BAB III

OBJEK DAN METODE PENELITIAN

3.1.Objek Penelitian

Objek penelitian yang digunakan oleh penulis adalah Pegadaian Sederhana Bandung yang beralamat di Jln Sederhana no 35 .Adapun penjelasan mengenai objek penelitian adalah sebagai berikut.

3.1.1. Sejarah Singkat Perusahaan

Pada tahun 1990 berdiri sebuah usaha yaitu pegadaian yang didirikan oleh Ibu Tuti tetapi dia tidak menjalankan usaha nya sendiri namun masuh bergabung dengan pihak pemerintah dan pada tahun 2000 Ibu Tuti sudah dapat menjalankan usaha pegadaian ini sendiri karena sudah memiliki modal yang cukup dan usaha ini sudah berjalan sampai sekarang.

Batas waktu pegadaian 4 bulan dengan bunga 3% / Bulan dengan dena ½ dari persenan / Bulan.

Dan apabila emas yang di gadaikan jatuh tempo pihak pegadaian akan melelang barang tersebut.

3.1.2. Visi dan Misi Perusahaan

Visi

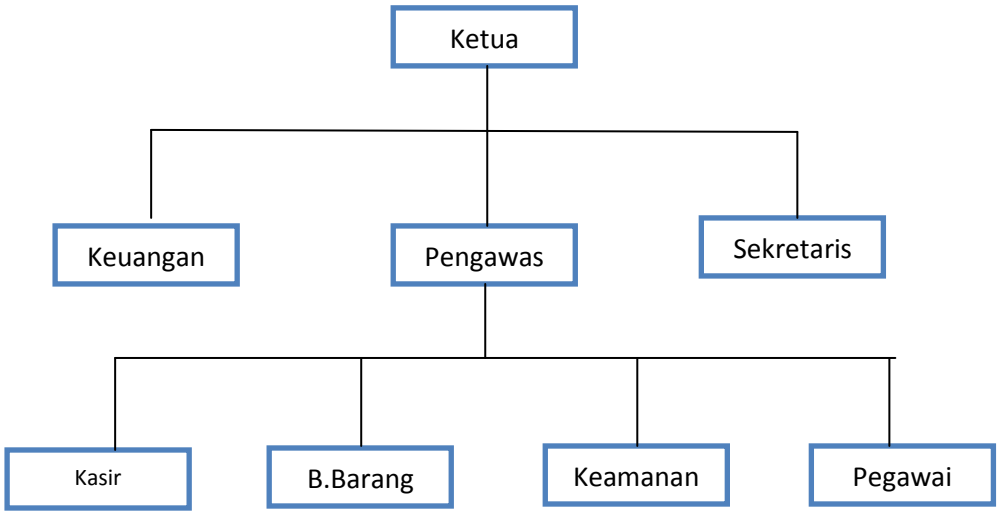
Menjadikan Pegadaian Sederhana Bandung ,menjadi pegadaian Pembiayaan Mikro dan kecil berbasis gadai bagi masyarakat mengeah ke bawah.

Misi

1. Membantu program pemerintah meningkatkan kesejahteraan rakyat khususnya golongan menengah ke bawah dengan memberikan solusi keuangan yang terbaik melalui penyaluran pinjaman skala mikro, kecil dan menengah atas dasar hukum gadai dan fidusia.
2. Memberikan manfaat kepada pemangku kepentingan dan melaksanakan tata kelola perusahaan yang baik secara konsisten.
3. Melaksanakan usaha lain dalam rangka optimalisasi sumber daya.

3.1.3. Struktur Organisasi Perusahaan

Struktur Organisasi perusahaan merupakan susunan kepengurusan atau pengelolaan dari sebuah organisasi, himpunan atau kelompok kerja dan juga dibagi-bagi tugasnya untuk melakukan aktivitas kerja sesuai dengan fungsi dan jabatannya. Dengan adanya sistem organisasi yang tegas, manajemen akan mengetahui siapa-siapa yang bertanggung jawab untuk setiap kegiatan dalam perusahaan dan juga dapat mengukur efektivitas tiap karyawan dalam lingkungan pekerjaan yang menjadi tugasnya.



Gambar 3.1.

Struktur Organisasi Pegadaian Sederhana Bandung

3.1.4. Deskripsi Tugas

Ketua

1. Mengendalikan seluruh kegiatan Pegadaian
2. Memimpin rapat
3. Menerima laporan
4. Mengambil keputusan

Keuangan

1. Membuat Laporan keuangan
2. Menyimpan dana pinjaman

Sekretaris

1. Membantu ketua dipegadaian
2. Menyampaikan pesan penting kepada ketua

Pengawas

1. Mengawasi jalan nya kegiatan Pegadaian

Kasir

1. Melayani konsumen
2. Menerima barang gadaian

B.Barang

1. Menyimpan Barang gadaian para konsumen

Keamanan

1. Menjaga dan mengamankan lingkungan Pegadaian

Pegawai

1. Membersihkan kantor pegadaian
2. Menyiap konsumsi untuk para pegawai Pegadaian

3.2. Metode Penelitian

Menurut Sugiyono (2004: 1) Metode penelitian merupakan cara ilmiah untuk mendapatkan data dengan tujuan dan kegunaan tertentu.

Dalam penelitian ini, metode yang digunakan yaitu :

1. Metode Deskriptif

   Metode ini merupakan metode penelitian yang menggambarkan dan menginterpretasikan objek sesuai dengan apa adanya. Metode ini merupakan metode penelitian yang digunakan untuk menangani Rumusan masalah yang pertama yaitu mengenai bagaimana Sistem Informasi Pegadaian yang berjalan di Pegadaian Sederhana Bandung.

2. Metode *Action*

Metode *Action* ini merupakan metode penelitian yang berupa tindakan dan manipulasi data dalam penelitian. Metode ini merupakan metode yang digunakan untuk menangani Rumusan masalah Perancangan Sistem Informasi Pegadaian pada Pegadaian Sederhana Bandung dan juga mengenai pengujian Sistem Informasi Pegadaianpada Pegadaian Sederhana Bandung.

3.2.1. Desain Penelitian

Kegunaan desain atau rancangan penelitian ini adalah untuk menuliskan bagaimana bentuk dan prosedur yang dijalankan terhadap penelitian yang telah dilakukan. Adanya suatu desain penelitian juga dapat menggambarkan secara detail dari aliran sistem dari data yang diinputkan serta prosesnya menjadi sebuah hasil yang dapat digunakan.

Menurut Jogiyanto (2005:34) “Desain sistem menentukan bagaimana suatu sistem akan menyelesaikan apa yang mesti diselesaikan, tahap ini menyangkut kegiatan pengkofigurasian dari suatu sistem sehingga setelah instalasi dari sistem benar-benar akan memuaskan rancang bangun yang telah deteteapkan pada akhir tahapan analisis sistem”

3.2.2. Jenis dan Metode Pengumpulan Data

Adapun jenis dan metode pengumpulan data yang penulis gunakan saat melakukan penelitian di Pegadaian Sederhana Bandung ini adalah :

3.2.2.1. Sumber data Primer

Sumber data Primer adalah data atau segala informasi yang diperoleh dan didapat oleh penulis langsung dari sumber-sumber, baik sumber individu maupun

sekelompok bagian dari objek penelitian seperti hasil dari Wawancara dan Observasi langsung pada objek yang diteliti dalam hal ini yaitu data yang didapat dari Pegadaian Sederhana Bandung :

1. Wawancara

   Wawancara merupakan Metode pengumpulan data melalui tatap muka dan tanya jawab langsung mengenai bagaimana prosedur Sistem Informasi Pegadaian yang ada di Pegadaian Sederhana Bandung. Wawancara dilakukan antara peneliti dengan Ketua untuk menanyakan bagaimana prosedur yang dijalan di Pegadaian Sederhana Bandung serta mengenai aplikasi *software* yang digunakan di Pegadaian saat ini.

2. Observasi

   Observasi merupakan Metode pengumpulan data melalui peninjauansecara langsung dan pencatatan informasi yang dilakukan oleh peneliti di Pegadaian Sederhana Bandung.

3.2.2.2. Sumber data Sekunder

Sumber data Sekunder yaitu data Primer yang telah diolah lebih lanjut dan disajikan dengan baik oleh pengumpul data primer atau pihak lain. Data sekunder dapat berupa tabel-tabel atau diagram dan juga segala informasi yang berasal dari literature yang ada hubungannya dengan teori-teori mengenai topik penelitian yang didapat dari Pegadaian Sederhana Bandung.

Sumber data sekunder dapat berupa data Dokumentasi, yaitu data yang diperoleh oleh peneliti dengan membaca buku-buku di perpustakaan dan mencari

referensi yang berkaitan dengan masalah yang akan dibahas oleh penulis seperti transkrip, surat kabar, mesin pencarian maupun dokumen seperti skripsi dll.

3.2.3. Metode Pendekatan dan Pengembangan Sistem

3.2.3.1. Metode Pendekatan Sistem

Pendekatan sistem yang penulis ambil adalah analisis dan perancangan terstruktur, digunakan untuk mendefinisikan dan mengilustrasikan organisasi dari sistem informasi secara berjenjang dalam bentuk modul dan sub modul, perancangan terstruktur juga menunjukan hubungan elemen data dan elemen kontrol antara hubungan modulnya.

Model analisis harus dapat mencapai tiga sasaran utama, yaitu menggambarkan apa yang dibutuhkan oleh pengguna, membangun dasar bagi pembuatan desain perangkat lunak, membatasi serangkaian persyaratan yang dapat divalidasi begitu perangkat lunak dibangun.

Dengan demikian perancangan terstruktur dapat memberikan penjelasan yang lengkap dan sistem dipandang dari elemen data menggunakan alat bantu seperti : diagram konteks yang merupakan diagram suatu proses yang menggambarkan ruang lingkup suatu sistem. *DFD (Data Flow Diagram)* merupakan diagram yang mepresentasikan grafik dari suatu sistem yang menggambarkan komponen-komponen sebuah sistem serta aliran data, kamus data yang dapat diartikan fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. *Entity Relationship Diagram(ERD)* merupakan suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem

secara abstrak dan tabel relasi yang merupakan suatu *file* yang terdiri dari beberapa *group* elemen yang berulang-ulang dan perlu diorganisasikan terlebih dahulu.

Adapun kelebihan penggunaan metode terstruktur yaitu :

1. Merupakan pendekatan visual yang *relative simple* sehingga membuat metode ini mudah dimengerti oleh pengguna
2. Merupakan metode yang telah diketahui secara umum di berbagai industri maupun instansi
3. Metode ini telah diterapkan sejak lama sehingga metode ini sudah matang dan layak digunakan
4. Metode ini memungkinkan untuk melakukan validasi antara berbagai kebutuhan

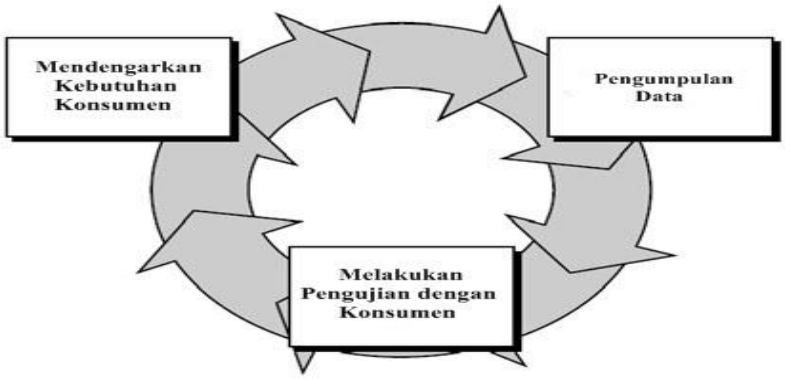
3.2.3.2. Metode Pengembangan Sistem

Metode Pengembangan Sistem merupakan kesatuan metode–metode, prosedur, konsep dan aturan-aturan yang digunakan untuk menyusun suatu sistem yang baru sebagai pengganti sistem yang lama secara keseluruhan atau memperbaiki sistem yang telah berjalan.

Metode yang digunakan dalam perancangan sistem ini yaitu menggunakan model *prototype*. yaitu suatu teknik analisis dan rancangan yangmemungkinkan pemakai ikut serta dalam mementukankebutuhan dan pembentukan sistem apa yang akan dikerjakan untuk memenuhi kebutuhan tersebut. Metode*prototype*

paling baik digunakan untuk mengembangkansistem yang didefinisikan kurang baik dan cocok untuk menerapkan sistem kecil:

Metode *prototype* terdapat 3 (tiga) tahapan untuk dapat mengembangkan suatu perangkat lunak seperti gambar dibawah ini :



Gambar 3.2 Model *prototype*

(Sumber : Roger S. Pressman, Ph.D. Rekayasa Perangkat Lunak**)**

Tahapan tersebut antara lain :

1. Pada tahap ini, pengembang dan pelanggan bertemu dan mendefiniskan objektif keseluruhan dari perangkat lunak, mengidentifikasi segala kebutuhan yang diketahui.
2. Pada tahap kedua, pengembang melakukan pengumpulan data yang berfokus pada penyajian dari aspek-aspek perangkat lunak tersebut yang akan nampak bagi pelanggan atau pemakai.
3. Pada tahap ketiga ini, konsumen atau pemakai mengevaluasi dan dipakai untuk menyaring kebutuhan pengembangan perangkat lunak. Pada tahap ini dimungkinkan perangkat lunak untuk di *setting* ulang untuk memenuhi

kebutuhan pelanggan dan pada saat yang sama memungkinkan pengembang untuk secara lebih baik memahami apa yang harus dilakukannya.

Alasan penyusun menggunakan metode *prototype* ini, karena metode *prototype* ini memiliki kelebihan-kelebihan sebagai berikut :

1. Metode ini cukup efektif sebagai paradigma dalam rekayasa perangkat lunak, karena mendapatkan kebutuhan dan aturan yang jelas yang disetujui oleh pelanggan, dalam pembuatan perangkat lunak bisa dilakukan secara cepat dan memungkinkan untuk merubah kembali perangkat lunak agar sesuai dengan kebutuhan pelanggan, walaupun pada umumnya *prototype* akan dihilangkan dan dibuat perangkat lunak yang sebenarnya.
2. Ketika diperlihatkan versi pertamanya atau tampilan *input* dan format *output* dalam media kertas pelanggan bisa langsung mengetahui apakah perangkat lunak tersebut bisa memenuhi kebutuhannya atau tidak.

3.2.3.3. Alat Bantu Analisis dan Perancangan

Dalam perancangan suatu sistem diperlukan beberapa alat bantu beruparepresentasi grafik yang dapat mempermudah dalam menggambarkan komponen-komponen yang ada, proses yang terjadi dan membuat usulan pemecahan masalah secara logika. Alat Bantu yang digunakan yaitu*Flow Map*, Diagram Konteks, *Data Flow Diagram (DFD),* Kamus Data dan Basis data.

1. ***Flow map***

   Flow Map merupakan bagan alir yang menunjukan arus dari laporan dan formulir termasuk tembusan. *Flow Map* digunakan untuk menganalisis bagaimana hubungan antara sub kerja yang akan menggerakkan sistem.

Setelah diketahui bagian-bagian yang terlibat dalam sistem, maka akan diketahui berapa jumlah entitas yang terkait dengan sistem yang dianalisis dan dirancang. Secara garis besar suatu *flowmap* digambarkan sebagai pemetaaan hubungan antara bagian-bagian kerja melalui dokumen, baik berupa laporan maupun formulir yang dinotasikan melalui simbol-simbol.

2. Diagram Konteks

Diagram konteks merupakan pola penggambaran yang berfungsi untuk memperlihakan interaksi Sistem Informasi tersebut dengan lingkungan dimana sistem tersebut ditempatkan.

3. *Data Flow Diagram (DFD)*

Diagram yang menggunakan notasi–notasi untuk menggambarkan arus dari data sistem biasa disebut Diagram Arus Data *(Data Flow Diagram / DFD). DFD* sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data itu mengalir atau lingkunagn fisik dimana data itu akan disimpan.

DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur.*DFD* merupakan alat yang cukup popular karena dapat menggambarkan arus data didalam sistem dengan terstruktur dan jelas.

Simbol-simbol yang digunakan dalam *DFD* terdiri dari 4 macam, yaitu : proses (*process*), aliran data (*data flow*), simpan data (*data store*), terminator (*external entity*).

Berikut uraian singkat mengenai 4 simbol tersebut :

a. Proses (*process*)

   Process adalah simbol yang mengubah suatu data dari suatu bentuk menjadi bentuk yang lain. Dengan kata lain, proses menerima masukan data dan mengeluarkan keluaran data lain yang telah diproses.

b. Aliran Data (*data flow*)

   Data flow atau aliran data adalah aliran yang menunjukkan perpindahan data dari satu bagian ke bagian lain dalam suatu sistem. *Data flow* dalam *DFD* digambarkan dengan tanda panah dan diberi keterangan disampingnya yang menunjukkan data yang mengalir.

c. Simpanan Data (*data store*)

   Data store adalah tempat penyimpanan data dalam suatu sistem, baik secara manual maupun secara elektronik. Simpanan data digunakan jika suatu proses perlu menggunakan lagi data tersebut.

d. Terminator (*external entity*)

   External entity adalah lingkungan luar dari sistem, sumber data menunjukkan suatu organisasi atau perseorangan yang memasukkan data ke sistem. Tujuan data menunjukkan suatu organisasi /peseorangan yang menerima data yang dihasilkan oleh sistem. Sumber dan tujuan data mempunyai satu simbol yang sama.

4. Kamus Data

Kamus data atau data *dictionary* atau disebut juga dengan istilah sistem data *dictionary* adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi.Dengan menggunakan kamus data, analis sistem dapat mendefinisikan data yang mengalir di sistem dengan lengkap.Pada tahap analisis, Kamus Data dapat digunakan sebagai alat komunikasi antara analisis sistem dengan pemakai sistem tentang data yang mengalir.Pada tahap perancangan sistem, kamus data digunakan untuk merancang input, merancang laporan-laporan dan *database*.

Kamus data harus mencerminkan keterangan yang jelas tentang data yang dicatatnya, maka kamus data harus memuat hal-hal berikut ini :

1. Nama arus data

   Karena kamus data dibuat berdasarkan arus data yang mengalir di diagram arus data, maka nama dari arus data juga harus dicatat di kamus data, sehingga mereka yang membaca diagram arus data dan memerlukan penjelasan lebih lanjut tentang suatu arus data tertentu di diagram arus data dapat langsung mencarinya dengan mudah.

2. Alias

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

4. Bentuk data

   Bentuk dari data yang mengalir dapat berupa dokumen dasar atau formulir, dokumen hasil cetakan komputer, laporan tercetak, tampilan di layar *monitor, variable, parameter, field.*

5. Arus Data

   Arus data menunjukan darimana data mengalir dan kemana data akan menuju. Keterangan arus data ini perlu dicatat di kamus data supaya memudahkan mencari arus data ini di diagram arus data.

6. Atribut

7. Atribut merupakan nama field dari data.

5. Perancangan Basis Data

1. Normalisasi

Normalisasi merupakan suatu proses untuk mengubah suatu relasi yang memiliki masalah anomaly atau tidak normal kedalam dua buah relasi atau lebih yang tidak memiliki masalah tersebut. Normalisasi digunakan untuk melakukan verifikasi terhadap tabel-tabel yang telah dibuat, sehingga tidak menimbulkan masalah saat data diperbaharui ataupun dihapus.

Apabila terdapat kesulitan pada saat menambah, menghapus, mengubah, membaca pada suatu database maka relasi tersebut dipecahkan pada beberapa tabel lagi.

Bentuk – bentuk normalisasi :

1. Bentuk *Unnormal*

   Pada bentuk *unnormal*, tidak ada keharusan mengikuti suatu format tertentu, data mungkin tidak lengkap atau terduplikasi. Data dikumpulkan apa adanya sesuai dengan kedatangannya.

2. Bentuk Normal kesatu

   Bentuk normal kesatu *(1NF)* terpenuhi jika sebuah tabel tidak memiliki Atribut bernilai banyak (*Multiple Attribute*) atau lebih dari satu atribut dengan domain nilai yang sama.

3. Bentuk Normal Kedua

   Bentuk normal kedua *(2NF)* terpenuhi jika pada tabel, semua atribut yang tidak termasuk dalam *key* primer memiliki ketergantungan fungsional (KF) pada *key* primer secara utuh. Sebuah tabel dikatakan tidak memenuhi *2NF*, jika ketergantungannya hanya berifat parsial.

4. Bentuk Normal ketiga

   Bentuk Normal ketiga *(3NF)* merupakan kriteria alternatif, karena jika kriteria *BCNF* yang ketat tidak dapat terpenuhi.Suatu relasi dikatakan dalam bentuk normal ketiga *(3NF)* jika sudah memenuhi bentuk normal kedua dan setiap atribut bukan kunci tidak memiliki dependensi transitif terhadap *primary key*.

5. Bentuk Normal *Boyce Codd*

Suatu relasi disebut memenuhi bentuk normal *Boyce Codd* jika dan hanya jika suatu penentu (determinan) adalah Kunci kandidat (atribut yang bersifat unik).

2. Tabel Relasi

Tabel Relasi bertujuan untuk membuat hubungan antar tabel agar terdapat relasi antara tabel yang satu dengan tabel yang lainnya. Kolom tabel menunjukan atribut dari *file*. Atribut ini menunjukan *item* data atau *field.*Kesimpulan nilai dari *field* atau *item* data disebut juga dengan dominan.

3. *Entity Relationship Diagram (ERD)*

Menurut Al-Bahra (2004:82), "*ERD* adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam system secara abstrak". *ERD* merupakan model jaringan data yang menekankan pada struktur-struktur dan *relationship data.*

Dari relasi tabel terdapat kardinalitas yaitu :

1. Satu ke satu *(One to One)*

   Hubungan *one to one relationship* ialah antara *file* pertama dengan *file* kedua adalah satu banding satu.

2. Satu ke banyak *(One to Many)*

   Hubungan *one to many relationship* ialah sebuah relasi merupakan tahap dimana hubungan antara *file* pertama dengan *file* kedua adalah satu berbanding.

3. Banyak ke banyak *(Many to Many)*

   Hubungan *many to many relationship* ialah sebuah relasi antara suatu *file* dengan *file* yang keduanya mempunyai relasi banyak berbanding banyak.

3.2.3.4. Pengujian *Software*

Cara pengetesan / pengujian sebuah *software* dapat dibagi menjadi 2 cara yaitu *Whitebox Testing* dan *Blackbox Testing*. Pengertian dari kedua metode tersebut adalah :

WHITEBOX TESTING

Pengertian *Whitebox Testing*

WhiteboxTesting merupakan cara pengujian dengan melihat ke dalam modul untuk meneliti kode-kode program yang ada, dan menganalisis apakah ada kesalahan atau tidak. Jika ada modul yang menghasilkan output yang tidak sesuai dengan proses bisnis yang dilakukan, maka baris-baris program, variabel, dan parameter yang terlibat pada unit tersebut akan dicek satu persatu dan diperbaiki, kemudian di-compile ulang.

Dengan menggunakan *whitebox*akan didapatkan kasus uji yang:

1. Menguji semua keputusan logikal
2. Menguji seluruh *Loop* yang sesuai dengan batasannya
3. Menguji seluruh struktur data internal yang menjamin validitas

Kelebihan *Whitebox Testing*:

1. Kesalahan Logika

   Digunakan pada sintaks '*if*' dan pengulangan. Dimana *Whitebox Testing*akan mendeteksi kondisi-kondisi yang tidak sesuai dan mendeteksi kapan proses pengulangan akan berhenti.

2. Ketidaksesuaian asumsi

   Menampilkan asumsi yang tidak sesuai dengan kenyataan, untuk di analisa dan diperbaiki.

3. Kesalahan ketik
4. Mendeteksi bahasa pemrograman yang bersifat *case sensitive*

Kelemahan *Whitebox Testing*:

Untuk perangkat lunak yang tergolong besar, *Whitebox Testing* dianggap sebagai strategi yang tergolong boros, karena akan melibatkan sumber daya yang besar untuk melakukannya.

BLACKBOX TESTING

Blackbox testing adalah metode pengujian perangkat lunak yang tes fungsionalitas dari aplikasi yang bertentangan dengan struktur internal atau kerja (lihat pengujian *whitebox*).pengetahuan khusus dari kode aplikasi / struktur internal dan pengetahuan pemrograman pada umumnya tidak diperlukan. Uji kasus dibangun di sekitar spesifikasi dan persyaratan, yakni, aplikasi apa yang seharusnya dilakukan. Menggunakan deskripsi eksternal perangkat lunak,

termasuk spesifikasi, persyaratan, dan desain untuk menurunkan uji kasus.Tes ini dapat menjadi fungsional atau non-fungsional, meskipun biasanya fungsional. Perancang uji memilih input yang valid dan tidak valid dan menentukan output yang benar. Tidak ada pengetahuan tentang struktur internal benda uji itu.

Metode uji dapat diterapkan pada semua tingkat pengujian perangkat lunak: unit, integrasi, fungsional, sistem dan penerimaan.Ini biasanya terdiri dari kebanyakan jika tidak semua pengujian pada tingkat yang lebih tinggi, tetapi juga bisa mendominasi unit testing juga.

Metode ujicoba *blackbox* memfokuskan pada keperluan fungsional dari software. Karna itu ujicoba blackbox memungkinkan pengembang software untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program. Ujicoba *blackbox* bukan merupakan alternatif dari ujicoba *whitebox*, tetapi merupakan pendekatan yang melengkapi untuk menemukan kesalahan lainnya, selain menggunakan metode *whitebox*.

Ujicoba *blackbox* berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya :

1. Fungsi-fungsi yang salah atau hilang
2. Kesalahan *interface*
3. Kesalahan dalam struktur data atau akses *databaseeksternal*
4. Kesalahan performa
5. kesalahan inisialisasi dan terminasi