

BAB II

TINJAUAN PUSTAKA

II.1. Perancangan

Tujuan dari perancangan secara umum adalah untuk didistribusikan kepada user. Rancangan model merupakan suatu gambaran yang menjelaskan suatu bentuk atau model. Secara umum rancangan model yang diusulkan mempunyai dua bentuk yaitu *physical* model dan *logical* model. Bentuk *physical* model menunjukkan bagaimana nantinya sistem secara fisik diterapkan. Model ini biasanya digambarkan dalam bentuk diagram alir sedangkan *logical* model digambarkan dengan diagram arus data, model ini menjelaskan kepada user bagaimana nantinya fungsi-fungsi dalam sistem informasi secara logika akan bekerja.

Perancangan adalah suatu proses pemilihan dan pemikiran yang menghubungkan fakta-fakta berdasarkan asumsi-asumsi yang berkaitan dengan masa datang dengan menggambarkan dan merumuskan kegiatan-kegiatan tertentu yang diyakini diperlukan untuk mencapai tujuan-tujuan tertentu dan menguraikan bagaimana pencapaiannya. (Anggraheni Rukmana ; 2012)

Adapun langkah-langkah perancangan sistem adalah sebagai berikut :

1. Mengidentifikasi masalah (*identify*), terdiri dari :
 - a. Mengidentifikasi penyebab masalah,
 - b. Mengidentifikasi titik keputusan,
 - c. Mengidentifikasi personil kunci.

2. Memahami kerja dari sistem yang ada (*Understand*), terdiri dari :
 - a. Menentukan jenis penelitian,
 - b. Merencanakan jadwal,
 - c. Membuat penugasan penelitian,
 - d. Membuat agenda wawancara,
 - e. Mengumpulkan hasil penelitian.
3. Menganalisa sistem (*Analyze*) terdiri dari :
 - a. Menganalisa kelemahan system,
 - b. Menganalisa kebutuhan informasi pemakai.
4. Membuat laporan hasil analisa (*Report*) terdiri dari :
 - a. Pelaporan bahwa analisa telah selesai dilakukan.
 - b. Meluruskan kesalahan pengertian apa yang telah ditemui dan dianalisa.

II.2. Aplikasi

Aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Contoh utama aplikasi adalah pengolah kata, lembar kerja, memanipulasi foto dan pemutar media. Beberapa aplikasi yang digabung bersama menjadi suatu paket disebut sebagai suatu paket atau *suite* aplikasi (*application suite*). Contohnya adalah *Microsoft Office* dan *OpenOffice.org*, yang menggabungkan suatu aplikasi pengolah kata, lembar kerja dan beberapa aplikasi lainnya. Aplikasi-aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan tiap aplikasi. Sering kali, mereka memiliki

kemampuan untuk saling berinteraksi satu sama lain sehingga menguntungkan pengguna. Contohnya, suatu lembar kerja dapat dibenamkan dalam suatu dokumen pengolah kata walaupun dibuat pada aplikasi lembar kerja yang terpisah. . (Dahlan Abdullah ; 2013 : 152)

Jenis-jenis *Software* Aplikasi :

1. *Software* aplikasi hiburan, contohnya yaitu *winamp* untuk mendengarkan musik, *games* dan sebagainya untuk hiburan.
2. *Software* aplikasi pendidikan yaitu *software* yang digunakan untuk mempelajari tentang pendidikan atau pengetahuan.
3. *Software* aplikasi bisnis yaitu *software* yang digunakan untuk aplikasi bisnis
4. *Software* aplikasi khusus

II.3. *Chatting*

Chatting adalah suatu kegiatan komunikasi melalui jaringan komputer ataupun internet. Pada aplikasi *chat*, seseorang dapat menyampaikan informasi kepada orang lain melalui *text*. *Chats* selain menyenangkan juga efektif digunakan. Dengan *chatting* lewat internet kita bisa mendapat banyak teman dari penjuru dunia. Faktor kenyamanan merupakan hal penting dalam berbagi informasi. Misalnya, jika seorang atasan ingin melihat laporan hasil kerja karyawan, mengirimkan laporan tersebut melalui *e-mail* serasa tidak efektif apabila kita memerlukan komunikasi yang interaktif. Dengan *chat* dapat tercipta kegiatan komunikasi interaktif yang efektif.

Skenario normal untuk sistem aplikasi *chatting* client-server ini adalah sebagai berikut :

1. *Server* dijalankan pertama kali, pada saat ini *server* dikatakan dalam kondisi *listening* dan siap untuk menerima permintaan koneksi dari *client*.
2. *Client* siap melakukan koneksi ke *server* dengan menggunakan perintah *connect* , *server* lalu akan memproses permintaan koneksi dari *client* dengan menggunakan perintah *accept*.
3. *Server* mengirimkan balasan / notifikasi atas permintaan koneksi dari *client*, isinya adalah informasi bahwa *client* sudah terkoneksi dengan *server*. Bila notifikasi dari *server* sudah diterima maka *client* sudah terkoneksi dengan *server*, pada saat ini terjadi maka *client* dan *server* sudah bisa berkomunikasi saling mengirimkan pesan.
4. Bila ada permintaan koneksi dari *client* yang lain maka *server* akan melakukan hal yang sama pada *point* (1), (2) dan (3) diatas.
5. Masalah akan timbul pada *client* jika IP *address*-nya di blocking oleh *server*, jika ini terjadi maka *client* tidak akan bisa terkoneksi dengan *server*, yang terlihat dengan adanya notifikasi dari *server*.
6. *Server* berhak untuk meng-kick (menendang) *client* yang dikehendaki dari sistem.
7. Jika *client* ingin keluar dari sistem maka tinggal memutuskan koneksi dari *server* dengan memanggil perintah *close*.
8. *Client* sudah tidak terkoneksi ke *server*.
9. *Server* berada dalam keadaan *listening* dan siap untuk memproses permintaan koneksi dari *client* yang lain.

II.4. *Message Voice*

Message Voice adalah salah satu media komunikasi yang memberikan kemudahan pengguna untuk dapat merekam suara dalam *chatting*. *Message voice* juga membutuhkan jaringan *internet* sebagai media transmisinya. Aplikasi *Message voice* telah banyak dilakukan oleh berbagai pihak, namun masih sedikitnya pengembangan aplikasi *Message voice* dengan layanan koneksi jaringan lokal seperti halnya koneksi jaringan *Wifi*. Berbagai jenis aplikasi yang terkenal saat ini telah banyak beredar dan digunakan oleh banyak pengguna dengan layanan *internet*. Akan tetapi juga ditemukan beberapa pengguna yang tertarik menggunakan aplikasi *Message voice* pada jaringan *Wifi* karena penggunaan jaringan *Wifi* jauh lebih murah dibanding penggunaan *internet*. Melihat kebutuhan dan fungsi yang diinginkan oleh pengguna perlu pengembangan aplikasi yang dapat diimplementasikan dengan jaringan lokal untuk dapat terhubung dengan aplikasi *Message voice*.

II.5. *Android*

Android merupakan sistem operasi yang berisi *middleware* serta aplikasi-aplikasi dasar. Basis sistem operasi *android* yaitu kernel *linux* 2.6 yang telah diperbaharui untuk *mobile device*. Pengembangan aplikasi *android* menggunakan bahasa pemrograman *java*. Yang mana konsep-konsep pemrograman *java* berhubungan dengan Pemrograman Berbasis Objek (OOP)). Selain itu pula dalam pengembangan aplikasi *android* membutuhkan *software development kit* (SDK) yang disediakan *android*, SDK ini memberi jalan bagi

programmer untuk mengakses *application programming interface (API)* pada *android*. (Arzan Muharom, Rinda Cahyana, H. Bunyamin M.Kom ; 2013 : 2)

II.5.1. Komponen Dasar *Android*

Android UI framework, bersama dengan bagian lain dari *Android*, mengandalkan konsep baru disebut sebuah *intent*. Maksud adalah sebuah penggabungan ide-ide seperti pesan *windowing*, tindakan, menerbitkan-dan model berlangganan, antar-proses komunikasi, dan aplikasi pendaftar. *Android* juga memiliki dukungan luas untuk sumber daya, yang meliputi elemen akrab dan *file* seperti *string* dan *bitmap*, serta beberapa *item* yang tidak begitu akrab seperti *XMLbased* melihat definisi. *Framework* kerja ini membuat penggunaan sumber daya dengan cara baru untuk membuat penggunaannya mudah, intuitif, dan nyaman. Berikut adalah contoh di mana ID sumber daya adalah otomatis dihasilkan untuk sumber daya didefinisikan dalam *file XML*. Setiap ID otomatis dihasilkan di kelas ini sesuai untuk baik elemen dalam *file XML* atau seluruh *file* itu sendiri. Di mana pun Anda ingin menggunakan definisi tersebut XML, Anda akan menggunakan dihasilkan ID gantinya. Tipuan ini membantu banyak ketika datang ke lokalisasi. Konsep lain yang baru di *Android* adalah penyedia konten. Sebuah penyedia konten adalah abstraksi pada sumber data yang membuatnya tampak seperti emitor dan konsumen tenang jasa. *Database SQLite* mendasari membuat fasilitas penyedia konten kuat alat untuk pengembang aplikasi. (Windu Gata, Grace Gata, Nia Kusuma Wardhani ; 2012 : 66)

II.5.2. *Android SDK*

Android SDK (Software Development Kit) adalah *tools API (Application Programming Interface)* yang diperlukan untuk mengembangkan aplikasi pada *platform Android* menggunakan bahasa pemrograman *Java*. Beberapa fitur-fitur *Android* yang paling penting adalah mesin *Virtual Dalvik* yang dioptimalkan untuk perangkat *mobile*, *integrated browser* berdasarkan *engine open source WebKit*, *Grafis* yang dioptimalkan dan didukung oleh *libraries grafis 2D, grafis 3D* berdasarkan spesifikasi *open ES 1.0* (Opsional akselerasi perangkat keras), kemudian *SQLite* untuk penyimpanan data (*database*). Fitur-fitur *android* lainnya termasuk media yang mendukung audio, video, dan gambar, juga ada fitur *bluetooth*, *EDGE*, *3G* dan *WiFi*, dengan fitur kamera, *GPS*, dan kompas. Selanjutnya fitur yang juga turut disediakan adalah lingkungan *Development* yang lengkap dan kaya termasuk perangkat *emulator*, *tools* untuk *debugging*, profil dan kinerja memori, dan *plugin* untuk IDE *Eclipse*. (Alicia Sinsuw, Xaverius Najoran ; 2013 : 2)

II.6. Jaringan

Pengertian jaringan komputer merupakan sekumpulan komputer serta perangkat - perangkatlain pendukung komputer yang saling terhubung dalam suatu kesatuan. Media jaringankomputer dapat melalui kabel-kabel atau tanpa kabel sehingga memungkinkan penggunajaringan komputer dapat saling melakukan pertukaran informasi seperti dokumen dan data. Dapat juga melakukan pencetakan pada printer yang sama dan bersama-sama memakai perangkat keras dan perangkat lunak yang terhubung dengan jaringan. (Choirul Muallifah;2013:2)

II.6.1. Jenis-Jenis Jaringan

Jaringan komputer adalah kombinasi perangkat keras, perangkat lunak, dan pengkabelan (*cabling*), yang memungkinkan berbagai alat komputasi berkomunikasi satu sama lain. Dalam suatu jaringan komputer, terjadi sebuah proses komunikasi antar entiti atau perangkat yang berlainan sistemnya. Entiti atau perangkat ini adalah segala komunikasi mengirim, dan, menerima.

Untuk berkomunikasi mengirim dan menerima antara dua entiti dibutuhkan pengertian di antara kedua belah pihak. Pengertian ini lah yang dikatakan sebagai protokol. Jadi protokol adalah himpunan aturan-aturan main yang mengatur komunikasi data. Protokol mendefinisikan apa yang dikomunikasikan, bagaimana dan kapan terjadinya komunikasi.

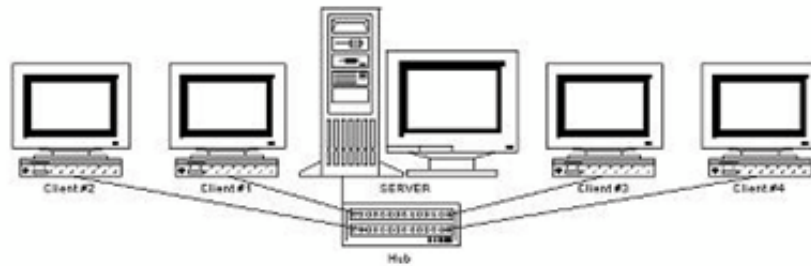
II.6.2 Jenis-Jenis Jaringan Komputer

Secara umum Jaringan Komputer dibagi atas tiga jenis yaitu Local Area Network, Wide Area Network dan Metorpolitan Area Network.

II.6.2.1 *Local Area Network* (LAN)

Local Area network (LAN), merupakan jaringan milik pribadi di dalam sebuah gedung atau kampus. Jaringan jenis ini seringkali digunakan untuk menghubungkan komputer-komputer pribadi dan *workstation* dalam kantor suatu perusahaan atau pabrik- pabrik untuk saling bertukar informasi serta memakai bersama (*share*) sumber daya (*resource*), misalnya printer. LAN seringkali menggunakan teknologi transmisi kabel tunggal. LAN tradisional beroperasi pada kecepatan mulai 10 sampai 100 Mbps (mega bit/detik) dengan *delay* rendah (*mikro*

second) dan mempunyai faktor kesalahan yang kecil. LAN modern dapat beroperasi pada kecepatan yang lebih tinggi, sampai ratusan Megabit/detik. Gambar II.1 menunjukkan sebuah jaringan LAN sederhana.



Gambar II.1 Contoh Jaringan LAN

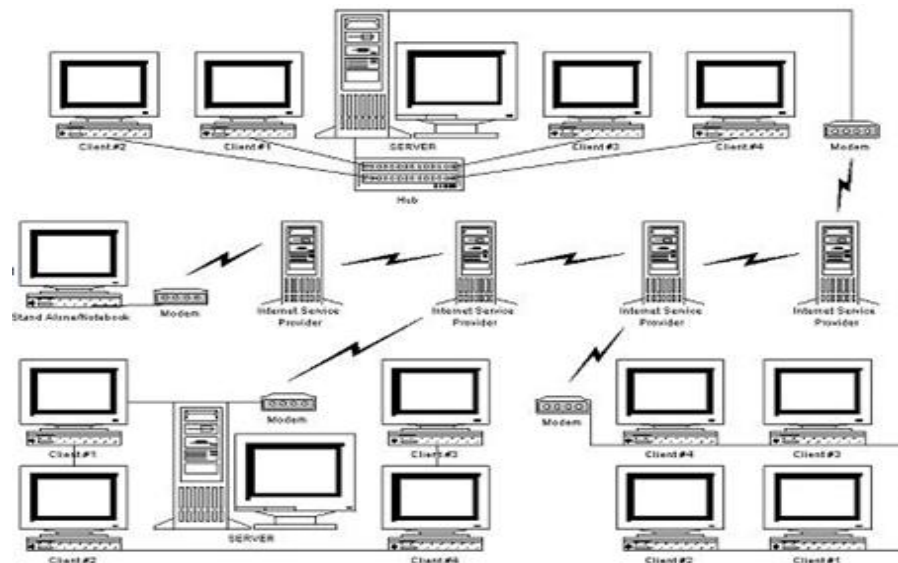
Sumber : Choirul Mualifah ; 2013

II.6.2.2 Wide Area Network (WAN)

Wide Area Network (WAN) mencakup daerah geografis yang luas, seringkali mencakup negara atau benua. WAN terdiri dari kumpulan mesin yang bertujuan untuk menjalankan program aplikasi. Mesin-mesin ini biasa disebut *host*. Didalam literature juga biasa disebut sebagai *End System*.

Host dihubungkan dengan sebuah *subnet* (bagian jaringan yang lebih kecil). Tugas subnet adalah membawa pesan dari host ke host lainnya, seperti halnya sistem telepon yang membawa isi pembicaraan dari pembicara ke pendengar. Dengan memisahkan aspek komunikasi murni sebuah jaringan dari aspek-aspek aplikasi (*host*), rancangan jaringan lengkap menjadi jauh lebih sederhana. Pada sebagian besar WAN, subnet terdiri dari dua komponen, yaitu kabel transmisi dan

element switching. Kabel transmisi (disebut juga sirkuit, channel, atau trunk) memindahkan bit-bit dari satu mesin ke mesin lainnya. Gambar II.2 menunjukkan sebuah jaringan WAN.



Gambar II.2 Jaringan WAN

Sumber : Choirul Mualifah ; 2013

II.6.2.3 Metropolitan Area Network (MAN)

Metropolitan Area Network (MAN) pada dasarnya merupakan versi LAN yang berukuran lebih besar dan biasanya memakai teknologi yang sama dengan LAN. MAN dapat mencakup kantor-kantor perusahaan yang berdekatan dan dapat dimanfaatkan untuk keperluan pribadi (perusahaan atau instansi). MAN hanya memiliki sebuah atau dua buah kabel dan tidak mempunyai elemen *switching*, yang berfungsi untuk mengatur paket melalui beberapa output kabel. Adanya elemen *switching* membuat rancangan menjadi lebih sederhana.

II.6.3 *Client-server*

Suatu model umum yang diterapkan untuk komunikasi dalam sebuah jaringan adalah model *client-server*. Konsepnya sederhana, sebuah proses/aplikasi dimana *client* melakukan permintaan (*request*) untuk suatu informasi atau mengirim sebuah perintah ke suatu aplikasi *server*.

Aplikasi *server* akan menerima permintaan dari *client*, kemudian memproses berdasarkan permintaan tersebut dan mengembalikan suatu hasil dari pemrosesan yang sudah dilakukan. Sehingga tugas utama dari *server* adalah melakukan *listen* untuk suatu koneksi, sedangkan *client* mencoba koneksi ke *server*. Setelah koneksi terjadi, hubungan pertukaran data antara *client* dan *server* diwakili suatu *stream* input dan output. Beberapa contoh pengaplikasian server adalah:

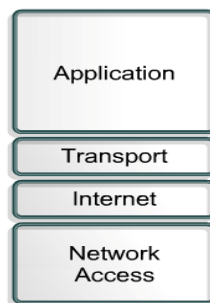
- a) Web server adalah node yang menyediakan layanan untuk akses halaman web.
- b) DNS server adalah node yang menyediakan layanan untuk menerjemahkan nomor IP menjadi alamat domain.
- c) Mail server adalah node yang menyediakan tempat penyimpanan (*storage*) *e-mail*.
- d) Proxy server adalah node yang menyaring informasi yang terdapat pada jaringan, membatasi kegiatan dalam suatu jaringan seperti menyimpan *cache*. Alamat web yang pernah dikunjungi *client*. Proxy juga dapat membatasi ukuran file yang dapat di *download* oleh *client*, juga berperan sebagai *firewall* untuk mengamankan jaringannya dari serangan virus.
- e) DHCP server adalah server yang bertugas untuk membagikan alamat IP kepada *client* pada jaringan *subnet*nya secara otomatis. Mencatat semua nomor mesin

client yang meminta (*request*) IP dari server ini, memberikan hak ataupun menolak memberikan IP address kepada suatu *client*.

Pada konsep *Client-Server* ini memudahkan manajemen jaringan, karena pada jaringan ini terdapat satu node yang menjadi sumber untuk memberikan pelayanan bagi *client*. Dengan berbeda-bedanya layanan yang diberikan oleh *server* maka dengan konsep ini, *client* dapat mengakses lebih dari satu *server* bergantung kepada kebutuhan *client* itu sendiri.

II.6.4 Protokol TCP/IP

Protokol TCP/IP hanya dibuat atas empat lapisan saja: *Network Access*, *Internet*, *Transport*, dan *Application*. Protokol TCP/IP memiliki 4 lapisan seperti ditunjukkan Gambar II.3.



Gambar II.3 Model TCP/IP

Sumber : Choirul Muallifah ; 2013

Beberapa protokol yang terdapat pada lapisan *Application* adalah:

- a) File Transfer Protocol (FTP).
- b) Hypertext Transfer Protocol (HTTP).
- c) Simple Mail Transfer Protocol (SMTP).

- d) Domain Name Service (DNS).
- e) Trivial File Transfer Protocol (TFTP).

Beberapa protocol yang terdapat pada lapisan Transport adalah:

- a) Transfer Control Protocol (TCP)
- b) User Datagram Protocol (UDP).

II.7. Wifi

Wi-Fi adalah singkatan dari Wireless Fidelity yaitu seperangkat standaryang digunakan untuk komunikasi jaringan lokal tanpa kabel (Wireless Local Area Network-WLAN). Awalnya *wifi* ditujukan untuk penggunaan perangkat nirkabel dan Jaringan Area Lokal (LAN),namun saat ini lebih banyak digunakan untuk mengakses *internet*. Hal ini memungkinkan seseorang dengan komputer dengan kartu nirkabel (*wireless card*) atau *personal digital assistant* (PDA) untuk terhubung dengan internet dengan menggunakan titik akses (*hotspot*) terdekat. *Wifidirancang* berdasarkan spesifikasi IEEE 802.11. Sekarang ini ada empat variasi dari 802.11, yaitu:802.11a, 802.11b, 802.11g, and 802.11n. Spesifikasi b merupakan produk pertama *wifi*. Variasi g dan n merupakan salah satu produk yang memiliki penjualan terbanyak pada 2005.(*Sony Bahagia Sinaga ; 2012 : 47*) Spesifikasi *wifi* yaitu :

Tabel II.1. Spesifikasi Wifi

Spesifikasi	Kecepatan	Frekuensi Band	Cocok dengan
802.11b	11 Mb/s	2.4 GHz	B
802.11a	54 Mb/s	5 GHz	A
802.11g	54 Mb/s	2.4 GHz	b,g
802.11n	100 Mb/s	2.4 GHz	b,g,n

Versi *wifi* yang paling luas dalam pasaran ASsekarang ini (berdasarkan dalam IEEE 802.11b/g) beroperasi pada 2.400 MHz sampai 2.483,50 MHz. Dengan begitu mengijinkan operasi dalam 11channel (masing-masing 5 MHz), berpusat difrekuensi berikut :

1. Channel 1 - 2,412 MHz
2. Channel 2 - 2,417 MHz
3. Channel 3 - 2,422 MHz
4. Channel 4 - 2,427 MHz
5. Channel 5 - 2,432 MHz
6. Channel 6 - 2,437 MHz
7. Channel 7 - 2,442 MHz
8. Channel 8 - 2,447 MHz
9. Channel 9 - 2,452 MHz
10. Channel 10 - 2,457 MHz
11. Channel 11 - 2,462 MHz

Secara teknis operasional, *wifi* merupakan salahsatu varian teknologi komunikasi dan informasi yang bekerja pada jaringan dan perangkat WLAN(*wireless local area network*). Dengan kata lain, *wifi* adalah sertifikasi merek dagang yang diberikan pabrikan kepada perangkat telekomunikasi(*internet*) yang bekerja di jaringan WLAN dan sudah memenuhi kualitas kapasitas interoperasi yang dipersyaratkan. (Sony Bahagia Sinaga ; 2012 : 47)

Teknologi *internet* berbasis *wifi* dibuat dan dikembangkan sekelompok insinyur Amerika Serikat yang bekerja pada *Institute of Electrical and Electronics Engineers* (IEEE) berdasarkan standar teknis perangkat bernomor 802.11b, 802.11a dan 802.16. Perangkat *wifi* sebenarnya tidak hanya mampu bekerja di jaringan WLAN, tetapi juga di jaringan *Wireless Metropolitan Area Network* (WMAN). Karena perangkat dengan standar teknis 802.11b diperuntukkan bagi perangkat WLAN yang digunakan di frekuensi 2,4 GHz atau yang lazim disebut frekuensi ISM (*Industrial, Scientific dan Medical*). Sedangkan untuk perangkat yang berstandar teknis 802.11a dan 802.16 diperuntukkan bagi perangkat WMAN atau juga disebut Wi-Max, yang bekerja di sekitar pita frekuensi 5 GHz. (Sony Bahagia Sinaga ; 2012 : 47).

Hotspot adalah definisi untuk daerah yang dilayani oleh satu Access Point Wireless LAN standar 802.11a/b/g, dimana pengguna (user) dapat masuk kedalam Access Point secara bebas dan mobile menggunakan perangkat sejenis notebook, PDA atau lainnya (Deris Stiawan, *Wireless Fundamental, Instalation & Implemetations*, 2008). Sedangkan hotspot (Wi-Fi) yang akan dibahas dalam penelitian ini adalah sarana terkoneksi jaringan internet tanpa kabel, dengan

menggunakan standar wireless fidelity, namun demikian dalam menjalankan hotspot diperlukan sarana lain, seperti Notebook, laptop, PDA yang memiliki fasilitas wireless LAN. Konsep teknologi nirkabel adalah teknologi yang menghubungkan dua piranti untuk bertukar data tanpa media kabel. Data dihubungkan melalui media gelombang cahaya tertentu (seperti teknologi infra merah pada remote TV) atau gelombang radio (seperti bluetooth pada komputer dan ponsel) dengan frekuensi tertentu.

Keunggulan dan Kelemahan Jaringan Hotspot (Wi-Fi) menurut (Priyambodo 2005: 5) adalah sebagai berikut:

1. Keunggulan

- a. Biaya Pemeliharaan murah
- b. Infrastruktur berdimensi kecil
- c. Pembangunannya cepat
- d. Mudah dan murah untuk direlokasi
- e. Mendukung Portabilitas

2. Kelemahan

- a. Biaya Peralatan mahal
- b. Delay yang sangat besar
- c. Kesulitan karena masalah propagasi radio
- d. Mudah untuk terinterferensi
- e. Kapasitas jaringan kecil
- f. Keamanan/kerahasiaan data kurang terjamin

II.8. Komponen Utama Jaringan Wi-Fi

Terdapat empat komponen utama untuk membangun jaringan Wi-Fi, yaitu sebagai berikut :

1. Access Point

Komponen yang berfungsi menerima dan mengirimkan data dari adapter wireless. Access Point mengonversi sinyal frekuensi radio menjadi sinyal digital atau sebaliknya. Komponen tersebut bertugas seperti sebuah hub/switch pada jaringan ethernet. Satu Access Point berdasarkan teori dapat menampung beberapa hingga ratusan client. Walaupun demikian, Access point direkomendasikan dapat menampung maksimal 40-an client.

2. Wireless LAN Interface

Wireless LAN Interface adalah peralatan yang dipasang di Desktop/mobile PC, peralatan yang dikembangkan secara massal adalah dalam bentuk PCMCIA (Personal Computer Memory Card International Association) card, PCI card maupun melalui port USB (Universal Serial Bus).

3. Mobile/ Desktop PC

Komponen perangkat akses untuk klien, mobile PC pada umumnya sudah terpasang port PCMCIA (Personal Computer Memory Card International Association), sedangkan Desktop PC harus ditambahkan PCI (Peripheral Component Interconnect) Card, serta USB (Universal Serial Bus) Adapter.

4. Antena external (optional)

Antena external ini digunakan untuk memperkuat daya pancar. Antena dapat dirakit sendiri oleh user. contoh : wajan bolik, antena kaleng, dll.

II.9. Data Flow Diagram (DFD)

Data Flow Diagram (DFD) adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi. DFD ini sering disebut juga dengan nama Bubble chart, Bubble diagram, model proses, diagram alur kerja, atau model fungsi.

DFD ini adalah salah satu alat pembuatan model yang sering digunakan, khususnya bila fungsi-fungsi sistem merupakan bagian yang lebih penting dan kompleks dari pada data yang dimanipulasi oleh sistem. Dengan kata lain, DFD adalah alat pembuatan model yang memberikan penekanan hanya pada fungsi sistem.

DFD ini merupakan alat perancangan system yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan ,sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program. (Dahlan Abdullah ; 2013 : 154)

II.10. UML (Unified Modelling Language)

Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML kita dapat membuat model untuk semua jenis

aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan peranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C. 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)*. Sejarah UML sendiri cukup panjang. Sampai era tahun 1990 seperti kita ketahui puluhan metodologi pemodelan berorientasi objek telah bermunculan di dunia. Diantaranya adalah: *metodologi booch*, *metodologi coad*, *metodologi OOSE*, *metodologi OMT*, *metodologi shlaer-mellor*, *metodologi wirfs-brock*, dsb. Masa itu terkenal dengan masa perang metodologi (*method war*) dalam pendesainan berorientasi objek. Masing-masing metodologi membawa notasi sendiri-sendiri, yang mengakibatkan timbul masalah baru apabila kita bekerjasama dengan group/perusahaan lain yang menggunakan metodologi yang berlainan. Dimulai pada bulan Oktober 1994 *Booch*, *Rumbaugh* dan *Jacobson*, yang merupakan tiga tokoh yang boleh dikata metodologinya banyak digunakan memelopori usaha untuk penyatuan metodologi

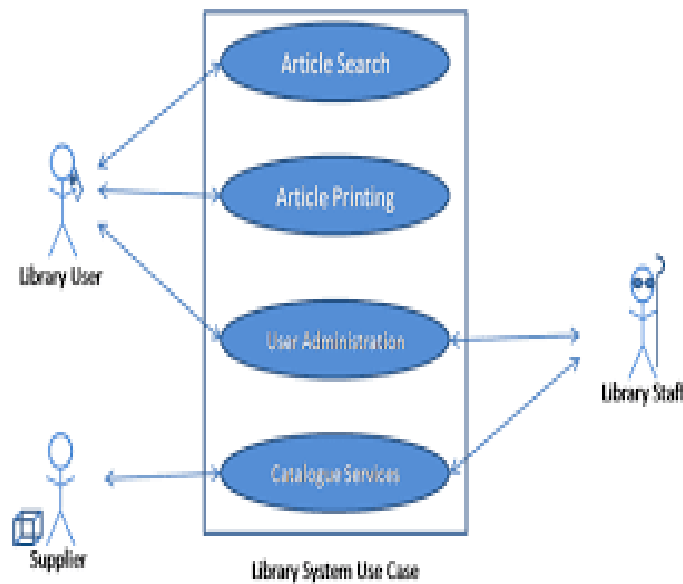
pendesainan berorientasi objek. Pada tahun 1995 dirilis *draft* pertama dari UML (versi 0.8). Sejak tahun 1996 pengembangan tersebut dikoordinasikan oleh *Object Management Group* (OMG – <http://www.omg.org>). Tahun 1997 UML versi 1.1 muncul, dan saat ini versi terbaru adalah versi 1.5 yang dirilis bulan Maret 2003. *Booch*, *Rumbaugh* dan *Jacobson* menyusun tiga buku serial tentang UML pada tahun 1999. Sejak saat itulah UML telah menjelma menjadi standar bahasa pemodelan untuk aplikasi berorientasi objek. (Yuni Sugiarti ; 2013 : 33)

Dalam pembuatan skripsi ini penulis menggunakan diagram *Use Case* yang terdapat di dalam UML. Adapun maksud dari *Use Case Diagram* diterangkan dibawah ini.

1. Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. *Use case diagram* dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem. Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use case* yang meng-*include* dieksekusi secara

normal. Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*. Sebuah *use case* juga dapat meng-*extend* *case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain. (Yuni Sugiarti ; 2013 : 41)



Gambar II.4. Contoh Use Case Diagram

Sumber : Yuni Sugiarti ; 2013

2. Class Diagram

Diagram kelas atau *class* diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Berikut adalah simbol-simbol pada diagram kelas :

Simbol	Deskripsi
	Package merupakan sebuah bungkus dari satu atau lebih kelas
	Kelas pada struktur sistem
	sama dengan konsep interface dalam pemrograman berorientasi objek
	relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity
	relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity
	relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus)
	relasi antar kelas dengan makna kebergantungan antar kelas
	relasi antar kelas dengan makna semua-bagian (whole-part)

Gambar II.5. Contoh Class Diagram

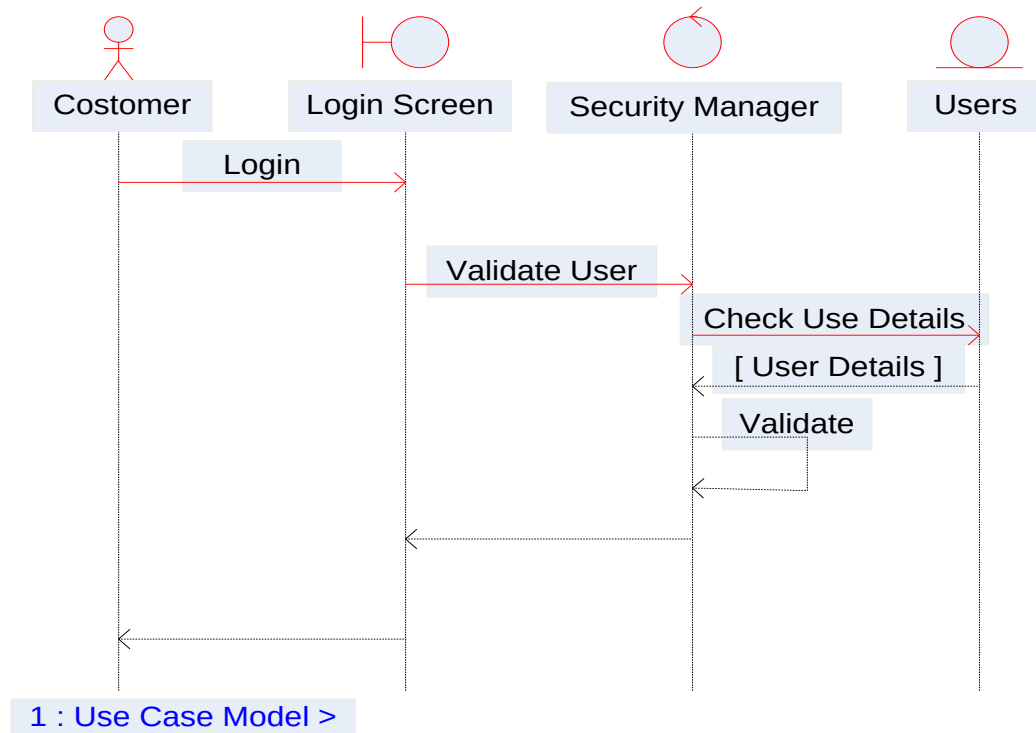
Sumber : Yuni Sugiarti ; 2013

3. Sequence Diagram

Diagram *Sequence* menggambarkan kelakuan/prilaku objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram *sequence* maka

harus diketahui objek-objek yang terlibat dalam sebuah *usecase* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Banyaknya diagram *sequence* yang harus digambar adalah sebanyak pendefinisian *usecase* yang memiliki proses sendiri atau yang penting semua *usecase* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram *sequence* sehingga semakin banyak *usecase* yang didefinisikan maka diagram *sequence* yang harus dibuat juga semakin banyak.



Gambar II.6. Contoh Sequence Diagram

Sumber : Yuni Sugiarti ; 2013

4. Activity Diagram

Activity diagram menggambarkan berbagai alur aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alur berawal, *decision* yang mungkin

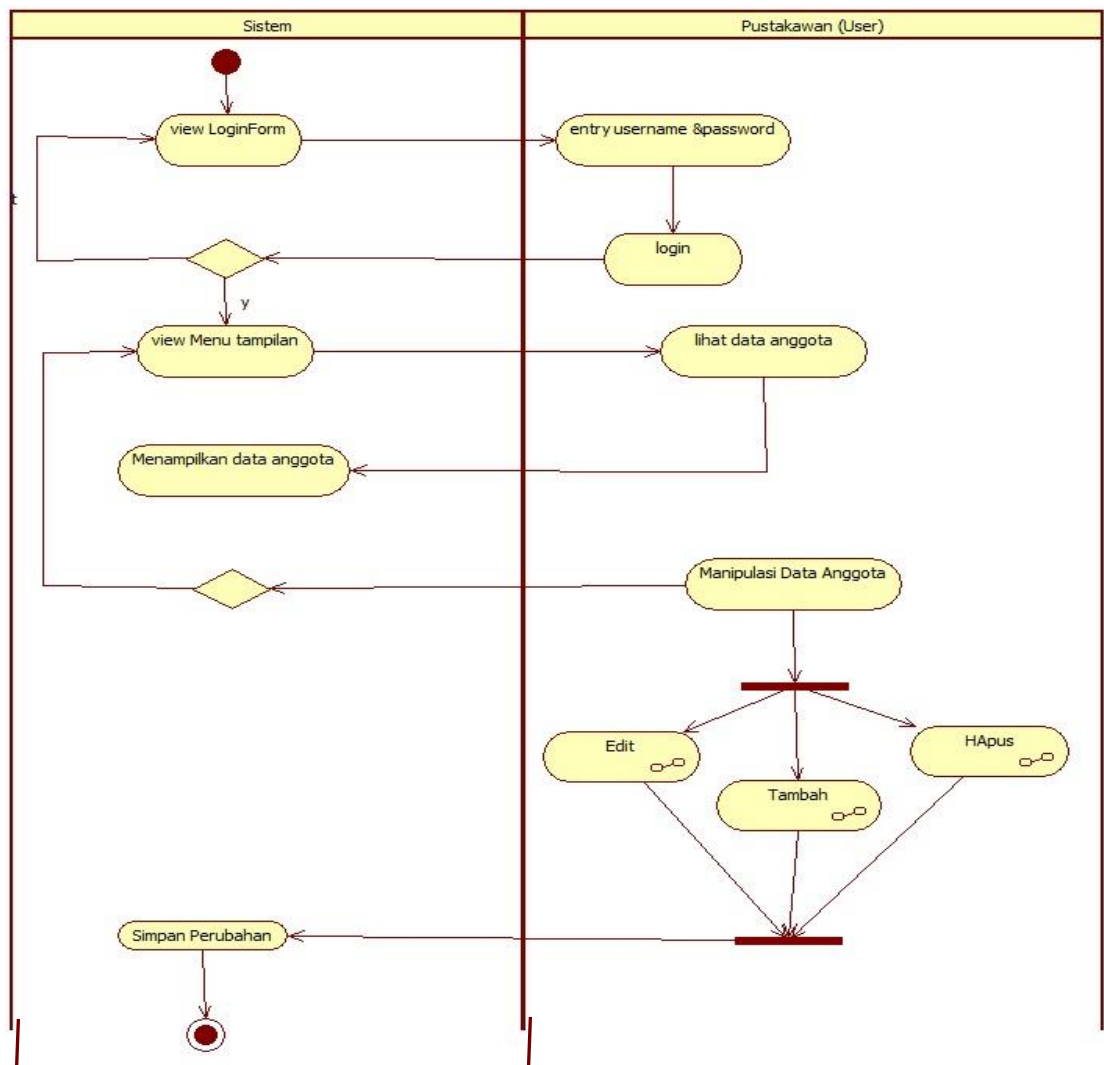
terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Activity diagram merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *states* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behaviour* internal sebuah sistem (dan interaksi antar *subsistem*) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari *level* atas secara umum.

Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan *behaviour* pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal.

Activity diagram dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.



Gambar II.7. Contoh Activity Diagram

Sumber : Yuni Sugiarti ; 2013