

Memulai Cordova

Juli 2015

Oleh : Feri Djuandi

Tingkat:



Pemula



Menengah



Mahir

Apache Cordova atau cukup disebut Cordova, adalah satu set *Application Programming Interface* (API) perangkat yang memungkinkan pengembang aplikasi mobile untuk mengakses fungsi perangkat asli seperti kamera atau *accelerometer* dengan menggunakan bahasa JavaScript. Dikombinasikan dengan framework antar muka seperti jQuery Mobile atau Dojo Mobile atau Sencha Touch, Cordova memungkinkan aplikasi mobile dikembangkan dengan hanya **HTML, CSS, dan JavaScript**.

Saat menggunakan API Cordova pengembang aplikasi dapat membangun sebuah aplikasi tanpa kode asli (Java, Objective-C, dan lain-lain), sebaliknya teknologi yang digunakan adalah teknologi web. Hasilnya adalah sebuah aplikasi web yang dijalankan di dalam perangkat itu sendiri secara lokal dan bukan pada server web yang terpisah. Oleh karena API JavaScript bekerja secara konsisten di beberapa platform perangkat dan dibangun pada standar web, maka aplikasi menjadi portabel untuk dijalankan pada platform-platform yang berbeda dengan modifikasi yang minimal atau tidak ada sama sekali.

Aplikasi yang dibangun menggunakan Cordova tetap perlu dikemas sebagai paket aplikasi menggunakan *Software Development Kit* (SDK) dari platform masing-masing. Maksudnya jika aplikasi tersebut akan dijalankan pada Android maka pengemasannya tetap harus menggunakan SDK Android, demikian pula untuk platform lainnya. Cordova tersedia untuk platform berikut: iOS, Android, Blackberry, Windows Phone, Palm WebOS, Bada, dan Symbian.

Apache Cordova dirilis pada Oktober 2012 sebagai projek tingkat atas dalam Apache Software Foundation (ASF). Melalui ASF, pengembangan projek Cordova di masa depan akan dipastikan tetap terbuka. Hal ini membuat Cordova tetap bersifat bebas dan *open source* di bawah lisensi Apache versi 2.0.

Artikel ini memberikan penjelasan dan pengenalan Cordova pada tingkat yang dasar dan berikut ini adalah langkah-langkah untuk memulai Cordova.

INSTALASI

Sebelum membangun aplikasi menggunakan API Cordova, terlebih dahulu Cordova harus diinstal pada sebuah komputer kerja. Silakan mengikuti langkah-langkah instalasi berikut ini:

A. Menginstal Node.js

Node.js adalah sebuah *engine* yang memungkinkan sebuah program JavaScript dapat berjalan pada sisi server (pada umumnya program JavaScript berjalan pada sisi client). Kemampuan Node.js ini dimanfaatkan untuk menjalankan program instalasi Cordova pada sebuah komputer.

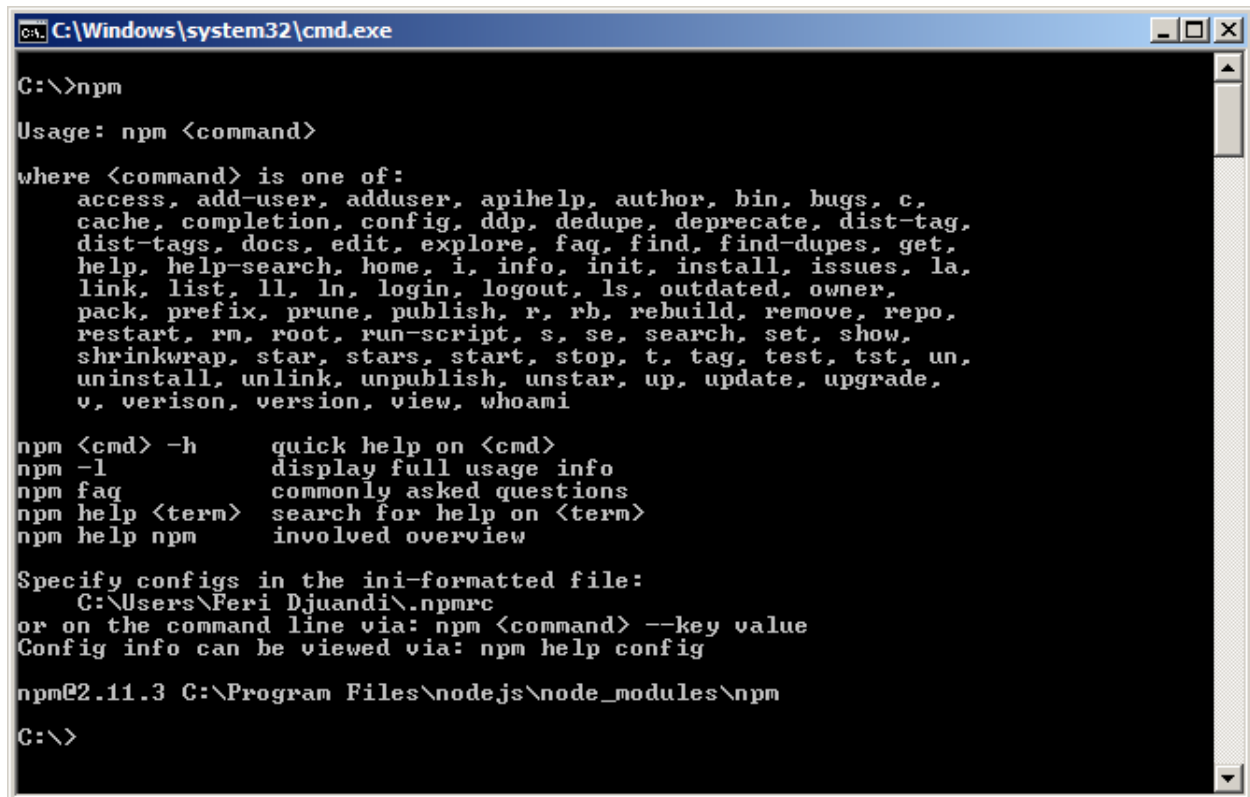
Silakan mengunduh node.js dari alamat situs di bawah ini dan lanjutkan dengan instalasinya.

<https://nodejs.org/>

Jika instalasi selesai dilakukan, lakukan uji coba dengan membuka sebuah *command shell* dan jalankan perintah:

```
npm
```

Jika instalasi telah benar dilakukan, maka tampilannya akan tampak seperti di bawah ini.



```
C:\Windows\system32\cmd.exe

C:\>npm

Usage: npm <command>

where <command> is one of:
  access, add-user, adduser, apihelp, author, bin, bugs, c,
  cache, completion, config, ddp, dedupe, deprecate, dist-tag,
  dist-tags, docs, edit, explore, faq, find, find-dupes, get,
  help, help-search, home, i, info, init, install, issues, la,
  link, list, ll, ln, login, logout, ls, outdated, owner,
  pack, prefix, prune, publish, r, rb, rebuild, remove, repo,
  restart, rm, root, run-script, s, se, search, set, show,
  shrinkwrap, star, stars, start, stop, t, tag, test, tst, un,
  uninstall, unlink, unpublish, unstar, up, update, upgrade,
  v, verison, version, view, whoami

npm <cmd> -h      quick help on <cmd>
npm -l            display full usage info
npm faq           commonly asked questions
npm help <term>   search for help on <term>
npm help npm      involved overview

Specify configs in the ini-formatted file:
  C:\Users\Feri Djuandi\.npmrc
or on the command line via: npm <command> --key value
Config info can be viewed via: npm help config

npm@2.11.3 C:\Program Files\nodejs\node_modules\npm
C:\>
```

B. Menginstal Cordova

Pastikan komputer terkoneksi ke Internet, kemudian buka sebuah command shell dan jalankan perintah berikut ini:

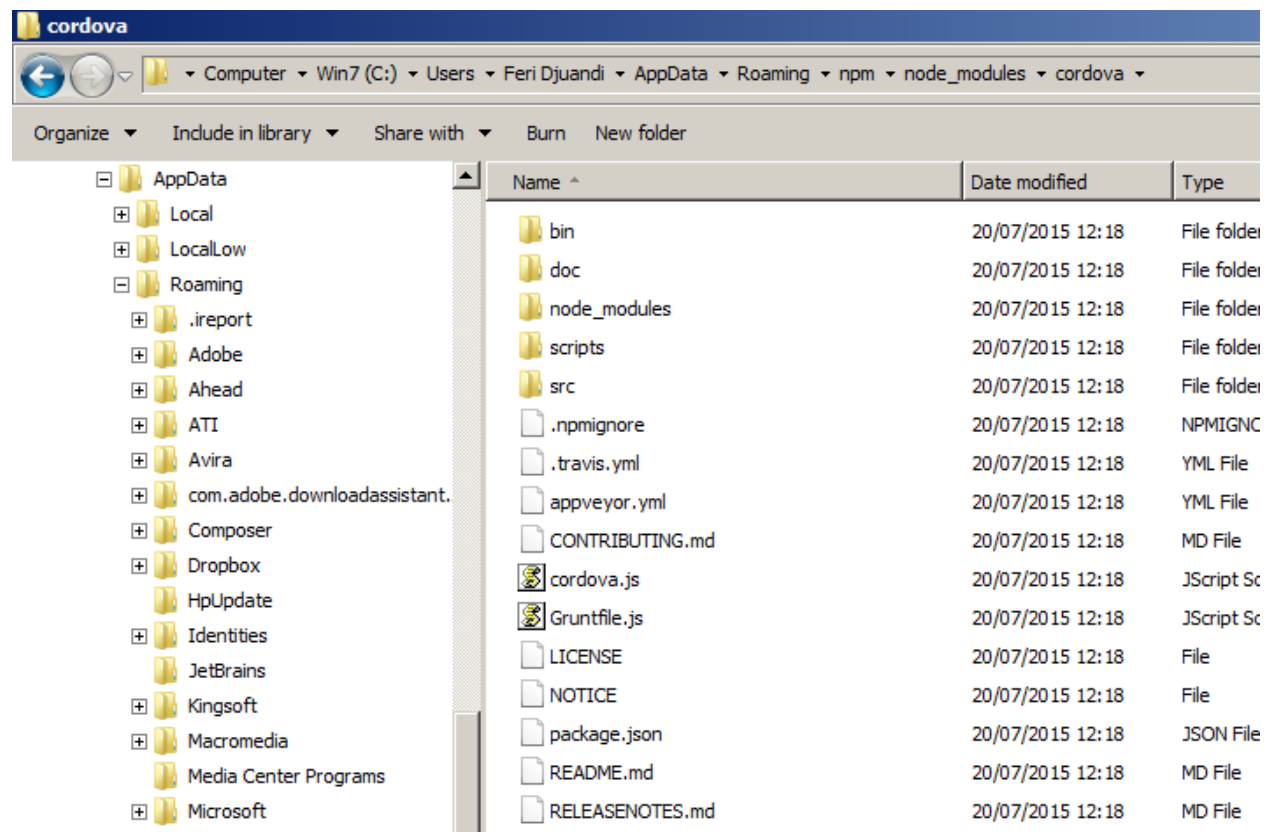
```
npm install -g cordova
```

Hasil instalasi tampak seperti berikut ini.

```
C:\Windows\system32\cmd.exe

C:\>npm install -g cordova
npm WARN engine cordova-js@4.0.0: wanted: {"node": ">=0.10.x"} <current: {"node": "0.12.7", "npm": "2.11.3"}>
npm WARN engine npm@1.3.4: wanted: {"node": ">=0.6", "npm": "1"} <current: {"node": "0.12.7", "npm": "2.11.3"}>
npm WARN engine xmlbuilder@2.2.1: wanted: {"node": ">=0.8.x && 0.10.x"} <current: {"node": "0.12.7", "npm": "2.11.3"}>
npm WARN installMany normalize-package-data was bundled with npm@1.3.4, but bundled package wasn't found in unpacked tree
C:\Users\Feri Djuandi\AppData\Roaming\npm\cordova -> C:\Users\Feri Djuandi\AppData\Roaming\npm\node_modules\cordova
cordova@5.1.1 C:\Users\Feri Djuandi\AppData\Roaming\npm\node_modules\cordova
├── underscore@1.7.0
├── q@1.0.1
├── nopt@3.0.1 <abbrev@1.0.7>
├── cordova-lib@5.1.1 <valid-identifier@0.0.1, osenv@0.1.0, properties-parser@0.2.3, bplist-parser@0.0.6, mime@1.2.11, unorm@1.3.3, semver@2.1.0, shelljs@0.3.0, dep-graph@1.1.0, rc@0.5.2, d8@0.4.4, cordova-app-hello-world@3.9.0, npmconf@0.1.16, through2@0.6.3, xcode@0.6.7, elementtree@0.1.6, glob@4.0.6, cordova-registry-mapper@1.1.8, init-package-json@1.7.1, tar@1.0.2, request@2.47.0, plist@1.1.0, npm@1.3.4, cordova-js@4.0.0>
C:\>
```

Lokasi hasil instalasi tampak seperti di bawah ini.



C. Membuat Aplikasi Cordova

Setelah semua instalasi berhasil dilakukan maka langkah selanjutnya adalah membuat aplikasi Cordova yang pertama. Tentunya aplikasi ini bentuknya sangat sederhana, namun tujuan yang terpenting disini adalah untuk memastikan aplikasi Cordova ini berhasil berjalan dengan baik pada sebuah perangkat mobile.

Membuat proyek baru

1. Buat sebuah folder kerja untuk proyek Cordova. Pada contoh ini folder tersebut diletakkan pada folder **D:\Cordova\workshop**. Anda bebas menentukan nama dan lokasi folder tersebut.
2. Buka sebuah command shell dan masuk ke dalam folder kerja di atas. Pada contoh ini akan dibuat sebuah proyek bernama **test1**.
3. Jalankan perintah berikut ini:

```
cordova create test1 com.exercise.test1 TestApp1
```

Setelah proyek berhasil dibuat, file-file proyek tersebut akan disimpan di dalam folder `..\Cordova\workshop\test1`. Silakan menuju folder tersebut dan melihat folder-folder dan file-file apa saja yang sudah terbentuk.

Menambahkan dukungan untuk platform Android

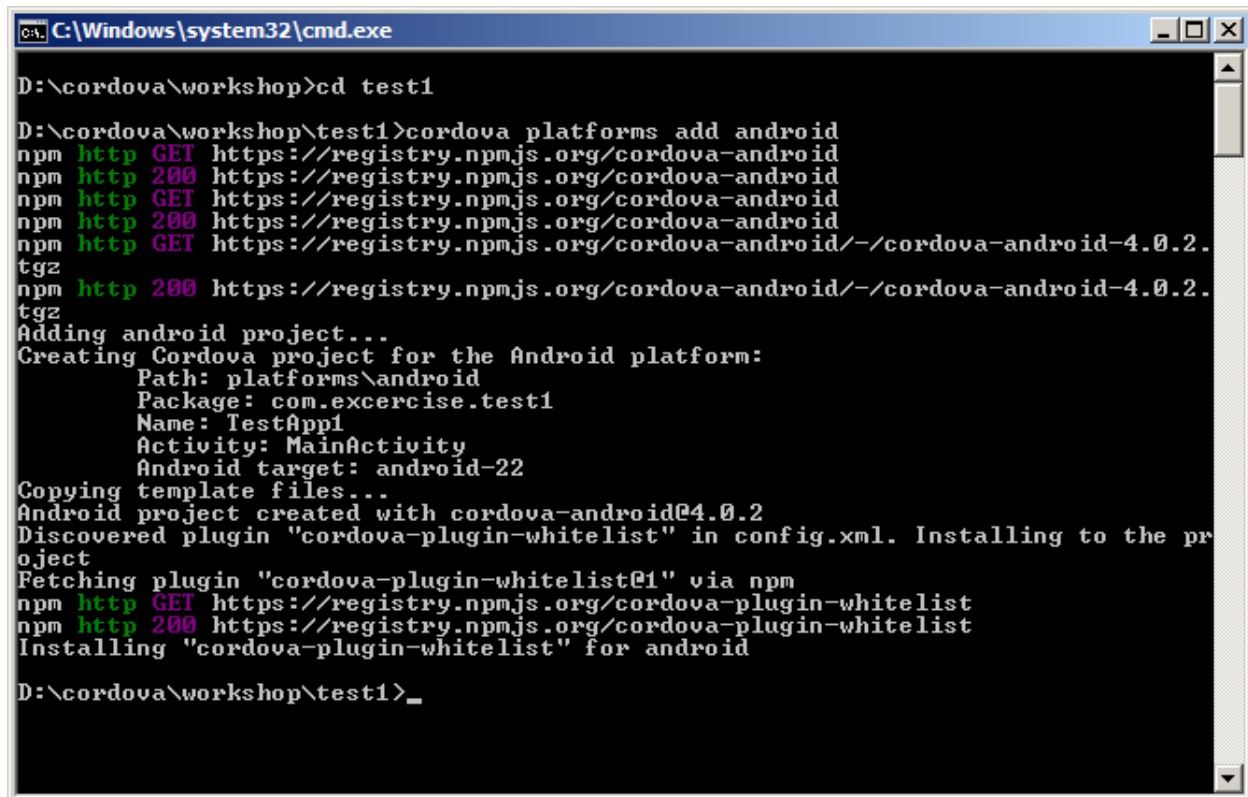
1. Pastikan Android SDK telah terinstal pada computer kerja. Jika belum terinstal, Android SDK bisa diperoleh dari situs berikut ini:

```
http://developer.android.com/sdk
```

2. Masuk ke dalam folder `..\Cordova\workshop\test1` dan jalankan perintah:

```
cordova platforms add android
```

Hasil eksekusi perintah tersebut diperlihatkan pada gambar berikut ini.



```
C:\Windows\system32\cmd.exe

D:\cordova\workshop>cd test1

D:\cordova\workshop\test1>cordova platforms add android
npm http GET https://registry.npmjs.org/cordova-android
npm http 200 https://registry.npmjs.org/cordova-android
npm http GET https://registry.npmjs.org/cordova-android
npm http 200 https://registry.npmjs.org/cordova-android
npm http GET https://registry.npmjs.org/cordova-android/-/cordova-android-4.0.2.tgz
npm http 200 https://registry.npmjs.org/cordova-android/-/cordova-android-4.0.2.tgz
Adding android project...
Creating Cordova project for the Android platform:
  Path: platforms\android
  Package: com.exercise.test1
  Name: TestApp1
  Activity: MainActivity
  Android target: android-22
Copying template files...
Android project created with cordova-android@4.0.2
Discovered plugin "cordova-plugin-whitelist" in config.xml. Installing to the project
Fetching plugin "cordova-plugin-whitelist@1" via npm
npm http GET https://registry.npmjs.org/cordova-plugin-whitelist
npm http 200 https://registry.npmjs.org/cordova-plugin-whitelist
Installing "cordova-plugin-whitelist" for android

D:\cordova\workshop\test1>_
```

Untuk menambahkan dukungan bagi platform lainnya, silakan mengikuti langkah-langkah serupa di atas.

Ikuti langkah-langkah berikut ini untuk menambahkan dukungan untuk platform iOS

1. Pastikan iOS SDK telah terinstal pada computer kerja.
2. Masuk ke dalam folder `..\Cordova\workshop\test1` dan jalankan perintah:

```
cordova platforms add ios
```

Catatan:

Pada artikel ini platform yang digunakan hanya Android, sehingga semua penjelasan yang diuraikan mengacu pada platform tersebut.

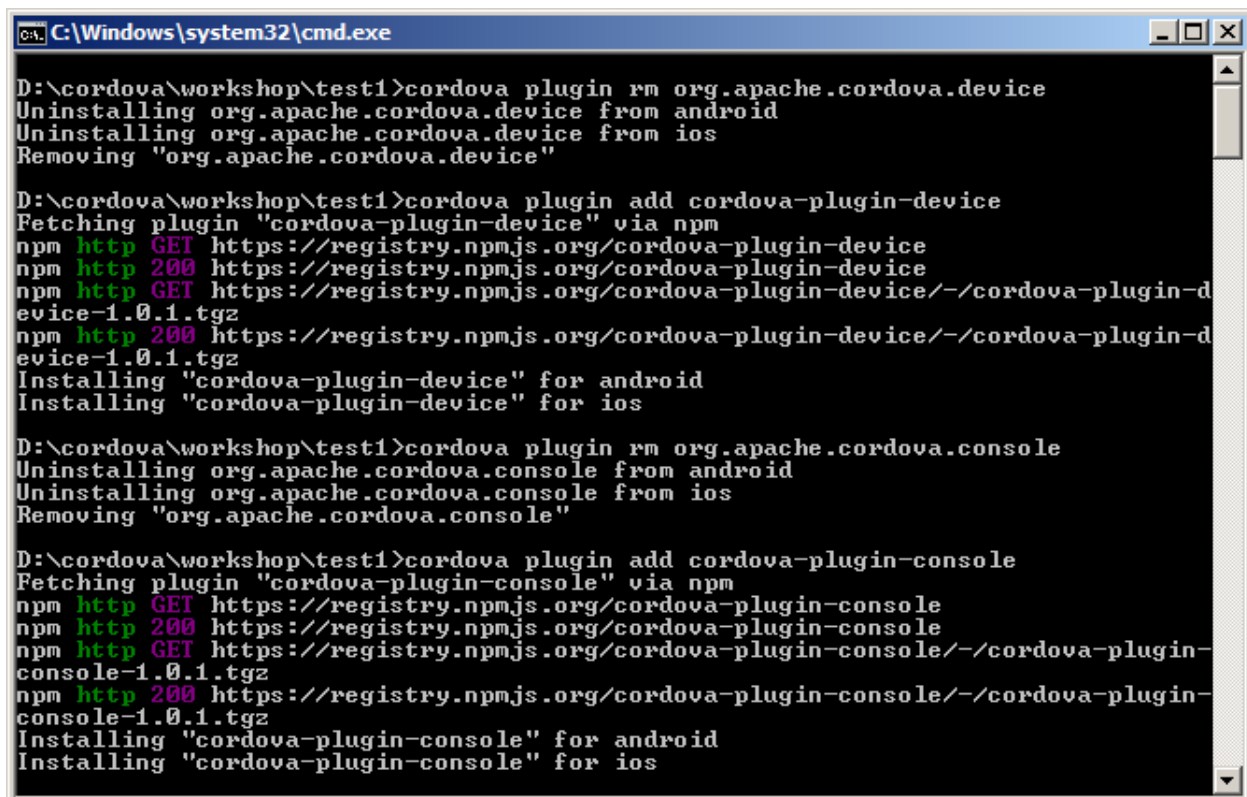
Menambahkan plugin dasar

Cordova bekerja pada perangkat mobile dengan memanfaatkan/mengakses plugin. **Device** dan **console** adalah dua plugin dasar yang perlu ditambahkan ke dalam aplikasi.

Masuk ke dalam folder `..\Cordova\workshop\test1` dan jalankan perintah:

```
cordova plugin add cordova-plugin-device
```

```
cordova plugin add cordova-plugin-console
```



```
C:\Windows\system32\cmd.exe

D:\cordova\workshop\test1>cordova plugin rm org.apache.cordova.device
Uninstalling org.apache.cordova.device from android
Uninstalling org.apache.cordova.device from ios
Removing "org.apache.cordova.device"

D:\cordova\workshop\test1>cordova plugin add cordova-plugin-device
Fetching plugin "cordova-plugin-device" via npm
npm http GET https://registry.npmjs.org/cordova-plugin-device
npm http 200 https://registry.npmjs.org/cordova-plugin-device
npm http GET https://registry.npmjs.org/cordova-plugin-device/-/cordova-plugin-device-1.0.1.tgz
npm http 200 https://registry.npmjs.org/cordova-plugin-device/-/cordova-plugin-device-1.0.1.tgz
Installing "cordova-plugin-device" for android
Installing "cordova-plugin-device" for ios

D:\cordova\workshop\test1>cordova plugin rm org.apache.cordova.console
Uninstalling org.apache.cordova.console from android
Uninstalling org.apache.cordova.console from ios
Removing "org.apache.cordova.console"

D:\cordova\workshop\test1>cordova plugin add cordova-plugin-console
Fetching plugin "cordova-plugin-console" via npm
npm http GET https://registry.npmjs.org/cordova-plugin-console
npm http 200 https://registry.npmjs.org/cordova-plugin-console
npm http GET https://registry.npmjs.org/cordova-plugin-console/-/cordova-plugin-console-1.0.1.tgz
npm http 200 https://registry.npmjs.org/cordova-plugin-console/-/cordova-plugin-console-1.0.1.tgz
Installing "cordova-plugin-console" for android
Installing "cordova-plugin-console" for ios
```

Menyesuaikan versi API Android

Saat hendak membuat (mem-*build*) sebuah paket aplikasi Cordova pada platform Android, Anda perlu memeriksa versi API Android yang terinstal pada komputer kerja – apakah sudah sesuai dengan versi target yang akan dibuat.

1. Silakan membuka file **project.properties** di dalam folder `..\workshop\test1\platforms\android`. File ini dibuat secara otomatis oleh API Cordova saat proyek ini dibuat.

Perhatikan bagian **target** di bawah ini.

```
# This file is automatically generated by Android Tools.
# Do not modify this file -- YOUR CHANGES WILL BE ERASED!
#
# This file must be checked in Version Control Systems.
#
# To customize properties used by the Ant build system edit
# "ant.properties", and override values to adapt the script to your
# project structure.
#
# To enable ProGuard to shrink and obfuscate your code, uncomment this (available properties: sdk.dir,
# user.home):
#proguard.config=${sdk.dir}/tools/proguard/proguard-android.txt:proguard-project.txt
# Project target.
target=android-21
android.library.reference.1=CordovaLib
```

Pada API Cordova yang digunakan oleh penulis, secara default target yang ditetapkan adalah android-22. Namun karena pada komputer kerja penulis versi API Android yang terinstal adalah 21 maka bagian ini harus diedit secara manual.

2. Buka file **AndroidManifest.xml** di dalam folder **..\workshop\test1\platforms\android**.

Sekali lagi pada bagian **android:targetSdkVersion**, ubah kode API Android sesuai dengan yang diinginkan.

```
<?xml version='1.0' encoding='utf-8'?>
<manifest android:hardwareAccelerated="true" android:versionCode="1" android:versionName="0.0.1"
package="com.exercise.test1" xmlns:android="http://schemas.android.com/apk/res/android">
  <supports-screens android:anyDensity="true" android:largeScreens="true"
android:normalScreens="true" android:resizeable="true" android:smallScreens="true"
android:xlargeScreens="true" />
  <uses-permission android:name="android.permission.INTERNET" />
  <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
  <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
  <application android:hardwareAccelerated="true" android:icon="@drawable/icon"
android:label="@string/app_name" android:supportsRtl="true">
    <activity android:configChanges="orientation|keyboardHidden|keyboard|screenSize|locale"
android:label="@string/activity_name" android:launchMode="singleTop" android:name="MainActivity"
android:theme="@android:style/Theme.Black.NoTitleBar"
android:windowSoftInputMode="adjustResize">
      <intent-filter android:label="@string/launcher_name">
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
  </application>
  <uses-sdk android:minSdkVersion="10" android:targetSdkVersion="21" />
</manifest>
```

Menjalankan projek pada perangkat Android

Setelah semua persiapan selesai dilakukan, sekarang saatnya untuk mencoba menjalankan aplikasi sederhana ini. Untuk menjalankan aplikasi ini dapat dilakukan melalui dua cara, yaitu (1) menggunakan emulator atau (2) langsung menggunakan perangkat Android.

Cara pertama bisa dilakukan jika Anda tidak memiliki perangkat Android. Namun cara ini tidak dianjurkan karena dibutuhkan waktu yang sangat lama untuk membuka program emulator Android dan memuat aplikasi ke dalamnya. Hal ini menjadi kendala besar saat uji coba yang perlu dilakukan berkali-kali karena banyaknya waktu yang terbuang.

Cara kedua lebih dianjurkan karena waktu instalasi aplikasi jauh lebih cepat dan kita langsung dapat merasakan jalannya aplikasi pada perangkat Android.

1. Sambungkan kabel data dari komputer ke perangkat Android. Pastikan perangkat Android terdeteksi oleh komputer.
2. Masuk ke dalam folder `..\Cordova\workshop\test1` dan jalankan perintah:

```
cordova run android
```

Contoh tampilan saat aplikasi di jalankan pada perangkat Android adalah sebagai berikut.



Untuk saat ini aplikasi sederhana ini tidak melakukan fungsi apa-apa selain menampilkan gambar di atas. Namun hasil ini sudah cukup mengindikasikan instalasi Cordova terinstal dengan baik pada komputer kerja dan aplikasi pun sudah bisa dibuat dan berjalan lancar.

Membuat file APK

Saat aplikasi dijalankan dengan perintah “run” seperti dijelaskan sebelumnya, Cordova membuat file APK yang terbentuk pada folder `..\workshop\test1\platforms\android\build\outputs\apk`.

Selanjutnya file APK tersebut disalin ke dalam perangkat mobile dan diinstal secara otomatis.

Jika tujuannya adalah hanya membuat (mem-build) file APK dan tanpa menjalankannya, maka perintah berikut ini dapat digunakan.

```
cordova build android
```

Hasil build akan dihasilkan pada folder “apk” yang sama seperti dijelaskan di atas.



Proverbs 1:7. The fear of the LORD is the beginning of knowledge: but fools despise wisdom and instruction.