BAB II

TINJAUAN PUSTAKA

2.1 Sistem Informasi

Sistem informasi adalah kerangka kerja mengkoordinasikan sumber daya (manusia, komputer) untuk mengubah masukan (input) menjadi keluaran (informasi), guna mencapai sasaran-sasaran perusahaan (Kadir, 2003).

Menurut Bodnar dan Hopwood (Kadir, 2003), sistem informasi adalah kumpulan perangkat lunak yang dirancang untuk mentransformasikan data kedalam bentuk informasi yang berguna.

Menurut Gelians, Oram dan Wiggins (Kadir, 2003), sistem informasi adalah suatu sistem buatan manusia secara umum terdiri atas sekumpulan komponen berbasis komputer dan manual yang dibuat untuk menghimpun, menyimpan dan mengelola data serta menyediakan informasi keluaran kepada para pemakai.

Menurut Hall (Kadir, 2003), sistem informasi adalah sebuah rangkaian prosedur formal dimana data dikelompokkan, diproses menjadi informasi, dan didistribusikan kepada pemakai.

Dari beberapa pengertian di atas dapat diambil kesimpulan, sistem informasi merupakan komponen yang terdiri dari manusia, teknologi informasi, dan prosedur kerja yang memproses, menyimpan, menganalisis dan mendistribusikan informasi untuk mencapai tujuan.

2.1.2 Komponen Sistem Informasi

Komponen sistem informasi yang disebut blok bangunan yaitu : blok masukan, blok model, blok teknologi, blok *database* dan blok kendali (Jogiyanto, 2005). Keenam blok tersebut saling berinteraksi satu dengan lainnya membentuk satu kesatuan untuk mencapai sasarannya.

1. **Blok masukan**

Blok masukan mewakili data yang masuk ke dalam sistem informasi, termasuk metode dan media untuk memperoleh data yang akan dimasukkan, yang dapat berupa dokumen dasar.

2. **Blok model**

Blok model terdiri dari kombinasi prosedur, logika dan model matematika yang akan memanipulasi/mentranspormasi data masukan dan data yang tersimpan dalam *database* untuk menghasilkan keluaran yang diinginkan.

3. **Blok Keluaran**

Blok keluaran adalah produk dari sistem informasi adalah keluaran berupa informasi yang berkualitas.

4. **Blok Teknologi**

Blok teknologi merupakan kotak alat dalam sistem informasi. Teknologi terdiri dari 3 bagian utama yaitu teknisi (*brainware*), perangkat lunak (*software*) dan perangkat keras (*hardware*).

5. **Blok *Database***

Merupakan kumpulan file data yang saling berhubungan yang diorganisasi sedemikian rupa agar dapat diakses dengan mudah dan cepat.

6. Blok Kendali

Pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah atau bila terlanjur terjadi kesalahan dapat langsung diatasi.

2.2 Arsitektur Sistem Informasi

Arsitektur sistem informasi disebut juga sebagai arsitektur teknologi informasi, arsitektur *enterprise* atau infrastruktur teknologi informasi. Arsitektur dari sistem merupakan sekumpulan dari model-model terhubung yang menggambarkan sifat dasar dari sebuah sistem. Keanekaragaman dari banyak model menggambarkan bagian berbeda dan aspek atau pandangan yang berbeda dari suatu sistem. Komponen merupakan blok pembangun : sistem dapat dibangun dengan cara menyatukan sekumpulan komponen berdasarkan aturan tertentu. Pandangan yang berbeda dari tiap komponen bukan berarti komponen-komponen tersebut berlaku sebagai sebuah sistem yang berdiri sendiri. Biasanya, sudut pandang dari suatu sistem terbagi menjadi beberapa sudut pandang yaitu : sudut pandang bisnis, sudut pandang fungsional dan sudut pandang teknis. Masing-masing dari sudut pandang tersebut dapat dipecah lagi menjadi beberapa bagian. Arsitektur enterprise terdiri dari empat komponen :

1. *Business Architecture*, yaitu mendeskripsikan tentang bagaimana proses bisnis untuk mencapai tujuan organsasi .
2. *Data Architecture*, adalah penggambaran bagaimana penyimpanan, pengelolaan dan pengaksesan data pada perusahaan.

3. *Apllication architecture,* merupakan pendeskripsian bagaimana suatu aplikasi dirancang dan bagaimana interaksi dengan aplikasi yang lain.
4. *Technology architecture,* yaitu gambaran infrastruktur perangkat lunak dan perangkat keras yang mendukung aplikasi dan bagaimana interaksinya dengan aplikasi yang lain.

2.3 Kerangka Kerja *The Open Group Architecture Framework* (TOGAF)

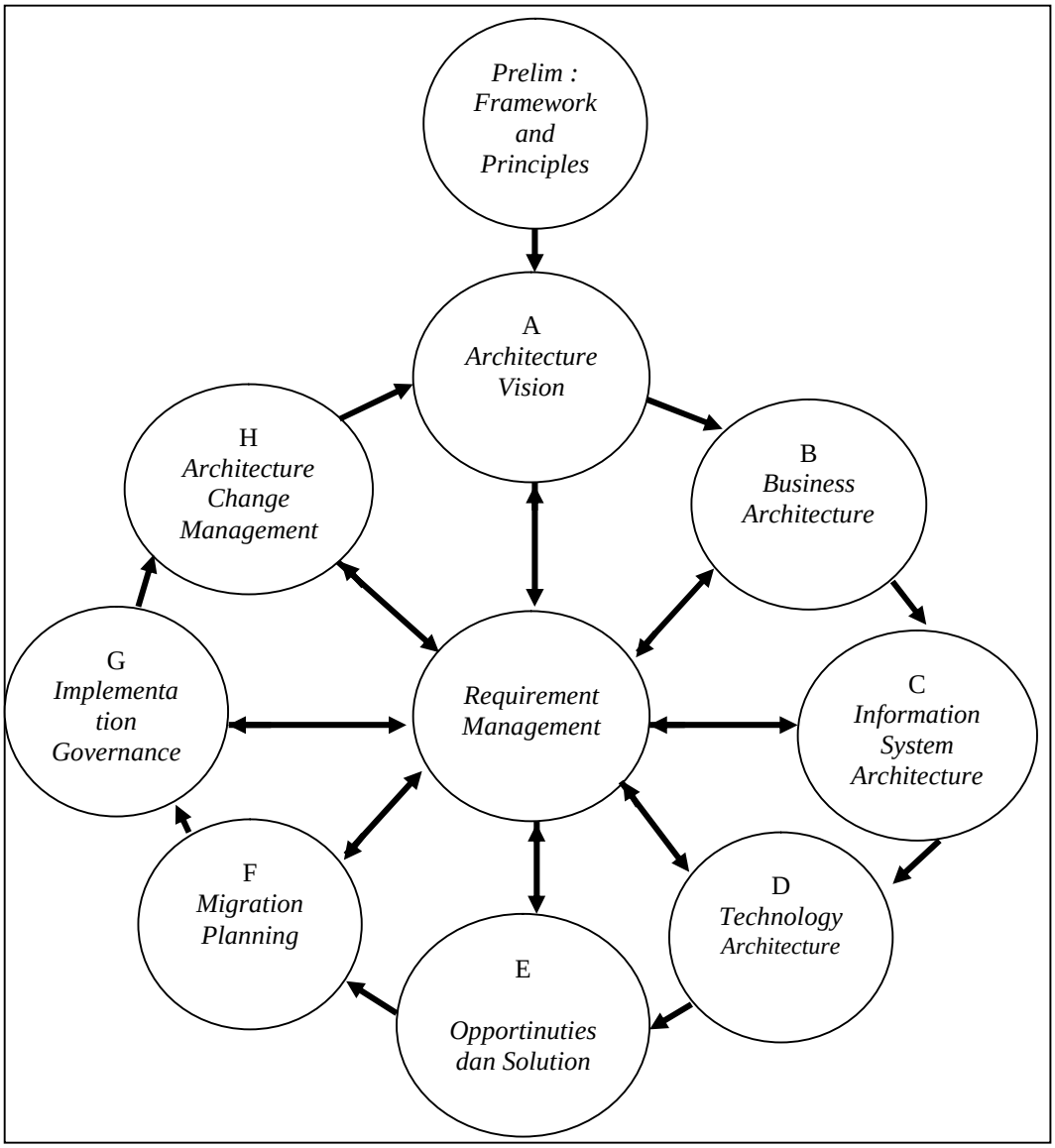
TOGAF merupakan kerangka kerja arsitektur *enterprise* yang dikembangkan oleh the open group's *Architecture Framework* pada tahun 1995 yang digunakan untuk mengembangkan arsitektur perusahaan. Pada awalnya TOGAF digunakan oleh Departemen Pertahanan Amerika Serikat, namun pada perkembangannya banyak digunakan pada berbagai bidang seperti industri manufaktur, perbankan, pendidikan dan lain sebagainya. TOGAF digunakan untuk mengembangkan arsitektur *enterprise,* dimana terdapat metode dan alat yang detail untuk mengimplementasikannya. Hal inilah yang membedakan dengan kerangka kerja arsitektur *enterprise* yang lain. Salah satu kelebihan dari kerangka kerja ini adalah sifatnya fleksibel dan *open source* (Setiawan, 2009a). TOGAF adalah salah satu metode yang paling banyak diterima untuk mengembangkan arsitektur perusahaan. TOGAF merupakan suatu kerangka kerja yang praktis, pasti dan dibuktikan dengan adanya tahapan-tahapan metode untuk mengembangkan dan mempertahankan arsitektur *enterprise* (Ugavina, 2009)

Secara umum TOGAF memiliki struktur dan komponen-komponen, yaitu :

1. *Architecture Development Method (ADM).* ADM merupakan bagian utama dari TOGAF yang menjelaskan bagaimana menentukan sebuah arsitektur *enterprise* secara khusus sesuai dengan kebutuhan.
2. *Foundation Architecture (Enterprise Continum). Foundation Architecture* merupakan sebuah "*framework-within-a-framework*" yang menyajikan gambaran hubungan bagi pengumpul arsitektur yang relevan dan menyediakan bantuan petunjuk pada waktu terjadi perpindahan abstraksi level yang berbeda. Di dalam *Foundation Architecture* terdapat tiga bagian yaitu *technical reference model, standard information,* dan *building block information base*.
3. *Resource base.* Pada bagian ini memberikan informasi berupa *guidelines, templates, checklist,* latar belakang informasi dan detail material pendukung ke yang membantu arsitek dalam penggunaan ADM.

TOGAF mewujudkan konsep *enterprise continuum* untuk mencerminkan tingkat abstraksi yang berbeda dalam sebuah proses pembangunan arsitektur. Dengan cara ini TOGAF memfasilitasi pemahaman dan kerjasama antar aktor pada tingkat yang berbeda. TOGAF menyediakan konteks bagi penggunaan dari beberapa kerangka kerja, model, dan asset arsitektur dalam hubungannya dengan TOGAF ADM. Dengan cara *enterprise continuum,* arsitek didorong untuk memanfaatkan sumber daya arsitektur lain yang relevan dan asset-asset. Selain itu TOGAF sebagai dasar arsitektur dalam mengembangkan teknologi informasi di suatu organisasi.

TOGAF terdiri atas 8 (delapan) fase yang berbentuk siklus (*cycle*). Pada fase ke 4 di fokuskan pengembangan arsitektur teknologi. Fase-fase dalam metode TOGAF dapat dilihat di gambar 2.1



Gambar 2.1 Proses Pengembangan TOGAF ADM (Lankhort & Drunen Hans van 2007)

Tahapan dari TOGAF ADM dapat dijelaskan sebagai berikut :

1. *Preliminary dan principle* (Tahapan A)

Tahapan persiapan (*Preliminary Stage*) merupakan tahapan *tools* untuk menentukan ruang lingkup *Enterprise Architecture (EA)* yang akan dikembangkan serta menentukan komitmen dengan manajemen dalam pengembangan EA.

2. *Architecture Vision* (Tahapan B)

   Menciptakan keragaman pandangan mengenai pentingnya arsitektur *enterprise* untuk mencapai tujuan organisasi yang dirumuskan dalam bentuk strategi serta menentukan ruang lingkup dari arsitektur yang akan dikembangkan. Pada tahapan ini berisikan kebutuhan-kebutuhan berkenaan dengan perancangan arsitektur sistem informasi yaitu profil organisasi, pendefinisan visi dan misi, tujuan organisasi, sasaran organisasai, proses bisnis organisasi, unit organisasi dan kondisi arsitektur saat ini.

3. *Business Architecture* (Tahapan C)

   Mendefinisikan kondisi awal arsitektur bisnis, menentukan model bisnis atau aktivitas bisnis yang diinginkan berdasarkan skenario bisnis. Pada tahapan ini *tools* dan *method* umum untuk permodelan seperti : *Unified Modeling Languange* (UML) bisa digunakan untuk membangun model yang diperlukan.

4. *Information System Architecture* (Tahapan D)

   Pada tahapan ini lebih menekankan pada aktivitas bagaimana arsitektur sistem informasi dikembangkan. Pendefinisian arsitektur sistem informasi dalam tahapan ini meliputi arsitektur data dan aristektur aplikasi yang akan digunakan dalam organisasi. Arsitektur data lebih memfokuskan bagaimana data

digunakan untuk kebutuhan fungsi bisnis, proses dan layanan. Teknik yang bisa digunakan yaitu *ER-Diagram, Class Diagram,* dan *Object Diagram.*

5. *Technology Architecture* (Tahapan E)

   Membangun arsitektur teknologi yang diinginkan dimulai penulisan jenis kandidat teknologi yang diperlukan dengan menggunakan *Technology Portofolio Catalog* yang meliputi perangkat lunak dan perangkat keras. Dalam tahapan ini juga mempertimbangkan alternatif-alternatif yang diperlukan dalam pemilihan teknologi.

6. *Opportunities and Solution* (Tahapan F)

   Pada tahap ini lebih menekan manfaat yang diperloleh dari arsitektur *enterprise* yang meliputi arsitektur bisnis, arsitektur data, arsitektur aplikasi dan arsitektur teknologi, sehingga menjadi dasar *stakeholder* untuk memilih dan menentukan arsitektur yang akan diimplementasikan.

7. *Migration Planning* (Tahapan G)

   Pada tahap ini akan dilakukan penilaian dalam menentukan rencana migrasi dari suatu sistem informasi. Biasanya pada tahapan ini untuk permodelannya menggunakan matrik penilaian dan keputusan terhadap kebutuhan utama dan pendukung dalam organisasi terhadap implementasi sistem informasi.

8. *Implementation Governance* (Tahapan H)

   Menyusun rekomendasi untuk pelaksanaan tatakelola implementasi yang sudah dilakukan, tatakelola yang dilakukan meliputi tatakelola organisasi, tatakelola teknologi informasi, dan tatakelola arsitektur.

9. *Management Change Architecture* (Tahapan I)

Menetapkan rencana manajemen arsitektur dari sistem yang baru dengan cara melakukan pengawasan terhadap perkembangan teknologi dan perubahan lingkungan organisasi, baik internal maupun eksternal serta menentukan apakah akan dilakukan siklus pengembangan arsitektur *enterprise*.

Dalam kerangka kerja ini terdapat kelebihan dan kelemahan. Menurut Mutyarini dan sembiring (2006) bahwa kelebihan TOGAF adalah sebagai berikut:

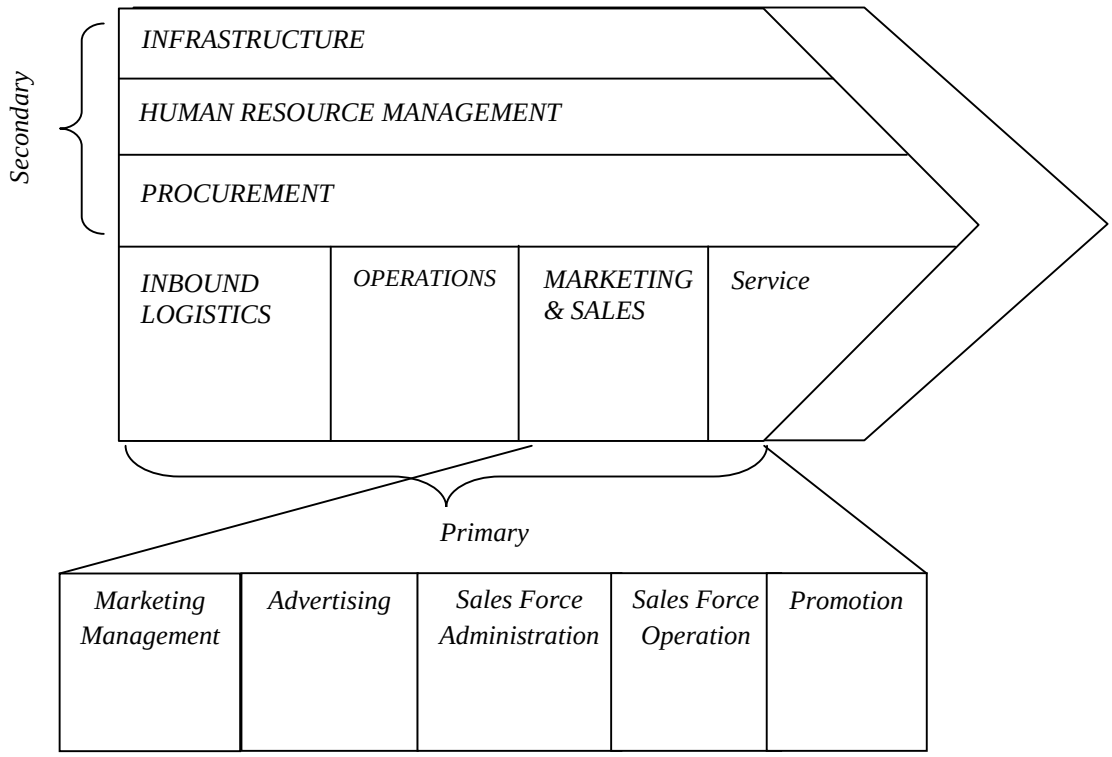
1. Fokus pada siklus implementasi (ADM) dan proses
2. Terdapat banyak area teknis arsitektur
3. *Reosource base* menyediakan banyak material referensi

Sedangkan kelemahan pada TOGAF adalah :

1. Tidak terdapat templates standar untuk seluruh domain seperti dalam membuat blok diagram tidak terdapat templates yang baku.
2. Tidak terdapat artefak yang dapat digunakan ulang (*ready made).*

2.4 *Value Chain* Michael E. Porter

Fungsi dari *value chain*, menurut Michael E. Porter (gambar 2.2) yaitu untuk mendeskripsikan cara melihat bisnis sebagai rantai aktivitas yang mengubah input menjadi *output* sehingga memiliki nilai bagi pelanggan (Porter, E.michael, 1985).



Gambar 2.2 *Value chain* Michael E. Porter

Value chain membagi dalam dua kategori, yaitu:

1. ***Primary activities***, (*line functions*) merupakan aktivitas utama dari organisasi yang melibatkan aktivitas-aktivitas sebagai berikut :
   a. ***Inbound logistics***, pada bagian ini terkait dengan penerimaan, penyimpanan, dan pendistribusian *input* menjadi produk.
   b. ***Operations***, semua aktivitas yang terkait dengan pengubahan input menjadi bentuk akhir dari produk, seperti produksi, pembuatan, pemaketan, perawatan, peralatan, fasilitas, operasi, jaminan kualitas, proteksi terhadap lingkungan.
   c. ***Outbond Logistics***, aktivitas yang terkait dengan pengumpulan, penyimpanan, distribusi secara fisik atau pelayanan terhadap pelanggan.

d. ***Marketing and Sales***, aktivitas yang terkait dengan pembelian produk dan layanan oleh pengguna dan mendorong untuk dapat membeli produk yang dibuat. Memiliki rantai nilai khusus, antara lain :

(1) *Marketing management*

(2) *Advertising*

(3) *Sales force administration*

(4) *Sales force operations*

(5) *Technical literature*

(6) *Promotion*

e. ***Service***, aktivitas yang terkait dengan penyedia layanan untuk meningkatkan atau merawat nilai suatu produk, seperti instalasi, perbaikan, pelatihan, suplai bahan, perawatan dan perbaikan bimbingan teknis.

2. ***Secondary activities***, (*staff* atau fungsi *overhead*) merupakan aktivitas pendukung yang membantu aktivitas utama. *Secondary activities* melibatkan beberapa bagian/fungsi, antara lain :

a. ***Firm infrastructure***, merupakan aktivitas, biaya dan aset yang berhubungan dengan menejemen umum, *accuanting*, keuangan, keamanan dan keselamatan sistem informasi serta fungsi lainnya.

b. ***Human Resources Management***, terdiri dari aktivitas yang terlibat seperti penerimaan, dengar pendapat, pelatihan, pengembangan, kompensasi untuk semua tipe personil, dan mengembangkan tingkat keahlian pekerja.

c. ***Research, Technology, and Sytem Development***, aktivitas yang terkait dengan biaya yang berhubungan dengan produk, perbaikan proses, perancangan peralatan, pengembangan perangkat lunak komputer, sistem telekomunikasi, kapabilitas basis data baru dan pengembangan dukungan sistem berbantuan komputer.

d. ***Procurement***, terkait dengan fungsi pembelian infut yang digunakan dalam *value chain* organisasi.

2.5 *Client/Server*

Client/server adalah arsitektur jaringan yang memisahkan *client* (biasanya aplikasi yang menggunakan *Grafical User Interface*) dengan *server*. Masing-masing *client* dapat me*request* atau informasi dari *server*. Komponen *client* juga sering disebut *front-end*, sementara komponen *server* disebut *back-end*. Komponen *client* dari aplikasi tersebut dijalankan dalam sebuah *workstation* dan menerima masukan data dari pengguna. Komponen *client* tersebut akan menyiapkan data yang dimasukkan oleh pengguna. Komponen *client* tersebut akan menyiapkan data yang dimasukkan oleh pengguna dengan menggunakan teknologi pemrosesan tertentu dan mengirimkannya kepada komponen *server* yang dijalankan di atas mesin *server*, umumnya dalam bentuk *request* terhadap beberapa layanan yang dimiliki oleh *server*.

Komponen *server* akan menerima *request* dari *client*, dan langsung memprosesnya dan mengembalikan hasil pemrosesan tersebut kepada *client*. *Client* pun menerima informasi hasil pemrosesan data yang dilakukan *server* dan

menampilkannya kepada pengguna, dengan menggunakan aplikasi berinteraksi dengan pengguna.

Sistem *client/server* didefinisikan sebagai sistem terdistribusi, tetapi ada beberapa perbedaan karakteristik yaitu :

1. ***Service* (layanan)**
   - Hubungan antara proses yang berjalan pada mesin yang berbeda
   - Pemisahan fungsi berdasarkan ide layanannya
   - *Server* sebagai *provider*, *client* sebagai konsumen
2. ***Sharing resources* (sumber daya)**:

   Server bisa melayani beberapa *client* pada waktu yang sama, dan meregulasi akses bersama untuk *share* sumber daya dalam menjamin konsistensinya.
3. ***Asymmetrical protocol* (protokol yang tidak simetris)**:

   Many-to-one relationship antara *client* dan *server*. *Client* selalu menginisiasikan dialog melalui layanan permintaan, dan *server* menunggu secara pasif *request* dari *client*.
4. **Transparansi lokasi**:

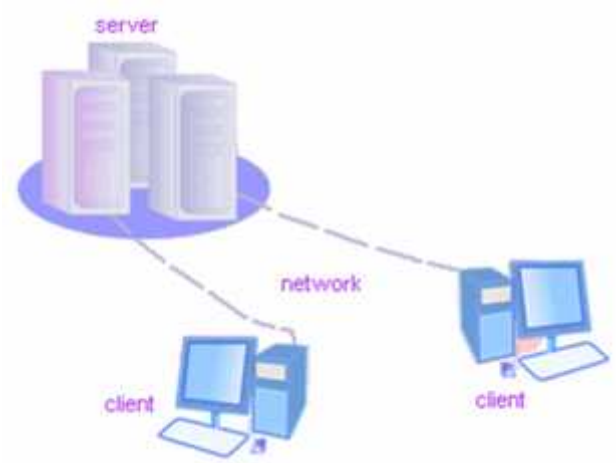
   Proses yang dilakukan *server* boleh terletak pada mesin yang sama atau pada mesin yang berbeda melalui jaringan. Lokasi *server* harus mudah diakses dari *client*.
5. ***Mix-and-Match*** :

   Perbedaan *server client platforms*
6. **Pesan berbasiskan komunikasi**;

Interaksi *server* dan *client* melalui pengiriman pesan yang menyertakan permintaan dan jawaban.

7. Pemisahan interface dan implementasi:

Server bisa di*upgrade* tanpa mempengaruhi *client* selama *interface* pesan yang diterbitkan tidak berubah.



Gambar 2.4 *Client/server*

Ada beberapa jenis *server* sebagai berikut :

1. *File Servers*

- File *server* vendors mengklaim bahwa mereka pertama menemukan istilah *client-server*.
- Untuk sharing file melalui jaringan

2. *Database Servers*

- *Client* mengirimkan SQL *requests* sebagai pesan pada database *server*, selanjutnya hasil perintah SQL dikembalikan.
- *Server* menggunakan kekuatan proses yang diinginkan untuk menemukan data yang diminta dan kemudian semua *record* dikembalikan pada *client*.

3. *Transaction Servers* (Transaksi Server)

- *Client* meminta *remote procedures* yang terletak pada *server* dengan sebuah SQL database *engine*.
- *Remote procedures* ini mengeksekusi sebuah grup dari SQL *statement*
- Hanya satu permintaan / jawaban yang dibutuhkan untuk melakukan transaksi

4. *Groupware Servers*

- Dikenal sebagai *Computer-supported cooperative working*
- Manajemen semi-struktur informasi seperti teks, image, bulletin boards dan aliaran kerja
- Data diatur sebagai dokumen

5. *Object Application Servers*

- Aplikasi *client/server* ditulis sebagai satu set objek komunikasi
- *Client objects* berkomunikasi dengan *server objects* melalui *Object Request Broker* (ORB)
- *Client* meminta sebuah method pada *remote object*

6. *Web Application Servers* (Aplikasi Web Servers)

- *World Wide Web* adalah aplikasi *client server* yang pertama yang digunakan untuk web.
- *Client dan servers* berkomunikasi menggunakan RPC seperti protokol yang disebut HTTP.

Dalam konteks database, *client* mengatur *interface* berfungsi sebagai *workstation* tempat menjalankan aplikasi basis data. *Client* menerima permintaan pemakai, memeriksa sintaks dan generate kebutuhan basis data dalam SQL atau bahasa yang lain. Kemudian meneruskan pesan ke *server*, menunggu response dan bentuk response untuk pemakai akhir. *Server* menerima dan memproses permintaan basis data kemudian mengembalikan hasil ke *client*.

Proses-proses ini melibatkan pemeriksaan autorisasi, jaminan integritas, pemeliharaan *data dictionary* dan mengerjakan *query* serta proses update. Selain itu juga menyediakan kontrol terhadap *concurrency* dan *recovery*.

Ada beberapa keuntungan jenis arsitektur ini adalah :

a. Akses basis data yang besar
b. Menaikkan kinerja
c. Jika *client* dan *server* diletakkan pada komputer yang berbeda kemudian CPU yang berbeda dapat memproses aplikasi secara paralel. Hal ini mempermudah merubah mesin *server* jika hanya memproses basis data.
d. Biaya untuk *hardware* dapat dikurangi
e. Hanya *server* yang membutuhkan *storage* dan kekuatan proses yang cukup untuk menyimpan dan mengatur basis data
f. Biaya komunikasi berkurang

g. Aplikasi menyelesaikan bagian operasi pada *client* dan mengirimkan hanya bagian yang dibutuhkan untuk akses basis data melewati jaringan, menghasilkan data yang sedikit yang akan dikirim melewati jaringan
h. Meningkatkan kekonsistenan
i. *Server* dapat menangani pemeriksaan *integrity* sehingga batasan perlu didefinisikan dan validasi hanya di satu tempat, aplikasi program mengerjakan pemeriksaan sendiri
j. Map ke arsitektur *open-system* dengan sangat alami

2.6 *Class Diagram*

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

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

Class memiliki tiga area pokok :

1. Nama

2. Atribut

3. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :

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

Hubungan Antar Class

a. Asosiasi, yaitu hubungan statis antar *class*. Umumnya menggambarkan class yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi class lain. Panah *navigability* menunjukkan arah *query* antar *class*.
b. Agregasi, yaitu hubungan yang menyatakan bagian ("terdiri atas..").
c. Pewarisan, yaitu hubungan hirarkis antar *class*. *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metoda *class* asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari class yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.
d. Hubungan dinamis, yaitu rangkaian pesan (*message*) yang di-*passing* dari satu *class* kepada *class* lain.