

repository.ub.ac.id

**PENGEMBANGAN APLIKASI PENYUSUNAN RANSUM UNTUK
SAPI POTONG (SUNRAN) BERBASIS WEB DENGAN METODE
PENYUSUNAN RANSUM *PEARSON SQUARE* DAN
IMPLEMENTASI *YII FRAMEWORK***

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Andrian Riza Hermawan

NIM: 105060800111028



PROGRAM STUDI INFORMATIKA/ILMU KOMPUTER
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

MALANG

2016

PENGESAHAN

PENGEMBANGAN APLIKASI PENYUSUNAN RANSUM UNTUK SAPI POTONG
(SUNRAN) BERBASIS WEB DENGAN METODE PENYUSUNAN RANSUM *PEARSON*
SQUARE DAN IMPLEMENTASI *YII FRAMEWORK*

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Andrian Riza Hermawan

NIM: 105060800111028

Skripsi ini telah diuji dan dinyatakan lulus pada
18 Januari 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Ismiarta Aknuranda, S.T., M.Sc., Ph.D.

NIK: 740719 06 1 1 0079

Aditya Rachmadi, S.ST., M.TI

NIK: 860421 16 1 1 0426

Mengetahui

Ketua Program Studi Informatika/Ilmu Komputer

Drs. Marji, MT.

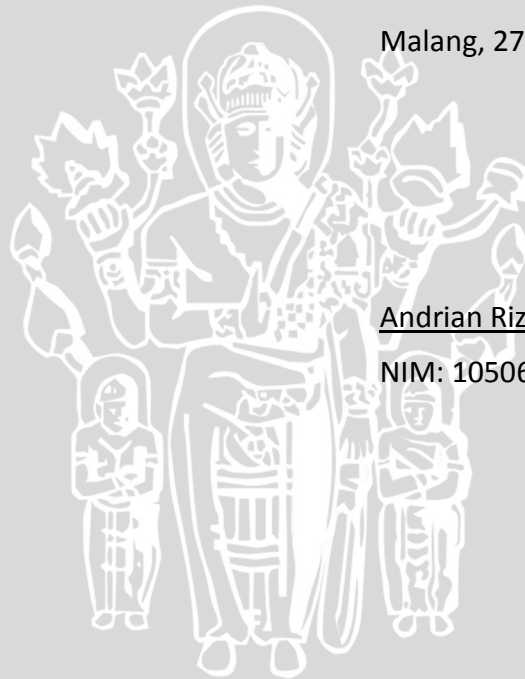
NIP: 19670801 199203 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 27 Januari 2016



Andrian Riza Hermawan

NIM: 105060800111028

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT karena berkat rahmat dan hidayah-Nya, penulis mampu menyelesaikan pengerjaan skripsi yang berjudul “Pengembangan Aplikasi Penyusunan Ransum untuk Sapi Potong (SunRan) Berbasis Web dengan Metode Penyusunan Ransum *Pearson Square* dan Implementasi YII *Framework*”. Penulisan skripsi ini bertujuan untuk memenuhi sebagian syarat dalam mendapatkan gelar Sarjana Komputer.

Pengerjaan skripsi ini telah mendapatkan dukungan serta bantuan dari berbagai pihak. Penulis menyampaikan ucapan terima kasih kepada:

1. Bapak Ismiarta Aknuranda, ST., M.Sc., Ph.D. selaku dosen pembimbing I serta dosen pembimbing akademik yang telah memberikan saran, dukungan dan arahan mengenai akademik serta membimbing penulis dalam pengerjaan skripsi ini.
2. Bapak Aditya Rachmadi, S.ST., M.TI selaku pembimbing II yang juga membimbing penulis dalam pengerjaan skripsi ini.
3. Ibu Dr. Ir. Herni Sudarwati, MS (dosen Fakultas Peternakan Universitas Brawijaya) selaku pihak pakar yang telah memberikan ilmu dalam bidang peternakan dalam pengerjaan skripsi ini.
4. Bapak Drs. Marji, MT. selaku Ketua Program Studi Informatika/Ilmu Komputer Program Teknologi Informasi dan Ilmu Komputer.
5. Bapak Denny Sagita Rusdianto, S.Kom., M.Kom. selaku Kepala Laboratorium Rekayasa Perangkat Lunak Program Teknologi Informasi dan Ilmu Komputer.
6. Segenap bapak dan ibu dosen Fakultas Ilmu Komputer Universitas Brawijaya yang telah memberikan ilmu pengetahuan selama penulis menempuh pendidikan di Fakultas Ilmu Komputer Universitas Brawijaya.
7. Ayahanda dan ibunda yang telah memberikan dukungan baik secara moril maupun materiil.

Penulis menyadari bahwa skripsi ini masih terdapat banyak kekurangan serta jauh dari kesempurnaan, sehingga penulis mengharapkan kritik dan saran agar bisa menyempurnakan skripsi ini. Penulis juga mengharapkan semoga skripsi ini bermanfaat kepada semua pihak, khususnya kepada penulis sendiri, kelompok peternak sapi potong dan industr yang melakukan penggemukan sapi potong.

Malang, 11 Januari 2016

Penulis

rhyan.3370@gmail.com

ABSTRAK

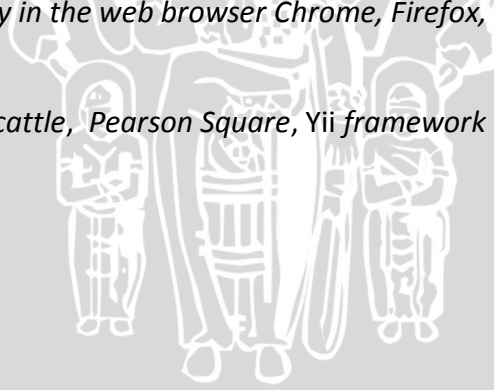
Ransum merupakan jumlah pakan yang diberikan pada ternak dalam 24 jam. Ransum yang ideal tidak hanya bisa memenuhi kebutuhan nutrisi harian, namun bisa juga untuk meningkatkan jumlah pertambahan berat badan sapi potong dalam sehari. Permasalahan yang muncul mengenai ransum yang ideal adalah penyusunannya yang tidak mudah jika dilakukan secara manual. Permasalahan tersebut menyebabkan penulis membuat aplikasi penyusun ransum ideal. Aplikasi yang dibuat ini menggunakan metode Pearson Square selama proses penyusunan ransum ideal. Pembuatan aplikasi penyusun ransum ini bertujuan agar tersusunnya ransum yang memenuhi kebutuhan nutrisi harian sapi potong yang sesuai dengan fisiologis hewan ternak. Aplikasi penyusun ransum ini merupakan aplikasi berbasis web yang dibangun menggunakan framework Yii di aplikasi pengembangan Netbeans IDE 8.0.2. Pengujian aplikasi penyusun ransum ideal ini dilakukan tiga kali, yaitu pengujian validasi umum, validasi implementasi Pearson Square, dan kompatibilitas web browser. Hasil pengujian validasi umum menunjukkan bahwa fungsi-fungsi yang ada di dalam aplikasi penyusun ransum telah berjalan dengan baik dan benar. Hasil pengujian validasi implementasi Pearson Square menunjukkan bahwa 2 dari 5 kasus uji memiliki nutrisi hasil penyusunan ransum oleh aplikasi SunRan dan oleh Microsoft Excel bernilai sama pada tingkat ketelitian 10⁻² (dalam satuan yang sama) dan juga menunjukkan persentase kesalahan yang kecil pada tingkat ketelitian 10⁻⁷. Pengujian kompatibilitas web browser menghasilkan kesimpulan bahwa aplikasi penyusun ransum ini menampilkan seluruh antarmuka dengan baik dan benar pada web browser Chrome, Firefox, dan Opera.

Kata kunci: ransum, sapi potong, *Pearson Square*, *framework* Yii

ABSTRACT

Ration is the amount of forage in 24 hours. Ideal Rations not only meet the daily nutritional needs, but also for improving the sheer number of beef cattle weight gain in a day. The problem that arise regarding the ideal rations is formulation. Formulation of ideal rations is difficult and need a lot of time to done manually. That problem lead author to develop an ideal rations formulation application. Rations formulation application which developed by author using Pearson's square during the process of formulating the ideal ration. Purpose of developing the application is rations that formulated by aplication meet the daily nutritional needs of beef cattle based from physiological condition. Rations formulation application which developed by authors is a web-based application developed using Yii framework in an development application NetBeans IDE 8.0.2. Testing processes of application is done 3 times. Three tests to application are the general validation test, Pearson Square implementation validation test, and web browser compatibility test. General validation testing results show that the functions in the rations formulation application has worked well and correctly. Result of Pearson Square implementation validation testing show that 2 of 5 cases have nutrients from the formulated rations by the application SunRan and by Microsoft Excel equal values at 10-2 accuracy level (in the same units) and also to show small percentage of errors at 10-7 accuracy level. Result of web browser compability test concluded that the rations formulation application display the entire interface correctly in the web browser Chrome, Firefox, and Opera.

Keywords: ration, beef cattle, Pearson Square, Yii framework



DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xii
DAFTAR LAMPIRAN	xv
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	2
1.5 Batasan masalah	2
1.6 Sistematika pembahasan.....	3
BAB 2 DASAR TEORI	5
2.1 Ransum	5
2.2 <i>Pearson square</i>	6
2.3 <i>Framework Yii</i>	8
2.4 <i>Unified Modeling Language (UML)</i>	9
2.5 Pengujian <i>black-box</i>	12
BAB 3 METODOLOGI PENELITIAN	14
3.1 Studi literatur.....	14
3.2 Analisa kebutuhan	15
3.3 Perancangan	15
3.4 Implementasi	15
3.5 Pengujian	16
3.6 Penulisan laporan	16
BAB 4 ANALISA KEBUTUHAN.....	17



4.1 Analisa data.....	17
4.2 Identifikasi aktor	17
4.3 Penyusunan daftar kebutuhan	18
4.4 Pemodelan <i>use case</i>	19
BAB 5 PERANCANGAN DAN IMPLEMENTASI	23
5.1 Perancangan	23
5.1.1 Penentuan kerangka kerja	23
5.1.2 Perancangan dengan <i>framework</i> Yii	24
5.2 Implementasi	53
5.2.1 Spesifikasi perangkat pengembang sistem.....	54
5.2.2 Implementasi fitur sistem	55
5.2.3 Implementasi kalkulasi <i>Pearson Square</i>	72
5.2.4 Implementasi basis data	72
5.2.5 Implementasi antarmuka.....	73
BAB 6 PENGUJIAN DAN ANALISA	83
6.1 Pengujian	83
6.1.1 Pengujian validasi umum	84
6.1.2 Pengujian validasi implementasi <i>Pearson Square</i>	90
6.1.3 Pengujian komparabilitas <i>web browser</i>	93
6.2 Analisa.....	94
6.2.1 Analisa pengujian validasi umum.....	94
6.2.2 Analisa pengujian implementasi <i>Pearson Square</i>	95
6.2.3 Analisa pengujian komparabilitas <i>web browser</i>	96
BAB 7 Penutup	97
7.1 Kesimpulan.....	97
7.2 Saran	97
DAFTAR PUSTAKA.....	99
LAMPIRAN A HASIL WAWANCARA PIHAK PAKAR.....	100
LAMPIRAN B SPESIFIKASI <i>USE CASE</i>	102
LAMPIRAN C DAFTAR NUTRISI HARIAN SAPI POTONG	111
LAMPIRAN D DAFTAR BAHAN PENYUSUN RANSUM	115
LAMPIRAN E HASIL PENGUJIAN VALIDASI IMPLEMENTASI <i>PEARSON SQUARE</i>..	117





DAFTAR TABEL

Tabel 2.1 Daftar beberapa bahan pakan penyusun ransum beserta nutrisinya	6
Tabel 2.2 Hasil perhitungan contoh penggunaan metode <i>Pearson Square</i>	7
Tabel 2.3 Perbandingan performa beberapa <i>framework</i> PHP.....	8
Tabel 2.4 Deskripsi komponen diagram <i>use case</i>	10
Tabel 2.5 Deskripsi simbol-simbol diagram kelas	11
Tabel 2.6 Deskripsi simbol-simbol <i>sequence diagram</i>	12
Tabel 4.1 Daftar aktor	18
Tabel 4.2 Daftar kebutuhan fungsional.....	18
Tabel 4.3 Daftar kebutuhan non-fungsional	19
Tabel 4.4 Deskripsi singkat <i>use case</i> mendaftarkan diri	20
Tabel 4.5 Deskripsi singkat <i>use case login</i>	20
Tabel 4.6 Deskripsi singkat <i>use case</i> melihat tabel nutrisi.....	21
Tabel 4.7 Deskripsi singkat <i>use case</i> melihat tabel komposisi ransum	21
Tabel 4.8 Deskripsi singkat <i>use case</i> menyusun ransum	21
Tabel 4.9 Deskripsi singkat <i>use case</i> mengatur hasil penyusunan ransum	21
Tabel 4.10 Deskripsi singkat <i>use case</i> membandingkan ransum.....	22
Tabel 4.11 Deskripsi singkat <i>use case logout</i>	22
Tabel 5.1 Spesifikasi perangkat lunak pengembang sistem	55
Tabel 5.2 Spesifikasi perangkat keras pengembang sistem.....	55
Tabel 6.1 Skenario kasus uji fungsi melihat tabel nutrisi dengan skenario benar	84
Tabel 6.2 Skenario kasus uji fungsi melihat tabel komposisi penyusun ransum dengan skenario benar.....	84
Tabel 6.3 Skenario kasus uji fungsi menyusun ransum dengan skenario benar .	84
Tabel 6.4 Skenario kasus uji fungsi menyusun ransum dengan skenario salah ...	85
Tabel 6.5 Skenario kasus uji fungsi mendaftarkan diri dengan skenario benar ...	85
Tabel 6.6 Skenario kasus uji fungsi mendaftarkan diri dengan skenario salah	86
Tabel 6.7 Skenario Kasus Uji Fungsi <i>Login / Masuk ke Sistem</i> dengan Skenario Benar	86
Tabel 6.8 Skenario Kasus Uji Fungsi <i>Login / Masuk ke Sistem</i> dengan Skenario Salah	86



Tabel 6.9 Skenario Kasus Uji Fungsi Mengatur Hasil Penyusunan Ransum dengan Skenario Benar	87
Tabel 6.10 Skenario Kasus Uji Fungsi Mengatur Hasil Penyusunan Ransum dengan Skenario Salah	87
Tabel 6.11 Skenario Kasus Uji Fungsi Membandingkan Ransum dengan Skenario Benar	88
Tabel 6.12 Skenario Kasus Uji Fungsi Membandingkan Ransum dengan Skenario Benar	88
Tabel 6.13 Skenario Kasus Uji Fungsi <i>Logout</i> / Keluar dari Sistem dengan Skenario Benar	88
Tabel 6.14 Hasil Pengujian Validitas	89
Tabel 6.15 Variabel tetap pengujian validasi implementasi <i>Pearson Square</i>	90
Tabel 6.16 Bahan hijauan, konsentrat dan keberhasilan menemukan ransum ideal pada pengujian validasi implementasi <i>Pearson Square</i>	91
Tabel 6.17 Hasil uji validitas implementasi <i>Pearson Square</i>	91
Tabel 6.18 Detail hasil penyusunan ransum dari TC01 melalui aplikasi SunRan..	91
Tabel 6.19 Detail hasil penyusunan ransum dari TC01 melalui Microsoft Excel ..	92
Tabel 6.20 Berat bahan penyusun dalam hasil penyusunan ransum dari TC01 melalui Microsoft Excel.....	92
Tabel 6.21 Persentase kesalahan dari TC01.....	92
Tabel 6.22 Persentase kesalahan hasil penyusunan ransum dari kasus uji.....	93
Tabel 6.23 Uji Validasi Kompabilitas pada <i>Web Browser</i> Chrome.....	93
Tabel 6.24 Uji Validasi Kompabilitas pada <i>Web Browser</i> Firefox	94
Tabel 6.25 Uji Validasi Kompabilitas pada <i>Web Browser</i> Opera.....	94
Tabel 6.26 Detail hasil penyusunan ransum dari TC06 melalui Microsoft Excel ..	95
Tabel 6.27 Berat bahan penyusun dalam hasil penyusunan ransum dari TC06 melalui Microsoft Excel.....	96



DAFTAR GAMBAR

Gambar 2.1 Struktur statis aplikasi Yii	9
Gambar 2.2 Contoh diagram <i>use case</i>	10
Gambar 2.3 Contoh diagram kelas.....	11
Gambar 2.4 Contoh <i>sequence diagram</i>	12
Gambar 3.1 Diagram alir penelitian.....	14
Gambar 3.2 <i>Timeline</i> pengerjaan skripsi.....	16
Gambar 4.1 Diagram analisa kebutuhan	17
Gambar 4.2 <i>Use case</i> Aplikasi Penyusunan Ransum (SunRan)	20
Gambar 5.1 Diagram perancangan	23
Gambar 5.2 Arsitektur aplikasi SunRan	24
Gambar 5.3 Diagram kelas <i>controller</i> aplikasi sunran	25
Gambar 5.4 Diagram kelas <i>model</i> aplikasi sunran.....	26
Gambar 5.5 Diagram kelas aplikasi SunRan	26
Gambar 5.6 Diagram urutan fungsi mendaftarkan diri	27
Gambar 5.7 Diagram urutan fungsi <i>login</i>	28
Gambar 5.8 Diagram urutan fungsi melihat tabel nutrisi.....	28
Gambar 5.9 Diagram urutan fungsi melihat tabel komposisi ransum.....	29
Gambar 5.10 Diagram urutan fungsi menyusun ransum	29
Gambar 5.11 Diagram urutan fungsi mengatur hasil penyusunan ransum (ubah nama ransum)	30
Gambar 5.12 Diagram urutan fungsi mengatur hasil penyusunan ransum (lihat detail)	31
Gambar 5.13 Diagram urutan fungsi mengatur hasil penyusunan ransum (hapus ransum)	32
Gambar 5.14 Diagram urutan fungsi membandingkan ransum	33
Gambar 5.15 Diagram urutan fungsi <i>logout</i>	34
Gambar 5.16 Diagram <i>entity-relationship</i> penyusunan ransum pada aplikasi SunRan.....	35
Gambar 5.17 Diagram <i>entity-relationship</i> bahan penyusun ransum pada aplikasi SunRan.....	36
Gambar 5.18 <i>Site map</i> aplikasi SunRan	36

Gambar 5.19 Perancangan antarmuka halaman beranda sistem SunRan	37
Gambar 5.20 Perancangan antarmuka nutrisi harian.....	38
Gambar 5.21 Perancangan antarmuka halaman bahan penyusun	39
Gambar 5.22 Perancangan antarmuka halaman pemilihan bahan untuk penyusunan ransum.....	40
Gambar 5.23 Perancangan antarmuka halaman notifikasi hasil penyusunan ransum	41
Gambar 5.24 Perancangan antarmuka halaman detail hasil penyusunan	42
Gambar 5.25 Perancangan antarmuka halaman daftar hasil penyusunan ransum tersimpan	44
Gambar 5.26 Perancangan antarmuka halaman detail hasil ransum penyusunan tersimpan	45
Gambar 5.27 Perancangan antarmuka halaman <i>update</i> hasil penyusunan ransum tersimpan	47
Gambar 5.28 Perancangan antarmuka halaman daftar hasil penyusunan ransum tersimpan untuk komparasi	48
Gambar 5.29 Perancangan antarmuka halaman detail komparasi hasil penyusunan ransum tersimpan	50
Gambar 5.30 Perancangan antarmuka halaman masuk.....	52
Gambar 5.31 Perancangan antarmuka halaman daftar	53
Gambar 5.32 Diagram implementasi	54
Gambar 5.33 Implementasi fitur pendaftaran.....	56
Gambar 5.34 Implementasi fitur <i>login</i>	57
Gambar 5.35 Implementasi fitur menampilkan tabel nutrisi	57
Gambar 5.36 Implementasi fitur menampilkan tabel bahan penyusun ransum .	58
Gambar 5.37 Implementasi fitur penyusunan ransum bagian pemilihan bahan .	58
Gambar 5.38 <i>Flowchart</i> tahap verifikasi data yang dipilih (bagian 1)	59
Gambar 5.39 <i>Flowchart</i> tahap verifikasi data yang dipilih (bagian 2)	60
Gambar 5.40 <i>Flowchart</i> tahap kalkulasi <i>Pearson Square</i> (bagian 1)	62
Gambar 5.41 <i>Flowchart</i> tahap kalkulasi <i>Pearson Square</i> (bagian 2)	63
Gambar 5.42 <i>Flowchart</i> tahap kalkulasi <i>Pearson Square</i> (bagian 3)	64
Gambar 5.43 <i>Flowchart</i> tahap menyimpan hasil penyusunan ransum (bagian 1)	67
Gambar 5.44 <i>Flowchart</i> tahap menyimpan hasil penyusunan ransum (bagian 2)	68



Gambar 5.45 Implementasi fitur manajemen hasil penyusunan ransum bagian menampilkan daftar hasil penyusunan ransum.....	70
Gambar 5.46 Implementasi fitur manajemen hasil penyusunan ransum bagian melihat detail hasil penyusunan ransum	70
Gambar 5.47 Implementasi fitur manajemen hasil penyusunan ransum bagian <i>update</i> hasil penyusunan ransum	70
Gambar 5.48 Implementasi fitur manajemen hasil penyusunan ransum bagian hapus hasil penyusunan ransum.....	71
Gambar 5.49 Implementasi fitur komparasi ransum bagian menampilkan daftar hasil penyusunan ransum	71
Gambar 5.50 Implementasi fitur komparasi ransum bagian lihat detail perbandingan hasil penyusunan ransum.....	71
Gambar 5.51 Implementasi fitur <i>logout</i>	72
Gambar 5.52 Implementasi basis data	73
Gambar 5.53 Implementasi antarmuka beranda.....	74
Gambar 5.54 Implementasi antarmuka daftar nutrisi harian.....	75
Gambar 5.55 Implementasi antarmuka bahan penyusun	75
Gambar 5.56 Implementasi antarmuka pemilihan bahan untuk penyusunan ransum	76
Gambar 5.57 Implementasi antarmuka notifikasi hasil penyusunan ransum.....	77
Gambar 5.58 Implementasi antarmuka detail hasil penyusunan.....	77
Gambar 5.59 Implementasi antarmuka daftar hasil penyusunan ransum tersimpan	78
Gambar 5.60 Implementasi antarmuka detail hasil penyusunan ransum tersimpan	79
Gambar 5.61 Implementasi antarmuka <i>update</i> hasil penyusunan ransum tersimpan	80
Gambar 5.62 Implementasi antarmuka daftar hasil penyusunan ransum tersimpan untuk komparasi	80
Gambar 5.63 Implementasi antarmuka detail komparasi hasil penyusunan ransum tersimpan	81
Gambar 5.64 Implementasi antarmuka masuk	82
Gambar 5.65 Implementasi antarmuka daftar	82
Gambar 6.1 Diagram pengujian dan analisa	83



DAFTAR LAMPIRAN

LAMPIRAN A HASIL WAWANCARA PIHAK PAKAR	100
LAMPIRAN B SPESIFIKASI <i>USE CASE</i>	102
B.1 <i>Use case</i> melihat tabel nutrisi	102
B.2 <i>Use case</i> melihat tabel komposisi ransum	102
B.3 <i>Use case</i> menyusun ransum	103
B.4 <i>Use case</i> mendaftarkan diri	105
B.5 <i>Use case login</i>	106
B.6 <i>Use case</i> mengatur hasil penyusunan ransum	107
B.7 <i>Use case</i> membandingkan ransum.....	108
B.8 <i>Use case logout</i>	109
LAMPIRAN C DAFTAR NUTRISI HARIAN SAPI POTONG	111
LAMPIRAN D DAFTAR BAHAN PENYUSUN RANSUM	115
LAMPIRAN E HASIL PENGUJIAN VALIDASI IMPLEMENTASI <i>PEARSON SQUARE</i> ..	117
E.1 Detail hasil penyusunan ransum dan persentase kesalahan dari TC02	117
E.2 Detail hasil penyusunan ransum dan persentase kesalahan dari TC03	118
E.3 Detail hasil penyusunan ransum dari dan persentase kesalahan TC04	119
E.4 Detail hasil penyusunan ransum dan persentase kesalahan dari TC05	121
LAMPIRAN F KELAS <i>CONTROLLER</i> DAN <i>MODEL</i> TIAP FUNGSI	123
F.1 Kelas <i>controller</i> dan <i>model</i> pada fungsi melihat tabel nutrisi	123
F.2 Kelas <i>controller</i> dan <i>model</i> pada fungsi melihat tabel komposisi	123
F.3 Kelas <i>controller</i> dan <i>model</i> pada fungsi menyusun ransum.....	124
F.4 Kelas <i>controller</i> dan <i>model</i> pada fungsi mendaftarkan diri.....	124
F.5 Kelas <i>controller</i> dan <i>model</i> pada fungsi <i>login</i>	124
F.6 Kelas <i>controller</i> dan <i>model</i> pada fungsi mengatur dan membandingkan ransum	125
F.7 Kelas <i>controller</i> dan <i>model</i> pada fungsi <i>logout</i>	125

BAB 1 PENDAHULUAN

1.1 Latar belakang

Hasil produksi ternak pada proses penggemukan hewan ternak sapi potong dibidang peternakan ditentukan oleh beberapa faktor penentu. Salah satu faktor penentu yang dimaksud yaitu kualitas pakan ternak (ransum) (Yulianto, 2012). Baik buruknya kualitas pakan ternak ditentukan oleh nutrisi yang dikandungnya. Kandungan nutrisi di dalam ransum seperti BK (Bahan Kering), PK (Protein Kasar), dan TDN (*Total Digestible Protein*). Ransum yang memenuhi nutrisi hewan ternak dapat meningkatkan kualitas maupun kuantitas hasil ternak. Peningkatan kuantitas hasil ternak berupa bertambahnya berat daging yang di produksi oleh hewan ternak. Peningkatan kualitas hasil ternak berupa daging sapi yang sehat dan bergizi. Ransum yang diberikan kepada hewan ternak diberikan secara berkelanjutan & teratur selama jangka waktu tertentu. Porsi ransum yang disusun sebanyak porsi harian si hewan ternak. Oleh karena itu, pemberian pakan ternak dianjurkan untuk memenuhi nutrisi harian hewan ternak.

Namun, pada umumnya peternak hanya memberikan pakan ternak semampu mereka. Sebagai contoh, mereka hanya memberikan hijauan (rumput-rumputan, dan lain sebagainya), sedangkan untuk memacu pertumbuhan bobot harian sapi yang tinggi, diperlukan pakan konsentrat sebagai pakan penguat (Yulianto, 2012). Alasan pakan konsentrat dianggap sebagai bahan penguat adalah pakan ini memiliki protein kasar yang tinggi (>20%). Protein yang tinggi ini nantinya memacu pertumbuhan hewan ternak. Sapi yang memperoleh pakan berupa rumput lapangan dan jerami padi akan meningkat berat badannya sebanyak 0,24kg/ekor/hari sedangkan jika ditambahkan konsentrat akan bertambah sebanyak 0,65kg/ekor/hari (Yulianto, 2012). Berdasarkan fakta tersebut, dapat diketahui bahwa efek dari pemberian pakan ternak (ransum) yang tidak sesuai dengan nutrisi yang diperlukan dalam sehari semalam akan mempengaruhi hasil produksi ternak.

Pemberian pakan ternak (ransum) yang sesuai dengan nutrisi yang diperlukan dalam sehari semalam (atau biasa disebut dengan ransum ideal) memerlukan suatu metode penyusunan ransum. Salah satu metode penyusunan ransum yaitu *Pearson Square*. Metode *Pearson Square* adalah metode yang sudah biasa dipakai untuk menyeimbangkan ransum (Umiyasih, 2007). Metode ini merupakan metode penyusunan ransum ideal yang mudah dan sederhana (Yulianto, 2012).

Fakta mengenai pengaruh pemberian pakan ternak terhadap produksi ternak menunjukkan bahwa diperlukan aplikasi yang memiliki fungsi utama menyusun ransum. Aplikasi ini dapat menerapkan metode penyusunan *Pearson Square* yang banyak dipakai, mudah, dan sederhana. Aplikasi penyusun ransum tersebut juga penulis sarankan berbasis web.

Aplikasi berbasis web merupakan aplikasi yang menggunakan akses internet. Aplikasi ini mempermudah penggunaannya saat pemakaian aplikasi karena

pengguna tidak memerlukan untuk menginstalasi aplikasi ke media akses baik berupa komputer maupun telepon genggam. Aplikasi ini juga memperingan media akses dikarenakan sumber daya yang terpakai sedikit. Penyebabnya yaitu proses yang terjadi dibebankan ke *server*. Keuntungan-keuntungan dari aplikasi berbasis web diatas membuat pengguna internet baik yang masih awam IT maupun mahir dibidang IT di seluruh dunia khususnya di Indonesia meningkat tiap tahunnya. Peningkatan yang dicapai pada tahun 2013 mencapai 13% dari tahun sebelumnya atau sebanyak sekitar 63 juta orang (pada tahun 2012) menjadi 71,19 juta orang ditahun 2013. Peningkatan ini mempermudah untuk penggunaan aplikasi berbasis web oleh pengguna internet diseluruh Indonesia.

Berdasarkan berbagai pertimbangan tersebut, penulis sebagai mahasiswa Program Teknik Informatika dan Ilmu Komputer (PTIIK) terdorong untuk membantu peternak untuk memudahkan merancang ransum untuk hewan ternak mereka. Bentuk bantuan dari penulis yaitu aplikasi berbasis *web* untuk merancang ransum yang ideal untuk pakan ternak. Aplikasi berbasis *web* tersebut penulis beri nama SunRan (Susun Ransum)

1.2 Rumusan masalah

Rumusan masalah yang diangkat pada skripsi ini antara lain:

1. Bagaimana merumuskan kebutuhan untuk sistem penyusunan ransum?
2. Bagaimana merancang dan mengimplementasikan sistem penyusunan ransum?
3. Apakah hasil pengujian terhadap sistem penyusunan ransum?

1.3 Tujuan

Adapun tujuan yang ingin dicapai yaitu membuat aplikasi penyusun ransum yang memenuhi kebutuhan nutrisi harian sapi potong yang sesuai dengan fisiologis hewan ternak.

1.4 Manfaat

Pembuatan aplikasi web untuk menyusun ransum ini diharapkan dapat bermanfaat untuk memudahkan peternak maupun perusahaan ternak sapi potong dalam menyusun ransum yang memenuhi kebutuhan harian hewan ternaknya sesuai dengan fisiologisnya.

1.5 Batasan masalah

Rumusan masalah yang dicantumkan sebelumnya membuat penulis membatasi masalah sebagai berikut:

- Hewan ternak yang dijadikan objek pada skripsi ini yaitu sapi potong jantan dan betina (tidak bunting)
- Fisiologis (berat badan) awal sapi potong ditentukan antara 100 kg – 500 kg

- Jika berat badan awal sapi potong yang berada diantara 100 kg – 500 kg dan tidak terdapat pada tabel nutrisi harian, maka berat badan awal sapi diikutkan ke berat badan awal di atasnya yang berada pada tabel nutrisi harian
- Nutrien-nutrien yang diharapkan terpenuhi antara lain BK (Bahan Kering), PK (Protein Kasar), TDN (*Total Digestible Protein*), Ca (*Calcium*), dan P (*Phosphor*)
- Kandungan nutrisi masing-masing bahan penyusun ransum diasumsikan sama dengan tabel nutrisi bahan penyusun ransum
- Metode perhitungan dalam penyusunan ransum menggunakan persegi empat Pearson (*Pearson Square*)
- Pembahasan difokuskan pada analisa kebutuhan sampai pengujian aplikasi *web* penyusun ransum ideal

1.6 Sistematika pembahasan

A Skripsi ini ditulis dengan sistematika penulisan sebagai berikut.

BAB I Pendahuluan

Bab ini memuat latar belakang, rumusan masalah, batasan masalah, tujuan dan manfaat penulisan skripsi ini

BAB II Dasar Teori

Bebagai teori & fakta pendukung penulisan skripsi ini dimuat dan dijelaskan pada bab ini

BAB III Metodologi Penelitian

Bab ini membahas mengenai metodologi peneliat pengerjaan skripsi ini. Bab ini terdiri dari beberapa sub bab antara lain studi literatur, analisis kebutuhan, perancangan, implementasi, pengujian, dan pengambilan kesimpulan

BAB IV Analisis Kebutuhan

Bab ini membahas mengenai analisis kebutuhan aplikasi penyusun ransum ideal yg dijadikan sebagai topik bahasan pada skripsi ini.

BAB V Perancangan dan Implementasi

Bab ini membahas proses perancangan aplikasi penyusun ransum ideal serta implementasinya.

BAB VI Pengujian dan Analisa

Bab ini membahas mengenai pengujian aplikasi penyusun ransum ideal beserta analisa hasil pengujian.

BAB VII Penutup

Bab ini memuat kesimpulan dari pembuatan dan pengujian aplikasi penyusunan ransum ideal. Bab ini juga mencantumkan saran-saran yang bisa dijadikan acuan untuk pengembangan selanjutnya.



BAB 2 DASAR TEORI

Bab II ini akan membahas teori - teori yang menjadi dasar pengerjaan skripsi. Teori yang akan dibahas yaitu mengenai ransum, perhitungan *Pearson Square*, *framework Yii*, UML (*Unified Modelling Language*), dan pengujian *black-box*. Berikut penjelasan masing - masing.

2.1 Ransum

Kata ransum memiliki beberapa arti. Kamus Besar Bahasa Indonesia (KBBI) mengartikan ransum sebagai bagian makanan yg sudah ditentukan ukurannya untuk setiap orang atau setiap ternak; bagian bahan makanan yg diberikan kpd penduduk atau ternak dng ukuran yg sudah ditentukan; catu makanan (Pusat Bahasa Departemen Pendidikan Nasional Republik Indonesia, 2008). Arti kata ransum (*ration*) pada buku Tabel Komposisi Pakan untuk Indonesia menyebutkan *the total amount of feed (diet) allotted to one animal for a 24-hours period* (Hartadi, 2005). Pengertian tersebut jika diterjemahkan ke bahasa Indonesia menjadi jumlah makanan yang dialokasikan ke seekor hewan dalam periode 24 jam. Buku yang dikarang oleh Yenni Nur Anggraeni dan Uum Umiyasih juga menyebutkan hal yang tidak berbeda jauh untuk pengertian istilah ransum yaitu campuran dua atau lebih bahan pakan yang diberikan untuk seekor ternak selama sehari semalam (Umiyasih, 2007). Ketiga pengetahuan dari berbagai sumber yang telah disebutkan diatas menyimpulkan bahwa pengertian dari istilah ransum adalah campuran dari dua atau lebih bahan pakan yang diberikan kepada seekor hewan ternak dalam periode 24 jam.

Bahan pakan yang menjadi campuran dalam menyusun ransum terdiri dari bahan pakan berserat, bahan pakan penguat / penyerasi (konsentrat), dan bahan tambahan khusus (vitamin dan mineral). Bahan pakan berserat merupakan bahan pakan yang memiliki serat kasar (BK) tinggi, yaitu >18% dari berat keseluruhan bahan pakan. Bahan pakan-bahan pakan yang termasuk bahan pakan berserat antara lain rumput-rumputan, *legume*, dan limbah pertanian. Bahan pakan penguat (konsentrat) adalah bahan pakan yang memiliki serat kasar (BK) sebanyak <18% dari berat keseluruhan pakan. Bahan pakan jenis konsentrat ini memiliki kandungan protein sebanyak $\geq 20\%$ dari berat badan keseluruhan. Bahan pakan yang termasuk dalam bahan pakan konsentrat anatara lain dedak, bungkil, dan lain sebagainya. Berikut daftar beberapa bahan penyusun ransum beserta nutrisi masing-masing.

Tabel 2.1 Daftar beberapa bahan pakan penyusun ransum beserta nutrisinya

Nama bahan	Jenis	Nutrisi				
		BK (%)	PK (%)	TDN (%)	Ca (%)	P (%)
Jerami padi kering	Pakan berserat	86	3,7	39	0,2	0,04
Rumput lapangan	Pakan berserat	24,4	8,2	56,2	0,37	0,23
Ampas tahu	Pakan penguat	14,6	30,3	77,9	0	0
Bungkil kelapa	Pakan penguat	88,6				

Sumber: Diadaptasi dari Yulianto (2012) dan Utomo (2012)

Ransum ideal sendiri memiliki pengertian “ransum yang diberikan selama 24 jam yang mengandung semua zat nutrisi (jumlah dan macam nutrisinya) dan perbandingan yang cukup untuk memenuhi kebutuhan gizi sesuai dengan tujuan pemeliharaan ternak”. Nutrien yang terpenuhi oleh ransum yang seimbang ada berbagai macam, seperti BK (Bahan Kering), PK (Protein Kasar), TDN (*Total Digestible Protein*), Ca (*Calcium*), dan P (*Phosphor*). Hasil wawancara dengan pihak pakar juga menyebutkan mengenai ransum ideal yaitu dari nutrisi harian yang diperlukan, yang perlu diperhatikan adalah komponen BK (Bahan Kering), PK (Protein Kasar), TDN (*Total Digestible Nutrient*), kalsium dan fosfor. Komponen BK, PK, dan TDN harus diatas kebutuhan nutrisi harian, sedangkan kalsium dan fosfor boleh dibawah nutrisi harian tapi kalsium diusahakan untuk tidak kurang dari fosfor (Lampiran A).

2.2 Pearson square

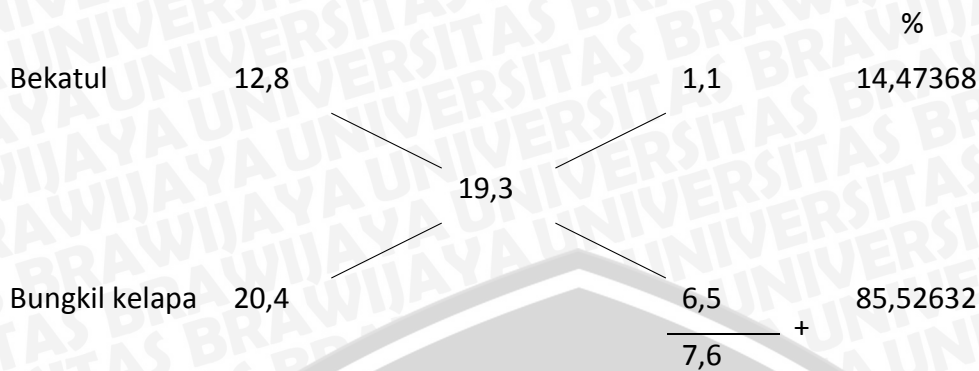
Metode *Pearson Square* adalah metode yang sudah biasa dipakai untuk menyeimbangkan ransum (Umiasih, 2007). Metode ini merupakan metode penyusunan ransum ideal yang mudah dan sederhana karena hanya berdasarkan satu kriteria (Yulianto, 2012). Kriteria yang menjadi dasar perhitungan *Pearson Square* contohnya BK (Bahan Kering), PK (Protein Kasar), atau TDN (*Total Digestible Nutrient*) saja. Contoh pengerjaan penyusunan menggunakan metode *Pearson Square* adalah sebagai berikut.

Bahan konsentrat bekatul dan bungkil kelapa diharapkan memenuhi nutrisi BK (Bahan Kering) sebanyak 1,75Kg dan PK (Protein Kasar) sebanyak 337,75gr. Bahan bekatul mengandung BK 88% dan PK 12,8%. Bahan bungkil kelapa mengandung nutrient BK sebanyak 88,6% dan PK sebanyak 20,4%.

$$\%PK \text{ diperlukan} = \frac{PK}{BK \cdot 1000} * 100 \quad (2.1)$$

$$\%PK \text{ diperlukan} = \frac{337,75}{1,75 \cdot 1000} * 100 = 19,3 \quad (2.2)$$

Persamaan di atas bertujuan untuk mengetahui jumlah pakan keseluruhan (Protein Kasar / PK) yang diberikan ke hewan ternak dalam sehari. Berdasarkan persamaan 2-2 diketahui PK yang diberikan sebanyak 19,3% dari keseluruhan ransum. Langkah berikutnya yaitu menghitung persentase untuk bekatul dan bungkil kelapa dalam penyusunan ransum.



Perhitungan di atas menunjukkan bahwa ransum terdiri dari sekitar 14,47% bekatul dan sekitar 85,56% bungkil kelapa. Setelah diketahui persentase untuk bungkil kelapa dan bekatul, dihitung berapa berat Bahan Kering (BK) dan Protein Kasar (PK) dari masing-masing bahan sebagai berikut.

$$\text{BK bekatul} = \text{BK diharapkan} * \frac{\% \text{ bekatul}}{100} \quad (2.3)$$

$$\text{BK bekatul} = 1,75 * \frac{14,47368}{100} = 0,253289 \quad (2.4)$$

$$\text{PK bekatul} = \text{PK diharapkan} * \frac{\% \text{ bekatul}}{100} \quad (2.5)$$

$$\text{PK bekatul} = 337,75 * \frac{14,47368}{100} = 1,496711 \quad (2.6)$$

$$\text{BK bungkil kelapa} = \text{BK diharapkan} * \frac{\% \text{ bungkil kelapa}}{100} \quad (2.7)$$

$$\text{BK bungkil kelapa} = 1,75 * \frac{85,52632}{100} = 1,496711 \quad (2.8)$$

$$\text{PK bungkil kelapa} = \text{PK diharapkan} * \frac{\% \text{ bungkil kelapa}}{100} \quad (2.9)$$

$$\text{PK bungkil kelapa} = 337,75 * \frac{85,52632}{100} = 305,3289 \quad (2.10)$$

Variabel BK diharapkan merupakan berat Bahan Kering (BK) yang akan dipenuhi oleh ransum. Variabel PK diharapkan merupakan berat Protein Kasar (PK) yang akan dipenuhi oleh ransum. Setelah berat PK dan BK diperoleh untuk masing-masing bahan penyusun ransum, dapat dilakukan rekapitulasi untuk mengecek apakah nutrisi hewan ternak terpenuhi atau tidak.

Tabel 2.2 Hasil perhitungan contoh penggunaan metode *Pearson Square*

Nama Bahan	BK(Kg)	PK (gr)
Bekatul	0,253289	32,42105
Bungkil kelapa	1,496711	305,3289
Total	1,75	337,75
Kebutuhan	1,75	337,75

Tabel 2.2 di atas menunjukkan total nutrisi (BK dan PK) dari keseluruhan ransum sama dengan kebutuhan nutrisi hewan ternak. Kesamaan nilai antara total dan kebutuhan diatas menunjukkan bahwa penyusunan ransum telah memenuhi kebutuhan nutrisi harian hewan ternak. Kebutuhan harian hewan ternak tersebut berdasarkan kondisi fisiologin hewan ternak sendiri.

Contoh di atas menunjukkan metode *Pearson Square* menunjukkan kemampuan dalam menyusun dua bahan penyusun ransum, namun berdasarkan wawancara dengan pihak pakar menjelaskan tentang penyusunan ransum menggunakan lebih dari dua jenis bahan, yaitu pertama, membagi rasio antara seluruh bahan hijauan dan bahan konsentrat. Jika bahan hijauan sebanyak 1 jenis maka langsung mengikuti rasio bahan hijauan. Jika bahan hijauan sebanyak 2 jenis maka BK untuk masing-masing hijauan sebanyak separuh dari BK yang diperlukan. Jika bahan hijauan sebanyak 3 jenis maka BK setiap hijauan sebanyak sepertiga dari BK yang diperlukan. Jika bahan konsentrat sebanyak 1 jenis maka langsung mengikuti rasio bahan konsentrat. Jika bahan konsentrat sebanyak 2 jenis maka digunakan metode *Pearson Square*. Jika bahan konsentrat sebanyak 3 jenis maka BK untuk satu bahan konsentrat sebanyak sepertiga dari BK yang diperlukan, sedangkan 2 bahan konsentrat lainnya menggunakan metode *Pearson Square* (Lampiran A).

2.3 Framework Yii

Framework Yii adalah *framework* (kerangka kerja) PHP berbasis komponen untuk pengembangan aplikasi *web* berskala besar (Badiyanto, 2013). *Framework* Yii ini berasal dari singkatan “Yes It Is!”. *Framework* PHP ini dibuat oleh Qiang Xue pada tanggal 1 Januari 2008. Versi stabil dari *framework* Yii yang paling baru adalah versi 1.1. Saat ini dikembangkan pula versi 2.0 tetapi masih dalam tahap beta. *Framework* Yii ini memiliki performa yang tinggi dibandingkan *framework* PHP lainnya. Tabel berikut menunjukkan perbandingan performa Yii dengan *framework* PHP lainnya.

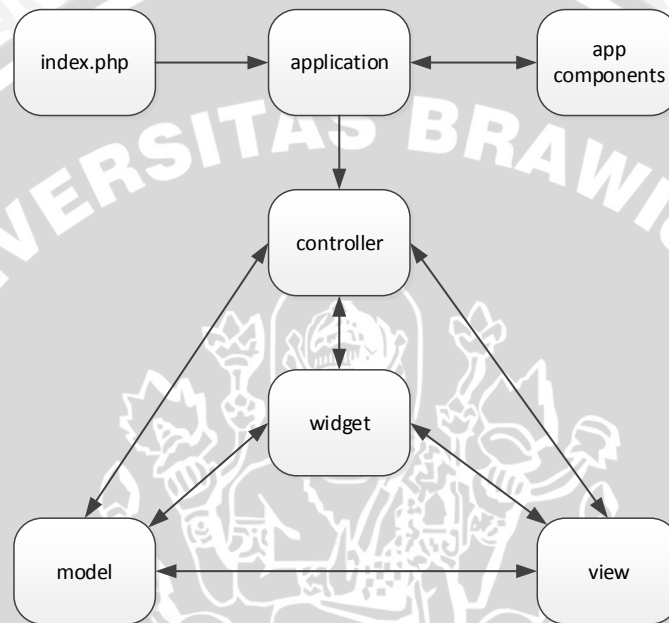
Tabel 2.3 Perbandingan performa beberapa *framework* PHP

Nama framework	RPS dengan APC	RPS tanpa APC
Yii 1.0.2	673	93
CodeIgniter 1.7.0	206	79
Zend 1.7.3	83	38
CakePHP 1.2.1	79	29
Prado 3.1.3	75	24
Symfony 1.2.2	50	18

Sumber: Diadaptasi dari Sharive (2013)

Tabel di atas menunjukkan bahwa Yii melampaui performa *framework* lainnya berdasarkan jumlah RPS (Request Per Second) atau Permintaan Per Detik. Keunggulan ini semakin meningkat saat APC (Alternative PHP Cache) dipergunakan.

Arsitektur yang dipakai oleh Yii adalah MVC (*Model-View-Controller*). *Framework* ini dibangun dengan menggunakan OOP (*Object Oriented Programming*) murni sehingga memudahkan penggunaannya saat penggunaan Yii dalam pembuatan aplikasi yang berbasis pemrograman berbasis objek. Arsitektur dari Yii jika digambarkan dalam diagram menjadi sebagai berikut.



Gambar 2.1 Struktur statis aplikasi Yii

Sumber: Sharive (2013)



2.4 Unified Modeling Language (UML)

UML merupakan singkatan dari *Unified Modeling Language* yang jika diterjemahkan ke Bahasa Indonesia menjadi bahasa pemodelan standar. Terjemahan dari singkatan tersebut menunjukkan bahwa dalam pemodelan menggunakan Bahasa yang terstandar. Widodo (2011) juga menulis dalam bukunya yang berjudul Menggunakan UML menyebutkan bahwa ketika kita membuat model menggunakan konsep *UML* ada aturan-aturan yang harus diikuti. Bagaimana element pada model-model yang kita but berhubungan satu dengan lainnya harus mengikuti standar yang ada (Widodo, 2011). Model yang mendapatkan standar dari UML antara lain diagram kelas, paket, *use case*, interaksi (dan *sequence*), komunikasi, *statechart*, aktivitas, komponen, dan *deployment*.

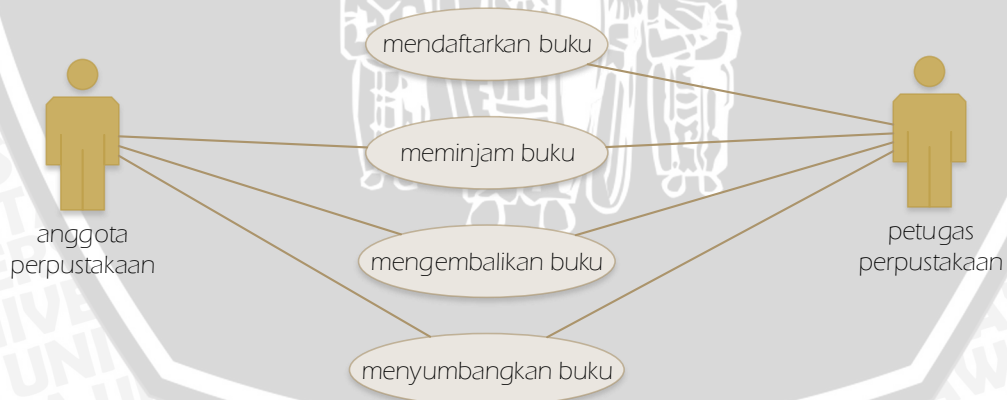
Diagram *use case* merupakan salah satu diagram dari UML yang menggambarkan *external view* dari sistem yang akan kita buat modelnya

(Widodo-2011). Diagram *use case* ini terdiri dari beberapa komponen yang akan dijelaskan pada Tabel 2.4 berikut.

Tabel 2.4 Deskripsi komponen diagram *use case*

No	Gambar	Nama	Deskripsi
1		Aktor	Pihak yang berperan dalam sistem. Orang / sistem yang menjalankan suatu sistem juga termasuk aktor.
2		<i>Use case</i>	Penggambaran dari fungsi yang berada pada sistem dan bisa berupa kelas, komponen atau <i>event</i> .
3		Relasi generalisasi	Relasi ini bisa untuk melakukan proses generalisasi dari komponen <i>use case</i> maupun aktor. Relasi ini menyederhanakan penggambaran aktor maupun <i>use case</i> yang memiliki kesamaan relasi.
4		Relasi ekstensi	Relasi ini menggambarkan ekstraksi suatu <i>use case</i> dari <i>use case</i> yang lebih kompleks.
5		Relasi inklusi	Relasi ini menggambarkan <i>use case</i> yang bisa dijalankan jika sudah menjalankan <i>use case</i> lainnya.

Gambar 2.2 berikut menggambarkan contoh dari penggunaan diagram *use case*.




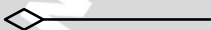
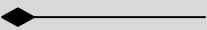



Gambar 2.2 Contoh diagram *use case*

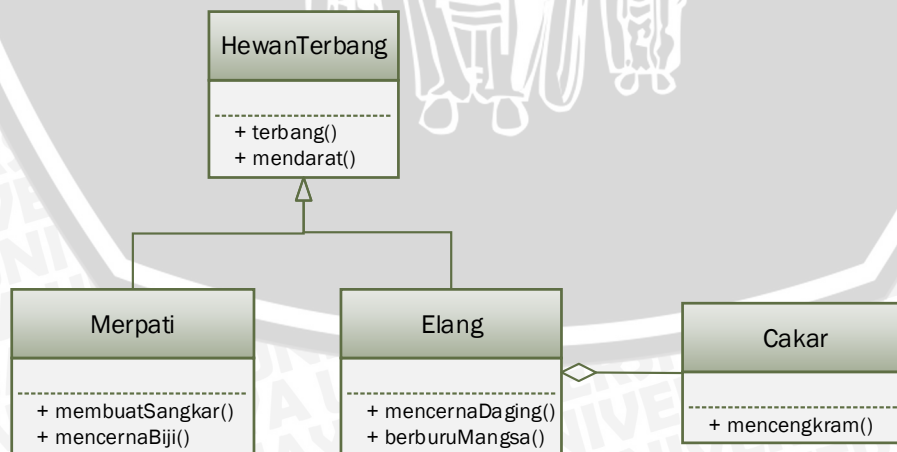
Diagram kelas adalah inti dari proses pemodelan objek (Widodo, 2011). Alasannya adalah diagram kelas terkait dengan diagram-diagram UML lainnya. Diagram kelas ini menggambarkan kelas-kelas beserta relasinya. Diagram kelas ini terdapat atribut, operasi, dan relasi. Atribut merupakan rincian dari suatu kelas yang bisa berupa variabel. Operasi adalah bagian di dalam kelas yang

biasanya berupa fungsi / *function*. Relasi merupakan hubungan antar kelas yang ada pada diagram kelas. Penggambaran diagram kelas dibuat menggunakan simbol-simbol yang dijelaskan pada Tabel 2.5 berikut.

Tabel 2.5 Deskripsi simbol-simbol diagram kelas

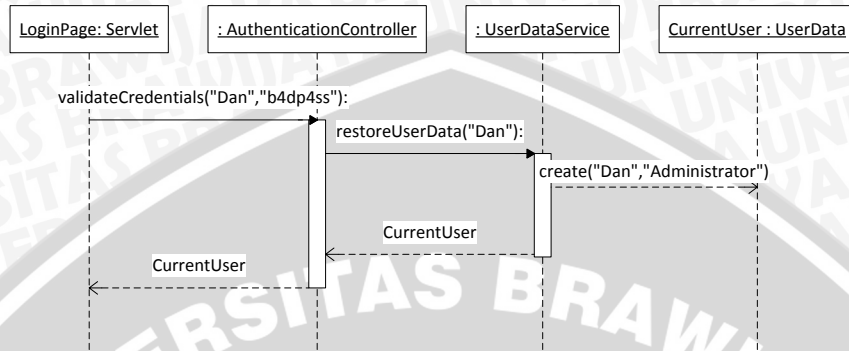
No	Simbol	Nama	Deskripsi
1		Kelas	Cetak biru dari objek yang memiliki atribut, metode, dan relasi dengan kelas lainnya
2		Relasi ketergantungan	Relasi ini menjelaskan bahwa suatu kelas tergantung pada kelas lainnya.
3		Relasi asosiasi	Relasi ini menjelaskan bahwa suatu kelas berhubungan dengan kelas lainnya namun tidak mempengaruhi / mengganggu kelas lainnya yang direlasikan.
4		Relasi agregasi	Relasi ini menggambarkan suatu kelas memiliki kelas lainnya yang direlasikan.
5		Relasi komposisi	Relasi ini menggambarkan suatu kelas merupakan bagian dari kelas yang direlasikan. Relasi komposisi ini tidak mengizinkan kelas yang dimiliki terasosiasi dengan kelas pemilik lainnya sebelum relasi komposisi yang dimiliki terputus.
6		Relasi generalisasi	Relasi ini menggambarkan suatu kelas adalah kelas yang lebih khusus / spesifik dari kelas lainnya yang lebih umum

Gambar 2.3 berikut menggambarkan contoh dari penggunaan diagram kelas.



Gambar 2.3 Contoh diagram kelas

Diagram di dalam UML selain diagram *use case* dan diagram kelas terdapat pula diagram urutan (*sequence diagram*). Diagram ini salah satu diagram yang dipakai untuk memodelkan skenario interaksi (Widodo, 2011). Diagram urutan ini menunjukkan bagaimana perilaku antar entitas dalam bekerja sama saat menjalankan sistem. Gambar 2.3 menampilkan contoh dari *sequence diagram*.



Gambar 2.4 Contoh *sequence diagram*

Sumber: Widodo (2011)

Tabel 2.6 berikut menjelaskan simbol-simbol yang ada di diagram urutan (*sequence diagram*).

Tabel 2.6 Deskripsi simbol-simbol *sequence diagram*

No	Simbol	Nama	Deskripsi
1		Objek	Entitas yang terlibat dalam interaksi. Objek ini memiliki garis hidup (<i>lifeline</i>)
2		Pesan (<i>message</i>)	Jenis komunikasi antar objek berupa pemanggilan metode, pembentukan instance, dan lin-lain
3		Pesan (<i>message</i>) pengembalian	Jenis komunikasi antar objek yang arahnya kembali (mengarah ke kiri)
4		<i>Activation</i>	Menunjukkan objek sedang focus ke sistem / mengerjakan sesuatu

2.5 Pengujian *black-box*

Pengujian *black-box* disebutkan oleh Roger S. Pressman bahwa *black-box testing, also called behavioral testing, focuses on the functional requirements of the software* (Pressman, 2010). Kutipan tersebut jika diterjemahkan ke Bahasa Indonesia menjadi pengujian *black-box*, disebut juga dengan pengujian perilaku,

berfokus pada kebutuhan fungsional dari perangkat lunak. Pengujian *black-box* ini membandingkan perilaku dari sistem saat berjalan dengan kebutuhan yang diharapkan untuk dipenuhi. Jika sistem menunjukkan perilaku yang sesuai dengan kebutuhan yang diharapkan maka sistem tersebut valid.

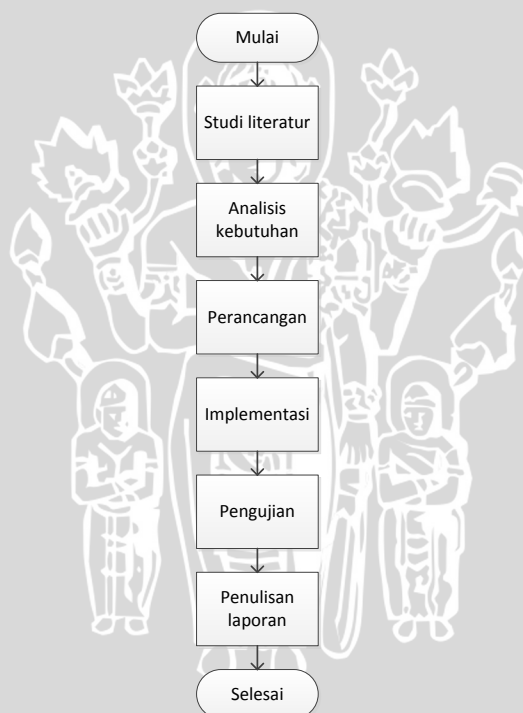
Pressman (2011) juga menyebutkan pada bukunya bahwa pengujian *black-box* ini cocok untuk mencari kesalahan / *error* berupa kesalahan atau hilangnya fungsi, kesalahan tampilan / antarmuka, kesalahan dalam struktur data maupun akses terhadap basis data eksternal, kesalahan perilaku maupun performa, dan kesalahan dalam inisialisasi serta penghentian.



BAB 3 METODOLOGI PENELITIAN

Metode-metode yang di lakukan dalam pengerjaan skripsi ini ada berbagai macam. Pengambilan dasar teori menggunakan metode studi literatur dan wawancara kepada pihak yang ahli (*expert*) dibidang peternakan. Metode pengambilan data pada skripsi ini baik berupa data bahan pakan maupun data nutrisi sapi diperoleh dengan cara sekunder. Data tersebut diperoleh melalui studi literatur. Pengujian yang akan dilakukan pada skripsi ini menggunakan metode *black box* yaitu, pengujian validasi umum, pengujian validasi implementasi *Pearson Square*, dan pengujian kompatibilitas *web browser*.

Penjelasan mengenai langkah - langkah yang akan dilakukan selama pengerjaan skripsi ini akan di jelaskan pada bab ini. Langkah - langkah yang akan dilakukan dimulai dari studi literatur, analisis kebutuhan, perancangan, implementasi, pengujian, dan penulisan laporan. Seluruh langkah – langkah tersebut jika digambarkan dalam diagram alir menjadi sebagai berikut.



Gambar 3.1 Diagram alir penelitian

3.1 Studi literatur

Tahapan dalam metodologi penelitian ini adalah tahapan untuk memahami teori-teori pendukung yang menjadi dasar pengerjaan skripsi ini. Teori-teori pendukung yang dipakai antara lain:

- Ransum
- *Pearson square*
- *Framework Yii*

- Unified Modelling Language (UML)
- Pengujian *Black-Box*

Teori-teori tersebut didapat dari berbagai sumber, baik dari buku, jurnal, maupun wawancara langsung dengan pihak pakar. Pihak pakar yang dimaksud adalah pihak yang ahli dibidang peternakan khususnya ruminansia (hewan pemamah biak) seperti sapi, kambing, kerbau, dan lain-lain.

3.2 Analisa kebutuhan

Tahapan berikutnya setelah studi literatur adalah analisis kebutuhan. Tahapan analisa kebutuhan ini menentukan kebutuhan-kebutuhan yang diperlukan dalam pengerjaan skripsi ini (khususnya dalam pembuatan aplikasi penyusunan ransum ideal). Penentuan kebutuhan ini yang akan membantu aplikasi dalam pemenuhan kebutuhan pengguna saat uji coba maupun saat aplikasi telah selesai. Kebutuhan-kebutuhan yang ditentukan akan dikategorikan menjadi kebutuhan fungsional dan kebutuhan non-fungsional. Kebutuhan-kebutuhan yang ditentukan berasal dari wawancara langsung dengan pihak pakar. Kebutuhan-kebutuhan juga dimodelkan dengan *use cases*.

3.3 Perancangan

Tahapan yang dilakukan setelah analisis kebutuhan adalah perancangan. Tahapan ini melakukan perancangan terhadap sistem agar memenuhi kebutuhan-kebutuhan yang telah ditentukan sebelumnya. Perancangan ini di dimodelkan kedalam bentuk beberapa diagram yang menggunakan standar Unified Modelling Language (UML), antara lain:

- Diagram kelas
- Diagram interaksi

Selain pemodelan beberapa diagram diatas, pada tahapan perancangan ini ditentukan pula beberapa hal, seperti penentuan kerangka kerja, perancangan arsitektur sistem, dan perancangan antarmuka (*interface*) aplikasi. Pemodelan data juga dilakukan pada tahapan perancangan ini.

3.4 Implementasi

Implementasi merupakan tahapan selanjutnya setelah melakukan tahapan perancangan. Tahapan implementasi ini dilakukan berdasarkan rancangan yang telah dibuat sebelumnya. Oleh karena itu, tahapan implementasi ini meliputi beberapa hal, antara lain:

- Pembuatan aplikasi penyusunan ransum
- Pembuatan basis data untuk menampung data yang diperlukan oleh aplikasi
- Memasukkan data yang diperlukan oleh aplikasi ke basis data
- Pembuatan tampilan (*interface*) aplikasi

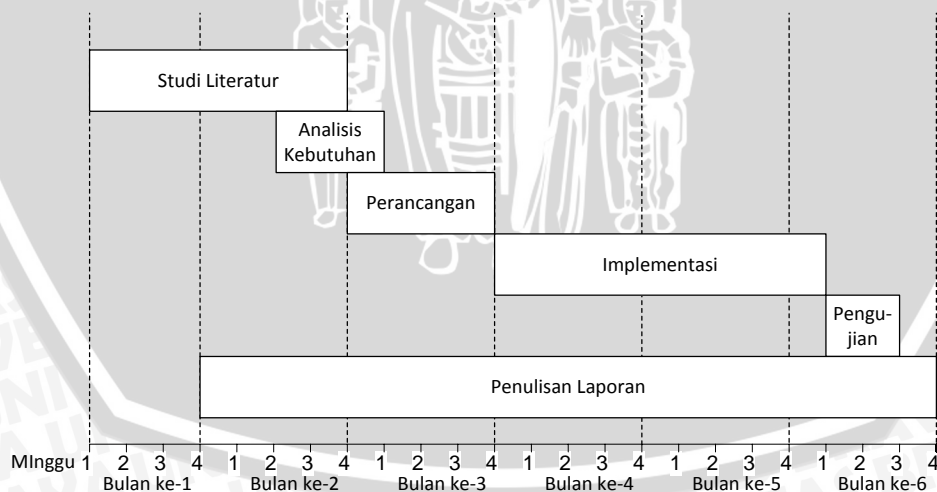
3.5 Pengujian

Tahapan pengujian ini dilakukan setelah tahapan implementasi selesai dilakukan. Tahapan ini dilakukan untuk mengecek apakah aplikasi telah memenuhi kebutuhan-kebutuhan yang telah ditentukan sebelumnya.

Metode pengujian yang dilakukan adalah *black box*. Pengujian yang dilakukan antara lain pengujian validasi umum, validasi implementasi *Pearson Square*, dan komabilitas terhadap aplikasi *web browser*. Pengujian validasi umum nantinya membandingkan perilaku yang diharapkan terhadap sistem dengan perilaku yang dilakukan oleh sistem. Kesesuaian perilaku dari sistem dengan kebutuhan yang diharapkan menentukan validasi dari sistem sendiri. Pengujian implementasi *pearson Square* membandingkan hasil penyusunan ransum oleh aplikasi SunRan dengan hasil penyusunan ransum oleh Microsoft Excel. Kesesuaian hasil pada tingkat ketelitian 10^{-2} menentukan validitas hasil penyusunan ransum. Pengujian implementasi *Pearson Square* ini juga dicari persentasi kesalahan hasil penyusunan ransum oleh aplikasi SunRan dengan hasil penyusunan ransum oleh Excel pada tingkat ketelitian 10^{-7} . Pengujian komabilitas terhadap aplikasi *web browser* dilakukan ke beberapa aplikasi *web browser*. Setiap pengujian ini membandingkan antarmuka yang ditampilkan kepada pengguna dengan rancangan antarmuka.

3.6 Penulisan laporan

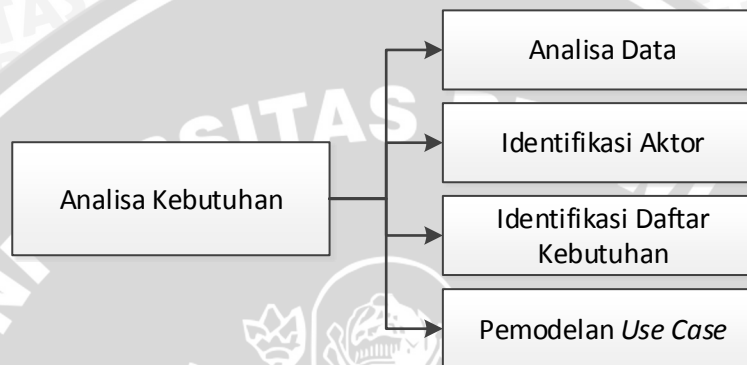
Penulisan laporan dilakukan saat tahapan-tahapan sebelumnya seperti analisa, perancangan, dan implementasi selesai dilakukan. Tahapan ini menulis mengenai tahapan-tahapan sebelumnya dan penjelasannya. Tahapan penulisan dan tahapan-tahapan sebelumnya dikerjakan dengan *timeline* sebagai berikut.



Gambar 3.2 *Timeline* pengerjaan skripsi

BAB 4 ANALISA KEBUTUHAN

Bab keempat ini akan membahas tentang analisa kebutuhan. Proses analisa kebutuhan ini merupakan proses awal yg akan mengidentifikasi kebutuhan yg diperlukan oleh aplikasi SunRan. Kebutuhan yang dimaksud seperti kebutuhan data, fungsi-fungsi aplikasi, dan lain-lain. Proses-proses di dalam pengerjaan analisa kebutuhan sistem antara lain analisa data, identifikasi aktor, identifikasi daftar kebutuhan, dan pemodelan *use case*. Proses-proses yang telah disebutkan diatas di ilustrasikan dalam Gambar 4.1 berikut.



Gambar 4.1 Diagram analisa kebutuhan

Bagian-bagian dari Gambar 4.1 di atas akan dibahas dalam sub-sub bab berikut.

4.1 Analisa data

Analisa data yang dilakukan ini merupakan tahapan dalam analisa kebutuhan yang mengidentifikasi struktur data yang diperlukan oleh sistem. Struktur data yang diperlukan oleh aplikasi SunRan ini antara lain data bahan penyusun ransum dan data kebutuhan nutrisi minimum dari sapi. Data dari bahan penyusun ransum terdiri dari nama dan nutrisinya. Dan data kebutuhan nutrisi minimum sapi yang dipakai di tentukan berdasarkan berat badan, jenis kelamin, dan penambahan berat badan yang diinginkan.

4.2 Identifikasi aktor

Tahap identifikasi aktor ini menentukan pihak-pihak yang terhubung langsung dengan aplikasi SunRan. Pihak-pihak ini yang menjadi aktor dalam pemodelan *use case* pada tahap berikutnya. Aktor-aktor yang telah diidentifikasi dapat dilihat pada tabel 4.3 berikut.

Tabel 4.1 Daftar aktor

Aktor	Deskripsi Aktor
Peternak praktis	Aktor ini adalah aktor yang bisa menjalankan beberapa fungsi di sistem SunRan, yaitu melihat tabel nutrisi, melihat tabel komposisi ransum, dan menyusun ransum.
Analisis	Aktor ini adalah aktor yang bisa melakukan beberapa fungsi yang dimiliki oleh aktor peternak seperti mendaftarkan diri, melihat tabel nutrisi, melihat tabel komposisi ransum, dan menyusun ransum. aktor ini juga bisa melakukan fungsi mendaftarkan diri dan <i>login</i> pada sistem. Aktor ini mampu mengatur hasil penyusunan ransum, membandingkan ransum, dan <i>logout</i> setelah masuk / <i>login</i> ke sistem.

4.3 Penyusunan daftar kebutuhan

Daftar kebutuhan yang dijelaskan pada sub-sub bab berikut terdiri dari daftar kebutuhan fungsional & non-fungsional. Daftar kebutuhan fungsional mengidentifikasi kebutuhan / *requirement* yang harus dipenuhi saat aplikasi telah selesai dibuat. Setiap kebutuhan fungsional disebut sebagai *Software Requirement Specification (SRS)* dan mendapatkan kode angka tertentu sebagai identifikasinya. Daftar kebutuhan fungsional bisa dilihat pada Tabel 4.2 berikut.

Tabel 4.2 Daftar kebutuhan fungsional

Kode Kebutuhan	Kebutuhan	Aktor	Use Case
SRS_001_1	Sistem mampu melakukan proses pendaftaran oleh aktor	Analisis	Mendaftarkan diri
SRS_001_2	Sistem mampu mengubah status pengguna menjadi masuk ke sistem	Analisis	<i>Login</i>
SRS_002_1	Sistem mampu menampilkan nutrisi harian	Peternak praktis dan analisis	Melihat tabel nutrisi
SRS_002_2	Sistem mampu menampilkan bahan-bahan komposisi penyusunan ransum	Peternak praktis dan analisis	Melihat tabel komposisi ransum
SRS_002_3	Sistem mampu menyusun ransum ideal	Peternak praktis dan analisis	Menyusun ransum
SRS_001_3	Sistem mampu melihat hasil penyusunan ransum yang tersimpan di sistem	Analisis	Mengatur hasil penyusunan ransum
SRS_001_4	Sistem mampu mengubah nama hasil penyusunan ransum yang tersimpan di sistem	Analisis	
SRS_001_5	Sistem mampu menghapus hasil penyusunan ransum yang tersimpan di sistem	Analisis	
SRS_001_6	Sistem mampu menampilkan perbandingan dari dua hasil penyusunan ransum untuk sapi potong dengan berat badan, jenis kelamin, dan pertambahan berat badan per hari yang sama	Analisis	Membandingkan ransum



Tabel 4.2 Daftar kebutuhan fungsional (lanjutan)

Kode Kebutuhan	Kebutuhan	Aktor	Use Case
SRS_001_7	Sistem mampu melakukan <i>logout</i> terhadap user yang sedang <i>login</i>	Analisis	<i>Logout</i>

Daftar kebutuhan fungsional pada tabel di atas didapat dari hasil wawancara langsung dengan pihak pakar. Hasil wawancara bisa dilihat pada Lampiran A.

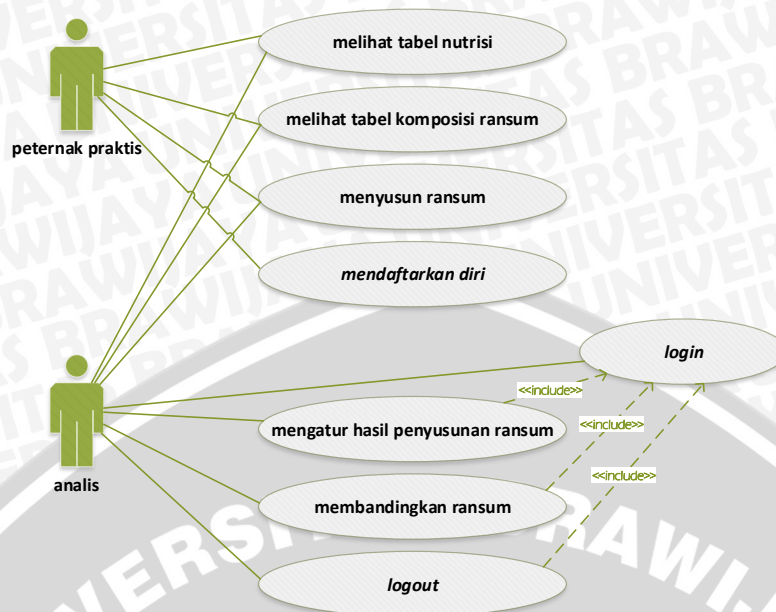
Selain kebutuhan fungsional yang harus dipenuhi oleh aplikasi SunRan, ada pula kebutuhan non-fungsional yang juga harus dipenuhi. Kebutuhan non-fungsional yang harus dipenuhi antara lain presisi (*precision*) dan portabilitas. Deskripsi dari kebutuhan non-fungsional tersebut terdapat di tabel 4.3 berikut.

Tabel 4.3 Daftar kebutuhan non-fungsional

Parameter	Deskripsi Kebutuhan
Presisi (<i>precision</i>)	Hasil penyusunan ransum oleh sistem memiliki kesamaan nutrisi hingga dua angka dibelakang koma (dalam satuan yang sama) dengan nutrisi hasil penyusunan ransum yang dilakukan oleh Microsoft Excel (secara <i>Pearson Square</i>).
Portabilitas	Sistem mampu menampilkan antarmuka sesuai dengan rancangan antarmuka pada beberapa aplikasi <i>web browser</i> , yaitu Chrome, Firefox, dan Opera.

4.4 Pemodelan *use case*

Aktor-aktor yang telah teridentifikasi pada sub bab sebelumnya akan dipakai dalam pemodelan *use case*. Pemodelan *use case* ini didasari atas standar Unified Modelling Language (UML). Masing-masing aktor akan terhubung dengan fungsi-fungsi yang bisa dilakukan, contohnya aktor pengguna terdaftar terhubung dengan *use case* menyusun ransum. Berikut diagram *use case* berdasarkan penjelasan Tabel 4.2 di atas.



Gambar 4.2 Use case Aplikasi Penyusunan Ransum (SunRan)

Penjelasan singkat dari masing-masing *usecase* pada Gambar 4.2 diatas akan dijabarkan pada tabel-tabel berikut.

Tabel 4.4 Deskripsi singkat use case mendaftarkan diri

Deskripsi Singkat Use Case	
Nama	Mendaftarkan diri
Tujuan	Untuk mendaftarkan data pengguna yang belum terdaftar agar bisa <i>login</i> .
Deskripsi	<i>Use case</i> ini memberikan fasilitas untuk pengguna yang belum terdaftar agar bisa mendaftarkan diri sehingga nantinya bisa masuk ke sistem.
Aktor	Analisis.

Penjelasan lebih detail dari *use case* mendaftarkan diri di atas bisa dilihat pada Lampiran B.4.

Tabel 4.5 Deskripsi singkat use case login

Deskripsi Singkat Use Case	
Nama	<i>Login</i>
Tujuan	Mengubah kondisi pengguna yang tidak masuk ke sistem (<i>guest user</i>) menjadi pengguna yang masuk ke sistem (<i>logged in user</i>).
Deskripsi	<i>Use case</i> ini memberikan fasilitas kepada pengguna yang tidak masuk ke sistem (<i>guest user</i>) menjadi pengguna yang masuk ke sistem (<i>logged in user</i>) sehingga bisa mengakses fitur-fitur pada sistem yang memerlukan persyaratan kondisi pengguna sebagai <i>logged in user</i> .
Aktor	Analisis.

Penjelasan lebih detail dari *use case login* di atas bisa dilihat pada Lampiran B.5.

Tabel 4.6 Deskripsi singkat *use case* melihat tabel nutrisi

Deskripsi Singkat <i>Use Case</i>	
Nama	Melihat tabel nutrisi
Tujuan	Menampilkan daftar nutrisi harian sapi potong dalam bentuk tabel.
Deskripsi	<i>Use case</i> ini menampilkan daftar nutrisi harian sapi potong untuk proses penggemukan.
Aktor	Peternak dan analis.

Penjelasan lebih detail dari *use case* melihat tabel nutrisi di atas bisa dilihat pada Lampiran B.1.

Tabel 4.7 Deskripsi singkat *use case* melihat tabel komposisi ransum

Deskripsi Singkat <i>Use Case</i>	
Nama	Melihat tabel komposisi ransum
Tujuan	Menampilkan daftar bahan penyusun ransum.
Deskripsi	<i>Use case</i> ini menampilkan daftar bahan penyusun ransum baik bahan hijauan dan konsentrat dalam bentuk tabel beserta nutrisinya.
Aktor	Peternak dan analis.

Penjelasan lebih detail dari *use case* melihat tabel komposisi ransum di atas bisa dilihat pada Lampiran B.2.

Tabel 4.8 Deskripsi singkat *use case* menyusun ransum

Deskripsi Singkat <i>Use Case</i>	
Nama	Menyusun ransum
Tujuan	Menyusun ransum agar mendapatkan ransum ideal.
Deskripsi	<i>Use case</i> ini membantu pengguna dalam menyusun ransum dengan bahan-bahan, fisiologi sapi potong (berat badan & jenis kelamin), dan penambahan per hari yang sudah ditentukan sebelumnya agar didapatkan ransum ideal.
Aktor	Peternak dan analis.

Penjelasan lebih detail dari *use case* bisa dilihat pada Lampiran B.3.

Tabel 4.9 Deskripsi singkat *use case* mengatur hasil penyusunan ransum

Deskripsi Singkat <i>Use Case</i>	
Nama	Mengatur hasil penyusunan ransum
Tujuan	Melihat secara detail, menghapus, dan mengubah (<i>update</i>) hasil penyusunan ransum.
Deskripsi	<i>Use case</i> ini membuat pengguna bisa untuk melihat secara detail, menghapus, dan mengubah (<i>update</i>) hasil penyusunan ransum yang sudah tersimpan sebelumnya.
Aktor	Analis.

Penjelasan lebih detail dari *use case* menyusun ransum di atas bisa dilihat pada Lampiran B.6.

Tabel 4.10 Deskripsi singkat *use case* membandingkan ransum

Deskripsi Singkat <i>Use Case</i>	
Nama	Membandingkan ransum
Tujuan	Membandingkan dua hasil penyusunan ransum.
Deskripsi	<i>Use case</i> ini membandingkan dua hasil penyusunan ransum yang sudah tersimpan sebelumnya. Hasil perbandingan akan ditampilkannya secara detail.
Aktor	Analisis.

Penjelasan lebih detail dari *use case* membandingkan ransum di atas bisa dilihat pada Lampiran B.7.

Tabel 4.11 Deskripsi singkat *use case* logout

Deskripsi Singkat <i>Use Case</i>	
Nama	<i>Logout</i>
Tujuan	Keluar dari sistem.
Deskripsi	<i>Use case</i> ini bertugas untuk mengubah kondisi / status pengguna yang awalnya sebagai pengguna yang masuk ke sistem (<i>logged in user</i>) menjadi pengguna yang tidak masuk ke sistem (<i>guest user</i>). Pengguna yang selesai memilih <i>use case</i> ini tidak bisa menjalankan <i>use case</i> yang memerlukan persyaratan pengguna yang masuk ke sistem (<i>logged in user</i>).
Aktor	Analisis.

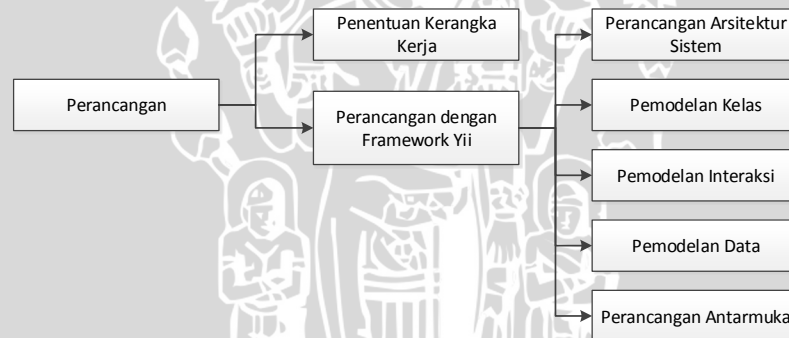
Penjelasan lebih detail dari *use case* logout di atas bisa dilihat pada Lampiran B.8.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Proses perancangan adalah tahapan awal yang akan dibahas di bab kelima ini. Proses perancangan ini terdiri dari penentuan kerangka kerja, perancangan arsitektur sistem, pemodelan kelas, pemodelan data, dan perancangan antarmuka. Bahasan berikutnya setelah perancangan yaitu implementasi. Pembahasan implementasi terdiri dari spesifikasi perangkat pengembang sistem, implementasi fitur sistem, implementasi kalkulasi *Pearson Square*, implementasi basis data, dan implementasi antarmuka.

5.1 Perancangan

Proses perancangan ini akan menyesuaikan bagian-bagian (perangkat lunak) di dalam kerangka kerja sudah ditentukan. Salah satu contoh penyesuaian yang dilakukan yaitu melakukan pengaturan koneksi basis data didalam *framework* Yii agar bisa terhubung dengan MySQL. Proses perancangan ini terdiri dari penentuan kerangka kerja dan perancangan dengan *framework* Yii. Proses perancangan dengan *framework* Yii terdiri dari beberapa tahapan, antara lain perancangan arsitektur sistem, pemodelan kelas, pemodelan interaksi, pemodelan data, dan perancangan antarmuka. Tahapan-tahapan dari perancangan dapat digambarkan sebagai berikut.



Gambar 5.1 Diagram perancangan

5.1.1 Penentuan kerangka kerja

Tahap penentuan kerangka kerja ini adalah tahapan yang menentukan kerangka kerja / *framework* yang akan di pakai selama proses perancangan dan implementasi. Kerangka kerja yang akan dipakai dalam pengembangan aplikasi SunRan adalah *framework* Yii. Pemilihan *framework* Yii disebabkan oleh performanya yang lebih baik daripada *framework* PHP lainnya. *Framework* Yii ini akan menjadi komponen utama karena *framework* ini berkerja layaknya pondasi dan gedung bagi aplikasi sunran. Fungsi-fungsi dan *method-method* yang sudah terdapat di dalam *framework* akan dipakai ulang selama pengembangan aplikasi SunRan.

5.1.2 Perancangan dengan *framework* Yii

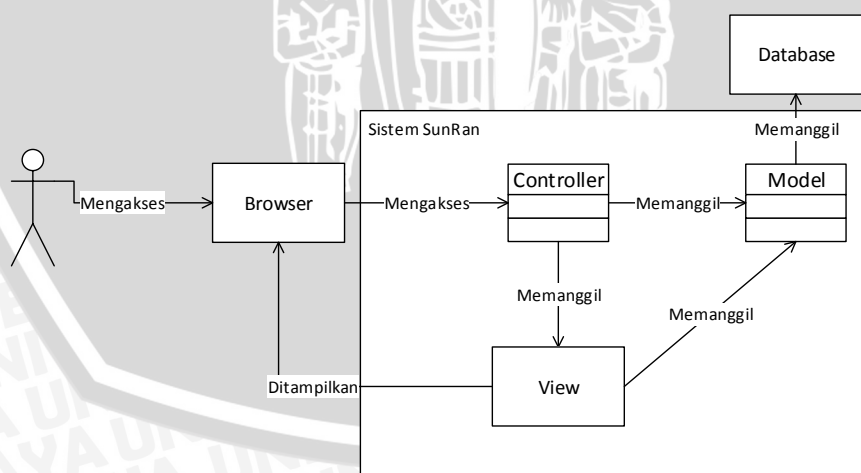
Perancangan dengan *framework* Yii ini adalah tahapan yang menentukan bagian-bagian di dalam kerangka kerja yang perlu di sesuaikan untuk proses implementasi serta perangkat lunak yang ditambahkan ke kerangka kerja. Salah satu contoh penyesuaian yaitu melakukan pengaturan koneksi basis data didalam *framework* Yii agar bisa terhubung dengan MySQL.

Perangkat lunak yang ditambahkan ke kerangka kerja selama masa pengembangan aplikasi SunRan adalah highchart dan htmltableui. Highchart berupa *extention* bagi *framework* Yii. Highchart mampu menampilkan data ke berbagai jenis grafik / bagan. Htmltableui juga berupa *extention* bagi *framework* Yii. Htmltableui mampu menampilkan data ke tabel secara rapi. Penggunaan highchart dan htmltableui akan memudahkan pengguna dalam memahami hasil dari penyusunan ransum.

Perangkat lunak berikutnya yang diperlukan dalam pengembangan aplikasi sunran adalah MySQL. MySQL ini adalah *Relational Database Management System* (RDBMS) yang bersifat *open-source*. Perangkat lunak ini yang melakukan manajemen penyimpanan data di dalam basis data dan juga memberikan data yang diminta oleh aplikasi SunRan.

5.1.2.1 Perancangan *arsitektur sistem*

Perancangan *arsitektur sistem* adalah perancangan perangkat lunak yang akan dipakai selama proses implementasi. *Arsitektur sistem* dari aplikasi SunRan ini menggunakan struktur MVC (*Model View Controller*). *Arsitektur* ini disesuaikan dengan *arsitektur* asli dari *framework* Yii yang juga menggunakan struktur MVC. Penggambaran *arsitektur* aplikasi SunRan dapat dilihat pada gambar berikut.



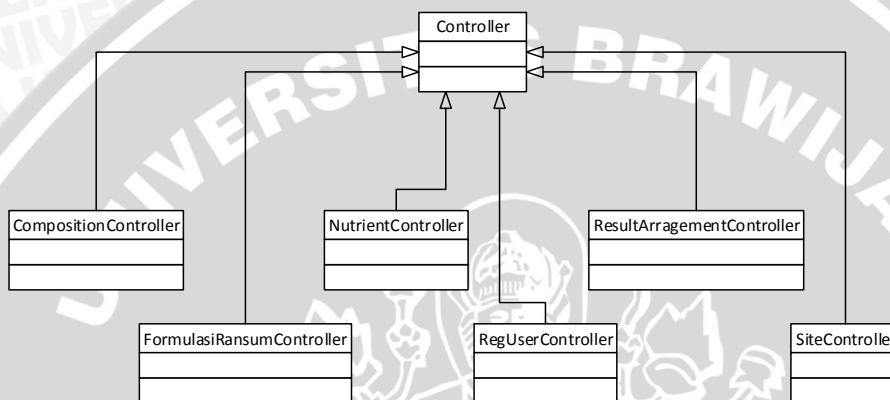
Gambar 5.2 Arsitektur aplikasi SunRan

Struktur MVC pada aplikasi SunRan sesuai dengan gambar diatas diawali dengan pengguna / *user* mengakses SunRan melalui aplikasi *web browser*. Pengguna saat mengakses SunRan melalui *browser* akan mengakses *controller*. *Controller* yang diakses tersebut akan melakukan instruksi yang sudah dibuat

sebelumnya. Proses yang dilakukan antara lain mengambil data yang berada di *database* melalui *model*, mengerjakan proses-proses, memanggil *view*, dan menyalurkan hasil proses ke halaman *view*. Tampilan halaman dan hasil proses yang diterima oleh *view* dari *controller* akan terlihat oleh pengguna.

5.1.2.2 Pemodelan kelas

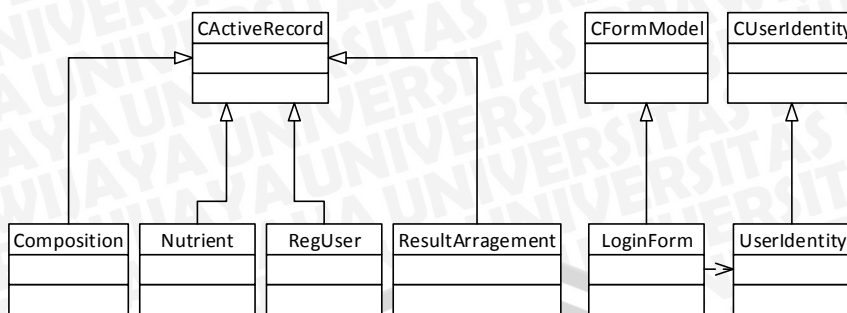
Pemodelan kelas ini merupakan pemodelan yang lebih detail dan lebih bersifat implementatif daripada pemodelan arsitektur. Masing-masing kelas yang akan didefinisikan didalam pemodelan kelas bisa memiliki relasi dengan kelas lainnya. Relasi antar kelas bisa berupa pewarisan (*inheritance*) dan ketergantungan.



Gambar 5.3 Diagram kelas *controller* aplikasi sunran

Diagram kelas pada Gambar 5.3 di atas merupakan diagram dari semua *controller* yang akan dibuat pada aplikasi SunRan. Keenam *controller* yang akan dibuat akan mewarisi kelas *controller* yang dimiliki oleh *framework* Yii. Enam *controller* tersebut memiliki nama akhiran *Controller*. Sebagian besar *controller* dibuat berdasarkan tabel yang berada di *database*. Masing-masing *controller* memiliki fungsi-fungsi / *actions* yang akan diakses saat mengunjungi halaman aplikasi SunRan. Kelas *CompositionController* mengatur pengolahan data pada tabel *composition*. Kelas *FormulasiRansumController* mengatur dalam penyusunan ransum, menampilkan formulasi hasil penyusunan dan penyimpanan formulasi ke *database*. Kelas *NutrientController* mengatur pengolahan data pada tabel *nutrient*. Kelas *RegUserController* pengolahan data pada tabel *reg_user*. Kelas *ResultArragementController* pengolahan data pada tabel *result_arragement* dan membandingkan 2 formulasi ransum. Kelas *SiteController* mengatur proses-proses umum pada aplikasi SunRan seperti *login* dan *logout*.

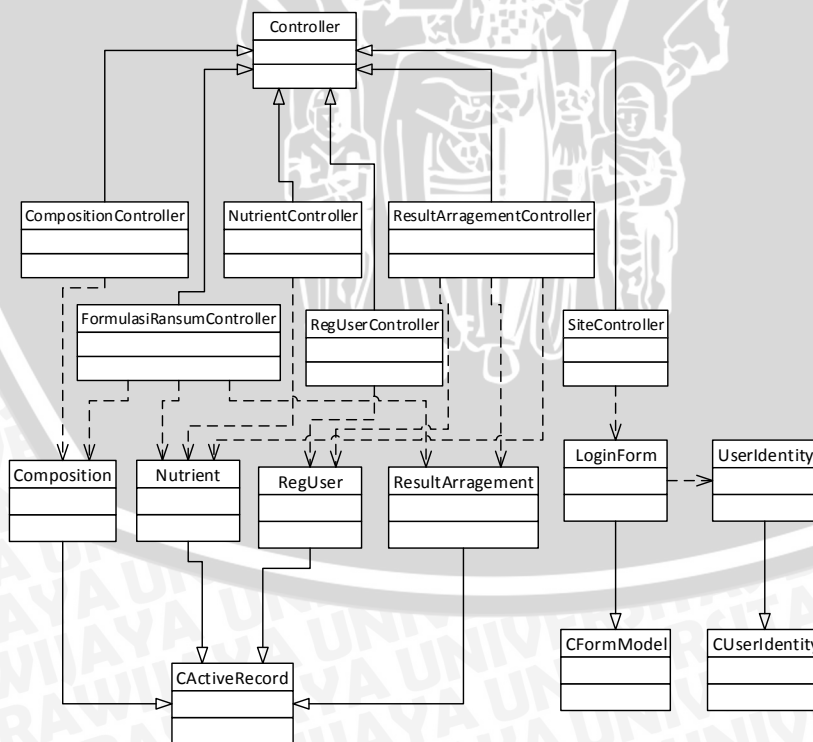
Pemodelan kelas *model* pada aplikasi SunRan sedikit berbeda dengan pemodelan kelas *controller*. Pemodelan kelas *model* pada aplikasi SunRan mewarisi salah satu antara kelas *CFormModel* atau *CActiveRecord*. Berikut diagram kelas *model* pada aplikasi SunRan.



Gambar 5.4 Diagram kelas *model* aplikasi sunran

Diagram 5.4 di atas menggambarkan pemodelan kelas-kelas *model* pada aplikasi SunRan. *Model* Composition, Nutrient, RegUser, dan ResultArragement mewarisi kelas CActiveRecord. Kelas CActiveRecord adalah kelas yang mengatur dalam proses CRUD (*Create, Read, Update, Delete*) terhadap tabel yang berada di *database*. Kelas Composition mengatur proses CRUD tabel composition. Kelas Nutrient mengatur proses CRUD tabel nutrient. Kelas RegUser mengatur proses CRUD tabel reg_user. Kelas ResultArragement mengatur proses CRUD tabel result_arragement. Kelas LoginForm mewarisi kelas CFormModel. Kelas CFormModel adalah kelas yang mengatur tentang pemrosesan *form* pada model. Kelas LoginForm sendiri berfungsi untuk mengatur *form* pada halaman *login*.

Kelas-kelas pada diagram kelas *controller* memiliki relasi dengan kelas-kelas yang digambarkan pada diagram kelas *model*. Relasi kelas-kelas tersebut digambarkan pada gambar berikut.

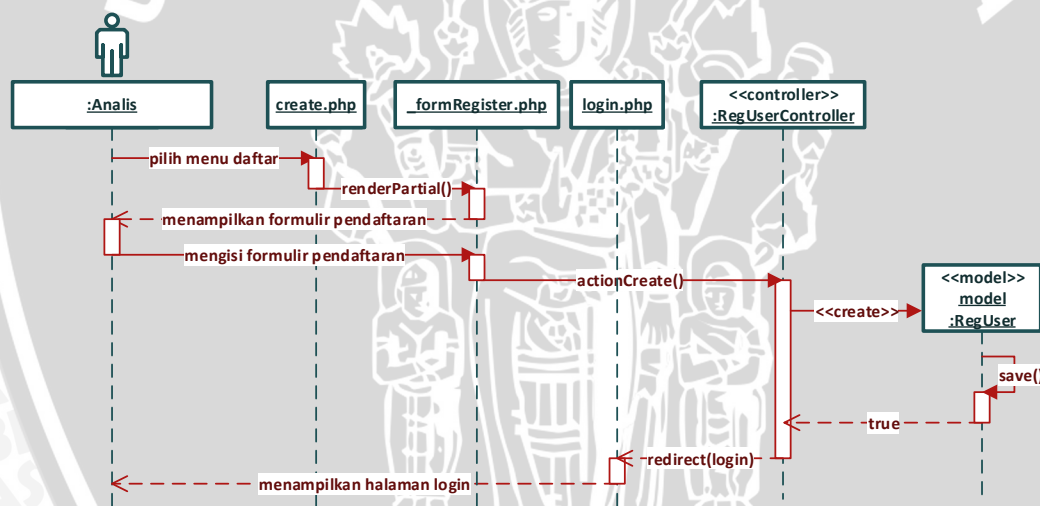


Gambar 5.5 Diagram kelas aplikasi SunRan

Relasi kelas-kelas *controller* dengan kelas-kelas *model* pada Gambar 5.5 di atas merupakan relasi asosiasi. Kelas *controller* *CompositionController* memiliki relasi asosiasi dengan kelas *model* *Composition*. Kelas *controller* *NutrientController* memiliki relasi asosiasi dengan kelas *model* *Nutrient*. Kelas *controller* *RegUserController* memiliki relasi asosiasi dengan kelas *RegUser*. Kelas *controller* *ResultArrangementController* memiliki relasi asosiasi dengan kelas *model* *ResultArrangement*, *Reguser*, dan *Nutrient*. Kelas *controller* *FormulasiRansumController* memiliki relasi asosiasi dengan kelas *model* *Composition*, *Nutrient*, dan *ResultArrangement*. Kelas *controller* *SiteController* memiliki relasi asosiasi dengan kelas *model* *LoginForm*.

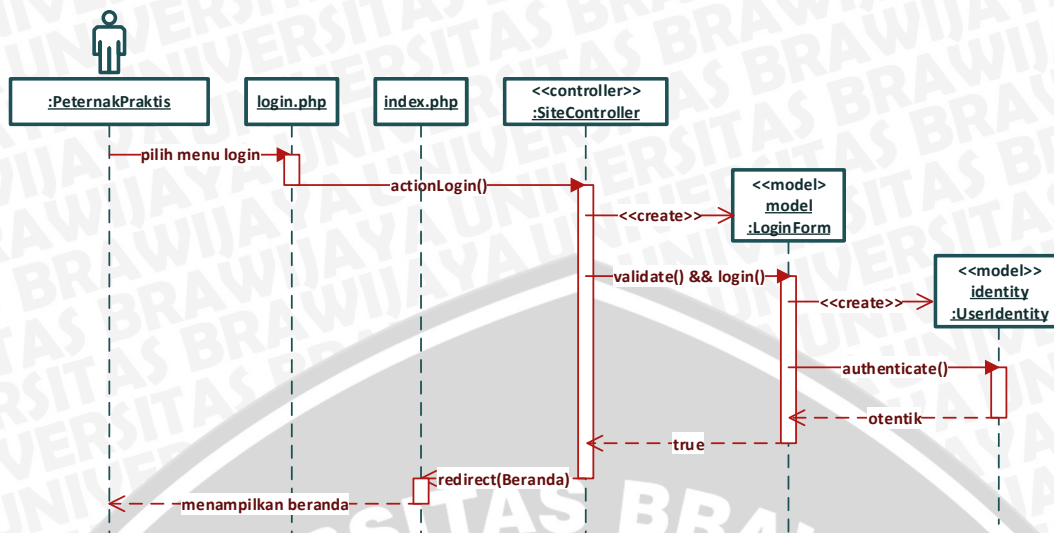
5.1.2.3 Pemodelan interaksi

Pemodelan interaksi pada sub bab ini merupakan perancangan yang menjelaskan interaksi antar objek pada sistem. Objek yang dimaksud bisa berupa actor, *controller*, maupun *model*. Interaksi-interaksi objek akan digambarkan pada diagram urutan / *sequence diagram*. Diagram urutan juga menjelaskan perintah / action dan pesan yang dijalankan oleh sistem pada suatu fungsi. Penggambaran *sequence diagram* pada sub bab ini sebanyak satu diagram untuk masing-masing fungsi yang berdasarkan *use case*.



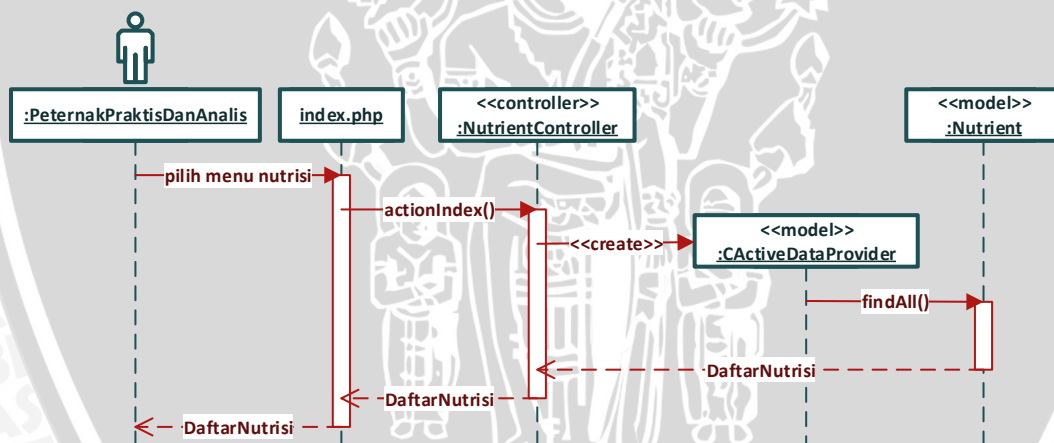
Gambar 5.6 Diagram urutan fungsi mendaftarkan diri

Gambar 5.16 merupakan *sequence diagram* dari fungsi mendaftarkan diri. Objek yang terlibat pada diagram ini adalah aktor analis, *controller* *RegUserController*, *model* *RegUser*, dan beberapa *view*. Objek *view* pada daigram diatas antara lain *create.php*, *_formRegister.php*, dan *login.php*



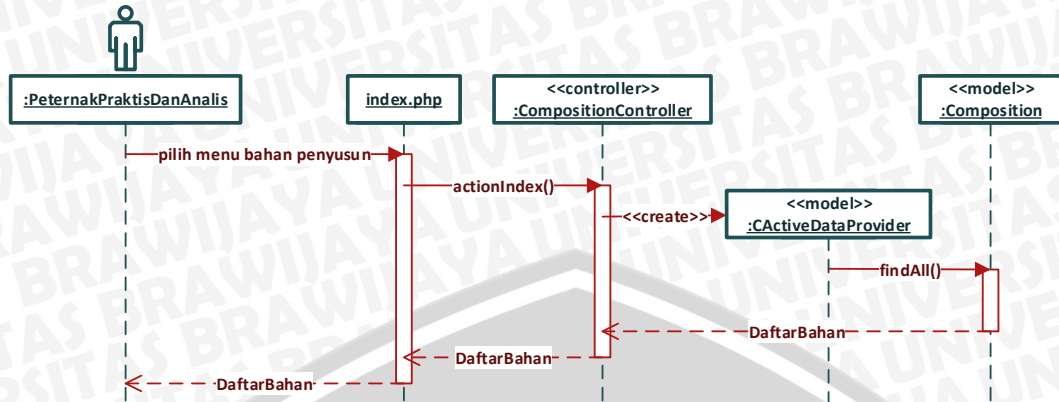
Gambar 5.7 Diagram urutan fungsi login

Gambar 5.7 merupakan *sequence diagram* dari fungsi login / masuk ke sistem. Objek yang terlibat pada diagram ini adalah aktor peternak praktis, *controller* SiteController, dua *model*, dan beberapa *view*. Objek *model* pada diagram di atas antara lain LoginForm dan UserIdentity. Objek *view* pada diagram di atas antara lain login.php dan index.php.



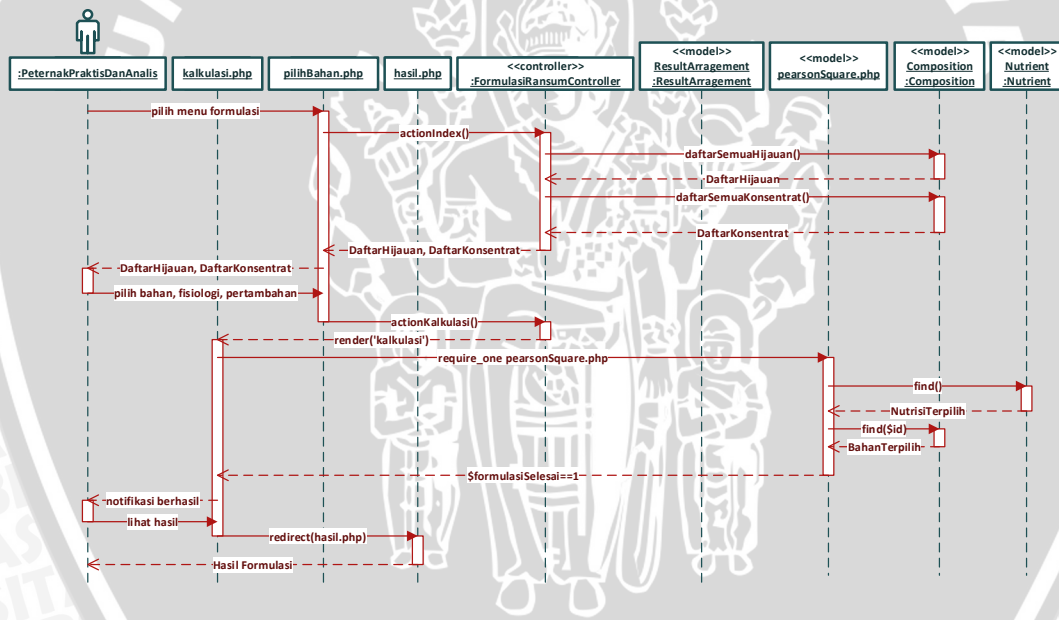
Gambar 5.8 Diagram urutan fungsi melihat tabel nutrisi

Gambar 5.8 merupakan *sequence diagram* dari fungsi melihat tabel nutrisi. Objek yang terlibat pada diagram ini adalah aktor peternak praktis dan analis, *controller* NutrientController, dua *model*, dan *view* index.php. Objek *model* pada diagram di atas antara lain Nutrient dan CActiveDataProvider.



Gambar 5.9 Diagram urutan fungsi melihat tabel komposisi ransum

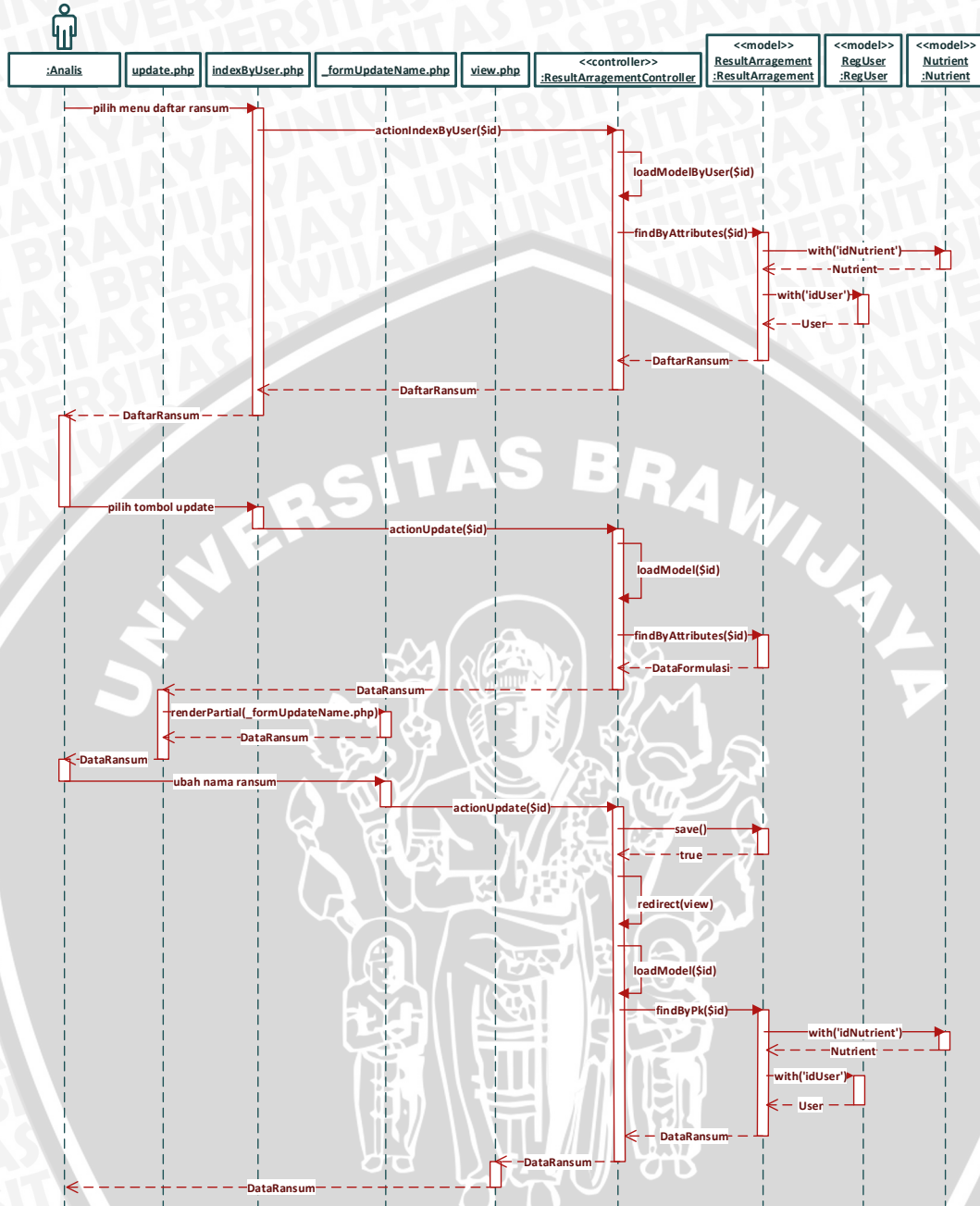
Gambar 5.9 merupakan *sequence diagram* dari fungsi melihat tabel bahan / komposisi penyusun ransum. Objek yang terlibat pada diagram ini adalah aktor peternak praktis dan analis, *controller* CompositionController, dua *model*, dan *view* index.php. Objek *model* pada diagram di atas antara lain Composition dan CActiveDataProvider.



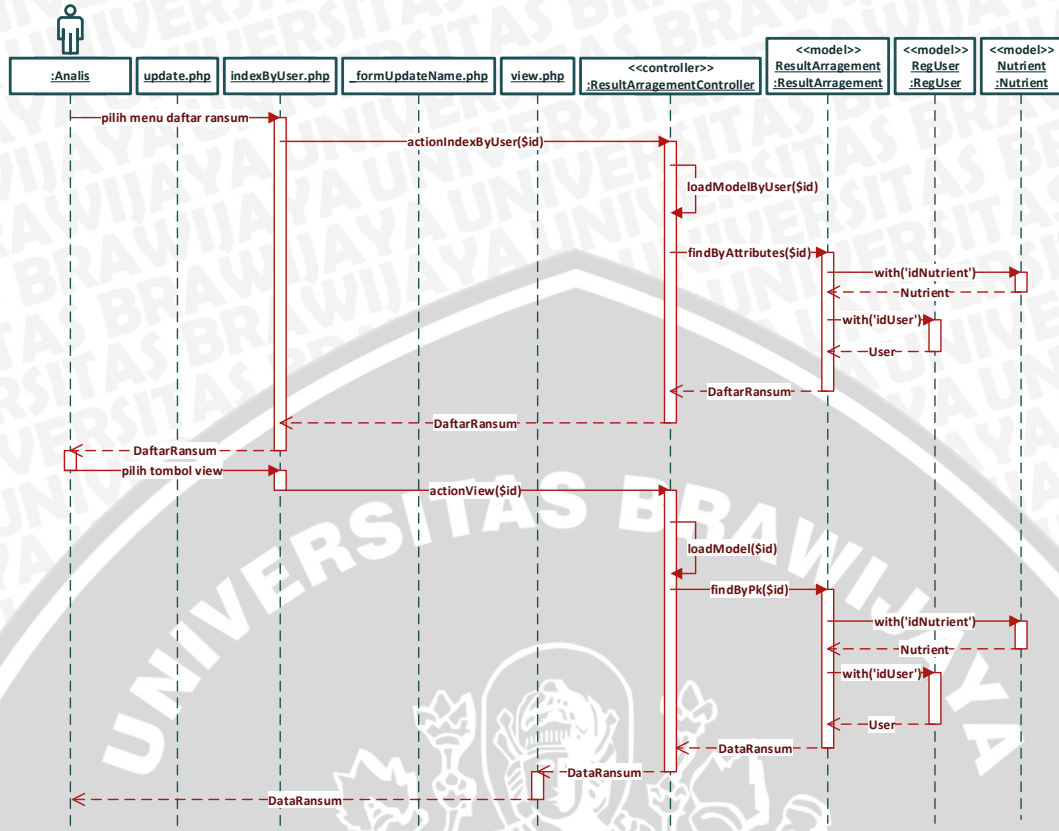
Gambar 5.10 Diagram urutan fungsi menyusun ransum

Gambar 5.10 merupakan *sequence diagram* dari fungsi menyusun ransum. Objek yang terlibat pada diagram ini adalah aktor peternak praktis dan analis, *controller* FormulasiRansumController, *model* ResultArrangement, *model* pearsonSquare, *model* Composition, *model* Nutrient dan beberapa *view*. Objek *view* pada diagram di atas antara lain kalkulasi.php, pilihBahan.php, hasil.php.

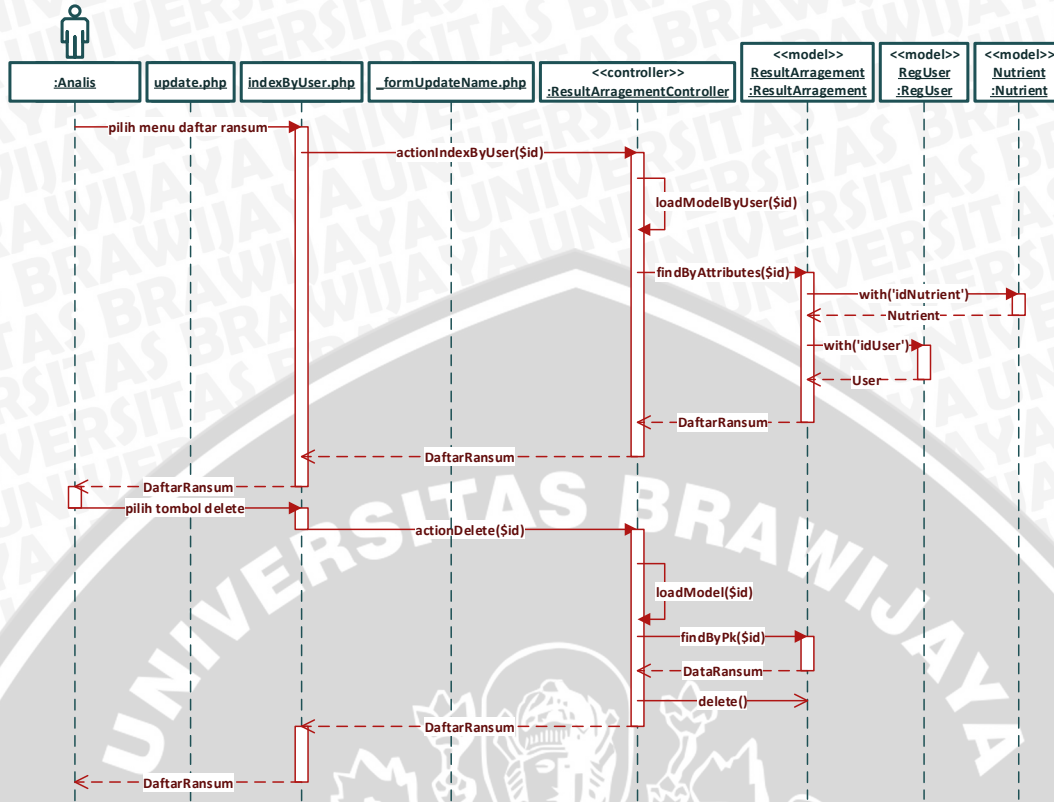




Gambar 5.11 Diagram urutan fungsi mengatur hasil penyusunan ransum (ubah nama ransum)

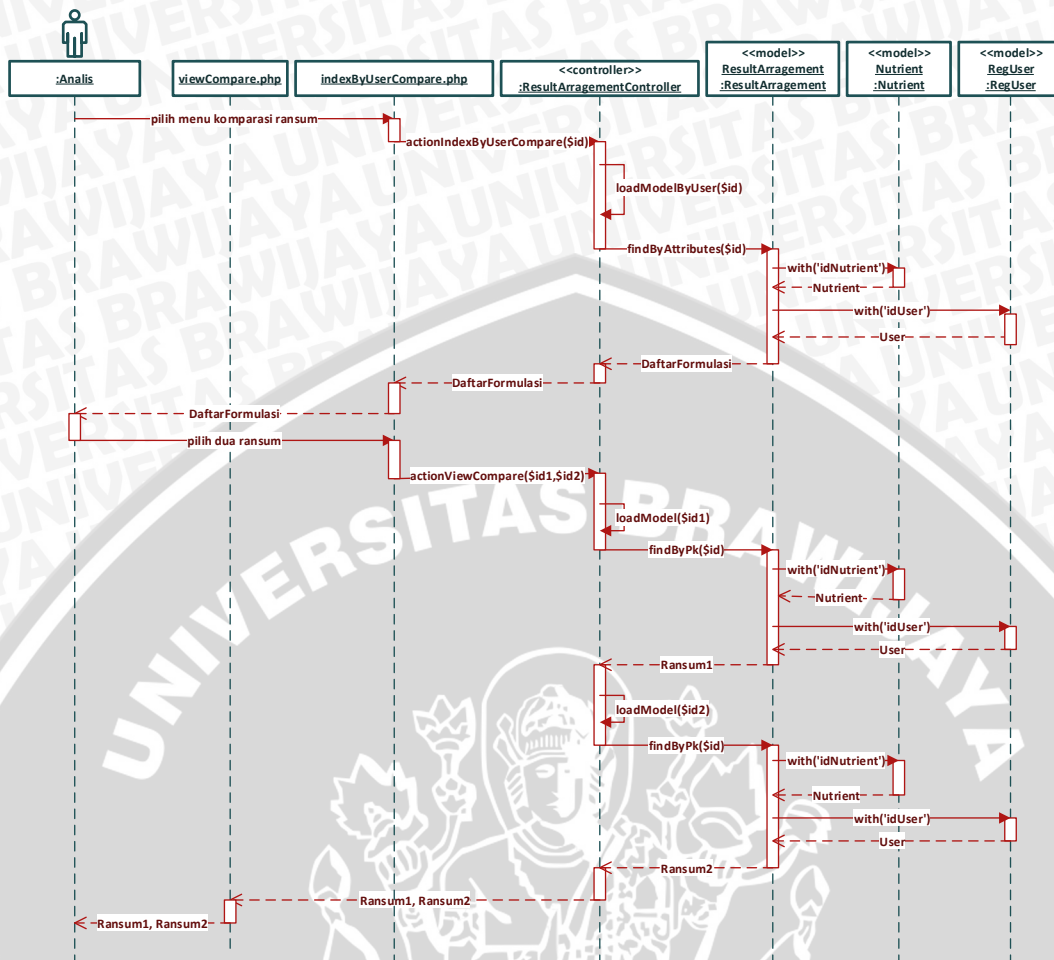


Gambar 5.12 Diagram urutan fungsi mengatur hasil penyusunan ransum (lihat detail)



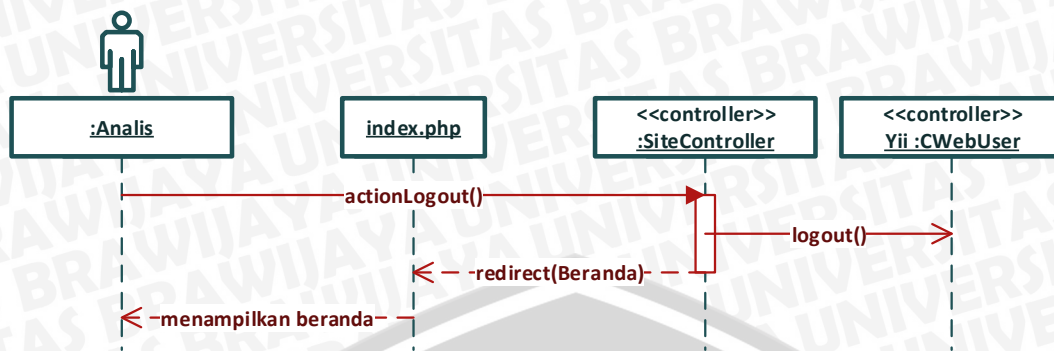
Gambar 5.13 Diagram urutan fungsi mengatur hasil penyusunan ransum (hapus ransum)

Gambar 5.11 hingga 5.13 merupakan *sequence diagram* fungsi mengatur hasil penyusunan ransum dari beberapa alur yang bisa dilakukan oleh aktor analis. Alur yang bisa dilakukan antara lain mengubah (*update*) nama ransum, melihat detail ransum, dan menghapus ransum. Objek yang terlibat pada ketiga diagram di atas antara lain aktor analis, *controller* ResultArrangementController, *model* ResultArrangement, *model* Composition, *model* Nutrient, dan beberapa *view*. Objek *view* pada diagram di atas antara lain *update.php*, *indexByUser.php*, *_formUpdateName.php*, dan *view.php*.



Gambar 5.14 Diagram urutan fungsi membandingkan ransum

Gambar 5.14 merupakan *sequence diagram* dari fungsi membandingkan ransum. Objek yang terlibat pada diagram ini adalah aktor analis, *controller* ResultArrangementController, *model* ResultArrangement, *model* Nutrient, *model* RegUser dan beberapa *view*. Objek *view* pada diagram di atas antara lain *viewCompare.php* dan *indexByUserCompare.php*.



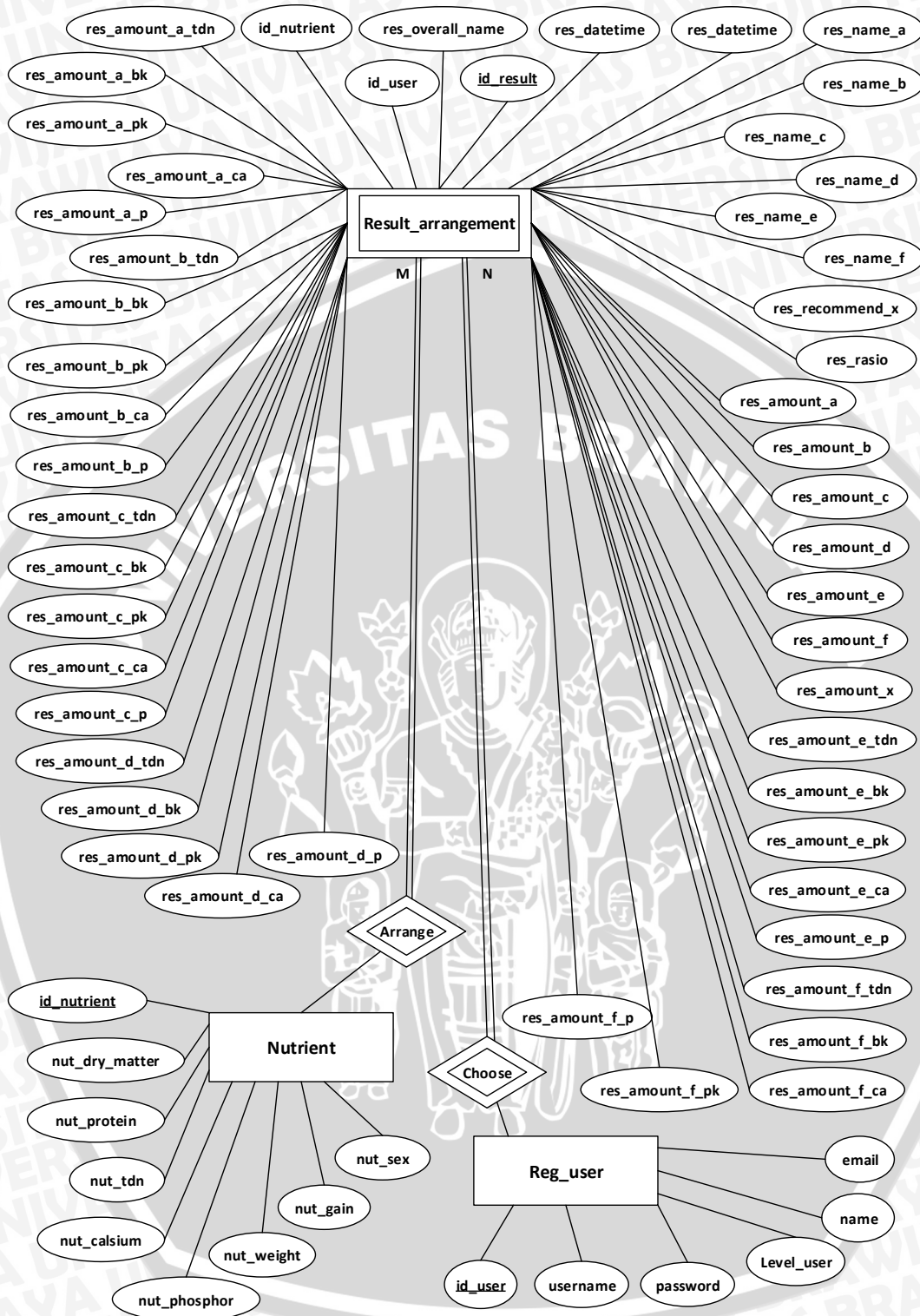
Gambar 5.15 Diagram urutan fungsi *logout*

Gambar 5.15 merupakan *sequence diagram* dari fungsi *logout* / keluar dari sistem. Objek yang terlibat pada diagram ini adalah aktor analis, *controller* SiteController, *controller* CwebUser dan *view* index.php.

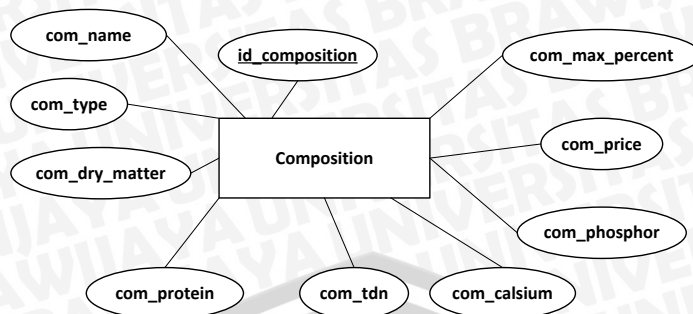
5.1.2.4 Pemodelan data

Aplikasi SunRan yang dibuat ini menggunakan berbagai macam data agar bisa berjalan dengan baik. Data yang akan dipakai perlu untuk di buat pemodelan terlebih dahulu. Pemodelan data yang digunakan digambarkan menggunakan diagram *entity-relationship*. Diagram *entity-relationship* dari data yang digunakan pada aplikasi SunRan ditunjukkan pada gambar berikut.





Gambar 5.16 Diagram *entity-relationship* penyusunan ransom pada aplikasi SunRan



Gambar 5.17 Diagram *entity-relationship* bahan penyusun ransum pada aplikasi SunRan

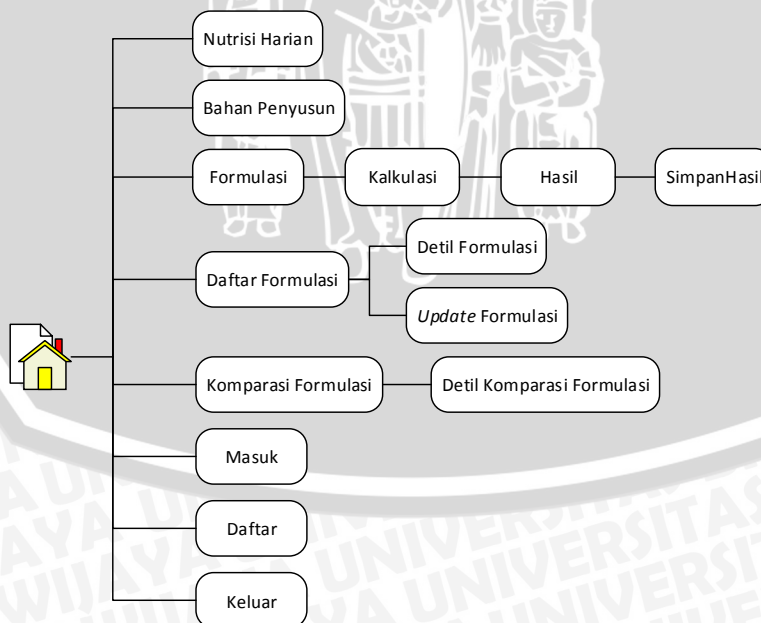
Diagram pada Gambar 5.15 menggambarkan entitas dari bahan penyusun ransum beserta atribut-atributnya. Data dari entitas ini juga dipakai saat proses penyusunan ransum.

5.1.2.5 Perancangan antarmuka

Perancangan antarmuka merupakan perancangan *interface* halaman yang akan ditampilkan selama pengguna mengakses aplikasi Sunran. Perancangan ini juga untuk memudahkan pengguna dalam memahami tampilan aplikasi SunRan. Penjelasan dan perancangan masing-masing antarmuka pada aplikasi SunRan akan dijelaskan pada sub bab berikut.

Perancangan site map

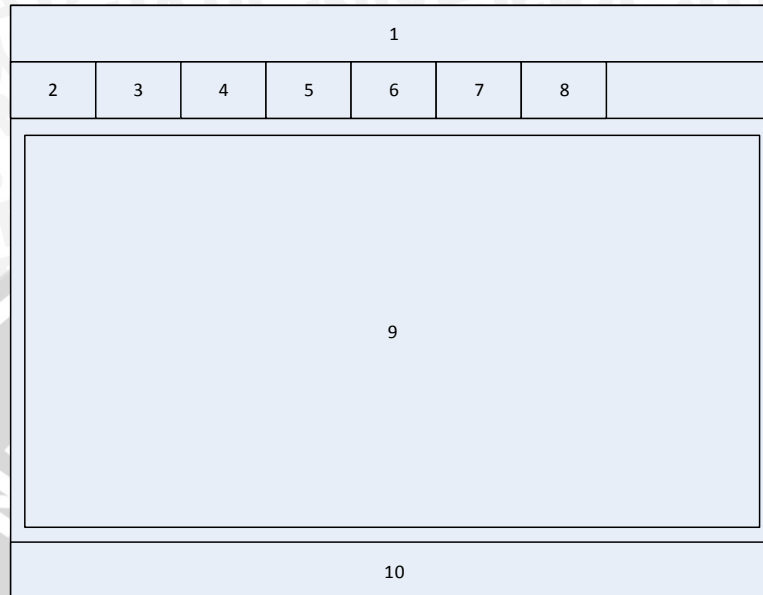
Site map dari aplikasi SunRan menggambarkan halaman yang bisa diakses oleh pengguna. Penggambaran *site map* dari aplikasi SunRan bisa dilihat pada gambar berikut.



Gambar 5.18 Site map aplikasi SunRan

Perancangan halaman heranda

Gambar 5.19 menampilkan rancangan dari halaman beranda dari aplikasi SunRan. Rancangan tersebut juga terdapat nomor petunjuk untuk masing-masing komponen yang ditampilkan.



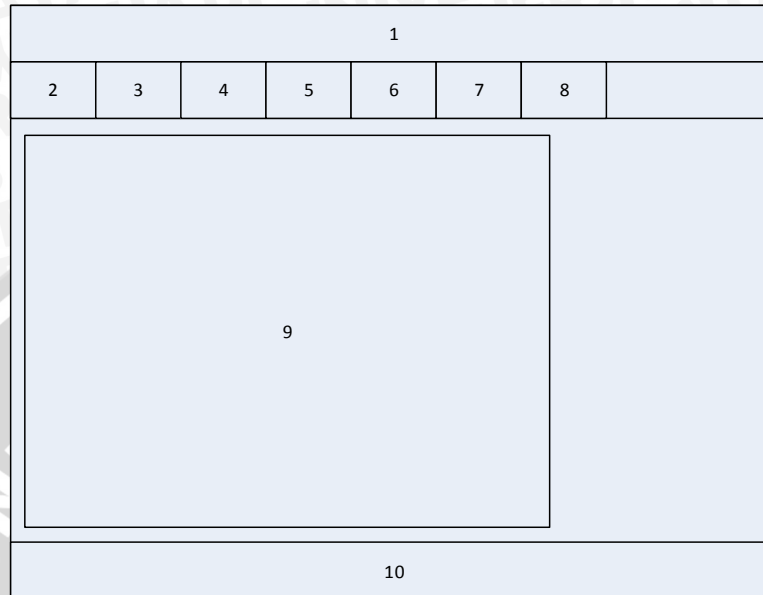
Gambar 5.19 Perancangan antarmuka halaman beranda sistem SunRan

Halaman beranda menampilkan gambaran umum mengenai apa saja yang bisa dilakukan oleh pengguna. Keterangan nomor-nomor petunjuk komponen pada Gambar 5.19 antara lain:

1. *Header* berisi judul sistem aplikasi.
2. Menu beranda yang akan mengarahkan ke halaman utama.
3. Menu nutrisi harian untuk memberikan informasi daftar nutrisi harian minimum.
4. Menu bahan penyusun untuk memberikan informasi daftar bahan penyusun ransum.
5. Menu formulasi ransum untuk melakukan penyusunan ransum
6. Menu daftar ransum untuk menampilkan ransum-ransum tersimpan yang merupakan hasil penyusunan sebelumnya agar bisa melakukan ubah / *update* nama formulasi, melihat detail ransum dan menghapus ransum.
7. Menu komparasi ransum untuk membandingkan 2 ransum tersimpan yang merupakan hasil penyusunan sebelumnya.
8. Menu keluar untuk *logout* / keluar dari sistem.
9. Konten halaman beranda yang menampilkan kemampuan yang bisa dilakukan oleh aplikasi SunRan.
10. *Footer* berisi hak cipta, tahun pembuatan, dan nama pengembang aplikasi SunRan.

Perancangan halaman nutrisi harian

Gambar 5.20 menampilkan rancangan dari halaman daftar nutrisi harian. Rancangan tersebut juga terdapat nomer petunjuk untuk masing-masing komponen yang ditampilkan.



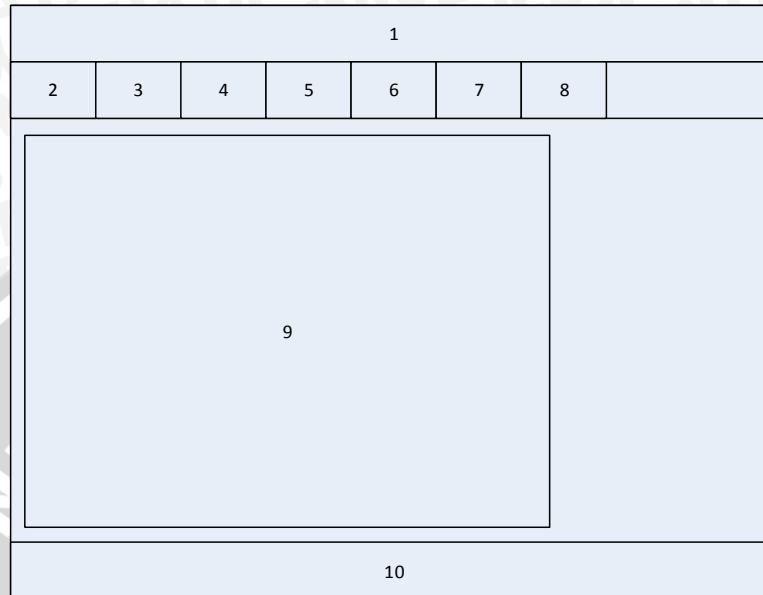
Gambar 5.20 Perancangan antarmuka nutrisi harian

Halaman nutrisi harian menampilkan daftar nutrisi minimum sapi potong yang akan digunakan dalam penyusunan ransum. Keterangan nomor-nomor petunjuk komponen pada Gambar 5.20 antara lain:

1. *Header* berisi judul sistem aplikasi.
2. Menu beranda yang akan mengarahkan ke halaman utama.
3. Menu nutrisi harian untuk memberikan informasi daftar nutrisi harian minimum.
4. Menu bahan penyusun untuk memberikan informasi daftar bahan penyusun ransum.
5. Menu formulasi ransum untuk melakukan penyusunan ransum
6. Menu daftar ransum untuk menampilkan ransum-ransum tersimpan yang merupakan hasil penyusunan sebelumnya agar bisa melakukan ubah / *update* nama formulasi, melihat detail ransum dan menghapus ransum.
7. Menu komparasi ransum untuk membandingkan 2 ransum tersimpan yang merupakan hasil penyusunan sebelumnya.
8. Menu keluar untuk *logout* / keluar dari sistem.
9. Tabel nutrisi harian sapi potong yang tersedia di dalam aplikasi SunRan.
10. *Footer* berisi hak cipta, tahun pembuatan, dan nama pengembang aplikasi SunRan.

Perancangan halaman bahan penyusun

Gambar 5.21 menampilkan rancangan dari halaman bahan penyusun. Rancangan tersebut juga terdapat nomer penunjuk untuk masing-masing komponen yang ditampilkan.



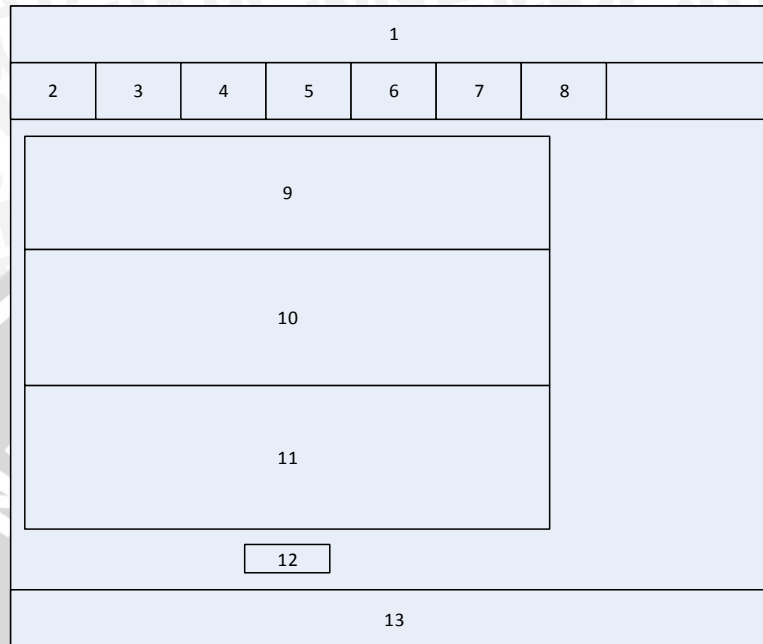
Gambar 5.21 Perancangan antarmuka halaman bahan penyusun

Halaman bahan penyusun menampilkan daftar bahan (hijauan dan konsentrat) yang akan digunakan dalam penyusunan ransum. Keterangan nomor-nomor petunjuk komponen pada Gambar 5.21 antara lain:

1. *Header* berisi judul sistem aplikasi.
2. Menu beranda yang akan mengarahkan ke halaman utama.
3. Menu nutrisi harian untuk memberikan informasi daftar nutrisi harian minimum.
4. Menu bahan penyusun untuk memberikan informasi daftar bahan penyusun ransum.
5. Menu formulasi ransum untuk melakukan penyusunan ransum
6. Menu daftar ransum untuk menampilkan ransum-ransum tersimpan yang merupakan hasil penyusunan sebelumnya agar bisa melakukan ubah / *update* nama formulasi, melihat detail ransum dan menghapus ransum.
7. Menu komparasi ransum untuk membandingkan 2 ransum tersimpan yang merupakan hasil penyusunan sebelumnya.
8. Menu keluar untuk *logout* / keluar dari sistem.
9. Tabel bahan-bahan penyusun yang tersedia di dalam aplikasi SunRan.
10. *Footer* berisi hak cipta, tahun pembuatan, dan nama pengembang aplikasi SunRan.

Perancangan halaman pemilihan bahan untuk penyusunan ransum

Gambar 5.22 menampilkan rancangan dari halaman pemilihan bahan untuk proses formulasi ransum. Rancangan tersebut juga terdapat nomer penunjuk untuk masing-masing komponen yang ditampilkan.



Gambar 5.22 Perancangan antarmuka halaman pemilihan bahan untuk penyusunan ransum

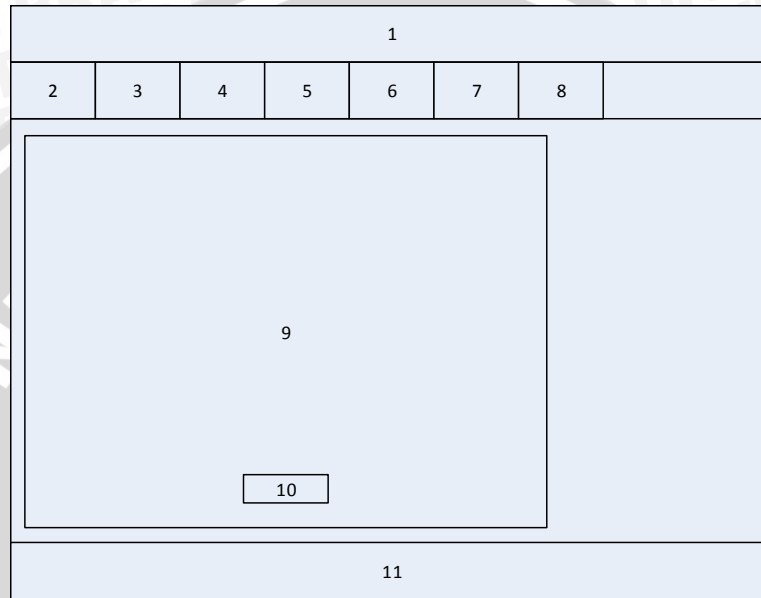
Halaman pemilihan bahan menampilkan *dropdown option* untuk data jenis kelamin sapi, berat badan, dan pertambahan berat badan per hari serta daftar bahan (hijauan dan konsentrat) yang akan digunakan dalam penyusunan ransum. Keterangan nomor-nomor petunjuk komponen pada Gambar 5.22 antara lain:

1. *Header* berisi judul sistem aplikasi.
2. Menu beranda yang akan mengarahkan ke halaman utama.
3. Menu nutrisi harian untuk memberikan informasi daftar nutrisi harian minimum.
4. Menu bahan penyusun untuk memberikan informasi daftar bahan penyusun ransum.
5. Menu formulasi ransum untuk melakukan penyusunan ransum
6. Menu daftar ransum untuk menampilkan ransum-ransum tersimpan yang merupakan hasil penyusunan sebelumnya agar bisa melakukan ubah / *update* nama formulasi, melihat detail ransum dan menghapus ransum.
7. Menu komparasi ransum untuk membandingkan 2 ransum tersimpan yang merupakan hasil penyusunan sebelumnya.
8. Menu keluar untuk *logout* / keluar dari sistem.
9. *Dropdown option* untuk data jenis kelamin sapi, berat badan, dan pertambahan berat badan per hari.
10. Daftar bahan hijauan penyusun ransum.
11. Daftar bahan konsentrat penyusun ransum.

12. Tombol memulai penyusunan ransum
13. Footer berisi hak cipta, tahun pembuatan, dan nama pengembang aplikasi SunRan.

Perancangan halaman notifikasi hasil penyusunan ransum

Gambar 5.23 menampilkan rancangan dari halaman notifikasi terntang berhasil atau tidaknya proses penyusunan ransum. Rancangan tersebut juga terdapat nomer penunjuk untuk masing-masing komponen yang ditampilkan.



Gambar 5.23 Perancangan antarmuka halaman notifikasi hasil penyusunan ransum

Halaman notifikasi hasil penyusunan ransum menampilkan notifikasi mengenai berhasil atau tidaknya dalam penemuan hasil penyusunan ransum. Keterangan nomor-nomor petunjuk komponen pada Gambar 5.23 antara lain:

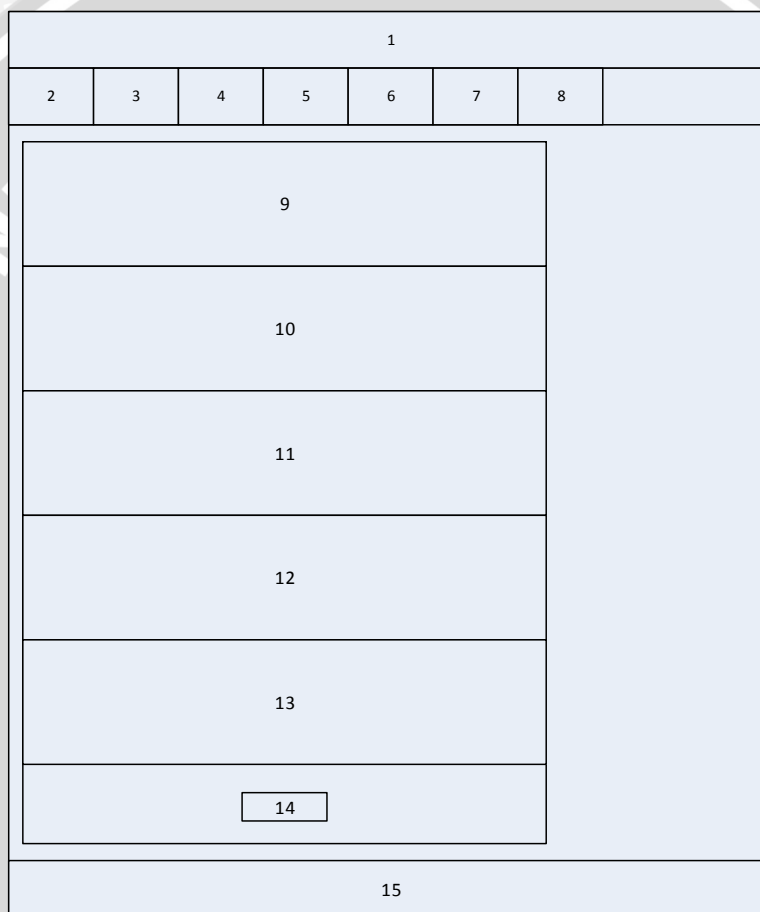
1. Header berisi judul sistem aplikasi.
2. Menu beranda yang akan mengarahkan ke halaman utama.
3. Menu nutrisi harian untuk memberikan informasi daftar nutrisi harian minimum.
4. Menu bahan penyusun untuk memberikan informasi daftar bahan penyusun ransum.
5. Menu formulasi ransum untuk melakukan penyusunan ransum
6. Menu daftar ransum untuk menampilkan ransum-ransum tersimpan yang merupakan hasil penyusunan sebelumnya agar bisa melakukan ubah / *update* nama formulasi, melihat detail ransum dan menghapus ransum.
7. Menu komparasi ransum untuk membandingkan 2 ransum tersimpan yang merupakan hasil penyusunan sebelumnya.
8. Menu keluar untuk *logout* / keluar dari sistem.
9. Tulisan notifikasi mengenai berhasil atau tidaknya dalam penemuan hasil penyusunan ransum.



10. Tombol untuk menuju halaman detail hasil penyusunan ransum (saat berhasil menemukan penyusunan ransum yang ideal). *Button* ini akan mengarahkan kembali ke halaman pemilihan bahan penyusun saat tidak ditemukan ransum yang ideal.
11. *Footer* berisi hak cipta, tahun pembuatan, dan nama pengembang aplikasi SunRan.

Perancangan halaman detail hasil penyusunan ransum

Gambar 5.24 menampilkan rancangan dari halaman detail hasil penyusunan ransum. Rancangan tersebut juga terdapat nomer penunjuk untuk masing-masing komponen yang ditampilkan.



Gambar 5.24 Perancangan antarmuka halaman detail hasil penyusunan

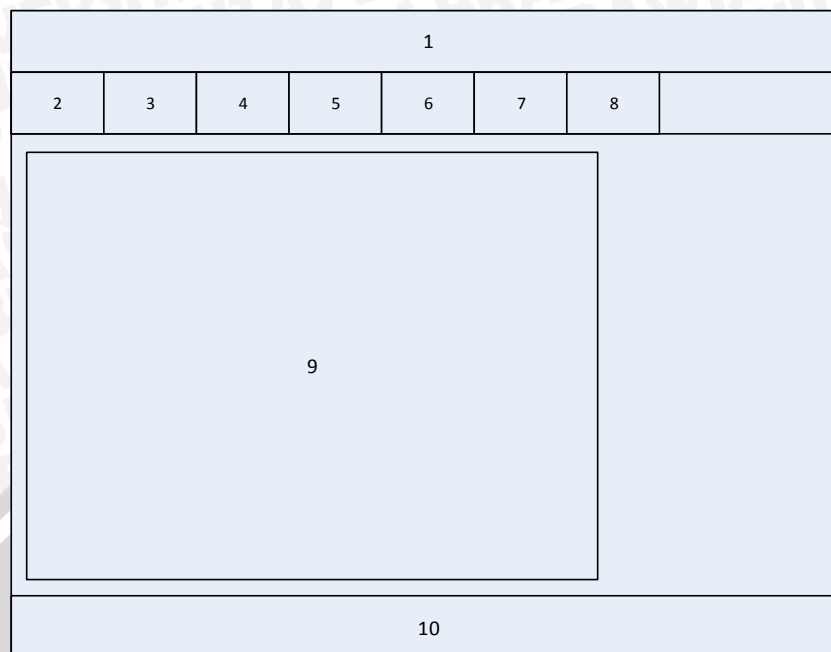
Halaman detail hasil penyusunan ransum menampilkan hasil penyusunan ransum berupa detail fisiologi sapi potong (jenis kelamin, berat badan awal, dan pertambahan per hari), beberapa atribut dari formulasi ransum (seperti nama formulasi dan tanggal penyusunan), dan nama-nama bahan penyusun ransum, tabel nutrisi dari masing-masing bahan penyusun ransum beserta kebutuhan nutrisi harian, diagram perbandingan antara keseluruhan nutrisi dari formulasi ransum dan kebutuhan nutrisi harian, diagram pembagian berat untuk masing-masing bahan penyusun, diagram penggambaran dari simulasi pertambahan

berat badan sapi potong selama 12 minggu. Keterangan nomor-nomor petunjuk komponen pada Gambar 5.24 antara lain:

1. *Header* berisi judul sistem aplikasi.
2. Menu beranda yang akan mengarahkan ke halaman utama.
3. Menu nutrisi harian untuk memberikan informasi daftar nutrisi harian minimum.
4. Menu bahan penyusun untuk memberikan informasi daftar bahan penyusun ransum.
5. Menu formulasi ransum untuk melakukan penyusunan ransum
6. Menu daftar ransum untuk menampilkan ransum-ransum tersimpan yang merupakan hasil penyusunan sebelumnya agar bisa melakukan ubah / *update* nama formulasi, melihat detail ransum dan menghapus ransum.
7. Menu komparasi ransum untuk membandingkan 2 ransum tersimpan yang merupakan hasil penyusunan sebelumnya.
8. Menu keluar untuk *logout* / keluar dari sistem.
9. Detail fisiologi sapi potong (jenis kelamin, berat badan awal, dan penambahan per hari), beberapa atribut dari formulasi ransum (seperti nama formulasi dan tanggal penyusunan), dan nama-nama bahan penyusun ransum.
10. Tabel nutrisi dari masing-masing bahan penyusun ransum beserta kebutuhan nutrisi harian.
11. Diagram perbandingan antara keseluruhan nutrisi dari formulasi ransum dan kebutuhan nutrisi harian.
12. Diagram pembagian berat untuk masing-masing bahan penyusun.
13. Diagram penggambaran dari simulasi penambahan berat badan sapi potong selama 12 minggu.
14. Tombol untuk menyimpan hasil penyusunan ransum.
15. *Footer* berisi hak cipta, tahun pembuatan, dan nama pengembang aplikasi SunRan.

Perancangan halaman daftar hasil penyusunan ransum tersimpan

Gambar 5.25 menampilkan rancangan dari halaman daftar hasil penyusunan yang sudah disimpan. Rancangan tersebut juga terdapat nomer penunjuk untuk masing-masing komponen yang ditampilkan.



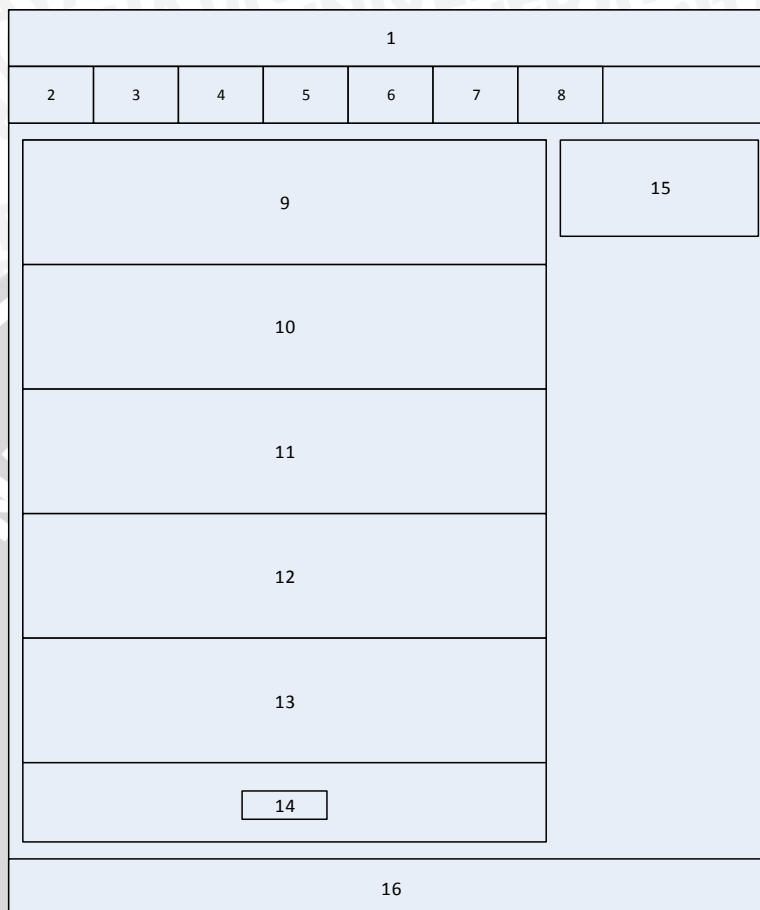
Gambar 5.25 Perancangan antarmuka halaman daftar hasil penyusunan ransum tersimpan

Halaman daftar hasil penyusunan ransum tersimpan menampilkan daftar hasil formulasi tersimpan yang merupakan hasil penyusunan sebelumnya beserta tombol untuk melihat detail, *update*, dan menghapus formulasi. Keterangan nomor-nomor petunjuk komponen pada Gambar 5.25 antara lain:

1. *Header* berisi judul sistem aplikasi.
2. Menu beranda yang akan mengarahkan ke halaman utama.
3. Menu nutrisi harian untuk memberikan informasi daftar nutrisi harian minimum.
4. Menu bahan penyusun untuk memberikan informasi daftar bahan penyusun ransum.
5. Menu formulasi ransum untuk melakukan penyusunan ransum
6. Menu daftar ransum untuk menampilkan ransum-ransum tersimpan yang merupakan hasil penyusunan sebelumnya agar bisa melakukan ubah / *update* nama formulasi, melihat detail ransum dan menghapus ransum.
7. Menu komparasi ransum untuk membandingkan 2 ransum tersimpan yang merupakan hasil penyusunan sebelumnya.
8. Menu keluar untuk *logout* / keluar dari sistem.
9. Tabel hasil-hasil formulasi tersimpan yang merupakan hasil penyusunan sebelumnya beserta tombol untuk melihat detail, *update*, dan menghapus formulasi.
10. *Footer* berisi hak cipta, tahun pembuatan, dan nama pengembang aplikasi SunRan.

Perancangan halaman detail hasil penyusunan ransum tersimpan

Gambar 5.26 menampilkan rancangan dari halaman detail hasil penyusunan ransum yang sudah tersimpan. Rancangan tersebut juga terdapat nomer penunjuk untuk masing-masing komponen yang ditampilkan.



Gambar 5.26 Perancangan antarmuka halaman detail hasil ransum penyusunan tersimpan

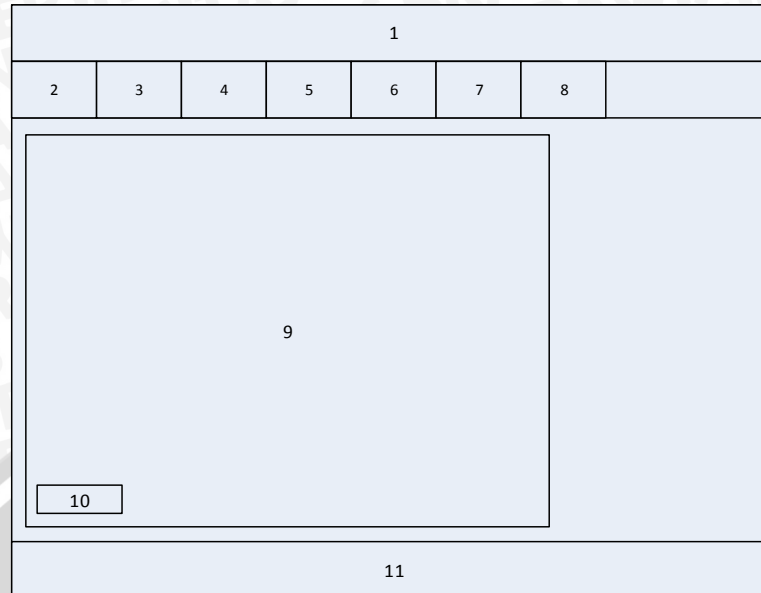
Halaman detail hasil penyusunan ransum tersimpan menampilkan berupa detail fisiologi sapi potong (jenis kelamin, berat badan awal, dan pertambahan per hari), beberapa atribut dari formulasi ransum (seperti nama formulasi dan tanggal penyusunan), dan nama-nama bahan penyusun ransum, tabel nutrisi dari masing-masing bahan penyusun ransum beserta kebutuhan nutrisi harian, diagram perbandingan antara keseluruhan nutrisi dari formulasi ransum dan kebutuhan nutrisi harian, diagram pembagian berat untuk masing-masing bahan penyusun, diagram penggambaran dari simulasi pertambahan berat badan sapi potong selama 12 minggu. Keterangan nomor-nomor petunjuk komponen pada Gambar 5.26 antara lain:

1. *Header* berisi judul sistem aplikasi.
2. Menu beranda yang akan mengarahkan ke halaman utama.
3. Menu nutrisi harian untuk memberikan informasi daftar nutrisi harian minimum.

4. Menu bahan penyusun untuk memberikan informasi daftar bahan penyusun ransum.
5. Menu formulasi ransum untuk melakukan penyusunan ransum
6. Menu daftar ransum untuk menampilkan ransum-ransum tersimpan yang merupakan hasil penyusunan sebelumnya agar bisa melakukan ubah / *update* nama formulasi, melihat detail ransum dan menghapus ransum.
7. Menu komparasi ransum untuk membandingkan 2 ransum tersimpan yang merupakan hasil penyusunan sebelumnya.
8. Menu keluar untuk *logout* / keluar dari sistem.
9. Detail fisiologi sapi potong (jenis kelamin, berat badan awal, dan penambahan per hari), beberapa atribut dari formulasi ransum (seperti nama formulasi dan tanggal penyusunan), dan nama-nama bahan penyusun ransum.
10. Tabel nutrisi dari masing-masing bahan penyusun ransum beserta kebutuhan nutrisi harian.
11. Diagram perbandingan antara keseluruhan nutrisi dari formulasi ransum dan kebutuhan nutrisi harian.
12. Diagram pembagian berat untuk masing-masing bahan penyusun.
13. Diagram penggambaran dari simulasi penambahan berat badan sapi potong selama 12 minggu.
14. Tombol untuk mencetak hasil penyusunan ransum.
15. Menu untuk melakukan *update* hasil penyusunan ransum.
16. *Footer* berisi hak cipta, tahun pembuatan, dan nama pengembang aplikasi SunRan.

Perancangan halaman *update* hasil penyusunan ransum tersimpan

Gambar 5.27 menampilkan rancangan dari halaman *update* hasil penyusunan ransum yang sudah tersimpan. Rancangan tersebut juga terdapat nomer penunjuk untuk masing-masing komponen yang ditampilkan.



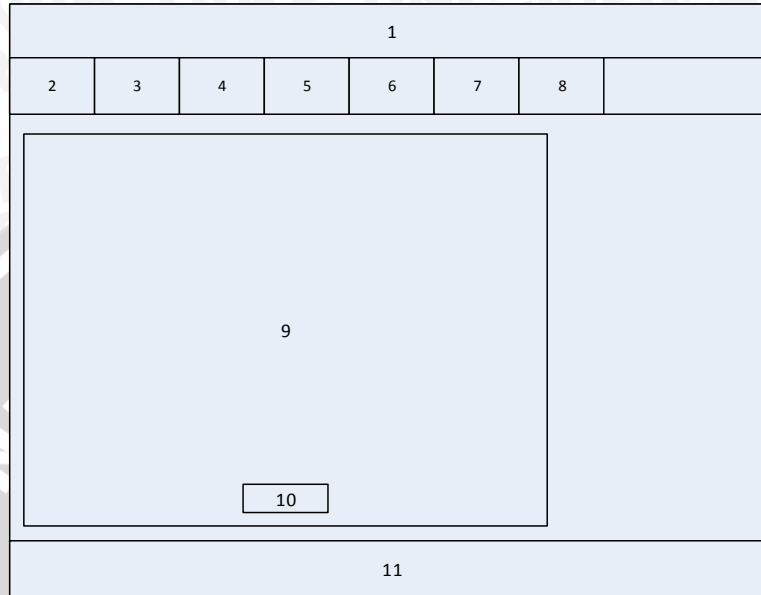
Gambar 5.27 Perancangan antarmuka halaman *update* hasil penyusunan ransum tersimpan

Halaman *update* hasil penyusunan ransum tersimpan menampilkan formulir untuk melakukan *update* formulasi. Keterangan nomor-nomor petunjuk komponen pada Gambar 5.27 antara lain:

1. *Header* berisi judul sistem aplikasi.
2. Menu beranda yang akan mengarahkan ke halaman utama.
3. Menu nutrisi harian untuk memberikan informasi daftar nutrisi harian minimum.
4. Menu bahan penyusun untuk memberikan informasi daftar bahan penyusun ransum.
5. Menu formulasi ransum untuk melakukan penyusunan ransum
6. Menu daftar ransum untuk menampilkan ransum-ransum tersimpan yang merupakan hasil penyusunan sebelumnya agar bisa melakukan ubah / *update* nama formulasi, melihat detail ransum dan menghapus ransum.
7. Menu komparasi ransum untuk membandingkan 2 ransum tersimpan yang merupakan hasil penyusunan sebelumnya.
8. Menu keluar untuk *logout* / keluar dari sistem.
9. Formulir untuk melakukan *update* formulasi.
10. Tombol untuk menyimpan *update* formulasi.
11. *Footer* berisi hak cipta, tahun pembuatan, dan nama pengembang aplikasi SunRan.

Perancangan halaman daftar hasil penyusunan ransum tersimpan untuk komparasi

Gambar 5.28 menampilkan rancangan dari halaman daftar hasil penyusunan ransum yang sudah tersimpan untuk dibandingkan. Rancangan tersebut juga terdapat nomer penunjuk untuk masing-masing komponen yang ditampilkan.



Gambar 5.28 Perancangan antarmuka halaman daftar hasil penyusunan ransum tersimpan untuk komparasi

Halaman daftar hasil penyusunan ransum tersimpan untuk komparasi menampilkan daftar hasil formulasi tersimpan yang merupakan hasil penyusunan sebelumnya beserta *checkbox* pada masing-masing hasil penyusunan agar bisa memilih untuk dibandingkan. Keterangan nomor-nomor petunjuk komponen pada Gambar 5.28 antara lain:

1. *Header* berisi judul sistem aplikasi.
2. Menu beranda yang akan mengarahkan ke halaman utama.
3. Menu nutrisi harian untuk memberikan informasi daftar nutrisi harian minimum.
4. Menu bahan penyusun untuk memberikan informasi daftar bahan penyusun ransum.
5. Menu formulasi ransum untuk melakukan penyusunan ransum
6. Menu daftar ransum untuk menampilkan ransum-ransum tersimpan yang merupakan hasil penyusunan sebelumnya agar bisa melakukan ubah / *update* nama formulasi, melihat detail ransum dan menghapus ransum.
7. Menu komparasi ransum untuk membandingkan 2 ransum tersimpan yang merupakan hasil penyusunan sebelumnya.
8. Menu keluar untuk *logout* / keluar dari sistem.
9. Tabel hasil formulasi tersimpan yang merupakan hasil penyusunan sebelumnya beserta *checkbox* pada masing-masing hasil penyusunan agar bisa memilih untuk dibandingkan.

10. Tombol untuk memulai membandingkan hasil penyusunan ransum
11. Footer berisi hak cipta, tahun pembuatan, dan nama pengembang aplikasi SunRan.

Perancangan halaman detail komparasi hasil penyusunan ransum tersimpan

Gambar 5.29 menampilkan rancangan dari halaman detail komparasi 2 hasil penyusunan ransum yang sudah tersimpan. Rancangan tersebut juga terdapat nomer penunjuk untuk masing-masing komponen yang ditampilkan.



1							
2	3	4	5	6	7	8	
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							

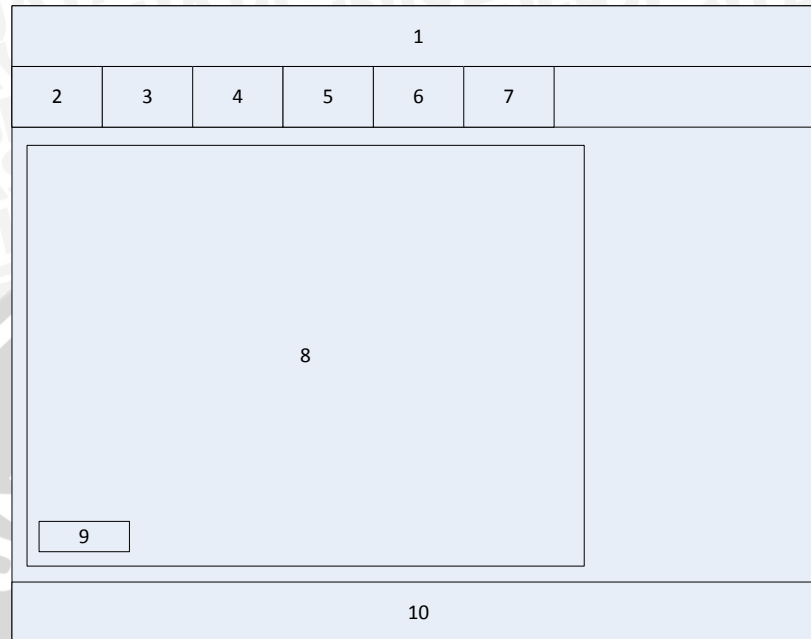
Gambar 5.29 Perancangan antarmuka halaman detail komparasi hasil penyusunan ransum tersimpan

Halaman detail komparasi hasil penyusunan ransum tersimpan menampilkan detail komparasi berupa detail fisiologi sapi potong (jenis kelamin, berat badan awal, dan pertambahan per hari), beberapa atribut dari formulasi ransum (seperti nama formulasi dan tanggal penyusunan), dan nama-nama bahan penyusun ransum, tabel nutrisi dari masing-masing bahan penyusun ransum beserta kebutuhan nutrisi harian, diagram perbandingan antara keseluruhan nutrisi dari formulasi ransum dan kebutuhan nutrisi harian, diagram pembagian berat untuk masing-masing bahan penyusun, diagram penggambaran dari simulasi pertambahan berat badan sapi potong selama 12 minggu. Keterangan nomor-nomor petunjuk komponen pada Gambar 5.29 antara lain:

1. *Header* berisi judul sistem aplikasi.
2. Menu beranda yang akan mengarahkan ke halaman utama.
3. Menu nutrisi harian untuk memberikan informasi daftar nutrisi harian minimum.
4. Menu bahan penyusun untuk memberikan informasi daftar bahan penyusun ransum.
5. Menu formulasi ransum untuk melakukan penyusunan ransum
6. Menu daftar ransum untuk menampilkan ransum-ransum tersimpan yang merupakan hasil penyusunan sebelumnya agar bisa melakukan ubah / *update* nama formulasi, melihat detail ransum dan menghapus ransum.
7. Menu komparasi ransum untuk membandingkan 2 ransum tersimpan yang merupakan hasil penyusunan sebelumnya.
8. Menu keluar untuk *logout* / keluar dari sistem.
9. Detail fisiologi sapi potong (jenis kelamin, berat badan awal, dan penambahan per hari) dan nama-nama bahan penyusun ransum dari masing-masing hasil formulasi.
10. Tabel total nutrisi dari kedua hasil formulasi ransum beserta kebutuhan nutrisi harian.
11. Diagram perbandingan antara keseluruhan nutrisi dari kedua formulasi ransum dan kebutuhan nutrisi harian.
12. Diagram penggambaran dari simulasi penambahan berat badan sapi potong selama 12 minggu.
13. Detail fisiologi sapi potong (jenis kelamin, berat badan awal, dan penambahan per hari), beberapa atribut dari formulasi ransum (seperti nama formulasi dan tanggal penyusunan), dan nama-nama bahan penyusun ransum dari hasil formulasi pertama.
14. Tabel nutrisi dari masing-masing bahan penyusun hasil formulasi ransum pertama beserta kebutuhan nutrisi harian
15. Diagram pembagian berat untuk masing-masing bahan penyusun hasil formulasi pertama.
16. Detail fisiologi sapi potong (jenis kelamin, berat badan awal, dan penambahan per hari), beberapa atribut dari formulasi ransum (seperti nama formulasi dan tanggal penyusunan), dan nama-nama bahan penyusun ransum dari hasil formulasi kedua.
17. Tabel nutrisi dari masing-masing bahan penyusun hasil formulasi ransum kedua beserta kebutuhan nutrisi harian
18. Diagram pembagian berat untuk masing-masing bahan penyusun hasil formulasi kedua.
19. Tombol untuk mencetak hasil penyusunan ransum.
20. *Footer* berisi hak cipta, tahun pembuatan, dan nama pengembang aplikasi SunRan.

Perancangan halaman masuk

Gambar 5.30 menampilkan rancangan dari halaman masuk / login. Rancangan tersebut juga terdapat nomer penunjuk untuk masing-masing komponen yang ditampilkan.



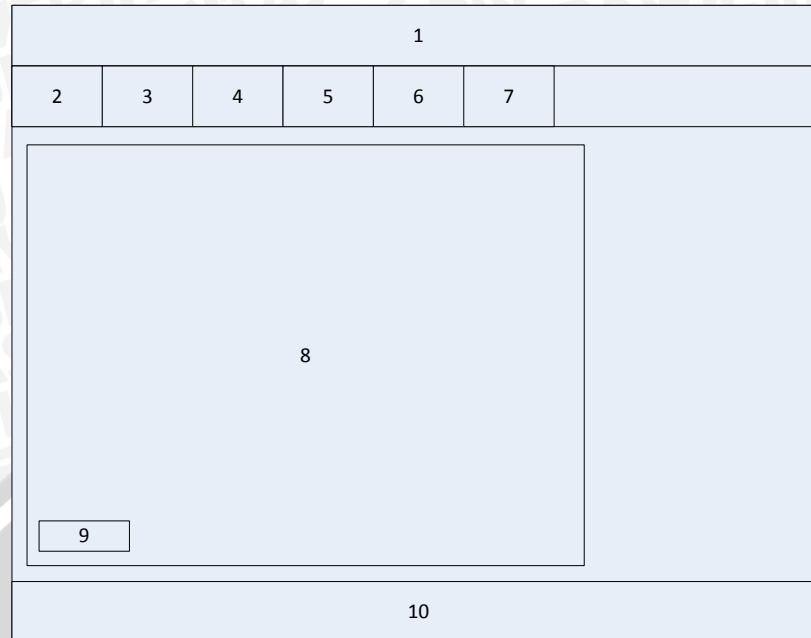
Gambar 5.30 Perancangan antarmuka halaman masuk

Halaman masuk menampilkan formulir untuk melakukan masuk / login ke aplikasi SunRan. Keterangan nomor-nomor petunjuk komponen pada Gambar 5.30 antara lain:

1. *Header* berisi judul sistem aplikasi.
2. Menu beranda yang akan mengarahkan ke halaman utama.
3. Menu nutrisi harian untuk memberikan informasi daftar nutrisi harian minimum.
4. Menu bahan penyusun untuk memberikan informasi daftar bahan penyusun ransum.
5. Menu formulasi ransum untuk melakukan penyusunan ransum
6. Menu masuk untuk *login* / masuk ke sistem.
7. Menu daftar untuk melakukan pendaftaran.
8. Formulir untuk mengisi *username* dan *password*.
9. Tombol untuk masuk / *login*.
10. *Footer* berisi hak cipta, tahun pembuatan, dan nama pengembang aplikasi SunRan.

Perancangan halaman daftar

Gambar 5.31 menampilkan rancangan dari halaman daftar. Rancangan tersebut juga terdapat nomer penunjuk untuk masing-masing komponen yang ditampilkan.



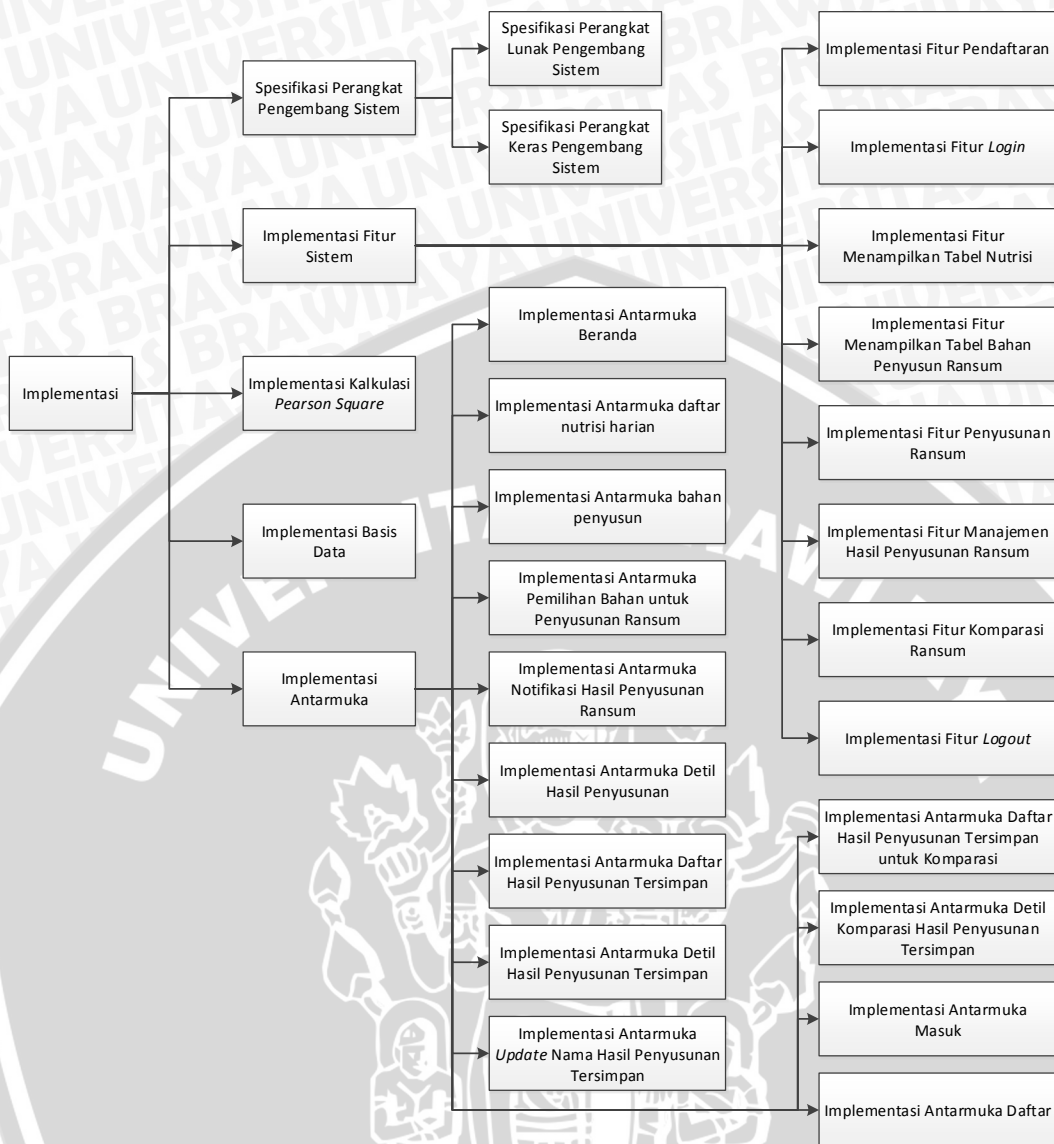
Gambar 5.31 Perancangan antarmuka halaman daftar

Halaman daftar menampilkan formulir untuk melakukan pendaftaran dengan mengisi data seperti *username*, *password*, nama, dan *email*. Keterangan nomor-nomor petunjuk komponen pada Gambar 5.31 antara lain:

1. *Header* berisi judul sistem aplikasi.
2. Menu beranda yang akan mengarahkan ke halaman utama.
3. Menu nutrisi harian untuk memberikan informasi daftar nutrisi harian minimum.
4. Menu bahan penyusun untuk memberikan informasi daftar bahan penyusun ransum.
5. Menu formulasi ransum untuk melakukan penyusunan ransum
6. Menu masuk untuk *login* / masuk ke sistem.
7. Menu daftar untuk melakukan pendaftaran.
8. Formulir untuk mengisi data *username*, *password*, nama, dan surel / *email*.
9. Tombol untuk melakukan pendaftaran.
10. *Footer* berisi hak cipta, tahun pembuatan, dan nama pengembang aplikasi SunRan.

5.2 Implementasi

Tahap implementasi ini berisi tentang penerapan / implementasi dari perancangan dan analisa pada bab-bab sebelumnya. Pembahasan pada bab ini terdiri dari spesifikasi perangkat pengembang sistem, implementasi fitur sistem, implementasi kalkulasi *Pearson Square*, implementasi basis data, dan implementasi antarmuka. Tahap pembahasan pada bab ini secara detail bisa dilihat pada gambar berikut.



Gambar 5.32 Diagram implementasi

SunRan yang merupakan aplikasi berbasis *web* dikembangkan dengan spesifikasi perangkat yang sudah ditetapkan oleh penulis. Pengembangan SunRan dilakukan dengan implementasi fitur, basis data, dan antarmuka seperti pada Gambar 5.32 di atas. Fitur yang diimplementasi seperti pendaftaran, *login*, menampilkan tabel nutrisi, dan penyusunan ransum. Fitur-fitur tersebut dikembangkan menggunakan bahasa pemrograman PHP, HTML, dan javascript pada *framework* Yii disertai dengan komponen-komponen lainnya.

5.2.1 Spesifikasi perangkat pengembang sistem

Pengembangan sistem SunRan mengacu pada hasil analisa kebutuhan pada bab empat. Pengembangan tersebut memerlukan perangkat yang mampu untuk dilakukan proses implementasi. Perangkat yang digunakan untuk implementasi SunRan memiliki spesifikasi yang dijelaskan pada sub-sub bab berikut.

5.2.1.1 Spesifikasi perangkat lunak pengembang sistem

Perangkat yang dipergunakan dalam implementasi menggunakan spesifikasi perangkat lunak / *software* yang dijelaskan pada Tabel 5.1 berikut.

Tabel 5.1 Spesifikasi perangkat lunak pengembang sistem

Nama komponen	Spesifikasi
Sistem Operasi	Windows 7 Ultimate 64-bit
<i>Web Browser</i>	Google Chrome versi 47.0.2526.73 m Mozilla Firefox versi 38.0.5 Opera versi 33.0.1990.115
<i>Tool</i>	Netbeans IDE 8.0.2. Aplikasi ini dipakai untuk menulis kode pemrograman untuk mengembangkan SunRan. XAMPP versi 3.2.1. Aplikasi ini berfungsi untuk membuat <i>web server</i> lokal pada perangkat agar sistem SunRan bisa dijalankan.

5.2.1.2 Spesifikasi perangkat keras pengembang sistem

Perangkat yang digunakan dalam implementasi memiliki spesifikasi perangkat keras / *hardware* seperti pada tabel 5.2 berikut.

Tabel 5.2 Spesifikasi perangkat keras pengembang sistem

Nama komponen	Spesifikasi
Prosesor	Intel(R) Core (TM) i5-2430M 2,40GHz
Memory (RAM)	8 GB
Kapasitas <i>Hard Disk Drive</i>	1 TB
Kartu Grafis	NVIDIA GeForce GT 525M
Ukuran Monitor	14 inci, 1366 x 768 piksel

5.2.2 Implementasi fitur sistem

Implementasi fitur-fitur yang ada di dalam sistem didasari dengan kebutuhan yang telah diidentifikasi sebelumnya dan menggunakan perangkat lunak yang telah ditentukan. Fitur-fitur yang di implementasi antara lain pendaftaran diri, *login*, menampilkan tabel nutrisi, menampilkan tabel bahan penyusun ransum, penyusunan ransum, komparasi ransum, dan *logout*. Penjelasan fitur-fitur dalam sistem SunRan dan spesifikasi pengembangan ada pada sub-sub bab berikut.

5.2.2.1 Implementasi fitur pendaftaran

Fitur pendaftaran memanggil tampilan formulir pendaftaran dan juga memproses data pengguna untuk disimpan ke basis data. Gambar 5.33 berisi cuplikan kode dari fitur ini.

```

1. $model=new RegUser('create');
2. if(isset($_POST['RegUser']))
3. {
4.     $model->attributes=$_POST['RegUser'];
5.     if(RegUser::hashPassword($model-
6. >password)===RegUser::hashPassword($model->confirmPassword))
7.     {
8.         $model->password=        RegUser::hashPassword($model-
9. >password);
10.        $model->confirmPassword=
11. RegUser::hashPassword($model->confirmPassword);
12.        if($model->save())
13.        {
14.            $this->redirect(array('site/login'));
15.        }
16.        $this->render('create',array(
17.            'model'=>$model
18.        ));
19.    }
20. }
21. $this->render('create',array('model'=>$model,));

```

Gambar 5.33 Implementasi fitur pendaftaran

Penjelasan implementasi fitur pendaftaran dari Gambar 5.33 yaitu:

1. Baris 1 berfungsi untuk membuat objek baru dari entitas pengguna.
2. Baris 2 berfungsi untuk menjalankan baris 3-17 jika terdapat data yang dikirim dari formulir pendaftaran.
3. Baris 4 berfungsi untuk menangkap data dari formulir pendaftaran.
4. Baris 5-6 berfungsi untuk konfirmasi kesamaan antara *password* dan konfirmasi *password*.
5. Baris 8-9 berfungsi untuk mengubah data *password* menjadi data enkripsi MD5.
6. Baris 10-11 berfungsi untuk mengubah data *confirmpassword* menjadi data enkripsi MD5
7. Baris 12 berfungsi untuk menyimpan data pengguna dan menentukan apakah proses pendaftaran berhasil atau tidak.
8. Baris 14 berfungsi untuk mengarahkan halaman ke halaman *login* jika proses pendaftaran berhasil.
9. Baris 16-18 berfungsi untuk menampilkan halaman pendaftaran jika proses penyimpanan gagal dengan mengirim ulang data yang gagal tersebut.
10. Baris 21 berfungsi untuk menampilkan halaman pendaftaran.

5.2.2.2 Implementasi fitur login

Fitur *login* / masuk ke sistem melakukan proses *login* terhadap pengguna yang tidak masuk ke sistem (*guest user*) menjadi pengguna yang masuk ke sistem (*logged in user*). Gambar 5.34 berisi cuplikan kode dari fitur ini.

```

1. $model=new LoginForm;
2. if(isset($_POST['ajax']) && $_POST['ajax']==='login-form')
3. {
4.     echo CActiveForm::validate($model);
5.     Yii::app()->end();
6. }
7. if(isset($_POST['LoginForm']))
8. {
9.     $model->attributes=$_POST['LoginForm'];
10.    if($model->validate() && $model->login())
11.        $this->redirect(Yii::app()->user->returnUrl);
12. }
13. $this->render('login',array('model'=>$model));

```

Gambar 5.34 Implementasi fitur *login*

Penjelasan implementasi fitur *login* dari Gambar 5.34 yaitu:

1. Baris 1 berfungsi untuk membuat objek formulir *login* baru.
2. Baris 2 berfungsi untuk menentukan ada tidaknya validasi ajax.
3. Baris 4-5 berfungsi untuk validasi data pada formulir *login*.
4. Baris 7 berfungsi untuk menentukan apakah ada data yang dikirim dari formulir *login*.
5. Baris 9 berfungsi untuk mendapatkan data yang dikirim dari formulir *login*.
6. Baris 10 berfungsi untuk melakukan verifikasi dan proses login terhadap data yang dikirim dari formulir *login*.
7. Baris 11 berfungsi untuk mengarahkan halaman ke halaman beranda jika proses *login* berhasil.
8. Baris 13 berfungsi untuk menampilkan halaman formulir *login* serta mengirim ulang data yang dikirim (jika ada).

5.2.2.3 Implementasi fitur menampilkan tabel nutrisi

Fitur menampilkan tabel nutrisi melakukan proses penjabaran secara detail kebutuhan nutrisi harian sapi potong berdasarkan jenis kelamin, berat badan, dan pertambahan per hari dalam bentuk tabel. Gambar 5.35 berisi cuplikan kode dari fitur ini.

```

1. $dataProvider=new CActiveDataProvider('Nutrient',
2. array('pagination'=>false));
3. $this->render('index',
4. array('dataProvider'=>$dataProvider));

```

Gambar 5.35 Implementasi fitur menampilkan tabel nutrisi

Penjelasan implementasi fitur menampilkan tabel nutrisi dari Gambar 5.35 yaitu:

1. Baris 1-2 berfungsi untuk mendapatkan data nutrisi dari basis data secara keseluruhan ke variabel.
2. Baris 3-4 berfungsi untuk menampilkan data yang berada di variabel ke tampilan *index*.

5.2.2.4 Implementasi fitur menampilkan tabel bahan penyusun ransum

Fitur menampilkan tabel bahan penyusun ransum melakukan proses penjabaran secara detail hingga menampilkan kandungan nutrisi dari bahan dalam bentuk tabel. Gambar 5.36 berisi cuplikan kode dari fitur ini.

```

1. $dataProvider=new CActiveDataProvider('Composition', array(
2.     'sort'=>array(
3.         'defaultOrder'=>'comp_name ASC'),
4.     'pagination'=>false));
5.     $this->render('index',array(
6.         'dataProvider'=>$dataProvider));
    
```

Gambar 5.36 Implementasi fitur menampilkan tabel bahan penyusun ransum

Penjelasan implementasi fitur menampilkan tabel bahan penyusun ransum dari Gambar 5.36 yaitu:

1. Baris 1-4 berfungsi untuk mendapatkan data bahan penyusun dari basis data secara keseluruhan dan diurutkan berdasarkan nama bahan serta menyimpannya ke variabel.
2. Baris 5-6 berfungsi untuk menampilkan data yang berada di variabel ke tampilan *index*.

5.2.2.5 Implementasi fitur penyusunan ransum

Fitur penyusunan ransum melakukan proses formulasi ransum berdasarkan data berat badan, jenis kelamin, penambahan per hari, dan bahan-bahan yang dipilih oleh pengguna. Implementasi ini dibagi dalam 4 bagian, yaitu memilih bahan, verifikasi data yang dipilih, kalkulasi *Pearson Square*, dan menyimpan hasil penyusunan ransum.

```

1. $hasilQuery1= Composition::model()->daftarSemuaHijauan();
2. $hasilQuery2= Composition::model()->daftarSemuaKonsentrat();
3. $this->render('pilihBahan',array(
4.     'hasilQuery1'=>$hasilQuery1,
5.     'hasilQuery2'=>$hasilQuery2));
    
```

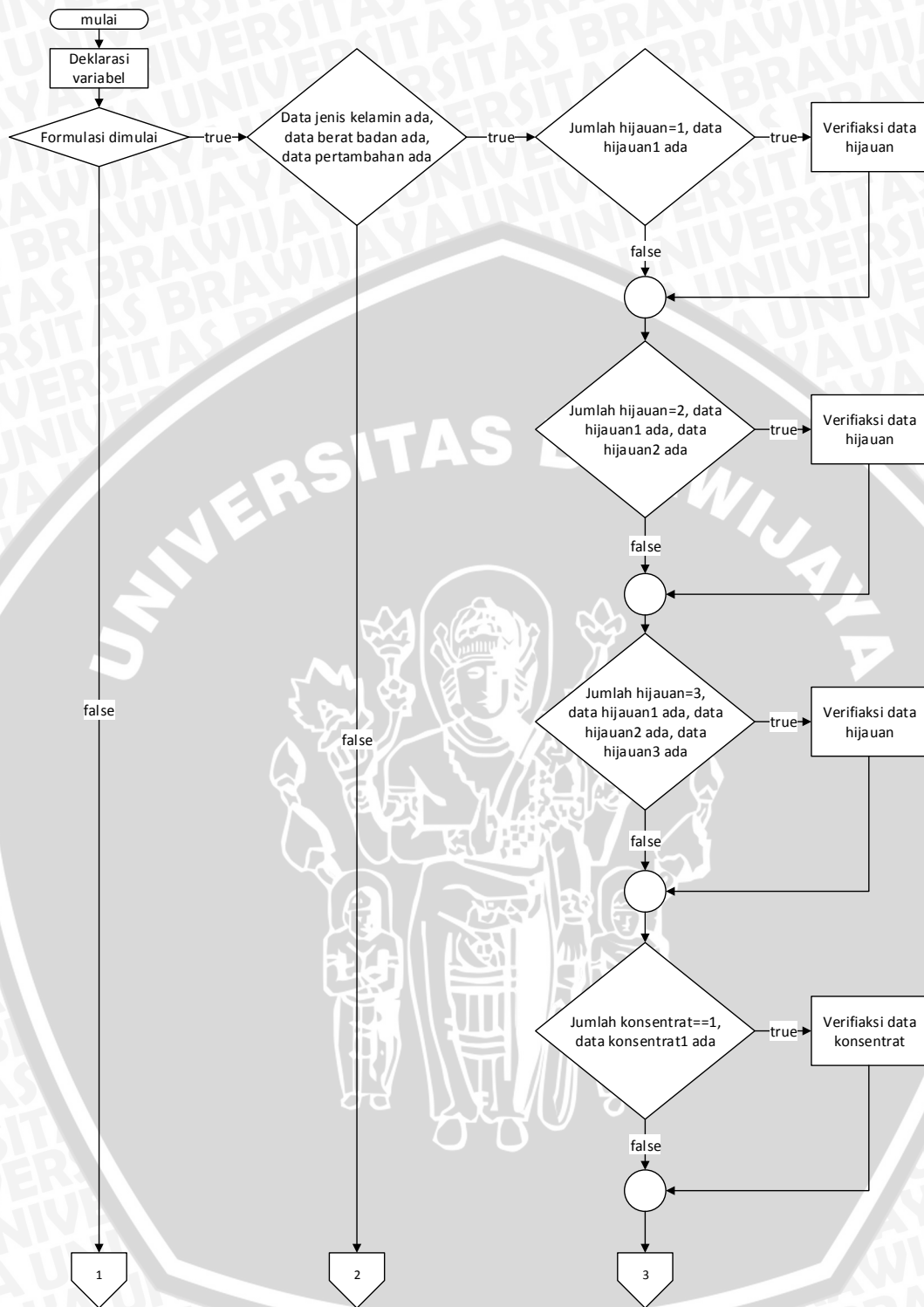
Gambar 5.37 Implementasi fitur penyusunan ransum bagian pemilihan bahan

Penjelasan implementasi fitur penyusunan ransum bagian pemilihan bahan dari Gambar 5.37 yaitu:

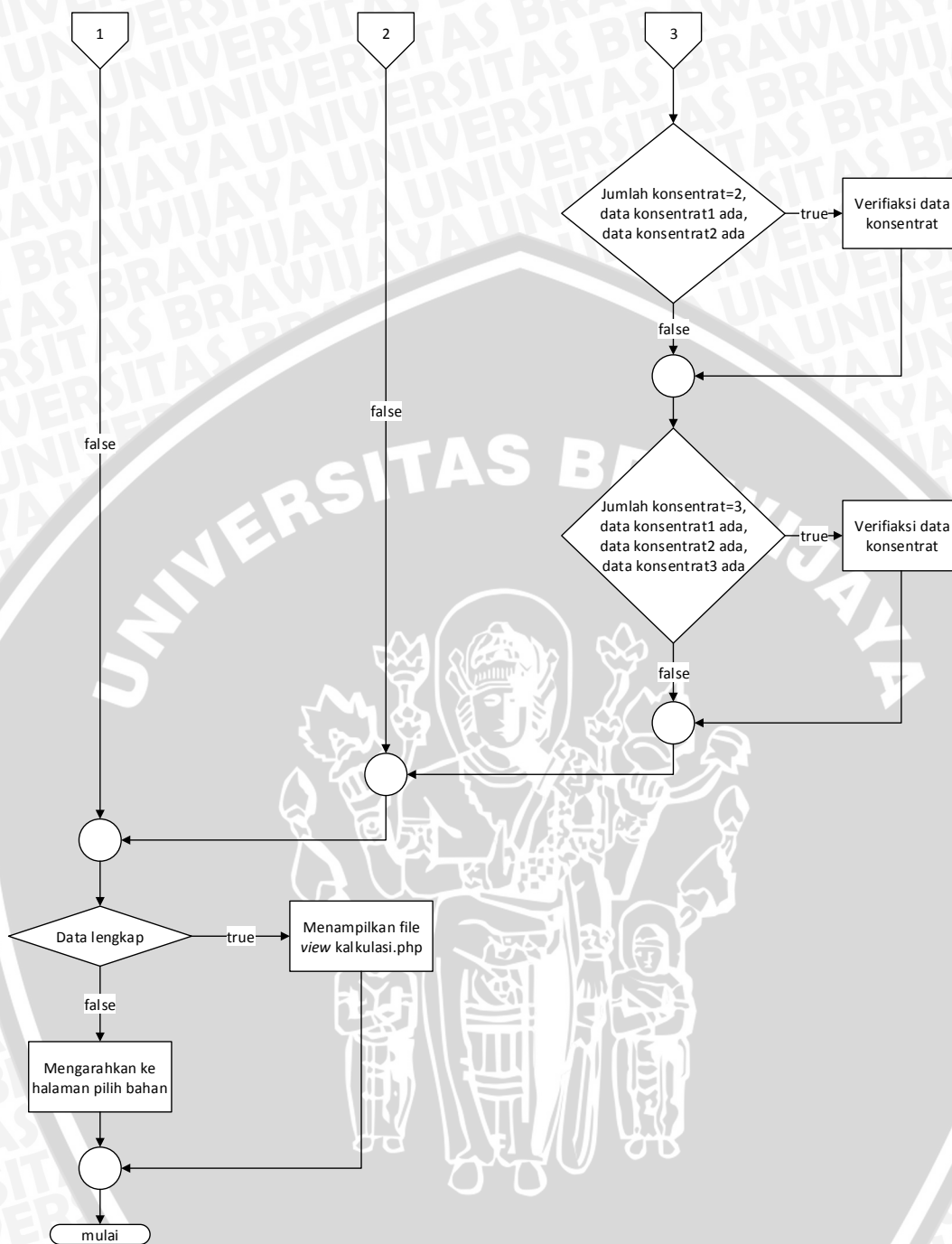
1. Baris 1 berfungsi untuk mendapatkan data bahan hijauan dari basis data serta menyimpannya di variabel.
2. Baris 2-3 berfungsi untuk mendapatkan data bahan konsentrat dari basis data serta menyimpannya di variabel.
3. Baris 4-6 berfungsi untuk menampilkan data dari kedua variabel.

Gambar 5.38 dan Gambar 5.39 Berikut menunjukkan alur dalam verifikasi data yang dipilih dan menyimpan hasil penyusunan ransum.





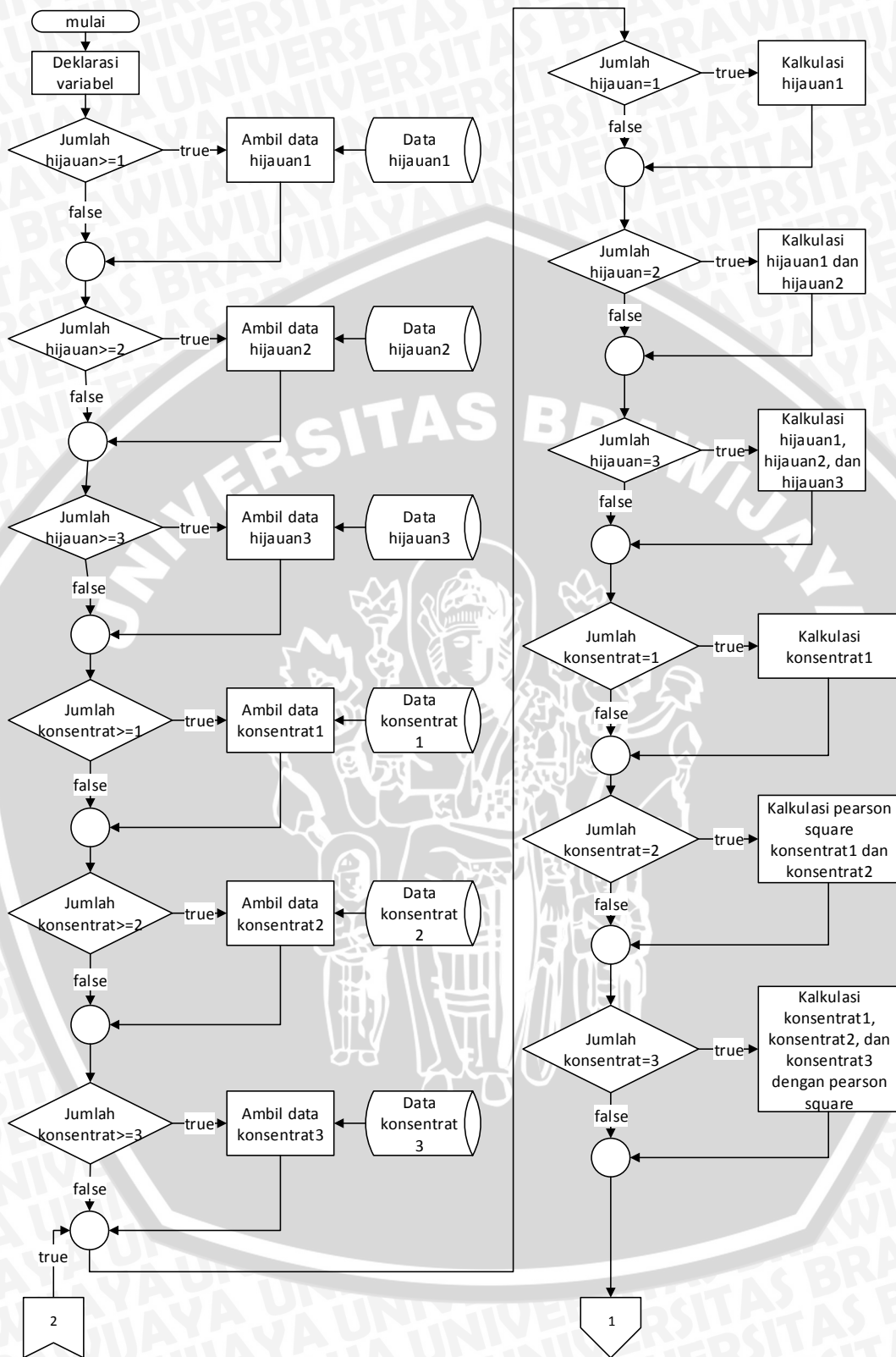
Gambar 5.38 Flowchart tahap verifikasi data yang dipilih (bagian 1)



Gambar 5.39 Flowchart tahap verifikasi data yang dipilih (bagian 2)

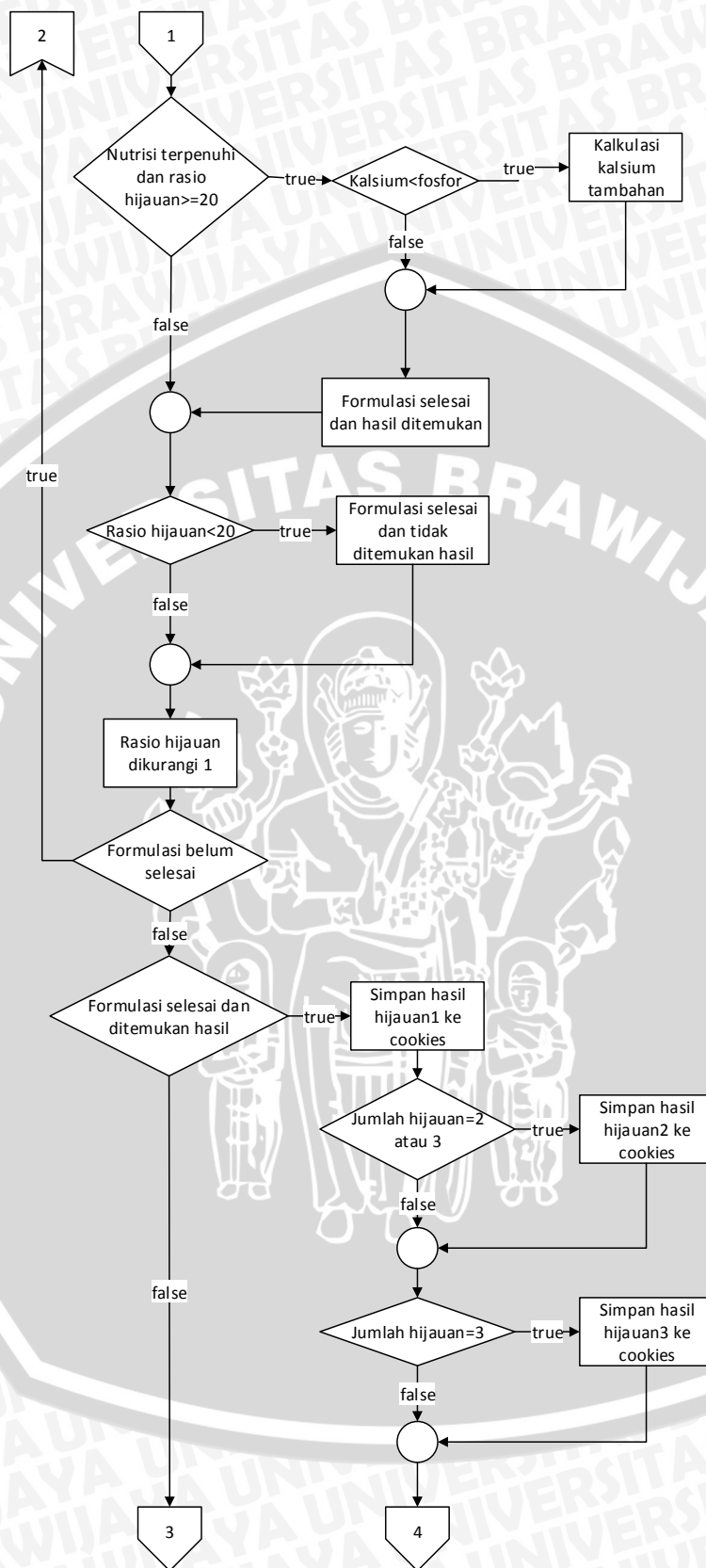
Kedua flowchart di atas diimplementasikan di controller *FormulasiRansumController*. Flowchart di atas menyimbolkan bahan hijauan pertama sebagai "hijauan1", bahan hijauan kedua sebagai "hijauan2", bahan hijauan ketiga sebagai "hijauan3", bahan konsentrat pertama sebagai "konsentrat1", bahan konsentrat kedua sebagai "konsentrat2", dan bahan konsentrat ketiga sebagai "konsentrat3". Penjelasan kedua flowchart di atas yaitu:

1. Sistem melakukan deklarasi variabel penanda dimulainya formulasi / penyusunan ransum.
2. Jika data jenis kelamin, berat badan, dan penambahan berat badan per hari ada pada *cookie* maka sistem menjalankan poin 3 (berikutnya). Jika tiga data tersebut tidak lengkap maka sistem menjalankan poin 9.
3. Jika jumlah hijauan sebanyak 1 dan data hijauan1 ada di *cookie*, sistem memverifikasi data hijauan. Jika dua syarat tersebut tidak terpenuhi maka sistem menjalankan poin 4 (berikutnya).
4. Jika jumlah hijauan sebanyak 2, data hijauan1, dan data hijauan2 ada di *cookie*, sistem memverifikasi data hijauan. Jika tiga syarat tersebut tidak terpenuhi maka sistem menjalankan poin 5 (berikutnya).
5. Jika jumlah hijauan sebanyak 3, data hijauan1, data hijauan2, dan data hijauan3 ada di *cookie*, sistem memverifikasi data hijauan. Jika empat syarat tersebut tidak terpenuhi maka sistem menjalankan poin 6 (berikutnya).
6. Jika jumlah konsentrat sebanyak 1 dan data konsentrat1 ada di *cookie*, sistem memverifikasi data konsentrat. Jika dua syarat tersebut tidak terpenuhi maka sistem menjalankan poin 7 (berikutnya).
7. Jika jumlah konsentrat sebanyak 2, data konsentrat1, dan data konsentrat2 ada di *cookie*, sistem memverifikasi data konsentrat. Jika tiga syarat tersebut tidak terpenuhi maka sistem menjalankan poin 8 (berikutnya).
8. Jika jumlah konsentrat sebanyak 3, data konsentrat1, data konsentrat2, dan data konsentrat3 ada di *cookie*, sistem memverifikasi data konsentrat. Jika dua syarat tersebut tidak terpenuhi maka sistem menjalankan poin 9 (berikutnya).
9. Jika data untuk penyusunan ransum telah lengkap maka sistem akan menampilkan *file* kalkulasi.php dan alur selesai. Jika data tidak lengkap maka sistem mengarahkan ke halaman pemilihan bahan dan alur selesai.

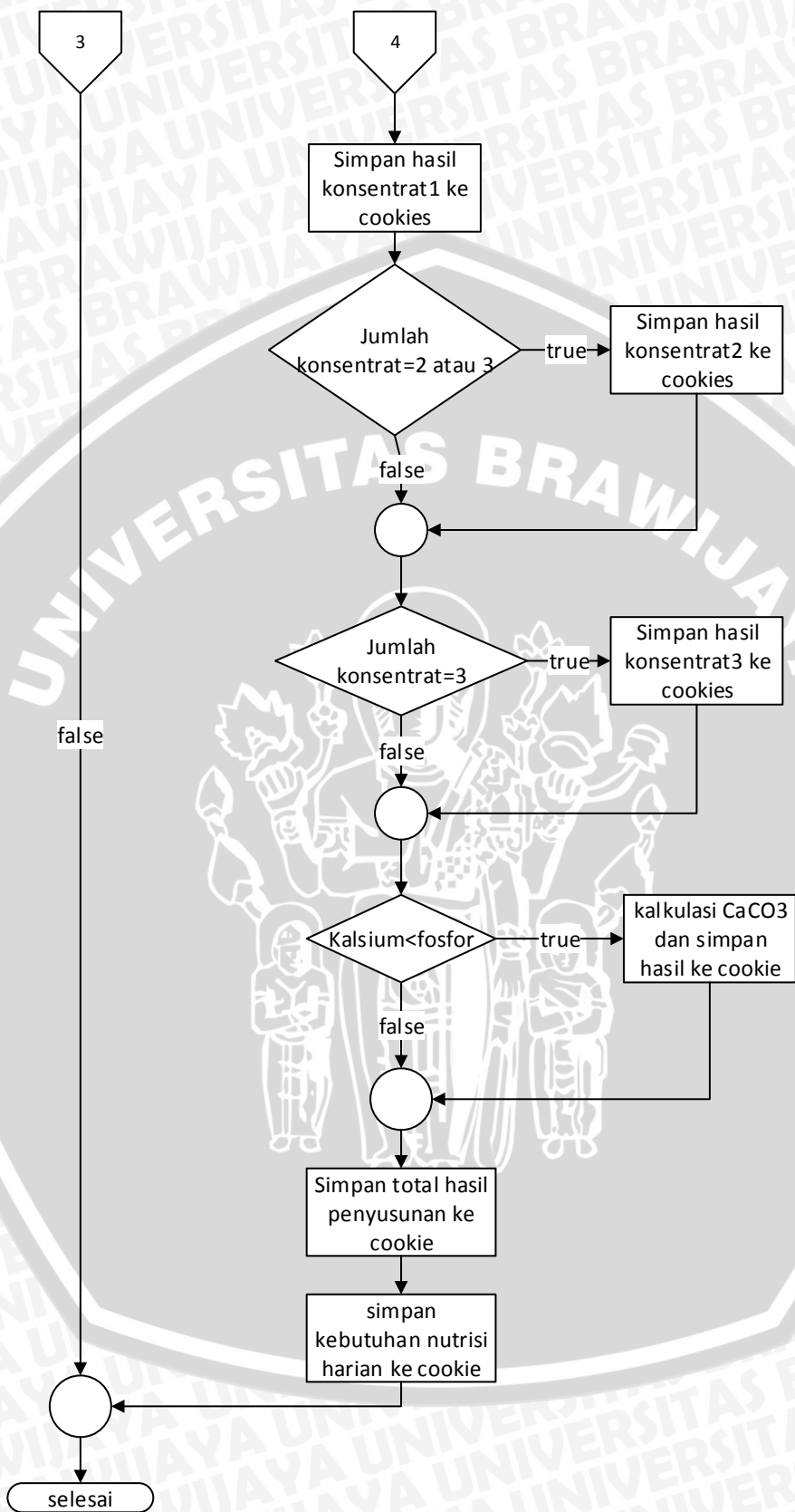


Gambar 5.40 Flowchart tahap kalkulasi *Pearson Square* (bagian 1)





Gambar 5.41 Flowchart tahap kalkulasi Pearson Square (bagian 2)



Gambar 5.42 Flowchart tahap kalkulasi Pearson Square (bagian 3)

Kedua *flowchart* di atas dibuat berdasarkan hasil wawancara dengan pihak pakar (Lampiran A). *Flowchart* di atas menyimbolkan bahan hijauan pertama sebagai “hijauan1”, bahan hijauan kedua sebagai “hijauan2”, bahan hijauan ketiga sebagai “hijauan3”, bahan konsentrat pertama sebagai “konsentrat1”, bahan konsentrat kedua sebagai “konsentrat2”, dan bahan konsentrat ketiga sebagai “konsentrat3”. Penjelasan kedua *flowchart* di atas yaitu:

1. Algoritma ini diawali dengan deklarasi variabel-variabel seperti rasio hijauan, jumlah hijauan, jumlah konsentrat, rasio konsentrat, jenis kelamin, berat badan, kebutuhan nutrisi harian, dan lain-lain.
2. Sistem mengambil data nutrisi bahan hijauan1 (dari basis data) jika jumlah hijauan lebih dari atau sama dengan 1.
3. Sistem mengambil data nutrisi bahan hijauan2 (dari basis data) jika jumlah hijauan lebih dari atau sama dengan 2. Jika jumlah hijauan kurang dari 2 maka sistem menjalankan poin 4 (berikutnya).
4. Sistem mengambil data nutrisi bahan hijauan3 (dari basis data) jika jumlah hijauan lebih dari atau sama dengan 3. Jika jumlah hijauan kurang dari 3 maka sistem menjalankan poin 5 (berikutnya).
5. Sistem mengambil data nutrisi bahan konsentrat1 (dari basis data) jika jumlah konsentrat lebih dari atau sama dengan 1.
6. Sistem mengambil data nutrisi bahan konsentrat2 (dari basis data) jika jumlah konsentrat lebih dari atau sama dengan 2. Jika jumlah konsentrat kurang dari 2 maka sistem menjalankan poin 7 (berikutnya).
7. Sistem mengambil data nutrisi bahan konsentrat2 (dari basis data) jika jumlah konsentrat lebih dari atau sama dengan 3. Jika jumlah konsentrat kurang dari 3 maka sistem menjalankan poin 8 (berikutnya).
8. Sistem menghitung hijauan1 jika jumlah hijauan sama dengan 1. Jika jumlah hijauan tidak sama dengan 1 maka sistem menjalankan poin 9 (berikutnya).
9. Sistem menghitung hijauan1 dan hijauan2 jika jumlah hijauan sama dengan 2. Jika jumlah hijauan tidak sama dengan 2 maka sistem menjalankan poin 10 (berikutnya).
10. Sistem menghitung hijauan1, hijauan2, dan hijauan3 jika jumlah hijauan sama dengan 3. Jika jumlah hijauan tidak sama dengan 3 maka sistem menjalankan poin 11 (berikutnya).
11. Sistem menghitung konsentrat1 jika jumlah konsentrat sama dengan 1. Jika jumlah konsentrat tidak sama dengan 1 maka sistem menjalankan poin 12 (berikutnya).
12. Sistem menghitung konsentrat1 dan konsentrat2 (menggunakan metode *Pearson Square*) jika jumlah konsentrat sama dengan 2. Jika

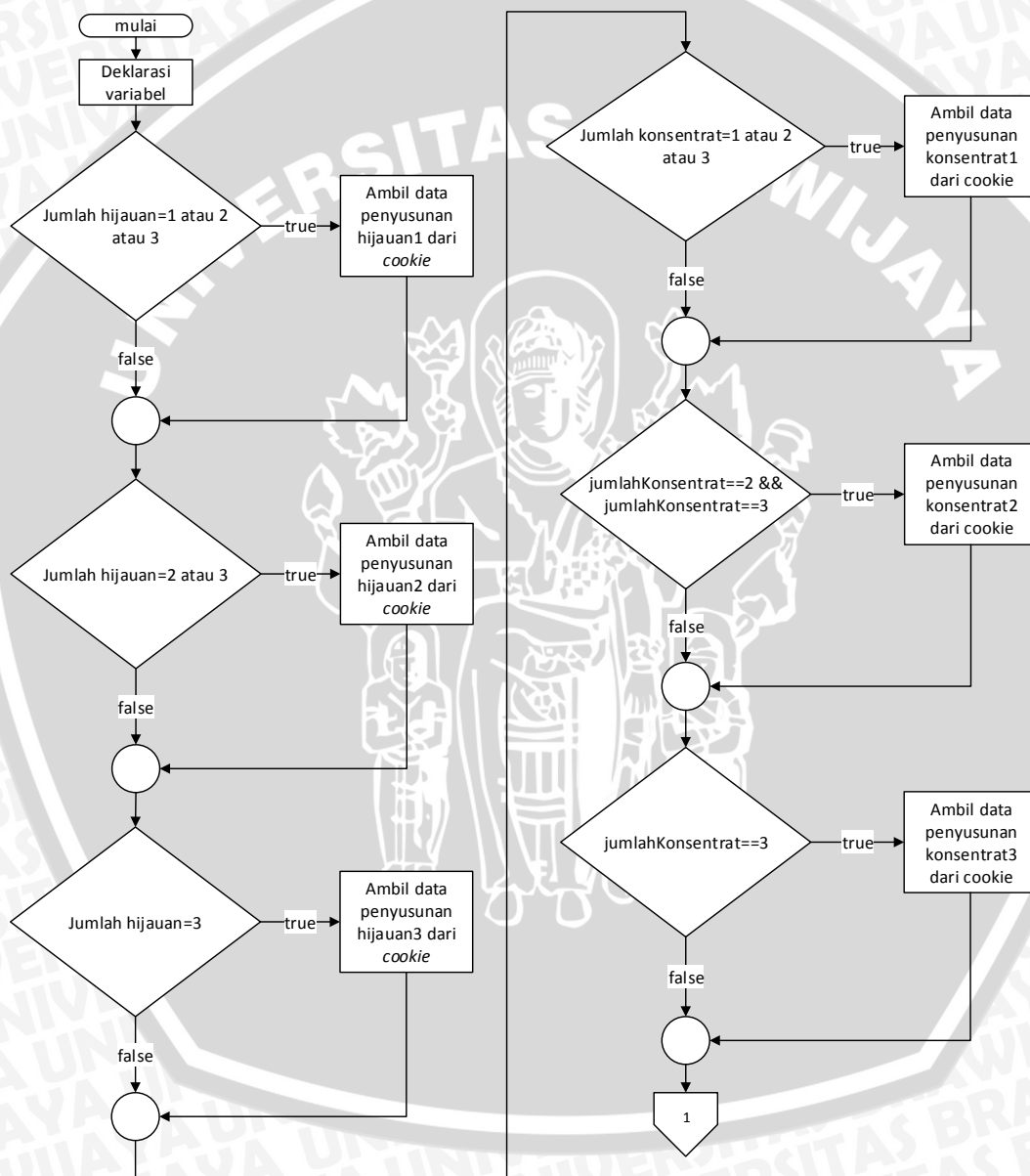
jumlah konsentrat tidak sama dengan 2 maka sistem menjalankan poin 13 (berikutnya).

13. Sistem menghitung konsentrat1, konsentrat2, dan konsentrat3 (konsentrat2 dan konsentrat3 dihitung menggunakan metode *Pearson Square*) jika jumlah konsentrat sama dengan 3. Jika jumlah konsentrat tidak sama dengan 3 maka sistem menjalankan poin 12 (berikutnya).
14. Jika nutrisi ransum yang disusun telah memenuhi kebutuhan harian dan rasio hijauan lebih dari 20% maka sistem menghitung kalsium tambahan (jika kalsium keseluruhan lebih sedikit daripada keseluruhan fosfor) dan memberi tanda ke variabel bahwa penyusunan ransum selesai dan ditemukan ransum ideal. Jika nutrisi ransum yang disusun tidak memenuhi kebutuhan harian atau rasio hijauan kurang dari 20% maka sistem menjalankan poin 15 (berikutnya).
15. Jika rasio hijauan kurang dari 20% maka sistem memberi tanda ke variabel bahwa penyusunan ransum selesai dan tidak ditemukan ransum ideal. Jika rasio hijauan lebih dari atau sama dengan 20% maka sistem menjalankan poin 16 (berikutnya).
16. Sistem mengurangi rasio hijauan sebanyak 1%
17. Jika penyusunan ransum belum selesai maka sistem menjalankan kembali poin 8. Jika penyusunan ransum selesai maka sistem menjalankan poin 18 (berikutnya).
18. Jika penyusunan ransum selesai dan ditemukan ransum ideal maka sistem menyimpan hasil perhitungan hijauan1 ke *cookies* dan dilanjutkan ke poin 19 (berikutnya). Jika penyusunan ransum selesai dan tidak ditemukan maka algoritma ini selesai.
19. Jika jumlah hijauan sama dengan 2 atau 3 maka sistem menyimpan hasil perhitungan hijauan2 ke *cookies*. Jika jumlah hijuan tidak sama dengan 2 atau 3 maka sistem menjalankan poin 20 (berikutnya).
20. Jika jumlah hijauan sama dengan 3 maka sistem menyimpan hasil perhitungan hijauan3 ke *cookies*. Jika jumlah hijuan tidak sama dengan 3 maka sistem menjalankan poin 21 (berikutnya).
21. Sistem menyimpan hasil perhitungan konsentrat1 ke *cookies* dan dilanjutkan ke poin 22 (selanjutnya).
22. Jika jumlah konsentrat sama dengan 2 atau 3 maka sistem menyimpan hasil perhitungan konsentrat2 ke *cookies*. Jika jumlah konsentrat tidak sama dengan 2 atau 3 maka sistem menjalankan poin 23 (berikutnya).
23. Jika jumlah konsentrat sama dengan 3 maka sistem menyimpan hasil perhitungan konsentrat3 ke *cookies*. Jika jumlah konsentrat tidak sama dengan 3 maka sistem menjalankan poin 24 (berikutnya).

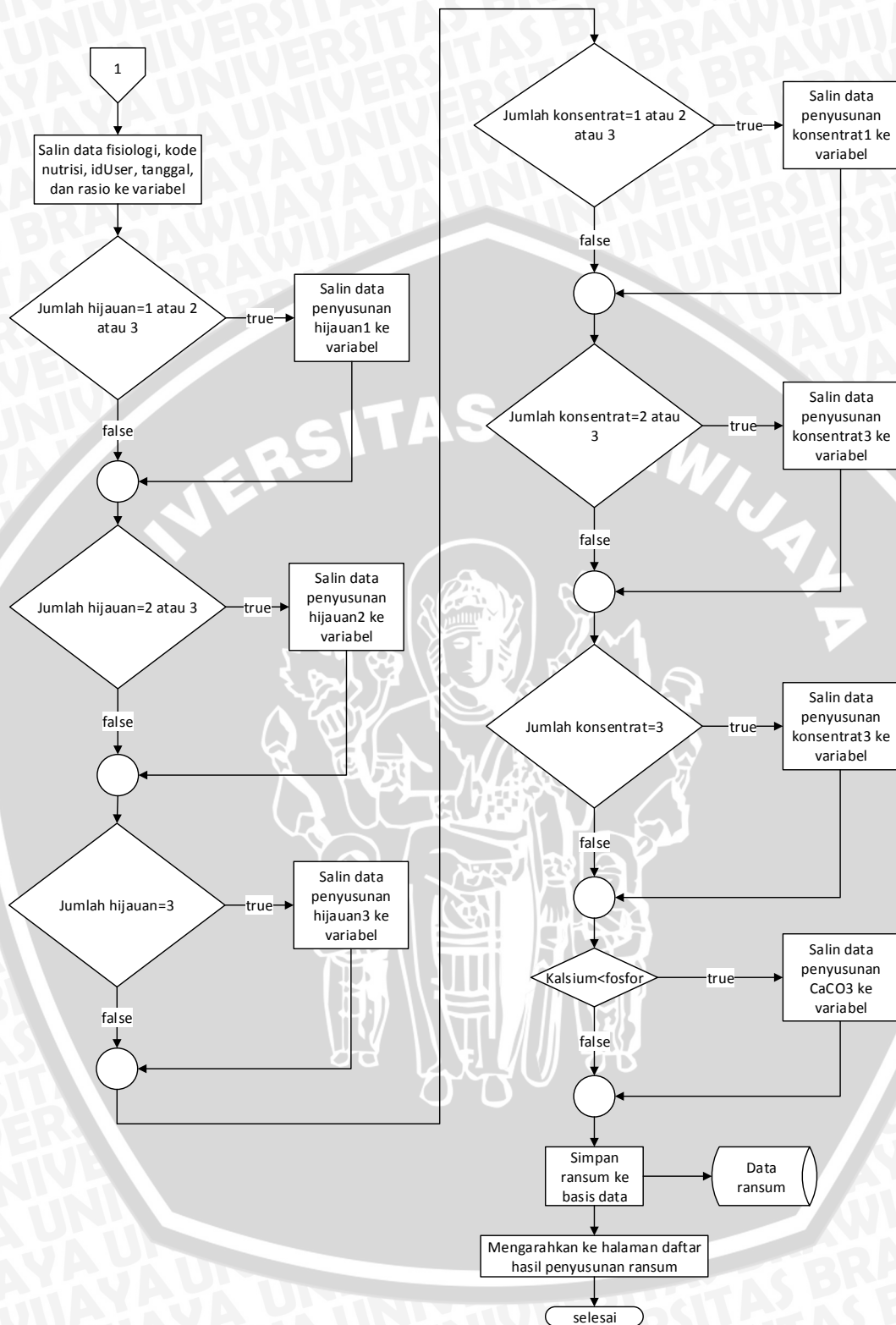
24. Jika jumlah keseluruhan kalsium lebih sedikit daripada keseluruhan fosfor maka sistem menghitung CaCO_3 yang diperlukan sebagai bahan rekomendasi, menyimpan hasil perhitungan tersebut ke *cookies*, dan menjalankan poin 25 (berikutnya).

25. Sistem menyimpan total hasil penyusunan ransum ideal dan kebutuhan nutrisi harian ke *cookie*.

Penjelasan lebih detail mengenai implementasi *Pearson Square* dapat dilihat sub bab 5.2.3.



Gambar 5.43 Flowchart tahap menyimpan hasil penyusunan ransum (bagian 1)



Gambar 5.44 Flowchart tahap menyimpan hasil penyusunan ransum (bagian 2)

Kedua *flowchart* di atas diimplementasikan di *controller* *FormulasiRansumController*. *Flowchart* di atas menyimbolkan bahan hijauan pertama sebagai “hijauan1”, bahan hijauan kedua sebagai “hijauan2”, bahan



hijauan ketiga sebagai “hijauan3”, bahan konsentrat pertama sebagai “konsentrat1”, bahan konsentrat kedua sebagai “konsentrat2”, dan bahan konsentrat ketiga sebagai “konsentrat3”. Penjelasan kedua *flowchart* di atas yaitu:

1. Sistem melakukan deklarasi variabel-variabel seperti jumlah hijauan, jumlah konsentrat, jenis kelamin, berat badan, dan lain-lain.
2. Jika jumlah hijauan sebanyak 1, 2, atau 3 maka sistem akan mengambil data penyusunan hijauan1 dari *cookie*.
3. Jika jumlah hijauan sebanyak 2 atau 3 maka sistem akan mengambil data penyusunan hijauan2 dari *cookie*. Jika jumlah hijauan tidak sebanyak 2 atau 3 maka sistem menjalankan poin 4 (berikutnya).
4. Jika jumlah hijauan sebanyak 3 maka sistem akan mengambil data penyusunan hijauan3 dari *cookie*. Jika jumlah hijauan tidak sebanyak 3 maka sistem menjalankan poin 5 (berikutnya).
5. Jika jumlah konsentrat sebanyak 1, 2, atau 3 maka sistem akan mengambil data penyusunan konsentrat1 dari *cookie*.
6. Jika jumlah konsentrat sebanyak 2 atau 3 maka sistem akan mengambil data penyusunan konsentrat2 dari *cookie*. Jika jumlah hijauan tidak sebanyak 2 atau 3 maka sistem menjalankan poin 7 (berikutnya).
7. Jika jumlah konsentrat sebanyak 3 maka sistem akan mengambil data penyusunan konsentrat3 dari *cookie*. Jika jumlah hijauan tidak sebanyak 3 maka sistem menjalankan poin 8 (berikutnya).
8. Sistem menyalin data fisiologi, tanggal penyusunan, rasio hijauan, dan lain-lain ke variabel.
9. Jika jumlah hijauan sebanyak 1, 2, atau 3 maka sistem akan menyalin data penyusunan hijauan1 ke variabel.
10. Jika jumlah hijauan sebanyak 2 atau 3 maka sistem akan menyalin data penyusunan hijauan2 ke variabel. Jika jumlah hijauan tidak sebanyak 2 atau 3 maka sistem menjalankan poin 11 (berikutnya).
11. Jika jumlah hijauan sebanyak 3 maka sistem akan menyalin data penyusunan hijauan3 ke variabel. Jika jumlah hijauan tidak sebanyak 3 maka sistem menjalankan poin 12 (berikutnya).
12. Jika jumlah konsentrat sebanyak 1, 2, atau 3 maka sistem akan menyalin data penyusunan konsentrat1 ke variabel.
13. Jika jumlah konsentrat sebanyak 2 atau 3 maka sistem akan menyalin data penyusunan konsentrat2 ke variabel. Jika jumlah hijauan tidak sebanyak 2 atau 3 maka sistem menjalankan poin 14 (berikutnya).
14. Jika jumlah konsentrat sebanyak 3 maka sistem akan menyalin data penyusunan konsentrat3 ke variabel. Jika jumlah hijauan tidak sebanyak 3 maka sistem menjalankan poin 15 (berikutnya).

15. Jika jumlah kalsium keseluruhan lebih sedikit daripada fosfor maka sistem mengambil data penyusunan CaCO₃ dari *cookie* ke variabel. Jika kalsium tidak lebih sedikit daripada fosfor maka sistem menjalankan poin 16 (berikutnya).
16. Sistem menyimpan hasil penyusunan ransum (yang telah disalin di variabel) ke basis data.
17. Mengarahkan halaman ke daftar hasil penyusunan ransum yang tersimpan di sistem.

Pembahasan lebih lengkap tentang implementasi tahap kalkulasi *Pearson Square* berada di sub bab 5.2.3.

5.2.2.6 Implementasi fitur manajemen hasil penyusunan ransum

Fitur manajemen hasil penyusunan ransum melakukan proses lihat detail, mengubah (*update*), dan menghapus hasil penyusunan ransum. Gambar 5.45 hingga Gambar 5.48 berisi cuplikan kode dari fitur ini.

```
1. $this->render('indexByUser',array('model'=>$this-
2. >loadModelByUser($id));
```

Gambar 5.45 Implementasi fitur manajemen hasil penyusunan ransum bagian menampilkan daftar hasil penyusunan ransum

Penjelasan implementasi fitur manajemen hasil penyusunan ransum bagian menampilkan daftar hasil penyusunan ransum dari Gambar 5.45 yaitu pada baris 1-2 berfungsi untuk menampilkan data berupa daftar hasil yang telah disusun dan disimpan sebelumnya pada tampilan *index*.

```
1. $this->render('view',array('model'=>$this->loadModel($id));
```

Gambar 5.46 Implementasi fitur manajemen hasil penyusunan ransum bagian melihat detail hasil penyusunan ransum

Penjelasan implementasi fitur manajemen hasil penyusunan ransum bagian melihat detail hasil penyusunan ransum dari Gambar 5.46 yaitu pada baris 1 berfungsi untuk menampilkan secara detail hasil penyusunan ransum pada tampilan *view*.

```
1. $model=$this->loadModel($id);
2. if(isset($_POST['ResultArrangement']))
3. {
4.     $model->attributes=$_POST['ResultArrangement'];
5.     if($model->save())
6.         $this->redirect(array('view','id'=>$model-
7. >id_result));
8. }
9. $this->render('update',array('model'=>$model));
```

Gambar 5.47 Implementasi fitur manajemen hasil penyusunan ransum bagian update hasil penyusunan ransum

Penjelasan implementasi fitur manajemen hasil penyusunan ransum bagian *update* hasil penyusunan ransum dari Gambar 5.47 yaitu:

1. Baris 1 berfungsi untuk mendapatkan data hasil penyusunan ransum serta menyimpannya di variable.

2. Baris 2-8 berfungsi untuk mendapatkan data *update* hasil penyusunan ransum dari formulir *update* dan melakukan proses *update* hasil penyusunan ransum di basis data. Jika proses *update* berhasil maka halaman diarahkan ke detail hasil penyusunan ransum. Baris-baris ini dijalankan jika ada data untuk melakukan *update*.
3. Baris 9 berfungsi untuk menampilkan formulir *update* hasil penyusunan ransum.

1.	<code>\$this->loadModel(\$id)->delete();</code>	
2.	<code>if(!isset(\$_GET['ajax']))</code>	
3.	<code> \$this->redirect(isset(\$_POST['returnUrl']))</code>	?
4.	<code> \$_POST['returnUrl'] : array('admin');</code>	

Gambar 5.48 Implementasi fitur manajemen hasil penyusunan ransum bagian hapus hasil penyusunan ransum

Penjelasan implementasi fitur manajemen hasil penyusunan ransum bagian hapus hasil penyusunan ransum dari Gambar 5.48 yaitu:

1. baris 1 berfungsi untuk melakukan penghapusan hasil penyusunan ransum.
2. Baris 2-4 berfungsi untuk mengarahkan ke halaman sebelumnya (daftar hasil penyusunan ransum) jika tidak mendapatkan permintaan menghapus hasil penyusunan ransum melalui *ajax*.

5.2.2.7 Implementasi fitur komparasi ransum

Fitur komparasi ransum melakukan proses perbandingan antara dua hasil penyusunan ransum secara detail dengan menampilkan hasil perbandingan menggunakan tabel dan diagram. Gambar 5.49 dan Gambar 5.50 berisi cuplikan kode dari fitur ini.

1.	<code>\$this->render('indexByUserCompare', array('model'=>\$this-</code>
2.	<code>>loadModelByUser(\$id));</code>

Gambar 5.49 Implementasi fitur komparasi ransum bagian menampilkan daftar hasil penyusunan ransum

Penjelasan implementasi fitur komparasi ransum bagian menampilkan daftar hasil penyusunan ransum dari Gambar 5.49 yaitu pada baris 1-2 berfungsi untuk menampilkan hasil penyusunan ransum yang telah dibuat dan disimpan sebelumnya pada tampilan *indexByUserCompare*.

1.	<code>\$this->render('viewCompare', array(</code>
2.	<code> 'hasil1'=>\$this->loadModel(\$id1),</code>
3.	<code> 'hasil2'=>\$this->loadModel(\$id2));</code>

Gambar 5.50 Implementasi fitur komparasi ransum bagian lihat detail perbandingan hasil penyusunan ransum

Penjelasan implementasi fitur komparasi ransum bagian lihat detail perbandingan hasil penyusunan ransum dari Gambar 5.50 yaitu pada baris 1-3 berfungsi untuk menampilkan detail perbandingan dua hasil penyusunan ransum pada tampilan *viewCompare*.



5.2.2.8 Implementasi fitur logout

Fitur *logout* melakukan proses mengubah kondisi / status dari pengguna dari pengguna yang masuk ke sistem (*logged in user*) menjadi pengguna yang tidak masuk ke sistem (*guest user*). Gambar 5.51 berisi cuplikan kode dari fitur ini.

```
1. Yii::app()->user->logout();  
2. $this->redirect(Yii::app()->homeUrl);
```

Gambar 5.51 Implementasi fitur logout

Penjelasan implementasi fitur *logout* dari Gambar 5.51 yaitu:

1. Baris 1 berfungsi untuk melakukan proses *logout* / keluar dari sistem.
2. Baris 2 berfungsi untuk mengarahkan halaman ke beranda.

5.2.3 Implementasi kalkulasi *Pearson Square*

Implementasi kalkulasi *Pearson Square* dilakukan pada pengembangan SunRan yang dipergunakan dalam proses penyusunan ransum. Implementasi kalkulasi ini menggunakan bahasa pemrograman PHP sesuai dengan Gambar 5.40 hingga Gambar 5.42 di atas. Implementasi dari kalkulasi *Pearson Square* dijadikan model pada aplikasi SunRan dengan nama *file pearsonSquare.php*.

5.2.4 Implementasi basis data

Pelaksanaan implementasi basis data saat pengembangan SunRan ini didasari oleh pemodelan yang ada di bab sebelumnya. Implementasi basis data dilaksanakan menggunakan MySQL dalam bentuk kode *query*. Hasil dari implementasi kode *query* digambarkan ke dragam konseptual *entity relationship* (relasi antar entitas) pada gambar berikut.



Gambar 5.52 Implementasi basis data

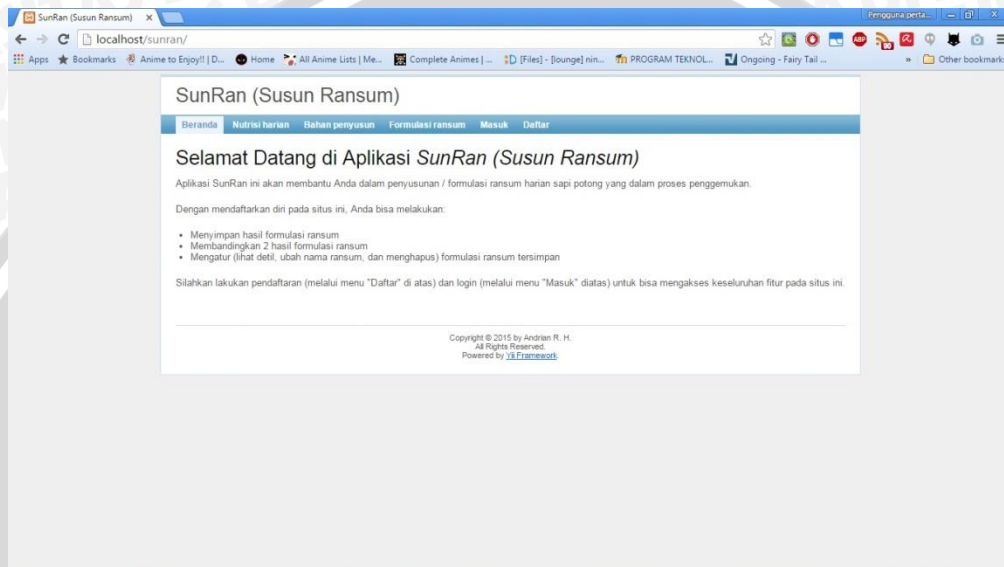
5.2.5 Implementasi antarmuka

Antarmuka / *interface* merupakan bagian dari aplikasi yang berinteraksi langsung dengan pengguna. Antarmuka dari sistem Sunran di implementasikan ke *file* yang termasuk *view* juga merupakan bagian dari sistem yang berintraksi langsung dengan pengguna. Implementasi dari antarmuka pada sistem Sunran didasari oleh rancangan antarmuka pada bab sebelumnya. Penjelasan dari

masing-masing implementasi antarmuka pada sistem SunRan akan disampaikan pada sub-sub bab berikut.

5.2.5.1 Implementasi antarmuka beranda

Implementasi halaman antarmuka beranda menampilkan menu, ucapan selamat, dan penjelasan singkat tentang fitur-fitur yang bisa diakses oleh pengguna. Implementasi dari halaman ini disesuaikan dengan rancangan pada bab sebelumnya. Gambar 5.53 berikut menampilkan implementasi halaman antarmuka beranda.



Gambar 5.53 Implementasi antarmuka beranda

5.2.5.2 Implementasi antarmuka daftar nutrisi harian

Implementasi halaman antarmuka daftar nutrisi harian menampilkan menu dan daftar nutrisi harian dalam bentuk tabel. Implementasi dari halaman ini disesuaikan dengan rancangan pada bab sebelumnya. Gambar 5.54 berikut menampilkan implementasi halaman antarmuka daftar nutrisi harian.

Berat (kg)	Kenaikan per Hari (kg/hari)	Jenis Kelamin	Berat Kering (kg)	TDN (kg)	Protein (gram)	Kalsium (gram)	Fosfor (gram)
100	0	Jantan	2.2	1	167	5	5
100	0.25	Jantan	2.6	1.3	303	10	7
100	0.5	Jantan	3	1.6	379	15	9
100	0.75	Jantan	3.2	1.9	448	20	11
100	1	Jantan	3.3	2.2	541	25	15
150	0	Jantan	3	1.4	231	6	6
150	0.25	Jantan	3.8	1.8	400	12	9
150	0.5	Jantan	4.2	2.2	474	16	10
150	0.75	Jantan	4.4	2.6	589	21	13
150	1	Jantan	4.5	3	607	27	16
200	0	Jantan	3.7	1.8	285	6	6
200	0.25	Jantan	4.5	2.2	470	11	9
200	0.5	Jantan	5.2	2.8	554	16	12
200	0.75	Jantan	5.4	3.2	622	21	15
200	1	Jantan	5.6	3.7	690	27	17
200	1.1	Jantan	5.6	3.9	714	30	18
250	0	Jantan	4.4	2	337	9	9
250	0.25	Jantan	5.3	2.6	534	12	10
250	0.5	Jantan	6.2	3.2	623	16	14
250	0.75	Jantan	6.4	3.8	693	21	17
250	1	Jantan	6.6	4.3	760	28	19
250	1.1	Jantan	6.6	4.6	782	30	20
300	0	Jantan	5	2.4	385	10	10

Gambar 5.54 Implementasi antarmuka daftar nutrisi harian

5.2.5.3 Implementasi antarmuka bahan penyusun

Implementasi halaman antarmuka bahan penyusun menampilkan menu dan tabel daftar bahan penyusun. Implementasi dari halaman ini disesuaikan dengan rancangan pada bab sebelumnya. Gambar 5.55 berikut menampilkan implementasi halaman antarmuka bahan penyusun.

No	Nama bahan	Tipe bahan	Bahan Kering (%)	Protein (%)	TDN (%)	Kalsium (%)	Fosfor (%)
1	Alfalfa	Hijauan	23	16	76	1.89	0.24
2	Ampas jeruk kering	Konsentrat	90	8.1	70	4.27	0.9
3	Ampas pati aren kering	Konsentrat	88	3.86	41	0	0
4	Ampas tahu	Konsentrat	14.6	30.3	77.9	0	0
5	Bagas (ampas tebu)	Konsentrat	90	1.9	47	0	0
6	Bekatul	Konsentrat	88	12.8	69.9	0.08	1.23
7	Bungkil kelapa	Konsentrat	88.6	21.3	78.7	0.17	0.62
8	Calopo	Hijauan	29	16	60	1.81	0.1
9	Centrosema	Hijauan	24.1	16.8	60.2	1.2	0.38
10	Cetro	Hijauan	25	22	61	1	0.36
11	Daging buah (pod) kakao kering	Konsentrat	92	10.06	47	0	0
12	Daun gliriside	Hijauan	27	19.1	76	0.67	0.19
13	Daun jagung	Hijauan	21	9.9	60	1.24	0.23
14	Daun kacang panjang	Hijauan	15	16.6	60.6	1.23	0.26
15	Daun kacang tanah	Hijauan	22.8	13.8	78.3	1.88	0.27
16	Daun kalandra	Hijauan	23	17	69	1.13	0.71
17	Daun kangkung	Hijauan	15	18.8	73	0.71	0.32
18	Daun kedelai	Hijauan	22.6	16.7	63.2	1.42	0.42
19	Daun kembang sepatu	Hijauan	19	17.8	67	1.67	0.52
20	Daun ketela pohon	Hijauan	26	20	71	0.99	0.56
21	Daun ketela rambat	Hijauan	24	11.3	74	1.79	0.24
22	Daun lamtoro	Hijauan	30	23.4	77	1.4	0.21
23	Daun mangla	Hijauan	16	12.5	72	1.47	0.18
24	Daun ...	Hijauan	22.2	16.6	73.6	0.62	0.18

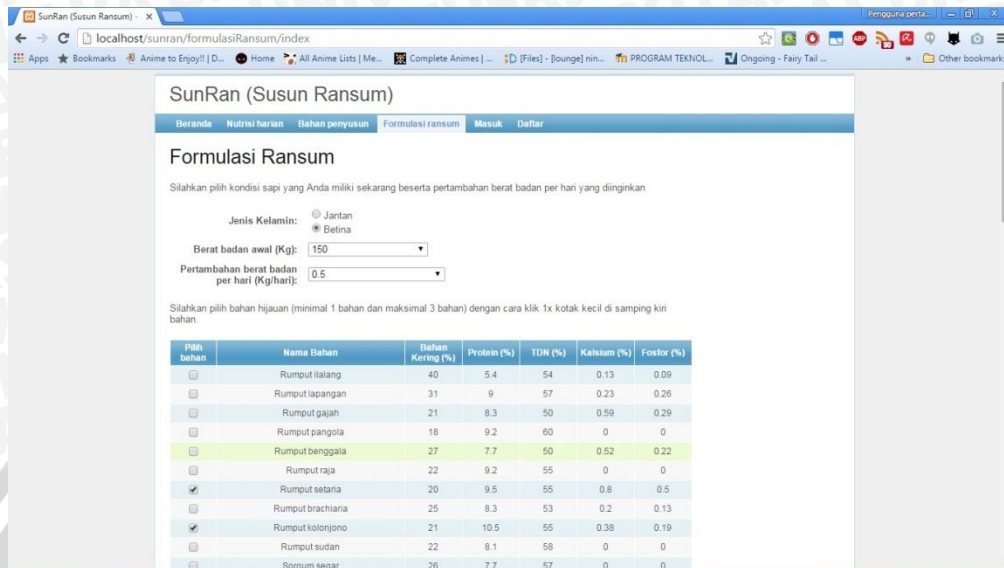
Gambar 5.55 Implementasi antarmuka bahan penyusun

5.2.5.4 Implementasi antarmuka pemilihan bahan untuk penyusunan ransum

Implementasi halaman antarmuka pemilihan bahan untuk penyusunan ransum menampilkan menu, opsi kondisi sapi potong (jenis kelamin, berat badan, dan pertambahan per hari), tabel pemilihan bahan (hijauan dan konsentrat), dan tombol untuk memulai proses penyusunan. Implementasi dari



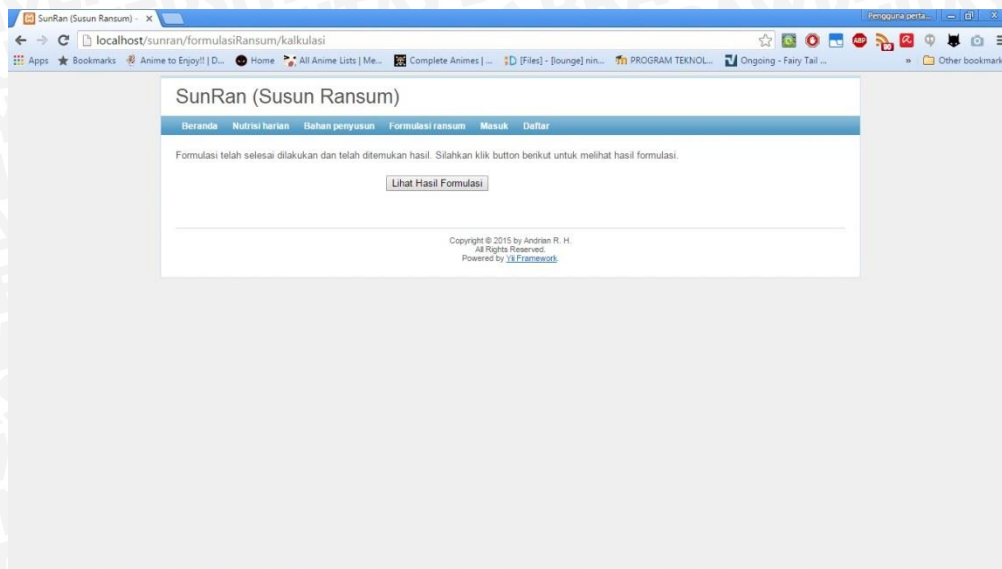
halaman ini disesuaikan dengan rancangan pada bab sebelumnya. Gambar 5.56 berikut menampilkan implementasi halaman antarmuka pemilihan bahan untuk penyusunan ransum.



Gambar 5.56 Implementasi antarmuka pemilihan bahan untuk penyusunan ransum

5.2.5.5 Implementasi antarmuka notifikasi hasil penyusunan ransum

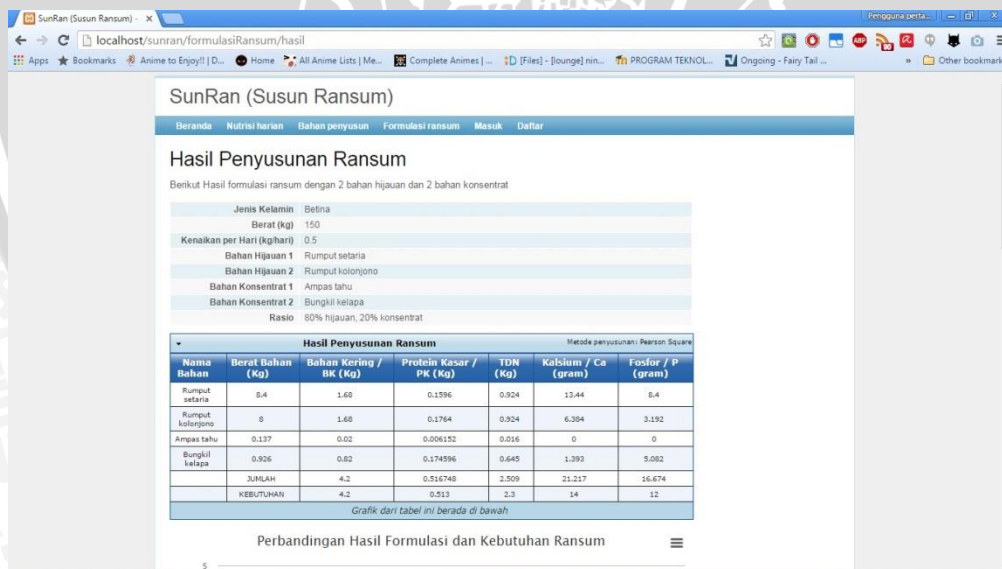
Implementasi halaman antarmuka notifikasi hasil penyusunan ransum menampilkan menu, notifikasi tentang berhasil atau tidaknya dalam proses penyusunan ransum, dan tombol yang mengarahkan halaman ke detail hasil penyusunan (jika hasil penyusunan ransum ditemukan). Tombol tersebut akan mengarahkan ke halaman detail hasil penyusunan ransum. Tombol tersebut akan digantikan dengan tombol yang mengarahkan halaman kembali ke halaman pemilihan bahan jika aplikasi SunRan tidak berhasil menyusun ransum ideal. Implementasi dari halaman ini disesuaikan dengan rancangan pada bab sebelumnya. Gambar 5.57 berikut menampilkan implementasi halaman antarmuka notifikasi hasil penyusunan ransum.



Gambar 5.57 Implementasi antarmuka notifikasi hasil penyusunan ransum

5.2.5.6 Implementasi antarmuka detail hasil penyusunan

Implementasi halaman antarmuka detail hasil penyusunan menampilkan menu, dan detail hasil penyusunan ransum menggunakan serta menyertakan tabel dan diagram untuk mempermudah dalam memahami hasil penyusunan ransum. Implementasi dari halaman ini disesuaikan dengan rancangan pada bab sebelumnya. Gambar 5.58 berikut menampilkan implementasi halaman antarmuka detail hasil penyusunan.



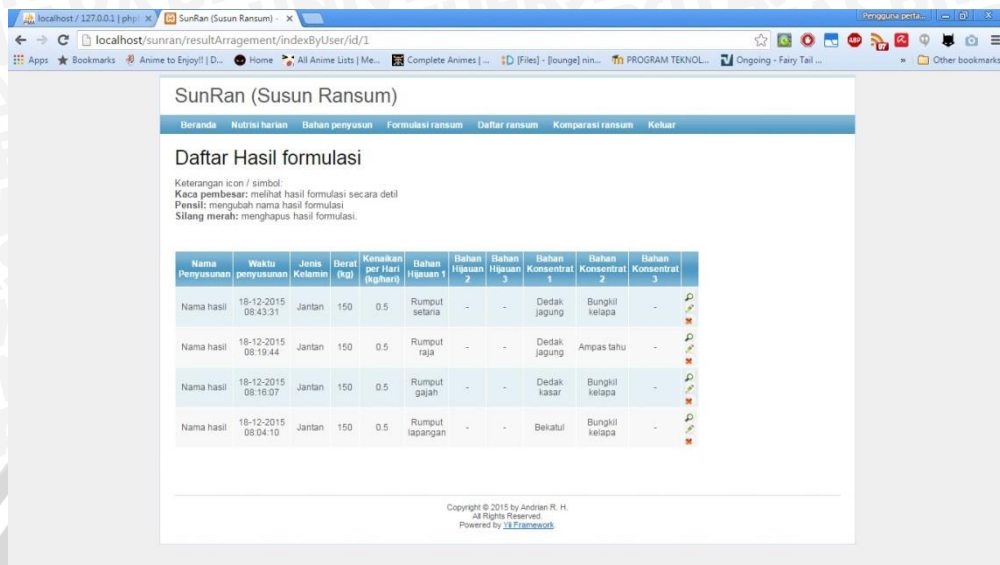
Gambar 5.58 Implementasi antarmuka detail hasil penyusunan

5.2.5.7 Implementasi antarmuka daftar hasil penyusunan ransum tersimpan

Implementasi halaman antarmuka daftar hasil penyusunan tersimpan menampilkan menu, daftar hasil penyusunan ransum yang tersimpan, dan tiga tombol untuk melihat detail, update, dan menghapus pada masing-masing hasil



penyusunan ransum. Implementasi dari halaman ini disesuaikan dengan rancangan pada bab sebelumnya. Gambar 5.59 berikut menampilkan implementasi halaman antarmuka daftar hasil penyusunan tersimpan.



Gambar 5.59 Implementasi antarmuka daftar hasil penyusunan ransum tersimpan

5.2.5.8 Implementasi antarmuka detail hasil penyusunan ransum tersimpan

Implementasi halaman antarmuka detail hasil penyusunan tersimpan menampilkan menu, detail hasil penyusunan ransum tersimpan pada sistem, dan tautan untuk *update* hasil penyusunan ransum yang ditampilkan. Implementasi dari halaman ini disesuaikan dengan rancangan pada bab sebelumnya. Gambar 5.60 berikut menampilkan implementasi halaman antarmuka detail hasil penyusunan tersimpan.

SunRun (Susun Ransum)

Detail formulasi Nama hasil

Operations
Ubah Nama Formulasi

Nama Penyusunan: Nama hasil
Waktu penyusunan: 15-12-2015 08:04:10
Jenis Kelamin: Jantan
Berat (kg): 150
Kenaikan per Hari (kg/hari): 0.5
Bahan Hijauan 1: Rumpul lapangan
Bahan Konsentrat 1: Bekatul
Bahan Konsentrat 2: Bungkil kelapa
Bahan yang direkomendasikan: CaCO3
Rasio: 80% hijauan, 20% konsentrat

Metode penyusunan: Pearson Square

Nama Bahan	Berat Bahan (kg)	Bahan Kering / BK (kg)	Protein Kasar / PK (kg)	TDN (kg)	Kalsium / Ca (gram)	Fosfor / P (gram)
Rumpul lapangan	10.839	3.36	0.3024	1.915	7.728	6.736
Bekatul	0.098	0.086	0.011023	0.06	0.069	1.059
Bungkil kelapa	0.851	0.754	0.160577	0.593	1.282	4.674
CaCO3 / Kapur	0.014972				3.38992	
JURAH	4.2	4.2	0.474	2.568	14.46892	14.469
KEBUTUHAN	4.2	4.2	0.474	2.2	16	10

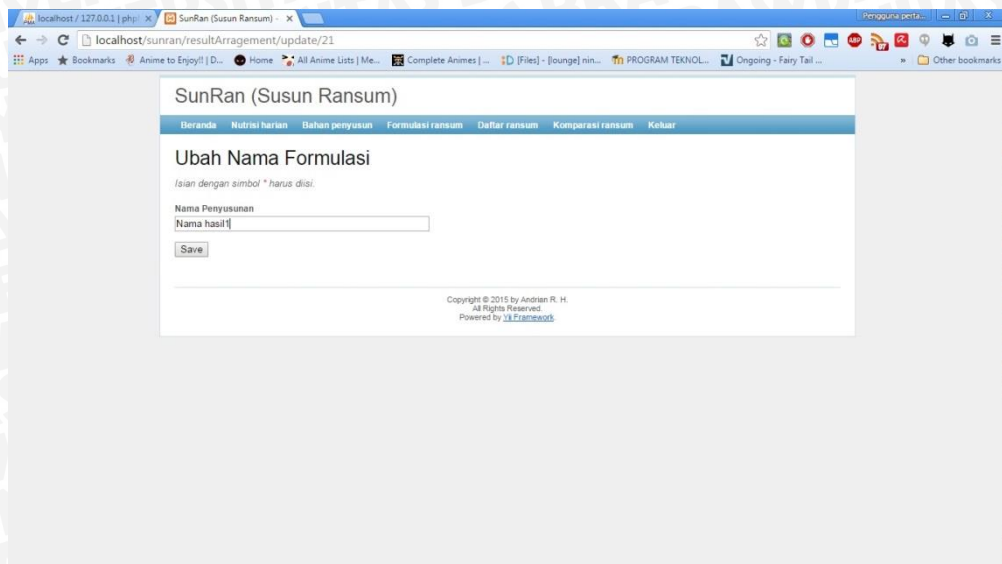
Grafik dari tabel ini berada di bawah

Perbandingan Hasil Formulasi dan Kebutuhan Ransum

Gambar 5.60 Implementasi antarmuka detail hasil penyusunan ransum tersimpan

5.2.5.9 Implementasi antarmuka update hasil penyusunan ransum tersimpan

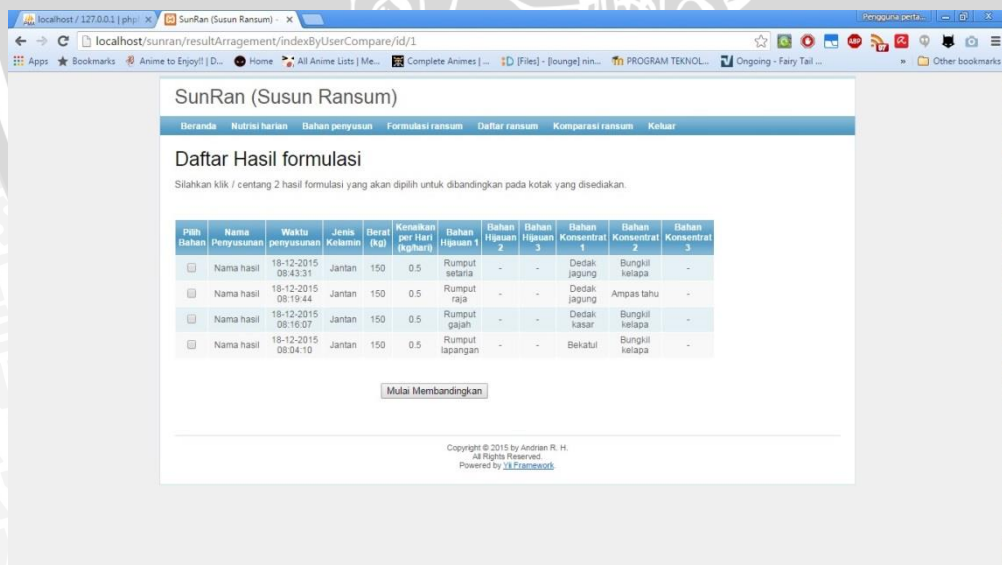
Implementasi halaman antarmuka *update* hasil penyusunan ransum tersimpan menampilkan menu, formulir untuk *update* hasil penyusunan ransum, dan tombol untuk menyimpan data pada formulir. Implementasi dari halaman ini disesuaikan dengan rancangan pada bab sebelumnya. Gambar 5.61 berikut menampilkan implementasi halaman antarmuka *update* hasil penyusunan ransum tersimpan.



Gambar 5.61 Implementasi antarmuka *update* hasil penyusunan ransum tersimpan

5.2.5.10 Implementasi antarmuka daftar hasil penyusunan ransum tersimpan untuk komparasi

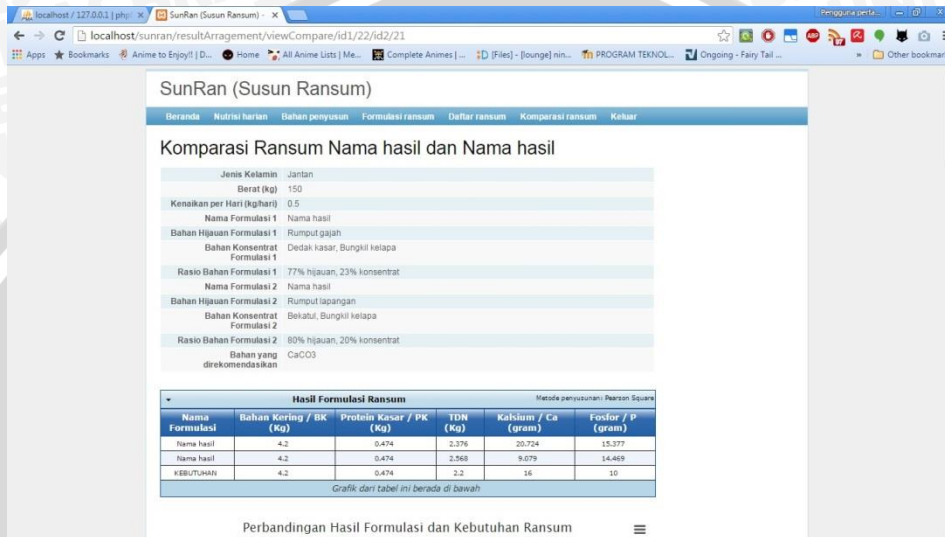
Implementasi halaman antarmuka daftar hasil penyusunan ransum tersimpan untuk komparasi menampilkan menu, daftar hasil penyusunan ransum yang tersimpan (beserta *checkbox* untuk memilih), dan tombol untuk memulai perbandingan. Implementasi dari halaman ini disesuaikan dengan rancangan pada bab sebelumnya. Gambar 5.62 berikut menampilkan implementasi halaman antarmuka daftar hasil penyusunan ransum tersimpan untuk komparasi.



Gambar 5.62 Implementasi antarmuka daftar hasil penyusunan ransum tersimpan untuk komparasi

5.2.5.11 Implementasi antarmuka detail komparasi hasil penyusunan ransum tersimpan

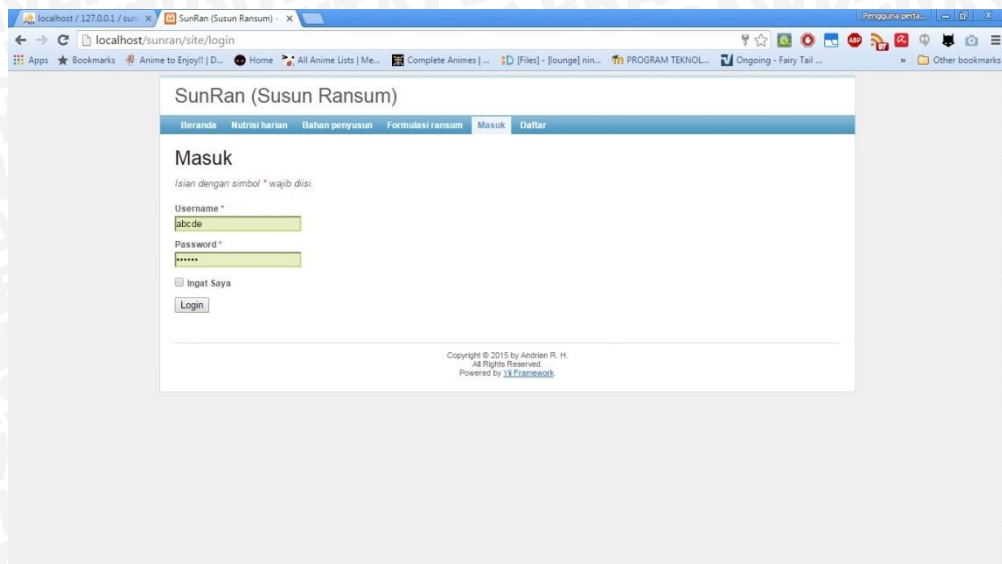
Implementasi halaman antarmuka detail komparasi hasil penyusunan ransum tersimpan menampilkan menu, dan detail perbandingan dua hasil penyusunan ransum beserta tabel dan grafik untuk mempermudah dalam memahami. Implementasi dari halaman ini disesuaikan dengan rancangan pada bab sebelumnya. Gambar 5.63 berikut menampilkan implementasi halaman antarmuka detail komparasi hasil penyusunan ransum tersimpan.



Gambar 5.63 Implementasi antarmuka detail komparasi hasil penyusunan ransum tersimpan

5.2.5.12 Implementasi antarmuka masuk

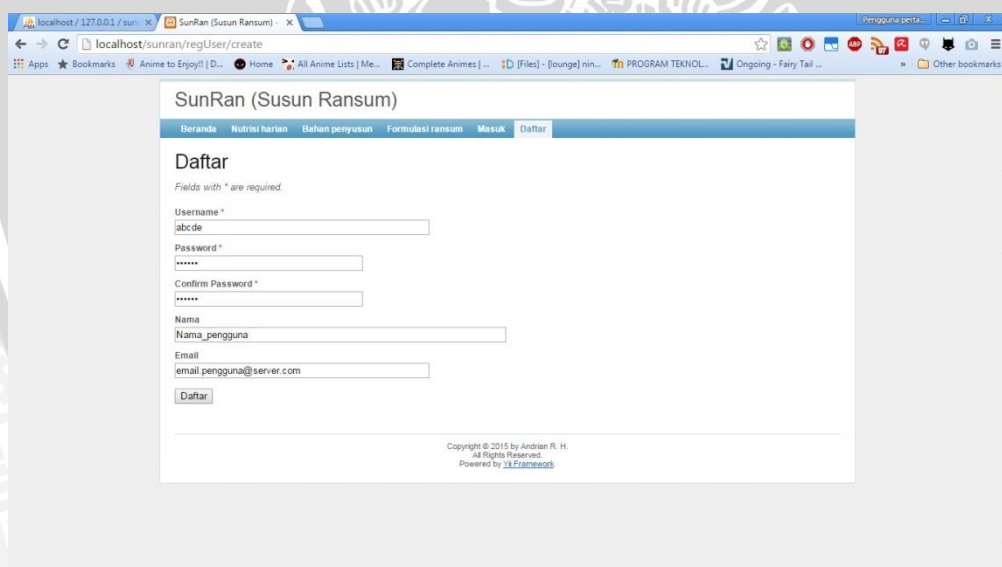
Implementasi halaman antarmuka masuk / login menampilkan menu, formulir untuk *login*, dan tombol untuk memulai proses *login* / masuk ke sistem. Implementasi dari halaman ini disesuaikan dengan rancangan pada bab sebelumnya. Gambar 5.64 berikut menampilkan implementasi halaman antarmuka masuk / *login*.



Gambar 5.64 Implementasi antarmuka masuk

5.2.5.13 Implementasi antarmuka daftar

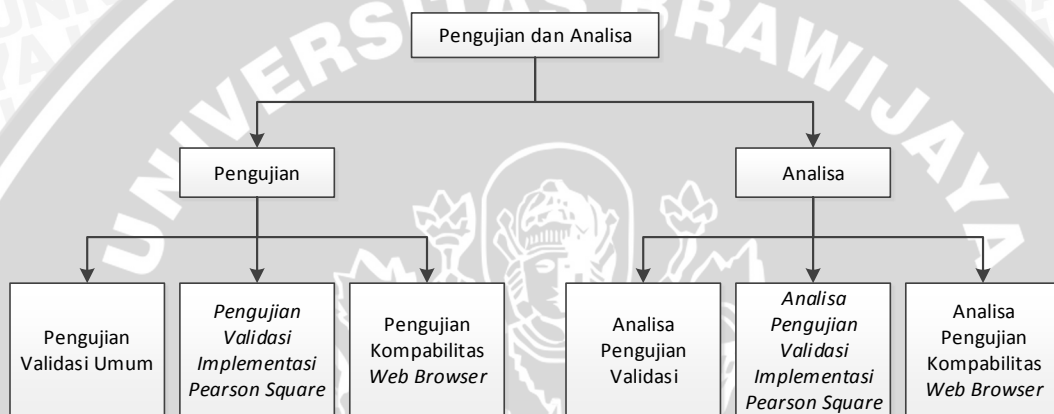
Implementasi halaman antarmuka daftar menampilkan menu, formulir untuk pendaftaran, dan tombol untuk memulai proses pendaftaran. Implementasi dari halaman ini disesuaikan dengan rancangan pada bab sebelumnya. Gambar 5.65 berikut menampilkan implementasi halaman antarmuka daftar.



Gambar 5.65 Implementasi antarmuka daftar

BAB 6 PENGUJIAN DAN ANALISA

Pengujian dan analisa merupakan tahapan yang dilakukan setelah implementasi. Tahapan pengujian dan analisa akan dijelaskan pada bab ini. Pengujian yang akan dibahas pada bab ini terdiri dari pengujian validasi, pengujian implementasi *Pearson Square*, dan komabilitas *web browser*. Pengujian validasi membandingkan perilaku / tindakan yang ditunjukkan oleh sistem dengan perilaku / tindakan yang diharapkan. Pengujian implementasi *Pearson Square* menguji apakah implementasi dari kalkulasi *Pearson Square* di sistem telah sesuai atau tidak. Pengujian komabilitas *web browser* merupakan pengujian dalam menjalankan / mengakses SunRan melalui beberapa aplikasi *web browser*. Tahap pengujian dan analisa bisa dilihat pada Gambar 6.1 berikut.



Gambar 6.1 Diagram pengujian dan analisa

6.1 Pengujian

Tahapan pengujian sesuai pada Gambar 6.1 di atas terdiri dari pengujian validasi umum, pengujian validasi implementasi *Pearson Square*, dan pengujian komabilitas *web browser*. Ketiga pengujian tersebut menguji implementasi yang telah dilakukan. Pengujian validasi umum menentukan apakah implementasi fungsi-fungsi sudah sesuai dengan kebutuhan dan *use case* yang telah dibuat. Pengujian validasi implementasi *Pearson Square* menguji apakah implementasi dari kalkulasi *Pearson Square* di sistem menghasilkan ransum dengan nutrisi yang sama dengan nutrisi ransum yang disusun oleh Microsoft Excel. Tingkat presisi nutrisi hasil pengusunan ransum pada sistem dan pada Microsoft Excel diharapkan mencapai 10^{-2} (dengan satuan yang sama). Pengujian validasi implementasi *Pearson Square* juga dapat diketahui persentase kesalahan nutrisi hasil penyusunan ransum dari aplikasi SunRan terhadap hasil penyusunan melalui Microsoft Excel pada tingkat ketelitian 10^{-7} (dengan satuan yang sama). Pengujian komabilitas *web browser* menentukan apakah implementasi antarmuka sudah sesuai dengan rancangan antarmuka melalui proses pengaksesan dengan aplikasi *web browser*.

6.1.1 Pengujian validasi umum

Pengujian validasi umum yang akan dilakukan menggunakan metode *black-box*. Metode ini memperhatikan apa yang dikerjakan oleh sistem sudah sesuai dengan kebutuhan atau belum. Jika fungsi dari sistem berjalan / menampilkan perilaku (*behavior*) yang sesuai dengan yang diharapkan maka fungsi tersebut valid. Jika fungsi dari sistem berjalan / menampilkan perilaku (*behavior*) yang tidak sesuai dengan yang diharapkan maka fungsi tersebut tidak valid. Masing-masing fungsi akan diuji menggunakan dua kasus pengujian (*test case*). Satu kasus pengujian dengan skenario yang benar / sesuai dengan alur utama (*main flow*). Kasus pengujian satunya dilakukan dengan skenario yang salah / tidak sesuai dengan alur utama. Beberapa fungsi pada aplikasi SunRan antara lain melihat tabel nutrisi, melihat tabel komposisi ransum dan *logout* hanya terdapat skenario benar saja. Skenario dari masing-masing kasus uji pada setiap fungsi dapat dilihat pada Tabel 6.1 hingga Tabel 6.13 berikut.

Tabel 6.1 Skenario kasus uji fungsi melihat tabel nutrisi dengan skenario benar

Kode kasus uji	TC01
Nama kasus uji	Pengujian fungsi melihat tabel nutrisi dengan skenario benar
Skenario	<ol style="list-style-type: none"> 1. Pengguna mengeklik menu nutrisi pada <i>menu bar</i> 2. Sistem menampilkan daftar nutrisi harian sapi potong

Tabel 6.2 Skenario kasus uji fungsi melihat tabel komposisi penyusun ransum dengan skenario benar

Kode kasus uji	TC02
Nama kasus uji	Pengujian fungsi melihat tabel komposisi penyusun ransum dengan skenario benar
Skenario	<ol style="list-style-type: none"> 1. Pengguna mengeklik menu bahan penyusun pada <i>menu bar</i> 2. Sistem menampilkan daftar bahan penyusun ransum

Tabel 6.3 Skenario kasus uji fungsi menyusun ransum dengan skenario benar

Kode kasus uji	TC03
Nama kasus uji	Pengujian fungsi menyusun ransum dengan skenario benar
Skenario	<ol style="list-style-type: none"> 4. Pengguna telah masuk / <i>login</i> ke sistem 5. Pengguna mengeklik menu formulasi pada <i>menu bar</i> 6. Sistem menampilkan opsi jenis kelamin, berat badan, dan penambahan berat badan per hari, dan daftar bahan penyusun ransum (hijauan dan konsentrat) 7. Pengguna memilih jenis kelamin, berat badan, dan penambahan berat badan per hari, dan daftar bahan penyusun ransum (hijauan dan konsentrat) 8. Pengguna menekan tombol mulai formulasi 9. Sistem melakukan proses penyusunan / formulasi ransum

Tabel 6.3 Skenario kasus uji fungsi menyusun ransum dengan skenario benar (lanjutan)

	<ol style="list-style-type: none"> 10. Sistem menampilkan halaman notifikasi mengenai keberhasilan sistem dalam menyusun ransum 11. Pengguna mengklik tombol lihat hasil formulasi 12. Sistem menampilkan hasil penyusunan ransum
--	--

Tabel 6.4 Skenario kasus uji fungsi menyusun ransum dengan skenario salah

Kode kasus uji	TC04
Nama kasus uji	Pengujian fungsi menyusun ransum dengan skenario salah
Skenario	<ol style="list-style-type: none"> 1. Pengguna telah masuk / <i>login</i> ke sistem 2. Pengguna mengklik menu formulasi pada <i>menu bar</i> 3. Sistem menampilkan opsi jenis kelamin, berat badan, dan penambahan berat badan per hari, dan daftar bahan penyusun ransum (hijauan dan konsentrat) 4. Pengguna memilih jenis kelamin, berat badan, dan penambahan berat badan per hari, dan daftar bahan penyusun ransum (hijauan dan konsentrat) 5. Pengguna mengklik tombol mulai formulasi 6. Sistem melakukan proses penyusunan / formulasi ransum 7. Sistem menampilkan halaman notifikasi mengenai ketidakberhasilan sistem dalam menyusun ransum 8. Pengguna mengklik tombol kembali pilih bahan 9. Sistem menampilkan opsi jenis kelamin, berat badan, dan penambahan berat badan per hari, dan daftar bahan penyusun ransum (hijauan dan konsentrat)

Tabel 6.5 Skenario kasus uji fungsi mendaftarkan diri dengan skenario benar

Kode kasus uji	TC05
Nama kasus uji	Pengujian fungsi mendaftarkan diri dengan skenario benar
Skenario	<ol style="list-style-type: none"> 1. Pengguna tidak sedang masuk / <i>login</i> ke sistem 2. Pengguna mengklik menu daftar pada menu bar 3. Sistem menampilkan formulir pendaftaran 4. Pengguna mengisi <i>username</i>, <i>password</i>, <i>confirm password</i>, nama, dan <i>email</i> 5. Pengguna mengklik tombol daftar 6. Sistem menyimpan data pengguna 7. Sistem mengarahkan halaman ke formulir <i>login</i>

Tabel 6.6 Skenario kasus uji fungsi mendaftarkan diri dengan skenario salah

Kode kasus uji	TC06
Nama kasus uji	Pengujian fungsi mendaftarkan diri dengan skenario salah
Skenario	<ol style="list-style-type: none"> 1. Pengguna tidak sedang masuk / <i>login</i> ke sistem 2. Pengguna mengeklik menu daftar pada <i>menu bar</i> 3. Sistem menampilkan formulir pendaftaran 4. Pengguna mengisi <i>username</i>, <i>password</i>, <i>confirm password</i>, nama, dan <i>email</i>. Data <i>username</i> yang diisi telah ada / tersimpan pada sistem 5. Pengguna mengeklik tombol daftar 6. Muncul pesan kesalahan / <i>error</i> bahwa <i>username</i> telah terpakai dan sistem tidak menyimpan data pengguna

Tabel 6.7 Skenario Kasus Uji Fungsi Login / Masuk ke Sistem dengan Skenario Benar

Kode kasus uji	TC07
Nama kasus uji	Pengujian fungsi <i>login</i> / masuk ke sistem dengan skenario benar
Skenario	<ol style="list-style-type: none"> 1. Pengguna tidak sedang masuk / <i>login</i> ke sistem 2. Pengguna mengeklik menu masuk pada <i>menu bar</i> 3. Sistem menampilkan formulir <i>login</i> 4. Pengguna mengisi <i>username</i> dan <i>password</i> 5. Pengguna mengeklik tombol <i>login</i> 6. Sistem mengubah kondisi / status pengguna menjadi masuk ke sistem 7. Sistem mengarahkan halaman ke halaman beranda

Tabel 6.8 Skenario Kasus Uji Fungsi Login / Masuk ke Sistem dengan Skenario Salah

Kode kasus uji	TC08
Nama kasus uji	Pengujian fungsi <i>login</i> / masuk ke sistem dengan skenario salah
Skenario	<ol style="list-style-type: none"> 1. Pengguna tidak sedang masuk / <i>login</i> ke sistem 2. Pengguna mengeklik menu masuk pada menu bar 3. Sistem menampilkan formulir <i>login</i> 4. Pengguna mengisi <i>username</i> dan <i>password</i> 5. Pengguna mengeklik tombol <i>login</i> 6. Sistem tidak mengubah kondisi / status pengguna 7. Sistem mengarahkan halaman ke halaman beranda

Tabel 6.9 Skenario Kasus Uji Fungsi Mengatur Hasil Penyusunan Ransum dengan Skenario Benar

Kode kasus uji	TC09
Nama kasus uji	Pengujian fungsi mengatur hasil penyusunan ransum dengan skenario benar
Skenario	<ol style="list-style-type: none"> 1. Pengguna telah masuk / <i>login</i> ke sistem 2. Pengguna telah menyimpan minimal 1 hasil penyusunan ransum 3. Pengguna mengeklik menu daftar ransum pada <i>menu bar</i> 4. Sistem menampilkan daftar hasil penyusunan ransum yang tersimpan 5. Pengguna mengklik tombol bergambar kaca pembesar (<i>view / lihat detail</i>) di samping kanan salah satu hasil penyusunan ransum 6. Sistem menampilkan detail hasil penyusunan ransum 7. Pengguna mengeklik menu daftar formulasi pada <i>menu bar</i> 8. Sistem menampilkan daftar hasil penyusunan ransum yang tersimpan 9. Pengguna mengklik tombol bergambar pensil (<i>update / ubah</i>) di samping kanan salah satu hasil penyusunan ransum 10. Sistem menampilkan formulir mengubah nama hasil penyusunan ransum 11. Pengguna mengisi nama hasil penyusunan ransum 12. Pengguna mengeklik tombol <i>save</i> 13. Sistem mengubah data nama hasil penyusunan ransum 14. Sistem mengarahkan halaman detail hasil penyusunan ransum 15. Pengguna mengeklik menu daftar ransum pada <i>menu bar</i> 16. Sistem menampilkan daftar hasil penyusunan ransum yang tersimpan 17. Pengguna mengklik tombol bergambar silang merah (<i>delete / hapus</i>) di samping kanan salah satu hasil penyusunan ransum 18. Sistem menampilkan pesan konfirmasi untuk menghapus hasil penyusunan ransum 19. Sistem menghapus hasil penyusunan ransum

Tabel 6.10 Skenario Kasus Uji Fungsi Mengatur Hasil Penyusunan Ransum dengan Skenario Salah

Kode kasus uji	TC10
Nama kasus uji	Pengujian fungsi mengatur hasil penyusunan ransum dengan dengan skenario salah
Skenario	<ol style="list-style-type: none"> 1. Pengguna telah masuk / <i>login</i> ke sistem 2. Pengguna belum menyimpan hasil penyusunan ransum 3. Pengguna mengeklik menu daftar ransum pada <i>menu bar</i> 4. Sistem mengarahkan halaman ke daftar hasil penyusunan ransum serta muncul pesan bahwa tidak ditemukan hasil penyusunan ransum

Tabel 6.11 Skenario Kasus Uji Fungsi Membandingkan Ransum dengan Skenario Benar

Kode kasus uji	TC11
Nama kasus uji	Pengujian fungsi membandingkan ransum dengan skenario benar
Skenario	<ol style="list-style-type: none"> 1. Pengguna telah masuk / <i>login</i> ke sistem 2. Pengguna telah menyimpan minimal 2 hasil penyusunan ransum yang memiliki data jenis kelamin, berat badan, dan pertambahan berat badan per hari yang sama 3. Pengguna mengeklik menu komparasi ransum pada menu bar 4. Sistem menampilkan daftar hasil penyusunan ransum yang tersimpan 5. Pengguna memilih dua ransum dengan cara mencentang kotak kecil di samping kiri hasil penyusunan ransum. Ransum yang dipilih adalah ransum yang memiliki data berat badan, jenis kelamin, dan perambahan berat badan perhari yang sama 6. Pengguna mengklik tombol mulai membandingkan 7. Sistem menampilkan perbandingan dua ransum

Tabel 6.12 Skenario Kasus Uji Fungsi Membandingkan Ransum dengan Skenario Benar

Kode kasus uji	TC12
Nama kasus uji	Pengujian fungsi membandingkan ransum dengan skenario salah
Skenario	<ol style="list-style-type: none"> 1. Pengguna telah masuk / <i>login</i> ke sistem 2. Pengguna telah menyimpan minimal 2 hasil penyusunan ransum yang memiliki data jenis kelamin, berat badan, dan pertambahan berat badan per hari yang sama 3. Pengguna mengeklik menu komparasi ransum pada menu bar 4. Sistem menampilkan daftar hasil penyusunan ransum yang tersimpan 5. Pengguna memilih dua ransum dengan cara mencentang kotak kecil di samping kiri hasil penyusunan ransum. Ransum yang dipilih adalah ransum yang memiliki data berat badan, jenis kelamin, dan perambahan berat badan perhari yang berbeda 6. Muncul pesan peringatan bahwa kedua ransum tidak bisa dibandingkan karena berat badan, jenis kelamin, dan ertambahan berat badan per hari yang berbeda

Tabel 6.13 Skenario Kasus Uji Fungsi Logout / Keluar dari Sistem dengan Skenario Benar

Kode kasus uji	TC13
Nama kasus uji	Pengujian fungsi <i>logout</i> / keluar dari sistem dengan skenario benar
Skenario	<ol style="list-style-type: none"> 1. Pengguna telah masuk / <i>login</i> ke sistem 2. Pengguna mengeklik menu keluar pada <i>menu bar</i> 3. Sistem mengubah kondisi / status pengguna menjadi keluar dari sistem 4. Sistem mengarahkan halaman ke halaman beranda

Skenario-skenario di diatas adalah proses yang dijalankan selama proses validitas fungsi-fungsi pada aplikasi SunRan. Hasil dari pengujian validasi fungsi-fungsi pada aplikasi SunRan dapat dilihat pada tabel berikut.

Tabel 6.14 Hasil Pengujian Validitas

Kode Kebutuhan	Kode Kasus Uji	Hasil yang Diharapkan	Hasil Aktual	Validitas
SRS_002_1	TC01	Sistem menampilkan daftar nutrisi harian	Sistem menampilkan daftar nutrisi harian	Valid
SRS_002_2	TC02	Sistem menampilkan bahan penyusun ransum	Sistem menampilkan bahan penyusun ransum	Valid
SRS_002_3	TC03	Sistem menyusun ransum serta menampilkan hasil penyusunan ransum	Sistem menyusun ransum serta menampilkan hasil penyusunan ransum	Valid
SRS_002_3	TC04	Muncul peringatan untuk memilih bahan konsentrat dan sistem tidak melakukan proses penyusunan / formulasi ransum	Muncul peringatan untuk memilih bahan konsentrat dan sistem tidak melakukan proses penyusunan / formulasi ransum	Valid
SRS_001_1	TC05	Data pengguna tersimpan dan halaman diarahkan ke formulir <i>login</i> / masuk	Data pengguna tersimpan dan halaman diarahkan ke formulir <i>login</i> / masuk	Valid
	TC06	Muncul pesan kesalahan / <i>error</i> bahwa <i>username</i> telah terpakai dan sistem tidak menyimpan data pengguna	Muncul pesan kesalahan / <i>error</i> bahwa <i>username</i> telah terpakai dan sistem tidak menyimpan data pengguna	Valid
SRS_001_2	TC07	Pengguna masuk ke sistem dan pengguna diarahkan ke halaman beranda	Pengguna masuk ke sistem dan pengguna diarahkan ke halaman beranda	Valid
	TC08	Pengguna tidak masuk ke sistem dan pengguna diarahkan ke halaman beranda	Pengguna tidak masuk ke sistem dan pengguna diarahkan ke halaman beranda	Valid
SRS_001_3, SRS_001_4, dan SRS_001_5	TC09	Sistem menampilkan secara detail, mengubah nama, dan menghapus hasil penyusunan ransum tersimpan	Sistem menampilkan secara detail, mengubah nama, dan menghapus hasil penyusunan ransum tersimpan	Valid
	TC10	Sistem mengarahkan halaman ke daftar hasil penyusunan ransum serta muncul pesan bahwa tidak ditemukan hasil penyusunan ransum	Sistem mengarahkan halaman ke daftar hasil penyusunan ransum serta muncul pesan bahwa tidak ditemukan hasil penyusunan ransum	Valid

Tabel 6.14 Hasil Pengujian Validitas (lanjutan)

Kode Kebutuhan	Kode Kasus Uji	Hasil yang Diharapkan	Hasil Aktual	Validitas
SRS_001_6	TC11	Sistem menampilkan perbandingan dua ransum	Sistem menampilkan perbandingan dua ransum	Valid
	TC12	Muncul pesan peringatan bahwa kedua ransum tidak bisa dibandingkan karena berat badan, jenis kelamin, dan ertambahan berat badan per hari yang berbeda	Muncul pesan peringatan bahwa kedua ransum tidak bisa dibandingkan karena berat badan, jenis kelamin, dan ertambahan berat badan per hari yang berbeda	Valid
SRS_001_7	TC13	Pengguna keluar dari sistem	Pengguna keluar dari sistem	Valid

6.1.2 Pengujian validasi implementasi *Pearson Square*

Pengujian validasi implementasi *Pearson Square* yang akan dilakukan untuk membuktikan hasil penyusunan ransum dari sistem dan dari Microsoft Excel memiliki nutrisi yang sama pada tingkat ketelitian 10^{-2} (dalam satuan yang sama). Pengujian ini dilakukan dengan cara membandingkan nutrisi bahan penyusun ransum pada hasil penyusunan ransum yang telah disusun oleh aplikasi SunRan dengan perhitungan *Pearson Square* yang ada di Microsoft Excel. Pengujian ini juga mencari persentase kesalahan pada nutrisi ransum yang disusun oleh aplikasi SunRan. Persentase kesalahan didapatkan melalui selisih nutrisi pada ransum yang disusun oleh aplikasi Sunran dengan nutrisi ransum yang dihitung melalui Microsoft Excel pada tingkat ketelitian 10^{-7} . Pengujian implementasi *Pearson Square* ini dilakukan beberapa kali dengan variabel tetap sebagai berikut.

Tabel 6.15 Variabel tetap pengujian validasi implementasi *Pearson Square*

Jenis Kelamin	Jantan
Berat (kg)	150
Kenaikan per Hari (kg/hari)	0.5
Jumlah bahan hijauan	Satu
Jumlah bahan konsentrat	Dua

Variabel tidak tetap yang dilakukan selama pengujian ini adalah bahan hijauan dan bahan konsentrat. Bahan hijauan, konsentrat, keberhasilan aplikasi SunRan dalam menemukan ransum ideal pada masing-masing pengujian dapat dilihat pada tabel berikut.

Tabel 6.16 Bahan hijauan, konsentrat dan keberhasilan menemukan ransum ideal pada pengujian validasi implementasi *Pearson Square*

Kode Kasus Uji	Bahan Hijauan	Bahan Konsentrat 1	Bahan Konsentrat 1	Keberhasilan menemukan ransum ideal
TC01	Rumput lapangan	Bekatul	Bungkil kelapa	Berhasil
TC02	Rumput gajah	Dedak kasar	Bungkil kelapa	Berhasil
TC03	Rumput raja	Dedak jagung	Ampas tahu	Berhasil
TC04	Rumput cetaria	Dedak jagung	Bungkil kelapa	Berhasil
TC05	Rumput kalanjono	Tongkol jagung	Kulit kedelai	Berhasil
TC06	Daun tebu	Bekatul	Dedak jagung	Tidak Berhasil

Validitas hasil pengujian dari keseluruhan kasus uji dapat dilihat pada tabel berikut.

Tabel 6.17 Hasil uji validitas implementasi *Pearson Square*

Kode Kasus Uji	Keberhasilan menemukan ransum ideal pada SunRan	Akurasi hasil hingga dua angka di belakang koma (satuan yang sama)	Validitas
TC01	Berhasil	Tidak terpenuhi	Tidak Valid
TC02	Berhasil	Tidak terpenuhi	Tidak Valid
TC03	Berhasil	Terpenuhi	Valid
TC04	Berhasil	Terpenuhi	Valid
TC05	Berhasil	Tidak terpenuhi	Tidak Valid
TC06	Tidak berhasil	Tidak relevan	Valid

Tiga tabel berikut adalah hasil pengujian validasi implementasi *Pearson Square* dengan kode uji TC01.

Tabel 6.18 Detail hasil penyusunan ransum dari TC01 melalui aplikasi SunRan

Nama Bahan	Berat Bahan (Kg)	Bahan Kering / BK (Kg)	Protein Kasar / PK (Kg)	TDN (Kg)	Kalsium / Ca (gram)	Fosfor / P (gram)
Rumput lapangan	10.839	3.36	0.3024	1.915	7.728	8.736
Bekatul	0.098	0.086	0.011023	0.06	0.069	1.059
Bungkil kelapa	0.851	0.754	0.160577	0.593	1.282	4.674
CaCO3 / Kapur	0.014972				5.38992	
	JUMLAH	4.2	0.474	2.568	14.46892	14.469
	KEBUTUHAN	4.2	0.474	2.2	16	10

Tabel 6.19 Detail hasil penyusunan ransum dari TC01 melalui Microsoft Excel

	BK(Kg)	PK (Kg)	TDN (Kg)	Ca (gr)	P (gr)
Rumput lapangan	3.36	0.3024	1.9152	7.728	8.736
Bekatul	0.0861176	0.0110231	0.0601962	0.0688941	1.0592471
Bungkil kelapa	0.7538824	0.1605769	0.5933054	1.2816	4.6740706
CaCO3				5.3908235	
Total	4.2	0.474	2.5687016	14.4693176	14.4693176
Kebutuhan	4.2	0.474	2.2	16	10

Tabel 6.20 Berat bahan penyusun dalam hasil penyusunan ransum dari TC01 melalui Microsoft Excel

Bahan Segar	Berat (Kg)
Rumput lapangan	10.8387097
Bekatul	0.0978610
Bungkil kelapa	0.8508830
CaCO3	0.0149745

Tabel 6.21 Persentase kesalahan dari TC01

Nama Bahan	Kesalahan Berat Bahan (%)	Kesalahan Bahan Kering / BK (%)	Kesalahan Protein Kasar / PK (%)	Kesalahan TDN (%)	Kesalahan Kalsium / Ca (%)	Kesalahan Fosfor / P (%)
Rumput lapangan	0.0026786	0	0	0.0104428	0	0
Bekatul	0.1420765	0.1366120	0.0005336	0.3259926	0.1536885	0.0233240
Bungkil kelapa	0.0137484	0.0156055	0.0000366	0.0514763	0.0312110	0.0015102
CaCO3	0.0167605				0.0167605	
	JUMLAH	0	0	0.0273152	0.0027482	0.0021953

Rata-rata dari seluruh persentase kesalahan di atas adalah 0.0389887%.

Detail hasil pengujian validasi implementasi *Pearson Square* dengan kode uji TC02 hingga TC05 di atas dapat dilihat pada Lampiran E.1 hingga E.4. Persentase kesalahan dari nutrisi ransum yang disusun oleh aplikasi Sunran dapat dilihat pada tabel berikut.

Tabel 6.22 Persentase kesalahan hasil penyusunan ransum dari kasus uji

Kode kasus uji	Persentase kesalahan (%)
TC01	0.0389887
TC02	6.4199148
TC03	0.0236629
TC04	0.0275560
TC05	0.0260935
Rata-rata	1.3072432

Persentase kesalahan untuk masing-masing kasus uji di atas diperoleh dengan cara menghitung rata-rata selisih nutrisi yang disusun oleh aplikasi SunRan dengan nutrisi ransum yang disusun oleh Microsoft Excel pada tingkat ketelitian 10^{-7} .

6.1.3 Pengujian kompatibilitas *web browser*

Pengujian kompatibilitas *web browser* yang akan dilakukan bertujuan untuk menguji tampilan / antarmuka apakah sesuai dengan rancangan yang sudah dibuat atau tidak. Jika antarmuka yang ditampilkan oleh sistem sesuai dengan yang diharapkan maka status pengujian tersebut valid. Jika antarmuka yang ditampilkan oleh sistem tidak sesuai dengan yang diharapkan maka status pengujian tersebut tidak valid. Aplikasi *web browser* yang akan dipakai pada pengujian ini antara lain Chrome, Firefox, dan Opera. Tabel 6.23 hingga Tabel 6.25 menjelaskan pengujian kompatibilitas sistem terhadap *web browser*.

Tabel 6.23 Uji Validasi Kompatibilitas pada *Web Browser Chrome*

Nama pengujian	Kompatibilitas pada <i>web browser</i> Chrome
Tujuan pengujian	Menguji validitas kinerja sistem SunRan dalam menampilkan antarmuka pada <i>web browser</i> Chrome
Prosedur pengujian	Mengakses setiap halaman pada sistem
Hasil yang diharapkan	Sistem mampu menampilkan antarmuka pada masing-masing halaman yang ada pada sistem sesuai dengan rancangan antarmuka
Hasil yang didapatkan	Sistem menampilkan antarmuka pada masing-masing halaman yang ada pada sistem sesuai dengan rancangan antarmuka dengan baik dan benar
Status validasi	Valid

Tabel 6.24 Uji Validasi Kompabilitas pada *Web Browser Firefox*

Nama pengujian	Kompabilitas pada <i>web browser</i> Firefox
Tujuan pengujian	Menguji validitas kinerja sistem SunRan dalam menampilkan antarmuka pada <i>web browser</i> Firefox
Prosedur pengujian	Mengakses setiap halaman pada sistem
Hasil yang diharapkan	Sistem mampu menampilkan antarmuka pada masing-masing halaman yang ada pada sistem sesuai dengan rancangan antarmuka
Hasil yang didapatkan	Sistem menampilkan antarmuka pada masing-masing halaman yang ada pada sistem sesuai dengan rancangan antarmuka dengan baik dan benar
Status validasi	Valid

Tabel 6.25 Uji Validasi Kompabilitas pada *Web Browser Opera*

Nama pengujian	Kompabilitas pada <i>web browser</i> Opera
Tujuan pengujian	Menguji validitas kinerja sistem SunRan dalam menampilkan antarmuka pada <i>web browser</i> Opera
Prosedur pengujian	Mengakses setiap halaman pada sistem
Hasil yang diharapkan	Sistem mampu menampilkan antarmuka pada masing-masing halaman yang ada pada sistem sesuai dengan rancangan antarmuka
Hasil yang didapatkan	Sistem menampilkan antarmuka pada masing-masing halaman yang ada pada sistem sesuai dengan rancangan antarmuka dengan baik dan benar
Status validasi	Valid

6.2 Analisa

Tahap analisa ini merupakan tahapan untuk mendapatkan kesimpulan terhadap hasil-hasil pengujian kepada sistem SunRan yang telah dilakukan sebelumnya. Pengujian yang dianalisa tak hanya pengujian validasi saja, namun pengujian kompabilitas *web browser*. Sub-sub bab berikut akan menjelaskan analisa dari masing-masing pengujian.

6.2.1 Analisa pengujian validasi umum

Analisa dari pengujian validasi terhadap fungsi-fungsi pada aplikasi SunRan didasari atas hasil validasi pada Tabel 6.14. Hasil validasi pada Tabel 6.14 menunjukkan bahwa seluruh implementasi fungsi-fungsi pada aplikasi SunRan telah valid atau telah berjalan dengan baik dan benar. Hasil-hasil validasi di atas juga menunjukkan bahwa validitas sistem sebesar 100%.

6.2.2 Analisa pengujian implementasi *Pearson Square*

Analisa dari pengujian implementasi *Pearson Square* pada aplikasi SunRan didasari atas hasil validasi pada Tabel 6.17 dan Tabel 6.22. kelima kasus uji di atas cukup untuk menunjukkan hasil pengujian implementasi *Pearson Square*. Pengujian ini menunjukkan bahwa dua dari lima hasil penyusunan ransum memiliki ketelitian hingga 10^{-2} terhadap hasil penyusunan ransum melalui Microsoft Excel (dengan metode *Pearson Square*).

Pengujian dengan kode uji TC01 memiliki perbedaan pada nutrisi kalsium dari bahan CaCO_3 . Nutrisi kalsium dari bahan CaCO_3 berdasarkan penyusunan oleh aplikasi SunRan sebanyak 5.38992 gram, sedangkan berdasarkan perhitungan di Excel sebanyak 5.3908235 gram. Perbedaan nilai nutrisi tersebut disebabkan oleh adanya perbedaan nutrisi kalsium serta fosfor pada bahan bekatul dan bungkil kelapa pada tingkat ketelitian 10^{-7} .

Pengujian dengan kode uji TC02 memiliki perbedaan pada nutrisi kalsium dari bahan bungkil kelapa. Nutrisi kalsium dari bahan bungkil kelapa berdasarkan hasil penyusunan oleh aplikasi SunRan sebanyak 1.64 gram, sedangkan berdasarkan perhitungan di Excel sebanyak 1.6399664 gram. Perbedaan tersebut disebabkan oleh adanya pembulatan angka pada proses penyusunan ransum pada aplikasi SunRan.

Pengujian dengan kode uji TC05 memiliki perbedaan pada berat bahan tongkol jagung. Berat bahan tongkol jagung berdasarkan hasil penyusunan oleh aplikasi SunRan sebanyak 0.31 Kg, sedangkan berdasarkan perhitungan di Excel sebanyak 0.3097795 Kg. Perbedaan tersebut disebabkan oleh adanya pembulatan angka pada proses penyusunan ransum pada aplikasi SunRan

Hasil pengujian kasus uji dengan kode TC06 (pada Tabel 6.17) menunjukkan bahwa tidak ditemukan ransum ideal karena setelah dilakukan proses penyusunan ransum masih belum bisa memenuhi nutrisi. Tidak ditemukannya ransum ideal dari TC06 berdampak tidak bisa dicarinya kesamaan nutrisi ransum (pada tingkat ketelitian 10^{-2}) dan persentase kesalahan pada kasus uji tersebut. Jika kasus uji TC06 dilakukan di Microsoft Excel, maka hasil akhir penyusunan ransum dapat dilihat pada tabel berikut.

Tabel 6.26 Detail hasil penyusunan ransum dari TC06 melalui Microsoft Excel

	BK(Kg)	PK (Kg)	TDN (Kg)	Ca (gr)	P (gr)
Daun tebu	0.84	0.04368	0.43176	3.948	2.856
Bekatul	3.3514721	0.4289884	2.342679	2.6811777	41.2231066
Dedak jagung	0.0085279	0.0008528	0.0070185	0	0
CaCO_3				37.4499289	
Total	4.2	0.4735212	2.7814575	44.0791066	44.0791066
Kebutuhan	4.2	0.474	2.2	16	10

Tabel 6.27 Berat bahan penyusun dalam hasil penyusunan ransum dari TC06 melalui Microsoft Excel

Bahan Segar	Berat (Kg)
Daun tebu	3.2941176
Bekatul	3.808491
Dedak jagung	0.0097129
CaCO3	0.1040276

Hasil lain dari pengujian ini (berdasarkan Tabel 6.22) menunjukkan bahwa dengan ketelitian 10^{-7} terdapat persentase kesalahan dari nutrisi hasil penyusunan ransum sebesar 1.3072432%. Persentase kesalahan sekecil itu berdasarkan diskusi dengan pihak pakar masih bisa diterima. Persentase kesalahan pada masing-masing kasus uji (dengan kode TC01 hingga TC05) dapat dilihat pada Lampiran E.1 hingga E.4.

6.2.3 Analisa pengujian komabilitas *web browser*

Analisa dari pengujian komabilitas *web browser* terhadap antarmuka pada sistem SunRan didasari atas hasil validasi pada Tabel 6.18 hingga Tabel 6.20. Hasil validasi pada tabel-tabel tersebut menunjukkan bahwa seluruh implementasi antarmuka pada sistem SunRan telah valid atau telah ditampilkan dengan baik dan benar. Hasil-hasil validasi di atas juga menunjukkan bahwa sistem SunRan kompatibel terhadap aplikasi *web browser* Chrome, Firefox, dan Opera.

BAB 7 PENUTUP

7.1 Kesimpulan

Proses perancangan, implementasi dan pengujian yang telah dilakukan dapat ditarik beberapa kesimpulan. Kesimpulan-kesimpulan tersebut antara lain:

1. Proses perumusan kebutuhan untuk aplikasi SunRan dilakukan dengan cara menganalisa hasil wawancara dengan pihak pakar lalu dilakukan pemodelan *use case*.
2. Proses perancangan aplikasi SunRan disesuaikan dengan kebutuhan-kebutuhan yang telah diidentifikasi. Proses perancangan yang dilakukan terdiri dari pemodelan kelas, interaksi, data, dan perancangan antarmuka. Proses implementasi aplikasi SunRan menggunakan beberapa perangkat lunak, yaitu *framework Yii*, *highchart* (sebagai *extention* dari *framework Yii*), *htmltableui* (sebagai *extention* dari *framework Yii*), dan *MySQL* (sebagai *Relational Database Management System / RDBMS*).
3. Hasil pengujian aplikasi SunRan dari segi validasi sistem menunjukkan keberhasilan sistem dalam menjalankan seluruh fungsi, antara lain fungsi pendaftaran diri, *login*, menampilkan tabel nutrisi, menampilkan tabek bahan penyusun, formulasi ransum, manajemen hasil penyusunan ransum, komparasi ransum, dan *logout*. Pengujian implementasi *Pearson Square* pada aplikasi SunRan menunjukkan bahwa 2 dari 5 kasus uji (yang disusun oleh aplikasi SunRan menggunakan metode *Pearson Square* dan menghasilkan ransum ideal) memiliki nutrisi ransum sama dengan nutrisi ransum yang disusun oleh *Microsoft Excel* (menggunakan metode *Pearson Square*) dengan tingkat kesamaan dua angka di belakang koma (dalam satuan yang sama). Pengujian implementasi *Pearson Square* juga menunjukkan bahwa persentase kesalahan yang kecil. Kecilnya persentase tersebut berdasarkan diskusi dengan pihak pakar masih bisa diterima. Pengujian *kompatibilitas web browser* terhadap sistem SunRan menunjukkan bahwa sistem SunRan dapat berjalan dengan baik dan benar pada aplikasi *web browser* *Chrome*, *Firefox*, dan *Opera*.

7.2 Saran

Aplikasi SunRan yang merupakan produk dalam pengerjaan skripsi ini merupakan aplikasi *prototype* sehingga masih belum layak untuk dirilis ke publik. Pengembangan lebih lanjut masih diperlukan jika ingin merilis sistem ini ke publik. Pengembangan lanjutan yang bisa diimplementasikan pada sistem Sunran antara lain:

1. Memberikan kemampuan untuk diakses melalui perangkat *mobile* baik melalui *smartphone* maupun *tablet*.
2. Memberikan kemampuan untuk mengubah dan menghapus bahan penyusun ransum yang ditambahkan oleh pengguna.
3. Memberikan kemampuan untuk memilih metode penyusunan ransum.

4. Fitur *administrator* yang bisa mengatur keseluruhan sistem.
5. Fitur menentukan harga bahan penyusun ransom.
6. Fitur kalkulasi harga keseluruhan bahan pada suatu ransom.
7. Fitur menentukan batas maksimal dan minimal untuk masing-masing bahan penyusun ransom.
8. Fitur membandingkan tiga hasil penyusunan ransom.
9. Pengaturan untuk menampilkan hasil penyusunan ransom secara detail atau secara sederhana (hanya data penting saja yang ditampilkan).
10. Efisiensi penggunaan *resource* selama proses penyusunan ransom.
11. Fitur mengunduh hasil penyusunan ransom dalam bentuk *file* berekstensi *.pdf maupun *.docx.
12. Fitur mencetak hasil penyusunan ransom baik secara detail maupun secara sederhana (hanya data penting saja yang dicetak).
13. Menyimpan hasil penyusunan ransom ke *cloud storage* dalam bentuk *file* berekstensi *.pdf maupun *.docx.
14. Mengirimkan hasil penyusunan ransom sebagai *file* lampiran pada surel.
15. Mengirimkan hasil penyusunan ransom kepada pengguna terdaftar lainnya.



DAFTAR PUSTAKA

- Badiyanto, 2013. *Buku Pintar Framework Yii*. Yogyakarta: Mediakom.
- Hartadi, H., Reksohardiprodjo, S. dan Tillman, A. D., 2005. *Tabel Komposisi Pakan untuk Indonesia*. 5th ed. Yogyakarta: Gajah Mada University Press.
- Kearl, L. C., 1982. *Nutrient Requirements of Ruminants in Developing Countries*. Utah: International Feedstuffs Insitute.
- Parakkasi, A., 1998. *Ilmu Nutrisi dan Makanan Ternak Ruminan*. Jakarta: UI-Press.
- Pressman, R. S., 2010. "Software Engineering: A Practitioner's Approach", edisi ke-7, McGraw-Hill, Amerika Serikat.
- Pusat Bahasa Departemen Pendidikan Nasional Republik Indonesia, 2008. Kamus Besar Bahasa Indonesia (KBBI) Daring. [online] Tersedia di <http://badanbahasa.kemdikbud.go.id/kbbi/index.php> [23 Desember 2015]
- Sharive, 2013. *Yii Framework: Menguasai Framework PHP Terbaik*. Yogyakarta: Lokomedia.
- Sharive, 2014. *Proyek Membangun Website dengan Yii*. Yogyakarta: Lokomedia.
- Sommerville, I., 2007. *Software Engineering*. 8th ed. Harlow: Pearson Education Limited.
- Umiyasih, U. dan Anggraeny, Y. N., 2007. *Petunjuk Teknis Ransum Seimbang, Strategi Pakan pada Sapi Potong*. Pasuruan: Pusat Penelitian dan Pengembangan Peternakan.
- Utomo, R., 2012. *Bahan Pakan Berserat untuk Sapi*. Yogyakarta: Citra Aji Parama.
- Widodo, P. P. dan Herlawati. 2011. *Menggunakan UML*. Bandung: Informatika.
- Yulianto, P. dan Saparinto, C., 2012. *Penggemukan Sapi Potong Hari Per Hari*. 2nd ed, Jakarta: Penebar Swadaya.



LAMPIRAN A HASIL WAWANCARA PIHAK PAKAR

Dr. Ir. Herni Sudarwati, MS. 2013. *Ransum dan Nutrisi Sapi Potong*. Diwawancara oleh Andrian Riza Hermawan. [tanya jawab langsung] Universitas Brawijaya, 15 Desember 2013.

No.	Pertanyaan Pewawancara	Jawaban Pihak Pakar
1	Apakah yang disebut dengan ransum itu?	Pakan yang diberikan kepada ternak selama 24 jam.
2	Apakah tujuan dalam penyusunan ransum?	Ransum disusun untuk memenuhi kebutuhan nutrisi ternak, contohnya pada sapi potong lokal.
3	Apakah dalam tujuan penyusunan ransum ini hanya diperuntukkan kepada sapi potong jantan?	Tidak, sapi potong betina juga bisa mengikuti proses penggemukan, kecuali sapi potong betina yang sedang bunting.
4	Kapan sapi potong bisa mengikuti proses penggemukan?	Sejak lepas sapih, yaitu sejak berhenti menyusui ke induknya (sekitar 3 bulan).
5	Sampai seberapa banyak pertambahan berat badan sapi potong selama proses penggemukan?	Dalam satu hari diharapkan bertambah bobot badan hingga 1 Kg/hari.
6	Apakah ada kriteria khusus dari penyusunan ransum agar sapi potong cepat bertambah gemuk?	Ada, yaitu dengan menyusun ransum ideal.
7	Kriteria seperti apa ransum ideal tersebut?	Ransum ideal adalah ransum mengandung semua zat makanan dalam kuantitas, kualitas serta perbandingan yang cukup untuk memenuhi kebutuhan zat makanan yang diperlukan ternak. Dari nutrisi harian yang diperlukan, yang perlu diperhatikan adalah komponen BK (Bahan Kering), PK (Protein Kasar), TDN (<i>Total Digestible Nutrient</i>), kalsium dan fosfor. Komponen BK, PK, dan TDN harus diatas kebutuhan nutrisi harian, sedangkan kalsium dan fosfor boleh dibawah nutrisi harian tapi kalsium diusahakan untuk tidak kurang dari fosfor.
8	Bagaimana cara dalam menyusun ransum ideal tersebut?	Bisa menggunakan metode <i>Pearson Square</i> .
9	Bagaimana cara penyusunan ransum menggunakan metode <i>Pearson Square</i> ?	Yaitu menentukan perbandingan (rasio) antara bahan hijauan dan bahan konsentrat. Rasio bahan hijauan maksimal 80% dan minimal 20%.
10	Apakah metode <i>Pearson Square</i> ini hanya terbatas dengan 2 bahan saja?	Tidak, metode <i>Pearson Square</i> bisa untuk bahan lebih dari 2 jenis.
11	Jika saya ingin menyusun ransum menggunakan bahan hingga 3	Bisa, dengan mengembangkan metode <i>Pearson Square</i> tersebut.

	bahan hijauan dan 3 bahan konsentrat dengan memakai metode <i>Pearson Square</i> apakah bisa?	
12	Bagaimana cara menyusunnya?	Pertama, membagi rasio antara seluruh bahan hijauan dan bahan konsentrat. Jika bahan hijauan sebanyak 1 jenis maka langung mengikuti rasio bahan hijauan. Jika bahan hijauan sebanyak 2 jenis maka BK (Bahan Kering) untuk masing-masing hijauan sebanyak separuh dari BK yang diperlukan. Jika bahan hijauan sebanyak 3 jenis maka BK setiap hijauan sebanyak sepertiga dari BK yang diperlukan. Jika bahan konsentrat sebanyak 1 jenis maka langung mengikuti rasio bahan konsentrat. Jika bahan konsentrat sebanyak 2 jenis maka digunakan metode <i>Pearson Square</i> . Jika bahan konsentrat sebanyak 3 jenis maka BK untuk satu bahan konsentrat sebanyak sepertiga dari BK untuk bahan konsentrat, sedangkan 2 bahan konsentrat lainnya menggunakan metode <i>Pearson Square</i> . Untuk kriteria yang dipakai waktu perhitungan <i>Pearson Square</i> pakai nutrisi PK (Protein Kasar).
13	Apakah ada kemungkinan nutrisi mineral dari hasil penyusunan ransum masih kurang dari nutrihari harian yang dibutuhkan?	Mungkin saja, terkadang kalsium dan fosfor bisa dibawah nutrisi harian. Tetapi, mineral usahakan sama banyak dengan fosfor.
14	Bagaimana menambahkan mineral kalsium agar bisa sebanyak mineral fosfor?	Dengan menambahkan kapur (CaCO_3) sebanyak $\frac{100}{36}$ terhadap selisih antara kalsium dengan fosfor dari hasil penyusunan ransum.
15	Fitur apa saja yang Anda butuhkan jika saya membuat aplikasi penyusun ransum?	Pertama, jelas fitur menyusun ransum, melihat data bahan dan nutrisi harian. Kalau fitur lainnya seperti bisa membandingkan, melihat, mengubah nama ransum, dan menghapus ransum juga bagus kalau dibuat juga.
	Pihak Pakar	Pewawancara
	Dr. Ir. Herni Sudarwati, MS NIP. 195402271983032001	Andrian Riza Hermawan 105060800111028

LAMPIRAN B SPESIFIKASI USE CASE

B.1 Use case melihat tabel nutrisi

Deskripsi umum

Use case ini diperuntukkan ke analis dan peternak praktis. Hasil dari *use case* ini adalah pengguna mampu melihat data nutrisi minimum yang harus dipenuhi untuk bisa menaikkan berat badan sapi pada berat badan dan jenis kelamin tertentu. Data dari tersebut akan dipakai dalam penyusunan ransum ideal.

Flow of events

Alur dasar

Use case dimulai saat pengguna memilih fungsi melihat tabel nutrisi harian

Muncul data nutrisi harian sapi

Pengguna dapat melihat data nutrisi minimal harian sapi yang harus dipenuhi untuk bisa menaikkan berat badan sapi pada berat badan tertentu.

Alur alternatif

Tidak ada.

Kebutuhan spesial

Tidak ada.

Pre-conditions

Tidak ada.

Post-conditions

Jika *use case* berhasil dijalankan, akan muncul data nutrisi minimal harian sapi yang harus dipenuhi untuk bisa menaikkan berat badan sapi pada berat badan tertentu. Jika *use case* tidak berhasil dijalankan, muncul notifikasi yang berisi bahwa data tidak ada.

Extension points

Tidak ada

B.2 Use case melihat tabel komposisi ransum

Deskripsi umum

Use case ini diperuntukkan ke analis dan peternak praktis. Hasil dari *use case* ini adalah pengguna mampu melihat nutrisi-nutrisi dari masing-masing bahan komposisi penyusun ransum. Data tersebut yang akan dipakai dalam penyusunan ransum ideal.

Flow of events

Alur dasar

Use case dimulai saat pengguna memilih fungsi lihat bahan penyusun ransum

Muncul data nutrisi harian sapi

Pengguna dapat melihat data nutrisi minimal harian sapi yang harus dipenuhi untuk bisa menaikkan berat badan sapi pada berat badan tertentu.

Alur alternatif

Tidak ada.

Kebutuhan spesial

Tidak ada.

Pre-conditions

Tidak ada.

Post-conditions

Jika *use case* berhasil dijalankan, akan muncul data nutrisi minimal harian sapi yang harus dipenuhi untuk bisa menaikkan berat badan sapi pada berat badan tertentu. Jika *use case* tidak berhasil dijalankan, muncul notifikasi yang berisi bahwa data tidak ada.

Extension points

Tidak ada

B.3 Use case menyusun ransum

Deskripsi umum

Use case ini digunakan untuk menyusun ransum oleh analis dan peternak praktis. Hasil dari *use case* ini adalah pengguna mendapatkan hasil penyusunan ransum dan menampilkan hasil penyusunan ransum secara detail. Khusus untuk analis juga bisa menyimpan hasil penyusunan ransum.

Flow of events

Alur dasar

Use case dimulai saat pengguna memilih fungsi penyusunan ransum.

1. Memilih kondisi fisiologi, penambahan berat badan per hari dan memilih bahan

Pengguna memasukkan fisiologi sapi potong (jenis kelamin dan berat badan), penambahan berat badan per hari dan memilih bahan (hijauan dan konsentrat) untuk dipakai pada proses penyusunan ransum

2. Menyusun ransum

Sistem melakukan penyusunan ransum menggunakan data fisiologi sapi potong (jenis kelamin dan berat badan), penambahan berat badan per hari dan memilih bahan (hijauan dan konsentrat) yang sudah dimasukkan

3. Menampilkan pemberitahuan tentang keberhasilan menyusun ransum

Sistem menampilkan pemberitahuan bahwa penyusunan ransum telah berhasil mendapatkan ransum ideal.

4. Menampilkan detail hasil penyusunan ransum

Sistem menampilkan detail hasil penyusunan ransum. Sistem juga menampilkan simulasi penambahan berat badan sapi potong hingga 12 bulan kedepan.

Alur alternatif

Salah memilih kondisi fisiologi, penambahan berat badan per hari dan memilih bahan

Di tahapan memilih kondisi fisiologi (jenis kelamin dan berat badan), penambahan berat badan per hari dan memilih bahan pada alur utama, Jika salah memilih kondisi fisiologi (jenis kelamin dan berat badan), penambahan berat badan per hari dan memilih bahan, sistem memberikan peringatan dan tidak bisa meneruskan tahapan berikutnya. Use case tetap berada di tahapan memilih kondisi fisiologi (jenis kelamin dan berat badan), penambahan berat badan per hari dan memilih bahan ke tahapan pada alur utama.

Tidak menemukan hasil penyusunan ransum ideal

Di tahapan menampilkan pemberitahuan tentang keberhasilan menyusun ransum pada alur utama, Jika hasil penyusunan ransum ideal tidak ditemukan, sistem menampilkan pemberitahuan bahwa tidak ditemukan ransum ideal. Use case tetap berada di tahapan menampilkan pemberitahuan tentang keberhasilan menyusun ransum pada alur utama.

Menyimpan hasil penyusunan ransum ideal

Di tahapan menampilkan detail hasil penyusunan ransum pada alur utama, Jika pengguna memilih fitur menyimpan hasil penyusunan ransum, sistem menyimpan hasil penyusunan ransum dan diarahkan ke daftar hasil penyusunan ransum tersimpan. Use case selesai / tidak dilanjutkan ke tahapan pada alur utama.

Kebutuhan spesial

Tidak ada.

Pre-conditions

Tidak ada.

Post-conditions

Jika use case berhasil dijalankan, hasil penyusunan ransum ditampilkan oleh sistem. Jika use case tidak berhasil dijalankan, hasil penyusunan ransum tidak didapat dan tidak ditampilkan

Extension points

Tidak ada

B.4 Use case mendaftarkan diri

Deskripsi umum

Use case ini diperuntukkan untuk analisis. Hasil dari use case ini adalah pengguna yang belum terdaftar menjadi terdaftar (*registered user*). *Username* yang dipakai bersifat unik sehingga tidak ada *username* yang sama.

Flow of events

Alur dasar

Use case dimulai saat pengguna yang belum terdaftar memilih fungsi mendaftarkan diri

1. **Memasukkan *username*, *password*, nama, dan *email* (surel)**
Pengguna yang belum terdaftar memasukkan *username*, *password*, nama, dan *email* (surel) pada tempat yang tersedia
2. **Mendaftarkan data yang telah dimasukkan ke sistem**
Sistem menyimpan data yang dimasukkan

Alur alternatif

***Username* sudah dipakai**

Di tahapan memasukkan *username*, *password*, nama, dan *email* (surel) pada alur utama, jika *username* yang dimasukkan sudah ada pada sistem, sistem memberikan notifikasi bahwa *username* telah dipakai. Use case dilanjutkan ke tahap memasukkan *username*, *password*, dan *email* (surel) pada alur utama.

***Password* tidak sama**

Di tahapan memasukkan *username*, *password*, dan *email* (surel) pada alur utama, jika *password* yang dimasukkan tidak sama, sistem memberikan notifikasi bahwa *password* tidak sama. Use case dilanjutkan ke tahap memasukkan *username*, *password*, dan *email* (surel) pada alur utama.

Kebutuhan spesial

Tidak ada.

Pre-conditions

Use case ini berjalan jika pengguna tidak dalam kondisi masuk ke sistem.

Post-conditions

Jika use case berhasil dijalankan, muncul notifikasi bahwa proses pendaftaran berhasil, data yang didaftarkan tersimpan pada sistem, dan diarahkan ke fungsi masuk (*login*). Jika use case tidak berhasil dijalankan, muncul notifikasi kesalahan dan data yang dimasukkan tidak tersimpan di sistem

Extension points

Tidak ada

B.5 Use case login

Deskripsi umum

Use case ini dipakai untuk masuk kedalam sistem bagi analis. Hasil dari *use case* ini adalah pengguna yang awalnya sebagai pengunjung biasa (*guest user*) menjadi *logged in user* dan bisa mengakses *use case* lain yang memerlukan syarat *logged in* / masuk ke sistem. Pengguna juga bisa memerintahkan sistem untuk mengingatkannya agar otomatis masuk ke sistem saat kunjungan berikutnya.

Flow of events

Alur dasar

Use case dimulai saat pengguna yang masih sebagai pengunjung biasa memilih fungsi masuk

1. Memasukkan *username* dan *password*

Pengguna yang belum *login* memasukkan *username* dan *password* pada tempat yang tersedia

2. Verifikasi data yang telah dimasukkan

Sistem melakukan verifikasi data *username* dan *password* yang dimasukkan dengan data *username* dan *password* yang tersimpan pada sistem

Alur alternatif

Verifikasi data gagal

Di tahapan verifikasi data yang telah dimasukkan pada alur utama, Jika tidak berhasil melakukan verifikasi, sistem mengarahkan ke beranda dengan kondisi pengguna tidak masuk / login ke sistem. Use case selesai / tidak dilanjutkan ke tahapan pada alur utama.

Mengingat pengguna

Di tahapan verifikasi data yang telah dimasukkan pada alur utama, Jika checkbox ingat saya dicentang, sistem menyimpan data yang terverifikasi agar otomatis masuk / login saat berkunjung kembali. Use case selesai / tidak dilanjutkan ke tahapan pada alur utama

Kebutuhan spesial

Tidak ada.

Pre-conditions

Use case ini berjalan jika pengguna tidak dalam kondisi masuk ke sistem.

Post-conditions

Jika *use case* berhasil dijalankan, pengguna masuk / *login* ke sistem. Jika *use case* tidak berhasil dijalankan, pengguna tetap menjadi pengunjung biasa (*guest user*).

Extension points

Tidak ada

B.6 Use case mengatur hasil penyusunan ransum

Deskripsi umum

Use case ini digunakan untuk mengatur hasil-hasil penyusunan ransum tersimpan di sistem oleh analis. Pengguna dapat melihat secara detail, mengubah (*update*) nama, dan menghapus hasil penyusunan ransum. Pengguna yang bisa memerintahkan sistem untuk menjalankan use case ini adalah pengguna yang sudah masuk (*login*) ke sistem.

Flow of events

Alur dasar

Use case dimulai saat pengguna yang sudah masuk ke sistem memilih fungsi mengatur hasil penyusunan ransum

1. Menampilkan daftar hasil penyusunan ransum

Sistem menampilkan daftar hasil penyusunan ransum yang sudah tersimpan sebelumnya.

2. Memilih fungsi *update* / ubah

Pengguna memilih untuk ubah / *update* hasil penyusunan ransum

3. Mengisi nama baru hasil penyusunan ransum

Pengguna mengisi nama hasil penyusunan ransum dengan data baru yang diinginkan pada tempat yang tersedia.

4. Mengubah (*update*) hasil penyusunan ransum

Sistem mengubah (*update*) hasil penyusunan ransum dengan data yang dimasukkan oleh pengguna

Alur alternatif

Tidak ditemukan hasil formulasi

Di tahapan menampilkan daftar hasil penyusunan ransum pada alur utama, Jika tidak ditemukan hasil formulasi yang tersimpan, sistem menampilkan tabel kosong. Use case tetap berada di tahap menampilkan daftar hasil penyusunan ransum.

Melihat hasil formulasi dengan detail

Di tahapan menampilkan daftar hasil penyusunan ransum pada alur utama, Jika pengguna memilih fitur melihat hasil penyusunan ransum secara detail, sistem menampilkan detail hasil penyusunan ransum. Use case berhenti / selesai.

Menghapus hasil formulasi

Di tahapan menampilkan daftar hasil penyusunan ransum pada alur utama, Jika pengguna memilih fitur menghapus hasil penyusunan ransum,

sistem menghapus hasil penyusunan ransum. Use case selesai / tidak dilanjutkan ke tahapan pada alur utama.

Kebutuhan spesial

Mesin cetak (*printer*) sanggup mencetak hasil penyusunan ransum dengan jelas saat menampilkan detail hasil penyusunan ransum.

Pre-conditions

Use case ini berjalan jika pengguna berada dikondisi masuk ke sistem.

Post-conditions

Jika *use case* berhasil dijalankan, pengguna berhasil melakukan manajemen hasil penyusunan ransum yang tersimpan di sistem. Jika *use case* tidak berhasil dijalankan, pengguna tidak berhasil melakukan manajemen hasil penyusunan ransum.

Extension points

Tidak ada

B.7 Use case membandingkan ransum

Deskripsi umum

Use case ini digunakan untuk membandingkan dua hasil penyusunan ransum yang sudah tersimpan di sistem agar bisa dilihat mana formulasi ransum yang lebih baik. Analis dapat memilih 2 formulasi ransum dan melihat secara detail perbandingan hasil penyusunan ransum.

Flow of events

Alur dasar

Use case dimulai saat pengguna yang sudah masuk ke sistem memilih fungsi membandingkan hasil penyusunan ransum

1. Menampilkan daftar hasil penyusunan ransum

Sistem menampilkan daftar hasil penyusunan ransum yang sudah tersimpan sebelumnya.

2. Memilih dua hasil penyusunan ransum

Pengguna memilih dua hasil penyusunan ransum untuk dibandingkan.

3. Menampilkan detail perbandingan 2 hasil penyusunan ransum

Sistem menampilkan detail perbandingan 2 hasil penyusunan ransum

Alur alternatif

Tidak ditemukan hasil formulasi

Di tahapan menampilkan daftar hasil penyusunan ransum pada alur utama, Jika tidak ditemukan hasil formulasi yang tersimpan, sistem menampilkan tabel kosong. Use case tetap berada di tahap menampilkan daftar hasil penyusunan ransum.

Memilih hasil ransum kurang dari dua

Di tahapan menampilkan daftar hasil penyusunan ransum pada alur utama, jika memilih kurang dari dua hasil formulasi yang tersimpan, sistem menampilkan peringatan bahwa diperlukan 2 hasil penyusunan ransum dengan data fisiologi (berat badan dan jenis kelamin) dan penambahan per hari yang sama untuk bisa melakukan perbandingan. Use case tetap berada di tahap menampilkan daftar hasil penyusunan ransum.

Memilih dua hasil ransum berbeda data fisiologi (berat badan dan jenis kelamin) dan penambahan per hari yang berbeda

Di tahapan menampilkan daftar hasil penyusunan ransum pada alur utama, jika memilih dua hasil ransum berbeda data fisiologi (berat badan dan jenis kelamin) dan penambahan per hari yang berbeda, sistem menampilkan peringatan bahwa diperlukan 2 hasil penyusunan ransum dengan data fisiologi (berat badan dan jenis kelamin) dan penambahan per hari yang sama untuk bisa melakukan perbandingan. Use case tetap berada di tahap menampilkan daftar hasil penyusunan ransum.

Kebutuhan spesial

Mesin cetak (*printer*) sanggup mencetak hasil penyusunan ransum dengan jelas saat menampilkan detail perbandingan hasil penyusunan ransum.

Pre-conditions

Use case ini berjalan jika pengguna berada dikondisi masuk ke sistem.

Post-conditions

Jika *use case* berhasil dijalankan, pengguna berhasil melihat detail perbandingan dua hasil penyusunan ransum yang tersimpan di sistem. Jika *use case* tidak berhasil dijalankan, pengguna tidak bisa melihat detail perbandingan dua hasil penyusunan ransum.

Extension points

Tidak ada

B.8 Use case logout

Deskripsi umum

Use case ini dipakai untuk keluar dari sistem bagi analis. Hasil dari *use case* ini adalah pengguna yang awalnya sebagai *logged in user* menjadi pengunjung biasa (*guest user*) dan tidak bisa mengakses *use case* lain yang memerlukan syarat *logged in* / masuk ke sistem.

Flow of events

Alur dasar

Use case dimulai saat pengguna yang sudah masuk ke sistem memilih fungsi keluar / *logout*

Melakukan proses *logout*

Sistem menjalankan proses *logout* atas pengguna yang sedang masuk dalam sistem

Alur alternatif

Tidak ada.

Kebutuhan spesial

Tidak ada.

Pre-conditions

Use case ini berjalan jika pengguna berada dikondisi masuk ke sistem.

Post-conditions

Jika *use case* berhasil dijalankan, pengguna berhasil keluar dari sistem. Jika *use case* tidak berhasil dijalankan, pengguna tidak berhasil keluar dari sistem.

Extension points

Tidak ada



LAMPIRAN C DAFTAR NUTRISI HARIAN SAPI POTONG

No.	Berat Badan (Kg)	Pertambahan Per Hari (Kg/hari)	Jenis Kelamin	BK (Kg)	PK (g)	TDN (Kg)	Ca (g)	P (g)
1	100	0	Jantan	2.2	167	1	5	5
2	100	0.25	Jantan	2.6	303	1.3	10	7
3	100	0.5	Jantan	3	379	1.6	15	9
4	100	0.75	Jantan	3.2	448	1.9	20	11
5	100	1	Jantan	3.3	541	2.2	25	15
6	150	0	Jantan	3	231	1.4	6	6
7	150	0.25	Jantan	3.8	400	1.8	12	9
8	150	0.5	Jantan	4.2	474	2.2	16	10
9	150	0.75	Jantan	4.4	589	2.6	21	13
10	150	1	Jantan	4.5	607	3	27	16
11	200	0	Jantan	3.7	285	1.8	6	6
12	200	0.25	Jantan	4.5	470	2.2	11	9
13	200	0.5	Jantan	5.2	554	2.8	16	12
14	200	0.75	Jantan	5.4	622	3.2	21	15
15	200	1	Jantan	5.6	690	3.7	27	17
16	200	1.1	Jantan	5.6	714	3.9	30	18
17	250	0	Jantan	4.4	337	2	9	9
18	250	0.25	Jantan	5.3	534	2.6	12	10
19	250	0.5	Jantan	6.2	623	3.2	16	14
20	250	0.75	Jantan	6.4	693	3.8	21	17
21	250	1	Jantan	6.6	760	4.3	28	19
22	250	1.1	Jantan	6.6	782	4.6	30	20
23	300	0	Jantan	5	385	2.4	10	10
24	300	0.25	Jantan	6	588	3	15	11
25	300	0.5	Jantan	7	679	3.7	19	14
26	300	0.75	Jantan	7.4	753	4.3	23	18
27	300	1	Jantan	7.5	819	5	28	21
28	300	1.1	Jantan	7.6	847	5.3	30	22
29	350	0	Jantan	5.7	432	2.6	12	12
30	350	0.25	Jantan	6.8	635	3.3	16	14
31	350	0.5	Jantan	7.9	731	4.1	20	16

No.	Berat Badan (Kg)	Pertambahan Per Hari (Kg/hari)	Jenis Kelamin	BK (Kg)	PK (g)	TDN (Kg)	Ca (g)	P (g)
32	350	0.75	Jantan	8.3	806	4.8	25	18
33	350	1	Jantan	8.5	874	5.6	30	21
34	350	1.1	Jantan	8.5	899	5.9	31	23
35	350	1.2	Jantan	8.5	923	6.2	32	24
36	400	0	Jantan	6.2	478	2.9	13	13
37	400	0.25	Jantan	7.5	664	3.7	16	15
38	400	0.5	Jantan	8.7	772	4.6	21	18
39	400	0.75	Jantan	9.1	875	5.4	26	21
40	400	1	Jantan	9.3	913	6.2	31	24
41	400	1.1	Jantan	9.4	942	6.6	32	25
42	400	1.2	Jantan	9.4	967	7	33	25
43	400	1.3	Jantan	9.3	988	7.2	33	26
44	450	0	Jantan	6.8	528	3.2	14	14
45	450	0.25	Jantan	8.2	703	4.1	18	17
46	450	0.5	Jantan	9.5	805	5	22	20
47	450	0.75	Jantan	10	911	5.9	26	23
48	450	1	Jantan	10.2	952	6.8	29	26
49	450	1.1	Jantan	10.2	975	7.2	30	27
50	450	1.2	Jantan	10.2	998	7.6	31	28
51	450	1.3	Jantan	10	1018	7.9	32	29
52	500	0	Jantan	7.4	567	3.4	15	15
53	500	0.25	Jantan	8.9	727	4.4	19	18
54	500	0.5	Jantan	10.3	831	5.4	23	21
55	500	0.75	Jantan	10.8	938	6.4	27	24
56	500	1	Jantan	11	975	7.4	30	27
57	500	1.1	Jantan	11.1	1005	7.8	30	27
58	500	1.2	Jantan	11	1026	8.2	31	28
59	500	1.3	Jantan	10.9	1048	8.6	32	28
60	500	1.4	Jantan	10.6	1063	9	33	29
61	100	0	Betina	2.4	178	1.1	4	4
62	100	0.25	Betina	2.9	321	1.3	13	10
63	100	0.5	Betina	3.1	391	1.7	14	11
64	100	0.75	Betina	3.2	460	2	20	14

No.	Berat Badan (Kg)	Pertambahan Per Hari (Kg/hari)	Jenis Kelamin	BK (Kg)	PK (g)	TDN (Kg)	Ca (g)	P (g)
65	100	1	Betina	3.3	527	2.3	26	18
66	150	0	Betina	3.3	234	1.6	5	5
67	150	0.25	Betina	4	414	1.9	13	11
68	150	0.5	Betina	4.2	513	2.3	14	12
69	150	0.75	Betina	4.4	552	2.7	19	15
70	150	1	Betina	4.5	623	3.1	25	18
71	200	0	Betina	4	299	1.8	6	6
72	200	0.25	Betina	4.9	492	2.3	10	10
73	200	0.5	Betina	5.6	577	2.8	14	13
74	200	0.75	Betina	5.5	639	3.3	19	16
75	200	1	Betina	5.6	707	3.8	23	18
76	250	0	Betina	4.8	264	2.1	7	7
77	250	0.25	Betina	5.8	486	2.7	12	12
78	250	0.5	Betina	6.2	564	3.3	13	13
79	250	0.75	Betina	6.5	644	3.9	18	15
80	250	1	Betina	6.6	724	4.5	23	18
81	250	1.1	Betina	6.6	757	4.8	25	20
82	300	0	Betina	5.5	303	2.4	9	9
83	300	0.25	Betina	6.7	526	3.1	13	13
84	300	0.5	Betina	7.1	604	3.8	14	14
85	300	0.75	Betina	7.4	717	4.5	17	15
86	300	1	Betina	7.6	764	5.2	21	18
87	300	1.1	Betina	7.3	797	6.1	24	20
88	350	0	Betina	6.1	340	2.7	10	10
89	350	0.25	Betina	7.4	557	3.5	15	15
90	350	0.5	Betina	8	637	4.3	15	15
91	350	0.75	Betina	8.3	717	5	15	15
92	350	1	Betina	8.5	797	5.8	18	18
93	350	1.1	Betina	8.5	829	6.1	20	19
94	350	1.2	Betina	8.4	860	6.4	21	20
95	400	0	Betina	6.8	377	3	11	11
96	400	0.25	Betina	8.3	579	3.9	15	15
97	400	0.5	Betina	8.8	657	4.7	15	15

No.	Berat Badan (Kg)	Pertambahan Per Hari (Kg/hari)	Jenis Kelamin	BK (Kg)	PK (g)	TDN (Kg)	Ca (g)	P (g)
98	400	0.75	Betina	9.2	739	5.6	16	16
99	400	1	Betina	9.4	819	6.5	18	18
100	400	1.1	Betina	9.4	850	6.8	19	19
101	400	1.2	Betina	9.2	883	7	20	19
102	450	0	Betina	7.4	441	3.3	12	12
103	450	0.25	Betina	9	590	4.2	16	16
104	450	0.5	Betina	9.6	671	5.1	17	17
105	450	0.75	Betina	10	750	6.1	18	18
106	450	1	Betina	10.2	831	7	19	19
107	450	1.1	Betina	10.2	857	7.4	20	20

Sumber: Kearl (1982)



LAMPIRAN D DAFTAR BAHAN PENYUSUN RANSUM

No.	Nama Bahan	Jenis Bahan	BK (%)	PK (%)	TDN (%)	Ca (%)	P (%)
1	Jerami padi segar	Konsentrat	40	4.3	40	0.21	0.05
2	Jerami padi kering	Konsentrat	86	3.7	39	0.2	0.04
3	Jerami jagung bagian atas segar	Konsentrat	28	8.2	5.7	0.54	0.11
4	Pucuk tebu	Konsentrat	39	4.5	56	0.39	0.06
5	Daging buah (pod) kakao kering	Konsentrat	92	10.06	47	0	0
6	Kulit buah kopi kering	Konsentrat	88	6.95	46	1.38	0.46
7	Bagas (ampas tebu)	Konsentrat	90	1.9	47	0	0
8	Kulit nanas	Konsentrat	12	14.5	68	0.26	0.09
9	Dedak kasar	Konsentrat	86	7.6	14	0.23	1.28
10	Grantek	Konsentrat	92	4.28	48	0	0
11	Ampas pati aren kering	Konsentrat	88	3.86	41	0	0
12	Ampas jeruk kering	Konsentrat	90	8.1	70	4.27	0.9
13	Kulit kacang	Konsentrat	82	4.9	45	0	0
14	Tongkol jagung	Konsentrat	90	2.61	49	0	0
15	Onggok	Konsentrat	86	1.61	80	0	0
16	Kulit kedelai	Konsentrat	89	20.3	56	0.59	0.17
17	Rumput ilalang	Hijauan	40	5.4	54	0.13	0.09
18	Rumput lapangan	Hijauan	31	9	57	0.23	0.26
19	Rumput gajah	Hijauan	21	8.3	50	0.59	0.29
20	Rumput pangola	Hijauan	18	9.2	60	0	0
21	Rumput benggala	Hijauan	27	7.7	50	0.52	0.22
22	Rumput raja	Hijauan	22	9.2	55	0	0
23	Rumput setaria	Hijauan	20	9.5	55	0.8	0.5
24	Rumput brachiaria	Hijauan	25	8.3	53	0.2	0.13
25	Rumput kolonjono	Hijauan	21	10.5	55	0.38	0.19
26	Rumput sudan	Hijauan	22	8.1	58	0	0
27	Sorgum segar	Hijauan	26	7.7	57	0	0
28	Rendeng	Hijauan	35	15.1	65	1.51	0.2
29	Cetro	Hijauan	25	22	61	1	0.36
30	Calopo	Hijauan	29	16	60	1.81	0.1

No.	Nama Bahan	Jenis Bahan	BK (%)	PK (%)	TDN (%)	Ca (%)	P (%)
31	Peuro	Hijauan	23	19.2	60	1.02	0.29
32	Siratro	Hijauan	21	16.6	49	0.85	0.17
33	Setilo	Hijauan	25	11.2	57	1.41	0.19
34	Alfalfa	Hijauan	23	16	76	1.89	0.24
35	Daun lamtoro	Hijauan	30	23.4	77	1.4	0.21
36	Daun gliriside	Hijauan	27	19.1	76	0.67	0.19
37	Daun turi	Hijauan	17	25	71	1.6	0.53
38	Daun kaliandra	Hijauan	23	17	69	1.13	0.71
39	Daun ketela pohon	Hijauan	26	20	71	0.99	0.56
40	Daun ketela rambat	Hijauan	24	11.3	74	1.79	0.24
41	Daun kangkung	Hijauan	15	18.8	73	0.71	0.32
42	Daun nangka	Hijauan	16	12.5	72	1.47	0.18
43	Daun kembang sepatu	Hijauan	19	17.8	67	1.67	0.52
44	Centrosema	Hijauan	24.1	16.8	60.2	1.2	0.38
45	Daun jagung	Hijauan	21	9.9	60	1.24	0.23
46	Daun kedelai	Hijauan	22.6	16.7	63.2	1.42	0.42
47	Daun kacang panjang	Hijauan	15	16.6	60.6	1.23	0.26
48	Daun kacang tanah	Hijauan	22.8	13.8	78.3	1.68	0.27
49	Daun tebu	Hijauan	25.5	5.2	51.4	0.47	0.34
50	Daun singkong	Hijauan	21.6	24.1	61.8	1.54	0.46
51	Daun ubi jalar	Hijauan	16.3	19.2	61.9	1.37	0.46
52	Daun pisang	Hijauan	23.3	16.6	73.5	0.57	0.18
53	Kulit pisang	Hijauan	13	7.1	59.1	0.33	0.21
54	Kulit singkong	Hijauan	30.6	6.6	73.1	0.33	0.21
55	Bekatul	Konsentrat	88	12.8	69.9	0.08	1.23
56	Dedak jagung	Konsentrat	87.8	10	82.3	0	0
57	Ampas tahu	Konsentrat	14.6	30.3	77.9	0	0
58	Bungkil kelapa	Konsentrat	88.6	21.3	78.7	0.17	0.62
59	Gaplek	Konsentrat	79.5	2.6	78.5	0.17	0.09
60	Singkong	Konsentrat	32.3	3.3	81.8	0.26	0.16
61	Ubi jalar	Konsentrat	32	3.2	83.9	0.28	0.23

Sumber: Yulianto (2012) dan Utomo (2012)

LAMPIRAN E HASIL PENGUJIAN VALIDASI IMPLEMENTASI PEARSON SQUARE

E.1 Detail hasil penyusunan ransum dan persentase kesalahan dari TC02

Detail hasil penyusunan ransum dari TC02 melalui aplikasi SunRan

Nama Bahan	Berat Bahan (Kg)	Bahan Kering / BK (Kg)	Protein Kasar / PK (Kg)	TDN (Kg)	Kalsium / Ca (gram)	Fosfor / P (gram)
Rumput gajah	15.4	3.234	0.268422	1.617	19.081	9.379
Dedak kasar	0.001	0.001	0.0001	0	0.003	0.017
Bungkil kelapa	1.089	0.965	0.205478	0.759	1.64	5.981
	JUMLAH	4.2	0.474	2.376	20.724	15.377
	KEBUTUHAN	4.2	0.474	2.2	16	10

Detail hasil penyusunan ransum dari TC02 melalui Microsoft Excel

	BK(Kg)	PK (Kg)	TDN (Kg)	Ca (gr)	P (gr)
Rumput gajah	3.234	0.268422	1.617	19.0806	9.3786
Dedak kasar	0.0013139	0.0000999	0.0001839	0.0030219	0.0168175
Bungkil kelapa	0.9646861	0.2054781	0.759208	1.6399664	5.981054
Total	4.2	0.474	2.3763919	20.7235883	15.3764715
Kebutuhan	4.2	0.474	2.2	16	10

Berat bahan penyusun dalam hasil penyusunan ransum dari TC02 melalui
Microsoft Excel

Bahan Segar	Berat (Kg)
Rumput gajah	15.4
Dedak kasar	0.0015278
Bungkil kelapa	1.0888105

Persentase kesalahan dari TC02

Nama Bahan	Kesalahan Berat Bahan (%)	Kesalahan Bahan Kering / BK (%)	Kesalahan Protein Kasar / PK (%)	Kesalahan TDN (%)	Kesalahan Kalsium / Ca (%)	Kesalahan Fosfor / P (%)
Rumput gajah	0	0	0	0	0.0020964	0.004265
Dedak kasar	34.5444444	23.8888889	0.1461988	100	0.7246377	1.0850694
Bungkil kelapa	0.0174014	0.0325358	0.000071	0.027395	0.0020474	0.0009031
JUMLAH	0	0	0	0.0164925	0.0019865	0.0034369

Rata-rata dari seluruh persentase kesalahan di atas adalah 6.4199148%.

E.2 Detail hasil penyusunan ransum dan persentase kesalahan dari TC03

Detail hasil penyusunan ransum dari TC03 melalui aplikasi SunRan

Nama Bahan	Berat Bahan (Kg)	Bahan Kering / BK (Kg)	Protein Kasar / PK (Kg)	TDN (Kg)	Kalsium / Ca (gram)	Fosfor / P (gram)
Rumput raja	15.273	3.36	0.30912	1.848	0	0
Dedak jagung	0.503	0.442	0.044158	0.363	0	0
Ampas tahu	2.726	0.398	0.120722	0.31	0	0
	JUMLAH	4.2	0.474	2.521	0	0
	KEBUTUHAN	4.2	0.474	2.2	16	10

Detail hasil penyusunan ransum dari TC03 melalui Microsoft Excel

	BK(Kg)	PK (Kg)	TDN (Kg)	Ca (gr)	P (gr)
Rumput raja	3.36	0.30912	1.848	0	0
Dedak jagung	0.4415764	0.0441576	0.3634173	0	0
Ampas tahu	0.3984236	0.1207224	0.310372	0	0
Total	4.2	0.474	2.5217894	0	0
Kebutuhan	4.2	0.474	2.2	16	10



Berat bahan penyusun dalam hasil penyusunan ransum dari TC03 melalui Microsoft Excel

Bahan Segar	Berat (Kg)
Rumput raja	15.27272727
Dedak jagung	0.502934345
Ampas tahu	2.728929078

Persentase kesalahan dari TC03

Nama Bahan	Kesalahan Berat Bahan (%)	Kesalahan Bahan Kering / BK (%)	Kesalahan Protein Kasar / PK (%)	Kesalahan TDN (%)	Kesalahan Kalsium / Ca (%)	Kesalahan Fosfor / P (%)
Rumput raja	0.0017857	0	0	0	0	0
Dedak jagung	0.0130544	0.0959393	0.0008255	0.1148376	0	0
Ampas tahu	0.1073343	0.1063304	0.000302	0.1198625	0	0
	JUMLAH	0	0	0.0313016	0	0

Rata-rata dari seluruh persentase kesalahan di atas adalah 0.0236629%.

E.3 Detail hasil penyusunan ransum dari dan persentase kesalahan TC04

Detail hasil penyusunan ransum dari TC04 melalui aplikasi SunRan

Nama Bahan	Berat Bahan (Kg)	Bahan Kering / BK (Kg)	Protein Kasar / PK (Kg)	TDN (Kg)	Kalsium / Ca (gram)	Fosfor / P (gram)
Rumput setaria	16.8	3.36	0.3192	1.848	26.88	16.8
Dedak jagung	0.243	0.213	0.021345	0.176	0	0
Bungkil kelapa	0.708	0.627	0.133455	0.493	1.065	3.885
	JUMLAH	4.2	0.474	2.517	27.945	20.685
	KEBUTUHAN	4.2	0.474	2.2	16	10

Detail hasil penyusunan ransum dari TC04 melalui Microsoft Excel

	BK(Kg)	PK (Kg)	TDN (Kg)	Ca (gr)	P (gr)
Rumput setaria	3.36	0.3192	1.848	26.88	16.8
Dedak jagung	0.2134513	0.0213451	0.1756704	0	0
Bungkil kelapa	0.6265487	0.1334549	0.4930938	1.0651327	3.8846018
Total	4.2	0.474	2.5167642	27.9451327	20.6846018
Kebutuhan	4.2	0.474	2.2	16	10

Berat bahan penyusun dalam hasil penyusunan ransum dari TC04 melalui Microsoft Excel

Bahan Segar	Berat (Kg)
Rumput setaria	16.8
Dedak jagung	0.2431109
Bungkil kelapa	0.7071655

Persentase kesalahan dari TC04

Nama Bahan	Kesalahan Berat Bahan (%)	Kesalahan Bahan Kering / BK (%)	Kesalahan Protein Kasar / PK (%)	Kesalahan TDN (%)	Kesalahan Kalsium / Ca (%)	Kesalahan Fosfor / P (%)
Rumput setaria	0	0	0	0	0	0
Dedak jagung	0.045597	0.2114428	0.0006219	0.1875999	0	0
Bungkil kelapa	0.118	0.0720339	0.0000995	0.0190238	0.0124626	0.0102515
	JUMLAH	0	0	0.0093673	0.000475	0.0019252

Rata-rata dari seluruh persentase kesalahan di atas adalah 0.027556%.

E.4 Detail hasil penyusunan ransum dan persentase kesalahan dari TC05

Detail hasil penyusunan ransum dari TC05 melalui aplikasi SunRan

Nama Bahan	Berat Bahan (Kg)	Bahan Kering / BK (Kg)	Protein Kasar / PK (Kg)	TDN (Kg)	Kalsium / Ca (gram)	Fosfor / P (gram)
Rumput kolonjono	16	3.36	0.3528	1.848	12.768	6.384
Tongkol jagung	0.31	0.279	0.007277	0.137	0	0
Kulit kedelai	0.63	0.561	0.113923	0.314	3.311	0.954
	JUMLAH	4.2	0.474	2.299	16.079	7.338
	KEBUTUHAN	4.2	0.474	2.2	16	10

Detail hasil penyusunan ransum dari TC05 melalui Microsoft Excel

	BK(Kg)	PK (Kg)	TDN (Kg)	Ca (gr)	P (gr)
Rumput kolonjono	3.36	0.3528	1.848	12.768	6.384
Tongkol jagung	0.2788016	0.0072767	0.1366128	0	0
Kulit kedelai	0.5611984	0.1139233	0.3142711	3.3110707	0.9540373
Total	4.2	0.474	2.2988839	16.0790707	7.3380373
Kebutuhan	4.2	0.474	2.2	16	10

Berat bahan penyusun dalam hasil penyusunan ransum dari TC05 melalui Microsoft Excel

Bahan Segar	Berat (Kg)
Rumput kolonjono	16
Tongkol jagung	0.3097795
Kulit kedelai	0.63056

Persentase kesalahan dari TC05

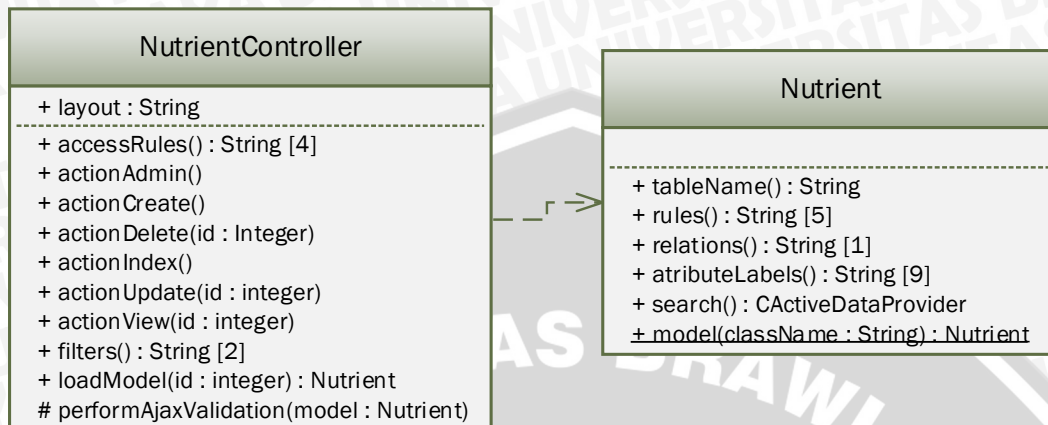
Nama Bahan	Kesalahan Berat Bahan (%)	Kesalahan Bahan Kering / BK (%)	Kesalahan Protein Kasar / PK (%)	Kesalahan TDN (%)	Kesalahan Kalsium / Ca (%)	Kesalahan Fosfor / P (%)
Rumput kolonjono	0	0	0	0	0	0
Tongkol jagung	0.0711679	0.0711679	0.0038299	0.2834467	0	0
Kulit kedelai	0.088813	0.035356	0.0002446	0.0862674	0.0021341	0.0039107
JUMLAH		0	0	0.0050507	0.0004395	0.0005084

Rata-rata dari seluruh persentase kesalahan di atas adalah 0.0260935%.

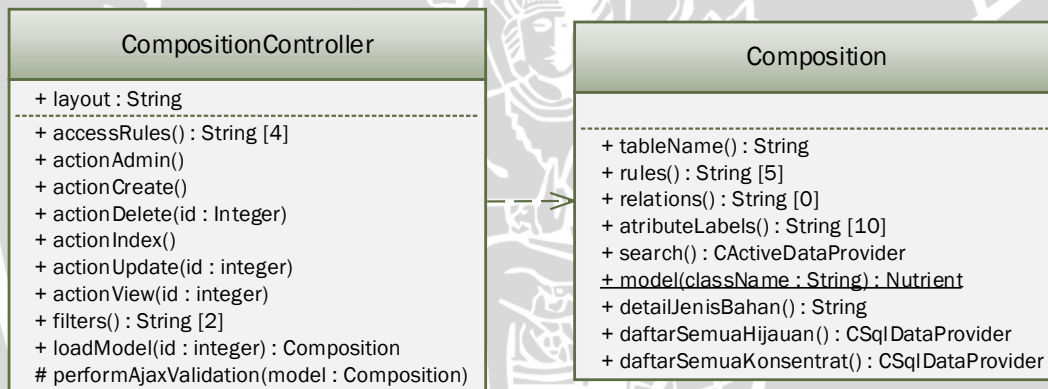


LAMPIRAN F KELAS *CONTROLLER* DAN *MODEL* TIAP FUNGSI

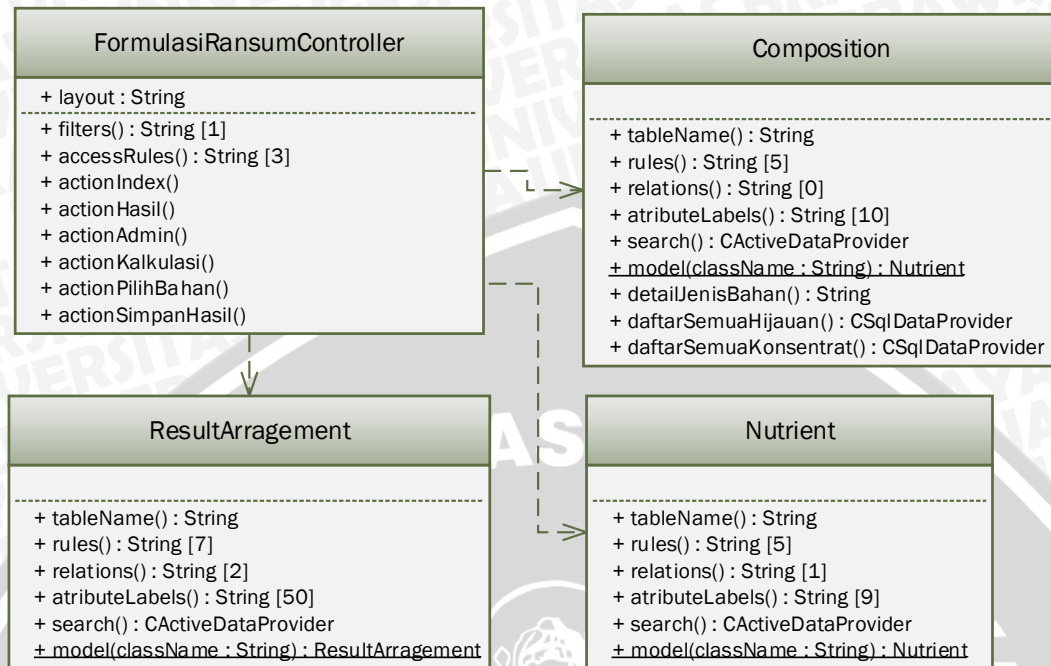
F.1 Kelas *controller* dan *model* pada fungsi melihat tabel nutrisi



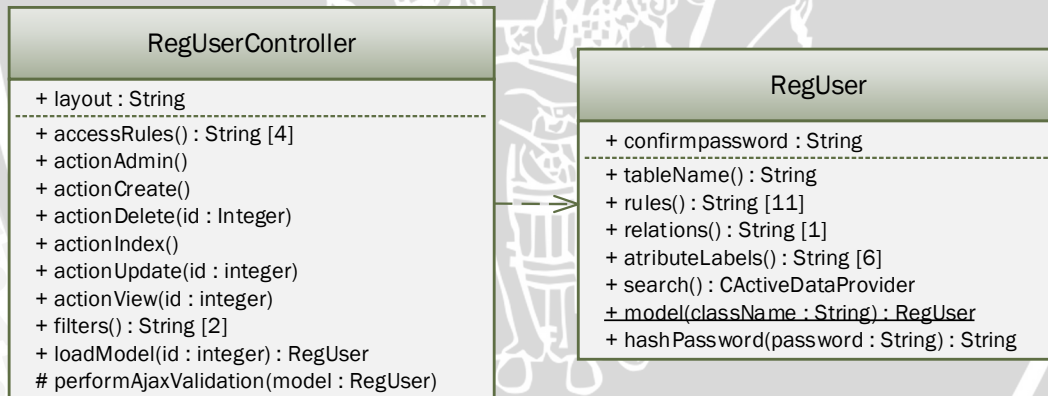
F.2 Kelas *controller* dan *model* pada fungsi melihat tabel komposisi



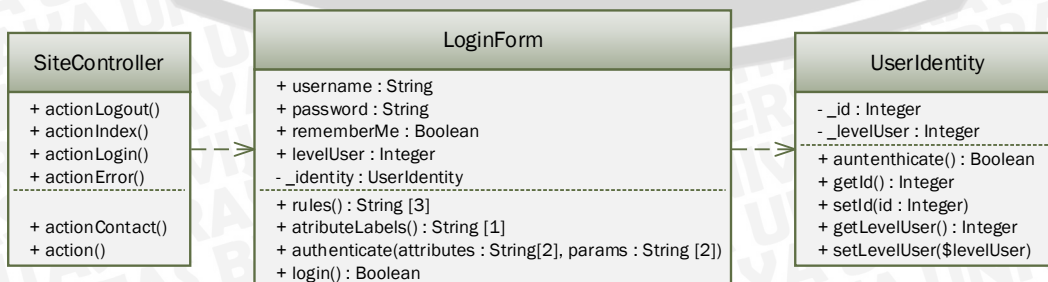
F.3 Kelas *controller* dan *model* pada fungsi menyusun ransum



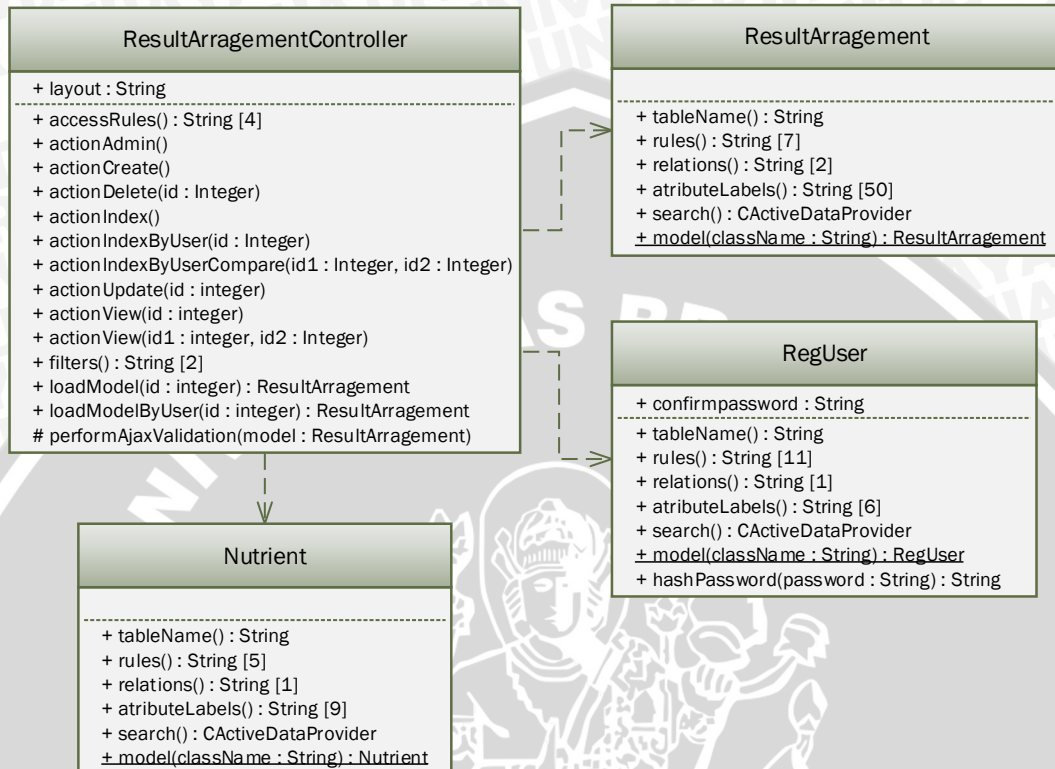
F.4 Kelas *controller* dan *model* pada fungsi mendaftarkan diri



F.5 Kelas *controller* dan *model* pada fungsi login



F.6 Kelas *controller* dan *model* pada fungsi mengatur dan membandingkan ransum



F.7 Kelas *controller* dan *model* pada fungsi *logout*

