



**SISTEM REKOMENDASI PEMILIHAN PRODUK BATIK PADA GRIYA  
BATIK DI KABUPATEN JEMBER MENGGUNAKAN  
METODE *WEIGHTED PRODUCT***

**SKRIPSI**

Oleh

**Bagus Ananda Permana**

**NIM 122410101004**

**PROGRAM STUDI SISTEM INFORMASI**

**UNIVERSITAS JEMBER**

**2017**



**SISTEM REKOMENDASI PEMILIHAN PRODUK BATIK PADA GRIYA  
BATIK DI KABUPATEN JEMBER MENGGUNAKAN  
METODE *WEIGHTED PRODUCT***

**SKRIPSI**

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat  
untuk menyelesaikan Program Studi Sistem Informasi (S1)  
dan mencapai gelar Sarjana Komputer

Oleh

**Bagus Ananda Permana**

**NIM 122410101004**

**PROGRAM STUDI SISTEM INFORMASI  
UNIVERSITAS JEMBER**

**2017**

## PERSEMBAHAN

Skripsi ini saya persembahkan untuk :

1. Almarhum Ayahanda Drs.Didik Fariadi dan Ibunda Renni Windarti
2. Kakak perempuan tercinta Shinta Anindya Ayu Oktavia beserta keluarga;
3. Adik perempuan tersayang Devi Permata Ayu Oktaria;
4. Harwinda Furqonia S yang selalu memberikan dukungan dan semangat;
5. Keluarga besar Ibu Moebanoe;
6. Sahabatku bersama dukungan dan doanya;
7. Guru-guruku sejak taman kanak-kanak sampai dengan perguruan tinggi;
8. Almamater Program Studi Sistem Informasi Universitas Jember.

**MOTO**

“Anda tidak bisa mengubah orang lain. Anda harus menjadi perubahan yang Anda harapkan dari orang lain”

(Mahatma Gandhi)



**PERNYATAAN**

Saya yang bertanda tangan di bawah ini:

Nama : Bagus Ananda Permana

NIM : 122410101004

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Sistem Rekomendasi Pemilihan Produk Batik pada Griya Batik di Kabupaten Jember Menggunakan Metode *Weighted Product*”, adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa ada tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika di kemudian hari pernyataan ini tidak benar.

Jember, 17 Maret 2017

Yang menyatakan,

Bagus Ananda Permana

NIM 122410101004

**SKRIPSI**

**SISTEM REKOMENDASI PEMILIHAN PRODUK BATIK PADA GRIYA  
BATIK DI KABUPATEN JEMBER MENGGUNAKAN  
METODE *WEIGHTED PRODUCT***

Oleh

**Bagus Ananda Permana**

**NIM 122410101004**

Pembimbing :

Dosen Pembimbing Utama : Drs. Antonius Cahya P, M.App., Sc., Ph.D

Dosen Pembimbing Pendamping : Nelly Oktavia Adiwijaya S.Si., MT.

**PENGESAHAN PEMBIMBING**

Skripsi berjudul “Sistem Rekomendasi Pemilihan Produk Batik pada Griya Batik di Kabupaten Jember Menggunakan Metode *Weighted Product*” telah diuji dan disahkan pada:

hari, tanggal : Jumat, 17 Maret 2017

tempat : Program Studi Sistem Informasi Universitas Jember.

Disetujui oleh:

Pembimbing I,

Pembimbing II,

Drs. Antonius Cahya P, M.App., Sc., Ph.D

NIP. 196909281993021001

Nelly Oktavia Adiwijaya S.Si., MT.

NIP. 198410242009122008

**PENGESAHAN PENGUJI**

Skripsi berjudul “Sistem Rekomendasi Pemilihan Produk Batik pada Griya Batik di Kabupaten Jember Menggunakan Metode *Weighted Product*” telah diuji dan disahkan pada:

hari, tanggal : Jumat, 17 Maret 2017

tempat : Program Studi Sistem Informasi Universitas Jember.

Tim Penguji,

Penguji I,

Penguji II,

Anang Andrianto, ST., MT  
NIP. 196906151997021002

Yanuar Nurdiansyah, ST., M.Cs  
NIP. 198201012010121004

Mengesahkan

Ketua Program Studi,

Prof. Drs. Slamin, M.Comp.Sc., Ph.D

NIP. 196704201992011001



## RINGKASAN

**Sistem Rekomendasi Pemilihan Produk Batik pada Griya Batik di Kabupaten Jember Menggunakan Metode *Weighted Product***; Bagus Ananda Permana, 122410101004; 2017; 143 halaman; Program Studi Sistem Informasi Universitas Jember.

Batik Jember merupakan sebuah karya yang berciri khas Jember. Batik Jember ini terkenal akan motif tembakau. Perkembangan motif Batik Jember juga menyesuaikan dengan ciri khas suatu daerah masing-masing yaitu kopi, kakao, buah naga, burung, pantai dan lain-lain. Hampir di setiap desa di Kabupaten Jember terdapat Griya Batik. Rata-rata Griya Batik yang terdapat di setiap desa memproduksi sendiri Batiknya mulai dari Batik tulis, semi tulis, hingga batik cap. Hasil produksi Griya Batik ini mulai banyak digunakan oleh warga jember, wisatawan lokal maupun mancanegara. Sayangnya hal tersebut tidak diikuti dengan kemudahan mencari batik yang sesuai dengan kriteria yang diinginkan. Sedangkan pemilik Griya Batik tidak menemukan media yang tepat untuk menginformasikan Griya Batiknya.

Sistem Rekomendasi Pemilihan Produk Batik pada Griya Batik di Kabupaten Jember Menggunakan Metode *Weighted Product* ini memberikan fasilitas kemudahan informasi bagi kedua belah pihak baik wisatawan maupun pemilik Griya Batik melalui media yang mudah diakses. Media tersebut adalah *website* yang dapat diakses dengan mudah dimana saja dengan hanya menggunakan koneksi internet. Selain itu, untuk memudahkan wisatawan dalam menentukan Batik yang sesuai dengan kriteria yang diinginkan, penulis membuat sistem pendukung keputusan dengan metode seleksi pengambilan keputusan *Weighted Product* (WP) dengan bahasa pemrograman PHP dan *MySQL* sebagai database. Dengan sistem pendukung keputusan diharapkan dapat membantu masyarakat Jember ataupun wisatawan dalam memilih Batik yang sesuai dengan kriteria yang diinginkan. Dari hasil penelitian yang dilakukan, aplikasi

dapat menjadi sebuah solusi sebagai media penghubung antara masyarakat dan pemilik Griya Batik untuk dapat menginformasikan Griya Batiknya.



## PRAKATA

Puji syukur kepada Allah SWT, atas segala limpahan rahmat, hidayat dan karuniaNya maka penulis dapat menyelesaikan skripsi yang berjudul “Sistem Rekomendasi Pemilihan Produk Batik pada Griya Batik di Kabupaten Jember Menggunakan Metode Weighted Product”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi Universitas Jember.

Penyusunan skripsi ini tidak lepas dari dukungan dan bantuan berbagai pihak. Oleh karena itu penulis menyampaikan terima kasih kepada:

1. Prof. Drs. Slamir, M.Comp.Sc., Ph.D., selaku Ketua Program Studi Sistem Informasi Universitas Jember;
2. Drs. Antonius Cahya P, M.App., Sc., Ph.D., selaku Dosen Pembimbing Utama dan Nelly Oktavia A, S.Si, MT., selaku Dosen Pembimbing Pendamping yang telah meluangkan waktu, pikiran dan perhatian dalam penulisan skripsi;
3. Seluruh Bapak dan Ibu dosen beserta staf karyawan di Program Studi Sistem Informasi Universitas Jember;
4. Ibunda Renni Windarti yang selalu mendukung dan mendoakan;
5. Kakak perempuan Shinta Anindya Ayu Oktavia beserta keluarga;
6. Adik perempuan Devi Permata Ayu Oktaria;
7. Keluarga besar Ibu Moebanoe;
8. Harwinda Furqonia Sobriani yang selalu mendukung, memberikan semangat dan motivasi;
9. Sahabat seperjuangan yang selalu menemani dan memberikan semangat serta doa Dimas Arifianto, Bagus Cahyo, Gede Bagoes, Riska Arimanudin, Rinaldi Dwi Sayoga, Yohanis Permadi, Tri Setyoko, Rizki kurniawan;
10. Angga Prastiyo;
11. Febrianto Rama Anji;

12. Teman-teman seperjuangan FORMATION angkatan 2012 dan semua mahasiswa Program Studi Sistem Informasi yang telah menjadi keluarga kecil bagi penulis selama menempuh pendidikan S1;
13. Keluarga besar Unit Kegiatan Mahasiswa Kesenian (UKMK) ETALASE periode 2014-2015, Yosafat Parulian, Moch.Fikri, Aglendy Rois, Dinda, Alfi dan anggota lain yang tidak dapat disebutkan satu persatu;
14. Chrisdianto (DUTA BATIK JEMBER)
15. Keluarga besar '*Second Story*' Firdaus Kurniawan, M.Alifi, Rayendra, Syahrizal, Dody Firdaus, Ipenk, Ugik, Danna, Arik, Cahya Anugerah, Jovan, Rizal, Nanda;
16. Keluarga besar *Crazy Squad*, Duta, Kofrok, Candra, Kober, Gevi, Silvy, Sonia, Patkay dan lainnya yang tidak dapat disebutkan satu persatu;
17. Semua pihak yang tidak dapat disebutkan satu persatu;

Dengan harapan bahwa penelitian ini nantinya akan terus berlanjut dan berkembang kelak, penulis juga menerima segala kritik dan saran dari semua pihak demi kesempurnaan skripsi ini. Akhirnya penulis berharap, semoga skripsi ini dapat bermanfaat.

Jember, 17 Maret 2017

Penulis

**DAFTAR ISI**

	Halaman
<b>SKRIPSI</b> .....	i
<b>PERSEMBAHAN</b> .....	ii
<b>MOTO</b> .....	iii
<b>PERNYATAAN</b> .....	iv
<b>SKRIPSI</b> .....	v
<b>PENGESAHAN PEMBIMBING</b> .....	vi
<b>PENGESAHAN PENGUJI</b> .....	vii
<b>RINGKASAN</b> .....	viii
<b>PRAKATA</b> .....	x
<b>DAFTAR ISI</b> .....	xii
<b>DAFTAR TABEL</b> .....	xvii
<b>DAFTAR GAMBAR</b> .....	xix
<b>BAB I. PENDAHULUAN</b> .....	1
<b>1.1 Latar Belakang</b> .....	1
<b>1.2 Rumusan Masalah</b> .....	2
<b>1.3 Tujuan dan Manfaat</b> .....	3
1.3.1 Tujuan.....	3
1.3.2 Manfaat.....	3
<b>1.4 Batasan Masalah</b> .....	4
<b>1.5 Sistematika Penulisan</b> .....	4

<b>BAB 2. TINJAUAN PUSTAKA</b> .....	6
<b>2.1 Penelitian Terdahulu</b> .....	6
<b>2.2 Weighted Product (WP)</b> .....	7
<b>BAB 3. METODOLOGI PENELITIAN</b> .....	11
<b>3.1 Jenis Penelitian</b> .....	11
<b>3.2 Tahapan Penelitian</b> .....	11
3.4.1 Tahapan Analisis Kebutuhan .....	12
3.4.2 Tahapan Desain Sistem .....	13
3.4.3 Tahapan Implementasi Sistem.....	15
3.4.4 Tahapan Pengujian Sistem .....	15
3.4.5 Tahapan Pemeliharaan Sistem .....	16
<b>BAB 4. PENGEMBANGAN SISTEM</b> .....	17
<b>4.1 Deskripsi Umum Sistem</b> .....	17
4.1.1 SOP ( <i>Statement of purpose</i> ).....	17
<b>4.2 Pengumpulan Data</b> .....	18
<b>4.3 Analisis Kebutuhan</b> .....	18
4.2.1 Kebutuhan Fungsional.....	18
4.2.2 Kebutuhan Non-Fungsional .....	18
<b>4.4 Desain Sistem</b> .....	19
4.3.1. <i>Business Process</i> .....	19
4.3.2. <i>Use Case Diagram</i> .....	20
4.3.3. <i>Use Case Skenario</i> .....	23
4.3.4. <i>Activity Diagram</i> .....	26

4.3.5. <i>Sequence</i> Diagram.....	29
4.3.6. <i>Class</i> Diagram.....	31
4.3.7. <i>Entity Relationship</i> Diagram (ERD) .....	33
<b>4.4 Implementasi</b> .....	<b>33</b>
<b>4.5 Pengujian</b> .....	<b>39</b>
4.5.1. Metode <i>White Box</i> .....	39
4.5.2. Metode <i>Black Box</i> .....	49
<b>BAB 5. HASIL DAN PEMBAHASAN</b> .....	<b>50</b>
<b>5.1 Hasil Pembuatan Sistem Rekomendasi Pemilihan Batik pada Griya Batik di Kabupaten Jember Menggunakan Metode <i>Weighted Product</i></b> .....	<b>50</b>
5.1.1 Tampilan Awal Sistem.....	50
5.1.2 Fitur Rekomendasi Pemilihan Produk Batik.....	51
5.1.3 Fitur <i>Login</i> .....	52
5.1.4 Fitur Mendaftar Griya Batik.....	54
5.1.5 Fitur Mengelola Griya Batik .....	55
5.1.6 Fitur Mengelola Data Batik.....	55
5.1.7 Fitur Mengelola Data <i>Admin</i> .....	57
5.1.8 Fitur Mengkonfirmasi Griya Batik.....	58
<b>5.2 Hasil Penerapan Perhitungan Metode <i>Weighted Product</i> (WP)</b> .....	<b>58</b>
5.2.1 Data Kriteria Batik .....	58
5.2.2 Data Batik pada Griya Batik .....	59
5.2.3 Penentuan <i>Range Weighted Product</i> .....	60
5.2.4 Data Nilai <i>Weighted Product</i> (Pakar).....	61

<b>5.3</b>	<b>Pengujian Sistem Rekomendasi Pemilihan Batik pada Griya Batik di Kabupaten Jember Menggunakan Metode <i>Weighted Product</i></b> .....	62
5.3.1	Kasus 1 .....	62
5.3.2	Kasus 2 .....	64
5.3.3	Hasil Perbandingan .....	67
<b>5.4</b>	<b>Implementasi Metode <i>Weighted Product</i> dalam Sistem Rekomendasi Pemilihan Batik pada Griya Batik di Kabupaten Jember</b> .....	70
<b>5.5</b>	<b>Pembahasan Pada Sistem Rekomendasi Pemilihan Batik pada Griya Batik di Kabupaten Jember</b> .....	73
5.5.1	Kelebihan Sistem.....	74
5.5.2	Kelemahan Sistem.....	74
<b>BAB 6. PENUTUP</b>	.....	75
<b>6.1</b>	<b>Kesimpulan</b> .....	75
<b>6.2</b>	<b>Saran</b> .....	76
<b>DAFTAR PUSTAKA</b>	.....	77
<b>LAMPIRAN</b>	.....	79
<b>LAMPIRAN A. <i>USE CASE</i> SKENARIO</b>	.....	79
A.1	<i>Use Case</i> Skenario Konfirmasi Griya Batik .....	79
A.2	<i>Use Case</i> Skenario Mengelola Data Admin.....	80
A.3	<i>Use Case</i> Skenario Mendaftar Griya Batik .....	81
A.4	<i>Use Case</i> Skenario Mengelola Griya Batik.....	82
A.5	<i>Use Case</i> Skenario Mengelola Data Batik .....	83
<b>LAMPIRAN B. <i>ACTIVITY DIAGRAM</i></b>	.....	85
B.1	<i>Activity Diagram</i> Mengkonfirmasi Griya Batik .....	85

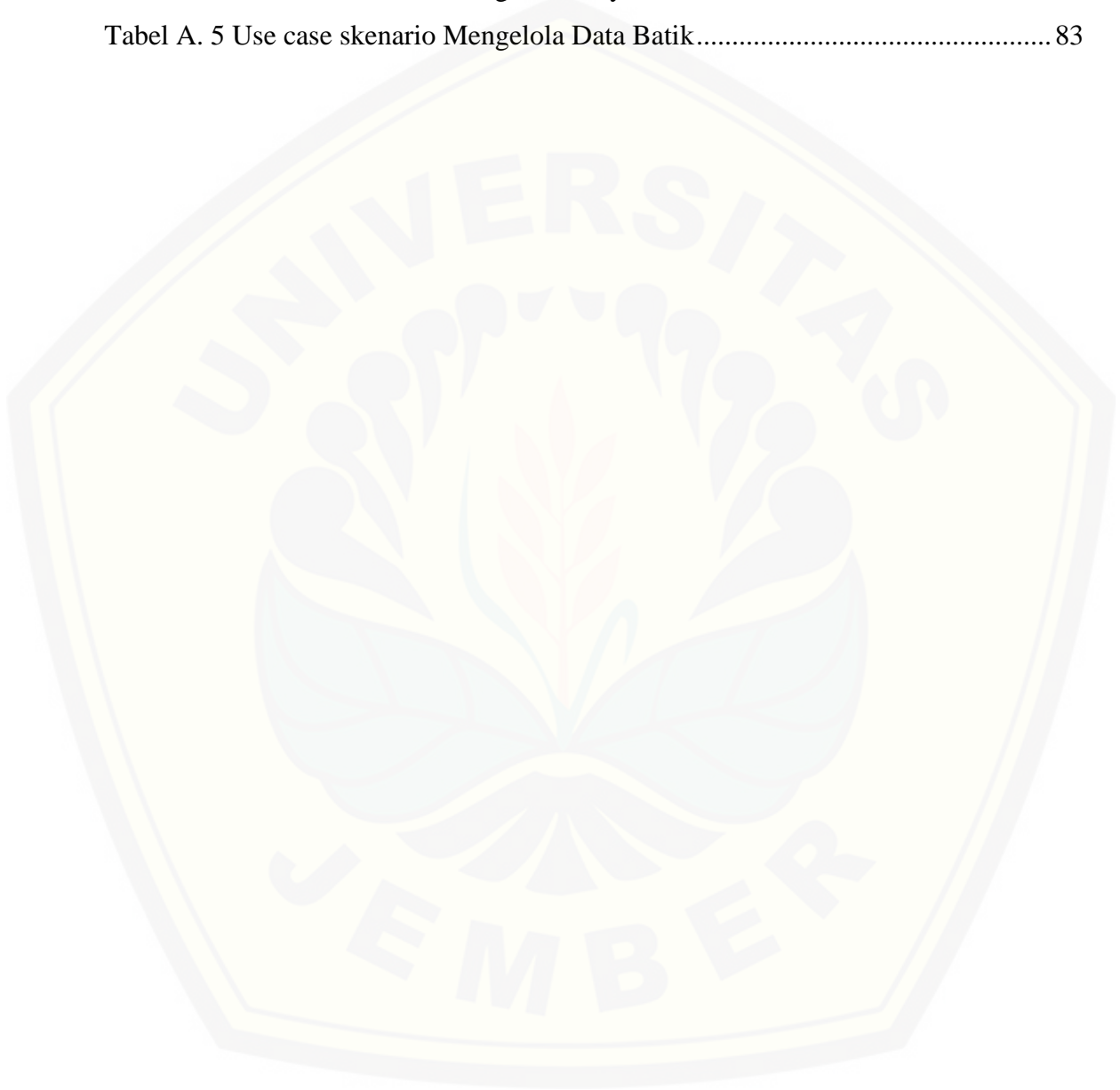


B.2	<i>Activity Diagram</i> Mengelola Data Admin .....	86
B.3	<i>Activity Diagram</i> Mendaftar Griya Batik .....	88
B.4	<i>Activity Diagram</i> Mengelola Griya Batik.....	88
B.5	<i>Activity Diagram</i> Mengelola Data Batik .....	89
<b>LAMPIRAN C. <i>SEQUENCE DIAGRAM</i></b> .....		91
C.1	<i>Sequence Diagram</i> Mengkonfirmasi Griya Batik .....	91
C.2	<i>Sequence Diagram</i> Mengelola Admin.....	91
C.3	<i>Sequence Diagram</i> Mengelola Griya Batik.....	92
C.4	<i>Sequence Diagram</i> Mengelola Data Batik .....	92
<b>LAMPIRAN D. KODE PROGRAM</b> .....		93
D.1.	Kode program <i>class controller</i> /Dashboard .....	93
D.2.	Kode program <i>class controller</i> /Data_batik.....	94
D.3.	Kode program <i>class controller</i> /Disease .....	96
D.4.	Kode program <i>class controller</i> /Griya .....	98
D.5.	Kode program <i>class controller</i> /Home.....	99
D.6.	Kode program <i>class controller</i> /Login .....	101
D.7.	Kode program <i>class model</i> /M_dashboard.....	104
D.8.	Kode program <i>class model</i> /M_data_batik.....	106
D.9.	Kode program <i>class model</i> /M_disease.....	109
D.10.	Kode program <i>class model</i> /M_griya.....	111
D.11.	Kode program <i>class model</i> /M_home .....	112
D.12.	Kode program <i>class model</i> /M_login.....	116
<b>LAMPIRAN E. PENGUJIAN <i>BLACK BOX</i></b> .....		120

DAFTAR TABEL

	Halaman
Tabel 2. 1 Contoh Tabel Kriteria dan Penilaian.....	7
Tabel 2. 2 Tabel Pembobotan Berdasarkan Tingkat.....	8
Tabel 4. 1 Data Kriteria.....	<b>Error! Bookmark not defined.</b>
Tabel 4. 2 Uji Validitas.....	<b>Error! Bookmark not defined.</b>
Tabel 4. 3 Uji Realibilitas.....	<b>Error! Bookmark not defined.</b>
Tabel 4.4 Deskripsi pembagian aktor sistem.....	22
Tabel 4.5 Deskripsi use case sistem.....	22
Tabel 4. 6 Usecase Skenario Memilih Batik.....	24
Tabel 4.7 Kode program function index().....	34
Tabel 4.8 Kode program function do_wp().....	35
Tabel 4. 9 Test case function do_wp().....	45
Tabel 5. 1 Data Kriteria Batik.....	58
Tabel 5. 2 Data Batik.....	59
Tabel 5. 3 Tabel Data Griya Batik.....	60
Tabel 5. 4 Nilai Tingkat Kepentingan.....	60
Tabel 5. 5 Range Nilai Tiap Subkriteria.....	60
Tabel 5. 6 Data Pembobotan Pakar.....	61
Tabel 5. 7 Data Pemilihan Kriteria Batik User Pertama.....	62
Tabel 5. 8 Tabel Cost dan Benefit User.....	62
Tabel 5. 9 Tabel Perhitungan Vektor S dan Vektor V.....	63
Tabel 5. 10 Data Pemilihan Kriteria Batik User Pertama.....	65
Tabel 5. 11 Tabel Cost dan Benefit User.....	65
Tabel 5. 12 Tabel Perhitungan Vektor S dan Vektor V.....	65
Tabel 5. 13 Perbandingan Perhitungan Manual dengan Sistem pada Kasus 1.....	67
Tabel 5. 14 Perbandingan Perhitungan Manual dengan Sistem pada Kasus 2.....	68
Tabel A. 1 Use case Konfirmasi Griya Batik.....	79

Tabel A. 2 Use case skenario Mengelola Data Admin .....	80
Tabel A. 3 Use case skenario Mendaftar Griya Batik.....	81
Tabel A. 4 Use case skenario Mengelola Griya Batik .....	82
Tabel A. 5 Use case skenario Mengelola Data Batik.....	83



## DAFTAR GAMBAR

	Halaman
Gambar 2. 1 Tahapan Metode Weighted Product.....	9
Gambar 3. 1 Model Waterfall (Sommerville, 2001).....	11
Gambar 3. 2 Diagram Alir Penerapan Metode Weighted Product (WP).....	13
Gambar 4. 1 Business Process Sistem .....	20
Gambar 4. 2 Use Case Diagram.....	21
Gambar 4. 3 Activity Diagram Memilih Batik .....	28
Gambar 4. 4 Sequence Diagram Memilih Batik.....	30
Gambar 4. 5 Class diagram.....	32
Gambar 4. 6 Entity Relationship Diagram.....	33
Gambar 4. 7 Listing program function do_wp() (1) .....	41
Gambar 4. 8 Listing program function do_wp() (2) .....	42
Gambar 4. 9 Listing program function do_wp() (3) .....	42
Gambar 4. 10 Listing program function do_wp() (4) .....	43
Gambar 4. 11 Listing program function do_wp() (5) .....	43
Gambar 4. 12 Diagram alir function do_wp ().....	44
Gambar 5. 1 Tampilan awal sistem.....	51
Gambar 5. 2 Tampilan Pengisian Nilai Bobot Setiap Kriteria Batik .....	51
Gambar 5. 3 Tampilan Rekomendasi Batik dari Sistem.....	52
Gambar 5. 4 Tampilan Login.....	52
Gambar 5. 5 Tampilan Homepage Owner .....	53
Gambar 5. 6 Tampilan <i>Homepage</i> Admin .....	53
Gambar 5. 7 Tampilan Halaman Mendaftar Griya Batik.....	54
Gambar 5. 8 Tampilan Halaman Ubah Griya Batik.....	55
Gambar 5. 9 Tampilan Halaman Menambah Batik .....	56
Gambar 5. 10 Tampilan Halaman Mengubah Batik .....	56
Gambar 5. 11 Tampilan Halaman Menghapus Data Batik .....	56

Gambar 5. 12 Tampilan Halaman Menambah Admin .....	57
Gambar 5. 13 Tampilan Halaman Mengubah Admin.....	57
Gambar 5. 14 Tampilan Halaman Konfirmasi Griya Batik .....	58
Gambar 5. 15 Kode Program metode weighted product (1) .....	71
Gambar 5. 16 Kode Program metode weighted product (2) .....	71
Gambar 5. 17 Kode Program metode weighted product (3) .....	72
Gambar 5. 18 Kode Program metode weighted product (4) .....	72
Gambar 5. 19 Kode Program metode weighted product (5) .....	73
Gambar 5. 20 Kode Program metode weighted product (6) .....	73
Gambar C. 1 Sequence diagram Mengkonfirmasi Griya Batik .....	91
Gambar C. 2 Sequence diagram Mengelola Admin.....	91
Gambar C. 3 Sequence diagram mengelola griya batik.....	92
Gambar C. 4 Sequence diagram mengelola data batik .....	92

## BAB I. PENDAHULUAN

Bab ini merupakan bab awal dari laporan tugas akhir. Pada bab ini akan dibahas tentang latar belakang, perumusan masalah, tujuan dan manfaat, batasan masalah, dan sistematika penulisan.

### 1.1 Latar Belakang

Batik Jember merupakan sebuah karya yang berciri budaya khas Jember. Batik Jember ini terkenal akan motif tembakau, yang merupakan salah satu komoditi yang paling banyak dihasilkan. Motif batik Jember ini juga menyesuaikan dengan perkembangan *trend* budaya lokal. Perkembangan diikuti dengan motif yang lebih banyak yaitu kakao, buah naga, kopi, bambu, dan motif burung berdasarkan ciri khas dari suatu daerah atau Kecamatan. Banyak pengrajin bahkan di setiap Kecamatan di Kabupaten Jember, berlomba lomba mengembangkan produk Batik Jember untuk nantinya dapat dipasarkan dan dikenal oleh masyarakat luas. Pengekspresian dari motif batik merupakan kesesuaian antara ciri khas yang dihasilkan dari Kabupaten Jember, kemudian akan menjadi sebuah inspirasi yang dapat dikembangkan.

Informasi dan pemasaran produk batik Jember masih belum ada kemajuan dan kurang dikenal oleh masyarakat luas, sehingga masyarakat jember sendiri enggan menggunakan batik lokal karya pengrajin lokal sehingga promosi batik Jember kurang optimal ditingkat lokal, nasional dan internasional (Solichah, 2009). Salah satu upaya untuk meningkatkan daya tarik masyarakat yaitu dengan mengikuti perkembangan *trend* yang disukai konsumen. Informasi mengenai Batik Jember dapat dilakukan dengan misalnya, pameran, koran, ataupun media yang berbasis *online*. Apabila dilihat lebih spesifik setiap cara memiliki kekurangan yaitu tidak setiap orang dapat mengetahui dengan jelas, kapan akan diadakan pameran, tidak tentu berlangganan koran ataupun terhubung dengan internet. Apabila dilihat dari segi efisiensi lebih menguntungkan menggunakan media berbasis *online* karena tidak terbatas oleh waktu dan kapanpun dapat diakses.

Banyak Griya Batik tersebar diseluruh Kabupaten Jember yang menyajikan macam macam produk batik yang dapat dilihat dari segi harga, motif atau corak batik dan bahan kain yang menjadi ciri khas dari masing masing Griya Batik. Namun, kebanyakan masyarakat bingung dalam memilih produk batik yang sesuai dengan keinginan dan kebutuhan. Selain karena banyaknya jenis batik, jarak juga menjadi penghalang konsumen untuk menjangkau pada setiap Griya Batik yang ada di Kabupaten Jember. Maka tidak menutup kemungkinan griya batik yang berada jauh dari kota yang kurang dalam informasi dan promosi akan berdampak pada minimnya angka penjualan.

Dari kondisi tersebut, maka perlu diadakan suatu penelitian untuk membuat sistem yang dapat memudahkan masyarakat atau konsumen dalam memilih batik dari seluruh griya batik yang ada di Kabupaten Jember, dengan kriteria yang sesuai dengan keinginan dan kebutuhan. Pemilihan batik yang dilakukan ini mampu mendukung pihak konsumen atau masyarakat dalam mengambil keputusan secara lebih efektif dan efisien dalam memilih produk batik di seluruh Griya Batik. Griya Batik juga lebih mudah dalam mencangkup pasar tanpa perlu khawatir kurangnya promosi. Pada penelitian ini dibutuhkan metode *Weighted Product* (WP) yang digunakan untuk menggolongkan produk batik ke dalam masalah yang bersifat *multicriteria* (ada banyak kriteria untuk mencapai tujuan), dengan konsumen atau masyarakat yang dapat menentukan sendiri bobot kepentingan dari masing-masing kriteria yang sesuai dengan keinginan dan kebutuhan yang ada.

## 1.2 Rumusan Masalah

Berdasarkan dari beberapa permasalahan yang telah diuraikan diatas, maka dapat diambil rumusan masalah sebagai berikut :

- a. Bagaimana mengimplementasikan metode *Weighted Product* untuk rekomendasi pemilihan produk batik pada griya batik di Kabupaten Jember?
- b. Bagaimana membangun sistem rekomendasi pemilihan produk batik pada griya batik di Kabupaten Jember menggunakan metode *weighted product* ?

### 1.3 Tujuan dan Manfaat

Berikut merupakan tujuan yang ingin dicapai dan manfaat yang ingin diperoleh dalam penelitian ini.

#### 1.3.1 Tujuan

Adapun tujuan yang ingin dicapai dalam penelitian ini adalah:

- a. Mengimplementasikan metode *Weighted Product* dalam sistem rekomendasi pemilihan produk batik pada griya batik di Kabupaten Jember.
- b. Membangun sistem rekomendasi pemilihan produk batik pada griya batik di Kabupaten Jember menggunakan metode *Weighted Product*.

#### 1.3.2 Manfaat

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

- a. Bagi Instansi

Sebagai masukan bagi instansi tentang produk batik yang menjadi pilihan wisatawan yang banyak diminati,

- b. Bagi Akademis

Hasil penelitian ini diharapkan dapat memberikan kontribusi dan masukan bagi siapa saja yang membutuhkan informasi yang berhubungan dengan judul penelitian ini. Selain itu, hasil penelitian ini merupakan suatu upaya untuk menambah varian judul penelitian yang ada di Program Studi Sistem Informasi Universitas Jember.

- c. Bagi Peneliti

Mengetahui bagaimana proses penerapan *weighted product* pada sistem rekomendasi pemilihan produk batik pada griya batik di Kabupaten Jember.

- d. Bagi pihak lain

Penelitian ini dapat dijadikan bahan referensi bagi peneliti lain yang ingin mengembangkan hasil penelitian ini di kemudian hari.



#### 1.4 Batasan Masalah

Terdapat beberapa batasan masalah yang diangkat sebagai parameter pengerjaan penelitian ini diantaranya sebagai berikut :

- a. Objek penelitian adalah batik tulis.
- b. Sistem yang dibangun berbasis *web*.
- c. Sistem ini menggunakan metode *Weighted Product* (WP).

#### 1.5 Sistematika Penulisan

Adapun sistematika penulisan skripsi setelah Bab 1. Pendahuluan adalah sebagai berikut:

- a. Bab 2. Tinjauan Pustaka

Bab ini memaparkan tinjauan terhadap hasil-hasil penelitian terdahulu berkaitan dengan masalah yang dibahas, landasan materi dan konsep prediksi, dan kajian teori metode analisis data yang berkaitan dengan masalah dalam penelitian.

- b. Bab 3. Metodologi Penelitian

Bab ini menguraikan tentang tempat dan waktu penelitian, metode penelitian, metode pengumpulan data, metode analisis data, dan teknik pengembangan sistem yang digunakan dalam penelitian.

- c. Bab 4. Pengembangan Sistem

Bab ini berisi uraian tentang langkah-langkah yang ditempuh dalam proses menganalisis dan merancang sistem yang hendak dibangun meliputi desain, kode program, dan pengujian sistem.

- d. Bab 5. Hasil dan Pembahasan

Bab ini memaparkan secara rinci pemecahan masalah melalui analisis yang disajikan dalam bentuk deskripsi dibantu dengan ilustrasi berupa tabel dan gambar untuk memperjelas hasil penelitian.

- e. Bab 6. Penutup

Bab ini terdiri atas kesimpulan atas penelitian yang telah dilakukan dan saran untuk penelitian selanjutnya



## BAB 2. TINJAUAN PUSTAKA

Pada bagian ini dipaparkan tinjauan yang berkaitan dengan masalah yang dibahas, kajian teori yang berkaitan dengan masalah, dan juga penelitian-penelitian terdahulu.

### 2.1 Penelitian Terdahulu

Pada penelitian, metode *Weighted Product* (WP) digunakan untuk melakukan perangkingan calon penerima beasiswa Peningkatan Prestasi Akademik (Ningrum, 2011). Penelitian ini menjelaskan proses perangkingan calon penerima beasiswa PPA tahun 2011 di FMIPA UNS dengan menggunakan metode *Weighted Product*. *Weighted Product* adalah suatu metode yang digunakan untuk mencari alternatif optimal dengan kriteria tertentu. Inti dari metode *Weighted Product* adalah menentukan nilai bobot untuk setiap kriteria, kemudian dilanjutkan dengan proses perangkingan diperoleh dengan mengurutkan nilai preferensi dari yang terbesar sampai yang terkecil. Kriteria yang digunakan pada penelitian tersebut antara lain nilai rata-rata rapor SMA kelas 12, penghasilan orang tua, jumlah tanggungan keluarga, dan pekerjaan orang tua.

Pada penelitian lainnya metode WP digunakan untuk melakukan pemilihan *homestay* bagi wisatawan di kawasan Desa Wisata Tanoker (Zukarnain, 2015). Penelitian ini menjelaskan proses pemilihan *homestay* bagi wisatawan menggunakan metode WP (*Weighted Product*). Inti dari metode *Weighted Product* adalah menentukan nilai bobot untuk setiap kriteria, kemudian dilanjutkan dengan proses perangkingan yang akan menyeleksi alternatif yang sudah diberikan. Hasil perankingan diperoleh dengan mengurutkan nilai preferensi dari yang terbesar sampai yang terkecil. Pada penelitian ini, akan dikembangkan sistem serupa dengan menambahkan fitur lokasi yang nantinya akan terintegrasi dengan *Google Maps*

sehingga wisatawan mengetahui lokasi *homestay* yang sesuai dengan kriteria yang diinginkan.

Hal-hal yang dapat diadopsi dari penelitian diatas adalah penggunaan metode *Weighted Product* (WP). Dalam penggunaannya perlu ditambahkan jenis kriteria yang digunakan agar proses pencarian Batik Jember lebih spesifik dan sesuai dengan keinginan konsumen. Tujuannya adalah agar metode yang digunakan sesuai dengan kasus yang diteliti. Selain itu, pada penelitian ini, nilai kepentingan bobot preferensi ditentukan oleh konsumen sendiri karena setiap konsumen memiliki keinginan yang berbeda dalam memilih Batik Jember.

## 2.2 *Weighted Product* (WP)

Metode WP merupakan metode yang menggunakan perkalian guna menghubungkan rating atribut, dimana rating setiap atribut harus dipangkatkan dahulu dengan bobot atribut yang bersangkutan (Tan, 2013). Menentukan kriteria yang akan digunakan, kemudian memberi nilai dari setiap kriteria tersebut. Tabel 2.1 menjelaskan contoh dari beberapa kriteria yang digunakan berdasarkan nilai dari setiap kriteria.

Tabel 2. 1 Contoh Tabel Kriteria dan Penilaian

No	Kriteria	Nilai	Bobot
1	$A_1$	$W_{01}$	$W_1$
2	$A_2$	$W_{02}$	$W_2$
3	$A_3$	$W_{03}$	$W_3$
4	$A_4$	$W_{04}$	$W_4$
5	$A_5$	$W_{05}$	$W_5$
6	$A_n$	$W_{0n}$	$W_n$

Sumber: (Alfita, 2010)

Pada tabel 2.1 dijelaskan bahwa kriteria yang digunakan diasumsikan dengan kriteria  $A_1$ , sampai  $A_n$ . Kemudian rating setiap kiteria diasumsikan dengan nilai  $W_{01}$

sampai  $W_{0n}$  dan bobot dari masing-masing kriteria diasumsikan dengan  $W_1$  sampai  $W_n$ . Menurut Kusumadewi (2006:75) bobot yang diberikan pada setiap nilai kriteria ditentukan berdasarkan tingkat kepentingan setiap kriteria. Pembobotan diberikan kepada setiap kriteria berdasarkan tingkat kepentingan dapat dilihat pada Tabel 2.2.

Tabel 2. 2 Tabel Pembobotan Berdasarkan Tingkat

No	Tingkat Kepentingan	Bobot
1	Sangat Rendah	1
2	Rendah	2
3	Cukup	3
4	Tinggi	4
5	Sangat Tinggi	5

Sumber: (Kusumadewi *et al*, 2006)

Dalam proses perhitungan metode *Weighted Product* (WP) menurut Kusumadewi *et al.* (2006) memiliki tiga tahapan, yaitu:

- a. Menghitung nilai perbaikan bobot menggunakan persamaan 2.1.

$$W_j = \frac{W_0}{\sum W_0} \quad 2.1$$

Keterangan :

$W_0$  = Nilai perbaikan bobot *user*

$W_j$  = Nilai Bobot dari *user*

$\sum W_0$  = Jumlah nilai bobot dari *user*

- b. Menghitung nilai preferensi untuk alternatif  $A_n$  dengan vektor S menggunakan persamaan 2.2

$$S_i = \prod_{j=1}^n (X_{ij})^{w_j} \quad 2.2$$

Keterangan :

$S_i$  = Nilai prefensi alternatif  $A_n$  (Vektor S)

$W_j$  = Nilai perbaikan bobot *user* (Positif jika *benefit*/ Negatif jika *cost*)

$X_{ij}$  = Nilai rating kinerja atribut ke-i terhadap subkriteria ke-j

- c. Menghitung nilai preferensi relatif dari setiap alternatif dengan menggunakan persamaan 2.3.

$$V_i = \frac{\prod_{j=1}^n (X_{ij})^{w_j}}{\prod_{j=1}^n (X_{j*})^{w_j}} \quad 2.3$$

Keterangan :

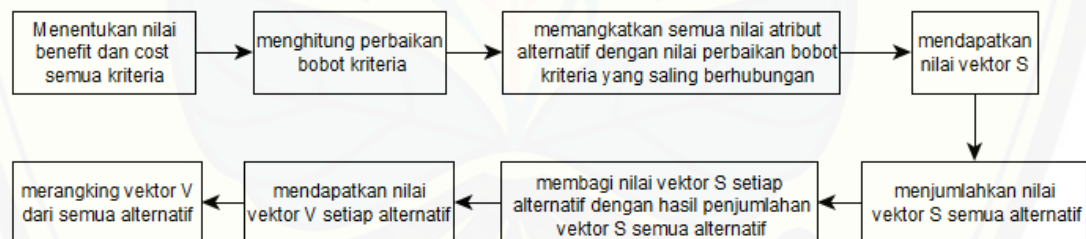
$V_i$  = Nilai preferensi relatif dari setiap alternatif (Vektor V)

$W_j$  = Nilai perbaikan bobot *user* (Positif jika *benefit*/ Negatif jika *cost*)

$X_{ij}$  = Nilai rating kinerja atribut ke-i terhadap subkriteria ke-j

$X_{j*}$  = Jumlah nilai preferensi alternatif (Vektor S)

Hasil dari nilai vektor V merupakan acuan yang digunakan untuk perbandingan dari alternatif yang dihitung. Nilai alternatif terbaik yang dipilih merupakan nilai vektor V terbesar dari setiap alternatif. Tahapan metode *Weighted Product* (WP) dapat dilihat pada Gambar 2.1.



Gambar 2. 1 Tahapan Metode *Weighted Product*

Tahapan metode *Weighted Product* (WP) pada Gambar 2.1, tahap awal yaitu menentukan nilai benefit dan cost semua kriteria. Kemudian menghitung perbaikan bobot kriteria, dimana apabila dijumlahkan harus sama dengan 1. Tahap selanjutnya, yaitu memangkatkan semua nilai atribut alternatif dengan nilai perbaikan bobot kriteria yang saling berhubungan. Pada tahap tersebut menghasilkan nilai vektor S. Seluruh nilai vektor S semua alternatif dijumlahkan. Lalu, membagi nilai vektor S 16

setiap alternatif dengan hasil penjumlahan vektor  $S$  semua alternatif. Setelah itu akan mendapatkan nilai vektor  $V$  setiap alternatif. Tahap terakhir yaitu meranking vektor  $V$  dari semua alternatif.



### BAB 3. METODOLOGI PENELITIAN

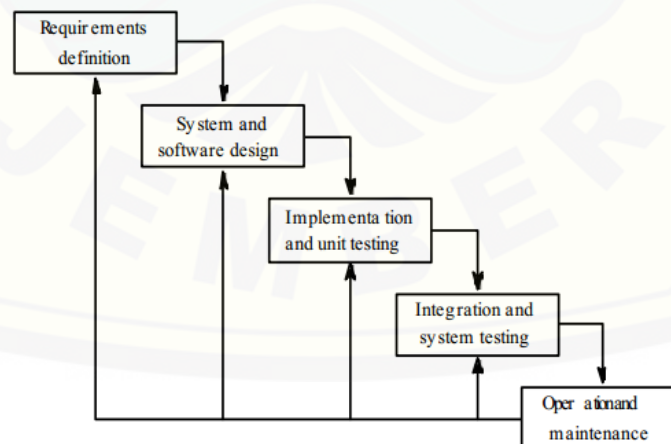
Bab ini menjelaskan tentang metode yang digunakan dalam merancang dan membangun sistem menggunakan metode pengembangan.

#### 3.1 Jenis Penelitian

Jenis penelitian yang digunakan pada penelitian ini merupakan jenis penelitian pengembangan, karena tujuan penelitian adalah untuk membangun sebuah sistem rekomendasi pemilihan produk Batik pada Griya Batik di Kabupaten Jember. Penelitian pengembangan bertujuan untuk membuat dan mengembangkan suatu produk yang efektif untuk digunakan. Penelitian ini bukan jenis penelitian yang ditunjukkan untuk menemukan teori atau menguji kebenaran dari suatu teori dalam bentuk eksperimentasi. Untuk mendukung konstruksi sistem ini, maka studi kasus dilakukan di 6 Griya Batik tulis di Kabupaten Jember.

#### 3.2 Tahapan Penelitian

Tahapan penelitian pengembangan sistem mengadopsi dari model *waterfall* dengan tahapan-tahapan seperti pada gambar berikut:



Gambar 3. 1 Model Waterfall (Sommerville, 2001)



### 3.4.1 Tahapan Analisis Kebutuhan

Analisis kebutuhan merupakan tahap untuk mengumpulkan data, informasi, serta mencari kebutuhan fungsional dan non fungsional sistem. Pada tahap ini, peneliti mencari permasalahan yang ada untuk dapat dianalisis kebutuhan yang diperlukan, sebagai solusi dari permasalahan yang muncul. Data-data yang telah didapat kemudian dikelompokkan menjadi kebutuhan fungsional dan non-fungsional sistem. Metode yang digunakan adalah sebagai berikut:

#### a. Metode Pengumpulan Data

Pengumpulan data dilakukan untuk memperoleh informasi yang dibutuhkan dalam mencapai tujuan penelitian. Pengumpulan data pada penelitian ini dilakukan melalui beberapa teknik pengumpulan data yaitu:

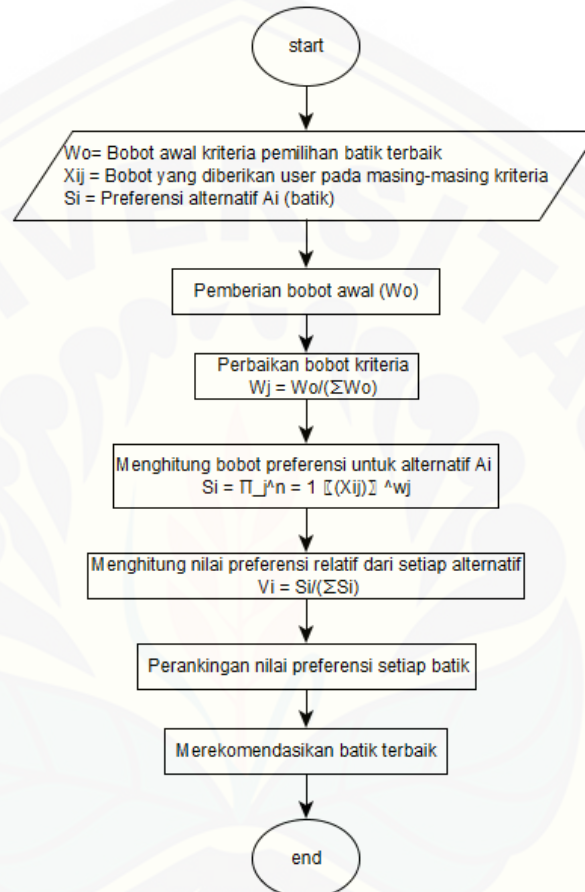
1. Melakukan wawancara kepada setiap pemilik griya batik di Kabupaten Jember untuk mendapatkan data dan informasi mengenai produk batik yang ada di griya batik.
2. Melakukan studi literature, jurnal, media, maupun internet mengenai pembangunan sistem rekomendasi produk bati griya batik di Kabupaten Jember seperti data kriteria dimasing-masing produk batik yakni harga, pewarnaan, motif, dan jenis kain yang digunakan.

Jenis data yang digunakan dalam penelitian adalah data primer dan data sekunder. Data primer merupakan data yang diperoleh melalui metode wawancara. Data sekunder merupakan data yang diperoleh dari studi literatur. Pengumpulan data dan informasi yang dilakukan antara lain data kriteria produk batik griya batik di Kabupaten Jember.

#### b. Metode Analisis Data

Tahap analisis data dimulai dengan menelaah data secara keseluruhan yang telah diperoleh dari tahap pengumpulan data. Data tersebut diantaranya yakni, data kriteria dari produk batik griya batik di Kabupaten Jember yakni harga, pewarnaan, motif, dan jenis kain. Langkah selanjutnya adalah pada setiap kriteria tersebut ditentukan

nilai bobot awal oleh konsumen atau *user* untuk diproses menggunakan metode *weighted product*. Adapun menganalisis data dengan menggunakan metode *weighted product* prosesnya dapat dilihat pada Gambar 3.2.



Gambar 3. 2 Diagram Alir Penerapan Metode *Weighted Product* (WP)

### 3.4.2 Tahapan Desain Sistem

Tahap ini meliputi tahap desain sistem dengan pembuatan diagram menggunakan tools *UML Visual Paradigm*. Diagram-diagram tersebut akan digunakan sebagai acuan pembuatan sistem pada tahap implementasi kebutuhan sistem. Konsep program menggunakan *Object-Oriented Programming* (OOP). Pada bagian desain, pengerjaan yang dilakukan diantaranya adalah:

a. *Business Process*

*Business Process* merupakan gambaran dari masuknya data serta data yang dihasilkan dari proses yang dijalankan sistem. Data yang dibutuhkan oleh sistem (*input*), keluaran data yang dihasilkan (*output*), media dari sistem (*uses*), dan tujuan dari pembuatan sistem (*goal*). *Input* yang dibutuhkan berupa data produk batik griya batik dan data griya batik. *Output* yang dihasilkan berupa hasil rekomendasi produk batik.

b. *Use Case Diagram*

*Use Case Diagram* merupakan dokumentasi yang menggambarkan fitur dan aktor yang mampu mengakses fitur-fitur pada sistem yang akan dibangun. Aktor yang dapat mengakses aplikasi ini yaitu pihak admin, user, dan owner griya. Fitur yang disediakan pada sistem adalah fitur pengelolaan data admin, fitur konfirmasi griya batik, fitur memilih batik, fitur mendaftar griya batik, fitur pengelolaan griya batik, dan fitur pengelolaan data batik.

c. *Use Case Scenario*

*Use case scenario* merupakan penjelasan alur sistem sesuai dengan yang tertera pada *use case diagram*, skenario juga menjelaskan reaksi yang akan terjadi pada sistem setelah menerima perlakuan dari aktor. *Use case scenario* menggambarkan keadaan normal saat setiap aktor mengakses aplikasi dan keadaan alternatif yang terjadi pada kondisi tertentu.

d. *Sequence Diagram*

*Sequence diagram* merupakan diagram yang digunakan untuk menggambarkan interaksi yang terjadi di dalam sistem sesuai dengan urutan dijalankannya sistem tersebut. Interaksi tersebut meliputi, aktor yang mengakses fitur tersebut sesuai dengan yang ada pada *use case diagram*, tampilan sistem, *controller*, model, dan pesan yang disampaikan jika terjadi suatu kondisi tertentu.

e. *Activity Diagram*

*Activity diagram* menggambarkan alur aktivitas dalam sistem yang sedang dirancang, meliputi awal alur dimulai, *decision* yang terjadi, dan bagaimana alur

berakhir yang dapat dilakukan oleh masing-masing aktor. *Activity diagram* juga menggambarkan aktivitas sistem setelah menerima perlakuan dari aktor tersebut.

f. *Class Diagram*

*Class diagram* merupakan sebuah spesifikasi yang menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class diagram* menggambarkan kelas, *function*, dan atribut yang dibutuhkan oleh seorang programmer dalam membangun sistem.

g. *Entity Relationship Diagram* (ERD)

*Entity relationship diagram* merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi.

### 3.4.3 Tahapan Implementasi Sistem

Setelah desain sistem telah selesai dilakukan maka selanjutnya pada tahap ini akan dilakukan pembuatan sistem dan implementasiannya. Pembuatan sistem meliputi penulisan kode program, dan pembuatan basis data. Penulisan kode program dilakukan menggunakan tools *sublime text 2* sebagai editor dengan bahasa pemrograman *PHP* sedangkan untuk manajemen basis data yang digunakan dalam membangun sistem yaitu *DBMS MySQL*.

### 3.4.4 Tahapan Pengujian Sistem

Pengujian digunakan untuk mengetahui sejauh mana sistem ini dapat berjalan. Testing berfungsi untuk mengetahui apakah sistem ini dapat berfungsi dengan baik sesuai dengan yang diharapkan. Serta untuk mengetahui letak kekurangan yang ada pada sistem ini. Terdapat dua metode yang digunakan untuk pengujian ini yakni :

a. *White box testing*

*White box testing* merupakan cara pengujian dengan melihat modul untuk yang telah dibuat dengan program – program yang ada. Pengujian ini dilakukan oleh (*developer*) pembuat program. Jika ada modul yang menghasilkan *output*

yang tidak sesuai, maka baris-baris program, variabel dan parameter yang terlibat pada unit tersebut satu persatu akan di cek dan diperbaiki, kemudian di *compile* ulang. (Pressman, 2001)

b. *Black box testing*

Berbeda dengan white box testing, *Black box testing* merupakan metode pengujian perangkat lunak yang memeriksa fungsionalitas dari aplikasi yang berkaitan dengan struktur *internal* atau kerja. Metode ini memfokuskan pada keperluan fungsionalitas dari *software* (Pressman, 2001).

#### 3.4.5 Tahapan Pemeliharaan Sistem

Pemeliharaan merupakan proses perawatan sistem setelah sistem digunakan oleh pengguna. Pemeliharaan dilakukan dengan mengecek kinerja sistem secara berkala. Pengecekan dilakukan apakah kinerja sistem masih berjalan dengan baik dan memperbaiki apakah terdapat kerusakan atau tidak.

## BAB 4. PENGEMBANGAN SISTEM

Bab ini menguraikan mengenai analisis kebutuhan, desain, implementasi, dan pengujian sistem yang digunakan dalam proses pengembangan sistem rekomendasi pemilihan produk pada batik griya batik di Kabupaten Jember menggunakan metode *weighted product*. Dimana tahapan analisis hingga pengujian yang dilakukan sesuai dengan metode pengembangan *waterfall*.

### 4.1 Deskripsi Umum Sistem

Deskripsi umum dari Sistem Rekomendasi Pemilihan Produk Batik pada Griya Batik di Kabupaten Jember Menggunakan Metode *Weighted Product* yang dibangun dalam penelitian ini akan dijelaskan lebih detail pada SOP (*statement of purpose*) sistem dan fungsi sistem.

#### 4.1.1 SOP (*Statement of purpose*)

Sistem Rekomendasi Pemilihan Produk Batik pada Griya Batik di Kabupaten Jember yang dibangun dalam penelitian ini merupakan sistem yang mampu menampilkan sebuah hasil rekomendasi dari pemberian bobot awal pada setiap kriteria yang dilakukan oleh *user* atau konsumen terhadap kriteria produk batik yang menjadi keinginan dan pilihan konsumen. Hasil penilaian tersebut berdasarkan analisis data dengan metode *weighted product*. Adapun kriteria yang digunakan merupakan data produk batik pada griya batik di Kabupaten Jember, yaitu diantaranya harga, pewarnaan, motif, dan jenis kain. Kriteria-kriteria tersebut ditentukan nilai bobot awalnya dimasing-masing kriteria. Nilai bobot tersebut dianalisis menggunakan metode *weigted product*, sehingga dihasilkan rekomendasi produk batik pada griya batik berdasarkan tingkat kebutuhan atau kepentingan user atau konsumen.

## 4.2 Pengumpulan Data

Pengumpulan data pada penelitian ini dilakukan melalui wawancara langsung ke setiap griya batik yang ada di Kabupaten Jember. Data yang diperlukan adalah data Batik setiap Griya Batik, beserta kriteria kriteria apa saja yang nantinya digunakan dalam membangun Sistem Rekomendasi Pemilihan Produk Batik.

## 4.3 Analisis Kebutuhan

Berdasarkan metode pengembangan sistem model *waterfall*, tahapan awal yang dilakukan adalah tahapan analisis kebutuhan. Tahapan ini dilakukan analisis terhadap kebutuhan-kebutuhan dari sistem yang dibangun, baik berupa kebutuhan fungsional maupun kebutuhan non-fungsional. Hasil analisis tersebut sangat mempengaruhi fungsionalitas sistem yang dibangun untuk dapat digunakan sesuai dengan fungsi dan kebutuhan pengguna.

### 4.2.1 Kebutuhan Fungsional

Kebutuhan fungsional sistem berisi fitur-fitur inti yang harus dipenuhi dalam sistem agar sistem mampu difungsikan sesuai dengan tujuan dan kebutuhan pengguna terhadap sistem itu sendiri. Kebutuhan fungsional dari sistem rekomendasi pemilihan produk batik pada griya batik di kabupaten jember menggunakan metode *weighted product* yaitu:

- a. Sistem mampu mengelola data admin (*view, insert, update, delete*)
- b. Sistem mampu mengelola data griya (*view, insert, update*)
- c. Sistem mampu mengelola data batik (*view, insert, update, delete*)
- d. Sistem mampu menampilkan rekomendasi produk batik pada konsumen

### 4.2.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional merupakan fitur-fitur yang dimiliki untuk mendukung sistem dalam memenuhi fungsionalitasnya untuk dapat memenuhi kebutuhan dari

pengguna. Kebutuhan non-fungsional dari sistem rekomendasi pemilihan produk batik pada griya batik di kabupaten jember yaitu :

- a. Sistem bekerja sesuai dengan fungsinya dan dapat dijalankan pada semua komputer dan *browser* yang berbeda.
- b. Tampilan dan bahasa komunikasi sistem mudah dimengerti oleh pengguna untuk memberikan kenyamanan pemakaian dan memudahkan pengoperasian.
- c. Sistem menggunakan *username* dan *password* untuk autentifikasi akses admin dan member terhadap sistem, sedangkan *user* biasa dapat langsung masuk kedalam sistem, karena sistem berbasis *free* untuk *user* biasa.

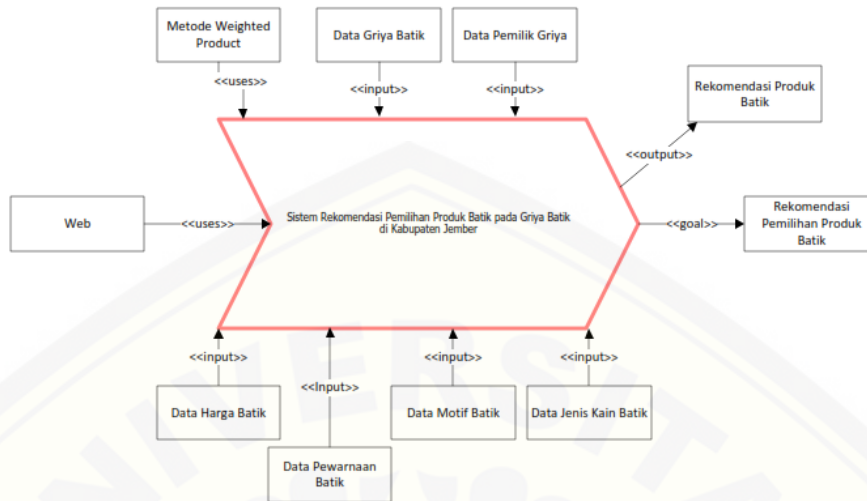
#### 4.4 Desain Sistem

Tahapan yang dilakukan setelah melakukan analisis kebutuhan sistem yaitu tahap perencanaan pembangunan sistem yang dapat digambarkan dengan desain sistem. sistem rekomendasi pemilihan batik pada griya batik di Kabupaten Jember ini meliputi *business process*, *use case diagram*, *scenario*, *activity diagram*, *sequence diagram*, *class diagram*, dan *entity relationship diagram*.

##### 4.3.1. *Business Process*

Selain dapat dideskripsikan dalam sebuah SOP (*Statement of Purpose*), gambaran umum sistem rekomendasi pemilihan produk pada batik griya batik di Kabupaten Jember menggunakan metode *weighted product* dapat digambarkan melalui sebuah *business process*. Seperti yang dapat kita lihat pada Gambar 4.1 menggambarkan data-data yang digunakan sebagai masukan, data keluaran, uses sistem yang dibangun, hingga goal dari dibangunnya sistem sendiri.

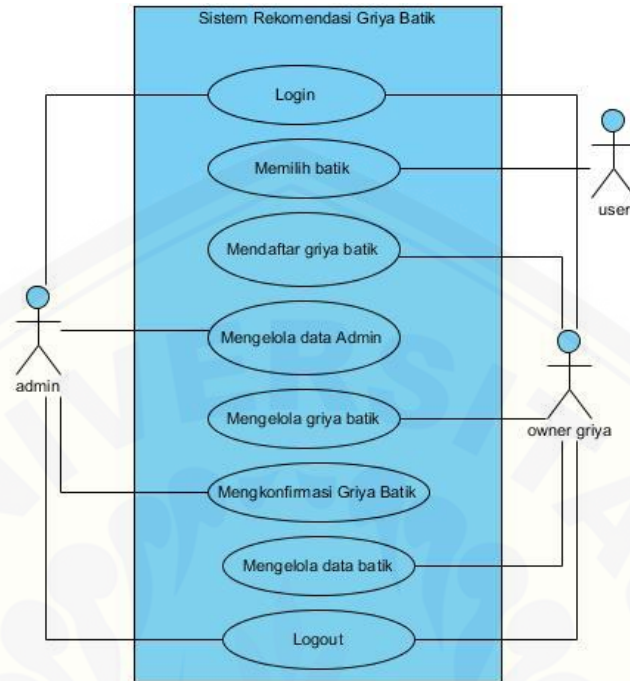




Gambar 4. 1 Business Process Sistem

#### 4.3.2. Use Case Diagram

*Use case* diagram merupakan pemodelan yang dibuat untuk dapat menggambarkan interaksi antara aktor sistem rekomendasi pemilihan produk pada batik griya batik di Kabupaten Jember menggunakan metode *weighted product* yang akan dibangun. Melalui *use case* diagram dapat diketahui interaksi yang dapat dilakukan aktor terhadap sistem sesuai dengan hak akses yang dimiliki oleh masing masing aktor atau pengguna. Pada Gambar 4.2 digambarkan *use case* diagram yang terdiri atas tiga aktor dengan delapan *use case*.



Gambar 4. 2 Use Case Diagram

Berdasarkan use case diagram pada Gambar 4.2 terdapat tiga aktor atau pengguna, yaitu admin, *owner* griya, dan *user*. Adapun deskripsi dari masing-masing aktor dapat dilihat pada Tabel 4.1.

Tabel 4.1 Deskripsi pembagian aktor sistem

No.	Aktor	Deskripsi
1	Admin	Aktor ini memiliki hak untuk menerima griya batik yang telah mengisi data-datanya secara benar dan dapat mengelola admin
2	User	Aktor yang memiliki akses terbatas pada sistem. Aktor ini hanya dapat melakukan pemilihan batik dengan menginputkan kriteria dan bobot terlebih dahulu dan kemudian akan di proses oleh sistem menggunakan metode WP
3	Owner Griya	Aktor yang dapat melakukan pendaftaran beserta ubah griya batiknya dan dapat mengelola batik.

Selain memiliki tiga aktor, dalam *use case* diagram juga terdapat delapan *use case*. Deskripsi dari *use case* tersebut dapat dilihat pada Tabel 4.5.

Tabel 4.2 Deskripsi *use case* sistem

No	Usecase	Deskripsi
1	Login	Usecase yang menggambarkan proses masuk ke dalam hak akses sistem masing masing aktor
2	Mengkonfirmasi Griya Batik	Usecase yang menggambarkan proses mengaktifkan atau non aktifkan Griya Batik
3	Mengelola data Admin	Usecase yang menggambarkan proses melihat, menambah, mengubah, dan menghapus data Admin
4	Memilih Batik	Usecase yang menggambarkan proses pemilihan batik sesuai dengan kriteria dan bobot yang diinputkan oleh <i>user</i>
5	Mendaftar Griya Batik	Usecase yang menggambarkan proses mendaftar Griya Batik
6	Mengelola	Usecase yang menggambarkan proses ubah data griya

Griya	batik
7	Mengelola data batik
	<i>Usecase</i> yang menggambarkan proses melihat, menambah, mengubah, dan menghapus data Batik
8	<i>Logout</i>
	<i>Usecase</i> yang menggambarkan proses keluar terhadap hak akses sistem masing-masing aktor

#### 4.3.3. Use Case Skenario

*Use case* skenario adalah dokumentasi terhadap kebutuhan fungsional sistem. *Use case* skenario sistem rekomendasi pemilihan produk batik pada griya batik di Kabupaten Jember menggunakan metode *weighted product* adalah sebagai berikut:

##### a. Skenario *Login*

Skenario *login* merupakan penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *use case* skenario *login*. Pada skenario tersebut dapat dilihat bagaimana alur perjalanan suatu fitur login antara aksi yang dilakukan oleh aktor yaitu admin dan owner griya ketika keadaan akan melakukan login, kemudian bagaimana reaksi dari sistem untuk merespon aksi tersebut sehingga aktor telah melakukan login pada sistem tersebut. Proses lengkap skenario *login* dapat dilihat pada Lampiran A Tabel A.1 *Use case* skenario *login*.

##### b. Skenario Mengkonfirmasi Griya Batik

Skenario Mengkonfirmasi Griya Batik merupakan penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengkonfirmasi griya batik. Pada skenario tersebut dapat dilihat bagaimana alur perjalanan suatu fitur pengaktifkan atau non-aktifkan griya batik antara aksi yang dilakukan oleh aktor yaitu admin, kemudian bagaimana reaksi dari sistem untuk merespon aksi tersebut sehingga aktor telah mengubah status konfirmasi pada sistem tersebut. Proses lengkap skenario mengkonfirmasi griya

batik dapat dilihat pada Lampiran A Tabel A.2 *Use case* skenario mengkonfirmasi griya batik.

c. Skenario Mengelola Data Admin

Skenario mengelola data Admin merupakan penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengelola data admin. Pada skenario tersebut dapat dilihat bagaimana alur perjalanan suatu fitur mengelola data admin antara aksi yang dilakukan oleh aktor yaitu admin ketika keadaan akan mengelola data admin, kemudian bagaimana reaksi dari sistem untuk merespon aksi tersebut sehingga aktor telah mengelola data admin pada sistem tersebut. Proses lengkap skenario mengelola data admin dapat dilihat pada Lampiran A Tabel A.3 *Use case* skenario mengelola data admin.

d. Skenario Memilih Batik

Skenario memilih batik merupakan penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario memilih batik. Pada skenario tersebut dapat dilihat bagaimana alur perjalanan suatu fitur memilih batik antara aksi yang dilakukan oleh aktor yaitu *user* ketika keadaan akan mengubah data pemilihan batik, kemudian bagaimana reaksi dari sistem untuk merespon aksi tersebut sehingga aktor telah memilih batik sistem tersebut. Proses lengkap skenario memilih batik dapat dilihat pada Tabel 4.6 *Use case* skenario memilih batik.

Tabel 4. 3 *Usecase* Skenario Memilih Batik

<b>ID</b>	: 03
<b>Name</b>	: Memilih Batik
<b>Participating Actor</b>	: <i>User</i>
<b>Pre Condition</b>	: Mengakses halaman <i>user</i>
<b>Post Condition</b>	: Menampilkan rekomendasi batik

**Normal Flow: Memilih Batik**

<b>Actor</b>	<b>System</b>
1. Membuka halaman web	
	2. Mengambil data batik pada

	database
	3. Menampilkan halaman daftar produk batik
4. Memilih tombol <i>show form</i>	
	5. Menagambil data kriteria pada <i>database</i>
	6. Menampilkan kriteria penilaian dengan pilihan tingkat kepentingan
7. Memilih tingkat kepentingan dari masing-masing batik untuk menghasilkan rekomendasi pemilihan batik	
	8. Menampilkan hasil rekomendasi pemilihan batik dan tabel hasil perhitungan dari tabel data batik pada <i>database</i> .

e. Skenario Mendaftar Griya Batik

Skenario mendaftar griya batik merupakan penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mendaftar griya batik. Pada skenario tersebut dapat dilihat bagaimana alur perjalanan suatu fitur menampilkan data kriteria antara aksi yang dilakukan oleh aktor yaitu *owner* griya ketika keadaan akan mendarat griya batik, kemudian bagaimana reaksi dari sistem untuk merespon aksi tersebut sehingga aktor telah mendaftar griya batik pada sistem tersebut. Proses lengkap skenario mendaftar griya batik dapat dilihat pada Lampiran A Tabel A.4 *Use case* skenario mendaftar griya batik.

f. Skenario Mengelola Griya Batik

Skenario mengelola griya batik merupakan penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengelola griya batik. Pada skenario tersebut dapat dilihat bagaimana alur perjalanan suatu fitur mengelola griya antara aksi yang dilakukan oleh aktor yaitu *owner* griya ketika keadaan akan mengelola griya, kemudian bagaimana reaksi dari sistem untuk merespon aksi tersebut sehingga aktor telah mengelola griya batik pada sistem tersebut. Proses lengkap skenario mengelola griya dapat dilihat pada Lampiran A Tabel A.5 *usecase* skenario mengelola griya.

g. Skenario Mengelola Data Batik

Skenario mengelola data batik merupakan penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengelola data batik. Pada skenario tersebut dapat dilihat bagaimana alur perjalanan suatu fitur mengelola data batik antara aksi yang dilakukan oleh aktor yaitu *owner griya* ketika keadaan akan mengelola data batik, kemudian bagaimana reaksi dari sistem untuk merespon aksi tersebut sehingga aktor telah mengelola data batik pada sistem tersebut. Proses lengkap skenario mengelola data batik dapat dilihat pada Lampiran A Tabel A.6 *Use case* skenario mengelola data batik.

h. Skenario *Logout*

Skenario *logout* merupakan penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *use case* skenario *logout*. Pada skenario tersebut dapat dilihat bagaimana alur perjalanan suatu fitur login antara aksi yang dilakukan oleh aktor yaitu admin dan *owner griya* ketika keadaan akan keluar dari sistem, kemudian bagaimana reaksi dari sistem untuk merespon aksi tersebut sehingga aktor telah keluar dari sistem tersebut. Proses lengkap skenario *logout* dapat dilihat pada Lampiran A Tabel A.7 *Use case* skenario *logout*.

#### 4.3.4. Activity Diagram

*Activity Diagram* menggambarkan aliran aktivitas sistem rekomendasi pemilihan produk batik pada griya batik di Kabupaten Jember menggunakan metode *weighted product* yang akan dibangun. Sistem rekomendasi pemilihan produk batik pada griya batik di Kabupaten Jember menggunakan metode *weighted product* ini memiliki delapan *activity diagram* yaitu sebagai berikut:

a. *Activity Login*

*Activity diagram login* dilakukan oleh admin dan *owner griya*. *Activity diagram login* menjelaskan tentang bagaimana sistem dapat menjalankan fungsi

melakukan autentifikasi hak akses semua aktor dalam menggunakan sistem, proses lengkapnya dapat dilihat pada Lampiran B Gambar B.1 *Activity diagram login*.

b. *Activity Diagram Mengkonfirmasi Griya Batik*

*Activity diagram mengkonfirmasi griya batik* dapat dilakukan oleh admin. *Activity diagram mengkonfirmasi griya batik* menjelaskan tentang bagaimana admin dapat mengubah status konfirmasi dari data griya batik pada sistem sehingga status konfirmasi berubah dan data griya batik tersimpan di *database*, proses lengkapnya dapat dilihat pada Lampiran B Gambar B.2 *Activity diagram mengkonfirmasi griya batik*.

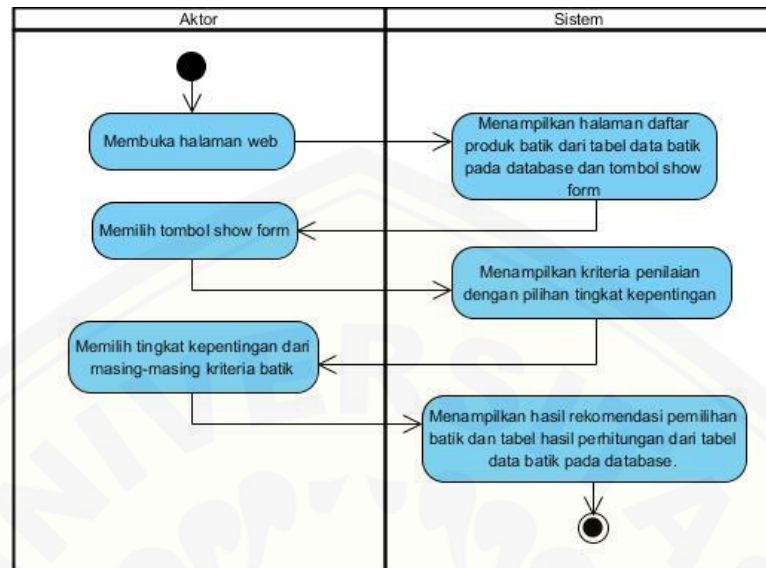
c. *Activity Diagram Mengelola Data Admin*

*Activity diagram mengelola data admin* dapat dilakukan oleh admin. *Activity diagram mengelola data admin* menjelaskan tentang bagaimana admin dapat mengelola data admin pada sistem, proses lengkapnya dapat dilihat pada Lampiran B Gambar B.3 *Activity diagram view kriteria*.

d. *Activity Diagram Memilih Batik*

*Activity diagram memilih batik* dapat dilakukan oleh *user*. *Activity diagram memilih batik* menjelaskan tentang bagaimana *user* dapat mengisi nilai bobot pada masing-masing kriteria pada sistem sehingga menghasilkan hasil pencarian batik, proses lengkapnya dapat dilihat pada Gambar 4.1 *Activity diagram memilih batik*.





Gambar 4. 3 Activity Diagram Memilih Batik

e. Activity Diagram Mendaftar Griya Batik

Activity diagram mendaftar griya batik dapat dilakukan oleh *owner* griya. Activity diagram mendaftar griya batik menjelaskan tentang bagaimana *owner* griya dapat mendaftar griya batik pada sistem proses lengkapnya dapat dilihat pada Lampiran B Gambar B.4 Activity diagram mendaftar griya batik.

f. Activity Diagram Mengelola Griya Batik

Activity diagram mengelola griya batik dapat dilakukan oleh *owner* griya. Activity diagram mengelola griya batik menjelaskan tentang bagaimana *owner* griya mengelola griya batik pada sistem. Activity diagram mengelola data pengguna dapat dilihat pada Lampiran B Gambar B.5 Activity diagram mengelola griya batik.

g. Activity Diagram Mengelola Data Batik

Activity diagram mengelola data batik dapat dilakukan oleh *owner* griya. Activity diagram mengelola data batik menjelaskan tentang bagaimana *owner* griya dapat mengelola data batik pada sistem, proses lengkapnya dapat dilihat pada Lampiran B Gambar B.6 Activity diagram mengelola data batik.

#### h. Activity Diagram Logout

*Activity* diagram *logout* menjelaskan tentang bagaimana sistem dapat menjalankan fungsi melakukan proses keluar dari sistem, proses lengkapnya dapat dilihat pada Lampiran B Gambar B.7 *Activity* diagram *logout*.

#### 4.3.5. Sequence Diagram

*Sequence* diagram adalah dokumentasi suatu diagram yang menampilkan urutan interaksi - interaksi antar objek di dalam sistem. *Sequence* diagram digunakan untuk menggambarkan skenario dan memodelkan aliran logika dalam sistem dengan cara *visual*. *Sequence* diagram dari sistem rekomendasi pemilihan batik pada griya batik di Kabupaten Jember menggunakan metode *weighted product* adalah sebagai berikut:

##### a. Sequence Diagram Login

Penggambaran *sequence login* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case login*. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram *login* lebih lengkap dijelaskan pada Lampiran C Gambar C.1 *sequence* diagram *login*.

##### b. Sequence Diagram Mengkonfirmasi Griya Batik

Penggambaran *sequence* diagram mengkonfirmasi griya batik digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* mengkonfirmasi griya batik. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram mengkonfirmasi griya batik lebih lengkap dijelaskan pada Lampiran C Gambar C.2 *sequence* diagram mengkonfirmasi griya batik.

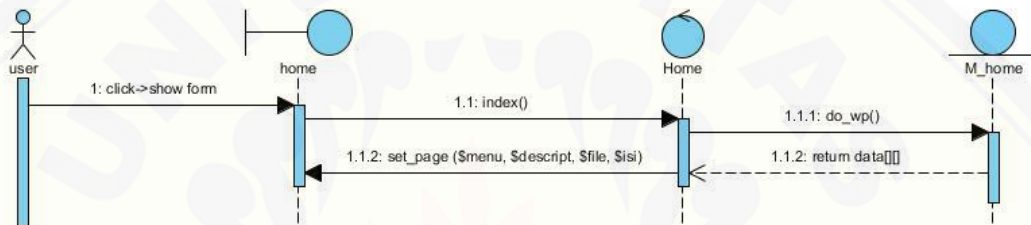
##### c. Sequence Diagram Mengelola Data Admin

Penggambaran *sequence* diagram mengelola data admin digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* mengelola data admin. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar.

*Sequence* diagram mengelola data admin lebih lengkap dijelaskan pada Lampiran C Gambar C.3 *sequence* diagram mengelola data admin.

d. *Sequence* Diagram Memilih Batik

Penggambaran *sequence* diagram memilih batik digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* memilih batik. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram memilih batik lebih lengkap dijelaskan pada Gambar 4.4 *sequence* diagram memilih batik.



Gambar 4. 4 *Sequence* Diagram Memilih Batik

e. *Sequence* Diagram Mendaftar Griya Batik

Penggambaran *sequence* diagram mendaftar griya batik digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* mendaftar griya batik. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram mendaftar griya batik lebih lengkap dijelaskan pada Lampiran C Gambar C.4 *sequence* diagram mendaftar griya batik.

f. *Sequence* Diagram Mengelola Griya Batik

Penggambaran *sequence* diagram mengelola griya batik digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* m mengelola griya batik. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram mengelola griya batik lebih lengkap dijelaskan pada Lampiran C Gambar C.5 *sequence* diagram mengelola griya batik.

g. *Sequence* Diagram Mengelola Data Batik

Penggambaran *sequence* diagram mengelola data batik digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* mengelola data

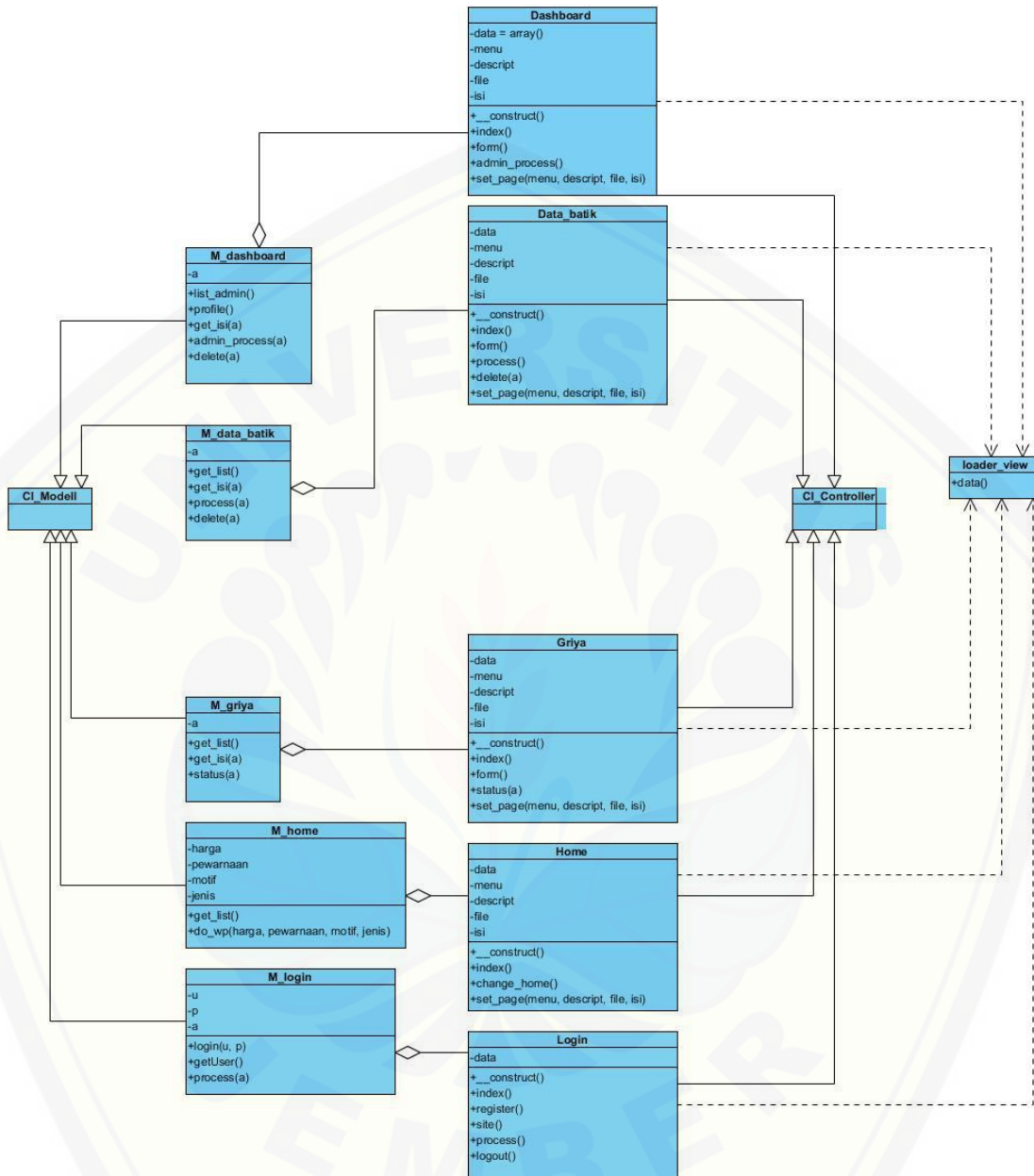
batik. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram mengelola data batik lebih lengkap dijelaskan pada Lampiran C Gambar C.6 *sequence* diagram mengelola data batik.

h. *Sequence* Diagram Logout

Penggambaran *sequence* diagram *logout* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* *logout*. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram *logout* lebih lengkap dijelaskan pada Lampiran C Gambar C.7 *sequence* diagram *logout*.

4.3.6. *Class* Diagram

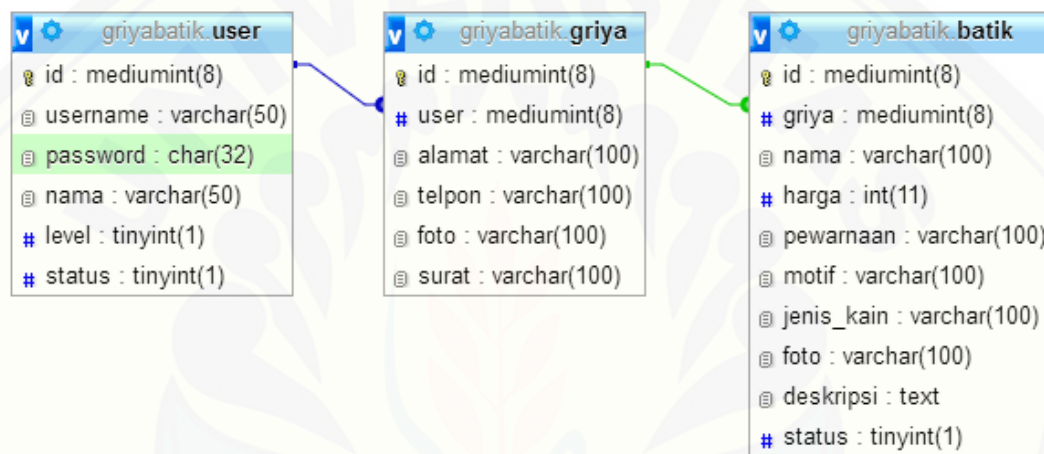
*Class* diagram menggambarkan hubungan antara kelas yang digunakan untuk membangun suatu sistem. Dalam paradigma OOP (*Object Oriented Programming*) terdapat 3 jenis kelas yaitu *model*, *view* dan *controller*. Berdasarkan *sequence* diagram yang telah dibangun, *class* diagram sistem rekomendasi pemilihan batik pada griya batik di Kabupaten Jember menggunakan metode *weighted product* terdiri dari 6 *class* *model*, 8 *class* *view* dan 6 *class* *controller*. *Class* diagram sistem dapat dilihat pada Gambar 4.5.



Gambar 4. 5 Class diagram

#### 4.3.7. Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) sistem rekomendasi pemilihan batik pada griya batik di Kabupaten Jember menggunakan metode *weighted product* merupakan gambaran komponen dan struktur *database* yang digunakan dalam pembuatan sistem. ERD yang diimplementasikan pada sistem ini terdiri dari 3 entitas yang dapat dilihat pada Gambar 4.6.



Gambar 4. 6 Entity Relationship Diagram

#### 4.4 Implementasi

Desain yang telah dibuat akan diimplementasikan ke dalam kode program. Beberapa hal yang dilakukan dalam tahap implementasi antara lain:

- Penulisan kode program (*coding*) menggunakan bahasa pemrograman *Page Hyper Text Pre-Processor (PHP)* dengan bantuan *framework Code Igniter (CI)*.
- Manajemen basis data menggunakan *DBMS MySQL*.

Kode program perhitungan metode *weighted product* terdapat di *class Home* pada *package controller* dan *class M\_home* pada *package models*. Penulisan kode program perhitungan metode *weighted product* dapat dilihat pada Tabel 4.4.

Tabel 4.4 Kode program *function* index()

No.	Kode
20	function index() {
21	\$this->session->set_userdata('home', ((\$this->session->userdata('home') == FALSE) ? FALSE : TRUE));
22	
23	\$isi = array( 24        'descript' => 'Data Batik' 25        , 'dataTable' => \$this->M_home->get_list() 26        , 'harga' => NULL 27        , 'pewarnaan' => NULL 28        , 'motif' => NULL 29        , 'jenis' => NULL 30        , 'add_button' => false 31        , 'back_button' => false 32    );
33	
34	if (isset(\$_POST['harga'])) {
35	\$isi['harga'] = \$_POST['harga'];
36	\$isi['pewarnaan'] = \$_POST['pewarnaan'];
37	\$isi['motif'] = \$_POST['motif'];
38	\$isi['jenis'] = \$_POST['jenis'];
39	
40	\$isi['dataTable'] = \$this->M_home->do_wp(\$isi['harga'], \$isi['pewarnaan'], \$isi['motif'], \$isi['jenis']);
41	}
42	
43	\$this->set_page('home', \$isi['descript'], 'home', \$isi);
44	}

Pada Tabel 4.4 merupakan kode program metode *weighted product* pada *class Home package controller* dengan *function* *index()* yang berfungsi untuk melakukan pemanggilan pada setiap *attribute* yang digunakan pada *function* perhitungan metode *weighted product*, dan dilanjutkan kepada *package* model yaitu *class M\_Home* pada *function* *do\_wp()*. Kode program *function* *do\_wp()* dapat dilihat pada Tabel 4.5.

Tabel 4.5 Kode program *function* *do\_wp()*

No	Kode
28	<code>function do_wp(\$harga, \$pewarnaan, \$motif, \$jenis) {</code>
29	<code>    \$query = \$this-&gt;db-&gt;query</code>
30	<code>    ('</code>
31	<code>        SELECT</code>
32	<code>        b.*</code>
33	<code>        , u.nama AS nama_griya</code>
34	<code>    FROM</code>
35	<code>        batik b</code>
36	<code>        , user u</code>
37	<code>        , griya g</code>
38	<code>    WHERE</code>
39	<code>        g.user = u.id</code>
40	<code>        AND b.griya = g.id</code>
41	<code>        AND u.status = 1</code>
42	<code>        AND b.status = 1</code>
43	<code>    );</code>
44	
45	<code>    \$list = \$query-&gt;result_array();</code>
46	<code>    \$list2 = array(array());</code>
47	
48	<code>        for (\$i = 0; \$i &lt; count(\$list); \$i++) {</code>
49	<code>            if (\$list[\$i]['harga'] &gt;= 500000) {</code>
50	<code>                \$v_harga = 5;</code>
51	<code>            } elseif (\$list[\$i]['harga'] &gt;= 300000 &amp;&amp; \$list[\$i]['harga'] &lt;</code>
52	<code>500000) {</code>
53	<code>                \$v_harga = 3;</code>
54	<code>            } elseif (\$list[\$i]['harga'] &lt; 300000) {</code>
55	<code>                \$v_harga = 1;</code>
56	<code>            }</code>
57	<code>            if (\$list[\$i]['pewarnaan'] == "Alami") {</code>
58	<code>                \$v_pewarnaan = 5;</code>
59	<code>            } else {</code>



---

```
60         $v_pewarnaan = 1;
61     }
62
63     if ($list[$i]['motif'] == "Tembakau") {
64         $v_motif = 5;
65     } else {
66         $v_motif = 1;
67     }
68
69     if ($list[$i]['jenis_kain'] == "Sutra") {
70         $v_jenis_kain = 5;
71     } else {
72         $v_jenis_kain = 1;
73     }
74
75     $list2[$i] = array(
76         0 => $list[$i]['id']
77         , 1 => $v_harga
78         , 2 => $v_pewarnaan
79         , 3 => $v_motif
80         , 4 => $v_jenis_kain
81     );
82 }
83
84 $sTot = 0;
85 $s = array();
86 $wTotal = ($harga + $pewarnaan + $motif + $jenis);
87 $wj = array(
88     0 => ($harga / $wTotal)
89     , 1 => ($pewarnaan / $wTotal)
90     , 2 => ($motif / $wTotal)
91     , 3 => ($jenis / $wTotal)
92 );
93 $jk = array(
94     0 => 0
95     , 1 => 1
96     , 2 => 1
97     , 3 => 1
98 );
99 // echo '<br /><br /><br /><br /><br /><br />';
100 // echo 'bobot -> ' . $wTotal . '<br />';
101 // print_r($wj);
102 for ($i = 0; $i < count($list2); $i++) { // Hitung S
103     $s[$i] = 0;
104     // echo '=====<br />';
105     for ($j = 1; $j < count($list2[$i]); $j++) {
```

---

---

```

106                                     if ($jk[$j - 1] == 0) {
107 //                                     echo '<br /> masuk 0 - ';
108                                     if ($j != 1) {
109 //                                         echo '[(!=1 sebelum) -> ' . $s[$i] . ' ]
                                         ';
110 //                                         echo '[proses -> ' . $list2[$i][$j] . ' ^
                                         ' . ((-1) * $wj[$j - 1]) . ' = ' . pow($list2[$i][$j], ((-1) * $wj[$j - 1])) .
                                         ' ]';
111                                     $s[$i] *= pow($list2[$i][$j], ((-1) *
                                         $wj[$j - 1]));
112 //                                         echo '[(!=1 sesudah) -> ' . $s[$i] . ' ]
                                         ';
113                                     } else {
114 //                                         echo '[(else sebelum) -> ' . $s[$i] . ' ]
                                         ';
115 //                                         echo '[proses -> ' . $list2[$i][$j] . ' ^
                                         ' . ((-1) * $wj[$j - 1]) . ' = ' . pow($list2[$i][$j], ((-1) * $wj[$j - 1])) .
                                         ' ]';
116                                     $s[$i] = pow($list2[$i][$j], ((-1) *
                                         $wj[$j - 1]));
117 //                                         echo '[(else sesudah) -> ' . $s[$i] . ' ]
                                         ';
118                                     }
119 // //                                     sTot = parseFloat(sTot) + parseFloat(s[i]);
120                                     } else {
121 //                                         echo '<br /> masuk 1 - ';
122                                     if ($j != 1) {
123 //                                         echo '[(!=1 sebelum) -> ' . $s[$i] . ' ]
                                         ';
124 //                                         echo '[proses -> ' . $list2[$i][$j] . ' ^
                                         ' . $wj[$j - 1] . ' = ' . pow($list2[$i][$j], $wj[$j - 1]) . ' ]';
125                                     $s[$i] *= pow($list2[$i][$j], $wj[$j
                                         - 1]);
126 //                                         echo '[(!=1 sesudah) -> ' . $s[$i] . ' ]
                                         ';
127                                     } else {
128 //                                         echo '[(else sebelum) -> ' . $s[$i] . ' ]
                                         ';
129 //                                         echo '[proses -> ' . $list2[$i][$j] . ' ^
                                         ' . $wj[$j - 1] . ' = ' . pow($list2[$i][$j], $wj[$j - 1]) . ' ]';
130                                     $s[$i] = pow($list2[$i][$j], $wj[$j -
                                         1]);
131 //                                         echo '[(else sesudah) -> ' . $s[$i] . ' ]
                                         ';
132                                     }
133 // //                                     sTot = parseFloat(sTot) + parseFloat(s[i]);

```

---

---

```
134         }
135     }
136     //     echo '<br /> Nilai S => ' . $s[$i];
137     $list[$i][count($list[$i])] = $s[$i];
138     $list2[$i][count($list2[$i])] = $s[$i];
139     //     echo '<br />';
140     $sTot += $s[$i];
141 }
142 //     echo 'sTot == ' . $sTot . '<br /><br />';
143 for ($i = 0; $i < count($list2); $i++) { // Hitung V
144 //     $v = ($s[$i] / $sTot) * 100;
145     $v = $s[$i] / $sTot;
146 //     $list[$i][count($list[$i])] = number_format((float)$v, 2, '.', '');
147     $list[$i][count($list[$i])] = $v;
148 //     $list2[$i][count($list2[$i])] = number_format((float)$v, 2, '.',
149 //     ');
150     $list2[$i][count($list2[$i])] = $v;
151 //     echo 'v-' . $i . '->' . $v . '<br />';
152 }
153 //     echo 'i == ' . count($list2);
154 //     for ($i = 1; $i < count($list2); $i++) { // urutkan berdasar nilai di list
155 //     indeks terakhir
156 //         $j = $i;
157 //         while ($j != 0 && ($list2[$j][(count($list2[$j]) - 1)] >
158 //         $list2[$j - 1][(count($list2[$j - 1]) - 1)]) {
159 //             $b = $list2[$j];
160 //             $list2[$j] = $list2[$j - 1];
161 //             $list2[$j - 1] = $b;
162 //             $j--;
163 //         }
164 //     }
165 //     for ($i = 1; $i < count($list); $i++) { // urutkan berdasar nilai di list
166 //     indeks terakhir
167 //         $j = $i;
168 //         while ($j != 0 && ($list[$j][(count($list[$j]) - 1)] > $list[$j -
169 //         1][(count($list[$j - 1]) - 1)]) {
170 //             $b = $list[$j];
171 //             $list[$j] = $list[$j - 1];
172 //             $list[$j - 1] = $b;
173 //             $j--;
174 //         }
175 //     }
176 //     return array("s" => $s, "v" => $v, "list" => $list2);
177 //     print_r($list2);
```

---

---

```
175         return $list;
176     }
177 }
```

---

Pada Tabel 4.8 merupakan kode program perhitungan metode *weighted program* pada *package* model *class* *M\_home fuction* *do\_wp()*. Pada kode program tersebut menjelaskan tahapan proses perhitungan metode *weighted product* yang menghasilkan hasil dari pemilihan *user* dari penilaian pada setiap kriteria yang ada. Hasil dari perhitungan tersebut adalah rekomendasi jenis batik yang telah disesuaikan dengan keinginan dan kebutuhan *user* yang telah diinputkan penilaian pada setiap kriteria pada kriteria batik dari hasil perhitungan terbesar sampai terkecil. Sehingga *user* dengan mudah memilih batik pada griya batik di Kabupaten Jember.

#### 4.5 Pengujian

Tahapan pengujian sistem merupakan suatu tahapan yang dilakukan secara sistematis untuk menguji dan mengevaluasi sistem dengan menggunakan sebuah metode pengujian sistem untuk mengetahui apakah kebutuhan sistem telah terpenuhi dan sistem layak untuk digunakan oleh pengguna. Agar pengujian yang dilakukan lebih valid, maka tahap pengujian sistem rekomendasi pemilihan produk batik pada griya batik di Kabupaten Jember menggunakan metode *weighted product* ini dilakukan dengan menggunakan dua metode, yaitu *white box* dan *black box*.

##### 4.5.1. Metode *White Box*

Pengujian sistem dengan metode *white box* dilakukan untuk menguji sistem dari segi desain dan kode program. Hal tersebut bertujuan untuk mengevaluasi apakah sistem mampu menghasilkan fungsi-fungsi, inputan, dan keluaran yang sesuai dengan spesifikasi dari kebutuhan sistem itu sendiri. Pengujian dengan metode *white box* dilakukan oleh peneliti dengan cara menghitung *independent path* yaitu dengan menggunakan suatu pengukuran kuantitatif *cyclomatic complexity*, listing program,

penentuan jalur independen, dan *test case*. Tahapan pengujian menggunakan metode *white box* meliputi:

a. *Listing Program*

*Listing* program merupakan baris-baris kode yang nantinya akan diuji. Setiap langkah dari kode-kode yang ada diberi nomor baik menjalankan *statement* biasa atau penggunaan kondisi dalam program.

b. Diagram alir

Diagram alir merupakan notasi yang digunakan untuk merepresentasikan aliran kontrol. Aliran kontrol yang digambarkan merupakan hasil penomoran dari listing program. Diagram alir digambarkan dengan node-node (simpul) yang dihubungkan dengan *edge-edge* (garis) yang menggambarkan alur jalannya program.

c. Kompleksitas siklomatik (*cyclomatic complexity*)

Kompleksitas siklomatik merupakan metrik perangkat lunak yang menyediakan ukuran kuantitatif dari kompleksitas logis suatu program. Bila digunakan dalam konteks teknik pengujian jalur dasar, nilai yang dihitung untuk kompleksitas siklomatik mendefinisikan jumlah jalur independen dalam basis set suatu program. Perhitungan kompleksitas siklomatik menggunakan rumus berikut ini.

$$V(G) = E - N + 2$$

Keterangan :

V(G) : Kompleksitas siklomatik

E : Jumlah *edge*

N : Jumlah *node*

d. Jalur Independen (*Independent Path*)

Jalur independen adalah setiap jalur yang melalui program yang memperkenalkan setidaknya satu kumpulan pernyataan-pernyataan pemrosesan atau kondisi baru. Bila dinyatakan dalam grafik alir, jalur independen harus

bergerak setidaknya sepanjang satu *edge* yang belum dilintasi sebelum jalur tersebut didefinisi.

e. Pengujian Basis Set (*Test Case*)

Pada bagian ini diberikan contoh data yang akan memaksa pelaksanaan jalur di basis set. Data yang dieksekusi dimasukkan ke dalam grafik alir apakah sudah melewati basis set yang tersedia. Sistem telah memenuhi syarat kelayakan perangkat lunak jika salah satu jalur yang dieksekusi setidaknya satu kali.

Pengujian sistem rekomendasi pemilihan batik pada griya batik di Kabupaten Jember menggunakan metode *weighted product* menggunakan metode *white box* akan diterapkan pada fitur yang dinilai dapat mewakili rekomendasi pemilihan batik pada griya batik di Kabupaten Jember sebagai berikut.

1) *Listing Program Metode Weighted Product*

```
28 function do_wp($harga, $pewarnaan, $motif, $jenis) {
29     $query = $this->db->query
30     (
31         SELECT
32             b.*
33             , u.nama AS nama_griya
34         FROM
35             batik b
36             , user u
37             , griya g
38         WHERE
39             g.user = u.id
40             AND b.griya = g.id
41             AND u.status = 1
42             AND b.status = 1
43     );
44
45     $list = $query->result_array();
46     $list2 = array(array());
47
48     for ($i = 0; $i < count($list); $i++) {
49         if ($list[$i]['harga'] >= 500000) {
50             $v_harga = 5;
51         } elseif ($list[$i]['harga'] >= 300000 && $list[$i]['harga'] < 500000) {
52             $v_harga = 3;
53         } elseif ($list[$i]['harga'] < 300000) {
54             $v_harga = 1;
55         }
56
57         if ($list[$i]['pewarnaan'] == "Alami") {
58             $v_pewarnaan = 5;
59         } else {
60             $v_pewarnaan = 1;
61         }
62     }
```

Gambar 4. 7 *Listing program function do\_wp() (1)*

```

63     if ($list[$i]['motif'] == "Tembakau") {
64         $v_motif = 5;
65     } else {
66         $v_motif = 1;
67     }
68
69     if ($list[$i]['jenis_kain'] == "Sutra") {
70         $v_jenis_kain = 5;
71     } else {
72         $v_jenis_kain = 1;
73     }
74
75     $list2[$i] = array(
76         0 => $list[$i]['id']
77         , 1 => $v_harga
78         , 2 => $v_pewarnaan
79         , 3 => $v_motif
80         , 4 => $v_jenis_kain
81     );
82 }
83
84 $sTot = 0;
85 $s = array();
86 $wTotal = ($harga + $pewarnaan + $motif + $jenis);
87 $wj = array(
88     0 => ($harga / $wTotal)
89     , 1 => ($pewarnaan / $wTotal)
90     , 2 => ($motif / $wTotal)
91     , 3 => ($jenis / $wTotal)
92 );
93 $jk = array(
94     0 => 0
95     , 1 => 1
96     , 2 => 1

```

Gambar 4. 8 Listing program function do\_wp() (2)

```

97     , 3 => 1
98 );
99 // echo '<br /><br /><br /><br /><br /><br />';
100 // echo 'bobot -> ' . $wTotal . '<br />';
101 // print_r($wj);
102 for ($i = 0; $i < count($list2); $i++) { // Hitung S
103     $s[$i] = 0;
104     echo '=====<br />';
105     for ($j = 1; $j < count($list2[$i]); $j++) {
106         if ($jk[(($j - 1) == 0)] {
107             echo '<br /> masuk 0 - ';
108             if ($j != 1) {
109                 echo ' [(!=1 sebelum) -> ' . $s[$i] . ' ] ';
110                 echo ' [proses -> ' . $list2[$i][[$j] . '^' . ((-1) * $wj[(($j - 1) == 0)] . ' = ' . pow($list2[$i]
111 [[$j], ((-1) * $wj[(($j - 1) == 0)])) . ' ] ';
112                 $s[$i] *= pow($list2[$i][[$j], ((-1) * $wj[(($j - 1) == 0)]));
113                 echo ' [(!=1 sesudah) -> ' . $s[$i] . ' ] ';
114             } else {
115                 echo ' [(else sebelum) -> ' . $s[$i] . ' ] ';
116                 echo ' [proses -> ' . $list2[$i][[$j] . '^' . ((-1) * $wj[(($j - 1) == 0)] . ' = ' . pow($list2[$i]
117 [[$j], ((-1) * $wj[(($j - 1) == 0)])) . ' ] ';
118                 $s[$i] = pow($list2[$i][[$j], ((-1) * $wj[(($j - 1) == 0)]));
119                 echo ' [(else sesudah) -> ' . $s[$i] . ' ] ';
120             }
121             // // sTot = parseFloat($sTot) + parseFloat($s[i]);
122         } else {
123             echo '<br /> masuk 1 - ';
124             if ($j != 1) {
125                 echo ' [(!=1 sebelum) -> ' . $s[$i] . ' ] ';
126                 echo ' [proses -> ' . $list2[$i][[$j] . '^' . $wj[(($j - 1) == 0)] . ' = ' . pow($list2[$i][[$j], $wj
127 [[$j] - 1]] . ' ] ';
128                 $s[$i] *= pow($list2[$i][[$j], $wj[(($j - 1) == 0)]);
129                 echo ' [(!=1 sesudah) -> ' . $s[$i] . ' ] ';
130             } else {

```

Gambar 4. 9 Listing program function do\_wp() (3)

```

128 //      echo ' [(else sebelum) -> ' . $s[$i] . ' ]';
129 //      echo ' [proses -> ' . $list2[$i][$j] . '^' . $wj[( $j - 1)] . ' = ' . pow($list2[$i][$j], $wj
[( $j - 1)] . ' ]';
130 //      $s[$i] = pow($list2[$i][$j], $wj[( $j - 1)]);
131 //      echo ' [(else sesudah) -> ' . $s[$i] . ' ]';
132 //      }
133 // //      sTot = parseFloat(sTot) + parseFloat(s[i]);
134 //      }
135 //      }
136 //      echo '<br /> Nilai S => ' . $s[$i];
137 //      $list[$i][count($list[$i])] = $s[$i];
138 //      $list2[$i][count($list2[$i])] = $s[$i];
139 //      echo '<br />';
140 //      $sTot += $s[$i];
141 //      }
142 //      echo 'sTot == ' . $sTot . '<br /><br />';
143 //      for ($i = 0; $i < count($list2); $i++) { // Hitung V
144 //      $v = ($s[$i] / $sTot) * 100;
145 //      $v = $s[$i] / $sTot;
146 //      $list[$i][count($list[$i])] = number_format((float)$v, 2, '.', '');
147 //      $list[$i][count($list[$i])] = $v;
148 //      $list2[$i][count($list2[$i])] = number_format((float)$v, 2, '.', '');
149 //      $list2[$i][count($list2[$i])] = $v;
150 //      echo 'v-' . $i . ' -> ' . $v . '<br />';
151 //      }
152 //      }
153 //      echo 'i == ' . count($list2);
154 //      for ($i = 1; $i < count($list2); $i++) { // urutkan berdasar nilai di list indeks terakhir
155 //      $j = $i;
156 //      while ($j != 0 && ($list2[$j][(count($list2[$j]) - 1)] > $list2[$j - 1][(count($list2[$j - 1]) - 1)])) {
157 //      $b = $list2[$j];
158 //      $list2[$j] = $list2[$j - 1];
159 //      $list2[$j - 1] = $b;
160 //      $j--;
161 //      }

```

Gambar 4. 10 Listing program function do\_wp() (4)

```

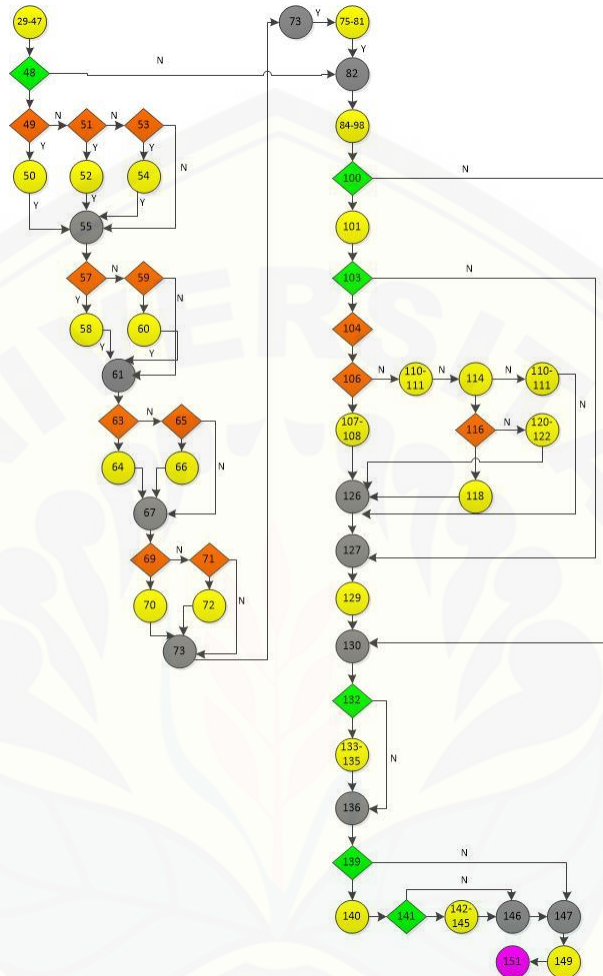
162 //      }
163 //      for ($i = 1; $i < count($list); $i++) { // urutkan berdasar nilai di list indeks terakhir
164 //      $j = $i;
165 //      while ($j != 0 && ($list[$j][(count($list[$j]) - 1)] > $list[$j - 1][(count($list[$j - 1]) - 1)])) {
166 //      $b = $list[$j];
167 //      $list[$j] = $list[$j - 1];
168 //      $list[$j - 1] = $b;
169 //      $j--;
170 //      }
171 //      }
172 //      }
173 //      return array("s" => $s, "v" => $v, "list" => $list2);
174 //      print_r($list2);
175 //      return $list;
176 //      }
177 //      }
178 //      }

```

Gambar 4. 11 Listing program function do\_wp() (5)



## 2) Diagram Alir

Gambar 4. 12 Diagram alir *function do\_wp ()*3) Kompleksitas Siklomatik (*Cyclomatic Complexity*)

*Function do\_wp ()*

$$V(G) = E - N + 2$$

$$V(G) = 70 - 54 + 2 = 18$$

4) Pengujian Basis Set (*Test Case*)

Pada Gambar 4.12 dapat dijelaskan bahwa node 1 yang bernilai 29-47, node 2 yang bernilai 50 merupakan baris koding yang sudah dipaparkan pada Gambar 4.7 sampai Gambar 4.11.

Tabel 4. 6 *Test case function do\_wp()*

<b>Jalur 1</b>	
<b>Test Case</b>	Jika berhasil menghitung vektor S dan vektor V
<b>Target yang diharapkan</b>	Menampilkan hasil perhitungan nilai S dan nilai V
<b>Hasil Pengujian</b>	Benar
<b>Path / jalur</b>	1-2-3-4-9-10-11-14-15-16-19-20-22-24-25-26-27-28-29-30- 31-32-33-40-41-42-43-44-45-46-47-48-49-50-51-52-53-54
<b>Jalur 2</b>	
<b>Test Case</b>	Jika berhasil menghitung vektor S dan vektor V
<b>Target yang diharapkan</b>	Menampilkan hasil perhitungan nilai S dan nilai V
<b>Hasil Pengujian</b>	Benar
<b>Path / jalur</b>	1-2-26-27-28-29-30-31-32-33-40-41-42-43-44-45-46-47-48- 49-50-51-52-53-54
<b>Jalur 3</b>	
<b>Test Case</b>	Jika berhasil menghitung vektor S dan vektor V
<b>Target yang diharapkan</b>	Menampilkan hasil perhitungan nilai S dan nilai V
<b>Hasil Pengujian</b>	Benar
<b>Path / jalur</b>	1-2-3-5-6-9-10-11-14-15-16-19-20-22-24-25-26-27-28-29-30- 31-32-33- 40-41-42-43-44-45-46-47-48-49-50-51-52-53-54
<b>Jalur 4</b>	
<b>Test Case</b>	Jika berhasil menghitung vektor S dan vektor V
<b>Target yang diharapkan</b>	Menampilkan hasil perhitungan nilai S dan nilai V
<b>Hasil Pengujian</b>	Benar
<b>Path / jalur</b>	1-2-3-5-7-9-10-11-14-15-16-19-20-22-24-25-26-27-28-29-30- 31-32-33- 40-41-42-43-44-45-46-47-48-49-50-51-52-53-54

<b>Jalur 5</b>	
<b>Test Case</b>	Jika berhasil menghitung vektor S dan vektor V
<b>Target yang diharapkan</b>	Menampilkan hasil perhitungan nilai S dan nilai V
<b>Hasil Pengujian</b>	Benar
<b>Path / jalur</b>	1-2-3-5-7-9-10-11-14-15-16-19-20-22-24-25-26-27-28-29-30-31-32-33- 40-41-42-43-44-45-46-47-48-49-50-51-52-53-54
<b>Jalur 6</b>	
<b>Test Case</b>	Jika berhasil menghitung vektor S dan vektor V
<b>Target yang diharapkan</b>	Menampilkan hasil perhitungan nilai S dan nilai V
<b>Hasil Pengujian</b>	Benar
<b>Path / jalur</b>	1-2-3-4-9-10-12-13-14-15-16-19-20-22-24-25-26-27-28-29-30-31-32-33- 40-41-42-43-44-45-46-47-48-49-50-51-52-53-54
<b>Jalur 7</b>	
<b>Test Case</b>	Jika berhasil menghitung vektor S dan vektor V
<b>Target yang diharapkan</b>	Menampilkan hasil perhitungan nilai S dan nilai V
<b>Path / jalur</b>	1-2-3-4-9-10-12-14-15-16-19-20-22-24-25-26-27-28-29-30-31-32-33- 40-41-42-43-44-45-46-47-48-49-50-51-52-53-54
<b>Jalur 8</b>	
<b>Test Case</b>	Jika berhasil menghitung vektor S dan vektor V
<b>Target yang diharapkan</b>	Menampilkan hasil perhitungan nilai S dan nilai V
<b>Hasil Pengujian</b>	Benar
<b>Path / jalur</b>	21-2-3-4-9-10-11-14-15-17-18-19-20-22-24-25-26-27-28-29-30-31-32-33- 40-41-42-43-44-45-46-47-48-49-50-51-52-53-54

<b>Jalur 9</b>	
<b>Test Case</b>	Jika berhasil menghitung vektor S dan vektor V
<b>Target yang diharapkan</b>	Menampilkan hasil perhitungan nilai S dan nilai V
<b>Hasil Pengujian</b>	Benar
<b>Path / jalur</b>	1-2-3-4-9-10-11-14-15-17-19-20-22-24-25-26-27-28-29-30-31-32-33- 40-41-42-43-44-45-46-47-48-49-50-51-52-53-54
<b>Jalur 10</b>	
<b>Test Case</b>	Jika berhasil menghitung vektor S dan vektor V
<b>Target yang diharapkan</b>	Menampilkan hasil perhitungan nilai S dan nilai V
<b>Hasil Pengujian</b>	Benar
<b>Path / jalur</b>	1-2-3-4-9-10-11-14-15-16-19-20-21-23-24-25-26-27-28-29-30-31-32-33- 40-41-42-43-44-45-46-47-48-49-50-51-52-53-54
<b>Jalur 11</b>	
<b>Test Case</b>	Jika berhasil menghitung vektor S dan vektor V
<b>Target yang diharapkan</b>	Menampilkan hasil perhitungan nilai S dan nilai V
<b>Hasil Pengujian</b>	Benar
<b>Path / jalur</b>	1-2-3-4-9-10-11-14-15-16-19-20-21-24-25-26-27-28-29-30-31-32-33- 40-41-42-43-44-45-46-47-48-49-50-51-52-53-54
<b>Jalur 12</b>	
<b>Test Case</b>	Jika berhasil menghitung vektor S dan vektor V
<b>Target yang diharapkan</b>	Menampilkan hasil perhitungan nilai S dan nilai V
<b>Hasil Pengujian</b>	Benar
<b>Path / jalur</b>	1-2-3-4-9-10-11-14-15-16-19-20-22-24-25-26-27-28-43-44-45-46-47-48-49-50-51-52-53-54

**Jalur 13**

<b>Test Case</b>	Jika berhasil menghitung vektor S dan vektor V
<b>Target yang diharapkan</b>	Menampilkan hasil perhitungan nilai S dan nilai V
<b>Hasil Pengujian</b>	Benar
<b>Path / jalur</b>	1-2-3-4-9-10-11-14-15-16-19-20-22-24-25-26-27-28-29-30-43-44-45-46-47-48-49-50-51-52-53-54

**Jalur 14**

<b>Test Case</b>	Jika berhasil menghitung vektor S dan vektor V
<b>Target yang diharapkan</b>	Menampilkan hasil perhitungan nilai S dan nilai V
<b>Hasil Pengujian</b>	Benar
<b>Path / jalur</b>	1-2-3-4-9-10-11-14-15-16-19-20-22-24-25-26-27-28-29-30-31-32-34-35-38-40-41-42-43-44-45-46-47-48-49-50-51-52-53-54

**Jalur 15**

<b>Test Case</b>	Jika berhasil menghitung vektor S dan vektor V
<b>Target yang diharapkan</b>	Menampilkan hasil perhitungan nilai S dan nilai V
<b>Hasil Pengujian</b>	Benar
<b>Path / jalur</b>	1-2-3-4-9-10-11-14-15-16-19-20-22-24-25-26-27-28-29-30-31-32-34-35-36-39-40-41-42-43-44-45-46-47-48-49-50-51-52-53-54

**Jalur 16**

<b>Test Case</b>	Jika berhasil menghitung vektor S dan vektor V
<b>Target yang diharapkan</b>	Menampilkan hasil perhitungan nilai S dan nilai V
<b>Hasil Pengujian</b>	Benar
<b>Path / jalur</b>	1-2-3-4-9-10-11-14-15-16-19-20-22-24-25-26-27-28-29-30-

---

31-32-33- 40-41-42-43-44-46-47-48-49-50-51-52-53-54

#### Jalur 17

<b>Test Case</b>	Jika berhasil menghitung vektor S dan vektor V
<b>Target yang diharapkan</b>	Menampilkan hasil perhitungan nilai S dan nilai V
<b>Hasil Pengujian</b>	Benar
<b>Path / jalur</b>	1-2-3-4-9-10-11-14-15-16-19-20-22-24-25-26-27-28-29-30- 31-32-33- 40-41-42-43-44-45-46-47-52-53-54

#### Jalur 18

<b>Test Case</b>	Jika berhasil menghitung vektor S dan vektor V
<b>Target yang diharapkan</b>	Menampilkan hasil perhitungan nilai S dan nilai V
<b>Hasil Pengujian</b>	Benar
<b>Path / jalur</b>	1-2-3-4-9-10-11-14-15-16-19-20-22-24-25-26-27-28-29-30- 31-32-33- 40-41-42-43-44-45-46-47-48-49-51-52-53-54

---

#### 4.5.2. Metode *Black Box*

*Black box testing* merupakan metode pengujian perangkat lunak yang memeriksa fungsionalitas dari aplikasi yang berkaitan dengan struktur internal atau kerja. Metode ini memfokuskan pada keperluan fungsionalitas dari software. Pengujian *black box* pada sistem rekomendasi pemilihan batik pada griya batik di Kabupaten Jember dilakukan untuk mengetahui apakah masukan dan keluaran dari sistem sesuai dengan kebutuhan fungsional atau tidak. Pengujian dilakukan pada setiap *usecase*. Hasil pengujian *black box* dapat dilihat pada Lampiran E (Pengujian *Black Box*).

## BAB 6. PENUTUP

Bab ini berisi mengenai kesimpulan dan saran dari peneliti tentang penelitian yang telah dilakukan. Kesimpulan dan saran tersebut diharapkan dapat digunakan sebagai acuan pada penelitian selanjutnya.

### 6.1 Kesimpulan

Kesimpulan dari penelitian yang telah dilakukan adalah:

- a. Mengimplementasikan metode *weighted product* digunakan pada sistem rekomendasi pemilihan batik pada griya batik di Kabupaten Jember untuk memberikan penyelesaian lebih mudah ketika melakukan pemilihan batik pada Griya Batik di Kabupaten Jember. Metode *weighted product* mendapatkan nilai rekomendasi pemilihan batik pada griya batik berdasarkan nilai bobot pada setiap produk dan bobot pada setiap kriteria. Langkah awal untuk mendapatkan rekomendasi pemilihan batik yaitu menentukan nilai bobot pada setiap nama atau produk batik oleh pemilik griya batik. Selanjutnya *user* memasukkan nilai bobot pada kriteria batik pada skala kepentingan yang ada. Dari penilaian yang diperoleh selanjutnya dihitung nilai perbaikan bobot, menentukan nilai *cost* dan benefit, serta dihasilkan nilai vektor S dan vektor V pada masing-masing nama atau produk batik. Hasil perhitungan vektor V akan diurutkan dari nilai terbesar sampai terkecil yang menjadi rekomendasi sistem dalam memilih batik yang sesuai dengan masukan *user* tersebut.
- b. Sistem rekomendasi pemilihan batik pada griya batik di Kabupaten Jember menggunakan metode *weighted product* berhasil dibangun. Sistem rekomendasi yang dibangun berbasis *website* dengan 3 hak akses yaitu admin dan *owner* griya dan *user* dengan fitur utama adalah fitur pemilihan batik yang dilakukan oleh *user*. Model pengembangan sistem yang diterapkan yaitu model *waterfall*. Hasil data dengan metode *weighted product* yang diperoleh dari *user* menunjukkan

bahwa produk batik yang direkomendasikan telah sesuai dengan masukan kriteria yang dilakukan oleh *user*. Hasil perhitungan metode *weighted product* dari dua sampel didapatkan rekomendasi pemilihan batik diposisi pertama Batik Tulis Bunga Tembakau, Batik Capung Bako kedua, Batik Tulis Mbako Lima ketiga. Sedangkan pada sampel kedua produk batik yang direkomendasikan adalah Batik Capung Bako diposisi pertama, Batik Tulis Tembakau kedua, Batik Tulis Sekar Jagad Pandhalungan ketiga.

## 6.2 Saran

Adapun saran yang ditujukan untuk memberikan masukan yang lebih baik yaitu :

- a. Sistem rekomendasi pemilihan batik pada griya batik di Kabupaten Jember yang akan dikembangkan selanjutnya perlu ditambahkan jumlah kriterianya, sehingga data kriteria yang digunakan sebagai rekomendasi pemilihan batik semakin beragam dengan keinginan dan kebutuhan *user*. Namun tetap memperhatikan jumlah kriteria yang terlalu banyak akan menimbulkan kurang minatnya *user*.
- b. Sistem rekomendasi pemilihan batik pada griya batik di Kabupaten Jember yang akan dikembangkan selanjutnya akan lebih baik jika bisa menyesuaikan tempat atau wilayah yang diinginkan dan dibutuhkan *user*, sehingga sistem rekomendasi pemilihan batik ini bisa digunakan untuk merekomendasikan pemilihan batik di seluruh wilayah Indonesia.



**DAFTAR PUSTAKA**

- Alfita, R. (2010). Perancangan Sistem Pendukung Keputusan Penentuan Prioritas Produk Unggulan Daerah Menggunakan Metode Weighted Product (WP).
- Kusumadewi, S., Hartati, S., Harjoko, A., & Wardoyo, R. (2006). *Fuzzy Multi-Attribute Decision Making (FUZZY MADM)*. Yogyakarta: Graha Ilmu.
- Ningrum. (2011). Perangkingan Calon Penerima Beasiswa Peningkatan Prestasi Akademik menggunakan metode Weighted Product (WP).
- Pressman, R. S. (2001). *Software Engineering a practitioner's approach* (5 ed.). New York, America: McGraw-Hill.
- Solichah, Z. (2009, Oktober 16). *www.antarnews.com*. Dipetik Mei 13, 2016, dari <http://www.antarnews.com/berita/158149/motiftembakau-jadi-khas-batik-jember>
- Sommerville, I. (2001). *Software Engineering*. Addison Wesley.
- Tan, F. (2013). *Metode Prototyping Dalam Pembuatan Sebuah Software*. Dipetik Mei 15, 2016, dari [blogger-sharearea.blogspot.com/2013](http://blogger-sharearea.blogspot.com/2013)
- Zukarnain. (2015). Sistem Informasi dan Rekomendasi dan Rekomendasi Pemilihan Homestay Bagi Wisatawan di Kawasan Desa Wisata Menggunakan Metode Weighted Product (WP).



## LAMPIRAN

## LAMPIRAN A. USE CASE SKENARIO

## A.1 Use Case Skenario Konfirmasi Griya Batik

Tabel A. 1 Use case Konfirmasi Griya Batik

<b>ID</b>	: 01
<b>Name</b>	: Konfirmasi Griya Batik
<b>Participating Actor</b>	: Admin
<b>Pre Condition</b>	: Admin sudah melakukan login
<b>Post Condition</b>	: Status <i>owner</i> griya batik aktif dan tersimpan di <i>database</i>

**Normal Flow**

<b>Actor</b>	<b>System</b>
1. Klik data Griya pada halaman admin	
	2. Mengambil data griya batik dari <i>database</i>
	3. Menampilkan halaman semua data griya batik
4. Klik aktifkan pada <i>field</i> aksi	
	5. Status griya batik berhasil diubah menjadi aktif dan tersimpan di <i>database</i>

**Alternative Flow Login Gagal**

<b>Actor</b>	<b>System</b>
1. Klik Login	
	2. Menampilkan login gagal

## A.2 Use Case Skenario Mengelola Data Admin

Tabel A. 2 Use case skenario Mengelola Data Admin

<b>ID</b>	: 02
<b>Name</b>	: Mengelola data Admin
<b>Participating Actor</b>	: Admin
<b>Pre Condition</b>	: Admin sudah melakukan <i>login</i>
<b>Post Condition</b>	: Data admin telah terkelola dan tersimpan di <i>database</i>

**Normal Flow: Melihat Data Admin**

<b>Actor</b>	<b>System</b>
1. Klik Dashboard pada halaman Admin	
	2. Berhasil menampilkan halaman daftar admin

**Normal Flow: Menambah Data Admin**

<b>Actor</b>	<b>System</b>
1. Klik tombol tambah	
	2. Menampilkan halaman form tambah Admin
3. Mengisi form	
	4. Menampilkan form yang sudah terisi
5. Klik Simpan	
	6. Menampilkan alert Tambah Data Berhasil dan data tersebut tersimpan ke <i>database</i>

**Alternative Flow: Batal Tambah Data Admin**

<b>Actor</b>	<b>System</b>
1. Klik Kembali	
	2. Kembali ke halaman data Admin

**Alternative Flow: Form Belum Terisi**

<b>Actor</b>	<b>System</b>
1. Klik Simpan	
	2. Menampilkan <i>alert "This is required"</i>

**Normal Flow: Mengubah Data Admin**

<b>Actor</b>	<b>System</b>
1. Pilih ubah pada field aksi	
	2. Menampilkan halaman form ubah Admin
3. Mengisi form ubah Admin	

	4. Menampilkan form ubah Admin yang sudah terisi
5. Klik Simpan	
	6. Menampilkan alert Edit Data Berhasil

**Alternative Flow: Batal Ubah Data Admin**

<b>Actor</b>	<b>System</b>
1. Klik Kembali	
	2. Kembali ke halaman data Admin

**Alternative Flow: Form Belum Terisi**

<b>Actor</b>	<b>System</b>
1. Klik Simpan	
	2. Menampilkan alert "This is required"

**Normal Flow: Menghapus Data Admin**

<b>Actor</b>	<b>System</b>
1. Pilih hapus pada field aksi	
	2. Menampilkan alert Apakah anda yakin menghapus data ini
3. Klik Oke	
	4. Menampilkan alert Hapus Data Berhasil

**Alternative Flow: Batal Hapus Data Admin**

<b>Actor</b>	<b>System</b>
1. Klik Batal	
	2. Kembali ke halaman data Admin

A.3 Use Case Skenario Mendaftar Griya Batik

Tabel A. 3 Use case skenario Mendaftar Griya Batik

<b>ID</b>	: 04
<b>Name</b>	: Mendaftar Griya Batik
<b>Participating Actor</b>	: Owner Griya
<b>Pre Condition</b>	: Mengakses form register
<b>Post Condition</b>	: Data griya batik terisi dan tersimpan di database

**Normal Flow: Registrasi Griya Batik**

<b>Actor</b>	<b>System</b>
1. Klik Register	
	2. Menampilkan halaman form pendaftaran Griya Batik
3. Mengisi form	

	4. Menampilkan form yang sudah terisi
5. Klik Simpan	
	6. Menampilkan alert Berhasil tambah data

**Alternative Flow: Form Belum Terisi Semua**

<i>Actor</i>	<i>System</i>
1. Klik Simpan	
	2. Menampilkan <i>alert</i> Tolong isikan kolom

A.4 Use Case Skenario Mengelola Griya Batik

Tabel A. 4 Use case skenario Mengelola Griya Batik

<b>ID</b>	: 04
<b>Name</b>	: Mengelola Griya Batik
<b>Participating Actor</b>	: Owner Griya Batik
<b>Pre Condition</b>	: Mengakses form data Griya Batik
<b>Post Condition</b>	: Mengisi form data Griya Batik dan tersimpan di database

**Normal Flow: Mengubah data Griya Batik**

<i>Actor</i>	<i>System</i>
1. Pilih ubah pada field aksi	
	2. Menampilkan halaman form ubah Griya
3. Mengisi form ubah Griya	
	4. Menampilkan form ubah Griya yang sudah terisi
5. Klik Simpan	
	6. Menampilkan alert Sukses ubah Griya

**Alternative Flow: Batal Ubah Griya**

<i>Actor</i>	<i>System</i>
3. Klik Kembali	
	4. Kembali ke halaman data Griya Batik

**Alternative Flow: Form Belum Terisi**

<i>Actor</i>	<i>System</i>
3. Klik Simpan	
	4. Menampilkan <i>alert</i> Tolong isikan

	kolom
--	-------

A.5 Use Case Skenario Mengelola Data Batik

Tabel A. 5 Use case skenario Mengelola Data Batik

<b>ID</b>	: 05
<b>Name</b>	: Mengelola Data Batik
<b>Participating Actor</b>	: Owner Griya
<b>Pre Condition</b>	: Mengakses form tambah Batik
<b>Post Condition</b>	: Data batik terkelola dan tersimpan di database
<b>Normal Flow: Melihat Data Batik</b>	
<b>Actor</b>	<b>System</b>
1. Klik Data Batik	
	2. Berhasil menampilkan halaman daftar Batik
<b>Normal Flow: Menambah batik</b>	
<b>Actor</b>	<b>System</b>
3. Klik Tambah	
	4. Menampilkan halaman form tambah Batik
5. Mengisi form	
	6. Menampilkan form yang sudah terisi
7. Klik Simpan	
	8. Menampilkan alert Tambah data berhasil
<b>Alternative Flow: Batal Tambah Batik</b>	
<b>Actor</b>	<b>System</b>
1. Klik Kembali	
	2. Kembali ke halaman data Griya Batik
<b>Alternative Flow: Form Belum Terisi Semua</b>	
<b>Actor</b>	<b>System</b>
1. Klik Simpan	
	2. Menampilkan alert Tolong isikan kolom
<b>Normal Flow: Mengubah batik</b>	
<b>Actor</b>	<b>System</b>
1. Pilih edit pada field aksi	
	2. Menampilkan halaman form ubah

	Batik
3. Mengisi form ubah Batik	
	4. Menampilkan form ubah Batik yang sudah terisi
5. Klik Simpan	
	6. Menampilkan alert Edit data berhasil

***Alternative Flow: Batal Ubah Batik***

<b><i>Actor</i></b>	<b><i>System</i></b>
1. Klik Kembali	
	2. Kembali ke halaman data Batik

***Alternative Flow: Form Belum Terisi***

<b><i>Actor</i></b>	<b><i>System</i></b>
1. Klik Simpan	
	2. Menampilkan <i>alert</i> Tolong isikan kolom

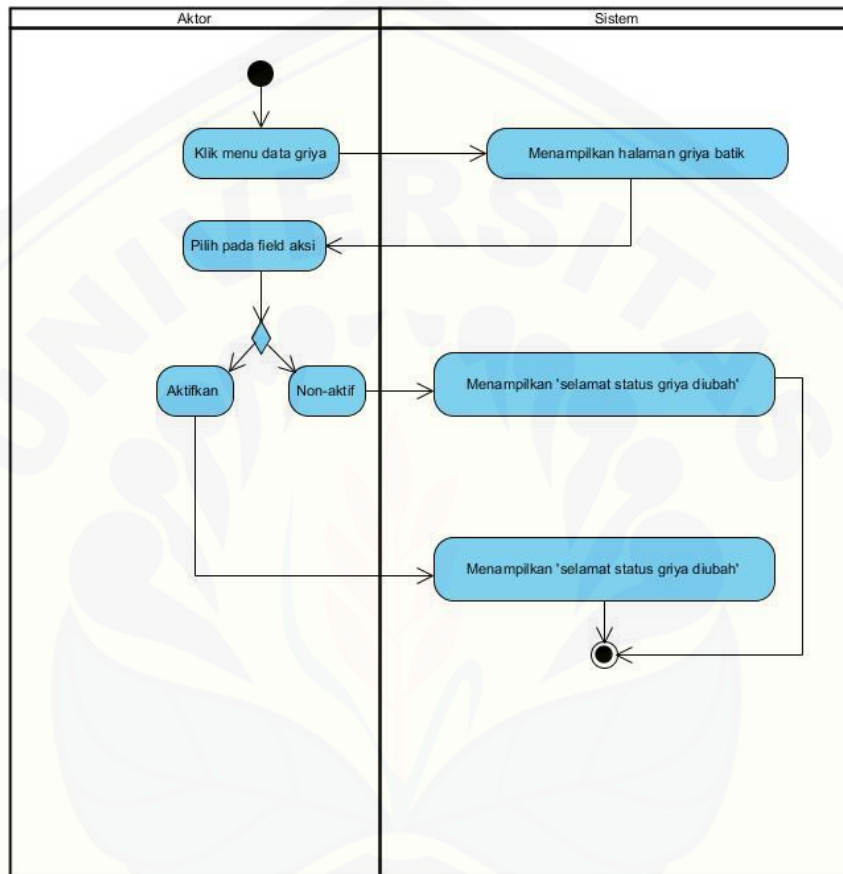
***Normal Flow: Menghapus Data Batik***

<b><i>Actor</i></b>	<b><i>System</i></b>
1. Pilih hapus pada field aksi	
	2. Menampilkan alert Apakah anda yakin menghapus data ini
3. Klik Oke	
	4. Menampilkan alert Sukses hapus Batik

***Alternative Flow: Batal Hapus Data Batik***

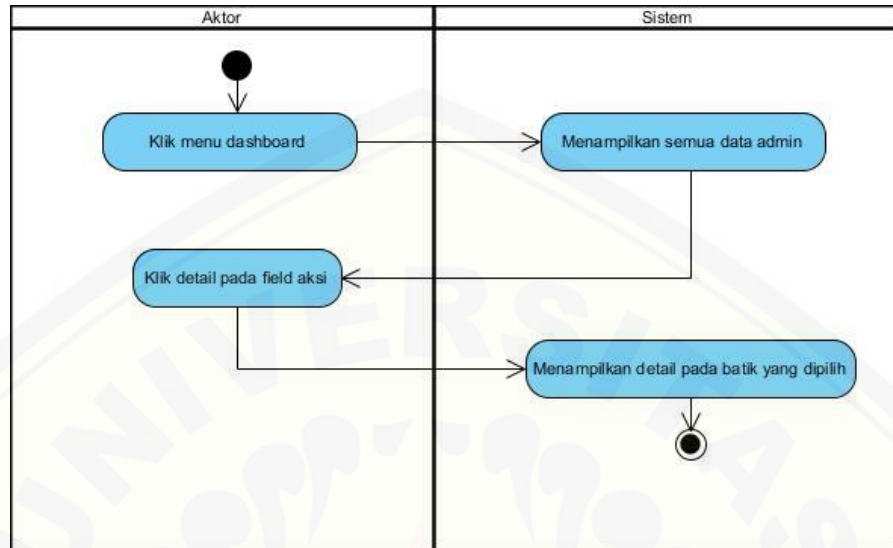
<b><i>Actor</i></b>	<b><i>System</i></b>
1. Klik Kembali	
	2. Kembali ke halaman daftar Batik



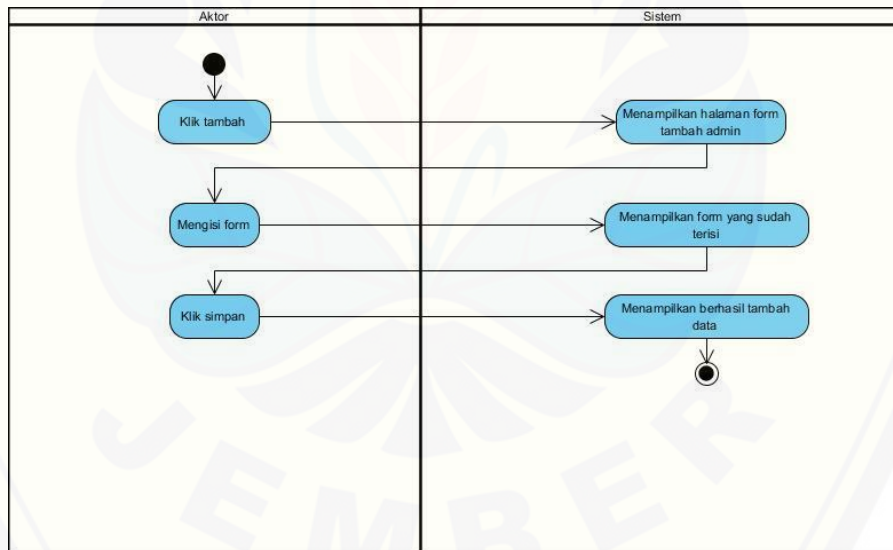
**LAMPIRAN B. ACTIVITY DIAGRAM****B.1 Activity Diagram Mengkonfirmasi Griya Batik**

Gambar B. 1 Activity Diagram Mengkonfirmasi Griya Batik

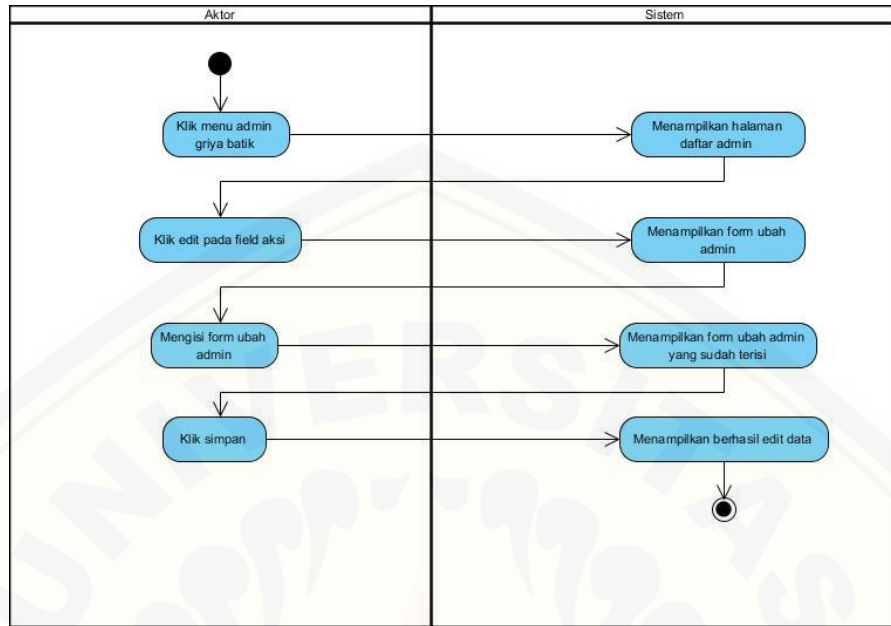
B.2 Activity Diagram Mengelola Data Admin



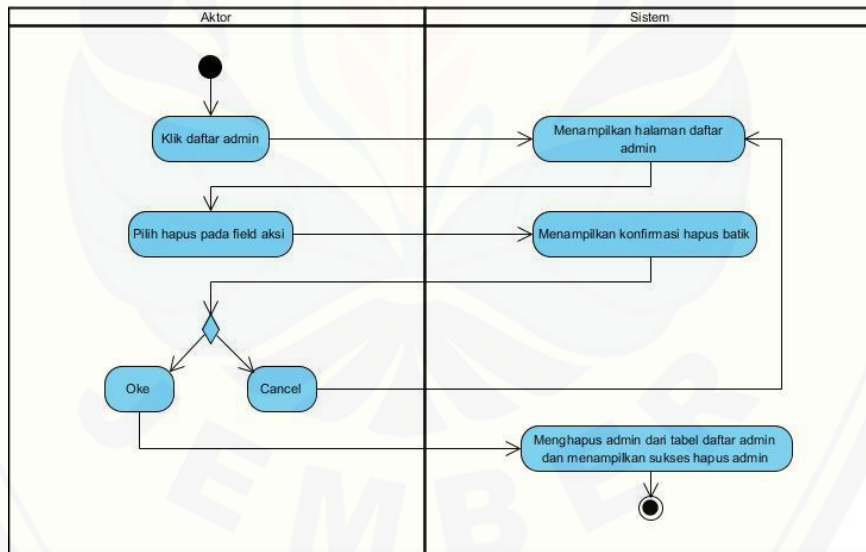
Gambar B. 2 Activity Diagram Mengelola Data Admin



Gambar B. 3 Activity Diagram Menambah Data Admin

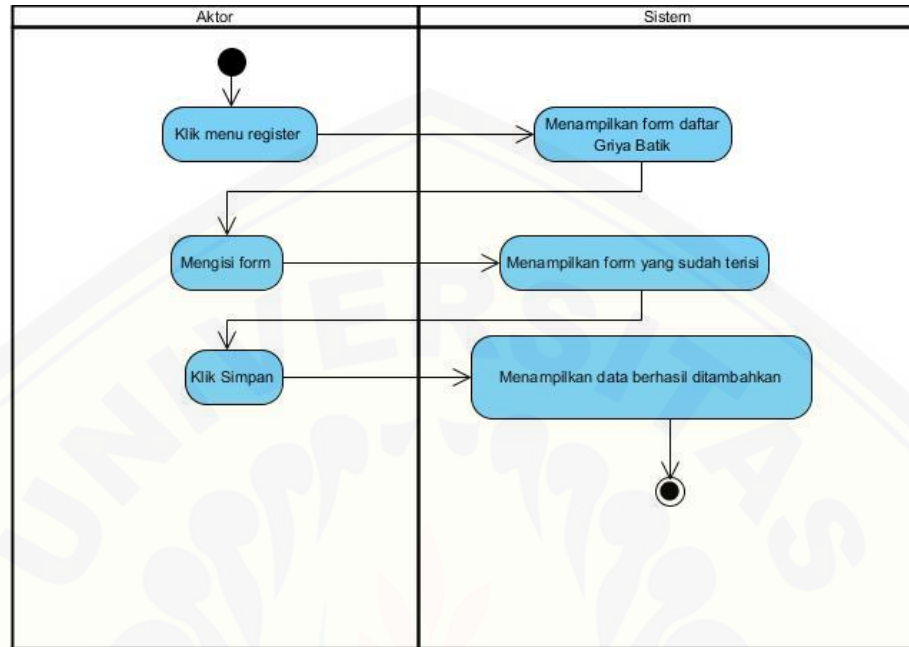


Gambar B. 4 Activity Diagram Mengubah Data Admin



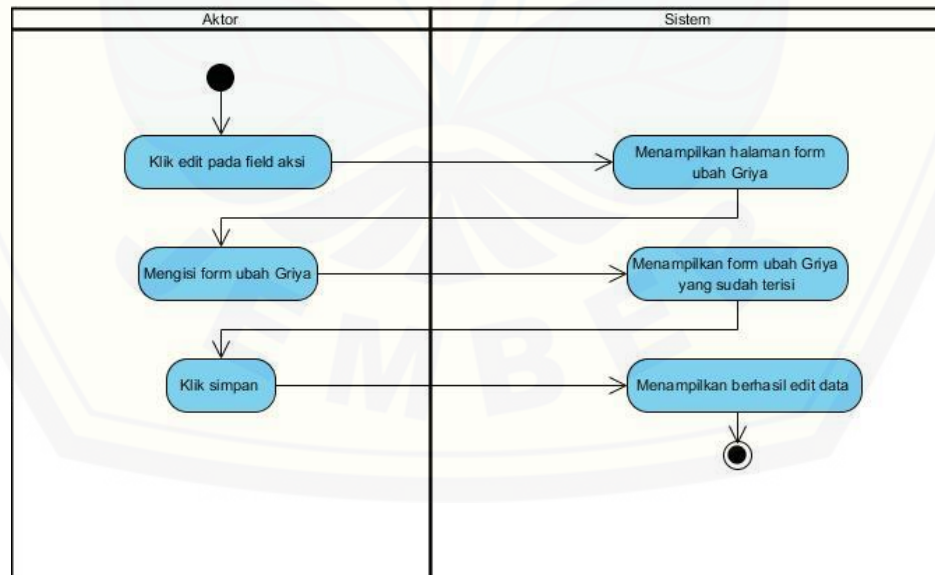
Gambar B. 5 Activity Diagram Menghapus Data Admin

B.3 Activity Diagram Mendaftar Griya Batik



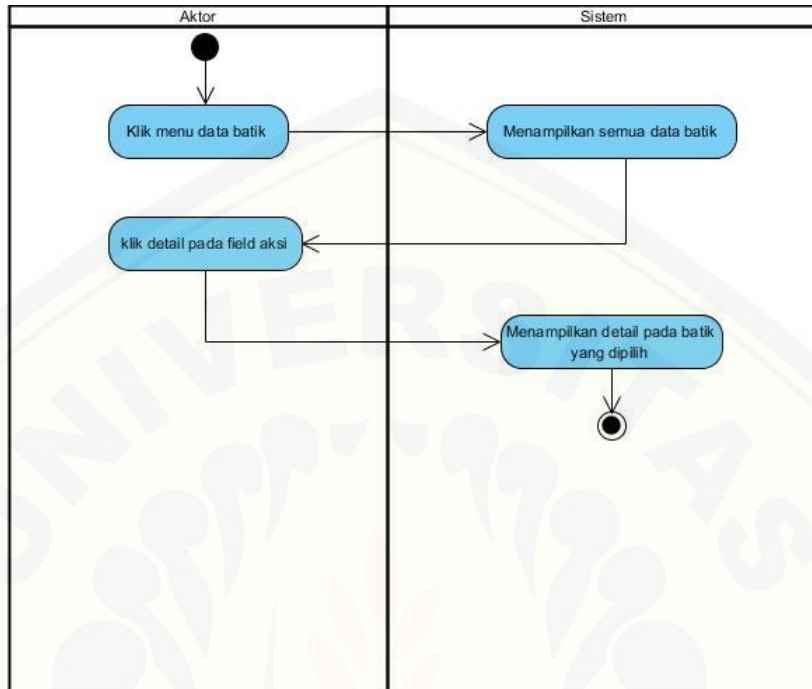
Gambar B. 6 Activity Diagram Mendaftar Griya Batik

B.4 Activity Diagram Mengelola Griya Batik

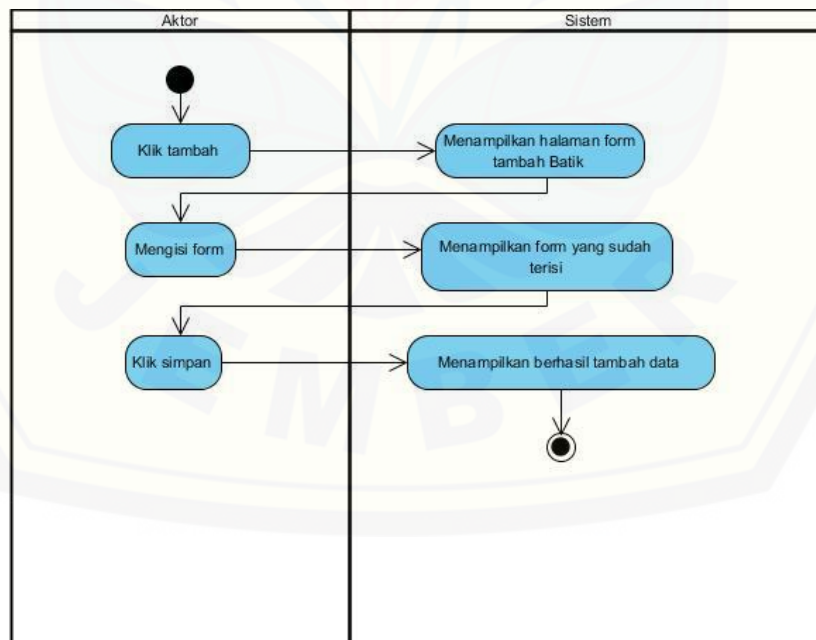


Gambar B. 7 Activity Diagram Mengelola Griya Batik

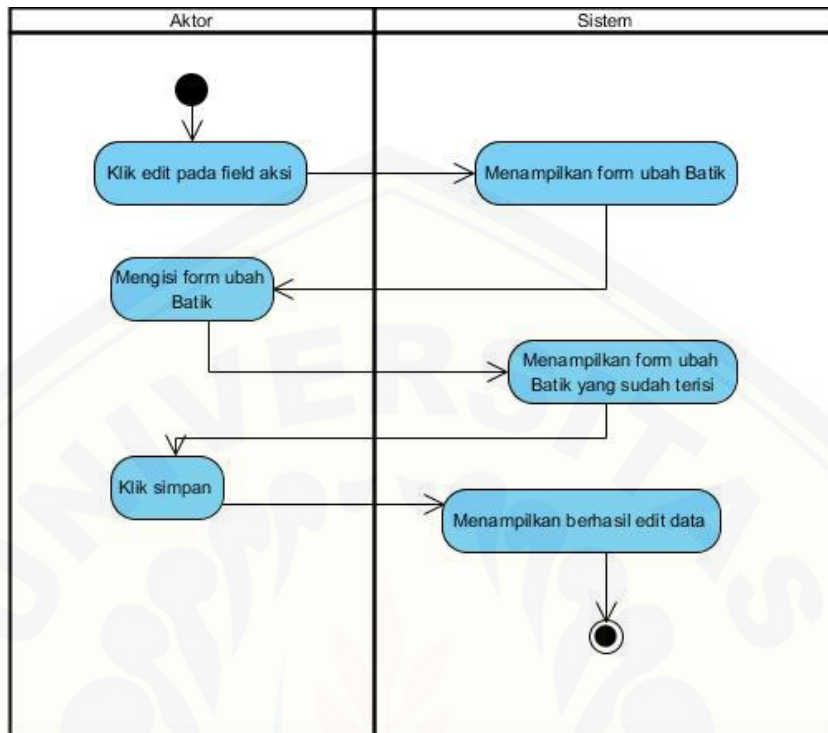
B.5 Activity Diagram Mengelola Data Batik



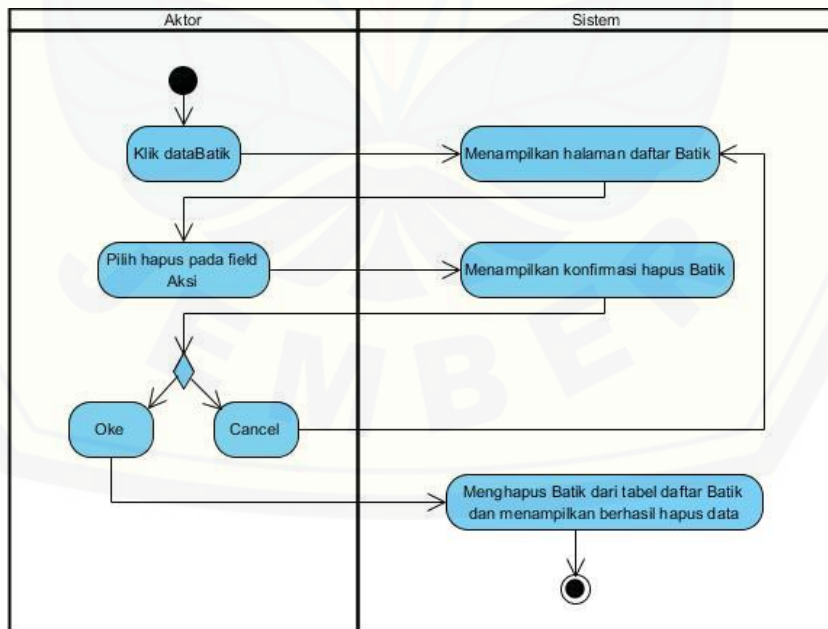
Gambar B. 8 Activity diagram Melihat Data Batik



Gambar B. 9 Activity diagram menambah data batik



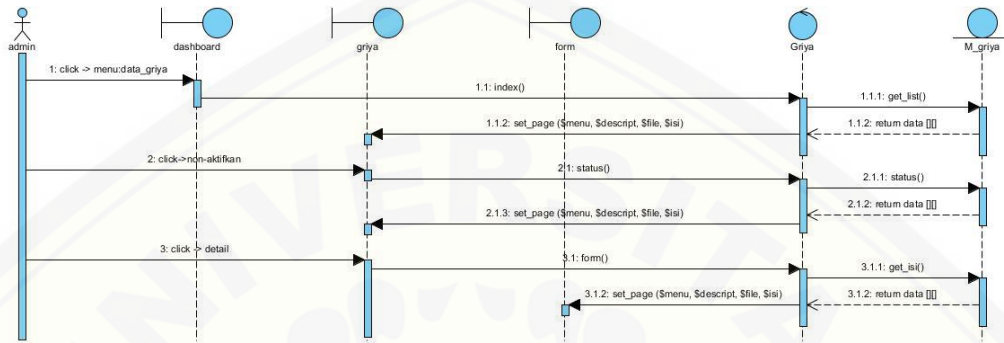
Gambar B. 10 Activity diagram Mengubah Data Batik



Gambar B. 11 Activity diagram Menghapus Data Batik

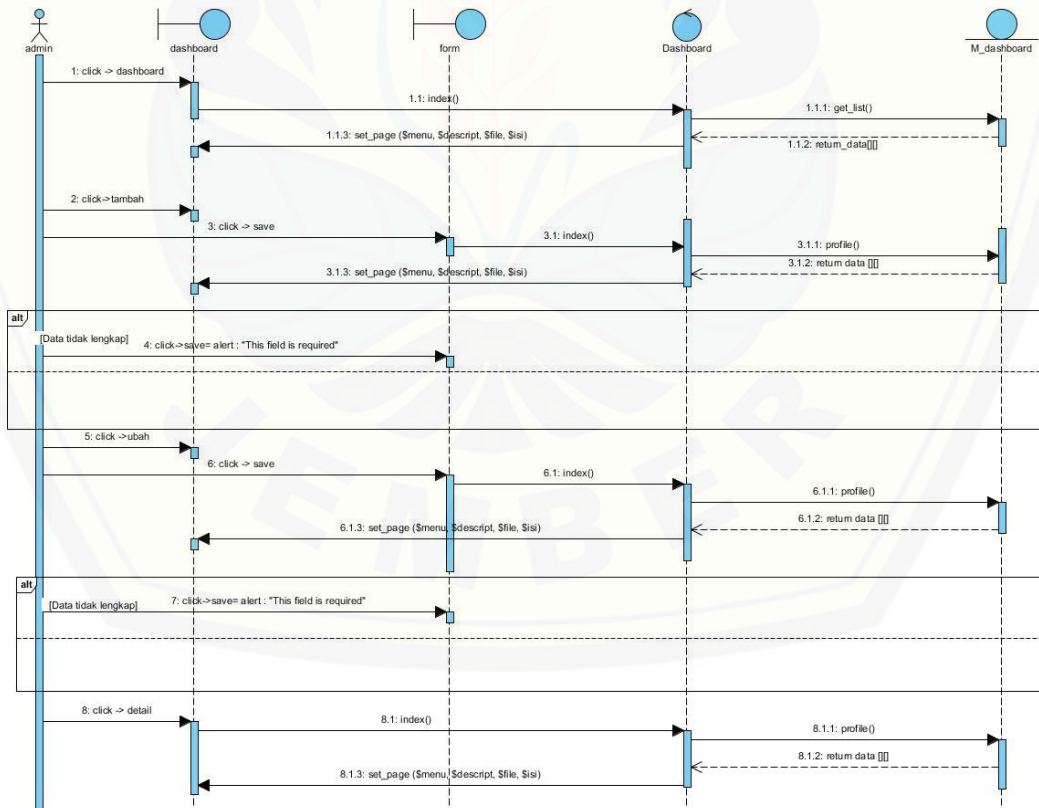
LAMPIRAN C. *SEQUENCE* DIAGRAM

C.1 *Sequence* Diagram Mengkonfirmasi Griya Batik



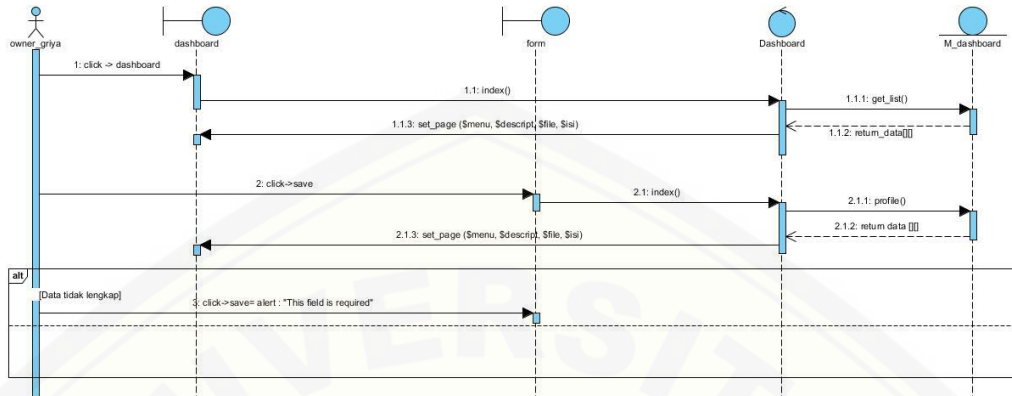
Gambar C. 1 *Sequence* diagram Mengkonfirmasi Griya Batik

C.2 *Sequence* Diagram Mengelola Admin



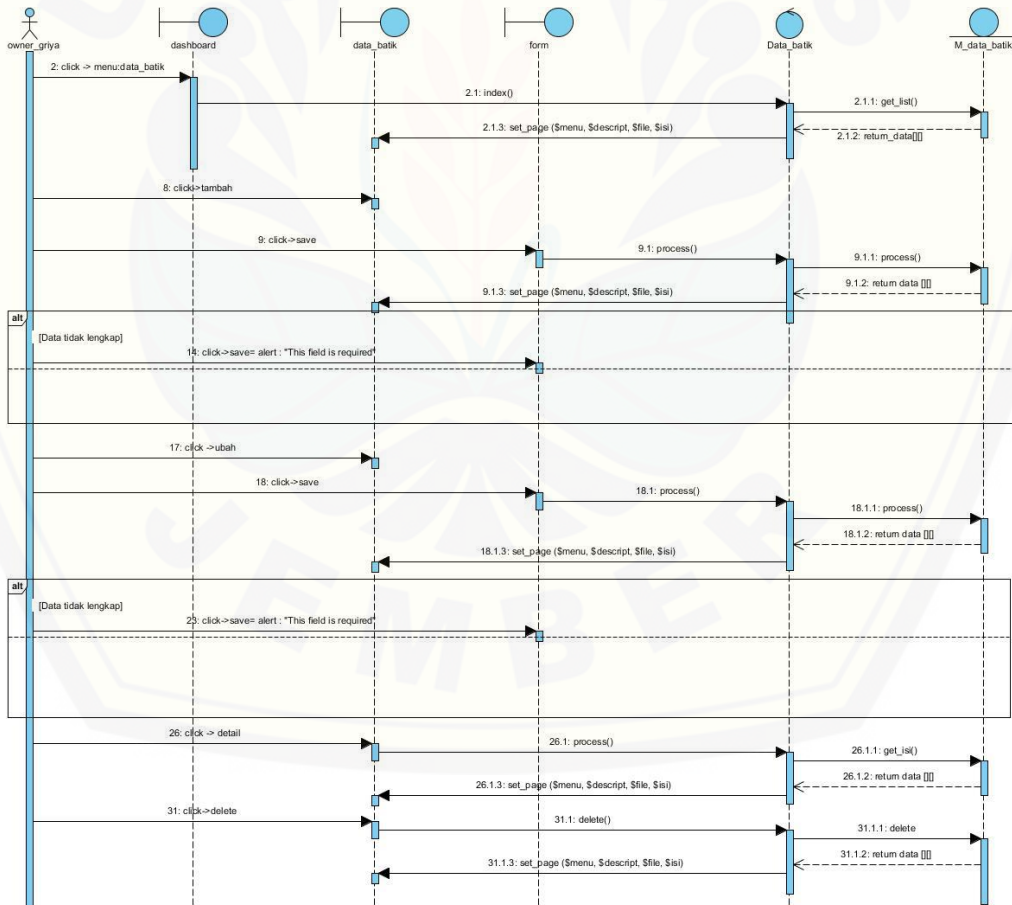
Gambar C. 2 *Sequence* diagram Mengelola Admin

### C.3 Sequence Diagram Mengelola Griya Batik



Gambar C. 3 Sequence diagram mengelola griya batik

### C.4 Sequence Diagram Mengelola Data Batik



Gambar C. 4 Sequence diagram mengelola data batik



**LAMPIRAN D. KODE PROGRAM**D.1. Kode program *class controller*/Dashboard

```
<?php

if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class Dashboard extends RAST_Control {

    private $data = array();

    function __construct() {
        parent::__construct();

        if ($this->session->userdata('level') > 0) {
        } else {
            redirect('login');
        }

        $this->load->model('M_dashboard');
    }

    function index() {
        $sisi = array(
            'descript' => "Info Griya"
            , 'dataTable' => (($this->session->userdata('level') == 1) ? $this-
            >M_dashboard->list_admin() : $this->M_dashboard->profile())
            , 'add_button' => (($this->session->userdata('level') == 1) ? true : false)
            , 'back_button' => false
        );

        $this->set_page('dashboard', $sisi['descript'], 'dashboard', $sisi);
    }

    function form() {
        $this->output->set_content_type('application/json');

        $id = $_POST['id'];
        echo json_encode($this->M_dashboard->get_isi($id));
    }

    function admin_process() {
        if ($this->session->userdata('level') < 1) {
```

```

        redirect('dashboard');
    }

    $post = $_POST;
    if ($this->M_dashboard->admin_process($post)) {
        $this->session->set_userdata('pesan_sistem', 'Selamat! ' . (($post['id'] == '') ?
'Penambahan' : 'Perubahan') . ' data admin, SUKSES!');
        $this->session->set_userdata('tipe_pesan', 'Sukses');
        redirect('dashboard');
    } else {
        $this->session->set_userdata('pesan_sistem', 'Maaf! ' . (($post['id'] == '') ?
'Penambahan' : 'Perubahan') . ' data admin, GAGAL! Silahkan periksa dan coba kembali');
        $this->session->set_userdata('tipe_pesan', 'Gagal');
        redirect('dashboard');
    }
}

function set_page($menu, $descript, $file, $isi) {
    $data['menu'] = $menu;
    $data['descript'] = $descript;

    $data['content'] = $this->load->view($file, $isi, TRUE);
    $this->load->view('template/template', $data);
}
}

```

## D.2. Kode program *class controller/Data\_batik*

```

<?php

if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class Data_batik extends RAST_Control {

    private $data;

    function __construct() {
        parent::__construct();

        $this->load->model('M_data_batik');
    }

    function index() {
        if ($this->session->userdata('level') < 1) {
            redirect('dashboard');
        }
    }
}

```

```

    }

    $sisi = array(
        'descript' => 'Data Batik'
        , 'dataTable' => $this->M_data_batik->get_list()
        , 'add_button' => true
        , 'back_button' => false
    );

    $this->set_page('data_batik', $sisi['descript'], 'data_batik', $sisi);
}

function form() {
    $this->output->set_content_type('application/json');

    $id = $_POST['id'];
    echo json_encode($this->M_data_batik->get_isi($id));
}

function process() {
    if ($this->session->userdata('level') < 1) {
        redirect('dashboard');
    }

    $post1 = $_POST;

    $file = rand(1000,100000)."_" . $_FILES['foto']['name'];
    $file_loc = $_FILES['foto']['tmp_name'];
    $file_size = $_FILES['foto']['size'];
    $file_type = $_FILES['foto']['type'];
    $folder="assets/images/photos/batik/";

    if (move_uploaded_file($file_loc,$folder.$file)) {
        $post = array_merge($post1, array("foto" => $file));

        if ($this->M_data_batik->process($post)) {
            $this->session->set_userdata('pesan_sistem', 'Selamat! ' .
(($post['id'] == "") ? 'Penambahan' : 'Perubahan') . ' data_batik, SUKSES!');
            $this->session->set_userdata('tipe_pesan', 'Sukses');
        } else {
            $this->session->set_userdata('pesan_sistem', 'Maaf! ' .
(($post['id'] == "") ? 'Penambahan' : 'Perubahan') . ' data_batik, GAGAL! Silahkan periksa
dan coba kembali');
            $this->session->set_userdata('tipe_pesan', 'Gagal');
        }
    } else {

```

```

        if ($this->M_data_batik->process($post)) {
            $this->session->set_userdata('pesan_sistem', 'Selamat! ' .
(($post['id'] == "") ? 'Penambahan' : 'Perubahan') . ' data_batik, SUKSES!');
            $this->session->set_userdata('tipe_pesan', 'Sukses');
        } else {
            $this->session->set_userdata('pesan_sistem', 'Maaf! ' .
(($post['id'] == "") ? 'Penambahan' : 'Perubahan') . ' data_batik, GAGAL! Silahkan periksa
dan coba kembali');
            $this->session->set_userdata('tipe_pesan', 'Gagal');
        }
    }
    redirect('data_batik');
}

function delete($a) {
    if ($this->session->userdata('level') < 1) {
        redirect('dashboard');
    }

    if ($this->M_data_batik->delete($a)) {
        $this->session->set_userdata('pesan_sistem', 'Selamat! User telah dihapus!');
        $this->session->set_userdata('tipe_pesan', 'Sukses');
    } else {
        $this->session->set_userdata('pesan_sistem', 'Maaf! User tidak terhapus! Silahkan
periksa dan coba kembali');
        $this->session->set_userdata('tipe_pesan', 'Gagal');
    }

    redirect('data_batik');
}

function set_page($menu, $descript, $file, $isi) {
    $data['menu'] = $menu;
    $data['descript'] = $descript;

    $data['content'] = $this->load->view($file, $isi, TRUE);
    $this->load->view('template/template', $data);
}
}
}

```

### D.3. Kode program class *controller*/Disease

```
<?php
```

```
if (!defined('BASEPATH'))
```

```
exit('No direct script access allowed');

class Disease extends RAST_Control {

    private $data;

    function __construct() {
        parent::__construct();

        if ($this->session->userdata('level') != 2) {
            redirect('dashboard');
        }

        $this->load->model('M_disease');
    }

    function index() {
        $sisi = array(
            'descript' => 'Kelola Admin'
            , 'dataTable' => $this->M_disease->get_list()
            , 'add_button' => true
            , 'back_button' => false
        );

        $this->set_page('disease', $sisi['descript'], 'disease', $sisi);
    }

    function form() {
        $this->output->set_content_type('application/json');

        $sid = $_POST['id'];
        echo json_encode($this->M_disease->get_isi($sid));
    }

    function process() {
        $post = $_POST;
        if ($this->M_disease->process($post)) {
            $this->session->set_userdata('pesan_sistem', 'Selamat! ' . (($post['id'] == "") ?
            'Penambahan' : 'Perubahan') . ' pelayanan, SUKSES!');
            $this->session->set_userdata('tipe_pesan', 'Sukses');
            redirect('disease');
        } else {
            $this->session->set_userdata('pesan_sistem', 'Maaf! ' . (($post['id'] == "") ?
            'Penambahan' : 'Perubahan') . ' penyakit, GAGAL! Silahkan periksa dan coba kembali');
            $this->session->set_userdata('tipe_pesan', 'Gagal');
            redirect('disease');
        }
    }
}
```

```

    }
}

function delete($a) {
    if ($this->M_disease->delete($a)) {
        $this->session->set_userdata('pesan_sistem', 'Selamat! Penyakit telah dihapus!');
        $this->session->set_userdata('tipe_pesan', 'Sukses');
        redirect('disease');
    } else {
        $this->session->set_userdata('pesan_sistem', 'Maaf! Penyakit tidak terhapus!
Silahkan periksa dan coba kembali);
        $this->session->set_userdata('tipe_pesan', 'Gagal');
        redirect('disease');
    }
}

function set_page($menu, $descript, $file, $sisi) {
    $data['menu'] = $menu;
    $data['descript'] = $descript;

    $data['content'] = $this->load->view($file, $sisi, TRUE);
    $this->load->view('template/template', $data);
}
}
}

```

#### D.4. Kode program *class controller/Griya*

```

<?php

if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class Griya extends RAST_Control {

    private $data;

    function __construct() {
        parent::__construct();

        if ($this->session->userdata('level') != 1) {
            redirect('dashboard');
        }

        $this->load->model('M_griya');
    }
}

```

```

function index() {
    $sisi = array(
        'descript' => 'Data Griya Batik'
        , 'dataTable' => $this->M_griya->get_list()
        , 'add_button' => false
        , 'back_button' => false
    );

    $this->set_page('griya', $sisi['descript'], 'griya', $sisi);
}

function form() {
    $this->output->set_content_type('application/json');

    $id = $_POST['id'];
    echo json_encode($this->M_griya->get_isi($id));
}

function status($a) {
    if ($this->M_griya->status($a)) {
        $this->session->set_userdata('pesan_sistem', 'Selamat! Status griya telah diubah!');
        $this->session->set_userdata('tipe_pesan', 'Sukses');
        redirect('griya');
    } else {
        $this->session->set_userdata('pesan_sistem', 'Maaf! Status griya gagal diubah!
Silahkan periksa dan coba kembali');
        $this->session->set_userdata('tipe_pesan', 'Gagal');
        redirect('griya');
    }
}

function set_page($menu, $descript, $file, $sisi) {
    $data['menu'] = $menu;
    $data['descript'] = $descript;

    $data['content'] = $this->load->view($file, $sisi, TRUE);
    $this->load->view('template/template', $data);
}
}
}

```

#### D.5. Kode program *class controller/Home*

```

<?php
if (!defined('BASEPATH'))
    exit('No direct script access allowed');

```

```
class Home extends RAST_Control {

    private $data;

    function __construct() {
        parent::__construct();

        if ($this->session->userdata('level') > 0) {
            redirect('dashboard');
        }

        $this->load->model('M_home');
    }

    function index() {
        $this->session->set_userdata('home', (($this->session->userdata('home') ==
FALSE) ? FALSE : TRUE));

        $sisi = array(
            'descript' => 'Data Batik'
            , 'dataTable' => $this->M_home->get_list()
            , 'harga' => NULL
            , 'pewarnaan' => NULL
            , 'motif' => NULL
            , 'jenis' => NULL
            , 'add_button' => false
            , 'back_button' => false
        );

        if (isset($_POST['harga'])) {
            $sisi['harga'] = $_POST['harga'];
            $sisi['pewarnaan'] = $_POST['pewarnaan'];
            $sisi['motif'] = $_POST['motif'];
            $sisi['jenis'] = $_POST['jenis'];

            $sisi['dataTable'] = $this->M_home->do_wp($sisi['harga'],
$sisi['pewarnaan'], $sisi['motif'], $sisi['jenis']);
        }

        $this->set_page('home', $sisi['descript'], 'home', $sisi);
    }

    function change_home() {
        $this->session->set_userdata('home', (($this->session->userdata('home')
== FALSE) ? TRUE : FALSE));
    }
}
```



```
    }  
  
    function set_page($menu, $descript, $file, $isi) {  
        $data['menu'] = $menu;  
        $data['descript'] = $descript;  
  
        $data['content'] = $this->load->view($file, $isi, TRUE);  
        $this->load->view('template/template', $data);  
    }  
}
```

#### D.6. Kode program *class controller/Login*

```
<?php  
if (!defined('BASEPATH')) exit('No direct script access allowed');  
  
class Login extends RAST_Control {  
  
    private $data;  
  
    public function __construct() {  
        parent::__construct();  
  
        $this->load->model('M_login');  
    }  
  
    function index() {  
        if ($this->M_login->getUser() > 0) {  
            redirect('dashboard');  
        } else {  
            $this->load->view('login');  
        }  
    }  
  
    function register() {  
        if ($this->M_login->getUser() > 0) {  
            redirect('dashboard');  
        } else {  
            $this->load->view('register');  
        }  
    }  
  
    //=====  
    // End of View  
    //=====
```

```
//
//
//
//
//
//=====
// Process
//=====

function site() {
  if ($_POST) {
    $u = $_POST['username'];
    $p = $_POST['password'];

    if (empty($u) || empty($p)) {
      redirect('login');
    } else {
      if ($this->M_login->login($u, $p)) {
        $this->session->set_userdata('pesan_sistem', 'Selamat! Anda berhasil login!');
        $this->session->set_userdata('tipe_pesan', 'Sukses');
        redirect('login');
      } else {
        $this->session->set_userdata('pesan_sistem', 'Username atau Password
SALAH!<br />Silahkan periksa dan coba kembali');
        $this->session->set_userdata('tipe_pesan', 'Gagal');
        redirect('login');
      }
    }
  }
}

function process() {
  $post1 = $_POST;

  $file1 = (($_FILES['foto']['name'] != "") ?
rand(1000,100000)."_" . $_FILES['foto']['name'] : "");
  $file_loc1 = $_FILES['foto']['tmp_name'];
  $file_size1 = $_FILES['foto']['size'];
  $file_type1 = $_FILES['foto']['type'];
  $folder1="assets/images/photos/griya/";

  $file2 = (($_FILES['surat']['name'] != "") ?
rand(1000,100000)."_" . $_FILES['surat']['name'] : "");
  $file_loc2 = $_FILES['surat']['tmp_name'];
  $file_size2 = $_FILES['surat']['size'];
  $file_type2 = $_FILES['surat']['type'];
```

```
$folder2="assets/images/photos/surat/";

if ($post1['id'] == " && (move_uploaded_file($file_loc1,$folder1.$file1)
&& move_uploaded_file($file_loc2,$folder2.$file2))) {
    $post = array_merge(array_merge($post1, array("foto" =>
$file1)), array("surat" => $file2));

    if ($this->M_login->process($post)) {
        $this->session->set_userdata('pesan_sistem', 'Selamat!
Pendaftaran Griya Batik, SUKSES!');
        $this->session->set_userdata('tipe_pesan', 'Sukses');
    } else {
        $this->session->set_userdata('pesan_sistem', 'Maaf!
Pendaftaran Griya Batik, GAGAL! Silahkan periksa dan coba kembali');
        $this->session->set_userdata('tipe_pesan', 'Gagal');
    }
} else if ($post1['id'] != " &&
(move_uploaded_file($file_loc1,$folder1.$file1) ||
move_uploaded_file($file_loc2,$folder2.$file2))) {
    $post = array_merge(array_merge($post1, array("foto" =>
$file1)), array("surat" => $file2));

    if ($this->M_login->process($post)) {
        $this->session->set_userdata('pesan_sistem', 'Selamat!
Pendaftaran Griya Batik, SUKSES!');
        $this->session->set_userdata('tipe_pesan', 'Sukses');
    } else {
        $this->session->set_userdata('pesan_sistem', 'Maaf!
Pendaftaran Griya Batik, GAGAL! Silahkan periksa dan coba kembali');
        $this->session->set_userdata('tipe_pesan', 'Gagal');
    }
} else {
    $this->session->set_userdata('pesan_sistem', 'Maaf! Pendaftaran
Griya Batik, GAGAL! Silahkan periksa dan coba kembali');
    $this->session->set_userdata('tipe_pesan', 'Gagal');
}
redirect('login');
}

function logout() {
    $this->session->sess_destroy();
    redirect('login');
}

//=====
// End of Process
```

```
//=====
}
```

#### D.7. Kode program *class modelM\_dashboard*

```
<?php
if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class M_dashboard extends RAST_Model {

    function list_admin() {
        $query = $this->db->query
            (
                SELECT
                    *
                FROM
                    user
                WHERE
                    status = 1
                    AND level = 1
            );

        return $query->result_array();
    }

    function profile() {
        $query = $this->db->query
            (
                SELECT
                    u.*
                    , g.alamat
                    , g.telpon
                    , g.foto
                    , g.surat
                FROM
                    user u
                    , griya g
                WHERE
                    status = 1
                    AND level = 2
                    AND u.id = g.user
                    AND u.id = ' . $this->session->userdata('user_id')
            );
    }
}
```

```

        return $query->result_array();
    }

    function get_isi($a) {
        $query = $this->db->query
        (
            SELECT
                *
            FROM
                user
            WHERE
                id = ' . $a
        );

        return $query->result_array();
    }

    function admin_process($a) {
        $query = FALSE;
        if ($a['id'] == "") {
            $query = $this->db->query
            (
                INSERT INTO user VALUES
                (
                    ' . "" . "" . ' .
                    ' . "" . $a['username'] . "" . ' .
                    ' . "" . md5($a['password']) . "" . ' .
                    ' . "" . $a['nama'] . "" . ' .
                    , 1
                    , 1
                )
            );
        } else {
            $query = $this->db->query
            (
                UPDATE user SET
                nama = ' . "" . $a['nama'] . "" . ' .
                , username = ' . "" . $a['username'] . "" . ' .
                ' . (($a['password'] != "") ? (' , password = ' . "MD5(" . $a['password'] . ")")
                : ") . ' .
                WHERE
                    id = ' . $a['id']
            );
        }

        if ($query) {

```

```
        return TRUE;
    } else {
        return FALSE;
    }
}

function delete($a) {
    $query = $this->db->query
    (
        UPDATE user SET
            status = 0
        WHERE
            id = ' . $a
    );

    if ($query) {
        return TRUE;
    } else {
        return FALSE;
    }
}
}
```

#### D.8. Kode program *class model/M\_data\_batik*

```
<?php

if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class M_data_batik extends RAST_Model {

    function get_list() {
        $query = $this->db->query
        (
            SELECT
                b.*
            FROM
                batik b
                , user u
                , griya g
            WHERE
                b.status = 1
                AND u.id = g.user
                AND g.id = b.griya
                AND u.id = ' . $this->session->userdata('user_id')
```

```

    );

    return $query->result_array();
}

function get_isi($a) {
    $query = $this->db->query
    (
        SELECT
        *
        FROM
        batik
        WHERE
        id = ' . $a
    );

    return $query->result_array();
}

function process($a) {
    $query = FALSE;

    if ($a['id'] == "") {
        $query = $this->db->query
        (
            SELECT
            id
            FROM
            griya
            WHERE
            user = ' . $this->session-
>userdata('user_id')
        );

        $id = $query->row()->id;

        $query = $this->db->query
        (
            INSERT INTO batik VALUES
            (
                ' . "" . "" . '
                , ' . $id . '
                , ' . "" . $a['nama'] . "" . '
                , ' . "" . $a['harga'] . "" . '
                , ' . "" . $a['pewarnaan'] . "" . '
                , ' . "" . $a['motif'] . "" . '
            )
        )
    }
}

```

```

        , ' . "" . $a['jenis_kain'] . "" . '
        , ' . "" . ((isset($a['foto'])) ?
$a['foto'] : '--') . "" . '
        , ' . "" . $a['deskripsi'] . "" . '
        , 1
    )
    );
//      , ' . "" . $a['foto'] . "" . '
    } else {
        $query = $this->db->query
        (
            UPDATE batik SET
            nama = ' . "" . $a['nama'] . "" . '
            , harga = ' . "" . $a['harga'] . "" . '
            , pewarnaan = ' . "" .
$a['pewarnaan'] . "" . '
            , motif = ' . "" . $a['motif'] . "" . '
            , jenis_kain = ' . "" .
$a['jenis_kain'] . "" . '
            , foto = ' . "" . ((isset($a['foto'])) ?
$a['foto'] : '--') . "" . '
            , deskripsi = ' . "" . $a['deskripsi']
. "" . '
            WHERE
            id = ' . $a['id']
        );
//      , foto = ' . "" . $a['foto'] . "" . '
    }
    if ($query) {
        return TRUE;
    } else {
        return FALSE;
    }
}

function delete($a) {
    $query = $this->db->query
    (
        UPDATE batik SET
        status = 0
        WHERE
        id = ' . $a
    );
    if ($query) {

```



```
        return TRUE;
    } else {
        return FALSE;
    }
}
}
```

#### D.9. Kode program *class model/M\_disease*

```
<?php
if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class M_disease extends RAST_Model {

    function get_list() {
        $query = $this->db->query
        (
            SELECT
            *
            FROM
            diseases
            WHERE
            status = 1
            ');

        return $query->result_array();
    }

    function get_isi($a) {
        $query = $this->db->query
        (
            SELECT
            *
            FROM
            diseases
            WHERE
            id = ' . $a

        );

        return $query->result_array();
    }

    function process($a) {
```

```
$query = FALSE;
if ($a['id'] == "") {
    $query = $this->db->query
    (
        INSERT INTO diseases VALUES
        (
            ' . "" . "" . '
            , ' . "" . $a['name'] . "" . '
            , ' . "" . $a['symptom'] . "" . '
            , 1
        )
    );
} else {
    $query = $this->db->query
    (
        UPDATE diseases SET
        name = ' . "" . $a['name'] . "" . '
        , symptom = ' . "" . $a['symptom'] . "" . '
        WHERE
        id = ' . $a['id']
    );
}

if ($query) {
    return TRUE;
} else {
    return FALSE;
}
}

function delete($a) {
    $query = $this->db->query
    (
        UPDATE diseases SET
        status = 0
        WHERE
        id = ' . $a
    );

    if ($query) {
        return TRUE;
    } else {
        return FALSE;
    }
}
}
```

D.10. Kode program *class model/M\_griya*

```
<?php
if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class M_griya extends RAST_Model {

    function get_list() {
        $query = $this->db->query
            ('
                SELECT
                    u.*
                    , g.*
                FROM
                    user u
                    , griya g
                WHERE
                    level = 2
                    AND u.id = g.user
            ');

        return $query->result_array();
    }

    function get_isi($a) {
        $query = $this->db->query
            ('
                SELECT
                    u.*
                    , g.alamat
                    , g.telpon
                    , g.foto
                    , g.surat
                FROM
                    user u
                    , griya g
                WHERE
                    level = 2
                    AND u.id = g.user
                    AND u.id = ' . $a

            ');

        return $query->result_array();
    }
}
```

```
function status($a) {
    $query = $this->db->query
        (
            UPDATE user SET
                status = IF(status = 0, 1, IF(status = 1, 0, status))
            WHERE
                id = ' . $a
        );

    if ($query) {
        return TRUE;
    } else {
        return FALSE;
    }
}
```

#### D.11. Kode program *class model/M\_home*

```
<?php

if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class M_home extends RAST_Model {

    function get_list() {
        $query = $this->db->query
            (
                SELECT
                    b.*
                    , u.nama AS nama_griya
                FROM
                    batik b
                    , user u
                    , griya g
                WHERE
                    g.user = u.id
                    AND b.griya = g.id
                    AND u.status = 1
                    AND b.status = 1
            );

        return $query->result_array();
    }
}
```

```
function do_wp($harga, $pewarnaan, $motif, $jenis) {
    $query = $this->db->query
    (
        SELECT
        b.*
        , u.nama AS nama_griya
        FROM
        batik b
        , user u
        , griya g
        WHERE
        g.user = u.id
        AND b.griya = g.id
        AND u.status = 1
        AND b.status = 1
        ');

    $list = $query->result_array();
    $list2 = array(array());

    for ($i = 0; $i < count($list); $i++) {
        if ($list[$i]['harga'] >= 500000) {
            $v_harga = 5;
        } elseif ($list[$i]['harga'] >= 300000 && $list[$i]['harga'] <
500000) {
            $v_harga = 3;
        } elseif ($list[$i]['harga'] < 300000) {
            $v_harga = 1;
        }

        if ($list[$i]['pewarnaan'] == "Alami") {
            $v_pewarnaan = 5;
        } else {
            $v_pewarnaan = 1;
        }

        if ($list[$i]['motif'] == "Tembakau") {
            $v_motif = 5;
        } else {
            $v_motif = 1;
        }

        if ($list[$i]['jenis_kain'] == "Sutra") {
            $v_jenis_kain = 5;
        } else {
            $v_jenis_kain = 1;
        }
    }
}
```

```

    }

    $list2[$i] = array(
        0 => $list[$i]['id']
        , 1 => $v_harga
        , 2 => $v_pewarnaan
        , 3 => $v_motif
        , 4 => $v_jenis_kain
    );
}

$sTot = 0;
$s = array();
$wTotal = ($harga + $pewarnaan + $motif + $jenis);
$wj = array(
    0 => ($harga / $wTotal)
    , 1 => ($pewarnaan / $wTotal)
    , 2 => ($motif / $wTotal)
    , 3 => ($jenis / $wTotal)
);
$jk = array(
    0 => 0
    , 1 => 1
    , 2 => 1
    , 3 => 1
);
// echo '<br /><br /><br /><br /><br /><br />';
// echo 'bobot -> ' . $wTotal . '<br />';
// print_r($wj);
for ($i = 0; $i < count($list2); $i++) { // Hitung S
    $s[$i] = 0;
// echo '=====<br />';
    for ($j = 1; $j < count($list2[$i]); $j++) {
        if ($jk[$j - 1] == 0) {
// echo '<br /> masuk 0 - ';
            if ($j != 1) {
// echo ' [(!=1 sebelum) -> ' . $s[$i] . ' ]';
// echo ' [proses -> ' . $list2[$i][$j] . ' ^ ' . ((-
1) * $wj[( $j - 1)]) . ' = ' . pow($list2[$i][$j], ((-1) * $wj[( $j - 1)])) . ' ]';
                $s[$i] *= pow($list2[$i][$j], ((-1) *
$wj[( $j - 1)]));
// echo ' [(!=1 sesudah) -> ' . $s[$i] . ' ]';
            } else {
// echo ' [(else sebelum) -> ' . $s[$i] . ' ]';
// echo ' [proses -> ' . $list2[$i][$j] . ' ^ ' . ((-
1) * $wj[( $j - 1)]) . ' = ' . pow($list2[$i][$j], ((-1) * $wj[( $j - 1)])) . ' ]';

```

```

//                                     $s[$i] = pow($list2[$i][$j], ((-1) * $wj[(($j
- 1))]);
//                                     echo ' [(else sesudah) -> ' . $s[$i] . ' ]';
//                                     }
// //                                     sTot = parseFloat(sTot) + parseFloat(s[i]);
//                                     } else {
// //                                     echo '<br /> masuk 1 - ';
//                                     if ($j != 1) {
// //                                     echo ' [(!=1 sebelum) -> ' . $s[$i] . ' ]';
// //                                     echo ' [proses -> ' . $list2[$i][$j] . ' ^ ' .
$wj[(($j - 1)] . ' = ' . pow($list2[$i][$j], $wj[(($j - 1))]) . ' ]';
//                                     $s[$i] *= pow($list2[$i][$j], $wj[(($j -
1))]);
// //                                     echo ' [(!=1 sesudah) -> ' . $s[$i] . ' ]';
//                                     } else {
// //                                     echo ' [(else sebelum) -> ' . $s[$i] . ' ]';
// //                                     echo ' [proses -> ' . $list2[$i][$j] . ' ^ ' .
$wj[(($j - 1)] . ' = ' . pow($list2[$i][$j], $wj[(($j - 1))]) . ' ]';
//                                     $s[$i] = pow($list2[$i][$j], $wj[(($j - 1))]);
//                                     echo ' [(else sesudah) -> ' . $s[$i] . ' ]';
//                                     }
// //                                     sTot = parseFloat(sTot) + parseFloat(s[i]);
//                                     }
//                                     }
// //                                     echo '<br /> Nilai S => ' . $s[$i];
//                                     $list[$i][count($list[$i])] = $s[$i];
//                                     $list2[$i][count($list2[$i])] = $s[$i];
// //                                     echo '<br />';
//                                     $sTot += $s[$i];
//                                     }
// //                                     echo 'sTot == ' . $sTot . '<br /><br />';
//                                     for ($i = 0; $i < count($list2); $i++) { // Hitung V
// //                                     $v = ($s[$i] / $sTot) * 100;
// //                                     $v = $s[$i] / $sTot;
// //                                     $list[$i][count($list[$i])] = number_format((float)$v, 2, '.', '');
// //                                     $list[$i][count($list[$i])] = $v;
// //                                     $list2[$i][count($list2[$i])] = number_format((float)$v, 2, '.', '');
// //                                     $list2[$i][count($list2[$i])] = $v;
// //                                     echo 'v- ' . $i . ' -> ' . $v . '<br />';
//                                     }

// //                                     echo 'i == ' . count($list2);
// //                                     for ($i = 1; $i < count($list2); $i++) { // urutkan berdasar nilai di list
indeks terakhir
// //                                     $j = $i;
// //                                     while ($j != 0 && ($list2[$j][count($list2[$j]) - 1]) > $list2[$j -
```

```

1][count($list2[$j - 1] - 1))] {
//          $b = $list2[$j];
//          $list2[$j] = $list2[$j - 1];
//          $list2[$j - 1] = $b;
//          $j--;
//          }
//          }
//          for ($i = 1; $i < count($list); $i++) { // urutkan berdasar nilai di list indeks
terakhir
          $j = $i;
          while ($j != 0 && ($list[$j][count($list[$j]) - 1] > $list[$j -
1][count($list[$j - 1] - 1)])) {
            $b = $list[$j];
            $list[$j] = $list[$j - 1];
            $list[$j - 1] = $b;
            $j--;
          }
        }

//          return array("s" => $s, "v" => $v, "list" => $list2);
//          print_r($list2);
//          return $list;
        }
}

```

#### D.12. Kode program *class model/M\_login*

```

<?php
if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class M_login extends RAST_Model {

    function login($u, $p) {
        $query = $this->db->query
        (
            SELECT
                id
                , nama
                , level
            FROM
                user
            WHERE
                username = " . $u . "
                AND password = " . md5($p) . "

```



```

                                AND status = 1
                                ');

if ($query->num_rows() == 1 && $query->row()->level > 0) {
    $this->session->set_userdata('user_id', $query->row()->id);
    $this->session->set_userdata('user_name', $query->row()->nama);
    $this->session->set_userdata('level', $query->row()->level);

    $this->session->set_userdata('visits_form', FALSE);
    $this->session->set_userdata('report_visits', FALSE);
    $this->session->set_userdata('report_sales', FALSE);
    $this->session->set_userdata('report_restock', FALSE);

    return TRUE;
} else {
    return FALSE;
}
}

function getUser() {
    if ($this->session->userdata('user_id') != FALSE && $this->session->userdata('level')
> 0) {
        return $this->session->userdata('level');
    } else {
        return NULL;
    }
}

function process($a) {
    $query = FALSE;

    if ($a['id'] == "") {
        $query = $this->db->query
        (
            INSERT INTO user VALUES
            (
                , "" . "" . '
                , ' . "" . $a['username'] . "" . '
                , ' . "MD5(" . $a['password'] . ")" . '
                , ' . "" . $a['nama'] . "" . '
                , 2
                , 0
            )
        );

        $query2 = $this->db->query

```

```

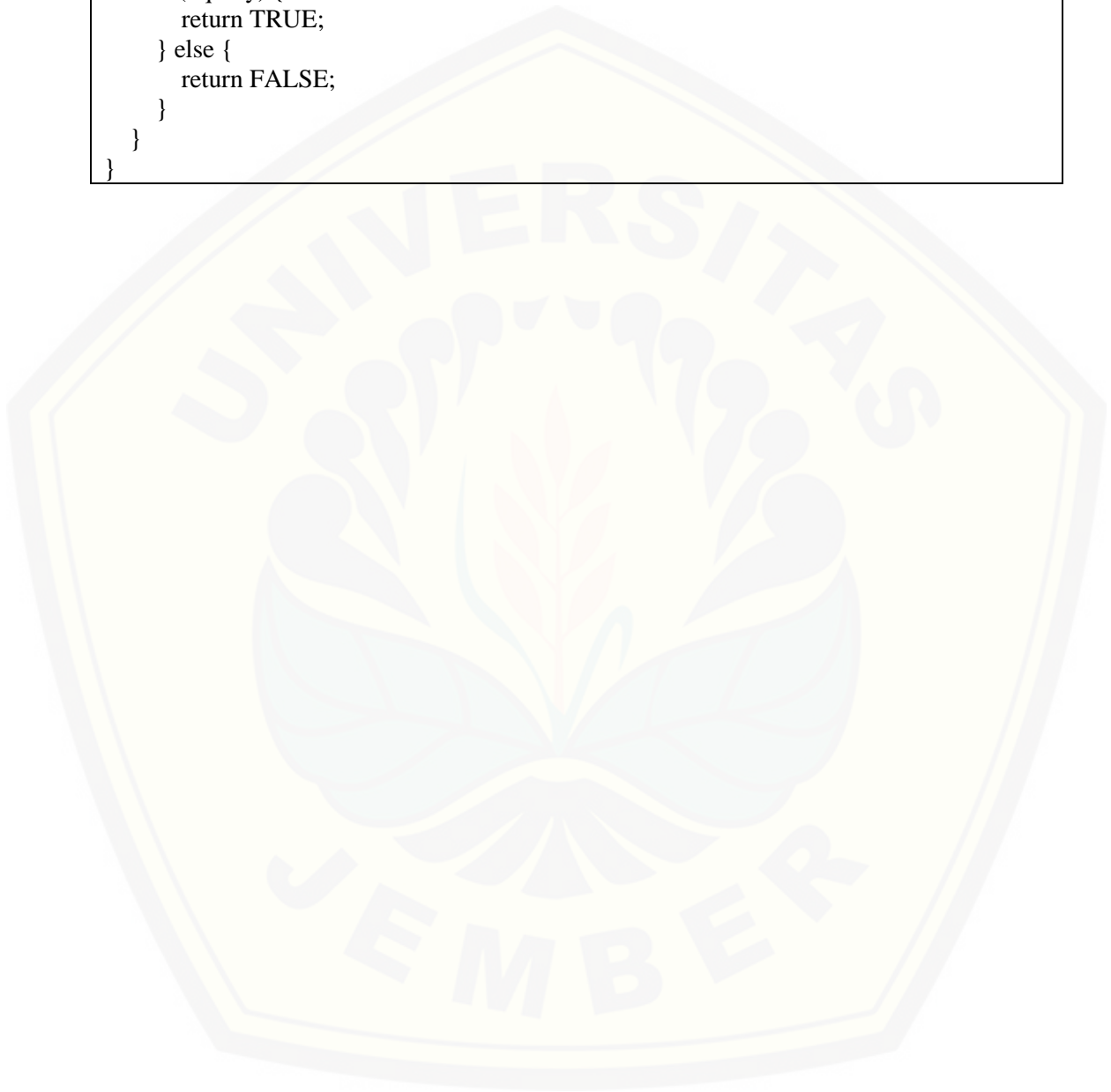
        (
            SELECT
                id
            FROM
                user
            WHERE
                username = "" . $a['username'] . ""
                AND password = "" .
md5($a['password']) . ""
                AND status = 0
        );
        $id = $query2->row()->id;

        $query = $this->db->query
        (
            INSERT INTO griya VALUES
            (
                , ' ' . $id . ' '
                , ' ' . $a['alamat'] . ' ' . '
                , ' ' . $a['telpon'] . ' ' . '
                , ' ' . $a['foto'] . ' ' . '
                , ' ' . $a['surat'] . ' ' . '
            )
        );
    } else {
        $query = $this->db->query
        (
            UPDATE user SET
            username = ' ' . $a['username'] . ' ' . '
            ' . (($a['password'] != "") ? (' , password = ' . "MD5(" . $a['password'] . ")")
: ") . '
            , nama = ' ' . $a['nama'] . ' ' . '
            WHERE
                id = ' . $a['id']
        );

        $query = $this->db->query
        (
            UPDATE griya SET
            alamat = ' ' . $a['alamat'] . ' ' . '
            , telpon = ' ' . $a['telpon'] . ' ' . '
            ' . (($a['foto'] != "") ? (' , foto = ' . $a['foto'] . ' ' . ' : ") . '
            ' . (($a['surat'] != "") ? (' , surat = ' . $a['surat'] . ' ' . ' : ") . '
            WHERE
                user = ' . $a['id']
        )
    }
}

```

```
);  
}  
  
if ($query) {  
    return TRUE;  
} else {  
    return FALSE;  
}  
}  
}
```



LAMPIRAN E. PENGUJIAN *BLACK BOX*

No	Fitur	Aksi	Hasil	Kesimpulan
1	Mengelola data Admin	Klik dashboard	Menampilkan semua data admin disertai tombol “Detail” dan “Ubah”	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Klik detail	Menampilkan detail masing-masing Admin	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Klik ubah	Menampilkan form beserta data yang akan diubah	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
2	Mengkonfirmasi Griya Batik	Klik data griya	Menampilkan semua data griya disertai tombol “Detail” dan “Aktifkan”	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Klik detail	Menampilkan detail masing-masing Griya	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Klik aktifkan	Griya yang diaktifkan berhasil login	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
3	Mendaftar Griya Batik	Klik register	Menampilkan form pendaftaran	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
4	Mengelola Griya Batik	Klik dashboard	Menampilkan detail griya batik	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
5	Mengelola Data Batik	Klik data batik	Menampilkan semua data batik disertai tombol “Tambah”, “Detail”, “Ubah”, dan “Hapus”.	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

		Klik tambah	Menampilkan form baru data batik	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Klik detail	Menampilkan detail data batik	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Klik Ubah	Menampilkan form beserta isi data sebelumnya	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Klik Hapus	Menampilkan pop up “Apakah yakin menghapus data ini?”	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

