BAB III

OBJEK DAN METODE PENELITIAN

3.1. Objek Penelitian

Objek penelitian adalah hal awal yang harus ditentukan dalam kegiatan penelitian sehingga penelitian dapat dilakukan secara efektif dan efisien sesuai dengan tujuan penelitian. Pemilihan dan penentuan objek penelitian yang tepat diharapkan dapat menunjang kegiatan selama penelitian, sehingga hal-hal yang diperlukan dalam penelitian akan mudah dicapai. Objek penelitian ini dilakukan di Strawbery's Salon yang beralamat di jalan Abdul Rahman Saleh no 12 Bandung.

3.1.1. Sejarah Singkat Perusahaan

Salon strawbery's didirikan pada tahun 2008 tepatnya 15 juni 2008.Pada awalnya salon ini berada di dalam gang lama kelamaan salon strawbery's berpindah tempat ke tempat yang lebih stategis dan gampang untuk di cari yang beralamat di jalan Abdul Rahman Saleh no 12 pemiliknya bernama Veylani dan salon ini memiliki 10 pegawai. Salon ini buka setiap hari mulai pukul 09.00 pagi s/d jam 18.00 sore. menawarkan berbagai macam perawatan Salon dan Spa dengan harga yg sangat terjangkau untuk segala kalangan

Dengan konsep seperti salon di korea yang meyajikan tata ruangan berwarna merah dan pink dengan di hiasi gambar ,model-model remaja yag ada di korea.Yang membuat para pelanggan betah berlama-lama berada di salon ini dan para pelayannya pun sangat ramah dalam memberikan pelayanan terhadap konsumen.

3.1.2. Visi dan Misi Strawbery's Salon

Visi:

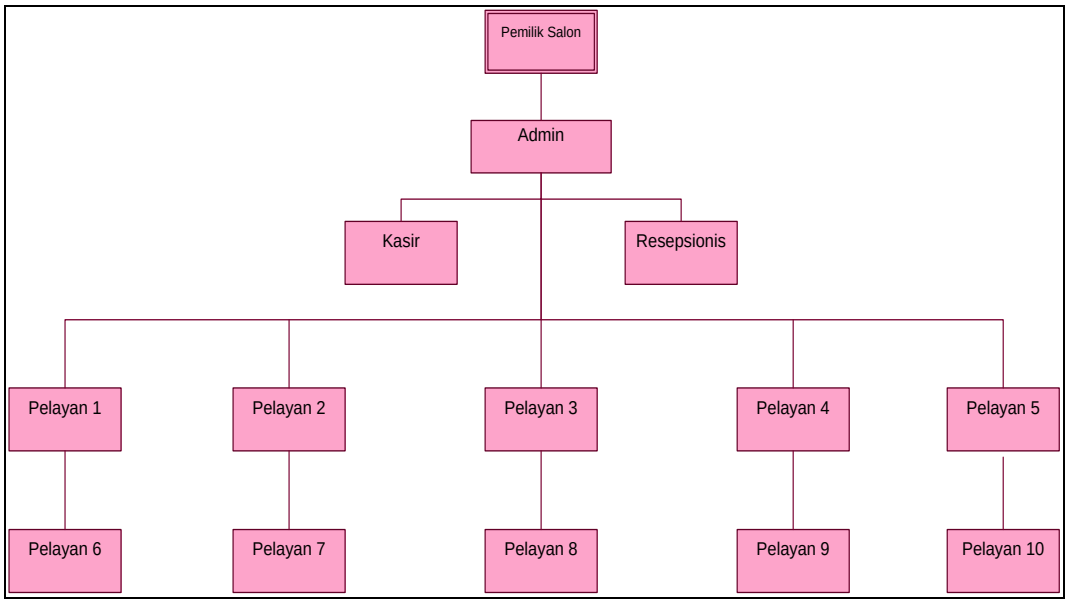
1. Memberikan pelayanan kecantikan,tata rias rambut,tata rias wajah dengan mengedepankan kepuasan pelanggan
2. Menjadikan Strawbery's unggul dalam mutu dan pelayanan
3. Mensejahterakan Karyawan

Misi:

1. Meningkatkan kualitas pelayanan kepada pelanggan secara terus-menerus
2. Menggunakan produk-produk berkualiatas dan aman dalam memberikan pelayanan/perwatan.
3. Memegang teguh prinsip kejujuran dan kedisiplinan dalam memberikan pelayanan.
4. Menjadikan kepuasan pelanggan sebagai tolak ukur keberhasilan salon
5. Menjadikan seluruh pelanggan Strawbery's Salon sebagai bagian yang terpenting dari Strawbery's Salon.
6. Komitmen kuat dalam menjaga kualitas pelayanan
7. Memberikan "Garansi pelayanan pasti memuaskan, gratis perawatan ulang bila pelanggan tidak puas".
8. Berusaha untuk memberikan manfaat yang positif bagi lingkungan di sekitar Strawbery's Salon.

3.1.3. Struktur Organisasi Perusahaan

Struktur organisasi merupakan landasan organisasi untuk menentukan pembagian tugas, tanggung jawab dan wewenang secara jelas. Sehingga koordinasi struktur dapat dilaksanakan dengan baik guna menunjang aktifitas salon. Dibawah ini adalah struktur organisasi yang ada pada Strawbery's Salon.



Gambar 3.1. Struktur Organisasi Strawbery's Salon

3.1.4. Deskripsi Tugas

1. Owner atau pemilik

a) Mengelola jalannya perusahaan agar tujuan dan hasilnya sesuai dengan kebutuhan perusahaan

b) Menerima laporan pelayanan salon

c) Bertanggung jawab atas seluruh kegiatan organisasi.

2. Admin

a) Bertugas mengelola sistem diantaranya mengupdate perawatan baru ,paket perawatan ,data pegawai baru

b) Bertugas membuat kartu pegawai (kasir)

3. Kasir

a) Bertugas melakukan perhitungan pembayaran di sistem dan mencetak struk pembayaran

b) Bertugas melakukan pembuatan laporan baik itu laporan perawatan salon(pendapatan salon) lalu di berikanya ke pemilik salon dan pembuatan laporan member (perawatan apa saja yang sudah di lakukan oleh member tersebut)

4. Resepsionis

a) Bertugas menyambut pelanggan yang datang ke salon lalu mengonputkan pelayanan apa saja yang di inginkan pelanggan lalu mencetak stuk nomor antrian.

b) Bertugas menawarkan pelanggan untuk menjadi member di salon strawbery’s

5.Pelayan

a) Hanya bertugas melakukan pelayanan yang di inginkan pelanggan tanpa berinteraksi dengan sistem

3.2. Metode Penelitian

Metode ialah suatu kerangka kerja untuk melakukan suatu tindakan, atau suatu kerangka berfikir menyusun gagasan, yang beraturan, berarah dan berkonteks, yang terkait (*relevant*) dengan maksud dan tujuan.

Penelitian adalah cara ilmiah untuk mendapatkan tujuan dan kegunaan tertentu. Seorang peneliti harus menentukan metode yang akan dipergunakan, dan ditentukannya metode penelitian, maka akan memandu seseorang peneliti mengenai urutan-urutan bagaimana penelitian dilakukan.

Metode penelitian merupakan suatu mekanisme, teknik atau tatacara bagaimana suatu penelitian dilaksanakan. Metode penelitian dianggap sebagai seperangkat pendekatan menyeluruh untuk mengumpulkan data dan menganalisis masalah-masalah tertentu mencakup teknik dan alat. Karena berupa sistem maka metode merupakan seperangkat unsur-unsur yang membentuk suatu kesatuan.

Metode penelitian deskriptip merupakan metode penelitian yang berusaha menggambarkan dan menginterpretasikan objek sesuai dengan apa adanya, menguraikan keadaan situasi pada tempat observasi, sehingga setelah dilakukannya penelitian kemudian melakukan analisis sehingga dapat diperoleh suatu kesimpulan. Maka dari itu untuk mendapatkan hasil penelitian untuk sesuai dengan gambaran atau interprestasi objek sesuai dengan apa adanya, Dalam penyusunan laporan ini, penulis menggunakan metode penelitian deskriptif

3.2.1. Desain Penelitian

Desain penelitian yang digunakan dalam penelitian ini bersifat deskriptif. Penelitian deskriptif adalah penelitian yang diarahkan untuk memberikan gejala-gejala,

fakta-fakta, atau kejadian-kejadian secara sistematis dan akurat mengenai sifat-sifat populasi atau daerah tertentu. Dalam hal ini berupa sistem informasi yang sedang berjalan serta entitas-entitas yang terkait dalam perusahaan atau instansi yang menjadi objek penelitian. Metode deskriptif dilakukan dengan cara mengumpulkan informasi aktual secara rinci mengenai masalah yang akan diteliti.

Berdasarkan uraian di atas penulis beranggapan bahwa metode deskriptif sesuai digunakan dalam penelitian tentang perancangan sistem informasi pelayanan pelanggan s, karena metode deskriptif dapat mendeskripsikan data yang aktual berupa gambaran dari masalah yang diteliti dan dapat mengetahui fenomena-fenomena yang terjadi pada sistem informasi yang berjalan di perusahaan atau instansi.

3.2.2. Jenis dan Metode Pengumpulan Data

Dalam melakukan penelitian ini penulis menggunakan metode deskriptif yaitu suatu metode dalam meneliti status sekelompok manusia suatu objek ,suatu set kondisi,suatu sistem penukiran ataupun mengenai fakta-fakta dan informasi dalam situasi atau kejadian dimasa sekarang secara sistematis faktual dan akurat.

3.2.2.1. Sumber Data Primer

Sumber data primer yaitu berupa teknik pengumpulan data yang diigunakan adalah sebagai berikut

1. Observasi

   Dalam hal ini penyusun melakukan pengamatan pada Strawbery’s salon untuk mendapatkan data secara umum dengan melihat langsung dan mengamati.

2. Wawancara

Dalam hal ini penyusun melakukan wawancara untuk melengkapi bahan yang sudah ada selama observasi. Penyusun melakukan tanya jawab langsung kepada pihak salon

3. Studi Pustaka.

Studi pustaka yaitu teknik pengumpulan data dan informasi melalui buku-buku dan sumber tertulis lainnya yang berhubungan dengan bidang penelitian, sehingga hasilnya dapat dijadikan sebagai dasar untuk penelitian yang dilaksanakan. Studi pustaka yang digunakan pada penelitian ini adalah karya-karya ilmiah dan buku-buku yang membahas tentang sistem informasi perkuliahan atau kegiatan belajar mengajar di instansi pendidikan, salah satunya universitas.

3.2.2.2. Sumber Data Sekunder

Sedangkan data sekunder diperoleh dengan cara melakukan kajian kepustakaan, yaitu mengkaji buku-buku literature, makalah, informasi melalui internet yang *relevan* dengan *variable* yang diteliti selain itu sumber data juga diperoleh dari instansi terkait yang ada hubungannya dengan *variable* yang digunakan dalam penelitian ini. Yang menjadi sumber dokumen sekunder penulis.

3.2.3. Metode Pendekatan dan Pengembangan Sistem

Adapun metode pengembangan dan pengembangan dari penelitian yang dilakukan sebagai berikut

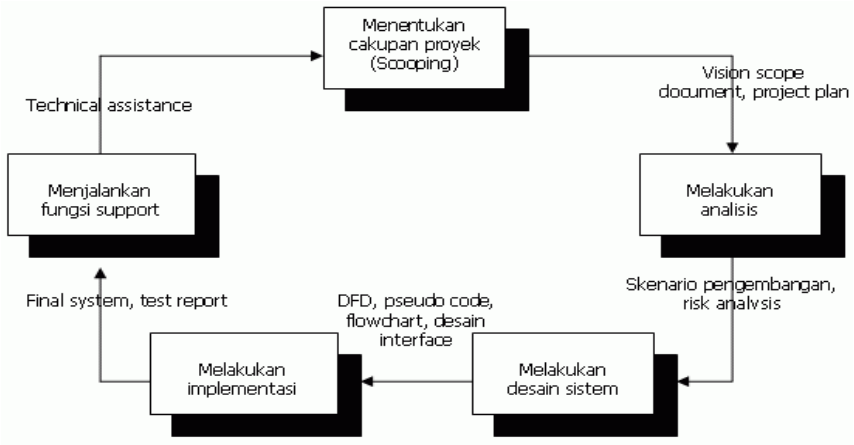
3.2.3.1. Metode Pendekatan Sistem

Metode Pendekatan sistem yang akan digunakan metode analisis dan pemrograman berorientasi objek. Alat-alat yang digunakan dalam pendekatan analisis dan pemrograman berorientasi objek yaitu dengan notasi UML dengan membuat enam

diagram yaitu, *Use case diagram, Activity diagram, Class diagram, Sequence diagram, Component diagram, Deployment diagram*. Sedangkan perancangan program dimulai dari *Design,* Perancangan, *Cooding*, *Testing*.

3.2.3.2. Metode Pengembangan Sistem

Dalam perancangan Sistem Informasi pelayanan pelanggan ini, penulis menggunakan metode *prototyping* yang merupakan salah satu metode pengembangan perangkat lunak yang banyak digunakan. Dengan metode *prototyping* ini pengembang dan pelanggan dapat saling berinteraksi selama proses pembuatan sistem. Desain penelitian ini dimodelkan dengan menggunakan model proses *prototype*. Model proses *prototype* merupakan suatu metode dalam pengembangan sistem yang menggunakan pendekatan untuk membuat sesuatu program dengan cepat dan bertahap sehingga dapat segera dievaluasi oleh pemakai *(user).* Berikut gambar dari metode pengembangan yang digunakan penulis :



Gambar 3.2 Metode Pendekatan *Prototype* Paradigma

(Sumber : Pressman, Roger S., 2002, *Rekayasa Perangkat Lunak*: Pendekatan Praktisi jilid Dua, Penerbit: Andi Offset, Yogyakarta).

Dari pengertian metode *prototype* diatas, penulis akan memberikan beberapa alasan mengapa penulis menggunakan metode pengembangan sistem dengan *prototype*, yaitu dikarenakan penulis akan lebih mudah dalam merancang sistem yang diinginkan dan dapat diterima oleh *user* sebagai pemakai.

Berikut ini beberapa tahapan-tahapan dalam Prototyping yang akan dipakai oleh penulis dalam merancang sebuah sistem, yaitu sebagai berikut:

a. Pengumpulan kebutuhan, User dan pengembang bersama-sama mendefinisikan format seluruh perangkat lunak, mengidentifikasikan semua kebutuhan, dan garis besar sistem yang akan dibuat.

b. Membangun prototyping, Membangun prototyping dengan membuat perancangan sementara yang berfokus pada penyajian kepada user (misalnya dengan membuat input dan format output)

c. Evaluasi protoptyping, Evaluasi ini dilakukan oleh user apakah prototyping yang sudah dibangun sudah sesuai dengan keinginann user. Jika sudah sesuai maka langkah 4 akan diambil. Jika tidak prototyping direvisi dengan mengulangu langkah 1, 2 , dan 3.

d. Mengkodekan sistem, Dalam tahap ini prototyping yang sudah di sepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai.

e. Menguji sistem, Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, harus dites dahulu sebelum digunakan. Pengujian ini dilakukan dengan White Box, Black Box, Basis Path, pengujian arsitektur dan lain-lain.

f. Evaluasi sistem, User mengevaluasi apakah sistem yang sudah jadi sudah sesuai dengan yang diharapkan . Jika ya, langkah 7 dilakukan; jika tidak, ulangi langkah 4 dan 5.

g. Menggunakan sistem, Perangkat lunak yang telah diuji dan diterima user siap untuk digunakan.

A. Keunggulan dan Kelemahan *Prototyping.*

Keunggulan *prototyping* adalah:

1. Adanya komunikasi yang baik antara pengembang dan pelanggan

2. Pengembang dapat bekerja lebih baik dalam menentukan kebutuhan pelanggan

3. Pelanggan berperan aktif dalam pengembangan sistem

4. Lebih menghemat waktu dalam pengembangan sistem

5.Penerapan menjadi lebih mudah karena pemakai mengetahui apa yang diharapkannya.

Kelemahan *prototyping* adalah :

Pelanggan kadang tidak melihat atau menyadari bahwa perangkat lunak yang ada belum mencantumkan kualitas perangkat lunak secara keseluruhan dan juga belum memikirkan kemampuan pemeliharaan untuk jangka waktu lama.

3.2.3.3. Alat Bantu Analisis dan Perancangan

a. *Unifed Modeling Language* (UML)

Menurut Bambang Hariyanto (2004 : 259) *UML adalah bahasa grafis untuk mendokumentasi,menspesifikasikan , dan membangun sistem perangkat lunak.*

UML berorientasi objek,menerapkan banyak level abstraksi, tidak bergantung proses pengembangan,tidak bergantung bahasa dari teknologi,pemaduan beberapa notasi diberagam metodologi,usaha bersama dari banyak pihak,didukung oleh kakas-kakas yang

diintegrasikan lewat XML.Standar UML dikelola oleh OMG(Object Management Group).

b. Diagram – Diagram UML

Diagram mengemukakan banyak hal,penggunaan notasi yang terdefinisi baik dan ekspresif adalah penting pada proses perangkat lunak :

1. Notasi standar memungkinkan pengembangan mendeskripsikan scenario atas rumusan arsitektur dan kemudian mengkomunikasikan secara tidak ambigu.
2. Notasi yang bagus membebaskan otak untuk berkonsentrasi pada masalah-masalah yang lebih lanjut.
3. Notasi yang baik memungkinkan mengeliminasi keperluan pemeriksaan konsistensi dan kebenaran keputusan-keputusan dengan menggunakan tool terotomatisasi.

1. *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. Seorang/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 duplikasi fungsionalitas 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.

2. *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.Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

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.

3. *Class Diagram*

Class 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.

Class memiliki tiga area pokok :

1. Nama dan *stereotype*

2. Atribut

3. *Method*

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.

Class dapat merupakan implementasi dari sebuah *interface*, yaitu *class* abstrak yang hanya memiliki metoda. *Interface* tidak dapat lang*Sun*g diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class*. Dengan demikian *interface* mendukung resolusi metoda pada saat *run-time*.Sesuai dengan perkembangan *class*

model, *class* dapat dikelompokkan menjadi *package*. Kita juga dapat membuat diagram yang terdiri atas *package*.

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.

4. *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. Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal.

Message digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah message. Untuk objek-objek yang memiliki sifat khusus, standar *UML* mendefinisikan *icon* khusus untuk objek *boundary*, *controller* dan *persistent entity*.

5. *Component* Diagram

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (dependency) diantaranya.Komponen piranti lunak adalah modul berisi code,baik berisi source code maupun binary code,baik *library* maupun *executabl*e,baik yang muncul pada *compile time* ,*link time* , maupun *run time.*Umumnya komponen terbentuk dari beberapa class dam/atau package, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa interface yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

6. *Deployment* Diagram

Diagram *deployment* digunakan untuk memodelkan aspek fisik dari sistem beriorentasi objek,yaitu memodelkan konfigurasi *node - node* pengolahan waktu jalan dan komponen-komponen yang tinggal di *node-node* itu.

3.2.4. Pengujian Software

Dalam pengujian software, penulis menggunakan pengujian ***black box*** karena lebih dipahami dan lebih mudah diketahui letak kesalahannya baik dari proses Programnya maupun tampilan atau output yang dihasilkan.

Pengujian *black box* :

1. Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak.
2. Disebut juga pengujian *behavioral* atau pengujian partisi.
3. Pengujian *blackbox* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program.
4. Pengujian *blackbox* berusaha menemukan :
5. Fungsifungsi yang tidak benar atau hilang
    a. Kesalahan *interface.*
    b. Kesalahan dalam struktur data atau akses *database* eksternal.
    c. Kesalahan kinerja.
    d. Inisialisasi dan kesalahan terminasi.
6. Dengan mengaplikasikan teknik blackbox maka kita menarik serangkaian *test case* yang memenuhi kriteria beirkut :
    a. *Test case* yang mengurangi, dengan harga lebih dari satu, jumlah *test case* tambahan yang harus di desain untuk mencapai pengujian yang dapat dipertanggungjawabkan.
    b. *Test case* yang member tahu kita sesuatu mengenai kehadiran atau ketidakhadiran kelas kesalahan, daripada member tahu kesalahan yang berhubungan hanya dengan pengujian