BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem

Menurut GOR [1] "*Sistem terdiri dari bagian – bagian yang beroperasi bersama untuk mencapai berberapa tujuan"*.

Dengan kata lain suatu sistem bukanlah merupakan suatu perangkat unsur – unsur yang dapat di identifikasikan sebagai kebersamaan yang menyatu, yang disebabkan tujuan atau sasaran yang sama.

Sedangkan menurut JOG [2] *"Sistem merupakan suatu jaringan kerja dari prosedur – prosedur yang saling berhubungan, berkumpul bersama – sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu"*.

Menurut JOG [2] Suatu sistem mempunyai karakteristik (sifat – sifat tertentu), yaitu :

1. Komponen - komponen *(Components)*

   Suatu sistem terdiri dari jumlah komponen yang saling berinteraksi, bekerjasama untuk membentuk suatu kesatuan komponen – komponen sistem berupa sub-sistem atau bagian – bagian dari sistem.

2. Batasan Sistem *(Boundary)*

   Merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini dapat dipandang

sebagai satu kesatuan dan menunjukan ruang lingkup *(scope)* dari sistem tersebut.

3. Lingkungan Luar Sistem *(Environment)*

   Segala sesuatu diluar batas dari sistem yang mempengaruhi operasi sistem yang bersifat menguntungkan atau merugikan.

4. Penghubung *(Interface)*

   Merupakan media penghubung antara satu sub-sistem dengan sub-sistem yang lainnya. Melalui penghubung ini memungkinkan sumber daya – sumber daya mengalir dari sub-sistem yang lainnya.

5. Masukan *(Input)*

   Energi yang dimasukan ke dalam sistem yang berupa masukan perawatan *(Maintenace Input)* agar sistem tersebut dapat beroperasi dan masukan sinyal *(Signal Input)* yang diproses unutk menghasilkan keluaran.

6. Keluaran *(Output)*

   Merupakan hasil dari energi yang telah diproses dan dapat diklarifikasikan antara lain menjadi output yang langsung digunakan, output yang bisa dijadikan input untuk sub-sistem yang lain atau kepada supra sistem serta output bagi sistem yang tidak digunakan lagi (sisa pembuangan) tapi bagi sistem lain sangat berguna.

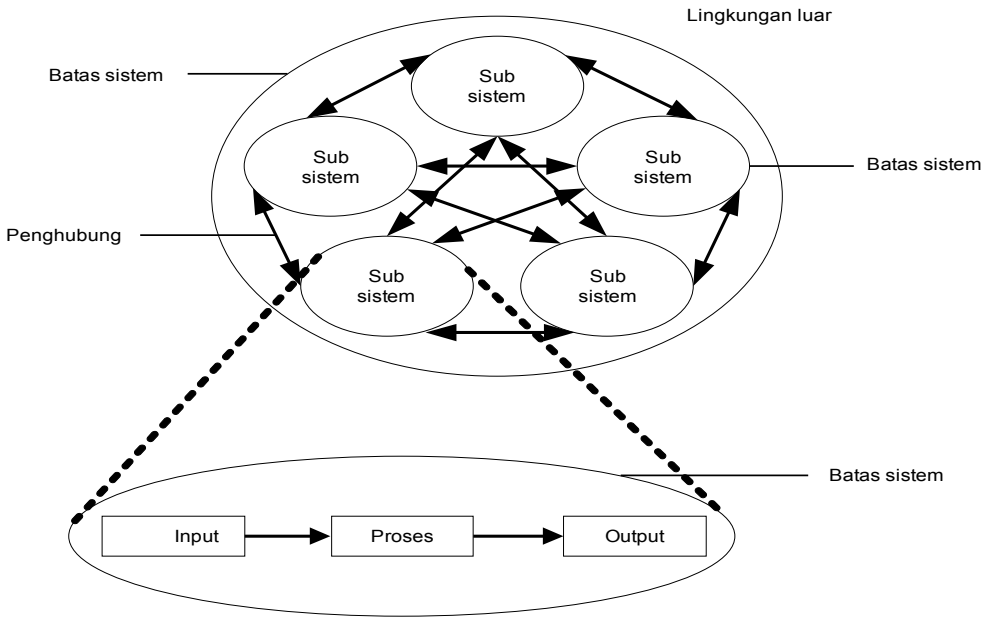
7. Pengolah *(Process)*

   Suatu sistem dapat mempunyai bagian pengolah yang akan merubah masukan menjadi keluaran.

8. Sasaran *(Objectives)* dan Tujuan *(Goal)*

Suatu sistem dikatakan berhasil jika mengenai sasaran atau tujuannya.

Karakterisitk atau sifat-sifat suatu sistem dapat kita lihat seperti pada gambar berikut ini :



Gambar 2.1 : Karakteristik Suatu Sistem

Sumber : Jogiyanto, HM, 2002, *Analisis & Desain Sistem Informasi,*

2.2 Konsep Dasar Informasi

Informasi adalah data yang sudah di proses dan di analisa sehingga mempunyai nilai yang berarti. Sumber dari informasi adalah data. Data merupakan bentuk jamak dari bentuk tunggal data *item*.

Informasi adalah *"Data yang telah diproses kedalam suatu bentuk tertentu yang mempunyai nilai nyata dan terasa bagi keputusan saat ini dan kebutuhan mendatang."* GOR [1]

Sedangkan menurut JOG [2] *"informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya"*

Agar bermanfaat, informasi harus memiliki kualitas sebagai berikut :

1. Relevan, yaitu menambah pengetahuan atau nilai bagi para pembuat keputusan, dengan cara mengurangi ketidakpastian, menaikkan kemampuan untuk memprediksi, atau menegaskan ekspektasi semula.
2. Dapat Dipercaya, yaitu bebas dari kesalahan atau bisa secara akurat menggambarkan kejadian atau aktivitas organisasi.
3. Lengkap, yaitu tidak menghilangkan data penting yang dibutuhkan oleh para pemakai.
4. Tepat waktu, yaitu disajikan pada saat yang tepat untuk mempengaruhi proses pembuatan keputusan.
5. Mudah dipahami, yaitu disajikan dalam format yang mudah dimengerti.
6. Dapat diuji kebenarannya, yaitu memungkinkan dua orang yang kompeten untuk menghasilkan informasi yang sama secara *independent.*

2.3 Konsep Dasar Sistem Informasi

Sistem informasi dapat diartikan sebagai suatu cara mengelola pekerjaan informasi dengan menggunakan pendekatan sistem. Sistem informasi merupakan

suatu proses sistem manusia dan mesin yang terintegrasi secara harmonis untuk menyediakan informasi yang dapat digunakan oleh *user* untuk mendukung fungsi operasi, manajemen dan pengambilan keputusan di dalam organisasi baik secara manual maupun dengan kemampuan teknologi informatika *(komputer).*

"Sistem Informasi adalah suatu sistem dari suatu sistem organisasi yang mempertemukan pengolahan transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dan laporan yang diperlukan" GOR [1]

Sistem Informasi merupakan proses yang diselenggarakan untuk mendapatkan informasi melalui suatu prosedur yang memanfaatkan sumber daya manajemen kedalam suatu keperluan tertentu.

Komponen sistem informasi adalah sebagai berikut :

1. *Hardware* (perangkat keras), terdiri dari komputer, printer, jaringan
2. *Software* (perangkat lunak)
3. Data, merupakan komponen dasar informasi
4. Manusia *(user)*
5. Prosedur, seperti buku penyusun operasional dan elektronik.

Kegiatan - kegiatan di dalam suatu informasi meliputi :

1. *Input*, kegiatan menyediakan data untuk diproses.
2. Proses, menggambarkan bagaimana suatu data diproses untuk menghasilkan informasi yang bernilai lebih.

3. *Output,* kegiatan yang menghasilkan keluaran berupa informasi hasil dari proses.
4. Penyimpanan, kegiatan untuk memelihara dan menyimpan data.
5. Kontrol (pengendali), suatu aktifitas untuk menjamin bahwa sistem informasi berjalan sesuai dengan jalan yang diharapkan.

2.4 Metode Pengembangan Sistem

Metode pengembangan sistem adalah menyusun suatu sistem yang baru untuk menggantikan sistem yang lama secara keseluruhan atau memperbaiki sistem yang telah ada.

Dalam metode pengembangan sistem ini, penulis menggunakan metode SDLC *(Systems Development Life Cycle). Menurut DAV [1], pada umumnya harus melalui beberapa tahap, diantaranya :*

1. Perencanaan

   Pada fase ini adalah fase kebikajsanaan, atau persiapan pra proyek dalam merancang/ mengembangkan satu sistem.

2. Analisis

   Fase ini adalah fase yang sangat penting dimana pada fase ini akan diputuskan sistem seperti apa yang akan dibuat, menilai masih layak tidaknya suatu sistem. Dengan memodelkan sistem dan memilih aternatif sistem yang paling baik.

3. Perancangan

   Fase perancangan adalah fase lanjutan dari analisis, dimana alaternatif yang sistem yang dipilih di desain baik untuk, Input, Output, Database dan File semuanya diintegrasikan agar biasa terimplementasi secara baik.

4. Implementasi

   Pada fase ini secara garis besar terdiri dari tiga bentuk kegiatan yaitu:

   a. *Programming and testing*. Yaitu mengkonversikan hasil perancangan ke dalam kegiatan operasi dengan menggunakan bahasa pemograman tertentu, dan mengetest semua program dan memastikan semua fungsi dapat berjalan secara benar.
   b. *Training*. Yaitu pelatihan sekaligus pengenalan sistem yg baru kepada user, dengan modul panduan sistem.
   c. *System hangeover*. Yaitu penggantian sistem lama ke dalam sistem yang baru dan kegiatan ini merupakan tanggung jawab tim desainer.

2.5 Pengenalan Object Oriented

Object oriented mencakup bidang aplikasi yang sangat luas. Para pengguna sistem komputer dan sistem lain yang didasarkan pada teknologi komputer merasakan efek *object oriented* dalam bentuk meningkatnya aplikasi *software* yang mudah digunakan dan servis yang lebih fleksibel, yang muncul dalam berbagai bidang industri. Sedangkan bagi *software engineer, objectoriented* berpengaruh dalam

bahasa pemrograman, metodologi rekayasa, manajemen proyek, *hardware* dan lain sebagainya.

Sebuah model objek menangkap struktur statis dari sistem dengan menggambarkan objek dalam sistem, hubungan antara objek, serta atribut dan operasi yang merupakan karakteristir setiap kelas dan objek. Model berorientasi objek lebih mendekati keadaan nyata.

1. Objek dan kelas

   Objek

   Objek didefinisikan sebagai konsep, abstraksi atau benda dengan batasan dan arti untuk suatu masalah. Objek menjelaskan dua tujuan, yaitu pengertian tentang dunia nyata dan dilengkapi dengan dasar praktek untuk implementasi komputer.

   Kelas

   Istilah kelas sering digunakan dari pada kelas - objek. Objek dalam kelas mempunyai atribut dan pola perilaku yang sama. Suatu kelas objek menggambarkan kumpulan dari objek yang mempunyai sifat (atribut), perilaku umum (operasi), relasi umum dengan objek lain. Objek dan kelas sering sama sebagai benda dalam deskripsi masalah.

2. Diagram Objek

   a. Kelas dan Objek

      Kelas dan objek adalah suatu kelas dan objek - objek yang berada pada kelas tersebut. Nama kelas adalah kata benda tinggal, atau kata sifat dan

kata benda. Nama dari kelas dan objek harus dapat menjelaskan objek tunggal dari suatu kelas.

b. Struktur objek dan hirarki kelas

Struktur kelas dibagi menjadi 2 macam, yaitu *Whole-Part Structure* dan *Gen-Spec Structure. Whole-Part Structure* memperlihatkan hirarki dari suatu kelas sebagai komponen dari kelas lain yang juga disebut sub objek. *Gen-Spec Structure* memperlihatkan kelas sebagai spesialisasi dari kelas diatasnya. Kelas yang mempunyai sifat umum disebut *generalization, superclass* atau *topclass,* sedangkan kelas yang mempunyai sifat khusus disebut *specialization.*

c. *Atribut*

Atribut menggambarkan data yang dapat memberikan informasi mengenai kelas atau objek dimana atribut tersebut berada.

d. *Metoda*

Metoda *(method)* disebut juga servis atau operator. Metoda adalah sub program yang tergabung dalam objek bersama - sama dengan atribut.

e. *Message*

Message (pesan) merupakan cara untuk berhubungan antara satu objek dengan objek lain.

2.6 Pengenalan UML

UML adalah notasi visual untuk menggambarkan konsep berorientasi objek. UML merupakan bahasa pemodelan yang paling sukses dari 3 (tiga) metode OO yang telah ada sebelumnya, yaitu *Booch, OMT,* dan *OOSE. UML* merupakan kesatuan dari ketiga metode pemodelan tersebut dan ditambah kemampuan lebih karena mengandung metode tambahan untuk mengatasi masalah pemodelan yang tidak dapat ditangani ketiga metode sebelumnya.

2.6.1 Pengertian UML

UML *(Unified Modeling Language)* adalah bahasa untuk menspesifikasi, memvisualisasi, membangun dan mendokumentasikan *artifacts* (bagian dari informasi yang digunakan atau dihasilkan oleh proses pembuatan perangkat lunak, *artifact* tersebut dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya. Selain itu UML adalah bahasa pemodelan yang menggunakan konsep orientasi objek. UML menyediakan notasi – notasi yang membantu memodelkan sistem dari berbagai perspektif. UML tidak hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan.

2.6.2 Bagian - bagian UML

Bagian – bagian utama dari UML *(Unified Modeling Languag)* yaitu :

1. *View*

View digunakan untuk melihat sistem yang dimodelkan dari beberapa aspek yang berbeda. *View* bukan melihat grafik, tapi merupakan suatu abtraksi yang berisi sejumlah diagram.

Beberapa jenis *view* dalam UML antara lain : *use case view, logical view, component view,* dan *deployment view.*

a) *Use case view*

Use case view digunakan untuk memahami dan menggunakan sistem yang kita modelkan. *View* ini melihat pada bagaiman actor dan *use case* beinteraksi.

b) *Logical view*

Logical view mengarah pada persyaratan *(requirement)* fungsional sistem. *View* ini melihat pada kelas - kelas dan hubungan antar kelas - kelas tersebut

c) *Component view*

Component view mengarah pada pengaturan *software. View* ini mengandung informasi mengenai komponen - komponen *software,* komponen tereksekusi *(executable* dan *library)* untuk sistem yang kita modelkan.

d) *Deployment view*

Deployment view memperlihatkan pemetaan setiap proses kedalam *hardware. View* ini paling bermanfaat ketika kita membuat model suatu

sistem yang diterapkan dalam lingkungan arsitektur yang terdistribusi dimana kita menerapkan aplikasi dan *server* pada lokasi yang berbeda.

2. *Diagram*

   Diagram berbentuk grafik yang menunjukkan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah diagram merupakan bagian dari suatu *view* tertentu dan ketika digambarkan biasanya dialokasikan untuk *view* tertentu.

Adapun jenis diagram antara lain: *use case diagram, class diagram, statechart diagram, sequence diagram, collaboration diagram, activity diagram, component diagram, deployment diagram.*

a) *Use case diagram*

   Menjelaskan manfaat sistem jika dilihat menurut pandangan orang yang berada diluar sistem *(actor)*. Diagram ini menunjukan fungsionalitas suatu sistem atau kelas dan bagaimana sistem berinteraksi dengan dunia luar. *Use case* diagram dapat digunakan selama proses analisis untuk menangkap *requirements sistem* dan untuk memahami bagaimana sistem seharusnya bekerja.

b) *Class diagram*

   Membantu dalam visualisasi struktur kelas - kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. *Class diagram* memperlihatkan hubungan antar kelas dan penjelasan detail tiap - tiap kelas didalam model desain dari suatu sistem.

c) *Statechart diagram*

Digunakan untuk memodelkan perilaku dinamis satu kelas atau objek. *Statechart* diagram memperlihatkan urutan keadaan sesaat *(state)* yang dilalui sebuah objek, kejadian yang menyebabkan sebuah transisi dari satu *state* atau aktivitas kepada yang lainnya, dan aksi yang menyebabkan perubahan satu *state* dan aktivitas

d) *Sequence diagram*

Menjelaskan interaksi objek yang disusun dalam suatu urutan waktu. *Sequence diagram* memperlihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu didalam *use case.*

e) *Collaboration diagram*

Melihat pada interaksi dan hubungan terstruktur antar objek. Tipe diagram ini menekankan pada hubungan *(relationship)* antar objek. *Collaboration diagram* digunakan sebagai alat untuk menggambarkan interaksi yang mengungkapkan keputusan mengenai perilaku sistem.

f) *Activity diagram*

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktivitas lainnya seperti *use case* atau interaksi.

g) *Component diagram*

Menggambarkan alokasi semua kelas dan objek kedalam komponen - komponen dalam desain fisik sistem *software.* Diagram ini memperlihatkan pengaturan dan kebergantungan antara komponen-komponen *software* seperti *source code, binary code,* dan komponen tereksekusi *(executable component)*.

h) *Deployment diagram*

Menggambarkan arsitektur fisik dari perangkat keras dan perangkat lunak sistem, menunjukkan hubungan komputer dengan perangkat *(nodes)* satu sama lain dan jenis hubungannya. Diagram ini memperlihatkan pemetaan *software* kepada *hardware.*

2.7 Basis Data

Basis Data terdiri dari dua kata, yaitu basis dan data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang atau berkumpul. Sedangkan Data adalah representasi fakta dunia nyata mewakili suatu objek seperti manusia, barang, hewan, peristiwa dan sebagainya.

Basis data merupakan kumpulan dari data - data yang saling terkait dan saling berhubungan satu dengan lainnya. Basis data adalah kumpulan kumpulan file yang saling berkaitan.

Salah satu pemakai sistem informasi manajemen *database* yaitu administrator basis data *(Database Administrator/DBA)* fungsi ini bertanggung jawab untuk melakukan koordinasi, pengawasan, dan pengelolaan *database.*

Basis data memiliki tujuan yang diantaranya adalah sebagai berikut :

1. Kemampuan dalam menangani data dalam jumlah besar.
2. Efisiensi pengolahan data yang meliputi *speed, space* dan *accuracy.*
3. Kemampuan pemakaian *resource* basis data secara bersama - sama *(Sharebility).*
4. Meniadakan duplikasi data yang tidak perlu dan menghindari inkonsistensi data.

2.8 Arsitektur Aplikasi

Arsitektur aplikasi dapat pula dikatakan sebagai struktur desain dari suatu sistem komputer dengan semua kelengkapanya yang telah siap untuk digunakan oleh *user.*

2.8.1 Pengertian Jaringan Komputer

Jaringan komputer adalah sekumpulan komputer pribadi saling berhubungan antara satu dengan yang lainnya menggunakan protokol komunikasi sehingga dapat saling berbagi informasi, program penggunaan bersama perangkat keras seperti *printer, harddisk.* Selain itu jaringan komputer dapat diartikan sebagai kumpulan sejumlah terminal komunikasi yang berada di berbagai lokasi yang terdiri dari lebih dari satu komputer yang saling berhubungan. Dalam sebuah jaringan komputer dapat diartikan sebagai kumpulan sejumlah terminal komunikasi yang berada di berbagai lokasi yang terdiri dari satu komputer yang saling berhubungan. Dalam sebuah

jaringan komputer biasanya terhubung sejumlah komputer ke sebuah/beberapa *server.*

Server adalah komputer yang difungsikan sebagai "pelayanan" pengiriman data dan penerimaan data diantara komputer - komputer yang tersambung.

Tujuan dibangunnya suatu jaringan komputer adalah membawa informasi secara tepat dan tanpa adanya kesalahan dari sisi pengirim (transmisi) menuju ke sisi penerima *(receiver)* melalui media komunikasi.

TCP/IP (*Trasmission Control Protocol/ Internet Protocol)* merupakan protokol *standard internet* yang digunakan untuk melakukan koneksi ke *internet* protokol. TCP/IP memiliki beberapa *subyek protocol* yang berbeda yang beroperasi pada lapisan yang berbeda dan mempunyai tugas masing - masing. Berkat adanya protokol ini setiap komputer dapat berhubungan secara fleksibel dengan *host - host* yang terkoneksi.

2.8.2 Jenis - jenis Jaringan Komputer

Jenis - jenis jaringan ada dua yaitu :

1. *Model peer to peer*

   Menurut model ini, setiap *host* dapat menawarkan layanan ke *peer* lain dan juga mengambil layanan dari *peer* lain. Model ini cocok untuk jaringan kecil.

2. *Model Client/Server*

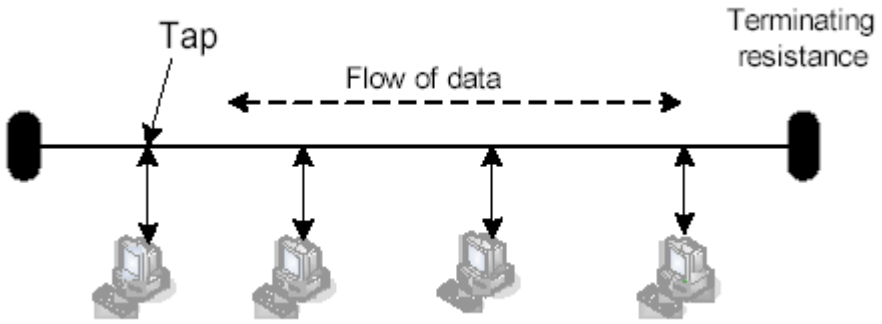
Model ini memisahkan secara jelas, mana yang dapat memberikan layanan jaringan *(server)* dan mana yang hanya memberikan layanan *(client)*. Beberapa komputer di*set-up* sebagai *server* yang memberikan sumber daya *(resource)* dari jaringan : *printer, modem,* dan saluran lainnya kepada komputer lain yang dikoneksi kejaringan yang berfungsi sebagai *client.*

2.8.3 Topologi Jaringan Komputer

Yang dimaksud topologi jaringan itu adalah susunan fisik bagaimana nodenode saling dihubungkan. Ada tiga topologi jaringan komputer, yaitu:

1. *Topologi Bus*

   Ethernet adalah contoh yang menggunakan *topologoi bus* yang menggunakan satu kawat (kabel) yang berfungsi sebagai medium untuk mentranmisikan data. *Node* yang merupakan bagian dari jaringan dihubungkan seluruhnya kekabel tersebut. *Node-node* yang terhubungkan mengirim dan menerima data jaringan melalui kabel sebagai pembawa sinyal dan melihat apakah data tersebut ditujukan buat dirinya.

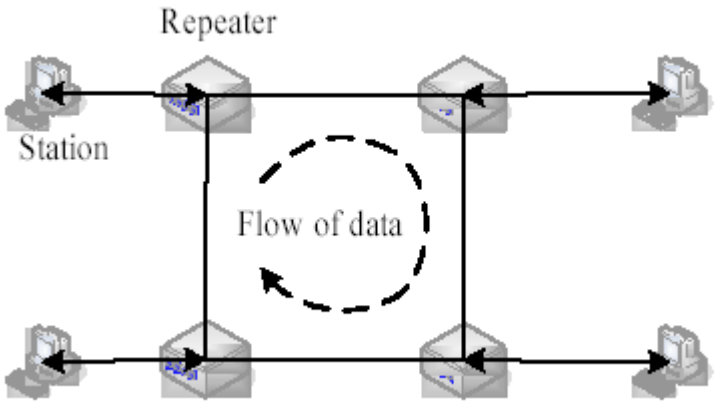


Gambar 2.2 *Topologi Bus*

Sumber : Stendy B.Sukur, 2003, *Perkembangan Hardware Komputer*.

2. *Topologi Ring*

Salah satu *node* dihubungkan dengan *node* yang ada didepan dan dibelakangnya sehingga membentuk *ring*. Setiap *node* mendapatkan giliran menggunakan jaringan dengan mengirim *"token"*. *Node* yang mendapat gilirandapat mengirim data dan *node* lain menerima data, serta melihat apakah data ditujukan kepadanya. Bila ditujukan buat dirinya, data pun disimpan, bila tidak data diteruskan ke *node* didepannya.



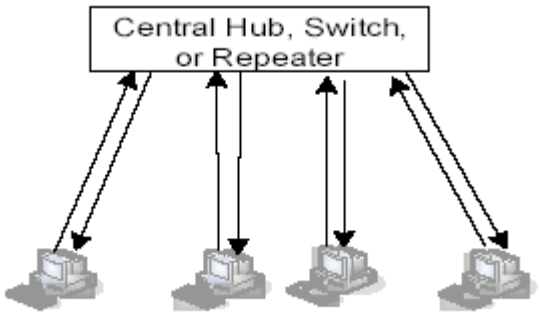
Gambar 2.3 *Topologi Ring*

Sumber : Stendy B.Sukur, 2003, *Perkembangan Hardware Komputer*.

3. *Topologi Star* atau Hub

Susunan atau skema dari topologi ini mirip sebuah bintang. Topologi ini memiliki satu hub pusat dari mana data ditransmisikan ke seluruh *node* dalam jaringan. Skema ini mempunyai kelebihan dibandingkan dua skema sebelumnya, yaitu bila terjadi kerusakan pada kabel tidak membawa dampak bagi seluruh *node,* tapi hanya node yang bersangkutan saja sehingga aktivitas jaringan tidak terganggu secara total. Ini berbeda dengan skema bus atau ring, dimana bila terjadi kerusakan pada kabel berakibat pada seluruh jaringan.

Topologi yang digunakan dalam merancang sistem ini adalah topologi *star* atau hub karena topologi ini mempunyai banyak kelebihan dibandingkan dengan topologi yang lain.



Gambar 2.4 *Topologi Star atau Hub*

Sumber : Stendy B.Sukur, 2003, *Perkembangan Hardware Komputer*.

2.8.4 Manfaat Jaringan Komputer

Manfaat yang didapat dari membangun jaringan komputer adalah sebagai berikut :

1. *Sharing Resources*

   Sharing resources bertujuan agar seluruh program, peralatan/ peripheral lainnya dapat dimanfaatkan oleh setiap orang yang ada pada jaringan komputer tanpa terpengaruh pada jaringan komputer tanpa terpengaruh oleh lokasi maupun pengaruh dari pemakai. Dengan kata lain, seorang pemakai yang letaknya jauh sekalipun dapat memanfaatkan data maupun informasi yang lainnya tanpa mengalami kesulitan.

2. Media komunikasi

Jaringan komputer memungkinkan terjadinya komunikasi antara pengguna, baik untuk *teleconference* maupun untuk mengirim pesan/ informasi lainnya.

3. Integrasi data

   Pembangunan jaringan komputer dapat mencegah ketergantungan pada komputer pusat. Setiap proses data tidak harus dilakukan 1 komputer saja, melainkan dapat didistribusikan ke tempat lainnya oleh sebab inilah maka dapat terbentuk data yang terintegrasi sehingga dengan demikian memudahkan pemakai untuk memperoleh dan mengolah informasi setiap saat.

4. Pengembangan dan Pemeliharaan

   Dengan adanya jaringan komputer ini, maka perkembangan peralatan dapat dilakukan dengan mudah dan menghemat biaya. Jaringan komputer bisa memudahkan pemakai dalam merawat *harddisk* dan peralatan lainnya. Misalnya untuk memberikan perlindungan serangan *virus* maka pemakai cukup memusatkan perhatian pada *harddisk* yang ada di komputer pusat.

5. Keamanan Data

   Sistem jaringan komputer memberikan perlindungan terhadap data jaminan keamanan data tersebut di berikan melalui pengaturan hak akses para pemakai dan *password,* serta teknik perlindungan terhadap *harddisk* sehingga data mendapatkan perlindungan yang efektif.

6. Sumber Daya Lebih Efisien dan Informasi Terkini

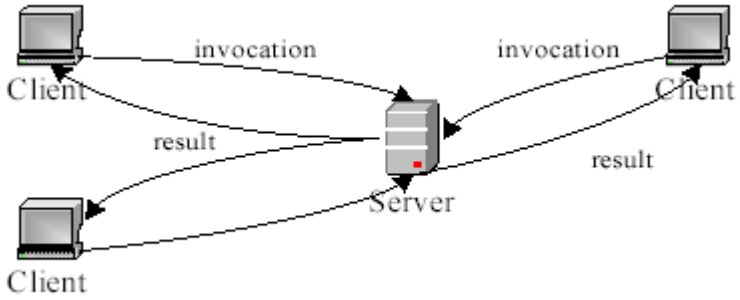
   Dengan adanya pemakaian sumber daya secara bersama-sama maka pemakai bisa mendapatkan hasil dengan maksimal dan kualitas yang tinggi. Selain itu data atau informasi yang di akses selalu terbaru, karena setiap ada perubahan yang terjadi dapat secara langsung di ketahui oleh setiap pemakai.

2.9 Pengertian Client / server

Sistem *client* - *server* mempunyai dua komponen utama yaitu komputer *client* dan komputer *server*. *Server* merupakan komputer induk yang melakukan pemrosesan terbanyak untuk memenuhi permintaan- permintaan dari komputer *client* dan bertindak sebagai *server database* yang menyimpan data. *Client* yaitu komputer atau *workstation* yang melakukan pengiriman permintaan- permintaan data pada *server* kemudian menampilkan data tersebut pada *interface* aplikasi yang dimilikinya. Selain itu *client* juga mempunyai kemampuan untuk mengubah atau menghapus data itu.

Sistem *client* - *server* merupakan suatu sistem *client* komputer yang melibatkan proses - proses *client* yang meminta suatu pelayanan data kepada computer *server* yang menyediakan layanan data tersebut. Sehingga *client* maupun *server* sama - sama melakukan pekerjaan. Dengan adanya kombinasi *client (Front-end)* dan *server (back-end)* ini maka kumpulan dari modul - modul program tidak dieksekusi dalam memori yang sama namun terbagi dalam komputer *client - server*.

Hal ini menjadikan konfigurasi bagi komputer *client* dan komputer *server* bisa berbeda seperti kapasitas memori, kecepatan prosesor atau alat masukan dan keluaran yang disesuaikan dengan fungsi kerja dari elemen - elemen tersebut. Bagi *server* yang menjalankan tugas pengelolaan suatu *database* digunakan suatu konfigurasi yang khusus menangani tugasnya tersebut dengan sistem operasi yang dikhususkan bagi *server* seperti windows NT *server, windows 2000 server,* sedangkan komputer *client* menggunakan konfigurasi yang umum bagi sebuah komputer desktop yang terhubung ke jaringan dengan sistem operasi seperti *windows 98, windows ME, windows Xp* dan lain - lain.



Gambar 2.5 *Model Client-server*

Sumber : Susanto Budi, 2003, *Pengantar Sistem Terdistribusi.*

2.10 Perangkat Lunak Pendukung

Perangkat lunak *(software)* yang digunakan penyusun dalam membuat sistem informasi akademik di SMAN Tanjungsari ini adalah : *Borland Delphi 7, Microsoft Access Data Base.*

2.10.1 Borland Delphi 7

Borland Delphi atau yang selanjutnya disingkat dengan *delphi* adalah bahasa pemrograman yang bekerja dengan sistem operasi sistem *windows. Delphi* merupakan bahasa pemrograman yang mempunyai kemampuan luas. Berbagai jenis aplikasi dapat dibuat dengan *delphi,* termasuk aplikasi untuk mengolah *teks, grafik,* angka, *database* dan aplikasi *web.*

Secara umum, kemampuan *Delphi* adalah menyediakan komponen - komponen dan bahasa pemrograman yang andal, sehingga memungkinkan untuk mebuat aplikasi sesuai dengan yang diharapkan, baik dari segi tampilan maupun dari segi kemampuan.

Untuk mempermudah pembuatan program aplikasi, *delphi* menyediakan fasilitas pemrograman yang lengkap. Fasilitas pemrograman tersebut dibagi dalam kedua kelompok, yaitu pemrograman berorientasi *object* (bentuk visual) dan bahasa pemrograman *procedure.* Secara ringkas, *object* adalah komponen yang mempunyai bentuk fisik dan biasanya dapat dilihat *(visual).* Sedangkan bahasa pemrograman secara singkat dapat disebut sebagai sekumpulan *teks* yang mempunyai arti tertentu dan disusun dengan aturan tertentu untuk menjalankan programnya. *Delphi* menggunakan struktur bahasa pemrograman *object pascal* yang sangat dikenal dikalangan pemrogram.

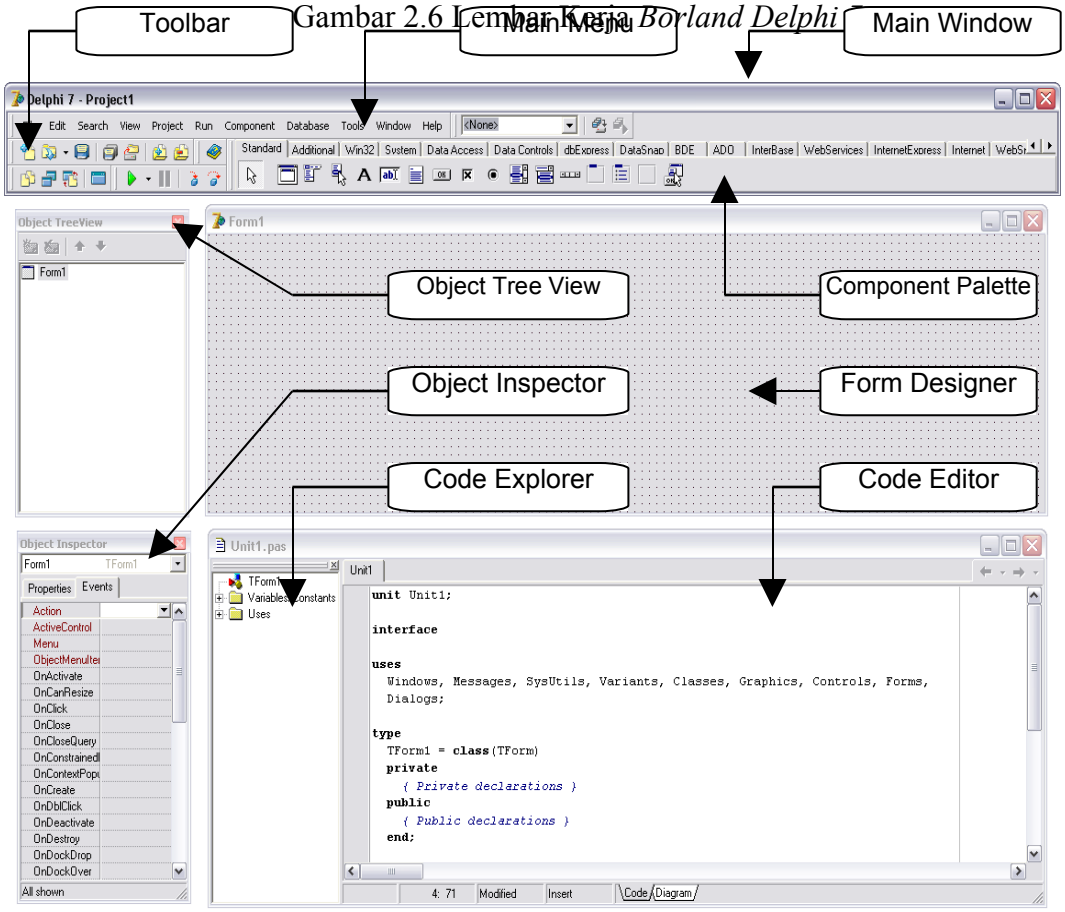
Delphi sebagai salah satu bahasa pemrograman dinilai memiliki kemampuan yang baik dengan kelebihan - kelebihan sebagai berikut :

1. Dalam pembuatan perangkat lunak, *delphi* menggunakan sebuah sistem RAD (*Rapid Application Development*) yang memanfaatkan bahasa pemrograman visual. RAD (*Rapid Application Development*) adalah sekumpulan strategi, motodologi dan perangkat terpadu yang ada dalam kerangka kerja yang disebut rekayasa informasi (*IE / Information Engineering*).
2. *Delphi* memberikan kemudahan dalam pembuatan perangkat lunak, dimana komponennya dapat dipakai ulang dan dapat dikembangkan.
3. Dalam Delphi telah didefinisikan model aplikasi dan model form yang dapat dipakai untuk membuat semua aplikasi dengan cepat.

2.10.2 Komponen Borland Delphi 7

Beberapa komponen yang dipakai dalam membuat program aplikasi adalah :

Saat pertama kali Borland Delphi 7. dipanggil maka akan muncul lima windows yaitu:



Gambar 2.6 Lembar Kerja *Borland Delphi 7*

1. *Main Window*

Terdapat tiga buah elemen yang masing-masing memiliki fungsi khusus yaitu *menu bar, speed bar* dan *component pallete*

a. *Main Menu*

Untuk mengatur semua *window* yang ada di lingkungan kerja *Delphi,* juga untuk mengelola proses desain aplikasi.



Gambar 2.7 *Menu Bar*

b. *Tool Bar*

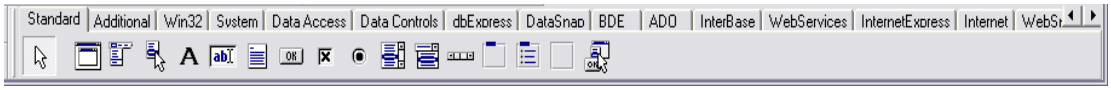
Menyediakan akses yang cepat bagi operasi - operasi yang sering digunakan, seperti membuka *file,* menyimpan *file,* fasilitas *cut* and *paste,* memilih *window* di dalam *Delphi,* eksekusi program aplikasi, serta membuka dan memilih *form* dalam *unit.*



Gambar 2.8 *Speed Bar*

c. *Component Pallete*

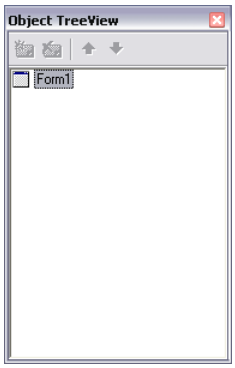
Component pallete adalah sebuah catalog yang berisi objek - objek *visual* dan objek aplikasi lainnya yang dapat ditempatkan pada *form.* Digunakan untuk memilih objek yang digunakan dalam pembuatan aplikasi dan kemudian meletakkan objek tersebut pada rancangan *form.*



Gambar 2.9 *Component Pallete*

2. *Object Tree View Window*

Object tree view menampilkan diagram pohon dari komponen-komponen yang bersifat *visual* maupun *nonvisual* yang telah terdapat dalam *form,* data *module* atau *frame. Object tree view* juga menampilkan hubungan logika antarkomponen.



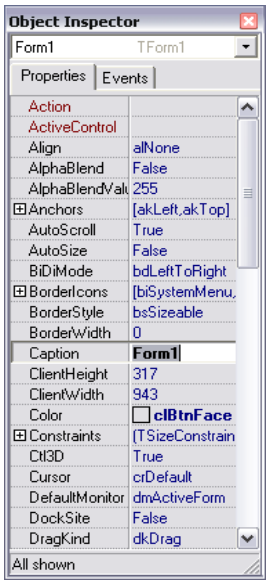
Gambar 2.10 *Object Tree View*

3. Object Inspector Window

Terletak di bagian kiri bawah. Terdiri dari dua halaman yaitu halaman *properties* dan halaman *events*. Halaman *properties* menampilkan semua *properties* untuk objek yang terpilih dalam *windows form*. Halaman *events* mendaftar semua *events* yang akan ditanggapi oleh objek yang bersangkutan.

a. Halaman *Properties*

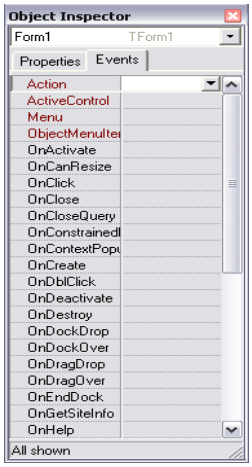
Halaman *properties* digunakan untuk menentukan setting suatu objek. Suatu objek biasanya mempunyai beberapa *properties* yang dapat diatur langsung dari lembar *properties* pada jendela *Object Inspector* maupun diatur lewat kode program. *Setting properties* akan menentukan cara kerja dari objek yang bersangkutan saat program aplikasi dijalankan.



Gambar 2.11 *Object Inspector Properties*

b. Halaman *Event*

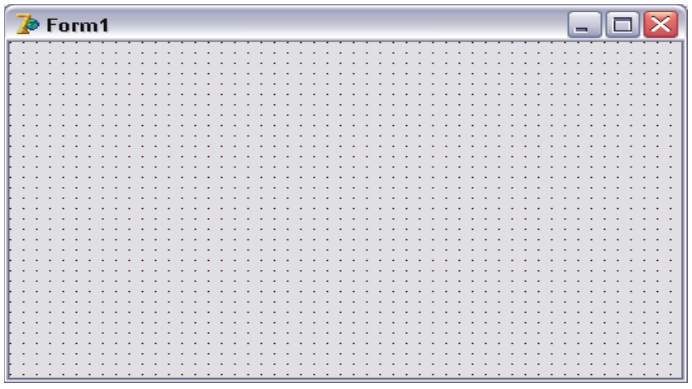
Event adalah peristiwa yang diterima oleh suatu objek, misalnya *klik, drag,* tunjuk dan lain-lain. *Event* yang diterima objek akan membuat *Borland Delphi* 7 menjalankan kode program yang ada di dalamnya.



Gambar 2.12 *Object Inspector Events*

4. *Form Designer Window*

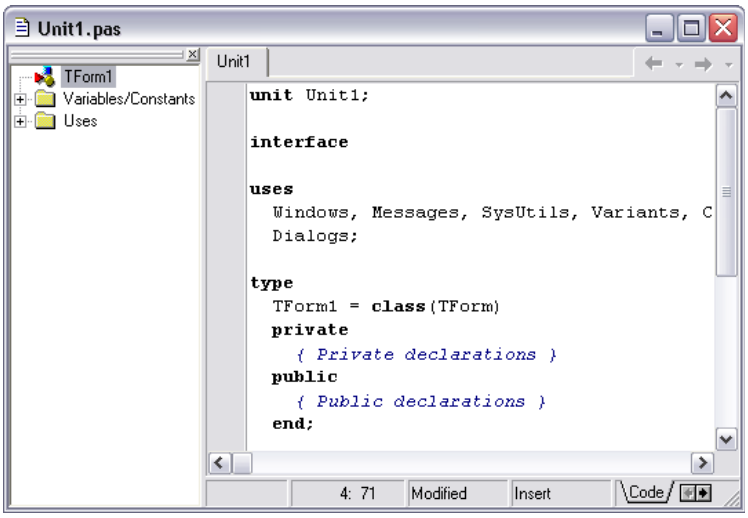
Form adalah suatu objek yang dipakai sebagai tempat bekerja program aplikasi. *Form* berbentuk jendela dan dapat dibayangkan sebagai kertas atau meja kerja yang dapat dilukisi atau diletakkan ke objek - objek lain dalamnya. Pada saat membuat suatu program aplikasi baru, biasanya ditampilkan dengan judul *Form 1* dan terletak di bagian kanan atas *windows editor program.*



Gambar 2.13 *Windows Form*

5. *Code Editor Window*

Dipakai untuk mengedit program objek *Pascal.* Memiliki beberapa halaman yang masing-masing menyimpan sebuah unit program, yang digunakan untuk menuliskan perintah yang harus dilakukan oleh komponen - komponen yang ada pada *form.* Pada *Code Editor window* juga terdapat *Code Explorer* yang digunakan untuk memudahkan pemakai berpindah antar *file unit* yang terdapat di dalam *code editor.*



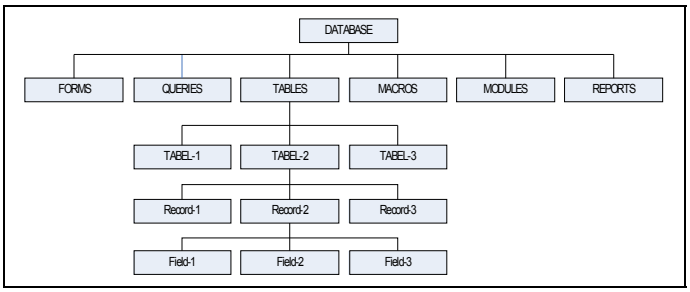
Gambar 2.14 *Windows Editor Program*

2.10.3 Microsoft Access (MS-Access)

Microsoft Access (MS-Access) adalah *software* RDBMS (*Relational Datbase Management System*) berbasis *windows,* yang terakhir dikeluarkan oleh *Microsoft* sebagai bagian dari paket *Micrososft Office.*

2.10.4 Struktur Database Microsoft Access (MS-Access)

MS-Access sebagai pengolahan *database* mempunyai komponen-komponen yang memebentuk struktur *database* secara terpadu (*integrated*) antara data dengan unsur pengolahannya.



Gambar 2.15 Struktur *Database* Pada *MS-Access*

Sumber: [Tim Pengajar Laboratorium Komputer, 2002, *Diktat Kompuetr II,* Universitas Padjajaran, Bandung]

Berdasarkan struktur *database* tersebut tampak bahwa secara garis besarnya suatu *database* pada *MS-Access* terdiri dari 6 komponen utama sebagai onjek yang mempunyai fungsi masing - masing, yaitu:

1. Tabel (*Table*), berupa kumpulan tabel data yang merupakan komponen utama dari suatu *database*. Serta sebagai tempat menyimpan data yang diolah dalam format baris dan kolom.
2. *Query* (*Queries*), merupakan objek *database* yang berfungsi untuk mengolah data yang tesimpan pada tabel untuk menghasilkan informasi yang *dibutuhkan.*
3. *Form,* atau formulir elektronik berfunsi untuk memasukan (*Input*) data terhadap suatu tabel. Serta digunakan untuk menampilkan, mengisi, dan mengubah data yang ada didalam tabel.
4. *Reports*, digunakan untuk *output* (menampilkan atau mencetak) informasi, baik berasal dari tabel ataupun dari *query*.

5. *Macro* (*Macros*), merupakan program sederhana yang dapat dibuat untuk mengatur urutan-urutan perintah atau fungsi yang tersedia dalam *MS-Access* dalam pengolahan suatu *database*.
6. *Module* (*Modules*), merupakan urutan-urutan perintah pada *MS-Access* untuk mengolah data dengan menggunakan bahasa pemrograman tertentu.

2.10.5 Rational Rose

Rational Rose adalah *software* yang memiliki perangkat - perangkat pemodelan secara visual untuk membangun suatu solusi dalam rekayasa *software* dan pemodelan bisnis. *Rational Rose* dikeluarkan oleh perusahaan *software* bernama *Rational Software,* perusahaan yang mencetuskan ide pembentukan konsorsium bagi perusahaan - perusahaan yang memakai standar UML sebagai bahasa pemodelan di perusahaannya, *Rational Rose* memakai UML sebagai bahasa pemodelannya, ditambah beberapa fitur lain yang membuat *Rational Rose* menjadi *software* pemodelan *visual* yang terkemuka. Bebarapa *fitur* yang terkemuka diantaranya *Rational Rose* memiliki *Rational Unified Process* (RUP) selain itu, *Rational Rose* memiliki kemampuan membuat solusi *client/server,* yang kemudian diterapkan dan didistribusikan dalam lingkungan perusahaan. *Rational Rose* memiliki keunggulan, diantaranya :

1. Bahasa yang digunakan adalah bahasa pemodelan standar yaitu UML, akan meningkatkan komunikasi intra tim.

2. *Rational Rose* mendukung *round-trip engineering,* sehingga kita dapat men-*generate* model kedalam kode (*java, C++, Visual Basic, Delphi* dan sebagainya) dan melakukan *reverse engineering* untuk menampilkan arsitektur *software* dari kode yang ada. Hal ini dapat dilakukan secara bolak - balik sebagai proses *interative* selama proses rekayasa *software.*
3. Model dan kode senantiasa singkron selama dalam *development cycle.*
4. Membangun *software* menggunakan *Rational Rose* memudahkan dalam memperbaiki *software* tersebut karena apabila suatu saat ditemukan requirement baru, dapat digambarkan kembali *software* tersebut dalam UML.
5. Para *user Rational Rose* dapat berkomunikasi walaupun bekerja dalam system operasi yang berbeda *(Windows* atau *UNIX).*
6. Dengan menggunakan *Rose Web Publisher* suatu tim dapat mengkomunikasikan model dan spesifikasinya dalam *web browser.*
7. Mendukung rekayasa *software* untuk sistem *client/server* sehingga *Rational Rose* merupakan *software* pemodelan *visual* yang tangguh dalam lingkungan *client/server, e-business,* dan lingkungan perusahaan terdistribusi (kantor - kantornya terletak dalam tempat yang terpisah - pisah).

2.11 Kasus yang dianalisis / dirancang

Dalam proses kegiatan belajar mengajar diperlukan sebuah sistem pengolahan nilai terkomputerisasi dikarenakan sistem yang sebelumnya masih menggunakan sistem manual dan membutuhkan sistem yang lebih efektif dan efisien.

1. Sistem Informasi Akademik adalah suatu yang memberikan layanan informasi yang berupa data, dalam hal ini yang berhubungan dengan akademik.
1. Nilai adalah angka atau sesuatu yang dihasilkan dari evaluasi proses kegiatan yang menunjukan kemampuan sejauh mana proses kegiatan dikuasai dan dimengerti .