BAB II

LANDASAN TEORI

2.1 Pengertian Sistem

Menurut JOG [3] sistem adalah suatu kesatuan atau kumpulan yang terdiri dari dua atau lebih komponen atau subsistem yang saling berinteraksi untuk mencapai suatu tujuan tertentu.

Pada dasarnya setiap kegiatan sistem dalam suatu organisasi mempunyai tujuan yang hendak dicapai. Suatu kegiatan dapat berjalan karena ada sistem. Ada dua kelompok pendekatan dalam mendefinisikan sistem, yaitu yang menekankan pada prosedur dan komponen atau elemen. Pendekatan sistem yang menekankan pada prosedur, sistem yaitu suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau menyelesaikan suatu sasaran tertentu (JOG [3]).

Sedangkan pendekatan sistem yang lebih menekankan pada elemen dan komponen sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai tujuan tertentu (JOG [3]).

Kedua pendekatan tersebut benar, yang berbeda adalah cara pendekatannya kepada sistem. Pendekatan sistem yang menekankan pada elemen dan komponen yang merupakan definisi yang lebih banyak diterima karena dapat memudahkan dalam menganalisa dan mengembangkan suatu sistem sehingga tujuan atau sasaran yang telah ditentukan dapat tercapai dengan baik.

Karakteristik atau sifat-sifat sebuah sistem adalah sebagai berikut:

a. Komponen (C*omponent*)

   Suatu sistem yang terdiri dari sejumlah komponen yang saling terinteraksi. Bekerja sama membentuk satu kesatuan yang utuh.

b. Batasan Sistem (*Boundary*)

   Batasan sistem merupakan daerah yang membatasi antara suatu sistem yang satu dengan sistem yang lain atau dengan lingkungan luar sistem.

c. Lingkungan Luar Sistem (*Environment*)

   Lingkungan luar sistem merupakan batas atau faktor-faktor luar dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan merugikan sistem.

d. Penghubung (*Interface*)

   Penghubung merupakan suatu media penghubung antara suatu subsistem dengan suatu subsistem yang lainnya.

e. Masukkan (*Input*)

   Masukan merupakan energi yang dimasukkan kedalam sebuah sistem. Masukan dapat berupa masukan perawatan dan masukan sinyal.

f. Keluaran (*Output*)

   Keluaran merupakan hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna bagi pemakai dan sisa pembuangan. Keluaran dapat merupakan masukkan bagi subsistem lainnya.

g. Pengolah (*Process*)

Suatu sistem mempunyai suatu bagian pengolah dari data atau masukan yang akan merubah dan menghasilkan suatu keluaran berupa informasi yang berguna.

h. Sasaran atau Tujuan (*Objective*/*Goal*)

Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang dihasilkan oleh sistem untuk mencapai sasaran atau tujuannya.

2.2 Konsep Dasar Informasi

Informasi merupakan faktor yang sangat penting dalam suatu sistem untuk pengambilan suatu keputusan.

Informasi menurut JOG [3] adalah **hasil dari pengolahan dan pemrosesan data menjadi bentuk yang lebih berguna, bermanfaat dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian (*event*) yang nyata (*fact*) yang digunakan untuk pengambilan keputusan.**

2.2.1 Siklus Informasi

Data merupakan sumber dari informasi yang sangat berguna bagi pemakainya. Bentuk tunggal dari data adalah datum atau data item. Data dapat berbentuk simbol-simbol berupa huruf-huruf, angka-angka, bentuk-bentuk suara, sinyal-sinyal, dan gambar-gambar.

Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan yang nyata JOG [3]. Kejadian-kejadian (*event*) merupakan suatu yang terjadi pada saat tertentu JOG [3]. Data merupakan bentuk yang masih mentah sehingga perlu diolah melalui suatu proses atau model untuk menjadi suatu informasi.

Data akan ditangkap sebagai *input*, kemudian diproses lewat suatu model dan seterusnya membentuk suatu siklus informasi.

2.2.2 Kualitas Informasi

Kualitas informasi tergantung dari tiga hal, yaitu:

a. Akurat berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bias atau menyesatkan. Akurat juga berarti informasi harus menjelaskan dan mencerminkan maksud dari informasi tersebut.
b. Tepat waktu berarti informasi yang datang kepada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi karena informasi merupakan landasan di dalam pengambilan keputusan.
c. Relevan berarti informasi mempunyai manfaat bagi penerima. Relevansi informasi untuk satu orang dengan orang lain berbeda. Hal ini sangat tergantung kepada penerima dan yang membutuhkan informasi.

Kualitas informasi tidak tergantung dari berapa biaya untuk mendapatkannya tetapi tergantung dari nilai informasi tersebut dan manfaatnya.

2.2.3 Nilai Informasi

Nilai informasi ditentukan dari dua hal, yaitu manfaat dan biaya untuk mendapatkan informasi tersebut. Suatu informasi dikatakan bernilai apabila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya. Kegunaan informasi adalah untuk mengurangi hal ketidakpastian didalam proses pengambilan keputusan tentang suatu keadaan.

2.3 Pengertian Sistem Informasi

Sistem informasi menurut JOG [3] adalah

> **Sebagai suatu sistem didalam suatu organisasi yang merupakan kombinasi dari orang-orang, fasilitas, teknologi, media, prosedur-prosedur dan pengendalian yang ditujukan untuk mendapatkan jalur komunikasi penting, memproses tipe transaksi rutin, memberikan sinyal kepada manajemen terhadap kejadian-kejadian internal dan eksternal dan menyediakan suatu dasar informasi yang penting untuk pengambilan keputusan**.

Tujuan dari sistem informasi adalah untuk mengumpulkan data, menyimpan data, dan menghasilkan informasi untuk pemakai.

> **Sistem informasi dapat terdiri dari kompoen-komponen yang saling berhubungan, yaitu blok masukan (*input block*), blok model (*model block*), blok keluaran (*output block*), blok teknologi (*technology block*), blok dasar data (*database block*), dan blok kendali (*controls block*) JOG [3].**

Sebagai suatu sistem ke enam blok tersebut saling berinterkasi satu dengan yang lain membentuk kesatuan untuk mencapai sasarannya. Dalam suatu sistem keenam blok tersebut tidak dapat terpisahkan dan saling berhubungan satu sama lain agar sistem dapat berjalan dengan lancar dan sesuai dengan prosedur yang berlaku.

2.4 Metode Pendekatan Sistem

Pemrograman berorientasi objek pertam-tama dibicarakan pada akhir tahun 1960 menggunakan bahasa SIMULA. Pada tahun 1970, bahasa pemrograman Smaltalk dikembangkan oleh Xerox PARC. Pada saat itu sebagian lain dari dunia menggunakan COBOL dan FORTRAN dengan metode dekomposisi fungsional. Perubahan terjadi selama beberapa dekade sampai dikenalnya faktor utama, yaitu

konsep dasar berorientasi objek, teknologi dasar pembangunan sistem, kondisi sistem, dan penggunaan model.

Metode pendekatan sistem Berorientasi Objek (*Object oriented*) menurut HAD [2] adalah mengorganisasi perangkat lunak sebagai kumpulan dari objek tertentu yang memiliki struktur data dan perilakunya.

Metodologi pengembangan sistem berorientasi objek mempunyai tiga karakteristik utama, yaitu:

a. Encapsulation

Encapsulation (pengkapsulan) merupakan dasar untuk pembatasan ruang lingkup program terhadap data yang diproses.

b. Inheritance

Inheritance (pewarisan) adalah teknik yang menyatakan bahwa anak dari objek akan mewarisi data/atribut dan metoda dari induknya langsung.

c. Polymorphism

*Polymorphism (*polimorfisme) yaitu konsep yang menyatakan bahwa sesuatu yang sama dapat mempunyai bentuk dan perilaku berbeda.

2.4.1 Model Berorientasi Objek

Sebuah model objek menangkap struktur statis dari sistem dengan menggambarkan objek dalam sistem, hubungan antara objek, serta atribut dan operasi yang merupakan karakteristik 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 daripada 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 dan semantik umum. Objek dan kelas-objek sering sama sebagai benda dalam deskripsi masalah.

2. Diagram Objek

   a. Kelas dan Objek

   Kelas-&-Objek adalah suatu kelas dan objek-objek yang berada pada kelas tersebut. Nama kelas adalah kata benda tunggal, atau kata sifat dan kata benda. Nama dari kelas-&-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 *service* atau *operator*. Metoda adalah subprogram yag tergabung dalam objek bersama-sama dengan atribut.

e. Message

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

2.4.2 Analisis Berorientasi Objek (OOA)

Analisis sistem menurut HAD [2] adalah proses menentukan kebutuhan sistem, apa yang harus dilakukan sistem untuk memenuhi kebutuhan klien, bukanlah bagaimana sistem tersebut diimplementasikan.

Melihat dari definisi diatas, maka seorang analis sistem akan terfokus pada dunia user, domain aplikasi atau permasalahan serta tanggung jawab dari sistem.

Dalam analisis berorientasi objek, terdapat lima lapisan (tidak bersifat sequensial), yaitu :

1. Menentukan lapisan Subjek
2. Menentukan lapisan Kelas dan Objek
3. Menentukan lapisan struktur

4. Menentukan lapisan *attribute*
5. Menentukan lapisan *service*

2.4.3 Desain Berorientasi Objek (OOD)

Desain adalah kegiatan untuk mengambil atau menentukan spesifikasi dari perilaku luar yang dapat diamati dan menambah detail yang diperlukan untuk implementasi sistem komputer (secara nyata), termasuk interaksi manusia, pengelolaan tugas/aksi, pengelolaan data secara detail.

Melihat definisi diatas, maka seorang desain sistem kegiatannya terfokus dengan tugas mengubah hasil analisis (suatu spesifikasi) ke dalam implementasi perangkat keras atau perangkat lunak secara umum.

Dalam desain berorientasi objek, terdapat lima lapisan kegiatan (tidak bersifat sequensial), yaitu :

1. Menentukan lapisan Subjek
2. Menentukan lapisan Kelas dan Objek
3. Menentukan lapisan struktur
4. Menentukan lapisan *attribute*
5. Menentukan lapisan *service*

2.5 Pengenalan UML

UML gabungan dari metode Booch, Rumbaugh (OMT) dan Jacobson, Tetapi UML ini akan mencakup lebih luas daripada OOA&D. Pada pertengahan pengembangan UML dilakukan standarisasi proses dengan OMG (*Object*

Management Gruop) dengan harapan UML akan menjadi bahasa standar pemodelan pada masa yang akan datang.

2.5.1 Pengertian UML

Menurut A.SUH [1] UML (*Unified Modeling Language*) merupakan sistem arsitektur yang bekerja dalam OOAD dengan satu bahasa yang konsisten untuk menentukan, visualisasi, mengkonstruksi, dan mendokumentasikan *artifact* yang terdapat dalam sistem *software*.

UML merupakan pengganti dari metode analisis berorientasi objek dan desain berorientasi objek (OOA&D) yang dimunculkan sekitar akhir tahun 80-an dan awal tahun 90-an. 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.

Tujuan penggunaan UML :

1. Memodelkan suatu sistem (bukan hanya perangkat lunak) yang menggunakan konsep berorientasi objek.
2. Menciptakan suatu bahasa pemodelan yang dapat digunakan baik oleh manusia maupun mesin.

2.5.2 Bagian-bagian UML

Bagian-bagian utama dari UML adalah sebagai berikut :

1. View

View digunakan untuk melihat sistem yang dimodelkan dari beberapa aspek yang berbeda. *View* bukan melihat grafik, tapi merupakan suatu abstraksi yang berisi sejumlah diagram. Beberapa jenis *view* dalam UML antara lain:

- *Use Case View*

  Use case view membantu kita untuk memahami dan menggunakan sistem yang kita modelkan. *View* ini melihat pada bagaimana *actor* dan *use case* berinteraksi. Terdapat beberapa diagram yang digunakan dalam *use case view*, yaitu: *use case diagram, sequence diagram, collaboration diagram,* dan *activity diagram*. *View* ini digunakan terutama untuk pelanggan, perancang (*designer*), pengembang (*developer*), dan penguji sistem (*tester*).

- *Logical* View

  Logical view mengarah pada persyaratan (*requirements*) fungsionalitas sistem. *View* ini melihat pada kelas-kelas dan hubungan antarkelas-kelas tersebut. Diagram dalam *view* ini adalah: *class diagram, sequence diagram, collaboration diagram*, dan *statechart diagram* untuk model dinamisnya. *View* ini digunakan untuk perancang (*designer*) dan pengembang (*developer*).

- *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. Hanya ada satu jenis diagram yang digunakan pada *view* ini

yaitu *component diagram*. *View* ini digunakan untuk pengembang (*developer*).

- *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. *View* ini hanya memiliki satu diagram, yaitu *deployment view*. *View* ini digunakan untuk pengembang (*developer*), pengintegrasi (*integrator*) dan penguji (*tester*).

2. Diagram

Diagram merupakan penjelasan secara grafis menganai elemen-elemen dalam sistem. Tipe diagram yang berbeda-beda ini membantu kita untuk melihat sistem dari perspektif yang berbeda-beda.

- *Use Case Diagram*

  Menjelaskan manfaat sistem jika dilihat menurut pandangan orang yang berada diluar sistem atau keadaan lingkungan sistem yang dilihat *user* (*actor*) dan bukan bagaimana fungsi yang ada didalam sistem. Diagram ini menunjukan fungsionalitas suatu sistem atau kelas dan bagaimana sistem berinteraksi dengan dunia luar.

- *Class Diagram*

  Membantu kita dalam visualisasi struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. *Class diagram*

memperlihatkan hubungan antarkelas dan penjelasan detail tiap-tiap kelas didalam model desain (dalam *logical view*) dari suatu sistem.

- *Squence Diagram*

  Menjelaskan interaksi objek yang disusun dalam urutan waktu. Diagram ini secara khusus berasosiasi dengan *use case*. *Squence diagram* memperllihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu didalam *use case*. Tipe diagram ini sebaiknya digunakan diawal tahap desain atau analisis karena kesederhanaannya dan mudah untuk dimengerti.

- *Collaboration diagram*

  Melihat pada interaksi dan hubungan struktur antarobjek. Tipe diagram ini menekankan pada hubungan (*relationship*) antarobjek, sedangkan *sequence diagram* menekankan pada urutan kejadian.

- *Activity Diagram*

  Memodelkan alur kerja (*work flow*) sebuah proses bisnis dan urutan aktifitas dalam suatu proses. Menggambarkan rangkaiaan aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga dugunakan untuk aktivitas lainnya seperti *use case* atau interaksi.

- *Statechart Diagram*

  Merperlihatkan 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* atau aktivitas. *Satechart diagram* khususnya digunakan untuk memodelkan tarap-tarap diskrit dari sebuah siklus hidup objek, sedangkan *activity diagram* paling cocok digunakan untuk memodelkan urutan aktivitas dalam suatu proses.

- *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 components*)

- *Deployment Diagram*

  Setiap model hanya memiliki satu diagram *deployment*. Diagram ini memperlihatkan pemetaan *software* kepada *hardware*.

2.5.3 *Indikator Multiplicity*

Multiplicity menentukan banyaknya objek yang terlibat dalam relasi. *Multiplicity* menentukan banyaknya object yang terhubung satu dengan yang lainnya. *Indicator multiplicity* terdapat pada masing-masing akhir garis relasi, baik pada asosiasi maupun agregasi. Beberapa contoh *multiplicity* adalah :

1	Tepat satu
0..*	Nol atau lebih
1..*	Satu atau lebih

0..1	Nol atau satu
5..8	range 5 s.d 8
4..6,9	range 4 s.d. 6 dan 9

2.5.4 *Business Modeling*

Business modeling menurut A.SUH [1] dapat didefinisikan sebagai segala teknik pemodelan yang digunakan untuk menggambarkan sebuah proses bisnis. *Business modeling* dapat digunakan untuk meninjau, meningkatkan, dan membuat sebuah bisnis. *Business modeling* menggambarkan semua *workflow* yang terjadi dalam suatu organisasi. *Workflow* adalah urutan aktifitas yang menghasilkan suatu nilai atau objek yang dapat dilihat bagi suatu pihak atau entitas tertentu.

Proses Bisnis

Dalam suatu organisasi terdapat banyak proses bisnis. Proses bisnis adalah sekumpulan aktifitas yang dirancang untuk menghasilkan keluaran tertentu bagi *customer* tertentu.

Didalam *business modeling* terdapat dua jenis model yaitu: model use case bisnis (*business use case model)* dan model objek bisnis (*business object model)* yang digunakan untuk memodelkan proses-proses bisnis yang terdapat dalam organisasi.

1. Model use case bisnis (*business use case model)* adalah model yang menggambarkan proses-proses bisnis dari sebuah bisnis atau organisasi dan interaksi proses tersebut dengan pihak luar, seperti para *customer* dan *partner*. Model use case bisnis diilustrasikan dalam beberapa *use case diagram*.

Model use case bisnis mempunyai elemen dengan *streotype* bernama *business actor* dan *business use case*.

2. Model objek bisnis (*business object model*) adalah model objek yang menggambarkan realisasi dari *business use case* (*business use case realization*). Model objek bisnis bertugas mengenali semua orang yang bekerja dan benda yang terlibat dalam bisnis dan bagaimana satu sama lain berhubungan.

   Model objek bisnis memiliki elemen dengan *streotype* yaitu: *business use case realization, business worker* dan *business entity*.

2.5.5 *Requirement*

Requirement menurut A.SUH [1] **adalah kondisi atau kemampuan yang harus dipenuhi oleh *software* aplikasi yang akan dibuat. *Requirement* untuk *software* aplikasi dibagi menjadi dua kelompok, yaitu *functional requirement* dan *non-functional requirement.***

Functional requirement menentukan tindakan yang harus dapat dimainkan sebuah *software* aplikasi. *Functional requirement* mencakup fitur, kemampuan, dan faktor keamanan *software* aplikasi.

Non-functional requirement menggambarkan atribut dari *software* aplikasi dan lingkungannya. *Non-functional requirement* dibagi menjadi *usability, reliability, performance* dan *supportability*.

2.5.6 Model Analisis

Model analisis menggambarkan realisasi dari *use-case-use-case* dalam *use-case model*, dan bertindak sebagai abstraksi dari model desain. Tujuan akhir sebuah model analisis adalah untuk membuat pemetaan awal mengenai perilaku yang diisyaratkan dalam sistem aplikasi ke dalam elemen-elemen pemodelan.

Elemen model yang terdapat dalam model analisis disebut kelas analisis. Kelas analisis adalah kelas berstereotype "*boundary*", "*control*", atau "*entity*" yang menggambarkan konsep awal mengenai "benda" dalam sistem aplikasi yang memiliki tanggung jawab dan perilaku.

2.6 Basis Data

Basis Data adalah suatu kelompok/kumpulan data operasional lengkap dari suatu perusahaan yang dikelola dan disimpan secara baik dan rapi dengan menggunakan cara-cara tertentu yang dibantu komputer agar informasi yang disediakan kepada pengguna dapat dilakukan secara optimal.

Sistem basis data menurut MAR [4] adalah **suatu sistem untuk menyusun dan mengelola *record-record* menggunakan komputer untuk menyimpan atau merekam serta memelihara data operasional lengkap sebuah organisasi/perusahaan sehingga mampu menyediakan informasi yang optimal yang diperlukan pemakai untuk proses mengambil keputusan.**

2.7 Arsitektur Aplikasi

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

2.7.1 Pengertian Jaringan Komputer

Jaringan Komputer menurut SOP [6] merupakan **sekelompok komputer otonom yang saling berhubungan antara satu dengan yang lainnya menggunakan protokol komunikasi melalui media komunikasi hingga dapat saling berbagi informasi, program-program, penggunaan bersama perangkat keras seperti printer, harddisk, dan sebagainya.**

2.7.2 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 diset-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*.

Dalam perancangan sistem informasi penjadwalan kuliah ini jenis jaringan yang digunakan adalah *client/server*.

2.7.3 Topologi Jaringan Komputer

Yang dimaksud topologi jaringan adalah susunan fisik bagaimana *node-node* saling dihubungkan. Ada tiga jenis topologi jaringan yang digunakan, yaitu :

1. Topologi Bus

   Topologi bus menghubungkan sejumlah komputer dengan satu sirkuit tunggal yang bisa terbuat dari kabel kembar (*twisted wire*), kabel coaxial, atau kabel serat optic. Semua sinyal dipancarkan ke semua arah jaringan, dengan *software* khusus untuk mengidentifikasi komponen mana yang menerima pesan. Jika satu dari komputer dalam jaringan gagal bekerja, tidak ada satupun komputer dalam jaringan yang terpengaruh. Topologi ini biasa digunakan untuk jaringan lokal (LAN).

2. Topologi Ring

   Jaringan ring tidak tergantung pada komputer induk dipusat dan tidak perlu mati jika satu dari komponen komputer tidak berfungsi dengan baik. Setiap komputer dalam jaringan dapat berkomunikasi secara langsung dengan komputer lain, dan setiap komputer memproses aplikasinya sendiri-sendiri. Data melalui ring dari satu komputer ke komputer lain dan mengalir searah.

3. Topologi Star

   Jaringan star terdiri dari komputer induk dipusat yang dihubungkan dengan komputer atau terminal yang lebih kecil. Topologi ini bermanfaat untuk aplikasi dimana pemrosesan harus terpusat dan data dijalankan secara lokal. Semua komunikasi antara titik dalam jaringan harus melewati komputer pusat. Karena komputer pusat merupakan pengawas lalulintas bagi komputer dan terminal lain dalam jaringan, komunikasi dalam jaringan akan terhambat jika komputer induk berhenti berfungsi.

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

2.7.4 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.

2. Media komunikasi

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

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.

2.8 Pengertian *Client / Server*

Client adalah setiap komponen dari sebuah sistem yang meminta layanan atau sumber daya (*resources*) dari komponen sistem lainnya. *Server* adalah setiap komponen sistem yang menyediakan layanan atau sumber daya ke komponen sistem lainnya. Sistem *client/server* dirancang untuk memisahkan layanan basisdata dari *client*, dengan penghubungnya menggunakan jalur komunikasi data. Layanan basisdata diimplementasikan pada sebuah komputer yang berdaya guna, yang memungkinkan manejemen tersentralisasi, keamanan, dan berbagi sumber daya.

2.9 Perangkat Lunak Pendukung

Perangkat lunak pendukung dalam pembuatan perangkat lunak penjadwalan pengajar yaitu diantaranya

2.9.1 Sekilas Tentang Borland Delphi 6.0

Borland Delphi merupakan suatu bahasa pemrograman yang memberikan berbagai fasilitas pembuatan aplikasi visual. Keunggulan bahasa pemrograman ini terletak pada produktifitas, kualitas, pengembangan perangkat lunak, kecepatan kompilasi, pola desain yang menarik serta diperkuat dengan pemrogramannya yang terstruktur. Keunggulan lain dari Delphi adalah dapat digunakan untuk merancang program aplikasi yang memiliki tampilan seperti program aplikasi lain yang berbasis Windows.

Khusus untuk pemrograman database, Borland Delphi menyediakan fasilitas objek yang kuat dan lengkap yang memudahkan programmer dalam membuat program.

Borland Delphi 6.0 merupakan pengembangan dari program Borland Delphi 5.0. lebih banyak keunggulan dan fasilitas baru yang diberikan oleh Borland Delphi 6.0 dibandingkan dengan versi-versi sebelumnya

2.9.2 Microsoft Access

Microsoft Access merupakan aplikasi pengolahan database model relasional yang berbasis windows. Maksud dari database itu sendiri adalah merupakan kumpulan arsip data berbentuk tabel yang saling berkaitan untuk menghasilkan informasi yang paling optimal bagi berbagai keperluan. Pengolahan data dengan

sistem database model relational ini, disamping dapat menghasilkan informasi yang lengkap dan beragam sesuai dengan keperluan pemakai, juga dapat mencapai tingkat efisiensi penyimpanan dan entri data serta efektifitas pengolahan data yang lebih tinggi dibanding dengan sistem pengolahan data non-database. Sistem pengolahan data model relational ini kemudian dikenal dengan sebutan sistem pengolahan model database (DBMS). Setiap database ini memiliki beberapa tabel data yang saling berkaitan dan pada setiap tabel terdiri dari beberapa *record* (baris) dan *field* (identik dengan kolom).

2.9.3 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, 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 sistem 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)

2.10 Penjadwalan

Penjadwalan adalah Sebagai proses pengalokasian sumber – sumber untuk memilih tugas yang tepat dalam jangka waktu tertentu.

Penjadwalan dapat pula didefinisikan sebagai pembuatan suatu daftar tertentu yang tersusun atau teratur berdasarkan kebutuhan agar suatu kegiatan atau aktifitas dapat berjalan dengan rapi. Dengan demikian penjadwalan kuliah merupakan penjadwalan atau pembuatan suatu susunan daftar kuliah yang tersusun atau teratur agar kegiatan belajar mengajar dapat berlangsung.

Proses penjadwalan banyak digunakan dalam berbagai bidang diantaranya penjadwalan mesin produksi, penjadwalan keberangkatan kereta api, penjadwalan penyelesaian proyek, penjadwalan kuliah dan lain sebagainya.

Dalam pembuatan atau penyusunan jadwal seorang penyusun jadwal dihadapkan pada persoalan bagaimana mengalokasikan sumberdaya yang ada sehingga waktu yang diperlukan untuk memproses aktivitas – aktivitas dapat dioptimasi dengan baik.

Dalam skripsi ini akan dibahas mengenai penjadwalan kuliah. Dalam penyusunan jadwal kuliah terdapat tiga buah komponen yang saling terkait yang terdiri dari dua sumber yaitu pengajar (dosen) dan ruangan, dan satu kegiatan atau aktivitas yaitu kuliah.