekerjaan berdasar *job order* yang telah ditentukan, *engineer* ini juga dapat melakukan inspeksi terhadap pekerjaan-pekerjaan tersebut.
- c. *Certified Staff, engineer* tingkat ini dapat melakukan pekerjaan semua jenjang, mulai dari pengerjaan *job order* biasa, melakukan inspeksi, melakukan *RII Release*, hingga membuat *CRS*.

2.5 Desain Sistem

Setelah tahap analisis dan perancangan sistem selesai dilakukan, maka analisis sistem telah mendapatkan gambaran dengan jelas apa yang harus dikerjakan. Lalu tahap selanjutnya adalah desain sistem.

Desain sistem adalah tahap setelah analisis dari siklus pengembangan sistem pendefinisian dari kebutuhan-kebutuhan fungsional dan persiapan untuk rancang bangun implementasi, menggambarkan bagaimana suatu sistem dibentuk.

“Pada tahap desain secara umum, komponen-komponen sistem informasi dirancang dengan tujuan untuk dikomunikasikan dengan pemakai sistem, bukan pemrogram. Komponen sistem informasi yang didesain adalah model, *input*, *output*, *database*, teknologi, dan kontrol” (Jogiyanto, 2003:211).

Analisis sistem dapat mendesain model dari sistem informasi yang diusulkan dalam bentuk *physical system* dan *logical model*. Bagan alir sistem (*system flowchart*) merupakan alat yang tepat untuk menggambarkan *physical system*. Simbol-simbol bagan alir sistem ini menunjukkan secara tepat arti fisiknya seperti simbol terminal, *harddisk*, dan laporan-laporan.

Logical model dari sistem informasi lebih menjelaskan kepada pemakai sistem bagaimana nantinya fungsi-fungsi pada sistem informasi secara logika akan bekerja. *Logical model* dapat digambarkan dengan diagram arus data (*data flow diagram*). Arus data pada *data flow diagram* dapat dijelaskan dengan kamus data atau *data dictionary*. Sketsa dari *physical system* dapat menjelaskan kepada pemakai sistem bagaimana nantinya sistem secara fisik akan diterapkan.

Maka dari itulah pada akhirnya *physical system* dan *logical model* sangat diperlukan di tahap desain sistem ini, karena sangat berguna untuk menjelaskan kepada pemakai, pemrogram dan ahli teknik yang terlibat tentang kerja sistem.

2.6 Interaksi Manusia dan Komputer

Menurut Shanti (2005), istilah Interaksi Manusia dan Komputer (*Human Computer Interaction*) sebenarnya telah lama dipelajari oleh para ahli pada masa

perang dunia kedua dengan munculnya keperluan untuk menghasilkan sistem persenjataan yang efektif sehingga dipelajarilah interaksi manusia dengan mesin pada saat itu. Hal ini kemudian mendorong munculnya ketertarikan para peneliti di bidang ini dan membentuk suatu perkumpulan peneliti di bidang ergonomi (*Ergonomi Research Society*).

2.6.1 Konsep Dasar Interaksi Manusia dan Komputer

Komputer dan peralatan terkait lainnya harus dirancang dengan pemahaman bahwa penggunaannya memiliki tujuan atau tugas khusus dan ingin menggunakannya sesuai dengan karakteristik tugas yang akan diselesaikan. Pada kenyataannya, masih sering kita jumpai kesalahan-kesalahan kecil dalam pengoperasian suatu sistem yang teraplikasi. Sebagai contoh, menu pilihan “*Save*” dan “*Delete*” diklasifikasikan dalam satu kelompok yang sama sebagai “Operasi File”, jika *user* kurang teliti dan memilih menu “*Delete*” padahal yang dimaksud adalah pilihan “*Save*” ditambah dengan tidak adanya mekanisme konfirmasi/dialog dalam eksekusi proses tersebut maka hal ini dapat merugikan *user*.

Menurut Shanti (2005) dalam jurnalnya, komputer diperkenalkan sebagai “*user friendly*” dan “*easy to use*”. Agar dua hal tersebut dapat terpenuhi maka perancang sistem perlu mengetahui bagaimana berpikir dalam lingkup tugas *user* yang sesungguhnya untuk kemudian menerjemahkannya ke dalam sistem.

Tidak mudah merancang sistem yang konsisten dan handal yang dapat mengantisipasi semua ketidaktelitian *user*. Interface bukanlah aspek yang dapat dibuat pada saat akhir, desainnya merupakan satu kesatuan dengan keseluruhan sistem. Desainer tidak hanya memberikan suatu tampilan yang “cantik” namun

juga harus dapat mendukung pekerjaan yang dilakukan oleh user dan dapat menghindari kesalahan-kesalahan kecil.

2.6.2 Interaksi Manusia dan Komputer pada Web

Hal-hal yang perlu diperhatikan dalam proses mendesain/menyusun suatu aplikasi berbasis web adalah:

1. Balance

Menyeimbangkan penyusunan komponen di layar.

2. Symmetry

Berbeda dengan *balance*, kesimetrisan sebuah desain menimbang tiap element di tiap sisi layar.

3. Regularity

Penciptaan sebuah desain dengan ukuran normal dan standarisasi dalam sebuah aplikasi, yaitu meliputi penggunaan ukuran hingga ke jarak antar komponen yang terdapat dalam layar.

4. Predictability

Dengan menyusun komponen yang sederhana dan telah umum digunakan hingga mudah ditebak pengguna yang baru pertama kali menggunakan aplikasi.

5. Sequentiality

Penglihatan yang tertuju pada suatu tempat yang dianggap atraktif. Secara intuitif, penglihatan menuju ke :

- a) Kontrol yang lebih terang.
- b) Element yang terisolasi.
- c) Gambar daripada teks.

- d) Warna yang menyolok.
- e) Kontrol yang lebih besar.
- f) Bentuk yang tidak standar.

6. Economy

Memberi *style* pada font, warna, serta pengaturan yang tidak berlebihan.

7. Unity

Tidak memberi jarak yang berlebih pada sebuah desain aplikasi.

8. Proportion

Sebuah desain yang dianggap proporsional tidak selalu berbentuk bujur sangkar, tetapi harus memiliki perbandingan yang sesuai dengan ukuran layar secara jamak.

9. Simplicity

Melakukan desain yang mengesankan keseragaman sehingga desain terlihat sederhana. Kesederhanaan dapat ditimbulkan dengan melakukan proses *alignment* yang tidak terlalu banyak serta bentuk komponen yang umum.

10. Groupings

Pengelompokan komponen berdasarkan fungsi serta penataan visual yang efisien.

Adapun beberapa elemen desain web yang perlu dipahami agar desain web yang dibuat berhasil secara artistik dan juga memenuhi kaidah *usability* serta *user friendly*, adalah:

1. Line

Bukan hanya sekedar garis yang ditampilkan dalam sebuah web, tetapi garis yang secara tidak langsung mampu membentuk batasan antar elemen dan juga kesan sebuah obyek dua dimensi.

2. Color

Khusus untuk pemilihan warna dalam desain web, disarankan untuk memilih warna dalam *web safe pallete* yang hanya terdiri dari 216 warna.

3. Volume

Diasumsikan sebagai ilusi yang terjadi dari sebuah bentuk dua dimensi. Ilusi tersebut akan terkesan sebagai bentuk 3D bagi pengguna dalam sebuah situs.

4. Movement

Pergerakan tidak hanya ditimbulkan oleh animasi, tetapi juga kesan dinamis yang muncul saat perubahan warna terjadi antar elemen disebuah situs, misalnya elemen atas dengan warna *absolute*, sedangkan bagian bawah ditempati oleh warna gradasi.

5. Space

Merupakan ruang yang tersisa dari sebuah situs. Sebagai contoh, desain situs yang terletak ditengah dan menyisakan ruang kosong di bagian kanan dan kiri untuk memenuhi prinsip *balance* dan simetris (*Symmetry*).

2.7 Testing Software

Menurut Romeo (2003:3) testing software adalah proses mengoperasikan software dalam suatu kondisi yang di kendalikan, untuk verifikasi apakah telah berlaku sebagaimana telah ditetapkan (menurut spesifikasi), mendeteksi *error*, dan validasi apakah spesifikasi yang telah ditetapkan sudah memenuhi keinginan atau kebutuhan dari pengguna yang sebenarnya. Verifikasi adalah

pengecekan atau pengetesan entitas-entitas, termasuk software, untuk pemenuhan dan konsistensi dengan melakukan evaluasi hasil terhadap kebutuhan yang telah ditetapkan. Validasi adalah melihat kebenaran sistem, apakah proses yang telah dilakukan adalah apa yang sebenarnya diinginkan atau dibutuhkan oleh *user*. Jadi, dapat disimpulkan bahwa testing merupakan tiap-tiap aktifitas pengumpulan informasi yang dibutuhkan untuk melakukan evaluasi atau mengukur suatu atribut dari software.

Testing software dilakukan untuk mendapatkan informasi *reliable* terhadap software dengan cara termudah dan paling efektif, antara lain:

1. Apakah software telah siap digunakan?
2. Apa saja resikonya?
3. Apa saja kemampuannya?
4. Apa saja keterbatasannya?
5. Apa saja masalahnya?
6. Apakah telah berlaku seperti yang diharapkan?

2.7.1 Test Case

Test case merupakan suatu tes yang dilakukan berdasarkan pada suatu inisialisasi, masukan, kondisi ataupun hasil yang telah ditentukan sebelumnya.

Adapun kegunaan dari *test case* ini, adalah sebagai berikut:

1. Untuk melakukan testing kesesuaian suatu komponen terhadap spesifikasi (*Black Box Testing*).
2. Untuk melakukan testing kesesuaian suatu komponen terhadap desain (*White Box Testing*).

Menurut Ganesan (2010) tujuan dasar dari penulisan *test case* adalah untuk melakukan validasi cakupan *testing* dari sebuah aplikasi. *Test case* yang ditulis dengan baik dapat membuat siklus *testing* menjadi lebih efisien. Sebuah *test case* yang baik dapat dengan mudah menentukan apakah suatu fitur dari aplikasi bekerja dengan benar.

Tabel 2.1 Contoh Test Case Pada Halaman Login.html

Test Case	Check Item	Test case Objective	Steps to Execute	Test Data / Input	Expected Result
TC-001	Log-in Page	Leave all fields as blank and click Log-in button	Click Log-in		By leaving all fields as blank and on click Log-in button then mandatory symbol (*) should appear in front of Username and Password fields
TC-002	Username	Enter Invalid Username	NA	Username : Jackk	By entering invalid Username then an error message should appear as " Please Enter Valid Username "
TC-003	Username	Enter valid Username	NA	Username : Jack	It should allow the user to proceed
TC-004	Password		NA		The password field should display the encrypted format of the text typed as (****)

Test Case	Check Item	Test case Objective	Steps to Execute	Test Data / Input	Expected Result
TC-005	Password	Enter wrong password	NA	Password : ***	By entering invalid password then an error message should appear as "Please Enter Correct Password "
TC-006	Password	Enter Correct password	NA	Password : *****	It should allow the user to proceed
TC-007	Log-in button	Correct Inputs	Click Log-in		It should lead the user to the respect page
TC-008	Forgot Password	Check hyperlink on Forgot Password label			while mouse over of the label an hand icon should display
TC-009	Forgot Password		Click Forgot Password		User can recover the password using the "Forgot Password" link page
TC-010	Registration	Check hyperlink on Registration label			while mouse over of the label an hand icon should display
TC-011	Registration		Click Registrati on		On click "Registration " page should redirect to the User Registration page

2.7.2 Black Box Testing

Black box testing, dilakukan tanpa pengetahuan detil struktur internal dari sistem atau komponen yang ditest, juga disebut sebagai *behavioral testing*,

specification-based testing, input / output testing atau functional testing. Black box testing berfokus pada kebutuhan fungsional pada software, berdasarkan pada spesifikasi kebutuhan dari *software*. Kategori *error* yang akan diketahui melalui *black box testing* adalah sebagai berikut:

- a) Fungsi yang hilang atau tidak benar.
- b) *Error* dari antar muka.
- c) *Error* dari struktur data atau akses eksternal database.
- d) *Error* dari kinerja atau tingkah laku.
- e) *Error* dari inisialisasi dan terminasi.

Test di desain untuk menjawab pertanyaan sebagai berikut:

1. Bagaimana validasi fungsi yang akan dites?
2. Bagaimana tingkah laku kinerja dari sistem yang akan dites?
3. Kategori masukan apa saja yang bagus digunakan untuk *test case*?
4. Apakah sebagian sistem sensitif terhadap suatu nilai masukan tertentu?
5. Bagaimana batasan suatu kategori masukan ditetapkan?
6. Sistem mempunyai toleransi jenjang dan volume data apa saja?
7. Apa saja akibat dari kombinasi data tertentu yang akan terjadi pada operasi dari sistem?