



**PERANCANGAN SISTEM INFORMASI FORMULASI RANSUM AYAM
PETELUR MENGGUNAKAN METODE TRIAL & ERROR BERBASIS
ANDROID**

TUGAS AKHIR

Diajukan kepada Jurusan Manajemen Informatika

Untuk Memenuhi Salah Satu Syarat Guna Mencapai Gelar Ahli Madya

Dalam Bidang Manajemen Informatika

Oleh :

BRIANS PRATAMA

NIM. 14 205 019

PROGRAM DIPLOMA D.III MANAJEMEN INFORMATIKA

FAKULTAS EKONOMI DAN BISNIS

ISLAM INSTITUT AGAMA ISLAM NEGERI (IAIN)

BATUSANGKAR

2018

SURAT PERNYATAAN KEASLIAN

Saya yang bertanda tangan di bawah ini:

Nama : BRIANS PRATAMA
NIM : 14 205 019
Tempat/ Tanggal Lahir : Batuasangkar/ 14 Desember 1995
Fakultas : Ekonomi dan Bisnis Islam
Jurusan : Manajemen Informatika

Menyatakan dengan sesungguhnya bahwa Tugas Akhir saya yang berjudul **“PERANCANGAN SISTEM INFORMASI FORMULASI RANSUM AYAM PETELUR MENGGUNAKAN METODE TRIAL & ERROR BERBASIS ANDROID”** adalah benar karya saya sendiri bukan plagiat kecuali yang dicantumkan sumbernya.

Apabila di kemudian hari terbukti bahwa Tugas Akhir ini plagiat, maka saya bersedia menerima sanksi sesuai dengan ketentuan perundang-undangan yang berlaku. Demikian pernyataan ini saya buat dengan sebenarnya untuk digunakan sebagaimana mestinya.

Batusangkar, 5 Februari 2018

Saya yang menyatakan



Brians Pratama

NIM. 14 205 019

PERSETUJUAN PEMBIMBING

Pembimbing penulis Tugas Akhir atas Nama : **BRIANS PRATAMA**,
Nim : 14 205 019 dengan judul, “ **PERANCANGAN SISTEM INFORMASI
FORMULASI RANSUM AYAM PETELUR MENGGUNAKAN METODE
TRIAL & ERROR BERBASIS ANDROID**” memandang bahwa Tugas Akhir
yang bersangkutan telah memenuhi persyaratan ilmiah dan dapat disetujui untuk
dilanjutkan ke Sidang Munaqasyah.

Dengan persetujuan ini diberikan untuk dapat dipergunakan sebagaimana
mestinya.

Batusangkar, Februari 2018

**Ketua Jurusan
Manajemen Informatika,**

Pembimbing,


Iswandi, M. Kom
Nip. 19700510 200312 1004


Adriyendi, M. Kom
Nip. 19770127 200912 1 002

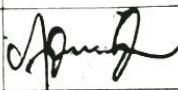
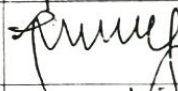

Mengetahui,
**Dekan Fakultas Ekonomi Dan Bisnis Islam
IAIN Batusangkar**



Dr. Ika Atsani, SH, M.Hum
NIP. 19750303 199903 1 004

PENGESAHAN TIM PENGUJI

Tugas Akhir yang berjudul “PERANCANGAN SISTEM INFORMASI FORMULASI RANSUM AYAM PETELUR MENGGUNAKAN METODE TRIAL & ERROR BERBASIS ANDROID” oleh BRIANS PRATAMA Nim. 14 205 019, telah diujikan pada Sidang Munaqasyah Institut Agama Islam Negeri (IAIN) Batusangkar, pada hari Rabu tanggal 14 Februari 2018 dan dinyatakan telah dapat diterima sebagai salah satu syarat untuk mencapai gelar Ahli Madya Program Diploma III (D.III) Manajemen Informatika.


No.	Nama Penguji	Jabatan	Tanggal	Tanda Tangan
1.	Adriyendi, M.Kom NIP. 19770127 200912 1 002	Ketua Sidang	20/2/2018	
2.	Iswandi, M.Kom NIP. 19700510 200312 1 004	Anggota	20/2/2018	
3.	Fitra Kasma Putra, M Kom NIP. 19850207 201503 1 004	Anggota	20/2/2018.	

Mengetahui,

Dekan Fakultas Ekonomi dan Bisnis Islam

IAIN Batusangkar




Dr. Ulya Atsani, S.H., M.Hum
19750303 199903 1 004

ABSTRAK

**JUDUL TUGAS AKHIR : PERANCANGAN SISTEM INFORMASI
FORMULASI RANSUM AYAM PETELUR
MENGUNAKAN METODE TRIAL &
ERROR BERBASIS ANDROID**

NOMOR INDUK : 14 205 019

JURUSAN : MANAJEMEN INFORMATIKA

DOSEN PEMBIMBING : ADRIYENDI, M.Kom

Penelitian ini dilakukan bertujuan untuk mempermudah para peternak ayam petelur dalam melakukan proses penghitungan formulasi ransum mereka sendiri. Penulis menemukan adanya beberapa kesulitan yang dialami peternak dalam menentukan ransum mereka yaitu belum tersedianya aplikasi yang dapat menghitung ransum dengan mudah dan cepat. Hal ini mengakibatkan peternak membutuhkan waktu yang lama dalam menentukan formulasi ransum yang sesuai. Dengan memanfaatkan perkembangan *Teknologi Android* saat ini, diharapkan dapat membantu proses penghitungan ransum dengan mudah dan cepat. Alat bantu perancangan sistem menggunakan UML (Unified Modeling Language), Sedangkan perangkat lunak yang digunakan dalam sistem adalah menggunakan bahasa pemrograman Java dengan Eclipse sebagai perantara. Dengan memanfaatkan sistem komputerisasi diharapkan dapat membantu proses pendataan dan dengan memanfaatkan pemrograman android sebagai software aplikasi diharapkan dapat menggantikan cara yang kurang efektif dan efisien serta diharapkan dapat mempermudah pengolahan dan penghitungan formulasi ransum

***Kata kunci* : Sistem Informasi, Ransum, Java, android**

KATA PENGANTAR



Puji syukur penulis haturkan atas kehadiran Allah SWT, atas segala rahmat dan karunia-Nya serta kesempatan menggenggam ilmu, sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul **“PERANCANGAN SISTEM INFORMASI FORMULASI RANSUM AYAM PETELUR MENGGUNAKAN METODE TRIAL & ERROR BERBASIS ANDROID”**

Lantunan Salawat dan salam senantiasa terkirim untuk Rasulullah SAW sang revolusioner sejati yang telah membawa pelita penerang bagi umat manusia di muka bumi ini, yakni Nabi Muhammad SAW

Tugas Akhir ini penulis susun untuk memberikan sumbangan pemikiran kepada almamater serta memenuhi sebagian persyaratan mencapai derajat Diploma III Jurusan Manajemen informatika Institut Agama Islam Negeri Batusangkar

Dalam penulisan Tugas Akhir ini penulis banyak mendapat bimbingan dan bantuan baik moril maupun materil dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis menyampaikan ucapan terima kasih kepada:

1. Bapak Dr. H. Kasmuri, M.A, selaku Rektor Institut Agama Islam Negeri Batusangkar
2. Bapak Dr.Ulya Atsani, SH, M.Hum selaku Dekan Fakultas Ekonomi dan Bisnis Islam IAIN Batusangkar
3. Bapak Iswandi, M.Kom selaku Ketua Jurusan Manajemen Informatika IAIN Batusangkar
4. Bapak Adriyendi, M.Kom selaku Pembimbing yang telah membimbing penulis dalam menyelesaikan penulisan Tugas Akhir ini
5. Segenap Dosen serta Karyawan/I IAIN Batusangkar
6. Asgim (Ayahanda) dan Detafifianis (Ibunda) dan Keluarga besar yang selalu memberikan semangat, serta do'a yang tiada henti-hentinya

7. Seluruh teman-teman keluarga besar Jurusan Manajemen Informatika yang selalu bersama-sama mengukir kenangan indah selama mengikuti perkuliahan di IAIN Batusangkar
8. Pihak-pihak yang tidak dapat penulis sebutkan satu persatu, yang telah memberikan motivasi dan semangat serta sumbangan pemikirannya kepada penulis sehingga selesainya laporan ini.

Penulis sadar bahwasanya Tugas Akhir ini jauh dari kesempurnaan, oleh karena itu kritikan dan saran yang sifatnya membangun sangat penulis harapkan untuk kesempurnaan tugas akhir ini. Penulis juga berharap semoga penulisan tugas akhir ini memberikan manfaat kepada kita semua. Amin.

Akhirnya kepada Allah SWT jualah penulis bermohon dan bersujud semoga keikhlasan yang diberikan akan dibalas-Nya. ***Amin Ya Robbal'alamin.***

Batusangkar, Februari 2018

Penulis

DAFTAR ISI

HALAMAN JUDUL	
LEMBAR KEASLIAN DATA	
LEMBAR PESETUJUAN PEMBIMBING	
LEMBAR PENGESAHAN PENGUJI	
ABSTRAK	i
KATA PENGANTAR.....	ii
DAFTAR ISI.....	iv
DAFTAR GAMBAR.....	vii
DAFTAR TABEL.....	viii
BAB I PENDAHULUAN	1
A. Latar Belakang Masalah.....	1
B. Identifikasi Masalah	2
C. Batasan Masalah.....	2
D. Rumusan Masalah	2
E. Tujuan Penelitian	3
F. Kegunaan Penelitian.....	3
G. Metodologi Penelitian	4
H. Sistematika Penulisan.....	4
BAB II LANDASAN TEORI	6
A. Konsep Dasar Sistem Informasi.....	6
1. Pengertian sistem.....	6
2. Pengertian Informasi.....	6
3. Pengertian Sistem Informasi.....	7
4. Komponen Sistem Informasi	7
B. Perancangan	8
1. Pengertian Perancangan Sistem.....	8
2. Sasaran Perancangan Sistem	9
C. Trial And Error.....	9
D. Ransum.....	10

1.	Pengertian Ransum	10
2.	Tehnik Penghitungan Ransum.....	10
E.	Android	12
1.	Pengertian	12
2.	Sejarah <i>Android</i>	13
3.	Arsitektur <i>Android</i>	15
b.	Linux Kernel	17
F.	Perangkat Lunak Pembangun Sistem.....	17
1.	<i>Java Software Development Kit</i> (SDK Java).....	17
2.	<i>Eclipse Software Development Kit</i> (SDK <i>Eclipse</i>).....	18
3.	<i>Android Software Development Kit</i> (SDK <i>Android</i>)	18
4.	<i>Android Package</i> (APK)	18
5.	<i>Android Virtual Devices</i> (AVD).....	18
6.	Emulator	19
7.	Unified Modeling Language (UML)	19
1.	Use Case Diagram.....	20
2.	Sequence Diagram	21
3.	Collaboration Diagram.....	24
4.	Activity Diagram.....	24
5.	Class Diagram	26
6.	Component Diagram	27
8.	Object Oriented Programming (OOP)	29
9.	Java.....	30
10.	Eclipse.....	30
BAB III ANALISA DAN PERANCANGAN.....		32
A.	Analisa Actor	32
1.	<i>Unified Modeling Language</i> (UML)	32
2.	Use case Diagram	32
3.	Sequence Diagram.....	33
4.	Activity Diagram	38
5.	Class Diagram.....	39

6.	Collaboration Diagram	39
7.	Struktur Program	40
B.	Analisa Rancangan.....	40
a.	Rancangan Beranda	40
b.	Rancangan halaman Menu Utama.....	41
c.	Rancangan halaman Hitung ransum	42
d.	Rancangan output Halaman Hitung Ransum.....	42
e.	Rancangan Halaman Informasi	43
f.	Rancangan halaman cek informasi pakan	44
g.	Rancangan Sub Item Cek Informasi Pakan	44
h.	Rancangan halaman Pakan ke Ternak	45
i.	Rancangan output halaman Pakan Ke Ternak.....	46
j.	Rancangan halaman Ternak Ke Pakan	46
k.	Rancangan output halaman Ternak Ke Pakan.....	47
l.	Rancangan halaman About	47
BAB IV	Kesimpulan dan Saran.....	48
A.	Kesimpulan	48
B.	Saran.....	48

DAFTAR PUSTAKA

FLOWCHART

LISTING PROGRAM

DAFTAR GAMBAR

Gambar 2.1	Arsitektur Andoid	17
Gambar 2.2	Eclipse ADT	32
Gambar 3.1	Use case Diagram	33
Gambar 3.2	Sequence Diagram Hitung Ransum.....	35
Gambar 3.3	Sequence Diagram About.....	35
Gambar 3.4	Sequence Diagram Informasi	36
Gambar 3.5	Sequence Diagram Informasi Pakan	36
Gambar 3.6	Sequence Diagram Pakan Ke Ternak.....	37
Gambar 3.7	Sequence Diagram Ternak Ke Pakan.....	38
Gambar 3.8	Activity Diagram User	39
Gambar 3.9	Class diagram	40
Gambar 3.10	Collaboration Diagram	40
Gambar 3.11	Struktur Program	41
Gambar 3.12	Halaman Beranda	41
Gambar 3.13	Halaman Menu Utama.....	42
Gambar 3.14	Hitung Ransum.....	43
Gambar 3.15	Output Halaman Hitung Ransum	44
Gambar 3.16	Halaman Informasi	44
Gambar 3.17	Halaman Cek Informasi Pakan.....	45
Gambar 3.18	Sub Item Cek Informasi Pakan.....	46
Gambar 3.19	Halaman Pakan Ke Ternak.....	46
Gambar 3.20	Output Halaman Pakan Ke Ternak.....	47
Gambar 3.21	Halaman Ternak Ke Pakan.....	47
Gambar 3.22	Output Halaman Ternak Ke Pakan.....	48
Gambar 3.21	Halaman About.....	49

DAFTAR TABEL

Tabel 2.1 Tabel penghitungan Ransum.....	10
Tabel 2.2 Daftar sibol – simbol UML.....	19

BAB I

PENDAHULUAN

A. Latar Belakang Masalah

Ransum adalah susunan bahan pakan yang seimbang dan tepat untuk ternak, sehingga mencukupi kebutuhan nutrisinya dalam satu hari. Keberadaan ransum dalam dunia peternakan sangatlah penting, karena dapat mempengaruhi hasil produksi dari ternak itu sendiri. Setiap peternakan baik peternak yang masih ulung ataupun peternak yang sudah maju tidak luput dari yang namanya ransum. Namun pada dasarnya peternak yang sudah menerapkan penghitungan ransum bagi ternaknya hanya peternak yang tergolong maju saja, sedangkan peternak yang masih ulung tidak melakukan perhitungan ransum tersebut, dengan alasan perhitungan ransum yang tergolong sangat sulit dan memerlukan keterampilan dalam menghitungnya. Perlu adanya metode penyusunan ransum yang tepat sehingga tercipta komposisi yang baik dan benar. Bila nutrisi ternak tercapai dengan baik, otomatis produktivitas akan baik pula. Usaha peternakan yang baik memiliki rancangan ransum yang baik untuk ternaknya. Diusahakan tiap peternakan memiliki rancangan tersebut sendiri, tidak hanya bergantung pada komposisi complete feed pabrik. (Buku-PBMT:2012)

Dengan memiliki komposisi ransum sendiri, peternak dapat mengambil bahan pakan yang potensial di sekitarnya sehingga biaya untuk pakan dapat diminimalisir. Seperti limbah pasar, limbah rumah makan, dsb. Sampai sekarang penghitungan ransum ayam ini masih hanya beberapa orang yang dapat melakukannya dan dengan cara yang beragam pula sehingga, hasil dan waktu yang dibutuhkan tidak dapat digunakan dengan semestinya dan tidak semua orang dapat menghiung formulasi ransum mereka sendiri. Oleh karena itu penulis bermaksud untuk merancang sebuah aplikasi untuk mempermudah para peternak dari yang

masih pemula sampai peternak yang sudah mahir di bidangnya. Perancangan Sistem Informasi Formulasi Ransum Ayam Petelur akan sangat membantu peternak ayam dalam menentukan formulasi ransum mereka sendiri.

Berdasarkan uraian diatas penulis mencoba membahasnya dalam tugas akhir yang berjudul **“Perancangan Sistem Informasi Formulasi Ransum Ayam Petelur Menggunakan Metode Trial & Error berbasis Android”**.

B. Identifikasi Masalah

Berdasarkan latar belakang masalah diatas, maka dapat diidentifikasi beberapa masalah sebagai berikut :

1. Sulitnya menentukan ransum ternak karena tidak semua orang memiliki keterampilan dalam menghitungnya.
2. Perhitungan dengan menggunakan Microsoft Exel masih tergolong sangat sulit bagi *User* yang tidak memiliki basic dalam peternakan dan aplikasi perhitungan Microsoft Exel.
3. Belum tersedianya sistem informasi yang dapat menghitung formulasi ransum dengan mudah.

C. Batasan Masalah

Agar tidak terjadi pembahasan masalah yang menyimpang dari judul, maka penulis memberi batasan masalah sebagai berikut :

1. Ransum yang akan dihitung yaitu ransum ayam petelur.
2. Metode yang akan digunakan dalam menghitung ransum yaitu metode *Trial And Error*.

D. Rumusan Masalah

Berdasarkan latar belakang masalah yang penulis jabarkan, maka dapat dirumuskan beberapa masalah sebagai berikut :

1. Bagaimana membuat sistem informasi untuk menghitung ransom dengan mudah dan semua orang dapat menghitung ransom mereka sendiri.
2. Bagaimana membuat sistem informasi perhitungan formulasi ransom yang dapat di cerna dengan mudah oleh *User*.
3. Bagaimana membuat sebuah sistem informasi penghitungan formulasi ransom yang proses penghitungannya dapat dilakukan kapanpun dan dimana saja.

E. Tujuan Penelitian

Adapun tujuan yang ingin dicapai dari permasalahan ini adalah untuk:

1. Merancang sebuah sistem informasi yang dapat melakukan perhitungan ransom tanpa harus berbelit-belit.
2. Merancang sebuah sistem informasi bagi *User* untuk dapat berbagi data formulasi dengan *User* lainnya.
3. Merancang sebuah sistem informasi yang semua orang dapat menggunakannya.

F. Kegunaan Penelitian

Kegunaan penelitian ini adalah :

1. Sebagai sarana menerapkan dan pengembangan ilmu yang telah penulis dapatkan selama masa perkuliahan.
2. Sebagai tambahan referensi bagi pembaca yang akan melakukan penelitian lebih lanjut.
3. Dapat memberikan sumbangan pemikiran untuk masyarakat, sehingga dapat merasakan manfaat dari perkembangan teknologi informasi.
4. Sebagai salah satu syarat untuk memperoleh gelas Ahli Madya (Amd) Program Diploma III Manajemen Informatika pada Institut Agama Islam Negeri (IAIN) Batusangkar.

G. Metodologi Penelitian

Dalam penulisan tugas akhir ini, penulis menggunakan beberapa metode antara lain :

1. Penelitian Lapangan (*Field Research*)

Dalam penelitian ini penulis mendapat data langsung dari hasil peninjauan ke lapangan, yaitu pada Rumah-rumah Kosan Khusus Puteri di Selingkup Kampus IAIN Batusangkar dan mengadakan wawancara pada bagian yang berhubungan langsung terhadap pembahasan yang penulis teliti.

2. Penelitian Perpustakaan (*Library Research*)

Penelitian ini dilakukan dengan cara mengumpulkan bahan-bahan dari buku-buku, karangan ilmiah, ataupun tulisan yang berhubungan dengan permasalahan yang dibahas dalam penelitian ini.

3. Penelitian Laboratorium (*Laboratory Research*)

Penulis melakukan pengolahan data yang telah dikumpulkan dengan menggunakan komputer sebagai alat bantu dalam pembuatan tugas akhir ini

H. Sistematika Penulisan

Sistematika penulisan laporan tugas akhir ini dibagi dalam IV BAB yang disusun sebagai berikut :

1. BAB I PENDAHULUAN

Bab ini merupakan penguraian mengenai latar belakang masalah, identifikasi masalah, batasan dan rumusan masalah, tujuan dan kegunaan penelitian, metodologi penelitian, sistematika penulisan.

2. BAB II LANDASAN TEORI

Bab ini berisi teori yang diambil dari buku-buku panduan dan referensi lainnya.

3. BAB III ANALISA DAN PERANCANGAN

Bab ini membahas analisa sistem yang sedang berjalan dan rancangan sistem yang diusulkan.

4. BAB IV PENUTUP

Bab ini berisi kesimpulan yang didapat selama pembuatan laporan tugas akhir serta saran-saran yang akan menjadi masukan bagi perkembangan sistem selanjutnya.

BAB II

LANDASAN TEORI

A. Konsep Dasar Sistem Informasi

1. Pengertian sistem

Menurut Jogiyanto (2005) sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. Sedangkan menurut Gordon B. Davis dalam Tata Sutabri (2005) sistem bisa berupa abstrak atau fisis. Sistem yang abstrak adalah susunan yang teratur dari gagasan-gagasan atau konsepsi yang saling bergantung. Sistem yang bersifat fisis adalah serangkaian unsur yang bekerja sama untuk mencapai suatu tujuan. Dari defenisi diatas, dapat disimpulkan bahwa sistem merupakan setiap kumpulan komponen atau subsistem yang saling berhubungan dan saling tergantung satu sama lain yang dirancang untuk mencapai suatu tujuan.

2. Pengertian Informasi

Menurut Robert N. Anthony dan John Dearden dalam Jogiyanto (2005) Informasi dapat didefenisikan data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. sumber dari informasi adalah data. Data merupakan kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Informasi yang baik harus memenuhi kriteria sebagai berikut:

a. Akurat (*Accurate*)

Informasi yang diperoleh harus bebas dari kesalahan-kesalahan dan tidak menyesatkan serta harus mencerminkan maksudnya. Informasi diharuskan akurat karena dari sumber informasi sampai ke penerima informasi kemungkinan banyak timbul gangguan (*noise*) yang dapat merubah atau merusak informasi tersebut.

b. Tepat Waktu (*timeliess*)

Informasi yang datang ada penerimanya tidak boleh terlambat. Informasi yang telah usang tidak akan mempunyai nilai, hal ini disebabkan karena informasi

merupakan landasan dalam pengambilan keputusan, jika terlambat maka akan berakibat fatal pada suatu organisasi, instansi maupun perusahaan.

c. Relevan (*Relevance*)

Informasi tersebut bermanfaat untuk pemakainya. Relevansi informasi untuk tiap-tiap orang dengan yang lainnya berbeda. Karena informasi yang baik hanya akan dihasilkan oleh data yang baik dengan pemrosesan data yang tepat.

3. Pengertian Sistem Informasi

Menurut Robert A. Laitch dan K.Roscoe Bavis dalam Jogiyanto (2005) sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

Sedangkan menurut O'Barian dalam Yakub (2005) menyebutkan bahwa sistem informasi merupakan kombinasi teratur dari orang-orang, perangkat keras (hardware), perangkat lunak (software), jaringan komunikasi, dan sumber data yang mengumpulkan, mengubah, dan menyebarkan informasi dalam sebuah organisasi.

Definisi umum dari sistem informasi adalah sebuah sistem yang terdiri atas rangkaian subsistem informasi terhadap pengolahan data untuk menghasilkan informasi yang berguna dalam pengambilan keputusan.

4. Komponen Sistem Informasi

Sistem informasi terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan (*Building Block*), dimana masing-masing blok ini saling berinteraksi satu sama lainnya membentuk satu kesatuan untuk mencapai tujuannya.

Adapun blok-blok tersebut adalah sebagai berikut:

a. Blok Masukan (*Input Block*)

Meliputi metode-metode dan media untuk menangkap data yang akan dimasukkan, dapat berupa dokumen-dokumen dasar.

b. Blok Model (*Model Block*)

Terdiri dari kombinasi prosedur, logika dan model matematika yang berfungsi memanipulasi data untuk menghasilkan keluaran tertentu.

c. Blok Keluaran (*Output Block*)

Berupa keluaran dokumen dan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

d. Blok Teknologi (*Technology Block*)

Untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran serta membantu pengendalian dari sistem secara keseluruhan.

e. Blok basis data (*Database Block*)

Merupakan kumpulan data yang berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer serta perangkat lunak untuk memanipulasinya.

f. Blok Kendali (*Controls Block*)

Meliput masalah pengendalian yang berfungsi mencegah dan menangani kesalahan/kegagalan sistem. (Sutabri, 2005)

B. Perancangan

1. Pengertian Perancangan Sistem

Menurut Robert J. Verzello dan John Reuter III dalam Jogiyanto (2005) perancangan sistem merupakan Pendefenisian dari kebutuhan- kebutuhan fungsional dan persiapan untuk rancang bangun implementasi menggambarkan bagaimana suatu sistem di bentuk. Sedangkan menurut Jhon Burch dan Gary Grudnitski dalam Jogiyanto (2005) pengertian perancangan sistem adalah Penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam suatu kesatuan yang utuh dan berfungsi.

Perancangan sistem merupakan sekumpulan prosedur yang dilakukan untuk mengubah spesifikasi logis menjadi disain yang dapat diimplementasikan ke sistem komputer organisasi. (Diana & Setiawati, 2011)

Jadi, Perancangan Sistem merupakan bagaimana cara mengorganisasikan sistem ke dalam subsistem-subsistem, serta alokasi subsistem-subsistem ke komponen-komponen perangkat keras, perangkat lunak, serta prosedur-prosedur.

2. Sasaran Perancangan Sistem

- a. Sasaran-sasaran yang akan dicapai dalam perancangan suatu sistem adalah:
- b. Perancangan sistem harus berguna, mudah dipahami dan nantinya mudah digunakan.
- c. Perancangan sistem harus dapat mendukung tujuan utama perusahaan.
- d. Perancangan sistem harus efisien dan efektif untuk dapat mendukung pengolahan data transaksi manajemen dan mendukung keputusan yang diambil oleh pihak manajemen.
- e. Perancangan sistem harus dapat mempersiapkan rancangan bangunan yang terinci untuk masing-masing komponen dari sistem informasi.

C. Trial And Error

Aktivitas coba-coba untuk belajar diteliti dengan seksama oleh ahli psikologi Amerika Serikat Edward L. Thorndike. Proses belajar dengan coba-coba disebut juga *trial and error*. Ketika hewan atau manusia memiliki keinginan atau menghadapi permasalahan (stimulus) maka mereka akan melakukan suatu respon terhadap stimulus tersebut. Semakin sering stimulus datang dan respon-respon diberikan maka akan muncul suatu proses belajar, yaitu mengetahui respon mana yang berhasil dan respon mana yang gagal.

Belajar melalui *trial and error* menuntut suatu kesabaran karena terjadi secara bertahap. Sebelum seseorang berhasil menguasai atau menetapkan respon terbaik maka seringkali ia dihadapkan pada kesalahan-kesalahan. Secara bertahap kita akan mengetahui bagaimana cara menyelesaikan sesuatu dengan cara yang

terbaik dan tidak mengulangi kesalahan-kesalahan yang pernah dilakukan. Karena itu ada istilah "kesalahan adalah cara untuk menjadi sukses." Istilah tersebut terutama terjadi melalui aktivitas *trial and error*.

D. Ransum

1. Pengertian Ransum

Ransum adalah susunan bahan pakan yang seimbang dan tepat untuk ternak, sehingga mencukupi kebutuhan nutrisinya dalam satu hari.

2. Tehnik Penghitungan Ransum

Menyusun ransum pada hakekatnya sama dengan mencampur bahan – bahan pakan yang dimiliki dengan perbandingan tertentu agar campuran tersebut dapat memenuhi kebutuhan ayam untuk berproduksi dengan baik. Ada berbagai cara yang dapat ditempuh untuk mencapai ini. Yang banyak dilakukan orang untuk penyusunan ransum ayam petelur saat ini dengan cara coba – coba (*Trial and Error*) cara ini relatif mudah bila bahan yang digunakan tidak banyak jenisnya. Ketepatan penyusunan ransum, dengan spesifikasi atau batasan yang banyak, sulit dilakukan metode ini. Disamping itu, pertimbangan harga minimum yang juga sulit dilakukan dengan metode ini. (B.Gunawan,1982)

Contoh tabel penghitungan ransum.

Tabel 2.1 Tabel Penghitungan Ransum

Sumber: <http://anuragaja.staff.ipb.ac.id/files/2012/04/Buku-PBMT.pdf>

NO	BAHAN PAKAN	PK	ME	LK	SK	KOMPOSISI	PK	ME	LK	SK
1	jagung halus	8,5	3370	2,47	1,97	26	2,21	876,2	0,6422	0,5122
2	tepung ikan	52,1	3080	5,36	0,99	25	13,025	770	1,34	0,2475
3	dedak halus	8,15	1630	6,48	21,89	16	1,304	260,8	1,0368	3,5024
4	minyak makan	0	8450	0	0	5	0	422,5	0	0
5	tepung tapioka	1,8	3720	1,3	1,8	17	0,306	632,4	0,221	0,306
6	feses kerbau	10,92	0	1,49	18,45	9	0,9828	0	0,1341	1,6605
Total						98	17,8278	2961,9	3,3741	6,2286
kebutuhan						100	18	2900	8	6

Berikut adalah keterangan dari istilah dalam tabel ransum diatas :

1. Protein Kasar (PK) : Semua ikatan yang mengandung Nitrogen (N), baik protein sesungguhnya(true protein) maupun zat-zat yang mengandung protein, tapi bukan protein.
2. Lemak Kasar (LK) : Kadar lemak dalam pakan dapat diketahui melalui ekstrak yang dilarutkan dalam ether, meski zat-zat lain juga larut di dalamnya. Karena itu, kadar lemak yang menjadi acuan perhitungan lebih tepat disebut lemak kasar(LK).
3. Metabolisme Energi (ME) : Nilai energi yang terhimpun pada zat-zat yang dapat dicerna dikurangi nilai energi yang keluar sebagai air kencing (urine) dan gas-gas usus.
4. Serat Kasar (SK) : komponen karbohidrat yang terdiri atas polisakarida yang tidak larut (selulosa dan hemiselulosa) serta lignin. Dalam ikatan lignoselulosa lignin memiliki koefisiensi cerna sangat rendah. Semakin tua tanaman, kandungan ligninnya semakin tinggi. Jerami padi termasuk bahan pakan dengan kandungan lignin tinggi sehingga sulit dicerna.

Berikut Langkah – langkah menghitung ransum ayam dengan menggunakan metode trial and error.

- a. Tentukan kebutuhan nitrien ayam yang diinginkan (kebutuhan nutrien dapat dilihat di buku panduan pemeliharaan yang dikeluarkan masing-masing breeder).
- b. Tentukan bahan pakan dengan mempertimbangkan pakan berkualitas sebagai sumber protein, energi, mineral, vitamin, mudah diperoleh, dapat dijamin kontinuitasnya, dan harga yang sesuai
- c. Cari data kandungan nutrien serta ada tidaknya faktor pembatas pada bahan baku yang akan kita gunakan. Tabel-tabel kandungan nutrien bahan baku pakan dari berbagai sumber dapat digunakan, tetapi harus dipertimbangkan yang paling dekat dengan kondisi sebenarnya.

- d. Mulai dilakukan penyusunan formula ransum yang diawali dengan penentuan penggunaan bahan baku pakan dalam jumlah keseluruhan 100.
- e. Hitung kandungan nutrisi setiap bahan baku pakan, terutama protein kasar, energi metabolis, serat kasar, mineral (Ca, P, Na dan Cl), asam amino lisin, metionin.
- f. Jumlah kandungan nutrisi dari setiap bahan baku pakan disesuaikan dengan kebutuhan nutrisi ternak. Bila belum mencukupi, lakukan perubahan penggunaan bahan baku ransum sampai dapat nutrisi yang sesuai (coba-coba)
- g. Bila kandungan nutrisi ransum sudah sesuai dengan kebutuhan nutrisi yang dikehendaki berarti formulasi sudah sesuai.

E. Android

1. Pengertian

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. *Android* merupakan generasi baru *platform mobile* yang memberikan kesempatan kepada pengembang untuk melakukan pengembangan sesuai dengan yang diharapkan. Sistem operasi yang mendasari *Android* merupakan lisensi di bawah naungan GNU, General Public License Versi 2 (GPLv2), yang biasadikenal dengan istilah *Copyleft*. Istilah *copyleft* ini merupakan lisensi yang setiap perbaikan oleh pihak ketiga harus terus jatuh di bawah *terms*.

Distribusi *Android* berada di bawah lisensi Apache Software (ASL/Apache2), yang memungkinkan untuk distribusi kedua atau seterusnya. Pengembang aplikasi *Android* diperbolehkan untuk mendistribusikan aplikasi mereka di bawah skema lisensi apapun yang mereka inginkan.

Pengembang memiliki beberapa pilihan dalam membuat aplikasi yang berbasis *Android*. Namun kebanyakan pengembang menggunakan *Eclipse* sebagai IDE untuk merancang aplikasi mereka. Hal ini dikarenakan *Eclipse* mendapat

dukungan langsung dari Google untuk menjadi IDE pengembangan aplikasi *Android*.

Aplikasi *Android* dapat dikembangkan pada berbagai sistem operasi, diantaranya adalah: Windows XP/Vista/7, Mac OS X (Mac OS X 10.48 atau yang lebih baru) dan Linux

2. Sejarah *Android*

Pada saat perilisan perdana *Android* pada tanggal 5 November 2007, *Android* bersama Open Handset Alliance mendukung pengembangan standar terbuka pada perangkat seluler tersebut. Di sisi lain, Google merilis kode-kode *Android* di bawah lisensi Apache. Sehingga terdapat dua jenis *distributor* sistem operasi *Android* yaitu yang mendapat dukungan penuh dari Google dan yang mendapat dukungan penuh dari Open Handset Distribution (OHD). Telepon selular pertama yang menggunakan sistem operasi *Android* adalah HTC Dream yang dirilis pada 22 Oktober 2008. Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja *Android* ARM Holdings, Atheros Communication yang diproduksi oleh Asustek Komputer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp dan Vodafone Group Plc.

Hingga saat ini terdapat beberapa versi dari sistem operasi *Android*, antara lain:

a. *Android* versi 1.1

Dirilis pada 9 Maret 2009. *Android* versi ini dilengkapi dengan adanya jam, alarm, *voice search*, pengiriman pesan dengan Gmail dan pemberitahuan *email*.

b. *Android* versi 1.5 (Cupcake)

Dirilis pada Mei 2009. Terdapat pembaruan dari versi 1.1 diantaranya adalah fitur *upload* video ke Youtube dan gambar ke Picasa langsung dari telepon, dukungan *bluetooth* A2DP, kemampuan terhubung secara otomatis ke headset *bluetooth*, animasi layar, dan *keyboard* pada layar yang dapat disesuaikan dengan sistem.

c. Android versi 1.6 (Donut)

Dirilis pada September 2009. Pembaruan yang terdapat pada versi ini diantaranya adalah proses pencarian yang lebih baik, penggunaan baterai indikator dan kontrol applet VPN. Fitur lainnya adalah memungkinkan pengguna untuk memilih foto yang akan dihapus, kamera, camcorder dan galeri yang diintegrasikan, CDMA/EVDO, 802.1x, VPN, Gestures, *Text-to-speech engine*.

d. Android versi 2.1 (Eclair)

Dirilis pada 3 Desember 2009. Perubahan yang dilakukan adalah pengoptimalan *hardware*, peningkatan Google Maps 3.1.2, perubahan UI dengan *browser* baru dan dukungan HTML5, daftar kontak yang baru, dukungan *flash* untuk kamera 3.2 MP, digital zoom dan *bluetooth* 2.1.

e. Android versi 2.2 (Froyo)

Dirilis pada 20 Mei 2010. Versi *Android* inilah yang sekarang banyak digunakan sebagai standar sistem operasi mereka. Terdapat perubahan yang cukup signifikan dari versi sebelumnya diantaranya adalah kerangka aplikasi memungkinkan penggunaan dan penghapusan komponen yang tersedia, Dalvik Virtual Machine (DVM) yang dioptimalkan untuk perangkat *mobile*, grafik di 2D dan 3D berdasarkan *libraries* OpenGL, SQLite, mendukung berbagai format audio dan video, GSM, *bluetooth*, EDGE, 3G, Wifi, kamera, *Global Positioning System* (GPS), kompas dan accelerometer.

f. Android versi 2.3 (GingerBread)

Dirilis pada 6 Desember 2010. Beberapa perbaikan fitur dari versi sebelumnya adalah SIP-based VoIP, *Near Field Communications* (NFC), *gyroscope* dan sensor, *multiple cameras support*, *mixable audio effect* dan *download manager*.

g. *Android* versi 3.0 (Honeycomb)

Dirilis tahun 2011. *Android* versi ini dirancang khusus untuk tablet, sehingga terdapat perbedaan dari fitur UI (*User Interface*). Honeycomb sengaja dibuat untuk layar yang lebih besar dan juga dapat mendukung *multiprocessor*.

h. *Android* versi 4.0 (Ice Cream)

Versi ini masih dalam pengembangan. Dari berbagai informasi menyebutkan bahwa versi Ice Cream merupakan gabungan antara versi Gingerbread dengan Honeycomb. Sehingga bisa digunakan untuk ponsel maupun tablet. Dan kemungkinan dirilis pada quarter ke 4 tahun 2011.

3. *Arsitektur Android*

Secara garis besar arsitektur *Android* dapat dijelaskan dan digambarkan sebagai berikut:

a. *Application dan Widget*

Application dan Widget ini adalah *layer* dimana kita berhubungan dengan aplikasi saja. Di *layer* terdapat aplikasi inti termasuk klien *email*, program SMS, kalender, peta, *browser*, kontak, dan lain-lain. Semua aplikasi ditulis dengan menggunakan bahasa pemrograman JAVA.

b. *Application Framework*

Application Framework adalah *layer* untuk melakukan pengembangan / pembuatan aplikasi yang akan dijalankan di sistem operasi *Android*, karena pada *layer* inilah aplikasi dapat dirancang dan dibuat, seperti *content provider* yang berupa SMS dan panggilan telepon.

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

- a) Views
- b) Content Provider
- c) Resource Manager

- d) Notification Manager
- e) Activity Manager
- f) Libraries

Libraries adalah *layer* tempat fitur-fitur *Android* berada, biasanya para pengembang aplikasi mengakses *libraries* untuk menjalankan aplikasinya

a. *Android Runtime*

Layer yang membuat aplikasi *Android* dapat dijalankan di mana dalam prosesnya menggunakan implementasi Linux. Dalvik *Virtual Machine* merupakan mesin yang membentuk dasar kerangka aplikasi *Android*.

Di dalam *Android Runtime* dibagi menjadi dua bagian yaitu:

1) *Core Libraries*

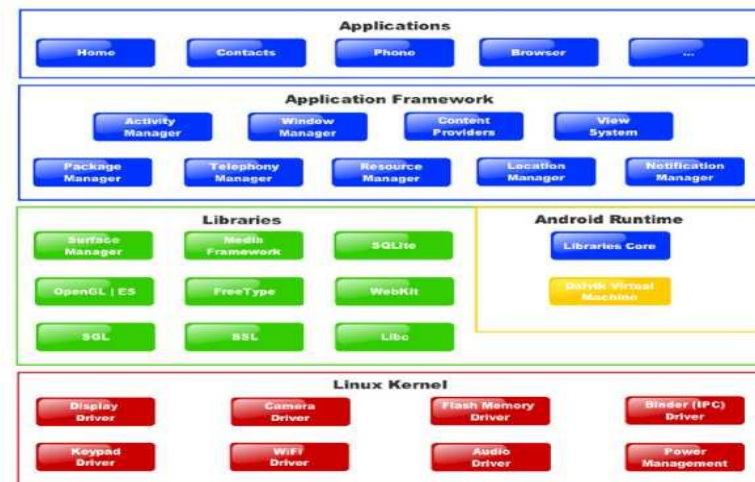
Aplikasi *Android* dibangun dalam bahasa Java, sementara DVM bukan merupakan *virtual machine* untuk Java. Sehingga diperlukan *libraries* yang berfungsi untuk menterjemahkan bahasa Java/C yang ditangani oleh *Core Libraries*.

2) *Dalvik Virtual Machine*

Virtual Machine berbasis *register* yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien, dimana merupakan pengembangan yang mampu membuat Linux kernel untuk melakukan *threading* dan manajemen tingkat rendah.

b. Linux Kernel

Linux Kernel adalah *layer* dimana inti sistem operasi dari *Android* itu berada. Berisi file system yang mengatur system processing memory, resource, drivers, dan sistem-sistem operasi *Android* lainnya. Linux Kernel yang digunakan



Android adalah Linux Kernel release.

Gambar 2.1 Arsitektur *Android*

(Sumber : NazruddinSafaat H.2012. *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC berbasis Android*)

Smartphone adalah sebuah telepon yang kegunaan dasarnya sama dengan telepon biasa yang dapat dibawa kemana-mana dan tidak perlu disambungkan dengan kabel, namun memiliki kemampuan tingkat tinggi dengan fungsi yang menyerupai komputer.

F. Perangkat Lunak Pembangun Sistem

1. *Java Software Development Kit (SDK Java)*

SDK Java merupakan kebutuhan utama bagi programmer untuk membuat dan menjalankan java. Komponen JDK antara lain compiler(javac), interpreter(java) disebut juga java virtual machine atau java runtime environment, applet viewer(appletviewer), debugger(jdb), java class library(jcl), header dan stub generator(javah), dan yang paling penting yaitu java documentation(javadoc).

2. *Eclipse Software Development Kit (SDK Eclipse)*

SDK *Eclipse* adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (*platform-independent*).

3. *Android Software Development Kit (SDK Android)*

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

3. *Android Developer Tools (ADT)*

Definisi *Android Developer Tools* adalah suatu perangkat tambahan koneksi di *Eclipse* dengan *Android SDK*, sehingga bisa lebih memudahkan membuat program untuk *Android* di *Eclipse*.

4. *Android Package (APK)*

APK adalah paket aplikasi *Android* (***Android*** PacKage). APK umumnya digunakan untuk menyimpan sebuah aplikasi atau program yang akan dijalankan pada perangkat *Android*. APK pada dasarnya seperti zip file, karena berisi dari kumpulan file, dapat diperoleh melalui berbagai metode, seperti menginstal sebuah aplikasi melalui Market, download dari sebuah situs web, atau membuat sendiri dengan bahasa Java.

5. *Android Virtual Devices (AVD)*

AVD adalah konfigurasi dari emulator sehingga kita dapat menjalankan perangkat *Android* sesuai model yang dipilih, misalkan *Android* 1.5 atau 2.2. Untuk dapat menjalankan emulator.

6. Emulator

Emulator adalah Aplikasi yang memungkinkan sebuah software dari sebuah platform berjalan di atas platform lain. Emulator menjalankan kode-kode software pada virtual machine. Sehingga software tersebut mengira bahwa ia sedang berjalan di atas platform aslinya. Dalam hal ini emulator *Android* dapat membuat *virtual phone* di komputer seperti telepon genggam berbasis sistem operasi *Android* yang sebenarnya.

7. Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah bahasa pemodelan secara grafis untuk menspesifikasikan, memvisualisasikan, membangun, dan mendokumentasikan seluruh rancangan sistem perangkat lunak. Penggunaan model ini bertujuan untuk mengidentifikasi bagian-bagian yang termasuk dalam lingkup sistem yang dibahas dan bagaimana hubungan antara sistem dengan subsistem maupun sistem lain di luarnya.

Selain itu UML adalah bahasa pemodelan yang menggunakan konsep orientasi object. UML dibuat oleh Grady Booch, James Rumbaugh, dan Ivar Jacobson di bawah bendera Rational Software Corp [HAN98]. UML menyediakan notasi-notasi yang membantu memodelkan sistem dari berbagai perspektif. UML tidak hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan.

Berikut ini definisi *Unified Modeling Language* (UML) menurut para ahli:

Menurut (Hend, 2006) “Unified Modeling Language (UML) adalah bahasa yang telah menjadi standard untuk visualisasi, menetapkan, membangun dan mendokumentasikan artifak suatu sistem perangkat lunak”.

Menurut (Adi Nugroho : 2005). “Unified Modeling Language (UML) adalah alat bantu analisis serta perancangan perangkat lunak berbasis objek”.

Menurut (Joomla dari <http://soetrasoft.com> : 2007). “Unified Modeling Language (UML) merupakan standard modeling language yang terdiri dari kumpulan-kumpulan diagram, dikembangkan untuk membantu para pengembang sistem dan software agar bisa menyelesaikan tugas-tugas seperti: Spesifikasi, Visualisasi, Desain Arsitektur, Konstruksi, Simulasi dan testing serta Dokumentasi”.

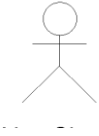


Berdasarkan beberapa pendapat yang dikemukakan diatas dapat ditarik kesimpulan bahwa “ *Unified Modeling Language (UML)* adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasikan, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis OO (*Object Oriented*)”. *Unified Modelling Language (UML)* adalah sebuah bahasa untuk menentukan, visualisasi, konstruksi, dan mendokumentasikan *artifacts* dari sistem *software*, untuk memodelkan bisnis, dan sistem *nonsoftware* lainnya. *Artifacts* adalah sepotong informasi yang digunakan atau dihasilkan dalam suatu proses rekayasa *software*. *Artifacts* dapat berupa model, deskripsi, atau software. Untuk membuat suatu model, UML memiliki diagram grafis yang diberi nama berdasarkan sudut pandang yang berbeda-beda terhadap sistem dalam proses analisa atau rekayasa

UML menyediakan 10 macam diagram yang merupakan salah satu alat bantu yang sangat handal dalam mengembangkan system berorientasi objek. Ada 6 jenis diagram yang ditangani oleh UML, yakni:

1. Use Case Diagram

Use Case Diagram adalah suatu kumpulan urutan interaksi di antara user dengan system untuk mencapai suatu tujuan di mana use case ini menggambarkan kebutuhan fungsional suatu system tanpa menampilkan struktur internal system. Simbol Use Case Diagram dapat dilihat pada tabel 2.2.

Tabel 2.2 Simbol Use Case Diagram*Sumber (Nugroho,2005)*

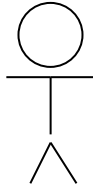
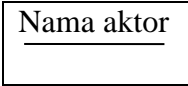

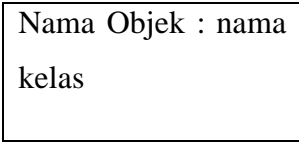

No	Simbol	Nama	Keterangan
1		Actor	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan use case.
2		Use Case	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor.
3		System	Menspesifikasikan paket yang menampilkan sistem secara terbatas.

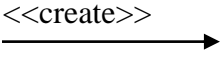
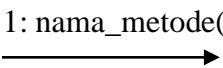
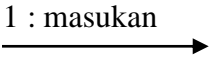
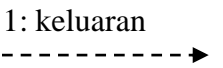
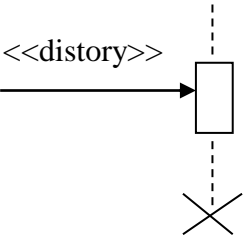
2. Sequence Diagram

Sequence diagram adalah suatu diagram yang menggambarkan interaksi antar obyek dan mengindikasikan komunikasi diantara obyek-obyek tersebut. Diagram ini juga menunjukkan serangkaian pesan yang dipertukarkan oleh obyek-obyek yang melakukan suatu tugas atau aksi tertentu. Obyek-obyek tersebut kemudian diurutkan dari kiri ke kanan, aktor yang menginisiasi interaksi biasanya ditaruh di paling kiri dari diagram. Daftar simbol Sequence Diagram dapat dilihat pada tabel 2.3.

Tabel 2.3 Simbol-simbol yang ada pada Sequence Diagram

Sumber : (Rekayasa Perangkat Lunak, Rosa A.S, M. Shalahuddin, 2013 : 165)

NO	SIMBOL	KETERANGAN
1	<p>Aktor</p>  <p>Atau</p>  <p>Tanpa waktu aktif</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat, walaupun simbol aktor itu orang belum tentu merupakan orang biasanya dinyatakan dengan menggunakan kata kerja frase nama use case.</p>
2	<p>Garis hidup</p> 	<p>Menyatakan kehidupan suatu objek.</p>
3	<p>Objek</p> 	<p>Menyatakan objek yang berinteraksi pesan .</p>
4	<p>Waktu aktif</p> 	<p>Menyatakan Objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif.</p>

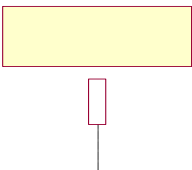
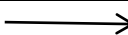
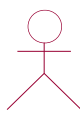
5	Pesan tipe create 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
6	Pesan tipe call 1: nama_metode() 	Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau diri sendiri.
7	Pesan tipe send 1 : masukan 	Menyatakan bahwa suatu objek mengirimkan data / masukkan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8	Pesan tipe return 1: keluaran 	Menyatakan bahwa suatu objek telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
9	Pesan tipe destroy 	Menyatakan bahwa suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jik ada create maka ada destroy.

3. Collaboration Diagram

Collaboration diagram adalah perluasan dari objek diagram. Objek diagram menunjukkan objek-objek dan hubungannya dengan yang lain. Collaboration diagram menunjukkan pesan-pesan objek yang dikirim satu sama lain. Collaboration diagram dapat dilihat pada tabel 2.4.

Tabel 2.4 Simbol-simbol yang ada pada Collaboration Diagram



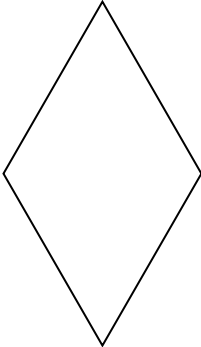

Sumber : (Rosa A.S, M. Shalahuddin, 2013)


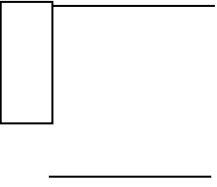
No	Simbol	Nama	Keterangan
1		Object dan lifeline	Orang tempat, benda, kejadian atau konsep yang ada dalam dunia nyata yang penting bagi suatu aplikasi yang saling berinteraksi.
2		Message	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktivitas yang terjadi.
3		Actor	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan use case.

4. Activity Diagram

Activity diagram adalah representasi secara grafis dari proses dari proses dan control flow dan berfungsi untuk memperlihatkan alur dari satu aktivitas ke aktivitas yang lain serta menggambarkan perilaku yang kompleks. Simbol Activity Diagram dapat dilihat pada tabel 2.5.

Tabel 2.4 Simbol-simbol yang ada pada Activity Diagram*Sumber : (Rosa A.S, M. Shalahuddin, 2013 : 162)*

N O	SIMBOL	KETERANGAN
1	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3	Percabangan / decision 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4	Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
5	Pengabungan / join	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.

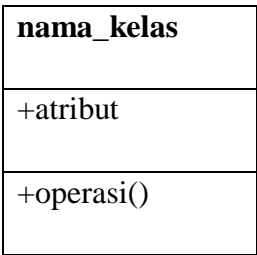

		
6	Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.



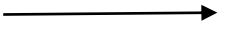

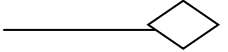
5. Class Diagram

Kelas (*class*) adalah sebuah spesifikasi yang jika diinstansi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan perancangan berorientasi objek. Kelas menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/fungsi) (Tohari, 2014). Class diagram dapat dilihat pada tabel 2.6.

Tabel 2.6 Simbol-simbol yang ada pada Class Diagram

Sumber : (Rekayasa Perangkat Lunak, Rosa A.S, M. Shalahuddin, 2013 : 146)

NO	GAMBAR	KETERANGAN
1	Kelas 	Kelas pada struktur sistem.
2	antarmuka/interface  nama_interface	Sama dengan konsep interface dalam pemrograman berorientasi objek.

3	Asosiasi association 	/	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity.
4	Asosiasi berarah 		Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity.
5	Generalisasi 		Relasi antar kelas dengan makna generalisasi spesialisasi (umum khusus).
6	Kebergantungan 		Kebergantungan antar kelas.
7	Agregasi aggregation 	/	Relasi antar kelas dengan makna semua bagian (whole part).

6. Component Diagram

Component diagram adalah diagram yang menunjukkan organisasi dan kebergantungan di antara sekumpulan komponen. Diagram ini memodelkan pandangan implementasi fisik dari sistem.

Langkah-langkah Penggunaan Unified Modeling Language (UML)

Menurut (Afif Amrullah:2002). “Langkah-langkah penggunaan Unified Modeling Language (UML) sebagai berikut:

- 1) Buatlah daftar business process dari level tertinggi untuk mendefinisikan aktivitas dan proses business process yang mungkin muncul.
- 2) Petakan use case untuk setiap business process untuk mendefinisikan dengan tepat fungsional yang harus disediakan oleh sistem, kemudian perhalus use case diagram dan lengkapi dengan requirement, constraints dan catatan-catatan lain.
- 3) Buatlah deployment diagram secara kasar untuk mendefinisikan arsitektur fisik sistem.
- 4) Definisikan requirement lain non fungsional, security dan sebagainya yang juga harus disediakan oleh sistem.
- 5) Berdasarkan use case diagram, mulailah membuat activity diagram.
- 6) Definisikan obyek-obyek level atas package atau domain dan buatlah sequence dan/atau collaboration untuk tiap alir pekerjaan, jika sebuah use case memiliki kemungkinan alir normal dan error, buat lagi satu diagram untuk masing-masing alir.
- 7) Buatlah rancangan user interface model yang menyediakan antamuka bagi pengguna untuk menjalankan skenario use case.

Berdasarkan model-model yang sudah ada, buatlah class diagram. Setiap package atau domain dipecah menjadi hirarki class lengkap dengan atribut dan metodenya. Akan lebih baik jika untuk setiap class dibuat unit test untuk menguji fungsionalitas class dan interaksi dengan class lain.

Setelah class diagram dibuat, kita dapat melihat kemungkinan pengelompokkan class menjadi komponen-komponen karena itu buatlah component diagram pada tahap ini. Juga, definisikan test integrasi untuk setiap komponen meyakinkan ia bereaksi dengan baik.

Perhalus deployment diagram yang sudah dibuat. Detilkan kemampuan dan *requirement* piranti lunak, sistem operasi, jaringan dan sebagainya. Petakan komponen ke dalam node.

Mulailah membangun sistem. Ada dua pendekatan yang tepat digunakan:

- a. Pendekatan *use case* dengan mengassign setiap use case kepada tim pengembang tertentu untuk mengembangkan unit kode yang lengkap dengan test.
- b. Pendekatan komponen yaitu mengassign setiap komponen kepada tim pengembang tertentu.

Lakukan uji modul dan uji integrasi serta perbaiki model beserta codenya. Model harus selalu sesuai dengan code yang aktual. Perangkat lunak siap dirilis”.

8. Object Oriented Programming (OOP)

Object Oriented Programming (OOP) atau Pemrograman BerorientasiObjek adalah konsep pemrograman yang difokuskan pada penciptaan kelas yang merupakan abstraksi / blueprint / prototype dari suatu objek. Kelas ini harus mengandung sifat (data) dan tingkah laku (method) umum yang dimiliki oleh objek-objek yang kelak akan dibuat (diinstansiasi). Data dan method merupakan anggota (member) dari suatu kelas.

Pemrograman prosedural murni yang tidak menerapkan konsep object oriented (karena ada bahasa pemrograman prosedural yang juga sudah berorientasi objek, meskipun belum sepenuhnya) banyak menitikberatkan ke arah pembentukan fungsi-fungsi, sehingga di dalam program akan terdapat banyak sekali fungsi dan variabel yang menyulitkan pemrogram untuk mengelola dan mengembangkannya. Oleh karena itu, dengan memperhatikan kekurangan-kekurangan tersebut, maka dibentuklah bahasa yang menerapkan pendekatan object oriented untuk menyederhanakan fungsi-fungsi dan variabel-variabel ke dalam bentuk objek.

Dalam OOP dibutuhkan memory lebih besar dibandingkan dengan program prosedural (tradisional). Dua objek yang identik akan memerlukan dua area memory berbeda walaupun dari sisi data dan proses keduanya memiliki jumlah dan jenis yang sama. Hal ini disebabkan karena data dan proses pada kedua objek tersebut dipisahkan oleh komputer.

Secara garis besar yang menjadi ciri dari OOP adalah adanya proses abstraksi (abstraction), pengkapsulan (encapsulation), penurunan sifat (inheritance), dan polimorfisme (polymorphism) pada objek-objek yang dibentuk.

9. Java

Java adalah sebuah teknologi yang diperkenalkan oleh Sun Microsystems pada pertengahan tahun 1990. Menurut definisi dari Sun, Java adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer standalone ataupun pada lingkungan jaringan.

Java2 adalah generasi kedua dari Java platform (generasi awalnya adalah Java Development Kit). Java berdiri di atas sebuah mesin interpreter yang diberi nama JVM (Java Virtual Machine). JVM inilah yang akan membaca bytecode dalam file .class dari suatu program sebagai representasi langsung program yang berisi bahasa mesin. Oleh Karena itu bahasa Java disebut sebagai bahasa pemrograman yang portable karena dapat dijalankan berbagai system operasi, asalkan pada system operasi tersebut terdapat JVM.

Platform Java terdiri dari kumpulan library, JVM, kelas-kelas yang dipaket dalam sebuah lingkungan rutin Java, dan sebuah compiler, debugger, dan perangkat lain yang dipaket dalam Java Development Kit (JDK). Java2 adalah generasi yang sekarang sedang berkembang dari platform Java. Agar sebuah program Java dapat dijalankan, maka file dengan ekstensi “.java” harus dikompilasi menjadi file bytecode. Untuk menjalankan bytecode tersebut dibutuhkan JRE (Java Runtime Environment) yang memungkinkan pemakai untuk menjalankan program Java, hanya menjalankan, tidak untuk membuat kode baru lagi. JRE berisi JVM dan library Java yang digunakan.

10. Eclipse

Eclipse adalah sebuah IDE (Integrated Development Environment) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (platform independent) (Akhmad Dharma Kasman, 2013).

Berikut ini adalah sifat dari eclipse:

- a. Multi platform : Target sistem operasi/Eclipse adalah Microsoft Windows, Linux, Solaris, AIX HP-UX dan Mac OS x.
- b. Multi language : Eclipse dikembangkan dengan bahasa pemrograman java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti CC++, Cobol, Phiton, Perl, PHP dan sebagainya.
- c. Multi role : Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan Web, dan lain sebagainya.

Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan *open source*, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan dengan komponen yang dinamakan *Plugin*. Tampilan awal Eclipse ADT dapat dilihat pada gambar 2.2.



Gambar 2.2 Eclipse ADT

BAB III

ANALISA DAN PERANCANGAN

A. Analisa Actor

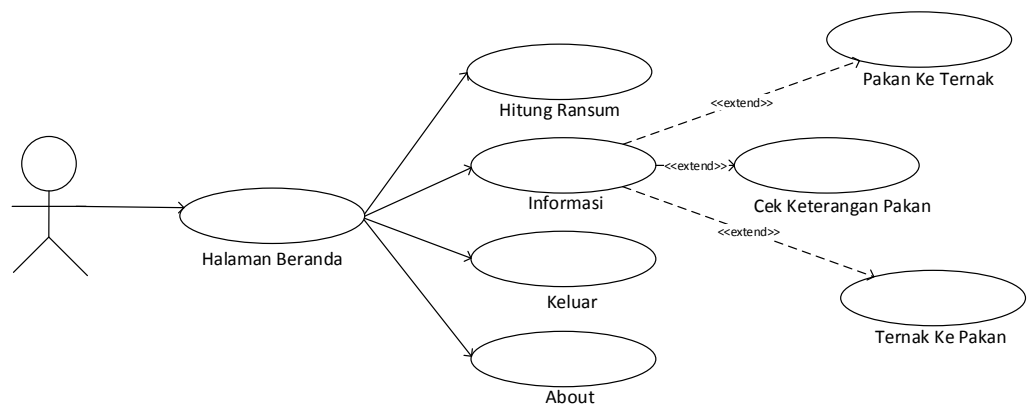
Untuk mengasilkan suatu rancangan yang baik maka diperlukan suatu bentuk aliran pembuatan rancangan sistim informasi formulasi ransum yang selanjutnya akan dijelaskan pada sub bab berikut.

1. *Unified Modeling Language (UML)*

Unified Modelling Language (UML) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.

2. Use case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem, yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use Case Diagram* dari sistem yang dirancang dapat digambarkan seperti pada gambar 3.1.



Gambar 3.1 Use Case Diagram.

Dari *Use Case Diagram* diatas maka urutan kegiatan yang dapat dilakukan oleh user adalah :

- a. Pada saat aplikasi dijalankan user akan dihadapkan dengan halaman beranda.
- b. Pada halaman menu utama user dihadapkan dengan beberapa pilihan halaman yaitu halaman hitung ransum, informasi, about dan keluar.
- c. Pada halaman informasi terdapat tiga buah menu pilihan yaitu halaman cek informasi pakan, pakan ke ternak, ternak ke pakan.
- d. Halaman cek informasi pakan berisikan tentang rincian dari bahan pakan tersebut.
- e. Halaman ternak ke pakan berisikan tentang perhitungan mencari jumlah pakan berdasarkan ternak yang dimiliki dan tambahan hasil berupa jumlah telur yang dihasilkan perbulan.
- f. Halaman pakan ke ternak berisikan tentang perhitungan mencari jumlah ternak berdasarkan pakan yang ada.
- g. Pada halaman about berisi tentang biodata si pembuat program.
- h. Pada button keluar berfungsi untuk menghentikan aplikasi.

3. Sequence Diagram

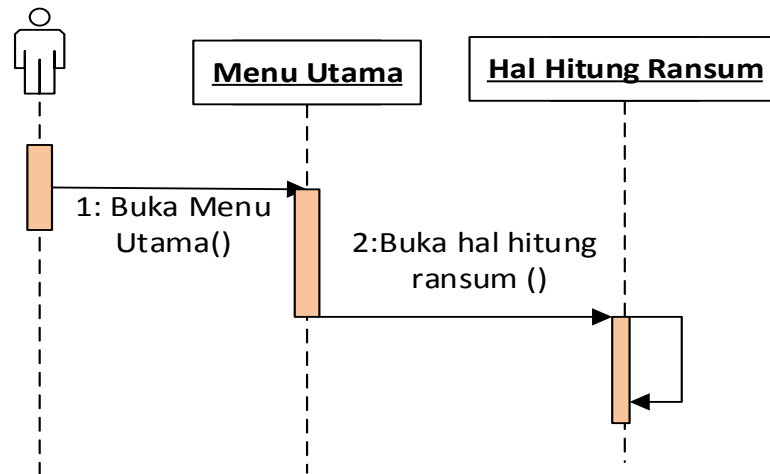
Sequence diagram menjelaskan interaksi antar objek di dalam dan di sekitar sistem berupa pesan (*message*) yang disusun dalam suatu urutan waktu yaitu urutan kejadian yang dilakukan oleh seorang aktor dalam menjalankan sistem.

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai response dari sebuah kegiatan untuk menghasilkan *output* tertentu. Adapun *sequence diagram* dapat dilihat dari gambar-gambar berikut :

a. Sequence Diagram Komposisi Ransum

Sequence diagram Komposisi Ransum pada halaman user dengan urutan event dan waktu user saat menghitung komposisi ransum.

Sequence diagram ini dapat dilihat pada Gambar 3.2.



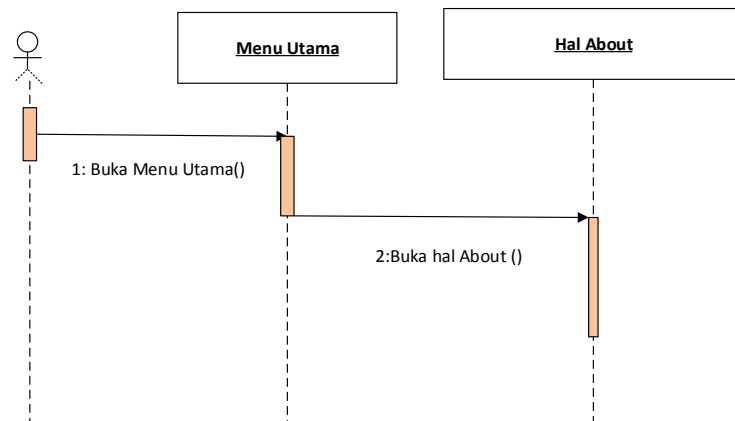
Gambar 3.2 Sequence Diagram Hitung Ransum

Keterangan Sequence Diagram Komposisi Ransum :

User membuka menu utama, pilih halaman Hitung ransum, sistem memproses ransum, jika data yang di isi tidak lengkap maka kembali inputkan komposisi, jika isi lengkap maka data diproses.

b. Sequence Diagram About

Sequence diagram About dapat dilihat pada Gambar 3.3.



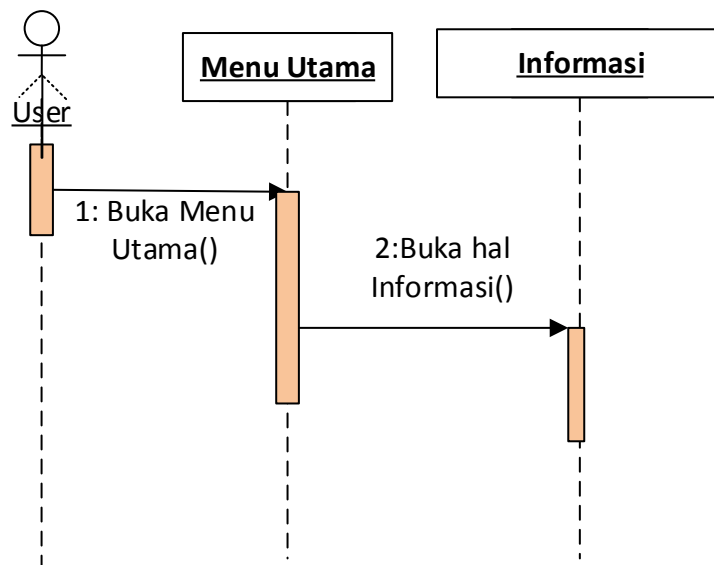
Gambar 3.3 Sequence Diagram About

Keterangan Sequence Diagram About :

User dapat melihat hal about dengan cara membuka menu utama terlebih dahulu.

c. Sequence Diagram Informasi

Sequence diagram Informasi dapat dilihat pada Gambar 3.4.



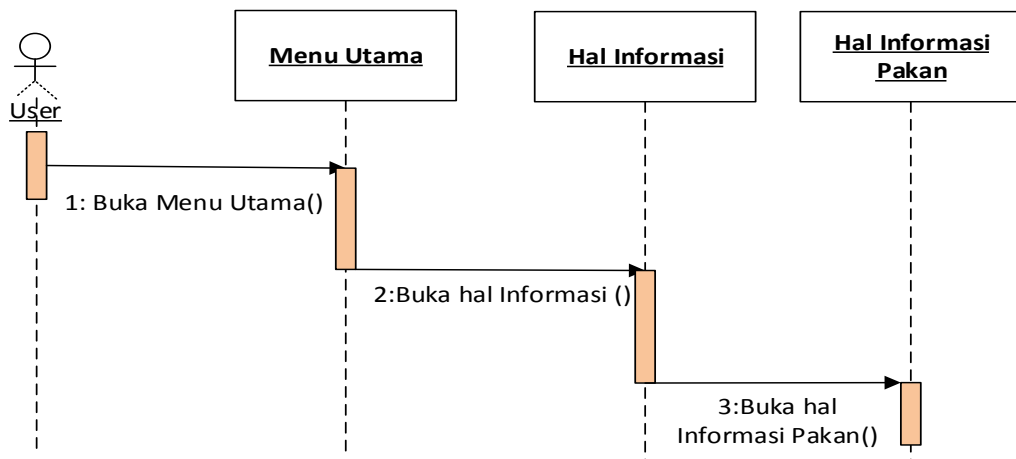
Gambar 3.4 Sequence Diagram Informasi

Keterangan Sequence Diagram Informasi :

User dapat melihat hal Informasi dengan cara membuka menu utama terlebih dahulu.

d. Sequence Diagram Informasi pakan

Sequence diagram Informasi Pakan dapat dilihat pada gambar 3.5



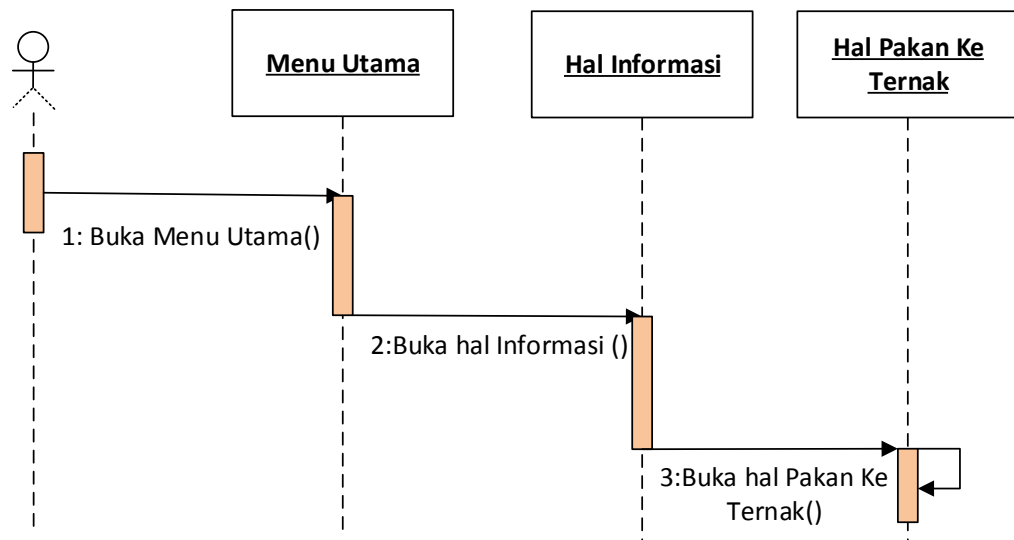
Gambar 3.5 Sequence Diagram Informasi Pakan

Keterangan Sequence Diagram Informasi Pakan :

User dapat melihat hal Informasi Pakan dengan cara membuka menu utama terlebih dahulu, kemudian dilanjutkan dengan membuka hal informasi.

e. Sequence Diagram Pakan Ke Ternak

Sequence diagram Pakan Ke Ternak dapat dilihat pada gambar 3.6



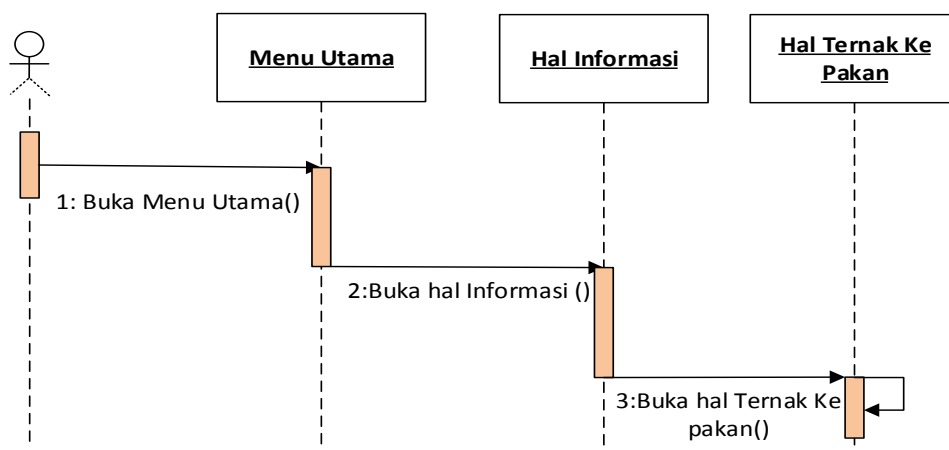
Gambar 3.6 Sequence Diagram Pakan Ke Ternak

Keterangan Sequence Diagram Pakan Ke Ternak :

User dapat melihat hal Pakan Ke Ternak dengan cara membuka menu utama terlebih dahulu, kemudian dilanjutkan dengan membuka hal informasi

f. Sequence diagram Ternak Ke Pakan

Sequence diagram Ternak Ke Pakan dapat dilihat pada gambar 3.7.



Gambar 3.7 Sequence Diagram Ternak Ke Pakan

Keterangan Sequence Diagram Ternak Ke Pakan :

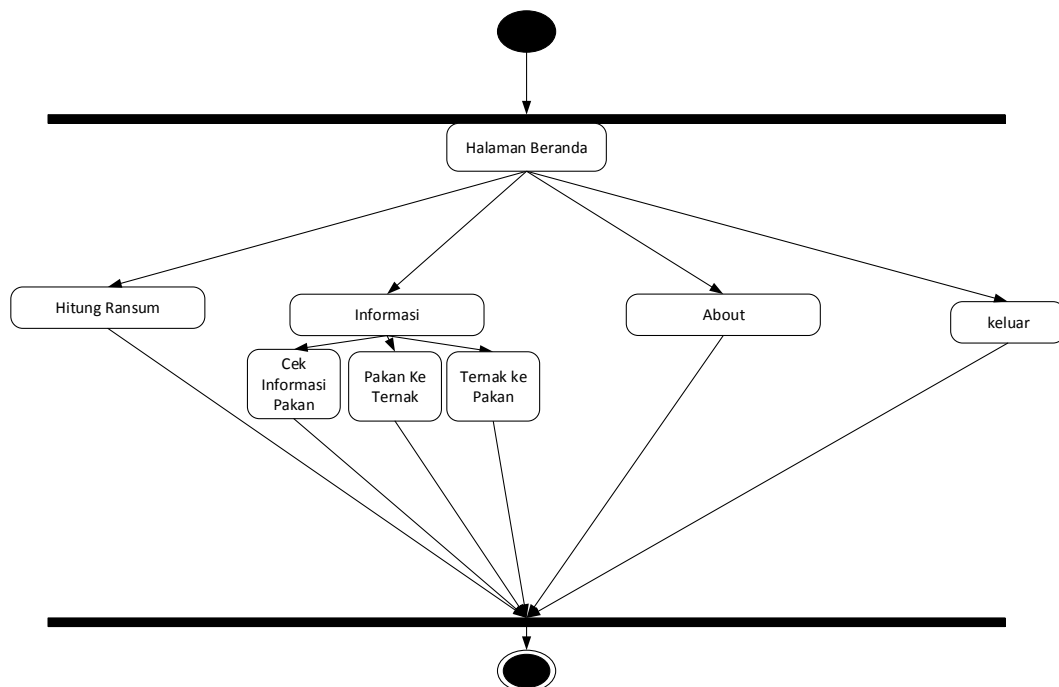
User dapat melihat hal Informasi dengan cara membuka menu utama terlebih dahulu, kemudian dilanjutkan dengan membuka hal Informasi.

4. Activity Diagram

Merupakan diagram yang menggambarkan berbagai alir aktifitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi dan bagaimana mereka berakhir.

a. Activity Diagram User

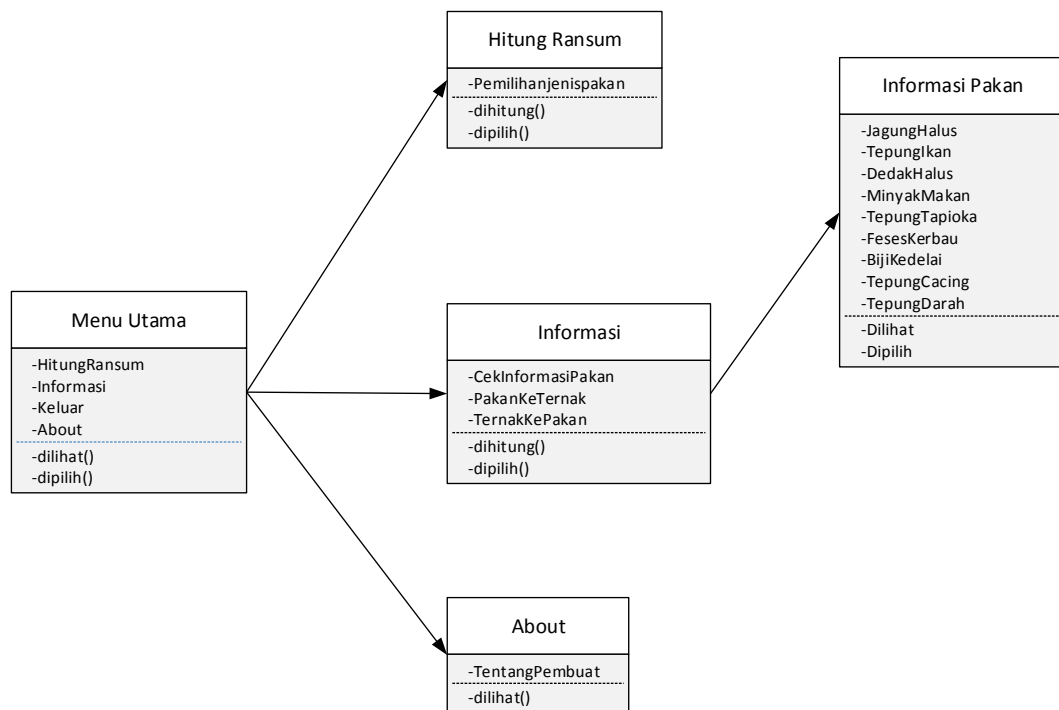
Activity diagram User menggambarkan aktivitas-aktivitas yang dapat dilakukan oleh User di dalam sistem. *Activity diagram* User dapat dilihat pada gambar 3.8.



Gambar 3.8 Activity Diagram User

5. Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class diagram* dari Sistem Informasi Penghitungan dapat digambarkan seperti pada gambar 3.9.

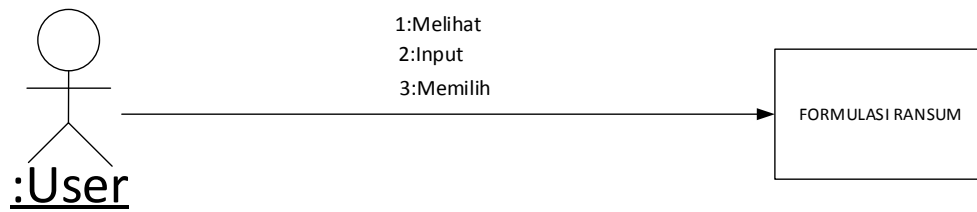


Gambar 3.9 Class Diagram

6. Collaboration Diagram

Collaboration diagram juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*, setiap *message* memiliki *sequence number*, dimana *message* dari level tertinggi memiliki nomor 1.

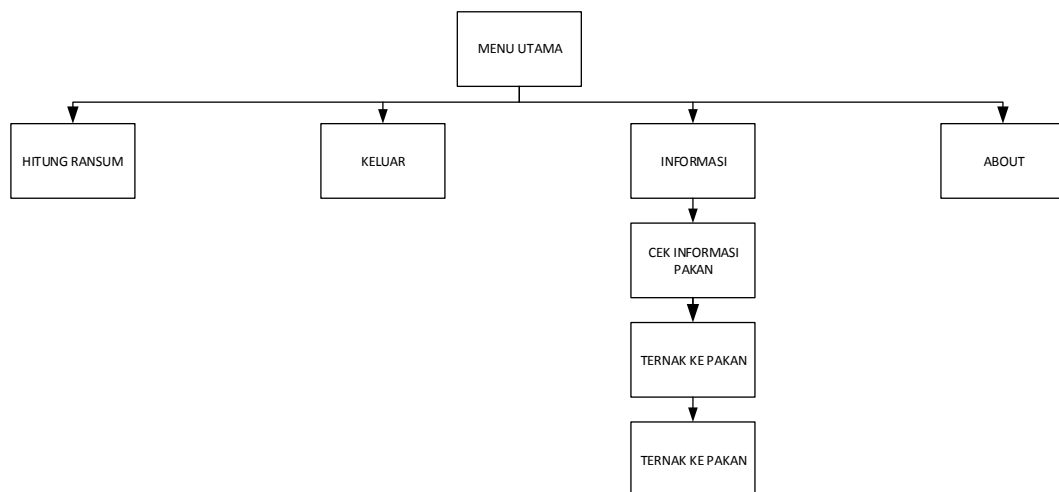
Message dari level yang sama memiliki 39 refix yang sama. Adapun *collaboration diagram* dari analisa dan penerapan aplikasi ini dapat digambarkan sebagai berikut :



Gambar 3.10 Collaboration Diagram

7. Struktur Program

Pada struktur program menggambarkan susunan modul-modul program yang dikerjakan pada sebuah sistem. Adapun bentuk dari struktur program dapat digambarkan seperti pada gambar 3.11.

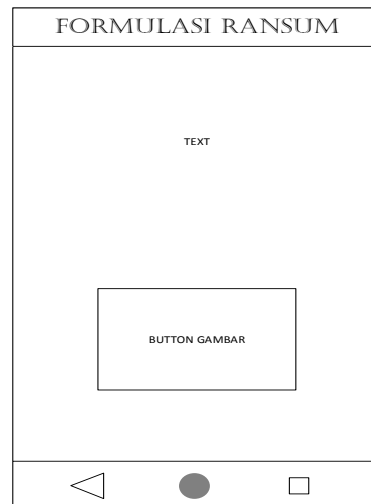


Gambar 3.11 Struktur Program

B. Analisa Rancangan

a. Rancangan Beranda

Beranda ini merupakan halaman pertama dari aplikasi ini. Pada halaman ini terdapat keterangan singkat tentang aplikasi formulasi ransum dan satu buah button untuk menuju ke menu utama. Rancangan beranda dapat dilihat pada gambar 3.12



Gambar 3.12 Halaman Beranda

b. Rancangan halaman Menu Utama

Rancangan Menu Utama memiliki empat buah button yaitu hitung ransum, informasi, keluar dan about. Adapun tampilan *Menu Utama* dapat dilihat dari gambar 3.13.



Gambar 3.13 Halaman Menu Utama

c. Rancangan halaman Hitung ransum

Pada rancangan Hitung Ransum user dapat menghitung ransum mereka sendiri. dapat dilihat pada gambar 3.14.

FORMULASI RANSUM	
Feses Kerbau	<input type="text"/> kg
Jagung Halus	<input type="text"/> kg
Tepung Ikan	<input type="text"/> kg
Dedak Halus	<input type="text"/> kg
Minyak Makan	<input type="text"/> kg
Tepung Tapioka	<input type="text"/> kg
Total Kebutuhan	<input type="text"/> kg
Total	<input type="text"/> kg
ME	<input type="text"/>
<input type="button" value="Hapus"/> <input type="button" value="Hitung"/>	

◀ ● ▶

Gambar 3.14 *Halaman Hitung Ransum*

d. Rancangan output Halaman Hitung Ransum

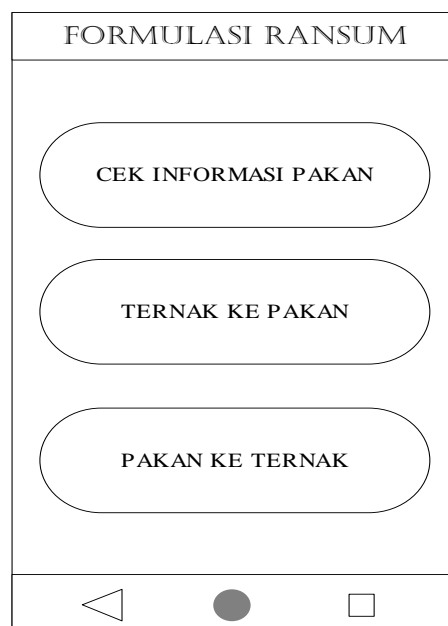
Rancangan output halaman hitung ransum dapat dilihat pada gambar 3.15



Gambar 3.15 *Output Hitung Ransum*

e. Rancangan Halaman Informasi

Halaman Informasi memiliki tiga buah menu yang dapat dipilih oleh user yaitu cek informasi pakan, ternak ke pakan, pakan ke ternak. Dapat dilihat pada gambar 3.16.



Gambar 3.16 *Halaman Informasi*

f. Rancangan halaman cek informasi pakan

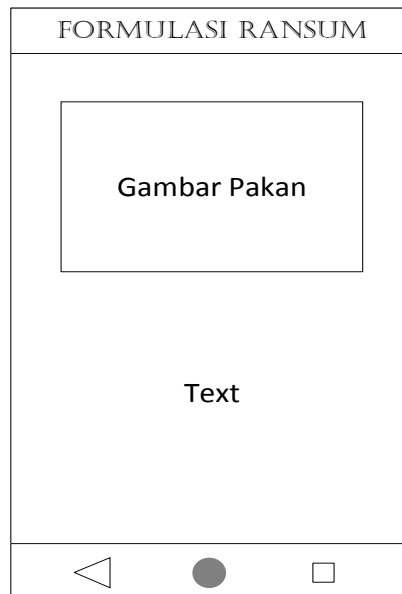
Pada halaman cek informasi pakan, user dapat memilih menu daftar informasi pakan. Halaman cek informasi pakan dapat dilihat pada gambar 3.17

FORMULASI RANSUM
Sub Item1
Sub Item2
Sub Item3
Sub Item4
Sub Item5
Sub Item6


Gambar 3.17 *Halaman Cek Informasi Pakan*

g. Rancangan Sub Item Cek Informasi Pakan

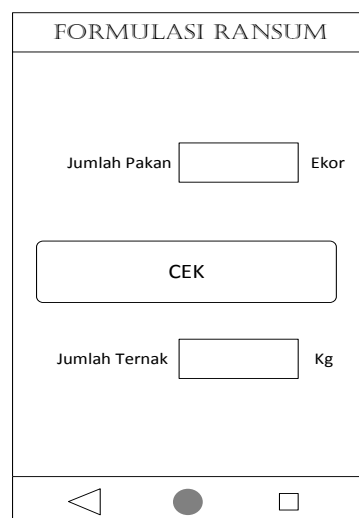
Pada halaman sub item cek informasi pakan user dapat melihat detail dari bahan pakan, rancangan sub item informasi pakan dapat dilihat pada gambar 3.18



Gambar 3.18 Sub Item Cek Informasi Pakan

h. Rancangan halaman Pakan ke Ternak

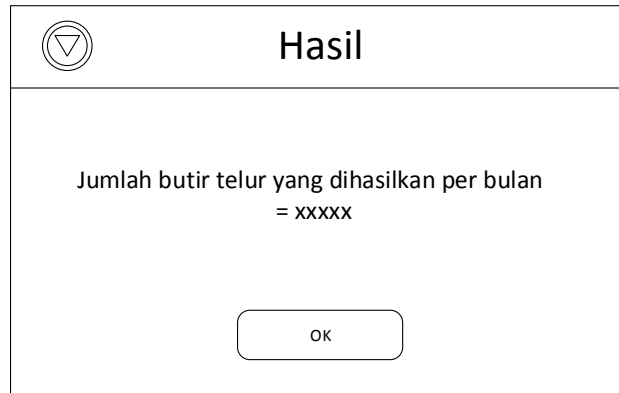
Pada halaman pakan ke ternak user dapat menghitung hasil ternaknya berdasarkan pakan yang ada. Halaman pakan ke ternak dapat dilihat pada gambar 3.19



Gambar 3.19 Halaman Pakan Ke Ternak

i. Rancangan output halaman Pakan Ke Ternak

Rancangan output halaman pakan ke ternak dapat dilihat pada gambar 3.20



Hasil

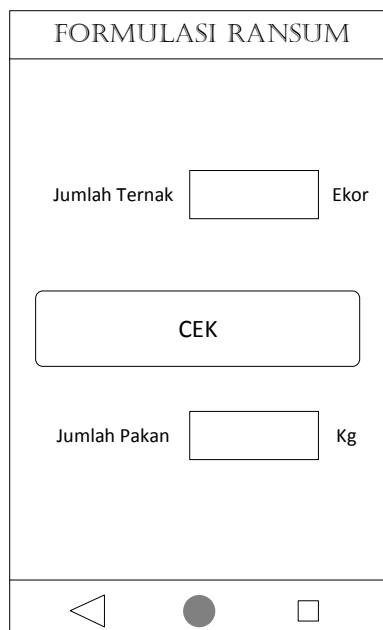
Jumlah butir telur yang dihasilkan per bulan
= xxxxx

OK

Gambar 3.20 *Halaman Pakan Ke Ternak*

j. Rancangan halaman Ternak Ke Pakan

Pada halaman ternak ke pakan user dapat menentukan jumlah kebutuhan pakan berdasarkan banyak ternaknya. Rancangan halaman ternak ke pakan dapat dilihat pada gambar 3.21



FORMULASI RANSUM

Jumlah Ternak Ekor

CEK

Jumlah Pakan Kg

Gambar 3.21 *Halaman Ternak Ke Pakan*

k. Rancangan output halaman Ternak Ke Pakan

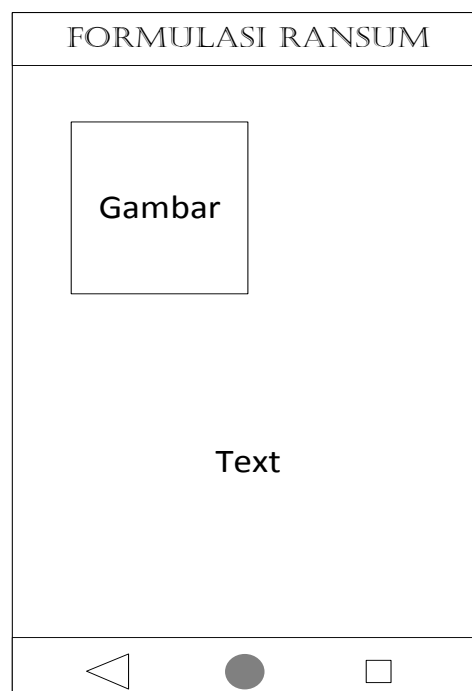
Desain output halaman pakan ke ternak dapat dilihat pada gambar 3.22



Gambar 3.22 Halaman Ternak Ke Pakan

l. Rancangan halaman About

Pada Halaman About dijelaskan tentang data pembuat program, Adapun Halaman About dapat dilihat pada gambar 3.23.



Gambar 3.23. Halaman About

BAB IV

Kesimpulan dan Saran

A. Kesimpulan

Dari uraian penulis pada pembahasan sebelumnya, maka pada bab ini penulis mengambil kesimpulan sebagai berikut :

1. Aplikasi formulasi ransum ini ditujukan untuk menghitung formulasi ransum bagi para peternak ayam petelur, mahasiswa, dan lainnya.
2. Penggunaan aplikasi ini berfungsi bukan hanya untuk menghitung ransum semata, namun juga dapat membandingkan antara jumlah ternak, bahan pakan dengan hasil telur ayam yang akan diperoleh.
3. Penggunaan aplikasi ini juga memiliki daftar harga pakan

B. Saran

Dari beberapa kesimpulan yang telah penulis kemukakan diatas, maka untuk itu penulis mencoba memberikan beberapa saran. Adapun saran-saran dari penulis adalah sebagai berikut :

1. Perlu adanya pengembangan yang lebih lanjut seperti melakukan penambahan jumlah jenis pakan.

DAFTAR PUSTAKA

- Faisal, M. *Sistem Informasi Manajemen: Jaringan*. Yogyakarta : Sukses Offset, 2008
- Grady Booch, J. R. *The Unified Modeling Language User Guide* . Addison Wesley Professional,2005.
- Hermawan, J. *Analisa Desain Dan Pemrograman Berorientasi Objek dengan UML Dan VB.Net*. Yogyakarta: Andi, 2004.
- <http://blog.codingwear.com> diakses pada tanggal 29 Agustus 2016
- Jogiyanto. *Analisis dan Desain Sistem Informasi Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*. Yogyakarta: CV. Andi Offset, 2005.
- Leo Willyanto Santoso “*Pelatihan microsoft visio 2010 profesional*”,pusat komputer,2013
- Zulfikar, 2013. *Manajemen Pemeliharaan Ayam Petelur Ras*.
- Nugroho, A. *Analisis Dan Perancangan Sistem Informasi Dengan Metodologi Berorientasi Objek*. Bandung: Informatika, 2005.
- Wahyono, T. *Sistem Informasi: Konsep Dasar, Analisis dan Desain dan Implementasi*. Yogyakarta: Graha Ilmu, 2004.
- Anggorodi. R. 1979. *Ilmu Makanan Ternak Dasar Umum*. Gramedia. Jakarta.
- <http://anuragaja.staff.ipb.ac.id/files/2012/04/Buku-PBMT.pdf>
- NazruddinSafaat H.2012.*Pemrograman Aplikasi Mobile Smartphone dan Tablet PC berbasis Android*
- <http://freetechebooks.com/ebook-2011/daftar-simbol-uml.html>

LAMPIRAN



KEMENTERIAN AGAMA REPUBLIK INDONESIA
INSTITUT AGAMA ISLAM NEGERI BATANGAS
FAKULTAS EKONOMI DAN BISNIS ISLAM

Jl. Sudirman No. 127 Lima Kaum Batangas Fax: (0752) 71150, 574221, 71890 Fax: (0752) 71879
Website: www.iainbatangas.ac.id e-mail: iaib@iainbatangas.ac.id

SURAT TUGAS

Nomor: B-*gly.e*/In 27/F IV/PP 00 9/11/2017

Dekan Fakultas Ekonomi dan Bisnis Islam dengan ini menugaskan Saudara :

Nama / NIP	Pangkat / Gol	Jabatan	Keterangan
Adriyendi, M.Kom 19770127 200912 1 002	Penata III/c	Lektor	Pembimbing

sebagai Pembimbing Tugas Akhir mahasiswa Jurusan Manajemen Informatika pada semester Ganjil Tahun Akademik 2017/2018, atas nama :

Nama : Brians Pratama
NIM : 14205019
Program Studi : Manajemen Informatika
Judul Proposal : *Perancangan Sistem Informasi Formulasi Ransum Ayam
Pecah menggunakan Metode Trial & Error Berbasis
Android*

Demikian surat tugas ini diberikan kepada yang bersangkutan, untuk dilaksanakan sebagaimana mestinya.

Batangas, 30 November 2017

an Dekan,

Dekan Bidang Akademik

Gandikabagaan



saahat Katuni, M.Ag

30603 200501 2 006



KEMENTERIAN AGAMA REPUBLIK INDONESIA
INSTITUT AGAMA ISLAM NEGERI BATUSANGKAR
LEMBAGA PENELITIAN DAN PENGABDIAN MASYARAKAT

Jl. Sudirman No.137 Kuburajo Lima Kaum Batusangkar 27213, Telp. (0752) 71150, Ext 135, Fax (0752) 71879
Website www.iainbatusangkar.ac.id e-mail: lpdm@iainbatusangkar.ac.id

09 Januari 2018

Nomor : B- 74 /ln.27/L.I/TL.00/ 01 /2018
Sifat : Biasa
Lampiran : 1 Rangkap
Perihal : **Mohon Penerbitan Surat Izin Penelitian**

Yth. Bupati Tanah Datar
Up. Kepala Kantor KESBANGPOL Kabupaten Tanah Datar
Batusangkar

Assalamu'alaikum Wr. Wb.
Dengan hormat,

Bersama ini disampaikan kepada Bapak/Ibu bahwa mahasiswa yang tersebut di bawah ini:

Nama/NIM : BRIANS PRATAMA / 14205019
Tempat/Tanggal Lahir : Batusangkar, 14 Desember 1995
Nomor Induk Keluarga : KTP. 1304071412950002
Fakultas : Ekonomi dan Bisnis Islam
Jurusan : Manajemen Informatika
Alamat : Jorong Balai Bungo Nagari Tanjung Kecamatan Sungayang
Kabupaten Tanah Datar

akan melakukan pengumpulan data untuk proses penulisan laporan hasil penelitiannya sebagai berikut:

Judul Penelitian : **Perancangan Sistem Informasi Formulasi Ransum Ayam Petelur Menggunakan Metode Trial & Error Berbasis Android**
Lokasi : Dinas Peternakan Kab. Tanah Datar
Waktu : 10 Januari 2018 s.d 10 Maret 2018
Dosen Pembimbing 1 : Adriyendi, M.Kom.
Dosen Pembimbing 2 : -

untuk itu, diharapkan kiranya Bapak/Ibu berkenan menerbitkan surat izin penelitian dalam rangka pelaksanaan penelitian mahasiswa yang bersangkutan.

Demikian disampaikan, atas bantuannya diucapkan terimakasih.



Ketua,

(Signature)
Yusrizal Efendi, S.Ag., M.Ag.

Tembusan:

1. Rektor IAIN Batusangkar (Sebagai Laporan).
2. Dekan Fakultas Ekonomi dan Bisnis Islam IAIN Batusangkar (Sebagai Laporan).



KEMENTERIAN AGAMA
INSTITUT AGAMA ISLAM NEGERI
BATUSANGKAR
FAKULTAS EKONOMI DAN BISNIS ISLAM

Jl. Sudirman No. 137 Kubu Raja Lima Kaum Batusangkar 27213 Telp. (0752) 71150, 574221, Fax. (0752) 71879
<http://www.iainbatusangkar.ac.id> e-mail: info@iainbatusangkar.ac.id

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

KARTU BIMBINGAN PENULISAN TUGAS AKHIR
PROGRAM D3 MANAJEMEN INFORMATIKA

Nim / Nama : 14205019/BRIANS PRATAMA
Jurusan : Manajemen Informatika
Dosen Pembimbing : ADRIYENDI, M.Kom
Judul Tugas Akhir : "PERANCANGAN SISTEM INFORMASI FORMULASI RANSUM AYAM PETELUR MENGGUNAKAN METODE TRIAL & ERROR BERBASIS ANDROID"

NO	Tanggal	Materi Bimbingan	Paraf
1	20-12-2017	Konsultasi Bab I	Adri
2	2-1-2018	Revisi Bab I	Adri
3	11-1-2018	Lengkap Bab I	Adri
4	16-1-2018	Revisi Bab II	Adri
5	17-1-2018	Revisi program semi-otomatisasi	Adri
6	22-1-2018	Lengkap Bab II	Adri
7	29-1-2018	Revisi materi output	Adri
8	1-2-2018	Program selesai sesuai design	Adri
9	5-2-2018	Acc vs / minor revisi	Adri
10			
11			
12			

Catatan : Setiap konsultasi dengan pembimbing kartu ini harap dibawa, diisi, dan diparaf oleh dosen pembimbing

Batusangkar, 5 Februari 2018
Tanda Tangan Mahasiswa

BRIANS PRATAMA
NIM. 14205019

Dosen Pembimbing Akademik

Dosen Pembimbing Tugas Akhir

ADRIYENDI, M.Kom

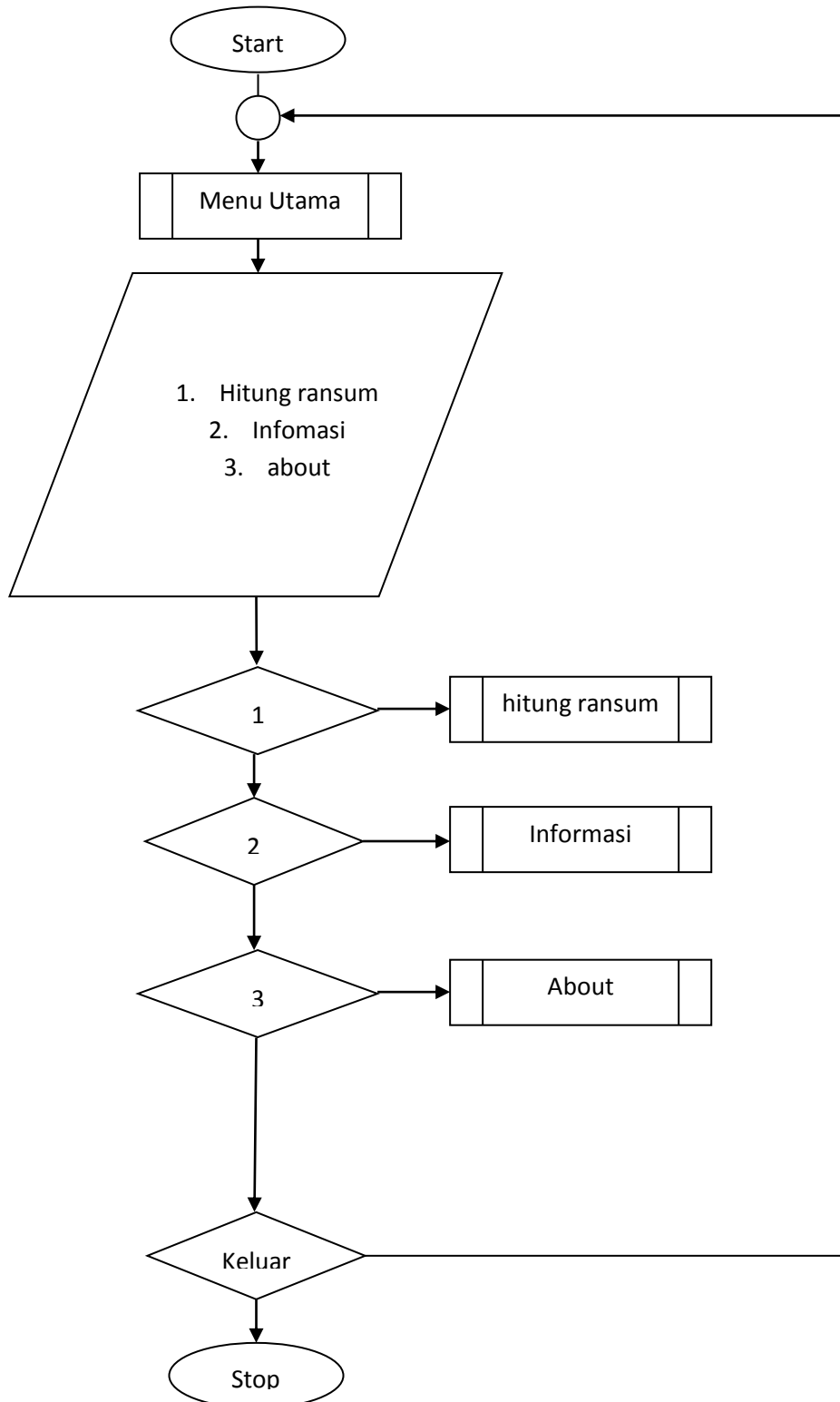
NIP. 1977012 7200912 1 002

ISWANDI, M.Kom

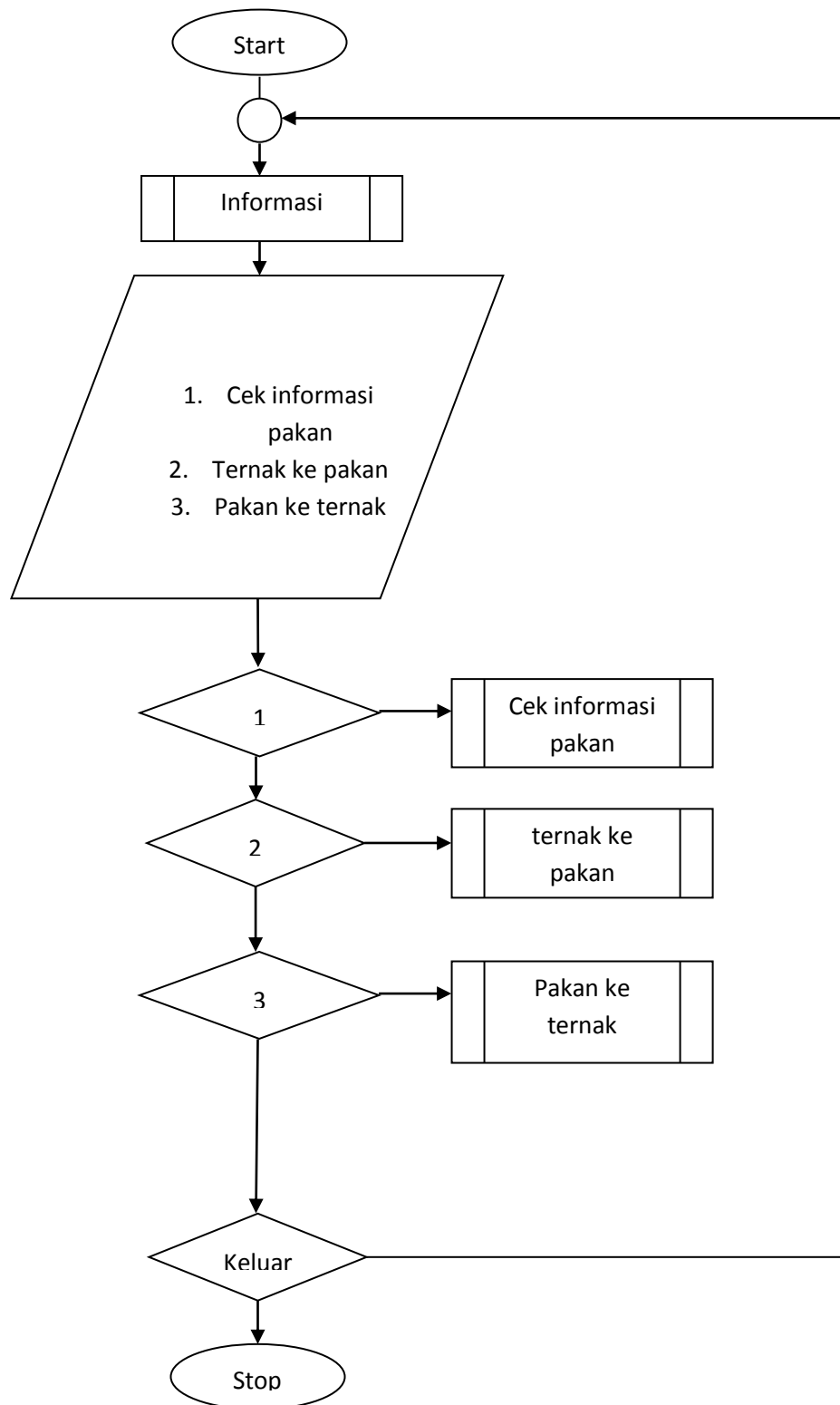
NIP. 19700510 200312 1 004

FLOWCHART

1. Flowchart Menu Utama



2. Flowchart hitung informasi



LISTING PROGRAM

Beranda

```
package
com.example.brianspratama;

import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.*;
import
android.view.View.OnClickListener;
import android.widget.ImageButton;

public class Halamanpertama
extends Activity implements
OnClickListener{

    ImageButton halamanutama;

    @Override

    protected void
onCreate(Bundle savedInstanceState)
{

        super.onCreate(savedInstanceState);

        setContentView(R.layout.acti
vity_halamanpertama);

        halamanutama=(ImageButton
)findViewById(R.id.imageButton1);

        halamanutama.setOnClickLis
tener((OnClickListener)this);
    }

    @Override

    public void onClick(View v){

        if
(v==halamanutama);

        Intent intent=new
Intent(this,MainActivity.class);

        startActivity(intent);

    }

}

}

Halaman utama

package
com.example.brianspratama;

import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.view.View;
```

```

import android.widget.*;
import android.content.Intent;

import
android.view.View.OnClickListener;

import android.app.AlertDialog;

import
android.content.DialogInterface;

import android.view.KeyEvent;

public class MainActivity extends
Activity implements
OnClickListener {

    Button btnhitung;
    Button btninformasi;
    Button btnabout;
    Button btnkeluar;

    @Override
    protected void
onCreate(Bundle savedInstanceState)
{

    super.onCreate(savedInstanceState);

    setContentView(R.layout.acti
vity_main);

    btnhitung = (Button)
findViewById(R.id.btnmulai);

                                btninformasi =
                                (Button)
                                findViewById(R.id.btninformasi);

                                btnabout = (Button)
                                findViewById(R.id.btnabout);

                                btnkeluar = (Button)
                                findViewById(R.id.btnkeluar);

                                btnhitung.setOnClickListener
                                ((OnClickListener) this);

                                btninformasi.setOnClickListe
                                ner((OnClickListener) this);

                                btnabout.setOnClickListener(
                                (OnClickListener) this);

                                btnkeluar.setOnClickListener
                                ((OnClickListener) this);

                                }

    @Override
    public void onClick(View v)
    {

        // TODO Auto-
        generated method stub

        if (v == btnhitung) {

            Intent intent =
            new Intent(this, Mulaihitung.class);

            startActivity(intent);

```

```

        }
        if (v == btninformasi)
    {
        Intent intent =
new Intent(this,
Informasiyam.class);

        startActivity(intent);
    }
    if (v == btnabout) {
        Intent intent =
new Intent(this, About.class);

        startActivity(intent);
    }
    if (v == btnkeluar) {
        close();
    }
}

@Override
public boolean
onCreateOptionsMenu(Menu menu)
{
    // Inflate the menu;
    this adds items to the action bar if it
    is present.

        getMenuInflater().inflate(R.m
enu.main, menu);

        return true;
    }

    public void close() {
        AlertDialog.Builder
builder = new
AlertDialog.Builder(this);

        builder.setMessage("Anda
Yakin Ingin Keluar dari Aplikasi?")

        .setCancelable(false)

        .setPositiveButton("YA",
new
DialogInterface.OnClickListener() {

            public void
onClick(DialogInterface dialog, int
id) {

                MainActivity.this.finish();

            }

        })

        .setNegativeButton("TIDAK"
,

```

```

        new
DialogInterface.OnClickListener() {

        public void
onClick(DialogInterface dialog, int
id) {

        dialog.cancel();

        }
}).show();

}

public boolean
onKeyDown(int keyCode, KeyEvent
event) {

        if (keyCode ==
KeyEvent.KEYCODE_BACK) {

                close();

        }

        return
super.onKeyDown(keyCode, event);

}
}

```

informasi

```

package
com.example.brianspratama;

```

```

import android.os.Bundle;
import android.app.Activity;
import android.app.AlertDialog;
import android.view.View;

```

```

import android.widget.*;

```

```

import
android.content.DialogInterface;

```

```

public class Informasi1 extends
Activity {
    EditText idternak,idpakan,ket;
    Button Cek;
    Button hapus;
    double bj,hb;
    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_inf
ormasi1);
        idternak=(EditText)
findViewById(R.id.ternak);
        idpakan=(EditText)
findViewById(R.id.pakan);
        Cek=(Button)
findViewById(R.id.bncek);
    }
    public void buttonHandler(View
view){
        if(idternak.getText().length()
== 0) {
            Toast.makeText(this,
"Angka pertama
masih kosong!",
            Toast.LENGTH_LONG).sho
w();

            return;
        }
    }
}

```

```

bj=Integer.parseInt(idternak.getText(
).toString());
double hb=bj*25;
double pkn=bj*0.1;
idpakan.setText(""+pkn);

```

```

AlertDialog.Builder builder =

```

```

        new
AlertDialog.Builder(this);

builder.setTitle("hasil");

builder.setMessage("jumlah butir
telur yang dihasilkan per bulan =
"+String.valueOf(hb))

.setPositiveButton("OK",
        new
DialogInterface.OnClickListener() {

            @Override
            public void
onClick(DialogInterface dialog,
            int which) {
                // TODO Auto-
generated method stub
                idternak.setText("");
                dialog.dismiss();
            }
        });
AlertDialog
dialogHasil = builder.create();
        dialogHasil.show();
    }

}

```

Mulai hitung

```

package
com.example.brianspratama;

import android.os.Bundle;
import android.app.Activity;
import android.app.AlertDialog;
import
android.content.DialogInterface;
import android.view.View;
import android.widget.*;

```

```

public class Mulaihitung extends
Activity {
    EditText feseskerbau,
jagunghalus, tepungikan, dedakhalus,
minyakmakan,
                tepungtapioka,
totala, totalkeb, me;
    Button hitung, hapus;
    String mak;

    @Override
    protected void
onCreate(Bundle savedInstanceState)
    {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.acti
vity_mulaihitung);
        feseskerbau =
(EditText)
findViewById(R.id.feseskerbau);
        jagunghalus =
(EditText)
findViewById(R.id.jagughalus);
        tepungikan =
(EditText)
findViewById(R.id.tepungikan);
        dedakhalus =
(EditText)
findViewById(R.id.dedakhalus);
        minyakmakan =
(EditText)
findViewById(R.id.minyakmakan);
        tepungtapioka =
(EditText)
findViewById(R.id.tepungtapioka);
        totala = (EditText)
findViewById(R.id.total);
        totalkeb = (EditText)
findViewById(R.id.totalkeb);
        me = (EditText)
findViewById(R.id.me);

```

```

        hitung = (Button)
findViewById(R.id.btnhitung);
        hapus = (Button)
findViewById(R.id.btnhapus2);
    }

    public void
buttonHandler(View view) {
    // TODO Auto-
generated method stub
    if
(feseskerbau.getText().length() == 0)
{
        Toast.makeText(this, "Angka
fesek kerbau masih kosong!",
        Toast.LENGTH_LONG).sho
w();
        return;
    }
    if
(jagunghalus.getText().length() == 0)
{
        Toast.makeText(this, "Angka
jagung halus masih kosong!",
        Toast.LENGTH_LONG).sho
w();
        return;
    }
    if
(tepungikan.getText().length() == 0)
{
        Toast.makeText(this, "Angka
tepung ikan masih kosong!",
        Toast.LENGTH_LONG).sho
w();
        return;
    }
    if
(dedakhalus.getText().length() == 0)
{

```

```

        Toast.makeText(this, "Angka
dedak halus masih kosong!",
        Toast.LENGTH_LONG).sho
w();
        return;
    }
    if
(minyakmakan.getText().length() ==
0) {
        Toast.makeText(this, "Angka
minyak makan masih kosong!",
        Toast.LENGTH_LONG).sho
w();
        return;
    }
    if
(tepungtapioka.getText().length() ==
0) {
        Toast.makeText(this, "Angka
tepung tapioka masih kosong!",
        Toast.LENGTH_LONG).sho
w();
        return;
    }
    if
(totaal.getText().length() == 0) {
        Toast.makeText(this, "Angka
total kebutuhan masih kosong!",
        Toast.LENGTH_LONG).sho
w();
        return;
    }
    double fk =
Integer.parseInt(feseskerbau.getText
().toString());
    double jh =
Integer.parseInt(jagunghalus.getText
().toString());

```

```

        double ti =
Integer.parseInt(tepungikan.getText(
).toString());
        double dh =
Integer.parseInt(dedakhalus.getText(
).toString());
        double mm =
Integer.parseInt(minyakmakan.getTe
xt().toString());
        double tt =
Integer.parseInt(tepungtapioka.getTe
xt().toString());
        double tot =
Integer.parseInt(totaal.getText().toSt
ring());
        double jmlfk = fk / tot
* 0;
        double jmljh = jh / tot
* 3370;
        double jmlti = ti / tot
* 3080;
        double jml dh = dh /
tot * 1630;
        double jmlmm = mm
/ tot * 8450;
        double jmltt = tt / tot
* 3720;
        double jumlahme =
jmlfk + jmljh + jmlti + jml dh +
jmlmm + jmltt;
        if (jumlahme >= 2890
&& jumlahme <= 2910)
            mak =
"maksimal";
        else if (jumlahme >=
0 && jumlahme <= 2889)
            mak = "belum
maksimal";
        else if (jumlahme >=
2911)
            mak = "belum
maksimal";
        else if
(jumlahme >= 2850 &&
jumlahme <= 2889)

```

```

            mak = "Hampir
Maksimal";
        double total = fk + jh
+ ti + dh + mm + tt;
        me.setText("" +
jumlahme);
        totalkeb.setText("" +
total);
        AlertDialog.Builder
builder = new
AlertDialog.Builder(this);
        builder.setTitle("Keterangan"
);
        builder.setMessage(
"ME =
" + String.valueOf(jumlahme) +
String.valueOf(mak))
        .setPositiveButton("OK",
new
DialogInterface.OnClickListener() {
            @Override
            public void
onClick(DialogInterface dialog, int
which) {
                // TODO Auto-generated
method stub
                me.setText("");
                dialog.dismiss();
            }
        });
        AlertDialog
dialogHasil = builder.create();
        dialogHasil.show();
        hapus.setOnClickListener(ne
w View.OnClickListener() {
            @Override

```



```

        Toast.makeText(getBaseContext(), "" + itemText,
            Toast.LENGTH_LONG).show();
        if (itemText.equals("Jagung Halus")) {
            Intent JHalus = getIntent();
            JHalus = new Intent(Keteranganpakan.this,
                Jagunghalus.class);
            startActivity(JHalus);
        }
        if (itemText.equals("Tepung Ikan")) {
            Intent Tikan = getIntent();
            Tikan = new Intent(Keteranganpakan.this,
                Tpikan.class);
            startActivity(Tikan);
        }
        if (itemText.equals("Dedak Halus")) {
            Intent DHalus = getIntent();
            DHalus = new Intent(Keteranganpakan.this,
                Dhalus.class);
            startActivity(DHalus);
        }
        if (itemText.equals("Minyak Makan")) {
            Intent MMakan = getIntent();
            MMakan = new Intent(Keteranganpakan.this,
                Mmakan.class);
            startActivity(MMakan);
        }
        if (itemText.equals("Tepung Tapioka")) {
            Intent TTapioka = getIntent();
            TTapioka = new Intent(Keteranganpakan.this,
                Ttapioka.class);
            startActivity(TTapioka);
        }
        if (itemText.equals("Feses Kerbau")) {
            Intent FKerbau = getIntent();
            FKerbau = new Intent(Keteranganpakan.this,
                Fkerbau.class);
            startActivity(FKerbau);
        }
        if (itemText.equals("Biji Kedelai")) {
            Intent BKedelai = getIntent();
            BKedelai = new Intent(Keteranganpakan.this,
                Bkedelai.class);
            startActivity(BKedelai);
        }
    });
}

@Override
public void onClick(View arg0) {

```

```

        // TODO Auto-
generated method stub
    }
}

Pakan ke ternak
package
com.example.brianspratama;

import android.os.Bundle;
import android.app.Activity;
import android.app.AlertDialog;
import android.view.View;
import android.widget.*;

import
android.content.DialogInterface;

public class PakanTernak extends
Activity {
    EditText idternak,idpakan,ket;
    Button Cek1;
    Button hapus;
    double bj,hb;
    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_pa
kan_ternak);
        idternak=(EditText)
findViewById(R.id.ternak1);
        idpakan=(EditText)
findViewById(R.id.pakan1);
        Cek1=(Button)
findViewById(R.id.cek1);
    }

    public void buttonHandler(View
view){
        if(idpakan.getText().length()
== 0) {

```

```

        Toast.makeText(this,
"Angka pertama
masih kosong!",
        Toast.LENGTH_LONG).sho
w();
        return;
    }

    hb=Integer.parseInt(idpakan.getText
().toString());
    double bj=hb*0.1;
    double trnak=bj*25;
    idternak.setText(""+trnak);

    AlertDialog.Builder builder =
        new
AlertDialog.Builder(this);
    builder.setTitle("hasil");

    builder.setMessage("jumlah ternak =
"+String.valueOf(bj))

    .setPositiveButton("OK",
        new
DialogInterface.OnClickListener() {

        @Override
        public void
onClick(DialogInterface dialog,
int which) {
            // TODO Auto-
generated method stub
            idternak.setText("");
            dialog.dismiss();
        }
    });
    AlertDialog
dialogHasil = builder.create();
    dialogHasil.show();
}
}

```