

# Bab 1

## Pengembangan Aplikasi Mobile

### 1.1 Tujuan

Dalam bab ini, kita akan mendiskusikan karakteristik dari perangkat mobile, dan bagaimana hal tersebut mempengaruhi pengembangan program untuk perangkat ini. Kita akan diperkenalkan kepada Java 2 Mobile Edition (J2ME), termasuk pentingnya configuration dan profilnya.

Pada akhir bab ini, pelajar diharapkan dapat menguasai :

- Mengidentifikasi karakteristik dari perangkat mobile
- Menjelaskan arsitektur J2ME
- Mengetahui peran atau aturan configuration dan profile
- Mengidentifikasi API yang disediakan oleh MIDP
- Menjelaskan daur hidup MIDlet

### 1.2 Perangkat Mobile

Perangkat mobile memiliki banyak jenis dalam hal ukuran, desain dan layout, tetapi mereka memiliki kesamaan karakteristik yang sangat berbeda dari sistem desktop.

- Ukuran yang kecil  
Perangkat mobile memiliki ukuran yang kecil. Konsumen menginginkan perangkat yang terkecil untuk kenyamanan dan mobilitas mereka.
- Memory yang terbatas  
Perangkat mobile juga memiliki memory yang kecil, yaitu primary (RAM) dan secondary (disk). Pembatasan ini adalah salah satu faktor yang mempengaruhi penulisan program untuk berbagai jenis dari perangkat ini. Dengan pembatasan jumlah dari memory, pertimbangan-pertimbangan khusus harus diambil untuk memelihara pemakaian dari sumber daya yang mahal ini.
- Daya proses yang terbatas  
Sistem mobile tidaklah setangguh rekan mereka yaitu desktop. Ukuran, teknologi dan biaya adalah beberapa faktor yang mempengaruhi status dari sumber daya ini. Seperti harddisk dan RAM, Anda dapat menemukan mereka dalam ukuran yang pas dengan sebuah kemasan kecil.
- Mengonsumsi daya yang rendah

Perangkat mobile menghabiskan sedikit daya dibandingkan dengan mesin desktop. Perangkat ini harus menghemat daya karena mereka berjalan pada keadaan dimana daya yang disediakan dibatasi oleh baterai-baterai.

- Kuat dan dapat diandalkan

Karena perangkat mobile selalu dibawa kemana saja, mereka harus cukup kuat untuk menghadapi benturan-benturan, gerakan, dan sesekali tetesan-tetesan air.

- Konektivitas yang terbatas

Perangkat mobile memiliki bandwidth rendah, beberapa dari mereka bahkan tidak tersambung. Kebanyakan dari mereka menggunakan koneksi wireless.

- Masa hidup yang pendek

Perangkat-perangkat konsumen ini menyala dalam hitungan detik kebanyakan dari mereka selalu menyala. Coba ambil kasus sebuah handphone, mereka booting dalam hitungan detik dan kebanyakan orang tidak mematikan handphone mereka bahkan ketika malam hari. PDA akan menyala jika anda menekan tombol power mereka.

## 1.3 Gambaran J2ME

### 1.3.1 Platform JAVA

Java dibuat pada tahun 1991 oleh James Gosling. Pada awalnya diberi nama Oak, dimana untuk menghormati pohon yang ada di luar jendela Gosling. Kemudian namanya diubah ke Java karena telah ada sebuah bahasa yang diberi nama Oak.

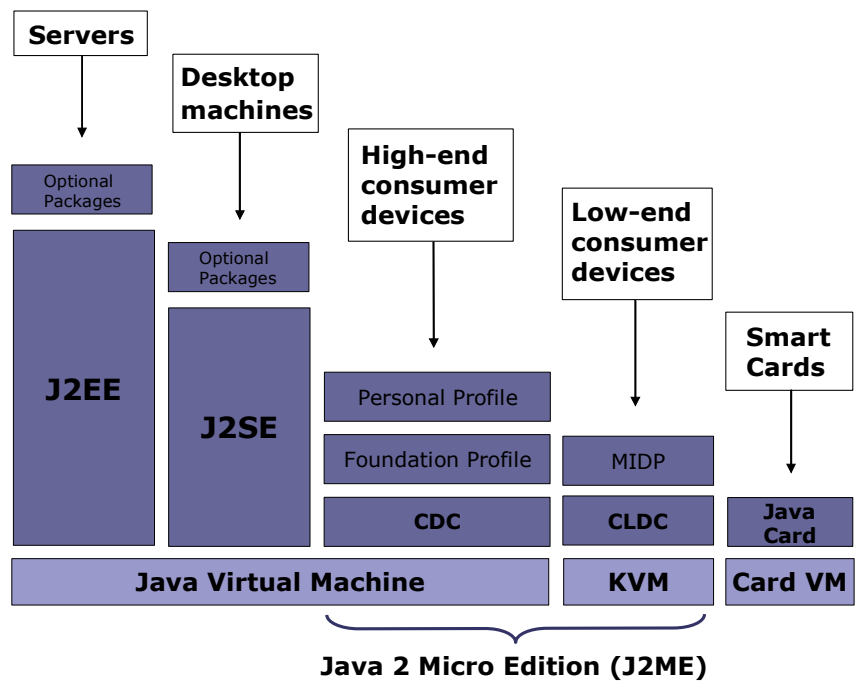
Motivasi sesungguhnya dari Java adalah kebutuhan akan sebuah bahasa yang bisa digunakan pada berbagai platform yang bisa dimasukkan ke dalam berbagai produk elektronik seperti pemanggang roti dan lemari es. Salah satu dari proyek pertama yang dikembangkan menggunakan JAVA sebuah remote kontrol yang diberi nama Star 7.

Pada saat yang sama, World Wide Web dan Internet berkembang sangat cepat. Gosling menyadari bahwa Java dapat digunakan untuk pemrograman Internet.

Dengan keluarnya versi 1.2, platform Java telah dipilah-pilah menjadi beberapa edisi :

The Standard Edition(J2SE), Enterprise Edition(J2EE), Mobile Edition(J2ME), dan JavaCard API.

J2SE – Java 2 Platform, Standard Edition	Aplikasi Desktop
J2EE – Java 2 Platform, Enterprise Edition	Aplikasi enterprise dengan fokus pada pengembangan sisi webserver, termasuk servlet,JSP,EJB, dan XML
J2ME – Java 2 Platform, Micro Edition	Perangkat Mobile
JavaCard	Smart Cards



**Gambar 1:** Platform Java

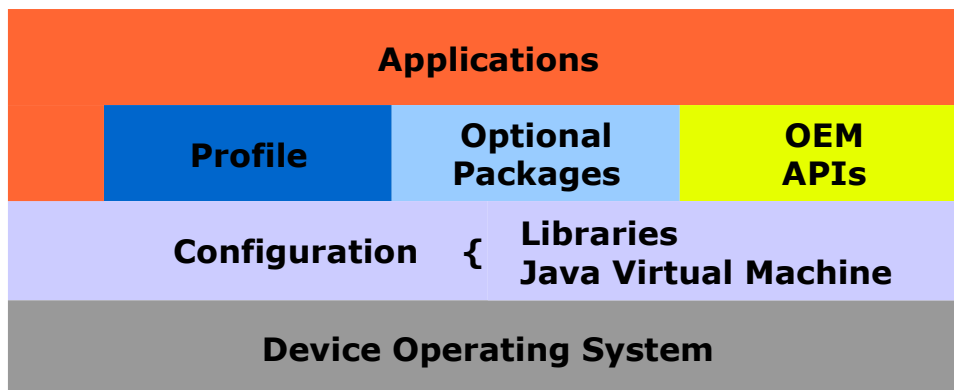
**1.3.2 Gambaran J2ME**

J2ME adalah satu set spesifikasi dan teknologi yang fokus kepada perangkat konsumen. Perangkat ini memiliki jumlah memori yang terbatas, menghabiskan sedikit daya dari baterai, layar yang kecil dan bandwidth jaringan yang rendah.

Dengan perkembangbiakan perangkat mobile konsumen dari telepon, PDA, kotak permainan ke peralatan-peralatan rumah, Java menyediakan suatu lingkungan yang portable untuk mengembangkan dan menjalankan aplikasi pada perangkat ini.

Program J2ME, seperti semua program JAVA adalah diterjemahkan oleh VM. Program-program tersebut dikompilasi ke dalam bytecode dan diterjemahkan dengan Java Virtual Machine(JVM).Ini berarti bahwa program-program tersebut tidak berhubungan langsung dengan perangkat. J2ME menyediakan suatu interface yang sesuai dengan perangkat. Aplikasi-aplikasi tersebut tidak harus dikompilasi ulang supaya mampu dijalankan pada mesin yang berbeda.

Inti dari J2ME terletak pada konfigurasi dan profile-profile. Suatu konfigurasi menggambarkan lingkungan runtime dasar dari suatu sistem J2ME. Ia menggambarkan core library, virtual machine, fitur keamanan dan jaringan.



**Gambar 2:** *Arsitektur J2ME*

Sebuah profile memberikan library tambahan untuk suatu kelas tertentu pada sebuah perangkat. profile-profile menyediakan user interface(UI) API,persistence, messaging library, dan sebagainya.

Satu set library tambahan atau package tambahan menyediakan kemampuan program tambahan. Pemasukan package ini ke dalam perangkat J2ME dapat berubah-ubah karena tergantung pada kemampuan sebuah perangkat. Sebagai contoh, beberapa perangkat MIDP tidak memiliki Bluetooth built-in, sehingga Bluetooth API tidak disediakan dalam perangkat ini.

### **1.3.3 Configuration**

Suatu configuration menggambarkan fitur minimal dari lingkungan lengkap Java runtime. Untuk menjamin kemampuan portabilitas dan interoperabilitas optimal diantara berbagai macam perangkat yang dibatasi sumber dayanya(memory, prosesor, koneksi yang dibatasi), configuration tidak menggambarkan fitur tambahan. Suatu configuration J2ME menggambarkan suatu komplemen yang minimum dari teknologi JAVA. Adalah merupakan tugas profile-profile untuk menggambarkan tambahan library untuk suatu kategori perangkat tertentu.

configuration menggambarkan:

- Subset bahasa pemrograman JAVA
- Kemampuan Java Virtual Machine(JVM)
- Core platform libraries
- Fitur sekuriti dan jaringan

### **1.3.4 Profile**

Suatu profile menggambarkan set-set tambahan dari API dan fitur untuk pasar tertentu, kategori perangkat atau industri. Sementara configuration menggambarkan library dasar, profile-profile menggambarkan library yang penting untuk membuat aplikasi-aplikasi efektif. Library ini memasukkan user interface, jaringan dan penyimpanan API.

## **1.4 CLDC**

The Connected Limited Device Configuration (CLDC) menggambarkan dan menunjuk pada area berikut ini:

- Fitur Bahasa Java dan Virtual Machine(VM)
- Library dasar(java.lang.\*,java.util.\*)
- Input/Output(java.io.\*)
- Keamanan
- Jaringan
- Internationalization

### **1.4.1 Fitur yang hilang**

Fitur tertentu dari J2SE yang dipindahkan dari CLDC adalah :

- Finalization of class instances
- Asynchronous exceptions
- Beberapa error classes
- User-defined class loaders
- Reflection
- Java Native Interface (JNI)
- Thread groups dan daemon threads

Reflection, Java Native Interface (JNI) dan user-defined class loaders potensial menjadi lubang keamanan. JNI juga membutuhkan memory yang intensif sehingga dimungkinkan untuk tidak mendapat dukungan dari memory rendah sebuah perangkat mobile.

### **1.4.2 Karakteristik perangkat CLDC**

Perangkat yang diincar oleh CLDC mempunyai karakteristik sebagai berikut:

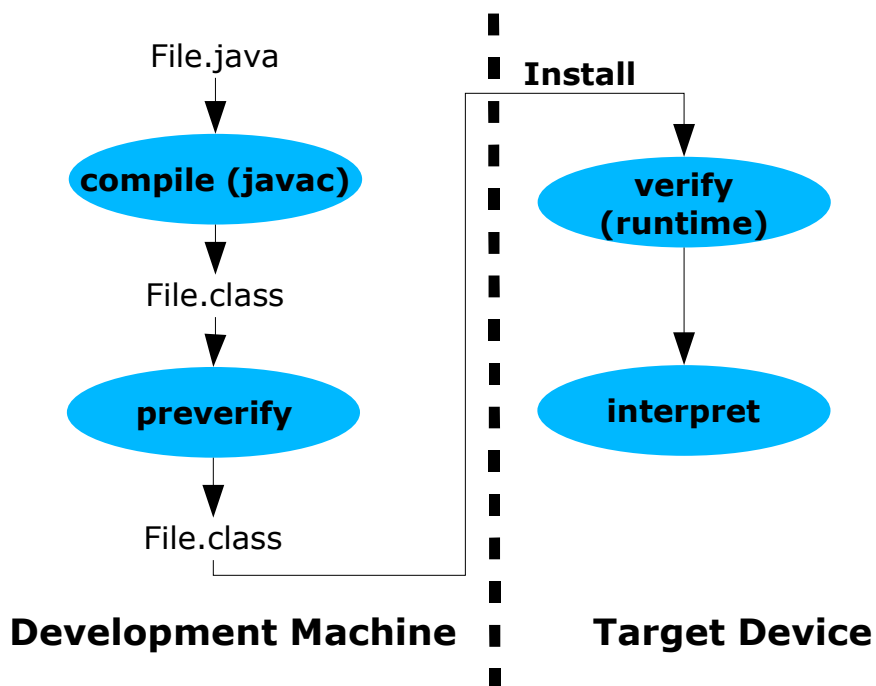
- Memory minimal 192kb untuk platform Java.
- Prosesor dengan 16 atau 32 bit.

- Mengonsumsi sedikit daya.
- Terbatas, koneksi jaringan yang sementara dengan pembatasan bandwidth(biasanya wireless).

CLDC tidak menggambarkan instalasi dan daur hidup sebuah aplikasi, antarmuka(UI) dan penanganan peristiwa(event handling). Adalah merupakan tugas profile yang berada di bawah CLDC untuk menggambarkan area ini. Secara khusus, spesifikasi MIDP menggambarkan daur hidup aplikasi MIDP (MIDlet), library UI dan event handling(javax.microedition.lcdui.\*).

### 1.4.3 Verifikasi Class

Spesifikasi CLDC memerlukan semua class untuk melewati proses verifikasi dua tingkat. Verifikasi pertama dilaksanakan diluar perangkat sebelum instalasi pada perangkat. Verifikasi kedua terjadi pada perangkat selama runtime dan dilaksanakan oleh KVM.

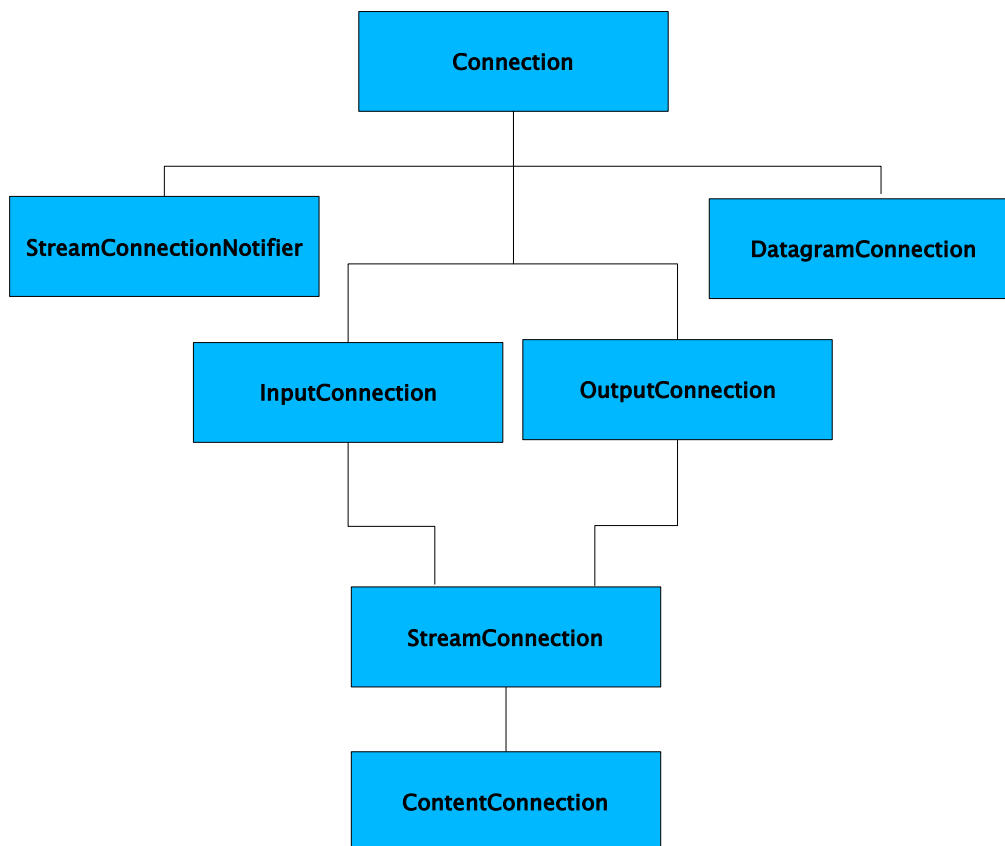


**Gambar 3:** Proses Verifikasi dua tingkat

### 1.4.4 Generic Connection Framework

The Generic Connection Framework menyediakan API dasar untuk koneksi dalam CLDC. Framework ini menyediakan suatu pondasi umum untuk koneksi seperti HTTP, Socket, dan Datagrams. GCF menyediakan suatu set API yang umum dan biasa yang memisahkan semua jenis koneksi. Perlu dicatat bahwa tidak semua jenis koneksi dibutuhkan untuk diterapkan oleh perangkat MIDP.

Hirarki interface yang dapat diperluas dari GFC membuat proses penyamarataan menjadi mungkin. Jenis koneksi baru mungkin bisa ditambahkan ke dalam framework ini dengan memperluas hirarki ini.



**Gambar 4:** Hirarki koneksi GCF

## 1.5 CDC

Connected Device Configuration (CDC) adalah super set dari CLDC. CDC menyediakan lingkungan Java runtime yang lebih luas dibandingkan CLDC dan lebih dekat kepada lingkungan J2SE.

CDC Java Virtual Machine (CVM) mendukung penuh Java Virtual Machine (JVM). CDC berisi semua API dari CLDC. CDC menyediakan suatu subset yang lebih besar dari semua class J2SE.

Seperti CLDC, CDC tidak menggambarkan setiap class UI. Library UI digambarkan oleh profile-profile di bawah configuration ini.

Semua class yang terdapat dalam CDC datang dari package ini:

- java.io
- java.lang
- java.lang.ref
- java.lang.math
- java.net
- java.security
- java.security.cert
- java.text
- java.util
- java.util.jar
- java.util.zip

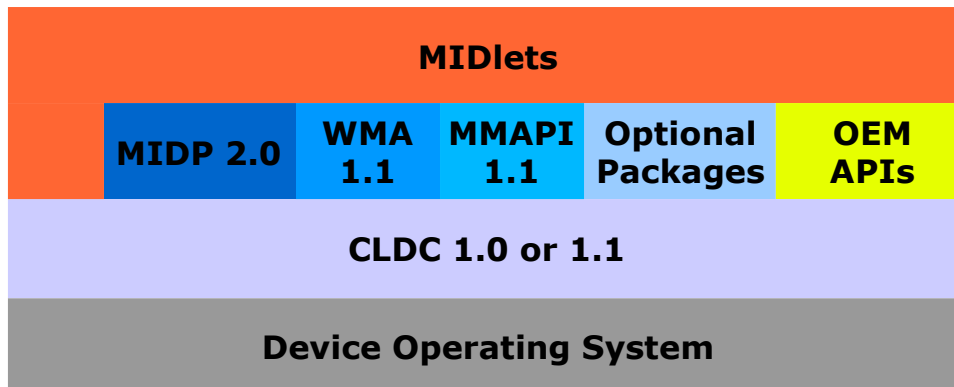
CDC juga memasukkan di dalamnya GCF. CDC memerlukan jenis koneksi tambahan seperti file dan dukungan datagram.

## 1.6 JTWI

The Java Technology for the Wireless Industry (JTWI) menetapkan satu set jasa dan spesifikasi standar. Berdasar spesifikasi JTWI, kata kuncinya adalah "untuk memperkecil fragmentasi API di dalam pasar telepon mobile, dan untuk mengirim spesifikasi yang dapat diprediksi, spesifikasi yang jelas untuk perangkat pabrik, operator, dan pengembang aplikasi".

Dengan penyesuaian kepada JTWI, banyak aplikasi akan berjalan di suatu set yang lebih luas pada perangkat. Perangkat pabrik juga akan beruntung karena sebuah aplikasi yang besar akan tersedia untuk perangkat mereka.





**Gambar 5:** *Komponen JTWI*

## 1.7 MIDP

The Mobile Information Device Profile (MIDP) berada di atas dari CLDC. Anda tidak bisa menulis aplikasi mobile hanya dengan menggunakan CLDC API. Anda harus tetap memanfaatkan MIDP yang mendefinisikan UI.

Spesifikasi MIDP, kebanyakan seperti CLDC dan API lainnya sudah digambarkan melalui Java Community Process (JCP). JCP melibatkan sebuah kelompok ahli berasal dari lebih dari 50 perusahaan, yang terdiri atas pabrik perangkat mobile, pengembang software. MIDP terus berkembang, dengan versi-versi masa depan yang telah lulus dari proses ketat JCP.

Spesifikasi MIDP menggambarkan suatu perangkat MID yang memiliki karakteristik-karakteristik ini sebagai batas minimum:

- Tampilan:
  - Ukuran Layar: 96x54
  - kedalaman tampilan: 1-bit
  - Ketajaman pixel: sekitar 1:1
- Masukan:
  - Satu atau lebih mekanisme user-input: satu keyboard, dua keyboard, atau touch screen
- Memory:
  - 256 kilobytes of non-volatile memory untuk implementasi MIDP.
  - 8 kilobytes of non-volatile memory for application-created persistent data
  - 128 kilobytes of volatile memory for the Java runtime (e.g., the Java heap)
- Jaringan:
  - dua jalur, wireless, bandwidth terbatas
- Sound:
  - Kemampuan untuk memainkan nada-nada

MIDP menggambarkan model aplikasi, UI API, penyimpanan dan jaringan yang kuat, permainan dan media API, kebijakan keamanan, penyebaran aplikasi dan ketetapan over-the-air.

## **1.8 MIDlet**

Suatu aplikasi MIDP disebut MIDlet. Perangkat application management software (AMS) berinteraksi langsung dengan MIDlet dengan method MIDlet create, start, pause, dan destroy.

MIDlet adalah bagian dari package javax.microedition.midlet. Sebuah MIDlet harus di-extend dengan class MIDlet. Dan dapat meminta parameter dari AMS seperti dirumuskan dalam application descriptor (JAD).

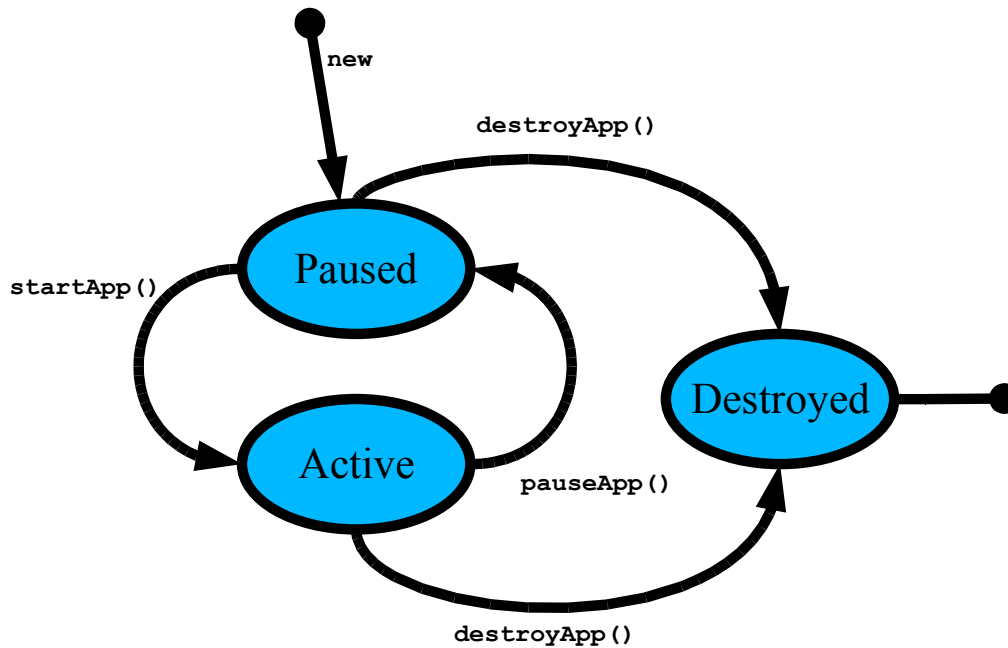
Suatu MIDlet tidak harus memiliki (dan memang harus tidak mempunyai) sebuah method public static void main(String[] argv). Method tersebut tidak akan dikenal lagi oleh AMS sebagai titik awal sebuah program.

### **1.8.1 Siklus MIDlet**

kehidupan MIDlet dimulai ketika di-instantiate oleh AMS. MIDlet pada awalnya masuk status "Pause" setelah perintah baru dibuat. AMS memanggil constructor public tanpa argumen dari MIDlet. Jika sebuah exception terjadi dalam constructor, MIDlet memasuki status "Destroyed" dan membuangnya segera.

MIDlet masuk ke dalam status "Active" atas pemanggilan method startUp() oleh AMS.

MIDlet masuk ke dalam status "Destroyed" ketika AMS memanggil method destroyApp(). Status ini juga kembali diakses ketika method notifyDestroyed() kembali dengan sukses kepada aplikasi. Dengan catatan bahwa MIDlet hanya bisa memasuki status "Destroyed" sekali dalam masa hidupnya.



**Gambar 6:** Daur hidup MIDlet

### 1.8.2 MIDlet suites

Aplikasi-aplikasi MIDlet dibungkus dan dikirim kedalam perangkat sebagai MIDlet suites. Sebuah MIDlet suite terdiri dari Java Archive (JAR) dan sebuah tambahan Java Application Descriptor (JAD).

File JAD adalah suatu file teks yang berisi satu set atribut-atribut, beberapa dibutuhkan.

## 1.9 Latihan

### 1.9.1 Apa keuntungan menggunakan java sebagai pengembangan dan platform runtime untuk perangkat mobile?

- Aplikasi yang sangat portable
- Kaya, tergambar dengan baik menghubungkan ke perangkat
- Penggunaan memori yang rendah(KVM)
- Lingkungan runtime yang aman
- aplikasi yang dinamis

### ***1.9.2 Apa yang memotivasi anda untuk menulis program untuk perangkat mobile?***

- Tantangan tentang penulisan mengoptimalkan aplikasi
- Pengalaman pelajaran baru
- Faktor kesenangan