BAB II

LANDASAN TEORI

2.1. E-Voting

2.1.1. Definisi E-Voting

1. ***E-Voting*** adalah suatu metoda pengumpulan suara dengan menggunakan perangkat elektronik
2. ***Remote voting*** adalah suatu cara/metoda pemberian suara pemilih pada suatu tempat dimana saja, seperti di rumah, di kantor atau dijalan, artinya dilokasi mana saja asal ada kesempatan bisa melakukan pemilihan suara.
3. ***Supervised voting*** adalah suatu proses pemilihan suara (memberikan suara) yang diawasi oleh petugas/pejabat pemilihan umum (KPU).
4. ***Remote electronic voting*** adalah pemilihan suara pada suatu tempat dengan menggunakan perangkat elektronik, tanpa pengawasan petugas/pejabat pemilihan umum (KPU)
5. ***Internet Voting*** adalah pemilihan suara melalui media internet, bisa dimana saja seperti dirumah , diperjalanan atau di tempat yang sudah disediakan

2.1.2. Macam-Macam Teknologi Voting

Ada enam macam teknologi voting yang umum digunakan yaitu:

1. Kertas Suara/ Surat Pemilihan (*Paper Ballots*)

Tekonologi ini adalah teknologi yang pertama dalam peradaban umat manusia dalam berdemokrasi, dimana kertas suara dijadikan dasar untuk

menghitung suara pemilih. Cara melakukan pemilihan adalah pemilih mengambil kertas suara yang sudah disediakan dalam bentuk formulir, nama-nama calon dan gambarnya sudah tercetak, setelah itu pemilih tinggal menusuk atau mencoblos photo atau symbol calon pilihannya dan memasukkannya kedalam suatu kotak suara yang sudah disediakan oleh petugas. Selanjutnya team atau anggota panitia akan melakukan penghitungan suara.

2. *Lever Machines*

Teknologi berikutnya (*Technological advance*) adalah *Lever Machines* yang dimulai diperkenalkan pada tahun 1892. Teknologi ini tidak terdapat dokumen suara. Pemilih memasukkan suara dalam suatu tempat dengan memilih daftar calon dan mengumpulkan masing-masing calon terpilih. Suara dicatat dan dihitung dengan *Lever Machines*

3. *Punchcards*

Teknologi *punchcard,* pertama kali dipakai untuk menghitung suara dengan menggunakan komputer yang dimulai pada tahun 1964. Dalam sistem ini, suara dicatat dengan memilih lubang-lubang padu kartu atau kertas komputer dan selanjutnya komputer akan membaca kartu suara. Kartu suara adalah sebagai dokumen suara pemilih yang tercatat.

Ada dua tipedari sistem *punchcard* yaitu kotak nomor dicetak pada kartu suara, dimana setiap kotak untuk pemilihan suara. Dan yang lainnya disebut *Datavote* yang mempunyai lubang-lubang pemilih yang menyatakan nama-nama kandidat atau memilih kandidatnya dengan cara melubangi kertas *punchcard* yang dicetak pada kartu suara.

4. *Marksense Form*

Teknologi ini dinamakan *optical scan* yang dimulai digunakan pada tahun 1980. Pada sistem ini pemilih menggunakan bentuk kertas dan menulis pada kotak atau bentuk oval berikut arah panah untuk mengarahkan calon pemilih. Jika suara sudah lengkap ditulis kemudian dibaca oleh komputer. Tulisan pemilih ditempatkan pada suatu tempat perhitungan dan selanjutnya akan dibaca melalui proses *optical scanning* dan langsung dihitung dengan bantuan mesin penghitung. Kira-kira 25 persen dari seluruh Negara telah menggunkan alat dengan cara seperti ini. Pada tahun 1992 telah meningkat dua kali lipat penggunaannya dan terus meningkat pemakainya

5. *Electronic Voting*

Teknologi *electronic voting* dimulai pada tahun 1970 yang disebut teknologi pencatatan langsung secara elektronik atau lebih dikenal dengan istilah DRE (*direct recording electronic*). Cara memilih dengan sistem ini adalah dengan memilih kandidat yang sudah tercetak pada layar komputer. Pemilih hanya menekan tombol pada display atau pada alat atau piranti yang mirip.

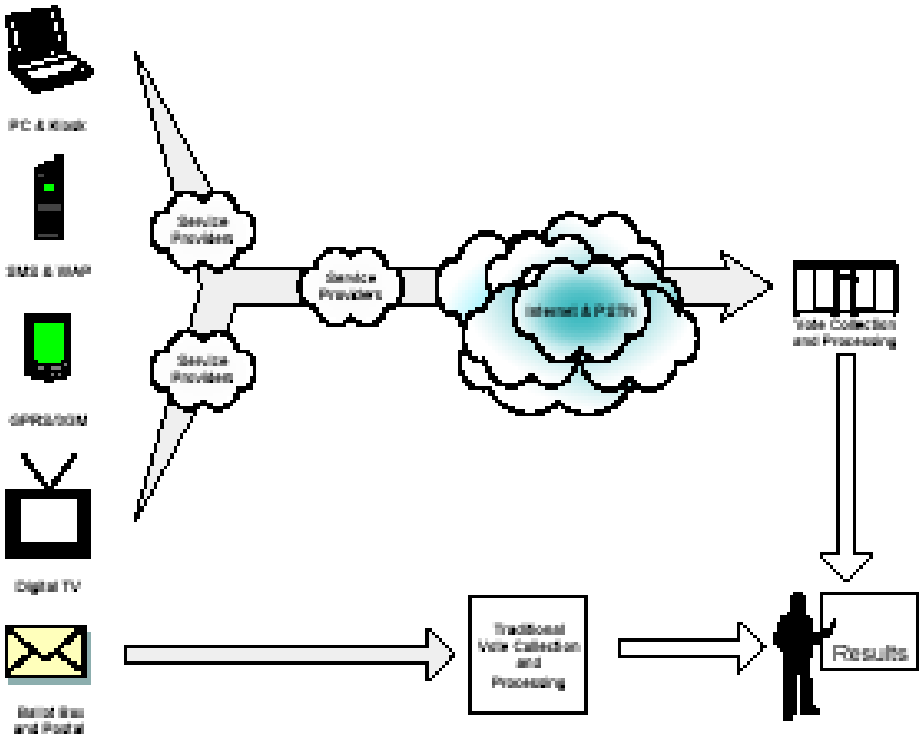
Contoh dari *electronic voting* adalah dengan menekan tombol suara pemilih langsung disimpan pada suatu piranti memori atau pada sirkit memori *non volatile*. Jika peralatan pemilihan menggunakan *keyboard* tulisan suara akan dicatat secara elektronik. Salah satu bentuk *electronic voting electronic voting* yang sedang dikembangkan adalah *Internet Voting.*

6. *Remote Voting*

Remote voting adalah suatu tempat pemungutan suara yang letaknya berjauhan atau pada tempat yang berbeda, teknologi ini bias menggunakan kertas yang dikirim melalui surat suara atau kartu suara, atau menggunakan suatu perangkat dengan kata lain bagaimana dokumen suara bisa dikirim ke suatu tempat untuk dihitung ditempat yang lain, cara ini tidak ada bedanya dengan pemungutan suara melalui surat , jadi *remote voting* adalah pemungutan suara dari tempat yang berbeda hanya saja cara atau media yang berbeda atau perangkat yang digunakan bisa berbeda-beda.**[3]**

2.1.3. Teknik Pengiriman E-Voting

Pengiriman *E-voting* dapat dilakukan dengan berbagai metoda atau dengan berbagai perangkat elektronik yang menjadi sarana pendukungnya.seperti contoh dibawah ini. Dalam gambar 2.1 adalah ilutrasi pengiriman *E-voting* dengan *multi flatform* artinya dengan sarana/ divice yang berbeda.



Gambar 2.1 Arsitektur pengiriman *e-voting*

1. Pemilih dapat memberikan suara melalui Internet

Dengan menggunakan bentuk web yaitu berupa halaman web di internet maka dapat mengirimkan input data dalam bentuk *alphanumeric*. Data yang dikirim sudah di enkripsi terlebih dahulu untuk menghindari dari penyadapan atau perubahan data.



Gambar 2.2 Media Internet

2. Pemilih dapat memberikan suara melalui telepon

Dalam pesawat telepon terbaru menggunakan sistim digital , pada tombol-tombol nya berbasis angka., tetapi dapat mengidentifikasi pernyataan yes atau no atau satu, dua ,tiga.



Gambar 2.3 Telepon Digital

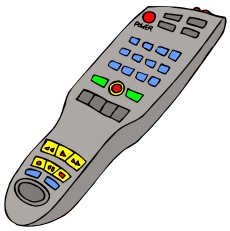
3. Pemilih dapat memberikan suara melalui pesan *Text SMS*

Pemilih diberikan dokumen petunjuk, user name dan PIN terlebih dahulu. Selanjutnya, sebelum mengirimkan pilihannya, pemilih juga diminta untuk memberikan nomor identitas atau nomor KTP sebagai salah satu usaha untuk mencegah pencurian identitas oleh orang lain. Jika pemilih sudah memberikan suara sebelumnya, tentu saja akan ditolak oleh sistem. Setelah mengirimkan SMS dengan format tersebut, pemilih akan mendapatkan SMS balasan berisi konfirmasi bahwa pilihan mereka telah diterima.



Gambar 2.4 Telepon seluler

4. **Pemilih dapat memberikan suara melalui TV digital dengan menggunakan *remote control* TV digital.**



Gambar 2.5 *Remote* TV digital

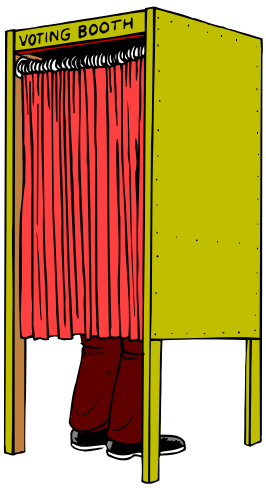
5. **Pemilih dapat memberikan suara melalui surat dan dikirim melalui *post***



Gambar 2.6 Kotak surat

6. Pemilih dapat memberikan suara melalui Kios (bilik suara)

Pemilih menyalurkan suaranya dengan cara datang ke suatu tempat yang sudah disediakan oleh panitia dan pemilih menyalurkan suaranya di bilik suara yang tertutup agar hasil pilihannya tidak diketahui oleh orang lain. Cara pemilihan seperti ini di sudah tidak asing lagi di Indonesia sudah.



Gambar 2.7 *Kios* (bilik suara)

2.2. Sistem

2.2.1. Analisis Sistem

Analisis sistem *(sistem analis)* merupakan tahapan yang sangat kritis dan penting karena kesalahan dalam tahap ini dapat mempengaruhi tahapan selanjutnya, Analisa sistem didefinisikan sebagai berikut :

"Penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya".

Dalam tahap analisis terdapat langkah-langkah sebagai berikut:

a. *Identify* yaitu mengidentifikasi masalah.
b. *Understand* yaitu memahami kerja dari sistem yang ada.
c. *Analyze* yaitu menganalisis sistem.
d. *Report* yaitu membuat laporan hasil analisis.

2.2.2. Desain Sistem

Desain sistem didefinisikan sebagai berikut :

"Penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi".

a. Tujuan Desain Sistem.

Desain sistem memunyai maksud dan tujuan sebagai berikut:

1. Untuk memenuhi kebutuhan kepada pemakai sistem.
2. Untuk memberikan gambaran yang jelas dan rancang bangun yang lengkap kepada pemrogram komputer dan ahli-ahli teknik yang terlibat.

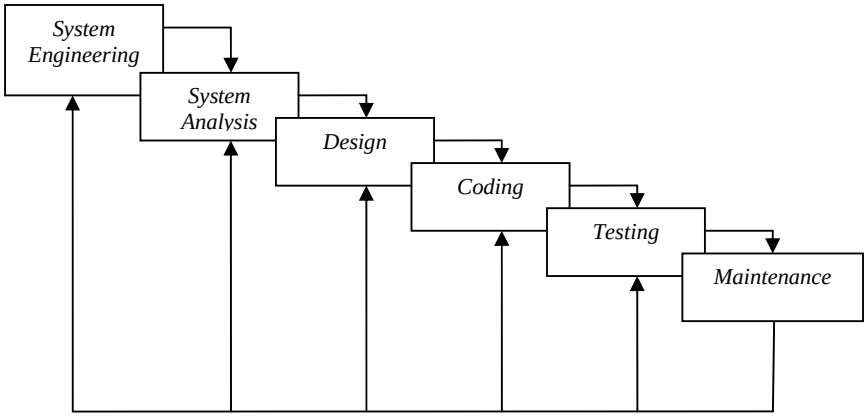
Sasaran-sasaran yang harus dicapai agar desain sistem mencapai tujuan:

1. Desain sistem harus berguna, mudah dipahami dan nantinya mudah digunakan.
2. Desain sistem harus dapat mendukung tujuan utama perusahaan.
3. Desain harus efektif dan efisien mendukung pengolahan transaksi yang, pelaporan manajemen dan mendukung keputusan yang akan dilakukan oleh manajemen, termasuk tugas-tugas lainnya yang tidak dapat dilakukan oleh komputer.

4. Desain sistem harus dapat mempersiapkan rancang bangun yang terinci untuk masing-masing komponen yang meliputi data dan informasi, simpanan data, metode-metode, prosedur-prosedur, orang-orang, perangkat keras, perangkat lunak dan pengendalian intern.

2.2.3. Model Perancangan Sistem

Langkah awal yang dilakukan dalam membangun sistem adalah dengan menentukan model sistem yang akan digunakan. Berikut ini akan dijelaskan mengenai model sistem *waterfall*. Dalam penelitian ini model sistem yang digunakan adalah model sistem *waterfall* dengan struktur seperti pada gambar 2.8 Berikut ini :



Gambar 2.8 Model Sistem Waterfall

Model sistem ini menuntut cara yang teratur dari suatu rangkaian yang mendekati perkembangan *software*, yang dimulai dengan suatu tingkatan kemajuan-kemajuan melalui analisa, perancangan, pengkodean, pengujian dan pemeliharaan. Model tersebut meliputi kegiatan aktifitas sebagai berikut :

a. *System/Information Engineering*

Merupakan bagian dari sistem yang terbesar dalam pengerjaan suatu proyek, dimulai dengan menetapkan berbagai kebutuhan dari semua elemen yang diperlukan sistem dan mengalokasikannya kedalam pembentukan perangkat lunak.

b. *System Analysis*

Merupakan tahap menganalisis hal-hal yang diperlukan dalam pelaksanaan proyek pembuatan perangkat lunak.

c. *Design*

Perancangan berdasarkan analisis data kedalam bentuk gambaran aplikasi yang dibuat.

d. *Coding*

Menerapkan hasil perancangan dari aplikasi yang dibangun kedalam bahasa pemrograman tertentu.

e. *Testing*

Pengujian terhadap aplikasi yang dibangun.

f. *Maintenance*

Tahap akhir dimana suatu perangkat lunak yang sudah selesai dapat mengalami perubahan-perubahan atau penambahan sesuai dengan permintaan *user*.

2.2.4. Alat Pengembangan Sistem

Untuk dapat melakukan langkah-langkah sesuai dengan yang diberikan oleh metodologi pengembangan sistem yang terstruktur, maka dibutuhkan alat

untuk melaksanakannya. Alat-alat yang digunakan dalam suatu metodologi umumnya berupa suatu gambar, diagram, atau grafik. Selain berbentuk gambar, alat-alat yang digunakan juga ada yang tidak berupa gambar atau grafik, seperti misalnya kamus data (*data dictionary*), *pseudocode*, serta formulir-formulir untuk mencatat dan menyajikan data, seperti :

1. **Bagan Alir Informasi**, bagian yang menggambarkan arus informasi berupa laporan, formulir, dan dokumen yang keluar atau masuk dari bagian – bagian tertentu dalam sistem.
2. **Bagan Alir Sistem (*System Flowchart*)**, merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem, yang menjelaskan urutan – urutan dari prosedur – prosedur yang ada didalam sistem.
3. **Bagan Hubungan Database (*Relational Database*)**, dimaksudkan untuk mengidentifikasikan kebutuhan dan hubungan dari file – file database yang digunakan.

2.3. *Software* Pendukung

2.3.1. Netbeans 6.5

Netbeans sebagai *IDE* ditujukan untuk memudahkan pemrograman Java. Dalam Netbeans, pemrograman dilakukan berbasiskan visual dan *event driven*. Seperti *IDE* lain, misalnya Borland Delphi dan Microsoft Visual Studio. Untuk membuat dialog atau *User Interface*, kita tidak perlu membuat teks program secara manual baris per baris, tetapi cukup klik pada *component pallete*. Teks program akan dihasilkan secara otomatis.

Untuk memakai Netbeans *IDE* maka kita harus terlebih dahulu mempunyai driver JDK yang akan mendukung pembuatan perangkat lunak dengan menggunakan Netbeans *IDE*, maka sebelum meng*install* Netbeans terlebih dahulu kita harus menginstal driver JDK. Netbeans yang dipakai pada pengembangan perangkat lunak ini adalah Netbeans 6.5 dan JDK 1.6.**[12]**

2.3.2. MySQL

MySQL adalah perangkat lunak *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis dibawah lisensi *GPL* (*General Public License*). Dimana setiap orang bebas untuk menggunakannya, tapi tidak boleh dijadikan produk turunan yang bersifat *Closed Source* atau komersial.

MySQL sebenarnya merupakan turunan salah satu konsep utama dalam *database* sejak lama, yaitu *SQL* (*Structure Query Language*). *SQL* adalah sebuah konsep pengoperasian *database*, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Keandalan suatu sistem *database* (DBMS) dapat diketahui dari cara kerja *Optimizer*-nya dalam melakukan proses perintah-perintah *SQL*, yang dibuat oleh *user* maupun program-program aplikasinya. Sebagai *database server*, MySQL dapat dikatakan lebih unggul dibanding *database server* lainnya dalam *query* data. Hal ini terbukti untuk *query* yang dilakukan oleh *single user*, kecepatan *query* MySQL bisa sepuluh kali lipat lebih cepat dari PostgreSQL dan lima kali lebih cepat dibanding Interbase.

2.4. *Unified Modeling Language* (UML)

Unified Modelling Language (UML) adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan software berbasis OO (*Object-Oriented*). UML sendiri juga memberikan standar penulisan sebuah sistem blue print, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema database, dan komponen-komponen yang diperlukan dalam sistem software.**[10]**

Meskipun UML adalah suatu standard dari Object Management Group (OMG), UML tidak hanya untuk memodelkan aplikasi perangkat lunak yang berorientasi objek. UML adalah suatu bahasa grafik yang dirancang untuk menjadi sangat fleksibel dan customizable. Hal ini memungkinkan kita untuk membuat beraneka jenis model, termasuk model untuk mengerti proses-proses bisnis, workflow, urutan dari query, aplikasi, database, arsitektur, dan lainnya**[9]**.

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan syntax/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*). **[10]**

2.4.1. Sejarah UML

Pendekatan analisa & rancangan dengan menggunakan model OO mulai diperkenalkan sekitar pertengahan 1970 hingga akhir 1980 dikarenakan pada saat itu aplikasi software sudah meningkat dan mulai komplek. Jumlah yang menggunakaan metoda OO mulai diuji cobakan dan diaplikasikan antara 1989 hingga 1994, seperti halnya oleh Grady Booch dari *Rational Software Co.,* dikenal dengan OOSE (*Object-Oriented Software Engineering*), serta James Rumbaugh dari *General Electric*, dikenal dengan OMT (*Object Modelling Technique*).**[11]**

Secara resmi bahasa UML dimulai pada bulan oktober 1994, ketika Rumbaugh bergabung Booch untuk membuat sebuah project pendekatan metoda yang uniform/seragam dari masing-masing metoda mereka. Saat itu baru dikembangkan draft metoda UML version 0.8 dan diselesaikan serta di release pada bulan oktober 1995. Bersamaan dengan saat itu, Jacobson bergabung dan UML tersebut diperkaya ruang lingkupnya dengan metoda OOSE sehingga muncul release version 0.9 pada bulan Juni 1996. Hingga saat ini sejak Juni 1998 UML version 1.3 telah diperkaya dan direspons oleh OMG (Object Management Group), Anderson Consulting, Ericsson, Platinum Technology, ObjectTime Limited, dll serta di pelihara oleh OMG yang dipimpin oleh Cris Kobryn.**[10]**

UML adalah standar dunia yang dibuat oleh *Object Management Group* (OMG), sebuah badan yang bertugas mengeluarkan standar-standar teknologi *object-oriented* dan *software component*. **[11]**

2.4.2. Konsep Dasar UML

Dari berbagai penjelasan rumit yang terdapat di dokumen dan buku-buku UML. Sebenarnya konsepsi dasar UML dapat dirangkum dalam tabel berikut ini :

Tabel 2.1 Abstraksi Konsep Dasar UML

Major Area	**View**	**Diagrams**	**Main Concepts**
Structural	static view	class diagram	Class, association, generalization, dependency, realization, antarmuka
	use case view	use case diagram	Use case, actor, association, extended, include, use case general-ization
	Implementation view, deployment view	Component diagram	Component, antarmuka, dependency, realization
		Deployment diagram	Mode, component, dependency,location
Dynamic	State machine view	State chart diagram	State event, transition, action
	Activity view	Activity diagram	State, activity, complemention, transition, fork, join
	Interaction view	Sequence diagram	Interaction, object, message, activation
		Collaboration diagram	Collaboration, interaction, collaboration role, message
Model management	Model managent view	Class diagram	Package, subsistem, model
Extensibility	All	All	Constraint, stereotype, tagget values

Abstraksi konsep dasar UML yang terdiri dari *structural classification*, *dynamic behavior*, dan *model management*, bisa kita pahami dengan mudah apabila kita melihat tabel diatas dari *Diagrams*. *Main concepts* bisa kita pandang

sebagai *term* yang akan muncul pada saat kita membuat diagram. Dan *view* adalah kategori dari diagaram tersebut.

2.4.3. Diagram-diagram UML

1. Use Case Diagram

Use case diagram menggambarkan sebuah fungsi yang dibutuhkan oleh sebuah sistem. Dalam hal ini ada kondisi yang agak beda, yaitu disini tingkah laku dan domain sistem. Sebuah use case merepresentasikan sebuah interaksi antara pengguna dengan sebuah sistem. Use case merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, membuat sebuah daftar aktifitas, dan sebagainya.

2. Class Diagram

Class merupakan inti dari pengembangan dan desain berorientasi objek, karena nantinya class ini akan menghasilkan sebuah objek. Class menggambarkan keadaan (atribut/property) suatu sistem dan metode/fungsi yang ada pada sistem tersebut.

Class diagram menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti pewarisan, asosiasi, dan lain-lain. Class memiliki tiga domain yaitu : Nama (dan stereotype), Atribut dan Metode. Atribut dan metode dapat memiliki salah satu sifat berikut :

1. Private, tidak dapat dipanggil diluar class yang bersangkutan
2. Protected, hanya dapat dipanggil oleh class yang bersangkutan dan anak-anak yang mewarisinya.
3. Public, dapat dipanggil oleh siapa saja.

3. Sequence Diagram

Sequnce diagram, secara khusus, menjabarkan behavior sebuah scenario tunggal. Diagram tersebut menunjukan sejumlah objek contoh dan pesan-pesan yang melewati objek-objek ini di dalam use case.

4. Activity Diagram

Activity Diagram adalah teknik untuk mendeskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. *Activity Diagram* mempunyai peran seperti halnya *flowchart,* akan tetapi perbedaannya dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku paralel sedangkan *flowcahrt* tidak bisa.

5. Deployment Diagram

Deployment/physical 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.

2.5. Basis Data

2.5.1. Pengertian Basis Data

Basis Data merupakan kumpulan data yang (arsip) yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa

pengulangan (*redudansi*) yang tidak perlu, untuk memenuhi berbagai kebutuhan. Atau bisa diartikan sebagai kumpulan *file*/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronis.

a. *Database Manajemen Sistem (DBMS*)

Pengelolaan basis data secara fisik tidak dilakukan oleh pemakai secara langsung, tetapi ditangani oleh sebuah perangkat lunak atau sistem yang khusus atau spesifik. Sistem ini yang akan menentukan bagaimana data diorganisasi, disimpan, diubah dan diambil kembali Disamping itu sistem ini juga menerapkan mekanisme pengaman data, pemakaian data secara bersama, pemaksaan keakuratan atau konsistensi data dan sebagainya. Perangkat lunak yang termasuk DBMS misalnya dBase II+, dBase IV, FoxBase, RBase, MS-*Access* dan *Borland-Paradox* atau *Borland-Interbase*, MS-SQL *Server*, CA-*Open Ingres*, *Oracle*, *Informix* dan *Sybase*.

b. Tujuan Basis Data

Basis Data pada prinsipnya ditujukan untuk pengaturan data agar terdapat kemudahan dalam pengambilan kembali data tersebut. Berikut ini terdapat beberapa tujuan dari Basis Data diantaranya yaitu :

1. Kecepatan dan kemudahan (*Speed)*
2. Efisiensi ruang penyimpanan (*Space)*
3. Keakuratan (*Accuracy)*
4. Ketersediaan (*Avaibility)*
5. Kelengkapan (*Completeness)*

6. Keamanan (*Security*)
7. Kebersamaan (*Sharability*)

c. Pengguna Basis Data

Pengguna basis data yang dibedakan berdasarkan cara mereka berinteraksi terhadap sistem diantaranya yaitu :

1. Programmer Aplikasi (*Application Programmer*)

   Pemakai yang berinteraksi dengan basis data dengan menggunakan *Data Manipulation Language* (DML) untuk membuat aplikasi dengan menggunakan bahasa pemrograman.

2. *User* Khusus (*Specialized User*)

   Pemakai yang membuat program aplikasi basis data untuk keperluan tertentu atau khusus.

3. *User* Mahir (*Casual User*)

   Pemakai yang berinteraksi dengan sistem tanpa membuat modul program.

4. *User* Umum (*Naïve User*)

   Pemakai berinteraksi dengan aplikasi basis data yang telah dibuat atau disediakan oleh sistem.

2.6. Object Oriented Programming (OOP)

OOP merupakan terobosan baru dalam pemrograman setelah pemrograman prosedural, pemrograman modular dan pemrograman abstraksi data. Ada beberapa konsep utama OOP antara lain :

Pemrograman berorientasi-objek menekankan konsep berikut:

1. Class : kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebagai contoh *'class of dog*' adalah suatu unit yang terdiri atas definisi-definisi data dan fungsi-fungsi yang menunjuk pada berbagai macam perilaku/turunan dari anjing.
2. Objek : membungkus data dan fungsi bersama menjadi suatu unit dalam sebuah program komputer; objek merupakan dasar dari modularitas dan struktur dalam sebuah program komputer berorientasi objek.
3. Abstraksi : kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti. Setiap objek dalam sistem melayani sebagai model dari "pelaku" abstrak yang dapat melakukan kerja, laporan dan perubahan keadaannya, dan berkomunikasi dengan objek lainnya dalam sistem, tanpa mengungkapkan bagaimana kelebihan ini diterapkan. Proses, fungsi atau metode dapat juga dibuat abstrak, dan beberapa teknik digunakan untuk mengembangkan sebuah pengabstrakan.
4. Enkapsulasi : memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam dari sebuah objek dengan cara yang tidak layak; hanya metode dalam objek tersebut yang diberi ijin untuk mengakses keadaannya. Setiap objek mengakses interface yang menyebutkan bagaimana objek lainnya dapat berinteraksi dengannya.

Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.

5. Polimorfisme : melalui pengiriman pesan. tidak bergantung kepada pemanggilan subrutin, bahasa orientasi objek dapat mengirim pesan; metode tertentu yang berhubungan dengan sebuah pengiriman pesan tergantung kepada objek tertentu di mana pesa tersebut dikirim. Contohnya, bila sebuah burung menerima pesan "gerak cepat", dia akan menggerakan sayapnya dan terbang. Bila seekor singa menerima pesan yang sama, dia akan menggerakkan kakinya dan berlari. Keduanya menjawab sebuah pesan yang sama, namun yang sesuai dengan kemampuan hewan tersebut. Ini disebut polimorfisme karena sebuah variabel tungal dalam program dapat memegang berbagai jenis objek yang berbeda selagi program berjalan, dan teks program yang sama dapat memanggil beberapa metode yang berbeda di saat yang berbeda dalam pemanggilan yang sama. Hal ini berlawanan dengan bahasa fungsional yang mencapai polimorfisme melalui penggunaan fungsi kelas-pertama.
6. *Inheritanc*e - Mengatur polimorfisme dan enskapsulasi dengan mengijinkan objek didefinisikan dan diciptakan dengan jenis khusus dari objek yang sudah ada, objek-objek ini dapat membagi (dan memperluas) perilaku mereka tanpa haru mengimplementasi ulang perilaku tersebut.

Dengan menggunakan OOP maka dalam melakukan pemecahan suatu masalah kita tidak melihat bagaimana cara menyelesaikan suatu masalah tersebut(terstruktur) tetapi objek-objek apa yang dapat melakukan pemecahan masalah tersebut. Sebagai contoh anggap kita memiliki sebuah departemen yang memiliki manager, sekretaris, petugas administrasi data dan lainnya. Misal manager tersebut ingin memperoleh data dari bagian administrasi maka manager tersebut tidak harus mengambilnya langsung tetapi dapat menyuruh petugas bagian administrasi untuk mengambilnya. Pada kasus tersebut seorang manager tidak harus mengetahui bagaimana cara mengambil data tersebut tetapi manager bisa mendapatkan data tersebut melalui objek petugas administrasi. Jadi untuk menyelesaikan suatu masalah dengan kolaborasi antar objek -objek yang ada karena setiap objek memiliki *job description* nya sendiri.