BAB II
TINJAUAN PUSTAKA

II.1 Pengertian Perangkat Lunak

Komputer merupakan mesin yang memproses fakta atau data menjadi informasi. Komputer di gunakan orang untuk meningkatkan hasil kerja dan memecahkan berbagai masalah. Yang menjadi pemroses data atau pemecah masalah itu adalah perangkat lunak. Defisnisi perangkat lunak sebagai berikut : "Berfungsi sebagai pengatur aktivitas kerja komputer dan semua intruksi yang mengarah pada sistem komputer. Perangkat lunak menjembatani interaksi user dengan komputer yang hanya memahami bahasa mesin."[3]

II.1.1 Klasifikasi Perangkat Lunak

Secara umum perangkat lunak ini dapat diklasifikasikan menjadi 2 [3], yaitu :

1. Perangkat Lunak Sistem Operasi

   Merupakan perangkat lunak yang berfungsi untuk mengkonfigurasikan komputer agar dapat menerima berbagai perintah dasar yang diberikan sebagai masukan. Perangkat lunak sistem juga merupakan sekumpulan program yang ditulis untuk melayani program-program yang lain.

   Contoh :

   a) *MS-DOS*

   b) *LINUX*

   c) *UNIX*

   d) *FREE BSD*

   e) *OS/2*

   f) *SUN OS (JAVA)*

   g) *WINDOWS*

   h) *MACINTOSH*, dll.

2. Perangkat Lunak Aplikasi

   Merupakan program siap pakai yang digunakan untuk aplikasi dibidang tertentu. Perangkat lunak aplikasi yang membantu user sehingga dapat bekerja lebih efektif dan efisien. Aplikasi ini dibagi atas beberapa bidang, antara lain :

   1. *Business & Office Application*

      Aplikasi perkantoran dipergunakan untuk menyelesaikan pekerjaan kantor seperti mengetik (*Word processing, document management*) membuat tabel kerja (*Worksheet*), membuat *Database* sederhana (*MS. Access*) ataupun mengolah image/citra sederhana.

      Contoh aplikasi perkantoran yang terpaket komplit :

      a) *Mocrosoft Office*

      b) *Koffice*

      c) *Star Office*

      d) *Abiword*

      e) *Open Office*

      f) *MS_Office for MAC,* dll.

      Aplikasi bisnis lainnya juga meliputi aplikasi pengolahan data keuangan, missal (*Personal Finance Application & Business Acounting Application*). Contoh :

      a) Aplikasi perhitungan neraca, rugi laba perusahaan.

      b) Aplikasi perhitungan pajak personal & perusahaan

      c) Aplikasi manajemen proyek

      d) Aplikasi persediaan stok barang dan penjualan

      e) Aplikasi pelayanan loket & kredit nasabah bank

      f) Aplikasi pemesanan tiket pesawat/kapal laut, dll.

   2. *Database Application*

      Aplikasi yang digunakan dalam pengolahan data baik yang berukuran kecil maupun besar bisa digunakan secara *stand-alone* (tunggal) maupun

dalam sistem berbasis jaringan lokal *Client server* maupun *webbase* (intranet maupun internet).

Contoh :

a) *MS SQL*

b) *FoxBase*

c) *Oracle*

d) *MySQL*

e) *PostgresSQL*

f) *SQL Server*, dll.

3. *Graphics Design Application*

   Aplikasi yang digunakan untuk membuat sketsa dua dimensi untuk desain ruangan, perabotan, mesin-mesin model industri, juga model tubuh manusia, hewan, dll.

   Contoh :

   a) *AutoCad*

   b) *Pro Design*

   c) *CorelDraw*

   d) *Adobe Ilistrator*, dll.

   Ada juga yang memasukan dalam golongan ini aplikasi *image*, *audio*, *video manipulator*, yaitu aplikasi yang digunakan untuk mengolah citra, suara hingga gambar 3 dimensi untuk pembuatan animasi, pengolahan *image, photo, audio, video editing*, dll.

   Contoh :

   a) *Adobe Photoshop, After Effect, Premire, & Audition*

   b) *Corel Draw, Corel RAVE, Corel Photo Paint*

   c) *Ulead Media studio*

   e) *Cool Edit Pro , Sound Force*

   f) *Macromedia Flash, Director*

   g) *3D Studio Max, Maya*, dll.

4. *Utility Tools & Antivirus*

   Software ini digunakan untuk operasi minor (*Background*) yang bekerja di belakang sistem, berguna untuk meningkatkan kinerja sistem operasi atau aplikasi dan juga performa *hardware*.

   Terdapat 2 jenis utilitas di dalam sistem operasi :

   a) *Utilitas Internal*, sudah termasuk dalam paket yang di instalasi dalam sistem operasi seperti *scandisk* di *Windows* atau perintah *fsck* (file *system check*) di *OS Linux.*

   b) *Utilitas eksternal*, yang harus diinstalasi kemudian pada OS, seperti *System Optimizer, PC Tools, Data/File Recovery, PC Maintenance, Internet Utilities, Norton System Work, Registry Clean & Repair, Hardware Repair.*

   Sementara untuk proteksi virus dan antivirus, contohnya adalah antivirus produk *Mc Affee, Grisoft, Norton, PandaSoft*, dll.

5. *Development Tools*

   Merupakan sistem bahasa pemrograman komputer yang dilengkapi dengan *compiler* tertentu sebagai sumber media penyusunan program aplikasi.

   Contoh :

   a) *Visual Basic*

   b) *Turbo Pascal*

   c) *Visual C++*

   d) *Turbo Assembler*

   e) *Phyton*

   f) *KDEvelopt*

   g) *Gdekstop*, dll.

   Termasuk dalam jenis aplikasi ini adalah web *development tools*, yaitu aplikasi yang digunakan untuk membuat/mendesain tampilan *website*

(homepage) di internet, seperti *FrontPage, Macromedi Dreamweaver, PHPedit, WebPage Editor*, dll.

6. *Communication Application*

   Aplikasi yang digunakan untuk membangun komunikasi antara satu komputer dengan komputer lain dalam jaringan komputer *local* maupun *global*.

   Contoh :

   a) *E-mail & Fax Application*

   b) *Remote Access*

   c) *Facebook chat, YM, GTalk, MSN*

   d) *Terminal Emulation*

   e) *Instant Messaging*

   f) *Web Browser*

   g) *Internet Telephony, dll.*

7. Jenis Aplikasi Lain

   a) *Multimedia Player,* seperti *WinAmp, XMMS, Power DVD, JetAudio, Windows Media Player*, dll.

   b) *Hacker Tools*, seperti *Find Password, Keygen/Crack Software License, Scanning Port*, dll.

   c) *Network & Security Application*, seperti *Firewall, Intruder Detections, Network Monitoring,* dll.

   d) *PC Games Apllication*, dan masih banyak lagi

II.2 Pengertian *Database*

Terdapat beberapa definisi basis data dari beberapa orang ahli basis data adalah sebagai berikut[2] :

1. *Database* adalah sekumpulan *data store* (bisa dalam jumlah yang sangat besar) yang tersimpan dalam *magnetic disk, optical disk, magnetic drum* atau media penyimpanan sekunder lainnya.

2. *Database* adalah sekumpulan program – program aplikasi umum yang bersifat "*batch*" yang mengeksekusi dan memproses data secara umum (seperti pencarian, peremajaan, penambahan, dan penghapusan terhadap data).
3. *Database* terdiri dari data yang akan digunakan atau diperuntukan terhadap banyak *user*, dimana masing – masing *user* (baik menggunakan teknik pemrosesan yang bersifat *batch* atau *online*) akan menggunakan data tersebut sesuai dengan tugas dan fungsinya, dan user lain dapat juga menggunakan data tersebut dalam waktu yang bersamaan.
4. *Database* adalah koneksi terpadu dari data – data yang saling berkaitan dari suatu *enterprise* (perusahaan, instansi pemerintah atau swasta).

Konsep dasar dari basis data adalah kumpulan dari catatan-catatan, atau potongan dari pengetahuan. Sebuah basis data memiliki penjelasan terstruktur dari jenis fakta yang tersimpan didalamnya: penjelasan ini disebut skema. Skema menggambarkan obyek yang diwakili suatu basis data, dan hubungan diantara obyek tersebut. Ada banyak cara untuk mengorganisasi skema, atau memodelkan struktur basis data: ini dikenal sebagai model basis data atau model data. Model yang umum digunakan sekarang adalah model relasional, yang menurut istilah layman mewakili semua informasi dalam bentuk tabel-tabel yang saling berhubungan dimana setiap tabel terdiri dari baris dan kolom (definisi yang sebenarnya menggunakan terminologi matematika). Dalam model ini, hubungan antar tabel diwakili dengan menggunakan nilai yang sama antar tabel. Model yang lain seperti model hierarkis dan model jaringan menggunakan cara yang lebih eksplisit untuk mewakili hubungan antar tabel.

II.2.1 Tipe *Database*

Menurut Ling Liu[6] Terdapat 12 tipe *Database*, antara lain : *Operational Database, Analytical Database, Data warehouse, Distributed Database, End-user Database, ExternalDatabase, Hypermedia Databases on the web, Navigational Database, In-memoryDatabases, Document-oriented Databases, Real-time Databases, Relational Database.*

1. *Operational Database*

Database ini menyimpan data rinci yang diperlukan untuk mendukung operasi dari seluruh organisasi. Mereka juga disebut *subject-area Databases (SADB)*, transaksi *Database*, dan produksi *Database*. Contoh: *Database* pelanggan, *Database* pribadi, *Database* inventaris, akuntansi *Database.*

2. *Analytical Database*

Database ini menyimpan data dan informasi yang diambil dari operasional yang dipilih dan *eksternal Database*. Mereka terdiri dari data dan informasi yang dirangkum paling dibutuhkan oleh sebuah organisasi manajemen dan *End-user* lainnya. Beberapa orang menyebut analitis multi dimensi *Database* sebagai *Database, manajemen Database,* atau informasi *Database*.

3. *Data Warehouse*

Sebuah data *warehouse* menyimpan data dari saat ini dan tahun-tahun sebelumnya – data yang diambil dari berbagai *Database* operasional dari sebuah organisasi. Data *warehouse* menjadi sumber utama data yang telah diperiksa, diedit, standar dan terintegrasi sehingga dapat digunakan oleh para manajer dan pengguna akhir lainnya diseluruh organisasi profesional. Perkembangan terakhir dari data *warehouse* adalah dipergunakan sebagai *Shared nothing architecture* untuk memfasilitasi ekstrem scaling.

4. *Distributed Database*

Merupakan *Database*-kelompok kerja lokal dan departemen di kantor regional, kantor cabang, pabrik-pabrik dan lokasi kerja lainnya. *Database* ini dapat mencakup kedua segmen yaitu operasional dan *user Database*, serta data yang dihasilkan dan digunakan hanya pada pengguna situs sendiri.

5. *End-User Database*

Database ini terdiri dari berbagai *file* data yang dikembangkan oleh *end-user* di *workstation* mereka. Contoh dari ini adalah koleksi dokumen dalam *spreadsheet*, *word processing* dan bahkan *download file.*

6. *External Database*

Database ini menyediakan akses ke *eksternal*, data milik pribadi *online* – tersedia untuk biaya kepada pengguna akhir dan organisasi dari layanan komersial. Akses ke kekayaan informasi dari *Database eksternal* yang tersedia untuk biaya dari layanan *online* komersial dan dengan atau tanpa biaya dari banyak sumber di internet.

7. *Hypermedia Databases on the Web*

Merupakan kumpulan dari halaman-halaman *multimedia* yang saling berhubungan di sebuah situs web. Mereka terdiri dari *homepage* dan halaman *hyperlink* lain dari *multimedia* atau campuran media seperti teks, grafik, gambar foto, klip *video, audio* dll.

8. *Navigational Database*

Dalam navigasi *Database*, *queries* menemukan benda terutama dengan mengikuti referensi dari objek lain.

9. *In-memory Databases*

Database di memori terutama bergantung pada memori utama untuk penyimpanan data komputer. Ini berbeda dengan sistem manajemen *Database* yang menggunakan *disk* berbasis mekanisme penyimpanan. *Database* memori utama lebih cepat daripada dioptimalkan *disk Database* sejak Optimasi algoritma *internal* menjadi lebih sederhana dan lebih sedikit CPU mengeksekusi instruksi. Mengakses data dalam menyediakan memori lebih cepat dan lebih dapat diprediksi kinerja dari *disk*. Dalam aplikasi dimana waktu respon sangat penting, seperti peralatan jaringan telekomunikasi yang mengoperasikan sistem darurat, *Database* memori utama yang sering digunakan.

10. *Document-Oriented Databases*

Merupakan program komputer yang dirancang untuk aplikasi berorientasi dokumen. Sistem ini bisa diimplementasikan sebagai lapisan di atas sebuah *Database relasional* atau objek *Database*. Sebagai lawan dari *Database relasional*, dokumen berbasis *Database* tidak menyimpan data dalam tabel dengan ukuran seragam kolom untuk setiap *record*. Sebaliknya, mereka menyimpan setiap catatan sebagai dokumen

yang memiliki karakteristik tertentu. Sejumlah bidang panjang apapun dapat ditambahkan ke dokumen. Bidang yang dapat juga berisi beberapa bagian data.

11. *Real-Time Databases*

Merupakan sistem pengolahan dirancang untuk menangani beban kerja negara yang dapat berubah terus-menerus. Ini berbeda dari *Database* tradisional yang mengandung data yang terus-menerus, sebagian besar tidak terpengaruh oleh waktu. Sebagai contoh, pasar saham berubah dengan cepat dan dinamis. *Real-time processing* berarti bahwa transaksi diproses cukup cepat bagi hasil untuk kembali dan bertindak segera. *Real-time Database* yang berguna untuk akuntansi, perbankan, hukum, catatan medis, *multi-media*, kontrol proses, sistem reservasi, dan analisis data ilmiah.

12. *Relational Database*

Standar komputasi bisnis sejak tahun 2009, *relational Database* adalah *Database* yang paling umum digunakan saat ini. Menggunakan *table* untuk informasi struktur sehingga mudah untuk mencari.

II.2.2 Perangkat Lunak *Database*

Database dapat dibuat dan diolah dengan menggunakan suatu program komputer, yaitu yang biasa disebut dengan *software* (perangkat lunak). *Software* yang digunakan untuk mengelola dan memanggil kueri (*query*) *Database* disebut *Database Management System (DBMS)* atau jika diterjemahkan kedalam Bahasa Indonesia berarti “Sistem Manajemen Basis Data”.

Menurut Andri Kristanto[5] manajemen sistem basis data atau *Database manajemen system* dapat didefinisikan sebagai berikut :

“ Perangkat lunak yang didesain untuk membantu dalam hal pemeliharaan dan utilitas kumpulan data dalam jumlah besar “

DBMS terdiri dari dua komponen, yaitu *Relational Database Management System (RDBMS)* dan *Overview of Database Management System (ODBMS)*. RDBMS meliputi *Interface Drivers, SQL Engine, Transaction Engine, Relational*

Engine, dan *Storage Engine*. Sedangkan ODBMS meliputi Language *Drivers*, *Query Engine*, *Transaction Engine*, dan *Storage Engine.*

Sedangkan level dari *software*-nya sendiri, terdapat dua level *software* yang memungkinkan untuk membuat sebuah *Database* antara lain :

a) *High Level Software*

b) *Low Level Software.*

Yang termasuk didalam *High Level Software*, antara lain *Microsoft SQL Server, Oracle, Sybase, Interbase, XBase, Firebird, MySQL, PostgreSQL, Microsoft Access, dBase III, Paradox, FoxPro, Visual FoxPro, Arago, Force, Recital, dbFast, dbXL,Quicksilver, Clipper, FlagShip, Harbour, Visual dBase, dan Lotus Smart Suite Approach.* Sedangkan yang termasuk di dalam *Low Level Software* antara lain *Btrieve dan Tsunami Record Manager.*

II.3 Pengertian Jaringan Komputer (*Networking*)

Iwan Sofana[12] menyatakan bahwa jaringan komputer (*computer network*), adalah satu himpunan interkoneksi sejumlah komputer *autonomous* dalam suatu himpunan interkoneksi sejumlah komputer *autonomous*. Dalam bahasa yang popular dapat dijelaskan bahwa jaringan komputer adalah sekumpulan beberapa komputer (dan perangkat lain seperti *printer*, *hub* dan sebagainya) yang saling terhubung satu sama lain melaui media perantara. Media perantara ini biasa berupa media kabel ataupun media tanpa kabel (*nirkabel*). Informasi berupa data akan mengalir dari suatu komputer ke komputer lainnya atau dari satu komputer ke perangkat lain, sehingga masing-masing komputer yang terhubung tersebut biasa bertukar data atau berbagai perangkat keras.

II.3.1 Komponen – Komponen Jaringan Komputer

Terdapat tiga komponen utama dalam sebuah jaringan, yaitu :

1) *Host* atau *node*, yaitu sistem komputer yang berfungsi sebagai sumber atau penerima dari data yang dikirimkan. *Node* ini dapat berupa:

Server : komputer tempat penyimpanan data dan program-program aplikasi yang digunakan dalam jaringan,

Client : komputer yang dapat mengakses sumber daya (berupa data dan program aplikasi) yang ada pada *server*,

Shared pheriperal : peralatan-peralatan yang terhubung dan digunakan dalam jaringan (misalnya, *printer*, *scanner*, *harddisk*, *modem*, dan lain-lain).

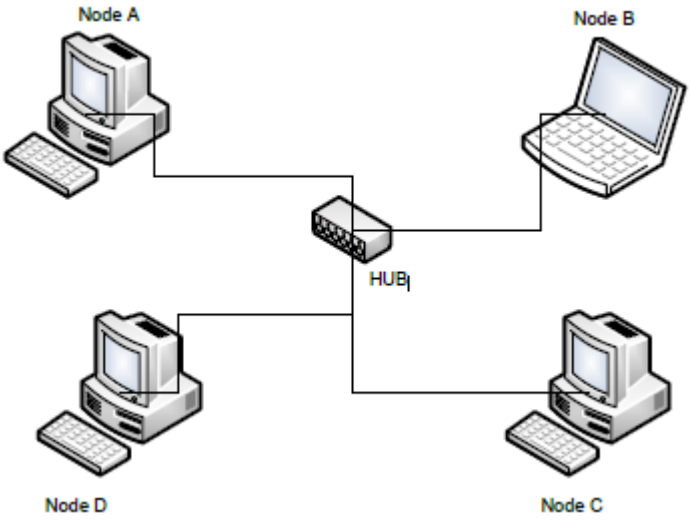
2) *Link*, adalah media komunikasi yang menghubungkan antara *node* yang satu dengan *node* lainnya. Media ini dapat berupa saluran transmisi kabel dan tanpa kabel,
3) *Software* (Perangkat Lunak), yaitu program yang mengatur dan mengelola jaringan secara keseluruhan. Termasuk di dalamnya sistem operasi jaringan yang berfungsi sebagai pengatur komunikasi data dan periferal dalam jaringan.

II.3.2 Tipe –Tipe Jaringan Komputer

Ada beberapa tipe jaringan komputer yang umumnya digunakan. Berikut ini beberapa klasifikasi tipe jaringan komputer yang ada :

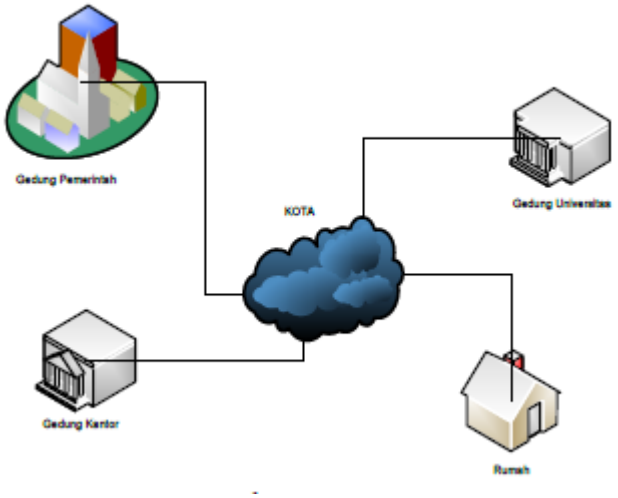
1. Berdasarkan letak geografis

a) *Local Area Network (LAN)*, jaringan ini berada pada satu bangunan atau lokasi yang sama dengan jarak antar komputer yang dihubungkannya bisa mencapai 5 sampai 10 km. dengan kecepatan transmisi data yang tinggi (mulai dari 10 Mbps sampai 100Mbps), dan menggunakan peralatan tambahan seperti *repeater*, *hub*, dan sebagainya.



Gambar II.1 LAN

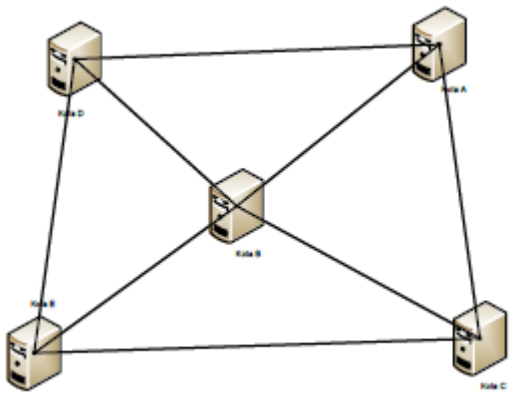
b) *Metropolitan Area Network (MAN),* jaringan ini merupakan gabungan beberapa *LAN* yang terletak pada satu kota (jangkauan 50-75 mil) yang dihubungkan dengan kabel khusus atau melalui saluran telepon, dengan kecepatan transmisi antara 34 sampai 155Mbps, dan menggunakan peralatan seperti *router*, telepon, *ATM switch*, dan antena parabola.



Gambar II.2 MAN

c) *Wide Area Network (WAN),* jaringan ini merupakan gabungan dari komputer *LAN* atau *MAN* yang ada di seluruh permukaan bumi ini yang

dihubungkan dengan saluran telepon, gelombang elektromagnetik, atau satelit. Dengan kecepatan transmisi yang lebih lambat dari 2 jenis jaringan sebelumnya, dan menggunakan peralatan seperti *router, modem, WAN switches.*



Gambar II.3 WAN

2. Berdasarkan arsitektur jaringan

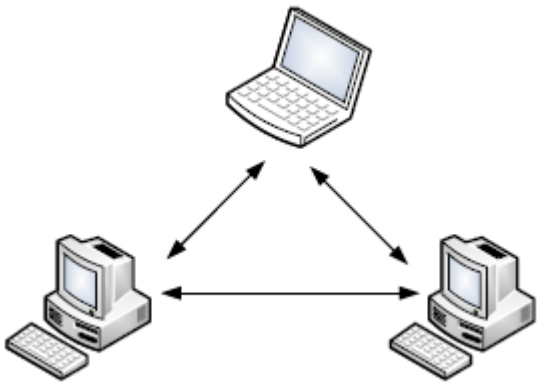
   a) Jaringan *peer to peer*

      Pada jaringan *peer to peer* semua komputer memiliki posisi setara/sejajar, dalam hierarki yang sama. Setiap komputer dapat menjadi klien terhadap komputer peer lainnya, setiap komputer dapat pula berbagi sumber daya dengan komputer yang berada dalam jaringan *peer-to-peer* ini. Sumber daya diletakkan secara desentralisasi pada setiap anggota jaringan, dan tidak memerlukan administrator jaringan.

      Aliran informasi bisa mengalir diantara dua komputer secara langsung, di mana pun. Namun, jaringan ini tidak sepenuhnya bebas tanpa kontrol, masih bisa digunakan *password* untuk memproteksi *file* dan *folder*, dapat juga diatur agar orang-orang tertentu tidak bisa menggunakan periferal tertentu.

Karena kemudahan pemasangan, pemeliharaan, serta biaya, jaringan ini lebih populer untuk jaringan dengan jumlah komputer yang sedikit (sekitar 2 sampai 20 komputer).

Sifat jaringan *peer to peer* digunakan untuk hubungan antara setiap komputer yang terhubung dalam jaringan komputer yang ada, sehingga komunikasi data terjadi antar komputer dengan hierarki yang sama karena setiap komputer dapat berfungsi sebagai *server* maupun *klien*.



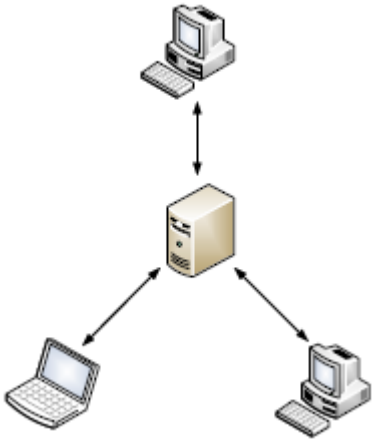
Gambar II.4 Jaringan Peer-to-peer

b) Jaringan berbasis *server* (*server-based network/server-client network*)

Pada jaringan *client-server* perangkat lunak yang mengontrol keseluruhan kerja jaringan berada pada *server*. Jaringan ini dapat menghubungkan ratusan komputer dengan tingkat keamanan yang tidak dimungkinkan dalam jaringan *peer-to-peer*.

Jaringan ini bisa diatur sehingga setiap klien harus *log on* ke *server* sebelum bisa memanfaatkan sumber daya yang terhubung ke *server*. *Server* lalu mengotentikasi klien dan memverifikasi bahwa komputer yang digunakan klien tersebut memiliki izin untuk *log on* ke jaringan,

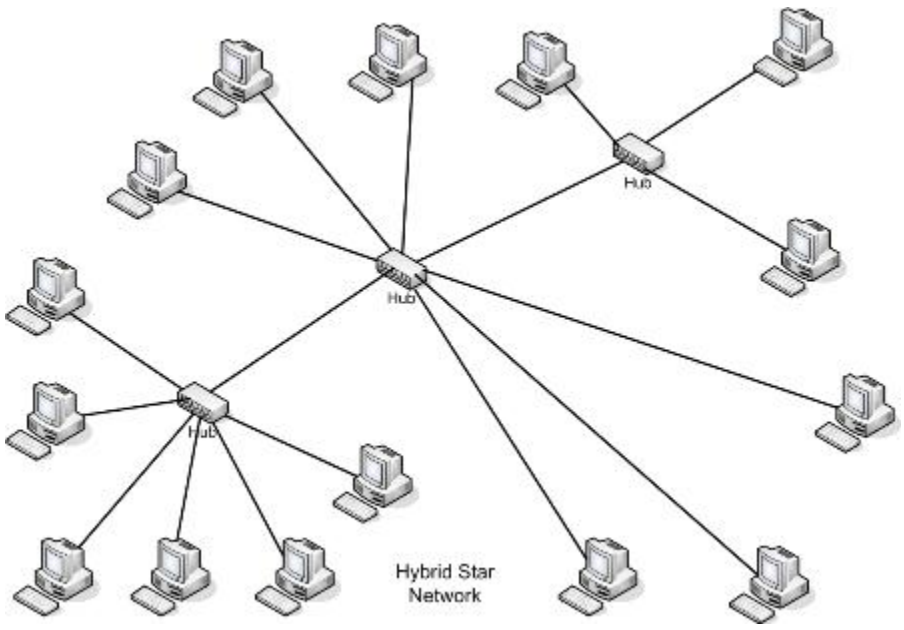
dengan memeriksa *username* dan *password* klien tersebut terhadap *Database* pada *server*.



Gambar II.5 Jaringan *Client Server*

c) Jaringan *Hybrid.*

Jaringan ini merupakan gabungan dari sifat pada *jaringan peer to peer* dan c*lient-server. Workgroup* yang terdiri dari beberapa komputer yang saling terhubung dapat mengelola sumber daya tanpa membutuhkan otorisasi dari *administrator* jaringan atau *server*. Pada jenis jaringan ini, terdapat pula sifat dari jaringan *client-server* sedemikian sehingga tingkat keamanan dapat lebih terjaga dan adanya *server* yang mempunyai suatu fungsi layanan tertentu, seperti sebagai *file server, print server, Database server, mail server*, dan lainnya.



Gambar II.6 Jaringan *Hybrid*

3. Berdasarkan teknologi transmisi

   a) Jaringan *switch*, merupakan jaringan yang penyampaian informasi dari pengirim ke penerima melalui mesin-mesin perantara atau saluran telepon,

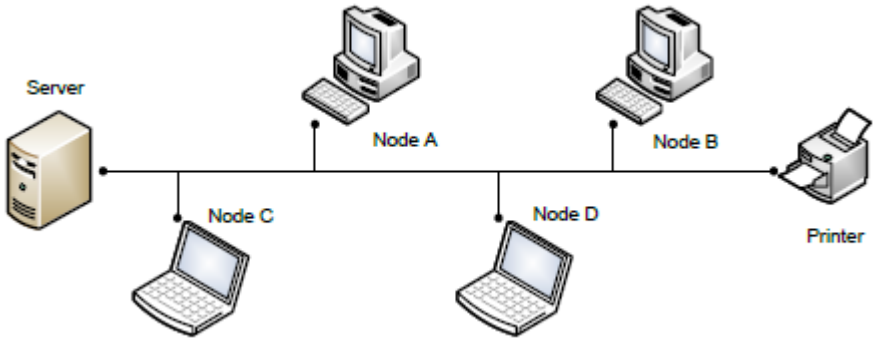
   b) Jaringan *broadcast*, merupakan jaringan yang penyampaian informasi dari pengirim ke penerima dilakukan secara *broadcast* (disiarkan ke segala arah) baik melalui saluran kabel maupun saluran tanpa kabel.

II.3.3 Topologi Jaringan

Topologi jaringan komputer adalah bentuk perancangan baik secara fisik maupun secara logik yang digunakan untuk membangun sebuah jaringan komputer. Adapun hal yang menjelaskan hubungan geometris antara unsur-unsur dasar penyusun jaringan, yaitu *node, link,* dan *station.*

II.3.3.1 Topologi *Bus*

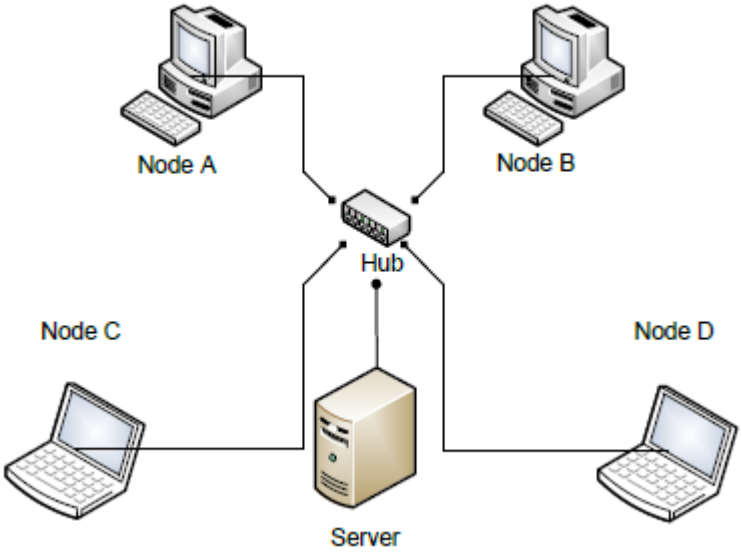
Topologi *Bus* adalah sebuah kabel tunggal atau kabel pusat dimana seluruh *workstation* dan *server* dihubungkan. Keunggulan topologi *Bus* adalah pengembangan jaringan atau penambahan *workstation* baru dapat dilakukan dengan mudah tanpa mengganggu *workstation* lain. Kelemahan dari topologi ini adalah bila terdapat gangguan di sepanjang kabel pusat maka keseluruhan jaringan akan mengalami gangguan.



Gambar II.7 Topologi *Bus*

II.3.3.2 Topologi *Star*

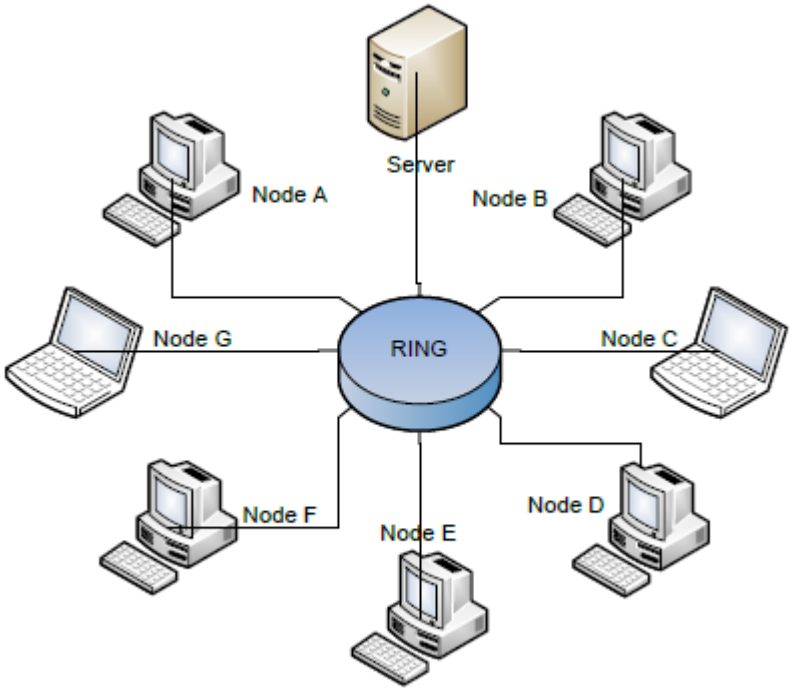
Pada topologi *star*, masing-masing *workstation* dihubungkan secara langsung ke *server* atau *hub*. Keunggulan dari topologi tipe *star* ini adalah bahwa dengan adanya kabel tersendiri untuk setiap *workstation* ke *server*, maka *bandwidth* atau lebar jalur komunikasi dalam kabel akan semakin lebar sehingga akan meningkatkan unjuk kerja jaringan secara keseluruhan. Dan juga bila terdapat gangguan di suatu jalur kabel maka gangguan hanya akan terjadi dalam komunikasi antara *workstation* yang bersangkutan dengan *server*, jaringan secara keseluruhan tidak mengalami gangguan. Kelemahan dari topologi *Star* adalah kebutuhan kabel yang lebih besar dibandingkan dengan topologi lainnya.



Gambar II.8 Topologi *Star*

II.3.3.3 Topologi *Ring*

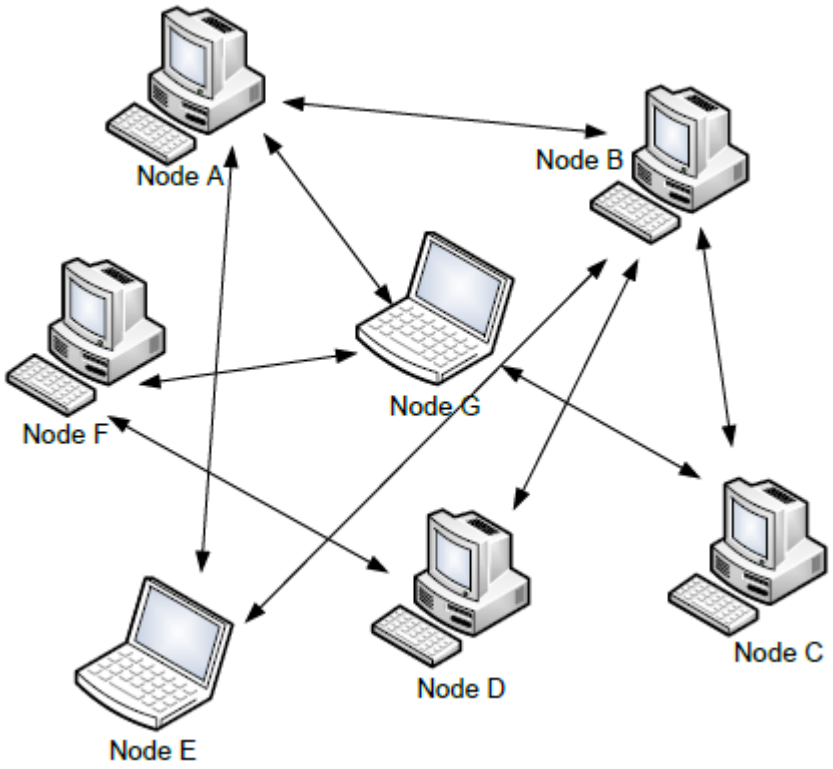
Didalam topologi *Ring* semua *workstation* dan *server* dihubungkan sehingga terbentuk suatu pola lingkaran atau cincin. Tiap *workstation* ataupun *server* akan menerima dan melewatkan informasi dari satu komputer ke komputer lain, bila alamat-alamat yang dimaksud sesuai maka informasi diterima dan bila tidak informasi akan dilewatkan. Kelemahan dari topologi ini adalah setiap *node* dalam jaringan akan selalu ikut serta mengelola informasi yang dilewatkan dalam jaringan, sehingga bila terdapat gangguan di suatu node maka seluruh jaringan akan terganggu. Keunggulan topologi *Ring* adalah tidak terjadinya *collision* atau tabrakan pengiriman data seperti pada topologi *Bus*, karena hanya satu *node* dapat mengirimkan data pada suatu saat.



Gambar II.9 Topologi *Ring*

II.3.3.4 Topologi *Mesh*

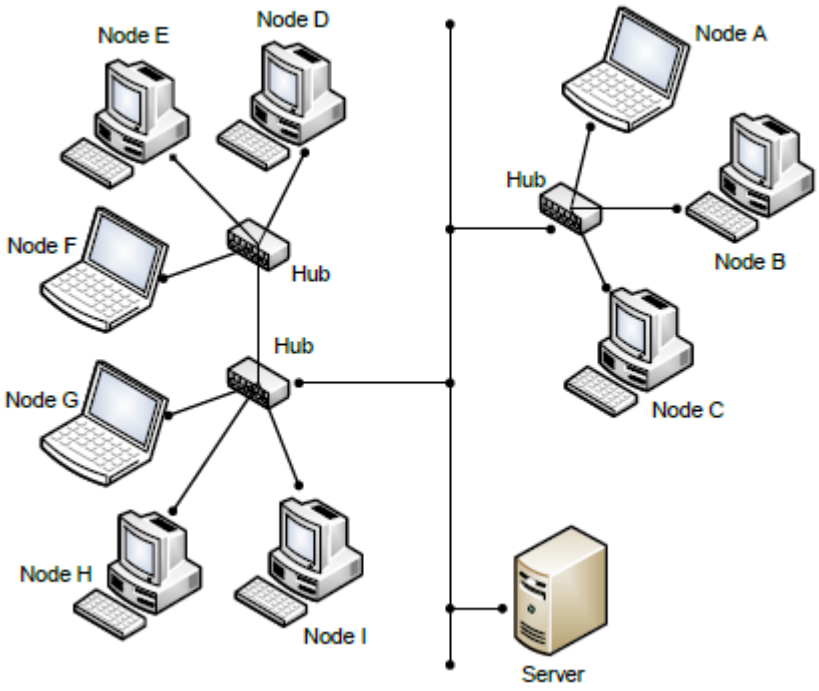
Pada topologi *mesh* tiap komputer terhubung langsung dengan komputer lain (*peer to peer*). Setiap komputer mempunyai jalur sendiri-sendiri dengan komputer lain, sehigga tidak akan terjadi *collision domain*. Kerugian dari penggunaan topologi ini adalah penggunaan *ethernet* dan kabel yang banyak sehingga dibutuhkan dana yang besar. Keuntungan dari penggunaan topologi ini adalah apabila ada salah satu jalur pada komputer putus, komputer masih dapat berhubungan dengan jalur yang lain.



Gambar II.10 Topologi *Mesh*

II.3.3.5 Topologi *Tree*

Topologi Jaringan Pohon (*Tree*) Topologi jaringan ini disebut juga sebagai topologi jaringan bertingkat. Topologi ini biasanya digunakan untuk interkoneksi antar sentral dengan hirarki yang berbeda. Untuk hirarki yang lebih rendah digambarkan pada lokasi yang rendah dan semakin keatas mempunyai hirarki semakin tinggi. Topologi jaringan jenis ini cocok digunakan pada sistem jaringan komputer. Kelebihan Seperti topologi star perangkat terhubung pada pusat pengendali / *Hub*. Tetapi *Hub* dibagi menjadi dua, *central Hub*, dan *secondary Hub*. Topologi tree ini memiliki keunggulan lebih mampu menjangkau jarak yang lebih jauh dengan mengaktifkan fungsi *Repeater* yang dimiliki oleh *Hub*. Kelemahan topologi *tree*, kabel yang digunakan menjadi lebih banyak sehingga diperlukan perencanaan yang matang dalam pengaturannya , termasuk di dalamnya adalah tata letak ruangan.



Gambar II.11 Topologi *Tree*

II.3.4 Protokol TCP/IP

II.3.4.1 TCP (*Transmission Control Protocol*)

Transmission Control Protocol (TCP) melakukan transmisi data per segmen, artinya paket data dipecah dalam jumlah yang sesuai dengan besaran paket, kemudian dikirim satu per satu hingga selesai. Agar pengiriman data sampai dengan baik, maka pada setiap paket pengiriman TCP akan menyertakan nomor seri (*sequence number*). Komputer *Client* yang menerima paket tersebut harus mengirim balik sebuah sinyal *ACKnowledge* dalam satu periode yang ditentukan. Bila pada waktunya komputer *Client* belum juga memberikan ACK, maka akan terjadi *time out* yang menandakan pengiriman paket gagal dan harus diulang kembali. Model protokol TCP disebut sebagai *connection oriented protocol*.

Port merupakan pintu masuk datagram dan paket data. *Port* data dibuat mulai dari 0 sampai 65.536. *Port* 0 sampai 1024 diesdiakan untuk layanan standar, seperti

FTP, TELNET, *Mail*, *Web* dan lainnya. *Port* ini lebih dikenal dengan nama *well-know port.*

Tabel II.1 *Port* TCP

No Port	Keterangan
21	FTP
23	Telnet
25	SMTP
80	HTTP/*Web*
110	POP3
161	SNMP
162	SNMP Trop

II.3.4.2 *IP* (*Internet Protokol*)

Internet protokol menggunakan IP address sebagai identitas. Pengiriman data akan dibungkus dalam paket dengan label berupa *IP address* si pengirim dan *IP address* penerima. Apabila *IP* penerima melihat pengiriman paket tersebut dengan identitas *IP address* yang sesuai, maka *datagram* tersebut akan diambil dan disalurkan ke *TCP* melalui *port*. *IP address* dibagi menjadi dua bagian, yaitu :

1. *Network ID* (menyatakan identitas jaringan)
2. *HOST ID* (menyatakan identitas komputer)

Sedangkan untuk penulisan *IP address* terbagi atas empat oktet, yang masing-masing mempunyai nilai maksimal 255 atau lebih jelasnya dapat dilihat pada sub bab berikut.

II.3.4.3 *IPv4*

Alamat IP versi 4 sering disebut dengan Alamat *IPv4* adalah sebuah jenis pengalamatan jaringan yang digunakan di dalam protokol jaringan *TCP/IP* yang menggunakan protokol IP versi 4. Panjang totalnya adalah 32-*bit*, dan secara teoritis dapat mengalamati hingga 4 miliar *host* komputer atau lebih tepatnya 4.294.967.296

host di seluruh dunia, jumlah *host* tersebut didapatkan dari 256 (didapatkan dari 8 bit) dipangkat 4 karena terdapat 4 oktet, sehingga nilai maksimal dari alamt IP versi 4 tersebut adalah 255.255.255.255 dimana nilai dihitung dari nol sehingga nilai nilai *host* yang dapat ditampung adalah 256x256x256x256 = 4.294.967.296 *host*. sehingga bila *host* yang ada diseluruh dunia melebihi kuota tersebut maka dibuatlah IP versi 6 atau *IPv6*.

II.3.4.4 *IP Addres*

Alamat IP (*Internet Protocol Address*) atau sering disingkat IP biasa digunakan untuk mengindentifikasi *interface* jaringan pada *host* dari suatu mesin. adalah deretan angka biner antar 32-*bit* sampai 128-*bit* yang dipakai sebagai alamat identifikasi untuk tiap komputer *host* dalam jaringan Internet. Panjang dari angka ini adalah 32-*bit* untuk *IPv4* atau IP versi 4, dan 128-*bit* untuk *IPv6* atau IP versi 6 yang menunjukkan alamat dari komputer tersebut pada jaringan Internet berbasis TCP/IP.

II.3.4.5 Pembagian Kelas IP

Alamat IP dibagi-bagi menjadi kelas-kelas yang masing-masing mempunyai kapasitas jumlah IP yang berbeda-beda.seperti terlihat pada tabel. Sebenarnya yang menjadi pembeda kelas IP versi 4 adalah pola biner yang terdapat dalam oktet pertama utamanya adalah bit-bit awal/*high-order bit*, tapi untuk lebih mudah mengingatnya, akan lebih cepat diingat dengan menggunakan representasi desimal.

Tabel II.2 Pembagian Kelas IP

Kelas Alamat IP	Oktet pertama (desimal)	Oktet pertama (biner)	Digunakan oleh
Kelas A	1–126	0xxx xxxx	Alamat *unicast* untuk jaringan skala besar.
Kelas B	128–191	10xx xxxx	Alamat *unicast* untuk jaringan skala menengah hingga skala besar.
Kelas C	192–223	110x xxxx	Alamat *unicast* untuk jaringan skala kecil.
Kelas D	224–239	1110 xxxx	Alamat *multicast* (bukan alamat *unicast*).
Kelas E	240–255	1111 xxxx	Direservasikan umumnya digunakan sebagai alamat percobaan (eksperimen) bukan alamat *unicast*.

Kelas A

Alamat-alamat kelas A diberikan untuk jaringan skala besar. Nomor urut bit tertinggi di dalam alamat IP kelas A selalu diset dengan nilai 0 (nol). Tujuh bit berikutnya untuk melengkapi oktet pertama akan membuat sebuah *network identifier*. 24 *bit* sisanya atau tiga oktet terakhir merepresentasikan *host identifier*. Ini mengizinkan kelas A memiliki hingga 126 jaringan, dan 16,777,214 *host* tiap jaringannya. Alamat dengan oktet awal 127 tidak diizinkan, karena digunakan untuk mekanisme *Interprocess Communication* (*IPC*) di dalam mesin yang bersangkutan.

Kelas B

Alamat-alamat kelas B dikhususkan untuk jaringan skala menengah hingga skala besar. Dua *bit* pertama di dalam oktet pertama alamat IP kelas B selalu diset ke bilangan biner10. 14 bit berikutnya untuk melengkapi dua oktet pertama, akan membuat sebuah *network identifier*. 16 bit sisanya dua oktet terakhir

merepresentasikan *host identifier*. Kelas B dapat memiliki 16,384 *network*, dan 65,534 *host* untuk setiap *network*-nya.

Kelas C

Alamat IP kelas C digunakan untuk jaringan berskala kecil. Tiga bit pertama di dalam oktet pertama alamat kelas C selalu diset ke nilai biner 110. 21 bit selanjutnya untuk melengkapi tiga oktet pertama akan membentuk *sebuah network identifier.* 8 bit sisanya sebagai oktet terakhir akan merepresentasikan *host identifier*. Ini memungkinkan pembuatan total 2,097,152 buah *network*, dan 254 *host* untuk setiap *network*-nya.

Kelas D

Alamat IP kelas D disediakan hanya untuk alamat-alamat IP *multicast*, sehingga berbeda dengan tiga kelas di atas. Empat bit pertama di dalam IP kelas D selalu diset ke bilangan biner 1110. 28 bit sisanya digunakan sebagai alamat yang dapat digunakan untuk mengenali *host.*

Kelas E

Alamat IP kelas E disediakan sebagai alamat yang bersifat "eksperimental" atau percobaan dan dicadangkan untuk digunakan pada masa depan. Empat bit pertama selalu diset kepada bilangan biner1111. 28 bit sisanya digunakan sebagai alamat yang dapat digunakan untuk mengenali *host*.

II.4 Teknologi *Chatting*

II.4.1 *Instant Messaging*

Instant Messaging atau yang biasa disebut *chatting* merupakan sebuah teknologi internet yang memungkinkan para pengguna dalam jaringan internet untuk mengirimkan pesan-pesan singkat secara langsung pada saat yang bersamaan (*real time*) dengan menggunakan teks kepada pengguna lainnya yang sedang terhubung ke jaringan yang sama.

Konsep yang digunakan oleh teknologi ini muncul pada awal-awal pengembangan sistem operasi *UNIX* dan jaringan internet. para pengguna yang sudah

masuk log dapat mengirimkan perintah berupa *talk*, *write*, dan *finger* untuk melihat siapa saja yang sudah masuk log dan akhirnya mengirimkan pesan singkat kepada mereka.

II.4.2 Sejarah *Instant Messaging*

Perkembangan *Instant Messaging* diawali ketika mulai maraknya orang menggunakan teknologi secara *online* pada awal tahun 1990 ketika orang - orang mulai meluangkan banyak waktu untuk mengakses internet.

Para pengembang piranti lunak menciptakan sebuah *software chat room*, dimana suatu grup atau perorangan dapat melihat serta mengirimkan pesan kepada sertiap orang yang ada pada "*room*" tersebut. Penyedia jasa online *Quantum Link* menawarkan fitur untuk dapat saling berkirim pesan dengan sesama pengguna komputer yang sedang *online*. Penerapan *Instant messaging* ini memiliki dasar yang tidak jauh berbeda, yaitu sebuah *chat room* yang ditujukan untuk dua orang.

Instant Messaging ini mulai meledak di internet pada November 1996, yaitu ketika *Mirabilis* memperkenalkan *ICQ* yang merupakan sebuah *Instant Messaging* yang dapat dipergunakan secara gratis bagi setiap orang. Pada tahun 1997, *AOL* menjadi *pioneer* dalam komunitas *online*. Hal ini dikarenakan *AOL* memberikan kemampuan bagi pengguna untuk dapat melakukan komunikasi dalam waktu yang sama dalam sebuah *chat room* dan *instant messaging*. Model *ICQ* inilah yang kemudian menjadi dasar yang penting dalam penggunaan *Instant Messaging* yang beredar sekarang ini.

Saat ini, perkembangan *Instant Messaging* telah mengalami kemajuan yang amat pesat. Penggunaan *Instant Messaging* yang awalnya hanya dapat diakses oleh para penggunanya melalui komputer, sekarang ini telah dapat diakses melalui telepon genggam. Dengan kemampuan yang dimiliki telepon genggam untuk mengakses internet, maka para pengguna *Instant Messaging* dapat mengakses dunia maya kapan saja dan dimana saja.

Di Indonesia sendiri, pengguna *Instant Messaging* cukup banyak. Umumnya mayoritas pengguna berasal dari kaum pelajar dan mahasiswa. Menjamurnya piranti lunak seperti *mig33*, *eBuddy*, dan lain-lain memungkinkan telepon genggam untuk ber'IM' ria, membuat para penggemar *Intsant Messaging* semakin dimanja.

II.4.3 *Instant Messenger (*IM*)*

Instant Messenger (IM) atau yang biasa disebut aplikasi *chatting* merupakan perangkat lunak yang memfasilitasi pengiriman pesan singkat (*instant messaging*), suatu bentuk komunikasi secara langsung antara dua orang atau lebih menggunakan teks yang diketik. Teks dikirim melalui komputer yang terhubung melalui sebuah jaringan, misalnya internet. Setelah penggunaan *email* yang mengubah cara orang berkomunikasi dari cara konvensional untuk mengirimkan surat, teknologi pengiriman pesan singkat (*instant messaging*) diciptakan untuk menutupi kelemahan *email* yang kadang-kadang kurang cepat dan tidak waktu nyata (*real-time*).

II.4.3.1 Sejarah Perkembangan *Instant Messenger* (IM)

Perkembangan pengirim pesan instan berbasis GUI diawali ketika mulai maraknya orang menggunakan teknologi secara *online* pada awal tahun 1990 ketika banyak orang mulai meluangkan waktu untuk mengakses internet. *E-mail* merupakan perkembangan komunikasi internet yang berkembang pesat pada saat itu, namun seiring dengan bertambahnya kebutuhan, *e-mail* dirasakan kurang cepat dalam mengirimkan pesan dan tidak dapat diketahui apakah orang yang dikirimi pesan melalui *e-mail* juga sedang daring dan membalas pesan secara cepat. Kebutuhan tersebut mendorong para pengembang perangkat lunak menciptakan sebuah program *chat room*, di mana suatu grup atau perorangan dapat melihat serta mengirimkan pesan kepada sertiap orang yang ada pada *room* tersebut. Aplikasi ini dimulai ketika penyedia jasa *online Quantum Link* (lebih dikenal dengan nama *American Online*) menawarkan fitur untuk dapat saling berkirim pesan dengan sesama pengguna

komputer yang sedang *online*. Penerapan pengiriman pesan instan ini memiliki dasar yang tidak jauh berbeda, yaitu sebuah *chat room* yang ditujukan untuk dua orang.

Seiring dengan perkembangannya, pengiriman pesan instan ini mulai meledak di Internet pada November 1996 saat Mirabilis memperkenalkan ICQ yang merupakan sebuah program pengirim pesan instan gratis, dan diikuti oleh AOL *Instant Messenger* (*AIM*) pada tahun 1997. *AOL* kemudian mengakuisisi Mirabilis.

Penggunaan pengirim pesan instan kemudian berkembang dari hanya bisa diakses melalui komputer menjadi dapat diakses juga melalui telepon genggam sehinggal memungkinkan para penggunanya mengakses layanan ini kapan saja dan di mana saja.

II.4.3.2 Fitur *Instant Messenger*

Beberapa fitur yang biasanya terdapat dalam sebuah aplikasi *chatting*, diantaranya :

a. Pesan singkat (*Instant messages*): Mengirimkan pesan kepada orang yang sedang *online* secara waktu nyata.
b. Obrolan (*Chat*): Menciptakan ruang obrolan (*chat room*) atau ruang diskusi dengan beberapa orang.
c. Tautan web (*Web links*): Berbagi pranala mengenai situs web favorit.
d. Video: Berbagi video.
e. Gambar (*Images*): Berbagi gambar.
f. Berkas (*Files*): Berbagi berkas komputer.
g. Bicara (*Talk*): Berfungsi seperti layaknya telepon.
h. Fasilitas dalam genggaman (*Mobile capabilities*): Mengirimkan pesan instan melalui telepon genggam.

II.5 *Android*

Menurut Nazrudin Safaat[11], *Android* adalah sistem operasi yang berbasis *Linux* untuk telepon seluler seperti telepon pintar dan komputer tablet. *Android* menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, *Google*

Inc. membeli *Android Inc*., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan *Android*, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk *Google*, *HTC*, *Intel*, *Motorola*, *Qualcomm*, T-*Mobile*, dan *Nvidia*.

Pada saat perilisan perdana *Android*, 5 November 2007, *Android* bersama *Open Handset Alliance* menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, *Google* merilis kode–kode *Android* di bawah lisensi *Apache*, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler.

Di dunia ini terdapat dua jenis distributor sistem operasi *Android*. Pertama yang mendapat dukungan penuh dari *Google* atau *Google Mail Services* (*GMS*) dan kedua adalah yang benar–benar bebas distribusinya tanpa dukungan langsung *Google* atau dikenal sebagai *Open Handset Distribution* (OHD).

Sekitar September 2007 sebuah studi melaporkan bahwa *Google* mengajukan hak paten aplikasi telepon seluler (akhirnya *Google* mengenalkan *Nexus One*, salah satu jenis telepon pintar *GSM* yang menggunakan *Android* pada sistem operasinya. Telepon seluler ini diproduksi oleh *HTC Corporation* dan tersedia di pasaran pada 5 Januari 2010).

Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja *Android ARM Holdings*, *Atheros Communications*, diproduksi oleh *Asustek Computer Inc*, *Garmin Ltd*, *Softbank*, *Sony Ericsson*, *Toshiba Corp*, dan *Vodafone Group Plc*. Seiring pembentukan *Open Handset Alliance*, *OHA* mengumumkan produk perdana mereka, *Android*, perangkat bergerak (*mobile*) yang merupakan modifikasi kernel *Linux* 2.6. Sejak *Android* dirilis telah dilakukan berbagai pembaruan berupa perbaikan bug dan penambahan fitur baru.

Telepon pertama yang memakai sistem operasi *Android* adalah *HTC Dream*, yang dirilis pada 22 Oktober 2008. Pada penghujung tahun 2009 diperkirakan di dunia ini paling sedikit terdapat 18 jenis telepon seluler yang menggunakan *Android*.

Beberapa versi *Android* yang telah di rilis diantaranya :

1. *Android* versi 1.1

   Pada 9 Maret 2009, *Google* merilis *Android* versi 1.1. *Android* versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, *voice search* (pencarian suara), pengiriman pesan dengan *Gmail*, dan pemberitahuan *email*.

2. *Android* versi 1.5 (*Cupcake*)

   Pada pertengahan Mei 2009, *Google* kembali merilis telepon seluler dengan menggunakan *Android* dan *SDK* (*Software Development Kit*) dengan versi 1.5 (*Cupcake*). Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke *Youtube* dan gambar ke *Picasa* langsung dari telepon, dukungan *Bluetooth A2DP*, kemampuan terhubung secara otomatis ke *headset Bluetooth*, animasi layar, dan *keyboard* pada layar yang dapat disesuaikan dengan sistem.

3. *Android* versi 1.6 (*Donut*)

   Donut (versi 1.6) dirilis pada bulan September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol *applet VPN*. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus; kamera, *camcorder* dan galeri yang dintegrasikan; *CDMA* / *EVDO*, *802.1x*, VPN, *Gestures*, dan *Text-to-speech engine*, kemampuan dial kontak, teknologi *text to change speech* (tidak tersedia pada semua ponsel; pengadaan resolusi VWGA.

4. *Android* versi 2.0/2.1 (*Eclair*)

   Pada 3 Desember 2009 kembali diluncurkan ponsel *Android* dengan versi 2.0/2.1 (*Eclair*), perubahan yang dilakukan adalah pengoptimalan *hardware*, peningkatan *Google Maps 3.1.2*, perubahan UI dengan *browser* baru dan dukungan *HTML5*, daftar kontak yang baru, dukungan *flash* untuk kamera 3,2 MP, *digital Zoom*, dan *Bluetooth* 2.1. Untuk bergerak cepat dalam persaingan perangkat generasi berikut, *Google* melakukan investasi dengan mengadakan

kompetisi aplikasi *mobile* terbaik (*killer apps* - aplikasi unggulan). Kompetisi ini berhadiah $25,000 bagi setiap pengembang aplikasi terpilih. Kompetisi diadakan selama dua tahap yang tiap tahapnya dipilih 50 aplikasi terbaik. Dengan semakin berkembangnya dan semakin bertambahnya jumlah *handset Android*, semakin banyak pihak ketiga yang berminat untuk menyalurkan aplikasi mereka kepada sistem operasi *Android*. Aplikasi terkenal yang diubah ke dalam sistem operasi *Android* adalah *Shazam*, *Backgrounds*, dan *WeatherBug*. Sistem operasi *Android* dalam situs Internet juga dianggap penting untuk menciptakan aplikasi *Android* asli, contohnya oleh *MySpace* dan *Facebook*.

5. *Android* versi 2.2 (*Froyo*: *Frozen Yoghurt*)

   Pada 20 Mei 2010, *Android* versi 2.2 (*Froyo*) diluncurkan. Perubahan-perubahan umumnya terhadap versi-versi sebelumnya antara lain dukungan *Adobe Flash* 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, intergrasi V8 *JavaScript* engine yang dipakai *Google Chrome* yang mempercepat kemampuan rendering pada *browser*, pemasangan aplikasi dalam *SD Card*, kemampuan *WiFi Hotspot portabel*, dan kemampuan auto update dalam aplikasi *Android* Market.

6. *Android* versi 2.3 (*Gingerbread*)

   Pada 6 Desember 2010, *Android* versi 2.3 (*Gingerbread*) diluncurkan. Perubahan-perubahan umum yang didapat dari *Android* versi ini antara lain peningkatan kemampuan permainan (*gaming*), peningkatan fungsi *copy paste*, layar antar muka (*User Interface*) didesain ulang, dukungan format video VP8 dan *WebM*, efek audio baru (*reverb, equalization, headphone virtualization*, dan *bass boost*), dukungan kemampuan *Near Field Communication* (*NFC*), dan dukungan jumlah kamera yang lebih dari satu.

7. *Android* versi 3.0/3.1 (*Honeycomb*)

   Android Honeycomb dirancang khusus untuk *tablet*. *Android* versi ini mendukung ukuran layar yang lebih besar. *User Interface* pada *Honeycomb*

juga berbeda karena sudah didesain untuk *tablet*. *Honeycomb* juga mendukung *multi prosesor* dan juga akselerasi perangkat keras (*hardware*) untuk grafis. *Tablet* pertama yang dibuat dengan menjalankan *Honeycomb* adalah *Motorola Xoom*. Perangkat *tablet* dengan *platform Android* 3.0 akan segera hadir di Indonesia. Perangkat tersebut bernama *Eee Pad Transformer* produksi dari Asus. Rencana masuk pasar Indonesia pada Mei 2011.

8. *Android* versi 4.0 (*ICS: Ice Cream Sandwich*)

   Diumumkan pada tanggal 19 Oktober 2011, membawa fitur *Honeycomb* untuk *smartphone* dan menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu kontak jaringan sosial, perangkat tambahan fotografi, mencari *email* secara *offline*, dan berbagi informasi dengan menggunakan NFC. Ponsel pertama yang menggunakan sistem operasi ini adalah *Samsung Galaxy Nexus*.

9. *Android* versi 4.1 (*Jelly Bean*)

   Android Jelly Bean yaang diluncurkan pada acara Google I/O lalu membawa sejumlah keunggulan dan fitur baru. Penambahan baru diantaranya meningkatkan input *keyboard*, desain baru fitur pencarian, UI yang baru dan pencarian melalui *Voice Search* yang lebih cepat. Tak ketinggalan *Google Now* juga menjadi bagian yang diperbarui. *Google Now* memberikan informasi yang tepat pada waktu yang tepat pula. Salah satu kemampuannya adalah dapat mengetahui informasi cuaca, lalu-lintas, ataupun hasil pertandingan olahraga. Sistem operasi *Android Jelly Bean* 4.1 muncul pertama kali dalam produk *tablet Asus*, yakni *Google Nexus 7*.

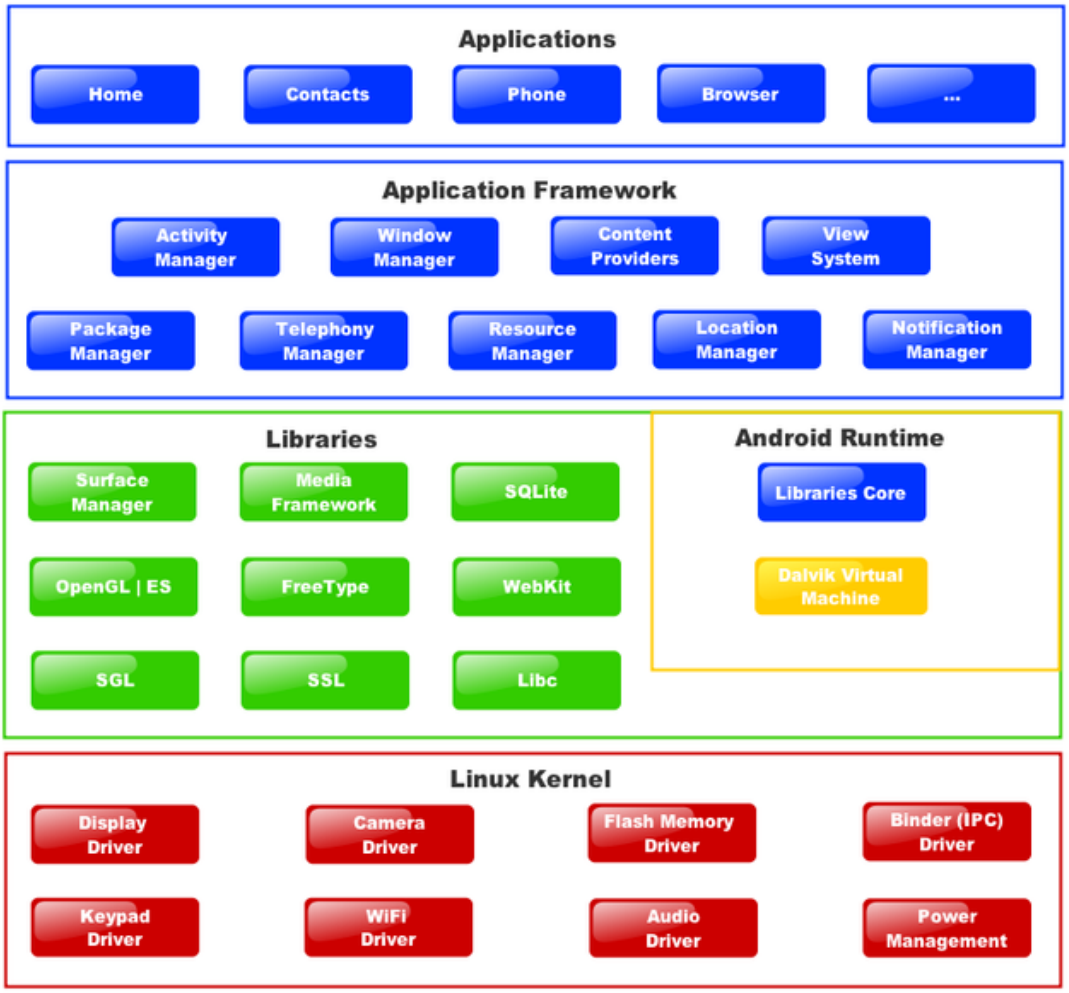
II.5.1 Fitur

Beberapa fitur yang tersedia pada *platform Android* antara lain :

1. Kerangka aplikasi: itu memungkinkan penggunaan dan penghapusan komponen yang tersedia.
2. *Dalvik mesin virtual*: mesin virtual dioptimalkan untuk perangkat telepon seluler.

3. Grafik: grafik di 2D dan grafis 3D berdasarkan pustaka *OpenGL*.
4. *SQLite*: untuk penyimpanan data.
5. Mendukung media: *audio*, *video*, dan berbagai format gambar (*MPEG4*, *H.264*, *MP3*, *AAC*, *AMR*, *JPG*, *PNG*, *GIF*)
6. *GSM*, *Bluetooth*, *EDGE*, *3G*, *4G* dan *WiFi* (tergantung piranti keras)
7. Kamera, *Global Positioning System (GPS)*, kompas, *NFC* dan *accelerometer* (tergantung piranti keras)
8. Lingkungan *Development* yang lengkap dan kaya termasuk perangkat *emulator*, *tools* untuk *debugging*, profil, dan kinerja memori, dan *plugin* untuk *Eclips IDE*
9. Market : Seperti kebanyakan *Handphone* yang memiliki tempat penjualan aplikasi, *market* pada *Android* merupakan katalog aplikasi yang dapat di *download* dan di *install* pada *Handphone* melalui internet

II.5.2 Arsitektur *Android*



Gambar II.12 Arsitektur *Android*

II.5.2.1 *Linux Kernel*

Android bukan *Linux*, akan tetapi *Android* dibangun di atas *Linux Kernel* yaitu versi 2.6 sehingga kehandalannya dapat dipercaya. Untuk inti sistem *service linux* yang digunakan seperti keamanan, manajemen memori, proses manajemen, *network*, dan *drive* model. Seperti yang terlihat digambar, *Linux Kernel* menyediakan *Driver Layer*, Kamera, *Keypad*, Kamera, *Wifi*, *Flash Memory*, Audio, dan *IPC* (*Interprocess Communication*) untuk mengatur aplikasi dan keamanan. Kernel juga bertindak sebagai lapisan abstrak antara *hardware* dan *software stacknya*

II.5.2.2 Libraries

Android menyertakan *libraries C/C++* yang digunakan oleh berbagai komponen dari sestem *Android*. Kemampuan ini disediakan kepada *Developer* aplikasi melalui *framework* aplikasi *Android*. beberapa inti *libraries* tercantum di bawah ini :

A. *System C library* – Variasi dari implementasi BSD-berasal pelaksanaan sistem standar C *library* (*libe*), sesuai untuk perangkat *embedded* berbasis *linux*.

B. *Media libraries – PaketVideo* berdasarkan *OpenCORE, library* mendukung pemutaran rekaman dan populer banyak format *audio* dan *video*, serta *file* gambar, termasuk *MPEG, H.264, MP3, AAC, AMR, JPG*. dan *PNG*

C. *Surface Manager* – mengelola akses ke subsistem layar

D. *LibWebCore* – mesin web modern yang *powerfull* yang baik *browser Android embeddable web*.

E. *SGL* – mesin grafis 2D

F. 3D *libraries* – sebuah pelaksanaan berdasarkan *openGL* ES 1,0 API; perpustakaan baik menggunakan perangkat keras akselarasi 3D (apabila tersedia) atau yang disertakan, sangat optimal 3D *software rasterizer*

G. *FreeType – bitmap* dan vektor *font rendering*

H. *SQLite* – mesin *Database* yang kuat dan ringan, dan penghubung tersedia untuk semua aplikasi.

II.5.2.3 *Android*-Runtime

Android terdiri satu *set core libraries* yang menyediakan sebagian besar fungsi yang sama dengan yang terdapat pada *core libraries* bahasa pemograman *java*. Setiap aplikasi menjalankan prosesnya sendiri dalam *Android*. Setiap alikasi menjalankan prosesnya sendiri dalam *Android*, dengan masing-masing instan dari mesin *virtual Dalvik* (*Dalvik VM*). *Dalvik* dirancang agar perangkat dapat menjalankan *multiple* VMs secara efisien. Mesin *Virtual Dalvik* mengeksekusi *file* dalam *Dalvik executable* (*.dex*), sebuah format yang dioptimalkan untuk memori yang

kecil. *Dalvik VM* berbasis, berjalan dan dikompilasi oleh *compiler* bahasa *Java* yang telah ditransformasikan ke dalam *.dex format* oleh *tool* "dx" yang telah di sertakan, *Dalvik VM* bergantung pada kernel *Linux* untuk berfungsi seperti *threading* dan manajemen *memori* tingkat rendahnya.

II.5.2.4 Framework-Aplikasi

Pengembang memiliki akses penuh menuju API *framework* yang sama, dengan yang digunakan oleh aplikasi inti. Arsitektur aplikasi dirancang agar komponen dapat digunakan kembali (*reuse*) denga mudah. Setiap aplikasi dapat memanfaatkan kemampuan ini dan aplikasi yang lain mungkin akan memanfaatkan kemampuan ini (sesuai denganbatasan keamanan yang didefinisikan oleh *framework*). Mekanisme yang sama memungkinkan komponen untuk diganti oleh pengguna. Semua pengguna merupakan rangkaian *set* layanan dan sistem, termasuk:

A. *View* yang kaya dan *extensible* yang dapat digunakan untuk membangun aplikasi, termasuk *list*, *grid*, kotak teks, tombol, dan bahkan sebuah *embeddable web*

B. *Content provider* yang memungkinkan aplikasi untuk mengakses data (seperti dari daftar kontak *help*) atau membagi data tersebut.

C. *Resource Manager*, yang menyediakan akses ke kode *non* sumber lokal seperti *string*, gambar, dan tata letak *file*

D. *Notification Manager* yang memungkinkan semua aplikasi menampilkan *alert* yang bisa dokostumisasi didalam status bar.

E. *Activity Manager* yang mengelola siklus hidup aplikasi dan menyediakan navigasi umum *backstack*.

II.5.2.5 Application

Android telah menyertakan aplikasi inti seperti *email Client*, *SMS*, kalender, peta, *browser*, kontak, dan lain-nya. Semua aplikasi tersebut ditulis dengan menggunakan bahasa pemrograman Java. Pada layer inilah *developer* atau kita

menempatkan aplikasi yang dibuat. Yang istimewa adalah pada *Android* semua aplikasi baik aplikasi inti (*native*) maupun aplikasi pihak ketiga berjalan pada layer aplikasi yang dibuat untuk *Android* akan memiliki akses yang setara dalam mengakses seluruh kemampuan *handset*, tanpa membedakan apakah itu merupakan aplikasi inti atau aplikasi pihak ketiga. Dalam kata lain dengan platform *Android* ini, Programmer atau *developer* secara penuh akan bisa mengkostumisasi perangkat *Android*nya.

II.6 Perangkat *Mobile*

Perangkat *mobile* atau biasa yang disebut ponsel merupakan perangkat telekomunikasi elektronik yang mempunyai kemampuan dasar yang sama dengan telepon konvensional saluran tetap, namun dapat dibawa ke mana-mana (*portabel*, *mobile*) dan tidak perlu disambungkan dengan jaringan telepon menggunakan kabel (nirkabel, *wireless*). Saat ini Indonesia mempunyai dua jaringan telepon nirkabel yaitu sistem *GSM* (*Global System* for *Mobile Telecommunications*) dan sistem *CDMA* (*Code Division Multiple Access*).

II.6.1 Fungsi dan Fitur

Selain berfungsi untuk melakukan dan menerima panggilan telepon, ponsel umumnya juga mempunyai fungsi pengiriman dan penerimaan pesan singkat (*short message service*, *SMS*). Ada pula penyedia jasa telepon genggam di beberapa negara yang menyediakan layanan generasi ketiga (3G) dengan menambahkan jasa *videophone*, sebagai alat pembayaran, maupun untuk televisi *online* di telepon genggam mereka. Sekarang, telepon genggam menjadi gadget yang multifungsi. Mengikuti perkembangan teknologi digital, kini ponsel juga dilengkapi dengan berbagai pilihan fitur, seperti bisa menangkap siaran radio dan televisi, perangkat lunak pemutar *audio* (MP3) dan *video*, kamera *digital*, *game*, dan layanan internet (*WAP*, *GPRS*, *3G*). Selain fitur-fitur tersebut, ponsel sekarang sudah ditanamkan fitur komputer. Jadi di ponsel tersebut, orang bisa mengubah fungsi ponsel tersebut

menjadi mini komputer. Di dunia bisnis, fitur ini sangat membantu bagi para pebisnis untuk melakukan semua pekerjaan di satu tempat dan membuat pekerjaan tersebut diselesaikan dalam waktu yang singkat.

II.6.2 Jenis Ponsel Berdasarkan Fungsi

Terdapat beberapa jenis ponsel berdasarkan fungsinya, antara lain :

1. Ponsel bisnis

   Ponsel jenis ini ditujukan untuk anda yang menginginkan perangkat bisnis dalam genggaman anda, biasanya ponsel yang telah memiliki kemampuan ini tergolong ponsel pintar (*smartphone*). Beragai aplikasi bisnis terdapat dalam ponsel ini dan dapat membuat pekerjaan kantor anda dapat dilihat dan dikerjakan dalam sebuah ponsel.

2. Ponsel hiburan

   Ponsel jenis ini merupakan ponsel berjenis *multimedia*, dimana semua aktivitas yang berhubungan dengan musik, seni, foto, sosial dan lainnya dapat anda atasi dengan sebuah ponsel. Banyak Ponsel jenis ini yang memiliki variannya tersendiri, seperti ponsel musik, ponsel kamera, dan ponsel jejaring sosial.

3. Ponsel *fashion*

   Ponsel jenis ini lebih banyak mengandalkan tampilannya, dan dapat membuat pemiliknya sangat puas meskipun dengan fitur yang terkesan sederhana. Tetapi dibalik itu semua, sebuah ponsel *fashion* dapat berharga berkali-kali lipat dari harga ponsel tercanggih. Dewasa ini dapat ditemukan ponsel yang berharga lebih mahal dari harga sebuah kendaraan, bahkan lebih mahal dari harga sebuah rumah.

4. Ponsel standar

   Ponsel jenis ini diperuntukan untuk anda yang menginginkan ponsel yang simpel, fitur yang disematkan dalam ponsel ini merupakan fitur inti, tanpa teknologi baru yang disematkan.

II.7 *Application Programming Interface* (*API*)

Application Programming Interface (*API*) atau aplikasi antarmuka pemrograman adalah metode spesifik yang ditentukan oleh sistem operasi komputer atau program aplikasi dimana seorang *programmer* menulis sebuah program aplikasi dapat membuat permintaan dari sistem operasi atau aplikasi lain. API juga merupakan kumpulan fungsi-fungsi *eksternal* yang disediakan *library windows* untuk mengatur kemampuan dan tingkah laku setiap element di *Windows* (dari tampilan di *desktop* hingga alokasi *memory*) sehingga dapat dimanfaatkan suatu program untuk meningkatkan kemampuan program tersebut. API menyediakan fungsi dan perintah dengan bahasa yang lebih terstruktur dan lebih mudah untuk dipahami oleh *programmer* bila dibandingkan dengan *System Calls*, hal ini penting untuk aspek *editing* dan pengembangan, sehingga *programer* dapat mengembangkan sistem dengan mudah. API juga dapat digunakan pada Sistem Operasi mana saja asalkan sudah ada paket-paket API nya.

Keuntungan memprogram dengan menggunakan API di antaranya :

1. Portabilitas. *Programmer* yang menggunakan API dapat menjalankan programnya dalam sistem operasi mana saja asalkan sudah ter- *install* API tersebut. Sedangkan *system call* berbeda antar sistem operasi, dengan catatan dalam implementasinya mungkin saja berbeda.
2. Lebih mudah dimengerti. API menggunakan bahasa yang lebih terstruktur dan mudah dimengerti daripada bahasa *system call*. Hal ini sangat penting dalam hal *editing* dan pengembangan.

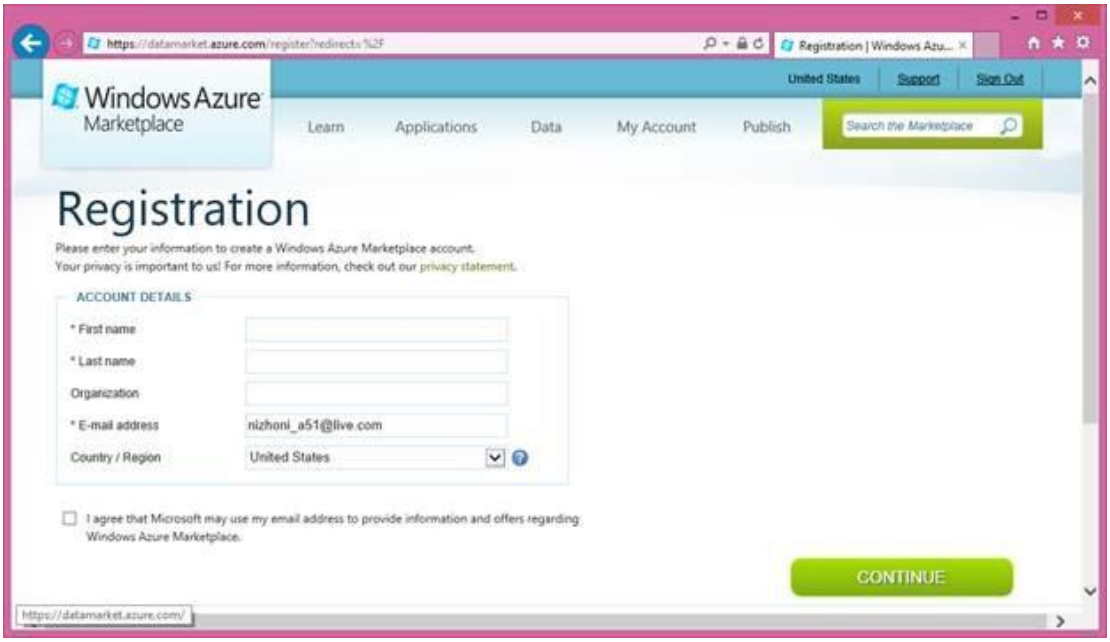
II.8 Microsoft Translator API

Microsoft translator atau *Bing Translator* adalah layanan yang disediakan oleh *Microsoft* untuk menerjemahkan teks atau seluruh halaman *web* kedalam bahasa yang berbeda. Saat ini *Microsoft Translator* menjadi alternatif yang banyak diminati oleh para *developer* mengingat Google tidak lagi menggratiskan layanan *Google Translate API*-nya.

Dengan adannya *Microsoft Translator* API, maka para pengembang (*developer*) pihak ketiga dapat memanfaatkan *Microsoft Translator* untuk mengembangkan aplikasi mereka. Berikut langkah-langkah dalam mengaktifkan *Microsoft Translator* API :

1. *Sign-In* dan *Register*

Sign-in ke *Windows Azure Marketplace*. Jika ini pertama kali, maka akan tampil form registrasi. Isi kolom yang tersedia kemudian klik "*Continue*".



Gambar II.13 Form registrasi *Windows Azure Marketplace*

Kemudian akan tampil halaman *Terms of Use*. Check pada "*I accept the Terms of Use*" dan klik "*Register*".

2. Berlangganan ke *Microsoft Translator* API

2.1 Cari layanan *Micrososft Translator* API di *Windows Azure Marketplace*

Setelah menyelesaikan Registrasi *Windows Azure Marketplace* (langkah 1), kemudian kunjungi https://datamarket.azure.com, pada kotak pencarian ketikan 'Translator' dan tekan *enter*.



Gambar 2.14 Kotak pencarian *Windows Azure Marketplace*

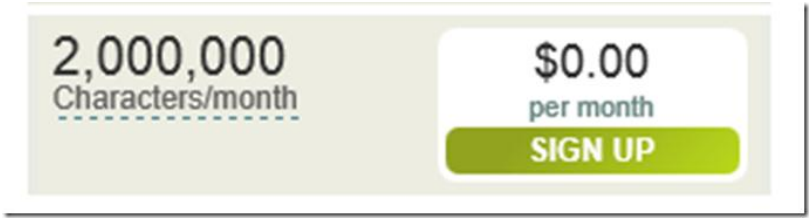
Pada hasil pencarian anda akan menemukan *Microsoft Translator API*



Gambar II.15 Hasil Pencarian

2.2 Berlangganan ke *Microsoft Translator* API

Klik *Microsoft Translator*, kemudian akan tampil halaman tawaran yang diberikan oleh *Microsoft Translator* API. Di samping kanan layar akan terlihat berbagai paket pemakaian perbulan. Pilihlah salah satu paket yang diberikan, mulai dari yang gratis ataupun yang berbayar. Pada penelitian ini penulis memilih paket gratis dengan volume pemakaian 2 juta karakter per bulan.



Gambar II.16 Paket Gratis Layanan *Microsoft Translator*

Klik '*Sign up'*.

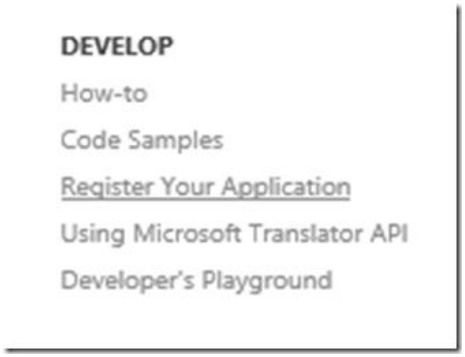
3. Mendapatkan Kridensial pengembang.

   Pada tahap ini diasumsikan telah:

   1. Menyelesaikan langkah 1 dan 2
   2. Telah terdaftar pada *Windows Azure Marketplace*
   3. Telah menggunakan *account azure marketplace* untuk berlangganan layanan *Microsoft Translator API*

3.1 Mendaftarkan Aplikasi pada *Windows Azure Marketplace*

Sign-in ke https://datamarket.azure.com/, pada bagian bawah halaman akan terlihat beberapa link. Klik '*Register Your Application'*.

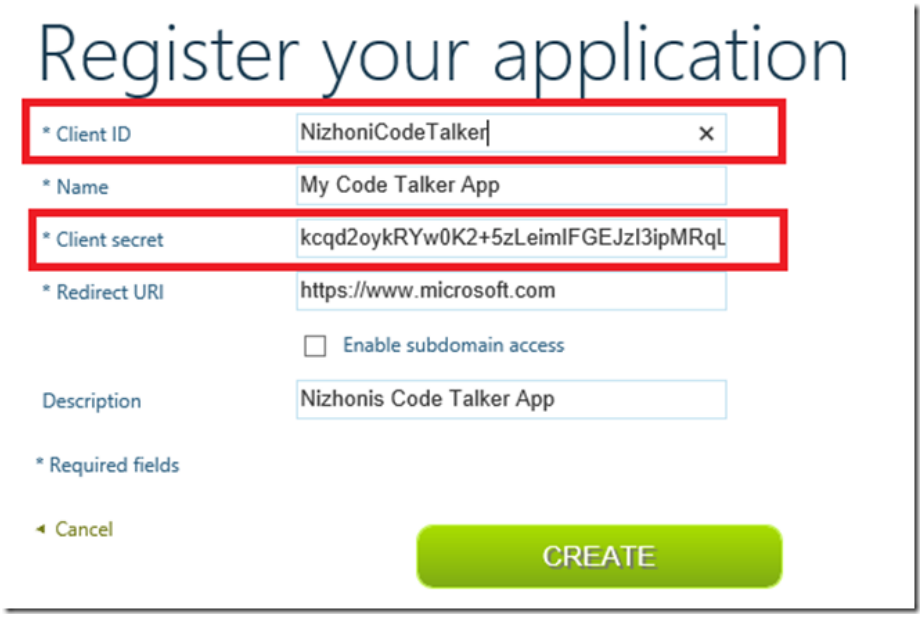


Gambar II.17 *Register Your Appplication*

3.2 Mendapatkan *Client ID* dan *Client Secret*

Client ID dan *Client Secret* digunakan untuk mengotentikasi aplikasi anda saat menggunakan layanan *Microsoft Translator API*

a) Isi *field Client ID* dan Name
b) *Field Client ID* sudah terisi secara otomatis. Jangan diubah.
c) Isi '*Redirect URL'* dengan URL https yang *valid* seperti https://microsoft.com



Gambar II.18 *Client ID* dan *Client Secret*

d) Klik '*Create'*

II.9 *Google Talk*

Google Talk (*GTalk*) adalah suatu layanan aplikasi berbasis web dari *Google* yang dibuat sebagai sarana untuk berkomunikasi melalui pesan instan dan *VoIP*. *Google talk* pertama kali dirilis pada tanggal 24 Agustus 2005. *Google Talk* disediakan secara gratis dan dapat diunduh serta diakses melalui *Google*. Layanan *Google Talk* hanya dapat digunakan apabila telah memiliki akun di *Google*.

Pesan instan yang ada di antara *server Google Talk* dan kliennya menggunakan protokol terbuka, dan XMPP, yang memungkinkan para pengguna klien XMPP lainnya untuk berkomunikasi dengan para pengguna *Google Talk*. Sementara itu, *VoIP* pada *google Talk* berbasis *Jingle protokol*. Teknologi yang dipakai dalam jaringan *server* google tidak dapat diketahui oleh publik. Layanan *Google Talk* hanya tersedia untuk sistem operasi yang menggunakan *Windows 2000, windows XP, windows Server 2003, windows Vista, dan Windows 7*. Perangkat lunak *Google Talk* tersedia juga untuk *BlackBerry, iPhone dan T-Mobile G1*. Dengan

dikeluarkannya gadget pada *Google Talk* membuat seluruh pengguna yang didukung oleh *Adobe Flash Player* dapat menggunakan *Google Talk*.

II.10 *Facebook Chat*

Facebook adalah sebuah layanan jejaring sosial yang diluncurkan pada bulan *Februari* 2004, dimiliki dan dioperasikan oleh *Facebook, Inc*. Pada September 2012, *Facebook* memiliki lebih dari satu miliar pengguna aktif, lebih dari separuhnya menggunakan telepon genggam. Pengguna harus mendaftar sebelum dapat menggunakan situs ini. Setelah itu, pengguna dapat membuat profil pribadi, menambahkan pengguna lain sebagai teman, dan bertukar pesan, termasuk pemberitahuan otomatis ketika mereka memperbarui profilnya. Selain itu, pengguna dapat bergabung dengan grup pengguna dengan ketertarikan yang sama, diurutkan berdasarkan tempat kerja, sekolah atau perguruan tinggi, atau ciri khas lainnya, dan mengelompokkan teman-teman mereka ke dalam daftar seperti "Rekan Kerja" atau "Teman Dekat".

Mulai 23 April 2008, *Facebook chat* dirilis ke seluruh pengguna *Facebook*. Pengguna dapat melakukan *chatting* dengan teman-teman *Facebook* mereka secara perorangan, atau pengguna dapat *chatting* dengan beberapa teman-teman secara simultan melalui fitur grup.

II.11 JSON

JSON (*JavaScript Object Notation*) adalah sebuah standarisasi dokumentasi yang digunakan untuk pertukaran komunikasi data secara *online*. Sama seperti XML *(eXtensible Markup Language*), keduanya adalah standar yang saat ini banyak digunakan . Standarisasi ini dibuat berdasarkan bagian dari bahasa pemprograman *JavaScript*, Standar *ECMA-262* Edisi ke-3 , Desember 1999. *JSON* merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh *programmer* keluarga C termasuk *C, C++, C#, Java, JavaScript, Perl, Python* dan lain-lain. Oleh karena sifat-sifat tersebut, menjadikan *JSON* ideal sebagai bahasa pertukaran data. Kelebihan

JSON dibandingkan *XML* adalah, *JSON* mendukung penggunaan beberapa jenis tipe data, mulai dari *String*, *Integer*, *Boolean*, *Double*, *Long* dan sebagainya. Selain itu dari sisi pengolahan (*parsing*) jauh lebih cepat dibandingkan *XML*. *JSON* terdiri dari dua struktur:

1. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel *Hash* (*hash table*), daftar berkunci (*keyed list*), atau *Associative Array*.
2. Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

II.11.1 Object pada JSON

Cara lama untuk membuat sebuah *object* adalah dengan menggunakan kata-kunci *new*.

```
var myJSON = new Object();
```

Metode ini telah diperbaiki. Sekarang untuk membuat *object* kosong, dapat digunakan dengan hanya mendefinisikan pasangan kurungkurawal.

```
var myJSON = {};
```

Seperti halnya kebanyakan *object* dasar *Javascript*, format data sangatlah fleksibel yang ditulis sebagai pasangan nama dan nilai. *Object* memiliki nama yang terkandung dalam *property object* yang biasanya disebut sebagai nama *object*. *Object* juga memiliki *property* nilai. Lihat contoh berikut:

```
var myFirstJSON = { "firstName" : "Firdaus ",
                    "lastName"  : "Musyafi",
                    "age"       :  23};
document.writeln(myFirstJSON.firstName); //Output Firdaus
document.writeln(myFirstJSON.lastName); //Output Musyafi
document.writeln(myFirstJSON.age);      // Output 22
```

Gambar II.19 Contoh Objek JSON

Penjelasan : *object* tersebut memiliki tiga *property* atau pasangan nama dan nilai. Nama adalah sebuah *string* (pada contoh di atas *firstName*, *lastName* dan *age*). Nilai dapat diisi dengan semua jenis *object Javascript* (semua di *Javascript* adalah *object*, nilai dapat diisi dengan *string*, angka, *array*, fungsi dan *object* lainnya). Pada contoh di atas nilai diisi dengan Firdaus, Musyafi dan 23. Firdaus dan Musyafi adalah *string* dan 23 adalah angka. Seperti anda lihat pada contoh, nilai dapat diisi dengan berbagai macam *object*.

JSON juga dapat diakses dengan cara seperti *associative array* :

```
var myFirstJSON = { "firstName"  : "Firdaus",
                    "lastName"   : "Musyafi",
                    "age"        : 23 };
document.writeln(myFirstJSON["firstName"]);        //Output
Firdaus
document.writeln(myFirstJSON["lastName"]);     //     Output
Musyafi
document.writeln(myFirstJSON["age"]); // Output 23
```

Gambar II.20 Akes *JSON* Menggunakan *Associative Array*

Jika *object* mengandung *object* lagi di dalamnya, tambahkan titik dan nama di belakangnya.

```
var myObject = {'color' : 'blue',
                'animal' : { 'cat' : 'friendly' } };
document.writeln(myObject.animal.cat); // output friendly
```

Gambar II.21 Akses *Object JSON*

II.12 *Web Service*

Web service adalah suatu sistem perangkat lunak yang dirancang untuk mendukung interoperabilitas dan interaksi antar sistem pada suatu jaringan. *Web service* digunakan sebagai suatu fasilitas yang disediakan oleh suatu *web site* untuk menyediakan layanan (dalam bentuk informasi) kepada sistem lain, sehingga sistem lain dapat berinteraksi dengan sistem tersebut melalui layanan-layanan (*service*) yang

disediakan oleh suatu sistem yang menyediakan *web service*. *Web service* menyimpan data informasi dalam format *XML*, sehingga data ini dapat diakses oleh sistem lain walaupun berbeda *platform*, sistem operasi, maupun bahasa *compiler*.

Web service bertujuan untuk meningkatkan kolaborasi antar pemrogram dan perusahaan, yang memungkinkan sebuah fungsi di dalam *Web Service* dapat dipinjam oleh aplikasi lain tanpa perlu mengetahui detil pemrograman yang terdapat di dalamnya.

Beberapa alasan mengapa digunakannya *web service* adalah sebagai berikut:

1. *Web service* dapat digunakan untuk mentransformasikan satu atau beberapa bisnis *logic* atau *class* dan objek yang terpisah dalam satu ruang lingkup yang menjadi satu, sehingga tingkat keamanan dapat ditangani dengan baik.
2. *Web service* memiliki kemudahan dalam proses *deployment*-nya, karena tidak memerlukan registrasi khusus ke dalam suatu sistem operasi. *Web service* cukup di-*upload* ke *web server* dan siap diakses oleh pihak-pihak yang telah diberikan otorisasi.
3. *Web service* berjalan di *port* 80 yang merupakan protokol standar HTTP, dengan demikian *web service* tidak memerlukan konfigurasi khusus di sisi *firewall*.

II.12.1 Arsitektur *Web Service*

Web service memiliki tiga entitas dalam arsitekturnya, yaitu:

1. *Service Requester* (peminta layanan)

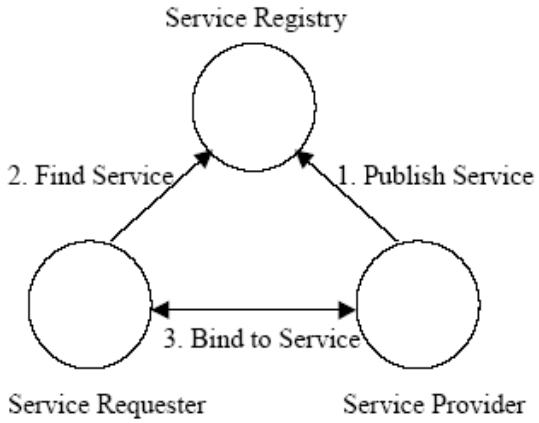
Berfungsi untuk menyediakan layanan/*service* dan mengolah sebuah *registry* agar layanan-layanan tersebut dapat tersedia.

2. *Service Provider* (penyedia layanan)

Berfungsi sebagai lokasi *central* yang mendeskripsikan semua layanan/*service* yang telah di-*register*.

3. *Service Registry* (daftar layanan)

Peminta layanan yang mencari dan menemukan layanan yang dibutuhkan serta menggunakan layanan tersebut



Gambar II.22 Arsitektur *Web Service*

II.12.2 Operasi-Operasi *Web Service*

Secara umum, *web service* memiliki tiga operasi yang terlibat di dalamnya, yaitu:

1.*Publish/Unpublish:* Menerbitkan/menghapus layanan ke dalam atau dari *registry*.

2.*Find*: *Service requestor* mencari dan menemukan layanan yang dibutuhkan.

3.*Bind*: *Service requestor* setelah menemukan layanan yang dicarinya, kemudian melakukan *binding* ke *service provider* untuk melakukan interaksi dan mengakses layanan/*service* yang disediakan oleh *service provider*.

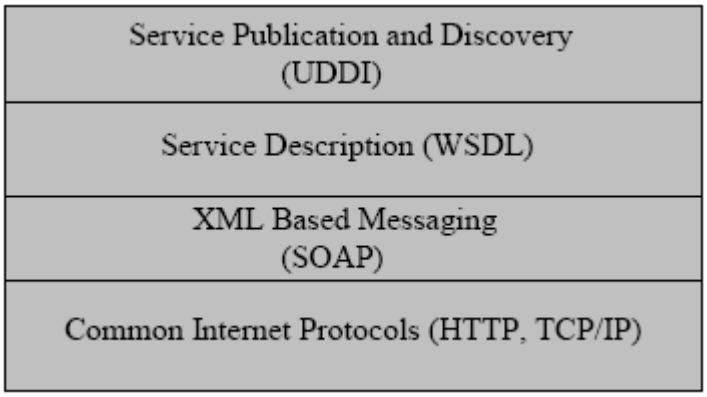
II.12.3 Komponen-Komponen *Web Service*

Web service secara keseluruhan memiliki empat layer komponen, yaitu:

1. Layer 1: Protokol internet standar seperti *HTTP*, *TCP*/*IP*

2. Layer 2: *Simple Object Access Protocol* (*SOAP*), merupakan protokol akses objek berbasis *XML* yang digunakan untuk proses pertukaran data/informasi antar layanan.

3. Layer 3: *Web Service Definition Language* (*WSDL*), merupakan suatu standar bahasa dalam format *XML* yang berfungsi untuk mendeskripsikan seluruh layanan yang tersedia.



Gambar II.23 Komponen *Web Service*

II.13 XMPP

Extensible Messaging dan Presence Protocol (*XMPP*) adalah protokol komunikasi untuk orientasi pesan *middleware* berbasis *XML (Extensible Markup Language*). Protokol ini awalnya bernama *Jabber*, dan dikembangkan oleh komunitas *open source Jabber* di tahun 1999 untuk *instant messaging* (*IM),* informasi keberadaan, dan pemeliharaan daftar kontak. Dirancang untuk diperluas, protokol juga telah digunakan untuk sistem mempublikasikan-berlangganan, sinyal untuk VoIP, *video*, *transfer file*, *game*, Aplikasi Internet seperti *smart grid*, dan layanan jejaring sosial.

Tidak seperti kebanyakan protokol *instant messaging*, *XMPP* didefinisikan dalam standar terbuka dan menggunakan pendekatan sistem terbuka pengembangan dan aplikasi, dimana siapa pun dapat menerapkan layanan *XMPP* dan beroperasi dengan implementasi organisasi lain. Karena *XMPP* adalah sebuah protokol terbuka, implementasi dapat dikembangkan dengan menggunakan lisensi perangkat lunak, meskipun banyak *server*, klien, dan implementasi *Library* didistribusikan sebagai perangkat lunak bebas dan *open source*, banyak implementasi perangkat lunak *freeware* dan juga yang komersial.

II.14 *SASL*

Simple Authentication and Security Layer (*SASL*) adalah suatu kerangka kerja untuk otentikasi dan keamanan data dalam protokol Internet. Ini memisahkan mekanisme otentikasi dari protokol aplikasi, dalam teori memungkinkan setiap mekanisme otentikasi yang didukung oleh *SASL* untuk digunakan dalam setiap protokol aplikasi yang menggunakan *SASL*. Mekanisme otentikasi juga dapat mendukung otorisasi *proxy*, fasilitas yang memungkinkan satu user untuk mengasumsikan identitas lain. Mereka juga dapat menyediakan keamanan data lapisan menawarkan integritas data dan layanan kerahasiaan data. *DIGEST-MD5* memberikan contoh mekanisme yang dapat memberikan lapisan data-keamanan. Protokol aplikasi yang mendukung *SASL* biasanya juga mendukung *Transport Layer Security (TLS)* untuk melengkapi layanan yang ditawarkan oleh *SASL*.

Sebuah mekanisme *SASL* menerapkan serangkaian tantangan dan tanggapan. Mekanisme *SASL* yang telah tetapkan meliputi:

a) "*EKSTERNAL*", di mana otentikasi implisit dalam konteks (misalnya, untuk protokol sudah menggunakan *IPsec* atau TLS)
b) "*ANONYMOUS*", untuk akses tamu yang tidak otentik
c) "*PLAIN*", mekanisme sandi *cleartext* sederhana.
d) "*OTP*", mekanisme *password* satu kali. *OTP* usang Mekanisme *SKEY*.
e) "*SKEY*", mekanisme S / KEY.
f) "*CRAM-MD5*", skema *challenge-response* sederhana berdasarkan *HMAC-MD5*.
g) "*DIGEST-MD5*", sebagian *HTTP Digest* skema *challenge-response* yang kompatibel berdasarkan *MD5*. *DIGEST-MD5* menawarkan lapisan keamanan data.
h) "*SCRAM*" tantangan-respon skema mekanisme berbasis modern dengan dukungan mengikat saluran
i) "*NTLM*", LAN mekanisme otentikasi Manajer NT

j) "*GSSAPI*", untuk *Kerberos* V5 otentikasi melalui *GSSAPI*. *GSSAPI* menawarkan lapisan data-keamanan.
k) "*BrowserID-AES128*", untuk *Mozilla Persona* otentikasi.
l) "*EAP-AES128*", untuk GSS otentikasi EAP.
m) *Gatekeeper* (& *GateKeeperPassport*), mekanisme tantangan-respon yang dikembangkan oleh *Microsoft* untuk MSN obrolan

II.15 *Unfied Modelling Language* (*UML*)

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 piranti 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 tiga 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*).

UML menyediakan 10 macam diagram untuk memodelkan aplikasi berorientasi objek, yaitu:

1. *Use Case Diagram* untuk memodelkan proses bisnis.

2. *Conceptual Diagram* untuk memodelkan konsep-konsep yang ada di dalam aplikasi.

3. *Sequence Diagram* untuk memodelkan pengiriman pesan (*message*) antar objek.

4. *Collaboration Diagram* untuk memodelkan interaksi antar objek.

5. *State Diagram* untuk memodelkan perilaku objek di dalam sistem.

6. *Activity Diagram* untuk memodelkan perilaku userdan objek di dalam sistem.

7. *Class Diagram* untuk memodelkan struktur kelas.

8. *Objek Diagram* untuk memodelkan struktur objek.

9. *Component Diagram* untuk memodelkan komponen objek.

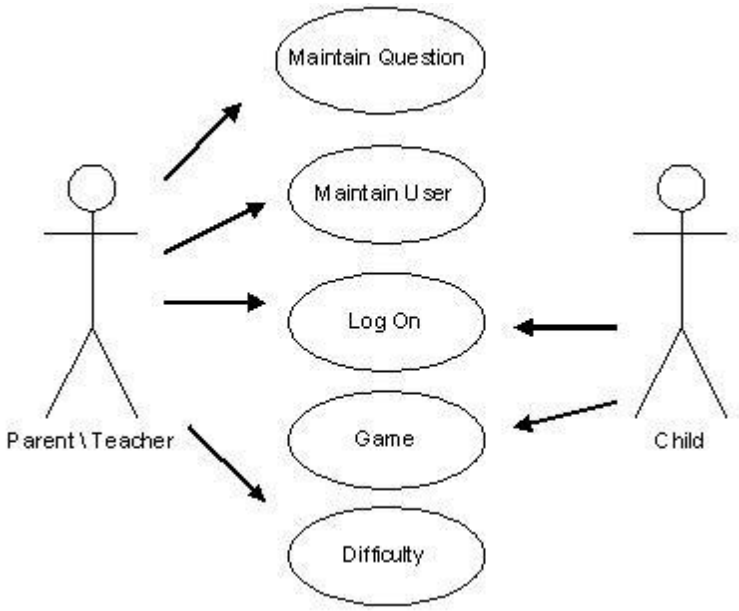
10. *Deployment Diagram* untuk memodelkan distribusi aplikasi.

Berikut akan dijelaskan 4 macam diagram yang paling sering digunakan dalam pembangunan aplikasi berorientasi objek, yaitu *use case diagram, sequence diagram, collaboration diagram,* dan *class diagram*.

A. *Use Case Diagram*

Use case diagram digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna sistem. *Use case diagram* terdiri atas diagram untuk *use case* dan *actor*. *Actor* merepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi.

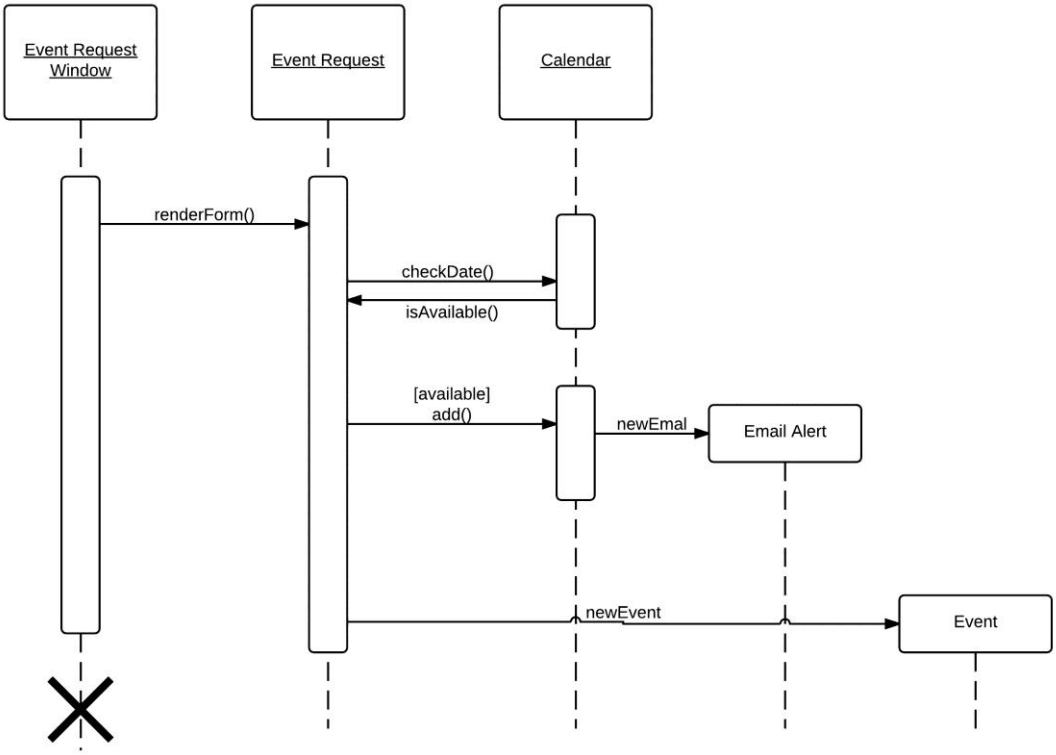
Use Case merepresentasikan operasi-operasi yang dilakukan oleh *actor*. *Use Ccase* digambarkan berbentuk *elips* dengan nama operasi dituliskan di dalamnya. *Actor* yang melakukan operasi dihubungkan dengan garis lurus ke *Use Case*.



Gambar II.24 ***Use Case Diagram***

B. *Sequence Diagram*

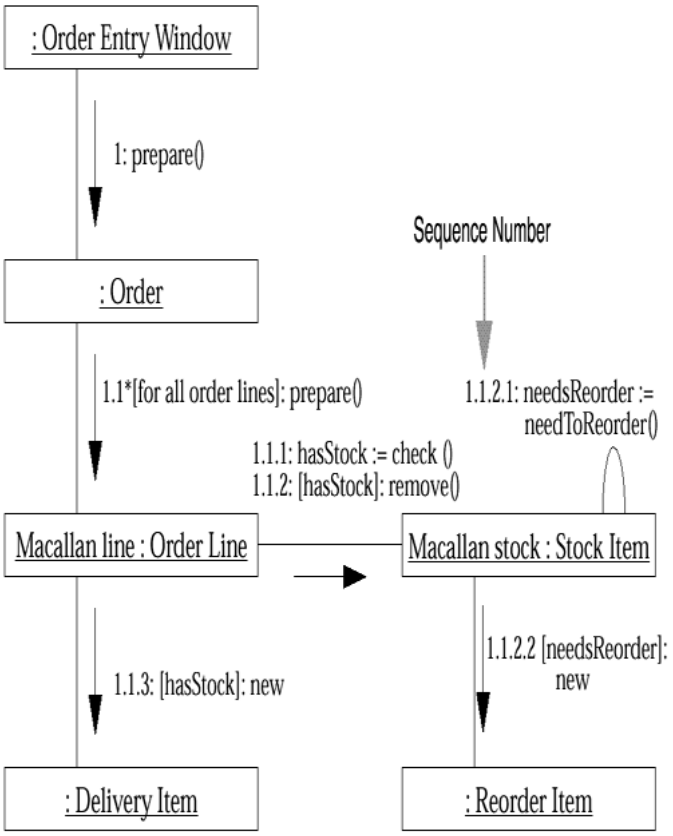
Sequence diagram menjelaskan secara detil urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari *use case*. Interaksi yang terjadi antar *class*, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi.



Gambar II.25 *Sequence Diagram*

C. *Collaboration Diagram*

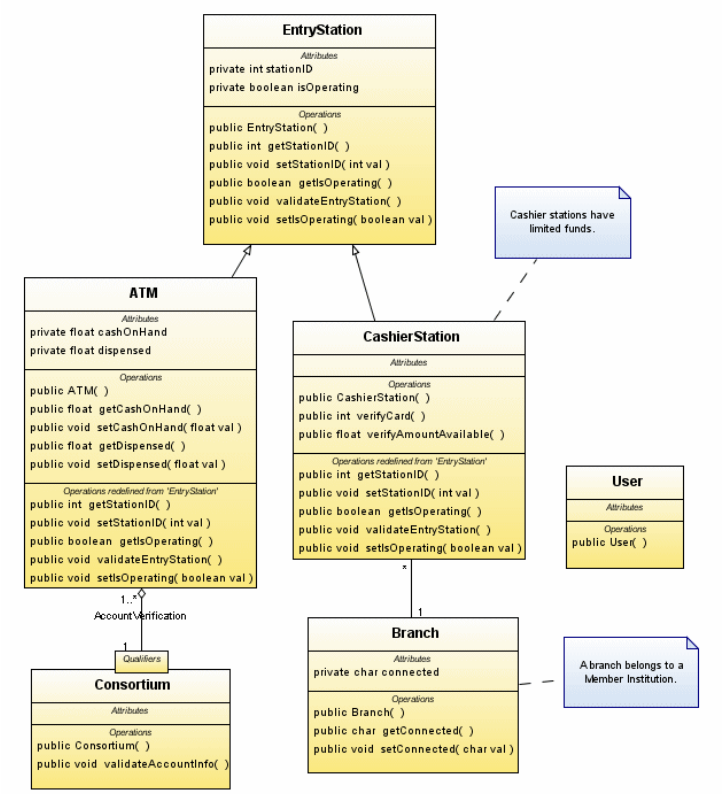
Collaboration diagram dipakai untuk memodelkan interaksi antar objek di dalam sistem. Berbeda dengan *sequence diagram* yang lebih menonjolkan kronologis dari operasi-operasi yang dilakukan, *collaboration* diagram lebih fokus pada pemahaman atas keseluruhan operasi yang dilakukan oleh objek.



Gambar II.26 *Collaboration Diagram*

D. *Class Diagram*

Class diagram merupakan diagram yang selalu ada di permodelan sistem berorientasi objek. *Class diagram* menunjukkan hubungan antar *class* dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan.



Gambar II.27 *Class Diagram*

II.16 *Tools* Implementasi

Tools implementasi merupakan peralatan-peralatan yang dibutuhkan dalam membangun perangkat lunak. Adapun *tools* implementasi yang digunakan untuk membuat aplikasi *chatting* dengan penerjemah otomatis adalah JDK sebagai perangkat pembangun *Java*, *Eclipse IDE* sebagai lingkungan pengembangan perangkat lunak, dan SQL sebagai sistem manajemen basis data.

II.16.1 JDK

JDK (*Java Development Kit*) adalah sebuah perangkat peralatan yang digunakan untuk membangun perangkat lunak dengan menggunakan bahasa pemrograman Java. JDK berjalan diatas sebuah *virtual machine* yang dinamakan JVM (*Java Virtual Machine*).

Dokumentasi JDK berisi spesifikasi API, deskripsi fitur, panduan pengembang, referensi halaman untuk perkakas JDK dan utilitas, demo, dan *link* ke informasi terkait. Dokumentasi ini juga tersedia dalam bundel yang dapat diunduh dan diinstal pada komputer.

II.16.2 *Eclipse IDE*

Eclipse adalah sebuat komunitas proyek *open source* yang difokuskan pada membangun sebuah *platform* pengembangan yang dapat diperluas, *runtime* dan kerangka aplikasi untuk membangun, menyebarkan dan mengelola perangkat lunak di seluruh daur hidup perangkat lunak.

Komunitas *Eclipse* memiliki lebih dari 60 proyek *open source*. Proyek-proyek ini secara konsep terbagi menjadi 7 kategori :

1. *Enterprise Development*
2. *Embedded and Device Development*
3. *Rich Client Platform*
4. *Rich Internet Applications*
5. *Application Frameworks*
6. *Application Lifecycle Management* (ALM)
7. *Service Oriented Architecture* (SOA)

Komunitas *Eclipse* juga didukung oleh ekosistem besar dan dinamis dari penyedia solusi teknologi informasi utama, *start-up inovatif*, universitas dan lembaga penelitian, dan individu yang memperluas, mendukung, dan melengkapi *platform Eclipse.*

II.16.3 *SQLite*

SQLite merupakan sebuah perpustakaan proses yang menerapkan kemandirian, tanpa *server*, tidak adanya konfigurasi mesin, dan basis data SQL transaksional. Kode untuk *SQLite* berada dalam domain publik dengan demikian bebas untuk digunakan untuk tujuan apapun, komersial atau swasta.

SQLite adalah mesin basis data *SQL* yang tertanam. Tidak seperti kebanyakan basis data *SQL* lainnya, *SQLite* tidak memiliki proses *server* yang terpisah. *SQLite* membaca dan menulis secara langsung ke *file disk* biasa. Basis data *SQL* lengkap dengan berbagai *tabel, indeks, triggers,* dan *views* terkandung dalam sebuah *file disk* tunggal.