

BAB V

PENUTUP

A. Kesimpulan

Bedasarkan dari pembahasan dan implementasi aplikasi MARB, maka dapat diambil kesimpulan sebagai berikut:

1. Aplikasi yang telah dibuat terdiri dari 2 menu utama, yaitu menu editor dan porting, sedangkan menu build untuk membuat update.zip yang nantinya di flash melalui custom recovery pada ponsel Android.
2. Hasil pengujian aplikasi MARB menunjukkan sistem dapat berjalan dengan baik. Hasil uji ini dilakukan di dua sistem operasi, yaitu Microsoft Windows XP (dengan virtualBox) dan Linux Kubuntu 14.04. Program berjalan dengan lancar dan tidak ada kendala apapun.

B. Saran

Pengembangan sistem MARB selanjutnya agar menulis ulang MARB dengan bahasa pemograman yang mendukung user interface. Agar semakin mudah dimengerti oleh pengguna dan developer custom rom mediatek.

DAFTAR PUSTAKA

Febriansyah. 2012. *Pembuatan custom ROM untuk HTC desire*. Skripsi. STMIK AKAKOM Yogyakarta

<http://pemula.linux.or.id/programming/bash-shell.html>. diakses pada 6 Desember 2015 pukul 10.14 WIB. Definisi bash Shell serta dasar-dasar pemrograman Bash Shell.

<https://www.techopedia.com/definition/8925/porting>. diakses pada 6 Desember 2015 pukul 10.51 WIB. Porting (Computer) Definition.

Krismuryanto, Bayu. 2014. *Remastering Android 4.1.2 Jellybean Untuk Multimedia Pada Samsung Galaxy Note 1 N7000*. Skripsi. STMIK AKAKOM Yogyakarta.

Nurul Kholis, I'mal. 2014. *Remastering sistem operasi android Untuk jaringan*. Skripsi. STMIK AKAKOM Yogyakarta.

Purnama, Dedhi. 2011. *Optimalisasi Sistem Operasi Android 2.1 Eclair pada Perangkat Mobile Htc Desire*. Skripsi. STMIK AMIKOM Yogyakarta.

LAMPIRAN

Script

Script add_busybox

```
#####  
#####  
#  
# project MTK Android ROM Builder  
# Copyright 2015 Blackant38 <ekoheriyanto8@gmail.com>  
#  
# MTK Android ROM Builder adalah projek open source yang di  
# khususkan  
# hanya untuk pribadi & atau edukasi.dimohon untuk tidak  
# menjualbelikan  
# atau menduplikat tanpa seizin pengembang.  
#  
#####  
#####  
#  
# exit 1 = busybox not created  
# exit 0 = busybox is now in working folder  
#  
  
if [ "$1" == "" ]  
then  
    clear  
  
elif [ "$1" == "show_help" ]  
then  
  
    clear  
    echo  
    echo "-----"  
    echo  
    echo "BusyBox merupakan file executable yang merupakan  
sekumpulan perintah UNIX"  
    echo "file ini dibutuhkan oleh beberapa aplikasi yang  
membutuhkan akses ROOT"  
    echo  
    echo "-----"  
    echo  
    echo -n "Tambahkan Busybox (y/n)?: "  
    read do_bb  
  
    if [ "$do_bb" == "n" ]  
    then  
        exit 1  
    fi  
fi  
  
if [ ! -d WORKING_* ]  
then
```

```

    echo tidak ditemukan folder kerja!
    exit 1
fi

cd WORKING_*

su_path=`find . -name su`

if [ "$su_path" == "" ]
then
    echo "binary 'su'tidak ditemukan di folder kerja!"
    cd ..
    exit 1
else
    echo
    echo "Ditemukan $su_path"
fi

version=`more ../tools/busybox_files/version.txt | sed -e
's/BusyBox \(.*\)$/\1/g`

if [ -e system/bin/busybox ]
then
    echo
    echo "folder kerja sudah terdapat /system/bin/busybox"

    if [ "$1" == "show_help" ]
    then
        echo -n "Ganti dengan Busybox $version (y/n)?: "
        read replace_bin_bb
        echo

        if [ "$replace_bin_bb" == "n" ]
        then
            echo "Dibatalkan"
        else
            echo "Mengganti /system/bin/busybox"
            cp -f ../tools/busybox_files/busybox system/bin/
        fi
    fi

    cd ..
    exit 0
fi

echo

if [ -d system/xbin ]
then
    echo Found /system/xbin

    if [ -e system/xbin/busybox ]
    then

        echo

```

```
echo "folder kerja sudah terdapat /system/xbin/busybox"

if [ "$1" == "show_help" ]
then
    echo -n "Ganti dengan Busybox $version (y/n)?: "
    read replace_bb
    echo
else
    replace_bb=n
fi

if [ "$replace_bb" == "n" ]
then
    cd ..
    exit 0
fi

echo "Menganti /system/xbin/busybox"
fi
else
    echo Membuat folder /system/xbin
    mkdir system/xbin
fi

echo "Menambahkan /system/xbin/busybox"
cp -f ../tools/busybox_files/busybox system/xbin/
cd ..

scripts/add_busybox_to_update_script

exit 0
```

Script add_busybox_to_update_script

```
#####
#####
#
# project MTK Android ROM Builder
# Copyright 2015 Blackant38 <ekoheriyanto8@gmail.com>
#
# MTK Android ROM Builder adalah projek open source yang di
khususkan
# hanya untuk pribadi & atau edukasi.dimohon untuk tidak
menjualbelikan
# atau menduplikat tanpa seizin pengembang.
#
#####
#####

if [ "$1" == "amend" ]
```

```

then
    convert_to_edify=0
else
    convert_to_edify=1
fi

cd WORKING_*
cd META-INF/com/google/android

echo

if [ -e update-script ]
then
    if [ "$convert_to_edify" == "0" ]
    then
        # Exit if all busybox symlinks already exist in update-script,
        # or a
        # busybox installer script exists in base dir of working
        # folder
        if [ "`grep -c \"symlink busybox\" update-script`" != "0" ] || \
        [ "`grep -m 1 \"busybox ln\" ../../../../../../*`" != "" ] || \
        [ "`grep -m 1 \"busybox --install\" ../../../../../../*`" != "" ]
        then
            cd ../../../../../../..
            exit 0
        fi

        #
        # Remove existing busybox references in update-script
        #
        search_run=`grep -c "run_program.*xbin/busybox --install"
update-script`
        search_perm=`grep -c "set_perm.*xbin/busybox" update-script`
        search_symlinks=`grep -c "symlink.*xbin/busybox " update-
script`

        if [ $search_run -gt 0 ] || [ $search_perm -gt 0 ] || [
$search_symlinks -gt 0 ]
        then
            echo "menghapus busybox /system/xbin/busybox di update-
script"
            sed -i \
            -e '/set_perm.*xbin\/busybox/d' \
            -e '/symlink.*xbin\/busybox/d' \
            -e '/run_program.*xbin\/busybox --install/d' \
            update-script
        fi

        echo "menambahkan installbusybox ke working folder"
        cp ../../../../../../tools/busybox_files/installbusybox
../../../../../../

```

```

echo "menambahkan installbusybox ke update-script"

if [ `grep -c "set_perm_recursive.*SYSTEM:xbin[ ]*$" update-
script` != 0 ]
then
sed -i -e 's/set_perm_recursive\(.*\)SYSTEM:xbin[
]*$/set_perm_recursive\1SYSTEM:xbin\nset_perm 0 1000 0755
SYSTEM:xbin\/busybox\nrun_program PACKAGE:installbusybox\n/'
update-script

elif [ `grep -c "init.goldfish.sh" update-script` != 0 ]
then
sed -i -e 's/set_perm 0 2000 0550
SYSTEM:etc\/init.goldfish.sh/set_perm 0 2000 0550
SYSTEM:etc\/init.goldfish.sh\nset_perm 0 1000 0755
SYSTEM:xbin\/busybox\nrun_program PACKAGE:installbusybox/g'
update-script

elif [ `grep -c "write_raw_image .*:boot.img BOOT:" update-
script` != 0 ]
then
sed -i -e 's/write_raw_image \(.*:boot.img\) BOOT:/set_perm
0 1000 0755 SYSTEM:xbin\/busybox\nrun_program
PACKAGE:installbusybox\nwrite_raw_image \1 BOOT:/' update-script

else
echo "set_perm 0 1000 0755 SYSTEM:xbin/busybox" >> update-
script
echo "run_program PACKAGE:installbusybox" >> update-script
fi

if [ `grep -c installbusybox update-script` == 0 ]
then
echo "Error: installbusybox tidak dapat ditambahkn ke
update-script!"
else

if [ ! -e ../../../../system/bin/busybox ]
then
echo "menambahkan symlink ke /system/bin/busybox"
sed -i -e 's/set_perm 0 1000 0755
SYSTEM:xbin\/busybox/set_perm 0 1000 0755
SYSTEM:xbin\/busybox\nsymlink \/system\/xbin\/busybox
SYSTEM:bin\/busybox/g' update-script
fi
fi

#####
###
# The update-script will be converted to updater-script when
built

#####
###

```



```

elif [ "$convert_to_edify" == "1" ]
then

    if [ -e ../../../../installbusybox ]
    then
        echo "menghapus BusyBox yang telah ada dan menambahkan
installbusybox ke update-script"
        sed -i \
            -e '/set_perm.*bin\/busybox/d' \
            -e '/symlink.*bin\/busybox/d' \
            -e '/run_program.*installbusybox/d' \
        update-script

        if [ -e ../../../../installbusybox ]
        then
            echo "menghapus installbusybox"
            rm -f ../../../../installbusybox
        fi

        echo

        elif [[ "$1" != "override" ]] && [[ `grep -c "symlink busybox"
update-script` != 0 || `grep -c "symlink /system/xbin/busybox "
update-script` != 0 ]]
        then
            echo "conten BusyBox telah terdapat di update-script"

            elif [[ "$1" != "override" ]] && [[ "`grep -m 1 \"busybox ln\"
../../../..//* 2>/dev/null`" != "" || "`grep -m 1 \"busybox --
install\" ../../../..//* 2>/dev/null`" != "" ]]
            then

                echo "scrip ini sudah memuat BusyBox symlinks"
                echo "tidak perlu perubahan pada script update-script"

            else

                echo "menambahkan BusyBox install command ke update-script"

                if [ `grep -c "set_perm_recursive.*SYSTEM:xbin[ ]*$" update-
script` != 0 ]
                then
                    sed -i -e 's/set_perm_recursive\(.*\)SYSTEM:xbin[
]*$/set_perm_recursive\1SYSTEM:xbin\nset_perm 0 1000 0755
SYSTEM:xbin\/busybox\nrun_program PACKAGE:system\/xbin\/busybox --
install -s SYSTEM:xbin\/n/' update-script

                    elif [ `grep -c "init.goldfish.sh" update-script` != 0 ]
                    then
                        sed -i -e 's/set_perm 0 2000 0550
SYSTEM:etc\/init.goldfish.sh/set_perm 0 2000 0550
SYSTEM:etc\/init.goldfish.sh\nset_perm 0 1000 0755
SYSTEM:xbin\/busybox\nrun_program PACKAGE:system\/xbin\/busybox --
install -s SYSTEM:xbin\/n/g' update-script

                    elif [ `grep -c "write_raw image .*:boot.img BOOT:" update-

```

```

script ` != 0 ]
    then
        sed -i -e 's/write_raw_image \(.*:boot.img\)
BOOT:/set_perm 0 1000 0755 SYSTEM:xbin\busybox\nrun_program
PACKAGE:system\bin\busybox --install -s
SYSTEM:xbin\nwrite_raw_image \1 BOOT:/' update-script

        else
            echo "set_perm 0 1000 0755 SYSTEM:xbin/busybox" >> update-
script
            echo "run_program PACKAGE:system\bin\busybox --install
-s SYSTEM:xbin" >> update-script
        fi

        if [ `grep -c "busybox --install -s" update-script` == 0 ]
        then
            echo "Error: BusyBox tidak dapat di tambahkan ke update-
script!"
        else

            if [ ! -e ../../../../../../system/bin/busybox ]
            then
                echo "menambahkan symlink ke /system/bin/busybox"
                sed -i -e 's/set_perm 0 1000 0755
SYSTEM:xbin\busybox/set_perm 0 1000 0755
SYSTEM:xbin\busybox\nsymlink \system\bin\busybox
SYSTEM:bin\busybox/g' update-script
            fi
        fi
    fi

else
    echo "Error: tidak ditemukan update script!"
fi

cd ../../../../../../..

scripts/fix_update_script_blanks

```

Script add_busybox_tool

```

#####
#####
#
# project MTK Android ROM Builder
# Copyright 2015 Blackant38 <ekoheriyanto8@gmail.com>
#
# MTK Android ROM Builder adalah projek open source yang di
khususkan

```

```

# hanya untuk pribadi & atau edukasi.dimohon untuk tidak
menjualbelikan
# atau menduplikat tanpa seizin pengembang.
#
#####
#####

if [ "$1" == "show_help" ]
then

    clear
    echo
    echo "-----"
    -----"
    echo
    echo "Fitur BusyBox Run-Parts memperbolehkan device untuk
menjalankan"
    echo "script di /system/etc/init.d saat reboot. "
    echo
    echo "-----"
    -----"
    echo

    echo -n "Tambahkan Busybox Run-Parts (y/n)?: "
    read do_runparts

    if [ "$do_runparts" == "n" ]
    then
        exit 0
    fi

elif [ "$1" == "for_apps2sd" ]
then

    echo
    echo "Optional: Busybox run-part dapat juga digunakan untuk
mengaktifkan fitur"
    echo "          App2sd.opsi ini memperbolehkan untuk menjalankan
script di folder,"
    echo "          /system/etc/init.d termasuk script app2sd."
    echo
    echo -n "Aktifkan app2sd dengan Busybox Run-Parts (y/n)?
(default: n): "
    read do_runparts

    if [ "$do_runparts" != "y" ]
    then
        exit 0
    fi

fi

```

```

cd WORKING_*

su_path=`find . -name su`

if [ "$su_path" == "" ]
then
    echo "'su' binary tidak ditemukan!"
    echo "Harus Root terlebih dahulu."
    cd ..
    exit 0
fi

#
# Create the folder
#

if [ ! -d system/etc/init.d ]
then
    echo
    echo "WARNING: tidak ditemukan folder /system/etc/init.d"
    echo
    echo "Membuat /system/etc/init.d folder ..."
    mkdir -p system/etc/init.d
    touch system/etc/init.d/placeholder
fi

cd ..
cd WORKING_*
if [ ! -d system/etc/init.d ]
then
    echo "tidak ditemukan folder /system/etc/init.d!"
    cd ..
    exit 0
fi
cd ..

echo
echo "karena device ini ($device) tidak memiliki file init.rc di
boot.img,"
echo "sebaiknya menggunakan custom kernel (zImage) yang support
busybox run-part"
echo
echo "Flash custom kernel dari forum device anda. bisa dicari di
xda-developer,"
echo -n "atur permission di /system/etc/init.d (y/n)?: "
read do_runparts

if [ "$do_runparts" == "n" ]
then
    echo "Dibatalkan"
    exit 0
fi

```

```

scripts/add_initd_perm_to_update_script

scripts/ensure_boot_extracted

if [ -d BOOT-EXTRACTED ]
then
  if [ -d WORKING_* ]
  then
    echo
  else
    exit 0
  fi
else
  exit 0
fi

if [ -d BOOT-EXTRACTED/boot.img-ramdisk ]
then
  cd BOOT-EXTRACTED/boot.img-ramdisk
else
  echo "Error: Folder BOOT-EXTRACTED/boot.img-ramdisk tidak
ditemukan !"
  exit 0
fi

test_property=`grep -c "on property:cm.filesystem.ready=1"
init.rc`
test_start=`grep -c "start sysinit" init.rc`
test_service=`grep -c "service sysinit" init.rc`
test_cm_sysinit=`grep -c "exec /system/bin/sysinit" init.rc`

if [ "$test_start" != "0" ] && [ "$test_service" != "0" ]
then
  echo "Busybox Run-Parts sudah terpasang di file init.rc!"
  cd ../../
  echo
  echo "Menghapus folder BOOT-EXTRACTED ..."
  rm -rf BOOT-EXTRACTED
  scripts/add_initd_perm_to_update_script
  exit 0

elif [ "$test_cm_sysinit" != "0" ]
then
  echo "sysinit /etc/init.d sudah dipasang di file init.rc!"
  cd ../../
  echo
  echo "Menghapus folder BOOT-EXTRACTED ..."
  rm -rf BOOT-EXTRACTED
  scripts/add_initd_perm_to_update_script
  exit 0

else

  cd ../../
  scripts/add_busybox

```

```

#cd WORKING_*
#bb_path=`find . -name busybox | grep -m 1 busybox | sed -e
's/\./g' -e 's/\/\\\/g'`
#cd ..

cd BOOT-EXTRACTED/boot.img-ramdisk

if [ `grep -c "start a2sd" init.rc` != 0 ]
then
echo
echo "Memastikan 'start sysinit' terdapat di init.rc ..."

sed -i -e 's/start a2sd/start sysinit/g' init.rc

echo
echo "Memastikan 'service sysinit' terdapat di init.rc..."

#sed -i -e 's/service a2sd[ ]*\system\bin\logwrapper[
]*\system\bin\sh[ ]*\system\bin\a2sd\service sysinit
\system\bin\logwrapper "$bb_path" run-parts
\system\etc\init.d/' init.rc
sed -i -e 's/service a2sd[ ]*\system\bin\logwrapper[
]*\system\bin\sh[ ]*\system\bin\a2sd\service sysinit
\system\bin\logwrapper \system\xbin\busybox run-parts
\system\etc\init.d/' init.rc

else

echo
echo Mengubah init.rc di ramdisk ...

if [ "$test_start" == "0" ] && [ "$test_property" == "0" ]
then
sed -i -e '0,/class_start /s//start sysinit\n\n
class_start /' init.rc

elif [ "$test_start" == "0" ] && [ "$test_property" != "0" ]
then
#
# jika a2sd sudah ada
#
sed -i -e 's/on property:cm.filesystem.ready=1/start
sysinit\n on property:cm.filesystem.ready=1/' init.rc
fi

if [ "$test_service" == "0" ]
then
#sed -i -e 's/service media /service sysinit
\system\bin\logwrapper "$bb_path" run-parts
\system\etc\init.d\n disabled\n oneshot\n\nservice media
/' init.rc
sed -i -e 's/service media /service sysinit
\system\bin\logwrapper \system\xbin\busybox run-parts
\system\etc\init.d\n disabled\n oneshot\n\nservice media
/' init.rc

```

```

        fi

    fi

    if [ "`grep sysinit init.rc`" == "" ]
    then
        echo
        echo "Error: tidak dapat mengubah isi init.rc"
    fi

    cd ../../
    scripts/add_initd_perm_to_update_script
    scripts/build_boot_img

fi

echo

fi

```

Script add_data_app_to_update_script

```

#####
#####
#
# project MTK Android ROM Builder
# Copyright 2015 Blackant38 <ekoheriyanto8@gmail.com>
#
# MTK Android ROM Builder adalah projek open source yang di
# khususkan
# hanya untuk pribadi & atau edukasi.dimohon untuk tidak
# menjualbelikan
# atau menduplikat tanpa seizin pengembang.
#
#####
#####
cd WORKING_*

if [ ! -e META-INF/com/google/android/update-script ]
then
    echo "Error: update-script tidak ditemukan!"
    cd ..
    exit 0
fi

cd ..
scripts/data_app_exists

```

```

if [ "$?" == 1 ]
then
    echo "update-script sudah memuat konten /data/app"
    exit 0

else

    cd WORKING_*
    cd META-INF/com/google/android

    echo "mengupdate update-script dengan entri /data/app"

    if [ `grep -c "delete DATA:app[ ]*" update-script` == 0 ]
    then
        if [ `grep -c "format .*DATA:" update-script` == 0 ]
        then
            if [ `grep -c "format .*SYSTEM:" update-script` == 0 ]
            then
                sed -i -e 'li delete DATA:app\n' update-script
            else
                sed -i -e 's/format\(.*\)SYSTEM:[
]*[\n]*/format\1SYSTEM:\n\ndelete DATA:app\n/' update-script
            fi
        else
            sed -i -e 's/format\(.*\)DATA:[
]*[\n]*/format\1DATA:\n\ndelete DATA:app\n/' update-script
            fi
        fi

        if [ `grep -c "copy_dir PACKAGE:data DATA:" update-script` == 0
        ]
        then
            sed -i -e 's/delete DATA:app[ ]*[\n]*/delete
DATA:app\ncopy_dir PACKAGE:data DATA:\n/' update-script
            fi

            sed -i -e 's/copy_dir PACKAGE:data DATA:[ ]*[\n]*/copy_dir
PACKAGE:data DATA:\nset_perm_recursive 1000 1000 0771 0644
DATA:app\n/' update-script

        fi

        cd ../../../../../../..

scripts/fix update script blanks

```

Script add_initd_perm_to_update_script

```

#####
#####
#
# project MTK Android ROM Builder
# Copyright 2015 Blackant38 <ekoheriyanto8@gmail.com>

```



```

#
# MTK Android ROM Builder adalah proyek open source yang di
# khususkan
# hanya untuk pribadi & atau edukasi.dimohon untuk tidak
# menjualbelikan
# atau menduplikat tanpa seizin pengembang.
#
#####
#####

cd WORKING_*

if [ -d system/etc/init.d ]
then

    echo
    echo "mengecek apabila update-script membutuhkan update untuk
/system/etc/init.d ..."

    cd META-INF/com/google/android

    if [ -e update-script ]
    then

        if [ `grep -c "set_perm_recursive .*SYSTEM:etc/init.d$"
update-script` == 0 ]
        then

            echo "update-script sedang di modif"

            # set_perm_recursive SYSTEM:
            grep_perm=`grep set_perm_recursive update-script | grep -m 1
"SYSTEM:[ ]*$"`

            if [ "$grep_perm" == "" ]
            then
                sed -i -e 's/copy_dir PACKAGE:system SYSTEM:/copy_dir
PACKAGE:system SYSTEM:\nset_perm_recursive 0 0 0777 0777
SYSTEM:etc\//init.d/g' update-script
            else
                sed -i -e 's/\(set_perm_recursive[ ]*[^\ ]*[ ]*[^\ ]*[
 ]*[^\ ]*[ ]*[^\ ]*[ ]*[^\ ]*\)/\1\nset_perm_recursive 0 0
0777 0777 SYSTEM:etc\//init.d/g' update-script
            fi

            if [ `grep -c "set_perm_recursive .*SYSTEM:etc/init.d$"
update-script` == 0 ]
            then
                echo "Error: tidak ditemukan baris update di update-
script"
            fi

            else
                echo "update-script sudah mendapatkan permissions set untuk
folder init.d"
            fi
        fi
    fi
fi

```

```

    fi

    else
        echo "Error: tidak ditemukan update-script!"
    fi

    cd ../../../../

fi

cd ..
scripts/fix_update_script_blanks

```

Script add_su_to_update_script

```

#####
#####
#
# project MTK Android ROM Builder
# Copyright 2015 Blackant38 <ekoheriyanto8@gmail.com>
#
# MTK Android ROM Builder adalah projek open source yang di
# khususkan
# hanya untuk pribadi & atau edukasi.dimohon untuk tidak
# menjualbelikan
# atau menduplikat tanpa seizin pengembang.
#
#####
#####

cd WORKING_*
cd META-INF/com/google/android

if [ -e update-script ]
then

    sed -i -e '/bin\su[ ]*$/d' update-script

    if [ -e ../../../../system/bin/su ]
    then

        if [ "`find ../../../../system/xbin -name su`" == "" ]
        then
            echo "memindahkan system/bin/su ke system/xbin/su"
            mkdir -p ../../../../system/xbin
            mv -f ../../../../system/bin/su ../../../../system/xbin/su
        else
            echo "/system/xbin/su ditemukan, memindahkan ke
/system/bin/su"
            rm -f ../../../../system/bin/su
        fi
    fi
fi

```

```

elif [ "`find ../../../../system/sbin -name su`" == "" ]
then
    echo "Error: /system/sbin/su adan /system/bin/su tidak
ditemukan!"
    cd ../../../../...
    echo
    exit 0
fi

echo
echo "update update-script dengan entri su dan symbolic link"

if [ `grep -c "set_perm_recursive.*SYSTEM:sbin[ ]*$" update-
script` != 0 ]
then
    sed -i -e 's/set_perm_recursive\(.*\)SYSTEM:sbin[
]*$/set_perm_recursive\1SYSTEM:sbin\nset_perm 0 0 06755
SYSTEM:sbin\su\nsymlink \system\sbin\su SYSTEM:bin\su\n/'
update-script

    elif [ `grep -c "init.goldfish.sh" update-script` != 0 ]
    then
        sed -i -e 's/set_perm 0 2000 0550
SYSTEM:etc\init.goldfish.sh/set_perm 0 2000 0550
SYSTEM:etc\init.goldfish.sh\nset_perm 0 0 06755
SYSTEM:sbin\su\nsymlink \system\sbin\su SYSTEM:bin\su/g'
update-script

        elif [ `grep -c "write_raw_image .*:boot.img BOOT:" update-
script` != 0 ]
        then
            sed -i -e 's/write_raw_image \(.*:boot.img\) BOOT:/set_perm 0
0 06755 SYSTEM:sbin\su\nsymlink \system\sbin\su
SYSTEM:bin\su\nwrite_raw_image \1 BOOT:/' update-script

        else
            echo "set_perm 0 0 06755 SYSTEM:sbin/su" >> update-script
            echo "symlink /system/sbin/su SYSTEM:bin/su" >> update-script
        fi

        if [ ! -d ../../../../system/bin ]
        then
            echo "Creating /system/bin folder"
            mkdir -p ../../../../system/bin
            touch ../../../../system/bin/placeholder
        fi

        if [ `grep -c "/sbin/su " update-script` == 0 ]
        then
            echo "Error: tidak dapat mengupdate update-script dengan SU"
        fi

    else
        echo "Error: tidak ditemukan updatescript!"
    fi
fi

```

```
cd ../../../../..  
scripts/fix_update_script_blanks
```

Script adjust_mnt

```
#####  
#####  
#  
# project MTK Android ROM Builder  
# Copyright 2015 Blackant38 <ekoheriyanto8@gmail.com>  
#  
# MTK Android ROM Builder adalah projek open source yang di  
# khususkan  
# hanya untuk pribadi & atau edukasi.dimohon untuk tidak  
# menjualbelikan  
# atau menduplikat tanpa seizin pengembang.  
#  
#####  
#####  
  
#  
# script tambahan jika terdapat perintah:  
# $1 = updater-script/update-script  
#  
  
if [ "$1" != "" ]  
then  
  
    if [ "$1" != "updater-script" ] && [ "$1" != "update-script" ]  
    then  
        echo "Error: Spesifik '$1' sebagai ganti dari updater-script  
atau update-script"  
        exit 1  
    fi  
  
    cd WORKING_*  
    working_folder=`pwd`  
    cd ..  
  
#  
# Find device name for lookup in edify_defs folder  
#  
if [ "$device_name" == "" ]  
then  
    device_name=`scripts/get_device_name`
```

```

if [ -e $working_folder/boot.img ]
then
    scripts/check_mt65xx_bootimg
    is_mt65xx_bootimg=$?
fi

else

    is_mt65xx_bootimg=0

echo
echo "Nama device adalah: $device_name"

change_mnt=`scripts/get_edify_def_val $device_name change_mnt`
change_mnt_dell=`scripts/get_edify_def_val $device_name
change_mnt_dell`

if [ "$change_mnt" == "yes" ] || [ "$change_mnt_dell" == "yes" ]
then
    param1=`scripts/get_edify_def_val $device_name param1`
    param2=`scripts/get_edify_def_val $device_name param2`
    param1_sdcard=`scripts/get_edify_def_val $device_name
param1_sdcard`
    param2_sdcard=`scripts/get_edify_def_val $device_name
param2_sdcard`
    param1_cache=`scripts/get_edify_def_val $device_name
param1_cache`
    param2_cache=`scripts/get_edify_def_val $device_name
param2_cache`
    fix_boot=`scripts/get_edify_def_val $device_name fix_boot`
    fix_boot2=`scripts/get_edify_def_val $device_name fix_boot2`
    fix_boot3=`scripts/get_edify_def_val $device_name fix_boot3`
    use_cache_not_tmp=`scripts/get_edify_def_val $device_name
use_cache_not_tmp`
    sys_mnt=`scripts/get_edify_def_val $device_name sys_mnt`
    cache_mnt=`scripts/get_edify_def_val $device_name cache_mnt`
    boot_mnt=`scripts/get_edify_def_val $device_name boot_mnt`
    data_mnt=`scripts/get_edify_def_val $device_name data_mnt`
    sdcard_mnt=`scripts/get_edify_def_val $device_name sdcard_mnt`
    webtop_mnt=`scripts/get_edify_def_val $device_name webtop_mnt`
    modem_mnt=`scripts/get_edify_def_val $device_name modem_mnt`
    preload_mnt=`scripts/get_edify_def_val $device_name
preload_mnt`
fi

cd WORKING_*
cd META-INF/com/google/android

if [ "$change_mnt" == "yes" ]
then

    if [ "$param1_cache" == "UNKNOWN" ]
    then

```

```

    param1_cache=$param1
fi
if [ "$param2_cache" == "UNKNOWN" ]
then
    param2_cache=$param2
fi
if [ "$param1_sdcard" == "UNKNOWN" ]
then
    param1_sdcard=$param1
fi
if [ "$param2_sdcard" == "UNKNOWN" ]
then
    param2_sdcard=$param2
fi

#####
# change_mnt==yes; updater-script
#####

if [ "$1" == "updater-script" ]
then
    sed -i \
        -e 's/format("MTD", "system")/format("'"$param1"'",
        "'"$param2"'", "'"$sys_mnt"'")/g' \
        -e 's/format("MTD", "preload")/format("'"$param1"'",
        "'"$param2"'", "'"$preload_mnt"'")/g' \
        -e 's/format("MTD", "cache")/format("'"$param1_cache"'",
        "'"$param2_cache"'", "'"$cache_mnt"'")/g' \
        -e 's/format("MTD", "boot")/format("'"$param1"'",
        "'"$param2"'", "'"$boot_mnt"'")/g' \
        -e 's/format("MTD", "userdata")/format("'"$param1"'",
        "'"$param2"'", "'"$data_mnt"'")/g' \
        \
        -e 's/format("'"$param1"'", "MTD",
        "system")/format("'"$param1"'", "'"$sys_mnt"'")/g' \
        -e 's/format("'"$param1"'", "MTD",
        "preload")/format("'"$param1"'", "'"$preload_mnt"'")/g' \
        -e 's/format("'"$param1_cache"'", "MTD",
        "cache")/format("'"$param1_cache"'", "'"$cache_mnt"'")/g' \
        -e 's/format("'"$param1"'", "MTD",
        "boot")/format("'"$param1"'", "'"$boot_mnt"'")/g' \
        -e 's/format("'"$param1"'", "MTD",
        "userdata")/format("'"$param1"'", "'"$data_mnt"'")/g' \
        \
        -e 's/mount("MTD", "system"/mount("'"$param1"'",
        "'"$param2"'", "'"$sys_mnt"'")/g' \
        -e 's/mount("MTD", "preload"/mount("'"$param1"'",
        "'"$param2"'", "'"$preload_mnt"'")/g' \
        -e 's/mount("MTD", "cache"/mount("'"$param1_cache"'",
        "'"$param2_cache"'", "'"$cache_mnt"'")/g' \
        -e 's/mount("MTD", "boot"/mount("'"$param1"'",
        "'"$param2"'", "'"$boot_mnt"'")/g' \
        -e 's/mount("MTD", "userdata"/mount("'"$param1"'",
        "'"$param2"'", "'"$data_mnt"'")/g' \
        -e 's/mount("MTD", "sdcard"/mount("'"$param1_sdcard"'",

```

```

""$param2_sdcard"", ""$sdcard_mnt""/g' \
\
-e 's/mount("""$param1"", "MTD",
"system"/mount("""$param1"", ""$sys_mnt""/g' \
-e 's/mount("""$param1"", "MTD",
"preload"/mount("""$param1"", ""$preload_mnt""/g' \
-e 's/mount("""$param1_cache"", "MTD",
"cache"/mount("""$param1_cache"", ""$cache_mnt""/g' \
-e 's/mount("""$param1"", "MTD",
"boot"/mount("""$param1"", ""$boot_mnt""/g' \
-e 's/mount("""$param1"", "MTD",
"userdata"/mount("""$param1"", ""$data_mnt""/g' \
-e 's/mount("""$param1_sdcard"", "MTD",
"sdcard"/mount("""$param1_sdcard"", ""$sdcard_mnt""/g' \
\
-e 's/\("""$webtop_mnt"", "\)???"/\1\webtop"/g' \
updater-script

if [ `grep -c "^mount.*webtop" updater-script` -gt 0 ]
then
echo "umount(\"/webtop\");" >> updater-script
fi

#
# Certain devices use/require a special way of flashing the
boot.img
#
# NOTE: :a;N;$!ba; = reads multi-line pattern as one-line
pattern
#

if [ -e updater-script.orig ] && [ "$fix_boot" != "yes" ]
then
grep_boot=`grep -c "^package_extract_file(\"boot.img\",
\"$boot_mnt\"");" updater-script.orig`
if [ $grep_boot != 0 ]
then
fix_boot=yes
fi
fi

if [ "$fix_boot" == "yes" ]
then
sed -i -e
':a;N;$!ba;s/assert(package_extract_file("boot.img",
"/tmp/boot.img"),\n[ ]*write_raw_image("/tmp/boot.img",
"boot"),\n[
]*delete("/tmp/boot.img));/package_extract_file("boot.img",
""$boot_mnt"");/g' updater-script

elif [ "$fix_boot2" == "yes" ]
then
sed -i -e 's/write_raw_image("/tmp/boot.img",
"boot")/write_raw_image("/tmp/boot.img", ""$boot_mnt""/g'
updater-script

```

```

elif [ "$fix_boot3" == "yes" ]
then
    sed -i -e 's/write_raw_image("\tmp\boot.img",
"boot")/run_program("\sbin\busybox", "dd", "if=\tmp\boot.img",
"of='$boot_mnt'")/g' updater-script
    fi

#
# Fix cases where some formats/mounts have different numbers
of args in the same file
#
# e.g.  format ext3 SYSTEM:
#       format DATA:
#

mount_four_args=`grep "^mount" updater-script | grep -c
$param2`
if [ "$param2_cache" != "$param2" ]
then
    mount_four_args2=`grep "^mount" updater-script | grep -c
$param2_cache`
else
    mount_four_args2=0
fi
if [ "$param2_sdcard" != "$param2" ]
then
    mount_four_args3=`grep "^mount" updater-script | grep -c
$param2_sdcard`
else
    mount_four_args3=0
fi

mount_four_args_total=$((mount_four_args+mount_four_args2+mount
_four_args3))
mount_all=`grep -c "^mount" updater-script`

if [ $mount_all -gt $mount_four_args_total ] && [
$mount_four_args_total -gt 0 ]
then
    sed -i \
        -e 's/mount("'"$param1"'",
        "'"$param2"'",/mount("'"$param1"'",/g' \
        -e 's/mount("'"$param1_cache"'",
        "'"$param2_cache"'",/mount("'"$param1_cache"'",/g' \
        -e 's/mount("'"$param1_sdcard"'",
        "'"$param2_sdcard"'",/mount("'"$param1_sdcard"'",/g' \
        updater-script
    fi

fmt_three_args=`grep "^format" updater-script | grep -c
$param2`

```



```

if [ "$param2_cache" != "$param2" ]
then
    fmt_three_args2=`grep "^format" updater-script | grep -c
$param2_cache`
    else
        fmt_three_args2=0
    fi

    fmt_three_args_total=$((fmt_three_args+fmt_three_args2))
    fmt_all=`grep -c "^format" updater-script`

    if [ $fmt_all -gt $fmt_three_args ] && [ $fmt_three_args -gt
0 ]
    then
        sed -i -e 's/format(""$param1""",
""$param2""",/format(""$param1""",/g' updater-script
        sed -i -e 's/format(""$param1_cache""",
""$param2_cache""",/format(""$param1_cache""",/g' updater-script
    fi

    #
    # Cover the case where the ROM uses a different filesystem
from what was defined
    # above (for both 'format' and 'mount')
    #
    # e.g. updater-script uses ext4 in CyanogenMod instead of
the expected ext3
    #

    if [ -e updater-script.orig ]
    then
        param1_old=`grep -m 1 "mount(.*$sys_mnt" updater-
script.orig | \
            sed -e 's/mount("\(^[^"]*\)\.*\/\1/g'`
        param2_old=`grep -m 1 "mount(.*$sys_mnt" updater-
script.orig | \
            sed -e 's/mount("\(^[^"]*" * [ ]*, [
]*"\(^[^"]*\)\.*\/\1/g'`

        if [ "$param1_old" != "" ] && [ "$param2_old" != "" ]
        then
            if [ "$param1_old" != "$param1" ]
            then
                sed -i -e 's/"$param1"/"$param1_old"/g' updater-
script
            fi
            if [ "$param2_old" != "$param2" ]
            then
                sed -i -e 's/"$param2"/"$param2_old"/g' updater-
script
            fi
        fi
    fi

    #

```

```

        # Check if the updater-script uses /cache instead of /tmp
for run_program
        # or writing images.
        #
        if [ `grep -c "/cache/boot.img" updater-script.orig` -gt 0
] && \
        [ `grep -c "\"/tmp/" updater-script` -gt 0 ]
        then
            use_cache_not_tmp=yes
        elif [ `grep -c "\"/tmp/" updater-script` -gt 0 ]
        then
            use_cache_not_tmp=no
        fi
    fi

    #
    # use_cache_not_tmp can be set to 'yes' in either edify_defs
file or from updater-script.orig
    #
    if [ "$use_cache_not_tmp" == "yes" ]
    then
        sed -i \
            -e 's/\\/tmp\\/\\/\\/cache\\/\\/g' \
            -e 's/\\(\"\\/cache\\/boot.img",
\\)\"boot\"/\\1\"'$boot_mnt'\"'/g' \
            updater-script
        fi

    #
    # Fix case where a shell script was used for mounting
instead of the 'mount' command
    #
    if [ `grep -c "mount(\\.*$sys_mnt\\", \\.*\\\"" updater-script`
-gt 0 ]
    then
        if [ `grep -c "auto \\$PARTITION \\$ANDROID_ROOT" updater-
script` -gt 0 ]
        then
            sed -i -e '/mount(\\.*'$sys_mnt'\"\", \\.*'/d' updater-
script
        fi
    fi

    if [ `grep -c "mount(\\.*$data_mnt\\", \\.*\\\"" updater-
script` -gt 0 ]
    then
        if [ `grep -c "auto \\$PARTITION \\$ANDROID_DATA" updater-
script` -gt 0 ]
        then
            sed -i -e '/mount(\\.*'$data_mnt'\"\", \\.*'/d' updater-
script
        fi
    fi
fi

```

```

mnt_list=( $data_mnt $sys_mnt $cache_mnt $boot_mnt
$sdcard_mnt )

for mnt in ${mnt_list[@]}
do
    if [ `grep -c "mount(\`.*$mnt\`, \`.*\`" updater-script`
-gt 0 ]
    then
        if [ `grep -c "mount $mnt" updater-script` -gt 0 ]
        then
            sed -i -e '/mount(".*"$mnt"', ".*"/d' updater-script
            fi
        fi
    done

#
# Alternative mount/unmount commands
#

use_sbin_mnt=0
use_sbin_bb=0

if [ -e updater-script.orig ]
then
    grep_sbin_mnt=`grep -c "run_program(\`/sbin/mount\`"
updater-script.orig`
    if [ $grep_sbin_mnt -gt 0 ]
    then
        use_sbin_mnt=1
    fi

    grep_sbin_bb=`grep -c "run_program(\`/sbin/busybox\`,
\`mount\`" updater-script.orig`
    if [ $grep_sbin_bb -gt 0 ]
    then
        use_sbin_bb=1
    fi

else

    if [ "$is_gs" == "1" ] || [ "$is_gs2" == "1" ] || [
"$is_gs2x" == "1" ]
    then
        use_sbin_mnt=1
    fi
fi

if [ $use_sbin_mnt == 1 ]
then
    sed -i \
    -e 's/^mount("[^"]*", "[^"]*", \(("[^"]*"\)
\("[^"]*"\)\/run_program("\`/sbin\/mount", \1, \2)/g' \
    -e
's/^unmount(\("[^"]*"\)\/run_program("\`/sbin\/umount", \1)/g' \

```

```

    updater-script
fi

if [ $use_sbin_bb == 1 ]
then
    sed -i \
        -e 's/^mount("[^"]*", "[^"]*", "[^"]*",
\("[^"]*"\\))/run_program("\\/sbin\\/busybox", "mount", \1)/g' \
        -e
's/^umount(\("[^"]*"\\))/run_program("\\/sbin\\/busybox", "umount",
\1)/g' \
    updater-script
fi

#
# MT65XX EMCC-based devices use "bootimg" rather than boot
#

if [ "$is_mt65xx_bootimg" == "1" ]
then
    sed -i \
        -e 's/boot.img\\", \\boot\\",/boot.img\\",
\\bootimg\\",/g' \
    updater-script
fi

#####
# change_mnt==yes; update-script
#####

elif [ "$1" == "update-script" ]
then
    sed -i \
        -e 's/format "$param1" "$param2" "$sys_mnt" 0/format
SYSTEM:/g' \
        -e 's/format "$param1" "$param2" "$sys_mnt"/format
SYSTEM:/g' \
        -e 's/format "$param1" "$param2"
"$preload_mnt"/format PRELOAD:/g' \
        -e 's/format "$param1_cache" "$param2_cache"
"$cache_mnt"/format CACHE:/g' \
        -e 's/format "$param1" "$param2" "$boot_mnt"/format
BOOT:/g' \
        -e 's/format "$param1" "$param2" "$data_mnt"/format
DATA:/g' \
        \
        -e 's/format "$param1" "$sys_mnt"/format "$param1"
SYSTEM:/g' \
        -e 's/format "$param1" "$preload_mnt"/format
"$param1" PRELOAD:/g' \
        -e 's/format "$param1_cache" "$cache_mnt"/format
"$param1_cache" CACHE:/g' \
        -e 's/format "$param1" "$boot_mnt"/format "$param1"
BOOT:/g' \

```

```

        -e 's/format "$param1" "$data_mnt"/format "$param1"
DATA:/g' \
    \
        -e 's/package_extract_file PACKAGE:boot.img
"$boot_mnt"/write_raw_image PACKAGE:boot.img BOOT:/g' \
        -e 's/run_program PACKAGE:sbin\busybox dd if=TMP:boot.img
of="$boot_mnt"/write_raw_image PACKAGE:boot.img BOOT:/g' \
        update-script

    sed -i \
        -e ':a;N;${ba;s/package_extract_file PACKAGE:boot.img
TMP:boot.img\nwrite_raw_image PACKAGE:boot.img BOOT:\ndelete
TMP:boot.img/write_raw_image PACKAGE:boot.img BOOT:/g' \
        update-script

    #
    # Cover the case where the ROM uses a different filesystem,
such that the update-script
    # wasn't adjusted in the above code.
    #
    # e.g. update-script uses ext4 in CyanogenMod instead of the
expected ext3
    #

    grep_other=`grep "format .* $sys_mnt[ ]*$" update-script`

    if [ "$grep_other" != "" ]
    then
        sed -i \
            -e 's/format .* "$sys_mnt"/format SYSTEM:/g' \
            -e 's/format .* "$preload_mnt"/format PRELOAD:/g' \
            -e 's/format .* "$cache_mnt"/format CACHE:/g' \
            -e 's/format .* "$boot_mnt"/format BOOT:/g' \
            -e 's/format .* "$data_mnt"/format DATA:/g' \
            \
            update-script
        fi
    fi

cd ../../../../../../..

#####
# All cases; update-script
#####

if [ "$1" == "update-script" ]
then

    #
    # Convert 'delete_recursive' of a partition to 'format', for
compatibility
    # with all of the scripts. This will be changed back when you
build ROM.
    #

```

```

    if [ "$sis_gs" == "1" ] || [ "$sis_gs2" == "1" ] || [ "$sis_gs2x"
== "1" ] \
    || [ "$sis_gs3_common" == "1" ] || [ "$sis_gs4_common" == "1"
]
]
then
    cd WORKING_*
    cd META-INF/com/google/android
    sed -i -e 's/delete_recursive \([^:]*:\)[ ]*$/format \1/g'
update-script
    cd ../../../../..
fi
fi

#####
# All cases; updater-script
#####

if [ "$1" == "updater-script" ]
then

    cd WORKING_*
    cd META-INF/com/google/android

    #
    # Fix cases where a mount hasn't been fixed yet
    #
    # e.g. If updater-script contains:
    #     mount("MTD", "userdata", "/data");
    #     But the other mount entries in file look like:
    #     mount("yaffs2", "MTD", ...);
    #
    grep_list=( yaffs2 ext3 )

    for str in ${grep_list[@]}
    do
        grep_result=`grep -c "^mount(\\"$str\"" updater-script`

        if [ $grep_result -gt 0 ]
        then
            sed -i \
                -e 's/^mount("MTD"/mount("\\"$str\"", "MTD"/g' \
                updater-script
            break
        fi
    done

    #
    # Galaxy S3 system partition fix
    #
    if [ "$sis_gs3_common" == "1" ] || [ "$sis_gs4_common" == "1" ]
    then
        sed -i \
            -e 's/mount("\\"$param1\"", "\\"$param2\"", "\\"$sys_mnt\"",
"\/system");//g' \
            -e

```

```

's/delete_recursive("\system");/unmount("\system");\nformat("'$
param1'", "'$param2'", "'$sys_mnt'",
"0");\nmount("'$param1'", "'$param2'", "'$sys_mnt'",
"\system");/g' \
    updater-script
    fi

    cd ../../../../..
fi

# Remove blank lines
cd WORKING_*
cd META-INF/com/google/android
sed -i -e '/^$/d' $1
cd ../../../../..

fi

```

Script apk_framework_install

```

#####
#####
#
# project MTK Android ROM Builder
# Copyright 2015 Blackant38 <ekoheriyanto8@gmail.com>
#
# MTK Android ROM Builder adalah projek open source yang di
khususkan
# hanya untuk pribadi & atau edukasi.dimohon untuk tidak
menjualbelikan
# atau menduplikat tanpa seizin pengembang.
#
#####
#####
clear

echo
echo "masukan file framework kedalam folder apktool"
echo
echo "ex:mediatk/framework-res.apk&medaitek-res.apk"
echo " :samsung/framework-res.apk&touchwiz-res.apk"
echo
echo "tekan enter untuk melanjutkan"
echo

read enterkey

clear
echo
echo "menginstall framework..."
echo

```

```

cd apktool/framework
if [ -e framework-res.apk ]
then
    echo "Install framewrok ..."
    apktool if framework-res.apk
fi
if [ -e framework-res.apk ]
then
    echo "Install framewrok ..."
    apktool if mediatek-res.apk
fi
exit 0

```

Script apktool

```

#####
#####
#
# project MTK Android ROM Builder
# Copyright 2015 Blackant38 <ekoheriyanto8@gmail.com>
#
# MTK Android ROM Builder adalah projek open source yang di
khususkan
# hanya untuk pribadi & atau edukasi.dimohon untuk tidak
menjualbelikan
# atau menduplikat tanpa seizin pengembang.
#
#####
#####

clear
echo
echo "=====
echo "                APKTOOL untuk Ubuntu                "
echo "=====
echo
echo "Pastikan file apktool sudah terinstall di sistem Ubuntu!"
echo
echo "jika belum teinstall silahkan install dengan cara berikut:"
echo "1) masuk folder tools/apktool "
echo "2) jalankan script apktool_installer "
echo
echo "tekan Enter untuk melanjutkan"
echo

read enterkey
clear

echo "masukan pilihan anda:"
echo
echo "f =install framework file(untuk apk system)"

```



```

echo "r =repack apk"                b =build apk"
echo "s =sign apk"                  x =keluar  "
echo
echo -n "masukan pilihan anda: "

read choose
if [ "$choose" == "x" ]
then
    exit 0
elif [ "$choose" == "r" ]
then
    scripts/decompile_apk
elif [ "$choose" == "b" ]
then
    scripts/build_apk
elif [ "$choose" == "f" ]
then
    scripts/apk_framework_install
else
    echo "pilihan yang anda masukan salah"
    continue
fi

```

Script app_file_to_working_folder

```

#####
#####
#
# project MTK Android ROM Builder
# Copyright 2015 Blackant38 <ekoheriyanto8@gmail.com>
#
# MTK Android ROM Builder adalah projek open source yang di
khususkan
# hanya untuk pribadi & atau edukasi.dimohon untuk tidak
menjualbelikan
# atau menduplikat tanpa seizin pengembang.
#
#####
#####
#
# argumen:
#
# $1 = nama file di original_update)
# $2 = "kernel_only" atau "name_given" - jika memilih ini
mengikuti $1
#

update_file=$1

echo "Selected $update file"

```

```

update_file2=`echo $update_file | tr ' ' '_`

if [ "$update_file" != "$update_file2" ]
then
    echo "Renaming to $update_file2"
    mv "original_update/$update_file" original_update/$update_file2
    update_file=$update_file2
fi

#
# membuat nama folder
#
build_dir=`scripts/set_working_folder_name $2`

#
#
#
echo
echo "Creating working folder $build_dir ..."
mkdir $build_dir

echo "Copying $update_file to $build_dir ..."
cp original_update/$update_file $build_dir
cd $build_dir

if [ "$update_file" != "UPDATA.APP" ]
then
    echo "Renaming to UPDATA.APP ..."
    mv -f $update_file UPDATA.APP
fi

echo
echo "Extracting system.img and boot.img using ZeBadger's script
..."
temp=`perl ../tools/huawei_files/split_updata.pl 2>/dev/null`

if [ ! -e output/system.img ]
then
    echo "Error: tidak ditemukan system.img "
fi

if [ ! -e output/boot.img ]
then
    echo "Error: tidak ditemukan boot.img"
fi

if [ ! -e output/system.img ] || [ ! -e output/boot.img ]
then
    cd ..
    exit 1
fi

mv -f output/boot.img .
mv -f output/system.img .
rm -rf output

```

```
rm -f UPDATA.APP

cd ..
scripts/img_files_to_working_folder no_create $2
exit $?
```

Script boot_img_tools

```
#####
#####
#
# project MTK Android ROM Builder
# Copyright 2015 Blackant38 <ekoheriyanto8@gmail.com>
#
# MTK Android ROM Builder adalah projek open source yang di
# khususkan
# hanya untuk pribadi & atau edukasi.dimohon untuk tidak
# menjualbelikan
# atau menduplikat tanpa seizin pengembang.
#
#####
#####

clear

while :
do

    clear

    echo
    echo "Boot image tools"
    echo "-----"
    echo
    echo "Masukan pilihan:"
    echo

    if [ -d WORKING_* ]
    then

        if [ -d BOOT-EXTRACTED ]
        then
            extracted=true
        else
            extracted=false
        fi

        cd WORKING_*
        if [ -e boot.img ]
        then
```

```

    boot_img=true
else
    boot_img=false
fi

if [ -d boot ]
then
    boot_folder=true
else
    boot_folder=false
fi

cd ..

if [ "$boot_img" == "true" ]
then
    echo "  s = lihat info boot.img"
fi

if [ "$extracted" == "true" ]
then
    echo "  b = Build boot.img di folder BOOT-EXTRACTED"
    echo "  n = Build folder boot NAND's di folder BOOT-
EXTRACTED"
    echo "  r = Hapus folder BOOT-EXTRACTED"
else

    if [ "$boot_img" == "true" ]
    then
        echo "  w = Extract kernel+ramdisk di boot.img"
        echo "  c = Convert boot.img ke folder boot NAND"
        echo "  p = ubah parameter 'command line' di boot.img"

    elif [ "$boot_folder" == "true" ]
    then
        echo "  y = Convert folder boot NAND ke boot.img"
        echo "  z = Extract kernel+ramdisk dari folder boot NAND"
    fi
fi

echo "  a = Extract kernel+ramdisk dari boot.img/recovery.img di
semua folder"
echo "  x = keluar"
echo
echo -n "? "

read enterLetter

if [ "$enterLetter" == "s" ]
then
    scripts/show_boot_img_info

elif [ "$enterLetter" == "w" ]
then

```

```
scripts/ensure_boot_extracted

elif [ "$enterLetter" == "r" ]
then
    rm -rf BOOT-EXTRACTED; echo; echo "Deleted"

elif [ "$enterLetter" == "b" ]
then
    scripts/prompt_build_boot
    scripts/build_boot_img

elif [ "$enterLetter" == "a" ]
then
    scripts/extract_boot_img

elif [ "$enterLetter" == "n" ]
then
    scripts/prompt_build_boot
    scripts/build_nand_boot

elif [ "$enterLetter" == "c" ]
then
    scripts/ensure_boot_extracted
    scripts/build_nand_boot

elif [ "$enterLetter" == "p" ]
then
    scripts/change_cmdline

elif [ "$enterLetter" == "y" ]
then
    scripts/ensure_nand_extracted
    scripts/build_boot_img

elif [ "$enterLetter" == "z" ]
then
    scripts/ensure_nand_extracted

elif [ "$enterLetter" == "x" ]
then
    exit 0
else
    echo "pilihan yang anda masukan salah"
    continue
fi

scripts/press_enter

done
```

Script build_apk

```
#####  
#####  
#  
# project MTK Android ROM Builder  
# Copyright 2015 Blackant38 <ekoheriyanto8@gmail.com>  
#  
# MTK Android ROM Builder adalah projek open source yang di  
# khususkan  
# hanya untuk pribadi & atau edukasi.dimohon untuk tidak  
# menjualbelikan  
# atau menduplikat tanpa seizin pengembang.  
#  
#####  
#####  
clear  
  
cd apktool/repack  
if [ -d apk_repack ]  
then  
    echo folder ditemukan  
else  
    echo tidak ditemukan folder!  
    exit 0  
fi  
apktool b apk_repack  
  
echo  
echo "tekan enter untuk melanjutkan"  
read enterkey  
clear  
echo  
echo "hasil build apk tedapat di folder apk_repack/dist"  
echo "silahkan sign apk melalui menu sign di apktool  "  
echo  
exit 0
```

Script build_boot_img

```
#####  
#####  
#  
# project MTK Android ROM Builder  
# Copyright 2015 Blackant38 <ekoheriyanto8@gmail.com>  
#  
# MTK Android ROM Builder adalah projek open source yang di  
# khususkan  
# hanya untuk pribadi & atau edukasi.dimohon untuk tidak  
# menjualbelikan  
# atau menduplikat tanpa seizin pengembang.
```

```

#
#####
#####

echo

if [ -d BOOT-EXTRACTED ]
then
    echo ditemukan folder BOOT-EXTRACTED , mengecek konten ...

    if [ -d BOOT-EXTRACTED/boot.img-ramdisk ]
    then
        echo Ditemukan boot.img-ramdisk

        if [ -e BOOT-EXTRACTED/zImage ]
        then
            echo Ditemukan zImage
        else
            echo Did not find BOOT-EXTRACTED/zImage
            exit 0
        fi

    else
        echo tidak dapat menemukan folder boot-extracted/boot.img-
ramdisk!
        exit 0
    fi

else
    echo tidak dapat menemukan folder BOOT-EXTRACTED!
    exit 0
fi

if [ -d WORKING_* ]
then
    echo folder ditemukan
else
    echo tidak ditemukan folder!
    exit 0
fi

if [ "$1" == "" ]
then

    cd WORKING_*

    if [ -e boot.img ]
    then

        cd ..
        scripts/check_kernel_offset
    fi
fi

```

```

res=$?

if [ "$res" != "0" ]
then
    exit 0
fi

else

if [ ! -e boot/initrd.gz ] || [ ! -e boot/zImage ]
then
    echo "boot.img tidak ditemukan di folderr!"
    cd ..
    exit 0
else

    base=0x11800000
    cmd_line="dsixda NAND"

    cd ..
    fi
fi

else

#
# Using hard-coded values
#

scripts/set_kernel_offset_files $1

if [ "$2" != "" ]
then
    base=$2

    if [ "$3" != "" ]
    then
        cmd_line=$3
    fi
fi

fi

#
# Check for MT65xx
#

cd WORKING_*
working_folder=`pwd`
cd ..
dec_offset=`scripts/get_boot_img_page_size $working_folder`
scripts/check_mt65xx_bootimg $working_folder $dec_offset
1>/dev/null
mt65xx=$?

```



```

if [ "$mt65xx" == "1" ]
then
    mkbootimg_src=mkbootimg_mt65xx.c
else
    mkbootimg_src=mkbootimg.c
fi
mkbootimg_out=mkbootimg

if [ `uname | grep CYGWIN` ]
then
    mkbootfs_file=mkbootfs.exe
    mkbootimg_file=$mkbootimg_out.exe
else
    mkbootfs_file=mkbootfs
    mkbootimg_file=$mkbootimg_out
fi

if [ -e tools/mkboot/$mkbootfs_file ]
then
    echo "Found $mkbootfs_file"
else
    echo
    echo "Compiling mkbootfs ..."
    cd tools/mkboot
    gcc -o mkbootfs mkbootfs.c 2>/dev/null
    cd ../..

    if [ -e tools/mkboot/$mkbootfs_file ]
    then
        echo mkbootfs sukses terkompile
    else
        echo "Error: mkbootfs gagal terkompile!"
        exit 0
    fi
fi

if [ -e tools/mkboot/$mkbootimg_file ]
then
    rm -f $mkbootimg_file
fi

echo
echo "Compiling mkbootimg ..."
cd tools/mkboot
gcc -c rsa.c
gcc -c sha.c
gcc rsa.o sha.o $mkbootimg_src -w -o $mkbootimg_out
rm *.o
cd ../..

if [ -e tools/mkboot/$mkbootimg_file ]

```

```

then
    echo "$mkbootimg_out berhasil terkompile"
else
    echo "Error: $mkbootimg_out tidak terkompile!"
    exit 0
fi

cp tools/mkboot/$mkbootfs_file BOOT-EXTRACTED/
cp tools/mkboot/$mkbootimg_file BOOT-EXTRACTED/
cd BOOT-EXTRACTED

echo
echo "membuat arsip ramdisk ..."
./$mkbootfs_file boot.img-ramdisk | gzip > ramdisk.gz

cd ..
cd WORKING_*

if [ -e boot.img ]
then
    cd ..
    echo

    size_orig=`scripts/get_boot_img_size`

    if [ "$base" == "" ]
    then
        echo "Attempting to determine kernel base address ..."
        base=`scripts/get_kernel_base_addr`
    fi

    echo "menggunakan base address dari $base"

    ramdisk_addr=`scripts/get_ramdisk_addr`

    echo "menggunakan ramdisk load address dari $ramdisk_addr"

    if [ "$3" == "" ] && [ "$1" == "" ]
    then
        echo "mencoba mendeteksi command line parameter ..."
        cmd_line=`scripts/get_cmdline`
    fi

    if [ "$cmd_line" == "" ]
    then

```

```

    echo "tidak ditemukan cmdline"
else
    echo "gunakan cmdline: $cmd_line"
fi

else
    cd ..
fi

#####
#
# Run mkbootimg to build new boot.img
#
#####

cd BOOT-EXTRACTED

echo
echo "Building new boot.img ..."

ramdisk_params=""
if [ "$ramdisk_addr" != "" ]
then
    ramdisk_params="--ramdiskaddr $ramdisk_addr"
fi

if [ "$cmd_line" == "" ]
then
    ./mkbootimg_file --kernel zImage --ramdisk ramdisk.gz -o
newBoot.img --base $base $ramdisk_params
else
    ./mkbootimg_file --kernel zImage --ramdisk ramdisk.gz --cmdline
"$cmd_line" -o newBoot.img --base $base $ramdisk_params
fi

if [ -e newBoot.img ]
then
    echo
    echo "newBoot.img terbuat"

    echo "pindah ke folder sebagai boot.img"
    cd ../WORKING_*
    mv -f ../BOOT-EXTRACTED/newBoot.img boot.img

    if [ "$size_orig" != "" ]
    then

        cd ..
        size_new=`scripts/get_boot_img_size`
        cd WORKING_*

        if [ "$size_new" != "" ]
        then
            echo

```

```

        echo "boot.img size"
        echo "Old: $size_orig"
        echo "New: $size_new"
    fi
fi

if [ -d boot ] && [ -e boot/initrd.gz ] && [ -e boot/zImage ]
then
    echo
    echo "menghapus NAND boot folder ..."
    rm -rf boot
fi

cd ..

echo
echo "menghapus BOOT-EXTRACTED folder ..."
rm -rf BOOT-EXTRACTED

exit_code=1

else
    echo
    echo "newBoot.img tidak terbuat!"

    rm $mkbootfs_file
    rm $mkbootimg_file

    cd ..
    exit_code=0
fi

exit $exit_code

```

Script build_rom

```

#####
#####
#
# project MTK Android ROM Builder
# Copyright 2015 Blackant38 <ekoheriyanto8@gmail.com>
#
# MTK Android ROM Builder adalah projek open source yang di
khususkan
# hanya untuk pribadi & atau edukasi.dimohon untuk tidak
menjualbelikan
# atau menduplikat tanpa seizin pengembang.
#
#####
#####

clear

```

```
echo

if [ -d WORKING_* ]
then
    echo ditemukan folder kerja
else
    echo tidak ditemukan folder kerja!
    exit 0
fi

cd WORKING_*

if [ -e META-INF/com/google/android/update-script ] && [ -e META-
INF/com/google/android/updater-script ]
then

    echo
    echo "Error: file update-script dan updater-script ditemukan di
folder kerja."
    echo "          Hapus salah satu untuk melanjutkan!"
    echo

    cd ..
    exit 0
fi

echo
echo "membersihkan file sampah ..."
echo

rm -f *.zip
rm -f testkey.*
rm -f signapk.jar
rm -f signed*.zip
rm -f *smali.jar

#
# hapus app cina
#
find . -name '*.DS_Store' -type f -delete

#
# hapus stackdump
#
find . -name sh.exe.stackdump -type f -delete

#
# hapus temp edif dev update script
#
if [ -d META-INF/com/google/android ]
then
    find META-INF/com/google/android/ -name 'sed*' -type f -delete
fi
```

```

#
# hapus temp file di root folder
#

file_list=`find . -maxdepth 1 -type f`

for filename in $file_list
do

    filename=`echo $filename | sed 's/\.\.\/g'`
    header_bytes=`od -A n -H -j 0 -N 4 $filename | sed 's/ //g'`

    if [ "$header_bytes" == "04034b50" ]
    then
        echo -n "Ditemukan file: $filename. Hapus (y/n)?: "
        read delete_file

        if [ "$delete_file" != "n" ]
        then
            echo "Menghapus $filename"
            rm -f $filename
        else
            echo "tidak dihapus $filename"
        fi
    fi
done

cd ..
scripts/prompt_nand_type
cd WORKING_*

echo
echo `pwd`
echo
ls -lrt
cd ..
echo

scripts/fix_pre_build

# Check null
cmd_line=`scripts/get_cmdline`

extreme=no
express=no
interactive=no

echo
echo
"=====
=====

```

```

echo
echo "pilih salah satu:"
echo
echo " 1 = Interactive Mode - recommended"
echo "      (guide semua build step by step)"
echo
echo " 2 = Lazy Mode - langsung build"
echo "      (Zipalign, Edify, sign ROM, auto-name ZIP)"
echo
echo " 3 = Express Mode - for advanced users"
echo "      (Zipalign, Edify, don't sign ROM, auto-name ZIP)"
echo
echo " 4 = Extreme Mode - for advanced users"
echo "      (No zipalign, no script conversion, don't sign ROM,
auto-name ZIP)"
echo
echo " 5 = Batal - jangan build"
echo
echo
"=====
=====
echo
echo -n "Number (default: 1): "

read build_mode

# Default is 1
if [ "$build_mode" == "5" ]
then
    echo "Dibatalkan"
    exit 0
elif [ "$build_mode" == "4" ]
then
    extreme=yes
elif [ "$build_mode" == "3" ]
then
    express=yes
elif [ "$build_mode" == "2" ]
then
    lazy=yes
    express=yes
else
    interactive=yes
fi

if [ "$extreme" == "no" ]
then
    scripts/do_zipalign for_build $express
fi

if [ "$?" == "1" ]
then
    echo "membatalkan build"
    exit 0
fi

```

```

cd WORKING_*

#
# Convert update-script (Amend) ke updater-script (Edify) untuk
file ZIP
#

if [ "$extreme" == "no" ] && [ -e META-
INF/com/google/android/update-script ]
then

    cd ..
    scripts/update_script_should_convert_back
    auto_convert="$?"
    cd WORKING_*

    if [ "$auto_convert" == "1" ] || [ "$express" == "yes" ]
    then
        convert_it=y
        want_file=updater-script

    else
        echo
        echo "-----"
        echo
        echo " update-script Terdeteksi di ROM"
        echo
        echo " custom recovery terbaru merekomendasikan untuk
mengganti"
        echo " updater-script (Edify) dengan update-script (Amend) di
ROM"
        echo
        echo -n " Gunakan updater-script di ROM(y/n)?: "
        read convert_it
    fi

    if [ "$convert_it" != "n" ]
    then

        # Remove installbusybox
        if [ -e installbusybox ]
        then
            cd ..
            scripts/add_busybox_to_update_script
            cd WORKING_*
        fi

        cd ..
        scripts/convert_update_script for_zip $express

        # ganti updater-script
        if [ "$?" != "1" ]

```



```

then

    cd WORKING_*

    if [ "$express" == "yes" ]
    then
        proceed_change=y
    else
        echo "Ganti (y/n)?"
        echo
        echo "  y = Ganti; updater-script dan update-binary akan
di gunakan di"
        echo "      file zip ROM; update-script berada di folder
kerja."
        echo "  n = Batal; updater-script dan update-binary akan
di hapus."
        echo

        echo -n "? (default: y): "
        read proceed_change
    fi

    echo

    if [ "$proceed_change" != "n" ]
    then

        if [ -e META-INF/com/google/android/updater-script ]
        then
            ../scripts/convert_to_unix META-
INF/com/google/android/updater-script
            want_file=updater-script
        else
            echo "tidak ditemukan updater-script, gunakan update-
script"
            want_file=update-script
        fi

        else
            echo
            echo "Membatalkan - akan menggunakan update-script di file
ZIP"
            want_file=update-script
            rm -fv META-INF/com/google/android/updater-script
            rm -fv META-INF/com/google/android/update-binary
            echo
        fi

        if [ "$auto_convert" == "1" ] && [ "$proceed_change" == "n"
]
    then
        echo "Kamu tidak bisa flash ROM tanpa updater-script."
        echo
        cd ..
        exit 0
    fi

```

```

        fi

        else
            echo "Gunakan update-script"
            cd WORKING_*
            want_file=update-script
        fi

        else
            echo
            echo
            echo "Gunakan update-script"
            want_file=update-script
        fi

        echo

else

        echo

        if [ -e META-INF/com/google/android/update-script ]
        then
            echo "ditemukan update-script (Amend)"
            want_file=update-script
        fi

        if [ -e META-INF/com/google/android/updater-script ]
        then
            echo "ditemukan updater-script (Edify)"
            want_file=updater-script
        fi

        echo
fi

if [ "$want_file" == "updater-script" ]
then
    script_not_want=*update-script*
    binary_not_want=""
else

    # Hapus run_program yang merefer ke busybox
    # diganti dg script installbusybox.
    if [ -e system/xbin/busybox ]
    then
        cd ..
        scripts/add_busybox_to_update_script amend
        cd WORKING_*
    fi

```

```

    script_not_want=*updater-script*
    binary_not_want=*update-binary*
fi

if [ "$cmd_line" == "dsixda Null" ]
then
    boot_not_want=boot.img
else
    boot_not_want=""
fi

#
# buat update.zip
#
echo
echo Membuat update.zip ...

zip -r -y -q update * -x *.cvs* *.git* *.svn* \
    *updater-script.orig* *update-binary.orig* *update-
script.orig* \
    $script_not_want $binary_not_want $boot_not_want

echo

# Hapus updater-script di folder kerja
cd META-INF/com/google/android
if [ -e update-script ]
then
    rm -f updater-script
    rm -f update-binary
fi

cd ../../../../..

if [ -e update.zip ]
then
    echo update.zip created
    echo
else
    echo "Error: update.zip not created!"
    cd ..
    exit 0
fi

#
# Sign update.zip
#

date_str=`date '+%m%d%y_%H%M%S'`

cd ..

```

```

device=`scripts/get_device_name | sed 's/ /_/g'`
cd WORKING_*

sign_update=no
sign_str=unsigned

if [ "$lazy" == "yes" ]
then
do_sign=y
else
do_sign=n
fi

if [ "$interactive" == "yes" ]
then
echo
echo "kami rekomendasikan untuk mensign ROM."
echo -n "Sign (y/n)?: "
read do_sign
fi

if [ "$do_sign" != "n" ]
then
sign_update=yes
sign_str=signed
fi

final_file=$device\_sign_str\_date_str.zip

if [ "$sign_update" == "no" ]
then
mv update.zip $final_file
res=$?

else
echo
echo Signing update.zip ...

cp ../tools/signapk_files/testkey.* .
cp ../tools/signapk_files/signapk.jar .

java -jar signapk.jar testkey.x509.pem testkey.pk8 update.zip
$final_file
res=$?

rm -f testkey.*
rm -f signapk.jar
rm -f update.zip
fi

if [ -e $final_file ] && [ "$res" == "0" ]
then

```

```

#
# ganti nama zip
#
if [ "$interactive" == "yes" ]
then
  cd ..
  final_file2=`scripts/set_update_name $final_file`
  cd WORKING_*
  if [ "$final_file2" != "$final_file" ]
  then
    mv -fv $final_file $final_file2
    final_file=$final_file2
  fi
fi

echo

#
# buat output folder
#
if [ -d ../OUTPUT_ZIP ]
then
  echo "ditemukan folder OUTPUT_ZIP"
else
  echo "Membuat folder OUTPUT_ZIP ..."
  mkdir ../OUTPUT_ZIP
fi

mv $final_file ../OUTPUT_ZIP/

echo
echo
echo
echo

"=====
=====
echo
echo "Selamat ... ROM berada di :"
echo
echo "----> OUTPUT_ZIP/$final_file"
echo
echo "Copy ke HH kemudian flash dengan custom recovery."
echo
echo

"=====
=====
echo
echo "NOTE 1: Jangan lupa triple wipe (factory reset, data/cache
wipe)"

if [ "$sign_update" == "no" ]
then
  echo
  echo "NOTE: Kamu harus mendisable signature checks di
recovery"
  echo "menu untuk flash this ROM!"

```

```

fi

echo
echo "Good luck!"

else

echo
echo "Tidak bisa membuat $final_file!"
echo

if [ "$sign_update" == "yes" ]
then
if [ `uname | grep Linux` ]
then
echo "Pastikan sudah menginstall Sun Java JDK"

elif [ `uname | grep CYGWIN` ]
then
echo "Pastikan kamu sudah menginstall Sun Java JDK di PC."
fi
fi
fi

cd ..

# Ramfung
echo

```

Script call_extract_kernel_and_modules

```

#####
#####
#
# project MTK Android ROM Builder
# Copyright 2015 Blackant38 <ekoheriyanto8@gmail.com>
#
# MTK Android ROM Builder adalah projek open source yang di
khususkan
# hanya untuk pribadi & atau edukasi.dimohon untuk tidak
menjualbelikan
# atau menduplikat tanpa seizin pengembang.
#
#####
#####

clear

echo
echo "Prosedur berikut akan mengekstrak kernel dan modul dari
ROM."

```

```

echo
echo "NOTE1: Folder MY_KERNEL yang sudah ada akan dihapus."
echo
echo "NOTE2: Jika folder kerja sudah ada, akan dihapus, dan akan
di backup dan"
echo "          di restore setelah kernel di ekstrak."
echo
echo "copy ROM baru, di folder 'original_update'."

scripts/press_enter

if [ -d WORKINGTEMP ]
then
    rm -rf WORKINGTEMP
fi

if [ -d WORKING_* ]
then
    mkdir WORKINGTEMP
    mv WORKING_* WORKINGTEMP 2>/dev/null

    if [ "$?" == "1" ]
    then
        echo "Error: folder kerja sedang digunakan. Pastikan
file/folder tidak ada yang terbuka."
        rm -rf WORKINGTEMP
        exit 0
    fi

fi

scripts/extract_kernel_and_modules

if [ -d MY_KERNEL ]
then

    if [ -d MY_DEVICE_WORKING/system ]
    then

        if [ `ls MY_DEVICE_WORKING/system | grep -m 1 prop` ]
        then

            echo
            echo "Mencopy *.prop ..."

            if [ ! -d MY_KERNEL/system ]
            then
                mkdir MY_KERNEL/system
            fi

            cp -v MY_DEVICE_WORKING/system/*.prop MY_KERNEL/system/
        fi

    fi

fi

```

```

rm MY_KERNEL/boot.img

echo
echo "-----"
-----"

echo
echo "files/folders berikut akan di copy ke folder MY_KERNEL:"
echo
ls MY_KERNEL

if [ -d MY_KERNEL/system ]
then
    echo
    echo "Di dalam folder /system:"
    ls MY_KERNEL/system
fi

echo
echo "-----"
-----"

echo
echo "kamu bisa menambahkan zImage ke folder BOOT-EXTRACTED,
jika kamu"
echo "sudah mengunpack boot.img di folder kerja.."
echo

else
    echo "Error: Kernel tidak terekstrak"
fi

if [ -d MY_DEVICE_WORKING ]
then
    echo "Menghapus folder kaerja ROM ini ..."
    rm -rf MY_DEVICE_WORKING
fi

if [ -d WORKINGTEMP ]
then
    working_folder=`ls WORKINGTEMP`
    echo "Merestore folder kerja sebelumnya $working_folder ..."
    mv WORKINGTEMP/* .
    rmdir WORKINGTEMP
fi

```

Script check_binaries

```

#####
#####
#
# project MTK Android ROM Builder
# Copyright 2015 Blackant38 <ekoheriyanto8@gmail.com>

```



```

#
# MTK Android ROM Builder adalah proyek open source yang di
# khususkan
# hanya untuk pribadi & atau edukasi.dimohon untuk tidak
# menjualbelikan
# atau menduplikat tanpa seizin pengembang.
#
#####
#####

bin_list=( clear gcc java perl sed wget od wc cpio zip unzip )
arg_list=( foo dumpversion version v -version V -version -version
-version -help -help )

show_header=1

for (( i = 0 ; i < ${#bin_list[@]} ; i++ ))
do

    if [ $show_header == 1 ]
    then
        clear
        echo
        echo
        echo
        echo
        echo "Sedang memuat"
        echo
        echo -n "Mohon tunggu... "
        for (( count = 0 ; $count <= $i ; count++ ))
        do
            echo -n "."
        done

        show_header=0

    else
        echo -n "."
    fi

    found_error=0

    bin=${bin_list[$i]}
    argmt=${arg_list[$i]}

    res=`$bin -$argmt 2>/dev/null`
    result=$?

    if [ "$result" == "127" ]
    then
        echo
        echo
        echo "Error: software '$bin' belum terinstall. "
        echo
        echo
    fi
done

```

```

    if [ "$bin" == "clear" ]
    then
        echo "          Mohon hubungi developer untuk informasi lebih
lanjut ;)"
        echo
    fi

    found_error=1

elif [ "$bin" == "sed" ] || [ "$bin" == "od" ]
then

    grep_cmd=`$bin --version | grep -m 1 GNU`

    if [ "$grep_cmd" == "" ]
    then
        echo
        echo
        echo "Error:  Anda belum menginstall versi '$bin' yang
benar."
        echo "          install '$bin' atau hubungi developer untuk
informasi lebih lanjut "
        echo

        found_error=1

    fi
fi

if [ $found_error == 1 ]
then
    #echo -n "lanjutkan (y/n)? (default: n): "
    #read proceedCheck
    proceedCheck=n

    if [ "$proceedCheck" != "y" ]
    then
        exit 1
    else
        show_header=1
    fi
fi

done

exit 0

```

Script check_install_dir

```
#####  
#####  
#  
# project MTK Android ROM Builder  
# Copyright 2015 Blackant38 <ekoheriyanto8@gmail.com>  
#  
# MTK Android ROM Builder adalah projek open source yang di  
# khususkan  
# hanya untuk pribadi & atau edukasi.dimohon untuk tidak  
# menjualbelikan  
# atau menduplikat tanpa seizin pengembang.  
#  
#####  
#####  
  
install_dir=`pwd`  
grep_result=`echo "$install_dir" | grep ".* [ ]*"`  
  
if [ "$grep_result" != "" ]  
then  
    echo  
    echo "Error: folder dengan nama \"$install_dir\" mengandung  
    karakter <space>."  
    echo "        mohon untuk memindah file ke folder tanpa karakter  
    <space>."  
    echo "        agar MARB dapat berjalan normal."  
  
    example_dir=`echo $install_dir | sed -e 's/\([^ ]*\) .*/\1//g'  
-e 's/\(.*)\|\.*/\1/g' -e 's/\|/|/g'`  
    echo  
    echo "        For example: $example_dir/marb"  
    echo  
  
    echo  
    exit 1  
else  
    exit 0  
fi
```

Script check_mt65xx_booting

```
#####  
#####  
#  
# project MTK Android ROM Builder  
# Copyright 2015 Blackant38 <ekoheriyanto8@gmail.com>  
#
```

```

# MTK Android ROM Builder adalah projek open source yang di
khususkan
# hanya untuk pribadi & atau edukasi.dimohon untuk tidak
menjualbelikan
# atau menduplikat tanpa seizin pengembang.
#
#####
#####

working_folder=$1
dec_offset=$2

if [ "$working_folder" == "" ]
then
    cd WORKING_*
    working_folder=`pwd`
    cd ..

    if [ "$dec_offset" == "" ]
    then
        dec_offset=`scripts/get_boot_img_page_size $working_folder`
    fi
fi

base_dir=`pwd`

# Check for MT65XX
cd $working_folder

if [ ! -e boot.img ]
then
    cd $base_dir
    echo "Error: tidak ditemukan boot.img"
    exit 0
fi

# Location of "KERNEL" string
str_offset=$(( $dec_offset + 8 ))
if [ "`od -A n -h -j $str_offset -N 6 boot.img | sed 's/ //g'`" \
    == "454b4e524c45" ]
then
    echo
    echo "ditemukan MT65xx kernel header"
    cd $base_dir
    exit 1
fi

cd $base_dir
exit 0

```

Script check_multiple_working

```
#####  
#####  
#  
# project MTK Android ROM Builder  
# Copyright 2015 Blackant38 <ekoheriyanto8@gmail.com>  
#  
# MTK Android ROM Builder adalah proyek open source yang di  
# khususkan  
# hanya untuk pribadi & atau edukasi.dimohon untuk tidak  
# menjualbelikan  
# atau menduplikat tanpa seizin pengembang.  
#  
#####  
#####  
  
num_working=`ls | grep -c "^WORKING_"`  
if [ $num_working -gt 1 ]  
then  
    echo  
    echo "Error: ada lebih dari satu folder 'working'"  
    exit 1  
fi  
  
exit 0
```

Script check_rom

```
#####  
#####  
#  
# project MTK Android ROM Builder  
# Copyright 2015 Blackant38 <ekoheriyanto8@gmail.com>  
#  
# MTK Android ROM Builder adalah proyek open source yang di  
# khususkan  
# hanya untuk pribadi & atau edukasi.dimohon untuk tidak  
# menjualbelikan  
# atau menduplikat tanpa seizin pengembang.  
#  
#####  
#####  
  
#  
# perintah script:  
#  
# $1 = "no_info"      - ekstrak kernel atau keperluan porting  
# $2 = "kernel_only" - pilihan lanjutan tetapi tetap menggunakan  
# $1
```

```

#
echo
echo "MEMBUAT FOLDER KERJA"
echo "======"
echo
echo "Pastikan terdapat minimal 1 ROM di folder
'original_update'!"
echo
echo "Masukan pilihan:"
echo
echo " 1 - Format yang di support"
echo " 2 - Batal"
echo ""
echo "Atau, tekan Enter untuk melanjutkan"
echo
echo -n "==>>? "
read proceed

if [ "$proceed" == "2" ]
then
    echo
    echo "Batal membuat folder kerja"
    exit 1

elif [ "$proceed" == "1" ]
then

    more tools/formats.txt

    echo
    echo -n "Tekan Enter untuk melanjutkan, atau ketik 'x' untuk
keluar: "
    read proceed

    if [ "$proceed" == "x" ]
    then
        exit 1
    fi

fi

scripts/make_backup_working

#
# cek banckup folder Working_
#
result=$?

if [ "$result" == "1" ]
then

    echo
    echo "Error: Folder kerja sedang digunakan. pastikan file tidak
di buka oleh aplikasi apapun."
    scripts/press_enter

```

```

exit 1

else

scripts/choose_rom $2
res=$?

if [ "$res" == "1" ] && [ "$2" == "kernel_only" ]
then
scripts/press_enter
fi

if [ `ls | grep -m 1 WORKING_` ] && [ "$res" != "1" ] && [ "$1"
== "" ]
then

#
# cek radio.img
#

echo
cd WORKING*
if [ -e radio.img ]
then
echo
echo "NOTE: radio.img ditemukan di folder kerja, hapus file
tersebut"
echo "bisa di hapus nanti sebelum membuild ROM."
else

if [ -e META-INF/com/google/android/update-script ]
then
cd ..
scripts/update_script_should_convert_back ignore_msg
res=$?
cd WORKING_*
fi
fi

if [ -e boot.img ]
then
boot_found=yes
elif [ -e boot/zImage ] && [ -e boot/initrd.gz ]
then
boot_found=yes
else
boot_found=no
fi

if [ "$boot_found"=="yes" ]
then

#
# prompt rom info
#

```

```

        if [ "$1" == "" ]
        then
            cd ..
            scripts/prompt_show_rom_info
        else
            cd ..
        fi

        else
            echo
            echo "BOOT.img tidak ditemukan!"
            cd ..
        fi
    fi
fi

if [ "$1" == "" ]
then
    scripts/press_enter
fi

exit $res

```

Script choose_rom

```

#####
#####
#
# project MTK Android ROM Builder
# Copyright 2015 Blackant38 <ekoheriyanto8@gmail.com>
#
# MTK Android ROM Builder adalah projek open source yang di
# khususkan
# hanya untuk pribadi & atau edukasi.dimohon untuk tidak
# menjualbelikan
# atau menduplikat tanpa seizin pengembang.
#
#####
#####
#
# fungsi scrip:
#
# $1 = "kernel_only" - jangan edit apapun setelah extraxt
#   = "name_given" - ganti nama, tetap menggunakan fungsi $1
# $2 = ROM di filder original_update (di tentukan dari if $1 =
# "name_given")
#

if [ "$1" == "name_given" ]
then

```



```

file_chosen=$2

if [ "$file_chosen" == "" ]
then
    echo "Error: Nama ROM tidak spesifik!"
    cd ..
    exit 0
fi

if [ ! -e original_update/$file_chosen ]
then
    echo "Error: tidak ditemukan original_update/$file_chosen!"
    exit 0
fi

else

while [ 1==1 ]
do

    echo
    echo "Mohon tunggu ..."
    echo

    #
    # Cari original_update
    #
    grep_cmd=`scripts/grep_roms`

    if [ "$grep_cmd" == "" ]
    then
        echo "Error: tidak di temukan ROM di folder
Original_update!"
        exit 0
    fi

    count=0
    rm -f temp.list

    echo >> temp.list
    echo "Available ROMs:" >> temp.list
    echo >> temp.list

    for filename in $grep_cmd
    do
        count=$((count+1))

        filename=`echo $filename | sed 's/temp_space/ /g'`

        # Nama file array
        file_array[$count]=$filename

        if [ "$filename" == "system.img" ]
        then
            filename="system.img and boot.img"

```

```

elif [ "$filename" == "factoryfs.rfs" ]
then
    filename="factoryfs.rfs, cache.rfs and zImage"

elif [ "$filename" == "system.rfs" ]
then
    filename="system.rfs, csc.rfs and boot.img"

elif [ "$filename" == "system.img.ext4" ]
then
    filename="system.img.ext4, tomb.img.ext4, cache.img.ext4
and boot.img"

elif [ "$filename" == "system.ext4.tar" ]
then
    filename="system.ext4.tar and boot.img"

elif [ "$filename" == "factoryfs.img" ]
then
    filename="factoryfs.img, hidden.img, cache.img and zImage"

elif [ "$filename" == "PDA.tar.md5" ]
then
    filename="PDA.tar.md5, CSC.tar.md5 and PHONE.tar.md5"

elif [ "$filename" == "PDA.tar" ]
then
    filename="PDA.tar, CSC.tar and PHONE.tar"
fi

echo " ($count) $filename" >> temp.list
done

more temp.list
rm -f temp.list

echo
echo -n "Masukan pilihan nomor (default=1, batal=0,
r=refresh): "

read enterNumber
echo

if [ "$enterNumber" == "0" ]
then
    exit 0
fi

if [ "$enterNumber" == "" ]
then
    enterNumber=1
fi

if [ "$enterNumber" == "r" ]
then

```

```

        continue
    fi

    if [ "`echo $enterNumber | sed 's/[0-9]*//'\`" == "" ] || [
"enterNumber"=="1" ]
    then
        file_chosen=${file_array[$enterNumber]}

        if [ "$file_chosen" == "" ]
        then
            echo "Error: Pilihan salah"
            continue
        else
            break
        fi
    else
        echo "Error: Pilihan salah"
        continue
    fi

done

fi

if [ ! -d "original_update/$file_chosen" ]
then

    if [ "$file_chosen" == "system.img" ]
    then
        scripts/img_files_to_working_folder create_working $1
        exit $?

    elif [ "$file_chosen" == "factoryfs.rfs" ] || [ "$file_chosen"
== "system.rfs" ]
    then
        scripts/rfs_files_to_working_folder create_working $1
        exit $?
    elif [ "$file_chosen" == "system.ext4.tar" ]
    then
        scripts/sext4_tar_files_to_working_folder create_working $1
        exit $?

    elif [ "`echo $file_chosen | grep -i \\.zip\$`" != "" ]
    then
        scripts/zip_file_to_working_folder "$file_chosen" $1
        exit $?

    elif [ "`echo $file_chosen | grep -i \\.tar\$`" != "" ]
    then
        scripts/tar_file_to_working_folder "$file_chosen" $1
        exit $?

    elif [ "`echo $file_chosen | grep -i \\.app\$`" != "" ]
    then
        scripts/app_file_to_working_folder "$file_chosen" $1

```

```

        exit $?
    fi

else
    if [ "`echo $file_chosen | grep -m 1 \"^WORKING_\"`" != "" ]
    then
        scripts/wkg_to_working_folder "$file_chosen" $1
        exit $?
    fi
fi
fi

```

Script convert_update_script

```

#####
#####
#
# project MTK Android ROM Builder
# Copyright 2015 Blackant38 <ekoheriyanto8@gmail.com>
#
# MTK Android ROM Builder adalah projek open source yang di
khususkan
# hanya untuk pribadi & atau edukasi.dimohon untuk tidak
menjualbelikan
# atau menduplikat tanpa seizin pengembang.
#
#####
#####

if [ "$1" == "" ]
then
    clear
fi

echo

if [ -d WORKING_* ]
then

    cd WORKING_*
    cd META-INF/com/google/android

else
    echo
    echo folder tidak ditemukan, buat baru!
    exit 0
fi

if [ "$1" == "" ]

```

```

then

    echo "Conversion Tool: update-script (Amend) ke updater-script
(Edify)"
    echo "-----"
    -----"
    echo -n "convert (y/n)?: "

    read proceed_now
    echo

    if [ "$proceed_now" == "n" ]
    then
        cd ../../../../...
        exit 0
    fi

    echo "-----"
    -----"
    echo

fi

if [ -e update-script ]
then

    if [ -e updater-script ]
    then

        if [ "$2" == "yes" ]
        then
            remove_it=y
        else
            echo "Error: terdapat update-script dan updater-script"
            echo "      di folder. Hapus salah satu secara manual."
            echo

            echo -n "Hapus updater-script dan lanjutkan conversion (y/n)?
"
            read remove_it
            echo
            fi

            if [ "$remove_it" != "n" ]
            then
                rm -vf updater-script
                echo
            else
                cd ../../../../...
                exit 0
            fi
        fi

    else
        echo "Error: tidak ditemukan update-script!"
        cd ../../../../...

```

```

    exit 0
fi

cd ../../../../../../..

if [ "$1" != "for_zip" ]
then
    echo "cek modifikasi di update-script sebelum di convert ..."
    scripts/fix_pre_build
fi

cd WORKING_*
cd META-INF/com/google/android

echo
echo "-----"
echo
echo "Menconfert ke updater-script, tunggu sebentar ..."
echo

cp update-script updater-script

orig_list=( DATA SYSTEM PRELOAD CACHE SDEXT BOOT SDCARD )
new_list=( userdata system preload cache sdext boot sdcard )
new_list_str="0 1 2 3 4 5 6"

echo "- Initial formatting ..."

if [ "$s2" == "1" ] || [ "$s2x" == "1" ] || [ "$s3_common" == "1" ] || [ "$s4_common" == "1" ]
then
    sed -i -e 's/format /delete_recursive /g' updater-script
fi

sed -i \
-e ':a;N;$!ba;s/,\\n//g' \
-e 's/assert[^\n]*[\\n]*//g' \
updater-script

sed -i -e 's/^[ \t]*//;s/[ \t]*$//g' updater-script
sed -i -e '/^#/d' updater-script

img_list=( boot radio bootloader )

```

```

for (( i = 0 ; i < ${#img_list[@]} ; i++ ))
do
img_name=${img_list[$i]}
str="write_raw_image PACKAGE:$img_name.img"
line_num=`sed -n -e '/'"$str"'/{=;q}' updater-script`

if [ "$line_num" != "" ]
then
str2="package_extract_file PACKAGE:$img_name.img
TMP:$img_name.img"
str3="delete TMP:$img_name.img"
line_num3=`sed -n -e '/'"$str3"'/{=;q}' updater-script`

if [ "$line_num3" != "" ]
then
line_num_temp2=$(( $line_num3 - 2 ))
str_temp=`sed -n -e '/'"$line_num_temp2"'p' updater-script`

if [ "$str_temp" == "$str2" ] && [ $line_num3 -lt $line_num
]
then
sed -i -e '/'"$line_num"'d' updater-script
continue
fi
fi
fi

sed -i -e
's/'"$str"'.*$/assert(package_extract_file("'"$img_name"'.img",
"\tmp\/'"$img_name"'.img"),\n
write_raw_image("\tmp\/'"$img_name"'.img", "'"$img_name"'),\n
delete("\tmp\/'"$img_name"'.img"));/g' updater-script

done
if [ `grep -c write_radio_image updater-script` -gt 0 ]
then
echo "- maaf tidak bisa 'write_radio_image' - tidak bisa di
convert ..."
sed -i -e 's/write_radio_image PACKAGE:radio.img//g' updater-
script
fi

#
# hboot.img
#
if [ `grep -c write_hboot_image updater-script` -gt 0 ]
then
echo "- maaf tidak bisa 'write_hboot_image' - tidak bisa di
convert ..."
sed -i -e 's/write_hboot_image PACKAGE:hboot.img//g' updater-
script
fi

echo "- copy_dir ..."

```

```

sed -i \
-e 's/copy_dir \(.*\) /package_extract_dir(\\"1\");/g' \
-e 's/\(package_extract_dir.*\) \(.*\) /1", "2/g' \
-e 's/\(package_extract_dir.*\) PACKAGE:\/1/g' \
-e
's/\(package_extract_dir.*\) SYSTEM:\([^]*\) /1/system\/2/g' \
-e
's/\(package_extract_dir.*\) PRELOAD:\([^]*\) /1/preload\/2/g' \
\
-e 's/\(package_extract_dir.*\) DATA:\([^]*\) /1/data\/2/g' \
\
-e 's/\(package_extract_dir.*\) TMP:\([^]*\) /1/tmp\/2/g' \
-e
's/\(package_extract_dir.*\) SDCARD:\([^]*\) /1/sdcard\/2/g' \
-e 's/\(package_extract_dir.*\) BOOT:\([^]*\) /1/boot\/2/g' \
\
-e
's/\(package_extract_dir.*\) CACHE:\([^]*\) /1/cache\/2/g' \
updater-script

echo "- set_perm_recursive ..."

sed -i \
-e 's/set_perm_recursive \(.*\) /set_perm_recursive(\1);/g' \
-e 's/\(set_perm_recursive.*[0-9]*\) \([0-9]*\) \([0-9]*\) \([0-9]*\) /1, 2, 3, 4, /g' \
-e 's/\(set_perm_recursive.*\) SYSTEM:) /1"\/system" /g' \
-e
's/\(set_perm_recursive.*\) SYSTEM:\([^]*\) /1"\/system\/2" /g' \
' \
-e
's/\(set_perm_recursive.*\) DATA:\([^]*\) /1"\/data\/2" /g' \
-e 's/\(set_perm_recursive.*\) TMP:\([^]*\) /1"\/tmp\/2" /g' \
\
-e
's/\(set_perm_recursive.*\) CACHE:\([^]*\) /1"\/cache\/2" /g' \
\
-e 's/\(set_perm_recursive.*\) \\/(.*)\([^]*\) /1\
"\/2\3" /g' \
updater-script

echo "- set_perm ..."

sed -i \
-e 's/set_perm \(.*\) /set_perm(\1);/g' \
-e 's/\(set_perm.*[0-9]*\) \([0-9]*\) \([0-9]*\) /1, 2, 3, /g' \
-e 's/\(set_perm.*\) SYSTEM:\([^]*\) /1"\/system\/2" /g' \
-e 's/\(set_perm.*\) DATA:\([^]*\) /1"\/data\/2" /g' \
-e 's/\(set_perm.*\) TMP:\([^]*\) /1"\/tmp\/2" /g' \
-e 's/\(set_perm.*\) CACHE:\([^]*\) /1"\/cache\/2" /g' \
-e 's/\(set_perm.*\) \\/(.*)\([^]*\) /1\
"\/2\3" /g' \
updater-script

echo "- run_program ..."

```



```

line_num=`sed -n -e '/run_program /{=;q}' updater-script`

while [ "$line_num" != "" ]
do
    sed -i \
        -e '$line_num' s/run_program \(.*\) /run_program(\\"1\");/g' \
        -e '$line_num' s/ /", "/g' \
        -e '$line_num' s/PACKAGE:\/\/g' \
        -e '$line_num' s/TMP:\/\/tmp\/\/g' \
        -e '$line_num' s/BOOT:\/\/boot\/\/g' \
        -e '$line_num' s/SYSTEM:\/\/system\/\/g' \
        -e '$line_num' s/DATA:\/\/data\/\/g' \
        updater-script

    sed -i \
        -e '$line_num'
s/run_program(\"\/\([^\^\/]*\)\" \(.*\) /package_extract_file(\"1\",
\"\/tmp\/1\");\nrun_program(\"\/tmp\/1\"2/g' \
        updater-script

    prog_name=`sed -n -e '$line_num'
s/package_extract_file(\"^\", [ ]*\"([^\]*\)\" );/1/p' updater-
script`

    if [ "$prog_name" != "" ]
    then
        sed -i \
            -e '$line_num' s/package_extract_file(\"([^\]*\)\", [
] *\"([^\]*\)\" );/package_extract_file(\"1\", \"2\");\nset_perm(0, 0,
0777, \"2\");/g' \
            updater-script
        fi

        line_num=`sed -n -e '/run_program /{=;q}' updater-script`
    done

echo "- package_extract_file ..."

sed -i \
    -e 's/package_extract_file PACKAGE:/package_extract_file /g' \
    -e 's/package_extract_file
\(.*\)/package_extract_file(\\"1\");/g' \
    -e 's/(package_extract_file[^\,]*\) \(.*)/1", "2/g' \
    -e
's/(package_extract_file.*) (\\"SYSTEM:([^\]*\)\" )/1(\\"system\/2)
/g' \
    -e
's/(package_extract_file.*) (\\"DATA:([^\]*\)\" )/1(\\"data\/2)/g'
\
    -e
's/(package_extract_file.*) (\\"TMP:([^\]*\)\" )/1(\\"tmp\/2)/g' \
    -e
's/(package_extract_file.*) SYSTEM:([^\]*\)\" )/1/system\/2)/g'
\
    -e 's/(package_extract_file.*) DATA:([^\]*\)\" )/1/data\/2)/g'
\

```

```

-e 's/(package_extract_file.*\)TMP:\([^)]*\))/\1/tmp/\2/g' \
-e
's/(package_extract_file.*\)CACHE:\([^)]*\))/\1/cache/\2/g' \
  updater-script

echo "- show_progress ..."

sed -i \
  -e 's/show_progress \(.*)/show_progress(\1);/g' \
  -e 's/(show_progress.*\) \(.*)/\1, \2/g' \
  updater-script

echo "- set_progress ..."

sed -i \
  -e 's/set_progress \(.*)/set_progress(\1);/g' \
  updater-script

echo "- sleep ..."

sed -i \
  -e 's/sleep \(.*)/sleep(\1);/g' \
  updater-script

echo "- delete/delete_recursive ..."

sed -i \
  -e 's/delete \(.*)/delete("\1");/g' \
  -e 's/delete_recursive \(.*)/delete_recursive("\1");/g' \
  -e 's/(delete.*\)SYSTEM:\1/system//g' \
  -e 's/(delete.*\)PRELOAD:\1/preload//g' \
  -e 's/(delete.*\)DATA:\1/data//g' \
  -e 's/(delete.*\)SDCARD:\1/sdcard//g' \
  -e 's/(delete.*\)CACHE:\1/cache//g' \
  -e 's/(delete.*\)TMP:\([^)]*\))/\1/tmp/\2/g' \
  -e 's/(delete.*\)TMP:\1/tmp//g' \
  -e 's/(delete.*\)SDEXT:\1/sdext//g' \
  updater-script

echo "- symlink ..."

sed -i \
  -e 's/symlink \(.*)/symlink("\1");/g' \
  -e 's/(symlink("[^ ]*)[ ]*\([^"]*\));\)/\1, "\2/g' \
  -e 's/(symlink.*\)SYSTEM:\(.*)/\1/system/\2/g' \
  -e 's/(symlink.*\)DATA:\(.*)/\1/data/\2/g' \
  updater-script

echo "- write_raw_image ..."

sed -i \
  -e 's/write_raw_image \(.*)/write_raw_image("\1");/g' \
  -e 's/(write_raw_image.*\^[,]*) \(.*)/\1, "\2/g' \
  -e 's/(write_raw_image.*\)PACKAGE:\1/g' \
  -e 's/(write_raw_image.*\)BOOT:\1boot/g' \

```

```

-e 's/\(write_raw_image.*\)TMP:\([^)]*\))/\1\|tmp\|2)/g' \
-e 's/\(write_raw_image.*\)CACHE:\([^)]*\))/\1\|cache\|2)/g' \
updater-script

echo "- format and mount ..."

# Add 'mount' before package_extract_dir and delete.*
for new in ${new_list[@]}
do
  if [ "$new" == "userdata" ]
  then
    new2=data
  else
    if [ "$new" == "sdcard" ]
    then
      continue
    fi
    new2=$new
  fi

  line_num_del=`sed -n -e '/delete.*("\|'"$new2"'\.*)/{=;q}'
updater-script`
  line_num_pkg=`sed -n -e '/package_extract_dir([\^,]*,
"\|'"$new2"'\.*)/{=;q}' updater-script`

  use_pkg=0
  use_del=0

  if [ "$line_num_del" == "" ] && [ "$line_num_pkg" == "" ]
  then
    continue
  fi

  if [ "$line_num_del" == "" ]
  then
    use_pkg=1
  else
    if [ "$line_num_pkg" == "" ]
    then
      use_del=1
    else
      if [ "$line_num_del" -lt "$line_num_pkg" ]
      then
        use_del=1
      else
        use_pkg=1
      fi
    fi
  fi

  if [ $use_del == 1 ]
  then
    sed -i \
      -e '0,/\(delete.*\)("\|'"$new2"'\.*)/s//mount("MTD",
      ""'"$new"'", "\|'"$new2"'");\n\1("\|'"$new2"'2)/' \
updater-script

```

```

else
    sed -i \
        -e '0,/package_extract_dir(\([^,]*\),
"\/'"$new2"'\'(.*)\)/s//mount("MTD", "'"$new"'",
"\/'"$new2"'');\npackage_extract_dir(\1, "\/'"$new2"'\'2\)/' \
        updater-script
    fi
done

for i in $new_list_str
do

    orig=${orig_list[${i}]}
    new=${new_list[${i}]}

    new2=$new
    if [ "$new" == "userdata" ]
    then
        new2=data
    fi

    sed -i -e 's/format "'$orig'"/format("MTD", "'"$new"'");/g'
updater-script
    sed -i -e 's/format \(.*) "'$orig'"/format("\1", "MTD",
"'"$new"'");\nmount("\1", "MTD", "'"$new"'", "\/'"$new2"'");/g'
updater-script

    if [ "$new" == "boot" ]
    then

        if [ -e ../../../../../../boot/initrd.gz ] && [ -e
../../../../../../../../boot/zImage ]
        then
            sed -i -e 's/\(format(.*)\//boot\);)/\1\nmount("MTD",
"boot", "\/boot");/g' updater-script

            grep_mnt=`grep "mount(.*) \//boot\";" updater-script`

            if [ "$grep_mnt" == "" ]
            then
                sed -i -e 's/\(mount(.*)\//system\);)/\1\nmount("MTD",
"boot", "\/boot");/g' updater-script
            fi
        fi

        continue
    fi

    grep_mnt=`grep "mount(\\"MTD\\", \\"$new\\", \"/$new2\\");" updater-
script`

    grep_fmt2=`grep "format(\\"[A-Za-z0-9]*\\", \\"MTD\\", \\"$new\\");"
updater-script`

```

```

if [ "$grep_fmt2" != "" ]
then

    grep_mnt2=`grep "mount(\\"[A-Za-z0-9]*\\", \\"MTD\\", \\"$new\\",
\\"/$new2\\");" updater-script`

    if [ "$grep_mnt" != "" ] && [ "$grep_mnt2" != "" ]
    then
        sed -i -e '/mount("MTD", ""'$new'"" , "\\"'$new2'""");/d'
updater-script
    fi
fi

grep_fmt=`grep "format(\\"MTD\\", \\"$new\\");" updater-script`

if [ "$grep_fmt" != "" ]
then

    if [ "$grep_mnt" != "" ]
    then
        # Delete the line
        sed -i -e '/mount("MTD", ""'$new'"" , "\\"'$new2'""");/d'
updater-script

        # Add it after the format
        sed -i -e 's/format("MTD", ""'$new'"" );/format("MTD",
""'$new'"" );\nmount("MTD", ""'$new'"" , "\\"'$new2'""");/g' updater-
script
    fi
fi

done

line_num=`sed -n -e '/format /{=;q}' updater-script`

while [ "$line_num" != "" ]
do
    sed -i \
    -e '$line_num' s/format \(.*\)/format(\\"\\1\\");/g' \
    -e '$line_num' s/ /", "/g' \
    -e '$line_num' s/format\(.*\));/format\\1);\nmount\\1,
"???)";/g' \
    updater-script

    line_num=`sed -n -e '/format /{=;q}' updater-script`
done

if [ -e updater-script.orig ]
then
    line_num=`sed -n -e '/^mount(.*)?/?{=;q}' updater-script`

    while [ "$line_num" != "" ]
    do

        partial_str=`grep "^mount(" updater-script | grep -m 1 ??? | \

```

```

        sed \
        -e 's/"???");//g' \
        -e 's/"\\\\"/g' `

    orig_str=`grep -m 1 "$partial_str" updater-script.orig | sed -
e 's/\\/fwdslash/g'`

    if [ "$orig_str" != "" ]
    then
        sed -i \
        -e '$line_num' s/.*/'"$orig_str"/g' \
        -e '$line_num' s/fwdslash/\\/g' \
        updater-script
    else
        # We don't want to be stuck in an infinite loop
        break
    fi

    line_num=`sed -n -e '/^mount(.????/{=;q}' updater-script`
done
fi

echo "- unmount ..."
for new in ${new_list[@]}
do

    if [ "$new" == "userdata" ]
    then
        new2=data
    else
        new2=$new
    fi

    grep_mnt=`grep "mount(. * \"/$new\");" updater-script`

    if [ "$grep_mnt" != "" ]
    then
        echo "unmount(\"/$new2\");" >> updater-script
    fi

done

grep_mnt=`grep "mount(. * \"/data\");" updater-script`

if [ "$grep_mnt" != "" ]
then
    echo "unmount(\"/data\");" >> updater-script
fi

echo "- Final formatting ..."

sed -i \

```

```

-e 's/SYSTEM:\/\/system\/\/g' \
-e 's/TMP:\/\/tmp\/\/g' \
-e 's/DATA:\/\/data\/\/g' \
-e 's/SDCARD:\/\/sdcard\/\/g' \
-e 's/CACHE:\/\/cache\/\/g' \
-e 's/SDEXT:\/\/sdext\/\/g' \
-e 's/BOOT:\/\/boot\/\/g' \
updater-script

#
# parameter karakter aneh
#

sed -i \
-e 's\/\[sp\]\/ /g' \
-e 's\/\[cr\]\/\n/g' \
-e 's\/\[sc\]\/;/g' \
-e 's\/\[co\]\/,/g' \
updater-script

#
# shell udah maount
#

if [ `grep -c "run_program.*umount" updater-script` ]
then

    for new in ${new_list[@]}
    do
        if [ "$new" == "userdata" ]
        then
            new=data
        fi

        grep_mnt=`grep "mount(. * \"/$new\" );" updater-script`
        grep_umnt=`grep -c "umount /$new" updater-script`

        if [ $grep_umnt -gt 0 ] && [ "$grep_mnt" != "" ]
        then
            sed -i -e '/unmount("\/\"$new\"");/d' updater-script
        fi
    done

fi

sed -i \
-e 's/add_upgrade_property/add_upgrade_property();/g' \
-e 's/install_modem_package/install_modem_package();/g' \
updater-script

sed -i -e 's/write_blob_raw_image
PACKAGE:blob/assert(package_extract_file("blob", "\/tmp\/blob"),\n
write_blob_raw_image("\/tmp\/blob", "staging"),\n
delete("\/tmp\/blob"));/g' updater-script

```

```

#
# Clean up
#
sed -i \
  -e 's/\/"/);/");/g' \
  -e 's/;/ $/;/g' \
  updater-script

#
# hapus baris kosong
#
sed -i -e '/^$/d' updater-script

#
# Cool solution from the web :-)
#
# Delete duplicate, consecutive lines from a file (emulates
"uniq").
# First line in a set of duplicate lines is kept, rest are
deleted.
#
sed -i -e '$!N; /\^(.*)\n\1$/!P; D' updater-script

#
# mount point
#
cd ../../../../..
scripts/adjust_mnt updater-script
cd WORKING_*
cd META-INF/com/google/android

sed -i -e '/^$/d' updater-script

test_mnt=`grep "^mount(" updater-script`
test_mnt2=`grep "sbin/mount\", \"\" updater-script`
test_mnt3=`grep "boot.img\"\" updater-script | grep dev`

if [ "$test_mnt" != "" ] || [ "$test_mnt2" != "" ] || [
"$test_mnt3" != "" ]
then
  echo
  echo
  echo "Pastikan anda memasukkan Mount point sesuai dengan device
anda:"
  echo "-----"
  -----"

  if [ "$test_mnt" != "" ]
  then
    echo $test_mnt
  fi

  if [ "$test_mnt2" != "" ]

```



```

then
    echo $test_mnt2
fi

if [ "$test_mnt3" != "" ]
then
    echo $test_mnt3
fi

echo "-----"
-----"
fi

if [ -e updater-script ]
then

    cd ../../../../..
    scripts/check_which_update_binary
    cd WORKING_*
    cd META-INF/com/google/android

    echo
    echo "-----"
    -----"
    echo

    if [ "$2" != "yes" ]
    then
        echo "file updater-script telah terbuat di folder META-
INF/com/google/android"
        echo "tolong di review apabila terdapat potensi error."
        echo ""
    fi

    if [ "$1" != "for_zip" ]
    then
        mv -f update-script update-script.orig
        echo "file update-binary juga telah di ubah nama menjadi
update-binary.orig."
    else

        if [ "$2" == "no" ]
        then
            echo "jangan hapus file apapun di folder META-
INF/com/google/android."
            echo
        fi
    fi

    grep_fmt=`grep -n "^format(" updater-script`
    grep_mount=`grep -n "^mount(" updater-script`
    grep_huh=`grep -n "???" updater-script`

    if [ "$grep_huh" != "" ]

```

```

then
    echo
    echo "-----"
    -----"
    echo
    echo "NOTE: pastikan anda untuk fix manual beberapa command
yang sekiranya janggal"
    echo "        samakan dengan file ori updater-script. sesuaikan
perintahnya"
    echo
    echo $grep_huh | sed -e 's/;[ ]*/;\n/g'
    echo
fi

else
    echo "Error: tidak bisa membuat updater-script!"
fi

cd ../../../../..

```

Script create_edify_def

```

#####
#####
#
# project MTK Android ROM Builder
# Copyright 2015 Blackant38 <ekoheriyanto8@gmail.com>
#
# MTK Android ROM Builder adalah projek open source yang di
khususkan
# hanya untuk pribadi & atau edukasi.dimohon untuk tidak
menjualbelikan
# atau menduplikat tanpa seizin pengembang.
#
#####
#####

device=`scripts/get_variant_or_device_name`

if [ ! -e "tools/edify_defs/$device" ] && [ "$device" != "UNKNOWN"
]
then

    cd WORKING_*

    out=(`grep ^mount META-INF/com/google/android/updater-script |
grep block`)
    if [ "$out" == "" ]
    then
        out=(`grep "^run program(\\"/sbin/mount\\" META-

```

```

INF/com/google/android/updater-script | grep block`)
fi

if [ "$out" != "" ]
then
    echo
    echo
"=====
=====
    echo
    echo "WARNING! Device '$device' tidak bisa terdeteksi"

    file_path=./tools/edify_defs/$device

    echo "change_mnt=yes" >> $file_path
    echo >> $file_path

    mnt_name_list=( system cache data sdcard )
    var_name_list=( sys_mnt cache_mnt data_mnt sdcard_mnt )
    got_block_id=0
    found_param1=0

    for item in ${out[@]}
    do

        # Look for ext3/ext4 etc.
        if [ "`echo $item | grep -c \"mount(\\\\"ext\\\\" != 0 ] && [
$found_param1 == 0 ]
        then
            param1=`echo $item | sed -e 's/mount(\\\\"(.*)\\\\"/,/\\1/g'`
            echo >> $file_path
            echo "param1=$param1" >> $file_path
            echo "param2=EMMC" >> $file_path
            echo >> $file_path
            found_param1=1

            echo "param1_sdcard=vfat" >> $file_path
            echo "param2_sdcard=MTD" >> $file_path
            echo >> $file_path

            # Look for /dev/block/mmcblk0 etc.
            elif [ "`echo $item | grep -c block`" != 0 ]
            then
                got_block_id=1
                slash=\\\\"
                block_id=`echo $item | sed -e 's/\\/\\/g' -e 's/,/\\/g' -e
's/(\\|\\/)/'$slash'\\1/g'`

            elif [ $got_block_id == 1 ]
            then

                # Look for mount name
                if [ "`echo $item | grep -c /`" == "1" ]
                then
                    for (( i = 0 ; i < ${#mnt_name_list[@]} ; i++ ))

```

```

do
    mnt=${mnt_name_list[$i]}

    if [ "`echo $item | grep $mnt`" != "" ]
    then
        var=${var_name_list[$i]}
        echo "$var=$block_id" >> $file_path
        got_block_id=0
    fi
done
fi
fi

done

# Default ext type
if [ $found_param1 == 0 ]
then
    echo >> $file_path
    echo "param1=ext4" >> $file_path
    echo "param2=EMMC" >> $file_path
    echo >> $file_path

    echo "param1_sdcard=vfat" >> $file_path
    echo "param2_sdcard=MTD" >> $file_path
    echo >> $file_path
fi

boot_out=(`grep write_raw_image META-
INF/com/google/android/updater-script \
| grep boot.img | grep block`)

if [ "$boot_out" != "" ]
then
    slash=\\
    boot_mnt=`echo ${boot_out[1]} | sed -e 's/\\"(.*)\\",/\1/g'
-e 's/\\(\\)/'\$slash'\1/g'`

    if [ "`echo $boot_mnt | grep block`" != "" ]
    then
        echo "boot_mnt=$boot_mnt" >> $file_path
        echo "fix_boot2=yes" >> $file_path
    fi
fi

for (( i = 0 ; i < ${#var_name_list[@]} ; i++ ))
do
    var=${var_name_list[$i]}
    if [ "`grep $var $file_path`" == "" ]
    then
        echo "$var=???" >> $file_path
    fi
done

```

```

echo
echo "tools/edify_defs/$device:"
echo "-----"
echo
cat $file_path
echo
echo "-----"
echo
echo
fi

cd ..
fi

```

Script decompile_apk

```

#####
#####
#
# project MTK Android ROM Builder
# Copyright 2015 Blackant38 <ekoheriyanto8@gmail.com>
#
# MTK Android ROM Builder adalah projek open source yang di
khususkan
# hanya untuk pribadi & atau edukasi.dimohon untuk tidak
menjualbelikan
# atau menduplikat tanpa seizin pengembang.
#
#####
#####

clear
echo "====decompile apk===="
echo "masukan file *.apk yang akan di decompile"
echo "ke dalam folder apktool/repack."
echo
echo "script akan mendecompile 1 file apk saja"
echo "pada folder tersebut.jika file *.apk lain"
echo "yang terdapat pada folder tersebut akan "
echo "dihapus"
echo
echo "tekan enter untuk melanjutkan"
echo

read enterkey
clear

cd apktool/repack
rm -f *.apk
rm -f -r apk_repack

```

```

echo "masukan file apk daalam folder 'repack'."
echo
echo "tekan enter untuk melanjutkan"
echo

read enterkey
clear
apktool d *.apk -o apk_repack
exit 0

```

Script deodex_apk

```

#####
#####
#
# project MTK Android ROM Builder
# Copyright 2015 Blackant38 <ekoheriyanto8@gmail.com>
#
# MTK Android ROM Builder adalah projek open source yang di
khususkan
# hanya untuk pribadi & atau edukasi.dimohon untuk tidak
menjualbelikan
# atau menduplikat tanpa seizin pengembang.
#
#####
#####

while :
do

    clear

    echo
    echo "-----"
    echo
    echo "Deodex adalah proses mengembalikan kembali file .dex yang
sebelumnya "
    echo "sudah dioptimalisasi (.odex) menjadi file .dex."
    echo "Proses ini akan mengambil cache ODEX kemudian menyusun
kembali menjadi "
    echo "file APK saja. Deodex memungkinkan proses pengeditan APK
file menjadi lebih mudah."
    echo
    echo "ROM yang telah DeODEXed, memiliki paket aplikasi yang
telah di jadikan "
    echo "satu kembali dalam satu file APK,sehingga mengijinkan
modifikasi pada file "
    echo "APK seperti perubahan tema. Karena tidak ada kode/bagian
paket aplikasi yang "
    echo "lokasinya berbeda maka integritas paket tetap terjaga. "
    echo

```

```

echo "Proses Deodex memerlukan beberapa menit. "
echo
echo "-----"
-----"
echo

if [ ! -d WORKING_* ]
then
    echo Folder tidak ditemukan!
    scripts/press_enter
    exit 0
fi

cd WORKING_*

if [ ! -d system/app ]
then
    echo "tidak ditemukan folder system/app!"
    cd ..
    scripts/press_enter
    exit 0
fi

if [ ! -d system/framework ]
then
    echo "tidak ditemukan folder system/framework!"
    cd ..
    scripts/press_enter
    exit 0
fi

if [ -d preload/symlink/system/app ]
then

    cd preload/symlink/system/app

    num_preload_odex=`find . | grep -c "\.odex$"`
    num_preload_apk=`find . | grep -c "\.apk$"`
    num_sysapp_odex=`find ../../../../system/app | grep -c
"\.odex$"`
    num_sysfr_odex=`find ../../../../system/framework | grep -c
"\.odex$"`

    if [ $num_preload_odex == 0 ] && [ $num_sysapp_odex == 0 ]
    then
        echo "tidak ditemukan file ODEX di
/preload/symlink/system/app atau /system/app"
        echo

        elif [ $num_preload_odex == 0 ] && [ $num_preload_apk == 0 ]
        then

            if [ ! -e preload_apk_list ]
            then
                echo "Warning: tidak ditemukan APK/ODEX file di
/preload/symlink/system/app"

```

```

        echo
    fi

    # Case 3 - apk/odex files in preload
    elif [ $num_preload_odex -gt 0 ] || [ $num_preload_apk -gt 0 ]
    then

        echo "NOTE: $num_preload_apk APK dan $num_preload_odex ODEX
ditemukan di folder /preload."
        echo
        echo "pindah file ke /system/app (y/n)?"
        echo -n "masukan pilihan: "

        read move_files
        echo

        if [ "$move_files" == "n" ]
        then
            echo "WARNING: tidak dapat memindahkan!"
        else

            if [ $num_preload_apk -gt 0 ]
            then
                apk_list=`ls *.apk`
                echo $apk_list > preload_apk_list
                mv -fv *.apk ../../../../system/app/
            else
                touch placeholder
            fi

            if [ $num_preload_odex -gt 0 ]
            then
                mv -fv *.odex ../../../../system/app/
            fi

            fi

            echo
        fi

        cd ../../../../..
    fi

    num_odex_fr=`find system/framework | grep -c "\.odex$"`
    num_odex_app=`find system/app | grep -c "\.odex$"`

    echo "Found $num_odex_fr *.odex file di /system/framework"
    echo "Found $num_odex_app *.odex file di /system/app"

    if [ -e preload/symlink/system/app/preload_apk_list ]
    then
        echo
        echo "NOTE: apk akan di pindahkan kembali ke folder /preload"
        echo "setelah proses deodex selesai."
    fi

```



```

    echo
fi

if [ $num_odex_fr == 0 ] && [ $num_odex_app == 0 ]
then
    echo
    echo "tidak perlu deodex de-odex!"
    cd ..
    scripts/press_enter
    exit 0
fi

if [ $num_odex_fr == 0 ] && [ $num_odex_app -gt 0 ]
then
    echo
    echo "WARNING: harus mendeodex file di folder framework"
    echo "baru mendeodex file di folder app ."
    echo
fi

cd ..
scripts/convert_to_unix tools/deodex_files/api_level.txt
api_level=`scripts/get_api_level`
cd WORKING_*

echo
echo "Masukan pilihan:"
echo

if [ $num_odex_app -gt 0 ] || [ $num_odex_fr -gt 0 ]
then
    echo " bb = Back up semua folder yang akan di deodex"
fi

echo " v = Set versi OS Android (Current API level =
$api_level)"
echo

if [ $num_odex_fr -gt 0 ]
then
    echo " f = Deodex /system/framework"
fi
if [ $num_odex_app -gt 0 ]
then
    echo " a = Deodex /system/app"
fi
if [ $num_odex_app -gt 0 ] && [ $num_odex_fr -gt 0 ]
then
    echo " b = Deodex semua folder"
fi
echo " s = Deodex per file"
echo " x = keluar"

```

```

echo
echo -n "? "

read enterChoice

list1=( app )
list2=( framework )
list3=( framework app )

if [ "$enterChoice" == "a" ]
then
    dir_list=app
elif [ "$enterChoice" == "v" ]
then
    cd ..
    scripts/change_api_level
    continue
elif [ "$enterChoice" == "f" ]
then
    dir_list=framework
elif [ "$enterChoice" == "b" ]
then
    dir_list="framework app"
elif [ "$enterChoice" == "s" ]
then
    cd ..
    scripts/choose_single_deodex
    continue
elif [ "$enterChoice" == "bb" ]
then

    folder_list=( framework app )
    date_str=`date +%m%d%y_%H%M%S`

    for f in ${folder_list[@]}
    do

        backup_folder="old_`echo -n $f`_$date_str"
        cd system/$f

        echo
        echo "membuat folder back up $backup_folder diluar folder
utama ..."
        mkdir ../../../../$backup_folder
        cp -r * ../../../../$backup_folder/
        echo "selesai membackup folder $f folder."

        cd ../../..
    done

    cd ..
    scripts/press_enter
    continue

elif [ "$enterChoice" == "x" ]

```

```

then
    cd ..
    exit 0
else
    cd ..
    continue
fi

cd ..
scripts/folder_deodex_apk "$dir_list"

cd WORKING_*

num_odex_app=`find system/app | grep -c "\.odex$"`
num_apk_app=`find system/app | grep -c "\.apk$"`
num_odex_fr=`find system/framework | grep -c "\.odex$"`

echo
echo "$num_odex_app *.odex files remain in system/app"
echo "$num_odex_fr *.odex files remain in system/framework"

if [ $num_odex_fr == 0 ] && [ $num_odex_app -gt 0 ]
then
    echo
    echo "WARNING: harus mendeodex file di folder framework"
    echo "baru mendeodex file di folder app ."
fi

cd ..
scripts/press_enter

if [ $num_odex_fr == 0 ] && [ $num_odex_app == 0 ]
then
    exit 0
fi

done

```

Script do_zipalign

```

#####
#####
#
# project MTK Android ROM Builder
# Copyright 2015 Blackant38 <ekoheriyanto8@gmail.com>
#
# MTK Android ROM Builder adalah projek open source yang di
khususkan
# hanya untuk pribadi & atau edukasi.dimohon untuk tidak

```

```

menjualbelikan
# atau menduplikat tanpa seizin pengembang.
#
#####
#####

if [ "$1" != "for_build" ]
then
clear
echo

if [ -d WORKING_* ]
then
echo folder kerja ditemukan
else
echo tidak ditemukan folder kerja!
exit 1
fi
fi

if [ `uname | grep CYGWIN` ]
then
zipalign_file=zipalign.exe

elif [ `uname | grep Linux` ]
then
zipalign_file=zipalign

elif [ `sw_vers | grep -o Mac` ]
then
zipalign_file=zipalign.mac

else
echo Error: OS tidak terdeteksi, tidak bisa melanjutkan zipalign
exit 1
fi

return_val=0

if [ "$1" == "for_build" ]
then

if [ "$2" == "no" ]
then
echo
echo -n "apakah kamu akan mengoptimasi APK dengan
zipalign(y/n)?: "
read do_zipalign
fi

else
clear

```

```

echo
echo "-----"
-----"
echo
echo "Zipalign adalah prosedur untuk mengoptimasi file *.apk"
echo "di folder kerja untuk meminimalisir penggunaan RAM ."
echo
echo "-----"
-----"
echo

echo -n "zipalign (y/n)?: "
read do_zipalign
fi

if [ "$do_zipalign" == "n" ]
then
    exit 0
fi

echo
cd WORKING_*

grep_cmd=`find . -type f | grep \\.apk$ | sort -f`

name1=

for filename in $grep_cmd
do
    if [ "`echo $filename | grep .apk$`" == "" ]
    then

        if [ "$name1" == "" ]
        then
            name1=$filename
        else
            name1="$name1 $filename"
        fi

        continue

    elif [ "$name1" != "" ]
    then

        # Replace space with underscore temporarily
        name2=$filename
        original_name="$name1 $name2"
        filename=`echo $original_name | tr ' ' '_'`
        mv "$name1 $name2" $filename
        found_space=yes
        echo "Zipaligning $original_name ..."

    else
        found_space=no

```

```

    echo "Zipaligning $filename ..."
fi

new_file=$filename\_new
../tools/zipalign_files/$zipalign_file 4 $filename $new_file

if [ -e $new_file ]
then

    if [ "$found_space" == "yes" ]
    then
        mv -f $new_file "$name1 $name2"
        rm -f $filename

    else
        mv -f $new_file $filename
    fi

else

    return_val=1
    echo "Error: tidak bisa melanjutkan zipalig, membatalkan"
    if [ `uname | grep Linux` ]
    then
        echo "jika menggunakan LInux 64-bit, pastikan untuk
menginstall package ia32-libs"
    fi
    break
fi

name1=

done

cd ..
echo

if [ "$return_val" == "0" ]
then
    echo "zipalign selesai"
fi

exit $return_val

```

Script extract_boot_img

```

#####
#####
#
# project MTK Android ROM Builder
# Copyright 2015 Blackant38 <ekoheriyanto8@gmail.com>

```

```

#
# MTK Android ROM Builder adalah proyek open source yang di
# khususkan
# hanya untuk pribadi & atau edukasi.dimohon untuk tidak
# menjualbelikan
# atau menduplikat tanpa seizin pengembang.
#
#####
#####

#
# scripts akan extract konten boot.img yg ada di semua folder
#

clear

date_str=`date '+%m%d%y_%H%M%S'`
boot_dir=bootimg_`date_str`

echo
echo "Membuat folder `pwd`/$boot_dir ..."
mkdir $boot_dir

echo
echo "---> Masukkan file boot.img/recovery.img di folder berikut,
kemudian tekan enter <--"

scripts/press_enter

cd $boot_dir

if [ -e boot.img ] || [ -e recovery.img ]
then

    if [ -e recovery.img ]
    then
        echo "Mengubah recovery.img ke boot.img (untuk script)"
        mv -f recovery.img boot.img
    fi

    cd ..
    scripts/check_bootimg_header $boot_dir
    scripts/check_kernel_offset $boot_dir
    res=$?

    if [ "$res" == "0" ]
    then
        kernel_file=extract-kernel.pl
        ramdisk_file=extract-ramdisk.pl
    else
        rm -rf $boot_dir
        exit 0
    fi

    cd $boot_dir
    cp ../tools/extract_boot_files/$kernel_file extract-kernel.pl

```

```

cp ../tools/extract_boot_files/$ramdisk_file extract-ramdisk.pl

echo
echo Mengextract kernel ...
./extract-kernel.pl boot.img 2>/dev/null

if [ ! -e zImage ]
then
    echo "Error: No zImage found!"
else
    test_z=`od -A n -j 1 -N 4 zImage | sed 's/ //g'`
    if [ "$test_z" == "" ]
    then
        echo "Error: zImage is empty!"
    fi
fi

echo Mengextract ramdisk ...
./extract-ramdisk.pl boot.img 2>/dev/null

if [ ! -d boot.img-ramdisk ]
then
    echo "Error: tidak ditemukan folder ramdisk!"
fi

rm boot.img
rm extract-*.pl
cd ..
echo
echo "konten dari $boot_dir:"
echo
echo "`ls -l $boot_dir`"

else
    echo "Error: boot.img atau recovery.img tidak ada!"
    echo
    cd ..
    echo "Menghapus $boot_dir folder"
    rm -rf $boot_dir
    exit 0
fi

```

Script extract_kernel_and_modules

```

#####
#####
#
# project MTK Android ROM Builder
# Copyright 2015 Blackant38 <ekoheriyanto8@gmail.com>

```



```

#
# MTK Android ROM Builder adalah proyek open source yang di
# khususkan
# hanya untuk pribadi & atau edukasi.dimohon untuk tidak
# menjualbelikan
# atau menduplikat tanpa seizin pengembang.
#
#####
#####

clear

if [ -d MY_DEVICE_WORKING ]
then
    echo "Menghapus folder MY_DEVICE_WORKING"
    rm -rf MY_DEVICE_WORKING
fi

if [ -d MY_KERNEL ]
then
    echo "Menghapus Folder MY_KERNEL"
    rm -rf MY_KERNEL
fi

if [ -d BOOT-EXTRACTED ]
then
    echo "Menghapus folder BOOT-EXTRACTED"
    rm -rf BOOT-EXTRACTED
fi

clear
echo
echo
"*****"
echo
echo "--> Pilih ROM yang akan digunakan sebagai base <--"
echo
echo
"*****"

scripts/check_rom no_info kernel_only

# If working folder is open and cannot move it, handle the error
properly
if [ "$?" != "1" ]
then
    scripts/press_enter
else
    exit 1
fi

if [ -d WORKING_* ]
then

```

```

cd WORKING_*

if [ -e zImage ]
then
    mkdir ../BOOT-EXTRACTED
    cp -v zImage ../BOOT-EXTRACTED/
    cd ..

else
    cd ..
    scripts/ensure_boot_extracted
fi

else
    exit 0
fi

if [ ! -d BOOT-EXTRACTED ]
then
    echo "Error: folder BOOT-EXTRACTED tidak ditemukan"
    exit 0
fi

echo
echo "mengubah nama folder kerja ke MY_DEVICE_WORKING"
mv WORKING_* MY_DEVICE_WORKING

echo
echo "Membuat folder MY_KERNEL"
mkdir MY_KERNEL

#
# Copy base kernel to MY_KERNEL
# copy base modules/drivers to MY_KERNEL
#

echo "Mencopy zImage ke MY_KERNEL"
cp BOOT-EXTRACTED/zImage MY_KERNEL/

#
# We need this step so that we can determine base address
# when porting!
#

echo "Mencopy boot.img ke MY_KERNEL"
cp MY_DEVICE_WORKING/boot.img MY_KERNEL/

if [ -d MY_DEVICE_WORKING/system/lib/modules ]
then
    mkdir -p MY_KERNEL/system/lib/modules
    echo
    echo "Mencopy wireless (system/lib/modules) ..."
    cp -r MY_DEVICE_WORKING/system/lib/modules/*

```

```

MY_KERNEL/system/lib/modules/
fi

if [ -e MY_DEVICE_WORKING/system/lib/libacdk.so ]
then
    mkdir -p MY_KERNEL/system/lib
    echo
    echo "Mencopy libacdk.so (system/lib/libacdk.so) ..."
    cp -v MY_DEVICE_WORKING/system/lib/libacdk.so
MY_KERNEL/system/lib/
fi

if [ -e MY_DEVICE_WORKING/system/lib/libaudio.primary.default.so ]
then
    mkdir -p MY_KERNEL/system/lib
    echo
    echo "Mencopy libaudio.primary.default.so
(system/lib/libaudio.primary.default.so) ..."
    cp -v MY_DEVICE_WORKING/system/lib/libaudio.primary.default.so
MY_KERNEL/system/lib/
fi

if [ -e MY_DEVICE_WORKING/system/lib/libc.so ]
then
    mkdir -p MY_KERNEL/system/lib
    echo
    echo "Mencopy libc.so (system/lib/libc.so) ..."
    cp -v MY_DEVICE_WORKING/system/lib/libc.so MY_KERNEL/system/lib/
fi

if [ -e MY_DEVICE_WORKING/system/lib/libcam.camadapter.so ]
then
    mkdir -p MY_KERNEL/system/lib
    echo
    echo "Mencopy libcam.camadapter.so
(system/lib/libcam.camadapter.so) ..."
    cp -v MY_DEVICE_WORKING/system/lib/libcam.camadapter.so
MY_KERNEL/system/lib/
fi

if [ -e MY_DEVICE_WORKING/system/lib/libcam.campipe.so ]
then
    mkdir -p MY_KERNEL/system/lib
    echo
    echo "Mencopy libcam.campipe.so (system/lib/libcam.campipe.so)
..."
    cp -v MY_DEVICE_WORKING/system/lib/libcam.campipe.so
MY_KERNEL/system/lib/
fi

if [ -e MY_DEVICE_WORKING/system/lib/libcam.camshot.so ]
then
    mkdir -p MY_KERNEL/system/lib
    echo
    echo "Mencopy libcam.camshot.so (system/lib/libcam.camshot.so)
..."

```

```
cp -v MY_DEVICE_WORKING/system/lib/libcam.camshot.so
MY_KERNEL/system/lib/
fi

if [ -e MY_DEVICE_WORKING/system/lib/libcam.client.so ]
then
  mkdir -p MY_KERNEL/system/lib
  echo
  echo "Mencopy libcam.client.so (system/lib/libcam.client.so)
  ..."
  cp -v MY_DEVICE_WORKING/system/lib/libcam.client.so
MY_KERNEL/system/lib/
fi

if [ -e MY_DEVICE_WORKING/system/lib/libcam.paramsmgr.so ]
then
  mkdir -p MY_KERNEL/system/lib
  echo
  echo "Mencopy libcam.paramsmgr.so
(system/lib/libcam.paramsmgr.so) ..."
  cp -v MY_DEVICE_WORKING/system/lib/libcam.paramsmgr.so
MY_KERNEL/system/lib/
fi

if [ -e MY_DEVICE_WORKING/system/lib/libcam.utils.so ]
then
  mkdir -p MY_KERNEL/system/lib
  echo
  echo "Mencopy libcam.utils.so (system/lib/libcam.utils.so) ..."
  cp -v MY_DEVICE_WORKING/system/lib/libcam.utils.so
MY_KERNEL/system/lib/
fi

if [ -e MY_DEVICE_WORKING/system/lib/libcam_camera_exif.so ]
then
  mkdir -p MY_KERNEL/system/lib
  echo
  echo "Mencopy libcam_camera_exif.so
(system/lib/libcam_camera_exif.so) ..."
  cp -v MY_DEVICE_WORKING/system/lib/libcam_camera_exif.so
MY_KERNEL/system/lib/
fi

if [ -e MY_DEVICE_WORKING/system/lib/libcamdrv.so ]
then
  mkdir -p MY_KERNEL/system/lib
  echo
  echo "Mencopy libcamdrv.so (system/lib/libcamdrv.so) ..."
  cp -v MY_DEVICE_WORKING/system/lib/libcamdrv.so
MY_KERNEL/system/lib/
fi

if [ -e MY_DEVICE_WORKING/system/lib/libcamera_client.so ]
then
  mkdir -p MY_KERNEL/system/lib
  echo
```

```

    echo "Mencopy libcamera_client.so
(system/lib/libcamera_client.so) ..."
    cp -v MY_DEVICE_WORKING/system/lib/libcamera_client.so
MY_KERNEL/system/lib/
fi

if [ -e MY_DEVICE_WORKING/system/lib/libcameracustom.so ]
then
    mkdir -p MY_KERNEL/system/lib
    echo
    echo "Mencopy libcameracustom.so (system/lib/libcameracustom.so)
..."
    cp -v MY_DEVICE_WORKING/system/lib/libcameracustom.so
MY_KERNEL/system/lib/
fi

if [ -e MY_DEVICE_WORKING/system/lib/libfeatureio.so ]
then
    mkdir -p MY_KERNEL/system/lib
    echo
    echo "Mencopy libfeatureio.so (system/lib/libfeatureio.so) ..."
    cp -v MY_DEVICE_WORKING/system/lib/libfeatureio.so
MY_KERNEL/system/lib/
fi

if [ -e MY_DEVICE_WORKING/system/lib/libimageio.so ]
then
    mkdir -p MY_KERNEL/system/lib
    echo
    echo "Mencopy libimageio.so (system/lib/libimageio.so) ..."
    cp -v MY_DEVICE_WORKING/system/lib/libimageio.so
MY_KERNEL/system/lib/
fi

if [ -e MY_DEVICE_WORKING/system/lib/libspeech_enh_lib.so ]
then
    mkdir -p MY_KERNEL/system/lib
    echo
    echo "Mencopy libspeech_enh_lib.so
(system/lib/libspeech_enh_lib.so) ..."
    cp -v MY_DEVICE_WORKING/system/lib/libspeech_enh_lib.so
MY_KERNEL/system/lib/
fi

if [ -e MY_DEVICE_WORKING/system/lib/libspeexresampler.so ]
then
    mkdir -p MY_KERNEL/system/lib
    echo
    echo "Mencopy libspeexresampler.so
(system/lib/libspeexresampler.so) ..."
    cp -v MY_DEVICE_WORKING/system/lib/libspeexresampler.so
MY_KERNEL/system/lib/
fi

if [ -e MY_DEVICE_WORKING/system/lib/libSR_AudioIn.so ]
then

```

```

mkdir -p MY_KERNEL/system/lib
echo
echo "Mencopy libSR_AudioIn.so (system/lib/libSR_AudioIn.so)
..."
cp -v MY_DEVICE_WORKING/system/lib/libSR_AudioIn.so
MY_KERNEL/system/lib/
fi

if [ -e
MY_DEVICE_WORKING/system/lib/libwebrtc_audio_preprocessing.so ]
then
mkdir -p MY_KERNEL/system/lib
echo
echo "Mencopy libwebrtc_audio_preprocessing.so
(system/lib/libwebrtc_audio_preprocessing.so) ..."
cp -v
MY_DEVICE_WORKING/system/lib/libwebrtc_audio_preprocessing.so
MY_KERNEL/system/lib/
fi

if [ -d MY_DEVICE_WORKING/system/usr ]
then
mkdir -p MY_KERNEL/system/usr
echo
echo "Mencopy key layout (system/usr) ..."
cp -r MY_DEVICE_WORKING/system/usr/* MY_KERNEL/system/usr/
fi

if [ -d MY_DEVICE_WORKING/system/lib/hw ]
then
mkdir -p MY_KERNEL/system/lib/hw
echo
echo "Mencopy hw (system/lib/hw) ..."
cp -r MY_DEVICE_WORKING/system/lib/hw/* MY_KERNEL/system/lib/hw/
fi

if [ -e MY_DEVICE_WORKING/system/etc/vold.fstab ]
then
mkdir -p MY_KERNEL/system/etc
echo
echo "Mencopy sdcard mount info (system/etc/vold.fstab) ..."
cp -v MY_DEVICE_WORKING/system/etc/vold.fstab
MY_KERNEL/system/etc/
fi

if [ -e MY_DEVICE_WORKING/system/etc/vold.fstab.fat.nand ]
then
mkdir -p MY_KERNEL/system/etc
echo
echo "Mencopy sdcard mount info (system/etc/vold.fstab.fat.nand)
..."
cp -v MY_DEVICE_WORKING/system/etc/vold.fstab.fat.nand
MY_KERNEL/system/etc/
fi

if [ -e MY_DEVICE_WORKING/system/etc/vold.fstab.nand ]

```

```

then
  mkdir -p MY_KERNEL/system/etc
  echo
  echo "Mencopy sdcard mount info (system/etc/vold.fstab.nand)
  ..."
  cp -v MY_DEVICE_WORKING/system/etc/vold.fstab.nand
MY_KERNEL/system/etc/
fi

if [ -e MY_DEVICE_WORKING/system/etc/.directory ]
then
  mkdir -p MY_KERNEL/system/etc
  echo
  echo "Mencopy .directory (system/etc/.directory) ..."
  cp -v MY_DEVICE_WORKING/system/etc/.directory
MY_KERNEL/system/etc/
fi

if [ -e MY_DEVICE_WORKING/system/etc/agps_profiles_conf.xml ]
then
  mkdir -p MY_KERNEL/system/etc
  echo
  echo "Mencopy agps_profiles_conf.xml
(system/etc/agps_profiles_conf.xml) ..."
  cp -v MY_DEVICE_WORKING/system/etc/agps_profiles_conf.xml
MY_KERNEL/system/etc/
fi

if [ -e MY_DEVICE_WORKING/system/etc/apns-conf.xml ]
then
  mkdir -p MY_KERNEL/system/etc
  echo
  echo "Mencopy apns-conf.xml (system/etc/apns-conf.xml) ..."
  cp -v MY_DEVICE_WORKING/system/etc/apns-conf.xml
MY_KERNEL/system/etc/
fi

if [ -e MY_DEVICE_WORKING/system/etc/spn-conf.xml ]
then
  mkdir -p MY_KERNEL/system/etc
  echo
  echo "Mencopy spn-conf.xml (system/etc/spn-conf.xml) ..."
  cp -v MY_DEVICE_WORKING/system/etc/spn-conf.xml
MY_KERNEL/system/etc/
fi

if [ -d MY_DEVICE_WORKING/system/etc/.tp ]
then
  mkdir -p MY_KERNEL/system/etc/.tp
  echo
  echo "Mencopy module (system/etc/.tp) ..."
  cp -r MY_DEVICE_WORKING/system/etc/.tp/*
MY_KERNEL/system/etc/.tp/
fi

if [ -d MY_DEVICE_WORKING/system/etc/firmware ]

```

```
then
  mkdir -p MY_KERNEL/system/etc/firmware
  echo
  echo "Mencopy firmware (system/etc/firmware) ..."
  cp -r MY_DEVICE_WORKING/system/etc/firmware/*
MY_KERNEL/system/etc/firmware/
fi

if [ -d MY_DEVICE_WORKING/system/etc/wide-dhcpv6 ]
then
  mkdir -p MY_KERNEL/system/etc/wide-dhcpv6
  echo
  echo "Mencopy wide-dhcpv6 (system/etc/wide-dhcpv6) ..."
  cp -r MY_DEVICE_WORKING/system/etc/wide-dhcpv6/*
MY_KERNEL/system/etc/wide-dhcpv6/
fi

if [ -d MY_DEVICE_WORKING/system/etc/wifi ]
then
  mkdir -p MY_KERNEL/system/etc/wifi
  echo
  echo "Mencopy wifi (system/etc/wifi) ..."
  cp -r MY_DEVICE_WORKING/system/etc/wifi/*
MY_KERNEL/system/etc/wifi/
fi

if [ -d MY_DEVICE_WORKING/system/etc/bluetooth ]
then
  mkdir -p MY_KERNEL/system/etc/bluetooth
  echo
  echo "Mencopy bluetooth module (system/etc/bluetooth) ..."
  cp -r MY_DEVICE_WORKING/system/etc/bluetooth/*
MY_KERNEL/system/etc/bluetooth/
fi

if [ -e MY_DEVICE_WORKING/system/etc/boot_logo_updater ]
then
  mkdir -p MY_KERNEL/system/bin
  echo
  echo "Mencopy boot_logo_updater (system/bin/boot_logo_updater)
  ..."
  cp -v MY_DEVICE_WORKING/system/bin/boot_logo_updater
MY_KERNEL/system/bin/
fi

if [ -e MY_DEVICE_WORKING/system/bin/bootanimation ]
then
  mkdir -p MY_KERNEL/system/bin
  echo
  echo "Mencopy bootanimation (system/bin/bootanimation) ..."
  cp -v MY_DEVICE_WORKING/system/bin/bootanimation
MY_KERNEL/system/bin/
fi

if [ -e MY_DEVICE_WORKING/system/bin/bugmailer.sh ]
then
```



```
mkdir -p MY_KERNEL/system/bin
echo
echo "Mencopy bugmailer.sh (system/bin/bugmailer.sh) ..."
cp -v MY_DEVICE_WORKING/system/bin/bugmailer.sh
MY_KERNEL/system/bin/
fi

if [ -e MY_DEVICE_WORKING/system/bin/mtkbt ]
then
mkdir -p MY_KERNEL/system/bin
echo
echo "Mencopy mtkbt (system/bin/mtkbt) ..."
cp -v MY_DEVICE_WORKING/system/bin/mtkbt MY_KERNEL/system/bin/
fi

if [ -e MY_DEVICE_WORKING/system/bin/vold ]
then
mkdir -p MY_KERNEL/system/bin
echo
echo "Mencopy sdcard mount info (system/bin/vold) ..."
cp -v MY_DEVICE_WORKING/system/bin/vold MY_KERNEL/system/bin/
fi

if [ -e MY_DEVICE_WORKING/system/bin/wlan_loader ]
then
mkdir -p MY_KERNEL/system/bin
echo
echo "Mencopy wlan_loader (system/bin/wlan_loader) ..."
cp -v MY_DEVICE_WORKING/system/bin/wlanloader
MY_KERNEL/system/bin/
fi

if [ -e MY_DEVICE_WORKING/system/bin/wpa_supplicant ]
then
mkdir -p MY_KERNEL/system/bin
echo
echo "Mencopy wpa_supplicant (system/bin/wpa_supplicant) ..."
cp -v MY_DEVICE_WORKING/system/bin/wpa_supplicant
MY_KERNEL/system/bin/
fi

if [ -e MY_DEVICE_WORKING/system/etc/vold.conf ]
then
mkdir -p MY_KERNEL/system/etc
echo
echo "Mencopy sdcard mount info (system/etc/vold.conf) ..."
cp -v MY_DEVICE_WORKING/system/etc/vold.conf
MY_KERNEL/system/etc/
fi

if [ -e MY_DEVICE_WORKING/system/sbin/libmnlp_mt6572 ]
then
mkdir -p MY_KERNEL/system/sbin
echo
echo "Mencopy libmnlp_mt6572 (system/sbin/libmnlp_mt6572) ..."
cp -v MY_DEVICE_WORKING/system/sbin/libmnlp_mt6572
```

```

MY_KERNEL/system/sbin/
fi

if [ -e MY_DEVICE_WORKING/system/sbin/mnld ]
then
  mkdir -p MY_KERNEL/system/sbin
  echo
  echo "Mencopy mnld (system/sbin/mnld) ..."
  cp -v MY_DEVICE_WORKING/system/sbin/mnld MY_KERNEL/system/sbin/
fi

if [ -d MY_DEVICE_WORKING/system/modules ]
then
  mkdir -p MY_KERNEL/system/modules
  echo
  echo "Mencopy other modules (system/modules) ..."
  cp -r MY_DEVICE_WORKING/system/modules/*
MY_KERNEL/system/modules/
fi

rm -rf BOOT-EXTRACTED

```

Script fix_lib_modules

```

#####
#####
#
# project MTK Android ROM Builder
# Copyright 2015 Blackant38 <ekoheriyanto8@gmail.com>
#
# MTK Android ROM Builder adalah projek open source yang di
# khususkan
# hanya untuk pribadi & atau edukasi.dimohon untuk tidak
# menjualbelikan
# atau menduplikat tanpa seizin pengembang.
#
#####
#####

cd WORKING_*

if [ -d system/lib/modules ]
then
  cd system/lib/modules

  echo

  symlinks=( `find . -type l -exec ls -l {} \; | sed 's/.* .\/\([^
]*\) .*/\1/g'` )
  source files=( `find . -type l -exec ls -l {} \; | sed 's/.*->

```

```

//g'` )

for (( i=0 ; i < ${#symlinks[@]} ; i++ ))
do
    link=${symlinks[$i]}
    src=${source_files[$i]}

    if [ "`echo $src | grep /\`" == "" ]
    then

        rm -f $link
        echo "Symlinking /system/lib/modules/$src -> $link"

        sed -i -e 's/\(set_perm_recursive 0 0 0755 0644 SYSTEM:[
]*$\)/symlink \/system\/lib\/modules\/'$src'
SYSTEM:lib\/modules\/'$link'\n\1/g' ../../../../META-
INF/com/google/android/update-script
        fi

    done

    # For MT6577
    if [ `uname | grep CYGWIN` ]
    then

        # wlan.ko fix
        if [ ! -e wlan.ko ]
        then
            if [ -e wlan_mt6620.ko ]
            then
                wlan_src=wlan_mt6620.ko
            elif [ -e wlan_mt6628.ko ]
            then
                wlan_src=wlan_mt6628.ko
            fi
        fi

        if [ "$wlan_src" != "" ]
        then
            echo "Symlinking /system/lib/modules/$wlan_src -> wlan.ko"

            sed -i -e 's/\(set_perm_recursive 0 0 0755 0644 SYSTEM:[
]*$\)/symlink \/system\/lib\/modules\/'$wlan_src'
SYSTEM:lib\/modules\/wlan.ko\n\1/g' ../../../../META-
INF/com/google/android/update-script
            fi

        # p2p.ko fix
        if [ ! -e p2p.ko ]
        then
            if [ -e p2p_mt6620.ko ]
            then
                p2p_src=p2p_mt6620.ko
            elif [ -e p2p_mt6628.ko ]
            then

```

```

        p2p_src=p2p_mt6628.ko
    fi
fi

if [ "$p2p_src" != "" ]
then
    echo "Symlinking /system/lib/modules/$p2p_src -> p2p.ko"

    sed -i -e 's/\(set_perm_recursive 0 0 0755 0644 SYSTEM:[
]*$\)/symlink \/system\/lib\/modules\/'$p2p_src'
SYSTEM:lib\/modules\/p2p.ko\n\1/g' ../../../../META-
INF/com/google/android/update-script
    fi

fi

cd ../../..
echo
fi

cd ..

```

Script grep_roms

```

#####
#####
#
# project MTK Android ROM Builder
# Copyright 2015 Blackant38 <ekoheriyanto8@gmail.com>
#
# MTK Android ROM Builder adalah projek open source yang di
khususkan
# hanya untuk pribadi & atau edukasi.dimohon untuk tidak
menjualbelikan
# atau menduplikat tanpa seizin pengembang.
#
#####
#####

cd original_update

grep_cmd1=`find . -maxdepth 1 -type f | grep -i \\.zip$ | sed 's/
/temp_space/g' 2>/dev/null`
grep_cmd2=`find . -maxdepth 1 -type f | grep -i \\.tar$ | sed 's/
/temp_space/g' 2>/dev/null`
grep_cmd3=`find . -maxdepth 1 -type f -name system.img
2>/dev/null`
grep_cmd4=`find . -maxdepth 1 -type f | grep -i \\.app$ | sed 's/
/temp_space/g' 2>/dev/null`

grep_cmd5=`find . -maxdepth 1 -type f -name factoryfs.img
2>/dev/null`

```

```

grep_cmd6=`find . -maxdepth 1 -type f -name system.img.ext4
2>/dev/null`
grep_cmd7=`find . -maxdepth 1 -type f -name factoryfs.rfs
2>/dev/null`

grep_cmd8=`find . -maxdepth 1 -type f -name PDA.tar.md5
2>/dev/null`

grep_cmd9=`find . -maxdepth 1 -type d | grep "\./WORKING_" | sed
's/ /temp_space/g' 2>/dev/null`

grep_cmd10=`find . -maxdepth 1 -type f -name system.rfs
2>/dev/null`

grep_cmd=`echo $grep_cmd1 $grep_cmd2 $grep_cmd3 $grep_cmd4
$grep_cmd5 $grep_cmd6 $grep_cmd7 \
    $grep_cmd8 $grep_cmd9 $grep_cmd10 | sed -e 's/\.\\///g' |
sort -f`

cd ..
echo $grep_cmd

```

Script init_marb

```

#####
#####
#
# project MTK Android ROM Builder
# Copyright 2015 Blackant38 <ekoheriyanto8@gmail.com>
#
# MTK Android ROM Builder adalah projek open source yang di
khususkan
# hanya untuk pribadi & atau edukasi.dimohon untuk tidak
menjualbelikan
# atau menduplikat tanpa seizin pengembang.
#
#####
#####

chmod 755 tools/extract_boot_files/extract-*.pl
chmod 755 tools/mkboot/*
chmod 755 tools/apktool_files/*
chmod 755 tools/apktool_files/SigningTool/*
chmod 755 tools/zipalign_files/*
chmod 755 tools/deodex_files/api_level.txt
rm -f symlinks.log
rm -f tools/explore2fs_windows/explore*.txt
rm -f tools/ext2read_windows/ext2*.log
rm -f ext2*.log
rm -f sh.exe.stackdump
rm -f scripts/.DS_Store

```

```

rm -f tools/.DS_Store
rm -f original_update/.DS_Store
rm -f .DS_Store
rm -f unzip.log

bin_list=( mkbootfs mkbootfs.exe mkbootimg mkbootimg.exe rsa.o
sha.o )

for mkboot_bin in ${bin_list[@]}
do
    if [ -e tools/mkboot/$mkboot_bin ]
    then
        rm tools/mkboot/$mkboot_bin
    fi
done

```

Script port_rom

```

#####
#####
#
# project MTK Android ROM Builder
# Copyright 2015 Blackant38 <ekoheriyanto8@gmail.com>
#
# MTK Android ROM Builder adalah projek open source yang di
khususkan
# hanya untuk pribadi & atau edukasi.dimohon untuk tidak
menjualbelikan
# atau menduplikat tanpa seizin pengembang.
#
#####
#####

scripts/extract_kernel_and_modules

if [ -d MY_DEVICE_WORKING ]
then
    my_device=`scripts/get_build_prop_val ro.product.device
MY_DEVICE_WORKING`
    my_board=`scripts/get_build_prop_val ro.product.board
MY_DEVICE_WORKING`
    my_platform=`scripts/get_build_prop_val ro.board.platform
MY_DEVICE_WORKING`
fi

if [ -d MY_KERNEL ]
then
    echo
    scripts/press_enter
else

```

```

    exit 1
fi

#
# grab donor ROM
#

clear
echo
echo
"*****"
echo
echo "--> ROM dari device lain yang akan di port ke device base <-
-"
echo
echo
"*****"

scripts/check_rom no_info

if [ -d WORKING_* ]
then

    scripts/press_enter

    cd WORKING_*

    if [ -d system ]
    then
        if [ -e boot.img ]
        then
            cd ..
            scripts/ensure_boot_extracted
        fi
    else
        cd ..
        echo
        echo "keluar dan membersihkan folder ..."

        rm -rf MY_KERNEL
        rm -rf MY_DEVICE_WORKING
        exit 0
    fi

else
    echo "belum ada ROM yang dipilih"
    echo "keluar dan membersihkan folder ..."
    rm -rf MY_KERNEL
    rm -rf MY_DEVICE_WORKING
    exit 0
fi

if [ ! -d BOOT-EXTRACTED ]
then

```

```

    echo "Error: Folder BOOT-EXTRACTED tidak ditemukan"
    exit 0
fi

clear
echo
echo
"*****"
echo
echo " PORTING ROM"
echo
echo
"*****"

#
# Replace kernel, modify ramdisk and then build new boot.img
#

echo
echo "Mencopy MY_KERNEL/zImage to BOOT-EXTRACTED folder"
cp -f MY_KERNEL/zImage BOOT-EXTRACTED/

echo
echo "Mencopy MY_KERNEL/boot.img ke dalam folder kerja baru..."

cd WORKING_*
cp -f ../MY_KERNEL/boot.img .
cd ..
echo

#
# Rename *.rc files if required
#

board=`scripts/get_product_board`

cd BOOT-EXTRACTED/boot.img-ramdisk
file_list=`ls *.$board.rc 2>/dev/null`

if [ "$file_list" == "" ]
then
    board=`ls init.*.rc | sed -e 's/init.goldfish.rc//g' -e
's/init.\([^.*\)*\).rc/\1/g' -e '/^$/d'`
    file_list=`ls *.$board.rc 2>/dev/null`
fi

if [ "$file_list" != "" ]
then
    for rc_file in ${file_list[@]}
    do
        new_name=`echo $rc_file | sed -e
's/'"$board"'/'"$my_board"'/g`

```



```

if [ "$src_file" != "$new_name" ]
then
    echo "Mengganti nama ..."
    mv -v $src_file $new_name
    echo
fi
done
fi

#
# init.rc refers to busybox
#
if [ "`grep -o -m 1 busybox init.rc`" == "busybox" ]
then
    cd ../../WORKING_*
    find_busybox=`find . -name busybox`
    cd ..

    if [ "$find_busybox" == "" ]
    then
        echo
        echo "-----"
        echo
        echo "WARNING: file ramdisk sudah init.rc terhubung dengan
busybox, tetapi "
        echo "                tidak ditemukan di folder kerja. jika ROM di
build sekarang,"
        echo "                ada kemungkinan boot screen tidak bisa
tampil."
        echo
        echo "                ditemukan file berikut:"
        echo
        echo `grep busybox BOOT-EXTRACTED/boot.img-ramdisk/init.rc`
        echo
        echo "tips:"
        echo
        echo "- copy file busybox ke folder /system/xbin. file binary
Busybox terdapat "
        echo " di folder "tools"."
        echo "- pastikan file update-script terdapat baris script
berikut:"
        echo " set_perm 0 0 04755 SYSTEM:xbin/busybox"
        echo "- kemungkinan ada beberapa symlinks yang hilang di
/system/xbin"
        echo ""
        echo -n "jalankan script untuk memperbaiki masalah ini (y/n)?:"
        echo ""

        read proceed_fix
        if [ "$proceed_fix" != "n" ]
        then
            scripts/add_busybox
        else

```

```

        echo "tidak bisa di perbaiki"
    fi

    echo
    echo "Jika sudah siap untuk membuild boot.img, tekan enter
untuk melanjutkan ..."
    read enterKey
    fi

else
    cd ../../
fi

scripts/build_boot_img

#
# atur values in build.prop
#
if [ "$my_device" == "UNKNOWN" ]
then
    echo "WARNING: Tidak bisa memodif ro.product.device di
build.prop (device=UNKNOWN)"
else
    scripts/set_build_prop_val ro.product.device $my_device
fi

if [ "$my_board" == "UNKNOWN" ]
then
    echo "WARNING: Tidak bisa memodif ro.product.board di build.prop
(board=UNKNOWN)"
else
    scripts/set_build_prop_val ro.product.board $my_board
fi

if [ "$my_platform" == "UNKNOWN" ]
then
    echo "WARNING: Tidak bisa memodif ro.board.platform di
build.prop (platform=UNKNOWN)"
else
    scripts/set_build_prop_val ro.board.platform $my_platform
fi

#
# Copy module files
#

if [ -d MY_KERNEL/system ]
then
    cd WORKING_*
    echo
    echo "Mencopy modules/drivers ..."
    cp -rf ../MY_KERNEL/system/* system/

```

```

    cd ..
fi

#tidak bisa cek mount points

cd WORKING_*

if [ -d META-INF/com/google/android ]
then

    cd META-INF/com/google/android

    if [ -e updater-script.orig ]
    then
        rm -f updater-script.orig
    fi

    if [ -e update-binary.orig ]
    then
        rm -f update-binary.orig
    fi

    #
    # use device's original updater-script for mount points
    #
    if [ -e ../../../../../../MY_DEVICE_WORKING/META-
INF/com/google/android/updater-script ]
    then
        cp ../../../../../../MY_DEVICE_WORKING/META-
INF/com/google/android/updater-script updater-script.orig
    fi

    cd ../../../../

fi

cd ..

rm -rf MY_KERNEL
rm -rf MY_DEVICE_WORKING

echo
echo
"=====
echo
echo "kamu sekarang sudah memport ROM dengan MARB, anda tinggal"
echo "membuil ROM jika semua telah selesai di edit"
echo
echo "Tips"
echo "-----"
echo "jika ukuran ROM sangat besar kamu bisa menghapus beberapa"
echo "aplilasi bloatware yang sekiranya tidak difungsikan di
sistem"

```

```
echo "di folder kerja, tetapi jangan hapus file sistem, itu bisa"
echo "mengakibatkan ROM tidak bisa boot"
echo
echo "jika HH anda tidak bisa boot setelah melakukan proses
porting"
echo "dengan MARB, silahkan kirim email ke
ekoheriyanto8@gmail.com"
echo "jika anda menemukan bug"
```

Script root_acces

```
#####
#####
#
# project MTK Android ROM Builder
# Copyright 2015 Blackant38 <ekoheriyanto8@gmail.com>
#
# MTK Android ROM Builder adalah projek open source yang di
khususkan
# hanya untuk pribadi & atau edukasi.dimohon untuk tidak
menjualbelikan
# atau menduplikat tanpa seizin pengembang.
#
#####
#####

if [ "$1" == "" ]
then
clear
fi

if [ ! -d WORKING_* ]
then
echo "Tidak ditemukan folder kerja!"
scripts/press_enter
exit 0
fi

echo
echo "ROOTING"
echo "-----"
echo
echo "dalam rooting ini, yang akan kita lakukan adalah:"
echo
echo "- Menambahkan aplikasi Superuser dan su binary"
echo "- Menambahkan script di update-script"

if [ "$sgs" == "0" ]
then
echo "- Update default.prop di ramdisk"
fi
echo
```

```

echo "-----"
echo

cd WORKING_*

if [ ! -d system ]
then
    echo "Folder /system tidak ditemukan!"
    cd ..
    scripts/press_enter
    exit 0
fi

cd system

if [ ! -e build.prop ]
then
    echo "Error: tidak ditemukan file build.prop!"
    cd ../../
    scripts/press_enter
    exit 0
fi

su_path=`find . -name su`
super_path=`find . -iname Superuser.apk`

super_su_exists=0
if [ "$super_path" == "" ]
then
    super_path=`find . -iname SuperSU.apk`
    if [ "$super_path" != "" ]
    then
        super_su_exists=1
    fi
fi
overwrite=0

if [ "$su_path" != "" ] && [ "$super_path" != "" ]
then
    echo "di dalam folder /system sudah terdapat $su_path dan
$super_path"
    echo -n "ganti (y/n)?: "
    read do_su

    if [ "$do_su" != "y" ]
    then
        cd ../../
        exit 0
    else
        overwrite=1
        echo
        echo
    fi
fi

fi

```

```

echo "versi superuser apa yang akan di install?"
echo

echo " c - Chainfire's SuperSU"
echo " d - ChainsDD's version "
echo " v - Versi asli (tidak direkomendasikan)"
echo
echo " x - batalkan rooting (default)"
echo

echo -n "? "
read enterVersion
echo

if [ "$enterVersion" == "c" ]
then
    folder=chainfire
elif [ "$enterVersion" == "d" ]
then
    if [ "`grep -c ro.build.version.release=1 build.prop`" != "0" ]
    then
        folder=cupcake-donut
    else
        folder=current
    fi
elif [ "$enterVersion" == "v" ]
then
    folder=old
else
    echo "Dibatalkan"
    cd ../../
    exit 1
fi

if [ "$overwrite" == "1" ]
then
    echo
    echo "mengganti binay lama SU dan Superuser app ..."
    rm -fv $su_path
    rm -fv $super_path
fi

if [ -d xbin ]
then
    echo ditemukan /system/xbin
else
    echo membuat /system/xbin
    mkdir xbin
fi

if [ "$enterVersion" == "d" ]
then
    cp -fv ../../tools/su_files/chainsdd/Superuser.apk app

```

```

cp -fv ../../tools/su_files/chainsdd/$folder/su xbin
else
cp -fv ../../tools/su_files/$folder/app/Superuser.apk app
cp -fv ../../tools/su_files/$folder/bin/su xbin
fi

if [ "$super_su_exists" == "1" ] && [ "$enterVersion" == "c" ]
then
mv -fv app/Superuser.apk app/SuperSU.apk
fi

cd ../../

echo
scripts/add_su_to_update_script

if [ "$sgs" == "0" ]
then
scripts/root_boot_img
fi

echo
echo "Finished rooting"
echo

scripts/press_enter

```

Script zip_file_to_working_folder

```

#####
#####
#
# project MTK Android ROM Builder
# Copyright 2015 Blackant38 <ekoheriyanto8@gmail.com>
#
# MTK Android ROM Builder adalah projek open source yang di
khususkan
# hanya untuk pribadi & atau edukasi.dimohon untuk tidak
menjualbelikan
# atau menduplikat tanpa seizin pengembang.
#
#####
#####

update_file=$1

echo "pilih $update_file"

```

```

# Trim spaces in file name
update_file2=`echo $update_file | tr ' ' '_ '`

if [ "$update_file" != "$update_file2" ]
then
    echo "Mrngubah nama ke $update_file2"
    mv "original_update/$update_file" original_update/$update_file2
    update_file=$update_file2
fi

#
# Working folder name
#
build_dir=`scripts/set_working_folder_name $2`

#
# Creating folder structure
#

echo
echo "membuat folder $build_dir ..."
mkdir $build_dir

echo
echo "ekstrak file dari zip ..."

if [ `uname | grep CYGWIN` ]
then

    unzip -q original_update/$update_file -d $build_dir 2>&1 | tee
unzip.log
    unzip_result=$?

    rm -f bootloader.img recovery.img recovery_signed.img
userdata.img radio.img rcdata.img splash1.nb0 splash2.nb0 android-
info.txt

    replace_count=`grep -c replace unzip.log`
    rm -f unzip.log

    if [ $replace_count -gt 0 ]
    then
        echo
        echo
        echo "WARNING"
        echo "-----"
        echo "di dalam folder dalam file zip terdapat file sama,
tetapi dengan ukuran berbeda"
        echo "hal ini akan berakibat error pada custom ROM"
        echo "kami merekomendasikan menggunakan linux. anda bisa
membatalkan projek, atau melanjutkan"
        echo
        echo -n "apakah andan akan melanjutkan (y/n)?: "
        read do_ignore
    fi
fi

```



```

    if [ "$do_ignore" != "y" ]
    then
        echo
        echo "OK, dibatalkan"
        cd ..
        exit 1
    else
        echo
        echo "OK, dibiarkan, tapi rom anda akan kurang setabil!"
    fi
fi

else
    unzip -q original_update/$update_file -d $build_dir -x
    bootloader.img recovery.img recovery_signed.img userdata.img
    radio.img rdata.img splash1.nb0 splash2.nb0 android-info.txt
    2>/dev/null

    unzip_result=$?

fi

if [ "$unzip_result" == "9" ]
then
    echo
    echo "Error: tidak bisa extract ROM ROM!"
    cd ..
    exit 1
fi

cd $build_dir

#
# Check for the required IMG files
#

echo

if [ "`find . -maxdepth 1 | grep .img | grep -c system_`" == "1" ]
then
    echo "Menganti nama ..."
    mv -v system_*.img system.img
fi

rm -vf hboot*.img

if [ "`find . -maxdepth 1 | grep .img | grep -c boot_`" == "1" ]
then
    echo "Menganti nama ..."
    mv -v boot_*.img boot.img
fi

if [ "`find . -maxdepth 1 | grep .img | grep -c lib_`" == "1" ]
then

```

```

    echo "Menganti nama ..."
    mv -v lib_*.img lib.img
fi

if [ -e system.img ] && [ ! -d system ]
then

    echo
    echo "ditemukan system.img"
    mkdir ../temp_img
    mv system.img ../temp_img

    if [ -e boot.img ]
    then
        echo "ditemukan boot.img"
        mv boot.img ../temp_img
    fi

    if [ -e lib.img ]
    then
        echo "ditemukan lib.img"
        mv lib.img ../temp_img
    fi

    echo
    rm -vf *
    mv ../temp_img/* .
    rmdir ../temp_img

    cd ..
    scripts/img_files_to_working_folder no_create $2
    exit $?

fi

if [ -d patch ]
then

    echo
    echo "Error: ini merupakan ROM update, bukan full ROM!
membatalkan."
    cd ..
    exit 1

elif [ ! -d system ] && [ ! -d SYSTEM ]
then

    echo
    echo "Error: tidak ditemukan sistem!"
    cd ..
    exit 1

```

```

fi

if [ ! -e boot.img ]
then
  if [ -e boot/initrd.gz ] && [ -e boot/zImage ]
  then
    echo
    echo "NAND ROM terdeteksi"
    echo "..."

    cd ..
    scripts/ensure_nand_extracted no_clear
    scripts/build_boot_img
    cd WORKING_*

  elif [ -e BOOT/kernel ] && [ -e BOOT/base ] && [ -e BOOT/cmdline
] && [ -d BOOT/RAMDISK ]
  then
    echo "Partisi berada dalam format folder; folder BOOT harus
diconvert"
    echo "converted ke boot.img ..."

    cd ..
    scripts/build_boot_img_from_folder_form
    cd WORKING_*

  else

    build_dir=`pwd`
    cd ..
    scripts/check_no_bootimg $build_dir
    cd $build_dir

    if [ ! -e boot.img ]
    then
      echo "Error: boot img tidak ditemukan"
      echo
      cd ..
      echo "Menghapus $build_dir ..."
      rm -rf $build_dir
      echo
      exit 1
    fi
  fi
fi

if [ -d system/bin ]
then
  rm -fv system/bin/*.lnk
fi

if [ -d system/xbin ]
then
  rm -fv system/xbin/*.lnk

```

```

fi

if [ -d META-INF ]
then
    cd META-INF
    rm -f CERT.RSA CERT.SF MANIFEST.MF
    cd ..
else
    echo
    echo "Warning: tidak di temukan META-INF folder!"
fi

cd ..
echo

if [ "$2" != "kernel_only" ] && [ "$2" != "name_given" ]
then

    scripts/check_no_update_script
    error_code=$?

    cd WORKING_*
    cd META-INF/com/google/android

    if [ -e updater-script ] && [ -e update-script ]
    then
        echo
        echo "Warning: ditemukan updater-script dan update-script."
        echo -n "Hapus updater-script (y/n)? "

        read enterRemove
        echo

        if [ "$enterRemove" == "n" ]
        then
            echo "tetap menggunakan updater-script"
            rm -rf update-script

            cd ../../../../../../..
            scripts/check_no_update_script
            error_code=$?

            cd WORKING_*
            cd META-INF/com/google/android

        else
            echo "Removing updater-script"
            rm -f updater-script
        fi
    fi

fi

if [ -e update-binary ] && [ -e update-script ]
then

```

```

echo
echo "Warning: Ditemukan update-binary dan update-script."
echo -n "Hapus update-binary (y/n)? "
read enterRemove
if [ "$enterRemove" != "n" ]
then
    echo "MEnghapus update-binary"
    rm -f update-binary
else
    echo "jangan hapus update-binary"
fi
fi

if [ -e update-script ]
then

    sed -i -e '/^#/d' update-script
fi

rm -f update-script.bak

cd ../../../../../../..

scripts/remove_recovery
fi

echo
echo "Listing of META-INF/com/google/android:"

cd WORKING_*
cd META-INF/com/google/android
ls -l
cd ../../../../../../..

cd $build_dir

echo
echo Listing of $build_dir:
ls -l

echo

if [ -d system ]
then
    echo "Selesai membuat folder!"
    cd ..

    if [ "$error_code" == "1" ]
    then
        exit 1
    else
        exit 0
    fi
fi

```

```
    fi
else
    echo "Error: gagal membuat folder"
    cd ..
    scripts/press_enter
    exit 1
fi
cd ..
```