

RANCANG BANGUN KERNEL *ANDROID* PADA *SMARTPHONE ANDROID* XIAOMI REDMI NOTE 3 SNAPDRAGON DENGAN *TOOLCHAIN* GCC-LINARO 7.4 DAN *ANYKERNEL 2*

Hendrik Sagita Pratama Putra¹, Anang Efendi², Dwi Wibowo³

^{1,2,3}Program Studi Teknik Informatika, STT POMOSDA Nganjuk

e-mail: artra.rama@gmail.com, afendystt@gmail.com, wibowo@stt-pomosda.ac.id

ABSTRAK

Perkembangan *smartphone* semakin pesat dan modern sehingga manusia tidak bisa lepas dari pemakaian *smartphone* pada aktivitas sehari-hari. Mulai dari aktivitas berbelanja, sekolah, kantor, game, hingga fotografi, dari anak kecil hingga lansia. Dengan banyaknya aktivitas yang menggunakan *smartphone*, maka perkembangan *smartphone* menjadi semakin cepat. Hal tersebut menjadikan pengembangan dan support sistem operasi hanya difokuskan pada *smartphone* keluaran terbaru. *Smartphone* yang bertipe lama seperti Xiaomi Redmi Note 3 Snapdragon tidak lagi mendapatkan support maupun pengembangan software lagi, atau disebut *end of life*. Atas dasar ini maka penelitian untuk membuat *smartphone* bertipe lama yang telah *end of life* dapat mengejar ketertinggalan sehingga performa tidak terpaut jauh dengan *smartphone* keluaran terbaru dengan membangun sebuah kernel pada *Smartphone* Xiaomi Redmi Note 3 Snapdragon yang menggunakan metode *waterfall*. Hasil yang diperoleh yaitu *Smartphone* Xiaomi Redmi Note 3 Snapdragon dapat menggunakan *OS Android 9.0 Pie* (sebelumnya menggunakan *OS Android 6.0 Marshmallow*), Setelah dilakukan *Benchmarking* pada perangkat, juga terdapat peningkatan performa, yang sebelumnya mendapatkan skor 198962 (*CPU* : 179584, *Memory* : 12779, *I/O* : 6599) menjadi 399741 (*CPU* : 345236, *Memory* : 37262, *I/O* : 17243).

Kata kunci : *Android OS, Smartphone, Kernel*

PENDAHULUAN

Latar Belakang Masalah

Berkembang pesatnya teknologi informasi dewasa ini membuat manusia tidak lepas dari penggunaan *Handphone* dalam berbagai macam kebutuhan, mulai dari kebutuhan sehari-hari maupun untuk melakukan sebuah pekerjaan. Untuk memenuhi kebutuhan itu maka dikembangkan perangkat ponsel pintar atau *Smartphone*. Seperti perangkat komputer, maka *Smartphone* membutuhkan sistem operasi agar dapat digunakan. Dewasa ini dari banyaknya sistem operasi pada perangkat *mobile*, salah satu yang paling populer adalah sistem operasi berbasis *Android*, sebuah sistem operasi *Open Source* yang dikembangkan oleh Google (Safaat,2012).

Saat ini pengguna *smartphone* pada umumnya hanya menggunakan perangkatnya tanpa mengetahui bahwa sistem operasi yang tertanam pada perangkat mereka telah *obsolete*. Sehingga banyak celah keamanan yang terbuka yang memungkinkan pihak-pihak tidak bertanggung jawab untuk memanfaatkan celah tersebut.

Xiaomi Redmi Note 3 Snapdragon merupakan sebuah perangkat *mobile* yang dikembangkan oleh Xiaomi yang diluncurkan pada Pebruari 2016, sebuah perangkat yang cukup berusia sehingga *developer* Xiaomi telah memutuskan untuk mengakhiri masa *support* perangkat ini (*end of life*).

Saat ini pengguna Xiaomi Redmi Note 3 Pro masih sangat banyak, namun dengan status perangkat yang *end of life*, maka membuat banyak user melakukan berbagai cara agar pengembangan perangkat tidak berhenti, maka dilakukan pengembangan sistem operasi

maupun *Kernel*, dengan status yang *Open Source* maka memungkinkan pengembangan menjadi cepat dan tidak membuat perangkat berstatus *obsolete*.

Peluncuran sistem operasi resmi dari pengembang Xiaomi untuk perangkat Redmi Note 3 Snapdragon berakhir pada sistem operasi *Android Marshmallow*, yaitu versi sistem operasi *Android* ke-6. Saat ini sistem operasi *Android stable* terbaru saat peneliti memulai penelitian ini adalah versi *Android Pie* yaitu versi ke-9.

Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan di atas maka dapat di rumuskan permasalahan yang akan dibahas dalam penelitian yaitu:

Bagaimana membangun sebuah *Kernel* pada perangkat Xiaomi Redmi Note 3 Snapdragon dengan *Toolchain* GCC-Linaro 7.4 dan AnyKernel2 ?

Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah untuk membangun sebuah *kernel* pada *smartphone* Xiaomi Redmi Note 3 Snapdragon dengan *Toolchain* GCC-Linaro 7.4 dan AnyKernel2.

KAJIAN PUSTAKA

Pengertian *Smartphone*

Telepon pintar (*smartphone*) adalah telepon genggam yang mempunyai kemampuan tingkat tinggi, kadang-kadang dengan fungsi yang menyerupai komputer. Belum ada standar pabrik yang menentukan arti telepon pintar. (Jati W, 2011)

Sebagian besar pengguna hanya menganggap telepon pintar adalah sebuah telepon yang bisa mereka gunakan untuk mengambil foto ataupun berselancar di dunia maya, sosial media khususnya. Bagi pengguna lainnya telepon pintar bisa mempermudah kehidupan sehari-hari mereka, bahkan bisa menjadi sumber utama pemasukan. Karena saat ini, telepon pintar bisa menyamai kemampuan sebuah komputer.

Pengertian Sistem Operasi *Android*

Sistem operasi *Android* sendiri merupakan sebuah sistem operasi *Open Source* yang dikembangkan dan diluncurkan oleh *Google inc*, yang dikhususkan untuk diaplikasikan pada teknologi *smartphone*. Akan tetapi dengan perannya sebagai sebuah *Open Source* maka *Android* sendiri telah banyak berkembang pesat hingga merambah ke penggunaan pada perangkat lainnya seperti *Android TV*, *STB* dan *Android Wear*.

Sistem *Android* adalah sebuah sistem operasi berbasis linux yang ditulis menggunakan bahasa pemrograman *java*. Sehingga dapat dikatakan bahwa komponen dasar penyusun sistem *Android* tidak jauh berbeda dengan komponen dasar penyusun sistem operasi linux. (Safaat, 2011)

Tampilan antarmuka pada *Android* didasarkan pada manipulasi langsung, menggunakan masukan sentuh yang serupa dengan tindakan di dunia nyata, misalnya menggosok (*swiping*), mengetuk (*tapping*), dan mencubit (*pinching*), untuk memanipulasi obyek di layar. Masukan pengguna direspon dengan cepat dan juga tersedia antarmuka sentuh layaknya permukaan air, seringkali menggunakan kemampuan getaran perangkat untuk memberikan umpan balik haptik kepada pengguna.

Perangkat keras internal seperti akselerometer, giroskop, dan sensor proksimitas digunakan oleh beberapa aplikasi untuk merespon tindakan pengguna, misalnya untuk menyesuaikan posisi layar dari potret ke lanskap, tergantung pada bagaimana perangkat diposisikan, atau memungkinkan pengguna untuk mengarahkan kendaraan saat bermain balapan dengan memutar perangkat sebagai simulasi kendali setir.

Android Recovery

Android Recovery merupakan sebuah antarmuka yang berfungsi untuk melakukan backup, instalasi *Android ROM*, *Kernel*, *Firmware* dan sebagainya. *Android Recovery* dapat diakses dengan menekan tombol *power* dan *volume up* secara bersamaan ketika perangkat dalam keadaan dimatikan.

Linux

Linux adalah nama yang diberikan kepada sistem operasi komputer bertipe Unix. Linux merupakan salah satu contoh hasil pengembangan perangkat lunak bebas dan sumber terbuka utama. Seperti perangkat lunak bebas dan sumber terbuka lainnya pada umumnya, kode sumber Linux dapat dimodifikasi, digunakan dan didistribusikan kembali secara bebas oleh siapa saja. (Susri, 2010)

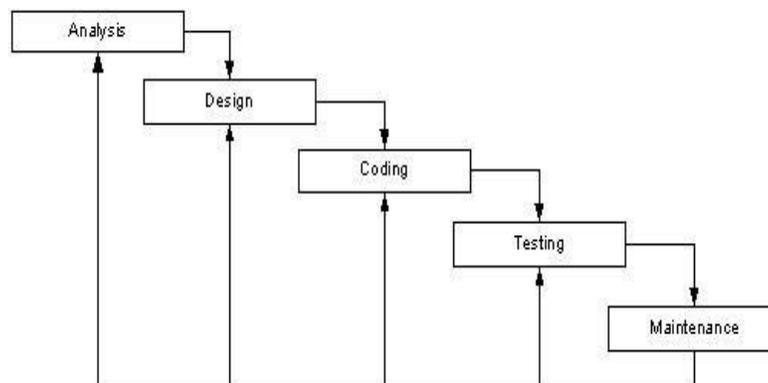
Pengertian Kernel

Dalam ilmu komputer, *Kernel* adalah suatu perangkat lunak yang menjadi bagian utama dari sebuah sistem operasi. Tugasnya melayani bermacam program aplikasi untuk mengakses perangkat keras komputer secara aman. (Stefan Brahler, 2010)

Karena akses terhadap perangkat keras terbatas, sedangkan ada lebih dari satu program yang harus dilayani dalam waktu yang bersamaan, maka *Kernel* juga bertugas untuk mengatur kapan dan berapa lama suatu program dapat menggunakan satu bagian perangkat keras tersebut. Hal tersebut dinamakan sebagai *multiplexing*.

Metode Perancangan Sistem

Dalam pengembangan sistem ini digunakan metode pengembangan *Waterfall*. Metode *Waterfall* menurut (Saputra, 2012) merupakan metode pengembangan perangkat lunak yang secara umum dilakukan oleh para peneliti sistem, dengan melalui beberapa tahapan penelitian yaitu *Analisis*, *Design*, *Coding*, *Testing* dan *Maintenance*. Metode pengembangan sistem model *Waterfall* dapat dilihat pada gambar 2.1



Gambar 2.1 Sistem Model *Waterfall*

ANALISIS DAN PERANCANGAN SISTEM

Analisis Sistem.

Analisis sistem merupakan tahap penguaraian dari suatu sistem yang utuh ke dalam bagian-bagian dengan maksud untuk mengevaluasi permasalahan sehingga dapat dibuat rancangan sistem yang baru sesuai dengan kebutuhan. Analisis sistem yang dilakukan adalah:

Suatu permasalahan pasti ada penyebabnya. Begitu pula dengan pengembangan *Kernel* Xiaomi Redmi Note 3 Snapdragon ini. Sistem ini dibuat dengan alasan antara lain:

Adanya penghentian pembaruan software produk ini yang disebabkan semakin banyaknya telepon seluler berbasis *Android* yang dikeluarkan oleh Xiaomi.

Bagaimana meningkatkan kemampuan dari Xiaomi Redmi Note 3 Snapdragon sehingga dapat mengejar kemampuan *smartphone* keluaran terbaru lainnya. Seluruh aktivitas dari hardware yang digunakan pada *smartphone Android* tergantung dari pengaturan yang dilakukan pada *Kernel Android* tersebut.

Analisis Data.

Tujuan analisis ini adalah untuk menentukan spesifikasi fungsi, kemampuan serta fasilitas dari *Kernel*. Kebutuhan yang dimaksud antara lain:

Kebutuhan kode sumber yang akan dikembangkan guna pengembangan *Kernel*. Sebuah *smartphone* tepatnya sebuah Xiaomi Redmi Note 3 Snapdragon yang akan menjadi objek penerapan.

Perlunya pengenalan yang mendalam mengenai dasar-dasar sistem linux yang merupakan dasar dari pengembangan *Kernel* ini.

Analisis Prosedur

Analisis prosedur atau proses sistem, sistem memberikan gambaran tentang sistem yang saat ini sedang berjalan. Analisis sistem bertujuan untuk mengetahui lebih jelas bagaimana cara kerja sistem tersebut, sehingga kelebihan dan kekurangan sistem dapat diketahui.

Perancangan Sistem

Perancangan sistem merupakan upaya untuk memulai membangun sistem yang akan dibuat. Perancangan ini dilakukan setelah mendapat gambaran dengan jelas apa yang harus dikerjakan. Agar lebih mudah untuk memahaminya, maka dibuatlah perancangan *Kernel Android* untuk Xiaomi Redmi Note 3 Snapdragon.

Perancangan Governor

Governor yang sesuai dengan kriteria diatas adalah *Governor Interactive* dan *Ondemand*. Kedua *Governor* ini memiliki kemampuan untuk mengatur frekuensi CPU sedemikian rupa yang kemudian menghasilkan sebuah manajemen CPU yang maksimal, sehingga cocok digunakan ketika proses multitasking ataupun gaming.

Governor sendiri pada sistem *Kernel Android* terletak pada bagian "*drivers/cpufreq/Kconfig*". Pada file *Kconfig* ini peneliti inputkan konfigurasi pemilihan *Governor Interactive* sebagai *Governor* bawaan yang diletakkan di antara *Governor* bawaan vendor.

```
config CPU_FREQ_DEFAULT_GOV_INTERACTIVE
bool "INTERACTIVE"
select CPU_FREQ_GOV_INTERACTIVE
select CPU_FREQ_GOV_PERFORMANCE
Use the CPUFreq Governor 'Interactive' as default.
config CPU_FREQ_GOV_INTERACTIVE
tristate "'Interactive' cpufreq Governor"
depends on CPU_FREQ
Interactive' - a "Interactive" optimized Governor!
```

Pada *script* diatas dapat dilihat peneliti menetapkan *Governor Interactive* sebagai *Governor* bawaan pada line yang paling bawah yaitu "Use the CPUFreq Governor 'Interactive' as default."

Ketika mengubah pengaturan *Kernel* peneliti juga diharuskan mengubah sebuah file yaitu "*drivers/cpufreq/Makefile*". File ini berisi bagian-bagian apa saja yang harus dikompilasi sehingga terbentuk file-file *Governor* yang dapat digunakan.

Script yang peneliti masukkan pada file ini adalah :

```
obj-$(CONFIG_CPU_FREQ_GOV_INTERACTIVE)+=cpufreq_Interactive.o
```

Script diatas berfungsi membuat sebuah file berekstensi “.o” yang merupakan hasil dari kompilasi kode sumber dalam bahasa C.

Setelah menetapkan konfigurasi *Governor* yang akan digunakan, maka peneliti membuat sebuah file “*cpufreq_Interactive.c*” yang berisi tentang kode sumber *Governor Interactive* yang ditulis dalam bahasa C. Peneliti meletakkannya pada bagian “*drivers/cpufreq/cpufreq_Interactive.c*”.

Selain itu diperlukan kode tambahan pada bagian “*include/linux/cpufreq.h*” yang berfungsi mengenali *Governor* yang peneliti masukkan tadi.

```
#elif defined(CONFIG_CPU_FREQ_DEFAULT_GOV_INTERACTIVE) extern struct cpufreq_Governor
cpufreq_gov_Interactive;
```

```
#define
```

```
CPUFREQ_DEFAULT_GOVERNOR (&cpufreq_gov_Interactive)
```

Script diatas peneliti masukkan pada bagian “*Cpufreq_Default*”. Pada bagian inilah yang bertugas melakukan pengenalan *Governor-Governor* yang dapat digunakan pada *Kernel*.

Compiling Kernel

Untuk proses *compiling Kernel* dilakukan pada sistem operasi linux. Akan tetapi, dibutuhkan sebuah *cross compiler* (penyilang kompiler) yang berfungsi agar mencegah terjadinya kekeliruan penetapan prosesor yang digunakan perangkat.

Pada komputer biasanya menggunakan prosesor dengan nama ARCH dan pada *Android* Xiaomi Redmi Note 3 Snapdragon menggunakan prosesor dengan nama arm64 sehingga peneliti melakukan langkah berikut :

Mengaktifkan *terminal* linux dengan klik kanan pada *root folder source code kernel*

Mendeklarasikan jenis hardware yang akan digunakan

```
Export ARCH=arm64
```

Mendeklarasikan *cross compiler* yang akan digunakan

```
Export CROSS_COMPILE=/home/kenzo/linaro/741/bin/aarch64-linux-gnu-
```

Kemudian peneliti membuat konfigurasi *hardware* dengan cara memanggil file *config* yang telah dibuat sebelumnya.

Peneliti mengetikkan perintah

```
make artra_defconfig
```

Berkas *Config* ini merupakan berkas *artra_defconfig* yang telah peneliti kembangkan.

Lalu mengetikkan perintah

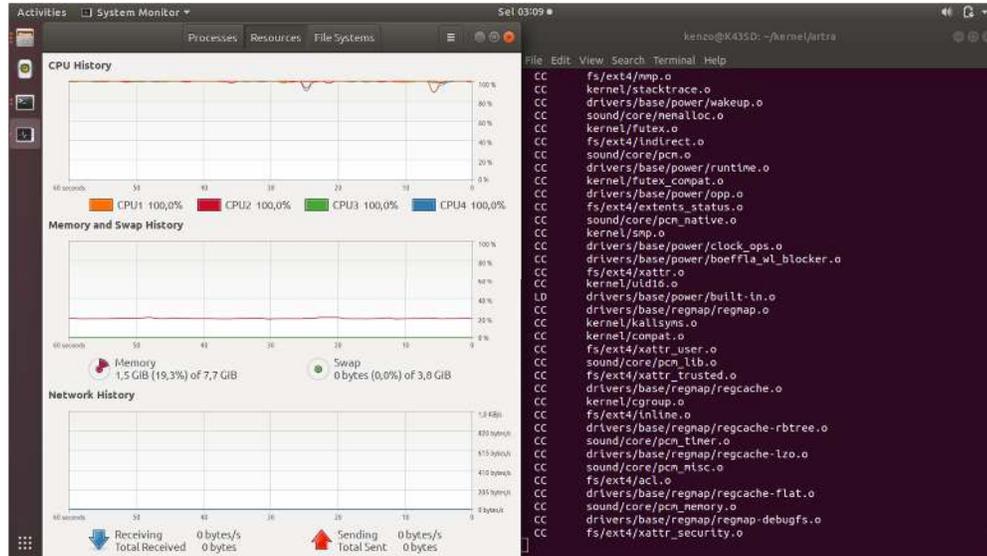
```
make -j 4
```

Perintah diatas tergantung pada jumlah inti prosesor dari komputer yang digunakan.

Karena peneliti menggunakan prosesor intel core i5 4 inti maka peneliti mengetikkan angka 4.

Menunggu hingga proses *compiling* selesai.

Hasil output terletak pada */arm64/boot/Image.gz-dtb*



Sumber : Data Diolah peneliti

Gambar 3.1 Proses Compiling Kernel

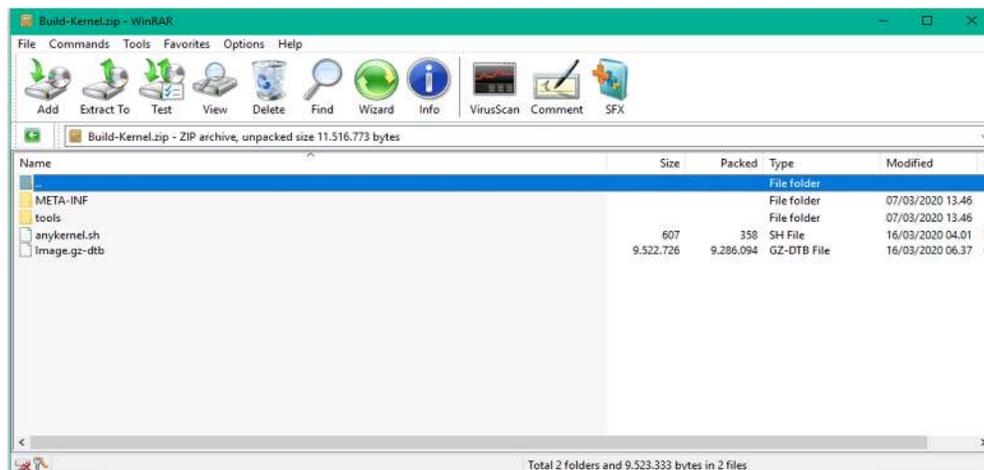
IMPLEMENTASI DAN PENGUJIAN

Implementasi Kernel

Tahap implementasi merupakan tahap inti dari pekerjaan sebuah proyek. Disinilah pembangunan komponen-komponen pokok sebuah sistem informasi dilakukan berdasarkan desain yang sudah dibuat. Implementasi sistem yang dimaksud merupakan proses pembuatan dan pemasangan sistem secara utuh, baik dari *hardware* (perangkat keras) maupun *software* (perangkat lunak).

Tampilan *Flashable Kernel*

Flashable kernel merupakan file kernel yang biasanya ber-ekstensi *zip* atau *img* yang dapat di pasang pada perangkat melalui *fastboot* maupun *recovery*

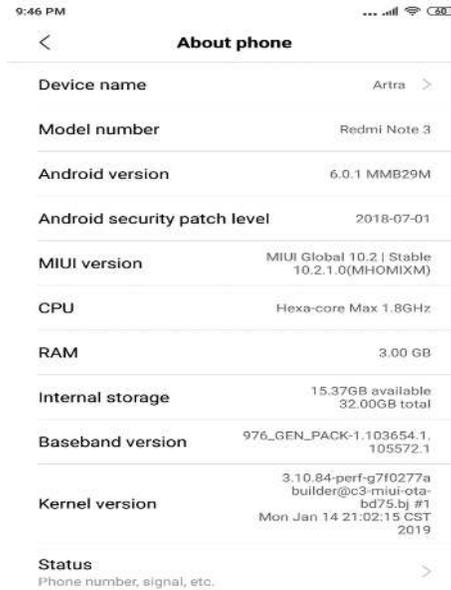


Sumber : Data Diolah Peneliti

Gambar 4.1 *Flashable Kernel*

Tampilan *Kernel Before*

Tampilan *about phone* dengan sistem operasi *official MIUI 10 Latest*, yang menggunakan *Android Marshmallow*.



Sumber : Data Diolah Peneliti
Gambar 4.2 *Kernel Official*

Flashing Kernel

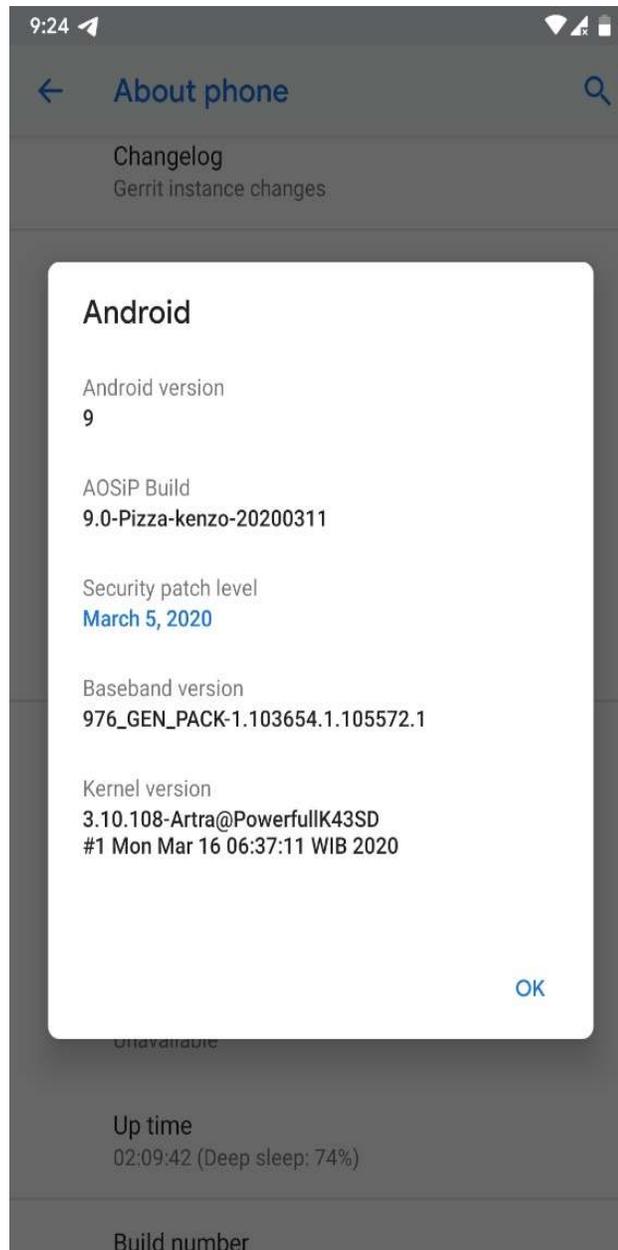
Proses pemasangan kernel atau yang disebut *flashing* dengan menggunakan *custom recovery*.



Sumber :Data Diolah Peneliti
Gambar 4.3 Proses *Flashing kernel*

Tampilan *Kernel After*

Tampilan *about phone* setelah dilakukan pemasangan kernel pada sistem operasi *custom Android Pie*.



Sumber : Data Diolah Peneliti

Gambar 4.4 *Kernel* yang dikembangkan

Pengujian *Kernel*

Pada tahap ini dilakukan pengujian fungsional kernel yang dilakukan dengan melakukan pengetesan pada setiap *hardware* yang terdapat pada *smartphone* dengan Metode *blackbox*.

Rencana Pengujian

Rencana pengujian yang akan dilakukan dengan menguji sistem secara *alpha*. berikut menggunakan data uji berdasarkan data yang telah didapat.

Tabel 4.1 Rencana pengujian

Item Uji	Detail Pengujian	Jenis Pengujian
<i>Flashing</i>	Verifikasi hasil <i>flashing</i>	<i>Black-box</i>
<i>Booting</i>	Verifikasi <i>booting</i>	<i>Black-box</i>
<i>Touchscreen</i>	Verifikasi respon	<i>Black-box</i>
<i>LCD</i>	Verifikasi tampilan	<i>Black-box</i>
<i>Audio</i>	Verifikasi suara	<i>Black-box</i>
<i>Radio</i>	Verifikasi <i>signal</i>	<i>Black-box</i>
<i>FM Radio</i>	Verifikasi fungsi	<i>Black-box</i>
<i>Sensor-sensor</i>	Verifikasi fungsi	<i>Black-box</i>
<i>USB</i>	Verifikasi data dan <i>charging</i>	<i>Black-box</i>
<i>Kamera</i>	Verifikasi fungsi	<i>Black-box</i>
<i>Keypad</i>	Verifikasi fungsi	<i>Black-box</i>
<i>LED</i>	Verifikasi fungsi	<i>Black-box</i>

Sumber :Data Diolah Peneliti

Pengujian menggunakan *benchmark* saat masih menggunakan sistem operasi dan kernel *official*, setelah menggunakan kernel yang dikembangkan, diharapkan hasil *benchmark* dapat ditingkatkan.



Sumber : Data Diolah Peneliti

Gambar 4.5 Benchmark

Kasus dan Hasil Pengujian (Pengujian Alpha)

Berdasarkan rencana pengujian yang telah disusun, maka dapat diperoleh hasil pengujian sebagai berikut :

Tabel 4.2 Hasil Pengujian

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
<i>Flashing</i>	<i>Flashing</i> sukses	Dapat dilakukan	Diterima
<i>Booting</i>	<i>Smartphone</i> dapat <i>booting</i> ke <i>homescreen</i>	Dapat dilakukan	Diterima
<i>Touchscreen</i>	<i>Touchsreen</i> berfungsi normal	Dapat dilakukan	Diterima
<i>LCD</i>	Panel menghasilkan warna yang sesuai	Dapat dilakukan	Diterima
<i>Audio</i>	<i>Voice</i> input, output, <i>jack audio</i> berfungsi	Dapat dilakukan	Diterima
<i>Radio</i>	<i>Signal</i> seluler, <i>bluetooth</i> dan data berfungsi	Dapat dilakukan	Diterima
<i>FM Radio</i>	Radio fm berfungsi	Dapat dilakukan	Diterima
<i>Sensor-sensor</i>	Sensor <i>accelerometer</i> , <i>light</i> , <i>orientation</i> , <i>proximity</i> , <i>gyroscope</i> dan <i>magnetic</i> berfungsi	Dapat dilakukan	Diterima
<i>USB</i>	Dapat melakukan transfer data dan charging	Dapat dilakukan	Dapat dilakukan
<i>Kamera</i>	Kamera belakang dan depan berfungsi	Dapat dilakukan	Diterima
<i>Keypad</i>	Tombol-tombol merespon	Dapat dilakukan	Diterima
<i>LED</i>	Dapat menyala sesuai fungsi	Dapat dilakukan	Diterima

Sumber : Data Diolah Peneliti

Hasil *benchmark* meningkat secara signifikan setelah kernel yang dikembangkan dipasangkan pada perangkat.



Sumber : Data Diolah Peneliti

Gambar 4.6 Benchmark kernel yang dikembangkan

Testimoni Testers

Berikut adalah testimoni oleh para testers yang peneliti kumpulkan menggunakan Google Forms;

Tabel 4.3 Testimoni para testers

No	Nama	Alamat	Email	Spesifikasi Perangkat	Hasil	Kesimpulan dan Saran
1.	Ahmad Fahmi	Semarang	slankri123@gmail.com	RAM 2GB ROM 16GB	Sukses	Semua berfungsi dengan lancar
2.	Satria	Tangerang	satrialvaro25@gmail.com	RAM 3GB ROM 32GB	Sukses	Semua berfungsi dengan lancar

Sumber : Data diolah peneliti

Kesimpulan Pengujian (Alpha)

Berdasarkan hasil pengujian dengan kasus sample uji dan testimoni testers yang telah dilakukan memberikan kesimpulan bahwa secara fungsional sistem sudah dapat menghasilkan output yang diharapkan.

KESIMPULAN

Berdasarkan hasil analisis rancang bangun kernel *Android* pada *smartphone* redmi note 3 snapdragon maka dapat diambil kesimpulan bahwa masalah yang terdapat pada *smartphone* redmi note 3 snapdragon dapat diselesaikan, sehingga sistem dapat berjalan lebih cepat dan tidak tertinggal dengan *smartphone* keluaran baru.

Dengan adanya pengembangan kernel maka kenyamanan user menjadi meningkat karena kinerja *smartphone* menjadi lebih baik dari sebelumnya.

DAFTAR PUSTAKA

- Bima Persada, Audio (2013). *Pembuatan dan pengembangan fitur Kernel Android pada sony xperia ray* (skripsi). Fakultas Matematika dan Ilmu Pengetahuan Alam. Jurusan Teknik Informatika. Universitas Sumatera Utara. Medan.
- Bun Seng, Ciu (2011). *Android : Dasar Pengoperasian, Optimasi Sampai Modifikasi*. Penerbit : Jasakom 2011 . ISBN:978-979-1090-53-7. Yogyakarta
- Troutman, Conan (2013). *XDA University: Adding Features to Your Freshly Compiled Kernel*. Adopted from <https://www.xda-developers.com/xda-university-adding-features-to-your-freshly-compiled-kernel/>, diakses pukul 22.51 tanggal 24 Oktober 2019.
- Tri Listyorini dan Anteng Widodo (2013). *Perancangan Mobile Learning Mata Kuliah Sistem Operasi Berbasis Android* (jurnal). Universitas Muria Kudus. Jurnal SIMETRIS Vol 3 No 1 April 2013. ISSN:2252-4983. Kudus.
- Safaat, Nazruddin (2012). *Android Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Penerbit : Informatika Bandung 2012. ISBN:978-602-1514-4-4. Bandung.
- Susrini (2010). *Creative Project LINUX untuk Pemula*. Penerbit : Grasindo 2012. ISBN:979-234800x. Jakarta.
- Jati W, Yohan (2010). *Google Android*. Penerbit : ANDI dan Elcom 2010. ISBN:978-979-2920-15-4. Yogyakarta