

## **BAB II**

### **LANDASAN TEORI**

#### **II.1 Penguat Sinyal (*Repeater*)**

*Repeater* adalah suatu alat elektronik yang menerima sinyal lemah dan mentransmisikannya kembali menjadi sinyal yang lebih kuat, sehingga dapat mencapai jarak yang lebih jauh (sumber: <http://en.wikipedia.org/wiki/Repeater>). Fungsi utama dari *repeater* adalah menerima sinyal dari satu segmen kabel LAN dan memancarkannya kembali dengan kekuatan yang sama dengan sinyal asli pada segmen (satu atau lebih) kabel LAN yang lain. Hal ini diperlukan karena sinyal-sinyal mengalami perlemahan dan perubahan bentuk selama transmisi (sumber: [www.oke.or.id/tutorial/jarkom.pdf](http://www.oke.or.id/tutorial/jarkom.pdf)).

Sebuah *repeater* tidak melakukan translasi atau filterisasi paket. Oleh karena *repeater* tidak melakukan apa-apa terhadap filterisasi atau pembatasan *traffic*, *repeater* beroperasi pada *physical layer*, lapisan (*layer*) pertama model OSI. *Repeater* seharusnya dilihat terutama sebagai penghubung *workstation* yang jauh. Dengan kata lain, *repeater* seharusnya digunakan untuk menambahkan jarak dari *network*, bukan kepadatannya. Dengan adanya *repeater* ini, jarak antara dua jaringan komputer bisa diperjauh.

#### **II.2 Teknologi *Bluetooth***

*Bluetooth* adalah sebuah teknologi komunikasi *wireless* (tanpa kabel) yang beroperasi dalam pita 2,4 GHz *unlicensed ISM (Industrial, Scientific, and Medical)* dengan menggunakan sebuah *frequency hopping transceiver* yang mampu menyediakan layanan komunikasi data dan suara secara *real-time* antara *host-host bluetooth* dengan jarak jangkauan layanan yang terbatas. *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 IEEE 802.11,

hanya saja pada *bluetooth* mempunyai jangkauan jarak layanan yang lebih pendek dan kemampuan *transfer* data yang lebih rendah. Pada dasarnya *bluetooth* diciptakan bukan hanya menggantikan atau menghilangkan penggunaan kabel di dalam 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.

Nama *bluetooth* berawal dari proyek prestisius yang dipromotori oleh perusahaan seperti Ericsson, IBM, Intel, dan Toshiba. Proyek ini di awal tahun 1998 dengan kode nama *bluetooth*, karena terinspirasi oleh seorang raja Viking (Denmark) yang bernama Harald Blatand. Raja Harald Blatand ini berkuasa pada abad ke-10 dengan menguasai sebagian besar daerah Denmark dan daerah Skandinavia pada masa itu. Dikarenakan daerah kekuasaannya yang luas, raja Harald Blatand ini membiayai para ilmuwan dan insinyur untuk membangun sebuah proyek berteknologi metamorfosis yang bertujuan untuk mengontrol pasukan dari suku-suku di daerah Skandinavia tersebut dari jarak jauh. Maka untuk menghormati ide raja Viking tersebut, proyek ini diberi nama Blatand yang berarti *bluetooth*.

Kali pertama dirilis untuk *bluetooth* versi 1.0 dan 1.0 B pada tanggal 26 Juli 1999 produk ini belum sempurna, karena mempunyai banyak masalah dan perusahaan manufaktur pendukungnya mengalami kesulitan dalam menerapkan teknologi ini pada produk mereka. Oleh sebab itu, pada bulan Oktober di tahun yang sama, dirilis *bluetooth* versi 1.2 dengan perbaikan antara lain:

- 1) Digunakannya *mask* pada perangkat *Hardware Device Address* (BD-ASSR) untuk melindungi pengguna dari *identity snooping* (pengintai) maupun *tracker*.
- 2) Penggunaan protokol tanpa nama sudah tersedia namun tidak diimplementasikan, sehingga konsumen tidak dapat menggunakannya.

3) *Adaptive Frequency Hopping (AFH)*, dengan memperbaiki daya tahan dari gangguan frekuensi radio yang digunakan oleh banyak orang di dalam *hopping sequence*.

4) Transmisi berkecepatan tinggi.

Dengan bertambahnya perusahaan manufaktur pendukung, antara lain *3Com, Ericsson, IBM, Intel, Lucent Technologies, Microsoft, Motorola, Nokia*, dan *Toshiba* yang lebih dikenal dengan nama *The Bluetooth SIG (Special Interest Group)*, maka teknologi ini pun mengalami perbaikan-perbaikan untuk versi 2.0-nya. Beberapa tambahan tersebut antara lain:

1) Diperkenalkannya *Non-hopping narrowband channels*. Pada *channel* ini bisa digunakan untuk memperkenalkan layanan *profile bluetooth* oleh berbagai *device* dengan volume yang sangat tinggi dari perangkat *bluetooth* secara simultan.

2) Tidak dienkripsinya informasi yang bersifat umum secara *realtime*, sehingga dasar kemacetan trafik informasi dan laju trafik ke tujuan dapat dihindari waktu ditransmisikan oleh perangkat dengan melewati setiap host dengan kecepatan tinggi.

3) Koneksi berkecepatan tinggi.

4) *Multiple speeds level*.

*Bluetooth* menggunakan salah satu dari dua jenis frekuensi *Spread Spectrum Radio* yang digunakan untuk kebutuhan wireless. Jenis frekuensi yang digunakan adalah *Frequency Hopping Spread Spectrum (FHSS)*, sedangkan yang satu lagi, yaitu *Direct Sequence Spread Spectrum (DHSS)* digunakan oleh IEEE802.11xxx. transceiver yang digunakan oleh *bluetooth* bekerja pada frekuensi 2.4 GHz *unlicensed*.

Secara umum karakteristik dari teknologi *bluetooth* dapat dilihat pada tabel 2.1.

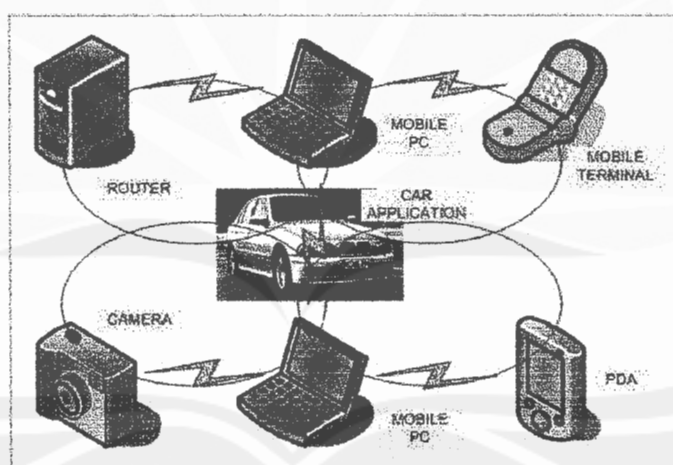
Tabel 2.1 Karakteristik utama teknologi *bluetooth*

Karakteristik	Deskripsi
<i>Physical Layer</i>	<i>Frequency Hopping Spread Spectrum (FHSS)</i>
<i>Frequency Band</i>	2.4 – 2.4835 GHz (ISM Band)
<i>Hop Frequency</i>	1.600 hop/detik
Kecepatan data	1 Mbps (raw)
Keamanan Data dan Jaringan	<ul style="list-style-type: none"> <li>- Tiga Mode Keamanan</li> <li>- Dua tingkat <i>device trust</i></li> <li>- Tiga tingkat keamanan layanan</li> <li>- Enkripsi <i>stream</i> untuk <i>confidentiality</i></li> <li>- <i>Challenge response</i> untuk <i>authentication</i></li> <li>- PIN-derived key</li> <li>- <i>Limited management</i></li> </ul>
Jangkauan	Sekitar 10 meter dan dapat diperluas sampai 100 meter
<i>Throughput</i>	720 kbps
Kelebihan	<ul style="list-style-type: none"> <li>- Tanpa kabel</li> <li>- Sinyal dapat menembus tembok atau halangan</li> <li>- Biaya relatif murah</li> <li>- Berdaya rendah, dan</li> <li>- <i>Hardware</i> yang relatif kecil</li> </ul>
Kekurangan	<ul style="list-style-type: none"> <li>- Kemungkinan terjadinya interferensi dengan teknologi lain yang menggunakan ISM band</li> <li>- Kecepatan pengiriman data relatif rendah</li> <li>- Sinyal yang lemah di luar batasan</li> </ul>

Jangkauan *bluetooth* tergantung dari *class radio*. Kebanyakan *device* menggunakan *class 2 radio* yang menghasilkan jangkauan sampai 10 meter dalam lingkungan yang bebas hambatan. Jangkauan ini cukup untuk mengganti aplikasi yang menggunakan kabel.

Sebuah perangkat yang memiliki teknologi *wireless bluetooth* akan mempunyai kemampuan untuk melakukan pertukaran informasi dengan

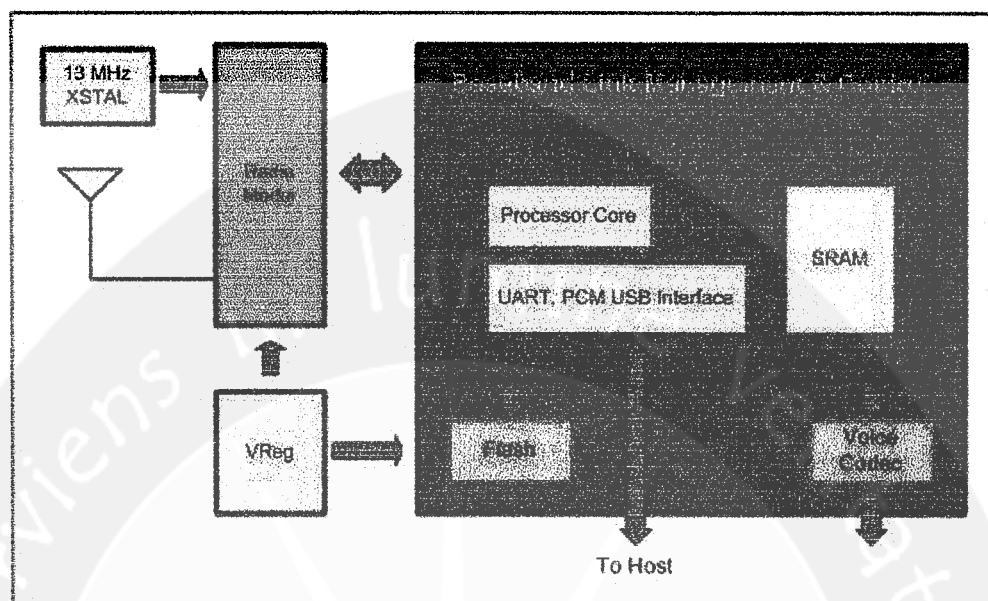
jarak jangkauan sampai dengan 10 meter (~30 feet), bahkan untuk daya kelas 1 bisa sampai pada jarak 100 meter. Sistem *bluetooth* menyediakan layanan komunikasi *point to point* maupun komunikasi *point to multipoint*. Produk *bluetooth* dapat berupa *PC card* atau *USB adapter* yang dimasukkan ke dalam perangkat. Perangkat-perangkat yang dapat diintegrasikan dengan teknologi *bluetooth* antara lain : *mobile PC*, *mobile phone*, *PDA (Personal Digital Assistant)*, *headset*, *kamera digital*, *printer*, *router* dan masih banyak peralatan lainnya. Aplikasi-aplikasi yang dapat disediakan oleh layanan *bluetooth* ini antara lain : *PC to PC file transfer*, *PC to PC file synch (notebook to desktop)*, *PC to mobile phone*, *PC to PDA*, *wireless headset*, *LAN connection via ethernet access point* dan sebagainya. Contoh modul aplikasi beberapa peralatan yang kemungkinan dapat menggunakan teknologi *bluetooth* dapat dilihat seperti gambar berikut (sumber: [www.ilmukomputer.com/2006/09/04/pengantar-teknologi-bluetooth/](http://www.ilmukomputer.com/2006/09/04/pengantar-teknologi-bluetooth/) - 29k -):



Gambar 2.1 Contoh modul aplikasi beberapa *Bluetooth*

Sistem *bluetooth* terdiri dari sebuah *radio transceiver*, *baseband link Management dan Control*, *Baseband (processor core, SRAM, UART, PCM USB Interface)*, *flash* dan *voice code*. sebuah *link manager*. *Baseband link controller* menghubungkan perangkat keras radio ke *baseband processing* dan *layer* protokol fisik. *Link manager* melakukan aktivitas-aktivitas protokol tingkat tinggi seperti melakukan *link setup*, autentikasi dan konfigurasi. Secara umum blok fungsional pada sistem *bluetooth* secara

umum dapat dilihat pada gambar berikut ini (sumber: [www.ilmukomputer.com/2006/09/04/pengantar-teknologi-bluetooth/](http://www.ilmukomputer.com/2006/09/04/pengantar-teknologi-bluetooth/) - 29k -):



Gambar 2.2 *Bluetooth system blocks*

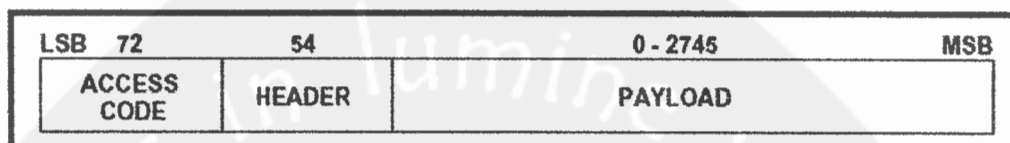
### IL2.1 Format Paket *Bluetooth*

*Bluetooth* menggunakan format paket seperti tampak pada gambar 2.3 paket terbagi ke dalam tiga bagian yaitu:

- 1) 72 bit kode akses
- 2) 54 bit header
- 3) Payload sebesar 0 – 2745 bit

Kode akses mempunyai tiga fungsi yaitu sinkronisasi, DC *offset compensation*, dan identifikasi *piconet*. *Sliding correlator* digunakan kode akses untuk sinkronisasi. Kode akses juga memuat *sequence* sebesar 4 bit untuk DC *offset compensation*. *Sequence* ini terletak di awal kode akses. Tiap *piconet* ditugaskan suatu pengenal yang diperoleh dari identifikasi perangkat *master* yang menghubungkan tiap paket terpisah ke suatu *piconet*. Proses untuk memperoleh nilai identifikasi *piconet* menjamin terdapatnya jarak minimum *Hamming* antara pengenal.

*Header* paket mengandung informasi berkaitan dengan hubungan antara *piconet*. Informasi yang termasuk di dalam *header* antara lain alamat anggota *piconet* (0-7), jenis paket, dan *general flow control*. *General flow control* terdiri dari *sequence number* dan *acknowledgment bit*. *Header* juga mengandung *header error control word*. *Payload* paket mempunyai besar yang bervariasi dan diproteksi dengan FEC (sumber: <http://forum.nokia.com>).



Gambar 2.3 Format paket *bluetooth*

Beberapa jenis paket telah dispesifikasikan untuk mendukung tiap jenis saluran. Jenis-jenis paket tersebut adalah sebagai berikut:

1) Paket tipe umum

Terdapat lima jenis paket yaitu ID, NULL, POLL, FHS, dan DMI.

Paket ID terdiri dari *reduced-length access code* sebesar 68 bit tanpa *header* dan *payload*. Paket ini digunakan untuk melakukan aktivitas seperti *paging*, *placing inquiries* dan mengirim respon. Paket ID merupakan satu-satunya paket yang mempunyai *reduced-length access code*. Paket ini sangat handal karena menggunakan *sliding-correlator* untuk penerimaan kode akses.

Paket NULL dan POLL terdiri dari kode akses dan *header* tanpa *payload*. Yang membedakan kedua paket ini adalah paket POLL meminta respon, sedangkan paket NULL tidak.

Paket FHS terdiri *payload* sebesar 240 bit termasuk penggunaan kode *Hamming*. Paket ini digunakan untuk mendukung beberapa tugas seperti sinkronisasi *clock*, pengaturan *paging*, dan deskripsi kode akses.

Paket DMI adalah paket yang sesuai dengan arsitektur paket ACL dan dapat dipertimbangkan sebagai paket ACL tetapi tidak terbatas pada saluran ACL saja. Paket ini digunakan untuk memberikan kendali informasi secara asinkron melalui saluran SCO dan juga membawa data atau kendali informasi melalui saluran ACL.

## 2) Paket ACL

Terdapat 7 jenis paket ACL, yaitu AUX1, DM1, DH1, DM3, DH3, DM5, dan DH5, yang semuanya dirancang untuk mendukung komunikasi data. Kecuali untuk paket AUX, semua paket diproteksi dengan skema ARQ.

## 3) Paket SCO

Paket SCO terdiri dari DV, HV1, HV2, dan HV3. paket SCO digunakan untuk membawa informasi suara. Kecuali untuk paket DV, paket SCO tidak menggunakan skema ARQ seperti pada paket ACL.

### II.2.2 Karakteristik *Transceiver Bluetooth*

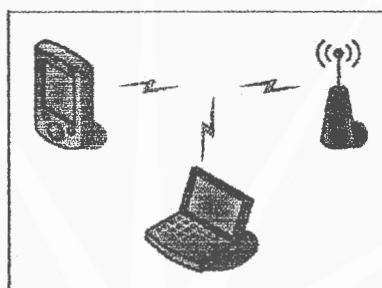
*Transceiver bluetooth* beroperasi pada frekuensi 2,4 GHZ *ISM (Industrial Scientific Medical)* yang secara tepat sebenarnya berada pada range frekuensi 2.400-2.483 GHZ yang terdiri atas 79 kanal. Kecepatan transfer data maksimum yang dapat dicapai adalah 1 Mbps. *Bluetooth* menggunakan kombinasi teknologi *packet* dan *circuit* untuk proses transmisinya. Masing-masing kanal tersebut dibagi lagi dalam *time slot* yang berselang selama 625  $\mu$ s. Setelah satu *packet* dikirim lewat sebuah frekuensi, maka kedua peralatan *Bluetooth* yang sedang melakukan komunikasi melakukan *tune* ulang dengan frekuensi yang berbeda. Secara efektif akan melakukan lompatan pada kecepatan 1600 lompatan perdetik melalui beberapa *time slot* yang berbeda. Inilah yang disebut dengan *frequency hopping*. Jarak jangkauan dari peralatan *bluetooth* sangat bergantung pada kelas daya dari peralatan radio yang digunakan. Untuk peralatan *mobile* umumnya digunakan peralatan radio kelas 2 yang memiliki jangkauan sampai 10 meter. Kelas ini berkaitan dengan *output power* yang digunakannya. Kelas 1 memiliki output power yang lebih besar. Dalam *transceiver bluetooth* ada tiga kelas pembagian daya yaitu :

- 1) Daya kelas 1 beroperasi antara 100 mW (20dBm) dan 1mW (0dBm) dan didesain untuk peralatan dengan jangkauan yang jauh hingga 100m.



- 2) Daya kelas 2 beroperasi antara 2,5 mW (4dBm) dan 0,25mW (-6dBm) dan didesain untuk peralatan dengan jangkauan yang jauh hingga 10m.
- 3) Daya kelas 3 beroperasi pada 1 mW (0dBm) dan didesain untuk peralatan dengan jangkauan pendek atau sekitar 1m.

Dari ketiga kelas di atas, yang sekarang ini dikembangkan adalah daya kelas 2 dengan jangkauan sekitar 10 m. Untuk mencegah bentrokan dengan berbagai daya dari peralatan yang berbeda, maka memungkinkan untuk menaikkan atau meningkatkan daya dari peralatan melalui *link manager protokol (LMP)* (sumber: [www.ilmukomputer.com/2006/09/04/pengantar-teknologi-bluetooth/](http://www.ilmukomputer.com/2006/09/04/pengantar-teknologi-bluetooth/) - 29k -).



Gambar 2.4 Interferensi yang terjadi pada *bluetooth*

Interferensi akan terjadi karena adanya tabrakan antara paket dari peralatan *bluetooth* yang digunakan dengan peralatan lain yang bekerja pada frekuensi yang berdekatan atau bahkan sama, sehingga saling *overlap* dalam domain waktu dan frekuensi. *Frequency hopping* juga mendukung munculnya interferensi ini, untuk itu beberapa peralatan *bluetooth* dilengkapi dengan sebuah teknologi akses yang disebut *frequency hop spread spectrum (FHSS)*.

### II.2.3 *Spread Spectrum*

Bagaimana data bisa bergerak di udara? *Wireless LAN* mengirimkan data melalui udara dengan menggunakan gelombang elektromagnetik dengan teknologi yang dipakai adalah *Spread-Spectrum Technology (SST)*. Dengan teknologi ini memungkinkan beberapa *user* menggunakan pita

frekuensi yang sama secara bersamaan. SST ini merupakan salah satu pengembangan teknologi *Code Division Multiple Access* (CDMA). Dengan urutan kode (*code sequence*) yang unik data dikirim ke udara dan diterima oleh tujuan yang berhak dengan kode tersebut. Dengan teknologi *Time Division Multiple Access* (TDMA) juga bisa diaplikasikan (data dikirim karena perbedaan urutan waktu/*time sequence*). Dalam teknologi SST ada dua pendekatan yang dipakai, yaitu:

- 1) *Direct Sequence Spread Spectrum* (DSSS), sinyal dikirim dalam pita frekuensi tertentu yang tetap sebesar 17 MHz. Prinsip dari metode *direct sequence* adalah memancarkan sinyal dalam pita yang lebar (17 MHz) dengan pemakaian pelapisan (*multiplex*) kode/*signature* untuk mengurangi *interferensi* dan *noise*. Untuk perangkat *wireless* yang bisa bekerja sampai 11 Mbps membutuhkan pita frekuensi yang lebih lebar sampai 22 MHz. Pada saat sinyal dipancarkan setiap paket data diberi kode yang unik dan berurut untuk sampai di tujuan, di perangkat tujuan semua sinyal terpancar yang diterima diproses dan disaring sesuai dengan urutan kode yang masuk. Kode yang tidak sesuai akan diabaikan dan kode yang sesuai akan diproses lebih lanjut.
- 2) *Frequency Hopping Spread Spectrum* (FHSS), sinyal ditransfer secara bergantian dengan menggunakan 1MHz atau lebih dalam rentang sebuah pita frekuensi tertentu yang tetap. Prinsip dari metode *frequency hopping* adalah menggunakan pita yang sempit yang bergantian dalam memancarkan sinyal radio. Secara periodik antara 20 sampai dengan 400ms (milidetik) sinyal berpindah dari kanal frekuensi satu ke kanal frekuensi lainnya.

Dengan teknologi DSSS maka untuk satu perangkat akan bekerja menggunakan 4 kanal (menghabiskan 20MHz, tepatnya 17MHz). Dalam implementasinya secara normal pada lokasi dan arah yang sama hanya 3 dari 13 kanal DSSS yang bisa dipakai. Parameter lain yang memungkinkan penggunaan lebih dari 3 kanal ini adalah penggunaan antena (*directional antenna*) dan polarisasi antena itu sendiri (*horisontal/vertikal*). Penggunaan

antena *Omni-directional* akan membuat sinyal ditransfer ke seluruh arah (360 derajat).

Teknologi FHSS ditujukan untuk menghindari *noise/gangguan* sinyal pada saat sinyal ditransfer, secara otomatis perangkat FHSS akan memilih frekuensi tertentu yang lebih baik untuk transfer data. Kondisi ini menjadikan satu keuntungan dibandingkan dengan DSSS.

#### II.2.4 Kelebihan *Bluetooth*

Teknologi *bluetooth* memberikan beberapa keuntungan kepada pengguna. Dengan metode jaringan *ad hoc*, komunikasi menggunakan *bluetooth* sangat menarik serta meningkatkan efisiensi dan mengurangi biaya. Kelebihan teknologi *bluetooth* antara lain:

1) Tanpa kabel

Teknologi *bluetooth* menggantikan kabel dengan hubungan komunikasi yang bervariasi seperti hubungan *mouse, keyboard, printer, modem, wireless, headset, microphone* yang terhubung ke komputer atau *mobile phone*.

2) Kemudahan dalam pertukaran *file*

*Bluetooth* memungkinkan pertukaran *file* di antara dua perangkat. Sebagai contoh, laptop para peserta rapat dapat saling bertukar informasi/data dengan peserta lainnya. *Mobile phone* berteknologi *bluetooth* dapat bertindak sebagai modem bagi laptop.

3) Sinkronisasi tanpa kabel

*Bluetooth* menyediakan sinkronisasi tanpa kabel secara otomatis dengan perangkat *bluetooth* lainnya. Sebagai contoh, informasi pribadi di dalam *address book* dan *date books* dapat disinkronisasi antara PDA, laptop, *mobile phone*, dan perangkat lainnya.

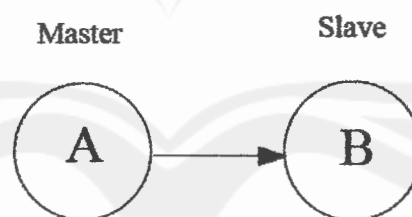
4) Hubungan internet

Teknologi *bluetooth* didukung oleh banyak perangkat dan aplikasi antara lain *mobile phone*, PDA, dan laptop. Suatu hubungan internet dapat dibentuk jika perangkat-perangkat tersebut dapat saling berhubungan.

### II.2.5 Topologi Jaringan *Bluetooth*

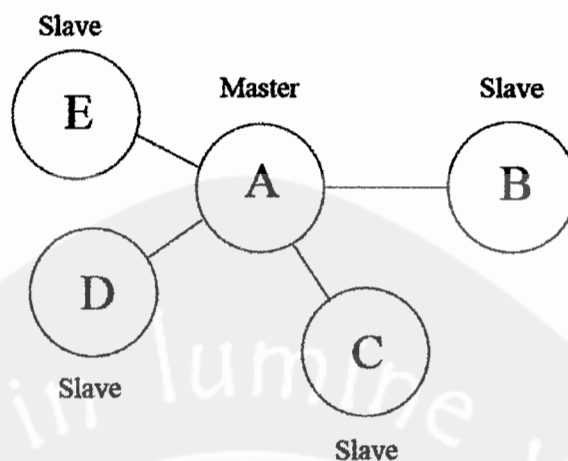
Teknologi *bluetooth* adalah teknologi yang berbasis pada jaringan ad hoc yang secara dinamik dapat berhubungan dengan perangkat yang terpisah. Disebut jaringan ad hoc karena tidak mempunyai jaringan konfigurasi yang tetap. Setiap kali ada tugas maka jaringan akan dibangun hanya untuk mengerjakan tugas saat itu dan kemudian dibongkar setelah pengiriman data telah selesai.

Jaringan *bluetooth* disebut juga *piconet*. Dalam kasus sederhana, dapat diartikan 2 *device* yang terkoneksi. *Device* yang menginisialisasi koneksi disebut *master* dan *device* lainnya adalah *slave*. Dalam jaringan *piconet* ini, koneksi yang terjadi dapat berbentuk *point-to-point application*, artinya pada aplikasi tersebut koneksi terjadi hanya antara 2 *device* saja, dengan sebuah *master* dan satunya lagi sebagai *slave*, seperti yang ditunjukkan pada gambar berikut (sumber: <http://wireless.klings.org>):



Gambar 2.5 Bentuk jaringan *piconet point-to-point*

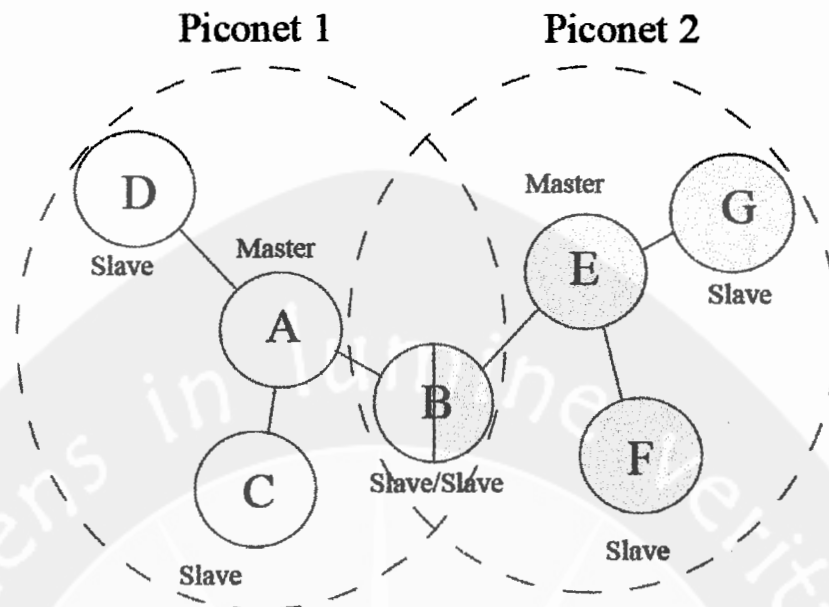
Selain itu, koneksi pada aplikasi juga dapat terjadi antara sebuah *master* dengan *slave* lain maksimal 7 buah *slave* yang disebut dengan *point-to-multipoint application*. Pada aplikasi *point-to-multipoint*, bentuk *piconet* seperti bentuk *star network*, dengan *master* sebagai node pusat, seperti yang ditunjukkan pada gambar berikut (sumber: <http://wireless.klings.org>):



Gambar 2.6 Bentuk jaringan *piconet point-to-multipoint*

*Slave* hanya mungkin mengirimkan data ketika diberikan hak oleh *master device (master piconet)* saat pengiriman, juga *slave* tidak berkomunikasi secara langsung satu dengan yang lainnya, tetapi semua komunikasi harus dilewatkan melalui *master*. *Slave* melakukan sinkronisasi *frequency hopping* mereka dengan *master* menggunakan *master's clock* dan *bluetooth address*.

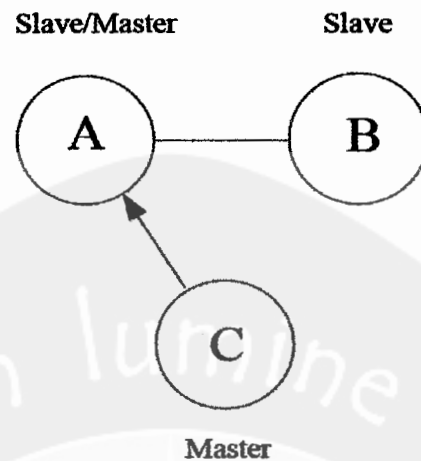
Sebuah *device* dalam satu *piconet* dapat berkomunikasi dengan *device* lain dalam *piconet* lainnya. Hubungan tersebut membentuk jaringan yang disebut *scatternet*. Secara sederhana, *scatternet* dapat diartikan koneksi yang terjadi antara 2 *piconet* atau lebih. Untuk lebih jelasnya lihat gambar contoh *scatternet* berikut (sumber: <http://wireless.klings.org>) :



Gambar 2.7 Scatternet

Pada gambar di atas ada satu *intermediate node* yang menghubungkan 2 *piconet*. *Intermediate node* harus *time-share*, artinya ia harus mengikuti *frequency hopping* dalam satu *piconet* saat itu. Hal ini mengurangi jumlah dari slot waktu yang tersedia untuk pengiriman data antara *intermediate node* dan *master*, sedikitnya memotong setengah dari waktu pengiriman.

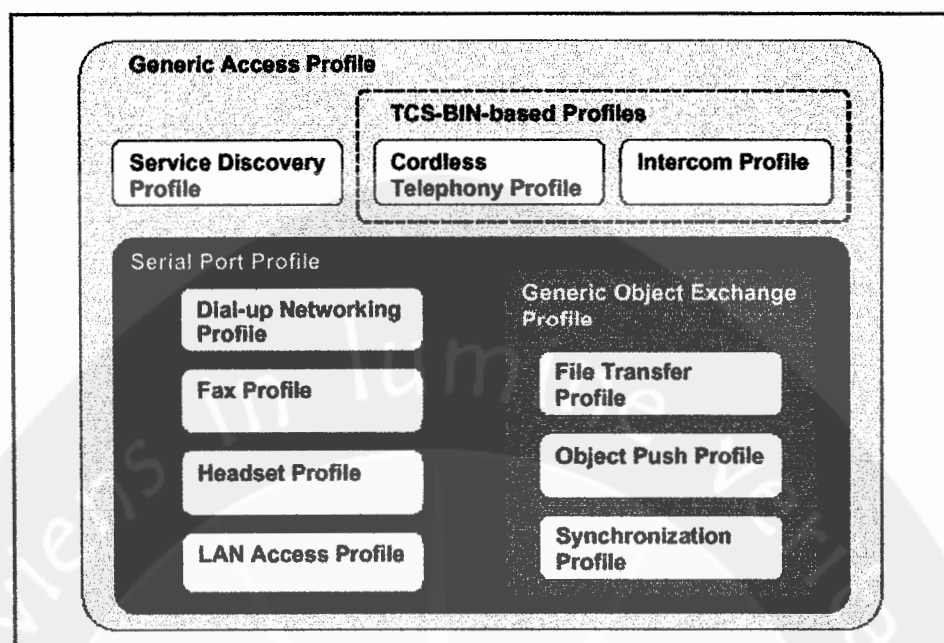
Pergantian peran memungkinkan 2 *device* untuk mengganti peran dalam suatu *piconet*. Contohnya, ada 2 buah *device* A dan B, *device* A melakukan koneksi ke *device* B seperti yang ditunjukkan pada gambar 2.2 di atas. Kemudian ada *device* C yang ingin bergabung ke *piconet* tersebut. *Device* C melakukan koneksi ke *master device*, A. Selama *device* A menginisialisasi koneksi, secara otomatis akan menjadi *master* dari koneksi antara *device* C dan A. Sekarang ada 2 *master*, maka ada 2 *piconet* dengan *device* A sebagai *intermediate node* di antara 2 *piconet* tersebut. *Device* A menjadi *master* untuk *device* B, tetapi menjadi *slave* untuk *device* C, seperti yang terlihat pada gambar di bawah ini (sumber: <http://wireless.klings.org>):



Gambar 2.8 Scatternet dengan 3 node

### II.2.6 Bluetooth Profiles

*Bluetooth profiles* dimaksudkan untuk memastikan *interoperability* diantara *bluetooth-enabled device* dan aplikasi-aplikasi dari pembuat dan vendor yang berbeda. *Profile* mendefinisikan peran dan kemampuan tipe spesifik dari aplikasi. Gambar berikut ini merepresentasikan *bluetooth profile* (sumber: <http://forum.nokia.com>):



Gambar 2.9 Bluetooth profiles

Profile dikeluarkan dengan *Bluetooth Specification v1.1* yang disebut *foundation profile*. Tabel berikut ini menjelaskan secara singkat mengenai profile tersebut:

Tabel 2.2 Bluetooth foundation profile

Profile	Deskripsi
<i>Generic Access Profile (GAP)</i>	Dasar dari semua <i>profile</i> dalam sistem <i>bluetooth</i> . GAP mendefinisikan dasar fungsionalitas <i>bluetooth</i> seperti prosedur koneksi, <i>device discovery</i> , dan <i>link management</i> .
<i>Serial Port Profile (SPP)</i>	Menyediakan <i>serial port (RS-232) emulation</i> berdasarkan RFCOMM bagian dari <i>bluetooth stack</i> .
<i>Dial Up Networking Profile (DUNP)</i>	Mendefinisikan fungsionalitas untuk penggunaan <i>bluetooth device</i> sebagai <i>Dial Up Networking Gateway</i> .
<i>FAX profile</i>	Mendefinisikan fungsionalitas untuk penggunaan



	<i>bluetooth device</i> sebagai FAX Gateway.
<i>Headset Profile</i>	Mendefinisikan fungsionalitas yang dibutuhkan untuk melakukan pengiriman suara seperti <i>wireless bluetooth headset</i> .
<i>LAN Access Point Profile</i>	Mendefinisikan fungsionalitas untuk penggunaan <i>bluetooth device</i> sebagai LAN access point.
<i>Generic Object Exchange Profile (GOEP)</i>	Menyediakan fungsionalitas untuk mendukung protokol pertukaran objek ( <i>Object Exchange</i> ) melalui jaringan <i>bluetooth</i> .
<i>Object Push Profile</i>	Menyediakan fungsionalitas untuk pertukaran <i>vCard</i> dan <i>vCalendar Object</i> , berdasarkan GOEP.
<i>File Transfer Profile</i>	Menyediakan fungsionalitas untuk pengendalian <i>folder</i> dan <i>copying/deleting/creating file</i> atau <i>folder</i> pada <i>bluetooth device</i> berdasarkan GOEP.
<i>Synchronization Profile</i>	Mendefinisikan fungsionalitas untuk <i>synchronization object stores</i> berisi <i>IrMC object</i> ( <i>vCard</i> , <i>vCalendar</i> , <i>vMessaging</i> , dan <i>vNotes object</i> ) antara <i>bluetooth device</i> , berdasarkan GOEP.
<i>Intercom Profile</i>	Memungkinkan <i>bluetooth device</i> untuk membangun jaringan komunikasi langsung yang mirip dengan <i>intercom communication</i> .
<i>The Cordless Telephony Profile</i>	Memungkinkan <i>bluetooth device</i> bertindak sebagai komunikasi telepon <i>cordless</i> tetap, seperti <i>ISDN gateway</i> .

Penggunaan *profile* menjamin *interoperability* antara *device* yang berbeda dari *Original Equipment Manufacturers (OEMs)* yang berbeda pula. Pengguna dapat membeli telepon selular dari satu *vendor* dan *headset*-nya dari *vendor* lain tetapi dapat bekerja dengan baik asalkan kedua *device* tersebut mengimplementasikan *headset profile*. Suatu *profile* baru akan

didefinisikan secara berkesinambungan oleh Bluetooth SIG Working Groups.

### II.2.7 Arsitektur *Bluetooth Protocol Stack*

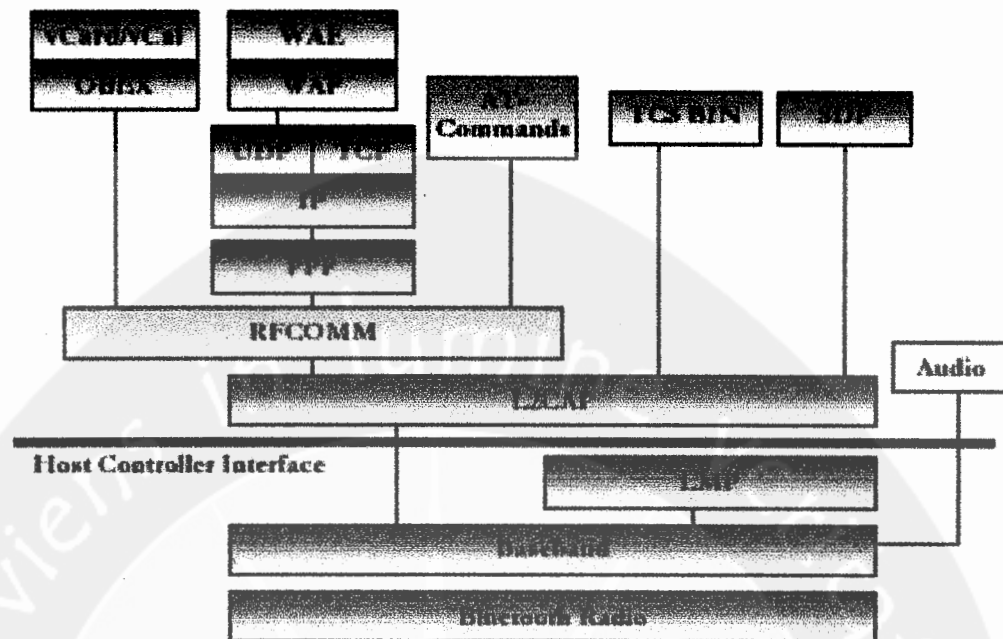
Protokol dibutuhkan untuk mengimplementasikan perbedaan *profile* dan penggunaan model. Setiap *profile* menggunakan sedikitnya beberapa bagian *protocol stack*. Supaya tercapai *interoperability* antara dua *bluetooth device*, keduanya harus mempunyai *profile* dan *protocol stack*.

*Bluetooth protocol stack* dibentuk dari protokol inti *bluetooth* dan protokol-protokol lain yang diadopsi ke dalam *bluetooth*. Protokol ini terdiri dari empat *layers (lapisan-lapisan)*, seperti yang ditunjukkan pada tabel berikut:

Tabel 2.3 Protokol dan *layer* di dalam *Bluetooth Protocol Stack*

<i>Layer Protokol</i>	<i>Protocol Stack</i>
<i>Bluetooth core protocol</i>	<i>Baseband, LMP, L2CAP, SDP</i>
<i>Cable replacement protocol</i>	RFCOMM
<i>Telephony control protocol</i>	TCS Binary, AT-commands
<i>Adopted protocol</i>	PPP, UDP/TCP/IP, OBEX, WAP, HID, vCard, vCal, IrMC, WAE, BNEP

Untuk lebih jelasnya, berikut akan ditampilkan gambar mengenai protokol-protokol di dalam *bluetooth* (sumber: <http://forum.nokia.com>):



Gambar 2.10 Bluetooth protocol stack

Deskripsi mengenai lapisan-lapisan tersebut dapat dilihat pada tabel berikut:

Tabel 2.4 Deskripsi dari lapisan protokol bluetooth

Lapisan	Deskripsi
Telephony control - AT commands	Melakukan pengontrolan <i>mobile phone</i> dan modem dalam <i>multiple usage model</i> . Selain itu AT commands juga digunakan sebagai layanan FAX.
Telephony Control System (TCS)	Protokol yang <i>bit-oriented</i> yang melakukan <i>call control signaling</i> dalam pembentukan hubungan data dan suara antara perangkat <i>bluetooth</i> . Selain itu juga melakukan <i>mobility management procedures</i> untuk menangani sekelompok TCS <i>bluetooth</i> .
Service Discovery Protocol (SDP)	Merupakan bagian yang penting di dalam <i>bluetooth</i> karena menyediakan dasar-dasar dari model penggunaan <i>bluetooth</i> . Dengan SDP, informasi perangkat, layanan, dan karakteristik layanan dapat diminta dan setelah itu hubungan antara dua atau lebih perangkat lunak

	<i>bluetooth</i> dapat dibentuk.
<i>Wireless Application Protocol (WAP)</i>	WAP <i>stack</i> dapat ditempatkan di atas RFCOMM atau L2CAP. Dapat mengurangi <i>overhead</i> dan menjadi solusi lebih baik untuk WAP menggunakan <i>bluetooth</i> .
<i>Object Exchange (OBEX)</i>	Digunakan untuk pertukaran objek secara sederhana dan langsung dengan menggunakan model <i>client-server</i> dan tidak tergantung pada mekanisme <i>transport</i> .
PPP	Digunakan bersama-sama dengan RFCOMM untuk melakukan koneksi <i>point-to-point</i> .
TCP/IP/UDP	Digunakan untuk melakukan komunikasi dengan perangkat <i>bluetooth</i> lain melalui internet.
RFCOMM	Berperan sebagai RS-232 <i>control</i> dan sinyal data melalui <i>bluetooth baseband</i> yang memberikan kemampuan <i>transport</i> bagi layanan di level atas yang menggunakan saluran serial sebagai mekanisme <i>transport</i> .
<i>Logical Link Control and Adaptation Protocol (L2CAP)</i>	Melakukan adaptasi terhadap protokol di lapisan atas melalui <i>baseband</i> dan menyediakan layanan data <i>connection-oriented</i> dan <i>connectionless</i> ke protokol di lapisan atas dengan kemampuan <i>multiplexing</i> , <i>segmentation and reassembly</i> , dan <i>group abstraction</i> . L2CAP besarnya sampai 64 KB dan hanya diperuntukkan bagi saluran ACL.
<i>Host Controller Interface (HCI)</i>	Menangani komunikasi antara <i>host</i> dan <i>bluetooth module</i>
<i>Link Manager Protocol (LMP)</i>	Membentuk hubungan antar perangkat <i>bluetooth</i> termasuk aspek keamanan seperti <i>authentication</i> dan enkripsi.
<i>Baseband dan Link Controller</i>	Memungkinkan <i>physical RF links</i> antara unit-unit <i>bluetooth</i> membentuk <i>piconet</i> . Sistem RF <i>bluetooth</i>

	adalah sistem FHSS dimana paket ditransmisikan dalam <i>time slot</i> dan frekuensi yang telah ditetapkan. Lapisan ini menggunakan prosedur <i>inquiry</i> dan untuk mensinkronkan <i>frequency hopping</i> transmisi dan <i>clock</i> dari perangkat <i>bluetooth</i> yang berbeda.
<i>Audio</i>	Dua <i>bluetooth device</i> yang mendukung <i>audio</i> dapat mengirim dan menerima <i>audio data</i> antara satu sama lain dengan membuka jaringan <i>audio</i> .
<i>Bluetooth Radio</i>	Melakukan modulasi dan demodulasi data untuk transmisi dan penerimaan data melalui udara

## II.2.8 Sistem Keamanan *Bluetooth*

### II.2.8.1 Kriteria Keamanan

Aspek keamanan merupakan hal penting yang perlu dipertimbangkan oleh semua teknologi, tak terkecuali teknologi *bluetooth*. Kriteria keamanan pada teknologi *bluetooth* adalah:

#### 4) *Availability*

Kemampuan suatu teknologi di dalam menjamin kelangsungan layanan di dalam jaringan terhadap serangan DoS (*Denial of Service*). Pertukaran informasi pada *bluetooth* dilakukan melalui medium udara sehingga memungkinkan serangan DoS.

#### 5) *Authenticaton*

Kemampuan suatu perangkat di dalam mengenali perangkat lain ketika saling berkomunikasi. Tanpa *autentication*, pihak yang tidak berhak dapat menyamar sebagai lawan komunikasi dan tanpa ijin memperoleh akses ke berbagai *resource* dan informasi-informasi penting serta mengacaukan sistem.

#### 6) *Confidentiality*

Kemampuan suatu perangkat di dalam menjamin bahwa suatu informasi tertentu tidak terbuka bagi pihak yang tidak berhak. Informasi-informasi

penting seperti informasi militer, mensyaratkan kerahasiaan. Oleh sebab itu, diperlukan teknik *authentication* yang baik, maka suatu komunikasi yang aman dapat dibentuk dengan kunci yang sudah disetujui.

7) *Integrity*

Kemampuan di dalam menjamin bahwa pesan yang dikirim tidak akan terkorupsi. Pesan terkorupsi dapat terjadi karena kesalahan-kesalahan seperti kerusakan di dalam propagasi radio atau serangan-serangan yang merusak di dalam jaringan.

8) *Authorization*

Proses di dalam menentukan apakah suatu perangkat dapat mempunyai akses ke suatu layanan tertentu. Proses ini yang menentukan suatu perangkat '*trusted*' atau '*untrusted*'. *Authorization* selalu melibatkan *authentication*.

9) *Non-repudiation*

Kemampuan di dalam menjamin bahwa pengirim pesan telah berhasil mengirimkan pesannya. Berguna untuk penghapusan dan penyisihan.

10) *Privacy*

Kemampuan di dalam melindungi dan mencegah informasi dari pengguna mengalir ke pengguna yang lain. Oleh karena *bluetooth* adalah sistem jaringan personal, maka ada kemungkinan pengguna akan mengasosiasikan dirinya dengan perangkat *bluetooth* tertentu seperti PDA. Jadi diperlukan sistem yang tidak menampilkan kegiatan dari pengguna tersebut.

### II.3 Teks

Teks merupakan salah satu objek atau bentuk data multimedia yang paling mudah disimpan dan dikendalikan. Teks dapat membentuk kata, surat atau narasi dalam multimedia yang menyajikan bahasa. Kebutuhan teks dalam multimedia bergantung pada kegunaan dari aplikasi multimedia tersebut. Misal aplikasi game *handphone* membutuhkan teks yang lebih sedikit sebagai petunjuk dari game tersebut sedangkan untuk ensiklopedia

dalam bentuk web membutuhkan teks yang lebih banyak agar memberikan informasi yang lengkap bagi pembacanya. Ada empat macam teks, yaitu: teks cetak (teks yang tercetak), teks hasil scan (teks cetak yang di *scan* oleh scanner yang hasilnya bernama *scanned text*), teks elektronik (teks yang dibaca komputer dan dikirim secara elektronis melalui jaringan), *hypertext* (teks yang mengacu pada proses *link*).

#### II.4 Teknologi Java

Java merupakan bahasa pemrograman berorientasi objek dan bebas *platform* yang dikembangkan oleh perusahaan bernama Sun Microsystem. Bidang usaha perusahaan ini adalah memproduksi perangkat elektronika rumah tangga seperti *tape*, televisi, dan perangkat lainnya. Java lahir karena ketidakpuasan seorang insinyur di Sun Microsystem bernama James Gosling. Ia tidak puas dengan kompiler C++ karena dinilai terlalu banyak menghasilkan *bug*, biaya yang besar, sangat bergantung terhadap *platform*. Kemudian ia membuat kompiler baru untuk mengatasi hal tersebut dan diberi nama Oak. Akan tetapi, pada perkembangannya bahasa Oak ini menjadi bahasa yang bisa digunakan untuk pemrograman secara umum yang dikenal dengan bahasa Java saat ini. Beberapa keunggulan teknologi Java antara lain:

- 1) Berbasis GUI
- 2) Berorientasi objek
- 3) Mendukung aplikasi web
- 4) *Multiplatform*
- 5) Tingkat keamanan yang lebih tinggi
- 6) Bisa digunakan pada aplikasi jaringan terdistribusi
- 7) Mendukung software *Mission-Critical*
- 8) *Multithread*

Produk pertama Java, yaitu versi 1.1 yang dikenal dengan nama Java 1 Compliant. Pada perkembangan selanjutnya, Sun Microsystem memperkenalkan Java versi 1.2 yang dikenal dengan Java 2 yang terdiri atas

JDK (*Java Development Kit*) dan JRE (*Java Runtime Environment*) versi

1.2. Java 2 ini dibagi menjadi 3 kategori, yaitu (<http://java.sun.com>):

1. *Java 2 Standard Edition (J2SE)*

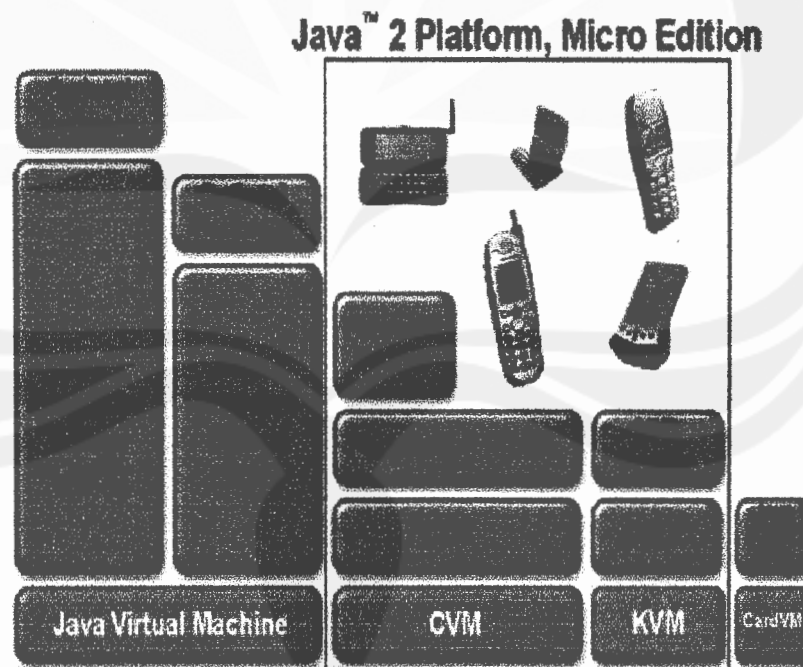
Kategori ini digunakan untuk menjalankan dan mengembangkan aplikasi-aplikasi Java pada level PC (*Personal Computer*).

2. *Java 2 Enterprise Edition (J2EE)*

Kategori ini digunakan untuk menjalankan dan mengembangkan aplikasi-aplikasi Java pada lingkungan *enterprise*, dengan menambah fungsionalitas Java semacam Java CORBA, *Servlet*, dan Java XML (*Extensible Markup Language*).

3. *Java 2 Micro Edition (J2ME)*

Kategori ini digunakan untuk menjalankan dan mengembangkan aplikasi-aplikasi Java pada *handheld device* seperti *handphone*, Palm, PDA, dan Pocket PC.



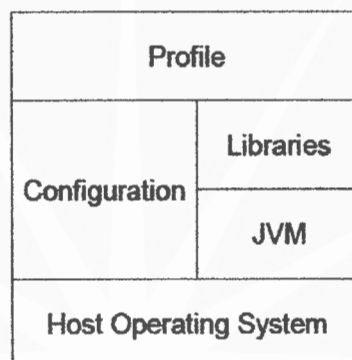
Gambar 2.11 Java 2 Technology



Saat ini ada satu kategori lain yaitu *Java Card*, yang khusus dikembangkan untuk membuat aplikasi-aplikasi pada *smart card*, misalnya aplikasi kartu telepon *CHIP*, kartu VISA, kartu SIM pada *handphone* dan aplikasi *mobile banking* yang saat ini sudah mulai umum digunakan.

## II.5 J2ME (*Java 2 Micro Edition*)

J2ME merupakan salah satu kategori Java yang digunakan pada pengembangan sistem mikro dan *embedded*, meliputi *handphone*, pager, PDA, dan lain-lain. Berikut adalah gambar arsitektur J2ME (sumber: <http://wireless.klings.org>):



Gambar 2.12 Arsitektur J2ME

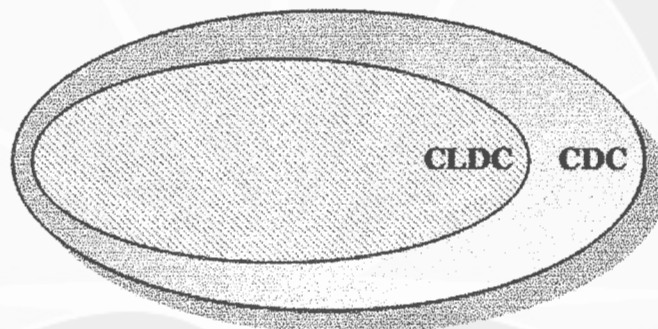
Komponen-komponen J2ME terdiri dari *Java Virtual Machine* (JVM) yang digunakan untuk menjalankan aplikasi Java pada *emulator* atau *handheld device*, Java API, dan *tools* lain seperti *J2ME Wireless Toolkit*.

### II.5.1 *Configuration* dan *Profile*

*Configuration* mendefinisikan minimum Java *Libraries* dan kapabilitas yang dimiliki para developer J2ME. Artinya, antara *mobile device* yang *Java enabled* maka akan ditemui *configuration* yang sama. Bila dianalogikan dengan sekelompok mobil maka sebuah mobil dengan mobil lainnya memiliki kesamaan, misalnya mempunyai roda, kaca spion, speedometer, dan lain-lain. Roda dan kaca spion itulah yang disebut sebagai

*configuration*. Jadi *configuration* hanyalah mengatur hal-hal yang berkaitan dengan kesamaan bukan mengatur hal-hal yang membedakan sehingga *configuration* memastikan *portabilitas* antar *device*.

Configuration ini ditentukan perkembangannya oleh JCP (*Java Community Process*), inilah badan non-profit yang berkuat dengan perkembangan teknologi Java. Saat ini ada dua standar *configuration* J2ME, yaitu: *Connected Device Configuration* (CDC) dan *Connected Limited Device Configuration* (CLDC). CDC lebih diarahkan untuk *powerful device* like *Internet TVs* dan *car navigation system*, sedangkan CLDC lebih diarahkan untuk *powerless device* seperti *mobile phone* dan PDA. Hubungan antara CLDC dengan CDC dapat digambarkan sebagai berikut (sumber: <http://developers.sun.com/mobility/getstart/articles/survey/>):



Gambar 2.13 Hubungan CLDC dengan CDC

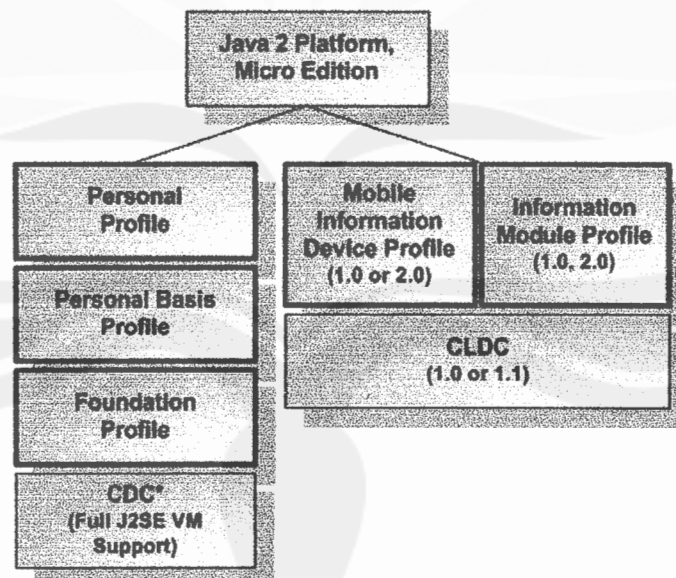
Sedangkan perbedaan mengenai CDC dan CLDC dapat dilihat pada table berikut:

Tabel 2.5 Perbedaan CDC dan CLDC

CLDC	CDC
Mengimplementasikan <i>subset</i> dari J2SE	Mengimplementasikan seluruh fitur pada J2SE
JVM yang digunakan dikenal dengan nama KVM (K <i>Virtual Machine</i> ) yaitu mesin virtual yang digunakan pada perangkat yang memiliki memori beberapa <i>kilobyte</i>	JVM yang digunakan dikenal dengan nama CVM (C <i>Virtual Machine</i> ) yaitu mesin virtual yang digunakan pada perangkat yang memiliki memori cukup besar

Digunakan pada perangkat <i>handheld</i> dengan ukuran memori terbatas (160 – 512 KB)	Digunakan pada perangkat <i>handheld</i> dengan ukuran memori minimal 2 MB
Processor: 16 atau 32 <i>bit</i>	Processor: 32 <i>bit</i>

*Profile* merupakan kebalikan dari *configuration*, yaitu mengatur hal-hal yang spesifik untuk sebuah *device* atau tipe *market*. Kembali pada analogi mobil di atas, sebuah mobil BMW tentu memiliki ciri spesifik yang tidak dimiliki oleh mobil Mercedes, dan sebaliknya. *Profile* mendefinisikan sekumpulan *Application Programming Interface* (API) yang terletak di atas *configuration* dan menawarkan akses kemampuan khusus ke *device*. *Mobile Information Device Profile* (MIDP) adalah *profile* yang digunakan dengan CLDC dan menyediakan sekumpulan API untuk digunakan oleh *mobile device*. API ini terdiri atas kelas untuk *user interface*, *persistent storage*, dan *networking*. Selain MIDP, terdapat *profile* seperti *Foundation Profile* (FP), *Personal Profile*, dan *Personal Digital Assistance* (PDA) (sumber: <http://developers.sun.com/mobility/getstart/articles/survey/>).



\*CDC 1.0 is based on J2SE 1.3

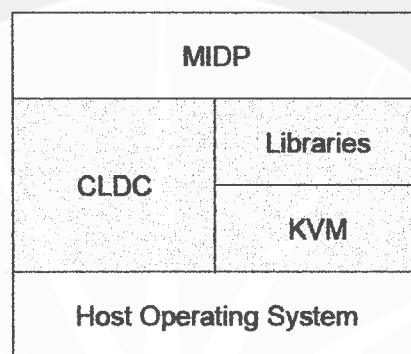
\*CDC 1.1 is based on J2SE 1.4

Gambar 2.14 J2ME Profiles

### II.5.2 *Connected Limited Device Configuration (CLDC)*

CLDC adalah suatu konfigurasi untuk *mobile device* yang ditujukan untuk *platform* Java yang membutuhkan memori kecil antara 160-512 KB. Spesifikasi CLDC adalah sebagai berikut (sumber: <http://wireless.klings.org>):

- 1) Mengimplementasikan subset dari J2SE
- 2) JVM yang digunakan dikenal dengan nama *K-Virtual Machine (KVM)*
- 3) Digunakan pada perangkat *handheld* dengan ukuran terbatas (160-512 KB)
- 4) Prosesor: 16 bit atau 32 bit



Gambar 2.15 Posisi CLDC dalam arsitektur J2ME

Gambar di atas menunjukkan CLDC adalah teknologi inti yang dirancang sebagai basis untuk satu atau lebih *profile*. Pada bagian ini secara detail CLDC diperlukan untuk pengembangan aplikasi *wireless* dengan MIDP implementasinya CLDC digunakan untuk program Java pada perangkat keras dengan ukuran memori terbatas, pada 160 sampai 512 KB. Akibatnya, fitur-fitur yang kurang penting untuk diimplementasikan dalam *handheld device* yang bersangkutan dari Java 2 harus dibuang.

Dua versi CLDC, yang telah didefinisikan, yaitu versi 1.0 dan versi 1.1. CLDC 1.1 memiliki beberapa fitur baru daripada CLDC 1.0. Memori yang dibutuhkan sebesar 160 KB untuk versi 1.0 dan 192 KB untuk versi 1.1. CLDC terdiri dari paket-paket, yaitu:

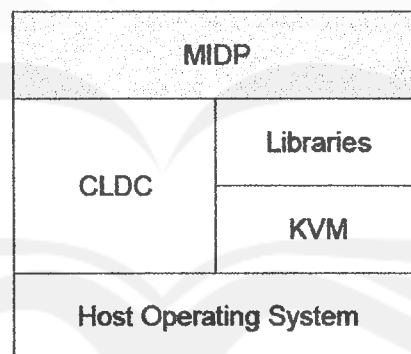
Tabel 2.6 Paket-paket CLDC

Paket	Kegunaan
java.io	Menyediakan <i>class</i> untuk input dan output melalui <i>data stream</i> .
java.lang	Menyediakan <i>class</i> dasar bagi bahasa pemrograman Java.
java.lang.ref	Menyediakan dukungan referensi.
java.util	Mengandung <i>collection class</i> dan fasilitas tanggal dan waktu.
javax.microedition.io	<i>Class</i> untuk GCF

Semua paket dalam CLDC berkoresponding dengan paket dari J2SE, kecuali `javax.microedition.io` yang diperkenalkan dalam CLDC.

### II.5.3 Mobile Information Device Profile (MIDP)

MIDP merupakan sekumpulan API yang terletak di atas CLDC seperti yang ditunjukkan pada gambar di bawah ini (sumber: <http://wireless.klings.org>):



Gambar 2.16 Posisi MIDP dalam arsitektur J2ME

MIDP menyediakan fitur seperti *user interface*, *networking support*, dan *persistent storage*. Dua versi MIDP yang telah dirilis saat ini, yaitu MIDP 1.0 dan MIDP 2.0

### II.5.3.1 MIDP 1.0

MIDP 1.0 merupakan hasil pengembangan dari JCP *expert group*, JSR 37. Spesifikasi MIDP 1.0 mendefinisikan arsitektur dan API yang berhubungan dalam pengembangan aplikasi *mobile information device*. Untuk dapat diklasifikasikan sebagai MID, sebuah device harus memiliki karakteristik sebagai berikut:

Tabel 2.7 MIDP requirements

Display:	Pixels: 96x54
Display depth:	1-bit
Pixel shape (aspect ratio):	approximately 1:1
Input:	One- or two- handed keyboard atau touch screen
Memory:	128 KB memory tetap untuk komponen MIDP 8 KB memory tetap untuk aplikasi 32 KB memory sementara untuk Java runtime environment
Networking	Two-way, wireless, possibly intermittent dengan limited bandwidth

MIDP menambahkan beberapa paket di atas CLDC, yaitu:

Tabel 2.8 Paket MIDP 1.0

Paket	Kegunaan
<code>javax.microedition.lcdui</code>	Menyediakan class untuk user interface.
<code>javax.microedition.midlet</code>	Mendefinisikan aplikasi MIDP dan interaksi antara aplikasi dengan lingkungan.
<code>javax.microedition.rms</code>	Menyediakan <i>persistent storage</i> (Record Management System).

Oleh karena MIDP berada di atas CLDC, maka MIDP juga mengandung semua paket CLDC. Selain itu, juga ditambahkan beberapa *interface* dan *class* baru dalam CLDC, diantaranya `HTTPConnection interface` untuk menangani koneksi HTTP dengan mewarisi fungsionalitas dalam GCF, dan `TimerTask class` dalam paket `java.util`.

### II.5.3.2 MIDP 2.0

MIDP 2.0 merupakan penyempurnaan MIDP 1.0 yang dikembangkan oleh JCP *expert group*, JSR 118. Spesifikasinya berdasarkan MIDP 1.0 hanya ada penambahan untuk *sound*-nya saja, jadi MIDlet yang dituliskan untuk MIDP 1.0 dapat dijalankan di lingkungan MIDP 2.0.

Paket dalam MIDP 2.0 sama seperti MIDP 1.0 dengan beberapa paket tambahan yaitu:

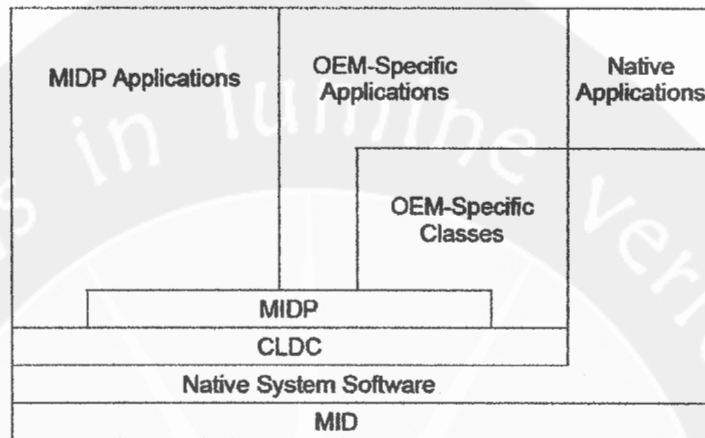
Tabel 2.9 Paket MIDP 2.0

Paket	Kegunaan
<code>javax.microedition.lcdui</code>	Menyediakan class untuk user interface.
<code>javax.microedition.midlet</code>	Mendefinisikan aplikasi MIDP dan interaksi antara aplikasi dengan lingkungan.
<code>javax.microedition.rms</code>	Menyediakan <i>persistent storage</i> ( <i>Record Management System</i> ).
<code>javax.microedition.lcdui.game</code>	Menyediakan fungsionalitas yang sangat berguna dalam pengembangan game.
<code>javax.microedition.media</code>	Menyediakan <i>audio building block</i> (ABB).
<code>javax.microedition.pki</code>	Menyediakan fungsionalitas untuk menangani <i>certificates</i> .

Fitur-fitur baru dalam MIDP GUI yang lebih baik dari sebelumnya, dukungan media seperti suara dan *video streaming*, dukungan untuk *game*, mendukung konektivitas seperti HTTPS, *datagram*, *sockets*, *server sockets* dan *serial port communication*, *push architecture* yang memungkinkan untuk mengaktifkan MIDlet ketika *device* menerima informasi dari *server*, *over-the-air* (OTA) *provisioning*, dan *end-to-end security*.

## II.5.4 MIDlets

Aplikasi MIDP disebut juga MIDlets. MIDlet biasanya didistribusikan dalam MIDlet suites yang tersedia di internet melalui WAP. Arsitektur MIDlet dapat anda lihat pada gambar di bawah ini (sumber: <http://wireless.klings.org>):

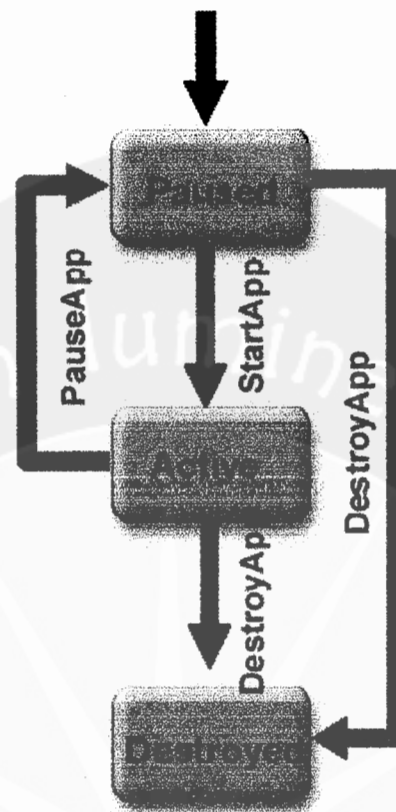


Gambar 2.17 Arsitektur MIDlet

### II.5.4.1 MIDlet Lifecycle (Daur Hidup MIDlet)

*Lifecycle* dari sebuah MIDlet ditangani oleh *Application Management Software* (AMS). AMS adalah sebuah lingkungan tempat siklus dari sebuah MIDlet, mampu untuk diciptakan, dijalankan, dihentikan, maupun dihilangkan. AMS sering disebut juga *Java Application Manager* (JAM). MIDlet memiliki beberapa *state* (keadaan), yaitu *pause*, *active*, dan *destroy*. Ketika masing-masing *state* dipanggil, beberapa *method* yang bersesuaian dipanggil. *Method-method* tersebut merupakan bawaan dari J2ME. Berikut ini adalah gambar mengenai daur hidup MIDlet (sumber: IF2281\_J2ME-MIDP.pdf):





Gambar 2.18 Daur hidup MIDlet

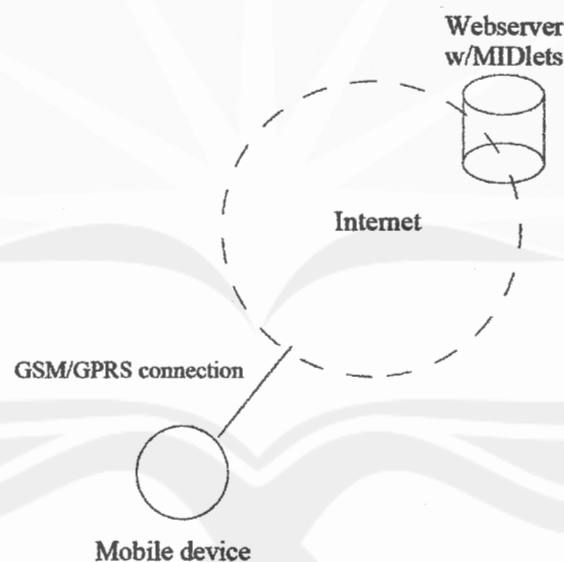
#### II.5.4.2 MIDlet Suites

MIDlet biasanya tersedia melalui MIDlet suites. MIDlet suite terdiri dari 2 *file*, yaitu sebuah *file* .jar dan .jad. *File Java ARchive* (JAR) mengandung semua *file-file* hasil kompilasi *file* java dalam bentuk *file* .class yang terkompres dan melakukan proses verifikasi format. *File Java Application Descriptor* (JAD) adalah sebuah *file plain text* yang mengandung semua informasi tentang sebuah MIDlet suite. Semua MIDlet harus dituliskan dalam *file* ini, ukuran *file* JAR harus dimasukkan dan benar, dan URL ke *file* JAR juga harus dituliskan dengan benar. Sebagai tambahan, versi MIDlet suite juga dimasukkan ke *file* JAD. Hal ini merupakan informasi yang diperlukan untuk MID. MID selalu *men-download file* JAD

terlebih dahulu dan memeriksa isinya. Jika MIDlet suite telah di-*install*, ia akan tahu jika ada versi terbaru yang tersedia. Ukuran *file* JAR adalah informasi yang penting, MID dapat mendeteksi apakah ada cukup *memory* untuk menginstal MIDlet suite. Jika semua syarat terpenuhi, maka MID akan menuju ke URL yang diberikan dan men-*download file* JAR.

#### II.5.4.3 MIDlet Deployment

MIDlet suite dapat disebar (deployed) ke *webserver* dan membuatnya tersedia untuk di *download* melalui HTTP atau *browser* yang mendukung WAP. Penyebaran melalui *webserver* adalah metode yang paling umum dalam menyebarkan MIDlet suite secara luas, tetapi MIDlet suite juga dapat dikirimkan ke MID melalui *bluetooth connection* atau *cable connection* (sumber: <http://klings>).

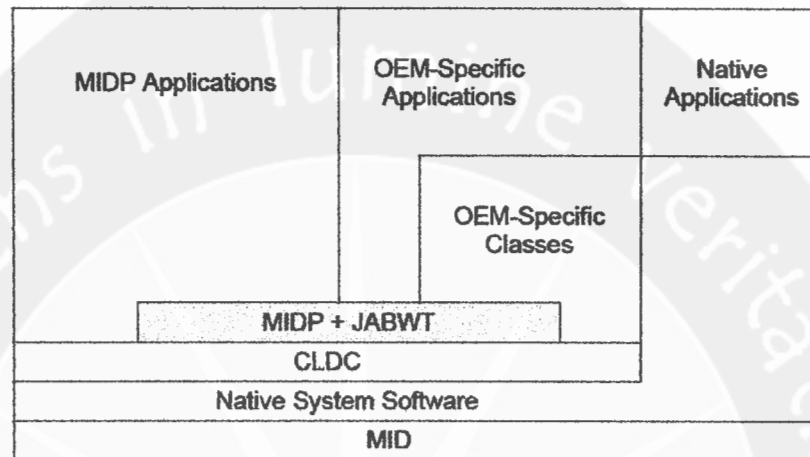


Gambar 2.19 MIDlet deployment melalui webserver

Gambar di atas menunjukkan bahwa *mobile device* melakukan koneksi ke internet menggunakan GSM atau GPRS *connection*. MIDlet dapat di *download* dari *webserver* di internet, secara sederhana dengan memasukkan URL *file* jad MIDlet yang diinginkan dalam HTTP atau WAP *browser*.

### II.5.5 Java Application Bluetooth Wireless Toolkit (JABWT)

JABWT didefinisikan oleh JCP *expert group*, JSR 82. JABWT merupakan paket pilihan yang digunakan untuk memprogram aplikasi menggunakan *Bluetooth* (sumber: <http://klings>).



Gambar 2.20 Posisi JABWT dalam arsitektur J2ME

Pada gambar di atas, JABWT terletak di atas CLDC dan diharapkan mewarisi semua kemampuan dari *profile* seperti MIDP. JABWT menggunakan GCF, didefinisikan dalam CLDC, untuk komunikasi *bluetooth*. JABWT terdiri dari 2 paket, yaitu:

Tabel 2.10 Paket dalam Java *Bluetooth*

Paket	Kegunaan
javax.bluetooth	Inti dari Bluetooth API.
javax.obex	Object Exchange (OBEX) API.