WEB

Internet and WWW

- Internet is a physical network connecting millions of computers using the same protocols for sharing/transmitting information (TCP/IP) → Jarkom
- Web or WWW is a collection of interlinked multimedia documents that are stored on the Internet and accessed using a common protocol (HTTP) → Web Prog

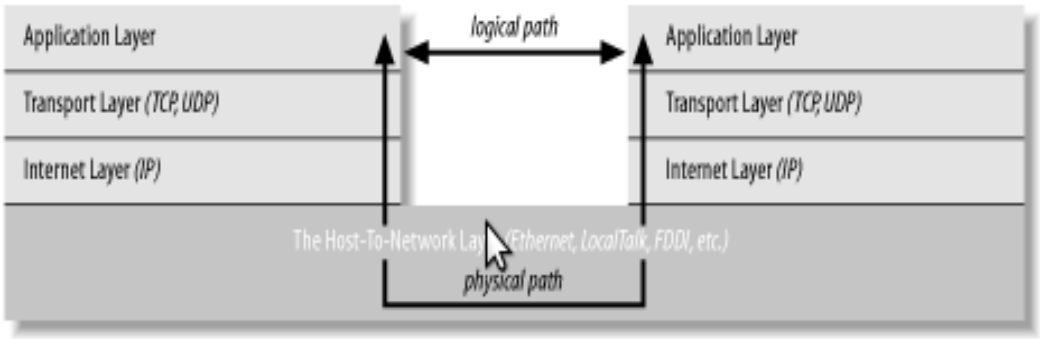
Apl Layer

- The layer that delivers data to the user is called the application layer
- Disinilah area Web Programming Course (change rapidly)
- Web 1.0

A bit about network

- Each machine on a network is called a node. Most nodes are computers, but printers, routers, bridges, gateways, dumb terminals, and drinking machines can also be nodes.
- Every network node has an address, a series of bytes that uniquely identify it

Layers of Network



- Physical Layer/ Network layer
- Internet Layer
- Transport Layer
- Application Layer

- Dikenal berberapa protocols untuk komunikasi data.These protocols define how the computers and applications running on the network communicate with one another. They allow computers running on different hardware platforms and different operating systems to share information.
- These protocols include the Simple Mail Transfer Protocol (SMTP) and Post Office Protocol (POP) for e-mail, the File Transfer Protocol (FTP) for file transfer, and the Network News Transfer Protocol (NNTP) for reading and posting to Internet newsgroups.
- Telnet, Gopher, ...

HTTP

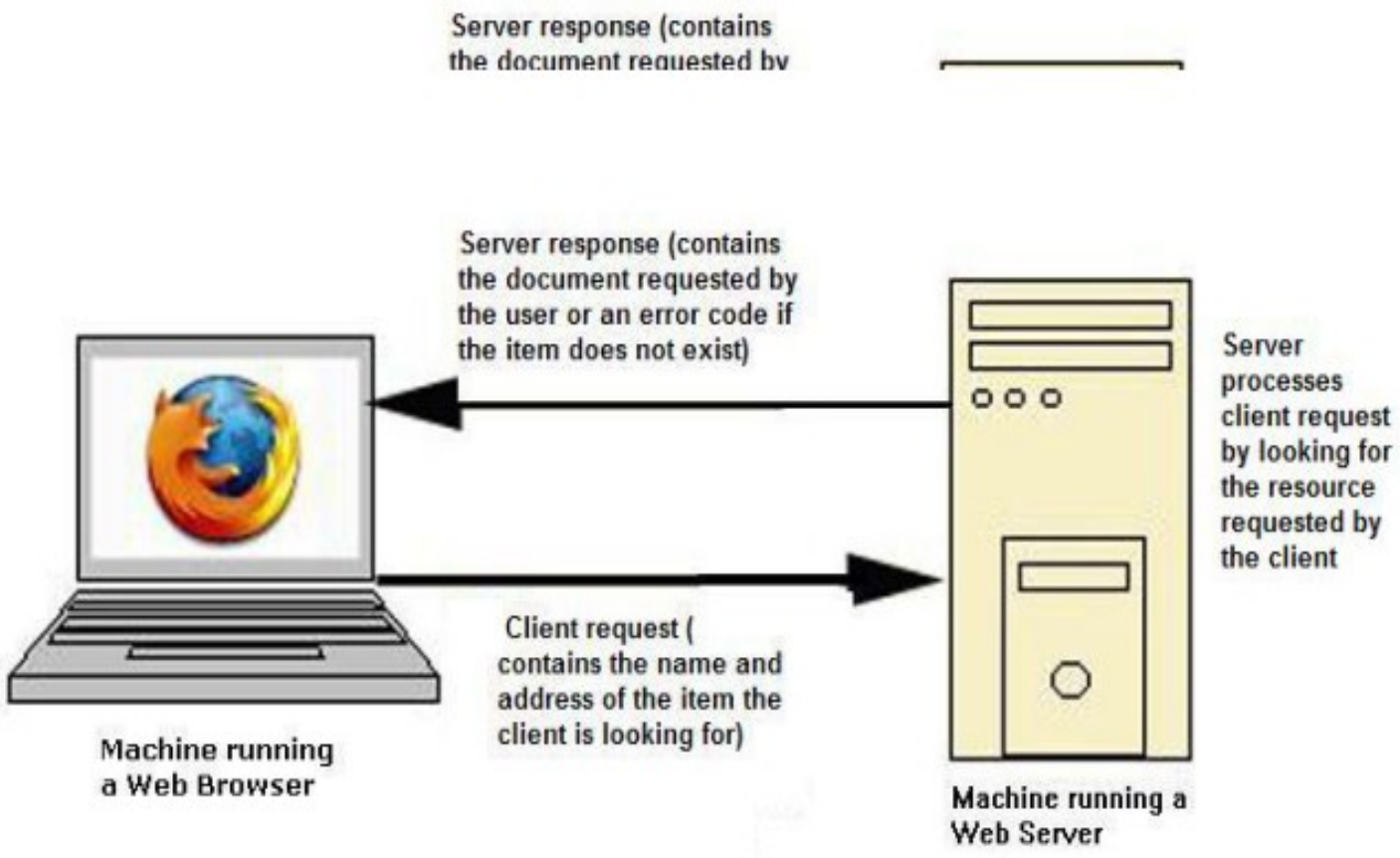
- The World Wide Web (Web for short) uses a protocol named the Hypertext Transfer Protocol (HTTP) to transfer hypermedia documents and other resources between server and client computers.
- These hypermedia documents are often referred to as Web pages and can contain links to other Web pages (hence the term hypertext)

HTTP as Client-Server

- Another very important difference is the fact that the HTTP is a stateless protocol. Users are not required to go through a logon process, which is typical of most client/server systems.
- In fact, the majority of HTTP data transfers are completely anonymous-beyond the machine address of the client, the server has no knowledge of who is retrieving data from it.
- In addition to the lack of user information, the HTTP protocol provides no mechanism for tracking how long a client may actually utilize information it has retrieved or for knowing what a client may have done before requesting a resource from the server.

HTTP as Client-Server

- HTTP is the standard protocol for communication between web browsers and web servers.
- HTTP specifies how a client and server establish a connection, how the client requests data from the server, how the server responds to that request, and finally, how the connection is closed
- HTTP connections use the TCP/IP protocol for data transfer


Server response (contains
the document requested by
Server response (contains
the document requested by
the user or an error code if
the item does not exist)
Server
processes
client request
by looking for
the resource
requested by
the client
Client request (
contains the name and
address of the item the
client is looking for)
Machine running
a Web Browser
Machine running a
Web Server

- 1. A client opens a connection with a server. Recall that any computer can act as a client, even if the computer is running a server application.
- 2. The client sends a request to the server. This request consists of a request method, a resource or service address, and possibly other header fields, and body content. These concepts are discussed in the following sections.
- 3. The server returns to the client a status line, possible header information, and (usually) an entity section.
- 4. The server closes the connection.

Client/Server – Web Server

- Accept request from client web browsers
- Respon the requests also naming and the address the data to client or
- Respon error if there is no data that be requested

HTTP Request

- GET
  - Add the data in the URL and send to the server
  - Limited data
- POST
  - Hide the data in the body's message
  - A lot data, complex

HTTP Respons

- The response begins with a response code, followed by a header full of metadata, a blank line, and the requested document or an error message
- The requested document is MIME-type (Multipurpose Internet Mail Extension), its contain the information to browser the content of the document (text, image, audio etc)

Client/Server – Web Client

- Usually → browser
- Provide the interface for user if request to server
- View the respon from server
- Send submit form and format it and can be understand in server
- Preview the respon from server

Addressing

- The Web uses a form of address known as a Universal Resource Identifier (URI) to identify data objects on servers.
- The URI for an object is independent of which protocol is used to access the data.
- An object's URI also provides no real clue as to what type of data is being identified.
- For example, a Web site dedicated to gardening may have a file named roses.htm, which is most likely an HTML document, and perhaps a file named roses.gif, which is probably a picture of roses.

- The more common form of a URI is known as a Universal Resource Locator (URL).
- Ex:
  - URI: //myserver.com/user1/default.htm
  - URL: http://myserver.com/user1/default.htm
  - URI: //ftp.myserver.com/demos/demo.zip
  - URL: ftp://ftp.myserver.com/demos/demo.zip

- URI can contain either absolute or relative addressing.
- Ex:
  - //myserver.com/home/file2.htm would specify an absolute path.
  - file3.htm would specify a relative path equivalent to //myserver.com/user1/file3.htm.
  - /file4.htm would specify a relative path equivalent to //myserver.com/file4.htm.

Static pages

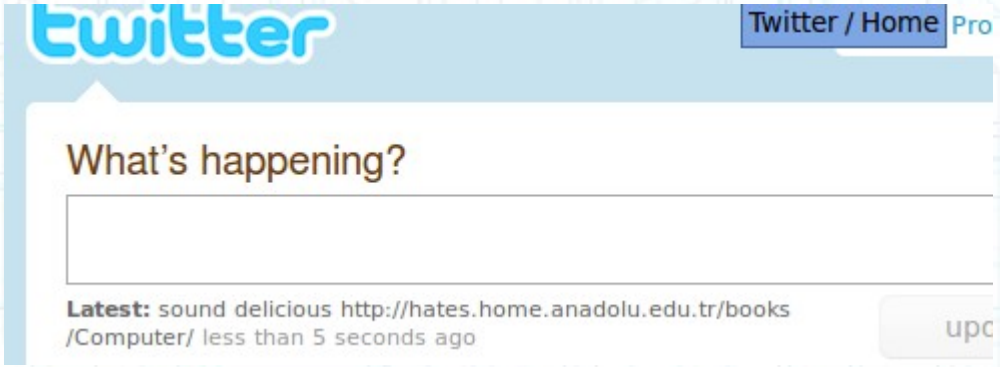
- Most Web pages are *static*
- contents (text/links/images) are the same each time it is accessed
- Ex: *online documents, most homepages*
- HyperText Markup Language (HTML) is used to specify text/image format

Ex. static pages (Web 1.0)

Dynamic pages

- As the Web moves towards online services and e-commerce, Web pages must also provide *dynamic* content
- pages must be fluid, changeable (e.g., rotating banners)
- must be able to react to the user's actions, request and process info, tailor services
- Ex: *amazon.com*

Ex. dynamic pages (Web 2.0)



Client side

- Can download program with Web page, execute on client machine
- simple, generic, but insecure
- Ex:
  - JavaScript, a scripting language for Web pages, developed by Netscape in 1995 uses a C++/Java-like syntax, so familiar to programmers, but simple
  - Java applets, can define small, special-purpose programs in Java called applets provides full expressive power of Java (but more overhead) good for more complex tasks or data heavy tasks, such as graphics

Server side

- Can store and execute program on Web server, link from Web page
- more complex, requires server privileges, but secure
- Ex:
  - CGI programming programs can be written to conform to the *Common Gateway Interface* when a Web page submits, data from the page is sent as input to the CGI program CGI program executes on server, sends its results back to browser as a Web page good if computation is large/complex or requires access to private data
  - Active Server Pages, Java Servlets, PHP, Server Side Includes vendor-specific alternatives to CGI provide many of the same capabilities but using HTML-like tags

A bit more about WEB

- The Hypertext Transfer Protocol (HTTP) is a standard that defines how a web client talks to a server and how data is transferred from the server back to the client
- HTTP can be used to transfer data in essentially any format, from TIFF pictures to Word documents to DBase files
  - (this is why some experts talk web as multimedia)

- However, far and away the most common format for data transferred over the Web and in some sense the Web's native format is the Hypertext Markup Language (HTML)
- But recently is XML, one of backbone Web 3.0