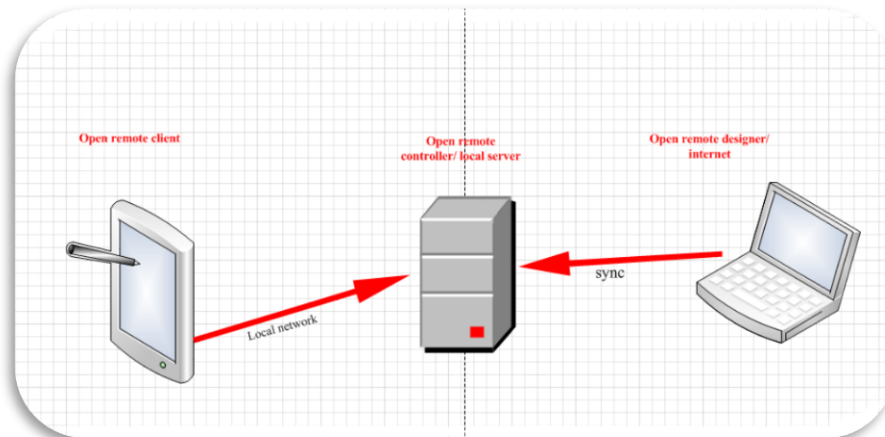


Definisi Open Remote

Open remote merupakan platform berbasis cloud yang berfungsi untuk memonitoring sebuah perangkat yang dapat diakses melalui Android, IPhone, IOS, dan perangkat lainnya. Platform ini terdiri dari tiga bagian, diantaranya :



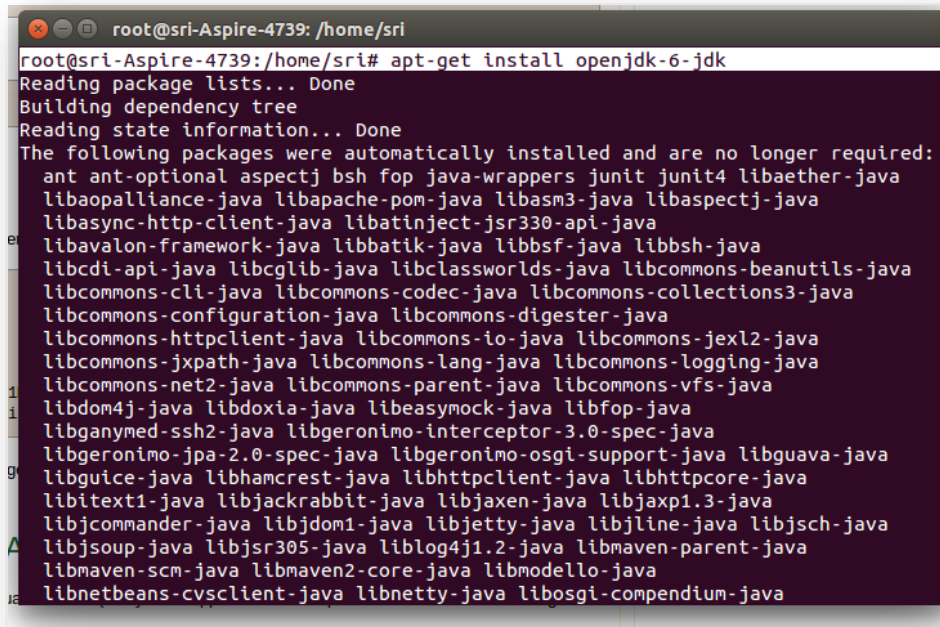
a. Gambar ilustrasi open remote

1. *Open remote controller*, berfungsi sebagai penghubung antara *client* dan *open remote designer* itu sendiri. Selain itu, *open remote controller* ini dapat membangun infrastruktur seperti : saklar, sensor, lampu, dan sebagainya yang sebelumnya telah didesain oleh *open remote designer*.
2. *Open remote client*, berfungsi sebagai interface pada Android, IOS, IPhone yang sebelumnya telah disinkronisasikan dengan local server (*open remote controller*).
3. *Open remote designer*, berfungsi sebagai tempat perancangan interface platform open remote sebelum akhirnya disinkronisasikan ke local server/ *open remote controller*.

🔧 Tutorial penggunaan platform open remote pada ubuntu server :

1. Tahap install open remote platform, dengan cara mengetikan command

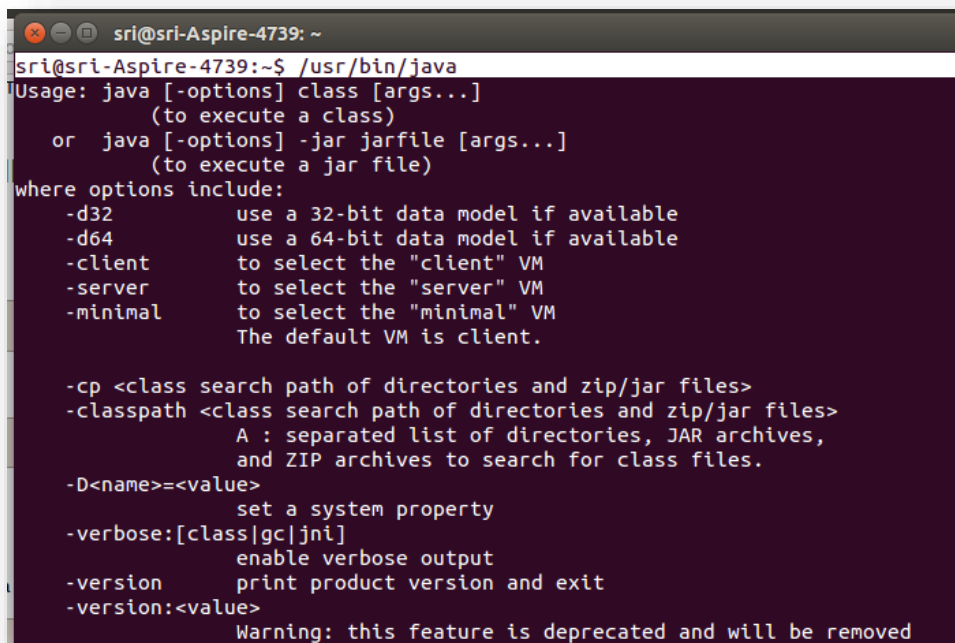
`apt-get install openjdk-6-jdk`



```
root@sri-Aspire-4739: /home/sri
root@sri-Aspire-4739: /home/sri# apt-get install openjdk-6-jdk
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
 ant ant-optional aspectj bsh fop java-wrappers junit junit4 libaether-java
 libaopalliance-java libapache-pom-java libasm3-java libaspectj-java
 libasync-http-client-java libatinject-jsr330-api-java
 libavalon-framework-java libbatik-java libbsf-java libbsh-java
 libcdi-api-java libcglib-java libclassworlds-java libcommons-beanutils-java
 libcommons-cli-java libcommons-codec-java libcommons-collections3-java
 libcommons-configuration-java libcommons-digester-java
 libcommons-httpclient-java libcommons-io-java libcommons-jexl2-java
 libcommons-jxpath-java libcommons-lang-java libcommons-logging-java
 libcommons-net2-java libcommons-parent-java libcommons-vfs-java
 libdom4j-java libdoxia-java libeasymock-java libfop-java
 libgeronimo-ssh2-java libgeronimo-interceptor-3.0-spec-java
 libgeronimo-jpa-2.0-spec-java libgeronimo-osgi-support-java libguava-java
 libguice-java libhamcrest-java libhttpclient-java libhttpcore-java
 libitext1-java libjackrabbit-java libjaxen-java libjaxp1.3-java
 libjcommander-java libjdom1-java libjetty-java libjline-java libjsch-java
 libjsoup-java libjsr305-java liblog4j1.2-java libmaven-parent-java
 libmaven-scm-java libmaven2-core-java libmodello-java
 libnetbeans-cvsclient-java libnetty-java libosgi-compndium-java
```

2. Cek instalasi java pada terminal dengan mengetikan command

`/usr/bin/java`

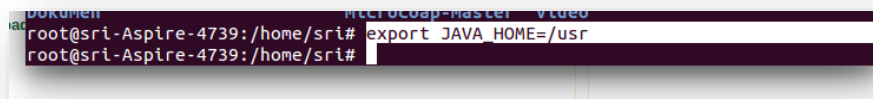


```
sri@sri-Aspire-4739: ~
sri@sri-Aspire-4739:~$ /usr/bin/java
Usage: java [-options] class [args...]
        (to execute a class)
   or java [-options] -jar jarfile [args...]
        (to execute a jar file)
where options include:
 -d32          use a 32-bit data model if available
 -d64          use a 64-bit data model if available
 -client      to select the "client" VM
 -server      to select the "server" VM
 -minimal     to select the "minimal" VM
              The default VM is client.

 -cp <class search path of directories and zip/jar files>
 -classpath <class search path of directories and zip/jar files>
        A : separated list of directories, JAR archives,
        and ZIP archives to search for class files.
 -D<name>=<value>
        set a system property
 -verbose:[class|gc|jni]
        enable verbose output
 -version     print product version and exit
 -version:<value>
        Warning: this feature is deprecated and will be removed
```

3. Mengatur java_home environment variable dengan memberikan command

`export JAVA_HOME=/usr`



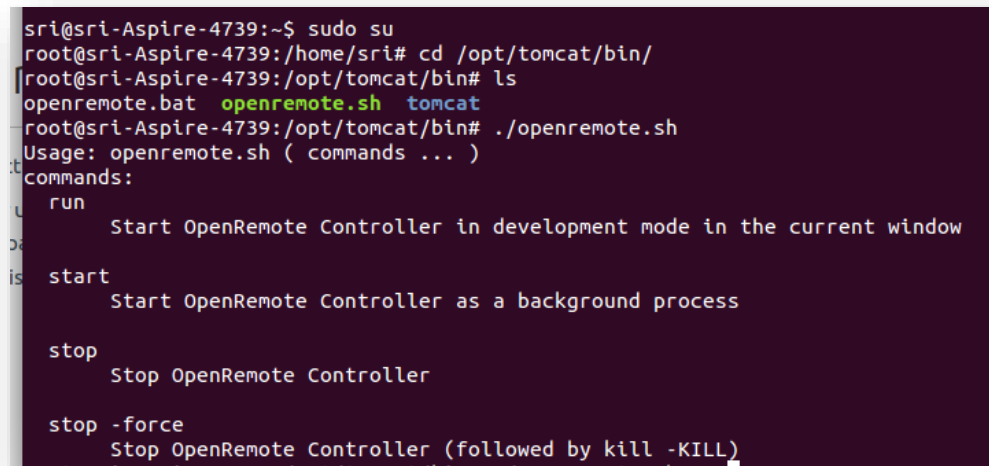
```
root@sri-Aspire-4739:/home/sri# export JAVA_HOME=/usr
root@sri-Aspire-4739:/home/sri#
```

4. Tahap selanjutnya download dan unzip open remote controller dari

<http://download.openremote.org>.

5. Jalankan open remote controller dengan melakukan command seperti dibawah ini :

- `cd /opt/OpenRemote-Controller/bin`



```
sri@sri-Aspire-4739:~$ sudo su
root@sri-Aspire-4739:/home/sri# cd /opt/tomcat/bin/
root@sri-Aspire-4739:/opt/tomcat/bin# ls
openremote.bat  openremote.sh  tomcat
root@sri-Aspire-4739:/opt/tomcat/bin# ./openremote.sh
Usage: openremote.sh ( commands ... )
commands:
  run
    Start OpenRemote Controller in development mode in the current window
  start
    Start OpenRemote Controller as a background process
  stop
    Stop OpenRemote Controller
  stop -force
    Stop OpenRemote Controller (followed by kill -KILL)
```

- pastikan open remote controller dapat dijalankan, jika belum dapat dijalankan ketikkan perintah dibawah ini pada terminal

`chmod +x openremote.sh`

- setelah itu jalankan open remote controller dengan command
`./openremote.sh run`

```
root@sri-Aspire-4739: /opt/tomcat/bin
root@sri-Aspire-4739:/home/sri# cd /opt/tomcat/bin/
root@sri-Aspire-4739:/opt/tomcat/bin# ls
openremote.bat  openremote.sh  tomcat
root@sri-Aspire-4739:/opt/tomcat/bin# ./openremote.sh run
Using CATALINA_BASE:   /opt/tomcat
Using CATALINA_HOME:   /opt/tomcat
Using CATALINA_TMPDIR: /opt/tomcat/temp
Using JRE_HOME:        /usr/lib/jvm/java-8-oracle

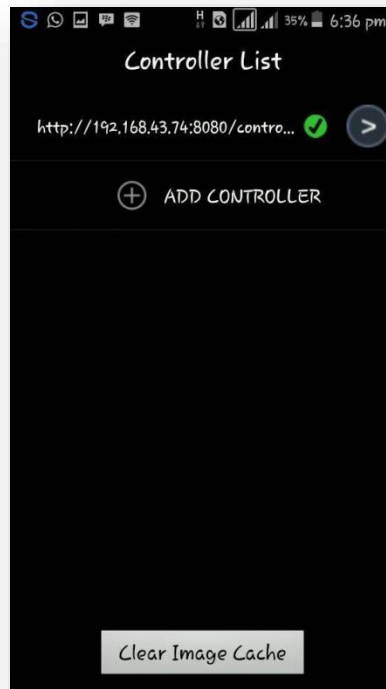
----- Logging -----
Console (stdout) threshold [CONTROLLER_CONSOLE_THRESHOLD]: INFO
System logs:
- Controller startup log [CONTROLLER_STARTUP_LOG_LEVEL]: DEBUG

-----
Sep 26, 2016 5:57:13 PM org.apache.catalina.core.AprLifecycleListener init
INFO: The APR based Apache Tomcat Native library which allows optimal performanc
e in production environments was not found on the java.library.path: /opt/tomcat
/webapps/controller/WEB-INF/lib/native
Sep 26, 2016 5:57:14 PM org.apache.coyote.http11.Http11Protocol init
```

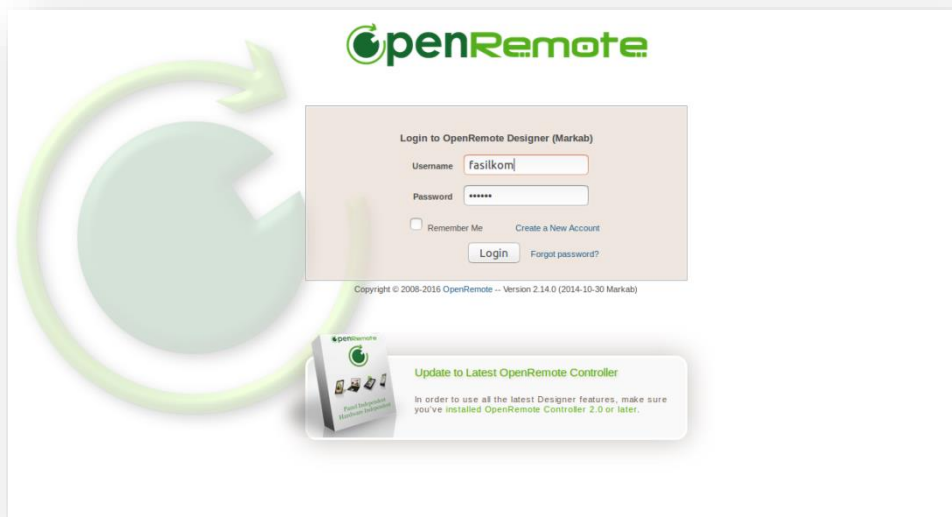
6. hubungkan open remote client dan open remote controller dengan mensinkronkan localhost:8080/controller/ ke device yang kita gunakan, seperti gambar dibawah ini :

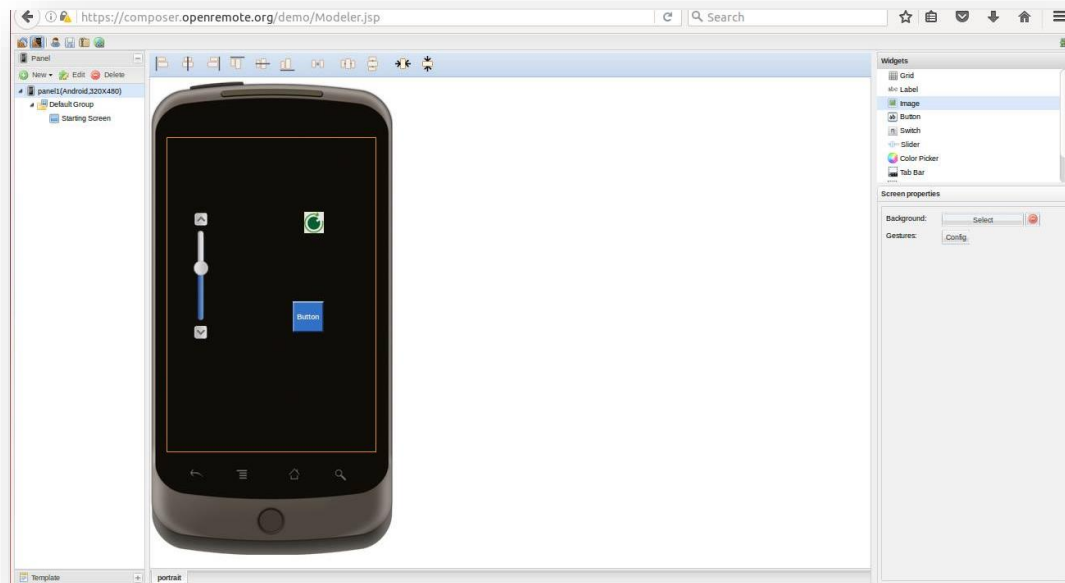
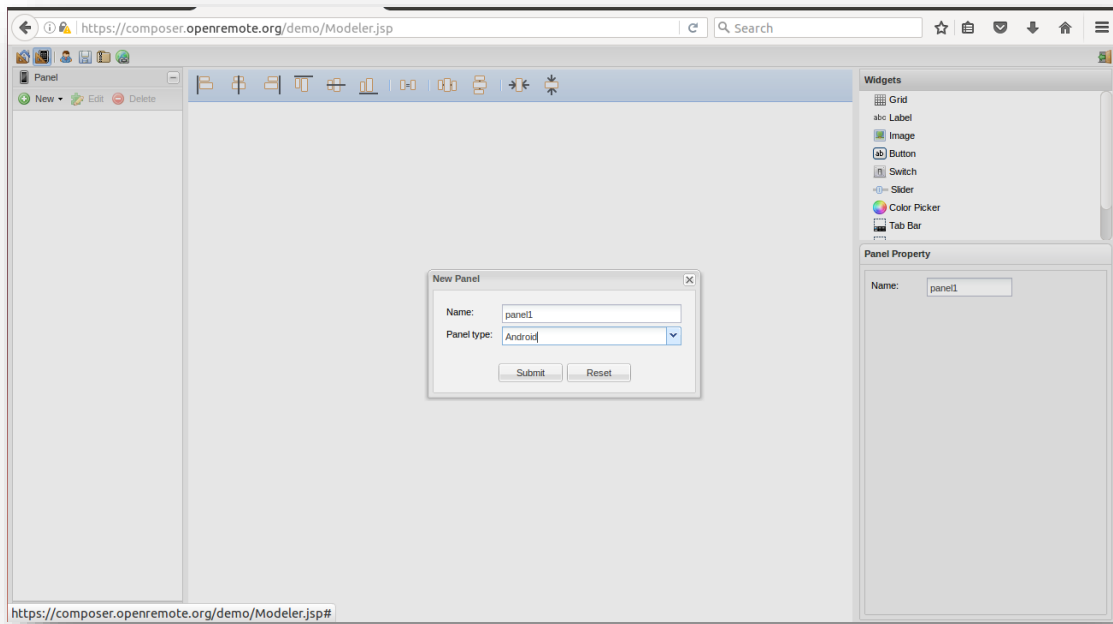


7. Setelah localhost dan client berhasil terhubung, maka di android akan muncul tanda ceklist berwarna hijau seperti gambar dibawah ini :



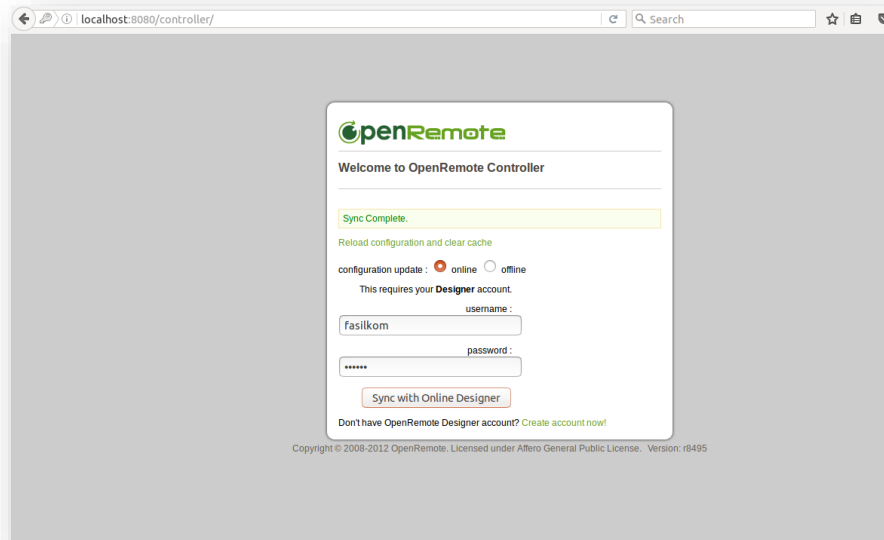
8. Setelah itu kita desain tampilan pada open remote designer , dibawah ini merupakan tahapan login hingga desain pada open remote designer :





(Gambar 8.1.) a. tampilan login ke open remote designer dengan menggunakan username dan password yang sebelumnya sudah kita aktivasi, b. tampilan untuk membuat panel baru dengan memasukkan nama dan type device yang digunakan, c. tampilan desain yang akan sinkronisasi dengan open remote client.

9. Setelah mendesain tampilan pada open remote designer, langkah selanjutnya adalah mensinkronkan tampilan pada open remote client dengan open remote designer, caranya kita akses localhost:8080/controller/ pada webserver seperti gambar dibawah ini :



10. Dibawah ini merupakan hasil tampilan dari open remote client dan open remote designer yang telah disinkronisasikan :

