

# J2ME - MIDP

Yohanes Nugroho

# Overview

- Device Non PC/Micro Device
- J2ME
- Konfigurasi dan Profile J2ME
- MIDP 1.0
- Mengembangkan aplikasi MIDP 1.0
  - langkah development
- Memprogram device MIDP 1.0
  - Tools yang diperlukan dan penggunaannya
  - Hello World
  - Penjelasan API MIDlet

# Device Non PC/Micro Device

- Selain PC, masih ada device lain yang memiliki komputer (mesin komputasi/prosessor) di dalamnya
  - Mainframe
    - Komputer yang “besar” (daya komputasi, memori, penyimpanan) dan umumnya tidak bergerak
  - Micro device
    - Alat yang memiliki prosessor dengan memori dan penyimpanan terbatas dan umumnya bersifat mobile
- Kita hanya akan membahas micro device

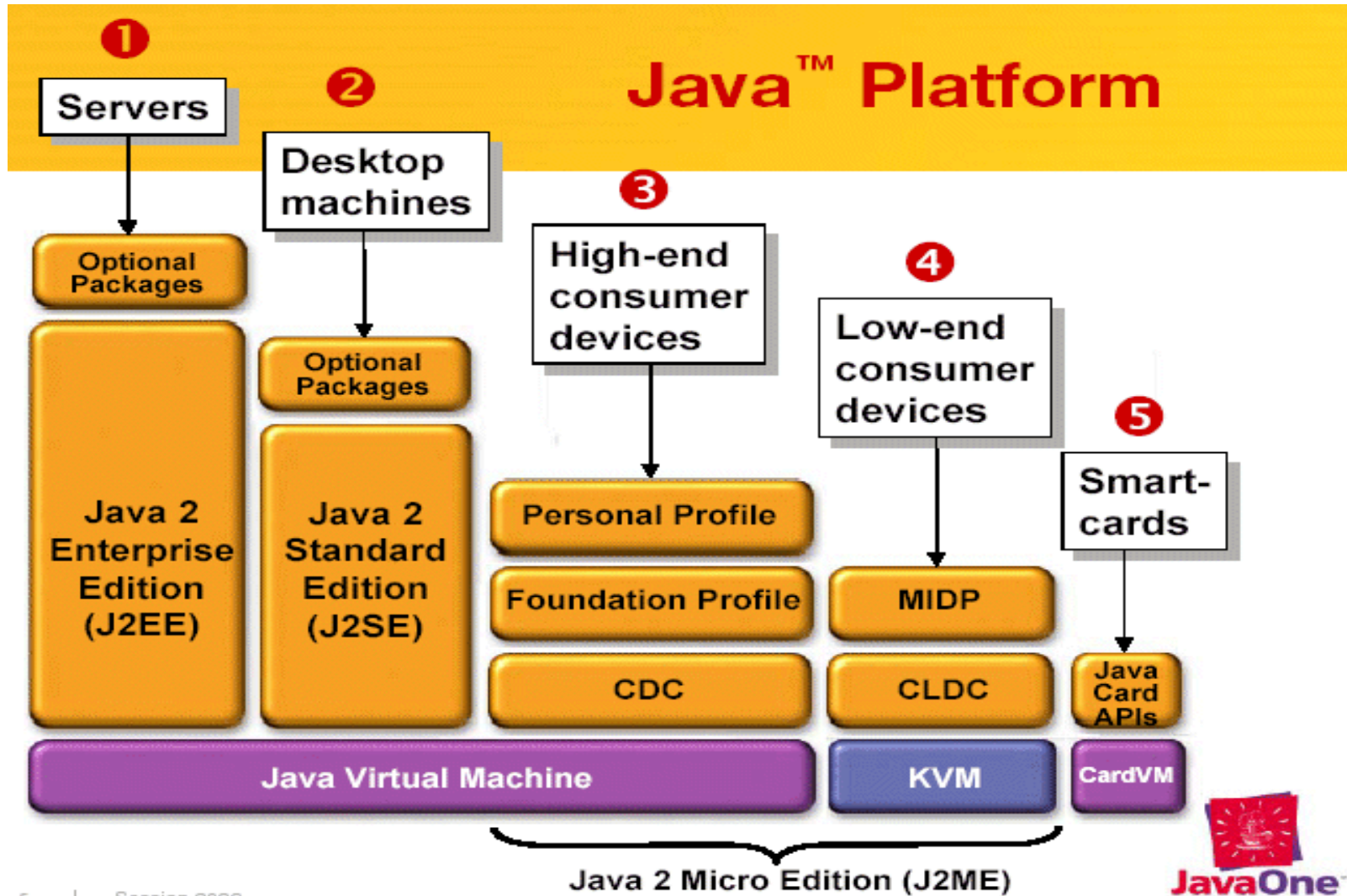
# Pemrograman Micro Device

- Umumnya dilakukan dalam bahasa C atau assembly
  - tapi kita akan melakukannya dalam Java
- Perbedaan dengan PC:
  - keterbatasan prosessor
  - keterbatasan input method (lebih sulit menginput teks)
  - keterbatasan memori
  - keterbatasan power (bettere)
- Untuk lebih jelas, lihat “Pemrograman Device Non PC” di kuliah Pemrograman Sistem

# J2ME

- Java 2 Micro Edition adalah edisi Java untuk device dengan kemampuan komputasi terbatas
  - Nama J2ME digunakan sebagai “payung” untuk aneka teknologi Java untuk device micro (di antaranya adalah Personal Java, JavaCard, MIDP, dll)
- Menggunakan bahasa Java yang sama, minus fitur tertentu (baik fitur bahasa maupun API)
  - fitur bahasa dan API yang ada dan tidak ada tergantung dari konfigurasi (kemampuan) dan profil (jenis) device

# Gambaran arsitektur Java



# Cakupan J2ME

- Device yang bisa diprogram dengan J2ME mulai dari smart card sampai PDA
- Masing-masing jenis device memiliki kemampuan komputasi yang berbeda
  - Smartcard memiliki memori  $<1\text{Mb}$ , dengan kecepatan prosessor yang rendah
  - PDA saat ini memiliki memori  $>8\text{Mb}$  dengan kecepatan prosessor yang tinggi
  - Device dengan kemampuan komputasi yang sama dibagi ke dalam **Configuration**

# Configuration

- Sebuah konfigurasi (*configuration*) mendefinisikan satu set library minimum dan kemampuan Virtual Machine minimum yang dimiliki sebuah device
- Device yang memiliki kemampuan pemrosesan yang sama dan batasan memori yang sama memiliki konfigurasi yang sama
- Contoh configuration
  - CDC (Connected Device Configuration)
  - CLDC (Connected Limited Device Configuration)

# Contoh *Configuration*: CLDC

- Punya koneksi terbatas (limited)
  - artinya user bisa online jika diperlukan
- Tidak mendukung tipe data floating point dan double
- Device mungkin hanya memiliki memori non volatile 128 Kb, memori volatile 8 kb untuk data persisten, dan 32 Kb untuk runtime Java
- Contoh device CLDC: Ponsel J2ME, PDA dengan koneksi wireless, pager dua arah

# Contoh *Configuration*: CDC

- Punya koneksi jaringan yang selalu online (connected)
- Mendukung seluruh fitur bahasa Java (JDK1.3)
- Device setidaknya memiliki 512kb memori non volatile dan 256 Kb memori volatile
- Contoh Device: PDA Phone

# Profile

- Profile adalah API yang diimplementasikan di atas (*on top of*) sebuah konfigurasi yang ditujukan untuk device dengan jenis/kegunaan yang serupa
  - Sebuah profile berdiri di atas sebuah configuration
- Contoh device serupa: ponsel
  - Device sejenis akan memiliki API yang serupa, misalnya untuk menampilkan teks, menerima input dari ITU Keypad (keypad telepon)
  - Secara teori bisa dibuat profile untuk jenis device apapun
- Profile yang paling terkenal adalah MID (Mobile Information Device)

# MIDP 1.0

- Mobile Information Device Profile versi 1.0 berada di atas CLDC
  - Hampir semua Ponsel J2ME mendukung profile ini
- MIDP memberikan batasan minimum bagi device dan satu Set API untuk peralatan mobile
  - API yang sangat generik yang pasti bisa didukung semua jenis device MIDP
  - Aplikasi MIDP disebut dengan MIDlet

# Batasan Minimum Device MIDP 1.0

- Display monokrom atau berwarna, minimal 96 x 54 piksel
- Input touch screen, keypad ITU, atau keyboard
- Koneksi wireless bandwidth terbatas yang tidak selalu online
- Memiliki batasan memori minimum CLDC

# API Non Standar

- Selain API Standar, mungkin ada API dari Vendor yang tersedia di device
  - Contoh: API untuk backlight atau vibrator (nada getar), API ini tidak masuk standar karena tidak semua device punya backlight atau vibrator
  - Device yang memakai API non standar tidak dijamin bisa berjalan di device lain
- API MIDP 1.0 adalah API Standar
  - Program yang hanya memakai API ini pasti berjalan di aneka device MIDP 1.0

## Keterbatasan API Standar MIDP 1.0

- Beberapa hal ini perlu diperhatikan:
  - Tidak ada API untuk mengakses address book
  - Tidak ada API untuk mengakses SMS User termasuk juga untuk mengirim SMS
  - Tidak ada API untuk mencegat phone call, atau melakukan phone call
  - Tidak ada API untuk mendapatkan lokasi cell ID saat ini
- Hal ini perlu ditegaskan di awal karena sering mengecewakan programmer yang ingin membuat program J2ME MIDP 1.0. Di MIDP 1.0

# Security MIDP 1.0

- Selain alasan dukungan hardware device (lihat 2 slide sebelumnya mengenai API non standar) ada masalah security
- Perhatikan contoh berikut:
  - device bisa mengakses internet (hanya program MIDP, Server, programmer, dan Tuhan yang tahu apa isi data yang dikirim)
  - Jika device bisa mengakses phone book atau SMS maka isi phone book atau SMS bisa dikirim ke orang lain di Internet
  - Hal ini berlaku juga untuk batasan yang lain (misal: SMS bisa dikirim via Internet, atau program bisa menghabiskan pulsa dengan mengirim SMS)

# Sekilas MIDP 2.0

- Secara singkat MIDP 2.0 adalah MIDP 1.0 + API untuk multimedia
  - Akses ke kamera dan perekaman Video
  - Mendukung aneka format Audio dan Video
  - Memiliki batasan minimum hardware yang lebih tinggi
- Device MIDP 2.0 saat ini masih relatif mahal (lebih mahal dari Device MIDP 1.0)
  - MIDP 2.0 Kompatibel/bisa menjalankan aplikasi MIDP 1.0

# Mengapa mengajarkan MIDP 1.0

- Materi yang akan dibahas hanyalah mengenai MIDP 1.0 dengan alasan:
  - Supaya lebih memahami batasan device
  - Supaya bisa dicoba langsung oleh sebagian besar peserta kuliah di device yang sesungguhnya
    - Device MIDP 1.0 sudah sangat murah (< 1 juta), rata-rata HP sudah mendukung MIDP, jadi pasti ada 1 teman yang punya HP ber-MIDP
  - MIDP 2.0 belum 100% menjadi standar (hanya ada di HP mahal, dengan banyak batasan)
    - Contoh batasan: ada HP MIDP 2.0 tanpa kamera

# Mengembangkan aplikasi MIDP 1.0

- Langkah pembuatan program:
  - [1] Menentukan program apa yang ingin dibuat
  - [2] Memahami dan “mengakali” (work around) batasan device
    - Jika ternyata tidak mungkin diprogram, pikirkan pendekatan lain
  - [3] Memprogram di emulator
  - [4] Mendeploy (menginstall) di device yang sebenarnya

# Menentukan program yang akan dibuat

- Tanpa memikirkan batasan device, pikirkan:
  - apakah program cukup nyaman dijalankan di device? (memakai spreadsheet di HP tentunya tidak nyaman)
  - apakah program berguna?
  - terlalu memikirkan batasan device akan membatasi kreativitas
- Mengapa tidak memikirkan batasan device lebih dulu?
  - beberapa batasan akan dapat “diakali” jika memang program cukup berguna

# Memahami dan mengakali batasan device [1]

- Jika konsep program sudah ada, pikirkan apakah device mampu menangani program tersebut?
  - Adakah cara untuk mengakalinya (work around it?)
- Contoh kasus: ebook reader
  - Masalah: ebook butuh data besar (contoh: Harry Potter 1 dalam format HTML yang telah di kompres butuh 180 Kb), memori HP tidak cukup
  - Solusi: buku didownload secara parsial dari web

# Memahami dan mengakali batasan device [2]

- Contoh kasus: messenger
  - Masalah: protokol Yahoo dan MSN terlalu rumit (butuh banyak memori)
  - Solusi: Buat protokol yang lebih sederhana dan program proxy yang menerjemahkan protokol Yahoo dan MSN yang rumit ke yang lebih sederhana
- Contoh kasus: web browser
  - Masalah: parsing web terlalu rumit
  - Ada server yang melakukan parsing, data yang sudah dalam bentuk sederhana diberikan ke program J2ME

# Memprogram di emulator

- Emulator adalah program yang meniru kerja alat lain
  - Dalam kasus ini meniru HP yang mendukung MIDP
- Memprogram di emulator akan mempermudah proses development:
  - tidak perlu memindah program ke device (melelahkan dan butuh waktu)
  - mudah didebug
- Kelemahan emulator:
  - tidak semua fitur device diemulasikan dengan tepat
  - kadang program berjalan dengan cepat di emulator tapi sangat lambat di device yang sesungguhnya

# Mendeploy ke device

- Test di device yang sesungguhnya harus dilakukan karena batasan emulator
- Instalasi/deployment dilakukan dengan:
  - Kabel data
  - Infrared
  - Bluetooth
  - WIFI
  - Koneksi Internet (CSD/GPRS untuk GSM, PDN untuk CDMA)

# Memprogram MIDP 1.0

- Untuk memprogram device (apapun) umumnya kita perlu tahu:
  - apa tools yang dibutuhkan
  - contoh program kecil berikut penjelasannya
  - cara menjalankan program di emulator
  - cara menginstall dan menjalankan program di device yang sesungguhnya
  - penjelasan API lain (untuk membuat program yang lebih besar/berguna)

# Tools yang dibutuhkan

- Tools wajib:
  - Compiler Java
  - Salah satu (tidak eksklusif, bisa dipakai bersama-sama) dari tools berikut:
    - WTK (Wireless Toolkit) dari Sun
    - SDK device tertentu
- Tools opsional:
  - IDE (Integrated Development Environment) untuk memudahkan penulisan program
  - Obfuscator untuk memperkecil program dan mencegah dekompilasi

# Contoh Program kecil: Hello World

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

public class Hello extends MIDlet {
    Form f = new Form("Hello World");
    public void startApp() {
        Display.getDisplay(this).setCurrent(f);
    }
    public void pauseApp() { }
    public void destroyApp
        (boolean unconditional) {}
}
```

# Penjelasan Hello World

- **Import package standar untuk MIDlet :**

```
import javax.microedition.midlet.*;  
import javax.microedition.lcdui.*;
```

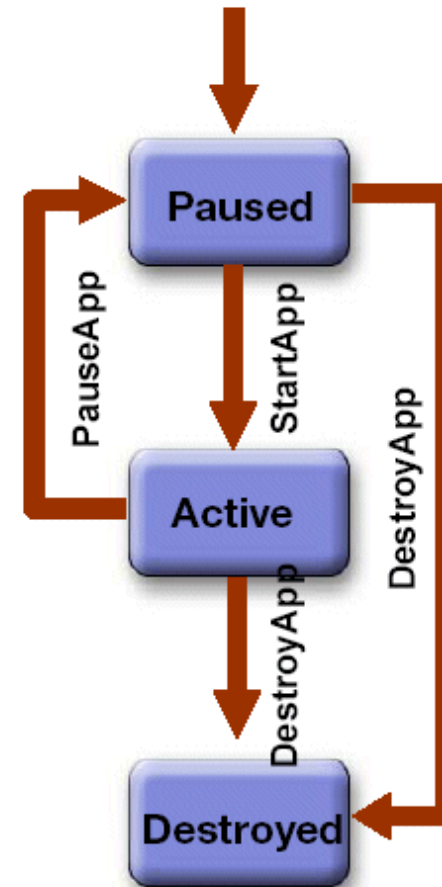
- **Sebuah Aplikasi Midlet harus diturunkan dari kelas `javax.microedition.midlet.MIDlet` dan harus mengimplementasikan 3 method abstrak `startApp()` , `pauseApp()` dan `destroyApp()` ;**

# Method Abstrak Kelas MIDLet

- `startApp()` dipanggil ketika aplikasi dimulai
- `pauseApp()` dipanggil ketika aplikasi dihentikan sementara
  - misalnya ketika ada telepon atau SMS datang
  - Catatan: tidak semua HP mengimplementasikan ini dengan benar (kadang `pauseApp()` tidak dipanggil)
- `destroyApp(boolean unconditional)` dipanggil ketika aplikasi akan dihentikan
  - jika `unconditional` bernilai `true` midlet harus berhenti, jika `false`, midlet boleh meminta agar tidak dihentikan
  - Hal yang dilakukan di `destroyApp()` misalnya adalah menyimpan data

# State MIDlet

- Gambaran state MIDlet sesuai method yang telah dijelaskan



# GUI Midlet Hello World

- Dalam satu saat di layar boleh ada satu objek turunan `Displayable`
  - Dalam contoh, digunakan objek `Form` (dengan konstruktor judul `Form`) yang merupakan salah satu objek turunan `Displayable`
- Objek yang aktif di layar diset dengan:  
`Display.getDisplay(objekMidlet).setCurrent(objekTurunanDisplayable)`
- Dalam contoh objek `MIDlet` adalah this

# Membuat dan Menjalankan MIDlet

- Jalankan WTK (Wireless Toolkit)
- Pilih new project (pilih jenisnya MIDP 1.0)
  - Beri nama Project: Hello
  - Beri nama MIDLet: Hello
- Buat file Hello.java (dengan isi seperti contoh) di direktori C:\WTK2.1\apps\Hello\src
- Klik Compile
- Jika sukses Klik Run

# Mempersiapkan file biner untuk HP

- Di HP (umumnya) tidak ada kompilator, jadi file perlu dibuatkan menjadi package
- Cara membuat package
  - pilih project --> package --> create package
  - hasilnya ada di C:\WTK2.1\apps\Hello\bin
- Akan terbentuk 2 file
  - File Jar: berisi kumpulan kelas dan resource
  - File Jad: deskripsi file jar
- Kirimkan file .jar dan .jad ke HP

# Menginstall ke HP [1]

- Alternatif offline:
  - Kabel data:
    - perlu software khusus sesuai HP
  - Infrared/Bluetooth
    - jika HP bisa dikirim file dari Windows (dengan OBEX/Object Exchange Protocol, yaitu dengan klik kanan dan memilih 'Send To Nearby Computer' atau 'Send To Bluetooth' ) maka kedua file bisa langsung dikirim (tapi kadang hanya .jar yang diperlukan)
    - Jika HP tidak mendukung OBEX, maka diperlukan software khusus

# Menginstall ke HP [2]

- Alternatif Online
  - Menggunakan WIFI/GSM/GPRS/PDN (Tergantung teknologi HP)
  - Letakkan file jad dan file jar di sebuah direktori yang sama di tempat yang bisa diakses via web,
    - misal di /home/ifxxxxx/public\_html di students.if.itb.ac.id
  - Arahkan wap browser ke alamat file JAD
    - misal  
<http://students.if.itb.ac.id/~ifxxxxx/Hello.jad>

# Memahami API Midlet

## Penjelasan Sekilas API MIDlet

# Aneka API MIDlet

- Hanya ada 4 kelompok API penting MIDlet
  - API MIDlet
  - GUI
    - Highlevel/lcdUI (untuk aplikasi biasa)
    - Lowlevel/canvas (untuk game)
  - Connection
  - RMS (Record Management System)
- Tidak akan di bahas lengkap
  - Lihat contoh di Demo Java
  - Lihat di buku yang membahas MIDP 1.0

# API GUI

- GUI dapat dibuat dengan komponen standar yang sudah disediakan (dalam package `javax.swing`)
  - List, Textbox, dll
  - Java akan secara otomatis menangani perbedaan antar device (ukuran layar, warna, dll)
- GUI dapat dibuat dengan canvas
  - Programmer menggambar sendiri semua yang diperlukan dan menangani perbedaan jenis layar (tidak akan dijelaskan di sini)
  - Umumnya dipakai oleh game

# LCDUI

- Dalam LCD UI digunakan konsep Layar (objek turunan Displayable)
  - Dalam satu saat hanya ada boleh 1 objek turunan Displayable yang muncul
  - Tidak ada multiwindow (layar terlalu kecil)
- Contoh turunan Displayable:
  - Form: Di dalam form boleh ada komponen-komponen seperti text box
  - List: hanya ada satu list yang muncul (misalnya untuk menu)
  - Alert: pesan peringatan ke user dalam satu layar penuh

# Contoh pemakaian LCDUI

...

```
Form f = new Form("Kuesioner");  
TextField tnama = new TextField("Nama",  
    "", 15, TextField.ANY);  
f.append(tnama);  
Displayable.getDisplay(this).setCurrent  
    (f);
```

....

# Command sebagai pengganti menu

- Untuk membuat menu digunakan objek Command
- Java yang akan mengatur bagaimana penampilan command
- Contoh:

```
Command c = new Command("Keluar",  
    Command.EXIT, 0);
```
- Command memiliki teks, Jenis aksi, dan prioritas (untuk mengurutkan menu)

# Menangkap Event

- Sebuah event di Form atau List bisa ditangkap untuk diproses
  - Misal jika isi list dipilih
  - caranya dengan memanggil `setCommandListener()` dengan parameter berupa Listener
- Listener adalah Objek yang mengimplementasikan interface `CommandListener`
  - Lihat lagi konsep Swing

# Interface CommandListener

- Hanya berisi satu method untuk melakukan aksi apa ketika command terjadi:

```
public interface CommandListener {  
public void commandAction(Command c,  
                             Displayable d);  
}
```

# Contoh: Hello World dengan menu [1]

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

public class Hello extends MIDlet implements
    CommandListener {
    Form f = new Form("Hello World");
    Command exit = new Command("Keluar", Command.EXIT,
        0);
public void startApp() {
    f.addCommand(exit);
    f.setCommandListener(this);
    Display.getDisplay(this).setCurrent(f);
}
}
```

# Contoh: Hello World dengan menu [2]

```
public void commandAction(Command c, Displayable d) {  
    if (c == exit) {  
        System.out.println("Destroyed");  
        notifyDestroyed();  
    }  
}
```

```
public void pauseApp() { }
```

```
public void destroyApp(boolean unconditional) {}  
}
```

# Connection

- API untuk koneksi ke Internet
  - Kelas ada pada package `javax.microedition.io`
  - Ada beberapa jenis koneksi yang dapat dilakukan (HTTP, IP Stream, IP Datagram) tapi sebuah HP hanya wajib mendukung HTTP
- Ketika dijalankan perlu diset oleh user agar MIDlet boleh mengakses internet

# Pemakaian Connection

- Pemakaian HTTP Connection untuk membuka koneksi HTTP (tanpa penanganan error) adalah sbb:

```
HttpConnection c = (HttpConnection)  
Connector.open(url);
```

```
InputStream is = c.openInputStream();
```

# RMS (Record Management System)

- Sistem persistensi MIDP adalah Record yang dikelola RMS (tidak menggunakan filesystem/file)
  - Ada dalam package javax.microedition.rms
- Beberapa Record (analogi: isi file) dikumpulkan dalam satu Record Store (analogi: file)
  - Satu MIDlet boleh memiliki beberapa RecordStore

# Identifikasi Record

- Record diidentifikasi dengan ID
  - ID adalah sebuah integer
  - ID otomatis diassign oleh RMS
  - ID Selalu menaik (record berikut akan memiliki  $ID > ID$  record saat ini), tapi mungkin tidak berurut (jika ada record yang dihapus)
  - Jika ada record yang dihapus ID tidak dipakai lagi

# Operasi RecordStore dan Record

- Operasi record store:
  - membuat/membuka/menutup/menghapus record store
- Operasi Record:
  - menulis/membaca/menghapus record dalam record store

# Membuat RecordStore

- Membuat atau membuka record store dilakukan dengan:

```
RecordStore rs =  
    RecordStore.openRecordStore("nama",  
    "vendor", "namasuite");
```

- Jika record store belum ada maka akan diciptakan
- Untuk menutup RecordStore:

```
rs.closeRecordStore();
```

# Menyimpan Record

- Record disimpan dengan `addRecord` (`addRecord` mengembalikan ID record baru)

```
int addRecord(byte [] data, int  
    offset, int numBytes)
```

- Data yang disimpan hanya bisa array of bytes
  - Gunakan kelas `java.io.DataOutputStream` untuk mengubah tipe data dasar apapun ke array of Bytes

# Konversi tipe data dasar ke array of bytes

```
ByteArrayOutputStream baos = new
    ByteArrayOutputStream();
DataOutputStream dos = new
    DataOutputStream(baos);
dos.writeInt(123);
/*UTF adalah string unicode*/
dos.writeUTF("Hello");
dos.flush();
byte result[] = baos.toByteArray();
```

# Membaca dan Menghapus Record

- Record dibaca berdasarkan ID-nya dengan hasil berupa array of byte

```
public byte[] getRecord(int  
    recordId)
```

- Array of byte bisa dikonversi ke tipe lain dengan `java.io.DataInputStream`
- Record dihapus berdasarkan RecordID

```
public void deleteRecord(int  
    recordId)
```

```
/*hasil adalah array of byte*/  
ByteArrayInputStream bais = new  
    ByteArrayInputStream(hasil);  
DataInputStream dis = new  
    DataInputStream(bais);  
int i = dis.readInt();  
/*UTF adalah string unicode*/  
String s = dis.readUTF();
```

# Penutup

- Pemrograman MIDlet hanyalah salah satu cara memprogram HP
- Masih ada alternatif lain (tergantung jenis HP)
  - C++ : Symbian, Windows Mobile, Palm
  - C : Mophun, Windows Mobile, Palm
  - Python: Symbian
- Sebagai mahasiswa informatika, cobalah memprogram benda apapun yang bisa diprogram :)