BAB II
LANDASAN TEORI

2.1. Konsep dasar analisis sistem

Analisis sistem dapat didefinisikan sebagai berikut:

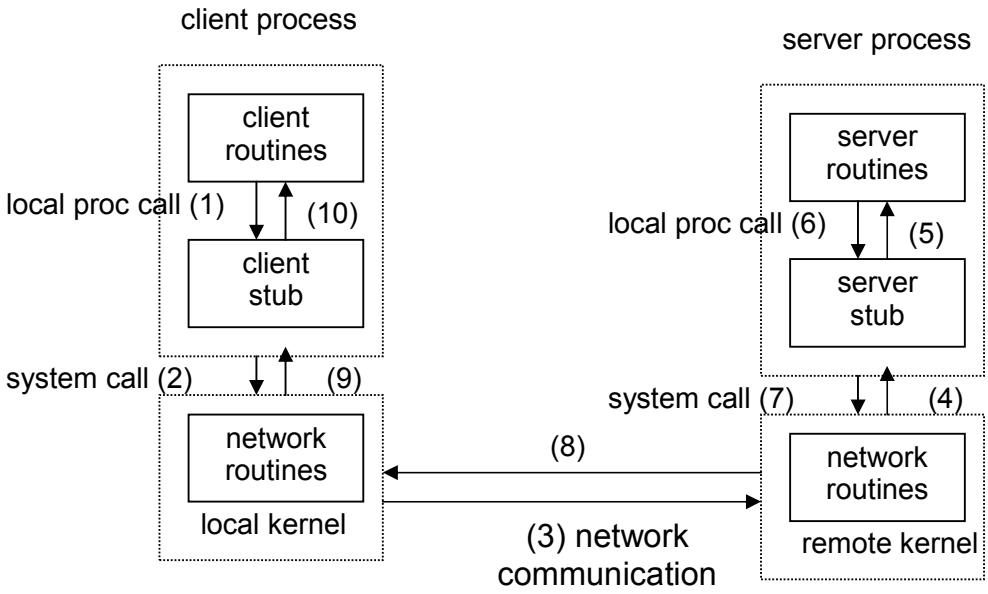
"*Analisis sistem adalah suatu penguraian dari sistem informasi ayng utuh ke dalam bagian – bagian komponennya dengan maksud untuk mengidentifikasikan danmengeveluasi permasalahan- permasalahan, kesempatan – kesempatan, hambatan – hambatan yang terjadi dan kebutuhan – kebutuhan yang diharapkan sehungga dapat diusulkan perbaikan – perbaikannya".* [8].

Analisis sistem bukan mempelajari proses mana yang harus ditangani oleh computer dan mana yang dikerjakan secara manual, tetapi penekanan dilakukan pada pemahaman secara detail dari suatu situasi. Menentukan kebutuhan apa yang harus dibuat.

2.2. *Remote Procedure Call* (RPC)

Adalah teknologi yang memungkinkan suatu prosedur atau subrutin yang ada pada suatu program dieksekusi pada komputer lain dalam suatu jaringan. RPC adalah suatu paradigma yang umum dalam implementasi *client/server*. RPC diinisialisasi oleh *client* dengan mengirim pesan *request* ke *remote* komputer untuk mengeksekusi prosedur tertentu. Respon dikembalikan kepada *client* sehingga aplikasi *client* dapat melanjutkan proses. Tujuan dari RPC ini adalah

untuk memudahkan pengembangan aplikasi, karena *client* dan *server* tidak perlu memikirkan detil jaringan seperti *socket* dan lain – lain.



Gambar 2.1. Model RPC
Sumber: Betty Purwandari, Universitas Indonesia, 2003

Langkah – langkah RPC :

1. Client memanggil lokal procedure, yang disebut *client stub*. Bagi *client, client stub* seolah-olah *server procedure* yang dipanggil. *Stub* melakukan *marshalling*: argumen untuk *remote procedure* diatur dalam paket dengan mengikuti suatu format standard, menjadi sebuah network message.
2. Melalui pemanggilan *system call* di lokal kernel, *client stub* mengirim *network message* ke *remote system*.
3. *Network message* ditransfer ke *remote system*, menggunakan *connection-oriented* maupun *connectionless protocol*.
4. *Server stub* menunggu *request* dari *client*, melakukan *unmarshalling* argumen yang ada di *network message*, dan melakukan konversi jika diperlukan.

5. *Server stub* menjalankan *local procedure call* yang memanggil *server procedure* yang sebenarnya.
6. Setelah *server procedure* selesai, *return values* dikembalikan ke *server stub*.
7. *Server stub* melakukan konversi (jika diperlukan), kemudian melakukan *marshalling* menjadi *network message,* dan mengirimkannya kembali ke *client stub.*
8. *Network message* ditransfer kembali ke *client stub*.
9. *Client stub* membaca *network message* dari *local kernel*.
10. *Client stub* mengkonversi r*eturn values* (jika diperlukan), kemudian mengembalikannya ke *client routine*, seperti *return values* dari *local procedure*.

Prosedur komunikasi pada RPC dapat digambarkan sebagai berikut:

Protocol Name	*Message from Client*	*Message from Server*	*Message from Client*
R (request)	Request		
RR (request- reply)	Request	Reply	
RRA (request-reply-acknowledge reply	Request	Reply	Acknowledge Reply

Gambar 2.2. Model RPC
Sumber: Betty Purwandari, Universitas Indonesia, 2003

1. R protokol:
   - Tidak ada *return value* dari prosedur di *server*.

- *Client* tidak memerlukan konfirmasi dari *server*, bahwa prosedur tersebut telah dilaksanakan.

2. RR protokol:
   - Paling banyak digunakan pada pertukaran pesan antara *client-server*.
   - Tidak perlu *acknowledge* secara eksplisit, karena jawaban dari *server* (*reply*) dapat dianggap sebagai *acknowledge* atas permintaan (*request*) dari *client*.
3. RRA protokol:
   - *Acknowledge reply* mengandung requestId & bersifat kumulatif, sehingga *server* dapat menghapus *entry* di *history* yg requestId-nya <= requestId yang ada di *acknowledge reply*.
   - Pengiriman pesan *acknowledge reply* tidak perlu memblok *client*.

Penanganan kesalahan:

- *Server crash*:
  - Dideteksi dengan mekanisme *timeout* oleh *client stub*.
- *Long-running server*:
  - Kanal kedua ke *remote server*, yang dapat digunakan oleh klien utk menghentikan *long-running server*.
- *Orphan server*:
  - Proses klien mati setelah memanggil RPC, tetapi sebelum memperoleh hasil dari *server*.
  - Bagaimana server dapat mendeteksi hal tsb?

2.3. Jaringan Komputer

Di akhir tahun 60-an, ARPA (*Advance Research Projects Agency*) pada Departemen Pertahanan AS mulai mendanai penelitian untuk melindungi jaringan – jaringan strategis dari serangan nuklir yang bisa menghancurkan sebagian besar jaringan. Ide dasarnya adalah menciptakan jaringan dari jaringan, atau sebuah *internetwork*, dengan banyak *link* antara setiap jaringan. Jika ada kegagalan pada *link* antara dua jaringan tidak akan merusak *internetwork* yang lainnya.

2.3.1. Internet

Adalah sebuah jaringan komputer yang terdiri dari berbagai macam ukuran jaringan komputer di seluruh dunia mulai dari PC, jaringan – jaringan lokal berskala kecil, hingga jaringan utama yang menjadi tulang punggung (*back bone)* internet. Semua perlengkapan jaringan dapat saling berkomunikasi lewat suatu aturan umum yang disebut protokol. Konsep internasional mengenai komunikasi dalam jaringan ini dituangkan dalam sebuah model yang disebut 7 *layer* OSI (*open system interconnection)*. Dalam internet, semua protokol yang ada didokumentasikan dalam RFC (*Request for Comment)*.

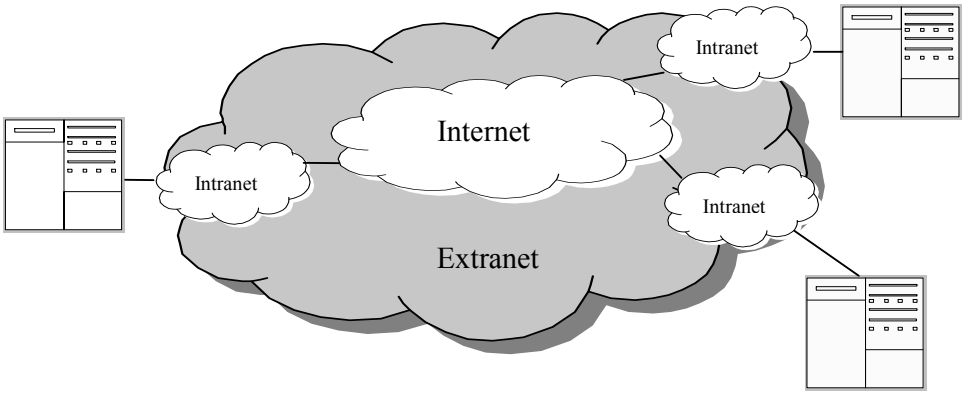
2.3.2. Intranet

Intranet sebenarnya adalah versi internet yang lebih kecil. Intranet beroperasi sebagai jaringan lokal. Awan intrernet terpisah dari awan intranet hanya dengan gateway; awan jaringan menyederhanakan representasi dari

internetwork, karena mereka menunjukan bahwa ada hubungan yang terdapat di dalam awan tersebut, tetapi strukturnya yang tepat dari awan tersebut adalah *immaterial* (tidak penting). Sebuah intranet mungkin terhubung dengan internet melalui gateway biasa, tetapi biasanya diantara kedua sistem digunakan *firewall* untuk menyaring data yang keluar masuk dari intranet organisasi.

2.3.3. Extranet

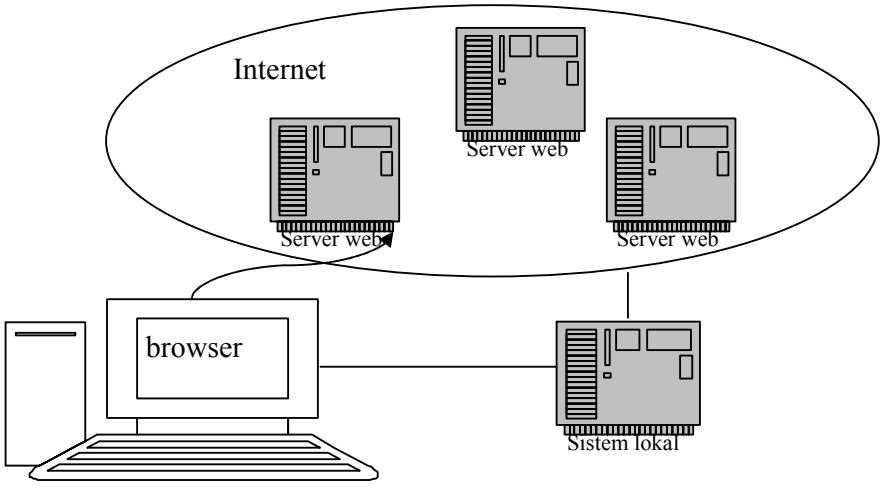
Extranet adalah jaringan pribadi yang menggunakan protokol internet, konektifitas jaringan, dan memungkinkan sistem telekomunikasi publik untuk membagi informasi perusahaan secara aman kepada pemasok, penjual, mitra, pelanggan atau kegiatan bisnis lainnya. Extranet dapat dilihat sebagai bagian dari intranet dari sebuah perusahaan yang diperluas untuk pengguna / *user* yang berada di luar perusahaan (umumnya menggunakan jaringan internet). Disebut juga dengen WAN, adalah jaringan komputer yang sangat luas yang dapat mencakup wilayah – wilayah geografis. WAN digunakan untuk menghubungkan beberapa LAN sehingga pengguna dan komputer disatu lokasi dapat berkomunikasi dengan pengguna dan komputer yang berbeda wilayah. Banyak WAN dibangun oleh perusahaan tertentu dan bersifat pribadi untuk perusahaan tersebut. Ada pula WAN yang dibangun oleh ISP, ISP menyediakan fasilitas koneksi dari LAN organisasi atau perusahaan ke internet, kebanyakan WAN dibangun dengan menggunakan *leased lines.* Selain menggunakan *leased lines* WAN juga dapat dibangun dengan metode *circuit switching* atau *packet switching*.



Gambar 2.3. Struktur Extranet
Sumber: Loshin Peter. 1998. Desain dan Implementasi Extranet. Elex Media Komputindo. Jakarta.

2.4. *World Wide Web*

World Wide Web (WWW), sering disingkat dengan web, adalah fasilitas *hypertext* untuk menampilkan data (text, gambar, bunyi, dan animasi – atau multimedia). Apa yang membuat web sangat handal adalah ide dari *hypertext*, yaitu data yang mempunyai hubungan ke data lain.



Gambar 2.4. World Wide Web
Sumber: Sidharta Lani. 1997. Internet Explorer 4.0. &
Outllook Express. Elex Media Komputindo. Jakarta.

2.4.1. Web Site

Web site awalnya merupakan suatu layanan sajian informasi yang menggunakan konsep *hyperlink*, yang memudahkan pengguna. Informasi yang disajikan menggunakan konsep multimedia (teks, gambar, animasi, suara, atau film.

2.4.2. Web Server

Web server adalah sistem yang digunakan untuk menyimpan dokumen – dokumen web, sistem ini akan melayani permintaan dokumen web dari *client*. *Web Server* merupakan program model *client/server* yang melayani dokumen – dokumen yang berbentuk *web browser* melalui sebuah protokol HTTP. Dalam dunia *web*, perangkat lunak *client*, yaitu *web browser* mempunyai tugas yang

sama yaitu menterjemahkan informasi yang diterima dari *web server* dan menampilkannya pada layer sistem pengguna. Oleh karena HTTP memungkinkan *web server* mengirimkan beragam data, seperti teks atau gambar, dan *web browser* harus bisa mengenali berbagai macam data yang akan diterimanya, dan selanjutnya menampilkannya dengan benar, teks harus ditampilkan sebagai teks dan gambar harus ditampilkan sebagai gambar.

2.4.3. Web Browser

Adalah program aplikasi yang menyediakan cara bagaimana *client* melihat dan berinteraksi dengan *web server* melalui *web.* Perkembangan serta semakin populernya *Graphical User Interface* (GUI) membuat banyak orang untuk membuat program *browser* yang menarik serta mudah digunakan. *Browser-browser web* yang modern dilengkapi dengan fasilitas – fasilitas yang mendukung tampilan multimedia berupa suara, animasi tiga dimensi, bahkan *video.* Program *browser* yang terkenal saat ini adalah *Netscape Navigator* dan *Microsoft Internet Explorer.*

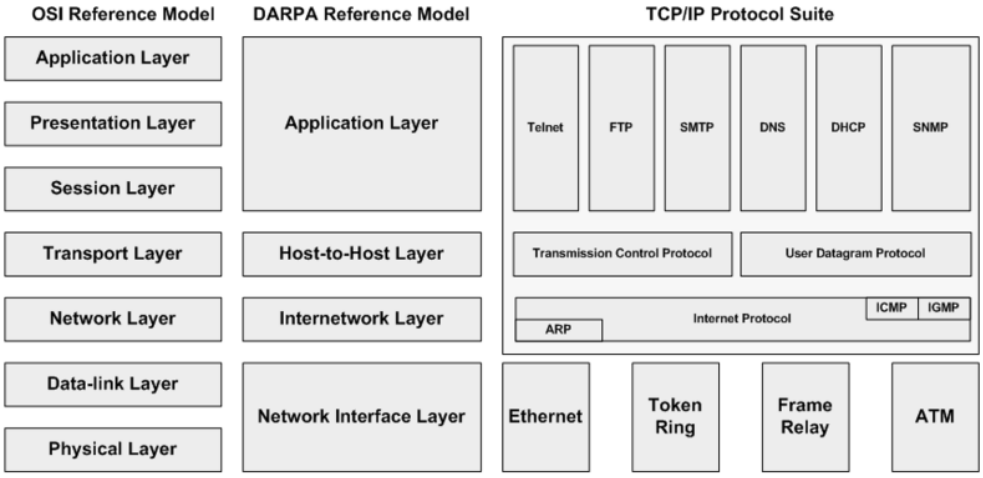
2.5. *Transmission Control Protocol/Internet Protocol* (TCP/IP)

Adalah aturan dalam komunikasi data dimana aturan tersebut harus diikuti oleh perusahaan-perusahaan dan produk-produk *software* agar produk mereka bisa *compatible* satu dengan yang lainnya. Suatu protokol menentukan bagaimana suatu *software* berkomunikasi dengan *software* lainnya, juga

menentukan bagaimana setiap bagian dari keseluruhan paket mengatur perjalanan informasinya.

Internet, dari awal memang sudah dibuat dengan menggunakan TCP/IP pada tingkat protokolnya yang memungkinkan sistem apapun yang terhubung kedalamnya bisa berkomunikasi dengan sistem lain tanpa memperdulikan bagaimana sistem masing-masing bekerja. Semua bagian didalam keluarga TCP/IP memiliki tugas tersendiri, misalnya mengirim *e-mail*, transfer *file*, menyediakan layanan *login* jarak jauh dan menangani informasi *routing* jaringan.

Protokol TCP bertanggung jawab memecah informasi ke dalam beberapa paket, sedangkan IP bertanggung jawab mengirimkan paket-paket tersebut sesuai tujuannya, kemudian TCP bertugas menyatukan kembali paket-paket itu sesuai dengan urutan yang telah ditentukan. Semua system yang langsung terhubung ke internet menjalankan TCP/IP, yang sering disebut sebagai *internet protocol suit*. Protokol komunikasi TCP/IP adalah satu – satunya persamaanyang dimiliki oleh sistem – sistem yang tehubung ke internet. Protokol TCP/IP selalu berevolusi seiring dengan waktu, mengingat semakin banyaknya kebutuhan terhadap jaringan komputer dan Internet. Pengembangan ini dilakukan oleh beberapa badan, seperti halnya Internet Society (ISOC), Internet Architecture Board (IAB), dan Internet Engineering Task Force (IETF). Macam-macam protokol yang berjalan di atas TCP/IP, skema pengalamatan, dan konsep TCP/IP didefinisikan dalam dokumen yang disebut sebagai Request for Comments (RFC) yang dikeluarkan oleh IETF.



Gambar 2.5. Arsitektur TCP/IP diperbandingkan dengan DARPA Reference Model dan OSI Reference Model
Sumber: wikipedia.org

Arsitektur TCP/IP tidaklah berbasis model referensi tujuh lapis OSI, tetapi menggunakan model referensi DARPA. Seperti diperlihatkan dalam diagram, TCP/IP merngimplemenasikan arsitektur berlapis yang terdiri atas empat lapis. Empat lapis ini, dapat dipetakan (meski tidak secara langsung) terhadap model referensi OSI. Empat lapis ini, kadang-kadang disebut sebagai *DARPA Model*, *Internet Model*, atau *DoD Model*, mengingat TCP/IP merupakan protokol yang awalnya dikembangkan dari proyek ARPANET yang dimulai oleh Departemen Pertahanan Amerika Serikat. Setiap lapisan yang dimiliki oleh kumpulan protokol (protocol suite) TCP/IP diasosiasikan dengan protokolnya masing-masing. Protokol utama dalam protokol TCP/IP adalah sebagai berikut:

Protokol lapisan aplikasi: bertanggung jawab untuk menyediakan akses kepada aplikasi terhadap layanan jaringan TCP/IP. Protokol ini mencakup protokol *Dynamic Host Configuration Protocol* (DHCP), *Domain Name System*

(DNS), Hypertext Transfer Protocol (HTTP), *File Transfer Protocol* (FTP), *Telnet*, *Simple Mail Transfer Protocol* (SMTP), *Simple Network Management Protocol* (SNMP), dan masih banyak protokol lainnya. Dalam beberapa implementasi *stack protokol*, seperti halnya Microsoft TCP/IP, protokol-protokol lapisan aplikasi berinteraksi dengan menggunakan antarmuka Windows Sockets (Winsock) atau NetBIOS over TCP/IP (NetBT).

Protokol lapisan antar-*host*: berguna untuk membuat komunikasi menggunakan sesi koneksi yang bersifat *connection-oriented* atau *broadcast* yang bersifat *connectionless*. Protokol dalam lapisan ini adalah Transmission Control Protocol (TCP) dan User Datagram Protocol (UDP).

Protokol lapisan *internetwork*: bertanggung jawab untuk melakukan pemetaan (*routing*) dan enkapsulasi paket-paket data jaringan menjadi paket-paket IP. Protokol yang bekerja dalam lapisan ini adalah *Internet Protocol* (IP), *Address Resolution Protocol* (ARP), Internet Control Message Protocol (ICMP), dan Internet Group Management Protocol (IGMP).

Protokol lapisan antarmuka jaringan: bertanggung jawab untuk meletakkan frame-frame jaringan di atas media jaringan yang digunakan. TCP/IP dapat bekerja dengan banyak teknologi transport, mulai dari teknologi transport dalam LAN (seperti halnya Ethernet dan Token Ring), MAN dan WAN (seperti halnya dial-up modem yang berjalan di atas Public Switched Telephone Network (PSTN), *Integrated Services Digital Network* (ISDN), serta *Asynchronous Transfer Mode* (ATM)).

2.6. HTTP (*Hypertext Transfer Protocol*)

HTTP adalah protokol untuk pertukaran *file* (teks, dambar, suara, video dan format lainnya) melalui situs *web*. HTTP bekerja diatas *Transmision Control Protoco* (TCP) yang menjamin sampainya data ditujuan dalam urutan yang benar. Bila satu kesalahan terjadi selama proses pengiriman, pihak pengirim akan mendapatkan pemberitahuan bahwa terjadi kesalahan. Karenanya *server* dan *client* tidak harus menyediakan mekanisme untuk memeriksa kesalahan transmisi data, namun demikian HTTP tidak memiliki apa yang disebut *session*, seperti halnya FTP, yang menjaga hubungan *server* dan *client* secara konsisten. Setiap halaman *web* yang dikirim akan melibatkan suatu proses penyambungan antara *client* dan *server*, baru kemudian melakukan transfer data, selesai transfer maka koneksi *server* dengan *client* akan diputus, sifat ini membuat HTTP sering disebut dengan istilah protokol *hit-and-run.*

Suatu halam *web* seringkali berisi file gambar, atau beberapa *file-file* lain. HTTP memaksa *server* untuk menjalin hubungan baru setiap kali hendak mengirim sebuah file. Ini tidak menguntungkan dan tidak efisien, mengingat proses hubungan putus-sambung semacam ini mengakibatkan beban bagi jaringan.

Standar baru protokol HTTP, yaitu HTTP/1.1 yang baru-baru ini diperkenalkan, dirancang untuk mengatasi masalah diatas. Web diarahkan kepenggunaan *persistent connection* (sambungan yang terus-menerus) secara lebih efisien. Dalam HTTP/1.1, *server* tidak akan memutuskan hubungan dengan *client* pada akhir transfer dokumen. Hubungan tetap dibuka untuk melayani jika

ada *request* lagi dalam waktu singkat. Hubungan baru akan diputus bila setelah melewati batas waktu tertentu *client* tidak mengirim *request* lagi.

2.7. Aplikasi Extranet

Semua aplikasi yang berjalan melalui sebuah *internetwork* dan menyatukan informasi dari sumber – sumber yang melewati batasan organisasi bisa dianggap sebagai aplikasi extranet. Hanya dengan meningkatkan intranet secara fisik tidak cukup untuk membuat sebuah extranet, walau hal itu menyediakan infrastruktur dimana aplikasi extranet bisa berjalan.

Desktop Network

Saat ini, Sistem *desktop* yang umum terdiri dari sebagian atau semua sarana dan aplikasi hubungan berikut ini:

a. Akses ke sumber jaringan lokal (LAN)
b. Akses ke intranet organisasi
c. Akses ke internet
d. *Client* GUI untuk sistem dan data *network*
e. Pelayanan e-mail organisasi
f. Pelayanan e-mail internet
g. Pelayanan informasi pribadi

Mengelola informasi yang memasuki *desktop* dari dan ke semua sumber ini (dan lebih) serta yang terpendting menjaga item-item yang penting tanpa kehilangan jejak akan informasi yang kurang penting kini semakin sulit. Masalahnya adalah

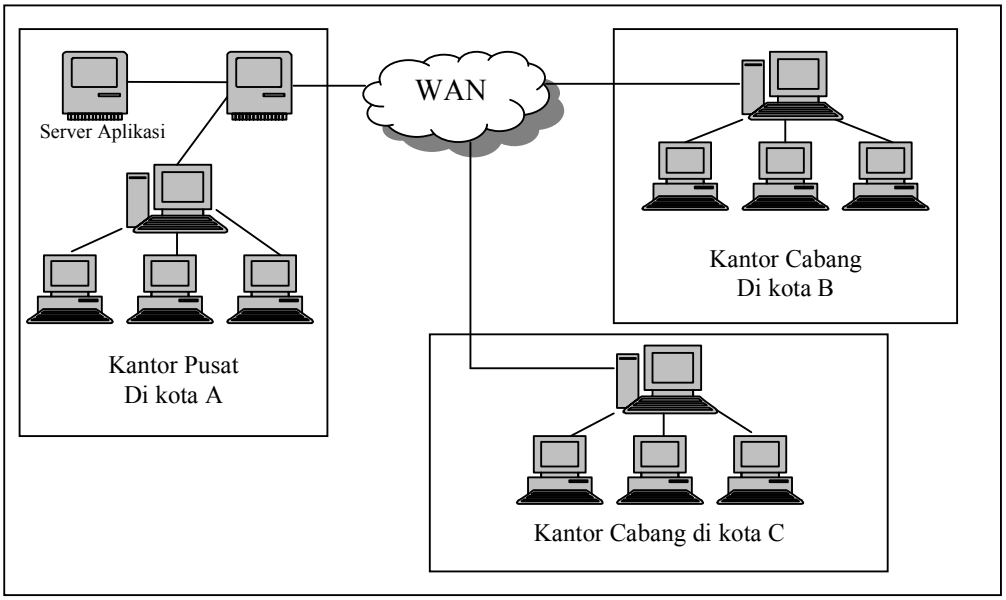
bahwa kita memiliki terlalu banyak sumber informasi dan terlalu sedikit aplikasi yang mampu mengkonsolidasi begitu banyak data dalam satu tempat.

Selama tahun 1997 *Netscape* dan *Microsoft* mengeluarkan produk yang akan mampu mengkonsolidasi informasi dari semua sumber yang terhubung pada *desktop*. Komponen objek terdistribusi juga melakukan pemunculannya. IBM dan *vendor* lain memperkenalkan sarana komponen yang penting untuk perdagangan internet, yaitu sebagai java beans dan sebagai kontrol activeX. Aplikasi yang dibangung dari java atau ActiveX akan hampir pasti sangat berguna, saat mereka bisa mengintegrasikan informasi yang terpisah-pisah dan mengakomodasi transaksi yang rumit. Untuk membuat aplikasi extranet *desktop* diperlukan suatu teknologi aplikasi terdistribusi antara lain:

a. *Distributed Component Object Model* (DCOM)
b. *Java Remote Method Invocation* (Java-RMI)
c. *Common Object Request Broker Architecture* (CORBA)
d. *Simple Object Access Protocol* (SOAP)
e. *Enterprise Java Beans* (EJB)

2.8. Struktur Aplikasi *Client/server* yang berbasis *Desktop*

Secara umum struktur jaringan komputer untuk aplikasi *Cient/Server* yang berbasis *desktop* adalah seperti gambar berikut ini:



Gambar 2.5. Strktur Apliksi *client/server* yang berbasis *Desktop*

2.9. *Hypertext Mark-up Language* (HTML)

HTML adalah file teks murni yang dapat dibuat dengan editor teks sembarangan. Dokumen ini dikenal sebagai sokumen *web pages*. Dokumen HTML merupakan dokumen yang disajikan dalam *browser*, dokumen ini umumnya berisi informasi ataupun *interface* aplikasi di dalam internet.

Dokumen HTML disusun oleh elemen-elemen. Elemen merupakan istilah bagi komponen- komponen dasar pembentukan dokumen HTML. Beberapa contoh seperti *head, body, tale, paragraph,* dan *list*. Elemen dapat berupa teks murni, atau bukakn teks, atau keduanya. Nama elemen ditunjukan dengan nama *tag*-nya. Suatu elemen di dalam dokumen HTML harus ditandai dengan penulisan tag-nya yang berpasangan. Ada beberapa elemen yang tidak mengharuskan tag-nya dituliskan secara berpasangan, elemen tersebut diantaranya adalah:

a. Paragraph dengan tag <p>

b. Ganti baris – line break dengan tag

c. Garis datar – horizontal rule dengan tag <hr>

d. List item dengan tag

Secara umum suatu elemen dalam HTML yang dinyatakan dengan tag-nya, dituliskan: <nama tag> … </nama tag>. Setiap dokumen terdiri atas tag head dan body. Elemen head berisi informasi tentang dokumen tersebut, dan elemen body berisi teks yang sebenarnya tersusun dari link, grafik, paragraph, dan elemen lainnya.