BAB II

LANDASAN TEORI

2.1 Telepon Selular

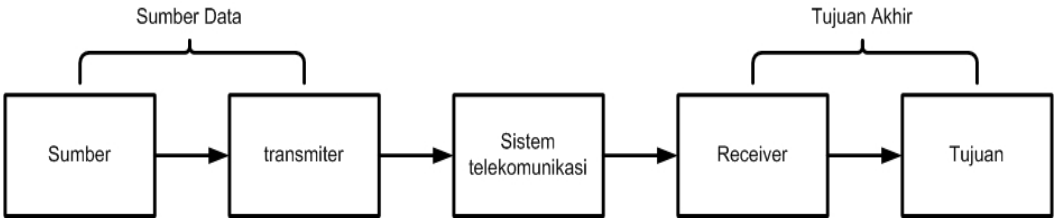
Handphone atau dengan nama lain telepon selular merupakan sebuah perangkat telekomunikasi elektronik yang mempunyai kemampuan dasar yang sama dengan telepon *fixed line* yang konvensional, namun dapat dibawa ke mana-mana (*portable*) dan tidak perlu disambungkan dengan jaringan telepon menggunakan kabel (nirkabel, wireless). Indonesia mempunyai dua jaringan telepon selular nirkabel saat ini, yaitu *Global System for Mobile Telecommunication* (GSM) dan *Code Division Multi Access* (CDMA).

Telepon selular selain berfungsi untuk melakukan dan menerima panggilan, umumnya juga mempunya fungsi pengiriman dan penerimaan pesan singkat atau *Short Message Service* (SMS). Telepon selular juga beredar di pasaran dengan harga yang cukup tinggi karena adanya tambahan fitur, seperti kamera, layanan *internet*, *Bluetooth* dan lain sebagainya. Selain itu juga terdapat *manufacture* telepon selular di beberapa negara yang menyiadakan layanan generasi ketiga (3G) yang menambahkan jasa *video call* maupun televisi *online* di ponsel mereka.

Selain kelebihan tersebut, kebanyakan Vendor telepon selular telah menyertakan fasilitas *Java Enable* pada produknya. Dengan adanya fitur *Java Enable* pada telepon selular, memungkinkan pengguna telepon selular tersebut semakin mudah untuk menambahkan sendiri aplikasi-aplikasi berbasisi Java 2 Micro Edition (J2ME).

2.2 Model Komunikasi

Model sederhana dari komunikasi, yang digambarkan pada diagram blok di bawah ini. Contoh lain adalah pertukaran signal suara antara dua telepon melalui jaringan yang sama. Elemen-elemen penting dari model adalah sebagai berikut :



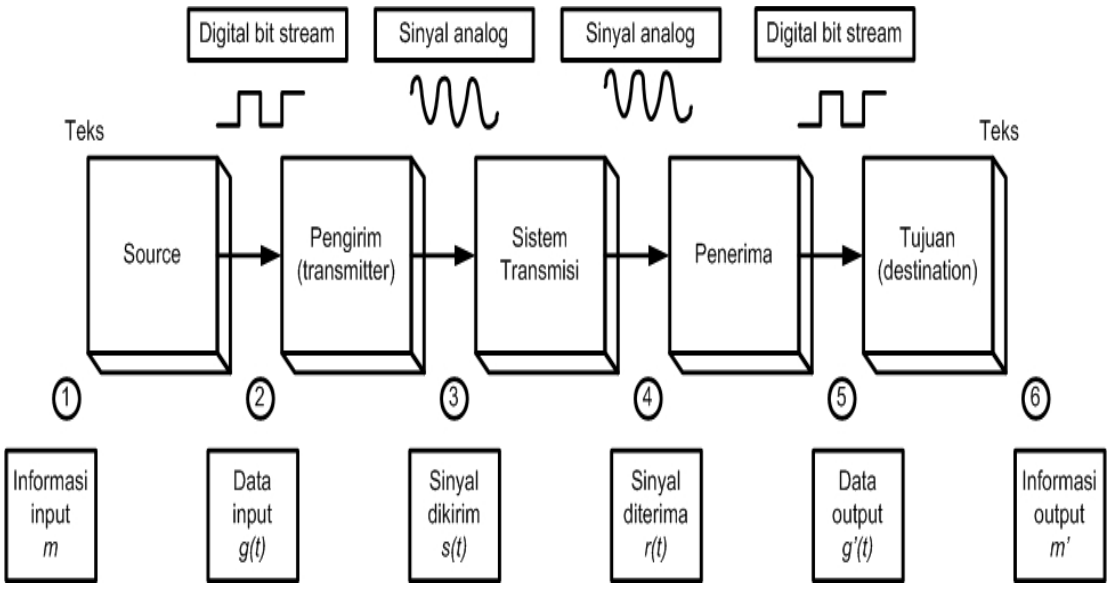
Gambar 2.1 Model Komunikasi secara Umum

Berikut keterangan dari gambar di atas :

1. *Source* (sumber) : alat ini menghasilkan data untuk ditransmisikan/dikirim.
2. *Transmitter* (pengirim) : biasanya, data dihasilkan oleh sistem sumber tidak dikirim secara langsung dalam bentuk yang telah hasilkan. *Transmitter* mengubah atau mengkodekan informasi sama seperti memproduksi signal elektromagnetik yang akan ditransmisikan secara berurutan.
3. *Transmission system* (sistem transmisi) : dapat berupa transmisi satu arah (sistem radio) atau dua arah (jaringan telkom).
4. *Receiver* (penerima) : menerima signal dari sistem transmisi dan mengubahnya kedalam bentuk yang dapat dimengerti oleh alat tujuan.
5. *Destination* (tujuan) : penerima data yang datang dari alat penerima (receiver).

2.3 Komunikasi Data

Konsep komunikasi membahas aspek dasar yang memfokuskan pada transmisi signal agar efektif dan efisien dari sumber ke tujuan.



Gambar 2.2 Penyederhanaan Konsep Komunikasi Data

Pada gambar di atas menerangkan bahwa, pada saat alat input (*source*) dan *transmitter* akan mengirimkan data kepada pengguna lain, dimana isi datanya berupa teks. Dan hasil input data dari teks yang telah di ketik berupa signal digital yang diteruskan ke memori dalam rentetan bit. Kemudian data tersebut dikirimkan ke salaluran transmisi melalui *transmitter*. Maka data bit dalam bentuk digital akan diubah ke bentuk analog oleh *transmitter*.

2.3.1 Transmisi Data

2.3.1.1 Konsep dan Istilah

Berikut terdapat beberapa konsep dan istilah dari transmisi data, antara lain :

1. Teminologi Transmisi

1) Media transmisi

Media transmisi dapat diklasifikasikan atas *guided* dan *unguided*. Signal yang telah merambat dikedua media tersebut umumnya telah dilakukan perubahan sehingga menjadi gelombang elektromagnetik. Pada media *guided*, gelombang sepenuhnya dikendalikan sepanjang media tersebut ada. Sedangkan *unguided* media dipakai untuk menyalurkan gelombang elektromagnetik tapi tidak mengendalikan mereka, contoh : perambatan melalui udara, ruang hampa dan air laut.

2) Direct link

Istilah *direct link* (hubungan langsung) terjadi jika signal yang bergerak langsung terjadi antara sumber ke tujuan tanpa perantara. Jika jarak kedua peralatan jauh dapat digunakan penguat signal yaitu *repeater* untuk signal digital atau *amplifier* untuk signal analog. *Direct link* biasanya terjadi pada media *guided* secara *point to point* pada medium yang sama.

3) Arah transmisi

Arah perambatan signal, dapat berupa *simplex*, *half duplex* atau *full duplex*. Pada transmisi *simplex*, signal hanya ditransmit satu arah saja dimana satu

station sebagai *transmiter* dan yang lainnya sebagai penerima. Arah transmisi *simplex* ini disebut juga dengan istilah *one way transmission*, dimana dengan *simplex* ini transmisi hanya bisa dilakukan satu arah saja. Contoh penggunaan teknologi ini adalah dalam proses pengiriman signal yang dilakukan oleh stasion radio atau televisi. Dengan teknologi arah transmisi *simplex* ini penerima hanya bersifat pasif dan tidak bisa memberikan respon terhadap siapa yang mengirimkan signal yang dia terima.



Gambar 2.3 Transmisi *Simplex*

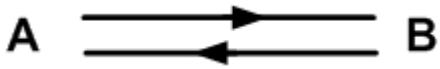
Dalam operasi *half duplex*, kedua station dapat melakukan transmisi, tapi tidak bisa bersamaan dan hanya sekali dalam suatu waktu. Disebut juga dengan istilah *either way transmission*. Transmisi bisa dilakukan dalam bentuk dua arah, namun tidak bisa dilakukan secara bersamaan. Fungsi mengirim dan menerima dari dua unit perangkat harus dilakukan secara bergantian. *Walkie-talkie* adalah contoh *half duplex* ini. Bila seseorang berbicara, dia tidak dapat berbicara sambil mendengarkan.



Gambar 2.4 Transmisi *Half Duplex*

Sistem *full duplex* ini dapat terjadi dengan menggunakan sebuah saluran komunikasi data atau dengan menggunakan dua saluran komunikasi data. *Full duplex* disebut juga dengan *both way transmission*. Merupakan komunikasi

dua arah yang dapat dilakukan secara bersamaan. Pada saat komunikasi terjadi, masing-masing unit yang berkomunikasi dapat melakukan pengiriman dan penerimaan data sekaligus. Contoh penggunaan teknologi dengan *full duplex* ini adalah pada telepon.



Gambar 2.5 Transmisi *Full Duplex*

Terminologi *simplex*, *half duplex* dan *full duplex* pada komunikasi data agak berbeda dengan sistem komunikasi radio. Pada sistem komunikasi radio digunakan istilah *simplex*, *semi duplex* dan *duplex*.

Simplex pada sistem komunikasi radio, merupakan sistem penyampaian informasi dengan menggunakan satu frekuensi kerja, namun arah aliran informasi tersebut dapat terjadi ke dua arah pada saat yang tidak bersamaan.

Semi duplex pada sistem komunikasi radio, merupakan sistem penyampaian informasi dengan menggunakan dua frekuensi kerja, masing-masing untuk *transmitter* dan *receiver*. Arah aliran informasi dapat terjadi ke dua arah pada saat yang tidak bersamaan.

Duplex pada sistem komunikasi radio, merupakan sistem penyampaian informasi dengan menggunakan dua frekuensi kerja, masing-masing untuk *transmitter* dan *receiver*. Arah aliran informasi dapat terjadi ke dua arah pada saat yang bersamaan.

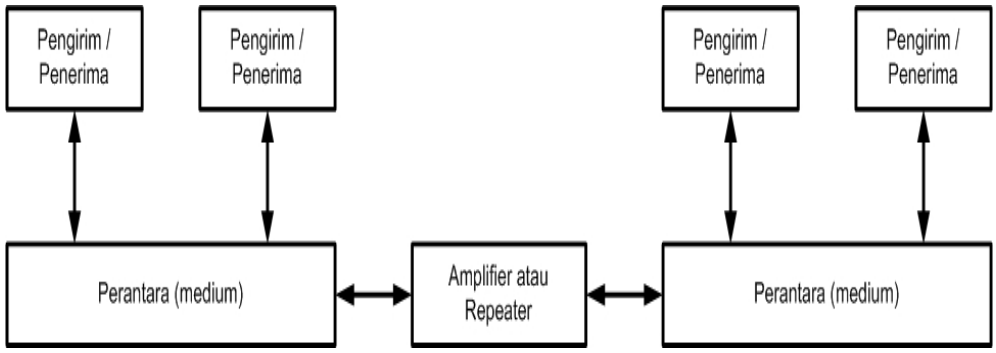
2. Konfigurasi Transmisi Data

Hubungan komunikasi data yang terjadi hanya ada dua macam yaitu, *point to point* dan *multipoint*.

Pada konfigurasi *point to point*, maka hanya ada dua perangkat yang berhubungan, dimana masing-masing beperan sebagai *transmitter* (pengirim) dan *receiver* (penerima). Sehingga media komunikasi dipakai hanya untuk kedua sistem perangkat tersebut. Lain halnya pada sistem yang tersambung dengan konfigurasi *multipoint*. Disini ada lebih dari dua perangkat yang disambung ke media komunikasi sehingga satu orang bicara maka semua yang terhubung ke media tersebut akan menerima informasinya.



Gambar 2.6 *Point to Point* pada Konfigurasi Media Transmisi



Gambar 2.7 *Multipoint* pada Konfigurasi Media Transmisi

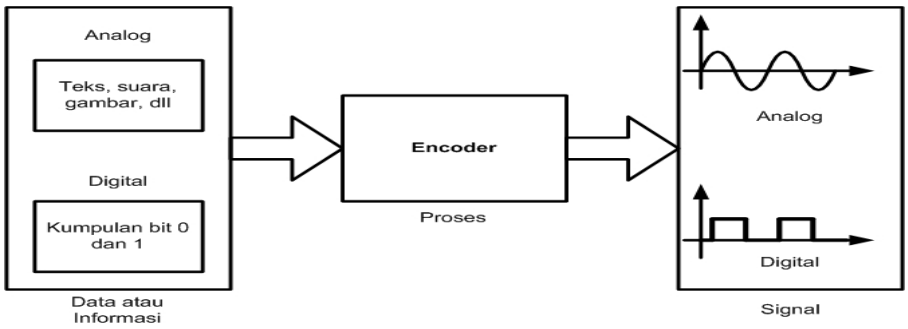
2.3.1.2 Pengiriman Data Analog dan Digital

1. Data

Data didefinisikan sebagai berikut : "Data adalah kenyataan atau fakta penting yang tercatat atau tercantum yang dapat diproses dan diinterpretasikan oleh komputer, telepon seluler, manusia dan lain sebagainya serta mempunyai arti yang bermacam-macam." Jadi segala fakta yang dicatat atau telah direkam dapat dikategorikan sebagai data. Ada dua jenis data yang dikenal saat ini, yaitu data analog (musik yang terekam di kaset, *graphic*, *video* yang terekam, dan lain-lain) dan data digital (bit 0,1 yang terekam di *harddisk*, memori).

2. Signal

Signal didefinisikan sebagai : "data atau informasi yang telah mengalami suatu proses sedemikian rupa sehingga dapat bergerak didalam saluran transmisi." Jadi jelas data yang diproses sehingga bisa pindah dari satu tempat ke tempat yang lain disebut signal. Signal dapat merambat dan besaran amplitudonya bervariasi terhadap fungsi waktu.

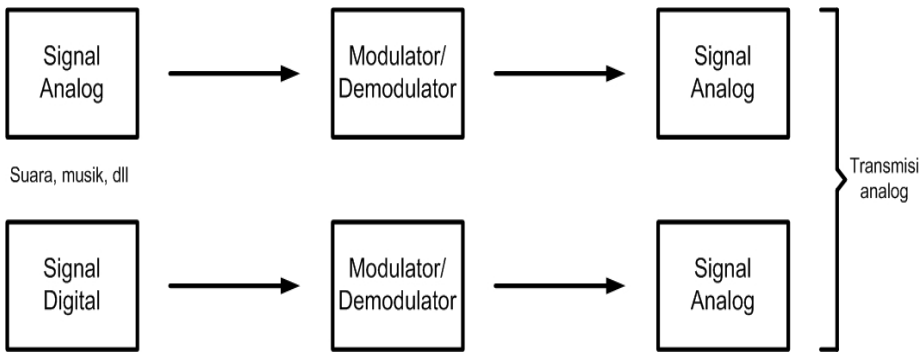


Gambar 2.8 Perubahan Data atau Informasi menjadi Signal

Ada dua jenis signal dasar yang dikenal saat ini, yaitu signal analog (musik yang terdengar, suara yang terucapkan keluar dan terdengar oleh telinga) dan signal digital (gelombang pulsa). Signal analog digunakan untuk mewakili data digital. Pada umumnya data analg berfungsi sebagai penunjuk waktu dan penetapan batas maksimal pemakaian frekuensi secara berkesamaan, seperti data yang dapat dijabarkan dengan signal elektromegnetik untuk keseluruhannya. Data digital dapat diterjemahkan dengan signal digital, dengan voltase yang berbeda untuk setiap dua digit binary.

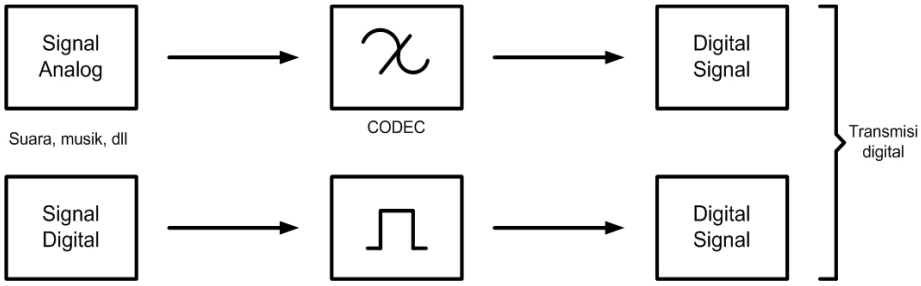
3. Transmisi

Signal jika ingin dirambatkan dalam saluran transmisi, perlu disesuaikan dengan karakteristik dan kondisi saluran yang akan dilewatinya. Pada transmisi analog berarti signal yang merambat pada saluran transmisi adalah analog, untuk itu ilustrasi pada gambar 2.9. Data digital dapat diterjemahkan menjadi signal analog dengan menggunakan *modulator/demodulator*.



Gambar 2.9 Data Analog dan Digital Ditransmisikan pada Saluran Analog

Dalam sebuah operasi yang hampir sama dengan operrasi di atas, data analog dapat ditransmisikan kedalam saluran digital. Alat yang dapat menyajikan fungsi tersebut adalah ‘codec’ (*Code Decoder*).



Gambar 2.10 Signal Analog dan Digital Ditransmisikan pada Saluran Digital

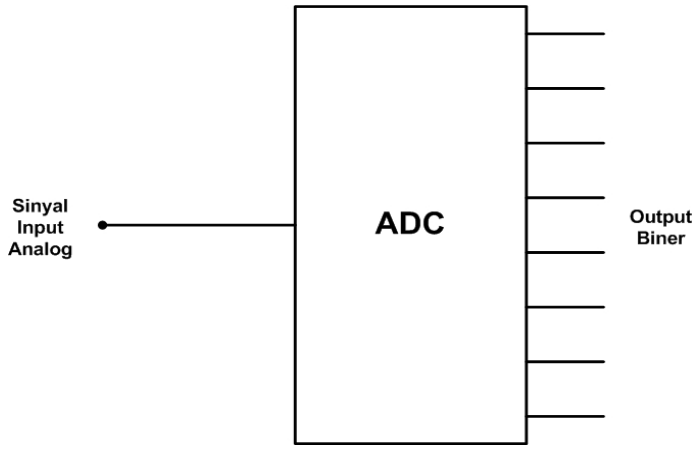
Pada intinya ‘codec’ mengambil sebuah signal analog yang langsung diubah kedalam deretan aliran bit, sedang pada bagian penerima akan melakukan proses kebalikannya sehingga mendapatkan kembali signal analognya. Pada gambar 2.10 menggambarkan proses tersebut. Baik signal analog maupun signal digital yang diubah agar mampu merambat kedalam saluran digital, proses inilah yang menghasilkan transmisi digital.

Transmisi analog berarti mentransmisikan signal analog tanpa memperhatikan pada konteks isi data, signal dapat mewakilkan signal analog (contoh : suara) atau signal digital (contoh : data binary yang lewat melalui modem). Dalam kasus lain, signal analog akan menjadi lemah (berkurang) setelah mencapai jarak tertentu.

2.4 *Analog to Digital Converter*

Sebuah perangkat *analog-to-digital converter* (ADC, A/D) digunakan untuk mengubah sebuah sinyal kontinu ke bentuk sinyal diskrit. Biasanya, ADC digunakan untuk mengubah input analog dalam besaran tegangan atau arus menjadi sejumlah angka digital.

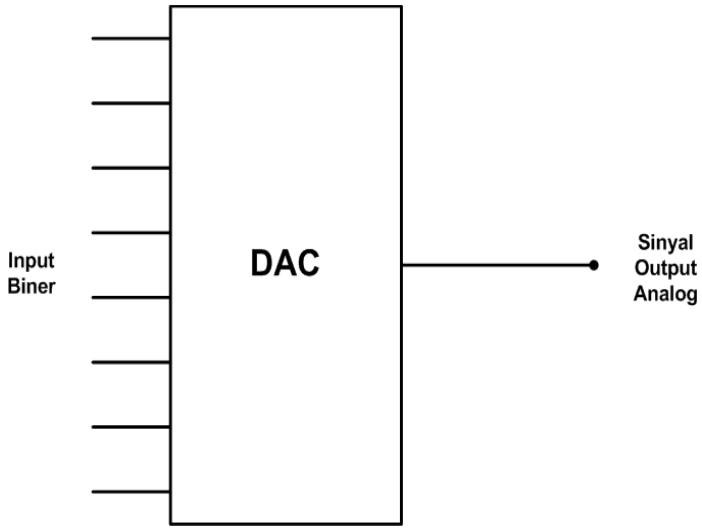
Resolusi perubahan dari sinyal analog ke bentuk sinyal diskrit digital ditentukan oleh nilai-nilai diskrit yang diperoleh dari sebuah rentang nilai analog. Hasil dari perubahan tersebut biasanya disimpan dalam bentuk bilangan-bilangan biner, sehingga resolusi biasanya diekspresikan dalam bentuk bit. Sebagai contoh, sebuah ADC dengan resolusi sebesar 8-bit dapat melakukan *encoding* sebuah input analog pada 256 level, karena $2^8 = 256$.



Gambar 2.11 Diagram Blok Perangkat ADC

2.5 *Digital to Analog Converter*

Digital-to-analog converter (DAC, D/A) bekerja dengan baik membalik arah dari proses ADC. Perangkat ADC digunakan untuk mengubah kode-kode digital (biner) ke sebuah bentuk sinyal analog pada besaran tegangan atau arus.



Gambar 2.12 Diagram Blok Perangkat DAC

Jenis DAC yang cukup banyak digunakan untuk aplikasi *high-fidelity* audio adalah *Pulse Width Modulation*. *Pulse Width Modulation* (PWM) merupakan jenis DAC yang paling sederhana. Tegangan atau arus yang stabil akan di-*switch* ke sebuah analog *low pass filter* dengan durasi yang ditentukan oleh input kode digital.

2.6 Media Transmisi

Media transmisi adalah suatu jalur antara pemancar dan penerima dalam sistem transmisi data. Media transmisi dapat diklasifikasikan menjadi *guided* dan *unguided* (dengan perantara dan tanpa perantara). Media transmisi akan dilewati oleh gelombang elektromagnetik, jadi signal yang ada baik itu analog ataupun digital harus diubah ke dalam bentuk gelombang elektromagnetik. Jenis media transmisi *guide seperti copper twisted pair* (sepasang kabel tembaga), *copper coaxial cable* (kabel tembaga coaxial) dan fiber optik. Sedangkan media transmisi *unguided* menghantarkan gelombang atau signal elektromagnetik tanpa melalui suatu perantara yang solid, yaitu melalui udara, baik di dalam atmosfir maupun di ruang angkasa. Bentuk transmisi ini biasa disebut juga dengan transmisi tanpa kabel (*wireless transmission*).

Kualitas dan karakteristik dari transmisi data ditentukan oleh media dan signalnya. Untuk transmisi *guided*, media yang dipakai lebih penting dalam menentukan pembatasan transmisi, sedangkan pada *unguided*, bandwidth dari signal yang dihasilkan oleh antena pemancar lebih penting. Yang menjadi kunci pokok dalam transmisi signal dengan antena adalah penentuan arah dalam memancarkan signal. Pada umumnya signal frekuensinya rendah (<UHF) memancar ke segala arah, signal merambat ke segala arah dari antena, dengan arah rambatan mengikuti kontur tanah. Sedangkan signal yang frekuensnya tinggi (>VHF) lebih dimungkinkan untuk memfokuskan arah pemancaran ini disebabkan karena sifat dari gelombang itu sendiri, yang bergerak secara garis lurus.

Hal yang penting dalam perancangan sistem transmisi data adalah kecepatan rambat data dan jarak yang ditempuh. Semakin tinggi kecepetan data semakin baik dan semakin pendek jaraknya semaik baik. Kecepatan data dan jarak transmisi ditentukan oleh sejumlah faktor. Faktor-faktor tersebut antara lain :

1. ***Bandwidth***

   Semakin besar *bandwidth* suatu media, semakin tinggi kecepatan data yang dilaluinya.

2. **Kesalahan-kesalahan dalam saluran transmisi**

   Kesalahan yang mungkin terjadi dalam media komunikasi bervariasi antara satu media dengan media lainnya. Untuk media *guided*, umumnya pada kabel *twisted pair* lebih banyak masalah terjadi dibandingkan dengan kabel coaxial. Dibandingkan dengan fiber optik, maka pada kabel coaxial lebih banyak terjadi gangguan.

3. **Intervensi (*Interferensi*)**

   Intervensi atau gangguan yang berupa sinyal yang frekuensinya bertumpang tindih, dapat mengubah atau menghilangkan signal pada transmisi *unguided*. Intervensi lebih banyak terjadi pada transmisi u*nguided* daripada *guided*. Pada transmisi *guided*, intervensi dapat terjadi karena pengaruh gelombang dari kabel yang berdekatan, misalnya pada *twisted pair* yang biasanya terdiri dari kabel-kabel yang berbeda disatukan dalam selongsong kabel. Lapisan pelindung kabel yang baik dapat meminimumkan terjadinya intervensi pada transmisi *guided*.

4. Jumlah penerima

Medium *guided* digunakan dalam membangun *point to point link* atau *shared link* dengan jumlah sambungan yang banyak. Masalahnya adalah bahwa setiap sambungan dapat menghasilkan redaman *(attenuation*) dan distorsi pada jaringan atau jalur kabel tersebut, sehingga dapat membatasi jarak dan atau kecepatan data.

Dalam perkembangannya, media transmisi dalam komunikasi tidak hanya menggunakan kabel (*wire*), melainkan terdapat juga media transmisi tanpa kebel, yang biasa disebut dengan *wireless*.

2.6.1 Media Transmisi Kabel (*wire*)

Salah satu contoh media trensmisi yang menggunakan kabel, yaitu telepon selular dengan komputer, komunikasi yang terjadi diantara keduanya menggunakan kabel data. Kabel data ini biasanya sudah termasuk dalam paket atau aksesoris yang terdapat pada telepon selular, sesuai dengan jenis dan seri dari telepon selular tersebut. Tipe koneksi kabel data tersebut biasanya berupa USB atau *communication port*.

2.6.2 Media Transmisi Tanpa Kabel (*wireless*)

Untuk transmisi *guided* atau dengan menggunakan kabel, pengiriman dan penerimaan signal dilakukan tanpa menggunakan antena. Untuk transmisi tanpa kabel antena memancarkan gelombang elektromagnetik ke medium transmisi (biasanya adalah udara) dan kemudian diterima oleh antena penerima melalui medium yang ada

disekitarnya (udara). Terdapat dua jenis konfigurasi dari *wireless transmission* yaitu satu arah (*directional*) dan segala arah (*omnidirectional*). Untuk transmisi satu arah, antena memfokuskan pancaran-pancaran elektromagnetik dimana antena pemancar dan antena penerima harus saling berhadapan dalam satu garis lurus. Sedangkan untuk transmisi ke segala arah, signal yang dipancarkan menyebar ke segala arah dan dapat diterima oleh banyak antena dari berbagai arah. Pada umumnya, semakin tinggi frekuensi dari signal, semakin memungkinkan untuk memfokuskannya pada pemancaran ke segala arah.

Penggunaan media kabel untuk berkomunikasi dirasakan kurang efektif, *fleksibel* dan *efisien*. Adapun beberapa media komunikasi *wireless* adalah sebagai berikut :

1. ***Infrared***

   Teknologi *infrared* merupakan teknologi pertama dan bersifat umum, seperti pada remote televisi. Prinsip kerja dari remote televisi berupa prosesor kecil pada remot yang akan melakukan penerjemahan pada saat penekanan tombol. Penekanan pada tombol tersebut merupakan instriksi bahasa mesin (*assembly*) yang dikirimkan melalui infrared ke TV, lalu data diubah kembali menjadi instruksi yang dikenali oleh TV tersebut.

   Media transmisi yang sering terdapat dalam telepon selular yaitu *infrared*. Terutama bagi beberapa telepon selular seri lama yang belum memiliki media transmisi jenis baru.

IrDA(*Infrared Data Association*) memiliki panjang gelombang sekitar 875 Nm. Sinar yang dihasilkan dan dipancarkan dari sebuah lampu LED. Terdapat dua versi yang telah dikembangkan pada *infrared,* yaitu versi 1.0 memiliki kecepatan dari 0,576 Kbps hingga 115,2 Kbps dan versi 2.0 memiliki kecepatan 0,576 Mbps hingga 1,152 Mbps.

2. *Wireless Fidelity* (Wi-Fi)

Wi-Fi (*Wireless Fidelity*) bekerja pada frekuensi sama dengan *bluetooth*, yaitu pada 2.4 GHz. Tetapi terdapat perbedaan, *bluetooth* menggunakan *Spread Spectrum Frequency Hopping* (SSFH), sedangkan Wi-Fi menggunakan *Direct Sequence Spread Spectrum* (DSSS).

Jadi, *spread* pada Wi-Fi akan lebih stabil dan tentunya lebih cepat dibandingkan dengan *bluetooth*. Wi-Fi memiliki kelemahan dalam hal keamanan dan rentan terhadap konflik dengan perangkat lain dalam waktu yang bersamaan. Wi-Fi yang dikenal dengan standar IEEE 802.11b pengoprasiannya mulai meluas dan beberapa operator di Amerika Serikat mengoprasikannya secara *Hot Spot* di lokasi startegis. Teknologi Wi-Fi pada telepon selular dewasa ini suda mulai ramai di pasaran, baik itu telepon selular yang berjenis GSM maupun CDMA.

2.7 *Bluetooth*

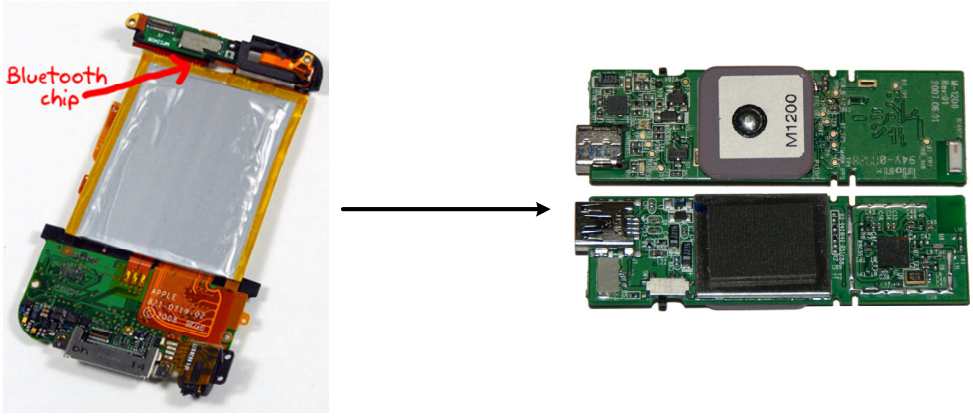
Bluetooth adalah sebuah teknologi komunikasi *wireless* (tanpa kabel) yang beroperasi dalam pita frekuensi 2,4 GHz *unlicensed* ISM dengan menggunakan sebuah *frequency hopping tranceiver* yang mampu menyediakan layanan komunikasi data dan

suara secara *real-time* antara *host-host bluetooth* dengan jarak jangkauan layanan yang terbatas (sekitar 10 meter).

ISM adalah *band* (*Industrial Scientific and Medical*) bebas lisensi yang ditetapkan oleh Komisi Komunikasi Federal (FCC). FCC menetapkan undang-undang yang mengatur pengoperasian piranti LAN nirkabel. Band ISM berada pada lokasi mulai dari 902 MHz, 2.4 GHz dan 5.8 GHz dengan lebar yang bervariasi dari sekitar 26 MHz hingga 150 MHz.

Spesifikasi *bluetooth* menyediakan definisi *link layer* dan *application layer* sehingga mendukung aplikasi data dan suara. Teknologi *bluetooth* juga dapat menembus benda padat dan bersifat *omni-directional* sehingga tidak memerlukan posisi *line-of-sight* seperti pada inframerah. Keamanan merupakan prioritas utama dalam pengembangan spesifikasi *bluetooth.*

Bluetooth sendiri dapat berupa *card* yang bentuk dan fungsinya hampir sama dengan *card* yang digunakan untuk *wireless local area network* (WLAN) dimana menggunakan frekuensi radio standar IEEE 802.11, hanya saja pada *bluetooth* mempunyai jangkauan jarak layanan yang lebih pendek dan kemampuan transfer data yang lebih rendah. Berikut gambaran secara fisik dari *bluetooth* :



Gambar 2.13 *Bluetooth* secara Fisik

Pada dasarnya *bluetooth* diciptakan bukan hanya untuk menggantikan atau menghilangkan penggunaan kabel didalam melakukan pertukaran informasi, tetapi juga mampu menawarkan fitur yang baik untuk teknologi *mobile wireless* dengan biaya yang relatif rendah, konsumsi daya yang rendah, *interoperability* yang menjanjikan, mudah dalam pengoperasian dan mampu menyediakan layanan yang bermacam-macam.

2.7.1 Keuntungan Teknologi *Bluetooth*

Teknologi *bluetooth* memiliki fitur-fitur utama ketangguhan (*robustness*), kebutuhan daya yang kecil (*low power consumption*) dan biaya yang murah. Perangkat *Bluetooth* menghindari *interferensi* dengan sistem lainnya dengan cara mengirimkan signal yang sangat lemah, berkisar 1 milliwatt. Sebagai perbandingan, kekuatan telepon selular dapat memancarkan suatu signal 3 watt.

Teknologi *bluetooth* saat ini telah mengglobal dan menjadi standar untuk menghubungkan berbagai *device* elektronik, mulai dari *mobile phone*, *portable computer*, *mobil, stereo headset*, sampai MP3 *player* dll.

Bluetooth juga memiliki konsep unik "profile", sehingga memungkinkan produk untuk dapat bekerja tanpa menginstall *driver*, selain itu *Bluetooth* juga menyediakan layanan sekuritas built-in. Terdapat 3 mode sekuritas pada spesifikasi *bluetooth*, yaitu :

1. Mode Keamanan 1 : *non-secure*
2. Mode Keamanan 2 : *service level enforce security*
3. Mode Keamanan 3 : *link level enforced security*

Device security levelnya terbagi menjadi 2, yaitu : *trusted device* dan *untrusted device*, sedangkan *service security* levelnya terbagi menjadi 3, yaitu :

a. Service yang membutuhkan *authorization* dan *authentication*
b. Service yang hanya membutuhkan *authentication*
c. Service yang terbuka untuk semua *device*

Selain layanan sekuritas yang memadai, teknologi *bluetooth* relatif mudah digunakan karena *Bluetooth* tidak memerlukan kita untuk melakukan apapun yang khusus untuk membuatnya bekerja.

Teknologi *bluetooth* juga sangat mendukung aplikasi *synchronous* dan *asynchronous*, sehingga memudahkan untuk mengimplementasikan *device-device* (perangkat-perangkat) yang saling berbeda untuk berbagai macam layanan, misalnya seperti suara dan internet. Selain itu, karena signalnya dapat menembus benda padat

dan bersifat *omni-directional*, maka tidak memerlukan posisi *line-of-sight* seperti pada inframerah.

Teknologi *bluetooth* menyediakan layanan komunikasi *point to point* maupun komunikasi *point to multipoint*. Selain itu, sistem *bluetooth* juga mendukung *bridging* jaringan, misalnya *mobile phone* yang menggunakan *chip bluetooth* dapat menggubungkan *piconet* lokal dengan jaringan GSM global.

2.7.2 Protokol – protokol *Bluetooth*

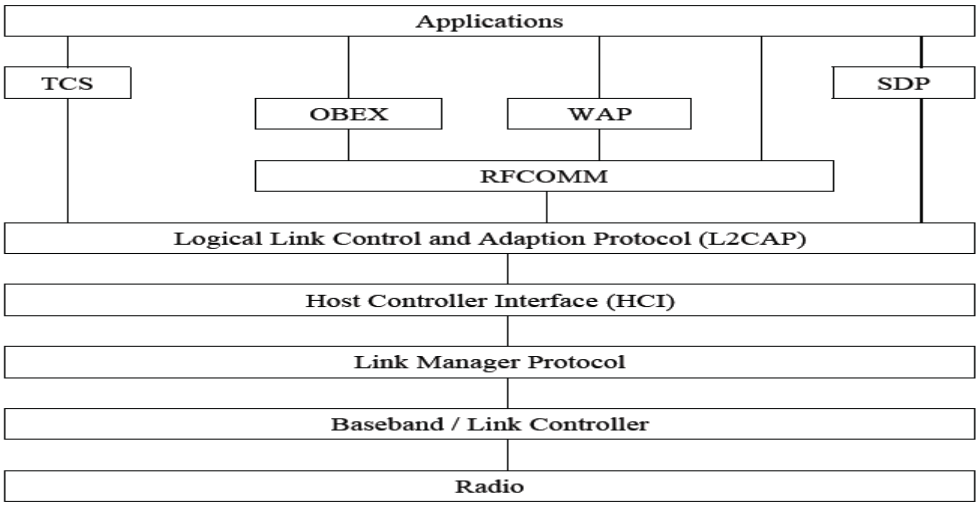
Protokol-protokol *bluetooth* dimaksudkan untuk mempercepat pengembangan aplikasi-aplikasi dengan menggunakan teknologi *bluetooth*.

Stack protokol *bluetooth* dapat dibagi ke dalam 4 layer, seperti pada tabel berikut :

Tabel 2.1 Tumpukan Layer pada *Stack Protocol Bluetooth*

Protokol Layer	Protokol pada Stack
Bluetooth Core Protokol	Baseband, LMP, L2CAP, SDP
Cable Replacement Protokol	RFCOMM
Telephony Control Protokol	TCS Binary, AT-Commands
Adopted Protokol	PPP, UDP/TCP/IP, OBEX, WAP, vCard, vCal, irMC, WAE

Bluetooth stack memiliki beberapa Layer. HCI merupakan layer yang memisahkan *hardware* dari *software* dan mengimplementasikan *software* dan *hardware*. Layer di bawah HCI mengimplementasikan *hardware*, sedangkan layer di atas HCI mengimplementasikan *software*. Lihat pada gambar di bawah ini :



Gambar 2.14 *Bluetooth Protocol Stack* (Secara Logik)

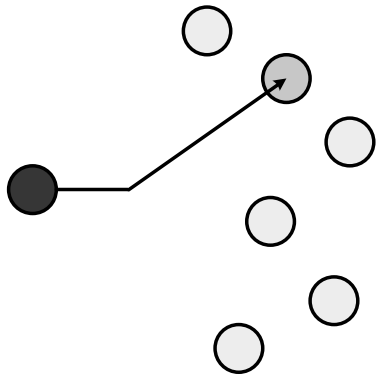
Tabel 2.2 Keterangan pada Gambar 2.14 Tentang *Bluetooth Protokol Stack*

Layer	Deskripsi
Applications	*Profile bluetooth* yang mengendalikan *developer* dalam menentukan bagaimana sebuah aplikasi harus menggunakan protokol *stack.*
Telephony Control System	Menyediakan *Telephony service.*
Service Discovery Protocol (SDP)	Digunakan untuk mencari *service* yang disediakan pada *remote bluetooth device.*
WAP and OBEX	Menyediakan *interface* pada bagian layer yang lebih tinggi pada protokol komunikasi yang lain.
RFCOMM	Menyediakan keterhubungan melalui *bluetooth* seperti yang digunakan pada standard serial (COM) port.
L2CAP	Merupakan layer yang menangani *packet segmentation and reassembly* (SAR). *Multiplexes* data dari layer yang lebih tinggi dan merubah merubah antar ukuran packet yang berbeda.
Host Control Interface (HCI)	Menangani komunikasi antara *host* dan *bluetooth module.*
Link Manager Protocol	Mengontrol dan konfigurasi *link* kepada *device* lain.
Baseband and Link Controller	Mengontrol akses dan mengirim paket data ke media fisik yang melewati *bluetooth* radio. Layanan-layanan yang disediakan pada layer ini antara lain: prosedur *inquiry* dalam menemukan *device*, fungsionalitas *device* dalam sinkronisasi *clock* dan membangun *link* pada frekuensi *hopping* yang tepat, *error correction*, *data whitening*, *flow control*, fungsi untuk men-*generate* kunci enkripsi.
Radio	*Modulate* dan *demodulate* data dari transmisi dan menerima di udara. *Bluetooth* Radio merupakan *layer* fisik yang menggunakan *frequency hopping spead spectrum* untuk memperkecil interferensi dari *device* lain yang menggunakan band ini. *Bluetooth* membagi frekuensi band 2.4 GHz menjadi 79 *channel* pada 1 MHz (dari 2.402 sampai 2.480 GHz) dan frekuensi hop 1600 kali per detik.

2.8 Komunikasi *Unicast*

Dalam jaringan komputer, transmisi *unicast* merupakan pengiriman paket informasi ke satu tujuan jaringan. Istilah "*unicast*" menurut analogi, merupaka "menyiarkan" yang berarti transmisi data yang sama untuk semua tujuan. Metode distribusi multi-mode yang lain seperti, *multicasting*, mirip dengan IP penyiaran, tetapi dilaksanakan secara lebih efisien.

Pengolahan pesan *unicast* digunakan untuk semua proses jaringan dimana *resource* yang unik atau pribadi, diminta untuk membuat sebagian besar jalur jaringannya sendiri. *Unicast* menggunakan dua jalur koneksi untuk melengkapi jalur transaksinya. Lihat pada gambar di bawah ini :

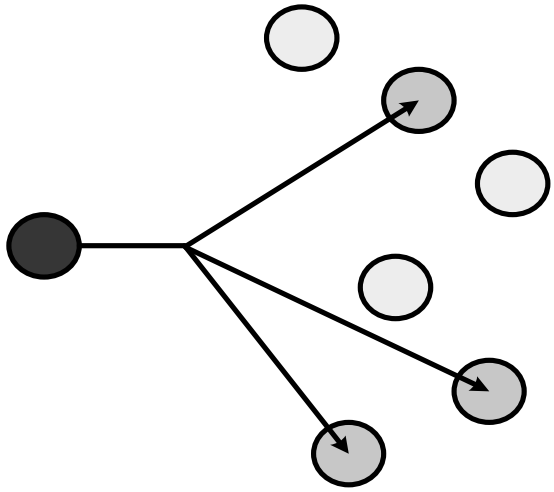


Gambar 2.15 Komunikasi *Unicast*

Aplikasi jaringan tertentu yang didistribusikan secara besar, terlalu sulit untuk diterapkan pada *Unicast*. Termasuk streaming media ke berbagai form. Istilah tersebut juga digunakan oleh penyedia layanan streaming. *Unicast* media server berbasis terbuka dan menyediakan streaming unik untuk setiap pengguna.

2.9 Komunikasi *Multicast*

Pengalamat *multicast* merupakan sebuah jaringan teknologi untuk penyampaian informasi kepada sekelompok tujuan secara sekaligus dengan menggunakan strategi yang paling efisien untuk menyampaikan pesan melalui setiap link jaringan yang hanya sekali, dan kemudian membuat salinan hanya ketika link ke beberapa tujuan split.



Gambar 2.16 Komunikasi *Multicast*

Kata "*multicast*" biasanya digunakan untuk merujuk ke IP *multicast* yang sering digunakan untuk streaming media dan televisi aplikasi Internet. IP *multicast* dalam pelaksanaan konsep *multicast*, terjadi di tingkat IP routing, dimana routers membuat jalur distribusi optimal untuk datagram yang dikirim ke alamat *multicast* tujuan secara real-time. Pada Data Link Layer, *multicast* menjelaskan satu-ke-banyak distribusi seperti Ethernet menangani *multicast*, Asynchronous Transfer Mode (ATM) *point-to-multipoint* sirkuit virtual atau *Infiniband multicast*.

2.10 Komunikasi *Broadcast*

Perbedaan yang mendasar antara radio broadcast dengan microwave adalah bahwa radio *broadcast* pemancarannya ke segala arah, sedangkan microwave dipancarkan lebih terfokus pada satu arah (*directional*). Selain itu, radio *broadcast* tidak memerlukan antena penerima yang khusus dan memiliki arah yang bebas.

Radio merupakan suatu stasuin atau terminal komunikasi umum yang beroperasi pada wilayah frekwensi antara 3 KHz -300 GHz. Dan menggunakan istilah radio yang mencakup frekwensi VHF dan sebagian frekwensi UHF, yaitu 30 MHz – 1 GHz. Wilayah frekwensi ini digunakan untuk radio FM, dan jaringan TV juga untuk aplikasi jaringan komunikasi data.

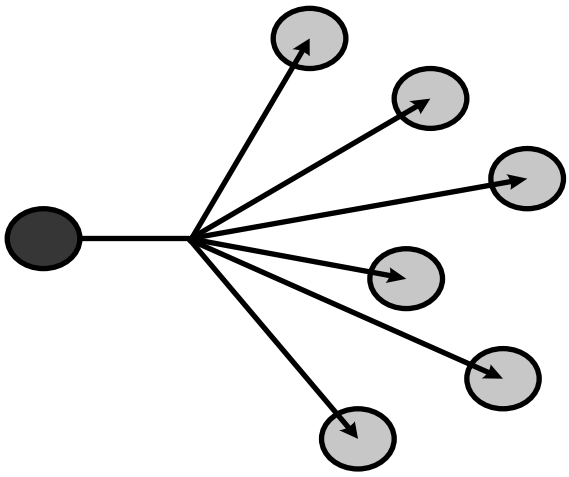
Wilayah antara 30 MHz – 1 GHz merupakan wilayah frekuensi yang efektif untuk komunikasi *broadcast*. Frekwensi diatas 30 MHz sampai dengan VHF (Gelombag Radio) akan bergerak di bawah permukaan ionosfer. Frekwensi ini sangat sensitif terhadap cuaca. Sedangkan pada frekwensi yang lebih tinggi seperti gelombang mikro maupun satelit, tidak terpengaruh dengan cuaca.

Dengan teknik perambatan pada satu garis pemancaran, gelombang radio menggunakan rumus $D = 7{,}14\ \sqrt{(Kh)}$ untuk menetukan jarak pengiriman dan penerimaan transmisi.

Semakin besar panjang gelombangnya, redaman yang terjadi pada gelombang radio pun relatif berkurang.

Sumber utama gangguan pada gelombang radio broadcast adalah gangguan dari banyak jalur pemancaran sinyal yang dipantulkan dari tanah, air maupun benda lainnya. Pantulan sinyal dapat menyebarkan garis edar sinyal menjadi lebih banyak (simpang siur) dan dapat mengganggu pemancaran gelombang radio yang tengah

berlangsung. Efek dari gangguan ini dapat dengan jelas terlihat pada siaran TV, dimana ketika pesawat udara melintas, gambar pada TV akan menjadi acak atau bertumpuk karena adannya pantulan sinyal dari pesawat tersebut.



Gambar 2.17 Komunikasi *Broadcast*

Sistem *broadcast* biasanya memungkinkan pengalamatan untuk semua simpul memakai kode khusus di field alamat. Ketika paket berkode ditransmisikan, paket diterima dan diproses pada simpul di jaringan. Dan *broadcast* itu sendiri memiliki saluran komunikasi tunggal yang dipakai bersama-sama oleh semua mesin yang ada pada jaringan. Pesan-pesan berukuran kecil, disebut paket dan dikirimkan oleh suatu mesin kemudian diterima oleh mesin-mesin yang lainnya. Bagian alamat pada paket berisi keterangan tentang kepada siapa paket ditujukan. Saat menerima sebuah paket, mesin akan cek bagian alamat, jika paket tersebut untuk mesin itu, maka mesin akan memproses paket itu. Jika bukan maka mesin mengabaikannya.

2.11 Teknologi *Wireless* Java

Teknologi *wireless Java* secara konsep, dapat dibagi dalam dua kategori, pertama untuk lokal dan kedua untuk area yang luas.

Peralatan yang termasuk dalam kategori pertama misalnya adalah *remote control* untuk membuka dan mengunci mobil, garasi, telepon *cordles* 900 MHz, peralatan mainan dengan radio control dengan menggunakan jaringan *wireless* dan lain sebagainya. Peralatan *wireless* jenis pertama ini hanya bekerja pada area yang jangkauannya tidak terlalu jauh.

Sedangkan peralatan jenis aplikasi yang kedua diantaranya adalah pager, telepon selular, PDA dan lain sebagainya. Jangkauan dari perangkat tersebut jauh lebih besar dari aplikasi jenis pertama. Karena jaringan yang ada di permukaan bumi berupa *cell tower*, peralatan komunikasi bergerak seperti telepon selular, dapat menerima layanan dari sebuah *wireless carrier* atau perusahaan yang mengoprasikan *cell tower* tersebut.

Aplikasi komunikasi bergerak, dalam perkembangan awal masing-masing vendor menghasilkan *platform* aplikasi dan sistem operasi sendiri. Sehingga sebuah telepon selular mempunyai *platform* aplikasi masing-masing.

Perbedaan aplikasi menyebabkan suatu *platform* aplikasi maupun sistem operasi dalam satu telepon selular tidak dapat dijalankan dalam peralatan telepon selular lainnya. Sehingga berakibat memperburuk pengembangan aplikasi yang baru.

Salah satu teknologi *Java* adalah "*write once run everywhere*", sehingga portabilitas *Java* merupakan suatu kekuatan yang dimiliki *Java*. *Java* dijalankan pada sistem operasi apapun tanpa tanpa harus dikompilasi ulang pada program *Java* yang

akan di jalankan di telepon selular tersebut. Untuk komunikasi bergerak, dengan perangkat yang kecil dan memori yang terbatas, *Sun Microsystem* mengenalkan *Java* 2 *Micro Edition* (J2ME), yang merupakan salah satu bagian teknologi *Java* yang digunakan untuk aplikasi *Java* yang berjalan pada perangkat *mobile* dan teknologi aplikasi *wireless*.

2.12 *Object Oriented Programming* (OOP)

Object Oriented Programming (OOP) dalam bahasa indonesia yaitu Pemrograman berbasis objek (PBO) merupakan metode implementasi yang diorganisasikan sebagai kumpulan objek yang bekerja sama, dimana masing-masing objek mempresentasikan dengan cepat dari kelas dan kelas-kelas merupakan anggota suatu hirarki yang disatukan melalui keterhubungan pewarisan.

Model data berorientasi objek dapat memberikan fleksibilitas yang lebih, kemudahan mengubah program dan digunakan luas dalamteknik piranti lunak skala besar. Lebih jauh lagi, pendukung PBO mengklaim bahwa PBO lebih mudah dipelajari bagi pemula, jika dibandingkan dengan pendekatan lainnya. Selain itu juga pendekatan PBO lebih mudah dikembangkan dan dirawat. Pemograman Berbasis Objek (PBO) menekankan beberapa konsep, antara lain sebagai berikut :

1. **Class**

   Kumpulan atas definisi data dan fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebuah class secara tipikal sebaiknya dapat dikenali oleh seorang non-programmer sekalipun, terkait dengan domain permasalahan yang ada dan kode

yang terdapat salam sebuah class sebaiknya bersifat mandiri dan independen. Dengan modularitas struktur dari sebuah program terkait dengan aspek-aspek dalam masalah yang akan diselesaikan melalui program tersebut. Cara seperti ini akan menyederhanakan pemetaan dari masalah ke sebuah program ataupun sebaliknya.

2. Objek

Membungkus data dan fungsi secara bersamaan menjadi suatu unit dalam sebuah program komputer, objek merupakan dasar dari sebuah modularitas dan struktur dalam sebuah program komputer berorientasi objek.

3. Abstraksi

Kamampuan sebuah progam untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokuskan kepada inti. Setiap objek dalam sisem, melayani sebagai model dari "pelaku" abstrak yang dapat melakukan kerja, laporan dan perubahan keadaanya serta dapat berkomunikasi dengan objek yang lainnya dalam sistem, tanpa mengungkapkan bagaimana kelebihan ini diterapkan. Proses, fungsi maupun metode dapat juga dibuat abstrak, dan beberapa teknik yang digunakan untuk mengembangkan sebuah pengabstrakan.

4. Enkaptulasi

Memastikan pengguna sebuah objek, tidak dapat mengganti keadaan dari sebuah objek edngan cara yang tidak layak dan hanya metode dalam objek tersebut yang diberi ijin untuk mengakses keadaannya. Setiap objek mengakses interface yang menyebutkan bagaimana objek lainnya dapat berinteraksi dengannya. Objek

lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.

5. ***Polimorfisme***

   Melalui penerimaan pesan. Tidak bergantung kepada pemanggilan subrutin, bahasa orientasi objek dapat mengirim pesan, metode tertentu yang berhubungan dengan sebuah pengiriman pesan tergantung kepada objek tertentu diman pesan tersebut dikirim.

6. ***Inheritas***

   Mengatur *polimorfisme* dan enkaptilasi dengan mengijinkan objek didefinisikan dan diciptakan dengan jenis khusus dari objek yang sudah ada. Obje-objek ini dapat membagi perilaku mereka tanpa harus mengimplementasikan kembali perilaku tersebut (bahasa berbasis objek tidak selalu memiliki inheritas).

Dengan menggunakan PBO, maka dalam melakukan pemecahan suatu masalah kita tidak melihat bagaimana cara menyelesaikan suatu masalah tersebut (terstruktur) melainkan objek-objek apa yang dapat melakukan pemecahan masalah tersebut.

2.13 *Unified Modelling Language* (UML)

Unified Modelling Language (UML) merupakan bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan dan membangun sistem perangkat lunak. UML tidak berdasarkan pada bahasa pemrograman tertentu. UML dapat digunakan untuk membuat model pada semua jenis aplikasi piranti lunak dimana

aplikasi tersebut dapat berjalan pada piraanti keras, sistem operasi dan jaringan serta ditulis dalam bahasa pemrograman apa saja. Karena UML menggunakan class dan operation dalam konsep dasarnya, maka UML lebih cocok untuk penulisan piranti lunak pada bahasa Perograman Berorientasi Objek (PBO) , seperti Java, C++, Visual Basic.net dan PBO lainnya.

2.13.1 Tujuan Perancangan UML

Perancangan UML mempunya beberapa tujuan utama. Tujuan-tujuan tersebut adalah sebagai berikut :

1. Menyediakan bahasa pemodelan visual yang ekspresif dan siap pakai untuk pengembangan dan pertukaran model-model yang berarti.
2. Menyediakan mekanisme perluasan dan spesifikasi untuk memperluas konsep inti.
3. Mendukung spesifikasi independen pada bahasa pemrograman dan proses pengembangan tertentu.
4. Menyediakan basis formal untuk bahasa pemodelan.
5. Mendukung konsep-konsep pengembangan level tinggi, seperti komponen, kolaborasi, *framework* dan *pattern*.

2.13.2 Diagram dan Teknik Pemodelan UML

Diagram mengambarkan banyak hal, penggunaan notasi yang terdefinisi dengan baik dan ekspresif sangat penting pada proses pengembangan perangkat lunak, antara lain sebagai berikut :

1. Notasi standar memungkinkan pengembang mendeskrispsikan rumusan arsitektur dan kemudian membuat komunikasi secara tidak ambigu.
2. Notasi yang bagus membebaskan otak untuk berkonsentrasi pada masalah-masalah yang lebih lanjut.
3. Notasi yang baik memungkinkan melakukan eliminasi keperluan konsitensi dan kebenaran keputusan menggunakan alat yang otomatis.

UML menyediakan beberapa diagram visual yang menunjukkan berbagai aspek dalam sistem. Diagram-diagram yang disediakan dalam UML adalah sebagai berikut :

1) Diagram kelas (*Class diagram*)

Diagram ini menunjukkan sekumpulan kelas, interface dan kolaborasi serta keterhubungannya. Diagram kelas ditujukan untuk pandangan statik terhadap sistem.

2) Diagram objek (*Object diagram*)

Menunjukkan sekumpulan objek dan keterhubungannya untuk menunjukkan potongan statik dari instan-instan yang ada pada diagram kelas. Serta untuk memperlihatkan satu prototipe atau kasus tertentu yang mungkin terjadi. Diagram objek menyediakan notasi grafis formal guna memodelkan objek, kelas dan saling keterhubungannya. Diagram objek berguna untuk abstract modelling dang

perancangan program-program sesungguhnya. Pada pendekatan ini, bentukan dasar dari sistem perangkat lunak adalah objek atau kelas.

3) Diagram use-case (*Use case diagram*)

Diagram ini menunjukkan sekumpulan kasus fungsional dan aktor serta keterhubungannya.

4) Diagram sekuen (*Sequence diagram*)

Diagram ini menunjukkan interaksi yang terjadi antar objek. Diagram ini merupakan pandangan dinamis terhadap sistem. Diagram ini menekankan pada basis urutan waktu dari pesan-pesan yang terjadi.

5) Diagram kolaborasi (*Collaboration diagram*)

Diagram ini juga merupakan diagram interaksi. Diagram ini menekankan pada organisasi struktur dari objek-objek yang mengirim dan menerima pesan.

6) Diagram state chart (*State chart diagram*)

Diagram ini berisi status, transisi, kejadian dan aktifitas. State chart merupakan pandangan dinamis dari sistem. Diagram ini penting dalam memodelkan perilaku antarmuka, kelas, kolaborasi dan menekankan pada urutan kejadian.

7) Diagram aktifitas (*Activity diagram*)

Diagram aktifitas menunjukkan aliran ektifitas pada sistem. Diagram ini memiliki pandangan dinamis dari sistem. Diagram ini digunakan umtuk memodelkan fungsi sistem dan menekankan pada aliran kendali diantara objek-objek.

8) Diagram komponen (*Component diagram*)

Menunjukkan organisasi dan hubungan ketergantungan diantara sekumpulan komponen. Diagram ini merupaka pandangan statik terhadap implementasi sistem.

9) Diagram pengembangan (*Deployment diagram*)

Menunjukkan konfigurasi pemrosesan saat jalan dan komponen-komponen yang terdapat didalamnya. Pilihan model dan diagram yang digunakan dipengaruhi oleh bagaimana persoalan ditangani dan bagaimana solusi dibentuk. Setiap sistem komplek perlu didekati melalui sekumpulan pandangan model yang hampir independent.

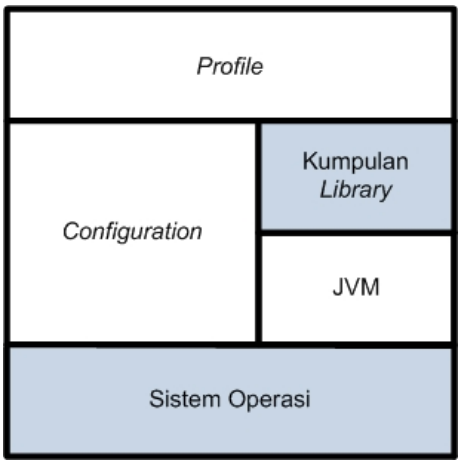
2.14 *Java 2 Micro Edition* (J2ME)

2.14.1 Pengertian J2ME

Java2 Micro Edition atau yang biasa disebut J2ME adalah lingkungan pengembangan yang didesain untuk meletakkan perangkat lunak Java pada barang elektronik beserta perangkat pendukungnya. Pada J2ME, jika perangkat lunak berfungsi baik pada sebuah perangkat maka belum tentu juga berfungsi baik pada perangkat yang lainnya. J2ME membawa Java ke dunia informasi, komunikasi, dan

perangkat komputasi selain perangkat komputer desktop yang biasanya lebih kecil dibandingkan perangkat komputer desktop. J2ME biasa digunakan pada telepon selular, pager, *Personal Digital Assistants* (PDA) dan sejenisnya.

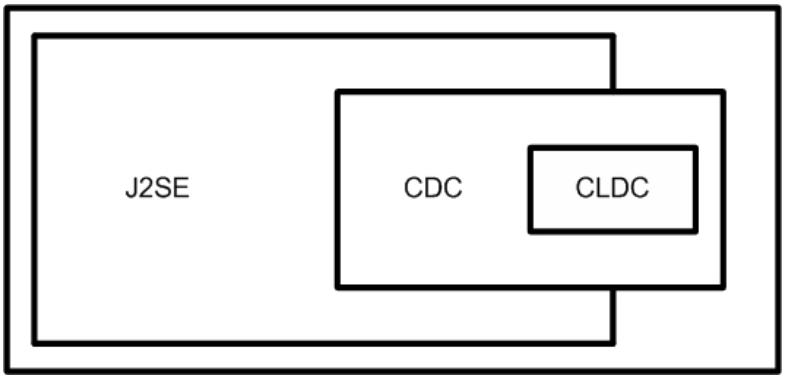
J2ME adalah bagian dari J2SE, karena itu tidak semua *library* yang ada pada J2SE dapat digunakan pada J2ME. Tetapi J2ME mempunya beberapa *library* khusus yang tidak dimiliki J2SE. Arsitektur J2ME dapat dilihat pada gambar berikut :



Gambar 2.18 Arsitektur J2ME

Teknologi J2ME juga memiliki beberapa keterbatasan, terutama jika diaplikasikan pada ponsel. J2ME sangat tergantung pada perangkat (*device*) yang digunakan, bisa dari segi merk ponsel, maupun kemampuan ponsel dan dukungannya terhadap teknologi J2ME. Misalnya, jika sebuah ponsel tidak memiliki kamera, maka jelas J2ME pada ponsel tersebut tidak dapat mengakses kamera. Keterbatasan lainnya adalah pada ukuran aplikasi, karena memori pada ponsel sangat terbatas. Sebagian besar ponsel tidak mengijinkan aplikasi J2ME menulis pada file karena alasan keamanan.

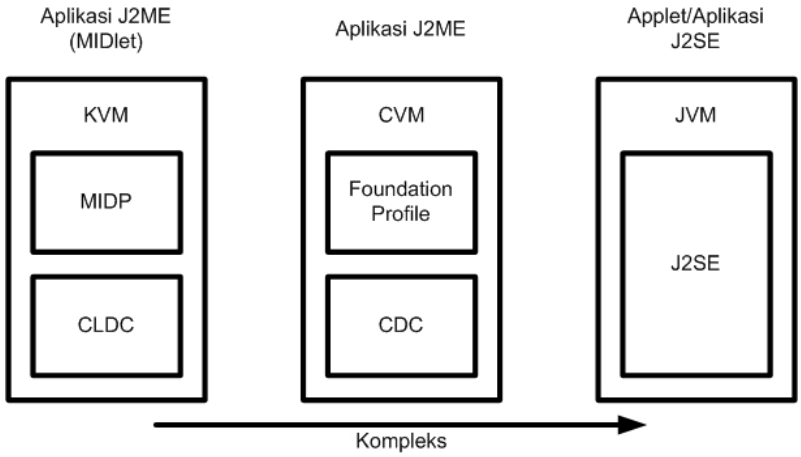
Configuration merupakan *Java library minimum* dan kapabilitas yang dipunya oleh para pengembang J2ME, yang maksudnya sebuah *mobile device* dengan kemampuan *Java* akan dioptimalkan untuk menjadi sesuai. *Configuration* hanyalah mengatur hal-hal tentang kesamaan sehingga dapat dijadikan ukuran kesesuaian antar *device*. Misalnya *library* untuk perubahan data teks menggunakan library UTF (*Unicode Transformation Format*). Dalam J2ME telah didefinisikan dua buah konfigurasi yaitu CLDC (*Connected Limites Device Configuration*) untuk perangkat kecil dan CDC (*Connected Device Configuration*) untuk perangkat yang lebih besar. Lingkup CLDC dan CDC dapat dilihat pada gambar di bawah ini :



Gambar 2.19 Lingkup Configuration

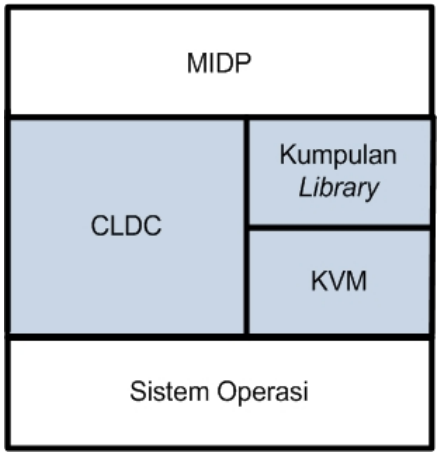
Profile berbeda dengan *configuration*, *profile* membahas sesuatu yang spesifik untuk sebuah perangkat. Sebagai contoh misalnya, sebuah sepeda dengan merk tertentu, tentunya mempunyai ciri spesifik dengan sepeda lain.

Dalam gambar J2ME terdapat dua buah *profile* yaitu MIDP dan *Foundation Profile.* Keterhubungan antara *configuration* dan *profile* yang ada pada J2ME beserta jenis mesin virtualnya dapat dilihat pada gambar berikut :



Gambar 2.20 Hubungan J2ME dan J2SE

2.14.2 *Connected Limited Device Configuration* (CLDC)



Gambar 2.21 Arsitektur J2ME

CLDC atau *Connected Limited Device Configuration* adalah perangkat dasar dari J2ME, spesifikasi dasar yang berupa *library* dan API yang diimplemetasikan pada J2ME, seperti yang digunakan pada telepon selular, pager dan PDA. Perangkat tersebut dibatasi dengan keterbatasan memori, sumber daya dan kemampuan memproses. Spesifikasi CLDC pada J2ME adalah spesifikasi minimal dari *package*,

kelas dan sebagian fungsi *Java Virtual Machine* yang dikurangi agar dapat diimplementasikan dengan keterbatasan sumber daya pada alat-alat tersebut, JVM yang digunakan disebut KVM (*Kilobyte Virtual Machine*). Posisi CLDC pada arsitektur J2ME dapat dilihat pada gambar di atas.

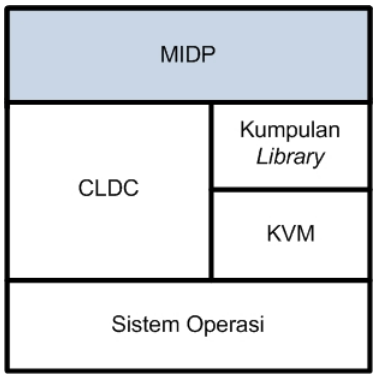
2.14.3 *Connected Device Configuration* (CDC)

CDC atau *Connected Device Configuration* adalah spesifikasi dari konfigurasi J2ME. CDC merupakan komunitas proses pada *Java* yang mempunyai standardisasi. CDC terdiri dari virtual machine dan kumpulan *library* dasar untuk dipergunakan pada profile industri. Implementasi CDC pada J2ME adalah *source code* yang menyediakan sambungan dengan macam-macam *platform*. Berikut adalah perbandingan antara CLDC dan CDC :

Tabel 2.3 Perbedaan CLDC dan CDC

CLDC	CDC
Mengimplementasikan sebagian dari J2SE	Mengimplementasikan seluruh fitur J2SE
JVM yang digunakan adalah KVM	JVM yang digunakan adalah CVM
Digunakan pada perangkat genggam (handphone, PDA, twoway pager) dengan memori terbatas (160 – 512 KB)	Digunakan pada perangkat genggam (internet TV, Nokia Communicator, car TV) dengan memori minimal 2MB
Prosesor : 16/32 bit	Prosesor : 32 bit

2.14.4 *Mobile Information Device Profile* (MIDP)



Gambar 2.22 Arsitektur J2ME

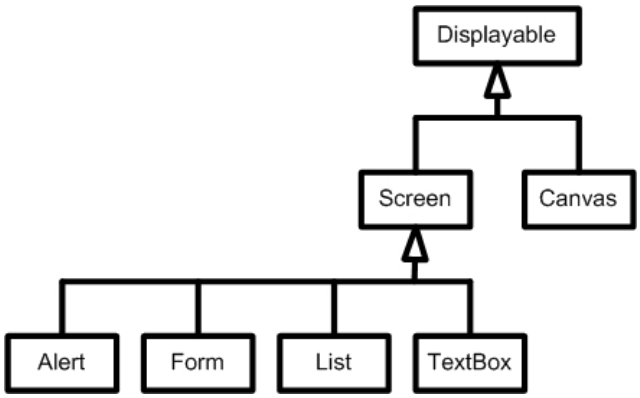
MIDP atau *Mobile Information Device Profile* adalah spesifikasi untuk sebuah profil J2ME. MIDP memiliki lapisan di atas CLDC, API tambahan untuk daur hidup aplikasi, antarmuka, jaringan dan penyimpanan persisten. Pada saat ini terdapat MIDP 1.0 dan MIDP 2.0. Fitur tambahan MIDP 2.0 dibanding MIDP 1.0 adalah API untuk multimedia. Pada MIDP 2,0 terdapat dukungan memainkan *tone*, *tone sequence* dan file WAV walaupun tanpa adanya *Mobile Media* API (MMAPI). Posisi MIDP pada arsitektur J2ME dapat dilihat pada gambar di atas. Berikut adalah perbandingan MIDP 1.0 dan MIDP 2.0 :

Tabel 2.4 Perbandingan MIDP 1.0 dan MIDP 2.0

Spesifikasi	MIDP 1.0	MIDP 2.0
Display	96 x 54	96 x 54
Kedalaman Display	1 -bit	1 -bit
Bentuk piksel (rasio aspek)	Mendekati 1:1	Medekati 1:1
Input	Keyboard dan touch screen	Keyboard dan touch screen
Memori	128 KB memori non-volatile untuk komponen MIDP. 8 KB memori non-volatile untuk data persistence yang dibuat oleh aplikasi. 32 KB memori volatile untuk JRE.	256 KB memori non-volatile untuk komponen MIDP. 8 KB memori non-volatile untuk data persistence yang dibuat oleh aplikasi. 128 KB memori volatile untuk JRE.

Jaringan	Dua arah, tanpa kabel (wireless)	Dua arah, tanpa kebel (wireless)
Library J2ME yang bukan merupakan *library* J2SE	javax.microedition.lcdui, javax.microedition.midlet, javax.microedition.rms	javax.microedition.lcdui, javax.microedition.midlet, javax.microedition.rms, javax.microedition.lcdui.game, javax.microedition.media, javax.microedition.pki
Multimedia		Memiliki kemampuan untuk memainkan file multimedia (suara dan video)

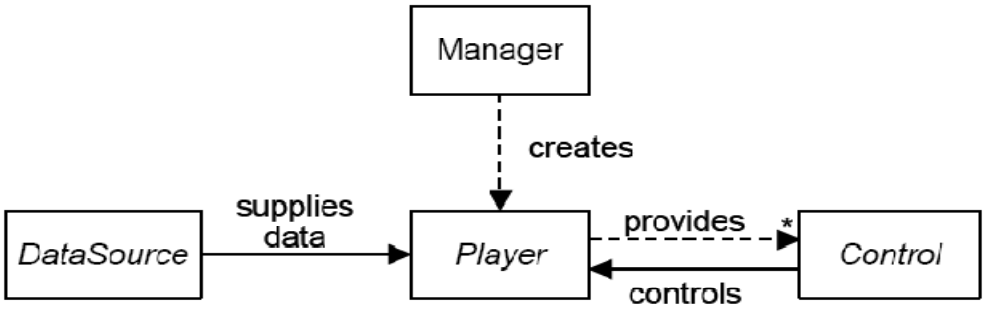
MIDP *User Interface* API memiliki API level tinggi dan level rendah. API level rendah berbasiskan penggunaan dari kelas abstrak *Canvas,* sedangkan kelas API level tinggi antara lain *Alert, Form, List* dan TextBox yang merupakan eksistensi dari kelas abstrak *Screen*. API level rendah lebih memberikan kemudahan kepada pengembang untuk memodifikasi sesuatu dengan kehendaknya, sedangkan API level tinggi biasanya hanya memberikan pengaksesan yang terbatas. Arsitektur antarmuka MIDP dapat dilihat pada gambar di bawah ini:



Gambar 2.23 MIDP User Interface

2.14.5 MMAPI

MMAPI dirancang untuk mendukung beragam tipe *file* media yang dapat dimainkan di dalam perangkat telepon selular, dan juga untuk menyediakan mekanisme pengambilan data (*data capturing*). MMAPI memfasilitasi sebuah kelas *factory* generik yang menginstansi objek-objek media *player* dari lokasi tertentu (objek-objek InputStream, maupun objek-objek DataSource). Sedangkan implementasi dari perangkat ponsel yang medukung MMAPI menyediakan kelas *player* konkrit untuk media-media yang didukung. *Player* dari perangkat tersebut memiliki fitur spesifik dan dapat dibuka melalui *interface* Control. Dan dapat berinteraksi dengan *player* dari sebuah ponsel melalui kontrol-kontrolnya, berikut gambaran yang menunjukkan arsitektur MMAPI :



Gambar 2.24 Arsitektur MMAPI

Kelas Manager adalah kelas *factory* statik di dalam MMAPI yang digunakan untuk menginstansi objek Player. Untuk membuat *instance* Player perlu memanggil *method* createPlayer() dari kelas Manager.

```
Manager.createPlayer( "capture://audio" );
Manager.createPlayer(stream, contentType);
```

Kelas DataSource mendefinisikan beberapa buah *method* abstrak yang mengizinkan *user* untuk menetukan bagaimana melakukan koneksi terhadap sumber data serta menjalankan atau menghentikan proses transfer data. Sebuah *instance* dari DataSource dapat berisi satu atau lebih objek SourceStream. DataSource menyembunyikan detail mengenai bagaimana data itu di baca dari *source*. DataSource dapat dibangun dari InputStream.

Spesifikasi MMAPI menyediakan interface Player, yang mendeklarasikan perilaku-perilaku umum dari semua kelas-kelas yang mengimplementasikan *interface* tersebut. Maka setiap jenis telepon selular akan mengimplementasikan *interface* tersebut secara berbeda untuk mengatasi tipe media yang akan dimainkan.

Kelas Control mendefinisikan fungsi untuk mengendalikan pemrosesan media. Setiap tipe media memiliki Control yang berbeda dari Player.

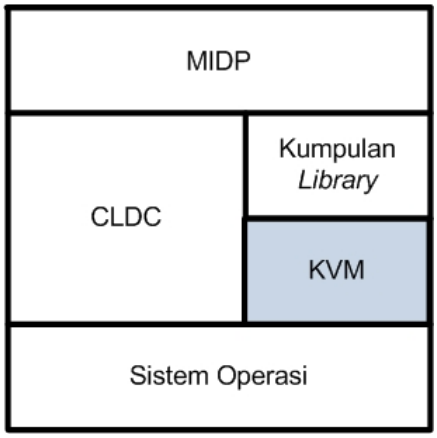
```
rc = (RecordControl) p.getControl( "RecordControl" );
return "RemoteAudioVolumeControl";
return "RemoteControl";
```

Untuk membuat aplikasi multimedia dalam ponsel, terlebih dahulu harus membuat *instance* dari objek Player berdasarkan sumber data (*DataResource*) yang ditentukan. Setelah objek Player terbetuk, maka dapat menggenerasi objek-objek

kontrol melalui objek Player yang ada. Berikut ini adalah keadaan-keadaan (*state*) yang dimiliki oleh objek *Player* :

1. CLOSED : keadaan ketika player telah melepas semua resource yang terikat, dan player tersebut tidak dapat digunakan kembali dengan memanggil method Player.Close().
2. UNREALIZED : keadaan ketika player telah diinstansi di *heap memory* namun *player* tersebut belum belum mangalokasikan resource apapun.
3. REALIZED : apabila *method* Player.realized() dipanggil dari keadaan UNREALIZED, maka *player* akan mendapatkan *resource* (berupa file media) yang diinginkan dan player akan berpindah ke keadaan REALIZED.
4. PREFETCHED : apabila *method* Player.prefetch() dipanggil, player akan menjalankan tugas-tugas untuk proses *start-up*. Setelah selesai, maka player akan berpindah ke keadaan PREFETCHED. Contoh tugas-tugas tersebut adalah mempersiapkan kontrol kamera, speaker, dan *resource* kontrol lainnya. Secara teori, *player* dapat dipindahkan dari keadaan PREFETCHED ke REALIZED dengan memanggil *method* Player.deallocate(). Namun pada kenyataannya, banyak jenis ponsel yang tidak mengimplementasikan *method* tersebut.
5. STARTED : dengan memanggil method Player.start(), maka dapat memulai menjalankan *player*. Setelah player tersebut berjalan, dapat dihentikan dengan memanggil *method* Player.stop() dan mengembalikan player ke keadaan PREFETCHED.

2.14.6 *Kilobyte Virtual Machine* (KVM)



Gambar 2.25 Arsitektur J2ME

KVM adalah paket JVM yang di desain untuk perangkat yang kecil. KVM mendukung sebagian dari fitur-fitur JVM, seperti misalnya KVM tidak mendukung operasi *floating-point* dan finalisasi objek. KVM diimplementasikan dengan menggunakan C sehingga sangat mudah beradaptasi pada tipe *platform* yang berbeda. Posisi KVM pada arsitektur J2ME dapat dilihat pada gambar di atas.

2.14.7 *C-Virtual Machine* (CVM)

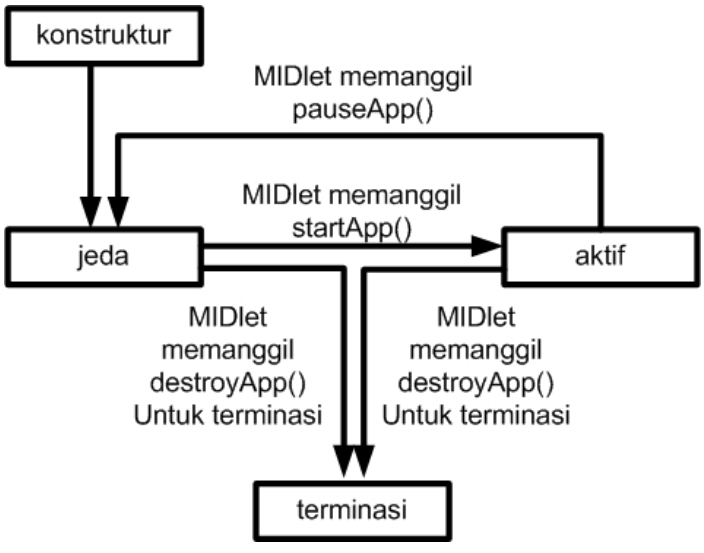
CVM adalah paket0020JVM optimal yang digunakan pada CDC. CVM mempunyai seluruh fitur dari *virtual machine* yang didesain untuk perangkat yang memerlukan fitur-fitur *Java* 2 *Virtual Machine*.

2.14.8 MIDlet

MIDlet adalah aplikasi yang ditulis untuk MIDP. Aplikasi MIDlet adalah bagian dari kelas javax.microedition.midlet.MIDlet yang didefinisikan pada MIDP. MIDlet berupa sebuah kelas abstrak yang merupakan sub kelas dari bentuk dasar aplikasi sehingga antarmuka antara aplikasi J2ME dan aplikasi manajemen pada perangkat dapat terbentuk.

2.14.8.1 Daur Hidup MIDlet

MIDlet terdiri dari beberapa metode yang harus ada, yaitu constructor(), protected void startApp() throws MIDletStateChangeException, protected void pauseApp(), protected void destroyApp(boolean unconditional) throws MIDletStateChangeExeption. Alur hidup MIDlet dapat dilihat pada gambar di bawah ini :



Gambar 2.26 Alur Hidup MIDlet

Ketika MIDlet dijalankan maka akan diinisialisi dengan kondisi *pause* dan dijalankan pauseApp(). Kondisi berikutnya adalah fungsi MIDlet dijalankan, yaitu pada startApp(). Metode yang ada tersebut diimplementasikan sebagai *protected*, hal ini dimaksudkan agar MIDlet tidak dapat memanggil metode tersebut. Pada saat pemakai keluar dari MIDlet, maka metode destroyApp() akan dijalankan sebelum MIDlet benar-benar tidak berjalan lagi. Metode notifyDestroyed() akan dipanggil sebelum MIDlet benar-benar tidak berjalan lagi, destroyApp() akan memanggil notifyDestroyed(), dan notifyDestroyed() akan memberi tahu *platform* untuk menterminasi MIDlet dan Membersihkan semua sumber daya yang mengacu pada MIDlet.

Dalam implementasinya, MIDlet memiliki struktur direktori antara laian sebagai berikut :

1. *src*

   menyimpan *source code* untuk MIDlet dan kelas lain yang diperlukan.

2. *res*

   Menyimpan sumber daya yang dibutuhkan oleh MIDlet, seperti misalnya gambar icon.

3. *lib*

   menyimpan file JAR atau ZIP yang berisi *library* tambahan yang dibutuhkan MIDlet.

4. *bin*

   menyimpan file JAR, JAD dan file *manifest* yang berisi muatan komponen MIDlet.

2.14.8.2 Atribut MIDlet

MIDlet juga memiliki Atribut *package* sebagai berikut :

Tabel 2.5 Nilai dan Fungsi Atribut MIDlet

Nama Atribut	Nilai dan Fungsi
MIDlet-Name	Nama MIDlet yang juga sebagai nama untuk file JAR-nya.
MIDlet-Version	Nomor versi dari MIDlet yang juga sebagai nomor versi pada file JAR-nya.
MIDlet-Vendor	Nama provider MIDlet.
MIDlet-n	Atribut yang mendeskripsikan MIDlet. Nilai n diganti dengan nilai numerik dimulai dari angka 1. Format nilai dari atribut ini terkait dengan atribut yang didesktipsikan selanjutnya.
Microedition-Profile	Versi dari spesifikasi MIDlet yang dapat berjalan. Dapat lebih dari saru versi yang dipisah dengan spasi.
Microedition-Configuration	Konfigurasi J2ME yang dibutuhkan untuk MIDlet.
MIDlet-Description	Deskripsi MIDlet.
MIDlet-Icon	Icon yang digunakan MIDlet.
MIDlet-Info-URL	URL dari file yang berisi informasi mengenai MIDlet.
MIDlet-Data-Size	Nilai minimum dari besarnya tempat penyimpanan persisten yang dibutuhkan (dalam byte) tidak termasuk yang dibutuhkan untuk menginstall aplikasi.
MIDlet-Jar-URL	URL dari file JAR.
MIDlet-Jar-Size	Ukuran file JAR dalam byte.
MIDlet-Install-Notify	Sebuah URL yang digunakan untuk melaporkan sukses atau gagalnya penginstallan MIDlet dari remote Server.
MIDlet-Delete-Confirm	Pesan yang ditampilkan ke pemakai sebelum MIDlet dihapus dari alat dimana MIDlet diisntall.
MIDlet-specificattributes	Pengembang MIDlet dapat menyediakan konfigurasi minimum untuk MIDlet dengan memasukkan atribut ini.

2.14.9 JAD (*Java Application Descriptor*)

Digunakan untuk mendeskripsikan isi aplikasi untuk keperluan pemetaan. File JAD berisi deskripsi file JAR (*Java Archive*) dan pemetaan atribut MIDlet, sedangkan file JAR berisi kumpulan kelas dan *resouce*.

2.14.10OTA (Over The Air)

OTA mengacu pada beberapa teknologi jaringan tanpa kabel. Dengan menggunakan OTA, provider MIDlet dapat menginstall MIDlet pada *web server* dan menyediakan link untuk men-*download* via WAP atau *internet microbrowser*.

2.15 NetBeans IDE 6.0

NetBeans adalah salah satu *open source* IDE berbasiskan *Java* dari *Sun Microsistems* yang berjalan diatas *Swing*. *Swing* merupakan sebuah teknologi Java untuk pengembangan aplkasi desktop yang *multiplatform*.

NetBeans IDE memungkinkan untuk membuat *project Java* dengan cepat dan gratis. NetBeans memiliki GUI *Designer* yang membuat para pembuat aplikasi *Java* menjadi lebih cepat.

NetBeans merupakan versi *open source* dari *Fort* IDE, sebuah IDE untuk pengembangan aplikasi *Java* yang dikembangkan oleh *Forte Software*, dimana *Forte* saat ini sudah diakuisisi oleh *Sun* dan menjadi salah satu produk penguat jajaran divisi *Software Sun*.

Berikut ini adalah berbagai komponen GUI yang digunakan dalam pemrograman *Java* pada NetBeans :

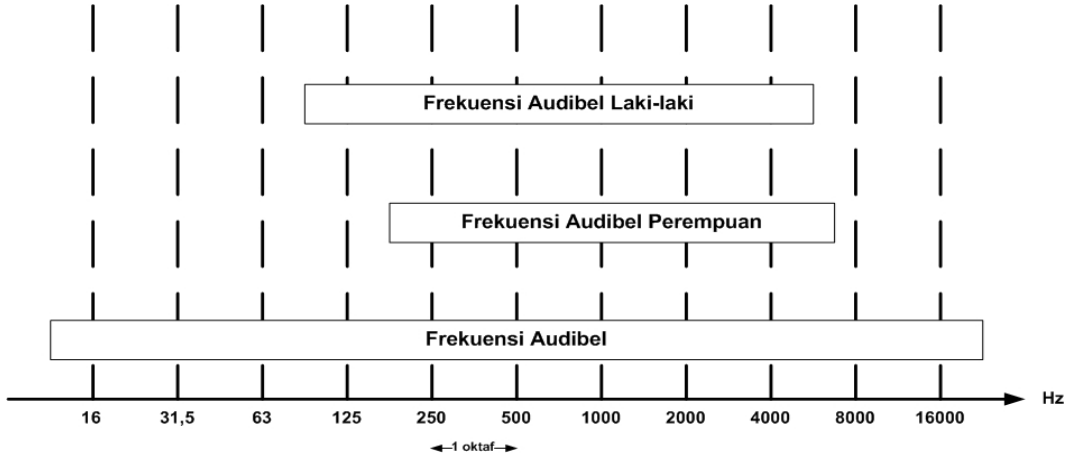
Tabel 2.6 Komponen GUI pada NetBeans

Komponen	Nama	Deskripsi
Swing Controls	Label	Untuk menampilkan teks atau gambar.
	Button	Untuk membuat subuah tombol yang akan digunakan untuk tujuan tertentu.
	Check Box	Digunakan untuk memilih salah satu atau beberapa item dari banyak pilihan tem yang disediakan.
	Radio Button	Untuk memilih item, dimana hanya salah satu item saja yang dipilih.
	Combo Box	Untuk memasukkan banyak pilihan item di dalamnya dan berbentuk seperti sebuah scroll. Sering juga disebut drop-down list.
	List	Mirip dengan combo box, hanya berbeda pada tampilannya saja. List berbentuk memanjang ke bawah. Secara fungsional tidak jauh berbeda dengan combo box, hanya list lebih memakan tempat jika dimasukkan dalam form.
	Progress Bar	Untuk memperlihatkan sudah sampai di mana suatu aplikasi berjalan. Sering digunakan saat aplikasi mulai loading.
	Table	Untuk menyisipkan tabel pada form.
Swing Menus	Menu Bar	Untuk menyisipkan sebuah wadah yang digunakan sebagai menu.
	Menu Item	Untuk mengisi Menu Bar yang berupa aneka macam menu.
Swing Windows	Dialog	Untuk membuat jendela dialog.
	Frame	Untuk menyisipkan frame pada form.
Abstract Window Toolkit (AWT)	Label	Sama dengan Label pada Swing Controls namun hanya bisa digunakan untuk teks.
	Button	Sama dengan Buton pada Swing Controls, digunakan untuk membuat tombol.
	Text Field	Untuk memasukkan teks yang bisa diedit kembali atau sebagai input dari suatu aplikasi.
	Check Box	Sama dengan Check Box pada Swing Controls, digunakan untuk memilih satu atau beberapa item.
	Choice	Sama dengan Combo Box pada Swing Controls.
	List	Sama dengan list pada Swing Controls.

2.16 Spektrum Frekuensi Suara Manusia

Suara merupakan sebuah fenomena disik yang menstimulasi indera pendengaran. Pada manusia, suara dapat didengar ketika frekuensi getaran berada diantara 15 Hz hingga 20 KHz. Hertz (Hz) merupakan unit frekuensi yang berarti

ahwa banyaknya getaran atau *cycle* tiap detik. Tingkat sensitifitas sistem pendengaran manusia berada pada rentang frekuensi 1000 Hz hingga 4000 Hz.



Gambar 2.27 Ilustrasi Suara dalam Domain Frekuensi

Frekuensi audibel dibagi menjadi beberapa tingkat rentang frekuensi, yaitu dalam satuan oktaf yang berarti interval antara dua frekuensi yang memiliki rasio sebesar 2:1. Sebagai referensi, besar frekuenai untuk nada middle C adalah 250 Hz.