BAB III

OBJEK DAN METODE PENELITIAN

3.1. Objek Penelitian

CV.Karya Pratam Gemilang Adalah Perusahaan yang berlokasi di Jalan Cikoang No. 45-A Bandung berstatus sebagai Perusahaan Non PMA – PMDN, menempati areal luas tanah + 134 m2 terdiri dari luas bangunan 50 m2 dan luas ruang usaha 50 m2. Selebihnya berfungsi sebagai halaman dan pekarangan.

3.1.1. Sejarah Singkat Perusahaan

CV.Karya Pratama Gemilang Bandung didirikan pada tanggal 20 Juni 1999, bertepatan dengan krisis moneter yang dialami oleh bangsa Indonesia. Ketika bangsa Indonesia mengalami Krisis Moneter, Pemutusan Hubungan Kerja dan jumlah pengangguran yang sangat tinggi.Pada saat itulah CV.Karya Pratama Gemilang Bandung didirikan dengan tujuan memperkecil jumlah pengangguran.

Dulu LKS yang pertama diterbitkan yaitu LKS Biologi untuk SMU. LKS ini merupakan kerjasama antara CV.Karya Pratama Gemilang Bandung sebagai penerbit dengan MGMP (Musyawarah Guru Mata Pelajaran) yakni perkumpulan guru-guru se-Jawa Barat.

Target pemasaran LKS ini yaitu kepada guru-guru yang ada di Kodya dan Kabupaten Bandung khususnya dan umumnya guru-guru yang ada di Jawa Barat.LKS ini dinamai (Motto) **SIGAP** (Siswa Giat Aktif dan Pintar) yang merupakan suatu Latihanr Kerja Siswa dengan tujuan membantu para guru dan murid untuk bisa dan terbiasa berlatih serta mempermudah proses belajar dan

mengajar. LKS yang sudah diterbitkan, pemasarannya cukup bagus, maka dirasakan perlu adanya penambahan LKS dan pembuatan buku paket baru. Setelah LKS untuk SMU/SMK/MA (seluruh mata pelajaran) sudah diterima dipasaran CV.Karya Pratam Gemilang Bandung juga mencetak LKS dan buku paket untuk SD,SMP (seluruh mata pelajaran) dan sekarang sudah mulai dipasarkan..

3.1.2. Visi dan Misi Perusahaan

Visi CV. Karya Pratama Gemilang yaitu ingin menjadi salah satu perusahaan yang maju di bidang percetakan, penjualan, dan supplier buku di kota Bandung. Serta bisa mengembangkan usaha secara luas.

Misi CV. Karya Pratama Gemilang yaitu :

a. memberikan pelayanan yang baik terhadap pelanggan.
b. Menjaga silaturahmi antara perusahaan dan pelanggan
c. Meningkatkan hubungan bsinis yang professional antara pelanggan dan perusahaan.
d. Meningkatkan kecerdasan generasi muda bangsa kita

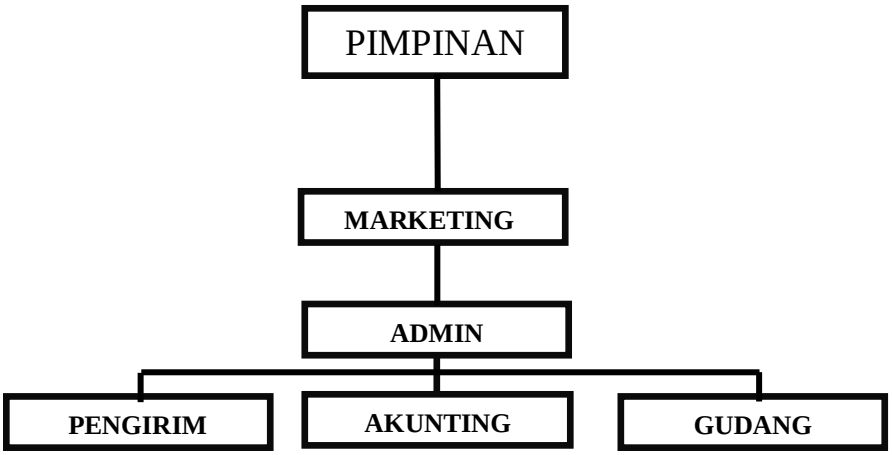
3.1.3. Struktur Organisasi Perusahaan

Struktur organisasi merupakan salah satu komponen dasaryang cukup penting dalam sebuah perusahaan, pembentukannya sesuai dengan kebutuhan perusahaan dan diatur sedemikian rupa sehingga mampu mendukung upaya untuk mencapai tujuan perusahaan yang diinginkan. Melalui sistem organisasi tersebut akan diatur tugas dan kewenangan masing-masing personil perusahaan serta hubungan serta pola kerja antar personil. Kejelasan mengenai

hubungan tersebut akan sangat besar artinya guna menghindari adanya kesalahan penanganan maupun kesalahan dalam menangani suatu pekerjaan. Tanpa adanya struktur organisasi yang baik, tidak mungkin suatu perusahaan akan berjalan dengan baik sesuai yang diharapkan.

Ini berarti bahwa sedikit banyak, sistem organisasi akan berpengaruh terhadap faktor dan kemampuan manajerial. Struktur organisasi Toko Buku Usaha dapat digambarkan sebagai berikut:

Gambar Struktur Organisasi



Gambar 3.1 Struktur Organisasi

3.1.4. Deskripsi tugas

1. Pimpinan

   a. Mengkoordinasikan dan mengendalikan kegiatan-kegiatan di perusahaan

   b. Mengkoordinasikan dan mengendalikan kegiatan pengadaan dan peralatan perlengkapan.

   c. Merencanakan dan mengembangkan sumber-sumber pendapatan serta pembelanjaan dan kekayaan perusahaan.

d. Mengendalikan uang pendapatan, hasil penagihan dari pelanggan.

e. Dalam melaksanakan tugas-tugas dan bertanggung jawab sepenuhnya atas tugas-tugas yang diemban.

f. Menawarkan visi dan imajinasi di tingkat tertinggi (biasanya bekerjasama dengan departemen-departemen terkait).

g. Memimpin rapat umum, dalam hal untuk memastikan pelaksanaan tata-tertib, keadilan dan kesempatan bagi semua untuk berkontribusi secara tepat, menyesuaikan alokasi waktu per item masalah, menentukan urutan agenda, dan menyimpulkan tindakan dan kebijakan.

h. Bertindak sebagai perwakilan organisasi dalam hubungannya dengan dunia luar.

i. Menjalankan tanggung jawab terhadap perusahaan sesuai dengan standar etika dan hukum, sebagai referensi dalam apapun standar dokumen kebijakan direktur yang digunakan.

2. Admin

a. Mencatat data pesanan yang masuk pada perusahaan

b. Mengecek ketersediaan barang

c. Membuat faktur untuk bagian lain

d. Mebuat laporan yang masuk kepda perusahaan

3. Akunting

a. Merencanakan dan mengontrol arus kas perusahaan (*cashflow*).

b. Mengumpulkan dan menganalisa data dan informasi keuangan untuk menghasilkan laporan keuangan yang dibutuhkan perusahaan.

c. Mengelola perencanaan, pelaporan dan pembayaran kewajiban pajak perusahaan.

d. Merencanakan dan mengkoordinasikan penyusunan anggaran perusahaan

4. Gudang

a. Menyiapkan barang - barang dengan benar dan cepat sesuai dengan pesanan yang diterima.

b. Memeriksa barang–barang yang untuk konsumen.

c. Mempacking barang - barang yang akan dikirim dengan benar.

d. Membantu mencek perlengkapan/ jumlah/ kondisi atas barang-barang yang akan dikirim.

e. Mengembalikan semua peralatan dan barang yang telah selesai digunakan.

f. Turut menjaga kebersihan, keselamatan kerja dan keutuhan barang di dalam gudang.

g. MembantuProses bongkar muat barang.

h. Melaporkan hal – hal yang mencurigakan.

5. Pengiriman

a. Mengecek data barang yang ada berupa satuan atau pack

b. Menghitung jumlah pack dan menyesuaikan dengan faktur

c. Melakukan pengiriman barang .

3.2 Metode Penelitian

Metode penelitian merupakan suatu, teknik atau cara untuk mencari, memperoleh, mengumpulkan atau mencatat data yang dapat digunakan untuk keperluan menyusun penelitian.

Berikut ini merupakan metode penelitian yang dilakukan dalam melaksanakan penelitian.

3.2.1. Desain Penelitian

Dalam melakukan penelitian, desain penelitian merupakan kerangka atau perincian prosedur kerja yang akan dilakukan pada waktu meneliti, sehingga diharapkan dapat memberikan gambaran dan arah mana yang akan dilakukan dalam melaksanakan penelitian tersebut, desain penelitian yang baik dapat memudahkan kita dalam melakukan penelitian dan sesuai dengan tujuan yang telah ditetapkan sebelumnya

Sesuai dengan tujuan yang ingin dicapai dalam penelitian ini, penulis mengumpulkan data sebagai bahan laporan dengan menggunakan metode deskriptif, yaitu suatu metode yang pada tahap pertama penulis melakukan dengan cara mengumpulkan data dan bahan yang diperlukan terlebih dahulu dan pada tahap berikutnya, penulis mengolah dan membahas sampai pada suatu kesimpulan yang pada akhirnya dapat dibuat suatu laporan.

3.2.2. Jenis & Metode Pengumpulan Data

Metode yang digunakan penulis dalam pengumpulan data adalah dengan cara mengumpulkan data dari sumber data primer dan sumber data sekunder, berikut ini adalah penjelasan singkat mengenai metode yang penulis gunakan.

3.2.2.1. Sumber Data Primer

Data Primer adalah data yang diperoleh langsung dari unit pengamatan / responden penelitian. Data primer bisa di dapatkan dengan cara berikut:

1. Wawancara (*Interview*)

Metode wawancara merupakan sesi tanya jawab yang dilaksanakan penulis untuk mewawancarai dengan pihak yang terkait. Dalam teknis wawancara ini, penulis berperan sebagai pewawancara dan mewawancarai pihak-pihak yang berhubungan dengan penelitian yang penulis lakukan. Penulis melakukan wawancara langsung kepada Pimpinan CV. Karya Pratama Gemilang mengenai permasalahan yang dibahas yaitu mengenai Sistem Informasi Penjualan Buku.

Wawancara ini dilakukan untuk mengetahui beberapa kekurangan tentang Sistem Informasi Penjualan, baik itu masalah layanan penjualan maupun interaksi antar *user*.

2. Observasi

Observasi merupakan metode pengumpulan data dengan tanpa harus menggunakan respon dari subjek yang diteliti. Teknik ini memungkinkan untuk mendapatkan temuan-temuan data yang mendukung pada penelitian ini. Dalam penelitian ini observasi dilakukan pada Pimpinan CV.Karya Pratama Gemilang.

3.2.2.2. Sumber Data Sekunder

Data Sekunder adalah data yang digunakan untuk mendukung data primer salah satunya dengan mengumpulkan dokumen-dokumen yang dianggap penting untuk penulisan laporan penelitian.

Studi Pustaka

Metode yang digunakan dengan cara mencari referensi buku-buku yang dapat diperlukan sebagai landasan teori maupun yang berhubungan dengan masalah pembuatan program aplikasi. Adapun buku referensi yang penulis

gunakan untuk penyusunan landasan teori adalah buku-buku yang berasal dari perpustakaan, sebagai contoh adalah buku karangan Jogiyanto.Albahra.

3.2.3. Metode Pendekatan & Pengembangan Sistem

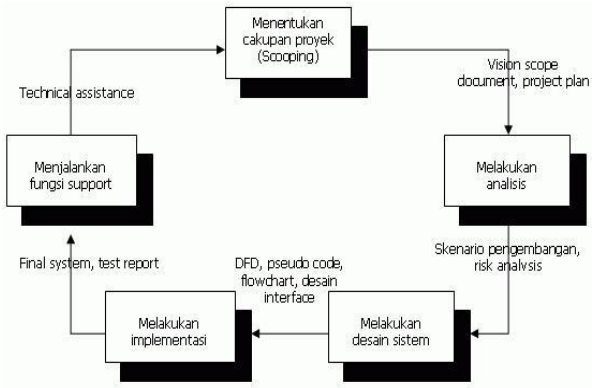
Metode pendekatan dan pengembangan sistem berisikan beberapa teknik yang digunakan dalam penelitian ini, diantaranya metode pengembangan sistem, metode pendekatan sistem, alat bantu analisis dan perancangan, dan pengujian software. Di bawah ini akan diuraikan tentang pemahaman dari metode tersebut.

3.2.3.1 Metode Pendekatan Sistem

Metode penelitian yang dilakukan adalah suatu metode deskriptif. Metode deskriptif adalah kegiatan menyimpulkan data mentah dalam jumlah yang besar sehingga hasilnya dapat ditafsirkan. Yang meliputi kegiatan mengelompokan, mengatur, mengurutkan data, atau memisahkan komponen atau bagian yang relevan dari keseluruhan data sehingga data mudah dikelola.

3.2.3.2 Metode Pengembangan Sistem

Dalam penerapan tahapan pengembangan sistem informasi penulis menggunakan sistem pengembangan *prototype*.. Berikut metodenya :



Gambar 3.2 Proses Prototype

Sumber: http://matakaca.blog.ugm.ac.id/files/2011/02/prototype_model.jpg

a) Pengumpulan kebutuhan

Pelanggan dan pengembang bersama-sama mendefinisikan format seluruh perangkat lunak, mengidentifikasikan semua kebutuhan.

b) Membangun *prototyping*

Membangun prototyping dengan membuat perancangan sementara yang berfokus pada penyajian kepada pelanggan (misalnya dengan membuat input dan format output)

c) Evaluasi *protoptyping*

Evaluasi ini dilakukan oleh pelanggan apakah prototyping yang sudah dibangun sudah sesuai dengan keinginann pelanggan. Jika sudah sesuai maka langkah D akan diambil.Jika tidak prototyping direvisi dengan mengulang langkah A, B , dan C.

d) Mengkodekan Sistem

Dalam tahap ini prototyping yang sudah disepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai

e) Menguji Sistem

Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, harus dites dahulu Pengujian ini dilakukan dengan pengujian Black Box.

f) Evaluasi Sistem

Pelanggan mengevaluasi apakah sistem yang sudah jadi sudah sesuai dengan yang diharapkan .

g) Menggunakan sistem

Perangkat lunak yang telah diuji dan diterima pelanggan siap untuk digunakan .

3.2.3.3. Alat Bantu Analisis dan Perancangan

Berikut ini adalah alat bantu analisis yang akan digunakan dalam merancang Sistem Informasi Penjualan:

Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C.

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax* atau semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (Object-Oriented Design), Jim Rumbaugh OMT (Object Modeling Technique), dan Ivar Jacobson OOSE (Object-Oriented Software Engineering).

A. *Use Case* diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Seoran atau sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

Use case diagram dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem. Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use case* yang meng-*include* dieksekusi secara normal. Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*. Sebuah *use case* juga dapat meng-*extend use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

B. Activity diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Activity diagram merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. *Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

C. Sequence diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar Sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan.

Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan

icon khusus untuk objek *boundary, controller* dan *persistent entity*.

D. Class Diagram

Class diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Class diagram menggambarkan struktur dan deskripsi *class, package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

Atribut dan metoda dapat memiliki salah satu sifat berikut :

a. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan
b. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya
c. *Public*, dapat dipanggil oleh siapa saja

Hubungan Antar *Class*

1. Asosiasi, yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain. Panah *navigability* menunjukkan arah *query* antar *class*.
2. Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas..”).
3. Pewarisan, yaitu hubungan hirarkis antar *class*. *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metoda *class* asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari *class* yang

diwarisinya. Kebalikan dari pewarisan adalah generalisasi.

4. Hubungan dinamis, yaitu rangkaian pesan (*message*) yang di-*passing* dari satu *class* kepada *class* lain. Hubungan dinamis dapat digambarkan dengan menggunakan *sequence diagram* yang akan dijelaskan kemudian.

E. *Deployment* Diagram

Deployment diagram menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisikal Sebuah *node* adalah server, *workstation*, atau piranti keras lain yang digunakan untuk men-*deploy* komponen dalam lingkungan sebenarnya. Hubungan antar *node* (misalnya TCP/IP) dan *requirement* dapat juga didefinisikan dalam diagram ini Perangkat Yang Mendukung UML.

Saat ini banyak sekali *tool* pendesainan yang mendukung UML, baik itu *tool* komersial maupun *opensource*. Beberapa diantaranya adalah:

1. *Rational Rose*
2. *Astah*
3. *Object Domain*
4. *Jvision*
5. *Objecteering*
6. *MagicDraw*
7. *Visual Object Modeller*

F. *Componen Diagram*

Bersifat statis. Diagram ini memperlihatkan organisasi serta ketergantungan sistem atau perangkat lunak pada komponen-komponen yang telah ada sebelumnya. Diagram ini berhubungan dengan diagram kelas dimana komponen secara tifikal dipetakan kedalam satu atau lebih kelas-kelas.

3.2.4. Pengujian Software

Dalam pembuatan suatu program baik berupa sistem informasi atau bukan,membutuhkan suatu teknik pengujian yang berguna dalam mengevaluasi program yang dibuat. Menurut Janner Simarmata (2010:300) Adapun teknik pengujian program dibagi menjadi dua, yaitu teknik pengujian perangkat lunak *white box* dan teknik pengujian *black box*.

a. White Box

Kadang disebut pula pengujian *glass box*, adalah metoda desain *test case* yang menggunakan struktur kontrol desain prosedural untuk memperoleh *test case* atau dengan kata lain bahwa pengujian dilakukan untuk memastikan bahwa operasi *internal* bekerja sesuai dengan spesifikasi dan semua komponen internal telah diamati dengan baik. Dengan menggunakan metoda pengujian ini perekayasa sistem dapat melakukan *test case* yaitu :

1. Memberi jaminan bahwa semua jalur *independent* pada suatu modul telah digunakan paling sedikit satu sekali.
2. Menggunakan semua keputusan logis pada sisi *true* dan *false*
3. Mengeksekusi semua *loop* sesuai dengan batasan
4. Menggunakan struktur data *internal* untuk menjamin valida

b. Black Box

Konsep Kotak Hitam (*Black Box*) digunakan merepresentasikan sistem cara kerja di dalamnya tidak tersedia untuk diinspeksi. Di dalam Kotak Hitam (*Black Box*), item-item yang diuji dianggap ”gelap” k yang diketahui hanya apa yang masuk dan apa yang keluar dari kotak hitam.

Pada pengujian Kotak Hitam (*Black Box*), kasus-kasus pengujian berdasarkan pada spesifikasi sistem. Rencana pengujian dapat dimulai sendiri mungkin di proses pengembangan perangkat lunak.

Teknik pengujian konvensional yang termasuk pengujian Kotak Hitam (*Black Box*) adalah sebagai berikut :

1. *Graph based testing* (pengujian yang didasarkan pada grafik)
2. *Equivalence partitioning* (menyekat persamaan)
3. *Comparison testing* (pengujian perbandingan)
4. *Orthogonal testing* (pengujian orthogonal)