53209/4/13
MILIK PERPUSTAKAAN ITS

APLIKASI MOBILE STRATEGIC DASHBOARD BERBASIS ANDROID UNTUK MONITORING PERSEBARAN PENYAKIT SAPI (STUDI KASUS : DINAS PETERNAKAN DAN KESEHATAN KABUPATEN MALANG)

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
Pada
Jurusan Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya
Oleh :

RSSI
005.43
Rez
a-1
2013

FAHRI REZA
NRP 5209 100 093

Surabaya, Oktober 2013

KETUA
JURUSAN SISTEM INFORMASI

Dr. Eng. FEBRILIYAN SAMOPA, S.Kom, M.Kom
NIP 19730219.199802.1.001

PERPUSTAKAAN ITS
Tgl. Terima 20-12-2013
Terima H
No Agenda Prp -

Selesai Revisi
28/10'13
R.P.K.

APLIKASI MOBILE STRATEGIC DASHBOARD BERBASIS ANDROID UNTUK MONITORING PERSEBARAN PENYAKIT SAPI (STUDI KASUS : DINAS PETERNAKAN DAN KESEHATAN KABUPATEN MALANG)

TUGAS AKHIR

Diajukan untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Business Intelligence/Decision Support System
Program Studi S-1 Jurusan Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh :
FAHRI REZA
NRP. 5209 100 093

Disetujui Tim Penguji: Tanggal Ujian : September 2013
Periode Wisuda : Desember 2013

Wiwik Anggraeni, S.Si., M.Kom **(Pembimbing I)**

Hatma Suryotrisongko, S.Kom., M.Eng **(Pembimbing II)**

Erma Suryani S.T, M.T, Ph.D **(Penguji 1)**

Raras Tyasnurita S.Kom., MBA **(Penguji 2)**

APLIKASI MOBILE STRATEGIC DASHBOARD BERBASIS ANDROID UNTUK MONITORING PERSEBARAN PENYAKIT SAPI (STUDI KASUS : DINAS PETERNAKAN DAN KESEHATAN KABUPATEN MALANG)

Nama Mahasiswa	: Fahri Reza
NRP	: 5209 100 093
Jurusan	: Sistem Informasi FTIF-ITS
Dosen Pembimbing I	: Wiwik Anggraeni, S.Si., M.Kom
Dosen Pembimbing II	: Hatma Suryotrisongko, S.Kom., M.Eng

ABSTRAK

Monitoring terhadap penyakit sapi dan persebarannya penting dilakukan oleh Dinas Peternakan. Sebelumnya, proses monitoring dilakukan dengan cara mendeteksi satu-persatu dilapangan. Cara seperti ini membutuhkan waktu yang relatif lama untuk menentukan hasil secara keseluruhan. Proses yang demikian panjang akan menjadi lebih praktis dan efisien jika dibantu dengan pengembangan teknologi yang dapat mengirimkan data secara cepat dan efisien. Untuk itu pengembangan aplikasi mobile dashboard yang berguna dalam monitoring persebaran penyakit sapi akan sangat membantu proses tersebut.

Melalui aplikasi mobile dashboard proses intepretasi data akan jauh lebih mudah. Data akan diproses, di rangkum dan tampil dalam bentuk visualisasi. Sehingga user akan lebih mudah mengetahui dan membaca hasil secara keseluruhan. Visualisasi data tersebut dapat diwujudkan dalam bentuk mobile dashboard berbasis android, yang sebelumnya diintegrasikan dengan aplikasi dalam device peternak.

Dengan adanya aplikasi ini Dinas Peternakan dapat lebih mudah memonitoring penyakit sapi dan melihat persebarannya di suatu wilayah. Pengambilan keputusan menjadi lebih mudah karena didasarkan pada KPI. Nantinya penggunaaan mobile dashboard ini juga dapat digunakan sebagai acuan untuk mengambil tindakan terhadap suatu daerah yang memiliki persebaran penyakit sapi tertinggi agar lebih lebih tepat sasaran.

Kata Kunci: *Penyakit sapi, Android, Mobile dashboard,visualisasi.*

STRATEGIC MOBILE APPLICATION DASHBOARD BASED ON ANDROID FOR MONITORING THE SPREAD OF COW DISEASE (CASE STUDY : DEPARTMENT OF ANIMAL HUSBANDRY AND HEALTH DISTRICT MALANG)

Student's Name	: Fahri Reza
NRP	: 5209 100 093
Major of Department	: Sistem Informasi FTIF-ITS
Supervisor	: Wiwik Anggraeni, S.Si., M.Kom
Co-Supervisor	: Hatma Suryotrisongko, S.Kom., M.Eng

ABSTRACT

Monitoring against the spread of cow disease done by Department of Animal Husbandry. Previously is done by detecting one by one in field. This method required a relatively long time to determine the overall outcome. Such a long process will be more practical and efficient if assisted with the development of technology that can transmit data quickly. For the development of mobile application that are useful in monitoring the spread of cow disease will greatly assist the process.

Through mobile dashboard application, data interpretation process will be much easier. Data will be processed, summarized and appear in the form of visualization. So the user can be realized in the form mobile dashboard based on android which earlier were intergration with the device application of cattleman.

With this application , Department of Animal Husbandry could more easily to monitor and view the cow disease within its range in area. Decision making become easier its base KPI. Later the use of mobile dasbhboard can

also be used as reference to take action an area that has the highest cow disease in order to be more precisely target.

Keywords: *. Cow Disease, Android, Mobile Dashboard, Visualization.*

KATA PENGANTAR

Segala puji dan syukur ditujukan kepada Allah yang telah melimpahkan kasih sayang dan karunia-Nya sehingga penulis dapat menyelesaikan Tugas Akhir berjudul:

"APLIKASI MOBILE STRATEGIC DASHBOARD BERBASIS ANDROID UNTUK MONITORING PERSEBARAN PENYAKIT SAPI (STUDI KASUS : DINAS PETERNAKAN DAN KESEHATAN KABUPATEN MALANG)"

Dengan ini, penulis hendak menyampaikan penghormatan dan terima kasih kepada pihak-pihak yang telah memberikan bantuan dan dukungan secara langsung maupun tidak langsung antara lain kepada :

1. Bapak, Ibu, dan Adikku tersayang dan seluruh keluarga besar yang selalu memberikan doa, semangat, dan motivasi selama pengerjaan tugas akhir ini.
2. Bapak Dr. Eng Febriliyan Samopa, S.Kom, M.Kom. selaku Ketua Jurusan Sistem Informasi ITS. Sekaligus sebagai dosen wali selama kuliah.
3. Ibu Wiwik Anggraeni dan Bapak Hatma Suryotrisongko, terima kasih atas bimbingan intensif yang telah diberikan, juga atas kesempatannya menjadi anak bimbing ibu bapak. Terima kasih kepada Bapak Hatma atas bimbingan intensif yang diberikan serta ilmu android yang diberikan dan bu wiwik yang senantiasa mendampingi penulis.
4. Ibu Erma Suryani dan Bu Raras Tyasnurita selaku dosen penguji yang telah bersedia menguji dan memberikan masukan yang sangat berarti bagi penulis.

5. Bapak dan Ibu dosen Jurusan Sistem Informasi-ITS yang telah dengan sabar memberi ilmunya selama penulis menempuh kuliah.
6. Seluruh staf karyawan Jurusan Sistem Informasi dan karyawan Fakultas Teknologi Informasi atas dukungannya sehingga tugas akhir ini dapat terselesaikan.
7. Kinara Ayu Hati Imananda Putri yang senantiasa mendampingi penulis dalam menyeleseaikan tugas akhir. Terimakasih atas kebersamaan dan semangatnya.
8. M. Adji Firmansyah, Irwan, Afas,Agung, Aldhi, yang telah berjuang bersama dalam menyelesaikan tugas akhir.
9. Bapak dan Ibu Dinas Peternakan dan Kesehatan hewan Kabupatan Malang yang tidak bisa penulis sebutkan satu persatu
10. Teman-teman admin lab SPK dan pengguna lab SPK yang selalu memberi semangat dan telah berjuang bersama untuk menyelesaikan tugas akhir.
11. Teman-teman Ae9is yang selalu memberikan semangat, dan rasa kekeluargaan yang terjalin selama ini.
12. Berbagai pihak yang belum sempat penulis sebutkan jasa-jasanya dalam mendukung penyusunan tugas akhir ini.

Apabila pada Tugas Akhir ini terdapat kekurangan, dengan kerendahan hati, penulis mengharapkan saran dan kritik yang membangun dari pembaca.

Surabaya, September 2013

Fahri Reza

DAFTAR ISI

HALAMAN JUDUL i
ABSTRAK vii
KATA PENGANTAR xi
Daftar Isi xiii
Daftar Gambar xvii
DAFTAR TABEL xix
BAB 1 PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Perumusan Masalah 4
1.3 Tujuan Tugas Akhir 4
1.4 Batasan Masalah 5
1.5 Manfaat 5
1.6 Sistematika Penulisan 6
BAB 2 DASAR TEORI 9
2.1 Penyakit Sapi 9
2.1.1 Penyakit Dismatosis 9
2.1.2 Penyakit Mastitis 10
2.1.3 Penyakit *Bovine Ephemeral Fever* (BEF) 10
2.2 *Strategic Dashboard* 11
2.3 KPI (*Key Performance Indicator*) 14
2.4 Chart 15
2.5 Aplikasi Mobile 16
2.6 Android 17
2.6.1 Android SDK (*Software Development Kit*) 17
2.6.2 ADT (*Android Development Tools*) 18
2.6.3 The Dalvik Virtual Machine 19
2.6.4 Aplication Framework 20
2.6.5 Libraries 21
2.6.6 Android Run Time 21
2.6.7 Linux Kernel 22
2.7 Web Service 22

2.7.1 Kelebihan Web Service 24
2.8 Java Scrips Object Notation (JSON) 25
2.9 AcharEngine 25
2.10 Hipotesa Sistem 29
BAB 3 METODE PENELITIAN 33
BAB 4 ANALISA DAN IMPLEMENTASI SISTEM 37
4.1 Analisa Tujuan Bisnis Dinas Peternakan 37
4.2 Identifikasi Jenis Dashboard dan Kelompok Pengguna 38
4.3 Kebutuhan Lingkungan 39
4.4 Identifikasi *Key Performance Indicator* 39
4.5 Perancangan *Dashboard* 43
4.5.1 Perencanaan Basis Data 43
4.5.2 Perencanaan Fungsional Sistem 47
4.5.2.1 Use Case 47
4.5.2.2 Robusness Diagram 50
4.6 Pembangunan Sistem 53
4.6.1 Gambaran Umum Aplikasi 53
4.6.2 Direktori dan File Kunci dalam Salinan Kerja Pengembang 54
4.6.3 Pembuatan fitur bagi petugas kecamatan 56
4.6.3.1 Pembuatan Fitur Input Data di Android 56
4.6.3.2 Pembuatan Web Service dengan JSON untuk Menginputkan Data 59
4.6.4 Pembuatan Aplikasi *Strategic Mobile Dashboard* Untuk Dinas Peternakan 68
4.6.4.1 Implementasi Library Achartengine pada Android Project 69
4.6.4.2 Pembuatan *Web Service* dengan JSON untuk Menampilkan Data 70
4.6.4.3 Pembuatan Fitur Melihat Grafik Berdasarkan Penyakit / Kecamatan 79
4.6.4.4 Pembuatan Fitur Melihat Grafik Berdasarkan Penyakit / Desa 88
4.6.4.5 Pembuatan Fitur Melihat Grafik Berdasarkan Penanganan / Kecamatan 97

4.6.4.6 Pembuatan Fitur Melihat Grafik Berdasarkan Penanganan / Desa 106
BAB 5 Uji coba dan analisis hasil 117
5.1 Validasi dan Verifikasi Sistem 117
5.1.1 Uji Coba Aplikasi dengan *Test Case* 117
5.1.1.1 Uji Coba Test Case Sisapi 117
5.1.1.2 Uji Coba *Test Case Strategic Mobile Dashboard* 120
5.1.2 Verifikasi Aplikasi dengan Berbagai Jenis Smartphone Android 128
5.1.3 Validasi dengan Sistem Pakar 132
5.2 Analisis Hasil Sistem 132
5.2.1 Analisis Persebaran Penyakit Sapi Berdasarkan Jenis Penyakit 133
5.2.2 Analisis Penanganan Penyakit Sapi Berdasarkan Jenis Obat / Kecamatan 170
5.2.3 Analisis Penanganan Penyakit Sapi Berdasarkan Jenis Obat / Desa 207
5.2.4 Analisis Persebaran Penyakit Sapi Berdasarkan Jenis Penyakit / Desa 209
BAB 6 KESIMPULAN DAN SARAN 213
6.1 Kesimpulan 213
6.2 Saran 214
DAFTAR PUSTAKA 215
LAMPIRAN A 219
LAMPIRAN B 223
BIODATA PENULIS 217

DAFTAR GAMBAR

Gambar 2. 1 Tabel call volume data (Few, 2007) 12
Gambar 2. 2 Grafik call volume data (Few, 2007)..... 12
Gambar 2. 3 Proses kerja DVM (Claudio Maia, 2008) ... 19
Gambar 2. 4 Arsitektur Android 22
Gambar 2. 5 Arsitektur WebService 23
Gambar 2. 6 Layer Komponen web service..... 24
Gambar 2. 7 Persiapan data 26
Gambar 2. 8 Inisiasisi dan konfigurasi grafik 27
Gambar 2. 9Kustomisasi warna 28
Gambar 2. 10 Menampilkan grafik 28
Gambar 2. 11 AndroidManifest.xml 29
Gambar 2. 12 Contoh grafik 29
Gambar 3. 1 Alur Pengerjaan Tugas Akhir.....36
Gambar 4. 1 Conceptual data Model Untuk Mobile Strategi Dashboard.....46
Gambar 4. 2 Physical Data Model Mobile Strategi Dashboard 47
Gambar 4. 3 robusness diagram log in 50
Gambar 4. 4 robusness diagram log out 51
Gambar 4. 5 Robusness Diagram tambah data 51
Gambar 4. 6 Robusness Diagram penyakit /kecamatan ... 52
Gambar 4. 7 Robusness Diagram penyakit / desa 52
Gambar 4. 8 robusness diagram penanganan / kecamatan 53
Gambar 4. 9 robusness diagram penanganan / desa 53
Gambar 4. 10 Desain Antar muka untuk fitur input data.. 68
Gambar 4. 11 Desain Antarmuka berdasarkan penyakit /kecamatan 81
Gambar 4. 12 Tampilan dashboard berdasarkan penyakit / kecamatan 88
Gambar 4. 13 Desain Antarmuka berdasarkan penyakit /desa..... 90

Gambar 4. 14 Tampilan dashboard berdasarkan penyakit / desa 97
Gambar 4. 15 Desain Antarmuka berdasarkan penyakit /kecamatan 99
Gambar 4. 16 Tampilan dashboard berdasarkan penanganan / kecamatan 106
Gambar 4. 17 Tampilan dashboard berdasarkan penanganan / desa 115
Gambar 5. 1 Screenshoot Aplikasi di Smartphone 1.....129
Gambar 5. 2 Screenshoot Aplikasi di Smartphone 2..... 130
Gambar 5. 3 Screenshoot Aplikasi di Smartphone 3..... 132
Gambar 6. 1 Form validasi dari Dinas Peternakan 1.....228
Gambar 6. 2 Form validasi dari dinas Peternakan 2..... 229
Gambar 6. 3 Form Validasi Dinas Peternakan 3..... 230
Gambar 6. 4 Berita acara Dinas Peternakan 231

DAFTAR TABEL

Tabel 2. 1 Jenis Chart 31
Tabel 4. 1 Tabel 4. 1 Key Perfomance Indicator Persebaran Penyakit Sapi 41
Tabel 4. 2 Daftar Atribut untuk Tabel Entitas "fact_peternakan" 43
Tabel 4. 3 Daftar atribut untuk Tabel Entitas "Username"44
Tabel 4. 4 Daftar atribut untuk Tabel Entitas "Kota" 44
Tabel 4. 5 Daftar atribut untuk Tabel Entitas "Kecamatan" 45
Tabel 4. 6 Daftar atribut untuk Tabel Entitas "Desa" 45
Tabel 4. 7 Daftar atribut untuk Tabel Entitas "dusun" 45
Tabel 4. 8 Daftar Use case Peternak 48
Tabel 4. 9 Daftar Use case Dinas Peternakan 49
Tabel 4. 10 Tabel Direktori Aplikasi sisi server 54
Tabel 4. 11 Tabel direktori Aplikasi sisi client(petugas Kecamatan) 54
Tabel 4. 12 Tabel direktori Aplikasi pada sisi client (dinas Peternakan) 55
Tabel 5. 1 Test Case Sisapi,,,,,,,,,,,,,,,,,,,,,........,,,,,,,,,,,,,,,,,,,,,,118
Tabel 5. 2 Test case Strategic Mobile Dashboard 121
Tabel 5. 3 Spesifikasi Smartphone 1 128
Tabel 5. 4 Spesifikasi Smarthphone 2 129
Tabel 5. 5 Spesifikasi Smarthphone 3 131
Tabel 5. 6 Table Analisis penyakit / kecamatan 134

BAB 1
PENDAHULUAN

Pada bab ini dijelaskan beberapa hal dasar yang meliputi latar belakang, tujuan, permasalahan, batasan masalah, metodologi serta sistematika penulisan tugas akhir, sehingga gambaran umum permasalahan dan pemecahan yang diambil dapat dipahami dengan baik

1.1 Latar Belakang

Pendeteksian persebaran dan jenis penyakit pada ternak seperti sapi merupakan hal yang sangat penting untuk dilakukan. Mengingat sapi merupakan salah satu komoditas dengan tingkat pengkonsumsian yang tinggi di Indonesia. Tindakan yang cepat dan tepat dapat mengurangi kemungkinan dan resiko terburuk yang dapat terjadi. Pendeteksian persebaran penyakit ini harusnya dilakukan oleh pihak-pihak terkait seperti peternak dan Dinas Peternakan.

Beberapa penelitian sebelumnya telah mengembangkan aplikasi yang diharapkan dapat memudahkan pendeteksian penyakit sapi. Penelitian yang dilakukan oleh Listiana (2011) menghasilkan sebuah aplikasi berbasis desktop. Dalam penelitiannya Listiana mengharapkan dengan dibuatnya aplikasi berbasis desktop tersebut, peternak tidak lagi tergantung pada dokter hewan. Namun, dalam pengaplikasiaanya Listiana mengungkapkan bahwa aplikasi berbasis desktop yang telah dibuat kurang memudahkan pengguna dalam mengoprasikannya. Pengguna/user diharuskan melakukan instalasi database dan aplikasi yang digunakan sebelum aplikasi tersebut dioperasikan (Listiana, 2011).

Berangkat dari permasalahan tersebut, sebuah penelitian baru dilakukan oleh Ardianto pada tahun 2012 dengan tujuan mempermudah pengguna dalam pengoprasian aplikasi

pendeteksian penyakit sapi. Ardianto (2012) mengembangkan aplikasi serupa namun berbasis mobile pada sistem operasi android. Aplikasi mobile berbasis android dipilih Ardianto dikarenakan teknologi *mobile communication* sudah menyentuh hampir semua kalangan di Indonesia (Pradana, 2007).

Ardianto (2012) mengembangkan aplikasi *mobile* dengan metode *knowledge representation* untuk mendeteksi dan memberikan penanganan dini penyakit sapi berbasis android. Dengan metode tersebut peternak hanya perlu membuka aplikasi mobile dalam *device* mereka dan memasukkan beberapa data. Kemudian peternak telah dapat melakukan pendeteksian dini terhadap penyakit sapi. Akan tetapi aplikasi mobile tersebut hanya dikhususkan untuk digunakan oleh peternak sapi. Di satu sisi Dinas Peternakan juga membutuhkan sebuah inovasi teknologi untuk melakukan monitoring persebaran penyakit sapi terutama di kabupaten Malang.

Sebelumnya monitoring yang dilakukan oleh Dinas Peternakan melalui pengumpulan data-data penyakit dari setiap daerah di kabupaten Malang. Namun metode demikian membutuhkan waktu yang relatif panjang, mulai pengumpulan data dilapangan hingga implementasi tindakan nyata terhadap penyakit sapi. Padahal setiap penyakit sapi yang terjadi di suatu daerah membutuhkan langkah strategis dan cepat dalam hal penanganannya. Teknologi informasi yang cepat dan akurat dibutuhkan oleh Dinas Peternakan untuk memonitoring persebaran penyakit sapi di kabupaten Malang.

Melihat kondisi demikian diperlukan adanya teknologi tepat guna yang dapat secara cepat dan otomatis mengirimkan data persebaran penyakit sapi kepada Dinas Peternakan. Dengan teknologi tersebut Dinas peternakan tidak perlu datang untuk memantau satu persatu penyakit sapi yang tersebar di kabupaten Malang. Dinas Peternakan dapat langsung memonitoring dimana saja dan kapan saja secara menyeluruh melalui data yang dikirimkan oleh peternak melalui aplikasi mobile pada *device*

mereka, yang terkoneksi langsung dengan aplikasi *mobile dashboard* milik Dinas Peternakan.

Maka dari itu dalam Tugas Akhir ini penulis mengusulkan sebuah pembuatan aplikasi *mobile dashboard* yang berguna untuk memonitoring sekaligus memvisualisasikan data persebaran penyakit sapi guna mengambil langkah *strategic* dalam hal pengambilan keputusan. Penyajian data dalam bentuk *dasboard* akan lebih memudahkan user dalam membaca data. Data yang tersaji merupakan rangkuman dari seluruh data, namun dalam bentuk yang lebih menarik, ringkas, sederhana dan lebih mudah dipahami.

Penggunaan aplikasi *mobile dashboard* ini akan membuat arus informasi mengalir semakin cepat. Data bersumber dari data yang diinputkan dalam aplikasi *mobile* android milik peternak yang sebelumya dibuat oleh Ardianto(2012), yang kemudian akan diintegrasikan secara langsung kepada aplikasi *mobile dashboard* Dinas Peternakan. Kemudian data yang telah diolah akan disajikan dalam bentuk visualisasi data misalnya diagram dan grafik. Pemvisualisasian data ini diharapkan dapat mempermudah Dinas Peternakan melakukan intepretasi data yang dikirimkan oleh seluruh peternak disebuah wilayah. Namun, untuk lebih mempermudah dan memperjelas sajian data bagi Dinas Peternakan, perlu ada sedikit modifikasi terhadap program yang dikembangkan oleh Ardianto (2012). Beberapa pengembangan program tersebut antara lain menambahkan KPI ke dalam program sebelumnya. Beberapa KPI ini antara lain jumlah ternak sapi yang terkena penyakit sapi berdasarkan kecamatan perbulan, jumlah ternak sapi yang terkena penyakit sapi berdasarkan desa perbulan, jumlah obat yang diberikan berdasarkan kecamatan perbulan, dan jumlah obat yang diberikan berdasarkan desa perbulan.

Dengan adanya *mobile dashboard* ini KPI yang ada lebih mudah divisualisasikan sehingga Dinas Peternakan dapat lebih mudah dan cepat melihat informasi. Informasi yang didapat dari *mobile dashboard* ini nantinya dibuat sebagai langkah strategis

untuk pengambilan kebijakan. Dengan melihat jumlah ternak yang terkena penyakit sapi di daerah Malang, Dinas Peternakan dapat mengambil formulasi kebijakan yang tepat dengan melihat hasil visualisasi dashboard yang ditampilkan. Pihak Dinas Peternakan pun nantinya dapat memberikan penanganan lebih efektif dan efisien seta tepat sasaran terhadap daerah dengan prosentasi penyakit sapi tertinggi di kabupaten Malang. Maka dengan aplikasi *mobile dashboard* ini memudahkan pihak Dinas Peternakan dalam pengambilan keputusan kedepannya.

1.2 Perumusan Masalah

Berdasarkan latar belakang diatas, maka permasalahan yang diangkat dalam tugas akhir ini antara lain :

1) Bagaimana cara menentukan KPI *mobile dashboard* sesuai dengan kebutuhan Dinas Perternakan dan kesehatan Kab. Malang, Jawa Timur?
2) Bagaimana proses penginputan KPI kedalam aplikasi *mobile* berbasis android yang telah dibuat sebelumnya?
3) Bagaimana memvisualisasikan data berdasarkan KPI dalam bentuk *mobile dashboard*?
4) Bagaimana cara menentukan prioritas penanganan persebaran penyakit sapi di daerah Malang?

1.3 Tujuan Tugas Akhir

Tujuan pembuatan tugas akhir adalah:

1. Mendapatkan indikator-indikator yang penting dalam persebaran penyakit sapi.
2. Mengaplikasikan input KPI kedalam aplikasi *mobile* berbasis android.

3. Mempermudah intepretasi data melalui penyajian visualisasi dalam bentuk *dashboard* secara *mobile*
4. Penentuan prioritas penanganan kasus didasarkan pada hasil visualisasi data yang menunjukkan jumlah tertinggi penyakit sapi dan persebarannya.

1.4 Batasan Masalah

Dari permasalahan yang telah disebutkan di atas, maka batasan-batasan dalam tugas akhir ini adalah sebagai berikut:

1) Data persebaran penyakit sapi yang diinputkan yang nantinya divisualisasikan di wilayah kabupaten Malang sesuai dengan KPI yang ada.
2) Untuk variable jenis penyakit sapi yang divisualisasikan ada 3 yaitu penyakit Mastitis, BEF dan Dismatosis dan variable jenis obat yang divisualisasikan yaitu obat antibiotik, obat anthelmentik dan obat antipiretik.
3) Aplikasi *Strategic* mobile dashboard ini berbasis android dengan menggunakan bahasa pemrogaman Java.

1.5 Manfaat

Manfaat yang diberikan oleh tugas akhir ini adalah dengan penggunaan aplikasi *mobile dashboard* berbasis android ini, memudahkan Dinas Peternakan dan Kesehatan mengetahui dan memonitoring penyakit sapi dan persebarannya di kabupaten Malang dengan mudah. Kelebihan yang ditawarkan aplikasi ini juga membuat Dinas Peternakan dan Kesehatan lebih cepat menentukan daerah mana yang harus mendapatkan penanganan dengan segera, dan formulasi kebijakan yang tepat sasaran.

1.6 Sistematika Penulisan

Uraian sistematika penulisan pada tugas akhir ini bertujuan agar perancangan dan pembuatan perangkat lunak yang dibahas menjadi mudah dipahami, jelas dan sistematis untuk tiap-tiap atau subbab bahasan.

Bab 1 Pendahuluan

Berisi gambaran umum yang membahas latar belakang, permasalahan yang dihadapi, batasan-batasan dan juga tujuan yang akan dicapai pada akhir pengerjaan Tugas Akhir ini.

Bab 2 Dasar Teori

Berisi teori-teori yang mendasari Tugas Akhir ini meliputi jenis penyakit sapi, *strategic dashboard*, KPI (*Key performance Indikator*), *Chart*, Android, dan *AchartEngine*.

Bab 3 Metodologi Penelitian

Mendeskripsi sistem secara umum, yang terdiri dari tahapan-tahapan dalam pengerjaan Tugas Akhir meliputi perumusan CSF dan KPI, pembuatan *storyboard* dan tampilan KPI, penyempurnaan aplikasi mobile android sebelumnya , pembuatan *mobile dashboard* , Uji coba dan Evaluasi.

Bab 4 Analisa dan Desain Sistem

Membahas analisa dan perancangan desain *strategic mobile dashboard* pada Dinas Peternakan yang terdiri dari beberapa tahapan proses.

Bab 5 Uji Coba dan Analisa Hasil

Membahas hasil uji coba aplikasi yang telah dilakukan dan analisis terhadap hasil uji coba.

Bab 6 Kesimpulan dan Saran

Berisi kesimpulan yang diambil dari pengerjaan tugas akhir dan saran untuk pengembangan selanjutnya.

BAB 2
DASAR TEORI

Pada bab ini akan diuraikan mengenai dasar-dasar teori yang digunakan dalam pengerjaan tugas akhir ini. Teori yang akan dibahas adalah Jenis penyakit sapi, *strategic dashboard*, KPI (*Key performance Indikator), Chart*, Android, *dan AchartEngine*.

2.1 Penyakit Sapi

Penyakit sapi yang digunakan adalah penyakit yang ciri-cirinya dapat dilihat dari kondisi fisik secara langsung, dengan demikian dapat dilakukan diagnosa secara mandiri tanpa perlu membawa ke dokter hewan untuk pemeriksaan laboratorium. Penyakit sapi yang digunakan merupakan penyakit sapi yang dapat menimbulkan kerugian bagi peternak sapi dan kejadiannya sering terjadi di Indonesia, berikut adalah penjelasan tentang penyakit yang digunakan

2.1.1 Penyakit Dismatosis

Penyakit Distomatosis adalah penyakit *infeksius parasite* yang menyerang hati dan saluran empedu. Penyebab distomatosis adalah cacing Fasciola gigantica dan Fasciola hepatica yang hidup dalam saluran empedu. Stadium infektif yakni cercaria dan metacercaria fasciola yang menempel pada hijauan pakan ternak maupun air minum yang masuk menuju tempat predileksi, yakni hepar dan ductus biliverus. Gejala klinis distomatosis berupa sembelit disertsi diare, kurus, letih, lesu, lemah, pucat dan bulunya berdiri, depresi dan perut membesar. Bentuk kronis menyebabkan sapi mengalami penurunan produktivitas dan hambatan pertumbuhan terutama pada pedet. Terjadi busung atau oedem di bawah leher atau disebut dengan *bottle jaw*. Diagnosa

MILIK PERPUSTAKAAN ITS
dapat diperkuat dengan pemeriksaan sampel feses hewan suspect distomatosis. (Tjahajati, 2012)

2.1.2 Penyakit Mastitis

Penyakit mastitis atau yang biasa disebut radang ambing adalah penyakit yang sering menyerang ternak sapi. Penyakit ini menyebabkan produksi susu menurun, kualitas susu ayng dihasilkan juga turun. Apabila sapi terserang penyakit ini makan ambing sapi mengalami pembengkakan dalam skala tidak normal. Tanda-tanda lain penyakit ini adanya radang, yang berupa kebengkakan, panas dalam rabaan, rasa sakit (hati-hati atas sepakan waktu memeriksa), warna yang kemerahan dan terganggunya fungsi, jelas dapat ditemukan pada waktu pemeriksaan. Air susu jadi 'pecah', bercampur endapan. Konsistensi air susu jadi lebih encer dan warna juga menjadi agak kebiruan, atau putih yang pucat.

Karena penyakit disebabkan oleh kuman, maka pengobatannya dengan antibiotika. Pengobatan secara topical dengan linimen kamfer atau kompres lainnya masih dapat dibenarkan sepanjang tidak menambah beratnya. Pengobatan radang ambing hampir selalu memerlukan obatobatan antimicrobial, terutama antibiotika. Antibiotika yang terbukti dapat digunakan adalah penisilin, sefalosporin, eritromisin, neomisin, novobiosin, oksitetrasiklin, dan steptomisin atau dihidrostreptomisin. Selain pengobatan infusi intramamer, juga dapat dilakukan penyuntikan intramuskuler maupun intravena. (Tjahajati, 2012)

2.1.3 Penyakit *Bovine Ephemeral Fever* (BEF)

Penyakit Bovine Ephemeral Fever (BEF) atau penyakit pada sapi three day sickness (demam tiga hari), merupakan penyakit menular yang bersifat akut ditandai dengan adanya demam dan kepincangan. Penyebab penyakit demam tiga hari ini adalah virus bernama Rhabdovirus dari family Rhabdoviridae. Gejala klinis yang tampak pada sapi jika terserang BEF adalah

demam tinggi mencapai 41 derajat celcius yang berlangsung 3 hari, sapi tampak lesu, depresi, nafsu makan sapi menurun, produksi susu menurun, pada saat sapi demam terjadi konstipasi berlanjut dengan diare, persendian kaki sapi membengkak disertai dengan kekakuan otot anggota gerak sehingga sapi menjadi pincang kemudian jatuh/roboh. Masa inkubasi penyakit ini berjalan sekitar 7-10 hari dan akan sembuh dalam waktu 5-7 hari sejak muncul gejala klinis. (Tjahajati, 2012)

2.2 *Strategic Dashboard*

Dashboard adalah sebuah antarmuka grafis yang berisi ukuran kinerja bisnis yang memungkinkan pengambilan keputusan manajerial. Definisi ini terdiri dari tampilan visual dari konsep *dashboard*, isi, dan tujuan digunakannya *dashboard*. Istilah "*dashboard*" mengacu pada informasi yang ditampilkan pada *single screen* yang digunakan untuk memantau apa yang terjadi di beberapa aspek bisnis. (Few, 2006).

Bussiness Intelligence Dashboard berguna untuk memvisualisasikan KPI dari objektif yang telah ditentukan. Sebelum pembuatan *dashboard* diperlukan pendefinisian dari objektif *dashboard*. Objektif pada umumnya dibagi menjadi tiga, yaitu strategis objektif, taktikal objektif, dan operasional objektif. Setelah itu baru ditentukan *Critical Success Factor* (CSF), *Key Performance Indicator* (KPI) dan *measurement*. Keseluruhan komponen tersebut (objektif, CSF, KPI dan *measurement*) akan dimasukkan ke dalam dokumen terstruktur yang disebut *Balance Scorecard*.

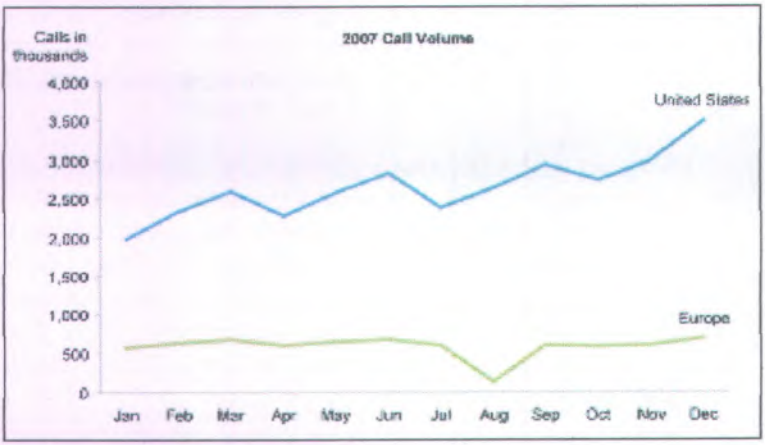
Visualisasi pada *dashboard* akan lebih cepat direspon dan dipersepsikan daripada dengan bahasa tulisan yang cenderung memakan waktu lebih lama dalam melakukan interpretasi. Perbandingan tampilan dapat dilihat pada gambar 2.1 dan 2.2 berikut ini :

2003 Call Volume (in thousands)

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
United States	1,983	2,343	2,593	2,263	2,574	2,838	2,382	2,634	2,938	2,739	2,983	3,493
Europe	574	636	573	593	644	679	593	139	599	583	602	690
	2,557	2,979	3,266	2,876	3,218	3,517	2,975	2,773	3,537	3,322	3,585	4,183

Gambar 2. 1 Tabel call volume data (Few, 2007)

Meskipun pada gambar 2.1 diatas hanya menampilkan 24 data, tetapi dalam membacanya akan membutuhkan waktu dan menggambarkan nilai yang disajikan juga sulit. Bandingkan dengan gambar berikut.



Gambar 2. 2 Grafik call volume data (Few, 2007)

Gambar 2.2 diatas adalah tampilan data yang sama namun disajikan dengan grafik garis yang sederhana. Dengan tampilan seperti itu kita akan dengan mudah mendapatkan fakta dari data yang ada tanpa melakukan usaha lebih dalam melakukan interpretasi.

Menurut Malik (2005), *Dashboard* harus memenuhi kriteria SMART yang terdiri dari beberapa elemen penting, yaitu :

- ***Synergetic.*** Harus ergonomis dan mempunyai visual yang efektif untuk pengguna guna mensinergikan informasi tentang aspek yang berbeda dalam *single screen.*
- ***Monitor KPI.*** Harus menampilkan KPI penting yang dibutuhkan untuk pengambilan keputusan yang efektif dimana dashboard dibutuhkan.
- ***Accurate.*** Informasi yang disajikan harus sepenuhnya akurat dalam rangka untuk mendapatkan kepercayaan penuh pengguna pada dashboard.
- ***Responsive.*** Harus mengenali ambang yang telah ditetapkan dengan menciptakan alert pengguna di samping presentasi visual di dashboard (misalnya, suara alarm, *e-mail, pager*) untuk menarik perhatian pengguna akan terjadinya hal penting.
- ***Timely.*** Harus menampilkan informasi terkini yang efektif bagi pengambilan keputusan. Informasi harus *real-time* dan tepat waktu.

Malik (2005) juga menyebutkan bahwa sebuah *dashboard* juga harus memiliki beberapa elemen canggih yang memenuhi kriteria IMPACT, yaitu :

- ***Interactive.*** Harus memungkinkan pengguna untuk melakukan drill down dan mendapatkan detail, sebab-sebabnya, dan lainnya.
- ***More data history.*** *Dashboard* harus memungkinkan pengguna untuk meninjau tren histori setiap KPI
- ***Personalized.*** Presentasi *dashboard* harus spesifik untuk masing-masing pengguna domain yang bertanggung jawab, mempunyai hak istimewa, terdapat pembatasan data, dan sebagainya.
- ***Analytical.*** Harus memungkinkan pengguna untuk melakukan analisis seperti analisis *what-if.*

- ***Collaborative.*** *Dashboard* harus memfasilitasi pengguna untuk melakukan pertukaran catatan penting dari pengamatan yang spesifik pada dashboard.
- ***Trackability.*** Harus memungkinkan setiap pengguna untuk menyesuaikan metrik yang ingin dilacak.

2.3 KPI (*Key Performance Indicator*)

Metrik digunakan dalam dashboard kinerja biasanya disebut kinerja utama Indikator (KPI) karena mereka mengukur seberapa baik organisasi atau individu melakukan suatu, taktis, strategis atau kegiatan operasional yang sangat penting untuk saat ini dan masa depan keberhasilan organisasi. (Eckerson, 2007)

Menurut Eckerson (2007), karakteristik dari KPI yang effectif adalah sebagai berikut.

1. ***Alligned***. KPI selalu selaras dengan strategi perusahaan dan tujuan.
2. ***Owned***. Setiap KPI, dimiliki oleh seorang individu atau kelompok pada sisi bisnis yang bertanggung jawab untuk hasilnya.
3. **Prediktif**. KPI mengukur nilai dari suatu bisnis
4. **Ditindaklanjuti**. KPI akan diisi dengan tepat waktu, ditindaklanjuti dengan data sehingga pengguna dapat meningkatkan kinerja sebelum terlambat.
5. **Sedikit jumlahnya**. KPI harus fokus pengguna pada tugas bernilai tinggi sedikit dan harus focus kepada nilai bisnis
6. **Mudah untuk memahami**. KPI harus langsung dan mudah dimengerti,tidak didasarkan pada indeks kompleks yang pengguna tidak tahu bagaimana mempengaruhi langsung.
7. **Seimbang dan terkait**. KPI harus menyeimbangkan dan memperkuat satu sama lain, tidak merusak proses satu sama lain .

8. **Memicu perubahan**. Tindakan pengukuran KPI yang harus memicu reaksi berantai perubahan positif dalam organisasi, terutama ketika dipantau oleh CEO.
9. **Standar**. KPI didasarkan pada definisi standar, aturan, dan perhitungan sehingga mereka dapat terintegrasi di dashboard seluruh organisasi.
10. ***Context Driven***. KPI menempatkan kinerja dalam konteks dengan target menerapkan dan ambang batas kinerja sehingga pengguna dapat mengukur kemajuan mereka atas waktu.
11. **Diperkuat dengan insentif**. Organisasi dapat memperbesar dampak KPI dengan melampirkan kompensasi atau *insentif* kepada mereka. Namun, mereka harus melakukan hal ini dengan hati-hati, menerapkan insentif hanya untuk dipahami
12. **Relevan**. KPI secara bertahap kehilangan dampaknya dari waktu ke waktu, sehingga harus dilakukan secara berkala .

2.4 Chart

Pendefinisan chart dan pemilihan chart penting dilakukan. Pemilihan tipe chart yang benar sangat penting untuk diputuskan untuk penggunaan chart. Grafik yang berbeda menampilkan data dalam cara yang sangat berbeda. Pemilihan chart yang terbaik akan membantu menampilkan data secara visual dengan cara yang lebih bermakna. (Harvey, 2010)

Di bawah ini adalah beberapa jenis-jenis chart :

- Grafik kolom: Sebuah grafik kolom, tidak seperti bar chart yang sering dibandingkan, menekankan variasi selama periode waktu. Dalam grafik kolom, kategori muncul horizontal dan nilai-nilai muncul secara vertikal, sedangkan dalam bar chart, kategori muncul secara vertikal. Variasi termasuk silinder, kerucut, dan limas bagan subtipe.

- Line chart: Sebuah grafik garis menunjukkan hubungan perubahan data selama periode waktu. Meskipun mirip dengan bagan daerah, yang menunjukkan kepentingan relatif dari nilai-nilai, line chart menekankan tren daripada jumlah perubahan.
- Pie chart: grafik Pie hanya berisi satu seri data grafik. Sebuah diagram pie menunjukkan hubungan antara bagian-bagian untuk keseluruhan. pilih salah satu diagram 2-D atau diagram pie 3-D.
- Bar chart: Sebuah diagram batang (bar horizontal) menekankan perbandingan antara item pada periode waktu tertentu. Ini jenis grafik juga mencakup silinder, kerucut, dan subtipe piramida.
- Grafik Area: Sebuah grafik daerah menunjukkan kepentingan relatif dari nilai-nilai dari waktu ke waktu. Bagan daerah ini mirip dengan line chart, tetapi karena daerah antara garis diisi, bagan daerah menekankan besarnya nilai tidak lebih dari line chart.

2.5 Aplikasi Mobile

Aplikasi yang melibatkan piranti bergerak dan komunikasi nirkabel. Aplikasi mobile merupakan aplikasi yang didesain untuk dijalankan di sebuah perangkat *mobile* pada sebuat tablet maupun *smartphone*. Aplikasi *mobile* sendiri sering berfungksi untuk menyediakan layananan yang sama dengan aplikasi yang dijalankan di PC, tetapi pada umumumnya aplikasi tersebut berukuran kecil dan memilki fungsi yang terbatas. (techopedia. 2010)

Aplikasi mobile berjalan pada sebuah perangkat *mobile* yang memiliki banyak jenis dalam hal ukuran, desain dan layout tetapi mereka memiliki kesamaan karakterisitik yang sangat berbeda dari sistem desktop. Beberapa karakteristik dari perangkat mobile antara lain adalah:

- Ukuran yang kecil
- Jenis dari perangkat ini. Dengan pembatasan jumlah dari memory, pertimbangan-pertimbangan khusus harus diambil untuk memelihara pemakaian dari sumber daya yang mahal ini.
- Daya proses yang terbatas
- Mengkonsumsi daya yang rendah

Berdasarkan karakteristik dari perangkat mobile tersebut, aplikasi *mobile* sangat cocok untuk diterapkan pada perangkat *mobile*. Sehingga sebuah aplikasi mobile tidak bisa menerapkan penulisan program yang rumit dan membutuhkan sumber daya komputasi yang besar. (J.E.N.I., 2010).

2.6 Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan sistem operasi yang terbuka bagi para pengembang untuk menciptakan aplikasi. (Safaat,2012). Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Tidak hanya menjadi system operasi di *smartphone*, saat ini android menjadi pesaing utama dari Apple pada system operasi *Table* PC. Pesatnya pertumbuhan Android selain faktor yang disebutkan diatas adalah karena Android itu sendiri adalah *platform* yang sangat lengkap baik itu sistem operasinya, aplikasi dan tool pengembangan. Market aplikasi android serta dukungan yang sangat tinggi dari komunitas open source di dunia, sehingga androiud terus berkembang pesat baik dasri segi teknologi mapun dari segi jumlah device yang ada di dunia.

2.6.1 Android SDK (*Software Development Kit*)

Android SDK adalah tools *Application Programming Interface* (API) yang diperlukan untuk mulai mengembangkan aplikasi pada sistem operasi Android menggunakan bahasa pemrograman Java.

Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang dikeluarkan oleh Google. Menurut Safaat (2012), beberapa fitur-fitur Android yang paling penting antara lain adalah :

- Framework aplikasi yang mendukung penggantian komponen dan *reusable*.
- Mesin virtual dalvik dioptimalkan untuk perangkat mobile.
- *Integrated* browser berdasarkan engine open source webkit.
- Grafis yang dioptimalkan dan didukung oleh librari grafis 2D, grafis 3D berdasarkan spesifikasi opengl ES 1,0.
- *SQLite* untuk menyimpsn data (Databasc).
- Media support yang mendukung audio, video, dan gambar (MPEG4, H264, MP3, AAC, AMR, JPG, PNG, GIF), GSM Telephony.
- Bluetooth, EDGE, 3G, dan Wifi.
- Kamera, GPS, Kompas, dan accelerator.

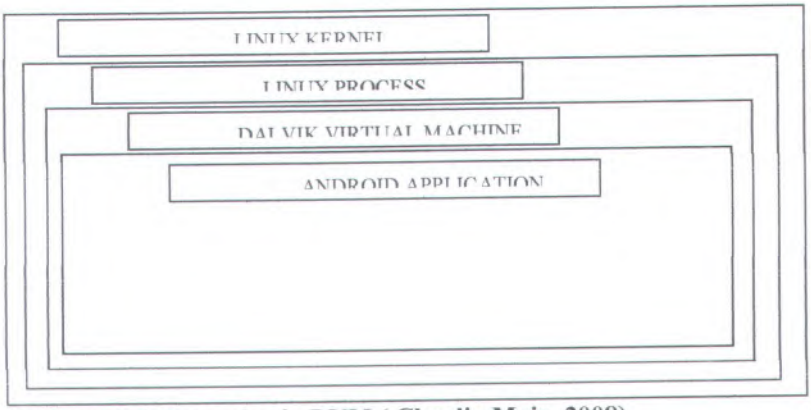
2.6.2 ADT (*Android Development Tools*)

Android Development Tools (ADT) adalah plugin yang didesain untuk IDE Eclipse yang memberikan kita kemudahan dalam mengembangkan aplikasi android dengan menggunakan IDE Eclipse. Dengan menggunakan ADT untuk Eclipse akan memudahkan kita dalam membuat aplikasi project android, membuat *GUI* aplikasi dan menambahkan komponen-komponen yang lainnya, begitu juga kita dapat melakukan running aplikasi menggunakan Adroid SDK melalui Eclipse. Dengan ADT kita juga dapat melakukan pembuatan

Package android (.apk) yang digunakan untuk distribusi aplikasi yang klita rancang.(Safaat,2012)

2.6.3 The Dalvik Virtual Machine

Salah satu elemen kunci dari Android adalah *Dalvik Virtual Machine* (DVM). (Safaat,2012) Android berjalan di dalam DVM, bukan pada *Java Virtual Machine* (JVM), banyak persamaan antara DVM dan JVM, tetapi Android menggunakan *Virtual Machine* sendiri yang dikostumisasi dan dirancang untuk memastikan bahwa beberapa fitur berjalan lebih efisien pada perangkat mobile. DVM merupakan sebuah mesin sederhana untuk menginterpretasikan java, yang secara menyeluruh dioptimalkan untuk sistem operasi Android dan dikembangkan untuk mengurangi penggunaan memori pada perangkat *mobile*. Dalvik tidak bisa membaca kode Java, namun Dalvik terdiri dari byte kode tersendiri yang disebut dengan dex maka file yang akan diekseksi memilih tipe file berkesteksi .dex. (Lacey, 2010).



Gambar 2. 3 Proses kerja DVM (Claudio Maia, 2008)

Semua akses yang dilakukan oleh aplikasi tersebut harus melalui Dalvik Virtual Machine terlebih dahulu. Jadi boleh dikatakan kalau aplikasi dengan ekstensi .dex tersebut adalah file *executable* untuk Android, yang bila kita berbicara tentang sistem operasi windows, maka file

.dex sama seperti file .exe pada windows. Sehingga dapat ditarik kesimpulan bahwa :

1. Dalvik VM merupakan tempat berjalannya aplikasi-aplikasi dari Android.
2. *Core Libraries* yang berada pada sistem Android sebagian besar merupakan core libraries yang terdapat pada bahasa pemrograman Java.
3. Setiap aplikasi yang dijalankan di Android, memiliki proses tersendiri dan memiliki instance masing-masing pada *Dalvik Virtual Machine*.
4. Dengan *Dalvik Virtual Machine* inilah, proses jalannya aplikasi menjadi efisien. Dalvik mengeksekusi file berekstensi dex yang disimpan dan dioptimalkan sehingga penggunaan memory di Android OS menjadi minimal.
5. *Dalvik Virtual Machine* bergantung pada kernal Linux (Android memakai Linux versi 2.6 untuk core sistemnya yang terdiri dari *security, memory management, process management, network stack* dan *driver model*) untuk fungsionalitas dasar seperti *threading* dan *low –level memory management*.

2.6.4 Aplication Framework

Android menawarkan kepada pengembang atau memberi kemampuan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi *resource*, menjalankan *service background*, mengatur alarm, dan menambahkan status notifikasi. Pengembang memiliki akses penuh menuju API framework. *Application framework* merupakan layer dimana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan

di sistem operasi Android, karena pada layer inilah dapat dirancang dan dibuat, seperti *content providers* yang berupa sms dan panggilan telepon. (Safaat,2012).

Komponen-komponen yang termasuk di dalam *Application Framework* adalah sebagai berikut :

- Views.
- Content Provider.
- Resource Manager.
- Notification Manager
- Activity Manager.

2.6.5 Libraries

Libraries merupakan layer pada Android dimana fitur-fitur Android berada, layer libraries diakses untuk menjalankan aplikasi.

2.6.6 Android Run Time

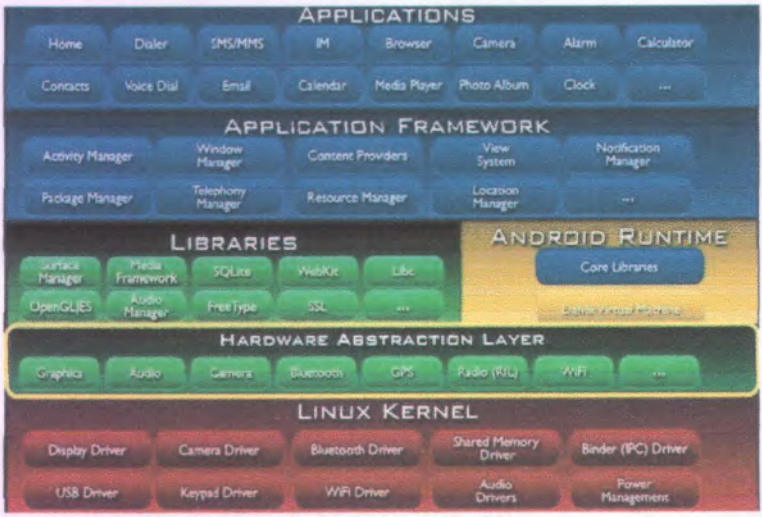
Layer yang membuat aplikasi Android dapat dijalankan dimana dalam prosesnya menggunakan implementasi Linux. *Dalvik Virtual Machine* (DVM) merupakan mesin yang membentuk dasar kerangka aplikasi Android. Di dalam Android *Run Time* dibagi menjadi dua bagian, yaitu :

- **Core Libraries** : Aplikasi Android dibangun dalam bahasa Java, sementara *Dalvik* sebagai virtual mesinnya bukan *Virtual Machine* Java, sehingga diperlukan sebuah libraries yang berfungsi untuk menterjemahkan bahasa Java ataupun C yang ditangani oleh *Core Libraries.*

- **Dalvik Virtual Machine** : Virtual mesin berbasis register yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien.

2.6.7 Linux Kernel

Linux kernel adalah layer dimana inti dari sistem operasi Android. Berisi file-file system yang mengatur sistem prosessing, memory, resource, drivers, dan sistem-sistem operasi android lainnya. (Safaat,2012).



Gambar 2. 4 Arsitektur Android

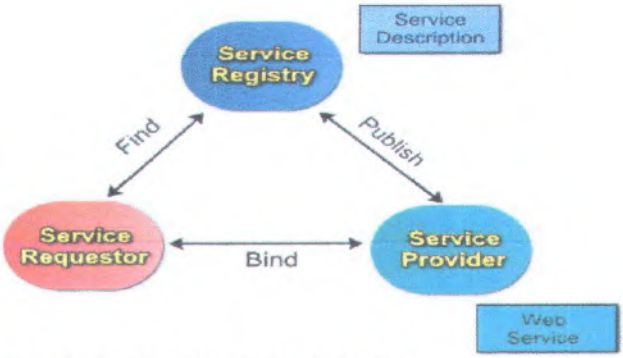
2.7 Web Service

Web service adalah suatu sistem perangkat lunak yang dirancang untuk mendukung *interoperabilitas* dan interaksi antar sistem pada suatu jaringan. *Web service* digunakan sebagai suatu fasilitas oleh suatu website untuk menyediakan layanan (dalam bentuk informasi) kepada sistem lain, sehinga sistem lain dapat berinteraksi dengan sistem tersebut melalui layanan yang

disediakan oleh suatu sistem yang menyediakan *web service*. *Web service* menyimpan data informasi dalam format XML, sehingga data ini dapat diakses oleh sistem lain walaupun berbeda platform, sistem operasi, maupun bahasa compiler. (Utama, 2010)

Web service bertujuan untuk meningkatkan kolaborasi antar pemrograman dan perusahaan, yang memungkinkan sebuah fungsi di dalam *web service* dapat dipinjam oleh aplikasi lain tanpa perlu mengetahui detil pemrograman yang terdapat di dalamnya. Adapun *web service* memiliki tiga entitas dalam arsitekturnya, yaitu :

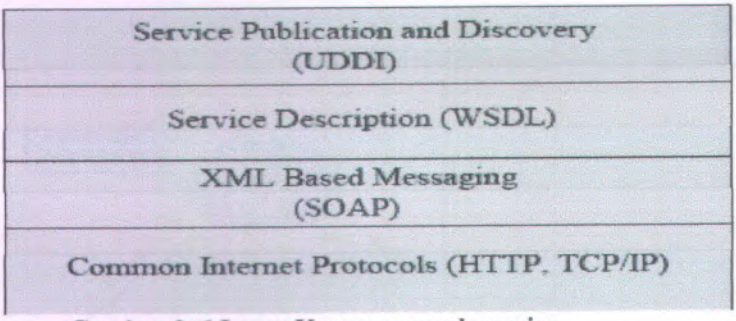
1. *Service Requester* (peminta layanan) berfungsi sebagai lokasi pusat yang mendekskripsikan semua layanan yang telah terdaftar.
2. *Service Provider* (penyedia layanan) berfungsi untuk menyediakan layanan dan mengolah sebuah registry agar layanan tersebut dapat tersedia
3. *Service Registry* (daftar layanan) berfungsi sebagai lokasi pusat yang mendeskripsikan semua layanan yang telah terdaftar



Gambar 2. 5 Arsitektur WebService

Web service secara keseluruhan memiliki tiga layer komponen seperti yang ditunjukkan pada gambar 2.6

1. Layer 1 : protokol internet standar seperti HTTP,TCP/IP
2. Layer 2 : *Simple Object Access Protocol* (SOAP), merupakan protokol akses objek berbasis XML yang digunakan untuk proses pertukaran data/informasi antar layanan.
3. Layer 3 : *Web Service Definition Language* (WSDL), merupakan suatu standar bahasa dalam format XML yang berfungsi untuk mendeskripsikan seluruh layanan yang tersedia.
4. Layer 4 : UDDI *(Universal Description, Discovery and Integration)* adalah sebuah service registry bagi pengalokasian webservice. UDDI mengkombinasikan SOAP dan WSDL untuk pembentukan sebuah registry API bagi pendaftaran dan pengenalan service.



Gambar 2. 6 Layer Komponen web service

2.7.1 Kelebihan Web Service

Beberapa kelebihan menggunakan *web service* adalah sebagai berikut :

1. *Web service* memiliki kemudahan dalam proses *deployment*-nya, karena tidak memerlukan registrasi khusus ke dalam suatu sistem operasi. *Web service* cukup di-upload ke web server dan

siap diakses oleh pihak-pihak yang telah diberikan otorisasi.

2. *Web service* berjalan di port 80 yang merupakan protokol standar HTTP, dengan demikian web service tidak memerlukan konfigurasi khusus di sisi firewall.
3. Aplikasi yang dibangun bisa terintegrasi dengan aplikasi lain dengan berbagai platform.
4. Security atau keamanan dari aplikasi bisa lebih terjaga dengan adanya otentikasi. (Utama, 2010)

2.8 Java Scrips Object Notation (JSON)

Java Script Object Notation (JSON) merupakan suatu format ringkas pertukaran data komputer. Formatnya berbasis teks dan terbaca manusia serta digunakan untuk merepresentasikan struktur data sederhana. Format JSON sering digunakan untuk mentransmisikan data terstruktur melalui suatu koneksi jaringan pada suatu proses yang disebut *serialisasi*. Format teks JSON benar menggunakan bahasa yang independen akan tetapi menggunakan konvensi yang akrab bagi *programmer* seperti C, C++, C#, Java, JavaScript, Perl, Phyton, dan lainnya.

JSON dibangun dalam 2 struktur :

- Kumpulan dari nama atau pasangan nilai.Dalam berbagai bahasa, ini direalisasikan sebagai object, record, struct, dictionary, hash table, keyed list, atau associative array.
- Daftar nilai, dalam banyak bahasa, ini dinyatakan sebagai *array*, vektor, *list*, atau urutan (JSON, 2013)

2.9 AcharEngine

AchartaEngine adalah pustaka untuk aplikasi Android. Mobile phone, tablet dan semua device yang berjalan menggunakan Android (http://achartengine.org/, 2013)

AchartEngine saat ini mengandung berbagai tipe chart :

- Line chart
- Scatter chart
- Time chart
- Bar chart
- Pie chart
- Doughnut chart
- Dial chart
- Combine chart

Semua tipe chart diatas dapat menggunakan multiple chart. Cara menggunakan *libarary* yaitu dengan mengimpor *library* kedalam project yang akan digunakan. Beberapa hal yang harus dilakukan dalam pembuatan chart dengan menggunakan *library AChartEngine* adalah sebagai berikut:

a. Menyiapkan activity dan data
Dalam pembuatan grafik diperlukan data dalam bentuk visual. Sehingga lebih bnyak dalam bentuk visual daripada text. Kode program dapat dilihat pada gambar 2.7

```
    int[] values = { 15, 3, 10 };
            String[] codes = new String[]
{
                        "Penyakit
Dismatosis",      "Penyakit      Mastitis",
"Penyakit BEF"
                    };
            CategorySeries  series  =  new
CategorySeries("Pie Graph");
    renderer.setChartTitle("Ternak    Sapi
yang Mati");
```

Gambar 2. 7 Persiapan data

Keterangan: Judul" Ternak Sapi yang mati " untuk memberikan nama untuk grafik, sedangakan array nilai berisi data untuk direpresentasikan dalam bentuk grafik.

b. Inisiasi dan Konfigurasi Grafik

Proses pembuatan grafik harus membuat object yang digunakan untuk merender grafik yaitu dengan memakai kelas XYmutipleSeriesRenderer , kemudian dilakukan proses konfigurasi. Proses dapat dilihat sebagai berikut :

```
    XYMultipleSeriesRenderer
mRenderer          =          new
XYMultipleSeriesRenderer();

      mRenderer.setChartTitle("Pe
rsebaran Penyakit Sapi");

      mRenderer.setXTitle("Jenis
Penyakit Sapi");
```

Gambar 2. 8 Inisiasisi dan konfigurasi grafik

Keterangan

- *mRenderer.setChartTitle* ("Persebaran Penyakit Sapi") ;Untuk Judul
- *mRenderer.setXTittle* ("Jenis Penyakit Sapi"); nama sumbu X.
- *mRenderer.setYTittle* ("Jumlah Penyakit Sapi"); nama sumbu Y.
- *mRenderer.seXAxisMax(5)* :Menentukan nila maksimum sumbu X.
- *mRenderer.setXAxisMin*(0) :Menentukan nila minimum sumbu X.

- *mRenderer.setYAxisMax*(5) :Menentukan nilai maksimum sumbu Y.
- *mRenderer.setYAxisMax*(50) :Menentukan nila maksimum sumbu Y.

c. Konfigurasi Grafik
Konfigurasi grafik pada beberapa jenis chart dapat dilakukan dengan mengganti warna ataupun mengganti jenis background pada chart. Kode program untuk *customisasi* dapat dilihat pada gambar 2.9 berikut ini :

```
renderer.setColor(Color.GREEN);
```

Gambar 2. 9Kustomisasi warna

Keterangan :

Renderer.setColor(color.Green); Untuk pemberian warna

d. Menampilkan Grafik
Untuk Menampilkan grafik ada beberapa konfigurasi yang harus dilakukan. Seperti pada gambar 2.10

```
    Intent          intent          =
ChartFactory.getBarChartIntent(context
, dataset,mRenderer, Type.DEFAULT);
            return intent;
```

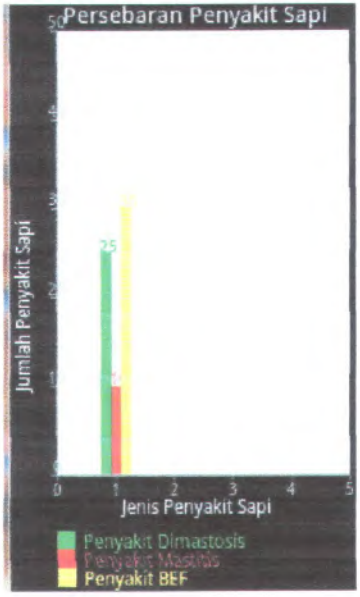
Gambar 2. 10 Menampilkan grafik

Keterangan :Grafik yang digunakan adalah bertipe barchart, maka perlu didapatkan intent dari kelas *ChartFactory* dengan memanggil method *getBarChartIntent* dengan melewatkan parameter context, dataset, *mRender* dengan typeBar default. Untuk menjalankan activity pelu melakukan tambahan *code* sebagai berikut

```
    <activity
android:name="org.achartengine
```

Gambar 2. 11 AndroidManifest.xml

Pada file *AndroidManifest.xml* perlu ditambahkan code 5.1 untuk menjalankan *activity* dari grafik yang sudah dibuat. Jika tidak ada masalah maka grafik tipe barchart akan bisa ditampilkan sebagai berikut



Gambar 2. 12 Contoh grafik

2.10 Hipotesa Sistem

Aplikasi *mobile dashboard* ini berguna untuk memvisualisasi data agar lebih mudah diintrepetasikan secara

mobile. Penyajian data dalam bentuk *dasboard* akan lebih memudahkan user dalam membaca data. Data yang tersaji merupakan rangkuman dari seluruh data, namun dalam bentuk yang lebih menarik, ringkas, sederhana dan lebih mudah dipahami. Dengan adanya *mobile dashboard* ini Dinas Peternakan akan lebih mudah dalam memonitoring persebaran penyakit sapi di kabupaten Malang.

Tahapan dari sistem *mobile dashboard* adalah sebagai berikut :

1. Mendapatkan indikator-indikator KPI yang dibutuhkan oleh Dinas Peternakan.
2. Setelah didapatkan KPI, dilakukan penambahan atribut pada aplikasi yang dibuat oleh Ardianto (2012)
3. Kemudian dilakukan pembuatan *Web service* dan koneksi database untuk melakukan penyimpanan data yang diinputkan pada aplikasi sebelumnya
4. Lalu dengan aplikasi mobile dashboard ini data akan diolah dan disajikan dalam bentuk visualisasi data misalnya diagram dan grafik.

Pada aplikasi *mobile dashboard*, ada beberapa fitur yang ditawarkan. Fitur-fitur tersebuat adalah :

1. Pada menu utama ada kategori dari dashboard yaitu berupa pilihan antara lain pilih berdasarkan penyakit atau obat.

2. Output dari Strategic Dashboard yang nanti disajikan

Tabel 2. 1 Jenis Chart

No	Key Performance Indicator (KPI)-Ukuran Keberhasilan	Jenis Chart yang digunakan
1	Jumlah ternak sapi yang terkena penyakit sapi berdasarkan kecamatan perbulan	Bar chart
2.	Jumlah ternak sapi yang terkena penyakit sapi berdasarkan desa perbulan	Bar Chart
3.	Jumlah obat yang diberikan berdasarkan kecamatan perbulan	Bar Chart
4.	Jumlah obat yang diberikan berdasarkan desa perbulan	Bar Chart

Dari penjelasan diatas, aplikasi mobile dashboard ini digunakan untuk memvisualiasasikan data yang berguna untuk memonitoring tingkat persebaran sapi di kabupaten Malang.

BAB 3
METODE PENELITIAN

Metodologi yang digunakan dalam tugas akhir ini adalah sebagai berikut :

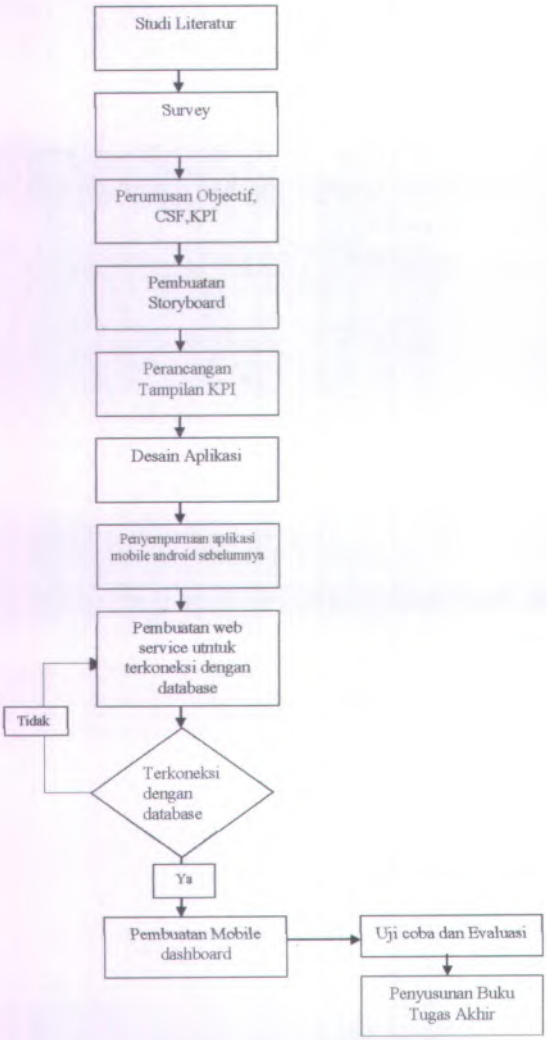
1. Pemahaman permasalahan dan studi literatur
   Pada tahapan ini dilakukan berbagai kajian pustakan terkait hal-hal yang mendukung pengerjaan tugas akhir ini. Pada tahapan ini juga mempelajari tentang indikator-indakator yang terkait dengan persebaran penyakit sapi, Android Mobile dan pembuatan dashboard. Dengan melakukan studi pustakan ini diharapkan dapat terindentifikasi konsep dasar yang harus dipahami sebelum membuat *mobile dashboard*.
2. Survey
   Survey dilakukan dalam bentuk wawancara dengan pihak terkait mengenai indikator-indikator persebaran penyakit sapi.
3. Penentuan dan Perumusan objectif , CSF dan KPI
   Setelah melakukan wawancara dan mengetahui gambaran umum dari permasalahan yang ada, selanjutnya dilakukan indentifikasi mengenai objectif yang ingin dicapai. Dan hal crtical yang mendukung tercapainya objectif. Penentuan Objectif, CSF dan KPI digunakan untuk memfokuskan *mobile dashboard* yang akan dibuat nantinya.
4. Pembuatan Storyboard
   Pembuatan rancangan awal dari storyboard ini digunakan untuk mematangkan konsep dari pembuatan dashboard yang dibuat benar-benar terkonsep dengan baik.
5. Perancangan Tampilan KPI
   Perancangan tampilan KPI digunakan untuk mendefiniskan tampilan KPI. Dengan adanya perancangan tampilan KPI

ini nantinya bagian-bagian dashboard menjadi terpertakan dan sesuai dengan keseluruhan taampilan pada dashboard.

6. Desain Aplikasi
   Desain aplikasi yang dibuat meliputi *use case diagram* dan *robusness diagram*. *Use case diagram* digunakan untuk menggambarkan berbagai macam fungsionalitas yang dapat dilakukan oleh pengguna terhadap sistem. *Robusness diagram* digunakan untuk menggambarkan iterasi antar object serta aliran data yang ditukarkan
7. Penyempurnaan aplikasi *mobile android* sebelumnya
   Pada tahapan ini dilakukan penyempurnaan terhadap aplikasi yang dibuat Ardianto (2012). Dalam tahapan ini dilakukan dengan penambahan beberapa KPI yang memang dibutuhkan oleh Dinas Peternakan Kabupaten Malang.
8. Pembuatan Web Service dan Koneksi database
   Pada tahap ini dilakukan penambahana pada fungsi web servis, dimana *web service* ini digunakan untuk tambahan tombol "simpan" pada aplikasi yang berfungsi untuk memasukkan indikator-indikator KPI yang dibutuhkan oleh Dinas Peternakan. Setelah terkoneksi dengan database, maka dibuat aplikasi *mobile dashboard*.
9. Pembuatan Mobile dashbord
   Pada tahapan ini dilakukan pembuatan aplikasi sesuai desain yang telah dibuat , serta pengimpentasian KPI yang telah disusun untuk dibuat dashboard. Aplikasi dibangun dengan bahasa pemprogaman java android yang dibangun menggunakan IDE Eclipse dan memanfaatkan database mySQL untuk menyimpan indikator-indikator data tentang persebaran penyakit sapi.
10. Uji coba dan Evaluasi
    Pada tahap uji coba dan evaluasi , aplikasi diuji dengan melihat pada storyboard apakah *mobile dashboard* yang dibuat sesuai dengan data apa tidak.

11. Pembuatan Buku TA
Membuat dokumentasi dari seluruh proses yang ada secara detail sesuai dengan permasalahan yang diangkat dalam Tugas Akhir

Secara Garis besar, metodologi yang sudah dijelaskan diatas dapat digambarkan dalam flowchart seperti yang ada pada gambar 3.1



Gambar 3. 1 Alur Pengerjaan Tugas Akhir

BAB 4
ANALISA DAN IMPLEMENTASI SISTEM

Pada bab ini akan dijelaskan mengenai desain dan implementasi *strategic dashboard* pada Dinas Peternakan dan Kesehatan kabupaten Malang yang terdiri dari beberapa proses. Pada proses analisa dilakukan analisa tujuan bisnis dari Dinas Peternakan. Pada tahap perancangan terdiri dari dari perancangan yang terkait dari pembuatan dashboard, penentuan *Key Performance Indicator* (KPI), perancangan desain dan alur dari pembuatan mobile *strategic dashboard* secara keseluruhan.

4.1 Analisa Tujuan Bisnis Dinas Peternakan

Analisa tujuan bisnis dari dinas peternakan didapat dari visi dan misi Dinas Peternakan dan kesehatan kabupaten Malang. Untuk mengetahui kondisi bisnis dari Dinas Peternakan dan Kesehatan maka dilakukan wawancara. Wawancara yang dilakukan anatara lain dengan Kepala bidang dinas Peternakan dan kesahatan kabupaten Malang. Salah satu misi dari Dinas Peternakan dan kesehatan kabupaten Malang yang didapat saat wawancara adalah menurunkan angka kematian / morbiditas ternak sapi di kabupaten Malang

Sebelumnya untuk proses interpretasi data menggunakan rekapan hasil dari Microsoft excel. Hasil rekapan data ini berdasarkan data yang didapat dari peternak, kemudian pihak dinas peternakan melakukan interpretasi data yang ada. Metode ini dirasa dinas peternakan kurang efektif dan efisien karena Dinas peternakan perlu memfilter dan mengintrepetasi data satu per satu sehingga butuh waktu cukup lama.

Beberapa permalasahan yang timbul ketika menerapkan metode yang sebelumnya antara lain :

1. Petugas Peternak yang ditugaskan di setiap desa, harus datang ke kantor kecamatan setiap kali pada saat

memberikan hasil rekapan penyakit sapi yang ada pada tiap desa.

2. Dinas Peternakan kesulitan melakukan proses rekap data setiap bulan mengenai jumlah penyakit sapi yang terjadi tiap kecamatan dan desa.
3. Dinas Peternakan dan kesehatan kesulitan mengintrepretasi hasil rekapan data secara realtime mengenai jumlah penyakit sapi yang terjadi pada setiap kecamatan dan desa.

Salah satu cara yang bisa dilakukan adalah mendeksi persebaran ternak yang terinfeksi secara realtime dan cepat. Sehingga Dinas Peternakan membutuhkan Aplikasi *Mobile Strategic Dashboard* yang membantu untuk memonitoring tingkat persebaran ternak sapi yang terinfeksi penyakit secara realtime yang kemudian digunakan untuk memberikan prioritas penanganan yang tepat sasaran nantinya.

Lingkungan Fisik :

Aplikasi *Strategic Mobile Dashboard* ini merupakan aplikasi berbasis android yang nantinya akan dijalankan dengan sistem *client-server* .

Lingkungan Teknologi :

Aplikasi Strategic *Mobile Dashboard* ini nantinya akan dijalankan dalam sistem operasi Android.

4.2 Identifikasi Jenis Dashboard dan Kelompok Pengguna

Berdasarkaan hasil identifikasi kebutuhan (sub bab 4.1), maka dapat diketahui pengguna dari *Mobile strategic Dashboard* adalah kepala Dinas Peternakan untuk memonitoring persebaran penyakit ternak sapi secara realtime. Sehingga bisa disimpulkan bahwa jenis dashboard yang cocok adalah *strategic dashboard* dalam bentuk mobile. Penggunaan *Mobile Strategic Dashboard* pada Dinas Peternakan dan Kesehatan untuk mempercepat menentukan daerah mana yang harus mendapatkan penanganan dengan segera, dan formulasi kebijakan yang tepat sasaran.

4.3 Kebutuhan Lingkungan

Hardware Requirements

Sistem yang digunakan menggunakan metode client-server. Untuk client menggunakan HP berbasis Android

Spesifikasi minimal untuk server :

- Space 1 GB
- OS Centos 5.5 x86_64
- Mysql 5
- Apache 2
- PHP 5
- B/W 2 GB/Bulan

Spesifikasi minimal client

- 2G Network GSM 850 / 900 / 1800 / 1900
- 3G Network HSDPA 900 / 2100
- GPRS Yes
- EDGE Yes
- Speed HSDPA, 7.2 Mbps
- WLAN Wi-Fi 802.11 b/g/n, Hotspot functionality
- OS Android OS, v2.3.5 (Gingerbread)
- CPU 830 MHz ARMv6
- Sensors Accelerometer, proximity, compass
- Browser WAP 2.0/xHTML, HTML
- GPS Yes, with A-GPS support

4.4 Identifikasi *Key Performance Indicator*

Metrik yang digunakan dalam tampilan dashboard disebut dengan *Key Performance Indicator* (KPI) karena metrik tersebut mengukur dan mendefiniskan seberapa baik suatu organisasi atau individu menunjukkan aktivitas operasional, taktikal atau strategis yang mendukung ketercapaian tujuan organisasi agar lebih tepat dan cepat diraih.

Untuk KPI Dinas Peternakan dan Kesahatan didapatkan dari penjabaran dari visi dan misi dari Dinas Peternakan yang dapat dilihat pada tabel 4.1 berikut

Tabel 4. 1 Tabel 4. 1 Key Perfomance Indicator Persebaran Penyakit Sapi

Tujuan	CSF	KPI	Measurement
Menurunkan angka kematian / morbiditas ternak sapi di kabupaten Malang	Monitoring terhadap penyakit sapi secara realtime	1. Jumlah ternak sapi yang terkena penyakit sapi berdasarkan kecamatan perbulan 2. Jumlah ternak sapi yang terkena penyakit sapi berdasarkan desa perbulan	Jumlah Jumlah

Tujuan	CSF	KPI	Measurement
	Monitoring terhadap penanganan yang diberikan secara realtime	1. Jumlah obat yang diberikan berdasarkan kecamatan perbulan 2. Jumlah obat yang diberikan berdasarkan desa perbulan	Jumlah Jumlah

4.5 Perancangan *Dashboard*

Pada tahap ini akan dijelaskan proses perancangan *mobile dashboard* dari kebutuhan Dinas Peternakan dan Kesehatan Kabupaten Malang.

4.5.1 Perencanaan Basis Data

Pada tugas akhir ini dilakukan perencanaan desain basis data sesuai dengan analisis kebutuhan dan KPI yang didapat. Basis data tersebut terdiri dari 5 tabel entitas antara lain :

- Table_fact_peternakan
- Table_user
- Table_kota
- Table_kecamatan
- Table_desa
- Table_dusun

a. Tabel_fact_peternakan

Tabel ini berfungsi untuk menyimpan segala infromasi yang terkait dengan data peternakan. Pada tabel 4.2 dijelaskan daftar atribut yang terkandung dalam "tabel fact_peternakan"

Tabel 4. 2 Daftar Atribut untuk Tabel Entitas "fact_peternakan"

Nama atribut	Tipe data	Keterangan
Id_peternakan	int	Nomor ID peternakan
Id_dusun	int	Nomor ID dusun
Nama_penyakit	Varchar (30)	Nama penyakit dari sapi
Nama_obat	Varchar(30)	Obat yang diberikan
Bulan	Varchar(30)	Tanggal penyakit sapi
Latitude	Varchar(30)	Posisi dari sapi yang diperiksa.

b. Table_user

Tabel ini berfungsi untuk menyimpan segala infromasi yang terkait dengan data *user/* pengguna. Pada tabel 4.3 dijelaskan daftar atribut yang terkandung dalam "tabel user"

Tabel 4. 3 Daftar atribut untuk Tabel Entitas "User"

Nama atribut	Tipe data	Keterangan
Username	Varchar(30)	Username
Password	Varchar (30)	Password
repassword	Varchar(30)	Verifikasi Password
Nama_lengkap	Varchar(30)	Tanggal penyakit sapi
Jekel	Varchar(30)	Jenis Kelamin
alamat	Varchar(30)	Alamat user
Nomor_tlp	Varchar(30)	Nomor telepon
Nomor_hp	Varchar(30)	Nomor hp

c. Kota

Tabel ini berfungsi untuk menyimpan segala infromasi yang terkait dengan nama kota. Pada tabel 4.4 dijelaskan daftar atribut yang terkandung dalam "tabel kota"

Tabel 4. 4 Daftar atribut untuk Tabel Entitas "Kota"

Nama atribut	Tipe data	Keterangan
Id_kota	int	Nomor ID kota
Nama_kota	Varchar (30)	Nama kota

d. Kecamatan

Tabel ini berfungsi untuk menyimpan segala infromasi yang terkait dengan nama kecamatan. Pada tabel 4.5 dijelaskan daftar atribut yang terkandung dalam "tabel kecamatan"

Tabel 4. 5 Daftar atribut untuk Tabel Entitas "Kecamatan"

Nama atribut	Tipe data	Keterangan
Id_kecamatan	int	Nomor ID kecamatan
kecamatan	Varchar (30)	Nama kecamatan

e. Desa

Tabel ini berfungsi untuk menyimpan segala infromasi yang terkait dengan nama desa. Pada tabel 4.6 dijelaskan daftar atribut yang terkandung dalam "tabel desa"

Tabel 4. 6 Daftar atribut untuk Tabel Entitas "Desa"

Nama atribut	Tipe data	Keterangan
Id_desa	int	Nomor ID desa
desa	Varchar (30)	Nama desa

f. Dusun

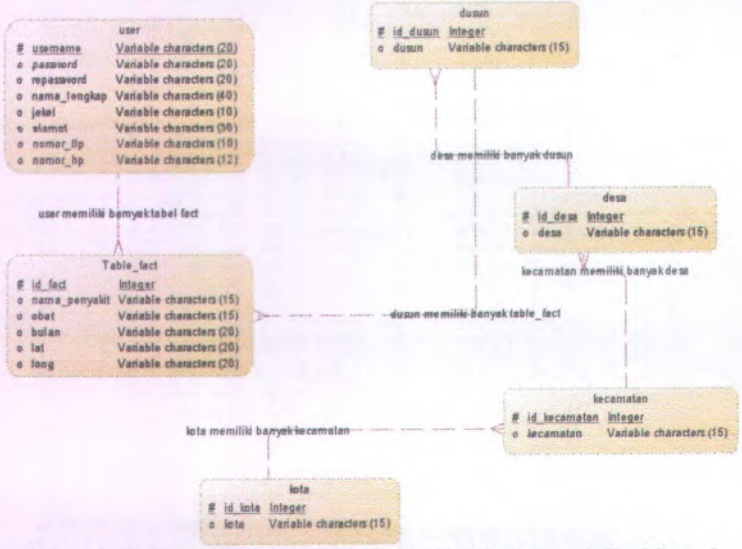
Tabel ini berfungsi untuk menyimpan segala infromasi yang terkait dengan nama dusun. Pada tabel 4.7 dijelaskan daftar atribut yang terkandung dalam "tabel dusun"

Tabel 4. 7 Daftar atribut untuk Tabel Entitas "dusun"

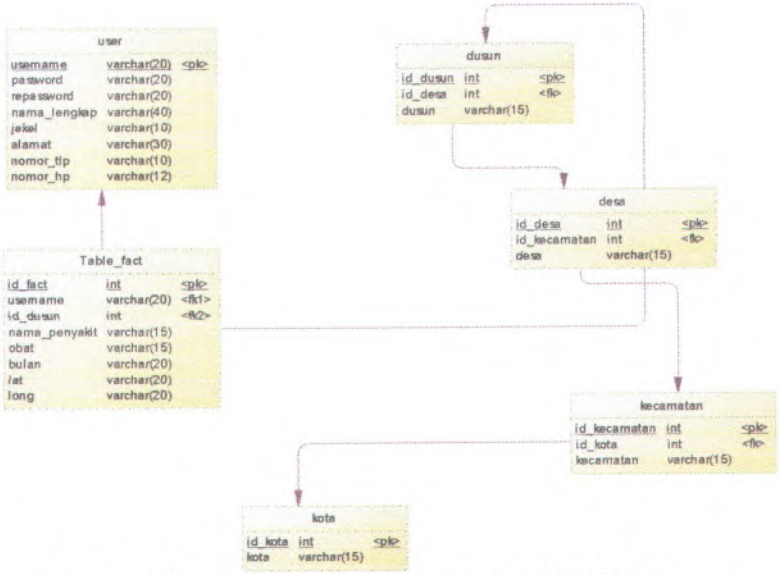
Nama atribut	Tipe data	Keterangan
Id_dusun	int	Nomor ID dusun
dusun	Varchar (30)	Nama dusun

Dari tabel-tabel diatas kemudian dilakukan pembuatan Concentual Data Model (CDM) dan Physical Data Model

(PDM) yang hasilnya dapat dilihat pada gambar 4.1 dan 4.2 berikut ini :



Gambar 4. 1 Conceptual data Model Untuk Mobile Strategi Dashboard



Gambar 4. 2 Physical Data Model Mobile Strategi Dashboard

4.5.2 Perencanaan Fungsional Sistem

Pada tahap ini akan dibahas mengenai perencanaan fungsional dari strategi *mobile dashboard* yang dibuat. Tahap ini merupakan tahap lanjutan dari perancangan basis data dimana tahap ini lebih mengambarkan spsesifikasi standar untuk mendokumentasikan perangkat lunak.

4.5.2.1 Use Case

Use Case diagram digunakan pada proses analisis untuk melihat *requirement sistem* dan untuk memahami bagaimana cara kerja sistem. Berikut use case yang ada :

a. Use Case Diagram Peternak

Use Case ini digunakan untuk memodelkan fungsi yang disediakan oleh sistem ketika digunakan oleh petugas kecamatan. Pada use case ini dijelaskan alur penginputan

data yang dilakukan petugas kecamatan ketika menginputkan data mengenai penyakit sapi yang diderita oleh tiap kecamatan dan desa. Lebih detail dapat dilihat pada tabel 4.8

Tabel 4. 8 Daftar Use case Peternak

No	Kode use case	Nama use case	Keterangan
1	UC-001	Use case Petugas Kecamatan log in	Petugas kecamatan melakukan log in sebelum melakukan insert data
2	UC-002	Use case Petugas kecamatan melakukan insert data	Petugas kecamatan melakukan insert data berdasarkan jenis penyakit , jenis obat, kota, kecamatan,desam dusun, dan posisi kordinat
3	UC-003	Use case pertugas kecamatan melakukan log out	Petugas kecamatan melakukan log out ketika sesudah menginputkan data

b. Use case Strategi Mobile Dashboard Diagram Dinas Peternakan

Use case ini digunakan untuk untuk memodelkan fungsi yang disediakan oleh sistem ketika menampilkan visualisasi data dalam bentuk grafik sesuai dengan jenis penyakit dan penanganannya sesuai pilihan pihak dinas peternakan. Lebih detail dapat dilihat pada tabel 4.9

Tabel 4. 9 Daftar Use case Dinas Peternakan

No	Kode use case	Nama use case	Keterangan
1	UC-001	Use case dapat memilih 2 jenis dashboard	Dinas Peternakan dapat memilih 2 jenis dashboard yaitu berdasarkan penyakit atau penangangan
2	UC-002	Use case memilih dashboard berdasarkan jenis penyakit	Dinas Peternakan memilih dashboard berdasarkan jenis penyakit
3	UC-003	Use case melihat grafik berdasarkan penyakit / kecamatan	Dinas Peternakan melihat grafik berdasarkan penyakit/ kecamatan
4	UC-004	Use case melihat grafik berdasarkan penyakit / desa	Dinas peternakan dapat melihat grafik berdasarkan penyakit/ desa
5	UC-005	Use case memilih dashboard berdasarkan jenis penanganan	Dinas perternakan dapat memilih dashboard berdasarkan jenis penanganan
6	UC-006	Use case melihat grafik berdasarkan penanganan / kecamatan	Dinas Peternakan dapat melihat grafik berdasarkan penanganan / kecamatan

No	Kode use case	Nama use case	Keterangan
7	UC-007	Use case melihat grafik berdasarkan penanganan / desa	Use case melihat grafik berdasarkan penanganan / desa

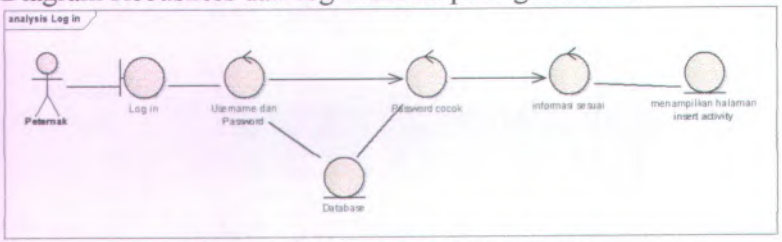
4.5.2.2 Robustness Diagram

Dari penjabaran use yang sudah dijabarkan diatas lalu dibuat Robusness Diagram. Robustness Diagram membantu untuk melihat alur tiap use case. Berikut Robussnes diagram yang ada :

a. **Robusness Diagram untuk Peternak**

1. **Robussness Diagram log in Peternak**

Pada diagram robusness log in peternak mengisi username dan password pada form log in. Jika username dan password sesuai maka akan masuk ke halaman input data penyakit. Diagram Robusness dari log in sesuai pada gambar 4.3

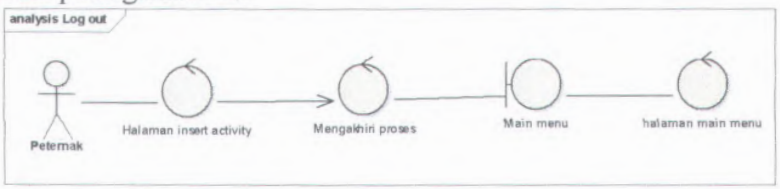


Gambar 4. 3 robusness diagram log in

2. **Robusness Diagram log out diagram log out**

Pada digram robusness log out ini peternak , peternak dapat melakukan log out dengan menekan tombol back yang berfungsi ke halaman sebelumnya ataupun dengan menekan tombol log out

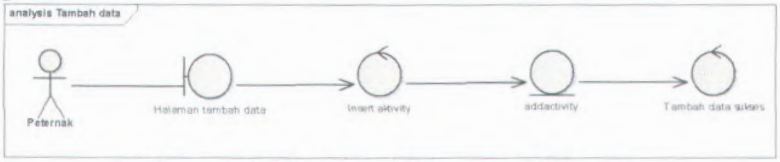
ketika pada halaman insert data. Diagram robusness dari log out sesuai pada gambar 4.4



Gambar 4. 4 robusness diagram log out

3. Robusness Diagram tambah data

Pada digram robusness tambah data data, petugas kecamatan masuk pada halaman insert atau tambah data, disini petugas kecamatan dapat mengisikan nama penyakit, obat, lokasi lalu menekan tombol simpan untuk menambahkan data ke server (database). Diagram robusness dari tambah data sesuai pada gambar 4.5

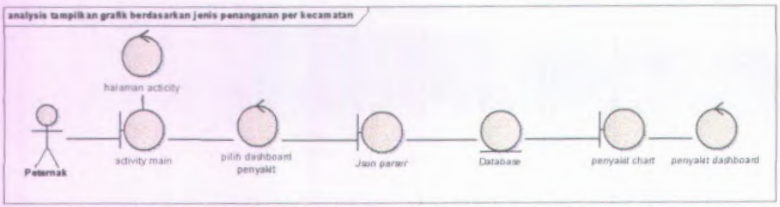


Gambar 4. 5 Robusness Diagram tambah data

b. Robusness Diagram untuk Dinas Peternakan

1. Robusness Diagram tampilkan grafik berdasarkan jenis penyakit per kecamatan

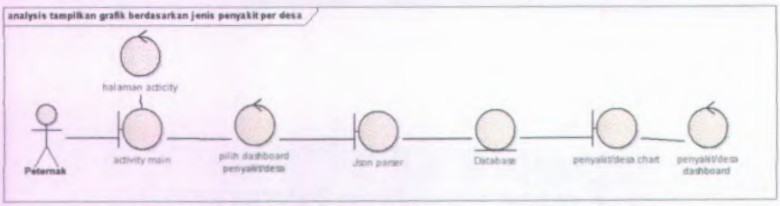
Pada diagram robusness tampilkan grafik berdasarkan jenis penyakit per kecamatan. Setelah memilih dashboard berdasarkan penyakit, maka akan muncul pilihan per kecamatan ataupun per desa. Kemudian dinas peternakan memilih opsi per kecamatan, disini akan ada pilihan jenis penyakit, bulan dan tahun sesuai dengan output yang dibutuhkan oleh dinas peternakan Diagram robusness dari penyakit / kecamatan sesuai pada gambar 4.6



Gambar 4. 6 Robusness Diagram penyakit /kecamatan

2. Robusness Diagram tampilkan grafik berdasarkan jenis penyakit per desa

Pada diagram robusness tampilkan grafik berdasarkan jenis penyakit per desa . Setelah memilih dashboard berdasarkan penyakit, maka akan muncul pilihan per kecamatan ataupun per desa. Kemudian dinas peternakan memilih opsi per desa, disini akan ada pilihan jenis penyakit, kecamatan , bulan dan tahun sesuai dengan output yang dibutuhkan oleh dinas peternakan. Diagram robusness dari penyakit / desa sesuai pada gambar 4.7

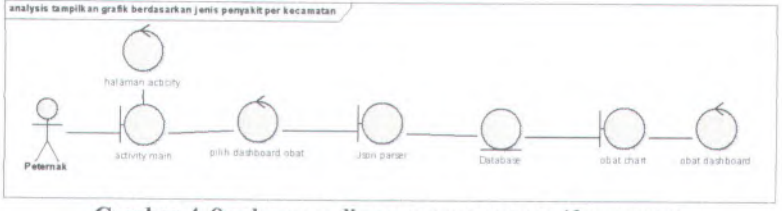


Gambar 4. 7 Robusness Diagram penyakit / desa

3. Robusness Diagram tampilkan grafik berdasarkan jenis penanganan per kecamatan

Pada diagram robusness tampilkan grafik berdasarkan jenis penanganan per kecamatan. Setelah memilih dashboard berdasarkan penanganan, maka akan muncul pilihan per kecamatan ataupun per desa. Kemudian dinas peternakan memilih opsi per desa, disini akan ada pilihan jenis obat, bulan dan tahun sesuai dengan output yang dibutuhkan oleh dinas peternakan.

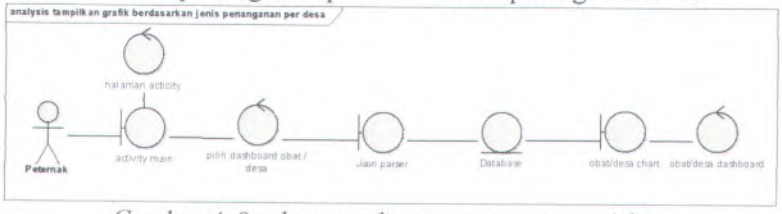
Diagram robusness dari penanganan per kecamatan sesuai pada gambar 4.8



Gambar 4. 8 robusness diagram penanganan / kecamatan

4. Robusness Diagram tampilkan grafik berdasarkan jenis penanganan per desa

Pada diagram robusness tampilkan grafik berdasarkan jenis penanganan per desa. Setelah memilih dashboard berdasarkan penanganan, maka akan muncul pilihan per kecamatan ataupun per desa. Kemudian dinas peternakan memilih opsi per desa, disini akan ada pilihan jenis obat, bulan dan tahun sesuai dengan output yang dibutuhkan oleh dinas peternakan. Diagram robusness dari penanganan per desa sesuai pada gambar 4.9



Gambar 4. 9 robusness diagram penanganan / desa

4.6 Pembangunan Sistem

Pada tahap ini akan menjelaskan tentang pembangunan aplikasi dari mobile strategic dashboard. Pada bagian ini yang akan dijelaskan adalah mengenai gambaran umum dari aplikasi , direktori dan source code.

4.6.1 Gambaran Umum Aplikasi

Aplikasi ini dibuat dengan mengunakan bahasa pemrogaman java pada sisi client kemudian pada sisi server

menggunakan bahasa pemrograman PHP. Untuk menghubungkan antara sisi client dan server dibutuhkan sebuah web servis sebagai pertukaran data. Sehingga meskpun berbeda bahasa pemrogaman akan tetap bisa melakukan pertukaran data. Untuk web service yang digunakan adalah berformat JSON (Java Script Object Notation).

4.6.2 Direktori dan File Kunci dalam Salinan Kerja Pengembang

Dalam pengembangan aplikasi, diperlukan sebuah direktori untuk menaruh aplikasi di dalam satu folder . Untuk direktori dari sisi server dapat dlihat pada tabel 4.10

Tabel 4. 10 Tabel Direktori Aplikasi sisi server

Path	Description
http://latihan.kopmaits.com/appmysql/	Tempat Menaruh seluruh file berenkstensi .php yang berisi query untuk melakukan koneksi antara server (server) dan client (android)

Untuk direktori dari aplikasi sisapi yang digunakan oleh petugas kecamatan dapat dilihat pada tabel 4.11

Tabel 4. 11 Tabel direktori Aplikasi sisi client(petugas Kecamatan)

Path	Description
E:\Trycode\SiSapi\src\com\SiSapi	Tempat menaruh seluruh class berekstensi .JAVA yang merupakan kode aplikasi untuk meninputkan hasil diagnosis dan lokasi
E:\Trycode\SiSapi\res	Tempat menaruh file yang berisi icon pada antarmuka sistem dimana pada folder ini

	masih terdapat 4 folder yaitu *drawable-hdpi,ldpi, mdpi, xdpi* yang menyesuaikan ukuran layar dari *smartphone*
E:\Trycode\SiSapi\res\layout	Tempat menaruh file yang berisi template css dan splashscreen sebagai antarmuka aplikasi.
E:\Trycode\SiSapi\res\values	Tempat menaruh file berekstensi xml yang berisi nilai untuk *interface* seperti nilai untuk *spinner*
E:\Trycode\SiSapi	Tempat menaruh seluruh file aplikasi dan file Android Manifest yang berisi *permission* untuk mengakses lokasi dan internet
E:\Trycode\SiSapi\assets	Tempat untuk menaruh *font file* untuk mempercantik tampilan antarmuka sistem

Untuk direktori dari aplikasi *strategic mobile dashboard* yang digunakan oleh petugas kecamatan dapat dilihat pada tabel 4.12

Tabel 4. 12 Tabel direktori Aplikasi pada sisi client (dinas Peternakan)

Path	*Description*
E:\Try code\TA_Dashboard\TA_Dashboard\ src\com\example\ta_dashboard	Tempat menaruh seluruh class berekstensi .JAVA yang merupakan kode aplikasi untuk menampilkan dashboard dalam bentuk grafik
E:\Try code\TA_Dashboard \res	Tempat menaruh file yang berisi icon pada antarmuka sistem dimana pada folder ini masih terdapat 4 folder yaitu *drawable-hdpi,ldpi, mdpi, xdpi* yang menyesuaikan ukuran

	layar dari *smartphone*
E:\Try code\TA_Dashboard \res\layout	Tempat menaruh file yang berisi template css dan splashscreen sebagai antarmuka aplikasi.
E:\Try code\TA_Dashboard \res\values	Tempat menaruh file berekstensi xml yang berisi nilai untuk *interface* seperti nilai untuk *spinner*
E:\Try code\TA_Dashboard	Tempat menaruh seluruh file aplikasi dan file Android Manifest yang berisi *permission* untuk internet
E:\Trycode\SiSapi\assets	Tempat untuk menaruh *font file* untuk mempercantik tampilan antarmuka sistem

4.6.3 Pembuatan fitur bagi petugas kecamatan

Pada bagian ini dijelaskan dokumentasi pembuatan aplikasi untuk fitur yang dikhususkan bagi petugas kecamatan ketika menginputkan data mengenai penyakit dan lokasi yang sebelumnya sudah dibuat oleh Wahyu (2012).

4.6.3.1 Pembuatan Fitur Input Data di Android

Dalam pembuatan fitur input data dalam aplikasi sebelumnya perlu menambahkan beberapa query agar bisa terkoneksi dengan internet lalu terkoneksi ke database secara online dan realtime. Berikut penambahan yang dilakukan antara lain :

1. **Menambahkan permission pada Android Manifest.xml**
   Penambahan *permission* pada Android Manifest.xml digunakan untuk memberikan *permission* /izin pada *device* ketika aplikasi me*request* akses ke *device.*

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/and
roid"
        package="com.SiSapi"
        android:versionCode="1"
        android:versionName="1.0" >

        <uses-sdk
            android:minSdkVersion="8" />
        <uses-permission
android:name="android.permission.INTERNET"/>
        <uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION
"/>
```

```
            <application
                android:icon="@drawable/logoapps"
                android:label="@string/app_name"
                android:theme="@drawable/logoapps">

            <activity
android:name="com.SiSapi.MainMenu"
                android:label="@string/app_name">
                <intent-filter>
                    <action
android:name="android.intent.action.MAIN" />
                    <category
android:name="android.intent.category.LAUNCHER" />
                </intent-filter>
            </activity>
    <activity
                android:name=".Diagnosa"
                android:label="@string/app_name">
            </activity>

             <activity
                android:name=".ReportDiagnosa"
```

```
                android:label="@string/app_name">
            </activity>

            <activity
                android:name=".KamusListview"
                android:label="@string/app_name">
            </activity>

    <activity android:name=".InsertActivity"


    android:label="@string/app_name"></activity>
            <activity android:name=".JSONActivity"
android:label="@string/app_name" ></activity>
            <activity android:name="Login"
android:label="@string/app_name"></activity>
            <activity android:name="CustomHttpClient"
android:label="@string/app_name"></activity>



     </application>

    </manifest>
```

Kode Program 4. 1 Permission pada file Android Manifest.xml

Pada file android Manifest.xml ini ditambahkan permission agar bisa mengakses lokasi dan internet yaitu dengan menambahkan kode

"***android.permission.ACCESS_FINE_LOCATION***" dan "***android.permission.INTERNET***"

4.6.3.2 Pembuatan Web Service dengan JSON untuk Menginputkan Data

Pembuatan *web service* berfungsi untuk menambahkan data yang dibuat dari sisi client dengan menggunakan bahasa pemrogaman Java di sisi client kemudian menggunakan bahasa pemrogaman PHP di sisi server. Ketika user (petugas kecamatan) menginputkan data, data tersebut dikirim melalui web server. Kemudian pada bagian server, data tersebut diolah oleh query mysql. Jika data tersebut sesuai dengan query maka data tersebut otomatis tersimpat di database (server).

Dibawah ini adalah potongan code java pada sisi client. Pada awal dilakukan pendifinisian inisisasi variable dan url yang merujuk pada *query* PHP yang berisi perintah sql utnuk penginputan data.

```
public class InsertActivity extends Activity  {

    TextView TeksPenyakit;
    TextView TeksObat;
    TextView TeksKota;
    TextView TeksKecamatan;
    TextView TeksDesa;
    TextView TeksDusun;
    TextView TeksLatitude;
    TextView TeksLongitude;
    RadioGroup penyakit;
    RadioGroup obat;
    Spinner
kota,kecamatan,kecamatan2,kecamatan1,kecamatanfix,des
a,desa1,desa2,desasby1,desasby2,desafix,dsnfix,dsn,ds
nmlg1,dsnmlg2;

  String Peternak1;
    private EditText txtLat;
```

```
        private EditText txtLng;
        private Button btnSimpan;
        Button btnLihatData,btnKembali;
        // Seusuaikan url dengan nama domain yang anda
gunakan
        private          String          url          =
"http://latihan.kopmaits.com/appmysql/addmakanan.php"
;
```

Kode Program 4. 2 inisiasi variabel

Berikut adalah potongan code ketika petugas kecamatan menekan tombol button.

```
     String lat =
URLEncoder.encode(txtLat.getText()

        .toString(), "utf-8");
                                String lng =
URLEncoder.encode(txtLng.getText()

        .toString(), "utf-8");
                                String
namaPeternak= URLEncoder.encode(Peternak1, "utf-
8");
                                url +=
"?namapenyakit=" + namapenyakit+ "&obat=" +
namaobat+ "&kota=" + namakota+

        "&kecamatan=" + namakecamatan+ "&desa=" +
namadesa+ "&dusun=" + namadusun+ "&lat=" + lat
                                    + "&lng="
+ lng + "&username="+ namaPeternak;
                                getRequest(url);
                            } catch
(UnsupportedEncodingException e) {
                                // TODO Auto-
generated catch block
```

```
        e.printStackTrace();
                        }

                        }
                });
```

Kode Program 4. 3 method simpan data

Pada saat tombol "simpan " ditekan makan sistem akan mengambil inputan untuk dikirim ke *web server.* Pada saat kode dikirim ke *web server*, kode tersebut di *encode* sehingga dapat diolah pada query PHP untuk menjalankan perintah SQL. Berikut method untuk menerima data dari server.

```
    public void getRequest(String Url) {
            /*Toast.makeText(this, "Tambah Data "
+ Url + " ", Toast.LENGTH_SHORT)
                        .show();*/
            HttpClient client = new
DefaultHttpClient();
            HttpGet request = new HttpGet(url);
            try {
                HttpResponse response =
client.execute(request);
                Toast.makeText(this, "Tambah
Data " + request(response) + " ",

      Toast.LENGTH_SHORT).show();
            } catch (Exception ex) {
                Toast.makeText(this, "Tambah
Data Gagal !", Toast.LENGTH_SHORT)
                                .show();
            }

      }
    public static String request(HttpResponse
```

```
response) {
            String result = "";

            try {
                  InputStream in =
response.getEntity().getContent();
                  BufferedReader reader = new
BufferedReader(
                              new
InputStreamReader(in));
                  StringBuilder str = new
StringBuilder();
                  String line = null;
                  while ((line =
reader.readLine()) != null) {
                        str.append(line +
"\n");
                  }
                  in.close();
                  result = str.toString();
            } catch (Exception ex) {
                  result = "Error";
            }
            return result;
      }
```

Kode Program 4. 4 *request* data ke server

Berikut ini adalah query SQL yang digunakan untuk menginputkan data , query ini berisi variable dan perintah untuk menginsertkan data ke database. Variabel-variabel yang diinputkan dalam query sql antara lain nama penyakit, obat yang diberika, tanggal pemeriksaan, lokasi sapi. Jika data berhasil diinputkan maka pesan sukses akan dikirim ke sisi client.

```
      <?php
            $namapenyakit=$_GET['namapenyakit'];
```

```
        $obat=$_GET['obat'];
         $kota=$_GET['kota'];
        $kecamatan=$_GET['kecamatan'];
        $desa=$_GET['desa'];
        $dusun=$_GET['dusun'];
        $lat=$_GET['lat'];
        $lng=$_GET['lng'];
        $username=$_GET['username'];

    $query =
    "INSERT INTO peternakan
(id,nama_penyakit,obat,bulan,kota,kecamatan,desa
,dusun,lat,lng,username) values
('','".$namapenyakit."','".$obat."',NOW(),'".$ko
ta."','".$kecamatan."','".$desa."','".$dusun."',
'".$lat."','".$lng."','".$username."')";
    $result = mysql_query($query, $link) or
die('Error query: '.$query);
    $query2 = "UPDATE penderita SET
jumlah_penderita=jumlah_penderita+1 WHERE
kecamatan_penderita = '".$kecamatan."' AND
nama_penyakit_penderita = '".$namapenyakit."'";
    $result2 = mysql_query($query2, $link) or
die('Error query: '.$query2);
       echo "SUCCESS";
    ?>
```

Kode Program 4. 5 Query Sql pada *script* PHP

Pembuatan desain antar muka

Desain antar muka dibuat sebagai interaksi antara sistem dengan pengguna. Pada halaman ini akan diisi oleh komponen-komponen seperti button, Radio Button (sebagai pilihan) dan Spinner. Berikut kode program untuk membuat tampilan anter muka untuk inputan user

```
    <ScrollView
```

```
xmlns:android="http://schemas.android.com/apk/res/android"
          android:layout_width="match_parent"
          android:layout_height="match_parent"
          android:fillViewport="true"
          android:orientation="vertical" >
      <LinearLayout
          android:layout_width="fill_parent"
          android:layout_height="fill_parent"
          android:orientation="vertical"

android:background="@drawable/backgroundaplikasi">
          <ImageView
            android:src="@drawable/logoapps"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:contentDescription="Logo"/>

          <TextView
            android:text="Nama Penyakit :"
            android:id="@+id/TeksPenyakit"
            android:textColor="@color/black"
               android:layout_width="fill_parent"

       android:layout_height="wrap_content"></TextView>
       <RadioGroup
           android:id="@+id/txtNamaObat"
           android:orientation="horizontal"
           android:layout_width="fill_parent"
                     android:layout_height="wrap_content"
                     android:padding="5px">

                     <RadioButton
                         android:id="@+id/antibiotik"
                         android:text="Antibiotik"

android:textColor="@color/black"/>
                     <RadioButton
                         android:id="@+id/antipiretik"
                         android:text="Antipiretik"

android:textColor="@color/black"/>
                     <RadioButton
                         android:id="@+id/anthelmentik"
```

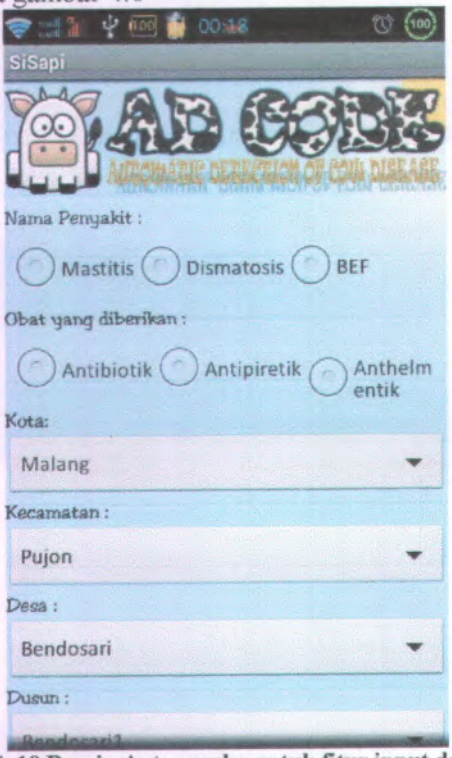
```
                        android:text="Anthelmentik"

android:textColor="@color/black"/>
            </RadioGroup>

        <TextView android:text="Kota:"
            android:id="@+id/TeksKota"
            android:textColor="@color/black"
                android:layout_width="fill_parent"

        android:layout_height="wrap_content"></TextView>
       <Spinner
              android:id="@+id/pilihkota"
              android:layout_width="fill_parent"
              android:layout_height="wrap_content" />

        <TextView
            android:text="Kecamatan :"
            android:id="@+id/TeksKecamatan"
            android:textColor="@color/black"
                android:layout_width="fill_parent"
        android:layout_height="wrap_content">
        </TextView>

         <Spinner
              android:id="@+id/planspinner"
              android:layout_width="fill_parent"
              android:layout_height="wrap_content" />

        <TextView android:text="Desa :"
            android:id="@+id/TeksDesa"
                android:textColor="@color/black"
                android:layout_width="fill_parent"

        android:layout_height="fill_parent"></TextView>
         <Spinner
              android:id="@+id/pilihdesa"
              android:layout_width="fill_parent"
              android:layout_height="fill_parent" />

        <TextView android:text="Dusun :"
            android:id="@+id/TeksDusun"
```

```
                android:textColor="@color/black"
                android:layout_width="fill_parent"

        android:layout_height="fill_parent"></TextView>

         <Spinner
               android:id="@+id/pilihdusun"
               android:layout_width="fill_parent"
               android:layout_height="fill_parent" />

      <TextView android:text="Latitude :"
            android:id="@+id/TeksLatitude"
                android:textColor="@color/black"
                android:layout_width="fill_parent"

android:layout_height="wrap_content"></TextView>

        <EditText android:text="" android:id="@+id/txtLat"
                android:layout_width="fill_parent"

        android:layout_height="wrap_content"></EditText>

        <TextView android:text="Longitude :"
            android:id="@+id/TeksLongitude"
                android:layout_width="fill_parent"
      <TextView android:text="Latitude :"
            android:id="@+id/TeksLatitude"
                android:textColor="@color/black"
                android:layout_width="fill_parent"

android:layout_height="wrap_content"></TextView>

        <EditText android:text=""
android:id="@+id/txtLat"
                android:layout_width="fill_parent"

        android:layout_height="wrap_content"></EditTex
t>

        <TextView android:text="Longitude :"
            android:id="@+id/TeksLongitude"
```

```
            android:layout_width="fill_parent"

        android:layout_height="wrap_content"></TextVie
w>

        <EditText android:text=""
            android:textColor="@color/black"
            android:id="@+id/txtLng"
               android:layout_width="fill_parent"

        android:layout_height="wrap_content"></EditTex
t>
```

Kode Program 4. 6 Desain antar muka untuk inputan data

Hasil desain antar muka yang dibuat untuk user (petugas kecamatan), Komponen spinner untuk menampilkan lokasi(Kota, kecamatan, Desa dan dusun) saling berkaitan (dependent) sehingga memudahkan user ketika memilih lokasi pencarian dapat dilihat pada gambar 4.6



Gambar 4. 10 Desain Antar muka untuk fitur input data

4.6.4 Pembuatan Aplikasi *Strategic Mobile Dashboard* Untuk Dinas Peternakan

Pembuatan Aplikasi *Strategic Mobile dashboard* untuk memonitoring penyakit sapi secara realtime dalam bentuk grafik memilki beberapa langkah, antara lain dengan

mengimplementasikan library Achartengine untuk pembuatan grafiknya.

4.6.4.1 Implementasi Library Achartengine pada Android Project

Dalam pembuatan *Aplikasi Mobile Strategic Dashboard* di android ada beberapa tahap yang perlu dilakukan

1. Pastikan telah mendownload library dari http://www.achartengine.org/
2. Kemudian untuk mengimport, library ke project yaitu dengan cara, klik kanan pada libs lalu pilih build path pada project. Pada Tab library , pilih add external jars kemudian pilih library yang sudah di download tadi.
3. Menambahkan permission dan tambahan pada Android Manifest.xml. Berikut potongan kode dari Android Manifest yang sudah ditambahkan beberapa permission seperti akses internet dan untuk menjalankan library achartengine

   Berikut implementasi kode program permission *library achartengine* dapat dilihat pada gambar 4.7

```
<uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="15" />
   <uses-permission
android:name="android.permission.INTERNET" />
    <uses-permission
android:name="android.permission.WRITE_EXTERNAL_S
TORAGE" />
    <application
        android:allowBackup="true"
        android:icon="@drawable/icon"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity android:name="splash"
android:label="@string/app_name">
```

```
            <intent-filter>
        <action
android:name="android.intent.action.MAIN" />
        <category
android:name="android.intent.category.LAUNCHER"
/>
    </intent-filter>
        </activity>
        <activity android:name="Login"
android:label="@string/app_name"></activity>
              <activity
android:name="org.achartengine.GraphicalActivity"
/>
```

Kode Program 4. 7 Permission dan Library Achartengine pada Android Manifest.xml

4.6.4.2 Pembuatan *Web Service* dengan JSON untuk Menampilkan Data

Pada Bagian ini akan dijelaskan tentang pembuatan *Web Service* yang digunakan untuk menampilkan data pada Aplikasi *Mobile Strategic Dashboard*. Terdapat 2 bagian utama yaitu pada class .java dan membuat perintah SQL. Untuk lebih detailnya akan dijelaskan pada bagian berikut ini.

1. **Membuat class Java untuk sinkronisasi dengan database**

Pada bagian ini akan dijelaskan menegenai potongan program yang berfungsi sebagai sinkronisasi dengan database yang ada pada server, dimana kelas ini menerima input data dari user dan mengirimlan ke web server dan menerima hasil client dari perintah SQL yang dijankan kemudian dikembalikan lagi hasilnya ke client. Kode programmnya sebagai berikut :

```
package com.example.ta_dashboard;

import java.io.BufferedReader;
```

```
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.UnsupportedEncodingException;
import java.util.List;

import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.client.utils.URLEncodedUtils;
import org.apache.http.impl.client.DefaultHttpClient;
import org.json.JSONException;
import org.json.JSONObject;

import android.util.Log;

public class JSONParser {
        static InputStream is = null;
    static JSONObject jObj = null;
    static String json = "";

    // constructor
    public JSONParser() {

    }

    // function get json from url
    // by making HTTP POST or GET mehtod
    public JSONObject makeHttpRequest(String url, String method,List
params) {

        // Making HTTP request
        try {

            // check for request method
```

```
        if(method == "POST"){
            // request method is POST
            // defaultHttpClient
            DefaultHttpClient httpClient = new DefaultHttpClient();
            HttpPost httpPost = new HttpPost(url);
            httpPost.setEntity(new UrlEncodedFormEntity(params));
            HttpResponse httpResponse = httpClient.execute(httpPost);
            HttpEntity httpEntity = httpResponse.getEntity();
            is = httpEntity.getContent();

        }else if(method == "GET"){
            // request method is GET
            DefaultHttpClient httpClient = new DefaultHttpClient();
            String paramString = URLEncodedUtils.format(params, "utf-
8");
            url += "?" + paramString;
            HttpGet httpGet = new HttpGet(url);

            HttpResponse httpResponse = httpClient.execute(httpGet);
            HttpEntity httpEntity = httpResponse.getEntity();
            is = httpEntity.getContent();
        }

    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    } catch (ClientProtocolException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }

    try {
        BufferedReader reader = new BufferedReader(new
InputStreamReader(
                is, "iso-8859-1"), 8);
        StringBuilder sb = new StringBuilder();
        String line = null;
        while ((line = reader.readLine()) != null) {
            sb.append(line + "\n");
```

```
        }
        is.close();
        json = sb.toString();
    } catch (Exception e) {
        Log.e("Buffer Error", "Error converting result " + e.toString());
    }

    // try parse the string to a JSON object
    try {
        jObj = new JSONObject(json);
    } catch (JSONException e) {
        Log.e("JSON Parser", "Error parsing data " + e.toString());
    }

    // return JSON String
    return jObj;

  }
}
```

Kode Program 4. 8 Sinkronisasi dengan database

Pada *kode program* diatas terdapat operator logika ketika user memilih *method* untuk menyimpan maka pada kelas sinkronisasi yang akan dipilih ada "POST" sebaliknya ketika user ingin mendapatkan dari *web server* maka method yang dipilih bernilai "GET"

2. Menampilkan hasil perintah dari SQL

Untuk menampilkan hasil perintah hasil SQL maka data yang ditampung harus berformat JSON. Jadi pada setiap data yang ditampung dalam sebuah array untuk ditampilkan di sisi client. Berikut *kode program* untuk menampilkan hasil perintah SQL di sisi client.

```
class LoadProfile extends AsyncTask<String,
String, String> {
            String sret="";
```

```
            ArrayList<Integer> nama_penyakit =
new ArrayList<Integer>();
            Integer[] nama_penyakitdashboard;

            ArrayList<String> kecamatan = new
ArrayList<String>();
            String[] kecamatan_dashboard;

        @Override
        protected void onPreExecute() {
            super.onPreExecute();
            pDialog = new
ProgressDialog(chart.this);
            pDialog.setMessage("Retrieving data.
Please wait...");
            pDialog.setIndeterminate(false);
            pDialog.setCancelable(false);
            pDialog.show();
             }
```

Kode Program 4. 9 Menampilkan hasil query SQL 1

Pada potongan *method* diatas digunakan untuk menampung *array* yang nantinya hasilnya akan dijalan pada SQL. Ketika *method* on Pre Excecute dijalankan akan menampilkan data makan akan muncul pesan

"Retrieving data. Please wait..."

Berikut adalah kode program untuk menjalankan proses ketika query didalam *background* :

```
  protected String doInBackground(String... args)
{
            List<BasicNameValuePair> params = new
ArrayList<BasicNameValuePair>();
            params.add(new
BasicNameValuePair("nama_penyakit", Penyakit));
            params.add(new
BasicNameValuePair("bulan", Bulan));
            params.add(new
```

```
BasicNameValuePair("tahun", Tahun));
            JSONObject json =
jParser.makeHttpRequest(base_url, "GET", params);
                  Log.d("All      Accounts:      ",
json.toString());
```

Kode Program 4. 10 Menampilkan hasil query SQL 2

Pada potongan method do *method* doInBackground berfungsi sebagai proses untuk menjalankan query sesuai dengan data yang diinputkan oleh *user* dan mengirimkan hasil inputan user ke lokasi *script* PHP melalui kelas sinkronisasi untuk menjalankan perintah SQL dan mengambil data dari *web server*

Berikut ini adalah kode program untuk menyimpan hasil array :

```
try {
                int success = json.getInt(TAG_SUCCESS);
                if (success == 1) {
                    JSONArray menuitemArray =
json.getJSONArray(TAG_ACCOUNT);
                  for (int i = 0; i <
menuitemArray.length(); i++) {


      nama_penyakit.add(Integer.parseInt((menuitemArray
.getJSONObject(i).getString("nama_penyakit"))));
                  nama_penyakitdashboard = new
Integer[nama_penyakit.size()];
                  nama_penyakitdashboard =
nama_penyakit.toArray(nama_penyakitdashboard);


      kecamatan.add((menuitemArray.getJSONObject(i).get
String("kecamatan")));
                  kecamatan_dashboard = new
String[kecamatan.size()];
                  kecamatan_dashboard =
kecamatan.toArray(kecamatan_dashboard);
```

```
                }

            }
        } catch (JSONException e) {
            e.printStackTrace();
        }
        return "";
         }
```

Kode Program 4. 11 Menampilkan hasil query SQL 3

Kode program diatas merupakan bagian dari method doInBackground yang berfungsi untuk menyimpan hasil perintah sql yang telah didapat dari query SQL ke dalam *array* yang sudah diinisiasikan diawal. Kode program dibawah ini adalah method dimana untuk menampilkan dalam bentuk grafik pada aplikasi *mobile strategic dashboard*.

```
  protected void onPostExecute(String file_url) {
            pDialog.dismiss();

            //Hasil.setText(sret);

            Intent barchart=new
Intent().setClass(chart.this,bargraph.class);
            Bundle c = new Bundle();
                  c.putIntegerArrayList("hasil",
nama_penyakit);

      c.putStringArrayList("namakecamatan",
kecamatan);
                  barchart.putExtras(c);
                  startActivity(barchart);
                  finish();

                }
```

Kode Program 4. 12 method untuk menampung hasil array SQL

```
public class bargraph extends Activity {
	ArrayList<Integer> y = new ArrayList<Integer>();
	Integer[] hasil;

	ArrayList<String> x = new ArrayList<String>();
	String[] hasil_kecamatan;



	 public void onCreate(Bundle savedInstanceState) {
		super.onCreate(savedInstanceState);

		Bundle c = getIntent().getExtras();
		y=c.getIntegerArrayList("hasil");
		x=c.getStringArrayList("namakecamatan");
		hasil = new Integer[y.size()];
	hasil = y.toArray(hasil);
	hasil_kecamatan = new String[x.size()];
	hasil_kecamatan = x.toArray(hasil_kecamatan);
				// Bar 1
				//int[] y = { 100, 80, 20, 250
};
				CategorySeries series = new
CategorySeries("Jumlah Penyakit per Kecamatan");
				for (int i = 0; i <
hasil.length; i++) {
					series.add("Bar " +
(i+1), hasil[i]);
				}


				XYMultipleSeriesDataset
dataset = new XYMultipleSeriesDataset();

	dataset.addSeries(series.toXYSeries());

				XYMultipleSeriesRenderer
mRenderer = new XYMultipleSeriesRenderer();

	mRenderer.setChartTitle("Persebaran Sapi");
```

```
        mRenderer.setXTitle("Kecamatan");
                                mRenderer.setYTitle("Jumlah
Penyakit Sapi");

        mRenderer.setAxesColor(Color.BLACK);

mRenderer.setLabelsColor(Color.WHITE);
                        mRenderer.setXAxisMin(0.5);
                        mRenderer.setYAxisMin(0);
                        mRenderer.setYAxisMax(25);
                        mRenderer.setBarSpacing(0.2);
                        mRenderer.setPanEnabled(true,
false);
                        mRenderer.setZoomEnabled(true);

mRenderer.setZoomButtonsVisible(true);
                        mRenderer.setZoomRate(1.1f);

mRenderer.setApplyBackgroundColor(true);

mRenderer.setBackgroundColor(Color.parseColor("#F5F5F5"));

mRenderer.setMarginsColor(Color.parseColor("#4E0263"));
                        mRenderer.setShowGrid(true);
                        mRenderer.setGridColor(Color.GRAY);
                        mRenderer.setLabelsTextSize(12);

mRenderer.setAxisTitleTextSize(12);

mRenderer.setChartTitleTextSize(12);



                        for (int abc = 0; abc <
hasil_kecamatan.length; abc++) {

mRenderer.addXTextLabel(abc+1, hasil_kecamatan[abc]);

mRenderer.setXLabelsAlign(Align.LEFT);
```

```
        mRenderer.setXLabelsAngle(45);
                                }

                            // Kustomisasi Bar 1
                                XYSeriesRenderer renderer =
new XYSeriesRenderer();


        renderer.setDisplayChartValues(true);

        renderer.setChartValuesSpacing((float)5.0);

        renderer.setChartValuesTextAlign(Align.RIGHT);
                                renderer.setColor(Color.BLUE);

        mRenderer.addSeriesRenderer(renderer);




                                Intent intent =
ChartFactory.getBarChartIntent(this, dataset,mRenderer,
Type.STACKED);
                                startActivity(intent);
        }


    }
```

Kode Program 4. 13 menampilkan hasil query sql ke dalam dashboard 1

4.6.4.3 Pembuatan Fitur Melihat Grafik Berdasarkan Penyakit / Kecamatan

Pada bagian ini akan dijelaskan pembuatan fitur melihat grafik berdasarkan penyakit / kecamatan. Langkah-Langkah pembuatan akan dijelaskan sebagai berikut.

1. **Membuat halaman untuk mengentry data berdasarkan penyakit per kecamatan**

Berikut source code untuk halaman entry data berdasarkan penyakit per kecamatan, dimana terdapat komponen Spinner yang berisi jenis penyakit, nama bulan , dan tahun yang akan dipilih oleh dinas Peternakan dan Kesehatan hewan

```
android:id="@+id/txt1"
        android:text="PILIH PENYAKIT"
        android:textSize="28dp"
        android:textStyle="bold"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center"/>

       <RadioGroup
           android:id="@+id/txtNamaCariPenyakit"
           android:orientation="horizontal"
           android:layout_width="fill_parent"
              android:layout_height="wrap_content"
              android:padding="2px">
                    <RadioButton

android:id="@+id/penyakitmastitis"
                              android:text="Mastitis"

android:textColor="@color/black"/>
                         <RadioButton

android:id="@+id/penyakitdismatosis"
                              android:text="Dismatosis"

android:textColor="@color/black"/>
                         <RadioButton
                              android:id="@+id/penyakitbef"
                              android:text="BEF"

android:textColor="@color/black"/>
             </RadioGroup>
```

Kode Program 4. 14 Desain Antar muka untuk entry data berdasarkan penyakit per kecamatan

Berikut ini adalah desain antar muka untuk entry data berdasarkan penyakit per kecamatan

Gambar 4. 11 Desain Antarmuka berdasarkan penyakit /kecamatan

2. Query SQL dan source code untuk menampilkan data berdarkan penyakit / kecamatan

Pada bagian ini dijelaskan mengenai pembuatan script PHP yang berisi perintah SQL. Ketika query dari sql dijalankan maka didapatkan data mengenai penyakit yang dipilih, bulan yang dipilih dan tahun yang dipilih.

Berikut adalah kode program untuk menampilkan query sesuai berdasarkan penyakit/kecamatan

```
$nama_penyakit=$_REQUEST['nama_penyakit'];
$bulan=$_REQUEST['bulan'];
$tahun=$_REQUEST['tahun'];
```

```
$q = mysql_query ("SELECT `kecamatan`,
COUNT(`nama_penyakit`) AS nama_penyakit FROM
kopmaits_latihan.peternakan WHERE
peternakan.nama_penyakit = '{$nama_penyakit}'
AND MONTH(peternakan.bulan)='{$bulan}' AND
YEAR(peternakan.bulan)='{$tahun}'
GROUP BY `nama_penyakit`, `kecamatan`
ORDER BY COUNT(`nama_penyakit`)");

$r = array();
$c = mysql_num_rows($q);
if($c>0){
    $r["peternakan"] = array();
    while($e = mysql_fetch_array($q)){
        $a = array();
        $a["nama_penyakit"]=
$e["nama_penyakit"];
        $a["kecamatan"]= $e["kecamatan"];

        array_push($r["peternakan"], $a);
    }
    $r["success"] = 1;
    // mengirimkan data JSON ke Android
    echo json_encode($r);
}else{
    $r["success"] = 0;
    echo json_encode($r);
}
mysql_close();
?>
```

Kode Program 4. 15 script PHP untuk menampilkan data berdasarkan penyakit /kecamatan

Kemudian dibuat script PHP untuk mengambil data dari *web server* , selanjutnya hasil tersebut ditampilkan ke sisi client dengan format JSON. Untuk itu dibutuhkan sebuah method yang digunakan untuk membaca hasil *array* yang nantinya ditampilkan dalam bentuk barchart . Berikut kode programnya

```
  protected String doInBackground(String... args) {
            List<BasicNameValuePair> params = new
ArrayList<BasicNameValuePair>();
            params.add(new
BasicNameValuePair("nama_penyakit", Penyakit));
            params.add(new
BasicNameValuePair("bulan", Bulan));
            params.add(new
BasicNameValuePair("tahun", Tahun));
            JSONObject json =
jParser.makeHttpRequest(base_url, "GET", params);
            Log.d("All Accounts: ", json.toString());
            try {
                int success =
json.getInt(TAG_SUCCESS);
                if (success == 1) {
                    JSONArray menuitemArray =
json.getJSONArray(TAG_ACCOUNT);
                    for (int i = 0; i <
menuitemArray.length(); i++) {


      nama_penyakit.add(Integer.parseInt((menuitemAr
ray.getJSONObject(i).getString("nama_penyakit"))));
                        nama_penyakitdashboard =
new Integer[nama_penyakit.size()];
                        nama_penyakitdashboard =
nama_penyakit.toArray(nama_penyakitdashboard);


      kecamatan.add((menuitemArray.getJSONObject(i).
getString("kecamatan")));
                        kecamatan_dashboard = new
String[kecamatan.size()];
                        kecamatan_dashboard =
kecamatan.toArray(kecamatan_dashboard);
```

```
                    }

                }
            } catch (JSONException e) {
                e.printStackTrace();
            }
            return "";
        }

        protected void onPostExecute(String file_url)
{
            pDialog.dismiss();

            //Hasil.setText(sret);

            Intent barchart=new
Intent().setClass(chart.this,bargraph.class);
            Bundle c = new Bundle();
                    c.putIntegerArrayList("hasil",
nama_penyakit);

       c.putStringArrayList("namakecamatan",
kecamatan);
                    barchart.putExtras(c);
                    startActivity(barchart);
                    finish();

            }
```

Kode Program 4. 16 Request data ke *web server*

3. Membuat barchart untuk dashboard berdasarkan penyakit / kecamatan

Pada kode program diatas hasil query yang diterima dan disimpan dalam array "hasil" dan "namakecamatan", selanjutnya dibuat kelas baru untuk membuat barchart. Berikut potongan kode

program untuk membuat barchart berdasarkan pernyakit / kecamatan.

```
public class bargraph extends Activity {
	ArrayList<Integer> y = new ArrayList<Integer>();
	Integer[] hasil;

	ArrayList<String> x = new ArrayList<String>();
	String[] hasil_kecamatan;



	 public void onCreate(Bundle savedInstanceState) {
		super.onCreate(savedInstanceState);

		Bundle c = getIntent().getExtras();
		y=c.getIntegerArrayList("hasil");
		x=c.getStringArrayList("namakecamatan");
		hasil = new Integer[y.size()];
	hasil = y.toArray(hasil);
	hasil_kecamatan = new String[x.size()];
	hasil_kecamatan = x.toArray(hasil_kecamatan);
				// Bar 1
				//int[] y = { 100, 80, 20, 250
};
				CategorySeries series = new
CategorySeries("Jumlah Penyakit per Kecamatan");
				for (int i = 0; i <
hasil.length; i++) {
					series.add("Bar " +
(i+1), hasil[i]);
				}


				XYMultipleSeriesDataset
dataset = new XYMultipleSeriesDataset();

	dataset.addSeries(series.toXYSeries());

				XYMultipleSeriesRenderer
mRenderer = new XYMultipleSeriesRenderer();
```

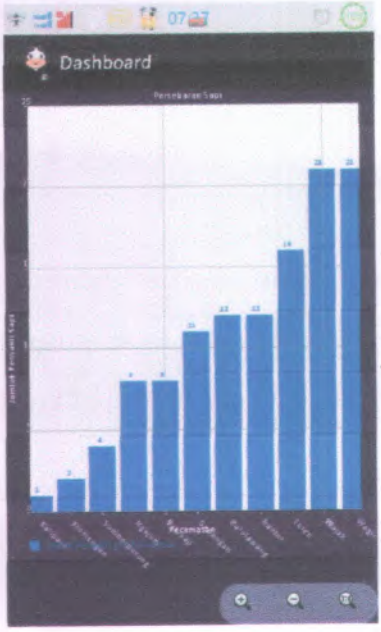
```
        mRenderer.setChartTitle("Persebaran Sapi");

        mRenderer.setXTitle("Kecamatan");
                                mRenderer.setYTitle("Jumlah
Penyakit Sapi");

        mRenderer.setAxesColor(Color.BLACK);

mRenderer.setLabelsColor(Color.WHITE);
                        mRenderer.setXAxisMin(0.5);
                        mRenderer.setYAxisMin(0);
                        mRenderer.setYAxisMax(25);
                        mRenderer.setBarSpacing(0.2);
                        mRenderer.setPanEnabled(true,
false);
                        mRenderer.setZoomEnabled(true);

mRenderer.setZoomButtonsVisible(true);
                        mRenderer.setZoomRate(1.1f);

mRenderer.setApplyBackgroundColor(true);

mRenderer.setBackgroundColor(Color.parseColor("#F5F5F5"));

mRenderer.setMarginsColor(Color.parseColor("#4E0263"));
                        mRenderer.setShowGrid(true);
                        mRenderer.setGridColor(Color.GRAY);
                        mRenderer.setLabelsTextSize(12);

mRenderer.setAxisTitleTextSize(12);

mRenderer.setChartTitleTextSize(12);



                        for (int abc = 0; abc <
hasil_kecamatan.length; abc++) {

mRenderer.addXTextLabel(abc+1, hasil_kecamatan[abc]);
```

```
mRenderer.setXLabelsAlign(Align.LEFT);

        mRenderer.setXLabelsAngle(45);
                                }

                            // Kustomisasi Bar 1
                                XYSeriesRenderer renderer =
new XYSeriesRenderer();


        renderer.setDisplayChartValues(true);

        renderer.setChartValuesSpacing((float)5.0);

        renderer.setChartValuesTextAlign(Align.RIGHT);
                                renderer.setColor(Color.BLUE);

        mRenderer.addSeriesRenderer(renderer);




                                Intent intent =
ChartFactory.getBarChartIntent(this, dataset,mRenderer,
Type.STACKED);
                                startActivity(intent);
        }


    }
```

Kode Program 4. 17 Menampilkan data berdasarkan penyakit / kecamatan dalam bentuk barchart

Pada kode program diatas berisi array "hasil" dan "namakecamatan" yang merupakan array dari jumlah dari nama_penyakit dan kecamatan. Untuk *x value* berdasarkan hasil array dari namakecamatan, kemudian *y value* berdasarkan hasil array dari hasil Untuk tampilan barchart disini berwarna "*blue*".

Hasil dapat dilihat pada gambar 4.8



Gambar 4. 12 Tampilan dashboard berdasarkan penyakit / kecamatan

4.6.4.4 Pembuatan Fitur Melihat Grafik Berdasarkan Penyakit / Desa

Pada bagian ini akan dijelaskan pembuatan fitur untuk melihat grafik berdasarkan penyakit / desa. Langkah-Langkah pembuatan akan dijelaskan sebagai berikut.

1. Membuat halaman untuk mengentry data berdasarkan penyakit per desa

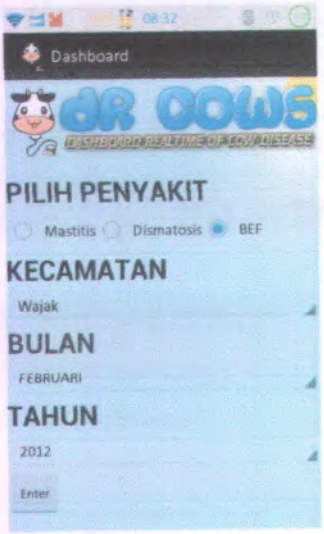
Berikut source code untuk halaman entry data berdasarkan penyakit per kecamatan, dimana terdapat komponen Spinner yang berisi jenis penyakit, nama kecamatan, nama bulan , dan tahun yang akan dipilih oleh dinas Peternakan dan Kesehatan hewan

```
TextView
        android:id="@+id/txt1"
        android:text="PILIH PENYAKIT"
        android:textSize="28dp"
        android:textStyle="bold"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center"/>

      <RadioGroup

android:id="@+id/txtNamaCariPenyakitDesa"
          android:orientation="horizontal"
          android:layout_width="fill_parent"
             android:layout_height="wrap_content"
             android:padding="2px">
                   <RadioButton

android:id="@+id/penyakitmastitis1"
                          android:text="Mastitis"

android:textColor="@color/black"/>
                    <RadioButton

android:id="@+id/penyakitdismatosis1"
                          android:text="Dismatosis"

android:textColor="@color/black"/>
                    <RadioButton

android:id="@+id/penyakitbef1"
                          android:text="BEF"

android:textColor="@color/black"/>
          </RadioGroup>

             <TextView
```

```
android:id="@+id/txt1"
android:text="KECAMATAN"
android:textSize="28dp"
android:textStyle="bold"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:gravity="center"/>
```

Kode Program 4. 18 Desain Antar muka untuk entry data berdasarkan penyakit per desa

Berikut ini adalah desain antar muka untuk entry data berdasarkan penyakit per desa



Gambar 4. 13 Desain Antarmuka berdasarkan penyakit /desa

2. Query SQL dan source code untuk menampilkan data berdarkan penyakit / desa

Pada bagian ini dijelaskan mengenai pembuatan script PHP yang berisi perintah SQL. Ketika query dari sql dijalankan maka

didapatkan data mengenai penyakit yang dipilih, bulan yang dipilih dan tahun yang dipilih.
Berikut ini adalah *script* PHP yang digunakan untuk memilih penyakit / desa.

```
$nama_penyakit=$_REQUEST['nama_penyakit'];
$bulan=$_REQUEST['bulan'];
$tahun=$_REQUEST['tahun'];
$kecamatan=$_REQUEST['kecamatan'];

$q = mysql_query ("SELECT `desa`,
COUNT(`nama_penyakit`) AS nama_penyakit FROM
kopmaits_latihan.peternakan WHERE
peternakan.nama_penyakit = '{$nama_penyakit}'
AND MONTH(peternakan.bulan)='{$bulan}' AND
YEAR(peternakan.bulan)='{$tahun}' AND
peternakan.kecamatan='{$kecamatan}'
GROUP BY `nama_penyakit`, `desa`
ORDER BY COUNT(`nama_penyakit`)");

$r = array();
$c = mysql_num_rows($q);
if($c>0){
    $r["peternakan"] = array();
    while($e = mysql_fetch_array($q)){
        $a = array();
        $a["nama_penyakit"]=
$e["nama_penyakit"];
        $a["desa"]= $e["desa"];

        array_push($r["peternakan"], $a);
    }
    $r["success"] = 1;
    // mengirimkan data JSON ke Android
    echo json_encode($r);
}else{
    $r["success"] = 0;
    echo json_encode($r);
}
```

```
mysql_close();
?>
```

Kode Program 4. 19 script PHP untuk menampilkan data berdasarkan penyakit /desa

Kemudian dibuat *script* PHP untuk mengambil data dari *web server*, selanjutnya hasil tersebut ditampilkan ke sisi client dengan format JSON. Untuk itu dibutuhkan sebuah method yang digunakan untuk membaca hasil *array* yang nantinya ditampilkan dalam bentuk barchart . Berikut kode programnya

```
protected String doInBackground(String... args) {
            List<BasicNameValuePair> params = new
ArrayList<BasicNameValuePair>();
            params.add(new
BasicNameValuePair("nama_penyakit", Penyakit));
            params.add(new
BasicNameValuePair("bulan", Bulan));
            params.add(new
BasicNameValuePair("tahun", Tahun));
            params.add(new
BasicNameValuePair("kecamatan", Kecamatan));
            JSONObject json =
jParser.makeHttpRequest(base_url, "GET", params);
            Log.d("All Accounts: ", json.toString());
            try {
                int success =
json.getInt(TAG_SUCCESS);
                if (success == 1) {
                    JSONArray menuitemArray =
json.getJSONArray(TAG_ACCOUNT);
                    for (int i = 0; i <
menuitemArray.length(); i++) {


      nama_penyakit.add(Integer.parseInt((menuitemAr
ray.getJSONObject(i).getString("nama_penyakit"))));
```

```
                            nama_penyakitdashboard =
new Integer[nama_penyakit.size()];
                            nama_penyakitdashboard =
nama_penyakit.toArray(nama_penyakitdashboard);


      desa.add((menuitemArray.getJSONObject(i).getSt
ring("desa")));
                            desa_dashboard = new
String[desa.size()];
                            desa_dashboard =
desa.toArray(desa_dashboard);


                    }

                }
            } catch (JSONException e) {
                e.printStackTrace();
            }
            return "";
        }

        protected void onPostExecute(String file_url)
{
            pDialog.dismiss();

            //Hasil.setText(sret);

            Intent barchartdesa=new
Intent().setClass(chartdesa.this,bargraphdesa.class);
            Bundle dc = new Bundle();
                    dc.putIntegerArrayList("hasil",
nama_penyakit);
                    dc.putStringArrayList("namadesa",
desa);
                    barchartdesa.putExtras(dc);
                    startActivity(barchartdesa);
```

```
                finish();

            }

                }
```

Kode Program 4. 20 Request data ke *web server*

3. Membuat barchart untuk dashboard berdasarkan penyakit / desa

Pada source diatas hasil query yang diterima dan disimpan dalam array "hasil" dan "namadesa", selanjutnya dibuat kelas baru untuk membuat barchart. Berikut potongan code untuk membuat barchart berdasarkan penyakit / desa

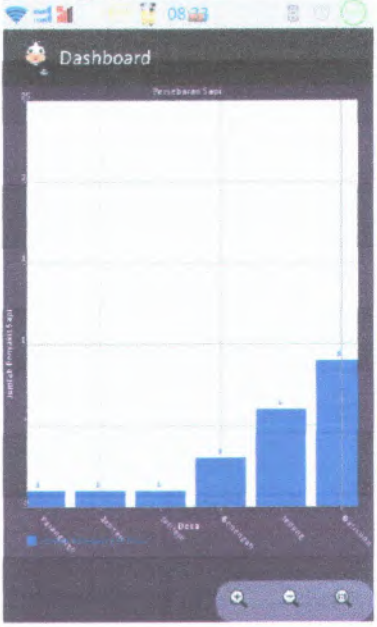
```
public void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);

          Bundle dc = getIntent().getExtras();
          y=dc.getIntegerArrayList("hasil");
          x=dc.getStringArrayList("namadesa");
         hasil = new Integer[y.size()];
      hasil = y.toArray(hasil);
      hasil_desa = new String[x.size()];
      hasil_desa = x.toArray(hasil_desa);

};
                        CategorySeries series = new
CategorySeries("Jumlah Penyakit per Desa");
                        for (int i = 0; i <
hasil.length; i++) {
                                series.add("Bar " +
(i+1), hasil[i]);
                        }
```

```
                        XYMultipleSeriesDataset dataset
= new XYMultipleSeriesDataset();

      dataset.addSeries(series.toXYSeries());

                        XYMultipleSeriesRenderer
mRenderer = new XYMultipleSeriesRenderer();

      mRenderer.setChartTitle("Persebaran Sapi");
                        mRenderer.setXTitle("Desa");
                        mRenderer.setYTitle("Jumlah
Penyakit Sapi");

      mRenderer.setAxesColor(Color.BLACK);

mRenderer.setLabelsColor(Color.WHITE);
                    mRenderer.setXAxisMin(0.5);
                     mRenderer.setYAxisMin(0);
                     mRenderer.setYAxisMax(25);
                     mRenderer.setBarSpacing(0.2);
                     mRenderer.setPanEnabled(true,
false);
                     mRenderer.setZoomEnabled(true);

mRenderer.setZoomButtonsVisible(true);
                     mRenderer.setZoomRate(1.1f);

mRenderer.setApplyBackgroundColor(true);

mRenderer.setBackgroundColor(Color.parseColor("#F5F5F5"));

mRenderer.setMarginsColor(Color.parseColor("#4E0263"));
                     mRenderer.setShowGrid(true);
                     mRenderer.setGridColor(Color.GRAY);
                     mRenderer.setLabelsTextSize(12);

mRenderer.setAxisTitleTextSize(12);
```

```
mRenderer.setChartTitleTextSize(12);



                    for (int abc = 0; abc <
hasil_desa.length; abc++) {
                        mRenderer.addXTextLabel(abc+1,
hasil_desa[abc]);

mRenderer.setXLabelsAlign(Align.LEFT);

      mRenderer.setXLabelsAngle(45);
                        }


                        XYSeriesRenderer renderer = new
XYSeriesRenderer();


      renderer.setDisplayChartValues(true);

      renderer.setChartValuesSpacing((float)5.0);

      renderer.setChartValuesTextAlign(Align.RIGHT);
                        renderer.setColor(Color.BLUE);

      mRenderer.addSeriesRenderer(renderer);



                        Intent intent =
ChartFactory.getBarChartIntent(this, dataset,mRenderer,
Type.STACKED);
                        startActivity(intent);
      }

    }
```

Kode Program 4. 21 Menampilkan data berdasarkan penyakit / desa dalam bentuk barchart

Pada *source code* diatas berisi array "hasil" dan "namadesa" yang merupakan array dari jumlah dari nama_penyakit dan desa. Untuk x value berdasarkan hasil array dari namadesa, kemudian y value berdasarkan hasil array dari hasil Untuk tampilan barchart disini berwarna "*blue*". Hasilnya dapat ditampilkan sepeerti gambar 4.10



Gambar 4. 14 Tampilan dashboard berdasarkan penyakit / desa

4.6.4.5 Pembuatan Fitur Melihat Grafik Berdasarkan Penanganan / Kecamatan

Pada bagian ini akan dijelaskan pembuatan fitur untuk melihat grafik berdasarkan penanganan / kecamatan Langkah-Langkah pembuatan akan dijelaskan sebagai berikut.

1. Membuat halaman untuk mengentry data berdasarkan penanganan per kecamatan

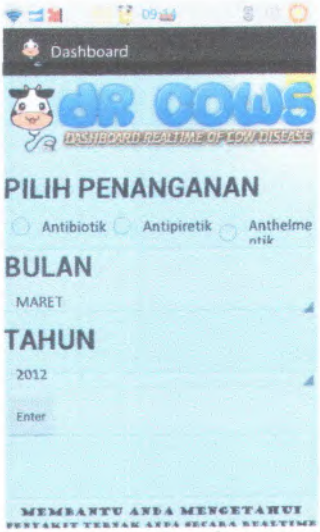
Berikut kode program untuk halaman entry data berdasarkan penanganan per kecamatan, dimana terdapat komponen Spinner yang berisi jenis penanganan, nama bulan , dan tahun yang akan dipilih oleh dinas Peternakan dan Kesehatan hewan

```
<TextView
        android:id="@+id/txt1"
        android:text="PILIH PENANGANAN"
        android:textSize="28dp"
        android:textStyle="bold"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center"/>

      <RadioGroup
          android:id="@+id/txtNamaCariObat"
          android:orientation="horizontal"
          android:layout_width="fill_parent"
             android:layout_height="wrap_content"
             android:padding="2px">
                   <RadioButton

android:id="@+id/obatAntibiotik"
                          android:text="Antibiotik"

android:textColor="@color/black"/>
                   <RadioButton

android:id="@+id/obatAntipiretik"
                          android:text="Antipiretik"

android:textColor="@color/black"/>
                   <RadioButton

android:id="@+id/obatAnthelmentik"
```

```
android:text="Anthelmentik"

android:textColor="@color/black"/>
        </RadioGroup>
```

Kode Program 4. 22 Desain Antar muka untuk entry data berdasarkan penanganan per kecamatan

Berikut ini adalah desain antar muka untuk entry data berdasarkan penanganan per kecamatan



Gambar 4. 15 Desain Antarmuka berdasarkan penyakit /kecamatan

2. Query SQL dan kode program untuk menampilkan data berdasarkan penanganan / kecamatan

Pada bagian ini dijelaskan mengenai pembuatan script PHP yang berisi perintah SQL. Ketika query dari sql dijalankan maka didapatkan data mengenai penanganan yang dipilih, bulan yang

dipilih dan tahun yang dipilih. Berikut *script* PHP yang digunakan untuk menampilkan penanganan / kecamatan

```
$obat=$_GET['obat'];
$bulan=$_GET['bulan'];
$tahun=$_GET['tahun'];

$q = mysql_query ("SELECT `kecamatan`,
COUNT(`obat`) AS obat FROM
kopmaits_latihan.peternakan WHERE
peternakan.obat = '{$obat}'
AND MONTH(peternakan.bulan)='{$bulan}' AND
YEAR(peternakan.bulan)='{$tahun}'
GROUP BY `obat`, `kecamatan` ");


$r = array();
$c = mysql_num_rows($q);
if($c>0){
    $r["peternakan"] = array();
    while($e = mysql_fetch_array($q)){
        $a = array();
        $a["obat"]= $e["obat"];
        $a["kecamatan"]= $e["kecamatan"];

        array_push($r["peternakan"], $a);
    }
    $r["success"] = 1;
    // mengirimkan data JSON ke Android
    echo json_encode($r);
}else{
    $r["success"] = 0;
    echo json_encode($r);
}
mysql_close();
?>
```

Kode Program 4. 23 script PHP untuk menampilkan data berdasarkan penanganan /kecamatan

Kemudian dibuat *script* PHP untuk mengambil data dari *web server* , selanjutnya hasil tersebut ditampilkan ke sisi client dengan format JSON. Untuk itu dibutuhkan sebuah method yang digunakan untuk membaca hasil *array* yang nantinya ditampilkan dalam bentuk barchart . Berikut kode programnya

```
  protected String doInBackground(String... args) {
            List<BasicNameValuePair> params = new
ArrayList<BasicNameValuePair>();
            params.add(new BasicNameValuePair("obat",
Obat));
            params.add(new
BasicNameValuePair("bulan", Bulan));
            params.add(new
BasicNameValuePair("tahun", Tahun));
            JSONObject json =
jParser.makeHttpRequest(base_url, "GET", params);
            Log.d("All Accounts: ", json.toString());
            try {
                int success =
json.getInt(TAG_SUCCESS);
                if (success == 1) {
                    JSONArray menuitemArray =
json.getJSONArray(TAG_ACCOUNT);
                    for (int ax = 0; ax <
menuitemArray.length(); ax++) {


obat_penyakit.add(Integer.parseInt((menuitemArray.get
JSONObject(ax).getString("obat"))));
                        obat_penyakitdashboard =
new Integer[obat_penyakit.size()];
                        obat_penyakitdashboard =
obat_penyakit.toArray(obat_penyakitdashboard);

      obat_kecamatan.add((menuitemArray.getJSONObjec
t(ax).getString("kecamatan")));
```

```
                                obatkecamatan_dashboard =
new String[obat_kecamatan.size()];
                                obatkecamatan_dashboard =
obat_kecamatan.toArray( obatkecamatan_dashboard);


                        }

                    }
                } catch (JSONException e) {
                    e.printStackTrace();
                }
                return null;
            }

            protected void onPostExecute(String file_url)
{
                pDialog.dismiss();

                //Hasil.setText(sret);

                Intent obatchart=new
Intent().setClass(chartobat.this,obatchart.class);
                Bundle obat2 = new Bundle();

          obat2.putIntegerArrayList("hasil_obat",
obat_penyakit);

          obat2.putStringArrayList("namakecamatan",
obat_kecamatan);
                        obatchart.putExtras(obat2);
                        startActivity(obatchart);
                        finish();
                }
                    }
```

Kode Program 4. 24 Request data ke *web server*

3. Membuat barchart untuk dashboard berdasarkan penanganan / kecamatan

Pada source code diatas hasil query yang diterima dan disimpan dalam array "hasil_obat" dan "namakecamatan", selanjutnya dibuat kelas baru untuk membuat barchart. Berikut potongan code untuk membuat barchart berdasarkan penganan / kecamatan.

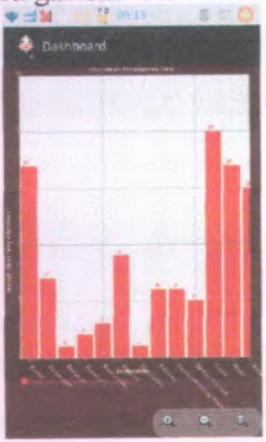
```
ArrayList<Integer> obt = new ArrayList<Integer>();
      Integer[] hasilobt;

      ArrayList<String> nmkcmtn = new
ArrayList<String>();
      String[] hasil_kcmtn;

       public void onCreate(Bundle savedInstanceState) {
             super.onCreate(savedInstanceState);

          Bundle obat2 = getIntent().getExtras();
          obt=obat2.getIntegerArrayList("hasil_obat");

nmkcmtn=obat2.getStringArrayList("namakecamatan");
          hasilobt = new Integer[obt.size()];
      hasilobt = obt.toArray(hasilobt);
      hasil_kcmtn = new String[nmkcmtn.size()];
      hasil_kcmtn = nmkcmtn.toArray(hasil_kcmtn);

                          CategorySeries series = new
CategorySeries("Obat yang Diberikan per Kecamatan");
                          for (int o = 0; o <
hasilobt.length; o++) {
                                series.add("Bar " +
(o+1), hasilobt[o]);
                          }


                          XYMultipleSeriesDataset dataset
```

```
= new XYMultipleSeriesDataset();

        dataset.addSeries(series.toXYSeries());

                                XYMultipleSeriesRenderer
mRenderer = new XYMultipleSeriesRenderer();

        mRenderer.setChartTitle("Persebaran Penanganan
Obat");

        mRenderer.setXTitle("Kecamatan");
                                mRenderer.setYTitle("Jumlah
Obat yang diberikan");

        mRenderer.setAxesColor(Color.WHITE);

mRenderer.setLabelsColor(Color.WHITE);
                                mRenderer.setXAxisMin(0.5);
                        mRenderer.setYAxisMin(0);
                        mRenderer.setYAxisMax(25);
                        mRenderer.setBarSpacing(0.2);

                        mRenderer.setPanEnabled(true,
false);
                        mRenderer.setZoomEnabled(true);

mRenderer.setZoomButtonsVisible(true);
                        mRenderer.setZoomRate(1.1f);



mRenderer.setApplyBackgroundColor(true);

mRenderer.setBackgroundColor(Color.parseColor("#F5F5F5"));

mRenderer.setMarginsColor(Color.parseColor("#6B1B02"));
                        mRenderer.setShowGrid(true);
                        mRenderer.setGridColor(Color.GRAY);
```

```
                    mRenderer.setLabelsTextSize(12);
                    mRenderer.setAxisTitleTextSize(12);
                   mRenderer.setChartTitleTextSize(12);

                     for (int kc = 0; kc <
hasil_kcmtn.length; kc++) {
                          mRenderer.addXTextLabel(kc+1,
hasil_kcmtn[kc]);

mRenderer.setXLabelsAlign(Align.LEFT);

      mRenderer.setXLabelsAngle(45);

                          }


                          XYSeriesRenderer renderer = new
XYSeriesRenderer();

      renderer.setDisplayChartValues(true);

      renderer.setChartValuesSpacing((float)5.0);

      renderer.setChartValuesTextAlign(Align.RIGHT);
                          renderer.setColor(Color.RED);

      mRenderer.addSeriesRenderer(renderer);


                          Intent intent =
ChartFactory.getBarChartIntent(this, dataset,mRenderer,
Type.STACKED);
                          startActivity(intent);
      }
      }
```

Kode Program 4. 25 Menampilkan data berdasarkan penanganan / kecamatan dalam bentuk barchart

Pada *source code* diatas berisi array "hasil_obat" dan "hasil_kcmtn" yang merupakan array dari jumlah dari nama_obat dan kecamatan. Untuk *x value* berdasarkan hasil array dari namakecamatan, kemudian *y value* berdasarkan hasil array dari hasil_obat. Untuk tampilan barchart disini berwarna "*red*". Hasilnya ditampilkan sperti gambar 4.12



Gambar 4. 16 Tampilan dashboard berdasarkan penanganan / kecamatan

4.6.4.6 Pembuatan Fitur Melihat Grafik Berdasarkan Penanganan / Desa

Pada bagian ini akan dijelaskan pembuatan fitur melihat grafik berdasarkan penanganan / desa. Langkah-Langkah pembuatan akan dijelaskan sebagai berikut.

1. **Membuat halaman untuk mengentry data berdasarkan penanganan per desa**

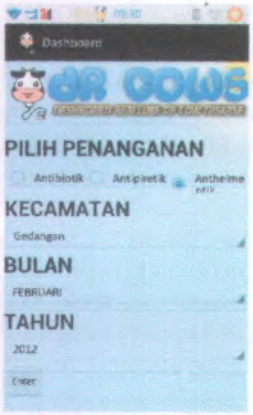
Berikut kode program untuk halaman entry data berdasarkan penanganan per kecamatan, dimana terdapat komponen Spinner yang berisi jenis penanganan,kecamatan nama bulan , dan tahun yang akan dipilih oleh dinas Peternakan

dan Kesehatan hewan. Berikut adalah potongan kode untuk mendesain halaman antar muka penanganan /desa

```
          <TextView
         android:id="@+id/txt1"
         android:text="PILIH PENANGANAN"
         android:textSize="28dp"
         android:textStyle="bold"
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:gravity="center"/>

       <RadioGroup
           android:id="@+id/txtNamaCariDesaObat"
           android:orientation="horizontal"
           android:layout_width="fill_parent"
              android:layout_height="wrap_content"
              android:padding="2px">
                    <RadioButton

android:id="@+id/obatAntibiotik1"
                          android:text="Antibiotik"

android:textColor="@color/black"/>
                     <RadioButton

android:id="@+id/obatAntipiretik1"
                          android:text="Antipiretik"

android:textColor="@color/black"/>
                     <RadioButton

android:id="@+id/obatAnthelmentik1"

android:text="Anthelmentik"

android:textColor="@color/black"/>
          </RadioGroup>
```

Kode Program 4. 26 Desain Antar muka untuk entry data berdasarkan penanganan per desa

Berikut ini adalah desain antar muka untuk entry data berdasarkan penanganan per desa



Kode Program 4. 27 Desain Antarmuka berdasarkan penanganan / desa

2. Query SQL dan source code untuk menampilkan data berdarkan penanganan / desa

Pada bagian ini dijelaskan mengenai pembuatan script PHP yang berisi perintah SQL. Ketika query dari sql dijalankan maka didapatkan data mengenai penanganan yang dipilih, kecamatan, bulan yang dipilih dan tahun yang dipilih. Berikut adalah *script* PHP untuk menampilkan data berdasarkan penanganan /desa

```
$obat=$_GET['obat'];
$bulan=$_GET['bulan'];
$tahun=$_GET['tahun'];
$kecamatan=$_GET['kecamatan'];

$q = mysql_query ("SELECT `desa`,
COUNT(`obat`) AS obat FROM
```

```
kopmaits_latihan.peternakan WHERE
peternakan.obat = '{$obat}' AND
peternakan.kecamatan = '{$kecamatan}'
AND MONTH(peternakan.bulan)='{$bulan}' AND
YEAR(peternakan.bulan)='{$tahun}'
GROUP BY `obat`, `desa` ");
$r = array();
$c = mysql_num_rows($q);
if($c>0){
    $r["peternakan"] = array();
    while($e = mysql_fetch_array($q)){
        $a = array();
        $a["obat"]= $e["obat"];
        $a["desa"]= $e["desa"];

        array_push($r["peternakan"], $a);
    }
    $r["success"] = 1;
    // mengirimkan data JSON ke Android
    echo json_encode($r);
}else{
    $r["success"] = 0;
    echo json_encode($r);
}
mysql_close();
?>
```

Kode Program 4. 28 script PHP untuk menampilkan data berdasarkan penanganan /desa

Kemudian dibuat script PHP untuk mengambil data dari *web server* , selanjutnya hasil tersebut ditampilkan ke sisi client dengan format JSON. Untuk itu dibutuhkan sebuah method yang digunakan untuk membaca hasil *array* yang nantinya ditampilkan dalam bentuk barchart . Berikut kode programnya

```
protected String doInBackground(String... args) {
            List<BasicNameValuePair> params = new
ArrayList<BasicNameValuePair>();
```

```
            params.add(new
BasicNameValuePair("nama_penyakit", Penyakit));
            params.add(new
BasicNameValuePair("bulan", Bulan));
            params.add(new
BasicNameValuePair("tahun", Tahun));
            params.add(new
BasicNameValuePair("kecamatan", Kecamatan));
            JSONObject json =
jParser.makeHttpRequest(base_url, "GET", params);
            Log.d("All Accounts: ", json.toString());
            try {
                int success =
json.getInt(TAG_SUCCESS);
                if (success == 1) {
                    JSONArray menuitemArray =
json.getJSONArray(TAG_ACCOUNT);
                    for (int i = 0; i <
menuitemArray.length(); i++) {


      nama_penyakit.add(Integer.parseInt((menuitemAr
ray.getJSONObject(i).getString("nama_penyakit"))));
                        nama_penyakitdashboard =
new Integer[nama_penyakit.size()];
                        nama_penyakitdashboard =
nama_penyakit.toArray(nama_penyakitdashboard);


      desa.add((menuitemArray.getJSONObject(i).getSt
ring("desa")));
                        desa_dashboard = new
String[desa.size()];
                        desa_dashboard =
desa.toArray(desa_dashboard);

                    }
```

```
                }
            } catch (JSONException e) {
                e.printStackTrace();
            }
            return "";
        }

        protected void onPostExecute(String file_url)
{
            pDialog.dismiss();

            //Hasil.setText(sret);

            Intent barchartdesa=new
Intent().setClass(chartdesa.this,bargraphdesa.class);
            Bundle dc = new Bundle();
                    dc.putIntegerArrayList("hasil",
nama_penyakit);
                    dc.putStringArrayList("namadesa",
desa);
                    barchartdesa.putExtras(dc);
                    startActivity(barchartdesa);
                    finish();

            }

                }
```

Kode Program 4. 29 Request data ke *web server*

3. Membuat barchart untuk dashboard berdasarkan penanganan / desa

Pada kode program diatas hasil query yang diterima dan disimpan dalam array "hasil_obat" dan "namadesa", selanjutnya dibuat kelas baru untuk membuat barchart. Berikut potongan code untuk membuat barchart berdasarkan penanganan / desa

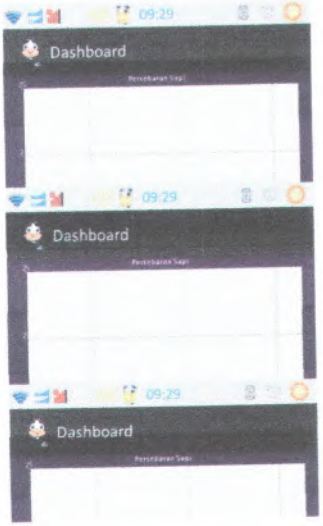
```
  Bundle dc = getIntent().getExtras();
            y=dc.getIntegerArrayList("hasil");
            x=dc.getStringArrayList("namadesa");
           hasil = new Integer[y.size()];
       hasil = y.toArray(hasil);
       hasil_desa = new String[x.size()];
       hasil_desa = x.toArray(hasil_desa);

                          CategorySeries series = new
CategorySeries("Jumlah Penyakit per Desa");
                          for (int i = 0; i <
hasil.length; i++) {
                                  series.add("Bar " +
(i+1), hasil[i]);
                          }



                          XYMultipleSeriesDataset dataset
= new XYMultipleSeriesDataset();

       dataset.addSeries(series.toXYSeries());

                          XYMultipleSeriesRenderer
mRenderer = new XYMultipleSeriesRenderer();

       mRenderer.setChartTitle("Persebaran Sapi");
                          mRenderer.setXTitle("Desa");
                          mRenderer.setYTitle("Jumlah
Penyakit Sapi");

       mRenderer.setAxesColor(Color.BLACK);

mRenderer.setLabelsColor(Color.WHITE);
                     mRenderer.setXAxisMin(0.5);
                      mRenderer.setYAxisMin(0);
                      mRenderer.setYAxisMax(25);
```

```
                    mRenderer.setBarSpacing(0.2);
                    mRenderer.setPanEnabled(true,
false);
                    mRenderer.setZoomEnabled(true);

mRenderer.setZoomButtonsVisible(true);
                    mRenderer.setZoomRate(1.1f);

mRenderer.setApplyBackgroundColor(true);

mRenderer.setBackgroundColor(Color.parseColor("#F5F5F5"));

mRenderer.setMarginsColor(Color.parseColor("#4E0263"));
                    mRenderer.setShowGrid(true);
                    mRenderer.setGridColor(Color.GRAY);
                    mRenderer.setLabelsTextSize(12);

mRenderer.setAxisTitleTextSize(12);

mRenderer.setChartTitleTextSize(12);



                    for (int abc = 0; abc <
hasil_desa.length; abc++) {
                        mRenderer.addXTextLabel(abc+1,
hasil_desa[abc]);

mRenderer.setXLabelsAlign(Align.LEFT);

      mRenderer.setXLabelsAngle(45);
                        }


                        XYSeriesRenderer renderer = new
XYSeriesRenderer();
```

```
        renderer.setDisplayChartValues(true);

        renderer.setChartValuesSpacing((float)5.0);

        renderer.setChartValuesTextAlign(Align.RIGHT);
                                renderer.setColor(Color.BLUE);

        mRenderer.addSeriesRenderer(renderer);



                                Intent intent =
ChartFactory.getBarChartIntent(this, dataset,mRenderer,
Type.STACKED);
                                startActivity(intent);
        }


    }
```

Kode Program 4. 30 Menampilkan data berdasarkan penanganan / desa dalam bentuk barchart

Pada kode program diatas berisi array "`hasil_obat`" dan "`hasil_desa`" yang merupakan array dari jumlah dari nama_obat dan desa. Untuk *x value* berdasarkan hasil array dari namadesa, kemudian *y value* berdasarkan hasil array dari hasil_obat. Untuk tampilan barchart disini berwarna "*blue*". Hasilnya ditampilkan seperti gambar 4.13



Gambar 4. 17 Tampilan dashboard berdasarkan penanganan / desa

BAB 5
UJI COBA DAN ANALISIS HASIL

Sebelum dilakukan uji coba, beberapa tahapan yang harus dilakukan terlebih dahulu yaitu validasi dan verifikasi sistem. Hal ini dilakukan untuk memastikan bahwa sistem yang dibuat sudah benar dan sesuai dengan kebutuhan.

5.1 Validasi dan Verifikasi Sistem

Validasi sistem dilakukan oleh pakar dari Dinas peternakan dan verifikasi dilakukan dengan menggunakan *test case* berdasarkan use case dan uji coba menggunakan jenis smartphone android.

5.1.1 Uji Coba Aplikasi dengan *Test Case*

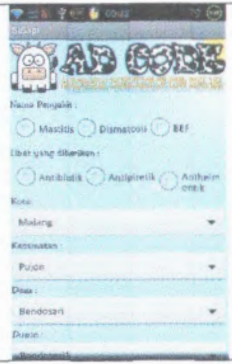
Proses pengujian dilakukan sesuai dengan use case yang dimiliki oleh aplikasi. Disini dibedakan antara antara use case yang dimiliki oleh petugas kecamatan untuk peternak dengan use case Aplikasi *Strategic Mobile Dashboard* yang digunakan oleh Dinas Peternakan.

5.1.1.1 Uji Coba Test Case Sisapi

Proses Pengujian *test case* dilakukan pada aplikasi Sisapi, sesuai dengan use case yang Sisapi yang dibahas pada bab sebelumnya. Berikut detailnya

Tabel 5. 1 Test Case Sisapi

No	Kode use case	Nama use case	Hasil yang diharapkan	Uji hasil dengan Sistem	Keterangan
1	UC-001	Use case Petugas Kecamatan log in	Petugas kecamatan melakukan log in sebelum melakukan insert data		Sesuai

2	UC-002	Use case Petugas kecamatan melakukan insert data	Petugas kecamatan melakukan insert data berdasarkan jenis penyakit , jenis obat, kota, kecamatan,desam dusun, dan posisi kordinat			Sesuai

3	UC-003	Use case pertugas kecamatan melakukan log out	Petugas kecamatan melakukan log out ketika sesudah menginputkan data			Sesuai

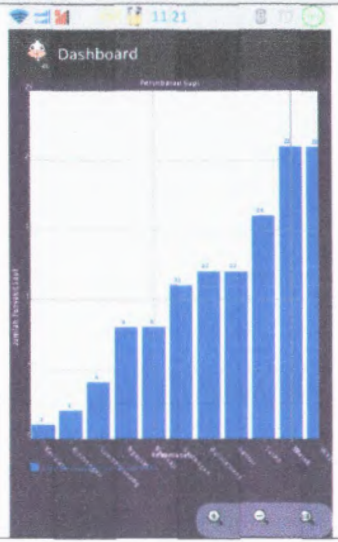
5.1.1.2 Uji Coba *Test Case Strategic Mobile Dashboard*

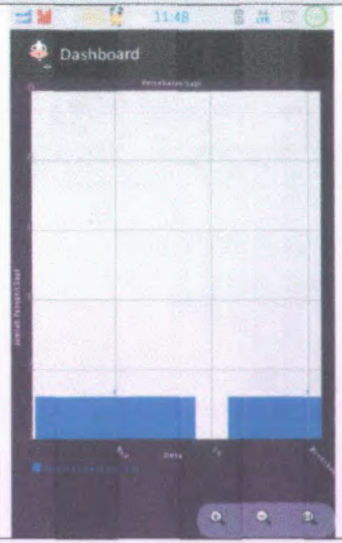
Proses Pengujian *test case* pada aplikasi Strategic Mobile Dashboard sesuai dengan use case pada aplikasi *strategic mobile dashboard.* Hasilnya dapat ditampilkan pada tabel 5.2

Tabel 5. 2 Test case Strategic Mobile Dashboard

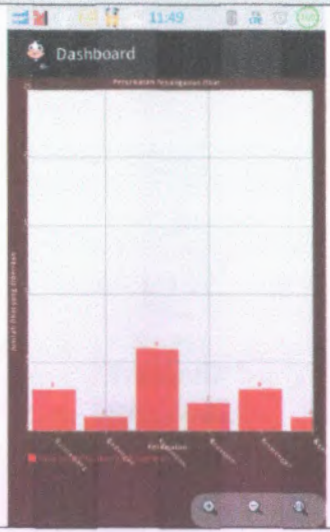
No	Kode use case	Nama use case	Hasil yang diharapkan	Uji hasil dengan Sistem	Keterangan
1	UC-001	Use case dapat memilih 2 jenis dashboard	Dinas Peternakan dapat memilih 2 jenis dashboard yaitu berdasarkan penyakit atau penangangan		Sesuai

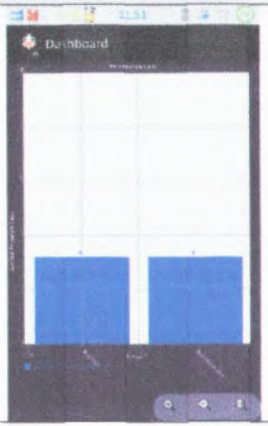
No	Kode use case	Nama use case	Hasil yang diharapkan	Uji hasil dengan Sistem	Keterangan
2	UC-002	Use case memilih dashboard berdasarkan jenis penyakit	Dinas Peternakan memilih dashboard berdasarkan jenis penyakit		Sesuai

No	Kode use case	Nama use case	Hasil yang diharapkan	Uji hasil dengan Sistem	Keterangan
3	UC-003	Use case melihat grafik berdasarkan penyakit / kecamatan	Dinas Peternakan melihat grafik berdasarkan penyakit/ kecamatan		Sesuai

No	Kode use case	Nama use case	Hasil yang diharapkan	Uji hasil dengan Sistem	Keterangan
4	UC-004	Use case melihat grafik berdasarkan penyakit / desa	Dinas peternakan dapat melihat grafik berdasarkan penyakit/ desa		Sesuai

No	Kode use case	Nama use case	Hasil yang diharapkan	Uji hasil dengan Sistem	Keterangan
5	UC-005	Use case memilih dashboard berdasarkan jenis penanganan	Dinas perternakan dapat memilih dashboard berdasarkan jenis penanganan		Sesuai

No	Kode use case	Nama use case	Hasil yang diharapkan	Uji hasil dengan Sistem	Keterangan
6	UC-006	Use case melihat grafik berdasarkan penanganan / kecamatan	Dinas Peternakan dapat melihat grafik berdasarkan penanganan / kecamatan		Sesuai

No	Kode use case	Nama use case	Hasil yang diharapkan	Uji hasil dengan Sistem	Keterangan
7	UC-007	Use case melihat grafik berdasarkan penanganan / desa	Use case melihat grafik berdasarkan penanganan / desa		Sesuai

5.1.2 Verifikasi Aplikasi dengan Berbagai Jenis Smartphone Android

Pada tahap ini dilakukan verifikasi aplikasi dengan cara menjalankan aplikasi di beberapa smartphone android dengan ukuran layar dan operating system yang berbeda. Berikut ini daftar beberapa Smartphone yang menjadi objek uji coba

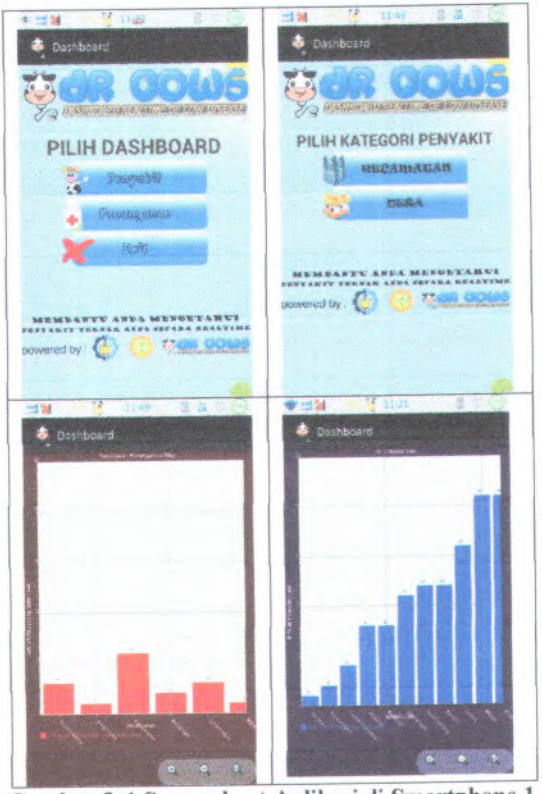
a. Smartfren Andromax I

Spesifikasi dari Smartphone yang digunakan untuk uji coba dapat dilihat pada tabel 5.3

Tabel 5. 3 Spesifikasi Smartphone 1

Sistem Operasi		Android 4.0.4 (Ice Cream Sandwich)
Jaringan		Dual mode (GSM & CDMA EVDO Rev -A)
Prosesor		Dual-core Snapdragon 1 Ghz
Layar Utama		4-inch LCD WVGA IPS technology, multitouchscreen, 5 finger touch point.
Internet		Browser : HTML; Zoom; Multitasking
Fitur Lain		Build in GPS

Screenshoot aplikasi yang dijalankan pada smartphone dapat dilihat di gambar 5.1



Gambar 5. 1 Screenshoot Aplikasi di Smartphone 1

Dari gambar 5.1 Aplikasi *strategic mobile dashboard* dapat berjalan di sistem operasi *Ice Cream Sandwich* (Android 4.0) . Maka dapat dianalisa bahwa aplikasi telah sesuai dan dapat berjalan di sistem operasi yang berbeda.

b. LG Google Nexus 4

Spesifikasi dari Smartphone kedua yang digunakan untuk uji coba dapat dilihat pada tabel 5.4

Tabel 5. 4 Spesifikasi Smarthphone 2

Sistem Operasi		Android 4.2 (Jelly Bean)
Jaringan		GSM/EDGE/GPRS (850, 900, 1800,

		1900 MHz)
Prosesor		Qualcomm Snapdragon S4 Pro CPU
Layar Utama		4,7"diagonal; 1280 x 768 pixel resolution (320 ppi); WXGA IPS; Corning Gorilla Glass 2
Internet		Browser : HTML; Zoom; Multitasking
Fitur Lain		GPS , Accelerometer, dll

Screenshoot aplikasi yang dijalankan pada smartphone dapat dilihat pada gambar 5.2

Gambar 5. 2 Screenshoot Aplikasi di Smartphone 2

Dari gambar 5.2 Aplikasi *strategic mobile dashboard* dapat berjalan di sistem operasi *Jely bean.* Maka dapat dianalisa bahwa

aplikasi telah sesuai dan dapat berjalan di sistem operasi yang berbeda.

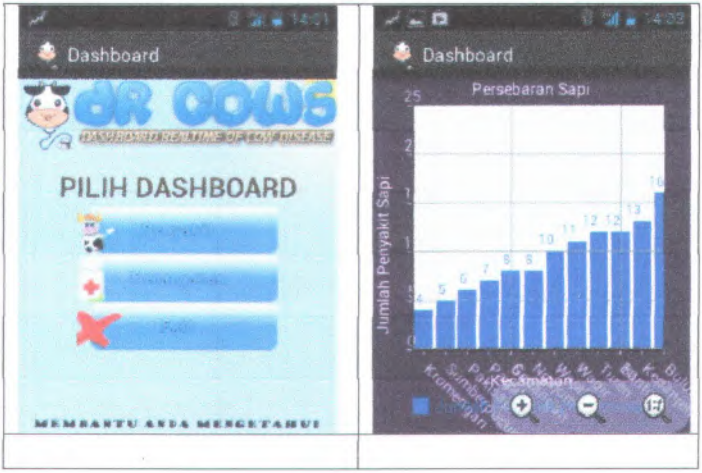
c. Samsung Galaxy Mini S5570

Spesifikasi dari Smartphone ketiga yang digunakan untuk uji coba dapat dilihat pada tabel 5.5

Tabel 5. 5 Spesifikasi Smarthphone 3

Sistem Operasi		Android 2.2 (Froyo)
Jaringan		GSM 850 / 900 / 1800 / 1900
Prosesor		Qualcomm MSM7227
Layar Utama		TFT capacitive touchscreen, 256K colors Size 240 x 320 pixels, 3.14 inches (~127 ppi pixel density)
Internet		Browser : HTML; Zoom;
Fitur Lain		GPS , Accelerometer, dll

Screenshoot aplikasi yang dijalankan pada smartphone diatas pada gambar 5.3



Gambar 5. 3 Screenshoot Aplikasi di Smartphone 3

Gambar 5.3 menujukkan aplikasi *strategic mobile dashboard* dapat berjalan di sistem operasi *froyo* (Android 2.2) .

Dari ketiga jenis OS Android yang diujikan, aplikasi mobile strategic dashboard lebih optimal dijalankan pada OS Jelly bean dengan layar sebesar 4 inch.

5.1.3 Validasi dengan Sistem Pakar

Setelah dilakukan verifikasi berdasarkan test case dan uji coba dengan berbagai macam smartphone maka dilakukan validasi oleh sistem pakar disini langsung kepada pihak dinas peternakan. Untuk lebih jelasnya dapat dilihat pada lampiran C

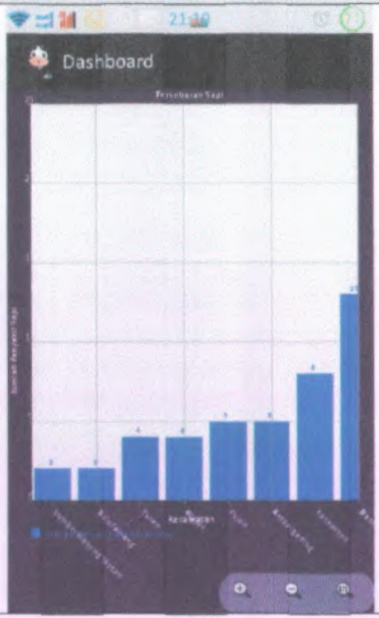
5.2 Analisis Hasil Sistem

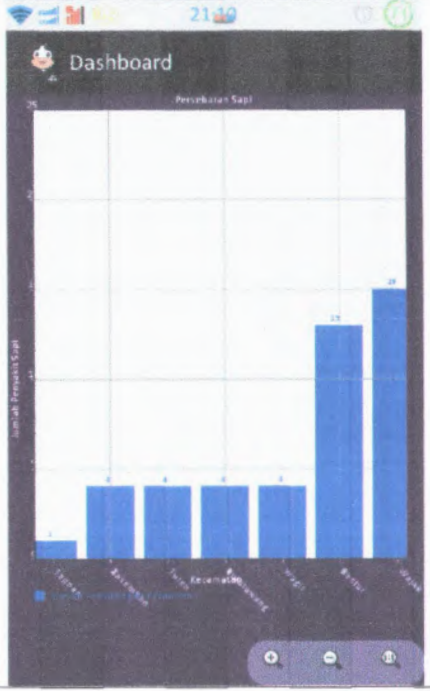
Analisis terhadap output yang dihasilkan oleh aplikasi mobile *strategic mobile dashboard* digunakan untuk mendapatkan data persebaran penyakit sekaligus penanganan pada setiap kecamatan maupun desa. Untuk menjalankan aplikasi ini ada beberapa tahapan yang harus dilakukan. Untuk lebih jelasnya pada lampiran A

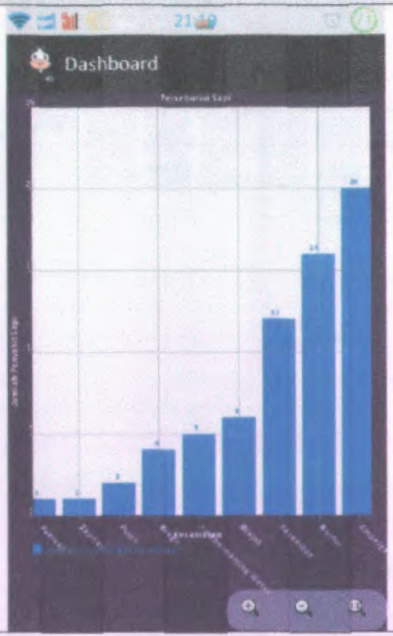
5.2.1 Analisis Persebaran Penyakit Sapi Berdasarkan Jenis Penyakit

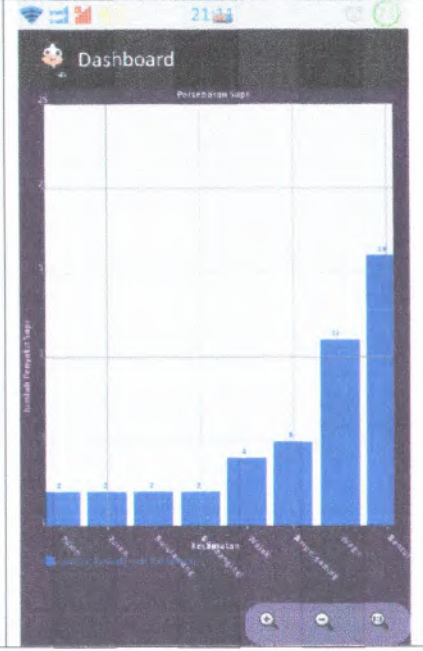
Analisis persebaran penyakit sapi dapat dilihat di setiap kecamatan yang ada di malang pada setiap bulan. Fitur ini menganalis penyakit tertentu, tergantung dari pilihan user pada bulan tertentu dan tahun tertentu. Nantinya dengan fitur ini, akan disajikan informasi mengenai persebaran penyakit pada tiap kecamatan. Hasil analisis lebih detail dapat dilihat pada tabel 5.6

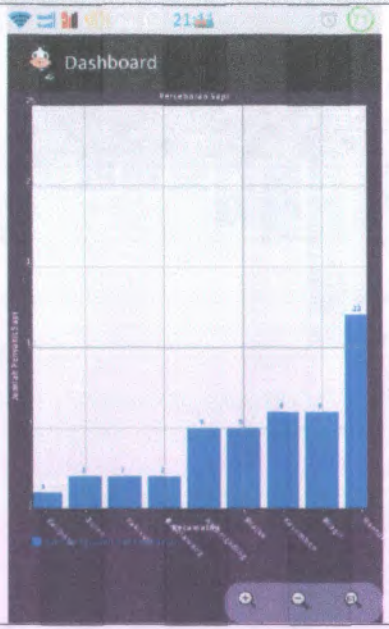
Tabel 5. 6 Table Analisis penyakit / kecamatan

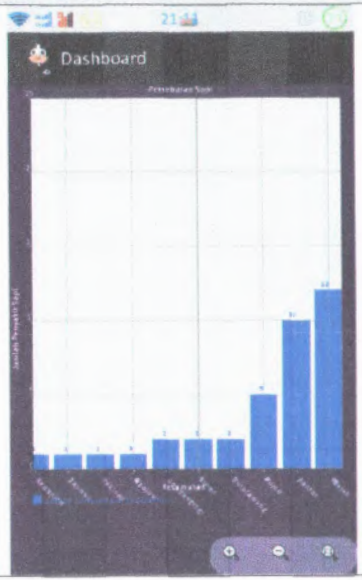
No	Nama Penyakit	Bulan	Screenshoot Aplikasi	Keterangan
1	Mastitis	Januari		Pada bulan januari, persebaran penyakit terdapat di 13 kecamatan dengan yang terbanyak ada di kecamatan Bantur sebanyak 13 kasus, kemudian di urutan kedua terbanyak terdapat di kasembon sebanyak 8 kasus, dan ketiga terbanyak berada di kecamatan Pujon dengan 5 kasus mastitis.

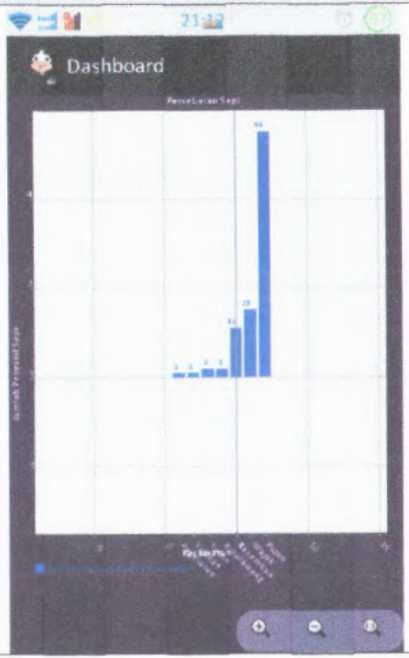
No	Nama Penyakit	Bulan	Screenshoot Aplikasi	Keterangan
2	Mastitis	Februari		Pada bulan februari, persebaran penyakit terdapat di 7 kecamatan dengan yang terbanyak ada di kecamatan Wajak sebanyak 15 kasus, kemudian di urutan kedua terbanyak terdapat di Bantur sebanyak 13 kasus, dan ketiga terbanyak berada di kecamatan wagir dengan 4 kasus mastitis.

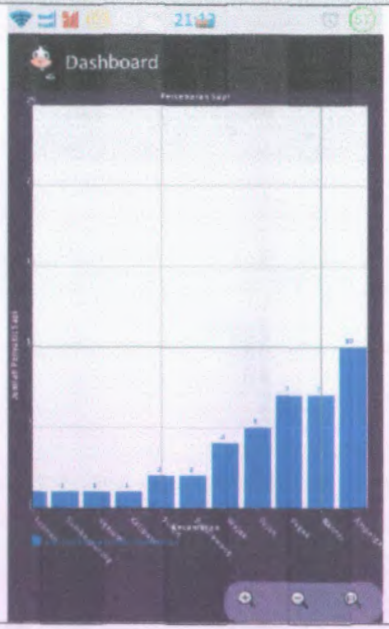
No	Nama Penyakit	Bulan	Screenshoot Aplikasi	Keterangan
3	Mastitis	Maret		Pada bulan maret, persebaran penyakit terdapat di 9 kecamatan dengan yang terbanyak ada di kecamatan Ampelgading sebanyak 20 kasus, kemudian di urutan kedua terbanyak terdapat di Bantur sebanyak 16 kasus, dan ketiga terbanyak berada di kecamatan Kasembon dengan 12 kasus mastitis.

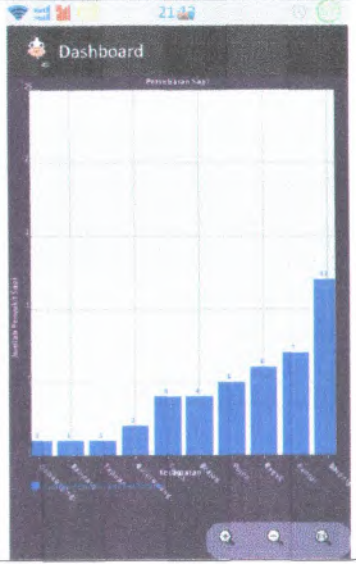
No	Nama Penyakit	Bulan	Screenshoot Aplikasi	Keterangan
4	Mastitis	April		Pada bulan April Mastitis, persebaran penyakit terdapat di 8 kecamatan dengan yang terbanyak ada di kecamatan bantur sebanyak 11 kasus, kemudian di urutan kedua terbanyak terdapat di Wagir sebanyak 11 kasus, dan ketiga terbanyak berada di kecamatan Ampelgading dengan 5 kasus mastitis.

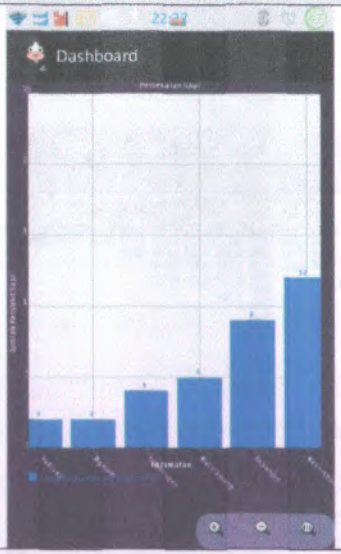
No	Nama Penyakit	Bulan	Screenshoot Aplikasi	Keterangan
5	Mastitis	Mei		Pada bulan Mei , persebaran penyakit terdapat di 9 kecamatan dengan yang terbanyak ada di kecamatan Bantur sebanyak 12 kasus, kemudian di urutan kedua terbanyak terdapat di wagir sebanyak 6 kasus, dan ketiga terbanyak berada di kecamatan Kasembon dengan 6 kasus mastitis.

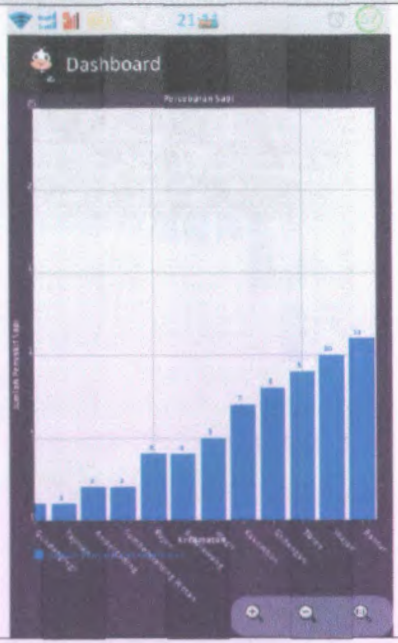
No	Nama Penyakit	Bulan	Screenshoot Aplikasi	Keterangan
6	Mastitis	Juni		Pada bulan Juni, persebaran penyakit terdapat di 10 kecamatan dengan yang terbanyak ada di kecamatan wajak sebanyak 12 kasus, kemudian di urutan kedua terbanyak terdapat di bantur sebanyak 10 kasus, dan ketiga terbanyak berada di kecamatan Pujon dengan 5 kasus mastitis.

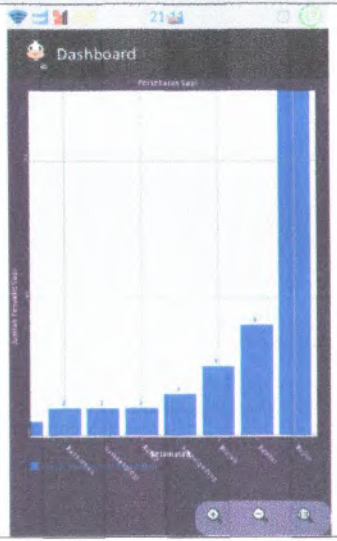
No	Nama Penyakit	Bulan	Screenshoot Aplikasi	Keterangan
10	Mastitis	Oktober		Pada bulan Oktober, persebaran penyakit terdapat di 7 kecamatan dengan yang terbanyak ada di kecamatan Pujon sebanyak 55 kasus, kemudian di urutan kedua terbanyak terdapat di wajak sebanyak 15 kasus, dan ketiga terbanyak berada di kecamatan kasembon dengan 11 kasus mastitis.

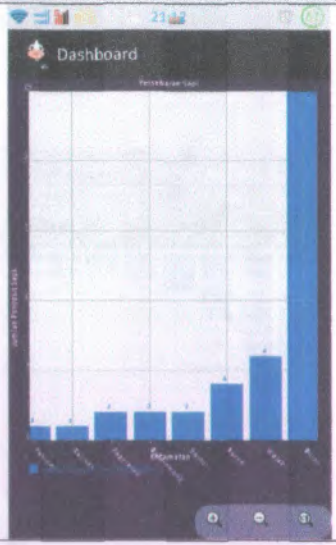
No	Nama Penyakit	Bulan	Screenshoot Aplikasi	Keterangan
11	Mastitis	November		Pada bulan November, persebaran penyakit terdapat di 11 kecamatan dengan yang terbanyak ada di kecamatan Ampelgading sebanyak 10 kasus, kemudian di urutan kedua terbanyak terdapat di Pagak sebanyak 8 kasus, dan ketiga terbanyak berada di kecamatan Bantur dengan 7 kasus mastitis.

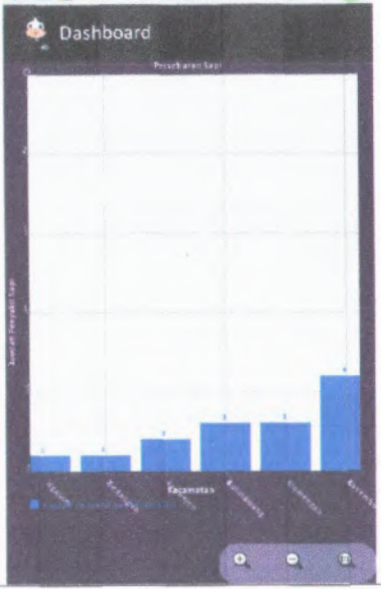
No	Nama Penyakit	Bulan	Screenshoot Aplikasi	Keterangan
12	Mastitis	Desember		Pada bulan Desember, persebaran penyakit terdapat di 10 kecamatan dengan yang terbanyak ada di kecamatan Ampelgading sebanyak 12 kasus, kemudian di urutan kedua terbanyak terdapat di Bantur sebanyak 7 kasus, dan ketiga terbanyak berada di kecamatan Pagak dengan 6 kasus mastitis.

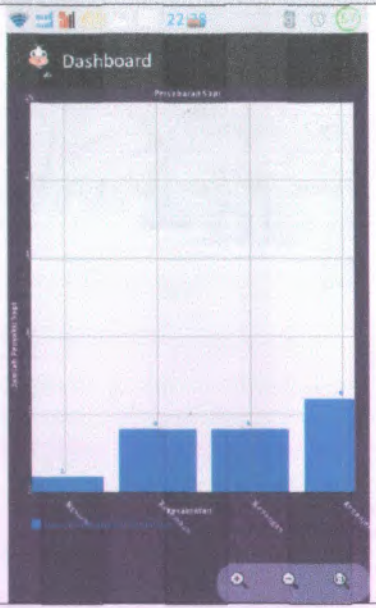
No	Nama Penyakit	Bulan	Screenshoot Aplikasi	Keterangan
13	Dismatosis	Januari		Pada bulan januari, persebaran penyakit terdapat di 6 kecamatan dengan yang terbanyak ada di kecamatan kasembon sebanyak 12 kasus, kemudian di urutan kedua terbanyak terdapat di kepanjen sebanyak 9 kasus, dan ketiga terbanyak berada di kecamatan Bululawang dengan 5 kasus Dismatosis.

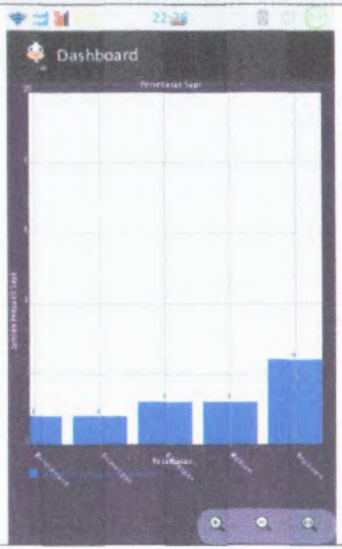
No	Nama Penyakit	Bulan	Screenshoot Aplikasi	Keterangan
7	Mastitis	Juli		Pada bulan juli, persebaran penyakit terdapat di 12 kecamatan dengan yang terbanyak ada di kecamatan Bantur sebanyak 11 kasus, kemudian di urutan kedua terbanyak terdapat di wajak sebanyak 10 kasus, dan ketiga terbanyak berada di kecamatan Turen dengan 9 kasus mastitis.

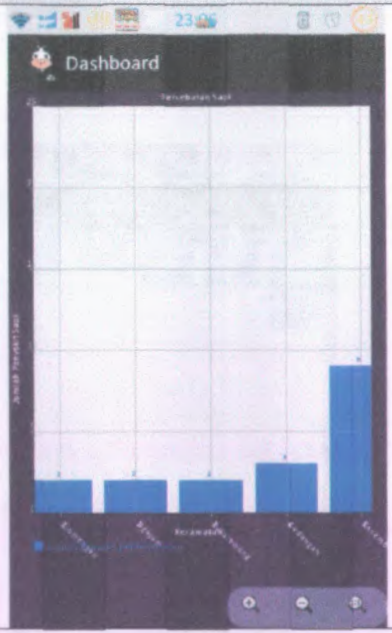
No	Nama Penyakit	Bulan	Screenshoot Aplikasi	Keterangan
8	Mastitis	Agustus		Pada bulan Agustus, persebaran penyakit terdapat di 8 kecamatan dengan yang terbanyak ada di kecamatan Pujon sebanyak 68 kasus, kemudian di urutan kedua terbanyak terdapat di bantur sebanyak 8 kasus, dan ketiga terbanyak berada di kecamatan Wajak dengan 5 kasus mastitis.

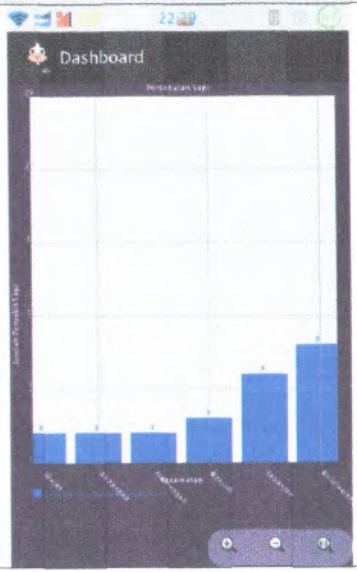
No	Nama Penyakit	Bulan	Screenshoot Aplikasi	Keterangan
9	Mastitis	September		Pada bulan September, persebaran penyakit terdapat di 8 kecamatan dengan yang terbanyak ada di kecamatan Pujon sebanyak 97 kasus, kemudian di urutan kedua terbanyak terdapat di Wajak 6 sebanyak 8 kasus, dan ketiga terbanyak berada di kecamatan Turen dengan 4 kasus mastitis.

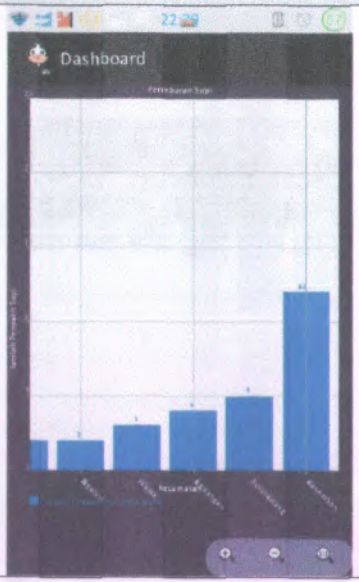
No	Nama Penyakit	Bulan	Screenshoot Aplikasi	Keterangan
14	Dismatosis	Februari		Pada bulan februari, persebaran penyakit terdapat di 6 kecamatan dengan yang terbanyak ada di kecamatan kasemmbon sebanyak 6 kasus, kemudian di urutan kedua terbanyak terdapat di kromengan sebanyak 3 kasus, dan ketiga terbanyak berada di kecamatan bululawang dengan 3 kasus dismatosis.

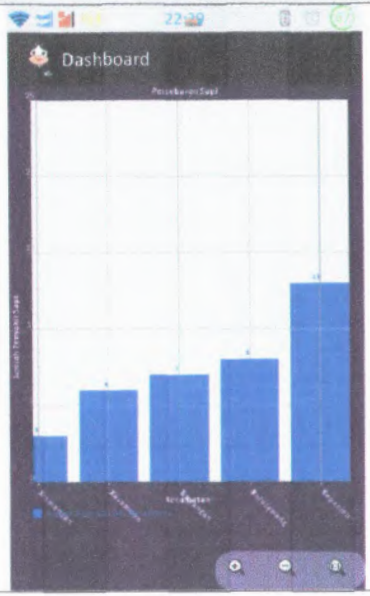
No	Nama Penyakit	Bulan	Screenshoot Aplikasi	Keterangan
15	Dismatosis	Maret		Pada bulan maret, persebaran penyakit terdapat di 4 kecamatan dengan yang terbanyak ada di kecamatan Kepanjen sebanyak 6 kasus, kemudian di urutan kedua terbanyak terdapat di kasembon sebanyak 4 kasus, dan ketiga terbanyak berada di kecamatan gedangan dengan 4 kasus dismatosis.

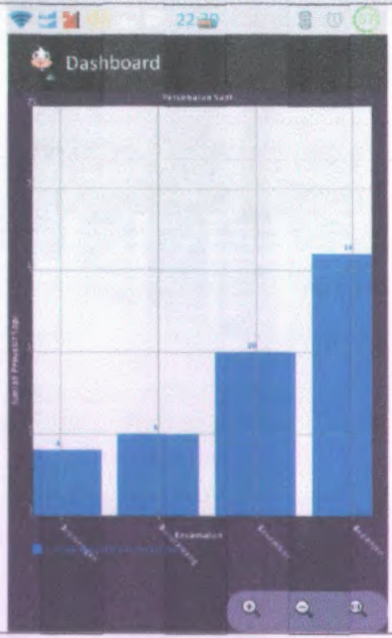
No	Nama Penyakit	Bulan	Screenshoot Aplikasi	Keterangan
16	Dismatosis	April		Pada bulan April, persebaran penyakit terdapat di 5 kecamatan dengan yang terbanyak ada di kecamatan kepanjen sebanyak 6 kasus, kemudian di urutan kedua terbanyak terdapat di Ngajum sebanyak 3 kasus, dan ketiga terbanyak berada di kecamatan gedangan dengan 3 kasus dismatosis.

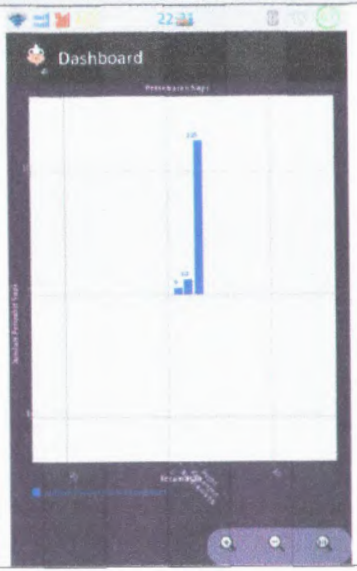
No	Nama Penyakit	Bulan	Screenshoot Aplikasi	Keterangan
17	Dismatosis	Mei		Pada bulan Mei, persebaran penyakit terdapat di 5 kecamatan dengan yang terbanyak ada di kecamatan kasembon sebanyak 10 kasus, kemudian di urutan kedua terbanyak terdapat di gedangan sebanyak 3 kasus, dan ketiga terbanyak berada di kecamatan bululawang dengan 2 kasus Dismatosis.

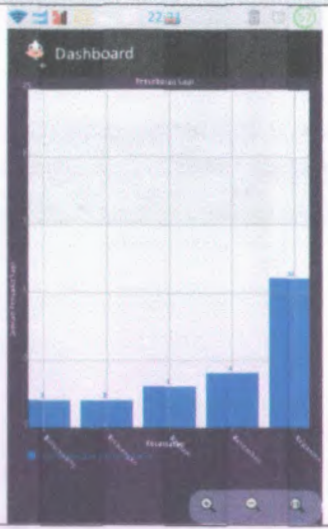
No	Nama Penyakit	Bulan	Screenshoot Aplikasi	Keterangan
18	Dismatosis	Juni		Pada bulan Juni, persebaran penyakit terdapat di 6 kecamatan dengan yang terbanyak ada di kecamatan bululawang sebanyak 8 kasus, kemudian di urutan kedua terbanyak terdapat di kepanjen sebanyak 6 kasus, dan ketiga terbanyak berada di kecamatan Ngajum dengan 3 kasus Dismatosis.

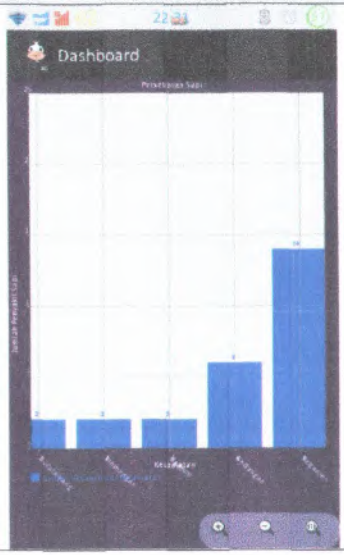
No	Nama Penyakit	Bulan	Screenshoot Aplikasi	Keterangan
19	Dismatosis	Juli		Pada bulan juli, persebaran penyakit terdapat di 6 kecamatan dengan yang terbanyak ada di kecamatan Kasemon sebanyak 12 kasus, kemudian di urutan kedua terbanyak terdapat di bululawang sebanyak 5 kasus, dan ketiga terbanyak berada di kecamatan Gedangan dengan 4 kasus Dismatosis.

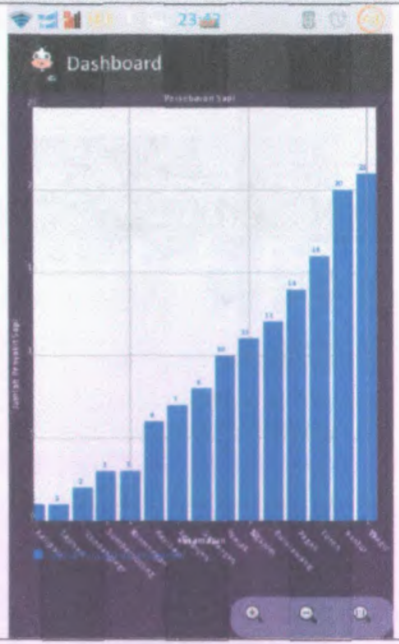
No	Nama Penyakit	Bulan	Screenshoot Aplikasi	Keterangan
20	Dismatosis	Agustus		Pada bulan Agustus, persebaran penyakit terdapat di 5 kecamatan dengan yang terbanyak ada di kecamatan Kepanjen sebanyak 13 kasus, kemudian di urutan kedua terbanyak terdapat di bululawang sebanyak 8 kasus, dan ketiga terbanyak berada di kecamatan gedangan dengan 7 kasus dismatosis.

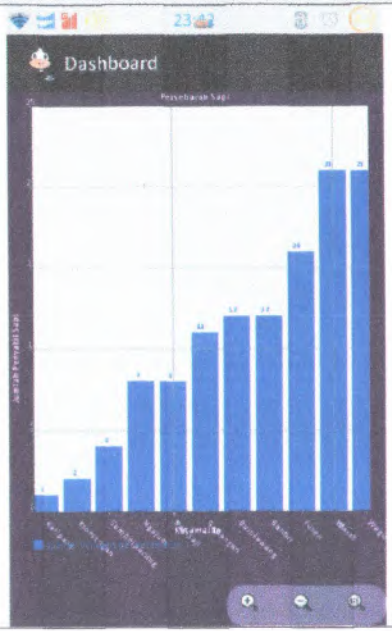
No	Nama Penyakit	Bulan	Screenshoot Aplikasi	Keterangan
21	Dismatosis	September		Pada bulan September, persebaran penyakit terdapat di 4 kecamatan dengan yang terbanyak ada di kecamatan kepanjen sebanyak 14 kasus, kemudian di urutan kedua terbanyak terdapat di kasembon sebanyak 10 kasus, dan ketiga terbanyak berada di kecamatan Bululawng dengan 5 kasus dismatosis.

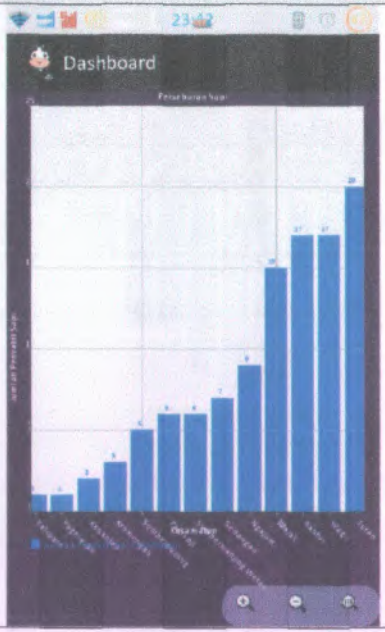
No	Nama Penyakit	Bulan	Screenshoot Aplikasi	Keterangan
22	Dismatosis	Oktober		Pada bulan Oktober, persebaran penyakit terdapat di 3 kecamatan dengan yang terbanyak ada di kecamatan Pujon sebanyak 125 kasus, kemudian di urutan kedua terbanyak terdapat di kepanjen sebanyak 12 kasus, dan ketiga terbanyak berada di kecamatan Bululawang dengan 5 kasus dismatosis.

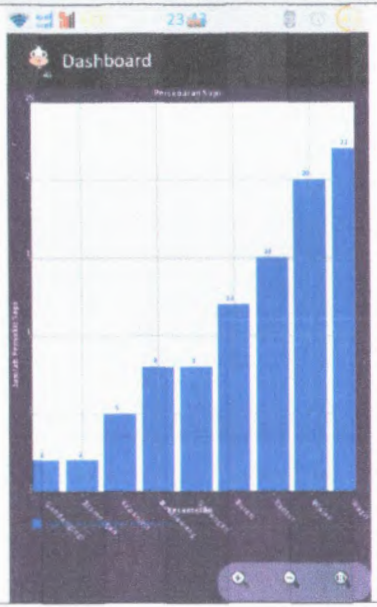
No	Nama Penyakit	Bulan	Screenshoot Aplikasi	Keterangan
23	Dismatosis	November		Pada bulan November, persebaran penyakit terdapat di 5 kecamatan dengan yang terbanyak ada di kecamatan kepanjen sebanyak 11 kasus, kemudian di urutan kedua terbanyak terdapat di Kasembon sebanyak 4 kasus, dan ketiga terbanyak berada di kecamatan Ngajum dengan 3 kasus dismatosis.

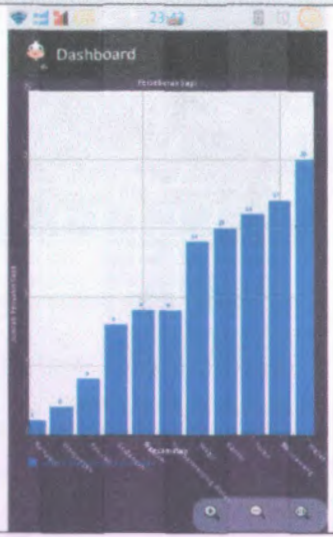
No	Nama Penyakit	Bulan	Screenshoot Aplikasi	Keterangan
24	Dismatosis	Desember		Pada bulan Desember, persebaran penyakit terdapat di 5 kecamatan dengan yang terbanyak ada di kecamatan kepanjen sebanyak 14 kasus, kemudian di urutan kedua terbanyak terdapat di Gedangan sebanyak 6 kasus, dan ketiga terbanyak berada di kecamatan Bululawang dengan sebanyak 2 kasus Dismatosis.

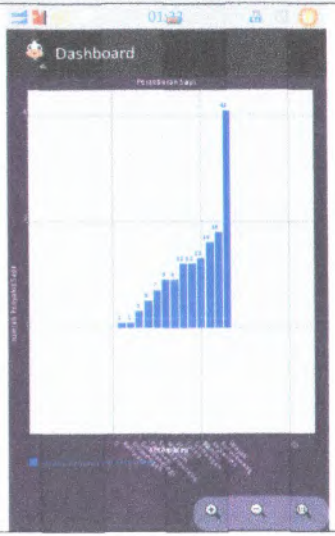
No	Nama Penyakit	Bulan	Screenshoot Aplikasi	Keterangan
25	BEF	Januari		Pada bulan januari, persebaran penyakit terdapat di 15 kecamatan dengan yang terbanyak ada di kecamatan wagir sebanyak 21 kasus, kemudian di urutan kedua terbanyak terdapat di bantur sebanyak 20 kasus, dan ketiga terbanyak berada di kecamatan Turen dengan 16 kasus BEF.

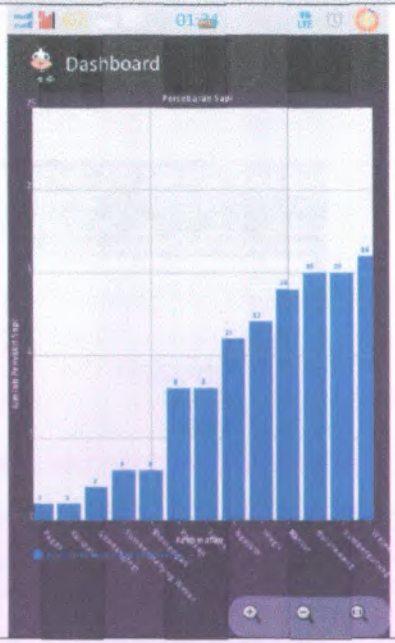
No	Nama Penyakit	Bulan	Screenshoot Aplikasi	Keterangan
26	BEF	Februari		Pada bulan februari, persebaran penyakit terdapat di 11 kecamatan dengan yang terbanyak ada di kecamatan Wagir sebanyak 21 kasus, kemudian di urutan kedua terbanyak terdapat di Wajak sebanyak 21 kasus, dan ketiga terbanyak berada di kecamatan Turen dengan 16kasus BEF.

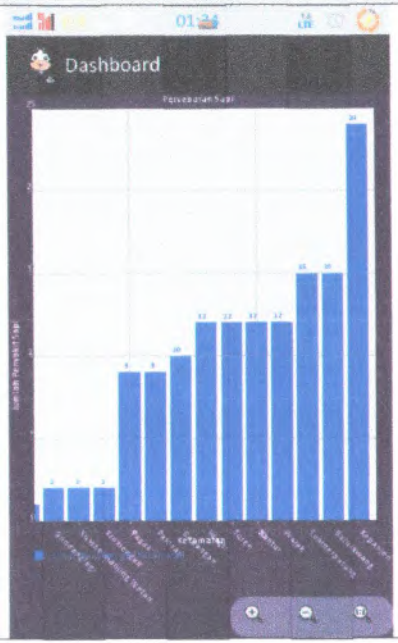
No	Nama Penyakit	Bulan	Screenshoot Aplikasi	Keterangan
27	BEF	Maret		Pada bulan maret, persebaran penyakit terdapat di 13 kecamatan dengan yang terbanyak ada di kecamatan Turen sebanyak 20 kasus, kemudian di urutan kedua terbanyak terdapat di Wagir sebanyak 17 kasus, dan ketiga terbanyak berada di kecamaan Bantur dengan 17 kasus BEF.

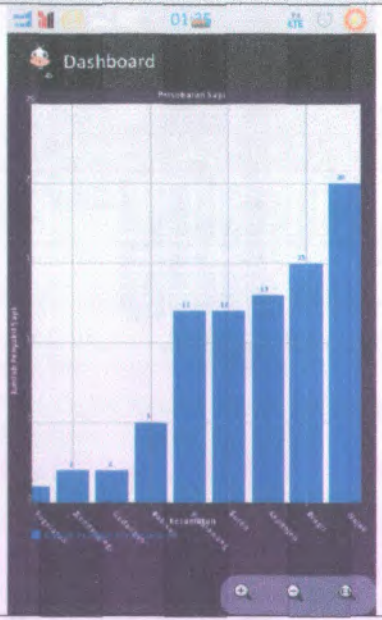
No	Nama Penyakit	Bulan	Screenshoot Aplikasi	Keterangan
28	BEF	April		Pada bulan April, persebaran penyakit terdapat di 9 kecamatan dengan yang terbanyak ada di kecamatan Wagir sebanyak 22 kasus, kemudian di urutan kedua terbanyak terdapat di Wajak sebanyak 20 kasus, dan ketiga terbanyak berada di kecamatan Bantur dengan 15 kasus BEF.

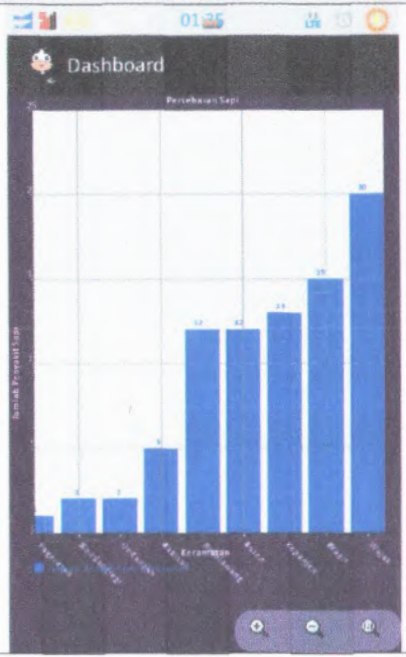
No	Nama Penyakit	Bulan	Screenshoot Aplikasi	Keterangan
29	BEF	Mei		Pada bulan Mei , persebaran penyakit terdapat di 11 kecamatan dengan yang terbanyak ada di kecamatan Wagir sebanyak 22 kasus, kemudian di urutan kedua terbanyak terdapat di Wajak sebanyak 20 kasus, dan ketiga terbanyak berada di kecamatan Kasembon dengan 15 kasus BEF.

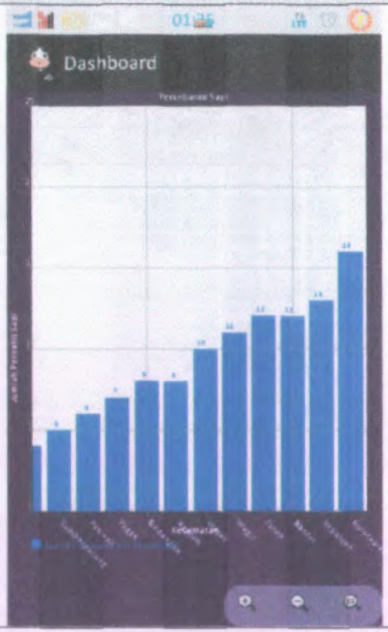
No	Nama Penyakit	Bulan	Screenshoot Aplikasi	Keterangan
30	BEF	Juni		Pada bulan Juni, persebaran penyakit terdapat di 12 kecamatan dengan yang terbanyak ada di kecamatan Wajak sebanyak 41 kasus, kemudian di urutan kedua terbanyak terdapat di Bululawang sebanyak 18 kasus, dan ketiga terbanyak berada di kecamatan Kepanjen dengan 16 kasus BEF.

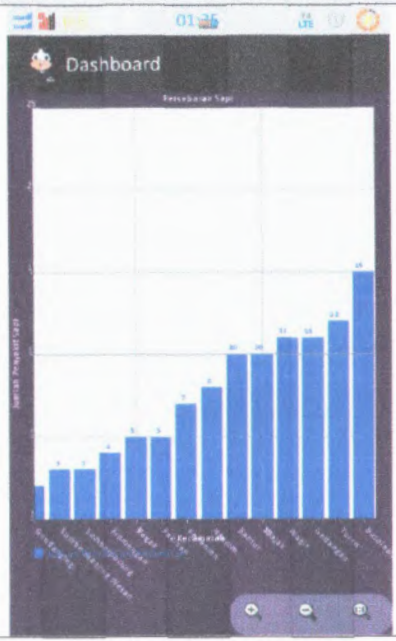
No	Nama Penyakit	Bulan	Screenshoot Aplikasi	Keterangan
31	BEF	Juli		Pada bulan juli, persebaran penyakit terdapat di 13 kecamatan dengan yang terbanyak ada di kecamatan Wajak sebanyak 16 kasus, kemudian di urutan kedua terbanyak terdapat di Bululawang sebanyak 15 kasus, dan ketiga terbanyak berada di kecamatan Sumberpucung dengan 15 kasus BEF.

No	Nama Penyakit	Bulan	Screenshoot Aplikasi	Keterangan
32	BEF	Agustus		Pada bulan Agustus, persebaran penyakit terdapat di 14 kecamatan dengan yang terbanyak ada di kecamatan Kepanjen sebanyak 24 kasus, kemudian di urutan kedua terbanyak terdapat di Bululawang sebanyak 15 kasus, dan ketiga terbanyak berada di kecamatan Sumberpucung dengan 15 kasus BEF.

No	Nama Penyakit	Bulan	Screenshoot Aplikasi	Keterangan
33	BEF	September		Pada bulan September, persebaran penyakit terdapat di 9 kecamatan dengan yang terbanyak ada di kecamatan Wajak sebanyak 20 kasus, kemudian di urutan kedua terbanyak terdapat di Turen sebanyak 17 kasus, dan ketiga terbanyak berada di kecamatan Bululawang dengan 15 kasus BEF.

No	Nama Penyakit	Bulan	Screenshoot Aplikasi	Keterangan
34	BEF	Oktober		Pada bulan Oktober, persebaran penyakit terdapat di 9 kecamatan dengan yang terbanyak ada di kecamatan Wajak sebanyak 20 kasus, kemudian di urutan kedua terbanyak terdapat di Wagir sebanyak 15 kasus, dan ketiga terbanyak berada di kecamatan kepanjen dengan 13 kasus BEF.

No	Nama Penyakit	Bulan	Screenshoot Aplikasi	Keterangan
35	BEF	November		Pada bulan November, persebaran penyakit terdapat di 12 kecamatan dengan yang terbanyak ada di kecamatan bululaang sebanyak 16 kasus, kemudian di urutan kedua terbanyak terdapat di Kepanjen sebanyak 13 kasus, dan ketiga terbanyak berada di kecamatan Bantur dengan 12 kasus BEF.

No	Nama Penyakit	Bulan	Screenshoot Aplikasi	Keterangan
36	BEF	Desember		Pada bulan Desember, persebaran penyakit terdapat di 14 kecamatan dengan yang terbanyak ada di kecamatan bululawang sebanyak 15 kasus, kemudian di urutan kedua terbanyak terdapat di Turen sebanyak 12 kasus, dan ketiga terbanyak berada di kecamatan Gedangan dengan 6 kasus BEF.

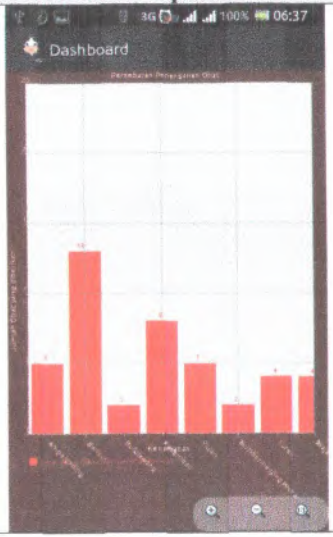
Dari tabel – tabel diatas didapatkan untuk tahun 2012, jenis penyakit Mastitis banyak terjadi di kecamatan Pujon sebanyak 248 kasus sapi. Untuk jenis penyakit sapi dismatosis persebaran terbanyak terdapat di Pujon dengan 125 kasus. Dan yang terakhir BEF lebih banyak tersebar di kecamatan Wajak dengan jumlah kasus 215.

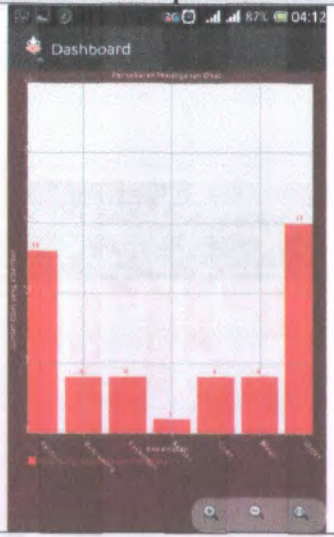
Merujuk dari hasil diatas, maka dapat dianalisis untuk penanganan lebih ditekankan pada ketiga kecamatan tersebut dikarenakan jumlah kasus ternak sapi yang terjangkit cukup tinggi.

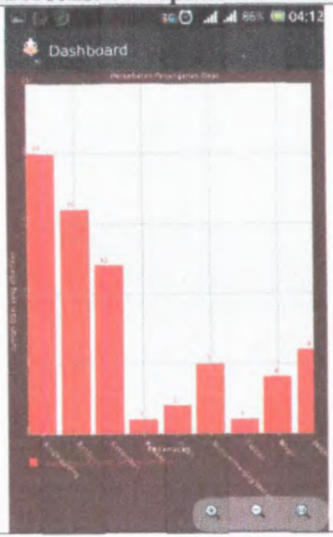
5.2.2 Analisis Penanganan Penyakit Sapi Berdasarkan Jenis Obat / Kecamatan

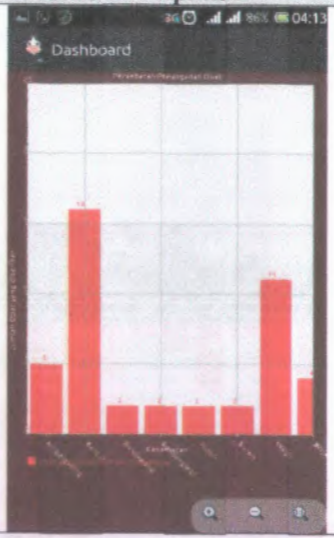
Analisis penanganan penyakit sapi dapat dilihat di setiap kecamatan yang ada di malang pada setiap bulan. Fitur ini menganalis jumlah pemberian obat yang dipilih berdasarkan pilihan user pada bulan tertentu dan tahun tertentu. Nantinya dengan fitur ini, akan disajikan informasi mengenai persebaran penyakit pada tiap kecamatan. Hasil analisis lebih detail dapat dilihat pada tabel dibawah ini

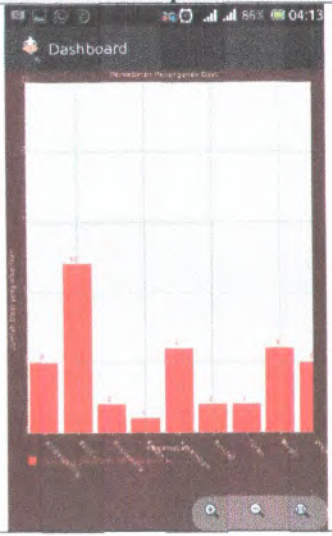
Tabel 5. 7 Analis penanganan/ kecamatan

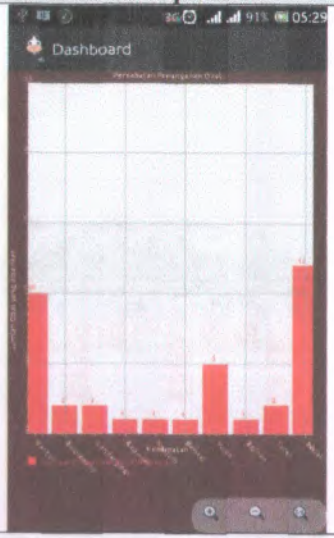
No	Nama obat	Bulan	Screenshoot Aplikasi	Keterangan
1	Antibiotik	Januari		Pada bulan Januari, jumlah pemberian obat antibiotik terdapat di 13 kecamatan dengan yang terbanyak ada di kecamatan Bantur sebanyak 13 penanganan, kemudian di urutan kedua terbanyak terdapat di kasembon sebanyak 8 penanganan, dan ketiga terbanyak berada di kecamatan Pujon dengan 5 penanganan.

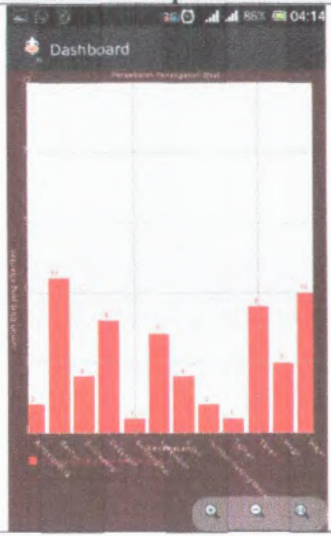
No	Nama obat	Bulan	Screenshoot Aplikasi	Keterangan
2	Antibiotik	Februari		Pada bulan Februari, jumlah pemberian obat antibiotik terdapat di 7 kecamatan dengan yang terbanyak ada di kecamatan Wajak sebanyak 15 penanganan, kemudian di urutan kedua terbanyak terdapat di Bantur sebanyak 13 penanganan, dan ketiga terbanyak berada di kecamatan wagir dengan 4 kasus penanganan.

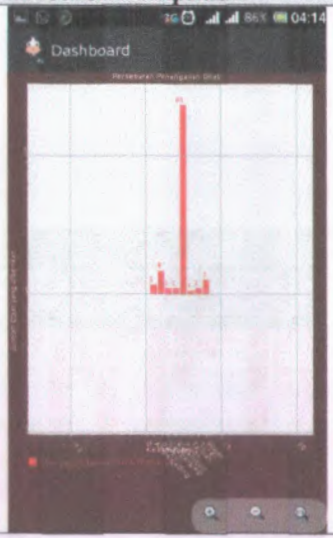
No	Nama obat	Bulan	Screenshoot Aplikasi	Keterangan
3	Antibiotik	Maret		Pada bulan maret, jumlah pemberian obat antibiotik terdapat di 9 kecamatan dengan yang terbanyak ada di kecamatan Ampelgading sebanyak 20 penanganan, kemudian di urutan kedua terbanyak terdapat di Bantur sebanyak 16 penanganan, dan ketiga terbanyak berada di kecamatan Kasembon dengan 12 penanganan .

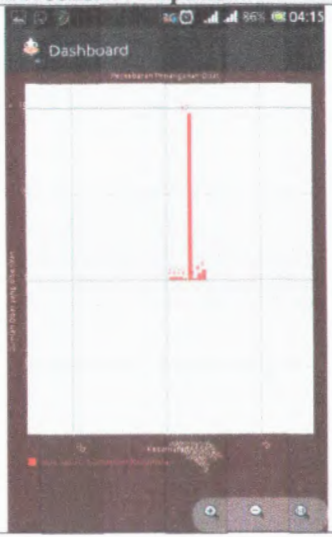
No	Nama obat	Bulan	Screenshoot Aplikasi	Keterangan
4	Antibiotik	April		Pada bulan April, jumlah pemberian obat antibiotik terdapat di 8 kecamatan dengan yang terbanyak ada di kecamatan bantur sebanyak 16 penanganan, kemudian di urutan kedua terbanyak terdapat di Wagir sebanyak 11 penanganan, dan ketiga terbanyak berada di kecamatan Ampelgading dengan 5 penanganan.

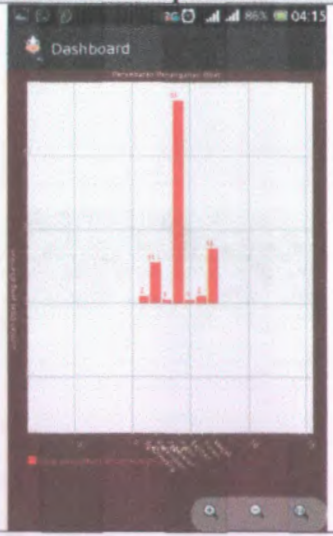
No	Nama obat	Bulan	Screenshoot Aplikasi	Keterangan
5	Antibiotik	Mei		Pada bulan Mei , jumlah pemberian obat antibiotik terdapat di 9 kecamatan dengan yang terbanyak ada di kecamatan Bantur sebanyak 12 penanganan, kemudian di urutan kedua terbanyak terdapat di wagir sebanyak 6 penanganan, dan ketiga terbanyak berada di kecamatan Kasembon dengan 6 penanganan.

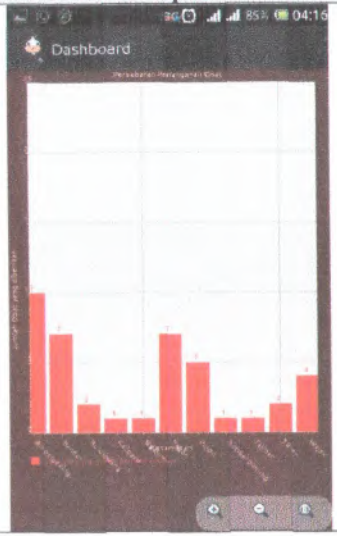
No	Nama obat	Bulan	Screenshoot Aplikasi	Keterangan
6	Antibiotik	Juni		Pada bulan Juni, jumlah pemberian obat antibiotik terdapat di 10 kecamatan dengan yang terbanyak ada di kecamatan wajak sebanyak 12 penanganan, kemudian di urutan kedua terbanyak terdapat di bantur sebanyak 10 penanganan, dan ketiga terbanyak berada di kecamatan Pujon dengan 5 penanganan.

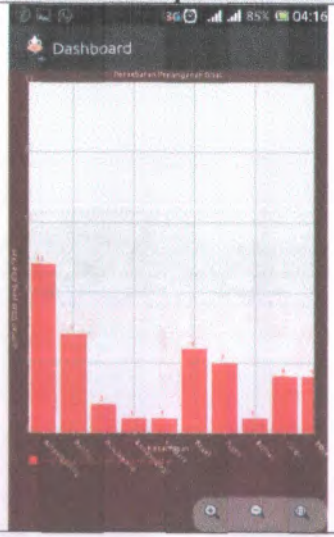
No	Nama obat	Bulan	Screenshoot Aplikasi	Keterangan
7	Antibiotik	Juli		Pada bulan Juli, , jumlah pemberian obat antibiotik terdapat di 12 kecamatan dengan yang terbanyak ada di kecamatan Bantur sebanyak 11 penanganan, kemudian di urutan kedua terbanyak terdapat di wajak sebanyak 10 penanganan, dan ketiga terbanyak berada di kecamatan Turen dengan 9 penanganan

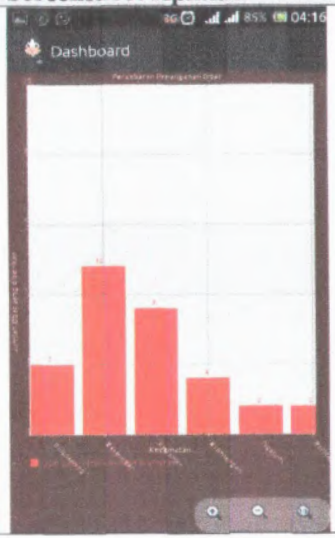
No	Nama obat	Bulan	Screenshoot Aplikasi	Keterangan
8	Antibiotik	Agustus		Pada bulan Agustus, jumlah pemberian obat antibiotik terdapat di 8 kecamatan dengan yang terbanyak ada di kecamatan Pujon sebanyak 68 penanganan, kemudian di urutan kedua terbanyak terdapat di bantur sebanyak 8 penanganan, dan ketiga terbanyak berada di kecamatan Wajak dengan 5 penanganan

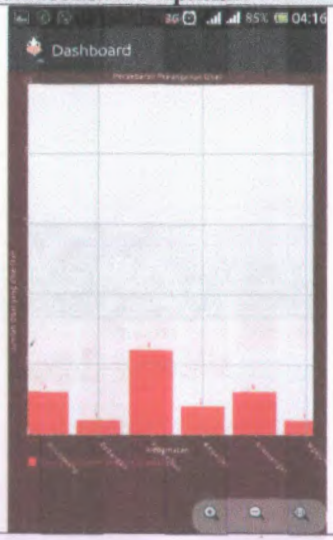
No	Nama obat	Bulan	Screenshoot Aplikasi	Keterangan
9	Antibiotik	September		Pada bulan September, jumlah pemberian obat antibiotik terdapat di 8 kecamatan dengan yang terbanyak ada di kecamatan Pujon sebanyak 97 penanganan, kemudian di urutan kedua terbanyak terdapat di Wajak 6 sebanyak 8 penanganan, dan ketiga terbanyak berada di kecamatan Turen dengan 4 penanganan.

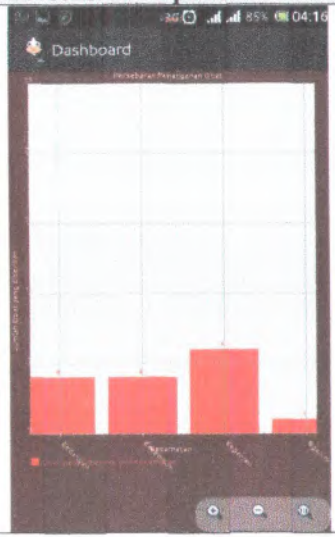
No	Nama obat	Bulan	Screenshoot Aplikasi	Keterangan
10	Antibiotik	Oktober		Pada bulan Oktober, jumlah pemberian obat antibiotik terdapat di 7 kecamatan dengan yang terbanyak ada di kecamatan Pujon sebanyak 55 penanganan, kemudian di urutan kedua terbanyak terdapat di wajak sebanyak 15 penanganan, dan ketiga terbanyak berada di kecamatan kasembon dengan 11 penanganan.

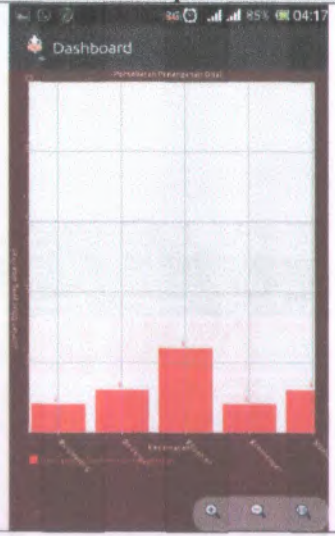
No	Nama obat	Bulan	Screenshoot Aplikasi	Keterangan
11	Antibiotik	November		Pada bulan November, jumlah pemberian obat antibiotik terdapat di 11 kecamatan dengan yang terbanyak ada di kecamatan Ampelgading sebanyak 10 penanganan, kemudian di urutan kedua terbanyak terdapat di Pagak sebanyak 8 penanganan, dan ketiga terbanyak berada di kecamatan Bantur dengan 7 penanganan.

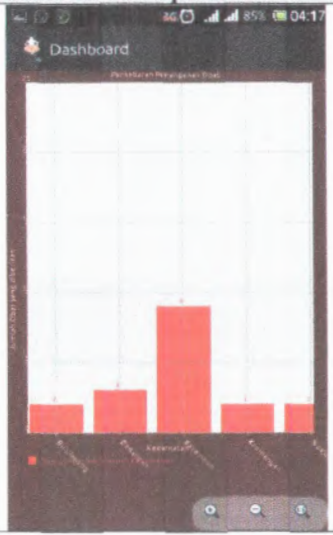
No	Nama obat	Bulan	Screenshoot Aplikasi	Keterangan
12	Antibiotik	Desember		Pada bulan Desember, jumlah pemberian obat antibiotik terdapat di 10 kecamatan dengan yang terbanyak ada di kecamatan Ampelgading sebanyak 12 penanganan, kemudian di urutan kedua terbanyak terdapat di Bantur sebanyak 7 penanganan, dan ketiga terbanyak berada di kecamatan Pagak dengan 6 kasus penanganan.

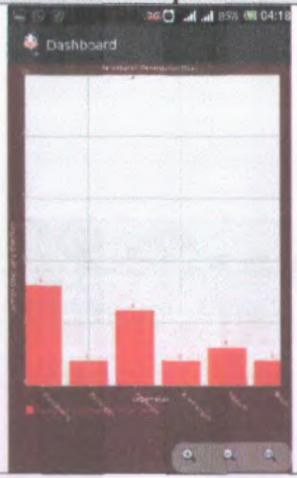
No	Nama obat	Bulan	Screenshoot Aplikasi	Keterangan
13	Antipiretik	Januari		Pada bulan januari, jumlah pemberian obat antipiretik terdapat di 6 kecamatan dengan yang terbanyak ada di kecamatan kasembon sebanyak 12 penanganan, kemudian di urutan kedua terbanyak terdapat di kepanjen sebanyak 9 penanganan, dan ketiga terbanyak berada di kecamatan Bululawang dengan 5 penanganan.

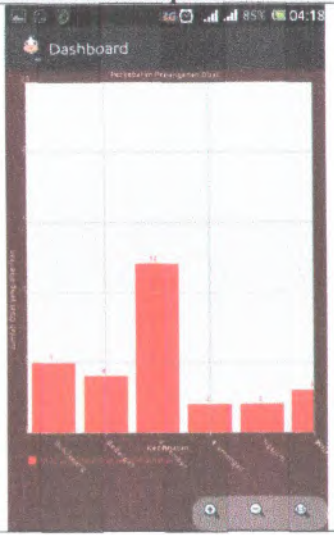
No	Nama obat	Bulan	Screenshoot Aplikasi	Keterangan
14	Antipiretik	Februari		Pada bulan februari, jumlah pemberian obat antipiretik terdapat di 6 kecamatan dengan yang terbanyak ada di kecamatan kasembon sebanyak 6 penanganan, kemudian di urutan kedua terbanyak terdapat di kromengan sebanyak 3 penanganan, dan ketiga terbanyak berada di kecamatan bululawang dengan 3 penanganan.

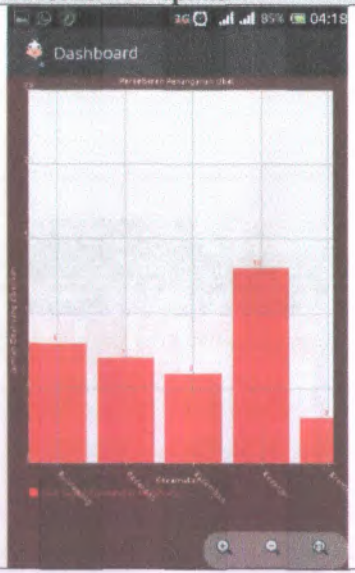
No	Nama obat	Bulan	Screenshoot Aplikasi	Keterangan
15	Antipiretik	Maret		Pada bulan maret, jumlah pemberian obat antipiretik terdapat di 4 kecamatan dengan yang terbanyak ada di kecamatan Kepanjen sebanyak 6 penanganan, kemudian di urutan kedua terbanyak terdapat di kasembon sebanyak 4 penanganan, dan ketiga terbanyak berada di kecamatan gedangan dengan 4 penanganan .

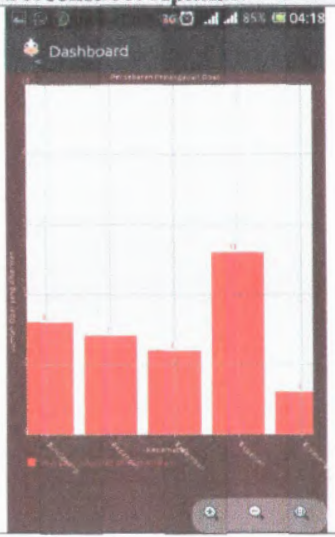
No	Nama obat	Bulan	Screenshoot Aplikasi	Keterangan
16	Antipiretik	April		Pada bulan April, jumlah pemberian obat antipiretik terdapat di 5 kecamatan dengan yang terbanyak ada di kecamatan kepanjen sebanyak 6 penanganan, kemudian di urutan kedua terbanyak terdapat di Ngajum sebanyak 3 penanganan, dan ketiga terbanyak berada di kecamatan gedangan dengan 3 penanganan.

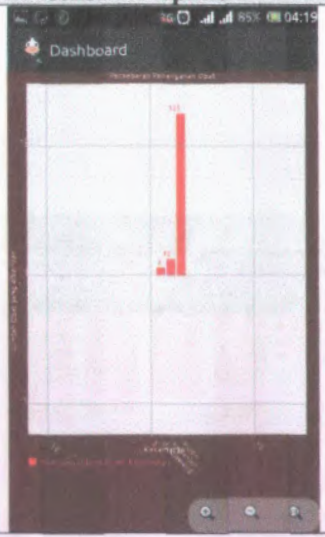
No	Nama obat	Bulan	Screenshoot Aplikasi	Keterangan
17	Antipiretik	Mei		Pada bulan Mei, jumlah pemberian obat antipiretik terdapat di 5 kecamatan dengan yang terbanyak ada di kecamatan kasembon sebanyak 9 penanganan, kemudian di urutan kedua terbanyak terdapat di gedangan sebanyak 3 penaganan, dan ketiga terbanyak berada di kecamatan bululawang dengan 2 penanganan.

No	Nama obat	Bulan	Screenshoot Aplikasi	Keterangan
18	Antipiretik	Juni		Pada bulan Juni jumlah pemberian obat antipiretik terdapat di 6 kecamatan dengan yang terbanyak ada di kecamatan bululawang sebanyak 8 penanganan, kemudian di urutan kedua terbanyak terdapat di kepanjen sebanyak 6 penanganan, dan ketiga terbanyak berada di kecamatan Ngajum dengan 3 penanganan.

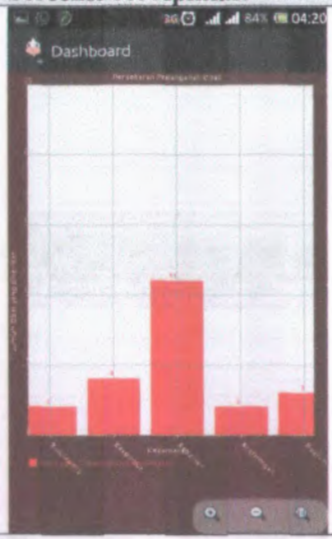
No	Nama obat	Bulan	Screenshoot Aplikasi	Keterangan
19	Antipiretik	Juli		Pada bulan juli, jumlah pemberian obat antipiretik terdapat di 6 kecamatan dengan yang terbanyak ada di kecamatan Kasemon sebanyak 12 penanganan, kemudian di urutan kedua terbanyak terdapat di bululawang sebanyak 5 penanganan, dan ketiga terbanyak berada di kecamatan Gedangan dengan 4 kasus penanganan.

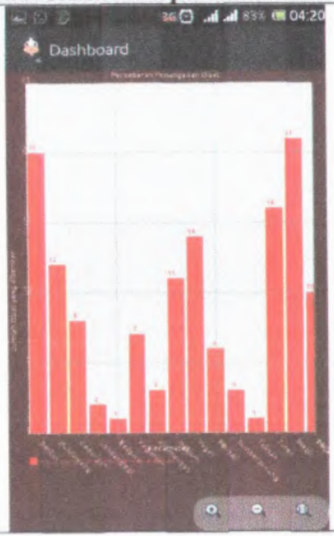
No	Nama obat	Bulan	Screenshoot Aplikasi	Keterangan
20	Antipiretik	Agustus		Pada bulan Agustus, jumlah pemberian obat antipiretik terdapat di 5 kecamatan dengan yang terbanyak ada di kecamatan Kepanjen sebanyak 13 penanganan, kemudian di urutan kedua terbanyak terdapat di bululawang sebanyak 8 penanganan, dan ketiga terbanyak berada di kecamatan gedangan dengan 7 penanganan.

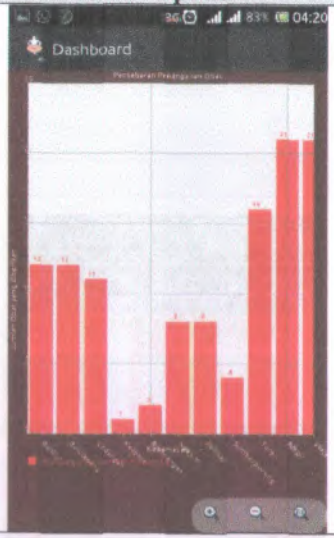
No	Nama obat	Bulan	Screenshoot Aplikasi	Keterangan
21	Antipiretik	September		Pada bulan September, jumlah pemberian obat antipiretik terdapat di 4 kecamatan dengan yang terbanyak ada di kecamatan kepanjen sebanyak 14 penanganan, kemudian di urutan kedua terbanyak terdapat di kasembon sebanyak 10 penanganan, dan ketiga terbanyak berada di kecamatan Bululawng dengan 5 penanganan.

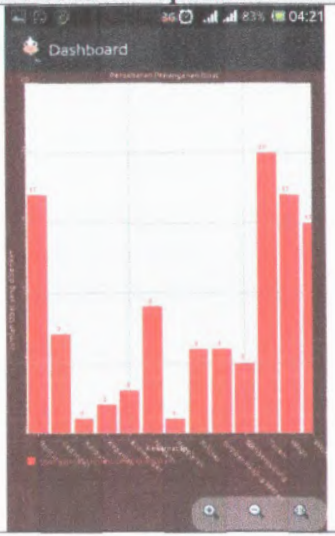
No	Nama obat	Bulan	Screenshoot Aplikasi	Keterangan
22	Antipiretik	Oktober		Pada bulan Oktober, jumlah pemberian obat antipiretik terdapat di 3 kecamatan dengan yang terbanyak ada di kecamatan Pujon sebanyak 125 penanganan, kemudian di urutan kedua terbanyak terdapat di kepanjen sebanyak 12 penanganan, dan ketiga terbanyak berada di kecamatan Bululawang dengan 5 penanganan.

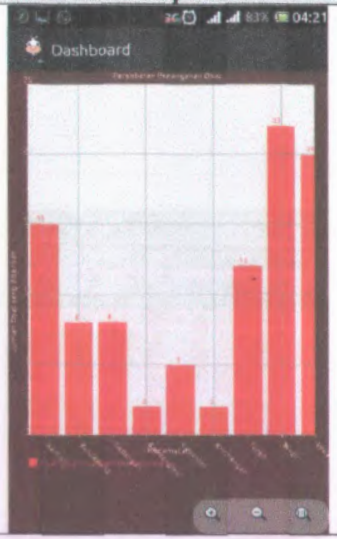
No	Nama obat	Bulan	Screenshoot Aplikasi	Keterangan
23	Antipiretik	November		Pada bulan November, jumlah pemberian obat antipiretik terdapat di 5 kecamatan dengan yang terbanyak ada di kecamatan kepanjen sebanyak 11 penanganan, kemudian di urutan kedua terbanyak terdapat di Kasembon sebanyak 4 penanganan, dan ketiga terbanyak berada di kecamatan Ngajum dengan 3 penanganan.

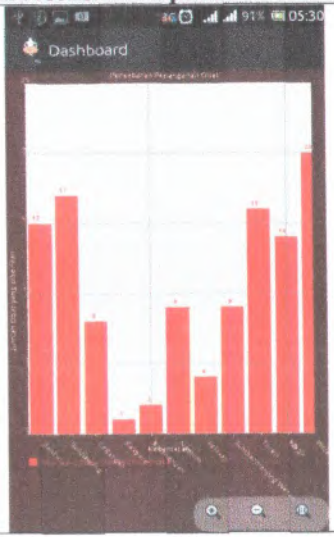
No	Nama obat	Bulan	Screenshoot Aplikasi	Keterangan
24	Antipiretik	Desember		Pada bulan Desember, jumlah pemberian obat antipiretik terdapat di 5 kecamatan dengan yang terbanyak ada di kecamatan kepanjen sebanyak 14 penanganan, kemudian di urutan kedua terbanyak terdapat di Gedangan sebanyak 6 penanganan, dan ketiga terbanyak berada di kecamatan Bululawang dengan sebanyak 2 penanganan

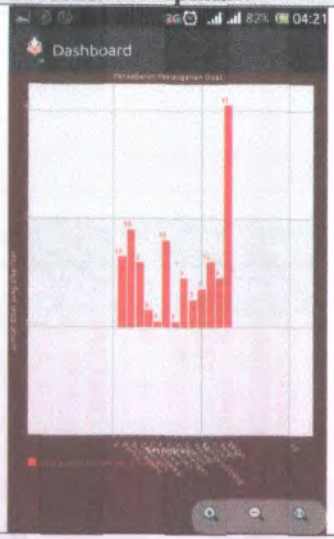
No	Nama obat	Bulan	Screenshoot Aplikasi	Keterangan
25	Anthelmentik	Januari		Pada bulan januari, jumlah pemberian obat anthelmentik terdapat di 15 kecamatan dengan yang terbanyak ada di kecamatan wagir sebanyak 21 penanganan, kemudian di urutan kedua terbanyak terdapat di bantur sebanyak 20 penanganan, dan ketiga terbanyak berada di kecamatan Turen dengan 16 penanganan.

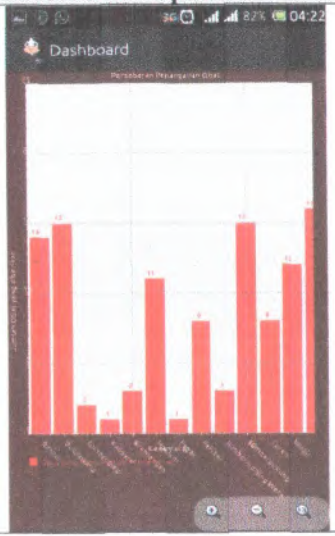
No	Nama obat	Bulan	Screenshoot Aplikasi	Keterangan
26	Anthelmentik	Februari		Pada bulan februari, jumlah pemberian obat anthelmentik terdapat di 11 kecamatan dengan yang terbanyak ada di kecamatan Wagir sebanyak 21 penanganan, kemudian di urutan kedua terbanyak terdapat di Wajak sebanyak 21 penanganan, dan ketiga terbanyak berada di kecamatan Turen dengan 16 penanganan.

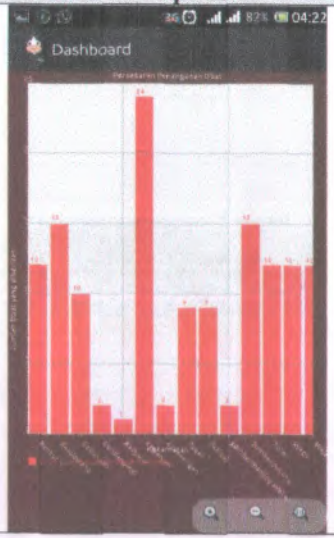
No	Nama obat	Bulan	Screenshoot Aplikasi	Keterangan
27	Anthelmentik	Maret		Pada bulan maret, jumlah pemberian obat anthelmentik terdapat di 13 kecamatan dengan yang terbanyak ada di kecamatan Turen sebanyak 20 penanganan, kemudian di urutan kedua terbanyak terdapat di Wagir sebanyak 17 penanganan, dan ketiga terbanyak berada di kecamaan Bantur dengan 17 penanganan.

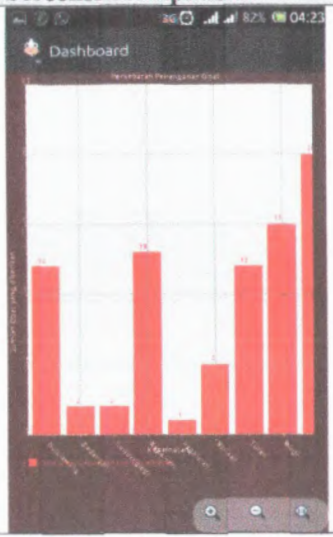
No	Nama obat	Bulan	Screenshoot Aplikasi	Keterangan
28	Anthelmentik	April		Pada bulan April, jumlah pemberian obat anthelmentik terdapat di 9 kecamatan dengan yang terbanyak ada di kecamatan Wagir sebanyak 22 penanganan, kemudian di urutan kedua terbanyak terdapat di Wajak sebanyak 20 penanganan, dan ketiga terbanyak berada di kecamatan Bantur dengan 15 penanganan.

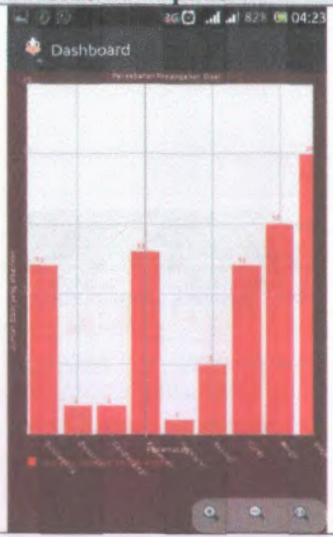
No	Nama obat	Bulan	Screenshoot Aplikasi	Keterangan
29	Anthelmentik	Mei		Pada bulan Mei , jumlah pemberian obat anthelmentik terdapat di 12 kecamatan dengan yang terbanyak ada di kecamatan Wajak sebanyak 20 penanganan, kemudian di urutan kedua terbanyak terdapat di Bantur sebanyak 17 penanganan, dan ketiga terbanyak berada di kecamatan Bululawang dengan 15 penanganan.

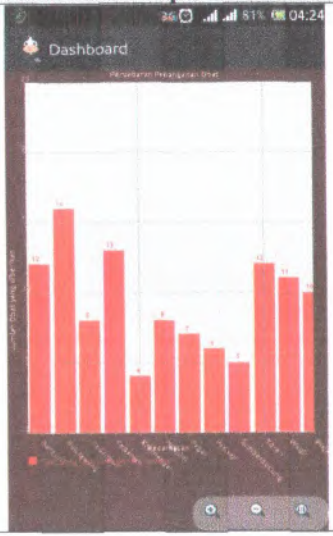
No	Nama obat	Bulan	Screenshoot Aplikasi	Keterangan
30	Anthelmentik	Juni		Pada bulan Juni, jumlah pemberian obat anthelmentik terdapat di 12 kecamatan dengan yang terbanyak ada di kecamatan Wajak sebanyak 41 penanganan, kemudian di urutan kedua terbanyak terdapat di Bululawang sebanyak 18 penanganan, dan ketiga terbanyak berada di kecamatan Kepanjen dengan 16 penanganan.

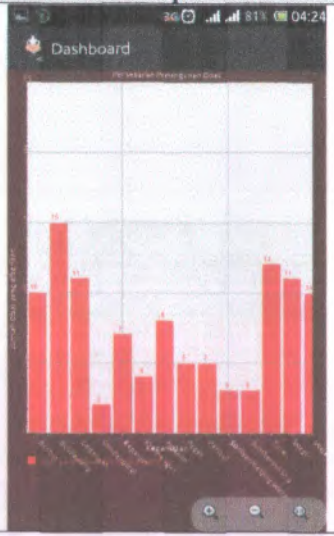
No	Nama obat	Bulan	Screenshoot Aplikasi	Keterangan
31	Anthelmentik	Juli		Pada bulan juli, jumlah pemberian obat anthelmentik terdapat di 13 kecamatan dengan yang terbanyak ada di kecamatan Wajak sebanyak 16 penanganan, kemudian di urutan kedua terbanyak terdapat di Bululawang sebanyak 15 penanganan, dan ketiga terbanyak berada di kecamatan Sumberpucung dengan 15 penanganan.

No	Nama obat	Bulan	Screenshoot Aplikasi	Keterangan
32	Anthelmentik	Agustus		Pada bulan Agustus, jumlah pemberian obat anthelmentik terdapat di 14 kecamatan dengan yang terbanyak ada di kecamatan Kepanjen sebanyak 24 penanganan ,kemudian di urutan kedua terbanyak terdapat di Bululawang sebanyak 15 penanganan, dan ketiga terbanyak berada di kecamatan Sumberpucung dengan 15 penanganan.

No	Nama obat	Bulan	Screenshoot Aplikasi	Keterangan
33	Anthelmentik	September		Pada bulan September, jumlah pemberian obat anthelmentik terdapat di 9 kecamatan dengan yang terbanyak ada di kecamatan Wajak sebanyak 20 penanganan, kemudian di urutan kedua terbanyak terdapat di Turen sebanyak 17 penanganan, dan ketiga terbanyak berada di kecamatan Bululawang dengan 15 penanganan.

No	Nama obat	Bulan	Screenshoot Aplikasi	Keterangan
34	Anthelmentik	Oktober		Pada bulan Oktober, jumlah pemberian obat anthelmentik terdapat di 9 kecamatan dengan yang terbanyak ada di kecamatan Wajak sebanyak 20 penanganan, kemudian di urutan kedua terbanyak terdapat di Wagir sebanyak 15 penanganan, dan ketiga terbanyak berada di kecamatan kepanjen dengan 13 penanganan.

No	Nama obat	Bulan	Screenshoot Aplikasi	Keterangan
35	Anthelmentik	November		Pada bulan November, jumlah pemberian obat anthelmentik terdapat di 12 kecamatan dengan yang terbanyak ada di kecamatan bululaang sebanyak 16 penanganan, kemudian di urutan kedua terbanyak terdapat di Kepanjen sebanyak 13 penanganan, dan ketiga terbanyak berada di kecamatan Bantur dengan 12 penanganan.

No	Nama obat	Bulan	Screenshoot Aplikasi	Keterangan
36	Anthelmentik	Desember		Pada bulan Desember, jumlah pemberian obat anthelmentik terdapat di 14 kecamatan dengan yang terbanyak ada di kecamatan bululawang sebanyak 15 penanganan , kemudian di urutan kedua terbanyak terdapat di Turen sebanyak 12 penanganan, dan ketiga terbanyak berada di kecamatan Gedangan dengan 6 penanganan.

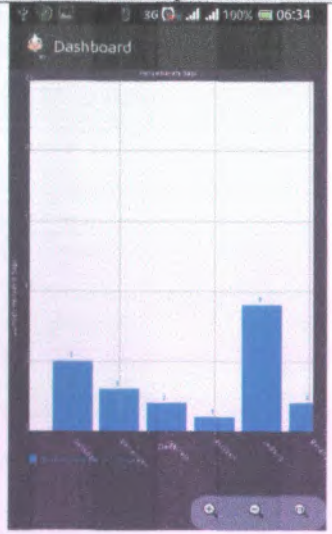
Dari tabel – tabel diatas didapatkan untuk tahun 2012, Penanganan dengan obat *antibiotik* banyak terjadi di kecamatan Pujon sebanyak 248 penanganan. Untuk tipe obat *Anthelmentik* persebaran terbanyak terdapat di Pujon dengan 125 penanganan. Dan yang terakhir tipe obat *Antipiretik* lebih banyak tersebar di kecamatan Wajak dengan jumlah penanganan 215.

Merujuk dari hasil diatas, maka dapat dianalisis untuk penanganan dan jumlah obat yang diberikan lebih ditekankan pada ketiga kecamatan tersebut dikarenakan jumlah kasus ternak sapi yang terjangkit cukup tinggi .

5.2.3 Analisis Penanganan Penyakit Sapi Berdasarkan Jenis Obat / Desa

Analisis penanganan penyakit sapi dapat dilihat di setiap kecamatan yang ada di malang pada setiap bulan. Fitur ini menganalis jumlah pemberian obat yang dipilih berdasarkan pilihan user pada kecamatan, bulan dan tahun tertentu. Nantinya dengan fitur ini, akan disajikan informasi mengenai persebaran penyakit pada tiap kecamatan. Hasil analisis lebih detail dapat dilihat pada tabel dibawah ini.

Tabel 5. 8 Analisis penanganan/desa

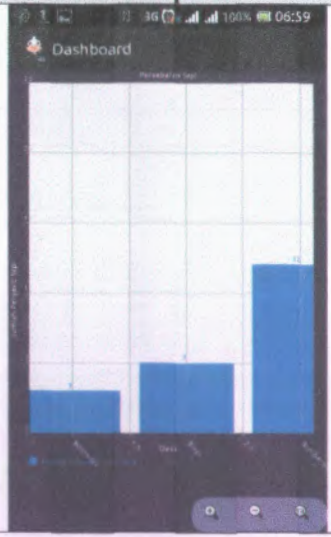
No	Nama obat	Bulan	Kecamatan	Screenshoot Aplikasi	Keterangan
1	Antibiotik	April	Wagir		Pada bulan April , jumlah pemberian obat antibiotik di kecamatan Wagir mencapai 22 penanganan . Desa yang menerima penanganan obat antibiotik terbanyak yaitu ada di desa jedang dengan 9 Antibiotik, kemudian desa kalisodo dengan 5 antibiotik dan yang terakhir di desa banengan dengan 3 antibiotik

Dari tabel diatas dapat dianalisis pada bulan April dan tahun 2012 diketahui ada 6 desa yang menerima bantuan obat antibiotik di kecamatan wagir . Dengan menggunakan aplikasi *strategic mobile dashboard* dapat mengetahui tingkat penanganan pemberian obat per desa yang memudahkan kepala dinas peternakan ketika ingin mengetahui proporsi jumlah pemberian obat di setiap desa berdasarkan kecamatan.

5.2.4 Analisis Persebaran Penyakit Sapi Berdasarkan Jenis Penyakit / Desa

Analisis persebaran penyakit sapi dapat dilihat di setiap kecamatan yang ada di malang pada setiap bulan. Fitur ini menganalis penyakit tertentu, tergantung dari pilihan user pada kecamatan, bulan dan tahun tertentu. Nantinya dengan fitur ini, akan disajikan informasi mengenai persebaran penyakit pada tiap kecamatan. Hasil analisis lebih detail dapat dilihat pada tabel dibawah ini

Tabel 5. 9 Analisis penyakit / desa

No	Nama obat	Bulan	Kecamatan	Screenshoot Aplikasi	Keterangan
1	BEF	Mei	Wagir		Pada bulan Mei , jumlah persebaran penyakit BEF di kecamatan Wagir mencapai 20 kasus . Desa dengan persebaran penyakit terbanyak terbanyak yaitu ada di desa sumberpucung dengan 12 kasus, kemudian desa blayu dengan 5 kasus dan yang terakhir di desa wonoayu dengan 3 kasus

Dari tabel diatas dapat dianalisis pada bulan Mei dan tahun 2012 diketahui ada 3 desa yang terkena penyakit *BEF* di kecamatan wagir . Dengan menggunakan aplikasi *strategic mobile dashboard* dapat mengetahui tingkat persebaran penyakit per desa yang memudahkan kepala dinas peternakan ketika ingin mengetahui proporsi jumlah persebaran penyakit di setiap desa berdasarkan kecamatan.

BAB 6

KESIMPULAN DAN SARAN

Pada bab ini diuraikan mengenai kesimpulan yang dapat ditarik dari pengerjaan Tugas Akhir ini. Selain itu juga diuraikan mengenai saran-saran yang dapat diperhatikan untuk pengembangan aplikasi kedepannya.

6.1 Kesimpulan

1. Aplikasi *Mobile Strategic Dashboard* dapat digunakan untuk membantu Dinas Peternakan memvisualisasikan data sehingga Dinas Peternakan dapat menentukan daerah mana yang harus mendapatkan penanganan dengan segera, dan formulasi kebijakan yang tepat sasaran. Aplikasi ini juga dapat digunakan untuk memonitor jumlah penyakit sapi per kecamatan, jumlah penyakit sapi per desa, jumlah obat per kecamatan, dan jumlah obat per desa.
2. Aplikasi *Mobile Strategic Dashboard* ini dibuat dengan bahasa pemrogaman java berbasis android di sisi client dan di sisi server menggunakan database *Mysql* berformat *JSON*.
3. Aplikasi Mobile Strategic Dashboard ini berjalan optimal dengan menggunakan system operasi *Jelly Bean* dengan ukuran layar sebesar 4 inch.
4. Untuk jenis penyakit mastitis dan dismatosis penanganan lebih ditekankan pada kecamatan pujon, sedangkan untuk BEF pada kecamatan Wajak dikarenakan dikarenakan jumlah kasus ternak sapi yang terjangkit cukup tinggi di kecamatan tersebut.

5. Aplikasi Mobile Strategic Dashboard berbasis android ini telah diuji coba dengan test case dan keseluruhan sesuai dengan use case yang direncenakan. Selain itu digunakan validasi pakar sebagai metode validasi bahwa aplikasi sudah sesuai dan mampu menjalankan fungsinya
6. Penggunaan *web service* berformat json memudahkan pertukaran data dari client ke server maupun dari server ke client dikarenakan data yang dikirimkan berbentuk *array* sehingga memiliki kesamaan dalam pengiriman dan penerimaan datanya.

6.2 Saran

1. Terkait dengan jumlah penyakit mastitis dan dimastosis yang terbanyak pada tahun 2012 terjadi di kecamatan pujon, maka dinas peternakan dapat melakukan penanganan khusus dan melakukan penambahan jumlah jenis obat sesuai dengan penyakit tersebut.
2. Menambah data mengenail hasil diagnosa penyakit untuk periode 2013 agar Aplikasi Mobile Strategic Dashboard ini lebih efisien dan realtime terhadap perubahan data yang terjadi.
3. Aplikasi Mobile Strategic Dashboard dapat ditambahkan berbagai pilihan dashboard yang sesuai keinginan dari user seperti piechart ataupun dalam bentuk grafik.

DAFTAR PUSTAKA

Claudio Maia, L. (2012). *Evaluating Android OS for Embedded Real-Time System.* Porto: Scholl of Engineering of the Polythechnic Insitute of Porto.

Eckerson, W. W. (2007). *Best Practices in Operational BI.* www.tdwi.org.

J.E.N.I. (2010). *blog.unsri.ac.id:*. Dipetik January 20, 2012, dari http://blog.unsri.ac.id/userfiles/JENI-J2ME-Bab01Pengembangan%20Aplikasi%20Mobile(2).pdf

JSON. (2013, July 23). *Introducing JSON.* Retrieved from www.json.org: www.json.org

Lacey, L. D. (2010). *A study on present and future of Google's Android.*

Listiana, N. (2011). *Penerapan Algortima Rough Set Untuk Deteksi Dan Penanganan Dini Penyaklit Sapi.* Surabaya: Tugas Akhir Jurusan Sistem Informasi.

Malik, S. (2005). *Enterprise Dashboard: Design and Best Practice for IT.* New Jersey: Willey.

Safaat, N. (2012). *Pemrogaman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android.* Bandung: Penerbit Informatika.

Technopedia. (2010). *Technopedia*. Dipetik January 2013, dari http://www.techopedia.com/definition/2953/mobileapplication-mobile-app

Tjahajati, I. (2012). *Berbagai Penyakit Pada Sapi.* Yogyakarta: PT. Citra Aji Parama .

Ardianto, A. (2012). *Pembuatan Sistem Pakar Untuk Pendeteksian Dan Penanganan Dini Pada Penyakit Sapi Berbasis Mobie Android Dengan Kajian Kinerja Teknik Knowledge Representation.* Surabaya: Tugas Akhir Sistem Informasi.

Nichol, S. (2004, November 3). *nusoapintro.htm.* Retrieved from http://www.scottnichol.com:http://www.scottnichol.com/nusoapintro.htm

Harvey, G. (2010). *http://www.dummies.com/.* Dipetik April 8, 2013, dari http://www.dummies.com/: http://www.dummies.com/how-to/content/how-to-choose-the-right-chart-type-in-excel-2010.html

http://achartengine.org/. (2013, april monday). Diambil kembali dari http://achartengine.org/

Utama, Y. (2010). *Teknik Pemrograman Web Service PHP dengan Menggunakan SOAP dan WSDL.* Yogyakarta: Andi Offset.

BIODATA PENULIS

Fahri Reza dilahirkan di Probolinggo pada 08 April 1991 sebagai anak pertama dari 3 bersaudara. Sampai dengan saat ini penulis telah menempuh pendidikan formal di SDN Sukabumi II, SMPN 1 Probolinggo, SMAN 1 Probolinggo, kemudian melanjutkan pendidikan S1 di jurusan Sistem Informasi Institut Teknologi Sepuluh Nopember Surabaya dengan bidang minat Business Intelligence/ Decision Support System (BI/DSS).

Selama menjadi mahasiswa, penulis aktif di organisasi kemahasiswaan Sistem Informasi (Staf Departemen Pengembangan Minat Bakat HMSI 2010-2011, Ketua UKM KOPMA ITS 2010-2011) Penulis dapat dihubungi melalui email di fahrireza09@gmail.com

PETUNJUK PENGGUNAAN APLIKASI STRATEGIC MOBILE DASHBOARD

1. Install Aplikasi pada smartphone Android anda
2. Pilih Kategori dashboard, pada aplikasi ini terdapat 2 pilihan dashboard yaitu berdasarkan penyakit dan berdasarkan penangananan
3. Untuk melihat penyakit per kecamatan, maka pilih kategori penyakit lalu pilih kecamatan. Lalu pilih nama bulan yang akan ditampilkan.
4. Untuk melihat penyakit per desa, maka pilih kategori penyakit lalu pilih desa. Kemudian pada pilihan,pilih kecamatan untuk melihat persebaran penyakit pada tingkat desa.
5. Untuk melihat penanganan per kecamatan, maka pilih kategori penanganan lalu pilih kecamatan. Lalu pilih nama bulan yang akan ditampilkan.
6. Untuk melihat penyakit per desa, maka pilih kategori penanganan lalu pilih desa. Kemudian pada pilihan,pilih kecamatan untuk melihat persebaran penanganan pada tingkat desa.

Hasil wawancara dengan Dinas Peternakan dan Kesehatan kabupaten Malang

1. Bagaimana metode pengelolaan data persebaran penyakit yang ada di Dinas Peternakan dan Kesehatan kebupaten Malang?

   Selama ini pengelolaan data penyakit sapi dikelola dengan bantuan perangkat lunak Ms. Excel. Sementara untuk pengumpulan data dari setiap kecamatan dilakukan secara manual dengan mencatat di excel kemudian di serahkan ke Dinas Peternakan secara langsung

2. Siapa saja yang berperan dan terlibat dalam mengelola data persebaran penyakit sapi ?
   - *Kepala Dinas(monitoring)*
   - *Peternak (menginputkan data)*

3. Masalah apa saja yang dihadapi oleh Dinas Peternakan dan Kesehatan Hewan dalam mengelola data tersebut ?

   Dengan cara sebelumnya kami mendapatkan beberapa kesulitan, diantaranya:
   - *Kesulitan dalam proses mengumpulkan dari setiap peternak yang ada di tiap-tiap kecamatan di Kabupaten Malang.*
   - *Pendokumentasian data masih terpisah-pisah dan belum terintegrasi,sehingga data ada yang tidak terekap dengan baik*
   - *Kepala dinas tidak bisa melihat data persebaran penyakit secara real-time*

4. Apa yang diharapkan dari perangkat lunak ini?

Perangkat lunak ini harus bisa mengatasi masalah-masalah tersebut seperti dengan peternak bisa pmenginputkan dalam database sehingga data tersebut dapat dengan mudah dikelola

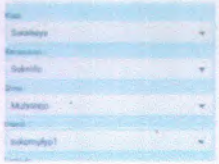
5. Bagaimana proses pengidentifikasian suatu daerah dikatakan rawan terhadap wabah penyakit sapi ? Apakah ada ukuran/standar minimal daerah dikatakan rawan terhadap penyakit sapi ?

Untuk pengkategorian daerah dikatakan rawan terhadap penyakit sapi bukan didasarkan pada jumlah atau banyaknya sapi perah yang sakit, akan tetapi didasarkan pada jenis penyakit sapi itu sendiri, misalkan dalam suatu daerah terdapat 1 sapi yang terkena Anthrax maka daerah tersebut dapat diakatakan rawan, dan harus segera dilakukan penanganan.

LAMPIRAN C

Hasil Validasi Sistem Oleh Dinas Peternakan kabupaten Malang

Form Evaluasi *Prototype Strategic Mobile Dashboard*

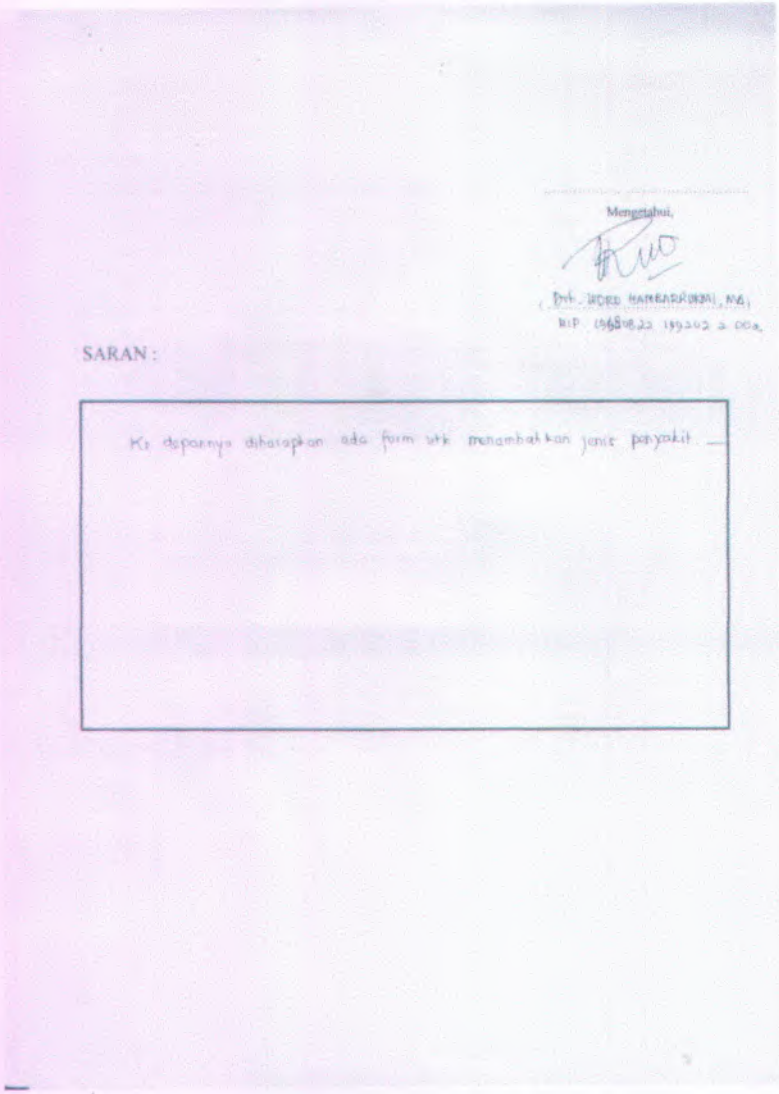
No	Jenis Tes	Hasil yang Diharapkan	Uji hasil dengan sistem	Keterangan
1	Menginputkan data hasil diagnosis penyakit dan obat yang diberikan	Terdapat inputan berupa *radio button* untuk memilih jenis penyakit dan obat yang akan diberikan		√
2	Menginputkan data dengan *dependent spinner* untuk menampilkan lokasi (kota, kecamatan , desa, dan dusun) dari sapi yang sakit	Muncul nama kecamatan ang sesuai dengan kota yang dipilih, muncul nama desa yang sesuai dengan kecamatan yang dipilih, muncul nama dusun yang sesuai dengan nama desa yang dipilih		√
3	Mendapatkan nilai GPS dari lokasi sapi yang sakit	Muncul nilai *latitude* dan *longitude* ketika user berhasil login dan masuk ke halaman input data		√
4	Memilih Pilihan Dashboard	Pada aplikasi *Strategic Mobile Dashboard* terdapat dua pilihan Dashboard		√
5	Memilih pilihan Dashboard berdasarkan jenis penyakit	Pada Aplikasi *Strategic dashboard* ada dua pilihan , salah satunya adalah berdasarkan jenis penyakit sapi		√
6	Memilih pilihan Dashboard	Muncul		

Gambar 6. 1 Form validasi dari Dinas Peternakan 1

Form Evaluasi *Prototype Strategic Mobile Dashboard*

	berdasarkan jenis penanganan	tampilan berdasarkan penanganan		√
7	Pada pilihan jenis penyakit, terdapat pilihan penyakit, bulan, dan tahun(misal penyakit **mastitis**, bulan **januari** , tahun **2013**)	Kemudian akan muncul tampilan dashboard dalam bentuk chat sesuai dengan pemilihan yang dipilih		√
8	Pada pilihan jenis penanganan, terdapat pilihan obat , bulan, dan tahun(misal obat **Anthelmentik**, bulan **Januari** , tahun **2013**)	Kemudian akan muncul tampilan dashboard dalam bentuk chat sesuai dengan pemilihan yang dipilih		√

Gambar 6. 2 Form validasi dari dinas Peternakan 2



Gambar 6. 3 Form Validasi Dinas Peternakan 3

BERITA ACARA HASIL WAWANCARA

Hari Senin, 08 Juli 2013 bertempat di Dinas Peternakan dan Kesehatan Hewan Kabupaten Malang telah dilakukan wawancara (hasil wawancara terlampir) sebagai tindak lanjut dalam proses identifikasi kebutuhan dan pengujian *prototype* aplikasi untuk penelitian tugas akhir oleh Saudara Mochammad Adji Firmansyah dan Fahri Reza, mahasiswa Jurusan Sistem Informasi Fakultas Teknologi Informasi Institut Teknologi Sepuluh Nopember Surabaya.

Materi wawancara terdiri dari :

1. Identifikasi Kebutuhan
2. Pengujian *prototype* GIS di Android
3. Pengujian *prototype Strategic Mobile Dashboard* di Android

Dari keseluruhan tahapan wawancara dan pemeriksaan hasil wawancara, dinyatakan bahwa hasil wawancara dan pengujian sistem tersebut telah dilaksanakan dan sesuai.

Demikian Berita Acara ini dibuat sesuai dengan yang telah dilaksanakan.

Mahasiswa 1.	Mahasiswa 2.
M Adji Firmansyah (NRP 5209100003)	Fahri Reza (NRP 5209100093)
Menyetujui, [illegible] NIP. [illegible]	

Gambar 6. 4 Berita acara Dinas Peternakan