



TESIS - KI142502

**MAKSIMALISASI RESPONSIVITAS KONEKSI  
INTERNET PADA SMARTPHONE DENGAN ANALISIS  
POLA KEBIASAAN PENGGUNA**

FADILAH FAHRUL HARDIANSYAH  
5112201046

DOSEN PEMBIMBING  
Prof.Dr.Ir. Joko Lianto Buliali  
Waskitho Wibisono, S.Kom., M.Eng., PhD

PROGRAM MAGISTER  
BIDANG KEAHLIAN KOMPUTASI CERDAS DAN VISUALISASI  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
2015



TESIS - KI142502

**MAXIMIZING RESPONSIVENESS OF SMARTPHONE  
INTERNET CONNECTION BASED ON USER  
BEHAVIOR ANALYSIS**

FADILAH FAHRUL HARDIANSYAH  
5112201046

SUPERVISOR  
Prof.Dr.Ir. Joko Lianto Buliali  
Waskitho Wibisono, S.Kom., M.Eng., PhD

MAGISTER PROGRAMME  
INFORMATICS ENGINEERING DEPARTMENT  
FACULTY OF INFORMATION TECHNOLOGY  
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY  
SURABAYA  
2015

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar  
Magister Komputer (M.Kom)  
di  
Institut Teknologi Sepuluh Nopember

Oleh :  
FADILAH FAHRUL HARDIANSYAH  
Nrp. 5112201046

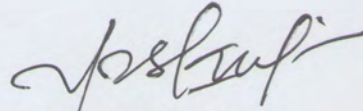
Tanggal Ujian : 16 Januari 2015  
Periode Wisuda : 2015 Gasal

Disetujui oleh :

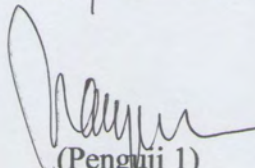
Prof.Dr.Ir. Joko Lianto Buliali, M.Sc  
NIP. 196707271992031002

  
(Pembimbing 1)

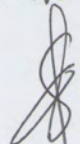
Waskitho Wibisono, S.Kom, M.Eng, Ph.D  
NIP. 197410222000031001

  
(Pembimbing 2)

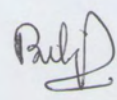
Dr. Ir. Raden Venantius Hari Ginardi, M.Sc  
NIP. 196505181992031003

  
(Penguji 1)

Darlis Herumurti, S.Kom, M.Kom  
NIP. 197712172003121001

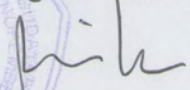
  
(Penguji 2)

Bilqis Amaliah, S.Kom, M.Kom  
NIP. 197509172001122002

  
(Penguji 3)



Direktur Program Rascasarjana,

  
Prof.Dr.Ir. Adi Soeprijanto, MT  
NIP. 196404051990021001

# MAKSIMALISASI RESPONSIVITAS KONEKSI INTERNET PADA SMARTPHONE DENGAN ANALISIS POLA KEBIASAAN PENGGUNA

Nama Mahasiswa : Fadilah Fahrul Hardiansyah  
NRP : 5112201046  
Pembimbing : 1. Prof.Dr.Ir Joko Lianto Buliali  
2. Waskitho Wibisono, S.Kom., M.Eng., PhD

## ABSTRAK

Peningkatan kemampuan *smartphone* berpengaruh terhadap peningkatan konsumsi energi pada *smartphone*. Banyak metode telah diusulkan untuk mengurangi konsumsi energi pada *smartphone*. Sebagian besar dari metode ini menggunakan kontrol koneksi Internet berdasarkan ketersediaan tingkat daya baterai terlepas dari kapan dan di mana pemborosan energi terjadi. Penelitian ini mengusulkan sebuah pendekatan baru untuk mengontrol koneksi Internet berdasarkan waktu *idle* menggunakan analisis pola perilaku pengguna. Pola perilaku pengguna digunakan untuk memprediksi durasi waktu *idle*. Kontrol koneksi Internet yang dilakukan selama waktu *idle*. Selama waktu *idle*, koneksi Internet secara berkala dinyalakan dan dimatikan dengan interval waktu tertentu. Pendekatan yang diajukan memiliki keunggulan dalam hal kenyamanan pengguna. Pemborosan energi pada saat *idle* berhasil dikurangi. Penghentian koneksi internet tidak mengganggu pengguna karena dilakukan pada saat *idle*. Selain itu tidak ada proses transfer data yang terhambat dalam waktu yang lama karena koneksi internet diehtikan dan diaktifkan secara berkala selama proses kontrol koneksi internet.

**Kata Kunci:** *Smartphone*, Kontrol Koneksi Internet, Waktu *Idle*, Pengenalan Pola, Kebiasaan Pengguna

# MAXIMIZING RESPONSIVENESS OF SMARTPHONE INTERNET CONNECTION BASED ON USER BEHAVIOR ANALYSIS

Nama Mahasiswa : Fadilah Fahrul Hardiansyah  
NRP : 5112201046  
Pembimbing : 1. Prof.Dr.Ir Joko Lianto Buliali  
2. Waskitho Wibisono, S.Kom., M.Eng., PhD

## ***ABSTRACT***

*The increase of smartphone ability is rapidly increasing the battery consumption. Many methods have been proposed to reduce smartphone battery consumption. Most of these methods use the internet connection control based on the availability of the battery power level regardless of when and where a waste of energy occurs. This research proposes a new approach to control the internet connection based on idle time using user behavior pattern analysis. User behavior pattern are used to predict idle time duration. During idle time internet connection periodically switched on and off by a certain time interval. The proposed method has advantages in term of user convenience. Battery energy wasted on idle time is reduced because of the internet connection control. Control of the internet connection does not interfere the user because it is implemented on idle time. Moreover there is no data transfer process delayed for a long time because internet connection periodically switched on and off by a certain time interval during control process.*

**Key Words:** *Smartphone, Internet Connection Control, Idle Time, Pattern Recognition, User Behavior*

## KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Dengan Mengucap puji syukur kepada Allah SWT atas segala limpahan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan tesis ini :

### **MAKSIMALISASI RESPONSIVITAS KONEKSI INTERNET PADA SMARTPHONE DENGAN ANALISIS POLA KEBIASAAN PENGGUNA**

Tesis ini disusun untuk memenuhi sebagian persyaratan guna memperoleh gelar sarjana Magister Komputer di Institut Teknologi Sepuluh Nopember. Penulis mengucapkan banyak terimakasih dan penghargaan yang sebesar-besarnya kepada semua pihak yang telah memberikan bantuan dan dukungan hingga tesis ini terselesaikan dengan baik, khususnya kepada:

1. Bapak Prof. Dr. Ir. Joko Lianto Buliali, M.Sc dan bapak Waskitho Wibisono, S.Kom., M.Eng., Ph.D yang telah membimbing penulis hingga tesis ini dapat terselesaikan dengan baik.
2. Orang tua penulis yang telah memberikan dukungan dan doa yang tiada henti. Dukungan dan doa mereka telah mengantarkan penulis dalam meraih segala yang dicita-citakan hingga saat ini.
3. Istri dan adik penulis yang telah memberikan dukungan penuh serta bantuan dan doa selama penulis menuntut ilmu di Institut Teknologi Sepuluh Nopember.
4. Bapak ibu dosen pengajar di Program Pasca Sarjana Teknik Informatika Institut Teknologi Sepuluh Nopember. Yang telah mengajarkan banyak ilmu bermanfaat.
5. Warga F-24, teman-teman Pasca FTIF angkatan 2012, dan pihak-pihak lain yang tidak dapat penulis sebutkan satu-persatu yang telah memberikan bantuan dalam penyelesaian tesis ini.

Dengan selesainya buku tesis ini, penulis berharap semoga buku ini dapat membawa manfaat bagi pembaca pada umumnya dan juga bagi penulis pada khususnya serta semua pihak yang berkepentingan. Penulis juga berharap agar tesis ini dapat dikembangkan lebih lanjut sehingga dapat benar-benar digunakan sebaik-baiknya untuk mendukung perkembangan ilmu pengetahuan. Penulis menyadari bahwa penulis adalah manusia biasa yang tidak luput dari kesalahan dan kekurangan. Untuk itu, kritik dan saran yang bersifat membangun kami harapkan untuk perbaikan selanjutnya.

Wassalamu'alaikum Wr. Wb.

Surabaya, 29 Januari 2015

Fadilah Fahrul Hardiansyah

## DAFTAR ISI

LEMBAR PENGESAHAN .....	i
ABSTRAK .....	iii
ABSTRACT .....	v
KATA PENGANTAR .....	vii
DAFTAR ISI .....	ix
DAFTAR GAMBAR .....	xi
DAFTAR TABEL .....	xiii
BAB 1 PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Perumusan Masalah .....	2
1.3 Pembatasan Masalah .....	3
1.4 Tujuan Penelitian .....	3
1.5 Manfaat dan Kontribusi Penelitian .....	3
BAB 2 DASAR TEORI DAN KAJIAN PUSTAKA .....	5
2.1 Pola Kebiasaan Pengguna .....	5
2.2 Waktu Respon <i>Smartphone</i> .....	6
2.3 <i>Smartphone Idle Time</i> .....	6
2.4 Monitoring Kinerja <i>Smartphone</i> .....	7
2.5 Koneksi Internet pada <i>Smartphone</i> .....	8
2.5.1 Kontrol Interval Transfer Paket Data .....	8
2.5.2 Sistem Cerdas untuk Penentuan <i>Offloading</i> .....	10
2.5.3 Penyeleksian Penerimaan Data .....	11
2.6 Background Services .....	12
BAB 3 METODE PENELITIAN .....	15
3.1 Tahapan Penelitian .....	15
3.2 Subjek Penelitian .....	15
3.3 Perancangan Data .....	17
3.3.1 Rancangan Data Masukan .....	17
3.3.2 Rancangan Data Keluaran .....	18
3.4 Perancangan Metode dan Algoritma .....	18
3.4.1 Tahap <i>Learning</i> Data Aktivitas <i>Smartphone</i> .....	18
3.4.1.1 Proses Pengambilan dan Penyimpanan Data .....	19
3.4.1.2 Ekstraksi Data Aktivitas <i>Smartphone</i> .....	22
3.4.1.3 <i>Learning</i> Data Hasil Ekstraksi .....	24
3.4.2 Tahap Penghematan Energi Baterai .....	25
3.4.2.1 Penentuan Kondisi <i>Idle Smartphone</i> .....	26
3.4.2.2 Proses Kontrol Koneksi Internet .....	27
3.4.2.3 Prediksi Durasi <i>Idle</i> .....	29
3.5 Skenario Uji Coba .....	31
3.5.1 Tahap <i>Learning</i> Data Aktivitas <i>Smartphone</i> .....	31
3.5.2 Tahap Kontrol Koneksi Internet .....	32
BAB 4 HASIL DAN PEMBAHASAN .....	33
4.1 Tahapan Uji Coba .....	33
4.2 Data Uji Coba .....	33

4.3	Hasil Uji Coba dan Analisis.....	37
4.3.1	Tahap <i>Learning</i> Data Aktivitas <i>Smartphone</i> .....	37
4.3.1.1	Uji Coba Pengambilan Data Aktivitas <i>Smartphone</i> .....	37
4.3.1.2	Uji Coba Ekstraksi Data Aktivitas.....	42
4.3.1.3	Uji Coba <i>Learning</i> Data Hasil Ekstraksi .....	43
4.3.2	Tahap Kontrol Koneksi Internet .....	44
4.3.2.1	Uji Coba Penentuan Kondisi <i>Idle</i> .....	45
4.3.2.2	Uji Coba Prediksi Durasi <i>Idle</i> .....	46
4.3.2.3	Uji Coba Proses Kontrol Koneksi Internet.....	47
4.3.3	Uji Coba Performa.....	50
4.3.3.1	Performa Metode Prediksi Durasi <i>Idle</i> .....	51
4.3.3.2	Performa Prediksi Durasi <i>Idle</i> Berdasarkan Pengguna Berbeda .....	53
4.3.3.3	Performa Metode KNN Berdasarkan Nilai K.....	54
4.3.3.4	Kesiapan Konektivitas Internet .....	54
4.3.3.5	Kelancaran Proses Transfer Data .....	57
4.3.3.6	Penghematan Konsumsi Energi.....	59
BAB 5	KESIMPULAN DAN SARAN.....	61
5.1	Kesimpulan .....	61
5.2	Saran .....	62
	DAFTAR PUSTAKA.....	63
	Lampiran 1. Sampel Data Awal Pengguna.....	65
	Lampiran 2. Sampel Data Hasil Ekstraksi.....	70
	Lampiran 3. Hasil Proses <i>Learning</i> .....	75
	Lampiran 4. Hasil Prediksi Durasi <i>Idle</i> .....	80
	Lampiran 5. Data Uji Coba Kesiapan Konektivitas Internet.....	85
	BIODATA PENULIS.....	87



## DAFTAR GAMBAR

Gambar 2.1 Rata-Rata Penggunaan (Kang, 2011).....	5
Gambar 2.2 Algoritma Pemberian Jeda Pengiriman Data (Hyeon, 2012).....	9
Gambar 2.3 Smart Offloading (Ayat Khairy, 2013).....	11
Gambar 2.4 Flowchart Penyeleksian Data (Kim, 2012).....	12
Gambar 3.1 Grafik Penggunaan CPU Satu Hari untuk (a) Pengguna 1, (b) Pengguna 2, (c) Pengguna 3, (d) Pengguna 4, (e) Pengguna 5 .....	16
Gambar 3.2 Tahap Learning Data Aktivitas Smartphone.....	19
Gambar 3.3 Tahapan Praproses .....	23
Gambar 3.4 Ekstraksi Data Aktivitas Smartphone .....	24
Gambar 3.5 Learning Data dengan Tiga Pendekatan Metode .....	25
Gambar 3.6 Tahap Penghematan Energi Baterai .....	26
Gambar 3.7 Kontrol Koneksi Internet .....	28
Gambar 3.8 Pencarian Durasi <i>Idle</i> dengan K-Means.....	29
Gambar 3.9 Pencarian Durasi <i>Idle</i> dengan K-nearest Neighbors .....	30
Gambar 4.1 Grafik Penggunaan CPU Dalam Waktu Satu Minggu untuk (a) Pengguna 1, (b) Pengguna 2, (c) Pengguna 3, (d) Pengguna 4, (e) Pengguna 5 ..	34
Gambar 4.2 Log Penyimpanan Data dengan Kondisi Smartphone Idle .....	41
Gambar 4.3 Log Penyimpanan Data dengan Kondisi Smartphone Aktif.....	41
Gambar 4.4 Log Penyimpanan Data dengan Kondisi Download Aktif.....	42
Gambar 4.5 Log Hasil Penentuan Kondisi Idle .....	46
Gambar 4.6 Uji Coba Ke-1 Fungsi Kontrol Koneksi Internet .....	48
Gambar 4.7 Uji Coba Ke-2 Fungsi Kontrol Koneksi Internet .....	49
Gambar 4.8 Uji Coba Ke-3 Fungsi Kontrol Koneksi Internet .....	50
Gambar 4.9 Grafik Tingkat Kesalahan Metode Prediksi Durasi Idle .....	52
Gambar 4.10 Tingkat Kesalahan Prediksi Durasi Idle.....	53
Gambar 4.11 Tingkat Kesalahan Metode KNN Berdasarkan Nilai K.....	54
Gambar 4.12 Uji Coba Kesiapan Koneksi Internet .....	56
Gambar 4.13 Grafik Uji Coba Transfer Data.....	58
Gambar 4.14 Konsumsi Energi Baterai Smartphone Saat Kondisi Idle .....	60

## DAFTAR TABEL

Tabel 4.1 Sampel Data Awal Pengguna 1 .....	42
Tabel 4.2 Hasil Penentuan Kondisi Idle Data Awal .....	43
Tabel 4.3 Hasil Penentuan Kondisi Idle Data Awal .....	43
Tabel 4.4 Centroid dan Center Cluster Pengguna 1 .....	44
Tabel 4.5 Sample Data Hasil Prediksi Durasi Idle.....	47
Tabel 4.6 Hasil Prediksi Durasi Idle pada Uji Coba Kesiapan Koneksi Internet .	55
Tabel 4.7 Hasil Uji Coba Konsumsi Energi.....	59

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Perkembangan sangat pesat dalam beberapa tahun terakhir. Kemampuan penyimpanan data, proses komputasi, Internet, dan resolusi layar terus ditingkatkan. Bahkan saat ini dilengkapi dengan berbagai sensor seperti *GPS (Global Positioning System)*, *accelerometer*, sensor cahaya dan berbagai sensor lain. Hal ini menyebabkan meningkatnya kebutuhan energi pada (Hyeon, 2012).

Koneksi Internet adalah salah satu faktor yang menyebabkan kebutuhan energi pada meningkat. Kebanyak aplikasi pada terhubung dengan Internet, seperti aplikasi jejaring sosial, permainan, portal berita, dan berbagai aplikasi lainnya. Aplikasi tersebut memiliki *service* yang selalu berjalan dibalik layar dan melakukan sinkronisasi data dengan *server*. Proses sinkronisasi data ini tentunya mengonsumsi banyak energi. Konsumsi energi ini terus berjalan sekalipun pengguna sedang tidak membutuhkan atau sedang memasuki waktu *idle* (Zhang, 2010).

Banyak pendekatan dan metode-metode yang diajukan oleh peneliti. Seperti metode *Smart Offloading* yang dilakukan dengan memprediksi konsumsi energi dan waktu eksekusi (Khairy, 2013). Metode ini menentukan apakah proses pengiriman data (*offloading*) dieksekusi atau tidak berdasarkan perkiraan konsumsi energi dan waktu eksekusinya. Metode penerimaan data secara selektif untuk menghindari energi yang terbuang karena penerimaan data yang tidak perlu (Kim, 2012). Selain itu ada pendekatan lain dengan mengontrol jeda waktu pengiriman dan penerimaan paket data saat kapasitas baterai tidak mencukupi (Hyeon, 2012). Pendekatan dan metode-metode tersebut berhasil memperpanjang masa hidup baterai.

Kebanyakan dari metode yang telah diajukan hanya mengacu pada ketersediaan energi baterai tanpa melihat kapan dan di mana letak pemborosan energi terjadi. Dengan hanya memperhatikan baterai maka saat ketersediaan energi baterai tinggi usaha pemangkasan konsumsi energi minimal. Sebaliknya saat ketersediaan energi baterai lemah maka banyak proses yang dihentikan yang

tentunya akan mengganggu kenyamanan pengguna. Selain itu ketersediaan energi baterai yang rendah bukan berarti pengguna tidak membutuhkan layanan dari perangkat.

Sebuah pendekatan baru diajukan dalam penelitian ini untuk mengontrol koneksi Internet dengan memberikan interval berdasarkan waktu *idle*. Pendekatan ini dilakukan dengan mempelajari kebiasaan pengguna untuk memprediksi berapa lama berada dalam kondisi *idle* (durasi *idle*). Durasi *idle* digunakan untuk menentukan interval waktu pada proses kontrol koneksi Internet. Kontrol koneksi Internet dilakukan dengan menghidupkan dan mematikan koneksi sesuai interval waktu yang didapatkan. Dengan pendekatan ini waktu respons bisa menyesuaikan kebutuhan pengguna dan tidak mengganggu aktivitas pengguna. Serta dapat mengurangi konsumsi energi yang tidak tepat guna.

## 1.2 Perumusan Masalah

Jika kontrol koneksi Internet dilakukan berdasarkan ketersediaan energi baterai maka ketika ketersediaan energi baterai rendah akan terjadi gangguan terkait kesiapan konektivitas Internet. Berbagai proses yang membutuhkan koneksi Internet akan terhambat karena penghentian koneksi Internet. Hal ini menjadi masalah ketika pengguna sedang membutuhkan layanan dari sedangkan ketersediaan energi baterai rendah dan konektivitas Internet sedang tidak siap. Hipotesis yang diajukan untuk mengatasi masalah tersebut adalah sebagai berikut:

“Jika kontrol koneksi Internet dilakukan dengan cara memberikan interval pada koneksi Internet berdasarkan durasi *idle* yang sesuai dengan pola kebiasaan pengguna maka pengguna tidak akan terganggu dengan kesiapan konektivitas Internet saat sedang dibutuhkan.”

Hipotesis itu memunculkan beberapa pertanyaan yang harus dijawab dalam penelitian ini. Di antaranya adalah:

1. Bagaimana memaksimalkan kesiapan konektivitas Internet dengan menggunakan analisis pola kebiasaan pengguna?
2. Bagaimana mempelajari pola kebiasaan pengguna untuk prediksi durasi *idle* ?

3. Bagaimana menentukan interval pada koneksi Internet berdasarkan durasi *idle* ?

### **1.3 Pembatasan Masalah**

Dari perumusan masalah yang terpapar pada sub-bab 1.2 diperoleh gambaran permasalahan yang cukup luas. Menyadari adanya keterbatasan waktu dan kemampuan, maka perlu adanya batasan masalah secara jelas dan terfokus. Berikut adalah batasan masalah dalam penelitian ini:

1. Data aktivitas pengguna direkam secara *realtime* pada selama satu minggu.
2. Pengujian dilakukan terhadap 5 orang pengguna dengan rutinitas keseharian berbeda. Penjelasan detail pengguna dijelaskan pada sub-bab 3.2.
3. Pengujian menggunakan perangkat *smartphone* dengan sistem operasi Android.

### **1.4 Tujuan Penelitian**

Berdasarkan perumusan masalah yang tertulis pada sub-bab 1.2, penelitian ini memunculkan beberapa tujuan yang hendak dicapai:

1. Melakukan kontrol koneksi Internet untuk memaksimalkan kesiapan konektivitas Internet pada .
2. Melakukan kontrol koneksi Internet dengan pemberian interval terhadap koneksi Internet pada saat dalam kondisi *idle*.
3. Mempelajari pola kebiasaan pengguna untuk memprediksi durasi *idle*.

### **1.5 Manfaat dan Kontribusi Penelitian**

Kontribusi dari penelitian ini adalah memberikan sebuah pendekatan baru untuk kontrol koneksi Internet yang dilakukan pada saat *idle* dengan pemberian interval berdasarkan durasi *idle* sesuai dengan pola kebiasaan pengguna. Dengan pendekatan ini pengguna tidak akan terganggu dengan kesiapan konektivitas Internet yang terhambat saat sedang dibutuhkan.

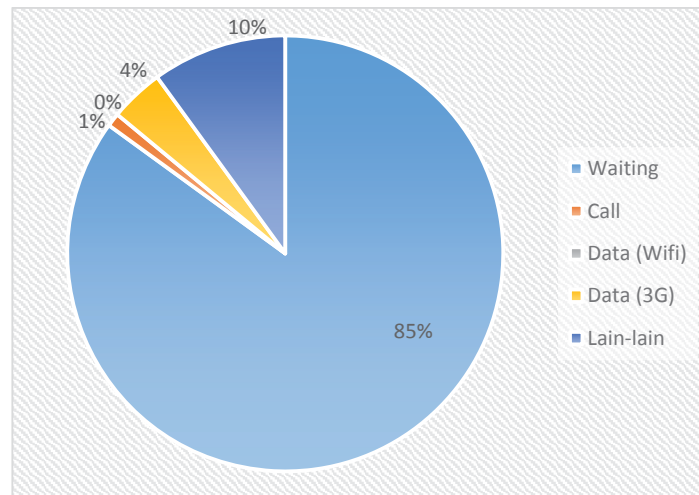
[Halaman ini sengaja dikosongkan]

## BAB 2

### DASAR TEORI DAN KAJIAN PUSTAKA

#### 2.1 Pola Kebiasaan Pengguna

Setiap pengguna pasti memiliki kebiasaan dalam penggunaan perangkat . Kebiasaan yang dimaksud di sini adalah kebiasaan pengguna untuk *idle* atau aktif dalam menggunakan perangkat yang biasanya bergantung pada waktu dan aktivitas pengguna di dunia nyata. Kebiasaan pengguna ini bisa dibagi dalam beberapa kondisi berdasarkan aktif atau tidaknya pengguna menggunakan perangkat. Berdasarkan hasil penelitian Kang (2011) keadaan atau kondisi yang paling sering dilakukan pengguna adalah kondisi *idle* dan *standby*. Pada kondisi ini Perangkat masih mengonsumsi energi (Zhang, 2010). Konsumsi energi saat *idle* ini bisa dikatakan percuma karena sebenarnya pengguna sedang tidak membutuhkan layanan.



Gambar 2.1 Rata-Rata Penggunaan (Kang, 2011)

Kita bisa membaca pola dari kebiasaan pengguna ini dengan berbagai cara dan parameter. Lee (2008) membaca pola pengguna dari kebiasaan perpindahan atau pergerakannya. Status perangkat seperti status layar, status *voice call*, status komunikasi data 3D, *network* dan status baterai juga bisa digunakan sebagai parameter untuk membaca pola kebiasaan pengguna (Kang, 2011).

## 2.2 Waktu Respon *Smartphone*

Waktu respons dapat didefinisikan sebagai kesiapan konektivitas Internet pada perangkat *smartphone*. Yaitu kondisi dimana koneksi Internet pada *smartphone* telah siap dan transfer data dapat dilakukan.

Banyak aplikasi penghemat daya baterai menggunakan metode kontrol koneksi Internet untuk menghemat daya baterai. Kontrol ini dilakukan dengan menghentikan dan mengaktifkan koneksi Internet sesuai aturan yang digunakan oleh masing-masing aplikasi. Namun model kontrol seperti ini dapat menimbulkan masalah saat penghentian koneksi Internet terjadi pada saat pengguna sedang aktif dan membutuhkan layanan dari perangkat *smartphone*.

Penelitian ini melakukan optimasi kesiapan konektivitas Internet pada *smartphone*. Pada penelitian ini dicari cara untuk melakukan kontrol koneksi Internet tanpa mengganggu pengguna dengan adanya penghentian koneksi Internet. Untuk itu kami mencoba melakukan kontrol koneksi Internet berdasarkan waktu *idle* pengguna dimana pengguna tidak membutuhkan layanan dari perangkat *smartphone*. Dengan melakukan kontrol koneksi Internet pada saat *idle* pengguna tidak akan terganggu dengan adanya penghentian koneksi Internet.

Pada penelitian ini juga digunakan interval untuk kontrol yang dilakukan. Proses penghentian dan pengaktifan koneksi Internet dilakukan dengan interval waktu tertentu. Hal ini ditujukan untuk mengoptimalkan kesiapan konektivitas Internet. Dengan cara ini koneksi koneksi Internet tetap dapat terjadi secara berkala meskipun perangkat memasuki kondisi *idle*.

## 2.3 *Smartphone Idle Time*

*Idle Time* pada diartikan sebagai kondisi di mana layar mati. Pada kondisi ini tetap menjalankan berbagai layanan yang dimilikinya mulai dari Internet, sistem, dan berbagai proses yang berjalan di balik layar. Proses dan layanan ini tentunya akan mengonsumsi banyak daya dan bisa dikatakan penggunaan daya ini kurang tepat guna atau tidak efisien.

Telah banyak peneliti melakukan uji coba untuk menghentikan konsumsi energi pada saat *idle time* ini. Cara yang paling banyak digunakan adalah dengan menghentikan berbagai proses yang berjalan di balik layar pada saat *idle time* dan



mengaktifkan kembali pada saat layar menyala. Cara ini sangat efektif untuk mengurangi konsumsi energi pada saat *idle time*.

Namun cara ini memiliki kelemahan dalam hal waktu respons perangkat untuk mengaktifkan kembali layanan-layanan yang dimatikan. Untuk mengaktifkan kembali layanan hingga terhubung dengan Internet butuh waktu yang cukup lama dan hal ini bisa mengganggu kenyamanan pengguna. Selain itu pada waktu *idle* ini bukan berarti pengguna tidak membutuhkan layanan dari perangkat. Bisa jadi pengguna sedang melakukan *download*, atau sedang menunggu pesan penting. Jika semua layanan dimatikan maka dapat mengganggu pengguna.

Dalam penelitian ini penulis mencoba mendefinisikan *idle time* bukan dari mati atau tidaknya layar namun dari aktivitas pengguna. Penulis mendefinisikan *idle time* sebagai kondisi di mana pengguna benar-benar tidak membutuhkan layanan apapun dari perangkat. Misalnya pengguna yang sedang tidur, bekerja yang tidak memungkinkan untuk menggunakan , dan aktivitas - aktivitas lain di mana pengguna tidak menggunakan . Kondisi pengguna ini akan diramalkan oleh sistem dengan mempelajari kebiasaan pengguna dalam menggunakan sehari-hari. Sistem akan menentukan apakah pengguna sedang menggunakan layanan dari atau tidak. Jika tidak berarti pengguna sedang memasuki kondisi *idle*.

#### **2.4 Monitoring Kinerja Smartphone**

Untuk mengenali aktivitas pengguna penulis melakukan monitoring terhadap kinerja . Ada beberapa aspek yang di monitor untuk mengenali aktivitas pengguna, di antaranya adalah CPU *usage*, transfer data dan konsumsi energi.

Salah satu aspek yang digunakan untuk menentukan aktivitas pengguna adalah CPU *usage*. Yang dimaksud dengan CPU *usage* di sini adalah seberapa besar penggunaan CPU untuk melakukan proses. Dari CPU *usage* ini dapat dilihat apakah pengguna sedang aktif atau tidak. Saat pengguna sedang aktif tentunya proses yang berjalan pada CPU akan tinggi sehingga penggunaan CPU tinggi sebaliknya saat pengguna sedang tidak menggunakan maka penggunaan CPU akan rendah. Jika penggunaan CPU ini terus di monitor dalam beberapa waktu maka kita dapat melihat pola dari aktivitas pengguna dalam menggunakan .

Aspek lain yang kami gunakan adalah *traffic* Internet atau lalu lintas pengiriman data. Dari *traffic* Internet kita juga dapat mengenali aktivitas pengguna. Saat pengguna sedang aktif menggunakan aplikasi yang berhubungan dengan Internet (misal aplikasi sosial media) tentunya *traffic* Internet akan tinggi. Namun saat pengguna sedang tidak menggunakan aplikasi tersebut maka *traffic* akan menurun. Namun kita tidak bisa benar-benar mengetahui aktivitas dari pengguna hanya dengan memonitor *traffic* Internet saja. Karena bisa jadi pengguna sedang aktif menggunakan namun tidak menggunakan aplikasi yang berhubungan dengan Internet.

Aspek selanjutnya yang dapat menjadi parameter untuk menentukan aktivitas pengguna adalah konsumsi energi. Saat pengguna sedang aktif menggunakan tentunya konsumsi energi akan meningkat. Sebaliknya saat pengguna sedang tidak menggunakan maka konsumsi energi akan lebih rendah.

## **2.5 Koneksi Internet pada *Smartphone***

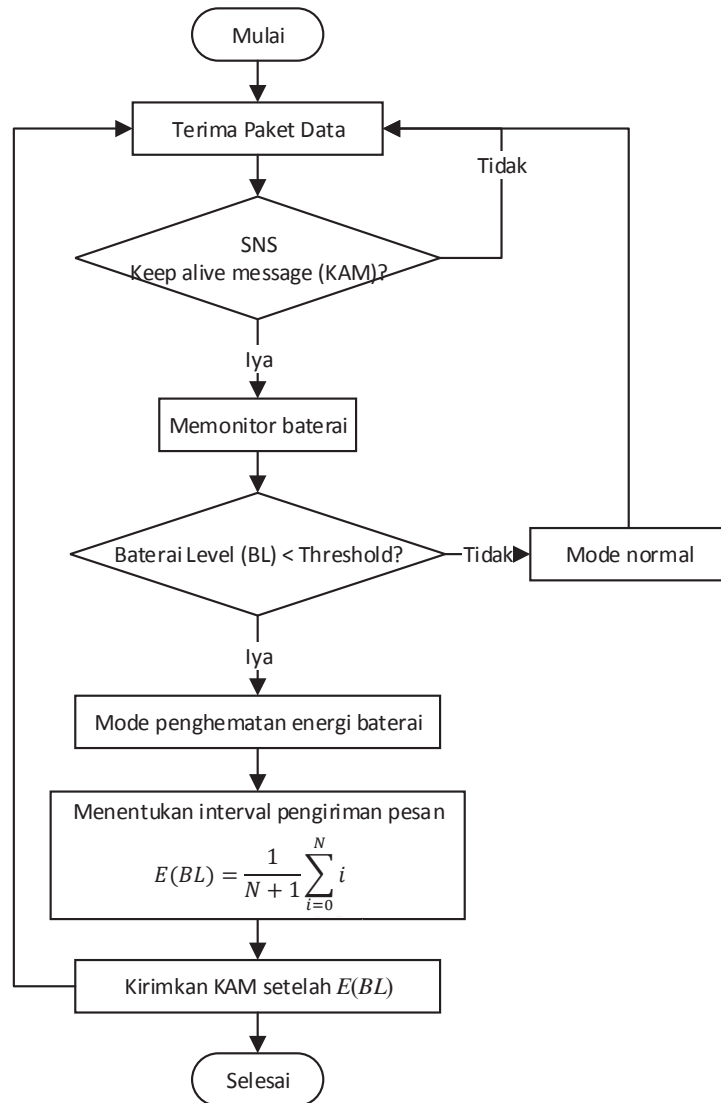
Saat ini telah dilengkapi dengan koneksi Internet berupa *mobile Internet* ataupun *wifi*. Untuk dapat menggunakan koneksi Internet biasanya sistem operasi telah menyediakan API tersendiri. Dengan API ini memungkinkan untuk setiap aplikasi untuk menggunakan koneksi Internet.

Koneksi Internet ini merupakan salah satu layanan yang mengonsumsi banyak energi. Banyak peneliti melakukan penelitian untuk efisiensi energi pada dengan melakukan kontrol pada koneksi Internet. Hyeon (2012) memberikan interval pengiriman paket data untuk memperpanjang waktu hidup baterai. Kim (2012) melakukan penyeleksian paket data yang akan diterima berdasarkan ketersediaan energi baterai.

### **2.5.1 Kontrol Interval Transfer Paket Data**

Metode ini diajukan oleh Yang Hyeon (2012). Ide dari metode ini adalah memberikan interval atau jeda untuk pengiriman sinyal berdasarkan ketersediaan energi baterai. Metode ini diterapkan pada aplikasi yang memiliki *SNS (Social Network Services)*. Umumnya aplikasi *SNS* secara periodik mengirimkan sinyal pada *server*. Metode ini digunakan untuk memberikan jeda pada pengiriman sinyal tersebut.

Pemberian jeda dilakukan dengan melihat kapasitas baterai. Semakin rendah kapasitas baterai maka semakin lama interval yang diberikan. Dengan demikian konsumsi energi untuk pengiriman sinyal tersebut juga terhambat dan otomatis masa hidup baterai akan lebih lama.



Gambar 2.2 Algoritma Pemberian Jeda Pengiriman Data (Hyeon, 2012)

Gambar 2.3 menunjukkan alur dari metode *signaling interval control*. Dalam *flowchart* tersebut terlihat bahwa metode ini fokus terhadap *battery level* dan diberikan *threshold* awal yang bernilai 50%. Jadi ketika ketersediaan energi baterai berada di atas 50% maka akan berjalan pada mode normal. Ketika

ketersediaan energi baterai di bawah 50% maka akan masuk pada *battery saving mode*.

Metode ini sangat efektif untuk memperpanjang masa hidup baterai. Namun karena metode ini hanya mengacu pada ketersediaan energi baterai maka penggunaan energi masih belum efisien. Saat ketersediaan energi baterai berada di atas 50% dan diberikan jeda untuk pengiriman sinyal, bisa jadi saat ini pengguna sedang membutuhkan layanan dari aplikasi *SNS*. Namun karena mendapatkan jeda aplikasi berjalan dengan *delay*. Sebaliknya saat ketersediaan energi baterai berada di atas 50% bisa jadi *user* sedang tidak menggunakan atau memerlukan layanan dari aplikasi *SNS*. Namun pengiriman sinyal berjalan normal.

Dengan demikian dapat disimpulkan bahwa metode ini sangat baik untuk memperpanjang masa hidup baterai. Namun tidak cukup baik dalam menggunakan energi secara efisien.

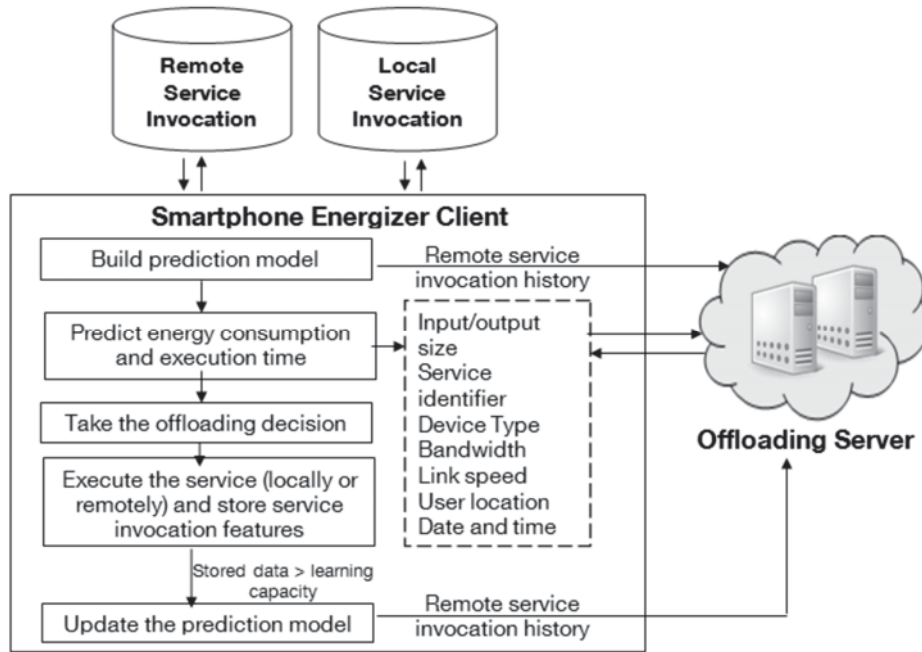
### **2.5.2 Sistem Cerdas untuk Penentuan *Offloading***

Ayat Khairy (2013) mengajukan *Smart Offloading* untuk efisiensi energi pada . Yang dimaksud dengan *Offloading* di sini adalah serangkaian proses untuk transfer data mulai dari *request* hingga *receive* data. Ayat Khairy membuat sebuah sistem cerdas untuk menentukan apakah sebuah *offloading* boleh dieksekusi atau tidak.

Prosesnya dimulai dengan melakukan prediksi untuk energi yang di konsumsi dan waktu untuk mengeksekusi *offloading* tersebut. Beberapa data seperti *bandwidth*, waktu, ukuran data, dan lain-lain digunakan untuk melakukan prediksi ini. Data-data ini dikirimkan ke *server* dan akan diolah oleh *server* untuk memprediksi konsumsi energi dan waktu eksekusinya. Kemudian hasil dikembalikan pada *client* untuk selanjutnya dilakukan pengambilan keputusan apakah *offloading* dieksekusi atau tidak.

Gambar 2.1 menunjukkan proses prediksi energi yang di konsumsi dan waktu eksekusi. Ketika memasuki *prediction mode* maka *client* akan mengirimkan data pada *offloading server*. Kemudian *server* mengembalikan hasil dari prediksi konsumsi energi dan waktu eksekusi kepada *client*. Setelah *client* mendapatkan

hasil prediksi maka *client* masuk pada *decision mode* untuk mengambil keputusan apakah *offloading* dieksekusi atau tidak.



Gambar 2.3 *Smart Offloading* (Ayat Khairy, 2013)

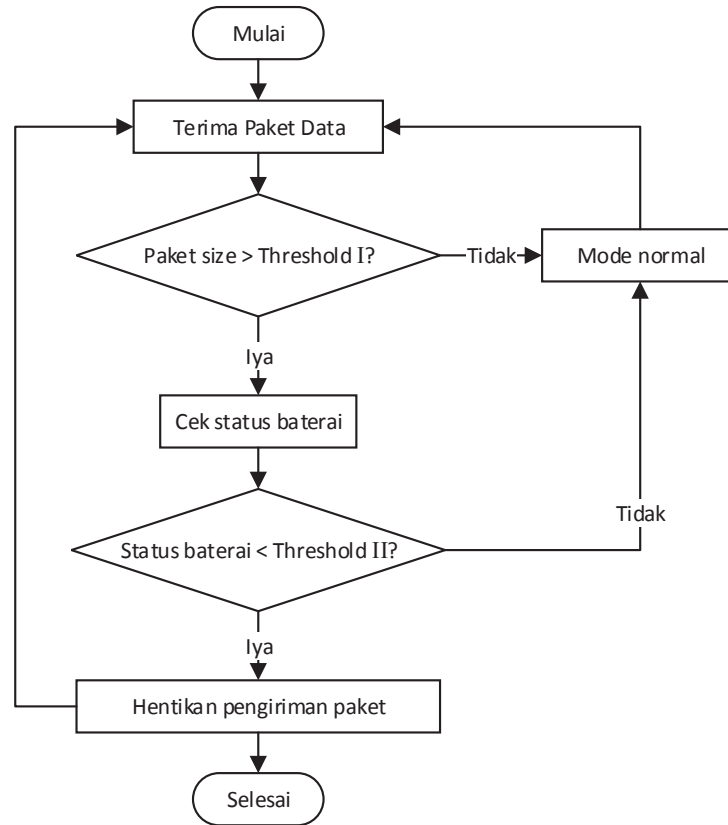
Metode ini akan sangat efektif jika menemukan *offloading* dengan konsumsi energi yang besar dan waktu eksekusi lama. Terlebih lagi kapasitas baterai tidak mencukupi untuk mengeksekusinya. Maka sistem ini dapat segera memutuskan untuk tidak mengeksekusi *offloading*. Sehingga konsumsi energi berlebih yang dapat menguras energi baterai dapat dihindari.

Namun jika menemui sebuah *offloading* dengan konsumsi energi kecil dan waktu eksekusinya juga relatif singkat, maka metode ini kurang efektif. Ketika menemui kasus seperti ini maka *client* dipaksa untuk melakukan pengiriman data. Pertama adalah mengirim data untuk prediksi. Kedua adalah pengiriman *request* untuk *offloading* itu sendiri. Sehingga *client* harus dua kali membuang energi dan waktu.

### 2.5.3 Penyeleksian Penerimaan Data

Metode lain diusulkan oleh Min Woo Kim (2012) untuk menangani permasalahan efisiensi energi. Beliau mengajukan sebuah pendekatan dengan melakukan penyeleksian terhadap data yang akan diterima oleh *client*. Ketika *client*

menerima sebuah paket, maka akan di lakukan pengecekan ukuran paket tersebut. Jika ukuran paket kurang dari *threshold* maka akan berjalan pada mode normal. Jika melebihi *threshold* maka akan dilakukan pengecekan kapasitas baterai saat ini. Jika ternyata kapasitas baterai lemah maka Transmisi paket digagalkan. Gambar 2.2 menjelaskan proses tersebut.



Gambar 2.4 Flowchart Penyeleksian Data (Kim, 2012)

Metode ini sangat efektif untuk memperpanjang waktu hidup baterai saat baterai ketersediaan energi baterai sedang kritis. Namun saat energi baterai melebihi *threshold* yang ditentukan maka tidak akan ada perlakuan khusus yang terjadi. Semuanya hanya akan berjalan seperti biasa.

## 2.6 Background Services

Sistem operasi pada memiliki berbagai layanan yang berjalan di balik layar yang biasa disebut *background services*. Layanan ini tetap berjalan ketika kita sedang menggunakan aplikasi ataupun memasuki waktu *idle* dan *standby*.

*Background services* ini memungkinkan untuk membuat komponen aplikasi yang tidak terlihat yang dapat melakukan proses komputasi tanpa memerlukan aksi langsung dari pengguna.

Layanan ini biasa digunakan oleh berbagai aplikasi untuk memonitor aplikasi, *update* ataupun data pada *server*. Contoh dari aplikasi yang menggunakan *background services* adalah aplikasi-aplikasi sosial media. Aplikasi sosial media pasti memiliki *background services* yang terus aktif untuk melakukan sinkronisasi data dengan *server*. Biasanya *services* ini akan terus berjalan sekalipun memasuki waktu *idle*. Hal inilah yang menjadi perhatian bagi banyak peneliti berkaitan dengan efisiensi baterai pada .

[Halaman ini sengaja di kosongkan]



## **BAB 3**

### **METODE PENELITIAN**

#### **3.1 Tahapan Penelitian**

Penelitian ini dilakukan dalam beberapa tahapan. Adapun tahapan yang dilakukan dalam penelitian ini adalah sebagai berikut:

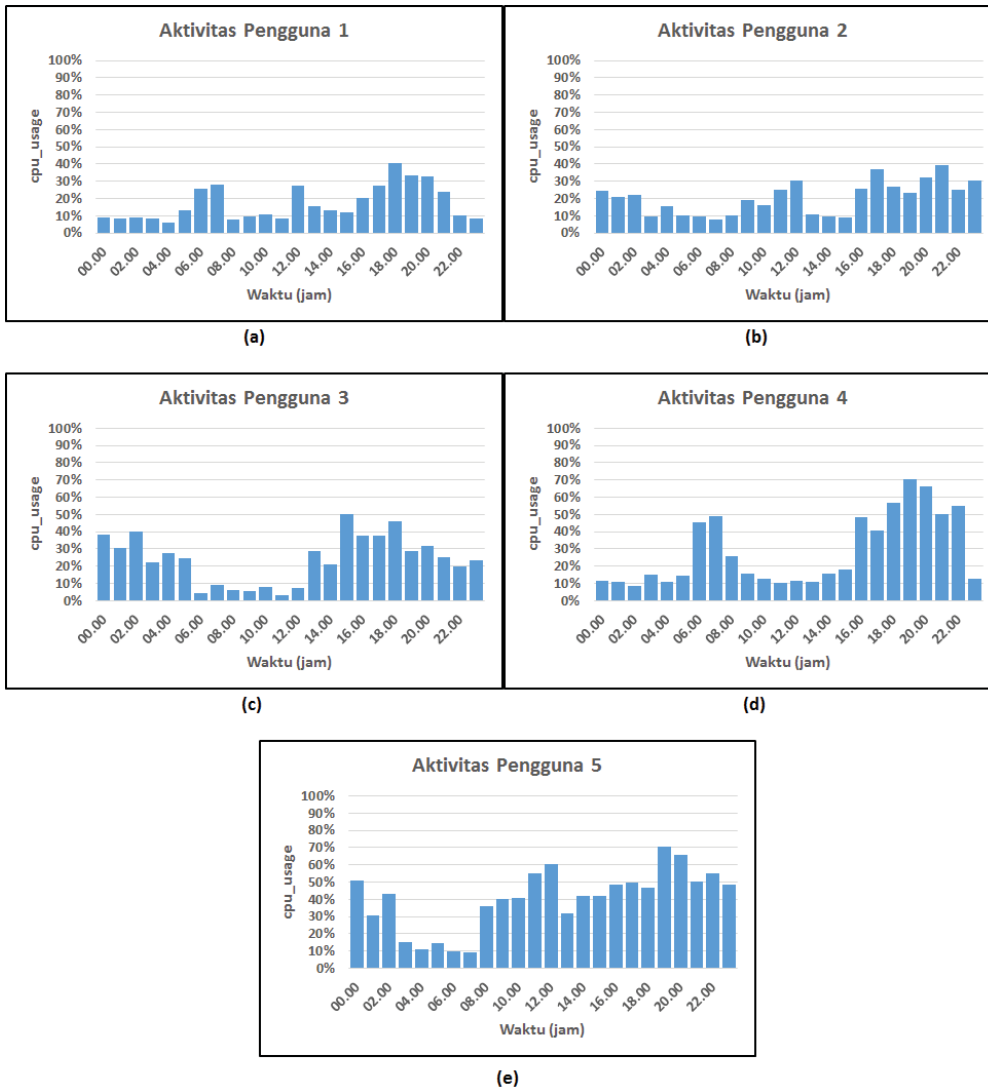
1. Identifikasi dan Perumusan Masalah
2. Studi Literatur
3. Perancangan Metode Dan Algoritma
4. Pengujian
5. Analisis Hasil Pengujian
6. Penyusunan Tesis

Identifikasi dan perumusan masalah telah diuraikan pada bab 1. Pada bab tersebut telah dijelaskan latar belakang dilakukannya penelitian ini, perumusan dan pembatasan masalah, serta tujuan dan kontribusi penelitian. Studi literatur diuraikan pada bab 2. Dalam tahap studi literatur dilakukan pengkajian terhadap beberapa referensi yang sesuai dengan topik penelitian ini, yang berguna untuk memecahkan permasalahan yang sedang diteliti. Referensi yang digunakan berupa buku, jurnal, dan artikel.

Perancangan metode dan algoritma dijelaskan pada bab 3. Perancangan metode dan algoritma menjelaskan rancangan data, perancangan metode, dan rencana uji coba yang digunakan untuk menyelesaikan permasalahan dalam penelitian ini. Langkah selanjutnya adalah pengujian dan analisis hasil pengujian yang dijelaskan pada bab 4. Pada bab tersebut dijelaskan proses uji coba, hasil uji coba, dan analisis hasil uji coba.

#### **3.2 Subjek Penelitian**

Subjek penelitian adalah sumber data dan informasi yang digunakan dalam penelitian ini. Adapun subjek dalam penelitian ini adalah lima orang pengguna dengan kebiasaan berbeda. Kebiasaan pengguna dilihat dari aktivitas penggunaan tiap hari. Aktivitas penggunaan diamati dari persentase CPU *usage*. Gambar 3.1 adalah grafik aktivitas tiap pengguna dalam satu hari.



Gambar 3.1 Grafik Penggunaan CPU Satu Hari untuk (a) Pengguna 1, (b) Pengguna 2, (c) Pengguna 3, (d) Pengguna 4, (e) Pengguna 5

Gambar 3.1 adalah grafik penggunaan CPU tiap pengguna yang menunjukkan aktivitas penggunaan perangkat oleh pengguna. Persentase CPU rendah menunjukkan bahwa pada waktu itu adalah waktu *idle* pengguna. Persentase CPU tinggi menunjukkan waktu aktif pengguna. Dari tinggi rendahnya persentase CPU dapat diamati waktu aktif dan *idle* pengguna.

Pengguna satu memiliki intensitas penggunaan yang rendah. Hal ini dilihat dari rata-rata persentase CPU yang rendah. Durasi waktu *idle* dalam hari itu juga tergolong tinggi, terlihat dari banyaknya waktu dengan persentase CPU

rendah. Pengguna dua memiliki intensitas penggunaan yang rendah, terlihat dari rata-rata persentase CPU yang rendah. Namun durasi waktu *idle* dalam satu hari lebih jarang dan kecil dibandingkan pengguna satu. Pengguna tiga memiliki intensitas penggunaan yang cukup tinggi terlihat dari rata-rata persentase CPU yang tinggi. Pengguna empat dan lima memiliki waktu-waktu aktif dengan persentase CPU yang sangat tinggi. Hal ini menunjukkan pengguna empat dan lima lebih aktif menggunakan perangkat dari pada pengguna satu, dua, dan tiga. Perbedaan antara pengguna empat dan lima adalah pengguna empat memiliki durasi *idle* yang lebih lama dibandingkan pengguna lima.

### **3.3 Perancangan Data**

Pada sub-bab ini dijelaskan rancangan data yang digunakan dalam penelitian. Rancangan data terdiri atas rancangan data masukan dan rancangan data keluaran. Data masukan adalah data yang akan diolah dengan metode yang diajukan untuk menghasilkan data keluaran.

#### **3.3.1 Rancangan Data Masukan**

Data masukan diambil secara langsung dari perangkat *smartphone* masing-masing pengguna. Ada beberapa data yang diambil dari pengguna sebagai data masukan:

1. Waktu (menit)
2. CPU *usage* (%)
3. *Download* (bytes)
4. *Upload* (bytes)
5. Status *Download*
6. Baterai (%)
7. Status layar
8. Status *charge*
9. *Latitude*
10. *Longitude*

Seluruh data tersebut digunakan untuk mempelajari pola kebiasaan pengguna. Aktivitas penggunaan perangkat *smartphone* dapat dilihat dari data nomor 2 hingga 8. Sedangkan data waktu (1), *latitude* (9) dan *longitude* (10) adalah salah

satu faktor yang mempengaruhi aktivitas pengguna. Jadi data tersebut kami gunakan karena memiliki hubungan dengan aktivitas pengguna.

### **3.3.2 Rancangan Data Keluaran**

Data keluaran adalah hasil dari pengolahan data masukan oleh sistem dengan metode yang diajukan dalam penelitian ini. Ada beberapa data luaran yang dihasilkan oleh sistem:

1. Durasi *idle* (menit)
2. Interval waktu (menit)

Hasil luaran dari sistem adalah durasi *idle* dan interval waktu. Durasi *idle* adalah hasil prediksi dari sistem. Kontrol koneksi Internet dilakukan selama durasi *idle* ini. Kontrol koneksi Internet dilakukan dengan mematikan dan menghidupkan koneksi Internet dalam interval waktu yang didapatkan dari hasil luaran sistem.

### **3.4 Perancangan Metode dan Algoritma**

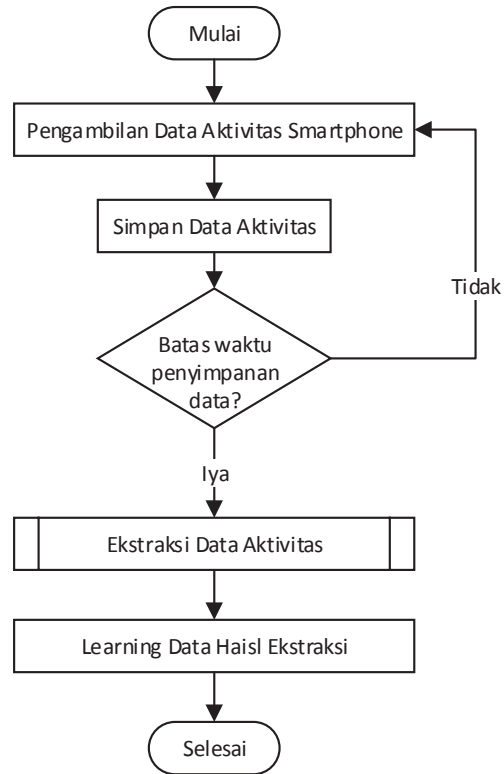
Sebuah pendekatan baru diajukan dalam penelitian ini untuk mengontrol koneksi Internet dengan memberikan interval berdasarkan waktu *idle*. Tujuan utama dari metode yang diajukan adalah melakukan kontrol koneksi Internet untuk memaksimalkan kesiapan konektivitas Internet. Kontrol koneksi Internet dilakukan dengan menghentikan dan mengaktifkan kembali koneksi Internet secara berulang dengan pemberian interval terhadap koneksi Internet pada saat dalam kondisi *idle*. Interval didapatkan dari hasil perhitungan durasi *idle*. Untuk menghitung durasi *idle*, sistem terlebih dahulu melakukan pembelajaran pola kebiasaan pengguna.

Ada dua tahapan utama yang dilakukan oleh sistem yang diusulkan dalam penelitian ini, yaitu tahapan pembelajaran pola kebiasaan pengguna dan tahap penghematan energi baterai. Dalam perancangan metode ini dijelaskan rancangan dari proses-proses yang ada dalam sistem. Rancangan yang dijelaskan berkaitan dengan dua tahapan utama dan proses-proses yang ada dalam sistem yang diusulkan.

#### **3.4.1 Tahap *Learning Data* Aktivitas *Smartphone***

Tahapan *learning* data aktivitas *smartphone* adalah tahapan awal yang dilakukan oleh sistem yang diusulkan. Tahapan ini bertujuan untuk mempelajari

pola kebiasaan pengguna dalam menggunakan *smartphone*. Gambar 3.2 menunjukkan alur proses pembelajaran data aktivitas *smartphone*.



Gambar 3.2 Tahap *Learning* Data Aktivitas *Smartphone*

Tahap ini berlangsung selama tujuh hari. Pada tahap ini sistem melakukan pengambilan dan penyimpanan data aktivitas *smartphone*. Data diambil secara langsung dari perangkat *smartphone* pengguna dan disimpan dalam sebuah *database* SQLite. Pengambilan dan penyimpanan data dilakukan dengan interval satu menit. Setelah data aktivitas *smartphone* terkumpul selama tujuh hari, kemudian sistem melakukan *learning* terhadap data yang telah didapat. Tujuan dari proses *learning* ini adalah untuk mendapatkan pola kebiasaan pengguna dalam menggunakan *smartphone*. Hasil dari proses *learning* ini digunakan untuk memprediksi durasi *idle* dalam tahap penghematan energi baterai. Berikut adalah perancangan untuk setiap proses dalam tahapan *learning* data aktivitas *smartphone*.

#### 3.4.1.1 Proses Pengambilan dan Penyimpanan Data

Proses pengambilan dan penyimpanan data dilakukan selama 7 hari. Data diambil secara langsung dari pengguna. Interval pengambilan data adalah 1 menit.

Kemudian data disimpan dalam sebuah *database* SQLite. Ada sepuluh data yang dibaca dan disimpan, yaitu waktu, CPU *usage*, *download*, *upload*, status *download*, baterai, status layar, status *charge*, *latitude*, dan *longitude*. Data direkam secara *realtime* setiap satu menit. Berikut ini adalah penjelasan proses pengambilan data untuk masing-masing atribut.

### 1. Waktu

Atribut waktu adalah waktu penyimpanan data dalam satuan menit. Waktu diambil dari sistem pada perangkat . Untuk mendapatkan data waktu ini digunakan Android API Calendar. Dari API tersebut didapatkan data jam dan menit saat ini yang kemudian disimpan dalam satuan menit. Berikut adalah potongan kode untuk pengambilan data waktu:

```
Calendar calendar = Calendar.getInstance();
time = ( c.get(Calendar.HOUR_OF_DAY) * 60 ) +
        c.get(Calendar.MINUTE);
```

Pada potongan kode diatas terlihat data jam dikalikan 60 dan ditambahkan dengan data menit. Dari rumusan tersebut didapatkan waktu dalam satuan menit. Dalam satu hari ada 1440 menit. Jadi rentang nilai untuk atribut waktu adalah 1 sampai dengan 1440.

### 2. CPU

CPU *usage* adalah persentase penggunaan CPU pada perangkat *smartphone*. Data persentase penggunaan CPU dibaca dari file `/proc/stat` pada sistem. File `/proc/stat` berisi informasi aktivitas kernel. Dari file ini dapat diketahui penggunaan CPU dalam interval waktu tertentu.

### 3. *Download* dan *Upload*

*Upload* dan *download* adalah besarnya data yang dikirim (*transmitted*) dan diterima (*received*) dalam satuan *bytes*. Untuk mendapatkan data ini digunakan TrafficStats API. Untuk data *download* menggunakan TrafficStats.getTotalRxBytes(), yaitu membaca total *received* data dalam satuan *bytes*. Sedangkan untuk *upload* menggunakan TrafficStats.getTotalTxBytes() yang membaca total *transmitted* data dalam satuan *bytes*. Berikut adalah potongan kode untuk pengambilan data *download* dan *upload*.

```
long download = TrafficStats.getTotalRxBytes();
long upload = TrafficStats.getTotalTxBytes();
```

#### 4. Status *Download*

Status *download* adalah status ada atau tidaknya proses *download file*. Status *download* memiliki nilai 0 jika tidak ada *download* dan 1 jika ada *download*. Untuk mendapatkan status *download* dari perangkat digunakan *DownloadManager* API. Berikut adalah potongan kode untuk membaca status *download* pada perangkat.

```
DownloadManager.Query query = new DownloadManager.Query();
DownloadManager downloadManager =
    (DownloadManager)ctx.getSystemService(Context.DOWNLOAD_SERVICE);
Cursor c = downloadManager.query(query);
int status =
    c.getInt(c.getColumnIndex(DownloadManager.COLUMN_STATUS));
if(status == DownloadManager.STATUS_RUNNING)
    return 1;
else
    return 0;
```

#### 5. Baterai

Baterai adalah ketersediaan energi baterai dalam persen. Untuk membaca data energi baterai digunakan Android Intent *ACTION\_BATTERY\_CHANGED*. Berikut adalah potongan kode pengambilan ketersediaan energi baterai dalam persen.

```
IntentFilter filter=new IntentFilter(Intent.ACTION_BATTERY_CHANGED);
Intent batteryStat=context.registerReceiver(null, filter);
int battery=batteryStat.getIntExtra(BatteryManager.EXTRA_LEVEL, -1);
```

#### 6. Status Layar

Status layar adalah kondisi layar menyala (1) atau mati (0). Status layar didapatkan dengan menggunakan Android *PowerManager* API. Berikut adalah potongan kode untuk membaca status layar.

```
PowerManager mPowerManager =
    (PowerManager) ctx.getSystemService(Context.POWER_SERVICE);
boolean screen = mPowerManager.isScreenOn();
if(screen)
    return 1;
else
    return 0;
```

#### 7. Status *charge*

Status *charge* adalah kondisi perangkat sedang melakukan pengisian daya baterai atau tidak. Status *charge* dari perangkat didapatkan dengan

Intent.ACTION\_BATTERY\_CHANGED. Berikut adalah potongan kode untuk membaca status *charge* pada perangkat.

```
int intStatus =
batteryStatus.getIntExtra(BatteryManager.EXTRA_PLUGGED, -1);
int pluggedStatus = 0;
if(intStatus==0)
    pluggedStatus = "onbattery";
else if(intStatus==BatteryManager.BATTERY_PLUGGED_USB)
    pluggedStatus = 1;
else if(intStatus==BatteryManager.BATTERY_PLUGGED_AC)
    pluggedStatus = 1;
else
    pluggedStatus = 0;
```

#### 8. Latitude dan longitude

*Latitude* dan *longitude* adalah data lokasi dari perangkat yang diambil dari lokasi *cell tower* terdekat atau WiFi. Data *latitude* dan *longitude* didapatkan dengan menggunakan LocationManager API. Berikut adalah potongan kode untuk *request* update perubahan data lokasi.

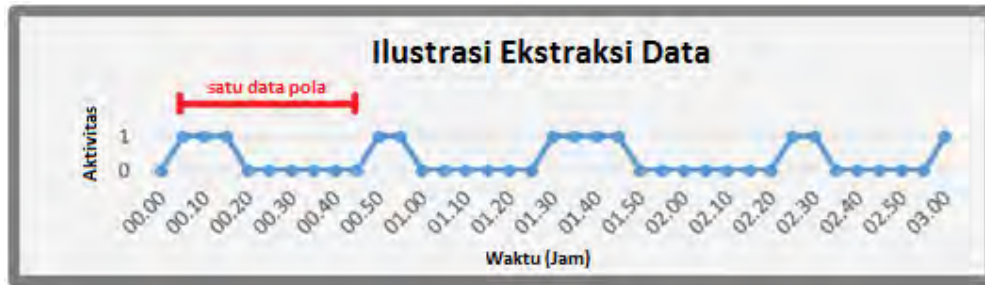
```
locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, 0, 0, locationListener);
```

Data yang diambil kemudian disimpan dalam *database* SQLite dalam *smartphone*. Lokasi penyimpanan data adalah /sdcard/dataBehavior, yaitu folder sdcard dalam *smartphone*, dengan nama *database* adalah dataBehavior. Data ini selanjutnya digunakan sebagai data *Training* untuk proses *learning*.

#### 3.4.1.2 Ekstraksi Data Aktivitas Smartphone

Proses ini bertujuan untuk mengekstraksi ciri dari data aktivitas *smartphone*. Langkah awal yang dilakukan dalam ekstraksi data adalah menentukan kondisi (aktif atau *idle*) dari setiap data. Setelah kondisi *idle* atau aktif ditentukan kemudian sistem melakukan penggabungan data. Setiap satu kelompok data aktif dan satu kelompok data *idle* di ekstrak menjadi satu data pola. Gambar 3.3 adalah ilustrasi pengelompokan data aktif dan *idle* menjadi satu pola.



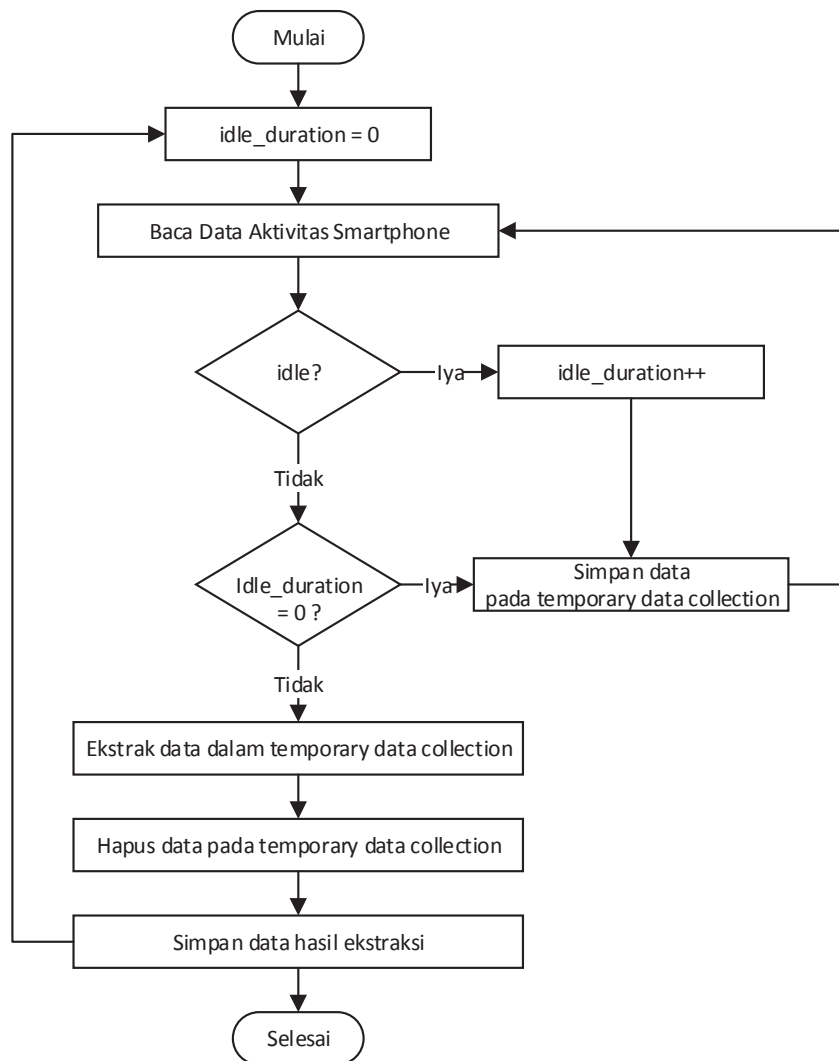


Gambar 3.3 Tahapan Praproses

Setiap satu kelompok data aktif dan satu kelompok data *idle* di ekstrak menjadi satu data pola. Pada gambar diilustrasikan data yang telah ditentukan status *idle* dan aktifnya dikelompokkan menjadi satu pola. Data terurut berdasarkan waktu. Data dengan nilai 0 artinya data tersebut *idle*, sedangkan data dengan nilai 1 adalah aktif.

Data digabung dengan metode rata-rata, kecuali data waktu dan koordinat (*latitude* dan *longitude*). Data waktu mengambil waktu awal pola. Data koordinat menggunakan fungsi *majority*. Data koordinat yang digunakan adalah data koordinat terbanyak dalam satu pola. Ada satu atribut tambahan yaitu *durasi\_idle*. *Durasi\_idle* didapatkan dengan menghitung lamanya waktu *idle* terjadi dalam sebuah pola.

Gambar 3.4 adalah alur proses ekstraksi data yang merupakan detail dari proses ekstraksi data pada gambar 3.2. Proses ekstraksi data diawali dengan membaca data aktivitas yang telah disimpan oleh sistem dalam *database*. Kemudian ditentukan kondisi *idle* atau aktif dari data tersebut. Jika data adalah *idle* maka *durasi\_idle* bertambah dan data ditambahkan pada temporal data collection. Jika data adalah aktif namun *durasi\_idle* sama dengan 0 maka data hanya ditambahkan pada data collection tanpa menambah *durasi\_idle*. Namun ketika data adalah aktif sedangkan *durasi\_idle* lebih dari 0 maka dilakukan ekstraksi. Data dalam temporary collection di ekstrak kemudian disimpan dalam *database* hasil ekstraksi. Kemudian semua data dalam temporary data collection di hapus dan *durasi\_idle* menjadi 0 kembali. Kemudian berlanjut untuk membaca data selanjutnya.



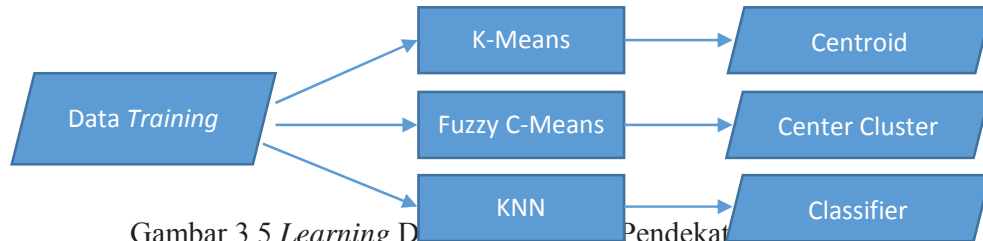
Gambar 3.4 Ekstraksi Data Aktivitas *Smartphone*

Data yang di ekstraksi dalam proses ini adalah data awal pengguna yang diambil dari perangkat . Sampel data awal pengguna dilampirkan pada lampiran 1. Hasil dari proses ekstraksi ini adalah data hasil ekstraksi yang kemudian digunakan untuk proses *learning*. Sampel data hasil ekstraksi dilampirkan dalam lampiran 2.

### 3.4.1.3 *Learning* Data Hasil Ekstraksi

Proses ini bertujuan untuk melakukan *learning* data hasil ekstraksi data aktivitas *smartphone*. Pada proses *learning* terdapat tiga pendekatan metode yang digunakan. Ketiga metode kami gunakan untuk mencari metode mana yang paling sesuai dengan data dalam penelitian ini. Gambar 3.5 menunjukkan tiga pendekatan

metode yang kami gunakan. Ketiga metode ini tidak digunakan secara bersamaan dalam satu proses. Namun tiap metode menjadi opsi pilihan untuk proses *learning* data.



Gambar 3.5 *Learning Data* dengan Pendekatan

1. *K-Means*

Dalam proses ini data akan di kelompokkan menjadi 5 *cluster*. Perhitungan jarak menggunakan fungsi euclidean distance. *Output* dari proses ini adalah *centroid* untuk tiap *cluster*. Selanjutnya centroid digunakan untuk prediksi durasi *idle*.

2. *K-Nearest Neighbors*

K-nearest neighbors adalah metode yang paling mudah diantara dua metode lain yang digunakan. Kami menggunakan nilai  $K = 5$ . Data *training* dikelompokkan berdasarkan durasi *idle*. Selanjutnya data ini menjadi classifier untuk mengklasifikasikan data baru.

3. *Fuzzy C-Means*

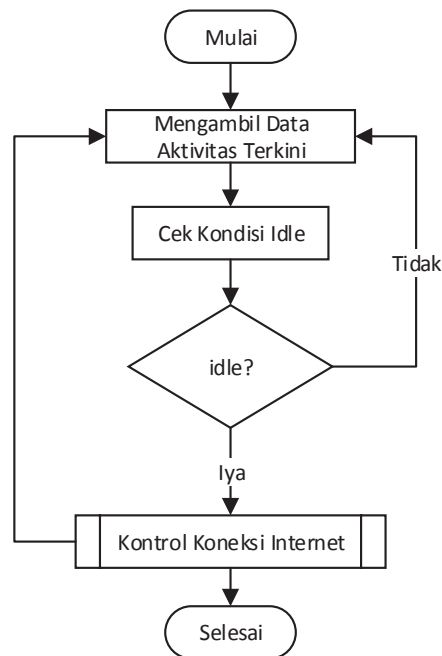
Skenario ketiga menggunakan *Fuzzy C-Means*. Data pola kebiasaan pengguna dikelompokkan menjadi 5 *cluster* dengan metode *Fuzzy C-Means*. Hasil dari proses *learning* dengan *Fuzzy C-Means* adalah *center cluster* untuk setiap cluster.

**3.4.2 Tahap Penghematan Energi Baterai**

Setelah tahap *learning* data berjalan selama 7 hari, kemudian sistem memasuki tahap penghematan energi baterai. Ada beberapa proses yang dilakukan dalam tahapan ini. Proses pertama adalah mengambil data aktivitas *smartphone* terkini. Data ini diambil secara *realtime* dari perangkat *smartphone* pengguna. Cara pengambilan data sama dengan proses pengambilan data pada tahap *learning* data aktivitas *smartphone*. Perbedaannya adalah pada tahap ini data tidak disimpan

dalam *database* untuk keperluan *learning*. Data diambil untuk keperluan proses kontrol koneksi Internet.

Proses selanjutnya adalah pengecekan kondisi *idle*. Untuk proses pengecekan kondisi *idle*, kami menggunakan aturan khusus yang dijelaskan pada sub-bab 3.4.2.1 mengenai penentuan kondisi *idle*. Hasil dari penentuan kondisi *idle* digunakan untuk menentukan proses selanjutnya yang akan dikerjakan oleh sistem. Jika hasil pengecekan adalah kondisi *smartphone* sedang aktif, maka sistem melakukan pengambilan data lagi pada menit berikutnya. Namun jika hasilnya adalah *idle* maka sistem melakukan proses kontrol koneksi Internet. Detil alur proses kontrol koneksi Internet dijelaskan pada *flowchart* selanjutnya tentang kontrol koneksi Internet.



Gambar 3.6 Tahap Penghematan Energi Baterai

#### 3.4.2.1 Penentuan Kondisi *Idle Smartphone*

Proses ini menentukan kondisi perangkat *smartphone* terkini apakah *idle* atau aktif. Kondisi *idle smartphone* ditentukan berdasarkan data persentase penggunaan CPU, kondisi layar, dan status *download* perangkat. Penentuan kondisi *idle* menggunakan aturan sebagai berikut :

```
if((layar==0 || CPUusage<25) && download_status==0)
    idle=true;
```

```
else  
    idle=false;
```

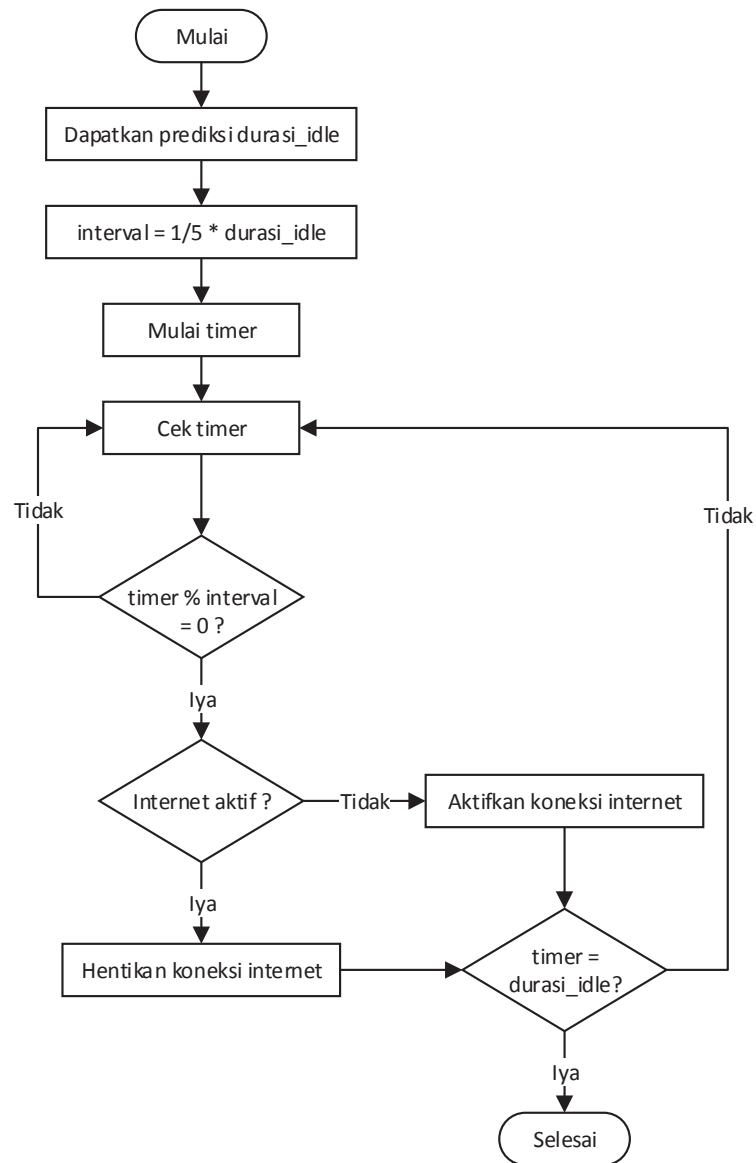
*Threshold* yang digunakan untuk persentase penggunaan CPU saat aktif dan *idle* adalah 25%. Jika CPU kurang dari *threshold* atau layar mati dan tidak ada proses *download*, maka kondisi *idle*. Jika ada proses *download* yang berjalan, kondisi adalah aktif meskipun kondisi layar mati dan CPU kurang dari *threshold*. Hal ini bertujuan untuk menghindari penghentian koneksi Internet saat ada proses *download* sedangkan kondisi layar mati dan penggunaan CPU rendah. Hasil dari proses penentuan kondisi *idle* ini digunakan untuk menentukan proses selanjutnya yang akan dilakukan oleh sistem. Jika hasil pengecekan adalah kondisi *smartphone* sedang aktif, maka sistem melakukan pengambilan data lagi pada menit berikutnya. Namun jika hasilnya adalah *idle* maka sistem melakukan proses kontrol koneksi Internet.

#### 3.4.2.2 Proses Kontrol Koneksi Internet

Ketika berada dalam kondisi *idle* maka sistem melakukan proses kontrol koneksi Internet. Dalam proses ini sistem terlebih dahulu melakukan prediksi *durasi\_idle* dan penghitungan interval waktu kontrol. Durasi proses kontrol Internet sesuai dengan hasil prediksi *durasi\_idle*. Selama proses kontrol, koneksi Internet dihentikan secara periodik dengan interval =  $1/5 * durasi\_idle$ . Gambar 3.3 menunjukkan alur proses kontrol koneksi Internet. Gambar tersebut adalah penjelasan detail dari proses kontrol koneksi Internet pada gambar 3.6 mengenai tahap penghematan energi.

Saat perangkat memasuki kondisi *idle*, sistem memprediksi berapa lama *idle* akan berlangsung. Setelah didapatkan perkiraan durasi *idle* kemudian dihitung interval waktu untuk jeda koneksi Internet. Kemudian koneksi Internet dihentikan secara berkala dengan jeda waktu yang diberikan. Kontrol ini berakhir setelah berjalan selama durasi *idle*. Koneksi Internet tidak dihentikan secara terus menerus selama durasi *idle*. Namun dalam proses ini dilakukan penghentian dan pengaktifan kembali koneksi Internet secara berulang dan bergantian. Misal hasil prediksi durasi *idle* adalah 60 menit, maka sistem akan interval yang digunakan adalah 12 menit ( $1/5 * durasi\_idle$ ). Selama proses kontrol, koneksi Internet dihentikan selama 12 menit kemudian diaktifkan kembali selama 12 menit dan dihentikan kembali 12

menit kemudian. Begitu seterusnya penghentian dan pengaktifan koneksi Internet dilakukan secara bergantian hingga proses kontrol berakhir.



Gambar 3.7 Kontrol Koneksi Internet

Tujuan dilakukannya penghentian dan pengaktifan kembali koneksi Internet secara bergantian dan berulang adalah untuk menjaga kelancara proses transfer data. Jika tidak ada interval maka proses transfer data terhenti ketika koneksi dihentikan (saat *smartphone idle*) dan baru berlanjut ketika koneksi Internet aktif (set *smartphone* aktif). Jika *idle* terjadi dalam durasi yang sangat lama maka data yang harusnya diterima pada saat *idle* tidak akan diterima hingga

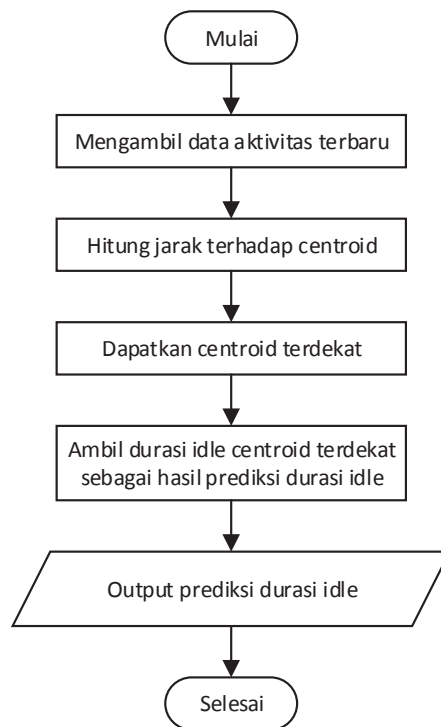
*smartphone* memasuki kondisi aktif dan koneksi Internet diaktifkan kembali. Akan terjadi hambatan dalam waktu yang cukup lama untuk proses transfer data

### 3.4.2.3 Prediksi Durasi *Idle*

Proses ini bertujuan untuk mendapatkan prediksi durasi *idle*. Prediksi dilakukan menggunakan tiga pendekatan metode, yaitu *K-Means*, *Fuzzy C-Means*, *KNN*. Ketiga metode tersebut digunakan untuk mencari metode yang paling sesuai untuk kasus dalam penelitian ini.

#### 1. *K-Means*

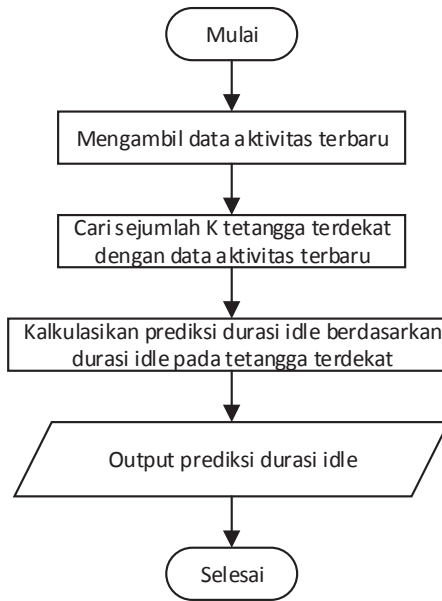
Data baru dikelompokkan ke dalam *cluster* berdasarkan *centroid* hasil proses *learning* dengan *k-means*. Pengelompokan dilakukan dengan mencari *centroid* terdekat terhadap data baru. Kemudian durasi *idle* pada *centroid* terdekat menjadi hasil dari prediksi *durasi\_idle*.



Gambar 3.8 Pencarian Durasi *Idle* dengan K-Means

#### 2. *K-Nearest Neighbors*

K-nearest neighbors adalah metode yang paling mudah diantara dua metode lain yang digunakan. Kami menggunakan nilai  $K = 5$ . Setiap ada data baru maka langsung dicari 5 tetangga terdekat dengan data tersebut. Kemudian dari 5 tetangga terdekat tersebut dihitung rata-rata *durasi\_idle*.



Gambar 3.9 Pencarian Durasi *Idle* dengan K-nearest Neighbors

### 3. Fuzzy C-Means

Prediksi durasi *idle* dengan menggunakan fuzzy c-means dilakukan dengan menghitung derajat keanggotaan data baru terhadap masing-masing center cluster. Dari derajat keanggotaan yang didapat kemudian dilakukan perhitungan durasi *idle* berdasarkan durasi *idle* masing-masing center cluster. Durasi *idle* pada center cluster ke- $n$  dikalikan dengan derajat keanggotaan data baru terhadap center cluster ke- $n$ . Kemudian dijumlahkan semua hasil perkalian durasi *idle* dan derajat keanggotaan data baru dari masing-masing Center cluster. Hasil dari perhitungan menjadi hasil prediksi durasi *idle*. Berikut adalah formula yang digunakan.

$$\text{prediksi} = \sum_{k=0}^n d_k m_k$$

Nilai  $n$  dalam formula tersebut adalah jumlah cluster,  $m$  dalam formula tersebut adalah durasi *idle* pada center cluster ke- $k$ , sedangkan  $m$  adalah derajat keanggotaan data baru terhadap center cluster ke- $k$ .



### 3.5 Skenario Uji Coba

Pengujian dilakukan untuk mengukur kesesuaian hasil dari metode yang diusulkan dengan hipotesis yang diajukan. Ada dua tahapan yang dilakukan dalam pengujian yaitu tahap *learning* data aktivitas *smartphone* dan tahap kontrol koneksi Internet. Berikut adalah penjelasan untuk masing-masing tahapan:

#### 3.5.1 Tahap *Learning* Data Aktivitas *Smartphone*

Uji coba ini dilakukan untuk melihat apakah proses pengambilan dan penyimpanan data aktivitas *smartphone* berjalan sesuai perancangan atau tidak. Selain itu juga dilakukan pengamatan terhadap proses *learning* data aktivitas. Untuk itu pada tahapan *learning* data aktivitas *smartphone* dilakukan tiga uji coba sebagai berikut:

1. Uji coba pengambilan data aktivitas *smartphone*  
Pengambilan data dilakukan secara langsung pada pengguna selama tujuh hari. Pengambilan data dilakukan setiap satu menit. Data yang telah diambil disimpan dalam *database* untuk digunakan dalam proses selanjutnya. Pengujian ini dilakukan untuk melihat bagaimana proses pengambilan data yang terjadi pada .
2. Uji coba ekstraksi data aktivitas  
Setelah tujuh hari mengambil dan menyimpan data aktivitas *smartphone* kemudian sistem melakukan proses ekstraksi data aktivitas. Pengujian dilakukan untuk melihat apakah hasil ekstraksi telah sesuai dengan format data input untuk proses *learning* atau tidak.
3. Uji coba fungsi *learning* data aktivitas  
Pengujian ketiga dilakukan terhadap proses *learning* data hasil ekstraksi. Proses *learning* ini menghasilkan data centroid untuk cluster data yang telah di kelompokkan dalam beberapa cluster. Centroid dan center cluster dari proses ini selanjutnya digunakan untuk prediksi durasi *idle* pada tahap kontrol koneksi Internet.

### 3.5.2 Tahap Kontrol Koneksi Internet

Pada tahap ini sistem melakukan pengambilan data aktivitas terkini dari perangkat. Kemudian dari data yang didapat sistem menentukan kondisi *smartphone idle* atau aktif. Jika kondisi *smartphone* adalah *idle* maka sistem melakukan proses prediksi durasi *idle*. Setelah prediksi durasi *idle* didapat maka sistem melakukan kontrol terhadap koneksi Internet berdasarkan durasi *idle* yang didapat. Untuk mengamati kinerja sistem pada tahap ini dilakukan beberapa uji coba seperti berikut:

1. Uji coba penentuan kondisi *idle*

Sistem melakukan pengecekan kondisi *idle smartphone* berdasarkan data aktivitas terkini dari perangkat. Status *idle* ditentukan berdasarkan persentase penggunaan CPU, status layar, dan status *download*. Aturan untuk penentuan kondisi *idle* dijelaskan pada sub-bab 3.4.2.1.

2. Uji coba prediksi durasi *idle*

Ketika hasil penentuan kondisi *idle* menunjukkan bahwa *smartphone* dalam kondisi aktif, maka sistem melakukan prediksi durasi *idle*. ada tiga pendekatan metode yang digunakan untuk proses prediksi durasi *idle*. ketiga metode digunakan untuk mencari metode yang paling sesuai dengan kasus pada penelitian ini.

3. Uji coba fungsi kontrol koneksi Internet

Setelah hasil prediksi *durasi\_idle* didapatkan, maka sistem melakukan pengontrolan koneksi Internet. Proses kontrol koneksi Internet dilakukan dengan durasi sesuai hasil prediksi *durasi\_idle*. Selama proses kontrol, koneksi Internet dihentikan secara berkala dengan interval waktu tertentu. Uji coba ini dilakukan untuk melihat bahwa kontrol koneksi Internet telah sesuai dengan *input* yang digunakan. *Input* yang digunakan adalah *durasi\_idle* hasil prediksi dari pengujian sebelumnya

## **BAB 4**

### **HASIL DAN PEMBAHASAN**

#### **4.1 Tahapan Uji Coba**

Uji coba dilakukan dalam dua tahapan yaitu tahap *learning* data aktivitas *smartphone* dan tahap kontrol koneksi Internet. Masing-masing tahapan uji coba memiliki beberapa sub pengujian. Berikut adalah tahapan uji coba beserta sub pengujian yang dilakukan dalam uji coba:

- a) Tahap *learning* data aktivitas *smartphone*
  1. Uji coba pengambilan data aktivitas *smartphone*
  2. Uji coba ekstraksi data aktivitas
  3. Uji coba *learning* data hasil ekstraksi
- b) Tahap kontrol koneksi Internet
  1. Uji coba penentuan kondisi *idle smartphone*
  2. Uji coba prediksi durasi *idle*
  3. Uji coba kontrol koneksi Internet
- c) Uji Coba Performa
  1. Performa metode prediksi durasi *idle*
  2. Performa prediksi durasi *idle* berdasarkan data pengguna berbeda
  3. Performa metode KNN berdasarkan nilai K
  4. Kesiapan Konektivitas Internet
  5. Kelancaran proses transfer data
  6. Penghematan konsumsi energi

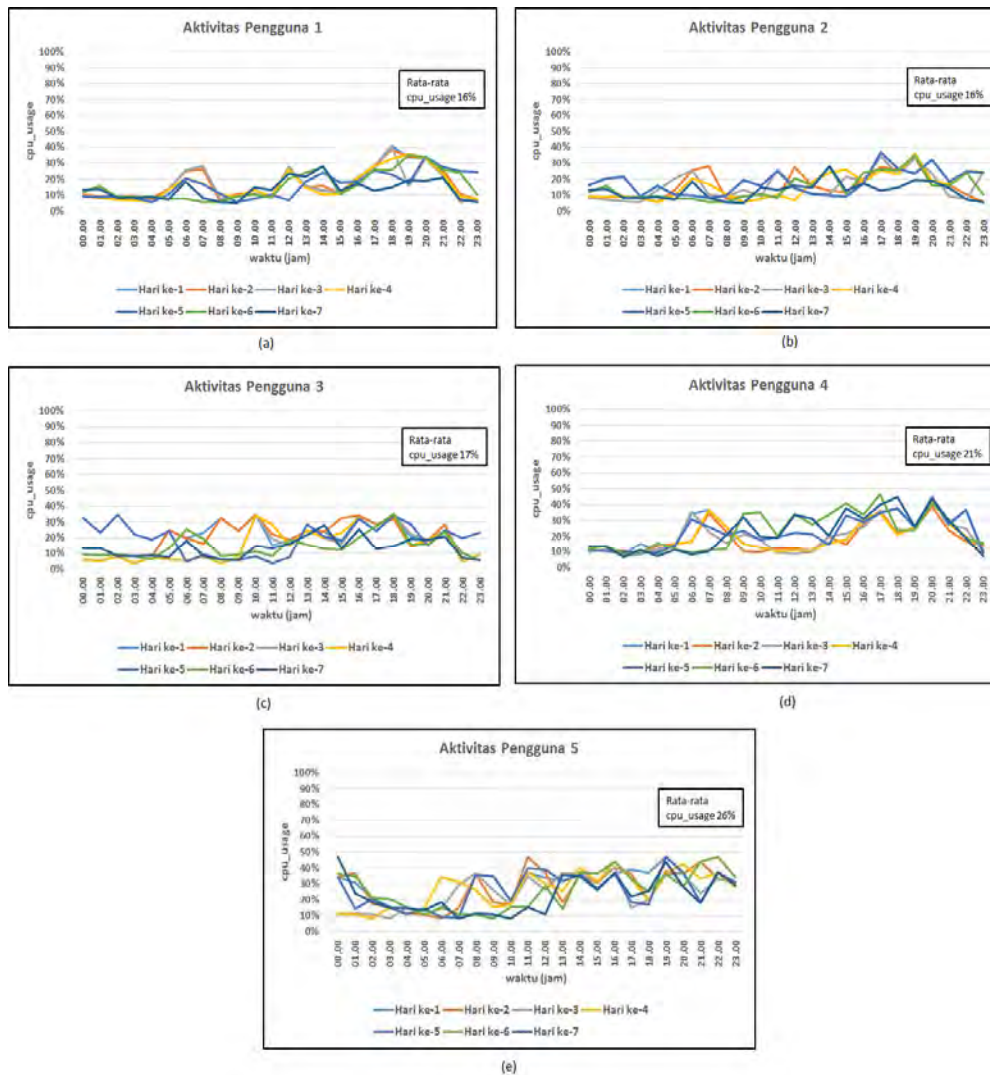
#### **4.2 Data Uji Coba**

Ada beberapa data yang digunakan dalam pengujian. data ini disesuaikan dengan uji coba yang dilakukan. Berikut adalah data-data yang digunakan dalam uji coba.

##### a) Data kebiasaan pengguna

Data kebiasaan pengguna merupakan data utama untuk metode yang diusulkan dalam penelitian ini. Data kebiasaan pengguna yang berbeda digunakan

berkaitan dengan tujuan penelitian untuk pengenalan kebiasaan pengguna. Data diambil dari lima pengguna dengan kebiasaan penggunaan yang berbeda.



Gambar 4.1 Grafik Penggunaan CPU Dalam Waktu Satu Minggu untuk (a) Pengguna 1, (b) Pengguna 2, (c) Pengguna 3, (d) Pengguna 4, (e) Pengguna 5

Perbedaan kebiasaan pengguna terletak pada rutinitas penggunaan oleh masing-masing pengguna serta seberapa aktif penggunaan dilakukan. Rutinitas penggunaan diamati dari waktu-waktu terjadinya *idle* dan aktivitas pada dalam keseharian pengguna. Waktu *idle* dan aktif pengguna terlihat dari persentase penggunaan CPU. Persentase penggunaan CPU tinggi menunjukkan perangkat aktif, sedangkan persentase penggunaan rendah menunjukkan kondisi *idle*. Dari

pengamatan waktu *idle* dan aktif dapat dilihat apakah rutinitas penggunaan *smartphone* teratur atau tidak. Perbedaan tingkat keaktifan pengguna dilihat dari rata-rata persentase penggunaan CPU *smartphone* setiap hari. Semakin tinggi rata-rata penggunaan CPU maka semakin aktif pengguna dalam menggunakan perangkatnya. Detail perbedaan kebiasaan pengguna dapat dilihat dari gambar 4.1. Gambar 4.1 adalah grafik persentase penggunaan CPU oleh masing-masing pengguna dalam waktu satu minggu.

Aktivitas penggunaan diamati dari persentase CPU *usage* . Gambar 4.1 menunjukkan aktivitas penggunaan tiap pengguna yang diamati dari persentase CPU *usage*. Berikut adalah penjelasan kebiasaan masing-masing pengguna berdasarkan gambar 4.1:

1. Pengguna 1

Grafik menjelaskan bahwa pengguna satu memiliki kebiasaan rutinitas penggunaan yang teratur dengan tingkat aktivitas penggunaan yang rendah. Rata-rata CPU *usage* yang rendah (16%) pada grafik pengguna satu menunjukkan bahwa pengguna satu jarang menggunakan perangkat . Tingkat aktivitas penggunaan perangkat rendah. Namun pengguna satu memiliki rutinitas kebiasaan penggunaan yang teratur. Terlihat dari waktu *idle* dan aktif pengguna dalam satu minggu relatif berada pada waktu-waktu yang sama.

2. Pengguna 2

Grafik menjelaskan bahwa pengguna dua memiliki kebiasaan rutinitas penggunaan yang tidak teratur dengan tingkat aktivitas penggunaan yang rendah. Pada grafik pengguna dua terlihat rata-rata CPU *usage* rendah (16%). Hal itu menunjukkan bahwa pengguna dua memiliki intensitas penggunaan yang rendah. Rutinitas kebiasaan penggunaan tidak teratur jika dibandingkan dengan pengguna satu.

3. Pengguna 3

Grafik menjelaskan bahwa pengguna tiga memiliki kebiasaan rutinitas penggunaan yang tidak teratur dengan tingkat aktivitas penggunaan

yang rendah. Pada grafik pengguna tiga terlihat rata-rata CPU *usage* rendah (17%). Hal itu menunjukkan bahwa pengguna tiga memiliki intensitas penggunaan yang rendah. Pengguna tiga kurang aktif menggunakan perangkat . Rutinitas kebiasaan penggunaan cukup teratur. Namun ada satu hari yang rutinitasnya jadwal rutinitasnya berbeda dengan hari lain.

#### 4. Pengguna 4

Grafik menjelaskan bahwa pengguna empat memiliki kebiasaan rutinitas penggunaan yang teratur dengan tingkat aktivitas penggunaan yang tinggi. Rata-rata CPU *usage* pada grafik pengguna ketiga menunjukkan nilai yang tinggi (21%). Hal ini menunjukkan bahwa pengguna 4 sangat aktif dalam menggunakan perangkat . Rutinitas kebiasaan penggunaan teratur. Waktu *idle* dan aktif memiliki kesamaan pada beberapa hari tertentu, meskipun ada beberapa hari yang terlihat berbeda.

#### 5. Pengguna 5

Grafik menjelaskan bahwa pengguna lima memiliki kebiasaan rutinitas penggunaan yang teratur dengan tingkat aktivitas penggunaan yang rendah. Rata-rata CPU *usage* pada grafik pengguna 5 menunjukkan nilai yang tinggi (26%). Hal ini menunjukkan pengguna lima sangat aktif dalam menggunakan . Intensitas penggunaan sangat sering. Rutinitas kebiasaan penggunaan teratur. Namun masih dapat dilihat dengan jelas waktu *idle* dan aktif yang rutin setiap harinya.

#### b) Data Durasi *idle*

Parameter durasi *idle* digunakan pada pengujian kontrol konektivitas Internet. Kontrol konektivitas Internet menggunakan interval yang dikalkulasikan dari durasi *idle*. Kontrol dilakukan selama durasi *idle*, dan selama kontrol diberikan jeda terhadap proses penghentian koneksi Internet berdasarkan interval yang didapat.

#### c) Data Status perangkat

Ada beberapa data dalam status perangkat yang digunakan dalam uji coba, antara lain kapasitas baterai, persentase CPU *usage*, status layar, status *idle* perangkat, status koneksi Internet, serta *download* dan *upload*.

### **4.3 Hasil Uji Coba dan Analisis**

Pada bagian ini ditampilkan hasil dari uji coba yang telah dilakukan beserta analisisnya. Ada dua tahapan utama dalam uji coba yang dilakukan yaitu uji coba fungsionalitas dan uji coba performa. Uji coba fungsionalitas dan performa masing-masing memiliki beberapa tahapan yang dilakukan. Hasil uji coba disajikan pada tiap tahap uji coba yang dilakukan.

#### **4.3.1 Tahap *Learning Data* Aktivitas *Smartphone***

Uji coba tahap pertama bertujuan untuk mengumpulkan data aktivitas *smartphone* dan melakukan *learning data*. Uji coba dilakukan dalam waktu 7 hari. Selama proses uji coba sistem menyimpan data aktivitas *smartphone* dalam *database*. Setelah data selama 7 hari tersimpan kemudian dilakukan proses ekstraksi data dan *learning*. Hasil dari proses *learning* digunakan untuk prediksi durasi *idle* pada tahap uji coba kontrol koneksi Internet.

Ada tiga proses yang di uji coba pada tahap ini, yaitu proses pengambilan data aktivitas *smartphone*, proses ekstraksi data aktivitas, dan proses *learning data* hasil ekstraksi. Berikut adalah detail hasil uji coba pada tiap tahapan:

##### **4.3.1.1 Uji Coba Pengambilan Data Aktivitas *Smartphone***

Pengambilan data adalah proses awal yang dilakukan sistem. Data diambil secara *realtime* dari pengguna. Proses pengambilan data berlangsung secara terus-menerus selama tujuh hari. Jeda waktu pengambilan data adalah satu menit. Data disimpan dalam *database* SQLite pada *smartphone* pengguna. Data ini selanjutnya digunakan untuk proses penentuan kondisi *idle* dan proses *learning*.

Ada sepuluh atribut yang dibaca dan disimpan sebagai data aktivitas pengguna, yaitu waktu, CPU *usage*, *download*, *upload*, status *download*, baterai, status layar, status *charge*, *latitude*, dan *longitude*. Data direkam secara *realtime* setiap satu menit. Berikut ini adalah penjelasan proses pengambilan data untuk masing-masing atribut.

## 1. Waktu

Atribut waktu adalah waktu penyimpanan data dalam satuan menit. Waktu diambil dari sistem pada perangkat . Untuk mendapatkan data waktu ini digunakan Android API Calendar. Dari API tersebut didapatkan data jam dan menit saat ini yang kemudian disimpan dalam satuan menit. Berikut adalah potongan kode untuk pengambilan data waktu:

```
Calendar calendar = Calendar.getInstance();
time = ( c.get(Calendar.HOUR_OF_DAY) * 60 ) +
        c.get(Calendar.MINUTE);
```

Pada potongan kode diatas terlihat data jam dikalikan 60 dan ditambahkan dengan data menit. Dari rumusan tersebut didapatkan waktu dalam satuan menit. Dalam satu hari ada 1440 menit. Jadi rentang nilai untuk atribut waktu adalah 1 sampai dengan 1440.

## 2. CPU

*CPU usage* adalah persentase penggunaan CPU pada perangkat *smartphone*. Data persentase penggunaan CPU dibaca dari file `/proc/stat` pada sistem. File `/proc/stat` berisi informasi aktivitas kernel. Dari file ini dapat diketahui penggunaan CPU dalam interval waktu tertentu.

## 3. *Download* dan *Upload*

*Upload* dan *download* adalah besarnya data yang dikirim (*transmitted*) dan diterima (*received*) dalam satuan *bytes*. Untuk mendapatkan data ini digunakan `TrafficStats` API. Untuk data *download* menggunakan `TrafficStats.getTotalRxBytes()`, yaitu membaca total *received* data dalam satuan *bytes*. Sedangkan untuk *upload* menggunakan `TrafficStats.getTotalTxBytes()` yang membaca total *transmitted* data dalam satuan *bytes*. Berikut adalah potongan kode untuk pengambilan data *download* dan *upload*.

```
long download = TrafficStats.getTotalRxBytes();
long upload = TrafficStats.getTotalTxBytes();
```

## 4. Status *Download*

Status *download* adalah status ada atau tidaknya proses *download file*. Status *download* memiliki nilai 0 jika tidak ada *download* dan 1 jika ada



*download*. Untuk mendapatkan status *download* dari perangkat digunakan *DownloadManager* API. Berikut adalah potongan kode untuk membaca status *download* pada perangkat.

```
DownloadManager.Query query = new DownloadManager.Query();
DownloadManager downloadManager =
    (DownloadManager)ctx.getSystemService(Context.DOWNLOAD_SERVICE);
Cursor c = downloadManager.query(query);
int status =
    c.getInt(c.getColumnIndex(DownloadManager.COLUMN_STATUS));
if(status == DownloadManager.STATUS_RUNNING)
    return 1;
else
    return 0;
```

## 5. Baterai

Baterai adalah ketersediaan energi baterai dalam persen. Untuk membaca data energi baterai digunakan Android Intent *ACTION\_BATTERY\_CHANGED*. Berikut adalah potongan kode pengambilan ketersediaan energi baterai dalam persen.

```
IntentFilter ifilter = new IntentFilter(
    Intent.ACTION_BATTERY_CHANGED);
Intent batteryStatus = context.registerReceiver(null, ifilter);
int battery = batteryStatus.getIntExtra(
    BatteryManager.EXTRA_LEVEL, -1);
```

## 6. Status Layar

Status layar adalah kondisi layar menyala (1) atau mati (0). Status layar didapatkan dengan menggunakan Android *PowerManager* API. Berikut adalah potongan kode untuk membaca status layar.

```
PowerManager mPowerManager =
    (PowerManager) ctx.getSystemService(Context.POWER_SERVICE);
boolean screen = mPowerManager.isScreenOn();
if(screen)
    return 1;
else
    return 0;
```

## 7. Status *charge*

Status *charge* adalah kondisi perangkat sedang melakukan pengisian daya baterai atau tidak. Status *charge* dari perangkat didapatkan dengan

Intent.ACTION\_BATTERY\_CHANGED. Berikut adalah potongan kode untuk membaca status *charge* pada perangkat.

```
int intStatus =
batteryStatus.getIntExtra(BatteryManager.EXTRA_PLUGGED, -1);
int pluggedStatus = 0;
if(intStatus==0)
    pluggedStatus = "onbattery";
else if(intStatus==BatteryManager.BATTERY_PLUGGED_USB)
    pluggedStatus = 1;
else if(intStatus==BatteryManager.BATTERY_PLUGGED_AC)
    pluggedStatus = 1;
else
    pluggedStatus = 0;
```

#### 8. Latitude dan longitude

*Latitude* dan *longitude* adalah data lokasi dari perangkat yang diambil dari lokasi *cell tower* terdekat atau WiFi. Data *latitude* dan *longitude* didapatkan dengan menggunakan LocationManager API. Berikut adalah potongan kode untuk *request* update perubahan data lokasi.

```
locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, 0, 0, locationListener);
```

Data yang diambil kemudian disimpan dalam *database* SQLite dalam *smartphone*. Lokasi penyimpanan data adalah /sdcard/dataBehavior, yaitu folder sdcard dalam *smartphone*, dengan nama *database* adalah dataBehavior. Data ini selanjutnya digunakan sebagai data *Training* untuk proses *learning*.

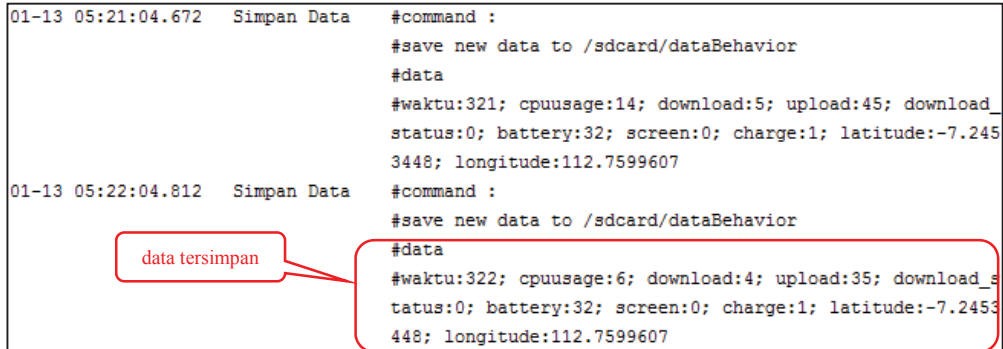
Pengujian dilakukan dengan memberikan beberapa perlakuan terhadap *smartphone* selama proses pengambilan data. Hal ini dilakukan untuk melihat perbedaan data yang disimpan oleh sistem. Selama proses pengujian dilakukan pengamatan terhadap *log* dari *smartphone*. Berikut adalah hasil pengamatan *log* yang dilakukan.

- *Smartphone* dalam keadaan *idle*

Semua aplikasi yang berjalan pada *smartphone* dihentikan sehingga *smartphone* berada dalam kondisi *idle*. Data *log smartphone* diamati ketika *smartphone* berada dalam kondisi *idle*. Layar dalam keadaan mati, tidak ada proses *download* dan tidak ada aplikasi yang berjalan. Gambar 4.2 adalah capture *log* dari *smartphone*. Pada gambar terlihat data CPU adalah 6%, *download\_status* 0, dan

screen 0. Sesuai dengan aturan untuk penentuan kondisi *idle*, data tersebut menunjukkan *smartphone* dalam kondisi *idle*.

```
01-13 05:21:04.672 Simpan Data #command :
                                #save new data to /sdcard/dataBehavior
                                #data
                                #waktu:321; cpuusage:14; download:5; upload:45; download_
                                status:0; battery:32; screen:0; charge:1; latitude:-7.245
                                3448; longitude:112.7599607
01-13 05:22:04.812 Simpan Data #command :
                                #save new data to /sdcard/dataBehavior
                                #data
                                #waktu:322; cpuusage:6; download:4; upload:35; download_s
                                tatus:0; battery:32; screen:0; charge:1; latitude:-7.2453
                                448; longitude:112.7599607
```

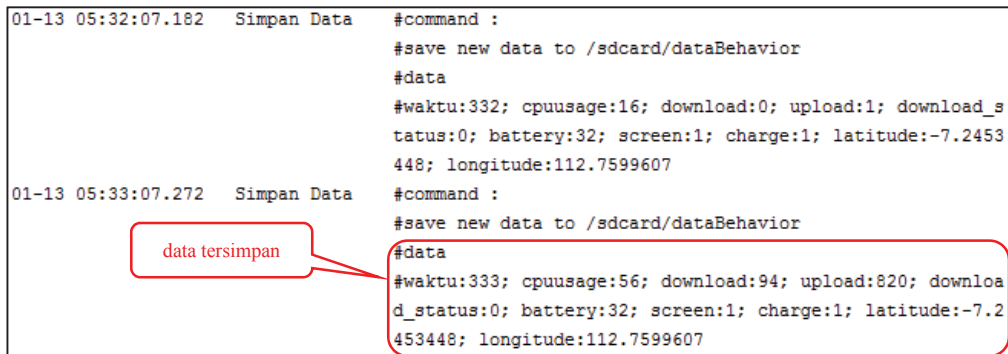


Gambar 4.2 Log Penyimpanan Data dengan Kondisi *Smartphone Idle*

- *Smartphone* dalam keadaan aktif

Sebuah aplikasi dijalankan pada *smartphone* sehingga *smartphone* masuk kondisi aktif. Data *log smartphone* diamati ketika *smartphone* berada dalam kondisi aktif. *Smartphone* menjalankan satu aplikasi agar *smartphone* tetap dalam kondisi aktif. Gambar 4.3 adalah capture *log* penyimpanan data dengan *smartphone* dalam kondisi aktif. Pada gambar terlihat data CPU adalah 56%, *download\_status* 0, dan screen 1. Data tersebut menunjukkan kondisi *smartphone* dalam kondisi aktif.

```
01-13 05:32:07.182 Simpan Data #command :
                                #save new data to /sdcard/dataBehavior
                                #data
                                #waktu:332; cpuusage:16; download:0; upload:1; download_s
                                tatus:0; battery:32; screen:1; charge:1; latitude:-7.2453
                                448; longitude:112.7599607
01-13 05:33:07.272 Simpan Data #command :
                                #save new data to /sdcard/dataBehavior
                                #data
                                #waktu:333; cpuusage:56; download:94; upload:820; downloa
                                d_status:0; battery:32; screen:1; charge:1; latitude:-7.2
                                453448; longitude:112.7599607
```



Gambar 4.3 Log Penyimpanan Data dengan Kondisi *Smartphone Aktif*

- *Smartphone* dalam status *download* aktif

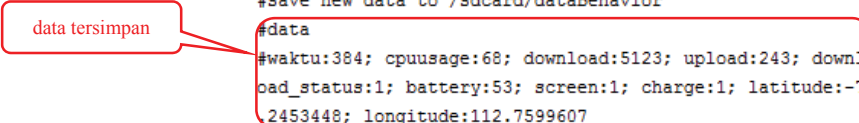
Pada *smartphone* dilakukan *download* data sehingga status *download* menjadi aktif. Data *log smartphone* diamati ketika *smartphone* berada dalam kondisi aktif. *Smartphone* menjalankan satu aplikasi agar *smartphone* tetap dalam kondisi aktif. Gambar 4.4 adalah capture *log* penyimpanan data dengan *smartphone* dalam kondisi aktif. Pada gambar terlihat data CPU adalah 56%, *download\_status*

0, dan screen 1. Data tersebut menunjukkan kondisi *smartphone* dalam kondisi aktif.

```

01-13 06:23:39.522  Simpan Data  #command :
                                #save new data to /sdcard/dataBehavior
                                #data
                                #waktu:383; cpuusage:89; download:7931; upload:223; downl
                                oad_status:1; battery:53; screen:1; charge:1; latitude:-7
                                .2453448; longitude:112.7599607
01-13 06:24:40.032  Simpan Data  #command :
                                #save new data to /sdcard/dataBehavior
                                #data
                                #waktu:384; cpuusage:68; download:5123; upload:243; downl
                                oad_status:1; battery:53; screen:1; charge:1; latitude:-7
                                .2453448; longitude:112.7599607

```



Gambar 4.4 Log Penyimpanan Data dengan Kondisi *Download* Aktif

#### 4.3.1.2 Uji Coba Ekstraksi Data Aktivitas

Data yang diproses dalam tahap praproses adalah seluruh data yang telah disimpan oleh sistem. Dalam pengujian ini data direkam dalam waktu 7 hari. Penyimpanan data dilakukan dengan jeda satu menit. Data yang tersimpan dalam tujuh hari adalah 10.080 data. Sampel data masing-masing pengguna yang direkam oleh sistem terlampir pada lampiran 1. Data tersebut adalah data awal yang selanjutnya digunakan dalam praproses untuk di ekstraksi. Tabel 4.1 berikut adalah sampel data awal pengguna 1.

Tabel 4.1 Sampel Data Awal Pengguna 1

waktu	CPUusage	download	upload	lat	lon	Download status	battery	layar	charge
460	5	0	0	-7.2453448	112.7599607	0	54	1	0
461	2	9	1	-7.2453448	112.7599607	0	54	1	0
462	3	0	0	-7.2453448	112.7599607	0	54	1	0
463	4	9	1	-7.2453432	112.7599603	0	54	1	0
464	60	0	0	-7.2453432	112.7599603	0	54	1	0
465	66	8	1	-7.2453432	112.7599603	0	54	1	0
466	16	0	0	-7.2453438	112.7599583	0	54	1	0
467	16	8	1	-7.2453438	112.7599583	0	54	1	0
468	14	0	0	-7.2453438	112.7599583	0	54	1	0
469	28	9	1	-7.2453457	112.7599598	0	54	1	0

Tabel 4.1 adalah sampel data awal dari pengguna 1. Langkah pertama yang dilakukan adalah menentukan kondisi (*idle* atau aktif) dari setiap baris data. Penentuan kondisi *idle* menggunakan aturan yang telah dirancang dan digunakan dalam proses penentuan kondisi *idle*. Hasil dari penentuan kondisi *idle* setiap baris data ditunjukkan dalam tabel 4.2. Pada tabel 4.2 terlihat ada baris yang berwarna

merah dan berwarna hijau. Baris yang berwarna merah menunjukkan data pada baris itu adalah data aktif. Sedangkan baris hijau menunjukkan data *idle*.

Tabel 4.2 Hasil Penentuan Kondisi *Idle* Data Awal

waktu	CPUusage	download	upload	lat	lon	Download status	battery	layar	charge
460	5	0	0	-7.2453448	112.7599607	0	54	1	0
461	2	9	1	-7.2453448	112.7599607	0	54	1	0
462	3	0	0	-7.2453448	112.7599607	0	54	1	0
463	4	9	1	-7.2453432	112.7599603	0	54	1	0
464	60	0	0	-7.2453432	112.7599603	0	54	1	0
465	66	8	1	-7.2453432	112.7599603	0	54	1	0
466	16	0	0	-7.2453438	112.7599583	0	54	1	0
467	16	8	1	-7.2453438	112.7599583	0	54	1	0
468	14	0	0	-7.2453438	112.7599583	0	54	1	0
469	28	9	1	-7.2453457	112.7599598	0	54	1	0

Setelah didapatkan kondisi *idle* dan aktif dari tiap data kemudian langkah selanjutnya adalah proses ekstraksi data. Setiap satu kelompok data aktif dan satu kelompok data *idle* di ekstrak menjadi satu data pola. Data digabung dengan metode rata-rata, kecuali data waktu dan koordinat (*latitude* dan *longitude*). Data waktu mengambil waktu awal pola. Data koordinat menggunakan fungsi *majority*. Data koordinat yang digunakan adalah data koordinat terbanyak dalam satu pola. Ada satu atribut tambahan yaitu *durasi\_idle*. *Durasi\_idle* didapatkan dengan menghitung lamanya waktu *idle* terjadi dalam sebuah pola. Tabel 4.3 adalah hasil ekstraksi dari tabel 4.2.

Tabel 4.3 Hasil Penentuan Kondisi *Idle* Data Awal

Durasi <i>idle</i>	waktu	CPUusage	download	upload	lat	lon	Download status	battery	layar	charge
6.0	460.0	23.333334	4.3333335	0.5	-7.2453448	112.7599607	0.0	54.0	1.0	0.0
4.0	466.0	18.5	4.25	0.5	-7.2453437	112.75996	0.0	47.0	1.0	0.0

Seluruh data awal untuk masing-masing pengguna di ekstraksi dengan cara yang sama dengan langkah-langkah di atas. Hasil dari proses ekstraksi data ini adalah data hasil ekstraksi untuk masing-masing pengguna yang di lampirkan pada lampiran 2. Selanjutnya data hasil ekstraksi ini digunakan untuk proses *learning*.

#### 4.3.1.3 Uji Coba *Learning* Data Hasil Ekstraksi

Proses ini bertujuan untuk melakukan *learning* data hasil ekstraksi dari data awal pengguna. Sampel data hasil ekstraksi untuk masing-masing pengguna dapat dilihat pada lampiran 2. Pada proses *learning* terdapat tiga pendekatan metode yang digunakan, yaitu K-Means, Fuzzy C-Means, dan KNN. Ketiga metode

kami gunakan untuk mencari metode mana yang paling sesuai dengan data dalam penelitian ini.

Hasil dari proses *learning* ini adalah centroid dan center cluster untuk tiap kelompok cluster. Centroid adalah hasil dari proses *learning* dengan K-Means. Sedangkan center cluster adalah hasil dari metode Fuzzy C-Means. Centroid dan center cluster ini selanjutnya digunakan untuk proses prediksi durasi *idle*. Tabel 4.4 adalah centroid dan center cluster hasil clustering data pengguna 1.

Tabel 4.4 Centroid dan Center Cluster Pengguna 1

<b>Centroid Pengguna 1</b>						
<b>durasi</b>	<b>waktu</b>	<b>CPUusage</b>	<b>down</b>	<b>up</b>	<b>latitude</b>	<b>longitude</b>
12	1152	35.1	38.9	3.8	-7.245357	112.760086
22	897	20.3	37.3	3.2	-7.245342	112.759964
4	356	16.4	13.3	0.8	-7.245341	112.759960
2	960	14.5	11.1	0.7	-7.245348	112.759950
9	938	34.9	12.4	0.9	-7.245382	112.760040
<b>Center Cluster Pengguna 1</b>						
<b>durasi</b>	<b>waktu</b>	<b>CPUusage</b>	<b>down</b>	<b>up</b>	<b>latitude</b>	<b>longitude</b>
40	662	11.7	10.8	0.9	-7.245445	112.759895
6	790	26.2	18.4	1.3	-7.245352	112.760086
16	842	20.8	19.4	1.6	-7.245325	112.760090
71	579	12.4	16.3	1.8	-7.245247	112.760010
112	707	14.4	20.0	2.5	-7.245263	112.759926

Centroid pada tabel 4.4 adalah hasil clustering data pengguna 1 menggunakan metode K-Means. Jumlah cluster yang digunakan adalah lima cluster dan jumlah attribute adalah tujuh. Center cluster pada tabel 4.4 adalah hasil clustering data pengguna 1 menggunakan metode Fuzzy C-Means. Jumlah cluster yang digunakan adalah lima cluster, nilai fuzziness adalah 1.5, dan attribute adalah tujuh. Data centroid dan center cluster ini selanjutnya digunakan untuk proses prediksi durasi *idle*. Data centroid dan center cluster ini merupakan hasil *learning* yang menunjukkan variasi durasi *idle* untuk tiap pengguna.

### 4.3.2 Tahap Kontrol Koneksi Internet

Tahapan uji coba selanjutnya adalah kontrol koneksi Internet. Setelah sistem melakukan penyimpanan data selama 7 hari kemudian sistem memasuki mode penghematan energi baterai. Pada mode ini sistem mengambil data terkini

dari perangkat kemudian dilakukan pengecekan status *idle* dari data yang diambil. Jika data menunjukkan kondisi *smartphone* dalam kondisi *idle* maka sistem melakukan kontrol terhadap koneksi Internet.

Pada tahap ini ada beberapa proses yang di uji coba, yaitu prose penentuan kondisi *idle*, proses prediksi durasi *idle*, dan proses kontrol koneksi Internet. Berikut adalah detil hasil pengujian yang telah dilakukan:

#### 4.3.2.1 Uji Coba Penentuan Kondisi *Idle*

Penentuan kondisi *idle* dilakukan setelah sistem mendapatkan data aktivitas terkini. Setelah sistem menyimpan data, kemudian sistem menentukan status aktif atau *idle* dari data tersebut. Kondisi *idle smartphone* ditentukan berdasarkan data persentase penggunaan CPU, kondisi layar, dan status *download* perangkat. Penentuan kondisi *idle* menggunakan aturan sebagai berikut :

```
if((layar==0 || CPUusage<25) && download_status==0)
    idle=true;
else
    idle=false;
```

*Threshold* yang digunakan untuk persentase penggunaan CPU saat aktif dan *idle* adalah 25%. Jika CPU kurang dari *threshold* atau layar mati dan tidak ada proses *download*, maka kondisi *idle*. Jika ada proses *download* yang berjalan, kondisi adalah aktif meskipun kondisi layar mati dan CPU kurang dari *threshold*. Hal ini bertujuan untuk menghindari penghentian koneksi Internet saat ada proses *download* sedangkan kondisi layar mati dan penggunaan CPU rendah.

Uji coba proses penentuan kondisi *idle* dilakukan dengan memberikan perlakuan khusus terhadap *smartphone*. Perlakuan yang diberikan pada *smartphone* adalah menjalankan sebuah aplikasi pada *smartphone* agar *smartphone* aktif, kemudian semua aplikasi dan membiarkan smarthone masuk kondisi *idle*. Dengan memberikan perlakuan tersebut *smartphone* memasuki kondisi aktif di awal kemudian selanjutnya masuk kondisi *idle*. Data yang disimpan oleh sistem dan hasil penentuan kondisi *idle* ditampilkan dalam *log* untuk melihat kesesuaian data yang disimpan dan hasil penentuan kondisi *idle smartphone*. Gambar 4.5 adalah capture *log smartphone* yang menunjukkan hasil penentuan kondisi *idle smartphone*.

```
01-13 06:42:24.532 Simpan Data #command :
                                #save new data to /sdcard/dataBehavior
                                #data
                                #waktu:402; cpuusage:83; download:150; upload:37; downloa
                                d_status:0; battery:59; screen:1; charge:1; latitude:-7.2
                                453448; longitude:112.7599607
01-13 06:42:24.642 UserBehav... device status : active
01-13 06:43:24.882 Simpan Data #command :
                                #save new data to /sdcard/dataBehavior
                                #data
                                #waktu:403; cpuusage:23; download:7; upload:5; download_s
                                tatus:0; battery:59; screen:0; charge:1; latitude:-7.2453
                                448; longitude:112.7599607
01-13 06:43:25.022 UserBehav... device status : idle
```

Status aktif

Status idle

Gambar 4.5 Log Hasil Penentuan Kondisi Idle

Gambar 4.5 menunjukkan sistem melakukan dua kali penyimpanan data yaitu pada jam 06.42 dan 06.43. Pada penyimpanan data pertama sistem penentuan kondisi *smartphone* berdasarkan data yang didapat adalah aktif. Karena persentase CPUUsage tinggi (83%) dan status layar aktif. Pada penyimpanan data kedua sistem menentukan kondisi *smartphone* adalah *idle* karena persentase CPUUsage dibawah 25% dan status layar mati. Ketika hasil dari penentuan kondisi *idle* menunjukkan *smartphone* dalam kondisi *idle*, maka selanjutnya sistem melakukan proses prediksi durasi *idle* dan kontrol Internet yang akan dibahas dalam uji coba selanjutnya. Hasil pengamatan ini menunjukkan bahwa proses penentuan kondisi *idle* telah berjalan sesuai aturan yang telah dirancang.

#### 4.3.2.2 Uji Coba Prediksi Durasi Idle

Ketika perangkat memasuki kondisi *idle* maka sistem melakukan prediksi durasi *idle*. Proses prediksi waktu *idle* dilakukan dengan tiga pendekatan metode yaitu metode *k-means*, *fuzzy c-means*, dan *k-nearest neighbors*. prediksi dengan metode *k-means* menggunakan *centroid* hasil proses *learning* dengan *k-means*. Dicari jarak terdekat data baru terhadap *centroid*. Proses prediksi dengan *fuzzy c-means* dilakukan dengan menggunakan *center cluster* hasil proses pembelajaran dengan *fuzzy c-means*. Tiap data baru dicari *membership* terhadap *center cluster*. Untuk kemudian dihitung prediksi durasi *idle* berdasarkan *membership* terhadap masing-masing *cluster*. Proses prediksi menggunakan metode *k-nearest neighbors* menggunakan data pola kebiasaan pengguna hasil proses ekstraksi ciri. Pada proses



ini dicari sebanyak k tetangga terdekat terhadap data baru. Nilai k yang digunakan untuk metode *k-nearest neighbors* adalah 5.

Prediksi durasi *idle* dilakukan sebanyak 100 kali. Hasil prediksi *idle* untuk masing-masing metode dibandingkan untuk mencari hasil yang paling sesuai. Tabel 4.5 adalah sebagian sampel hasil prediksi durasi *idle* untuk masing-masing metode dengan menggunakan data pengguna 1. Hasil prediksi secara lengkap ada pada lampiran 4.

Tabel 4.5 Sample Data Hasil Prediksi Durasi *Idle*

Durasi <i>Idle</i> (menit)			
Realitas	K-Means	Fuzzy CMeans	K-NN
24	32	27	29
29	32	27	29
40	32	27	57
49	46	63	52
77	72	50	117
64	32	27	54
69	68	63	46
62	68	63	60

#### 4.3.2.3 Uji Coba Proses Kontrol Koneksi Internet

Setelah melakukan prediksi durasi *idle*, proses selanjutnya yang dilakukan oleh sistem adalah kontrol koneksi Internet berdasarkan hasil prediksi *durasi\_idle* yang didapatkan. Proses kontrol koneksi Internet dilakukan dengan durasi sesuai *durasi\_idle* hasil prediksi. Selama proses kontrol koneksi Internet, sistem menghentikan koneksi Internet secara periodik dengan interval  $1/5 * durasi\_idle$ .

Uji coba ini dilakukan untuk melihat bahwa kontrol Internet telah sesuai dengan masukan yang digunakan dan algoritma yang telah dirancang. Masukan yang digunakan adalah durasi *idle* hasil prediksi dari pengujian sebelumnya. Kontrol Internet dilakukan pada waktu perangkat memasuki kondisi *idle*. Kontrol Internet dilakukan selama durasi *idle*. Proses kontrol dilakukan dengan menghentikan dan menghidupkan koneksi Internet berdasarkan interval waktu. Interval waktu didapat dari durasi *idle* dengan perhitungan yang telah dijelaskan pada sub-bab 3.4 tentang proses kontrol Internet.

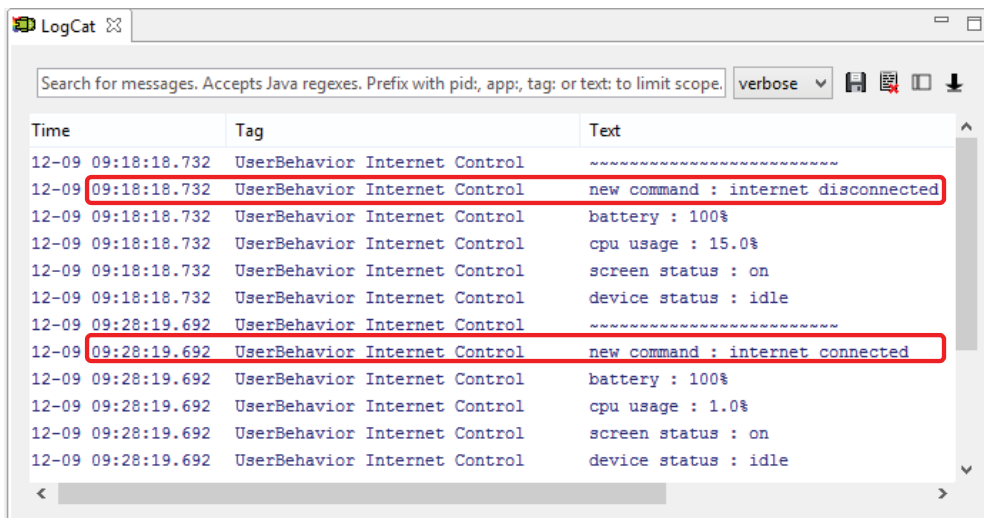
Uji coba fungsi kontrol koneksi Internet dilakukan sebanyak tiga kali dengan durasi *idle* yang telah ditentukan. Durasi *idle* yang digunakan adalah 10

menit, 60 menit, dan 180 menit. Tiga waktu durasi *idle* tersebut dipilih untuk mewakili durasi *idle* singkat, sedang, dan durasi *idle* panjang. Kontrol koneksi Internet dilakukan selama durasi *idle* yang telah ditentukan. Selama masa kontrol koneksi Internet dilakukan pemberian interval terhadap Internet. Interval yang diberikan adalah 1/5 dari durasi *idle*. Jadi selama kontrol dilakukan terjadi lima kali proses penghentian dan pengaktifan koneksi Internet. Namun interval tidak diberikan jika waktu interval kurang dari 3 menit. Berikut adalah hasil uji coba yang telah dilakukan:

- Uji coba 1

data masukan  
 $durasi\_idle = 10$  menit  
 $interval = \frac{durasi\_idle}{5} = 2$  menit

Uji coba pertama dilakukan dengan data masukan durasi *idle* 10 menit dan interval koneksi 2 menit. Interval tidak diberikan pada koneksi Internet karena interval kurang dari 3 menit. Jadi yang dilakukan oleh sistem hanya menghentikan koneksi Internet pada saat kontrol dimulai, dan kembali diaktifkan setelah kontrol berjalan selama durasi *idle* yaitu 10 menit. Pada gambar 4.6 terlihat koneksi dihentikan pada waktu 09.18 dan diaktifkan kembali pada waktu 09.28.



Gambar 4.6 Uji Coba Ke-1 Fungsi Kontrol Koneksi Internet

- Uji coba 2

data masukan:

$durasi\_idle = 60$  menit

$interval = \frac{durasi\_idle}{5} = 12$  menit



Gambar 4.7 Uji Coba Ke-2 Fungsi Kontrol Koneksi Internet

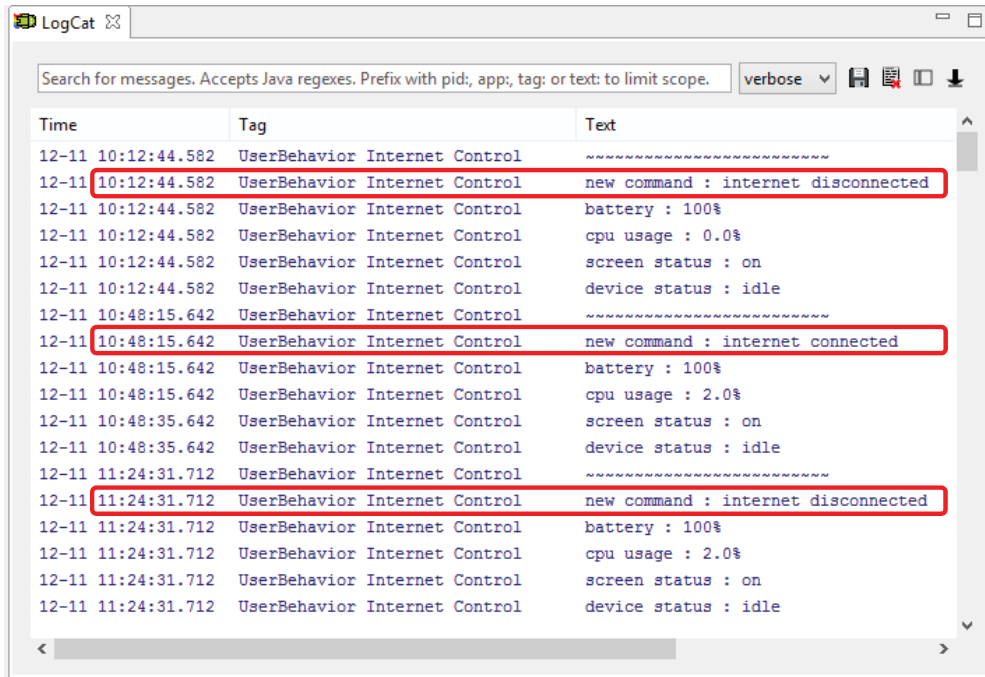
Gambar 4.7 adalah hasil uji coba kontrol koneksi Internet dengan data masukan durasi *idle* 60 menit dan interval koneksi 12 menit. Pada gambar terlihat koneksi Internet dihentikan pada waktu 10.15 kemudian pada waktu 10.27 koneksi Internet diaktifkan dan dihentikan kembali pada waktu 10.39. Interval yang terjadi adalah 12 menit. Hasil uji coba ini membuktikan bahwa interval untuk kontrol koneksi Internet telah berjalan sesuai dengan data masukan.

- Uji coba 3

data masukan:

$durasi\_idle = 180$  menit

$interval = \frac{durasi\_idle}{5} = 36$  menit



Gambar 4.8 Uji Coba Ke-3 Fungsi Kontrol Koneksi Internet

Gambar 4.8 adalah hasil uji coba kontrol koneksi Internet dengan parameter durasi *idle* 180 menit dan interval koneksi yang didapat adalah 36 menit. Pada gambar terlihat koneksi Internet dihentikan pada waktu 10.12 kemudian pada waktu 10.48 koneksi Internet diaktifkan dan dihentikan kembali pada waktu 11.24. Interval yang terjadi adalah 36 menit. Hasil uji coba ini membuktikan bahwa interval untuk kontrol koneksi Internet telah berjalan sesuai dengan data masukan.

Dari keseluruhan uji coba kontrol koneksi Internet yang telah dilakukan menunjukkan bahwa kontrol koneksi Internet telah berjalan sesuai parameter *input* yang diberikan. Tidak ada kesalahan yang terjadi dalam proses kontrol koneksi Internet.

### 4.3.3 Uji Coba Performa

Uji coba performa dilakukan untuk melihat performa dari metode yang diajukan dibandingkan dengan metode sebelumnya. Metode yang jadi pembanding adalah dua metode kontrol koneksi Internet. Metode pertama melakukan kontrol koneksi Internet berdasarkan ketersediaan daya baterai. Metode kedua melakukan kontrol koneksi Internet berdasarkan waktu *idle*. Namun berbeda dengan metode

yang diajukan pada penelitian ini, metode kedua melakukan kontrol Internet berdasarkan waktu *idle* tanpa adanya interval. Kedua metode tersebut adalah metode yang banyak digunakan dalam aplikasi-aplikasi penghemat daya baterai . Selain itu untuk mengamati performa dari metode yang diajukan dilakukan pengujian terhadap pengguna berbeda. Dengan pengujian terbut dapat diketahui efektivitas metode yang diajukan terhadap pengguna yang berbeda.

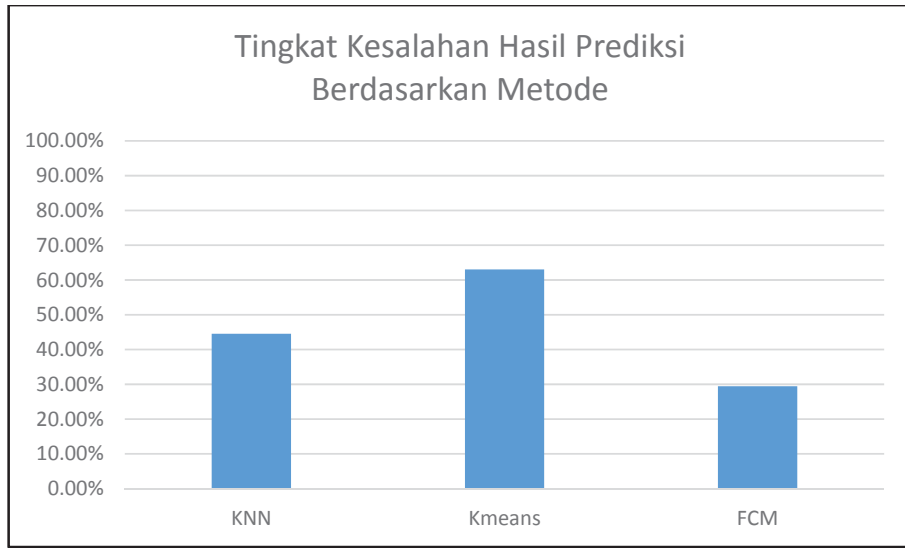
#### 4.3.3.1 Performa Metode Prediksi Durasi *Idle*

Proses prediksi waktu *idle* dilakukan dengan tiga pendekatan metode yaitu metode *k-means*, *fuzzy c-means*, dan *k-nearest neighbors*. prediksi dengan metode *k-means* menggunakan *centroid* hasil proses *learning* dengan *k-means*. Dicari jarak terdekat data baru terhadap *centroid*. Proses prediksi dengan *fuzzy c-means* dilakukan dengan menggunakan *center cluster* hasil proses pembelajaran dengan *fuzzy c-means*. Tiap data baru dicari *membership* terhadap *center cluster*. Untuk kemudian dihitung prediksi durasi *idle* berdasarkan *membership* terhadap masing-masing *cluster*. Proses prediksi menggunakan metode *k-nearest neighbors* menggunakan data pola kebiasaan pengguna hasil proses ekstraksi ciri. Pada proses ini dicari sebanyak k tetangga terdekat terhadap data baru. Nilai k yang digunakan untuk metode *k-nearest neighbors* adalah 5.

Prediksi durasi *idle* dilakukan sebanyak 100 kali. Hasil prediksi *idle* untuk masing-masing metode dibandingkan untuk mencari hasil yang paling sesuai. Untuk melihat akurasi dari masing-masing metode dicari nilai kesalahan dari tiap hasil prediksi. Rumusan yang digunakan untuk mencari tingkat kesalahan adalah sebagai berikut:

$$e = \frac{\sum_{i=1}^n ((p_i - r_i)/r_i)}{n} 100\% \quad (4.1)$$

$e$  adalah hasil perhitungan kesalahan.  $p_i$  adalah hasil prediksi ke- $i$  yang didapat sedangkan  $r_i$  adalah realitas durasi *idle* ke- $i$ .  $n$  adalah jumlah percobaan. Gambar 4.9 adalah grafik hasil perhitungan tingkat kesalahan masing-masing metode.



Gambar 4.9 Grafik Tingkat Kesalahan Metode Prediksi Durasi *Idle*

Grafik 4.9 merupakan grafik rata-rata tingkat kesalahan hasil prediksi berdasarkan tiga metode yang digunakan. Prediksi dilakukan menggunakan data dari lima pengguna. Pada grafik 4.9 terlihat hasil prediksi untuk durasi *idle* dengan metode *fuzzy c-means* memiliki tingkat kesalahan yang paling rendah diantara tiga metode yang digunakan, yaitu dengan tingkat kesalahan 29,44%. Berikut adalah rata-rata tingkat kesalahan untuk masing-masing metode dari hasil uji coba terhadap lima pengguna.

*fuzzy c-means* = 29,44 %

k-means = 63,06 %

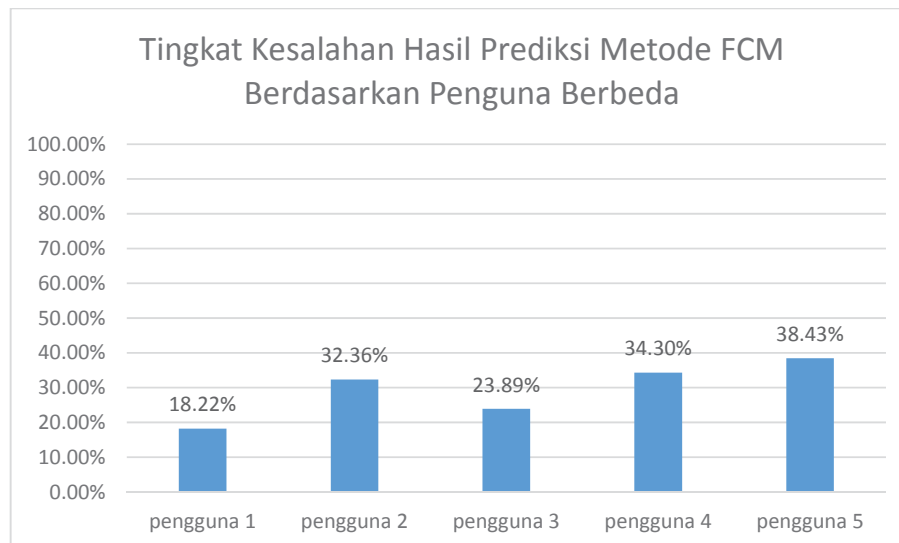
k-nearest neighbors = 44,55 %

Berdasarkan hasil uji coba didapatkan tingkat kesalahan terendah adalah menggunakan metode *fuzzy c-means*. Berdasarkan hasil pengujian ini, pengujian-pengujian selanjutnya menggunakan metode *fuzzy c-means* untuk pembelajaran pola kebiasaan pengguna dan prediksi durasi *idle*.

#### 4.3.3.2 Performa Prediksi Durasi *Idle* Berdasarkan Pengguna Berbeda

Untuk melihat efektifitas sistem terhadap pengguna berbeda, maka dilakukan analisis terhadap hasil prediksi durasi *idle* yang telah dilakukan terhadap lima pengguna berbeda. Sampel data hasil prediksi durasi *idle* untuk tiap pengguna terlampir pada lampiran 4.

Performa proses prediksi durasi *idle* dilihat dari prediksi durasi *idle* yang dihasilkan dibandingkan dengan realitas durasi *idle* sebenarnya. Dari prediksi durasi *idle* yang dihasilkan, dihitung tingkat kesalahan hasil prediksinya. Tingkat kesalahan hasil prediksi dibandingkan antar pengguna untuk melihat efektifitas sistem terhadap pola aktivitas pengguna yang berbeda.



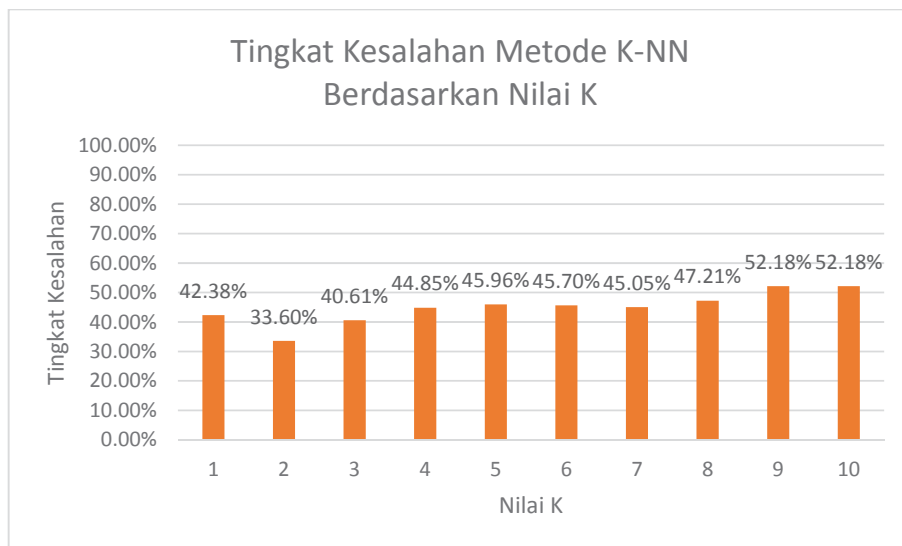
Gambar 4.10 Tingkat Kesalahan Prediksi Durasi *Idle*

Berdasarkan gambar 4.10 tingkat kesalahan terendah terjadi pada pengguna 1. Sedangkan tingkat kesalahan tertinggi adalah pengguna 5. Hal ini dikarenakan hasil prediksi durasi *idle* sangat dipengaruhi oleh pola kebiasaan masing-masing pengguna. Pengguna dengan kebiasaan penggunaan perangkat yang teratur lebih mudah dikenali dari pada kebiasaan yang sering berubah-ubah. Berdasarkan penjelasan kebiasaan masing-masing pengguna pada sub-bab 4.2, pengguna 5 memiliki rata-rata tingkat penggunaan CPU yang tinggi dengan rutinitas yang tidak teratur sehingga tingkat kesalahan prediksi durasi *idle* untuk pengguna 5 tinggi. Sedangkan pengguna 1 memiliki rata-rata tingkat penggunaan

yang rendah dan rutinitas penggunaan teratur, sehingga pengguna 1 memiliki tingkat kesalahan yang rendah.

#### 4.3.3.3 Performa Metode KNN Berdasarkan Nilai K

Uji coba ini dilakukan untuk mencari nilai K yang paling sesuai pada metode KNN. Uji coba dilakukan dengan menggunakan data dari pengguna 1. Dari data pengguna 1 dilakukan uji coba prediksi durasi *idle* menggunakan metode KNN. Parameter nilai K pada metode KNN diubah dengan beberapa nilai. Hasil dari prediksi dibandingkan dengan realitas durasi *idle* sebenarnya untuk menghitung tingkat kesalahan hasil prediksi yang didapat. Gambar 4.11 adalah grafik hasil uji coba yang telah dilakukan.



Gambar 4.11 Tingkat Kesalahan Metode KNN Berdasarkan Nilai K

Gambar 4.11 menunjukkan grafik tingkat kesalahan metode KNN berdasarkan nilai K. Dari grafik tersebut diketahui nilai K dengan tingkat kesalahan paling rendah adalah 2. Berdasarkan hasil uji coba ini untuk uji maka nilai K untuk metode KNN yang digunakan dalam penelitian ini adalah 2.

#### 4.3.3.4 Kesiapan Konektivitas Internet

Kesiapan konektivitas Internet dilihat dari kesiapan perangkat untuk melakukan transfer data melalui jaringan Internet. Uji coba dilakukan dengan melakukan pengecekan status koneksi Internet perangkat. Uji coba dilakukan dengan membandingkan kondisi perangkat saat menggunakan metode yang



diajukan dan metode pembandingan. Berikut adalah penjelasan metode yang dibandingkan:

a) Metode A (metode yang diajukan)

Metode A menggunakan metode yang diajukan dalam penelitian ini. Kontrol koneksi Internet dilakukan pada saat *idle*. selama kontrol dilakukan, koneksi dihentikan dan diaktifkan dalam interval waktu yang diberikan. Ketika perangkat dalam kondisi aktif koneksi Internet selalu aktif.

b) Metode B

Kontrol koneksi Internet pada waktu *idle* tanpa menggunakan interval. Pada metode ini koneksi Internet dihentikan selama perangkat dalam kondisi *idle*. Koneksi Internet diaktifkan kembali saat memasuki kondisi aktif. Ketika perangkat dalam kondisi aktif koneksi Internet selalu aktif.

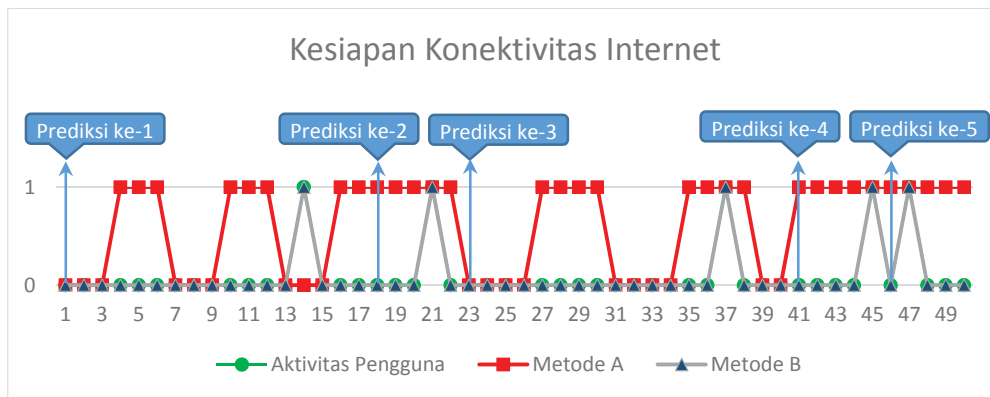
Uji coba dilakukan pada satu perangkat yang sama. Pada perangkat uji coba digunakan metode A dan B secara bergantian. Uji coba dilakukan dengan menggunakan data pengguna satu pada hari pertama, menit ke-1 hingga ke-50. Uji coba dilakukan dengan mengamati aktivitas pengguna dan koneksi Internet yang dikontrol dengan metode A dan B. Berdasarkan 50 menit data yang digunakan, terjadi lima kali proses prediksi durasi *idle* yaitu pada menit ke-1, 18, 23, 41, dan 46. Gambar 4.11 adalah grafik hasil uji coba yang telah dilakukan. Sedangkan data hasil prediksi ada pada tabel 4.6.

Tabel 4.6 Hasil Prediksi Durasi *Idle* pada Uji Coba Kesiapan Koneksi Internet

Prediksi ke	Realitas Durasi <i>Idle</i>	Prediksi Durasi <i>Idle</i>
1	14	17
2	7	5
3	16	18
4	4	5
5	1	5

Pada menit ke-1 data aktivitas menunjukkan nilai 0 yang artinya perangkat dalam kondisi *idle*. Maka pada menit tersebut sistem melakukan prediksi. Hasil prediksi pertama adalah 17 menit. Kemudian sistem melakukan kontrol terhadap koneksi Internet selama 17 menit kedepan, mulai menit ke-1 hingga menit ke-17.

Pada menit ke-18 sistem kembali melakukan pengecekan terhadap kondisi perangkat. Berdasarkan data aktivitas, pada menit 18 perangkat dalam kondisi *idle*. Maka sistem melakukan prediksi durasi *idle*. Hasil prediksi ke dua adalah 5 menit. Karena hasil prediksi menunjukkan durasi yang terlalu singkat, maka sistem tidak menghentikan koneksi Internet untuk lima menit kedepan. Kemudian kembali melakukan pengecekan dan prediksi pada menit 23, dengan hasil prediksi adalah 18 menit. Maka sistem melakukan kontrol selama 18 menit. Prediksi ke empat dan kelima terjadi pada menit ke-41 dan 46. Hasil dari dua prediksi terakhir adalah lima menit. Sehingga sistem tidak melakukan penghentian koneksi Internet. Hasil prediksi dan kontrol yang dilakukan digambarkan pada grafik 4.12.



Gambar 4.12 Uji Coba Kesiapan Koneksi Internet

Gambar 4.12 adalah hasil uji coba kesiapan konektivitas Internet. Pada grafik ditampilkan data aktivitas pengguna pada hari pertama, menit ke-1 hingga ke-50. Data terlampir pada lampiran 5. Aktivitas pengguna bernilai 0 (nol) menunjukkan tidak ada aktivitas pada menit tersebut, atau perangkat dalam kondisi *idle*. Sedangkan Aktivitas bernilai 1 menunjukkan pada waktu tersebut ada aktivitas yang dilakukan oleh pengguna. Data metode A dan metode B didapat dari ketersediaan konektivitas Internet. Pada gambar terlihat dengan metode B koneksi Internet selalu tersedia ketika pengguna melakukan aktivitas. Sedangkan pada saat *idle* tidak ada koneksi Internet.

Metode A (metode yang diajukan) melakukan kontrol koneksi Internet pada saat *idle* menggunakan interval. Pada saat perangkat memasuki kondisi *idle* maka sistem melakukan prediksi durasi *idle* yang akan terjadi. Pada gambar

tersebut terlihat pada menit pertama kondisi perangkat adalah *idle*. Maka pada waktu tersebut sistem melakukan prediksi durasi *idle*. Hasil prediksi pertama ini adalah 17 menit. Sehingga koneksi Internet dikontrol selama 17 menit. Selama proses kontrol, koneksi Internet dihentikan dan diaktifkan secara berulang dengan interval waktu tertentu. Interval waktu yang digunakan adalah  $1/5 * \text{durasi } idle$ . Jika pada prediksi pertama didapatkan durasi *idle* adalah 17 menit, maka interval yang digunakan adalah 3 menit. Pada 3 menit pertama koneksi Internet dihentikan. Kemudian pada 3 menit setelahnya (menit ke-4) koneksi Internet diaktifkan, dan dihentikan kembali pada menit ke-7. Proses ini terus berjalan hingga waktu kontrol usai.

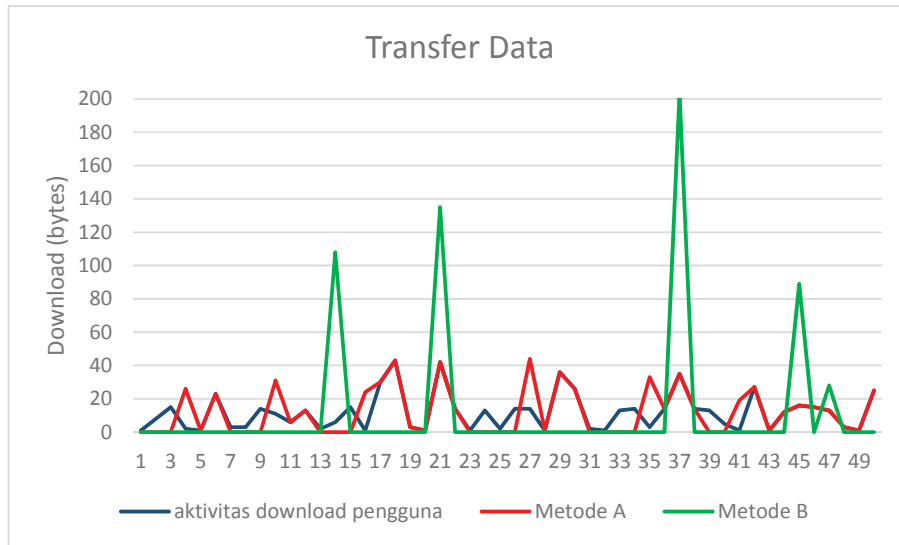
Hasil uji coba ini membuktikan bahwa metode yang kami ajukan berhasil melakukan kontrol koneksi Internet dengan menggunakan interval. Koneksi Internet dihentikan pada saat perangkat memasuki kondisi *idle*. Namun dalam proses kontrol koneksi Internet, metode yang diajukan melakukan penghentian dan pengaktifan koneksi Internet secara berulang berdasarkan interval waktu tertentu. Dengan penggunaan interval ini koneksi Internet lebih banyak tersedia dibandingkan dengan metode B.

#### 4.3.3.5 Kelancaran Proses Transfer Data

Pengujian ini dilakukan untuk melihat hambatan yang terjadi akibat dari proses kontrol koneksi Internet. Pengujian ini berkaitan dengan penggunaan interval pada metode yang kami ajukan. Penggunaan interval bertujuan untuk menghindari adanya paket data yang terhambat dalam jangka waktu yang lama akibat dari penghentian koneksi Internet. Pada pengujian ini juga dilakukan perbandingan antara metode yang diajukan dengan metode lain. Metode yang menjadi pembanding adalah metode kontrol koneksi Internet berdasarkan waktu *idle* tanpa menggunakan interval. Metode tersebut dipilih sebagai pembanding karena memiliki kesamaan dalam kontrol koneksi Internet yaitu berdasarkan waktu *idle*.

Uji coba dilakukan dengan melihat lalu lintas *download* data yang terjadi. Data yang digunakan adalah data pengguna 1 pada hari pertama, menit ke=1 hingga ke-50. Data *download* pada pengguna 1 ditampilkan pada grafik dan dibandingkan

dengan data *download* jika metode A (metode yang diusulkan) dan metode B digunakan. Gambar 4.13 adalah grafik hasil uji coba proses transfer data.



Gambar 4.13 Grafik Uji Coba Transfer Data

Gambar 4.13 menunjukkan aktivitas *download* pengguna, aktivitas *download* ketika menggunakan metode A, dan aktivitas *download* ketika menggunakan metode B. Pada data aktivitas *download* dengan metode B terlihat proses *download* sering berhenti dengan waktu yang lama. Penghentian *download* ini terjadi karena koneksi Internet dihentikan. Pada metode B, koneksi Internet dihentikan selama perangkat dalam kondisi *idle*. Jika kondisi *idle* terjadi dalam waktu yang lama maka tidak ada proses transfer data dalam jangka waktu yang lama. Proses transfer data berlanjut ketika koneksi Internet diaktifkan lagi. Pada saat itu antrian data yang akan masuk pada perangkat sudah cukup banyak. Sehingga ketika koneksi Internet aktif, proses transfer data langsung meningkat dengan sangat drastis.

Berbeda dengan metode B, metode yang kami ajukan melakukan kontrol koneksi Internet berdasarkan *idle* dan menggunakan interval. Ketika perangkat memasuki kondisi *idle* Internet dihentikan. Namun ada interval pengaktifan kembali koneksi Internet ditengah-tengah kondisi *idle*. Hal ini bertujuan untuk menghindari terjadinya hambatan dalam waktu yang cukup lama untuk proses transfer data. Pada gambar 4.13 terlihat proses transfer data terhenti ketika

perangkat memasuki kondisi *idle*, namun dalam kondisi ini perangkat masih bisa melanjutkan proses transfer data secara berkala.

Pengujian ini membuktikan bahwa metode yang kami ajukan memiliki keunggulan dalam hal kelancaran proses transfer data dibandingkan dengan metode B. Ketika proses kontrol Internet berlangsung, proses transfer data masih dapat terjadi secara berkala sesuai interval waktu yang diberikan. Sehingga tidak ada paket data yang terhambat dalam waktu yang sangat lama akibat proses pengontrolan koneksi Internet.

#### 4.3.3.6 Penghematan Konsumsi Energi

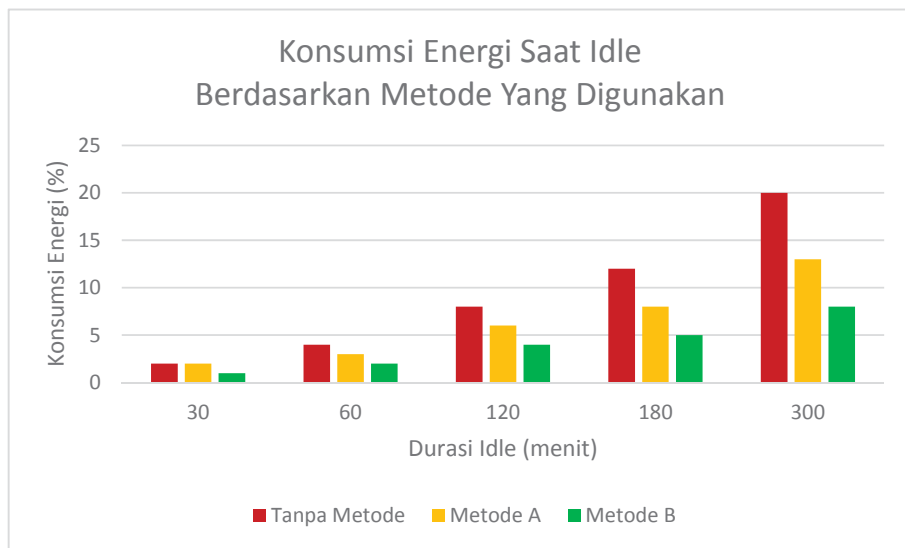
Pengujian ini dilakukan untuk melihat seberapa besar konsumsi energi dapat dihemat dengan menggunakan metode yang kami ajukan. Pada pengujian ini juga dilakukan perbandingan antara metode yang diajukan (metode A) dengan metode lain. Metode yang menjadi pembanding adalah metode kontrol koneksi Internet berdasarkan waktu *idle* tanpa menggunakan interval (metode B). Metode tersebut dipilih sebagai pembanding karena memiliki kesamaan dalam kontrol koneksi Internet yaitu berdasarkan waktu *idle*. Pada uji coba ini juga diamati ketersediaan energi baterai dengan kondisi perangkat tanpa menggunakan metode apapun. Uji coba dilakukan pada kondisi *idle* dengan durasi *idle* tertentu. Ketersediaan energi baterai diamati ketika perangkat memasuki kondisi *idle* dan ketika kondisi *idle* berakhir. Berikut adalah hasil uji coba yang telah dilakukan:

Tabel 4.7 Hasil Uji Coba Konsumsi Energi

Durasi <i>Idle</i> (menit)	Konsumsi Energi (%)		
	Tanpa metode apapun	Metode A	Metode B
30	2	2	1
60	4	3	2
120	8	6	4
180	12	8	5
300	20	13	8

Tabel 4.7 adalah hasil monitoring ketersediaan energi baterai selama waktu *idle*. Pada gambar terlihat dengan menggunakan metode B terjadi

penghematan energi yang cukup besar dibandingkan dengan kondisi perangkat tidak menggunakan metode apapun. Ketika perangkat menggunakan metode A (metode yang kami ajukan) juga terjadi penghematan energi baterai namun tidak sebesar metode B. Hal ini disebabkan adanya penggunaan interval pada metode kami. Dengan adanya interval, koneksi Internet tetap terjadi dalam interval waktu tertentu dimana koneksi Internet ini mengkonsumsi energi baterai. Sedangkan dengan metode A koneksi Internet benar-benar terhenti ketika *idle* sehingga tidak ada konsumsi energi untuk koneksi Internet.



Gambar 4.14 Konsumsi Energi Baterai *Smartphone* Saat Kondisi *Idle*

Pada gambar 4.14 terlihat bahwa dengan durasi *idle* berbeda, berbeda pula energi baterai yang dihemat pada metode A ataupun B. Semakin lama durasi *idle* terjadi semakin besar energi baterai yang dihemat. Hal ini membuktikan bahwa besarnya energi yang dihemat dengan metode yang kami ajukan sangat tergantung pada lama kondisi *idle* terjadi.

## **BAB 5**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Berdasarkan hasil uji coba dan analisis yang telah dilakukan dapat diambil kesimpulan sebagai berikut:

- Dengan mengamati dan melakukan pembelajaran terhadap pola kebiasaan pengguna kita bisa memprediksi berapa lama perangkat berada dalam kondisi *idle*.
- Di antara tiga metode (*K-means*, KNN, dan FCM) yang digunakan, metode dengan hasil prediksi durasi waktu *idle* terbaik adalah metode FCM. Metode FCM memiliki tingkat kesalahan yang paling rendah yaitu sekitar 29%.
- Tingkat keberhasilan pengenalan pola kebiasaan pengguna sangat dipengaruhi oleh rutinitas pengguna dalam menggunakan . Semakin teratur rutinitas pengguna, semakin mudah sistem mengenali kebiasaan pengguna dan semakin baik hasil prediksi durasi *idle* yang didapat.
- Konsumsi daya yang sia-sia pada saat *idle* berhasil dikurangi. Penghematan daya tergantung pada durasi dan intensitas waktu *idle*. Semakin lama dan sering kondisi *idle* terjadi maka daya yang dihemat semakin besar.
- Metode yang diajukan memiliki keunggulan dalam hal kenyamanan pengguna. Kesiapan konektivitas Internet selalu tersedia ketika pengguna aktif menggunakan perangkat. Selain itu proses transfer data masih tetap berjalan dengan interval waktu tertentu ketika memasuki kondisi *idle*. Sehingga pengguna tidak terganggu dengan adanya paket data yang terhambat dengan waktu yang cukup lama.

## 5.2 Saran

Kontrol koneksi Internet untuk mengurangi konsumsi energi pada perangkat *smartphone* merupakan topik yang menarik untuk dikembangkan. Dari banyak metode yang telah diajukan tentunya masih banyak aspek yang bisa dikembangkan. Salah satunya adalah optimasi kesiapan konektivitas Internet seperti yang dilakukan dalam penelitian ini. Untuk penelitian lebih lanjut pendekatan yang kami ajukan dapat dikembangkan pada aspek pengenalan pola kebiasaan pengguna untuk prediksi durasi *idle*. Penggunaan metode lain untuk pengenalan pola dapat digunakan untuk mencari hasil prediksi yang lebih baik. Selain itu pemilihan data atribut untuk digunakan dalam proses analisis kebiasaan pengguna masih membutuhkan penelitian lebih lanjut untuk mendapatkan hasil yang lebih baik.



## Lampiran 1. Sampel Data Awal Pengguna

### Sampel Data Awal Pengguna 1

waktu	CPUusage	download	upload	lat	lon	Download status	battery	layar	charge
1	5	1	0	-7.2453363	112.7599682	0	89	1	0
2	9	8	0	-7.2453363	112.7599682	0	89	1	0
3	10	15	1	-7.2453374	112.7599700	0	89	1	0
4	4	2	0	-7.2453374	112.7599700	0	89	1	0
5	4	1	0	-7.2453374	112.7599700	0	89	1	0
6	4	23	0	-7.2453374	112.7599610	0	89	1	0
7	5	3	0	-7.2453374	112.7599610	0	89	1	0
8	5	3	0	-7.2453374	112.7599610	0	89	1	0
9	10	14	0	-7.2453374	112.7599614	0	89	1	0
10	11	11	0	-7.2453374	112.7599614	0	89	1	0
11	5	6	0	-7.2453374	112.7599614	0	89	1	0
12	5	13	0	-7.2453374	112.7599612	0	89	1	0
13	4	2	0	-7.2453374	112.7599612	0	89	1	0
14	33	6	0	-7.2453374	112.7599612	0	89	1	0
15	6	15	0	-7.2453374	112.7599708	0	89	1	0
16	4	1	0	-7.2453374	112.7599708	0	89	1	0
17	13	30	3	-7.2453374	112.7599708	0	89	1	0
18	10	43	2	-7.2453374	112.7599615	0	89	1	0
19	4	3	0	-7.2453374	112.7599615	0	89	1	0
20	5	1	0	-7.2453374	112.7599615	0	89	1	0
21	28	42	4	-7.2453427	112.7599656	0	89	1	0
22	6	14	0	-7.2453427	112.7599656	0	89	1	0
23	9	1	0	-7.2453427	112.7599656	0	89	1	0
24	4	13	0	-7.2453374	112.7599656	0	89	1	0
25	5	2	0	-7.2453374	112.7599656	0	89	1	0
26	4	14	0	-7.2453374	112.7599656	0	89	1	0
27	7	14	1	-7.2453374	112.7599708	0	89	1	0
28	5	1	0	-7.2453374	112.7599708	0	89	1	0
29	13	36	4	-7.2453374	112.7599708	0	89	1	0
30	5	26	0	-7.2453327	112.7599656	0	89	1	0

### Sampel Data Awal Pengguna 2

waktu	CPUusage	download	upload	lat	lon	Download status	battery	layar	charge
1	28	0	0	-7.245345	112.759964	0	47	0	1
2	2	0	0	-7.2470546	112.75914	0	47	0	1
3	64	0	0	-7.245344	112.75996	0	47	0	1
4	3	0	0	-7.245344	112.75996	0	47	0	1
5	2	0	0	-7.245343	112.759964	0	47	0	1
6	20	0	0	-7.2453437	112.75996	0	47	0	1
7	4	0	0	-7.245344	112.75996	0	47	0	1
8	18	0	0	-7.2453437	112.759964	0	47	0	1
9	10	11	1	-7.2453437	112.75996	0	47	0	1
10	7	0	0	-7.245344	112.759964	0	47	0	1
11	30	13	1	-7.245344	112.759964	0	47	0	1
12	5	0	0	-7.2453437	112.759964	0	47	0	1
13	8	13	1	-7.245344	112.759964	0	47	0	1
14	3	0	0	-7.2453437	112.75996	0	47	0	1
15	3	13	0	-7.245344	112.759964	0	47	0	1
16	17	1	0	-7.245344	112.75996	0	47	0	1
17	6	13	1	-7.245343	112.759964	0	47	0	1
18	3	12	1	-7.2453446	112.759964	0	47	0	1
19	9	1	0	-7.2453437	112.75996	0	47	0	1
20	3	0	0	-7.2453437	112.75996	0	47	0	1
21	11	13	0	-7.245344	112.75996	0	47	0	1
22	3	13	0	-7.245344	112.75996	0	47	0	1
23	7	13	1	-7.2453437	112.759964	0	47	0	1
24	29	0	0	-7.245344	112.75996	0	47	0	1
25	13	1	0	-7.2453437	112.75996	0	47	0	1
26	4	13	0	-7.245343	112.759964	0	47	0	1
27	18	0	0	-7.245344	112.759964	0	47	0	1
28	31	0	0	-7.245345	112.75996	0	47	0	1
29	29	3	2	-7.2453446	112.75997	0	47	0	1
30	4	14	1	-7.245344	112.759964	0	47	0	1

### Sampel Data Awal Pengguna 3

waktu	CPUusage	download	upload	lat	lon	Download status	battery	layar	charge
1	5	13	0	-7.2453448	112.7599607	0	47	1	1
2	4	16	0	-7.2453448	112.7599607	0	47	1	1
3	4	9	0	-7.2453448	112.7599607	0	47	1	1
4	4	4	0	-7.2453437	112.7599612	0	47	1	1
5	5	1	0	-7.2453437	112.7599612	0	47	1	1
6	5	28	0	-7.2453436	112.7599588	0	47	1	1
7	4	1	0	-7.2453436	112.7599588	0	47	1	1
8	5	1	0	-7.2453436	112.7599588	0	47	1	1
9	6	15	0	-7.2453432	112.7599584	0	47	1	1
10	5	14	0	-7.2453432	112.7599584	0	47	1	1
11	4	1	0	-7.2453432	112.7599584	0	47	1	1
12	5	15	0	-7.2453447	112.7599584	0	47	1	1
13	15	1	0	-7.2453447	112.7599584	0	47	1	1
14	5	17	0	-7.2453447	112.7599584	0	47	1	1
15	5	13	0	-7.2453439	112.7599587	0	47	1	1
16	4	1	0	-7.2453439	112.7599587	0	47	1	1
17	5	2	0	-7.2453439	112.7599587	0	47	1	1
18	6	26	0	-7.2453440	112.7599583	0	47	1	1
19	4	2	0	-7.2453440	112.7599583	0	47	1	1
20	4	1	0	-7.2453440	112.7599583	0	47	1	1
21	6	13	0	-7.2453437	112.7599612	0	47	1	1
22	4	16	0	-7.2453437	112.7599612	0	47	1	1
23	3	1	0	-7.2453437	112.7599612	0	47	1	1
24	5	13	0	-7.2453443	112.7599587	0	47	1	1
25	5	2	0	-7.2453443	112.7599587	0	47	1	1
26	4	14	0	-7.2453443	112.7599587	0	47	1	1
27	6	13	0	-7.2453438	112.7599583	0	47	1	1
28	4	1	0	-7.2453438	112.7599583	0	47	1	1
29	5	2	0	-7.2453438	112.7599583	0	47	1	1
30	4	15	1	-7.2453436	112.7599585	0	47	1	1

#### Sampel Data Awal Pengguna 4

waktu	CPUusage	download	upload	lat	lon	Download status	battery	layar	charge
1	6	8	1	-7.2453443	112.7599578	0	79	0	0
2	10	0	0	-7.2453443	112.7599578	0	79	0	0
3	9	0	0	-7.2453445	112.7599613	0	79	0	0
4	6	0	0	-7.2453445	112.7599613	0	79	0	0
5	8	8	1	-7.2453445	112.7599613	0	79	0	0
6	0	0	0	-7.2453448	112.7599607	0	79	0	0
7	16	0	0	-7.2453448	112.7599607	0	79	0	0
8	0	0	0	-7.2453448	112.7599607	0	79	0	0
9	2	8	1	-7.2453432	112.7599603	0	79	0	0
10	0	0	0	-7.2453432	112.7599603	0	79	0	0
11	8	0	0	-7.2453432	112.7599603	0	79	0	0
12	1	9	1	-7.2453438	112.7599583	0	79	0	0
13	2	0	0	-7.2453438	112.7599583	0	79	0	0
14	4	0	0	-7.2453438	112.7599583	0	79	0	0
15	2	8	1	-7.2453457	112.7599598	0	79	0	0
16	0	0	0	-7.2453457	112.7599598	0	79	0	0
17	3	0	0	-7.2453437	112.7599612	0	79	0	0
18	1	8	1	-7.2453437	112.7599612	0	79	0	0
19	3	0	0	-7.2453437	112.7599612	0	79	0	0
20	6	0	0	-7.2453436	112.7599585	0	79	0	0
21	2	0	0	-7.2453436	112.7599585	0	79	0	0
22	8	10	1	-7.2453436	112.7599585	0	79	0	0
23	2	0	0	-7.2453436	112.7599585	0	79	0	0
24	7	8	1	-7.2453436	112.7599585	0	79	0	0
25	1	0	0	-7.2453436	112.7599585	0	79	0	0
26	3	0	0	-7.2453440	112.7599583	0	79	0	0
27	5	0	0	-7.2453440	112.7599583	0	79	0	0
28	59	61	12	-7.2453440	112.7599583	0	79	0	0
29	100	162	28	-7.2453436	112.7599585	0	79	0	0
30	94	304	26	-7.2453436	112.7599585	0	79	0	0

**Sampel Data Awal Pengguna 5**

waktu	CPUusage	download	upload	lat	lon	Download status	battery	layar	charge
1	22	70	8	-7.2453440	112.7599639	0	61	0	0
2	23	78	8	-7.2453440	112.7599639	0	61	0	0
3	21	51	6	-7.2453440	112.7599549	0	61	0	0
4	25	79	10	-7.2453440	112.7599549	0	61	0	0
5	23	69	9	-7.2453489	112.7599595	0	61	0	0
6	20	61	8	-7.2453489	112.7599595	0	61	0	0
7	39	71	10	-7.2453489	112.7599595	0	61	0	0
8	62	93	9	-7.2453442	112.7599583	0	61	0	0
9	87	103	37	-7.2453442	112.7599583	0	62	0	0
10	52	79	8	-7.2453442	112.7599583	0	62	0	0
11	52	52	9	-7.2453402	112.7599557	0	62	0	0
12	43	69	7	-7.2453402	112.7599557	0	62	0	0
13	19	79	28	-7.2453402	112.7599557	0	62	0	0
14	15	49	5	-7.2453438	112.7599583	0	62	0	0
15	21	14	3	-7.2453438	112.7599583	0	62	0	0
16	31	147	18	-7.2453438	112.7599583	0	62	0	0
17	11	35	5	-7.2453457	112.7599612	0	62	0	0
18	16	14	0	-7.2453457	112.7599612	0	62	0	0
19	2	17	0	-7.2453440	112.7599637	0	62	0	0
20	3	30	0	-7.2453440	112.7599637	0	62	0	0
21	35	96	19	-7.2453440	112.7599637	0	62	0	0
22	30	97	16	-7.2453457	112.7599612	0	62	0	0
23	18	89	16	-7.2453457	112.7599612	0	62	0	0
24	28	88	15	-7.2453457	112.7599612	0	62	0	0
25	30	105	15	-7.2453440	112.7599612	0	62	0	0
26	36	132	23	-7.2453440	112.7599612	0	62	0	0
27	22	72	13	-7.2453440	112.7599612	0	62	0	0
28	27	65	14	-7.2453440	112.7599644	0	62	0	0
29	29	66	12	-7.2453440	112.7599644	0	62	0	0
30	52	51	9	-7.2453440	112.7599644	0	62	0	0

## Lampiran 2. Sampel Data Hasil Ekstraksi

### Sampel Hasil Ekstraksi Data Pengguna 1

<i>durasi</i>	<i>waktu</i>	<i>CPUusage</i>	<i>down</i>	<i>up</i>	<i>latitude</i>	<i>longitude</i>	<b>Down status</b>	<i>battery</i>	<i>layar</i>	<i>charge</i>
14	0.001	0.081	0.115	0.003	-0.0805	0.626444	0	0.890	1	0
7	0.010	0.100	0.288	0.051	-0.0805	0.626444	0	0.890	1	0
8	0.026	0.149	0.166	0.015	-0.0805	0.626444	0	0.890	1	0
2	0.032	0.155	0.209	0.000	-0.0805	0.626444	0	0.890	1	0
87	0.033	0.084	0.123	0.022	-0.08049	0.626445	0	0.890	0	0
7	0.094	0.100	0.288	0.051	-0.0805	0.626444	0	0.890	1	0
16	0.099	0.075	0.190	0.023	-0.0805	0.626444	0	0.871	1	0
8	0.110	0.149	0.166	0.015	-0.0805	0.626444	0	0.870	1	0
2	0.115	0.155	0.209	0.000	-0.0805	0.626444	0	0.870	1	0
48	0.117	0.076	0.221	0.010	-0.0805	0.626444	0	0.870	1	0
7	0.255	0.167	0.301	0.011	-0.0805	0.626444	0	0.850	1	0
4	0.260	0.173	0.205	0.020	-0.0805	0.626444	0	0.850	1	0
6	0.263	0.370	0.301	0.087	-0.0805	0.626444	0	0.850	1	0
2	0.267	0.375	0.501	0.120	-0.0805	0.626444	0	0.850	1	0
5	0.275	0.368	0.559	0.080	-0.0805	0.626444	0	0.850	1	0
3	0.278	0.210	0.573	0.080	-0.0805	0.626444	0	0.850	1	0
17	0.281	0.391	0.175	0.026	-0.0805	0.626444	0	0.850	1	0
3	0.356	0.313	0.105	0.027	-0.0805	0.626444	0	0.850	1	0
33	0.358	0.122	0.095	0.030	-0.0805	0.626444	0	0.824	0	0
75	0.381	0.069	0.258	0.012	-0.0805	0.626444	0	0.820	0	0
4	0.433	0.450	0.262	0.030	-0.0805	0.626444	0	0.820	1	0
7	0.435	0.127	0.085	0.023	-0.0805	0.626444	0	0.820	1	0
58	0.440	0.071	0.048	0.012	-0.0805	0.626444	0	0.789	0	0
102	0.483	0.187	0.648	0.236	-0.0805	0.626444	0	0.766	0	0
4	0.560	0.528	0.086	0.010	-0.0805	0.626444	0	0.730	1	0
4	0.563	0.390	0.071	0.010	-0.0805	0.626444	0	0.730	1	0
4	0.565	0.145	0.071	0.020	-0.0805	0.626444	0	0.730	1	0
91	0.568	0.123	0.045	0.009	-0.0805	0.626444	0	0.730	0	0
26	0.631	0.077	0.332	0.052	-0.0805	0.626444	0	0.730	1	0
17	0.649	0.169	0.237	0.012	-0.0805	0.626444	0	0.659	1	0

## Sampel Hasil Ekstraksi Data Pengguna 2

<i>durasi</i>	<i>waktu</i>	<i>CPUUsage</i>	<i>down</i>	<i>up</i>	<i>latitude</i>	<i>longitude</i>	<i>Down status</i>	<i>battery</i>	<i>layar</i>	<i>charge</i>
35	0.001	0.159	0.019	0.013	-0.0805	0.626444	0	0.470	0	1
8	0.025	0.308	0.055	0.046	-0.0805	0.626444	0	0.470	1	1
5	0.031	0.118	0.045	0.000	-0.0805	0.626444	0	0.470	1	1
8	0.034	0.096	0.021	0.000	-0.0805	0.626444	0	0.470	1	1
3	0.040	0.227	0.006	0.027	-0.0805	0.626444	0	0.470	1	1
19	0.043	0.294	0.006	0.009	-0.0805	0.626444	0	0.470	0	1
4	0.056	0.188	0.014	0.021	-0.0805	0.626444	0	0.470	1	1
3	0.059	0.167	0.009	0.014	-0.0805	0.626444	0	0.470	1	1
46	0.061	0.136	0.008	0.007	-0.0805	0.626444	0	0.520	0	1
61	0.093	0.223	0.063	0.149	-0.0805	0.626444	0	0.580	0	1
43	0.135	0.077	0.014	0.018	-0.0805	0.626444	0	0.580	1	1
2	0.165	0.205	0.019	0.021	-0.08052	0.626491	0	0.580	1	1
10	0.167	0.483	0.042	0.103	-0.08052	0.626444	0	0.580	1	1
53	0.174	0.114	0.039	0.032	-0.08052	0.626444	0	0.580	0	1
4	0.210	0.403	0.015	0.021	-0.08052	0.626444	0	0.580	1	1
6	0.213	0.200	0.015	0.021	-0.08052	0.626444	0	0.580	1	1
50	0.217	0.066	0.025	0.019	-0.0805	0.626444	0	0.580	1	1
4	0.252	0.403	0.015	0.021	-0.0805	0.626444	0	0.765	1	1
97	0.259	0.062	0.029	0.017	-0.0805	0.626444	0	0.950	1	1
13	0.326	0.163	0.035	0.022	-0.0805	0.626444	0	0.950	1	1
4	0.335	0.403	0.015	0.021	-0.0805	0.626444	0	0.950	1	1
6	0.338	0.200	0.015	0.021	-0.0805	0.626444	0	0.950	1	1
54	0.342	0.074	0.028	0.018	-0.0805	0.626444	0	0.969	1	1
2	0.381	0.290	0.045	0.021	-0.08053	0.626463	0	1.000	1	1
8	0.392	0.106	0.041	0.010	-0.08052	0.626444	0	1.000	1	1
37	0.398	0.267	0.043	0.053	-0.08052	0.626444	0	1.000	0	1
58	0.424	0.120	0.052	0.078	-0.08052	0.626444	0	1.000	0	0
6	0.464	0.137	0.089	0.034	-0.0805	0.626444	0	1.000	0	0
8	0.468	0.541	0.029	0.010	-0.0805	0.626444	0	1.000	1	0
17	0.474	0.121	0.036	0.019	-0.0805	0.626444	0	1.000	1	0

### Sampel Hasil Ekstraksi Data Pengguna 3

<i>durasi</i>	<i>waktu</i>	<i>CPUusage</i>	<i>down</i>	<i>up</i>	<i>latitude</i>	<i>longitude</i>	<b>Down status</b>	<i>battery</i>	<i>layar</i>	<i>charge</i>
119	0.001	0.058	0.136	0.021	-0.0805	0.626444	0	0.532	0	1
43	0.083	0.069	0.136	0.033	-0.0805	0.626444	0	0.944	1	0
7	0.113	0.187	0.102	0.024	-0.0805	0.626444	0	0.950	1	0
7	0.197	0.187	0.102	0.024	-0.0805	0.626444	0	0.950	1	0
24	0.201	0.145	0.246	0.038	-0.0805	0.626444	0	0.831	1	0
10	0.224	0.193	0.270	0.029	-0.0805	0.626444	0	0.730	1	0
3	0.231	0.330	0.184	0.019	-0.0805	0.626444	0	0.730	1	0
6	0.233	0.333	0.404	0.057	-0.0805	0.626444	0	0.713	1	0
11	0.237	0.355	0.168	0.042	-0.0805	0.626444	0	0.630	1	0
6	0.244	0.233	0.168	0.076	-0.0805	0.626444	0	0.630	1	0
4	0.249	0.315	0.201	0.129	-0.0805	0.626444	0	0.660	1	0
9	0.251	0.128	0.141	0.006	-0.0805	0.626444	0	0.750	1	0
24	0.258	0.110	0.200	0.050	-0.0805	0.626444	0	0.750	1	0
3	0.274	0.380	0.147	0.152	-0.0805	0.626444	0	0.750	1	0
3	0.282	0.140	0.050	0.038	-0.0805	0.626444	0	0.750	1	0
4	0.284	0.245	0.176	0.071	-0.0805	0.626444	0	0.710	1	0
17	0.287	0.519	0.219	0.343	-0.0805	0.626444	0	0.371	1	0
7	0.299	0.603	1.000	0.253	-0.0805	0.626444	0	0.070	1	0
5	0.303	0.158	0.098	0.023	-0.0805	0.626444	0	0.070	1	0
9	0.307	0.284	0.170	0.038	-0.0805	0.626444	0	0.070	1	0
39	0.331	0.296	0.647	0.467	-0.0805	0.626444	0	0.218	0	0
12	0.358	0.309	0.441	0.343	-0.0805	0.626444	0	0.240	0	0
23	0.366	0.406	0.593	0.370	-0.0805	0.626444	0	0.173	0	0
7	0.382	0.603	0.930	0.253	-0.0805	0.626444	0	0.070	1	0
5	0.387	0.158	0.098	0.023	-0.0805	0.626444	0	0.070	1	0
9	0.390	0.284	0.170	0.038	-0.0805	0.626444	0	0.070	1	0
12	0.397	0.103	0.103	0.048	-0.0805	0.626444	0	0.070	1	0
13	0.405	0.077	0.059	0.000	-0.0805	0.626444	0	0.070	1	0
10	0.419	0.382	0.243	0.063	-0.0805	0.626444	0	0.230	1	0
3	0.426	0.180	0.197	0.000	-0.0805	0.626444	0	0.230	1	0



#### Sampel Hasil Ekstraksi Data Pengguna 4

<i>durasi</i>	<i>waktu</i>	<i>CPUusage</i>	<i>down</i>	<i>up</i>	<i>latitude</i>	<i>longitude</i>	<i>Down status</i>	<i>battery</i>	<i>Layar</i>	<i>charge</i>
125	0.001	0.117	0.047	0.084	-0.0805	0.626444	0	0.787	0	0
4	0.088	0.338	0.012	0.010	-0.0805	0.626444	0	0.688	1	0
76	0.090	0.078	0.016	0.011	-0.0805	0.626444	0	0.624	0	0
4	0.143	0.218	0.043	0.113	-0.0805	0.626444	0	0.510	1	1
12	0.146	0.115	0.042	0.010	-0.0805	0.626444	0	0.510	1	1
13	0.154	0.125	0.037	0.028	-0.08053	0.626444	0	0.519	1	1
3	0.163	0.283	0.037	0.041	-0.0805	0.626444	0	0.630	1	1
3	0.165	0.370	0.029	0.027	-0.0805	0.626444	0	0.630	1	1
62	0.167	0.120	0.049	0.085	-0.0805	0.626444	0	0.783	0	0
7	0.210	0.361	0.047	0.135	-0.0805	0.626444	0	0.560	1	0
9	0.215	0.150	0.027	0.018	-0.0805	0.626444	0	0.551	1	0
6	0.222	0.163	0.028	0.021	-0.0805	0.626444	0	0.520	1	0
7	0.226	0.123	0.044	0.076	-0.0805	0.626444	0	0.520	1	0
39	0.231	0.256	0.045	0.222	-0.0805	0.626444	0	0.423	1	0
9	0.258	0.334	0.028	0.023	-0.0805	0.626444	0	0.140	0	0
2	0.264	0.380	0.057	0.164	-0.0805	0.626444	0	0.140	1	0
4	0.265	0.405	0.035	0.041	-0.0805	0.626444	0	0.140	1	0
10	0.268	0.227	0.041	0.012	-0.0805	0.626444	0	0.140	0	0
14	0.275	0.139	0.038	0.032	-0.0805	0.626444	0	0.140	1	0
7	0.285	0.390	0.044	0.029	-0.0805	0.626444	0	0.140	1	0
6	0.290	0.195	0.038	0.034	-0.0805	0.626444	0	0.223	1	0
4	0.294	0.335	0.026	0.010	-0.0805	0.626444	0	0.390	1	0
2	0.297	0.250	0.034	0.062	-0.0805	0.626444	0	0.390	1	0
9	0.298	0.299	0.032	0.142	-0.0805	0.626444	0	0.390	1	0
7	0.304	0.234	0.039	0.012	-0.0805	0.626444	0	0.390	1	0
6	0.309	0.485	0.037	0.062	-0.0805	0.626444	0	0.390	1	0
12	0.313	0.424	0.064	0.086	-0.0805	0.626444	0	0.390	1	0
4	0.322	0.378	0.029	0.021	-0.0805	0.626444	0	0.390	1	0
4	0.324	0.260	0.025	0.000	-0.0805	0.626444	0	0.390	1	0
45	0.362	0.106	0.046	0.010	-0.0805	0.626444	0	0.381	1	0

### Sampel Hasil Ekstraksi Data Pengguna 5

<i>durasi</i>	<i>waktu</i>	<i>CPUusage</i>	<i>down</i>	<i>up</i>	<i>latitude</i>	<i>longitude</i>	<i>Down status</i>	<i>battery</i>	<i>layar</i>	<i>charge</i>
71	0.001	0.353	0.040	0.140	-0.0805	0.626444	0	0.633	0	0
4	0.050	0.278	0.010	0.000	-0.0805	0.626444	0	0.680	1	0
6	0.053	0.432	0.010	0.099	-0.0805	0.626444	0	0.680	1	0
4	0.057	0.283	0.006	0.016	-0.0805	0.626444	0	0.680	1	0
2	0.069	0.215	0.021	0.057	-0.0805	0.626444	0	0.680	1	0
13	0.071	0.478	0.011	0.034	-0.0805	0.626444	0	0.680	1	0
4	0.080	0.288	0.006	0.013	-0.0805	0.626444	0	0.680	1	0
5	0.083	0.116	0.038	0.013	-0.0805	0.626444	0	0.650	1	0
12	0.086	0.286	0.050	0.044	-0.0805	0.626444	0	0.800	1	1
13	0.094	0.327	0.174	0.388	-0.0805	0.626444	0	0.800	1	1
2	0.103	0.205	0.009	0.006	-0.0805	0.626444	0	0.800	1	1
3	0.109	0.217	0.007	0.000	-0.0805	0.626444	0	0.800	1	1
6	0.111	0.172	0.015	0.019	-0.0805	0.626444	0	0.800	1	1
77	0.115	0.147	0.002	0.001	-0.0805	0.626444	0	0.498	0	0
58	0.169	0.109	0.002	0.004	-0.0805	0.626444	0	0.580	0	0
17	0.224	0.113	0.008	0.004	-0.0805	0.626444	0	0.190	1	0
2	0.236	0.350	0.013	0.013	-0.0805	0.626444	0	0.190	1	0
6	0.238	0.420	0.015	0.055	-0.0805	0.626444	0	0.190	0	0
65	0.242	0.099	0.004	0.010	-0.0805	0.626444	0	0.760	0	0
3	0.287	0.437	0.005	0.008	-0.0805	0.626444	0	0.900	1	0
43	0.302	0.077	0.003	0.006	-0.0805	0.626444	0	0.750	1	0
2	0.332	0.205	0.004	0.006	-0.0805	0.626444	0	0.750	1	0
9	0.351	0.272	0.006	0.017	-0.0805	0.626444	0	0.480	1	1
2	0.363	0.270	0.016	0.044	-0.0805	0.626444	0	0.480	1	1
4	0.364	0.628	0.016	0.089	-0.0805	0.626444	0	0.480	1	1
6	0.367	0.153	0.014	0.032	-0.0805	0.626444	0	0.480	1	1
5	0.371	0.216	0.009	0.015	-0.0805	0.626444	0	0.480	1	1
4	0.460	0.335	0.005	0.003	-0.0805	0.626444	0	0.390	1	0
2	0.463	0.250	0.007	0.019	-0.0805	0.626444	0	0.390	1	0
9	0.465	0.299	0.007	0.044	-0.0805	0.626444	0	0.390	1	0

### Lampiran 3. Hasil Proses Learning

#### Centroid data pengguna 1

<b>durasi</b>	<b>waktu</b>	<b>CPUusage</b>	<b>download</b>	<b>upload</b>	<b>latitude</b>	<b>longitude</b>
11	0.733	0.235	0.619	0.127	-0.0805	0.626444
12	0.776	0.312	0.475	0.115	-0.0805	0.626445
1	0.327	0.168	0.252	0.048	-0.0805	0.626444
4	0.412	0.216	0.194	0.037	-0.0805	0.626444
9	0.704	0.244	0.146	0.024	-0.0805	0.626445

#### Center cluster data pengguna 1

<b>durasi</b>	<b>waktu</b>	<b>CPUusage</b>	<b>download</b>	<b>upload</b>	<b>latitude</b>	<b>longitude</b>
215	0.200	0.073	0.145	0.020	-0.0805	0.626444
6	0.572	0.259	0.293	0.052	-0.0805	0.626444
48	0.446	0.103	0.168	0.036	-0.0805	0.626444
101	0.450	0.124	0.274	0.082	-0.0805	0.626444
18	0.594	0.201	0.257	0.062	-0.0805	0.626444

**Centroid data pengguna 2**

<b>durasi</b>	<b>waktu</b>	<b>CPUusage</b>	<b>download</b>	<b>upload</b>	<b>latitude</b>	<b>longitude</b>
6	0.570	0.226	0.046	0.058	-0.08051	0.626444
3	0.635	0.274	0.013	0.011	-0.08052	0.626446
9	0.186	0.123	0.028	0.026	-0.08051	0.626446
4	0.619	0.304	0.014	0.045	-0.08051	0.626446
17	0.800	0.180	0.054	0.066	-0.08051	0.626448

**Center cluster data pengguna 2**

<b>durasi</b>	<b>waktu</b>	<b>CPUusage</b>	<b>download</b>	<b>upload</b>	<b>latitude</b>	<b>longitude</b>
50	0.374	0.119	0.030	0.037	-0.08051	0.626445
20	0.577	0.189	0.017	0.023	-0.08051	0.626446
174	0.474	0.105	0.021	0.013	-0.08051	0.626444
5	0.616	0.265	0.046	0.056	-0.08051	0.626446
95	0.383	0.107	0.022	0.016	-0.08051	0.626444

**Centroid data pengguna 3**

<b>durasi</b>	<b>waktu</b>	<b>CPUusage</b>	<b>download</b>	<b>upload</b>	<b>latitude</b>	<b>longitude</b>
2	0.512	0.385	0.493	0.264	-0.0805	0.626444
11	0.615	0.277	0.234	0.119	-0.0805	0.626444
7	0.702	0.179	0.150	0.038	-0.0805	0.626444
13	0.360	0.177	0.169	0.076	-0.0805	0.626444
8	0.533	0.177	0.147	0.030	-0.0805	0.626444

**Center cluster data pengguna 3**

<b>durasi</b>	<b>waktu</b>	<b>Cpuusage</b>	<b>download</b>	<b>upload</b>	<b>latitude</b>	<b>longitude</b>
45	0.437	0.143	0.247	0.136	-0.0805	0.626445
71	0.636	0.147	0.238	0.111	-0.0805	0.626444
133	0.450	0.087	0.145	0.049	-0.0805	0.626444
6	0.539	0.266	0.236	0.098	-0.0805	0.626445
16	0.573	0.228	0.247	0.120	-0.0805	0.626445

**Centroid data pengguna 4**

<b>durasi</b>	<b>waktu</b>	<b>CPUusage</b>	<b>download</b>	<b>upload</b>	<b>latitude</b>	<b>longitude</b>
14	0.597	0.256	0.040	0.038	-0.08051	0.626444
11	0.793	0.305	0.027	0.030	-0.0805	0.626444
1	0.388	0.445	0.595	0.102	-0.0805	0.626444
7	0.312	0.190	0.034	0.035	-0.0805	0.626444
1	0.482	0.394	0.018	0.003	-0.0805	0.626444

**Center cluster data pengguna 4**

<b>durasi</b>	<b>waktu</b>	<b>Cpuusage</b>	<b>download</b>	<b>upload</b>	<b>latitude</b>	<b>longitude</b>
178	0.337	0.124	0.040	0.061	-0.0805	0.626444
11	0.570	0.302	0.058	0.084	-0.08051	0.626443
70	0.463	0.166	0.053	0.083	-0.0805	0.626444
4	0.543	0.298	0.063	0.125	-0.0805	0.626445
34	0.605	0.248	0.134	0.203	-0.0805	0.626444

**Centroid data pengguna 5**

<b>durasi</b>	<b>waktu</b>	<b>CPUusage</b>	<b>download</b>	<b>upload</b>	<b>latitude</b>	<b>longitude</b>
2	0.200	0.212	0.074	0.171	-0.0805	0.626447
1	0.626	0.383	0.089	0.179	-0.0805	0.626445
2	0.830	0.446	0.054	0.045	-0.0805	0.626444
18	0.744	0.232	0.012	0.022	-0.0805	0.626445
4	0.500	0.247	0.008	0.018	-0.0805	0.626444

**Center cluster data pengguna 5**

<b>durasi</b>	<b>waktu</b>	<b>Cpuusage</b>	<b>download</b>	<b>upload</b>	<b>latitude</b>	<b>longitude</b>
5	0.606	0.306	0.026	0.056	-0.0805	0.626443
29	0.633	0.290	0.049	0.044	-0.08051	0.626451
12	0.612	0.390	0.083	0.214	-0.0805	0.626446
66	0.380	0.199	0.015	0.042	-0.0805	0.626446
124	0.423	0.220	0.018	0.050	-0.08051	0.626449

#### Lampiran 4. Hasil Prediksi Durasi *Idle*

##### Hasil Prediksi *Idle* Pengguna 1

Real	Prediksi		
	KNN	KMeans	FCM
16	16	1	18
9	9	1	5
142	87	1	107
10	7	1	6
91	95	4	100
3	2	4	5
9	9	1	5
11	10	12	8
10	4	11	6
8	8	1	5
8	5	1	5
6	10	1	5
14	9	9	17
9	17	1	5
4	4	4	5
8	7	9	5
4	4	9	5
3	7	9	5
117	2	11	100
10	10	11	6
7	7	12	5
15	15	11	18
5	5	12	5
8	16	9	5
6	4	12	5
2	2	12	5
22	7	9	18
3	7	9	5
16	16	1	18



### Hasil Prediksi *Idle* Pengguna 2

Real	Prediksi		
	KNN	KMeans	FCM
2	2	9	5
6	4	9	5
2	3	6	5
11	11	3	7
3	3	3	5
4	3	4	5
6	6	3	5
19	19	3	20
70	4	6	62
3	2	3	5
7	8	17	5
8	3	3	5
7	3	3	5
2	2	9	5
15	46	9	19
2	2	9	5
3	3	3	5
14	17	9	17
43	43	3	49
17	3	6	20
60	11	6	50
4	2	4	5
53	2	3	50
2	2	17	5
4	3	4	5
19	2	9	20
55	2	9	50
43	11	9	49
23	3	4	20

### Hasil Prediksi *Idle* Pengguna 3

Real	Prediksi		
	KNN	KMeans	FCM
113	113	13	128
8	9	13	5
8	3	13	5
12	8	2	14
13	113	13	15
8	12	2	5
13	13	8	15
8	2	2	5
8	8	8	5
5	5	2	5
44	44	2	45
6	4	13	5
3	3	7	5
6	9	13	5
12	12	2	14
13	13	8	15
11	3	8	11
7	7	2	5
13	13	8	15
4	4	11	5
12	12	2	14
25	24	11	16
7	7	13	5
130	64	13	132
7	7	13	5
19	19	13	15
12	12	2	14
8	8	13	5
7	7	11	5

#### Hasil Prediksi *Idle* Pengguna 4

Real	Prediksi		
	KNN	KMeans	FCM
10	6	1	11
5	5	1	4
35	35	1	33
12	12	7	11
6	6	7	4
34	34	11	33
7	9	11	5
4	4	7	4
4	28	7	4
4	2	7	4
86	7	11	69
53	76	7	55
54	57	7	59
5	5	1	4
8	8	11	8
3	3	7	4
7	4	1	5
7	7	7	5
57	8	7	66
2	2	14	4
8	18	11	8
4	2	7	4
11	4	11	11
2	3	14	4
3	5	11	4
2	2	14	4
17	9	11	11
6	9	7	4
7	14	7	5

### Hasil Prediksi *Idle* Pengguna 5

Real	Prediksi		
	KNN	KMeans	FCM
14	58	2	12
6	6	1	4
22	22	2	25
19	58	2	14
26	6	4	28
7	7	2	4
124	2	4	123
3	3	1	4
29	10	1	29
2	9	18	4
5	5	18	4
4	4	1	4
4	2	18	4
4	4	2	4
29	7	18	29
4	4	2	4
6	6	1	4
4	4	2	4
10	10	2	11
11	6	2	12
14	14	18	12
2	2	18	4
13	13	2	12
10	6	2	11
36	5	4	28
4	3	4	4
6	6	1	4
88	5	18	71
2	4	18	4

**Lampiran 5. Data Uji Coba Kesiapan Konektivitas Internet**

**Data aktivitas dan *download* pengguna 1**

Menit ke	Aktivitas ( <i>idle</i> / aktif)			Data <i>Download</i> (bytes)		
	Aktivitas pengguna	Metode A	Metode B	<i>download</i>	<i>Download</i> Metode A	<i>Download</i> Metode B
1	0	0	0	1	0	0
2	0	0	0	8	0	0
3	0	0	0	15	0	0
4	0	1	0	2	26	0
5	0	1	0	1	1	0
6	0	1	0	23	23	0
7	0	0	0	3	0	0
8	0	0	0	3	0	0
9	0	0	0	14	0	0
10	0	1	0	11	31	0
11	0	1	0	6	6	0
12	0	1	0	13	13	0
13	0	0	0	2	0	0
14	1	0	1	6	0	108
15	0	0	0	15	0	0
16	0	1	0	1	24	0
17	0	1	0	30	30	0
18	0	1	0	43	43	0
19	0	1	0	3	3	0
20	0	1	0	1	1	0
21	1	1	1	42	42	135
22	0	1	0	14	14	0
23	0	0	0	1	0	0
24	0	0	0	13	0	0
25	0	0	0	2	0	0
26	0	0	0	14	0	0
27	0	1	0	14	44	0
28	0	1	0	1	1	0
29	0	1	0	36	36	0
30	0	1	0	26	26	0
31	0	0	0	2	0	0
32	0	0	0	1	0	0
33	0	0	0	13	0	0
34	0	0	0	14	0	0
35	0	1	0	3	33	0

36	0	1	0	14	14	0
37	1	1	1	35	35	203
38	0	1	0	14	14	0
39	0	0	0	13	0	0
40	0	0	0	5	0	0
41	0	1	0	1	19	0
42	0	1	0	27	27	0
43	0	1	0	1	1	0
44	0	1	0	12	12	0
45	1	1	1	16	16	89
46	0	1	0	15	15	0
47	1	1	1	13	13	28
48	0	1	0	3	3	0
49	0	1	0	1	1	0
50	0	1	0	25	25	0

## DAFTAR PUSTAKA

- Feist, J. 2014. HTC One (M8) has Fastest Display Touch Response *Time*. <http://www.androidauthority.com/display-touch-response-365800/>. 17 April 2014 (09:49).
- Kim, M. W., D. G. Yun, J. M. Lee, dan S. G. Choi. 2012. Battery Life *Time* Extension Method Using Selective Data Reception on *Smartphone*. *International Conference of Information Networking*: 468–471.
- Hyeon, Y., M. W. Kim, J. M. Lee, dan S. G. Choi. 2012. Battery Life *Time* Extension Method By Using Signalling Interval Control. *International Conference Advanced Communication Technology*: 327–330.
- Khairy, A., H. H. Ammar, dan R. Bahgat. 2013. *Smartphone* Energizer: Extending *Smartphone's* Battery Life with Smart *Offloading*. *International Wireless Communications and Mobile Computing Conference*: 329–336.
- Liggett, B. 2011. E-Mili: New *Idle* Technology Could Make *Smartphone* 44% More Energy Efficient. <http://inhabitat.com/new-idle-technology-could-make-s-44-more-energy-efficient/>. 17 April 2014 (14:35).
- Meier, R. 2012. *Professional Android™ 4 Application Development*. 1st ed. John Wiley & Sons, Inc. Indianapolis.
- Lee, S. C., E. Lee, W. Choi, dan U. M. Kim. 2008. Extracting Temporal Behavior Patterns of Mobile User. *Networked Computing and Advanced Information Management 2*: 455–462.
- Kang, J. M., S. S. Seo, dan J. W.-K. Hong. 2011. Usage Pattern Analysis of *Smartphones*. *Network Operations and Management Symposium*: 1–8.
- Zhang, L., B. Tiwana, R. P. Dick, Z. Qian, Z. M. Mao, Z. Wang, dan L. Yang. 2010. Accurate Online Power Estimation and Automatic Battery Behavior Based Power Model Generation for *Smartphones*. *Hardware/Software Codesign and System Synthesis*: 105–114.
- Zahid, I. M. A. Ali, dan R. Nassr. 2011. Android *Smartphone*: Battery Saving Service. *Research and Innovation in Information Systems*: 1–4.

## BIODATA PENULIS

**Nama** : Fadilah Fahrul Hardiansyah  
**Tempat Lahir** : Jember  
**Tanggal Lahir** : 29 Januari 1989  
**Agama** : Islam  
**Alamat** : Jl. Sumatera IX-38  
RT 01, RW 01  
Sumpersari, Jember  
**Email** : fahrul@pens.ac.id  
**Phone** : +6283853501389



Penulis lahir pada awal tahun 1989 di kota Jember. Penulis menghabiskan masa kanak-kanak hingga remaja di kota Jember. Penulis lulus dari SDN Kepatihan 16 pada tahun 2001 dan melanjutkan sekolah di SMPN 4 Jember. Setelah lulus SMP pada tahun 2004, penulis melanjutkan sekolah di SMAN 1 Jember yang merupakan SMA kebanggaan kota Jember. Setelah lulus dari SMA penulis melanjutkan studi di Politeknik Elektronika Negeri Surabaya jurusan Teknik Informatika dalam jenjang Diploma III. Setelah lulus dari jenjang Diploma III penulis melanjutkan studi dalam jenjang Diploma IV di almamaternya. Pada tahun 2012 penulis melanjutkan studi S2 di Institut Teknologi Sepuluh Nopember jurusan Teknik Informatika, dan mendapatkan gelar Magister pada tahun 2015. Walaupun tidak ada bekal ilmu di bidang IT dari SMA, penulis dapat mengikuti dan menyerap banyak ilmu dari perkuliahan. Hingga penulis dapat menjadi dosen di almamaternya, Politeknik Elektronika Negeri Surabaya jurusan Teknik Informatika.

*Imagination is more important than knowledge*