



**PENGEMBANGAN SISTEM REKOMENDASI PEMILIHAN *SPARE PART*
PADA BENGKEL LANGGENG JAYA MOTOR DENGAN METODE
*WEIGHTED PRODUCT (WP)***

SKRIPSI

oleh

**Andhika Gita Wardhana
1224101011**

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS JEMBER
2019**



**PENGEMBANGAN SISTEM REKOMENDASI PEMILIHAN *SPARE PART*
PADA BENGKEL LANGGENG JAYA MOTOR DENGAN METODE
*WEIGHTED PRODUCT (WP)***

SKRIPSI

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

oleh

**Andhika Gita Wardhana
122410101011**

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS JEMBER
2019**

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Allah SWT, yang telah memberikan kemudahan dan kelancaran dalam mengerjakan skripsi ini;
2. Ayahanda Nurhadi dan Ibunda Sri Kuniawati;
3. Kakak Aditya Pradhana Putra dan Adik Azzhara Ayuniar Farissa;
4. Seluruh Keluarga Besar;
5. Sahabatku beserta doa dan dukungannya;
6. Guru-guruku sejak sekolah dasar hingga perguruan tinggi;
7. Almamater Program Studi Sistem Informasi Universitas Jember.

MOTTO

“Kerahkan hati, pikiran, dan jiwamu ke dalam aksimu yang paling kecil
sekalipun. Inilah rahasia kesuksesan”

(Swami Sivananda)



PERNYATAAN

Saya yang bertanda tangan dibawah ini:

Nama : Andhika Gita Wardhana

NIM : 122410101011

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Pengembangan Sistem Rekomendasi Pemilihan *Spare Part* Pada Bengkel Langgeng Jaya Motor Dengan Metode *Weighted Product (WP)*”, 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 adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika di kemudian hari pernyataan ini tidak benar.

Jember, 10 Juli 2019

Yang menyatakan,

Andhika Gita Wardhana

NIM 122410101011

SKRIPSI

**PENGEMBANGAN SISTEM REKOMENDASI PEMILIHAN *SPARE PART*
PADA BENGKEL LANGGENG JAYA MOTOR DENGAN METODE
*WEIGHTED PRODUCT (WP)***

Oleh

Andhika Gita Wardhana

122410101011

PEMBIMBING

Dosen Pembimbing Utama : M. Arief Hidayat, S.kom., M.Kom.

Dosen pembimbing Pendamping : Gayatri Dwi Santika, S.SI., S.Kom.

PENGESAHAN PEMBIMBING

Skripsi berjudul "Pengembangan Sistem Rekomendasi Pemilihan *Spare Part* Pada Bengkel Langgeng Jaya Motor Dengan Metode *Weighted Product (WP)*", telah diuji dan disahkan pada,

Hari, tanggal : Kamis, 11 Juli 2019

Tempat : Program Studi Sistem Informasi Universitas Jember

Disetujui oleh:

Pembimbing I,

Pembimbing II,

M. Arief Hidatat, S.kom., M.Kom.

Gayatri Dwi Santika, S.SI., S.Kom.

NIP 198101232010121003

NIP 760017013

PENGESAHAN

Skripsi berjudul ” Pengembangan Sistem Rekomendasi Pemilihan *Spare Part* Pada Bengkel Langgeng Jaya Motor Dengan Metode *Weighted Product* (WP)”, telah diuji dan disahkan pada,

Hari, tanggal : Kamis, 11 Juli 2019

Tempat : Program Studi Sistem Informasi Universitas Jember

Tim Penguji:

Penguji I,

Penguji II,

Anang Andrianto, S.T., M.T.

Yudha Alif Auliya, S.Kom., M.Kom.

NIP. 196906151997021002

NIP. 760018031

Mengesahkan

a.n Dekan

Wakil Dekan I Fakultas Ilmu Komputer

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

NIP. 196909281993021001

RINGKASAN

Sistem Rekomendasi Pemilihan *Spare Part* Pada Bengkel Langgeng jaya Motor Menggunakan Metode *Weighted Product* (WP); Andhika Gita

Wardhana, 122410101011; 2019; 212 Halaman; Program Studi Sistem Informasi Universitas Jember.

Bengkel Langgeng Jaya Motor yang terletak pada jalan Jawa 2 / 1A Kabupaten Jember, melayani penjualan *spare part* motor, servis motor, variasi motor, dan pembelian barang grosir. Setiap hari Bengkel Langgeng Jaya Motor melayani pembelian *spare part* sepeda motor. Banyak orang yang melakukan pembelian *spare part* mengalami kesulitan untuk menentukan merk *spare part* yang cocok untuk sepeda motornya. Sedangkan banyaknya pilihan merk *spare part* untuk dipilih oleh pembeli.

Sistem Rekomendasi Pemilihan *Spare Part* Pada Bengkel Langgeng jaya Motor Menggunakan Metode *Weighted Product* (WP) ini memberi fasilitas kemudahan untuk menentukan *spare part* terbaik untuk digunakan sepeda motor sesuai dengan merk daei sepeda motor tersrbut. Pembeli hanya perlu meberi tahu kriteria mulai dari merk sepeda motor dan tipe sepeda motor dan alternatif merk sepeda motor lainnya kepada karyawan bengkel, penulis membuat sistem pengambilan keputusan *weighted product* dengan bahasa pemrograman *PHP* dengan *framework* LARAVEL dan *MySQL* sebagai *database*. Dengan sistem pendukung diharapkan membantu pembeli *spare part* pada bengkel Langgeng Jaya Motor sesuai dengan kriteria yang diinginkan. Dari hasil penelitian yang dilakukan, aplikasi dapat menjadi sebuah solusi bagi pembeli dan pihak bengkel Langgeng Jaya Motor.

PRAKATA

Puji syukur kehadiran Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul Sistem Informasi Pengendalian Persediaan Stok Bahan Baku Menggunakan Metode *Silver Meal*. 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 bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Prof. Dr Saiful Bukhori, ST., M.Kom., selaku Ketua Program Studi Sistem Informasi Universitas Jember;
2. M. Arief Hidat, S.kom., M.Kom., selaku Dosen Pembimbing Utama dan Gayatri Dwi Santika, S.SI., M.Kom., selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi;
3. Seluruh Bapak dan Ibu dosen beserta staf karyawan di program studi sistem informasi;
4. Seluruh keluarga besar tersayang;
5. Pemilik Bengkel Langgeng Jaya Motor yang meluangkan waktunya dalam membantu penyelesaian skripsi ini;
6. Keluarga besar FORMATION angkatan 2012 yang telah menjadi keluarga selama menempuh pendidikan S1;
7. Teman-teman kontraan yang menyediakan tempat dan listriknya dalam mengerjakan skripsi ini.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna, oleh sebab itu penulis mengharapkan adanya masukan yang bersifat membangun dari semua pihak. Penulis berharap skripsi ini dapat bermanfaat bagi semua pihak.

Jember, 11 Juli 2019

Penulis

DAFTAR ISI

SKRIPSI	i
PERSEMBAHAN	ii
MOTTO.....	iii
PERNYATAAN	iv
PEMBIMBING	v
PENGESAHAN PEMBIMBING.....	vi
PENGESAHAN.....	vii
RINGKASAN	viii
PRAKATA	ix
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xiv
DAFTAR TABEL	xvi
BAB 1. PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	3
1.3. Tujuan	3
1.4. Batasan Masalah	3
1.5. Sistematika Penulisan	3
BAB 2. TINJAUAN PUSTAKA	5
2.1. Penelitian Terdahulu.....	5
2.2. Sistem Informasi	6
2.3. Sistem Penunjang Keputusan (SPK)	6
2.4. <i>Weighted Product (WP)</i>	8
BAB 3. METODELOGI PENELITIAN.....	11
3.1. Jenis Penelitian.....	11
3.2. Tempat Dan Waktu Penelitian	11
3.3. Tahapan Penelitian	11
3.3.1. Analisis Kebutuhan (<i>Requirements Definition</i>)	12
3.3.2. Sistem dan Desain Software (<i>System and Software Design</i>)	13
3.3.3. Implementasi Sistem (<i>Implementation and Unit Testing</i>)	14

3.3.4.	Pengujian Sistem (<i>Integration and System Testing</i>)	15
3.3.5.	Pemeliharaan (<i>Operation and Maintenance</i>)	15
3.4.	Gambaran Sistem.....	15
BAB 4. PENGEMBANGAN SISTEM		17
4.1.	Analisis Kebutuhan.....	17
4.2.	Tahapan Kebutuhan Sistem	19
4.2.1	Kebutuhan Fungsional	19
4.2.2	Kebutuhan Non Fungsional.....	19
4.3.	Desain Sistem.....	20
4.3.1	<i>Business Process</i>	20
4.3.2	<i>Usecase Diagram</i>	21
4.3.3	<i>Scenario</i>	24
4.3.4	<i>Activity Diagram</i>	29
4.3.5	<i>Sequence Diagram</i>	33
4.3.6	<i>Class Diagram</i>	38
4.3.7	<i>Entity Relationship Diagram</i>	39
4.4.	Penulisan Kode Program.....	39
4.5.	Pengujian Sistem.....	43
4.4.1.	<i>White Box Testing</i>	44
4.4.2.	<i>Black Box Testing</i>	46
BAB 5. HASIL DAN PEMBAHASAN		53
5.1.	Hasil Pengembangan Sistem.....	53
5.1.1.	Fitur <i>Login</i>	53
5.1.2.	Halaman Benranda.....	53
5.1.3.	Fitur Mengelola Data Perusahaan	54
5.1.4.	Fitur Mengelola Data <i>Merk</i>	56
5.1.5.	Fitur Mengelola Data <i>Spare Part</i>	58
5.1.6.	Fitur Mengelola Data Grup Barang	60
5.1.7.	Fitur Mengelola Data Kriteria.....	62
5.1.8.	Fitur Mengelola Data Sub Kriteria.....	64
5.1.9.	Fitur Mengelola Data Bobot Kriteria.....	66
5.1.10.	Fitur Mengelola Rekomendasi Pemilihan	67
5.2.	Pembahasan Rekomendasi Pemilihan <i>Spare Part</i>	69

5.2.1	Implementasi Perbaikan Nilai Bobot	69
5.2.2	Nilai Preferensi Relatif Alternatif	70
5.2.3	Nilai Preferensi Relatif Dari Setiap Alternatif.....	70
5.2.4	Perbandingan Hasil Perhitungan Manual dengan Sistem.....	71
BAB 6. PENUTUP		72
6.1.	Kesimpulan.....	72
6.2.	Saran	73
DAFTAR PUSTAKA		74
LAMPIRAN		75
A.	Lampiran <i>Scenario</i>	75
A1.	Lampiran <i>Scenario Login</i>	75
A2.	Lampiran <i>Scenario</i> Mengelola Data Perusahaan	76
A3.	Lampiran <i>Scenario</i> Mengelola Data <i>Merk</i>	81
A4.	Lampiran <i>Scenario</i> Mengelola Data <i>Spare Part</i>	87
A5.	Lampiran <i>Scenaro</i> Mengelola Data Grup Barang.....	91
A6.	Lampiran <i>Scenaro</i> Mengelola Data Kriteria	96
A7.	Lampiran <i>Scenaro</i> Mengelola Data Sub Kriteria	102
A8.	Lampiran <i>Scenaro</i> Mengelola Data Bobot Kriteria	108
A9.	Lampiran <i>Scenario</i> Mengelola Data Rekomendasi Perhitungan (Karyawan) 110	
A10.	Lampiran <i>Scenario Logout</i>	113
B.	Lampiran <i>Activity Diagram</i>	115
B1.	Lampiran <i>Activity Diagram</i> Mengelola Data Perusahaan	115
B2.	Lampiran <i>Activity Diagram</i> Mengelola Data <i>Merk</i>	116
B3.	Lampiran <i>Activity Diagram</i> Mengelola Data <i>Spare Part</i>	117
B4.	Lampiran <i>Activity Diagram</i> Mengelola Data Grup Barang.....	118
B5.	Lampiran <i>Activity Diagram</i> Mengelola Data Kriteria	119
B6.	Lampiran <i>Activity Diagram</i> Mengelola Data Sub Kriteria	120
B7.	Lampiran <i>Activity Diagram</i> Mengelola Data Bobot Kriteria	121
B8.	Lampiran <i>Activity Diagram</i> Mengelola Data Rekomendasi Perhitungan (Karyawan)	122
B9.	Lampiran <i>Activity Diagram Logout</i>	123
C.	Lampiran <i>Sequence Diagram</i>	124

C1.	Lampiran <i>Sequence Diagram</i> Login.....	124
C2.	Lampiran <i>Sequence Diagram</i> Mengelola Data <i>Merk</i>	125
C3.	Lampiran <i>Sequence Diagram</i> Mengelola Data <i>Sapre Part</i>	126
C4.	Lampiran <i>Sequence Diagram</i> Mengelola Data Grup Barang	127
C5.	Lampiran <i>Sequence Diagram</i> Mengelola Data Kriteria.....	128
C6.	Lampiran <i>Sequence Diagram</i> Mengelola Data Sub Kriteria	129
C7.	Lampiran <i>Sequence Diagram</i> Mengelola Data Bobot Kriteria	130
C8.	Lampiran <i>Sequence Diagram</i> Mengelola Rekomendasi Perhitungan (Karyawan)	130
C9.	Lampiran <i>Sequence Diagram</i> Logout	130
D.	Lampiran Kode Program	131
D1.	Lampiran Kode Program Fitur Mengelola Data Perusahaan	131
D2.	Lampiran Kode Program Fitur Mengelola Data <i>Merk</i>	137
D3.	Lampiran Kode Program Fitur Mengelola Data <i>Spare Part</i>	143
D4.	Lampiran Kode Program Fitur Mengelola Data Grup Barang	153
D5.	Lampiran Kode Program Fitur Mengelola Data Kriteria	160
D6.	Lampiran Kode Program Fitur Mengelola Data Sub Kriteria	167
D7.	Lampiran Kode Program Fitur Mengelola Data Bobot Kriteria	175
D8.	Lampiran Kode Program Mengelola Rekomendasi Pemilihan.....	180
E.	Lampiran Black Box	204

DAFTAR GAMBAR

Gambar 2. 1 Karakteristik SPK (Sumber:Turban, E., J. E. Aronson, 2005).....	7
Gambar 2. 2. Model Waterfall (Kersting, 2017)	Error! Bookmark not defined.
Gambar 4. 1 Business Process	21
Gambar 4. 2 Usecase Diagram	22
Gambar 4. 3 Activity Diagram Login	30
Gambar 4. 4 Activity Diagram Mengelola Rekomendasi Pemilihan	32
Gambar 4. 5 Sequence Diagram Lihat dan Tambah Data Perusahaan	34
Gambar 4. 6 Sequence Diagram Mengubah Data Perusahaan	34
Gambar 4. 7 Sequence Diagram Menghapus Data Perusahaan	35
Gambar 4. 8 Sequence Diagram Mengelola Rekomendasi Pemilihan	37
Gambar 4. 9 Class Diagram	38
Gambar 4. 10 Entity Relationship Diagram (ERD)	39
Gambar 4. 11 Diagram Alir Perhitungan	45
Gambar 5. 1 Fitur Login.....	53
Gambar 5. 2 Halaman Beranda	54
Gambar 5. 3 Fitur Mengelola Data Perusahaan	54
Gambar 5. 4 Tambah Data Perusahaan	55
Gambar 5. 5 Ubah Data Perusahaan	55
Gambar 5. 6 Hapus Data Perusahaan	56
Gambar 5. 7 Mengelola Data Merk.....	56
Gambar 5. 8 Tambah Data Merk.....	57
Gambar 5. 9 Ubah Data Merk	57
Gambar 5. 10 Hapus Data Merk	58
Gambar 5. 11 Fitur Mengelola Data Spare Part	58
Gambar 5. 12 Tambah Data Spare Part.....	59
Gambar 5. 13 Hapus Data Spare Part.....	59
Gambar 5. 14 Fitur Mengelola Data Grup Barang.....	60
Gambar 5. 15 Tambah Data Grup Barang	60
Gambar 5. 16 Ubah Data Grup Barang	61
Gambar 5. 17 Hapus Data Grup Barang	61

Gambar 5. 18 Fitur Mengelola Data Kriteria	62
Gambar 5. 19 Tambah Data Kriteria.....	62
Gambar 5. 20 Ubah Data Kriteria	63
Gambar 5. 21 Hapus Data Kriteria.....	63
Gambar 5. 22 Mengelola Data Sub Kriteria.....	64
Gambar 5. 23 Tambah Data Sub Kriteria.....	65
Gambar 5. 24 Ubah Data Sub Kriteria	65
Gambar 5. 25 Hapus Data Sub Kriteria	66
Gambar 5. 26 Fitur Mengelola Data Bobot Kriteria	66
Gambar 5. 27 Ubah Data Bobot Kriteria	67
Gambar 5. 28 Mengelola Data Rekomendasi Perhitungan (Pemilik)	67
Gambar 5. 29 Mengelola Data Rekomendasi Perhitunagn (Karyawan)	68
Gambar 5. 30 Melakukan Rekomendasi Perhitungan.....	68
Gambar 5. 31 Hasil Melakukan Rekomendasi Perhitungan	69

DAFTAR TABEL

Tabel 2. 1 Contoh Kriteria dan Penilaian (Alffita, 2010)	8
Tabel 2. 2 Pembobotan Berdasarkan Tingkat	8
Tabel 4. 1 Kriteria Spare Part.....	17
Tabel 4. 2 Kriteria Perusahaan	17
Tabel 4. 3 Kriteria Merk.....	18
Tabel 4. 4 Kriteria Harga	18
Tabel 4. 5 Kriteria Garansi.....	18
Tabel 4. 6 Kriteria Ketahanan	18
Tabel 4. 7 Deskripsi Aktor	23
Tabel 4. 8 Deskripsi Usecase Diagram	23
Tabel 4. 9 Scenario Mengelola Rekomendasi Pemilihan	26
Tabel 4. 10 Kode Program PenjualanController.php.....	40
Tabel 4. 11 Black Box Testing.....	46
Tabel 5. 1 Perbaikan Nilai Bobot Kriteria	69
Tabel 5. 2 Preferensi Relatif Alternatif	70
Tabel 5. 4 Perbandingan Hasil Manual dan Sistem	71

BAB 1. PENDAHULUAN

Bab ini merupakan langkah awal dari penulisan tugas akhir. Bab ini berisi latar belakang, rumusan masalah, batasan masalah, metodologi penelitian, dan sistematika penulisan.

1.1. Latar Belakang

Bengkel Langgeng Jaya Motor yang terletak pada jalan Jawa 2 / 1A Kabupaten Jember melayani penjualan *spare part* motor, servis motor, variasi motor, dan pembelian barang grosir. Setiap hari Bengkel Langgeng Jaya Motor melayani pembelian *spare part* sepeda motor.

Bengkel Langgeng Jaya Motor mengalami kesulitan dalam melayani pembeli, karena kebanyakan pembeli masih bingung untuk memilih *spare part* yang akan dibeli dan digunakan pada motornya. Pelayan atau pekerja pada Bengkel Langgeng Jaya Motor juga kesulitan dalam hal menjelaskan satu – persatu *spare part* yang ditanyakan pembeli karena banyaknya *spare part* yang sama dalam berbagai merk dan juga harga yang berbeda, contohnya *spare part* yang bermerk *spare part* asli dari jenis sepedanya seperti Honda, Yamaha, Suzuki dan lainnya, ada juga dari merk lain diluar jenis aslinya seperti aspira, indopart, federal dan lainnya.

Konsumen Bengkel Langgeng Jaya Motor juga merasa mengalami kesulitan dalam memilih *spare part* motor yang cocok dengan keinginan, kebutuhan dan kemampuan pembeli karena sekarang banyak macam varian *spare part* motor lengkap dengan keunggulan dan kelebihan. Hal ini tentunya akan mempersulit pembeli dalam menentukan pilihan yang tepat atau sesuai dengan pilihan yang diinginkan untuk kebutuhan motor. Dengan banyaknya produksi *spare part* motor, maka semakin banyak pula merk dan tipe *spare part* motor yang diproduksi dengan kelebihan dan kelemahannya masing-masing. Menentukan *spare part* yang terbaik dan cocok untuk pembeli itu tidak mudah. Karena perlu dibuat sebuah sistem pendukung keputusan yang membantu para pembeli untuk menentukan *spare part* motor yang cocok dengan pilihan pembeli dengan mengetahui rekomendasi

pemilihan *spare part* motor yang sesuai dengan keinginan, kebutuhan dan kemampuan pembeli tanpa harus menunggu bertanya pada pelayan Bengkel Langeng Jaya Motor.

Untuk menangani masalah dalam pelayanan *spare part* motor, penulis mengusulkan sistem pendukung keputusan yang dapat membantu dan menghasilkan keputusan kepada pembeli Bengkel Jaya Motor tentang *spare part* motor yang cocok dengan mengisi data kriteria dan bobot sesuai kebutuhan pemakai. Data bobot diisi oleh pihak Bengkel Langeng Jaya Motor dengan melihat macam – macam *spare part* dan spesifikasi dari *spare part* yang dijual oleh pihak bengkel. Dengan adanya sistem ini konsumen dapat lebih cepat mengetahui *spare part* yang cocok dan sesuai dengan keinginan, kebutuhan dan kemampuan keuangan dari pembeli.

Metode yang dipakai dalam sistem rekomendasi keputusan pemilihan *spare part* motor adalah *Weighted Product* (WP). Penerapan metode *Weighted Product* (WP) dalam sistem pemilihan *spare part* motor ini digunakan untuk memberikan rekomendasi pemilihan *spare part* motor sesuai dengan yang diharapkan oleh pembeli. Konsep ini banyak digunakan untuk menyelesaikan masalah keputusan secara praktis. Konsepnya sederhana dan mudah dipahami, kemampuan untuk mengukur kinerja relatif tepat dari keputusan dalam bentuk matematis yang sederhana. Sistem pendukung keputusan pemilihan *spare part* motor dengan metode *Weighted Product* (WP) dipilih karena mampu memilih alternatif terbaik dari sejumlah alternatif. Dalam hal ini alternatif yang dimaksud adalah *spare part* motor terbaik berdasarkan kriteria-kriteria yang ditentukan dengan langkah - langkah metode *Weighted Product*. Hasil dari proses metode *Weighted Product* (WP) ini dapat mengurutkan alternatif dari nilai yang terbesar ke nilai yang terkecil, sehingga diharapkan *spare part* motor yang direkomendasikan benar-benar sesuai dengan keinginan, kebutuhan, dan kemampuan keuangan pembeli. Berdasarkan permasalahan yang telah diuraikan. Maka penulis mengambil judul “Pengembangan Sistem Rekomendasi Pemilihan *Spare Part* Pada Bengkel Langeng Jaya Motor Menggunakan Metode *Weighted Product* (WP)”.

1.2. Rumusan Masalah

Melihat dan mengkaji latar belakang diatas, maka penulis menyimpulkan pokok permasalahan yang dihadapi adalah :

1. Bagaimana menerapkan metode Weighted Product untuk membangun sistem pemilihan *spare part* motor.
2. Bagaimana membangun aplikasi berbasis web untuk sistem pemilihan *spare part* motor menggunakan metode Weighted Product.

1.3. Tujuan

Adapun tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut:

1. Menghitung rekomendasi pemilihan *spare part* motor menggunakan metode *Weighted Product*.
2. Merancang dan membangun aplikasi berupa sistem pemilihan *spare part* motor menggunakan metode *Weighted Product*.

1.4. Batasan Masalah

Batasan masalah digunakan untuk memilih *spare part* terbaik dalam Pengembangan Sistem Rekomendasi Pemilihan *Spare Part* Pada Bengkel Langgeng Jaya Motor Menggunakan Metode *Weighted Product* (WP). Adapun batasan masalah yang ingin dicapai dalam penelitian ini adalah sebagai berikut:

1. Objek yang diteliti hanya data yang berhubungan dengan *spare part* pada Bengkel Langgeng Jaya Motor.
2. Sistem hanya sebagai alat bantu melakukan perhitungan pemilihan *spare part* terbaik dan keputusan akhir pemilihan tetap pada pembeli.
3. Sistem tidak melakukan proses jual beli *spare part*.

1.5. Sistematika Penulisan

Sistematika penulisan dan urutan skripsi ini disusun sebagai berikut:

1. Pendahuluan

Bab ini menjelaskan tentang latar belakang, perumusan masalah, tujuan dan manfaat, ruang lingkup studi dan sistematika penulisan.

2. Tinjauan Pustaka

Bab ini menjelaskan tentang materi, informasi, tinjauan pustaka, dan studi terdahulu yang menjadi kerangka pemikiran dalam penelitian.

3. Metodologi Penelitian

Bab ini menjelaskan tentang metode penelitian yang digunakan dalam penelitian.

4. Pengembangan Sistem

Bab ini menjelaskan tentang gambaran umum pengembangan sistem, pengujian kinerja, pemeliharaan operasi sistem informasi.

5. Hasil dan Pembahasan

Bab ini menjelaskan tentang hasil dan pembahasan dari penelitian yang dilakukan.

6. Penutup

Bab ini berisi tentang kesimpulan dari penelitian dan saran untuk penelitian selanjutnya.

BAB 2. TINJAUAN PUSTAKA

Pada bagian ini dipaparkan teori-teori dan pustaka yang akan dipakai dalam penelitian. Teori-teori ini berupa teori dari buku literatur dan jurnal. Berikut merupakan teori-teori yang dibahas dalam penelitian.

2.1. Penelitian Terdahulu

Penelitian sejenis yang berhubungan dengan penelitian ini dilakukan oleh Sasika Rani (2014) dengan judul “Sistem Pendukung Keputusan Pemilihan Sepeda Motor Berbasis WEB Dengan Metode *Weighted Product* (WP)” menunjukkan bahwa penerapan metode WP dapat memberikan solusi dalam permasalahan pemilihan sepeda motor dengan menggunakan kriteria-kriteria seperti harga, kapasitas mesin, pemakaian bahan bakar, pilihan warna, dan pilihan model.

Adapun penelitian lain yang dilakukan oleh zulkarnain (2015) dengan judul “Sistem Informasi dan Rekomendasi dan Rekomendasi Pemilihan Homestay Bagi Wisatawan di Kawasan Desa Wisata Menggunakan Metode *Weighted Product* (WP)”. Penelitian ini menjelaskan proses pemilihan *homestay* bagi wisatawan menggunakan metode WP. Inti dari metode ini adalah menentukan nilai bobot untuk setiap kriteria, kemudian melakukan proses perankingan yang akan menyeleksi alternatif yang sudah diberikan.

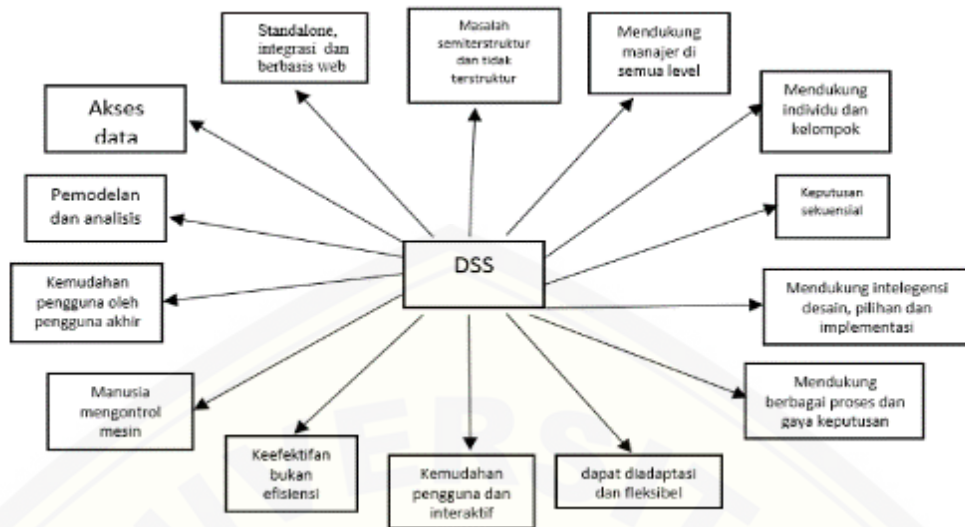
Berdasarkan hasil kedua penelitian terdahulu dapat diadopsi bahwa metode *Weighted Product* (WP) dalam penggunaannya dapat membantu dalam proses rekomendasi pemilihan *spare part* dengan ditambahkan jenis kriteria yang sesuai yaitu *Spare Part*, Perusahaan, *Merk*, Harga, Garansi dan Ketahanan. Selain itu pada penelitian ini, nilai kepentingan bobot preferensi juga harus ditentukan dengan menggunakan rentang bobot 1 - 5. Sehingga dapat menghasilkan rekomendasi yang terbaik bagi pembeli.

2.2. Sistem Informasi

Sistem informasi adalah suatu istilah yang dapat diartikan secara luas dan berbeda-beda. Dilihat dari susunan katanya, sistem informasi terdiri dari dua kata yaitu sistem dan informasi. Sistem adalah sekumpulan elemen-elemen yang saling berinteraksi untuk mencapai tujuan tertentu. Sebuah sistem memiliki maksud tertentu. Ada yang menyebutkan suatu tujuan (*goals*) dan ada yang menyebutkan untuk sasaran (*objectiveness*). Tujuan biasanya berhubungan dengan ruang lingkup yang luas dan sasaran biasanya dengan ruang lingkup yang sempit. Seringkali tujuan dan sasaran digunakan bergantian dan tidak dibedakan. Sedangkan informasi adalah data yang sudah diproses sehingga mempunyai nilai lebih. Dari penjabaran di atas maka dapat disimpulkan sistem informasi adalah jaringan kerja elemen-elemen yang saling berhubungan untuk mengolah data-data yang menghasilkan informasi yang berguna.

2.3. Sistem Penunjang Keputusan (SPK)

Sistem Penunjang Keputusan (SPK) atau disebut juga *Decision Support System* (DSS) merupakan suatu pendekatan dalam pengambilan keputusan dengan menggunakan data, *interface* yang mudah serta menggabungkan pemikiran pengambil keputusan. DSS menggunakan berbagai *model* dan dibangun dengan oleh suatu proses interaktif dan iteratif sehingga mendukung pada fase pengambilan keputusan disertai dengan memasukkan suatu komponen pengetahuan. Gambar 2.1 menjelaskan bahwa DSS memiliki karakteristik dalam menentukan pengambilan keputusan.



Gambar 2. 1 Karakteristik SPK (Sumber:Turban, E., J. E. Aronson, 2005)

Menurut definisi tersebut peneliti dapat menyimpulkan bahwa sistem penunjang keputusan tidak ditekankan untuk membuat keputusan secara mutlak, melainkan melengkapi kemampuan untuk mengolah informasi yang diperlukan untuk membuat keputusan. Dengan kata lain, Sistem Penunjang Keputusan membantu manusia dalam proses membuat keputusan, bukan menggantikan perannya dalam mengambil keputusan.

SPK merupakan implementasi teori-teori pengambilan keputusan yang telah diperkenalkan oleh ilmu-ilmu seperti operation research dan management science, hanya bedanya adalah bahwa jika dahulu untuk mencari penyelesaian masalah yang dihadapi harus dilakukan perhitungan iterasi secara manual (biasanya untuk mencari nilai minimum, maksimum, atau optimum), saat ini computer PC telah menawarkan kemampuannya untuk menyelesaikan persoalan yang sama dalam waktu relatif singkat. Sprague dan Watson mendefinisikan Sistem Pendukung Keputusan (SPK) sebagai sistem yang memiliki lima karakteristik utama yaitu (Ralph H. Sprague, J., & Watson, 1993) dalam (Alfita, 2011).

1. Sistem yang berbasis komputer.
2. Dipergunakan untuk membantu mengambil keputusan.
3. Untuk memecahkan masalah-masalah rumit yang mustahil dilakukan dengan kalkulasi manual.

4. Melalui cara simulasi yang interaktif.
5. Dimana data dan model analisis sebagai komponen utama

2.4. *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 Kriteria dan Penilaian (Alfita, 2010)

No	Kriteria	Nilai	Bobot
1	A_1	W_{01}	W_1
2	A_2	W_{012}	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

Pada tabel 2.1 dijelaskan bahwa kriteria yang digunakan diasumsikan dengan kriteria A_1 sampai A_n . Kemudian rating setiap kriteria diasumsikan dengan W_{01} sampai W_{0n} dan bobot dari masing – masing kriteria diasumsikan dengan W_1 sampai W_n . Menurut Kusumadewa (2006:75) bobot yang diberikan kepada 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 Pembobotan Berdasarkan Tingkat

No	Tingkat Kepentingan	Bobot
1	Sangat Rendah	1
2	Rendah	2
3	Cukup	3

4	Tinggi	4
5	Sangat Tinggi	5

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

- a. Menghitung nilai perbaikan bobot menggunakan persamaan 1

$$W_j = \frac{W_0}{\sum W_0} \quad 1$$

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

$$S_i = \prod_j^n = (X_{ij})^{w_j} \quad 2$$

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 3

$$V_i = \frac{\prod_{j=1}^n (X_{ij})^{w_j}}{\prod_{j=1}^n (X_{j*})^{w_j}} \quad 3$$

Keterangan :

V_i = Nilai preferensi 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. 2 Tahapan Metode *Weighted Product* (WP)

BAB 3. METODELOGI PENELITIAN

Pada bab ini akan dibahas mengenai metodologi penelitian, yaitu tahapan-tahapan yang dilalui oleh peneliti mulai dari analisa kebutuhan data untuk pengembangan sistem.

3.1. Jenis Penelitian

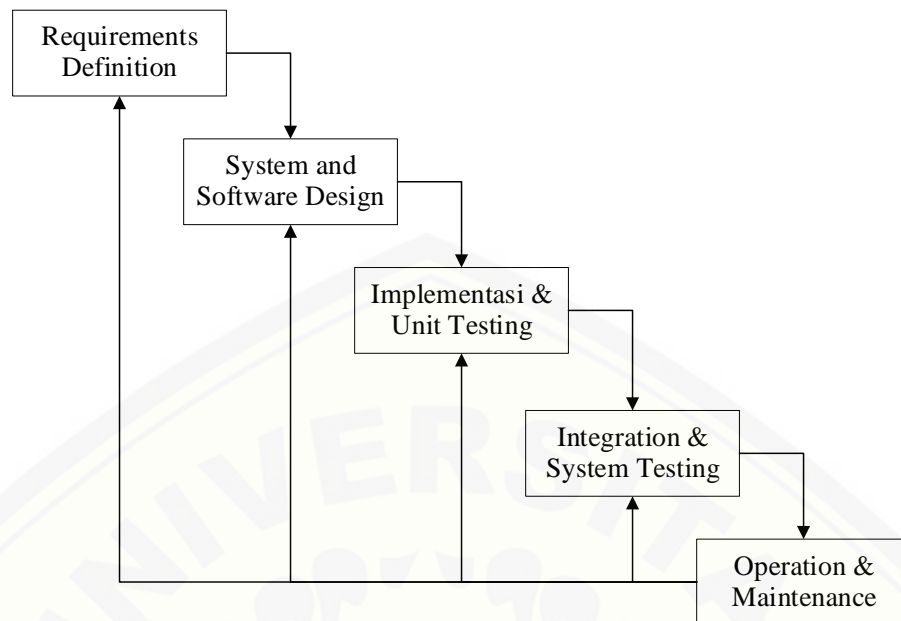
Jenis penelitian yang digunakan dalam membangun sistem rekomendasi pemilihan *spare part* merupakan penelitian pengembangan, karena tujuan penelitian adalah untuk membangun sebuah sistem rekomendasi pemilihan *spare part*. Penelitian pengembangan bertujuan untuk membuat dan mengembangkan suatu produk yang efektif untuk digunakan. Penelitian ini bukan jenis penelitian yang ditujukan untuk menemukan teori atau menguji kebenaran dari suatu teori dalam bentuk eksperimentasi.

3.2. Tempat Dan Waktu Penelitian

Objek penelitian yang dilakukan bertempat di Benkel Langgeng Jaya Motor Jalan Jawa 2 / 1A Kabupaten Jember. Waktu penelitian membutuhkan selama 3 bulan.

3.3. Tahapan Penelitian

Penelitian ini dikembangkan menggunakan metode Software Development Life Cycle (SDLC) dengan model Waterfall. Model waterfall merupakan salah satu model yang sistematis dan sekuensial yang mulai pada tingkatan kemajuan sistem sampai pada analisis, desain, kode, tes dan pemeliharaan ditunjukkan pada gambar 3.2



Gambar 3. 1 Model Waterfall (Kersting, 2017)

3.3.1. Analisis Kebutuhan (*Requirements Definition*)

Analisis kebutuhan merupakan tahap untuk mengumpulkan data dan informasi yang dibutuhkan untuk membangun sistem. Data tersebut dikelompokkan menjadi kebutuhan fungsional dan kebutuhan non fungsional. Untuk memahami sifat program yang akan dibangun, maka harus memahami informasi yang dibutuhkan untuk perangkat lunak, fungsi yang diperlukan, alur, kinerja dan *interface* dari program yang akan dibangun (Pressman, 2001).

1. Tahap Pengumpulan Data

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

a. Studi Literatur

Studi literatur berisi uraian tentang teori, temuan dan bahan penelitian lain yang diperoleh dari bahan acuan untuk dijadikan landasan kegiatan penelitian. Studi literatur berisi ulasan, rangkuman data pemikiran penulisan tentang beberapa sumber pustaka (dapat berupa artikel, buku, slide, informasi

internet dan lain-lain) tentang topik yang dibahas dan biasanya ditempatkan pada bab awal (Hasibuan, 2007).

b. Wawancara

Wawancara merupakan teknik pengumpulan data survei berupa tanya jawab peneliti dengan narasumber. Wawancara tersebut berupa percakapan langsung antar dua pihak atau lebih untuk mendapatkan informasi secara lisan dengan tujuan untuk memperoleh data yang dapat menjelaskan ataupun menjawab suatu permasalahan penelitian (Hasibuan, 2007).

Peneliti melakukan wawancara kepada pemilik bengkel untuk mengetahui permasalahan yang ada dan memperoleh data – data bengkel yang nantinya akan diolah dalam penelitian ini. Data dan informasi yang diperoleh dari tahapan ini antara lain data *spare part*, meliputi dari jenis, kualitas dan harga.

2. Tahap Pengolahan Data

Data yang telah diperoleh pada tahap pengumpulan data diolah untuk menentukan kebutuhan-kebutuhan atau fungsionalitas sistem informasi pengendalian persediaan bahan baku menggunakan metode *Weighted Product (WP)*.

3.3.2. Sistem dan Desain Software (*System and Software Design*)

Tahapan berikutnya adalah desain sistem menggunakan *Unified Modeling Language (UML)* yang dirancang menggunakan konsep *Object-Oriented Programming (OOP)*. Berikut pemodelan UML yang digunakan antara lain:

1. *Business Process*

Business process merupakan diagram yang menggambarkan proses dari sebuah sistem yang meliputi data apa yang diperlukan lalu data diolah untuk menghasilkan output yang diinginkan.

2. *Usecase Diagram*

Menggambarkan fitur-fitur yang tersedia pada aplikasi yang dibangun dan hak akses setiap actor yang terlibat.

3. *Scenario*

Scenario digunakan untuk menjelaskan cara kerja sistem berdasarkan tugas user yang terdapat pada *usecase diagram*. *Scenario* terdiri dari nama *usecase*, aksi aktor dan reaksi sistem.

4. *Sequence Diagram*

Sequence Diagram digunakan untuk menunjukkan interaksi antar objek pada sebuah sistem berupa pesan yang digambarkan terhadap waktu. *Sequence* merupakan blue print bagi programmer.

5. *Activity Diagram*

Activity Diagram digunakan untuk menggambarkan urutan aktivitas dalam sebuah proses. Aktivitas tersebut sesuai dengan *scenario* yang berisi tugas user dan reaksi sistem dan digambarkan dalam bentuk diagram.

6. *Class Diagram*

Class Diagram menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti pewarisan, asosiasi, dependensi dan lain-lain.

7. *Entity Relationship Diagram*

Entity Relationship Diagram menggambarkan struktur database yang akan dibangun pada sistem.

3.3.3. Implementasi Sistem (*Implementation and Unit Testing*)

Tahap implementasi dilakukan berdasarkan desain system yang selanjutnya diubah dalam bentuk program, yaitu :

1. Penulisan program menggunakan bahasa pemrograman *Page Hyper Text Processor (PHP)* dengan framework *Laravel*.
2. *Database Management System (DBMS)* yang digunakan adalah *MySQL* dengan menggunakan jaringan lokal aplikasi *XAMPP*.

3.3.4. Pengujian Sistem (*Integration and System Testing*)

Tahap pengujian sistem bertujuan untuk mengetahui sejauh mana sistem ini dapat berjalan. *Integration Testing* berfungsi untuk mengetahui apakah sistem ini dapat berfungsi dengan baik sesuai dengan yang diharapkan, serta untuk mengetahui letak kekurangan pada sistem. Penelitian ini melakukan pengujian sistem dengan cara sebagai berikut:

1. *White Box Testing*

White box testing merupakan cara pengujian dengan melihat modul yang telah dibuat dan program-program yang ada. Pengujian ini dilakukan oleh pembuat program (*developer*). Jika ada modul yang menghasilkan output yang tidak sesuai, maka baris-baris program, variabel dan parameter yang terlibat pada unit tersebut satu persatu akan dicek dan diperbaiki, kemudian di *compile* ulang (Pressman, 2001).

2. *Black Box Testing*

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

3.3.5. Pemeliharaan (*Operation and Maintenance*)

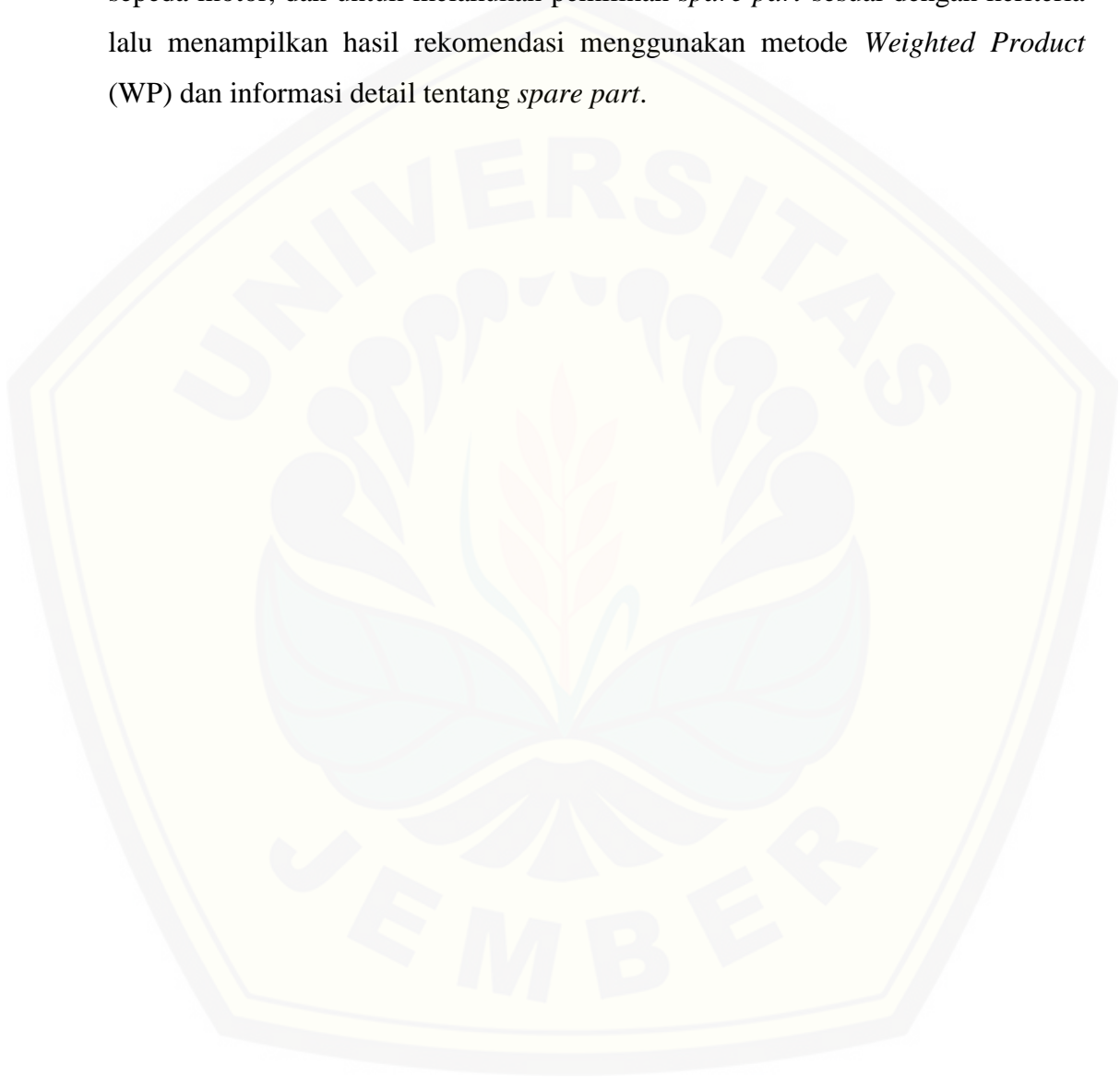
Tahap pemeliharaan dilakukan ketika sistem memiliki kesalahan yang belum terdeteksi sebelumnya, sehingga kesalahan-kesalahan sistem perlu diperbaiki. Pemeliharaan juga dilakukan apabila sistem mengalami perubahan-perubahan karena permintaan baru dari user.

3.4. Gambaran Sistem

Pengembangan aplikasi dari “Sistem Rekomendasi Pemilihan *Spare Part* Pada Bengkel Langgeng Jaya Motor dengan Metode *Weighted Product* (WP). Aplikasi ini merupakan aplikasi berbasis web yang dapat membantu pelanggan

bengkel untuk menentukan spare part motor yang akan dipakaikan pada kendaraannya.

Beberapa fitur utama dari aplikasi ini diantaranya adalah terdapat menu penambahan jenis dan dan merk spare part, penambahan sepeda motor dan jenis sepeda motor, dan untuk melakukan pemilihan *spare part* sesuai dengan kriteria lalu menampilkan hasil rekomendasi menggunakan metode *Weighted Product* (WP) dan informasi detail tentang *spare part*.



BAB 4. PENGEMBANGAN SISTEM

Pengembangan sistem ini dilakukan dengan menggunakan menggunakan menggunakan model *waterfall*. Model ini merupakan metodologi pengembangan perangkat lunak yang mengusulkan pendekatan kepada perangkat lunak sistematis dan sekuensial yang mulai pada tingkat kemajuan sistem pada seluruh analisis, desain, kode, pengujian dan pemeliharaan.

4.1. Analisis Kebutuhan

Proses analisis kebutuhan data yang diperlukan dalam sistem dilakukan dengan wawancara dengan pihak Bengkel Langgeng Jaya Motor. Wawancara difokuskan untuk mendapatkan data berupa kriteria, nilai kriteria, sub-kriteria, nilai sub-kriteria, dan nilai *range* dari tiap *spare part* yang ditetapkan pihak Bengkel Langgeng Jaya Motor. Berdasarkan hasil wawancara dan analisis ditetapkan 6 kriteria, 21 subkriteria, dan nilai *range* tiap subkriteria berbeda-beda sesuai dengan jenis *spare part*. Nilai-nilai tersebut dijelaskan dalam tabel-tabel berikut.

Tabel 4. 1 Kriteria Spare Part

Kriteria	Sub Kriteria	Nilai Sub Kriteria
<i>Spare Part</i> (Nilai = 5)	Sesuai	1
	Tidak Sesuai	0

Tabel 4. 2 Kriteria Perusahaan

Kriteria	Sub Kriteria	Nilai Sub Kriteria
Perusahaan (Nilai = 5)	Sesuai	1
	Tidak Sesuai	0

Tabel 4. 3 Kriteria Merk

Kriteia	Sub Kriteria	Nilai Sub Kriteria
Merk (Nilai = 5)	Sesuai	1
	Tidak Sesuai	0

Tabel 4. 4 Kriteria Harga

Kriteria	Sub Kriteria	Nilai Sub Kriteria
Harga (Nilai = 5)	Sangat Mahal	1
	Mahal	2
	Cukup	3
	Murah	4
	Sangat Murah	5

Tabel 4. 5 Kriteria Garansi

Kriteria	Sub Kriteria	Nilai Sub Kriteria
Garansi (Nilai = 4)	Tidak Baik	1
	Kurang Baik	2
	Cukup	3
	Baik	4
	Sangat Baik	5

Tabel 4. 6 Kriteria Ketahanan

Kriteria	Sub Kriteria	Nilai Sub Kriteria
Ketahanan (Nilai = 3)	Tidak Baik	1
	Kurang Baik	2
	Cukup	3
	Baik	4
	Sangat Baik	5

4.2. Tahapan Kebutuhan Sistem

Data kriteria dan subkriteria diatas selanjutnya digunakan dalam proses analisis kebutuhan sistem, berdasarkan SDLC *waterfall*. Analisis ini dilakukan terhadap objek penelitian untuk mendapatkan kebutuhan fungsional dan non fungsional dari sistem yang akan dibangun. Hasil dari analisis akan mempengaruhi fungsionalitas sistem yang akan dibangun untuk dapat digunakan sesuai dengan fungsi dan kebutuhan *user*.

4.2.1 Kebutuhan Fungsional

Kebutuhan fungsional sistem merupakan kebutuhan pokok yang harus dibangun agar sistem memiliki fitur yang dapat bekerja sesuai dengan apa yang diharapkan oleh pengguna. Kebutuhan fungsional sistem adalah sebagai berikut :

1. Sistem memiliki proses *login* (masuk) kedalam halaman utama sesuai dengan hak akses pengguna.
2. Sistem mampu mengelola kriteria, subkriteria, dan juga pembobotan yang digunakan sebagai bahan penilaian rekomendasi *spare part* yang telah ditetapkan pemilik bengkel.
3. Sistem mampu mengelola penilaian *spare part* untuk dijadikan acuan penentuan *spare part* yang akan dibeli.
4. Sistem mampu membuat penilaian untuk menentukan pemilihan rekomendasi *spare part*.
5. Sistem mampu logout (keluar) dari sistem sesuai dengan hak akses pengguna menuju halaman login awal.

4.2.2 Kebutuhan Non Fungsional

Kebutuhan non fungsional merupakan kebutuhan yang menjadi pendukung agar sistem yang dibangun dapat dengan mudah, aman dan nyaman digunakan oleh pengguna. Kebutuhan non fungsional demi mendukung fungsionalitas sistem adalah sebagai berikut :

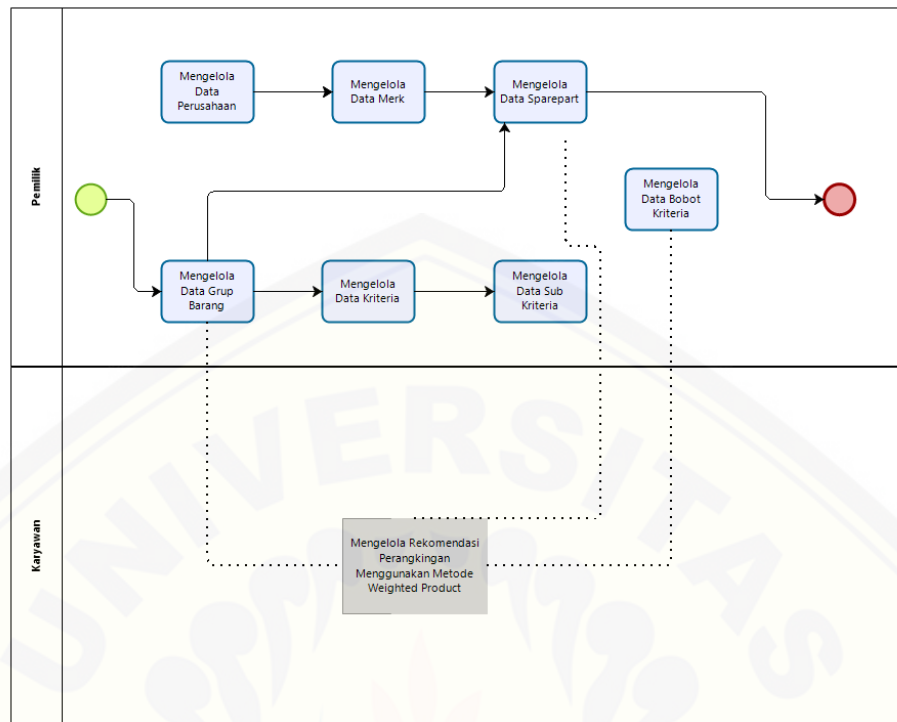
1. Sistem memiliki tampilan yang memudahkan terhadap pengguna (*user friendly*)
2. Sistem memiliki batasan hak akses untuk setiap pengguna yang masuk.
3. Sistem dapat dijalankan dengan *browser* yang berbeda.
4. Sistem mampu digunakan kapan saja dan dimana saja dengan syarat pengguna menggunakan akses internet, terkecuali sistem digunakan hanya pada server lokal.

4.3. Desain Sistem

Tahapan berikutnya adalah desain sistem menggunakan *Unified Modeling Language (UML)* yang dirancang menggunakan konsep *Object-Oriented Programming (OOP)*. Berikut pemodelan UML yang digunakan antara lain:

4.3.1 *Business Process*

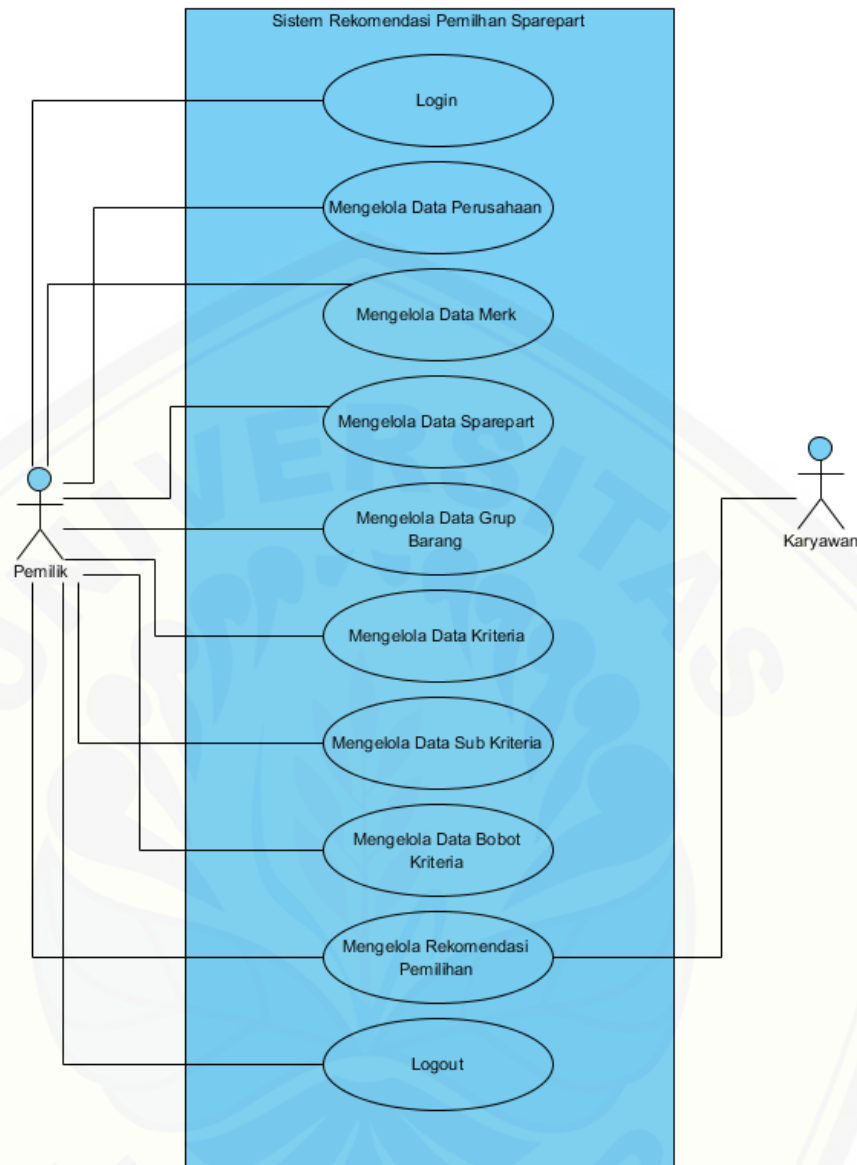
Business process merupakan diagram yang menggambarkan proses dari sebuah sistem yang meliputi data apa yang diperlukan lalu data diolah untuk menghasilkan output yang diinginkan.



Gambar 4. 1 Business Process

4.3.2 Usecase Diagram

Usecase Diagram merupakan kumpulan dari serangkaian kegiatan yang dapat dilakukan oleh sebuah sistem. *Usecase* diagram berisi tentang fitur yang akan dikembangkan dalam sistem penunjang keputusan pembelian padi terbaik, selain hal diagram ini juga berisi hak akses yang diberikan untuk setiap pengguna dalam mengakses setiap fitur yang ada. Gambar *usecase diagram* sistem penunjang keputusan pembelian padi terbaik dapat dilihat pada gambar 4.2 berikut:



Gambar 4. 2 Usecase Diagram

Gambar 4.2 menjelaskan use case diagram dari sistem yang akan dibuat, dimana terdapat dua aktor yang memiliki hak akses tersendiri dalam mengakses fitur yang akan dikembangkan. Fitur yang akan dikembangkan berjumlah sepuluh dua fitur dimana terdapat beberapa fitur yang menggunakan metode *Weighted Product* sebagai *include* agar fitur dapat berjalan sesuai dengan apa yang diharapkan.

Penjelasan definisi tugas aktor terkait dengan *usecase* dapat dilihat pada tabel 4.7, dimana pada tabel tersebut dijelaskan tugas dari aktor atau pengguna

terkait dengan fitur yang bisa dilakukan dalam sistem. Penjelasan lainnya dapat dilihat pada tabel 4.8, dimana pada tabel tersebut menjelaskan fungsionalitas dari setiap *use case* yang telah dibuat.

Tabel 4. 7 Deskripsi Aktor

No	Aktor	Hak Akses
1	Pemilik	Pemilik merupakan aktor yang memiliki hak akses untuk mengelola data perusahaan (menambah, mengubah, menghapus dan melihat), mengelola data merk (menambah, mengubah, menghapus dan melihat), mengelola data <i>spare part</i> (menambah, menghapus dan melihat), mengelola data grup barang (menambah, mengubah, menghapus dan melihat), mengelola data kriteria (menambah, mengubah, menghapus dan melihat), mengelola data bobot kriteria (mengubah, dan melihat), dan mengelola rekomendasi pemilihan.
2	Karyawan	Mengelola rekomendasi pemilihan.

Tabel 4. 8 Deskripsi Usecase Diagram

No	Usecase Diagram	Penjelasan
1	Login	Fitur yang menggambarkan proses login atau autentifikasi untuk masuk ke sistem
2	Mengelola Data Perusahaan	Fitur yang berfungsi untuk melihat data, menambah data, mengubah data, dan menghapus data perusahaan.
3	Mengelola Data Merk	Fitur yang berfungsi untuk melihat data, menambah data, mengubah data, dan menghapus data merk.
4	Mengelola Data <i>Spare part</i>	Fitur yang berfungsi untuk melihat data, menambah data, dan menghapus data <i>spare part</i> .

5	Mengelola Data Grup Barang	Fitur yang berfungsi untuk melihat data, menambah data, mengubah data, dan menghapus data grub barang
6	Mengelola Data Kriteria	Fitur yang berfungsi untuk melihat data, menambah data, mengubah data, dan menghapus data kriteria
7	Mengelola Data Sub Kriteria	Fitur yang berfungsi untuk melihat data, menambah data, mengubah data, dan menghapus data Sub kriteria
8	Mengelola Data Bobot Kriteria	Fitur yang berfungsi untuk melihat data, dan mengubah data bobot kriteria.
9	Mengelola Rekomendasi Pemilihan	Fitur yang berfungsi untuk melakukan rekomendasi <i>sapre part</i> sesuai dengan kebutuhan dari pembeli <i>sapre part</i> .
10	Logout	Fitur yang menggambarkan proses keluar dari system

4.3.3 Scenario

Skenario *Usecase* merupakan penjelasan alur kerja sistem secara rinci meliputi penjelasan prekondisi dan prakondisi juga aksi yang akan dikerjakan sistem ketika pengguna menggunakan sistem. Diagram ini juga menjelaskan *alternatif flow* atau kondisi alternatif ketika pengguna melakukan kesalahan dalam menggunakan sistem, diagram ini dibuat sesuai dengan fitur yang ada pada *use case diagram*.

1. Scenario Login

Skenario login menjelaskan tentang alur reaksi antara aktor atau pengguna dengan sistem dimulai dari pertama kali sistem dijalankan, skenario login dimulai dari mengisi form pada tampilan login hingga masuk pada tampilan dashboard sesuai dengan hak akses pengguna. Skenario utama dan alternatif pada skenario masuk ditunjukkan pada lampiran A (Lampiran *Scenario*).

2. *Scenario* Mengelola Data Perusahaan

Skenario mengelola data perusahaan merupakan diagram alur reaksi antara aktor atau pengguna dengan sistem, skenario mengelola data pengguna menjelaskan alur aksi utama ketika pengguna membuat data perusahaan, sedangkan untuk skenario alternatif berisi alur yang akan dijalankan oleh sistem ketika terdapat suatu data yang tidak diisi dan tidak valid. Skenario utama dan alternatif pada skenario ditunjukkan pada lampiran A (Lampiran *Scenario*).

3. *Scenario* Mengelola Data *Merk*

Skenario mengelola data merk merupakan diagram alur reaksi antara aktor atau pengguna dengan sistem, skenario mengelola data pengguna menjelaskan alur aksi utama ketika pengguna membuat data merk, sedangkan untuk skenario alternatif berisi alur yang akan dijalankan oleh sistem ketika terdapat suatu data yang tidak diisi dan tidak valid. Skenario utama dan alternatif pada skenario ditunjukkan pada lampiran A (Lampiran *Scenario*).

4. *Scenario* Mengelola Data *Spare Part*

Skenario mengelola data *spare part* merupakan diagram alur reaksi antara aktor atau pengguna dengan sistem, skenario mengelola data pengguna menjelaskan alur aksi utama ketika pengguna membuat data *spare part*, sedangkan untuk skenario alternatif berisi alur yang akan dijalankan oleh sistem ketika terdapat suatu data yang tidak diisi dan tidak valid. Skenario utama dan alternatif pada skenario ditunjukkan pada lampiran A (Lampiran *Scenario*).

5. *Scenario* Mengelola Data Grup Barang

Skenario mengelola data grup barang merupakan diagram alur reaksi antara aktor atau pengguna dengan sistem, skenario mengelola data pengguna menjelaskan alur aksi utama ketika pengguna membuat data grup barang, sedangkan untuk skenario alternatif berisi alur yang akan dijalankan oleh sistem ketika terdapat suatu data yang tidak diisi dan tidak valid. Skenario utama dan alternatif pada skenario ditunjukkan pada lampiran A (Lampiran *Scenario*).

6. *Scenario* Mengelola Data Kriteria

Skenario mengelola data kriteria merupakan diagram alur reaksi antara aktor atau pengguna dengan sistem, skenario mengelola data pengguna

menjelaskan alur aksi utama ketika pengguna membuat data kriteria, sedangkan untuk skenario alternatif berisi alur yang akan dijalankan oleh sistem ketika terdapat suatu data yang tidak diisi dan tidak valid. Skenario utama dan alternatif pada skenario ditunjukkan pada lampiran A (Lampiran *Scenario*).

7. *Scenario* Mengelola Data Sub Kriteria

Skenario mengelola data sub kriteria merupakan diagram alur reaksi antara aktor atau pengguna dengan sistem, skenario mengelola data pengguna menjelaskan alur aksi utama ketika pengguna membuat data sub kriteria, sedangkan untuk skenario alternatif berisi alur yang akan dijalankan oleh sistem ketika terdapat suatu data yang tidak diisi dan tidak valid. Skenario utama dan alternatif pada skenario ditunjukkan pada lampiran A (Lampiran *Scenario*).

8. *Scenario* Mengelola Data Bobot Kriteia

Skenario mengelola data bobot kriteria merupakan diagram alur reaksi antara aktor atau pengguna dengan sistem. Pada skenario ini dapat mengubah data dari bobot yang akan digunakan dalam perhitungan rekomendasi pemilihan *spare part*. Skenario utama dan alternatif pada skenario ditunjukkan pada lampiran A.

9. *Scenario* Mengelola Rekomendasi Pemilihan

Skenario mengelola rekomendasi pemilihan merupakan diagram alur reaksi antara aktor atau pengguna dengan sistem, skenario mengelola perhitungan dengan menggunakan metode *Weighted Product* (WP). Skenario utama dan alternatif pada skenario ditunjukkan pada tabel 4.9.

Tabel 4. 9 *Scenario* Mengelola Rekomendasi Pemilihan

Nama	Mengelola Rekomendasi Pemilihan
Aktor	Pemilik
Pre-Condition	Pemilik Memilih Menu Perhitungan
Pra-Condition	Pemilik berhasil melakukan perangkanan rekomendasi <i>spare part</i>
SKENARIO NORMAL	
“Menghitung Rekomendasi <i>Spare Part</i>”	
Aksi Aktor	Reaksi Sistem

1. Pilih menu Perhitungan	<p>2. Menampilkan halaman perhitungan dengan tabel.</p> <ol style="list-style-type: none">a. Keterangan<ol style="list-style-type: none">i. Idii. Perusahaaniii. <i>Merek</i>iv. <i>Spare Part</i>b. Menghitung Nilai Preferensi<ol style="list-style-type: none">i. S1ii. S2iii. S3iv. S4v. S5vi. S6c. Hasil Perkaliand. Totale. Ranging <p>Serta terdapat tombol Lakukan Perhitungan</p>
3. Klik tombol Lakukan Perhitungam	<p>4. Menampilkan modal show form lakukan perhiungan</p> <ol style="list-style-type: none">a. <i>Drobdown</i> Grub Barangb. <i>Drobdown</i> Perusahaan 1c. <i>Drobdown</i> Perusahaan 2d. <i>Drobdown</i> Perusahaan 3e. <i>Tab Conten Merk</i>

	Serta terdapat tombol Hitung dan Close
5. Klik tombol Hitung	
	6. Melakukan proses perhitungan
	7. Menampilkan halaman perhitungan dan tabel berisi perhitungan yang telah dilakukan. <ul style="list-style-type: none"> a. Keterangan <ul style="list-style-type: none"> i. Id ii. Perusahaan iii. <i>Merek</i> iv. <i>Spare Part</i> b. Menghitung Nilai Preferensi <ul style="list-style-type: none"> i. S1 ii. S2 iii. S3 iv. S4 v. S5 vi. S6 c. Hasil Perkalian d. Total e. Ranging Serta terdapat tombol Lakukan Perhitungan

SCENARIO ALTERNATIF

“Batal Menghitung Rekomendasi”

Aksi Aktor	Reaksi Sistem
5a. Klik tombol Close	6a. Menampilkan halaman perhitungan dengan tabel.

- a. Keterangan
 - i. Id
 - ii. Perusahaan
 - iii. Merek
 - iv. Spare Part
- b. Menghitung Nilai Preferensi
 - i. S1
 - ii. S2
 - iii. S3
 - iv. S4
 - v. S5
 - vi. S6
- c. Hasil Perkalian
- d. Total
- e. Ranging

Serta terdapat tombol Lakukan Perhitungan

10. *Scenario Logout*

Skenario logout adalah diagram alur yang menggambarkan bagaimana aksi aktor dan reaksi sistem ketika akan keluar dari sistem. Skenario utama pada skenario ditunjukkan pada lampiran A (Lampiran *Scenario*).

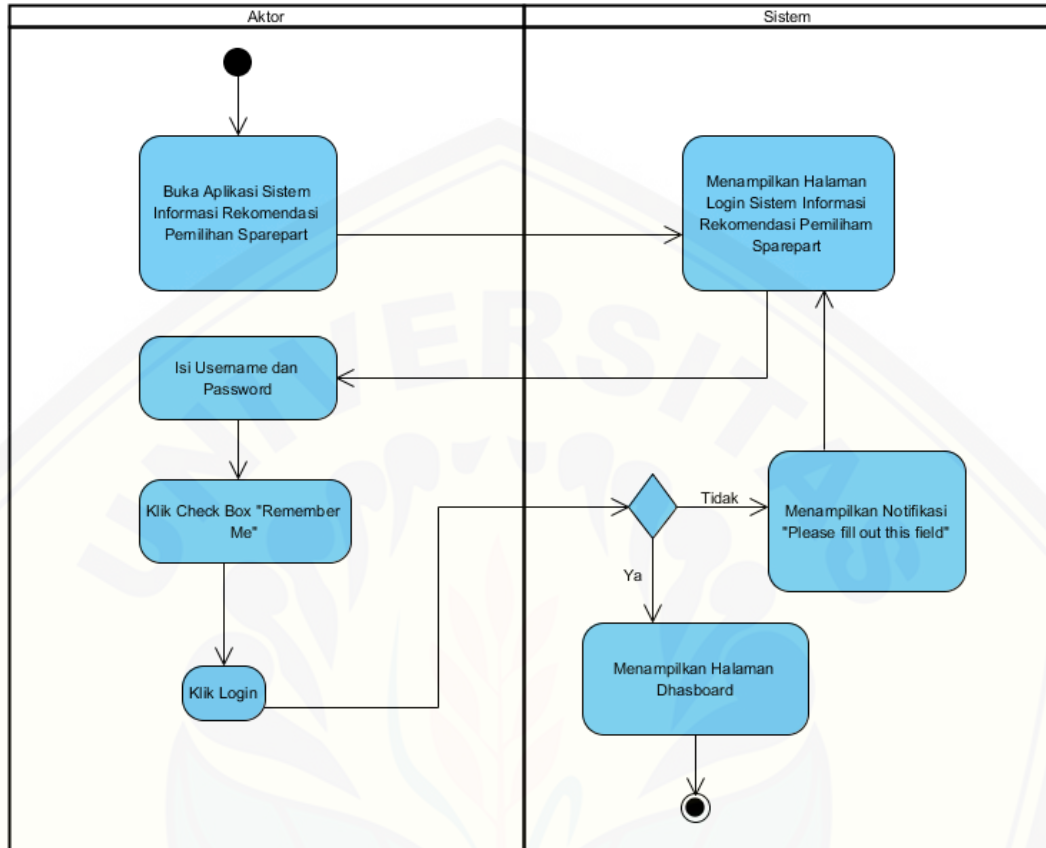
4.3.4 *Activity Diagram*

Activity Diagram digunakan untuk menggambarkan urutan aktivitas dalam sebuah proses. Aktivitas tersebut sesuai dengan *scenario* yang berisi tugas user dan reaksi sistem dan digambarkan dalam bentuk diagram.

1. *Activity Diagram Login*

Activity diagram login menggambarkan alur aktifitas dari proses login agar pengguna dapat mengakses fitur system. Actor yang melakukan login pada system ini adalah pemilik. Actor harus memiliki username dan password yang terdaftar

pada system agar dapat melakan login. *Activity diagram* dapat dilihat pada gamabar 4.7.



Gambar 4. 3 Activity Diagram Login

2. Activity Diagram Mengelola Data Perusahaan

Activity Diagram pada fitur mengelola data perusahaan dapat dilihat pada pembagian partisi berdasarkan aktor dan sistem. Diagram aktifitas dari setiap tombol dan reaksi yang akan dikerjakan sistem dalam mengelola data perusahaan dapat dilihat pada lampiran B (Lampiran *Activity Diagram*).

3. Activity Diagram Mengelola Data Merk

Activity Diagram pada fitur mengelola data merk dapat dilihat pada pembagian partisi berdasarkan aktor dan sistem. Diagram aktifitas dari setiap tombol dan reaksi yang akan dikerjakan sistem dalam mengelola data perusahaan dapat dilihat pada lampiran B (Lampiran *Activity Diagram*).

4. Activity Diagram Mengelola Data Spare Part

Activity Diagram pada fitur mengelola data *spare part* dapat dilihat pada pembagian partisi berdasarkan aktor dan sistem. Diagram aktifitas dari setiap tombol dan reaksi yang akan dikerjakan sistem dalam mengelola data perusahaan dapat dilihat pada lampiran B (Lampiran *Activity Diagram*).

5. *Activity Diagram* Mengelola Data Grup Barang

Activity Diagram pada fitur mengelola data grup barang dapat dilihat pada pembagian partisi berdasarkan aktor dan sistem. Diagram aktifitas dari setiap tombol dan reaksi yang akan dikerjakan sistem dalam mengelola data perusahaan dapat dilihat pada lampiran B (Lampiran *Activity Diagram*).

6. *Activity Diagram* Mengelola Data Kriteria

Activity Diagram pada fitur mengelola data kriteria dapat dilihat pada pembagian partisi berdasarkan aktor dan sistem. Diagram aktifitas dari setiap tombol dan reaksi yang akan dikerjakan sistem dalam mengelola data perusahaan dapat dilihat pada lampiran B (Lampiran *Activity Diagram*).

7. *Activity Diagram* Mengelola Data Sub Kriteria

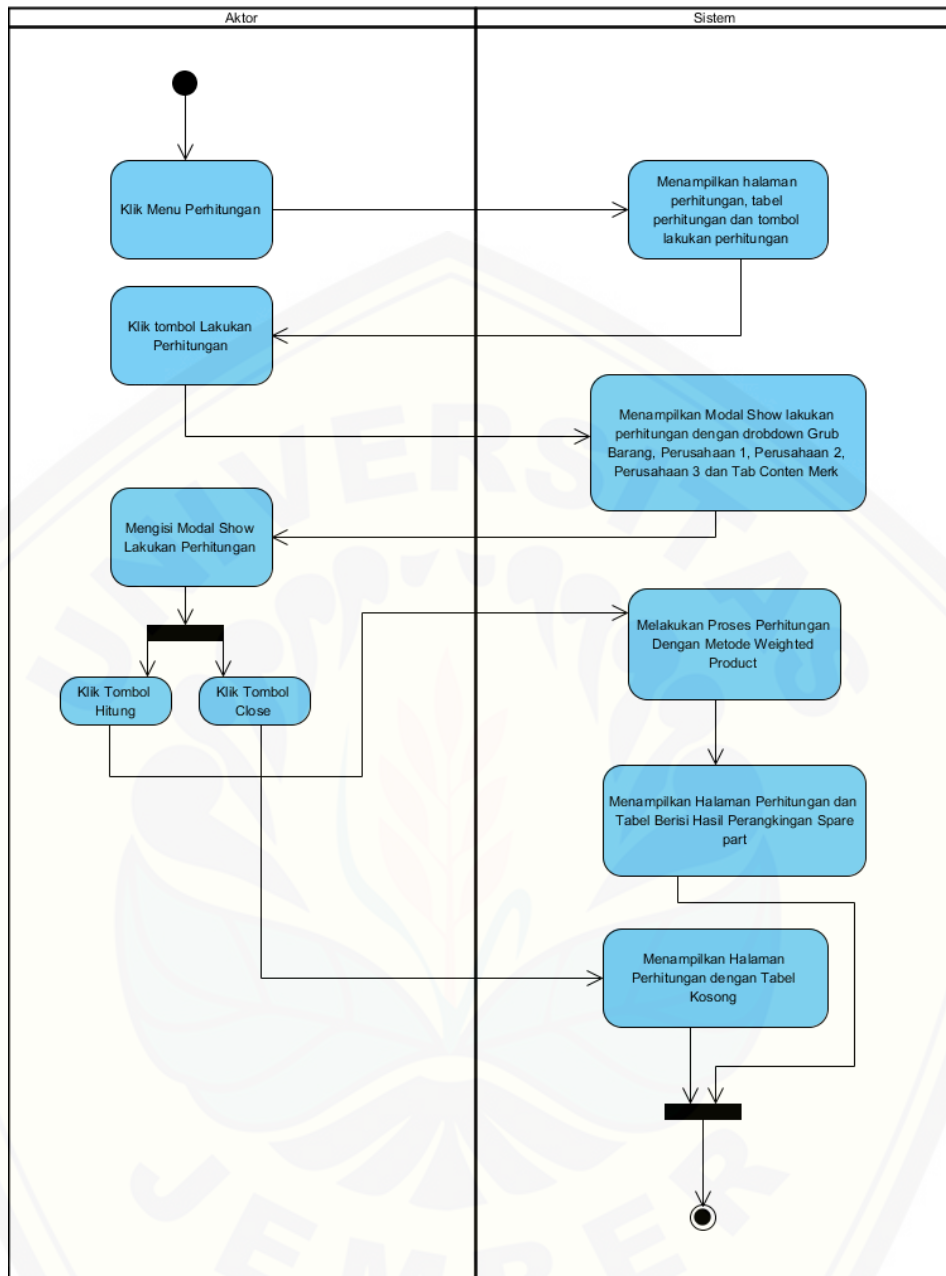
Activity Diagram pada fitur mengelola data sub kriteria dapat dilihat pada pembagian partisi berdasarkan aktor dan sistem. Diagram aktifitas dari setiap tombol dan reaksi yang akan dikerjakan sistem dalam mengelola data perusahaan dapat dilihat pada lampiran B (Lampiran *Activity Diagram*).

8. *Activity Diagram* Mengelola Data Bobot Kriteria

Activity Diagram pada fitur mengelola data bobot kriteria dapat dilihat pada pembagian partisi berdasarkan aktor dan sistem. Diagram aktifitas dari setiap tombol dan reaksi yang akan dikerjakan sistem dalam mengelola data perusahaan dapat dilihat pada lampiran B (Lampiran *Activity Diagram*).

9. *Activity Diagram* Mengelola Rekomendasi Perhitungan

Activity diagram mengelola rekomendasi perhitungan menggambarkan alur aktivitas proses perangkaan *spare part* yang dilakukan oleh aktor pemilik dan aktor karyawan, dan reaksi sistem ketika aktor-aktor tersebut memberikan aksi terhadap sistem. *Activity diagram* mengelola rekomendasi perhitungan terdapat pada gambar 4.8 berikut.



Gambar 4. 4 Activity Diagram Mengelola Rekomendasi Pemilihan

10. Activity Diagram Logout

Diagram aktifitas pada fitur *logout* dapat dilihat pada pembagian partisi berdasarkan aktor dan sistem. Diagram aktifitas dari setiap tombol dan reaksi yang akan dikerjakan sistem dalam *logout* dapat dilihat pada lampiran B (Lampiran Activity Diagram).

4.3.5 *Sequence Diagram*

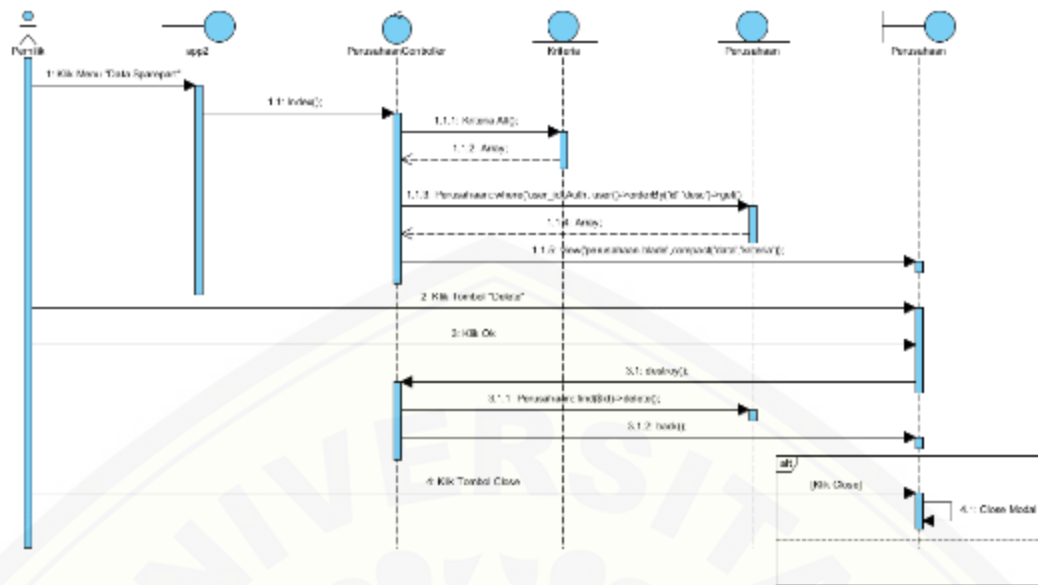
Sequence Diagram digunakan untuk menunjukkan interaksi antar objek pada sebuah sistem berupa pesan yang digambarkan terhadap waktu. *Sequence* merupakan *blue print* bagi *programmer*.

1. *Sequence Diagram Login*

Sequence diagram login merupakan langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses masuk. Actor yang dapat melakukan login adalah admin. Pada *sequence* ini terdapat class view login.blade.php, app2.php class controller LoginController.php, class model User.php. Penggambaran *sequence diagram* masuk digunakan untuk menjelaskan fungsi atau method yang akan dibuat seperti yang dapat dilihat pada lampiran C (Lampiran *Sequence Diagram*).

2. *Sequence Diagram Mengelola Data Perusahaan*

Sequence diagram ini menggambarkan scenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian atau event untuk mengelola data perusahaan yaitu menambah data perusahaan, mengubah data perusahaan dan menghapus data perusahaan. Actor yang dapat mengelola data perusahaan adalah admin. Pada *sequence* ini terdapat class view Perusahaan.blade.php, app2.blade.php, class controller PerusahaanController.php, class model Perusahaan.php serta method-method yang dibutuhkan dalam proses ini. *Sequence diagram* mengelola data perusahaan akan dijelaskan pada gambar 4.3 sampai 4.6 berikut.



Gambar 4. 7 Sequence Diagram Menghapus Data Perusahaan

3. Sequence Diagram Mengelola Data Merk

Sequence diagram ini menggambarkan scenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian atau event untuk mengelola data *merk* yaitu menambah data *merk*, mengubah data *merk* dan menghapus data *merk*. Actor yang dapat mengelola data *merk* adalah admin. Pada sequence ini terdapat class view merk.blade.php, perusahaan.blade.php, class controller MerkController.php, class model Merk.php serta method-method yang dibutuhkan dalam proses ini. *Sequence diagram* mengelola data *merk* terdapat pada lampiran C (Lampiran *Sequence Diagram*).

4. Sequence Diagram Mengelola Data Spare Part

Sequence diagram ini menggambarkan scenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian atau event untuk mengelola data *spare part* yaitu menambah data *spare part*, dan menghapus data *spare part*. Actor yang dapat mengelola data *spare part* adalah admin. Pada sequence ini terdapat class view sparepart.blade.php, merk.blade.php, class controller SparepartController.php, class model Sparepart.php serta method-method yang dibutuhkan dalam proses ini. *Sequence diagram* mengelola data *spare part* terdapat pada lampiran C (Lampiran *Sequence Diagram*).

5. *Sequence Diagram* Mengelola Data Grup Barang

Sequence diagram ini menggambarkan scenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian atau event untuk mengelola data grup barang yaitu menambah data grup barang, mengubah data grup barang dan menghapus data grup barang. Actor yang dapat mengelola data grup barang adalah admin. Pada sequence ini terdapat class view grupbarang.blade.php, app2.blade.php, class controller GrupBarangController.php, class model GrupBarang.php serta method-method yang dibutuhkan dalam proses ini. *Sequence diagram* mengelola data grup barang terdapat pada lampiran C (Lampiran *Sequence Diagram*).

6. *Sequence Diagram* Mengelola Data Kriteria

Sequence diagram ini menggambarkan scenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian atau event untuk mengelola data kriteria yaitu menambah data kriteria, mengubah data kriteria dan menghapus data kriteria. Actor yang dapat mengelola data kriteria adalah admin. Pada sequence ini terdapat class view kriteria.blade.php, grupbarang.blade.php, class controller KriteriaController.php, class model Kriteria.php serta method-method yang dibutuhkan dalam proses ini. *Sequence diagram* mengelola data kriteria terdapat pada lampiran C (Lampiran *Sequence Diagram*).

7. *Sequence Diagram* Mengelola Data Sub Kriteria

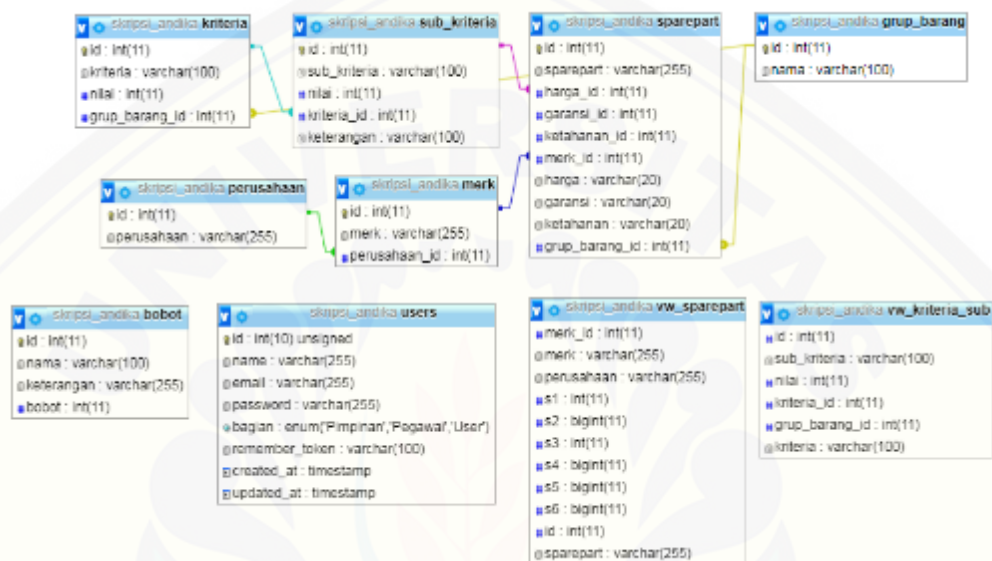
Sequence diagram ini menggambarkan scenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian atau event untuk mengelola data kriteria yaitu menambah data kriteria, mengubah data kriteria dan menghapus data kriteria. Actor yang dapat mengelola data kriteria adalah admin. Pada sequence ini terdapat class view subkriteria.blade.php, kriteria.blade.php, class controller SubKriteriaController.php, class model SubKriteria.php serta method-method yang dibutuhkan dalam proses ini. *Sequence diagram* mengelola data kriteria terdapat pada lampiran C (Lampiran *Sequence Diagram*).

8. *Sequence Diagram* Mengelola Data Bobot Kriteria

Sequence diagram ini menggambarkan scenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian atau event untuk

4.3.7 Entity Relationship Diagram

Entity Relationship Diagram menggambarkan struktur database yang akan dibangun pada sistem. Diagram ini menggambarkan desain database yang akan digunakan dalam sistem. ERD pendeteksi dan penentuan tingkat plagiarismedapat dilihat pada Gambar 4.10.



Gambar 4. 10 Entity Relationship Diagram (ERD)

4.4. Penulisan Kode Program

Tahap ini merupakan proses pembuatan aplikasi, yaitu merupakan proses implementasi dari hasil analisa kebutuhan yang dilakukan pada tahap perencanaan, analisa dan desain sistem. Penulisan kode program dilakukan menggunakan bahasa pemrograman PHP dengan *framework* Laravel.

Class PenjualanController.php merupakan *controller class*. *Class* ini berada pada direktori *app/http/controllers* yang telah disediakan sesuai dengan *framework Laravel*. Penulisan kode-kode pemrograman dalam kelas ini menggunakan beberapa *class* lain yang telah tersedia dalam *framework Laravel*. Penulisan kode pemrograman ditunjukkan pada tabel 4.11.

Tabel 4. 10 Kode Program PenjualanController.php

Kode Program
<pre><?php namespace App\Http\Controllers; use Illuminate\Http\Request; use App\Bobot; use App\GrupBarang; use App\Perusahaan; use DB; use Auth; class PenjualanController extends Controller { public function index(Request \$request) { \$bobot = Bobot::all(); \$kriteria = Bobot::all(); \$perusahaan = Perusahaan::all(); \$grupBarang = GrupBarang::all(); \$dt = \$request->toArray(); \$d = \$request; \$jbobot = 0 ; \$arr = array(); if (isset(\$dt['s1'])) { \$db = DB::table('vw_sparepart')->get(); \$jml = 0; \$i = 3; \$l = 0; foreach(\$d->merk as \$ke => \$r){</pre>

```
        foreach ($db as $k => $v) {
            $j = (3-($ke%3));
            $jbobot = $bobot[0]-
                >bobot+$i+$j+$bobot[1]-
                >bobot+$bobot[2]-
                >bobot+$bobot[3]->bobot;
            $arr[$l]['sp'] = $v->sparepart;
            $arr[$l]['id'] = $v->id;
            $arr[$l]['per'] = $v->perusahaan;
            $arr[$l]['merk'] = $v->merk;
            $arr[$l]['s1'] = pow((( $v->s1 ==
                $d->grup_barang)?1:0), ($bobot[0]-
                >bobot/$jbobot));
            $arr[$l]['s2'] = pow((( $v->s2 ==
                $d->s2)?3:1), ($i/$jbobot));
            $arr[$l]['s3'] = pow((( $v->s3 ==
                $r)?3:1), ($j/$jbobot));
            $arr[$l]['s4'] = round(pow(( $v-
                >s4), ($bobot[1]-
                >bobot/$jbobot)), 3);
            $arr[$l]['s5'] = round(pow(( $v-
                >s5), ($bobot[2]-
                >bobot/$jbobot)), 3);
            $arr[$l]['s6'] = round(pow(( $v-
                >s6), ($bobot[3]-
                >bobot/$jbobot)), 3);
            $arr[$l]['hPer'] =
                $arr[$l]['s1']*$arr[$l]['s2']*$arr[$l]['s3']*$ar
                r[$l]['s4']*$arr[$l]['s5']*$arr[$l]['s6'];
            $jml += $arr[$l]['hPer'];
            $l++;
        }
        if($ke%3 == 0){
            $i--;
        }
    }
}
```

```
        foreach ($arr as $k => $value) {
            $arr[$k]['tot'] = $arr[$k]['hPer']/$jml;
        }
    }

    $kk = array();
    foreach ($arr as $key => $row) {
        $kk[$key] = $row['tot'];
    }

    array_multisort($kk, SORT_DESC, $arr);
    $data = $arr;
    if(Auth::guest()){
        return
        view("admin.Penjualan2",compact('data','kriteria','grupBarang','p
        erusahaan'));
    }
    return
    view('admin.Penjualan',compact('data','kriteria','grupBarang','p
    erusahaan'));
}

public function merks($id){
    $data = Perusahaan::find($id)->merks;
    echo json_encode($data);
}

public function hargas($id){
    $data = DB::table('vw_kriteria_sub')->
    >where('grup_barang_id',$id)->get();
    echo json_encode($data);
}
}
```

Class ini berfungsi mengatur perbandingan menggunakan metode *Weighted Product* (WP). Transaksi tersebut diarahkan kedalam *function* yang berbeda. *Function-function* yang ada dalam *class* ini dijelaskan sebagai berikut:

1. *Function index(Request \$request)* berfungsi sebagai *function* yang pertama kali dijalankan ketika *class* ini dipanggil dan juga mengatur perhitungan untuk melakukan perbandingan *spare part*. *Function index()* mengambil data yang *spare part* dan data terkait lainnya yang dibutuhkan kemudian ditampilkan dalam *view class*.
2. *Function merks(\$id)* berfungsi untuk mengambil data dari tabel *merk* yang dibutuhkan.
3. *Function hargas(\$id)* berfungsi untuk mengambil data dari *database* grup barang untuk tabel kriteria.

Penulisan kode program untuk fitur – fitur lainnya pada Sistem Rekomendasi *Spare Part* Pada Bengkel Langgeng Jaya Motor Dengan Metode *Weighted Product* (WP) terdapat pada Lampiran D (Kode Program). Fitur – fitur tersebut meliputi fitur mengelola data perusahaan, mengelola data *merk*, mengelola data *spare part*, mengelola data grup barang, mengelola data kriteria, mengelola data sub kriteria, mengelola data bobot kriteria, Login dan Logout.

4.5. Pengujian Sistem

Pengujian dan evaluasi dimaksudkan untuk mengetahui sejauh mana sistem yang dibuat ini dapat berfungsi sesuai dengan proses transaksi dalam optimasi distribusi ini nanti sesuai dengan yang diharapkan oleh pada pengguna. Pada tahap ini dilakukan uji coba terhadap sistem optimasi untuk menemukan kesalahan-kesalahan yang mungkin terjadi serta melakukan perbaikan untuk lebih menyempurnakan kinerja *website* tersebut. Pengujian dilakukan dengan 2 metode, yaitu *White Box* dan *Black Box*. Pengujian menggunakan pengujian jalur dasar (*bash path testing*) yang didalamnya terdapat beberapa tahapan pengujian antara

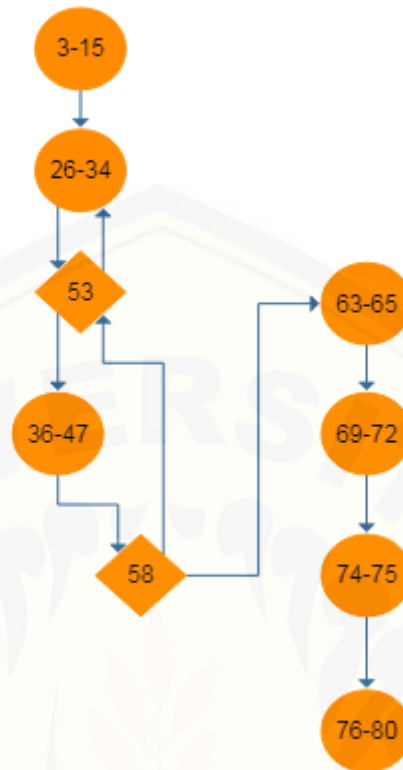
lain pembuatan diagram alir, penentuan jalur independen, penghitungan kompleksitas siklomatik, dan *test case*.

4.4.1. *White Box Testing*

Pengujian menggunakan pengujian jalur dasar (*bash path testing*) yang didalamnya terdapat beberapa tahapan pengujian antara lain pembuatan diagram alir, penentuan jalur independen, penghitungan kompleksitas siklomatik, dan *test case*.

1. Diagram Alir

Diagram alir merupakan notasi yang digunakan untuk merepresentasikan aliran control yang digambarkan dari hasil penomoran dari *listing* program. Diagram alir digambarkan dengan *node* (simpul) yang dihubungkan dengan *edge* (garis). Diagram alir fitur prediksi pada Sistem Rekomendasi *Spare Part* Pada Bengkel Langgeng Jaya Motor Dengan Metode *Weighted Product* (WP) dapat dilihat pada gambar 4.12 berikut.



Gambar 4. 11 Diagram Alir Perhitungan

2. Kompleksitas Siklomatik (*Cyclomatic Complexity*)

Kompleksitas siklomatik merupakan metrik perangkat lunak yang menyediakan ukuran kuantitatif dari kompleksitas logis suatu program. Kompleksitas siklomatik mendefinisikan jumlah jalur independen dalam suatu program. Perhitungan kompleksitas siklomatik menggunakan rumus berikut.

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

Keterangan:

$V(G)$ = Kompleksitas siklomatik

E = Jumlah *edge* (garis)

N = Jumlah *node* (simpul)

Penjelasan perhitungan kompleksitas siklomatik fitur prediksi pada Sistem Rekomendasi *Spare Part* Pada Bengkel Langgeng Jaya Motor Dengan Metode *Weighted Product* (WP) sebagai berikut.

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

$$= 10 - 9 + 2$$

$$= 3$$

3. Jalur Independen (*Independent Path*)

Jalur independen merupakan setiap jalur yang melalui program yang memperkenalkan setidaknya satu kondisi baru. Bila dinyatakan dalam grafik alir, jalur independen harus bergerak setidaknya sepanjang satu edge yang belum dilintasi sebelum jalur tersebut didefinisi. Jalur independen fitur prediksi pada Sistem Rekomendasi *Spare Part* Pada Bengkel Langgeng Jaya Motor Dengan Metode *Weighted Product* (WP) adalah sebagai berikut.

Jalur 1 : (3-15) – (26-35) – 53 – (36-47) – 58 – (63-65) – (69-72) – (74-75) – (76-80)

Jalur 2 : (3-15) – (26-35) – 53 – (26-35) – 53 – (36-47) – 58 – (63-65) – (69-72) – (74-75) – (76-80)

Jalur 3 : (3-15) – (26-35) – 53 – (26-35) – 53 – (36-47) – 58 – 53 – (36-47) – 58 – (63-65) – (69-72) – (74-75) – (76-80)

4.4.2. *Black Box Testing*

Black box testing merupakan metode pengujian perangkat lunak dari segi fungsionalitas *software* yang berkaitan dengan struktur internal atau kerja dengan tujuan untuk mengetahui apakah fungsi – fungsi, data input dan data output pada sistem sesuai dengan yang dibutuhkan oleh pengguna atau tidak. Hasil pengujian metode *black box* Sistem Rekomendasi *Spare Part* Pada Bengkel Langgeng Jaya Motor Dengan Metode *Weighted Product* (WP) dapat dilihat pada tabel 4.

Tabel 4. 11 *Black Box Testing*

No	Menu	Fungsi	Aksi	Hasil	Ket
1	<i>Login</i>	Digunakan pemilik untuk masuk sistem	Mengisi Username dan Password pada form login kemudian	<i>Login</i> berhasil dan menampilkan beranda	√

			menekan tombol login		
			Username dan Passwor tidak diisi	Menampilkan notifikasi "Please fill out this field"	√
2	Mengelola Data Perusahaan	Digunakan untuk lihat tabel perusahaan, tambah data perusahaan, mengubah data perusahaan, dan menghapus data perusahaan	Memilih tombol tambah perusahaan	Menampilkan <i>modal show</i> form tambah perusahaan	√
			Mengisi <i>modal show</i> form perusahaan dan klik tombol Simpan	Menyimpan data ke <i>database</i> dan menampilkan halaman perusahaan	√
			Form <i>modal show</i> perusahaan tidak terisi dan klik tombol Simpan	Menampilkan notifikasi "Please fill out this field"	√
			Klik tombol Cancel	Menampilkan halaman perusahaan	√
			Memilih tombol Update	Menampilkan <i>modal show</i> form <i>Upadate</i>	√
			Mengubah <i>modal show</i>	Menyimpan perubahandata	√

			form perusahaan dan klik tombol Simpan	ke <i>database</i> dan menampilkan halaman perusahaan	
			Form <i>modal show Update</i> tidak terisi dan klik tombol Simpan	Menampilkan notifikasi "Please fill out this field"	√
			Klik tombol Cancel	Menampilkan halaman perusahaan	√
			Memilih tombol Delete	Menampilkan peringatan "Apakah Anda Yakin Untuk Menghapus", tombol Ok dan Tombol Cancel	√
			Klik Tombol Ok	Menghapus data perusahaan terpilih dan menampilkan halaman perusahaan	√

			Klik Tombol Cancel	Menampilkan halaman perusahaan	√
3.	Mengelola Data Merk	Digunakan untuk lihat tabel merk, tambah data merk, mengubah data merk, dan menghapus data merk	Memilih tombol tambah merk	Menampilkan modal show form tambah merk	√
			Mengisi modal show form merk dan klik tombol Simpan	Menyimpan data ke database dan menampilkan halaman merk	√
			Form modal show merk tidak terisi dan klik tombol Simpan	Menampilkan notifikasi "Please fill out this field"	√
			Klik tombol Cancel	Menampilkan halaman merk	√
			Memilih tombol Update	Menampilkan modal show form Update data merk	√
			Mengubah modal show form merk dan klik tombol Simpan	Menyimpan perubahandata ke database dan menampilkan halaman merk	√

			Modal show form Update tidak terisi dan klik tombol Simpan	Menampilkan notifikasi "Please fill out this field"	√
			Klik tombol Cancel	Menampilkan Halaman merk	√
			Memilih tombol Delete	Menampilkan peringatan "Apakah Anda Yakin Untuk Menghapus", tombol Ok dan Tombol Cancel	√
			Klik Tombol Ok	Menghapus data perusahaan terpilih dan menampilkan halaman merk	√
			Klik Tombol Cancel	Menampilkan halaman merk	√
4.	Mengelola Data Spare Part	Digunakan untuk lihat tabel spare part, tambah data spare part, dan	Memilih tombol tambah spare part	Menampilkan modal show form tambah spare part	√
			Mengisi modal show form	Menyimpan data ke	√

		menghapus data <i>spare part</i>	<i>spare part</i> dan klik tombol Simpan	<i>database</i> dan menampilkan halaman <i>spare part</i>	
			Form <i>modal show form spare part</i> tidak terisi dan klik tombol Simpan	Menampilkan notifikasi "Please fill out this field"	√
			Klik tombol Cancel	Menampilkan halaman <i>spare part</i>	√
			Memilih tombol Delete	Menampilkan peringatan "Apakah Anda Yakin Untuk Menghapus", tombol Ok dan Tombol Cancel	√
			Klik Tombol Ok	Menghapus data perusahaan terpilih dan menampilkan halaman <i>spare part</i>	√

			Klik Tombol Cancel	Menampilkan halaman <i>spare</i> <i>part</i>	√
--	--	--	-----------------------	--	---



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

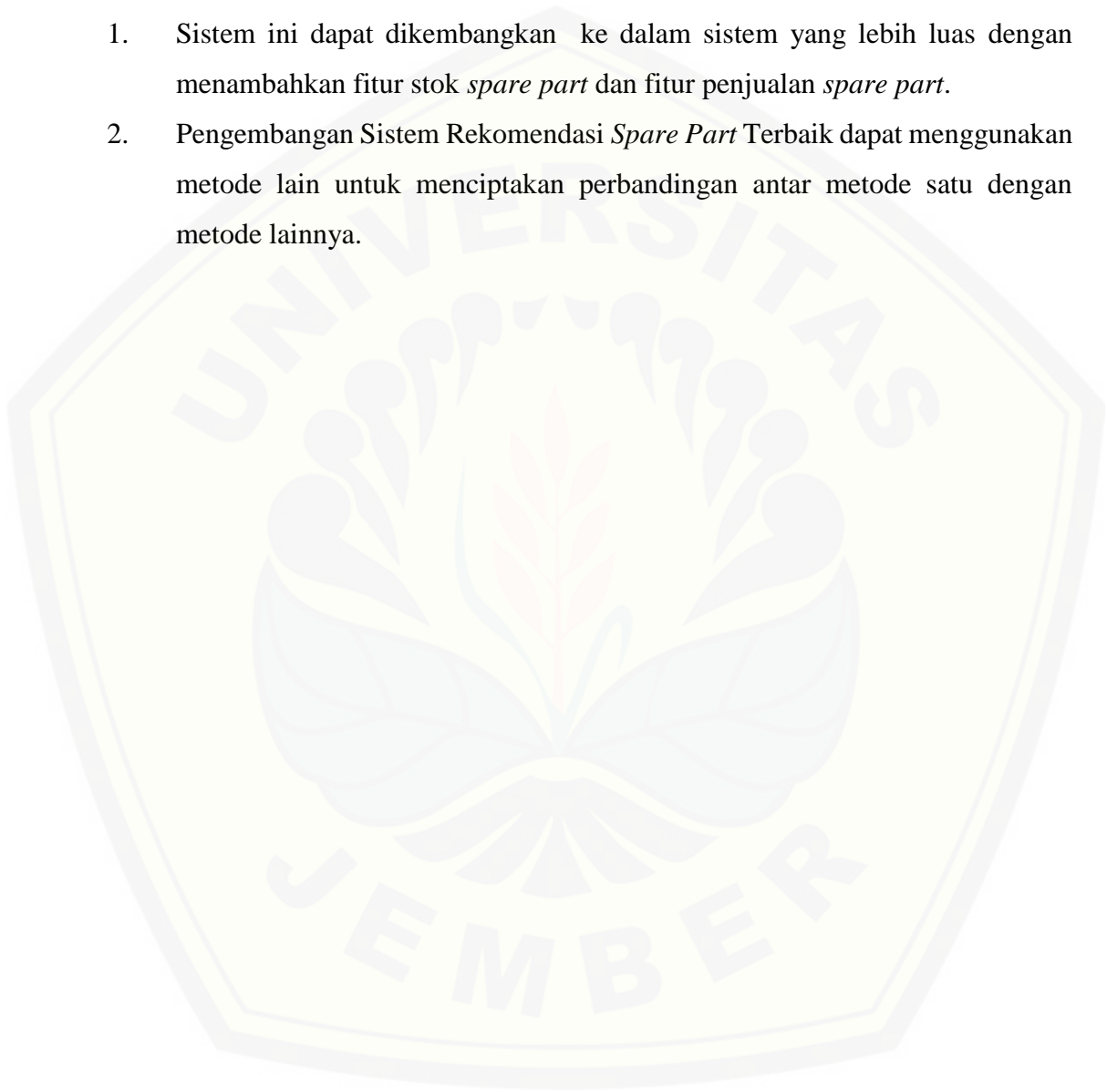
Kesimpulan dari hasil penelitian yang telah dilakukan oleh peneliti adalah sebagai berikut :

1. Penerapan rekomendasi *spare part* menggunakan metode *Weighted Product* (WP) dilakukan dengan menghitung normalisasi bobot kriteria untuk mendapatkan perbaikan nilai bobot setiap kriteria. Perkalian pangkat bobot sub kriteria dengan normalisasi bobot kriteria dan menghitung hasil akhir. Perhitungan akhir rekomendasi harus bernilai lebih dari 0, jika hasil 0 maka bukan termasuk rekomendasi dari *spare part* terbaik. Hasil rekomendasi *spare part* terbaik menghasilkan perbandingan yang berbeda, karena setiap pembeli memiliki nilai bobot kepentingan sesuai dengan kebutuhan.
2. Pengembangan Sistem Rekomendasi *Spare Part* Pada Bengkel Langgeng Jaya Motor Dengan Metode *Weighted Product* (WP) dibangun dengan bahasa pemrograman PHP dengan *framework* Laravel. *Database* yang digunakan MySQL dengan nama *database* "skripsi.andikas.sql". Sistem menggunakan *class* PenjualanController.php, *class* penjualan.blade.php untuk melakukan proses perhitungan dan *class* Bobot.php. *class* Perusahaan.php. GrupBarang.php untuk mengambil data. Penerapan menggunakan metode *weighted product* menghasilkan rekomendasi *spare part* dengan mengedepankan dari sisi harga sesuai dengan keinginan dari pembeli, untuk pencarian *spare part* Aki Kering, sepeda motor Honda, merk sepeda motor vario 125 dan rentang harga 170000-219900 menghasilkan rekomendasi *spare part* terbaik merk Willwod dengan nilai perhitungan 0,2199221. Dan hasil dari testing semua fitur dapat berjalan dengan baik.

6.2. Saran

Saran dari hasil penelitian yang telah dilakukan oleh peneliti adalah sebagai berikut :

1. Sistem ini dapat dikembangkan ke dalam sistem yang lebih luas dengan menambahkan fitur stok *spare part* dan fitur penjualan *spare part*.
2. Pengembangan Sistem Rekomendasi *Spare Part* Terbaik dapat menggunakan metode lain untuk menciptakan perbandingan antar metode satu dengan metode lainnya.



DAFTAR PUSTAKA

- Departemen Pendidikan Indonesia. (2008). *Kamus Besar Bahasa Indonesia*. Jakarta: Balai Pustaka.
- Hasibuan, M. S. (2007). *Manajemen Sumber Daya Manusia Perusahaan*. Bandung: PT. Bumi Aksa.
- Kersting, W. H. (2017). *Distribution System Modeling and Analysis*. Florida, United States: CRC Press.
- Pressman, R. S. (2001). *Software Engineering: A Practitioner's Approach, Fifth Edition*. New York: McGraw-Hill Higher Education ©2001.
- Rani, S. (2014). Sistem Pendukung Keputusan Pemilihan Sepeda Motor Berbasis WEB Dengan Metode *Weighted Product* (WP).
- Zulkarnain. (2015). Sistem Informasi dan Rekomendasi dan Rekomendasi Pemilihan Homestay Bagi Wisatawan di Kawasan Desa Wisata Menggunakan Metode *Weighted Product* (WP).
- Ralph H. Sprague, J., & Watson, H. J. (1993). *Decision Support Systems: Putting Theory Into Practice*. New Jersey: Prentice Hall.
- Alfita, R. (2011). Sistem Pendukung Keputusan Penentuan prioritas Pengembangan Industri Kecil Menengah Di Kabupaten Bangkalan Madura, Jurusan Teknik Informatika Fakultas, Universitas Trunojoyo.
- Turban, E., J. E. Aronson, dan T. L. (2005). *Decision support systems and intelligent systems*.
- Alfita, R. (2010). Perancangan Sistem Pendukung Keputusan Penentuan Prioritas Produk Unggulan Daerah Menggunakan Metode *Wighted Product* (WP).

LAMPIRAN

A. Lampiran Scenario

A1. Lampiran Scenario Login

Nama	Login
Aktor	Pemilik
Pre-Kondisi	Pemilik masuk ke sistem
Pra-Kondisi	Pemilik sudah melakukan login

SKENARIO NORMAL

“Login”

Aktor	Sistem
1. Membuka	Sistem
Rekomendasi	Pemilihan
<i>Spare Part</i>	
	2. Menampilkan halaman login
	Rekomendasi Pemilihan <i>Spare Part</i>
3. Mengisi E-mail dan Password	
4. Klik tombol Login	
	5. Menampilkan halaman Dashboard
	Rekomendasi Pemilihan <i>Spare Part</i>

SCENARIO ALTERNATIF

“E-mail dan password tidak diisi

Aktor	Sistem
4. Klik tombol Login	
	5. Menampilkan notifikasi “Please fill out this field”

SCENARIO ALTERNATIF

“E-mail dan password salah”

Aktor	Sistem
3. Mengisi E-mail dan Password	
4. Klik tombol Login	
	5. Menampilkan notifikasi "Please fill out this field"

A2. Lampiran *Scenario* Mengelola Data Perusahaan

Nama	Mengelola Data Perusahaan
Aktor	Pemilik
Pre-Condition	Pemilik Memilih Menu Data Sparepart
Pra-Condition	Pemilik berhasil melihat data perusahaan, tambah data perusahaan, mengubah data perusahaan dan hapus data perusahaan

SKENARIO NORMAL

"Melihat Data Perusahaan"

Aksi Aktor	Reaksi Sistem
1. Memilih Menu Sparepart	
	2. Mengambil data perusahaan dari <i>database</i>
	3. Menampilkan Halaman Perusahaan Berisi tabel Perusahaan : a. Id b. Perusahaan c. Aksi Serta terdapat tombol Tambah Perusahaan, List Merk, Update dan Delete

SKENARIO NORMAL	
“Menambah Data Perusahaan”	
Aksi Aktor	Reaksi Sistem
4. Klik Tombol Tambah Perusahaan	
	5. Menampilkan form tambah perusahaan yang berisi : <ul style="list-style-type: none"> a. Id (Hidden) b. Nama Perusahaan Serta terdapat tombol Simpan dan Kembali pada halam form tambah perusahaan.
6. Mengisi form tambah perusahaan	
7. Klik tombol Simpan	
	8. Menyimpan data perusahaan kedalam perusahaan pada <i>database</i>
	9. Mengambil data perusahaan di <i>database</i>
	10. Menampilkan Halaman Perusahaan Berisi tabel Perusahaan : <ul style="list-style-type: none"> a. Id b. Perusahaan c. Aksi Serta terdapat tombol Tambah Perusahaan, List Merk, Update dan Delete
SCENARIO ALTERNATIF	
“Data Belum Diisi”	

Aksi Aktor	Reaksi Sistem
6a. Tidak mengisi form tambah perusahaan	
7a. Klik tombol Simpan	
	8a. Menampilkan notifikasi “ Please fill out this field”

SCENARIO ALTERNATIF

“Batal Menambah Data”

Aksi Aktor	Reaksi Sistem
6b. Mengisi form tambah perusahaan	
7b. Klik tombol Batal	
	8b. Mengambil data perusahaan di <i>database</i>
	9b. Menampilkan Halaman Perusahaan Berisi tabel Perusahaan : <ul style="list-style-type: none"> a. Id b. Perusahaan c. Aksi Serta terdapat tombol Tambah Perusahaan, List Merk, Update dan Delete

SKENARIO NORMAL

“Mengubah Data Perusahaan”

Aksi Aktor	Reaksi Sistem
4. Klik tombol Update	
	5. Mengambil data perusahaan yang terpilih sesuai Id dari <i>database</i>

	<p>6. Menampilkan form tambah perusahaan yang berisi :</p> <ul style="list-style-type: none"> a. Id (Hidden) b. Nama Perusahaan <p>Serta terdapat tombol Simpan dan Kembali pada halaman form tambah perusahaan.</p>
7. Mengubah nama perusahaan	
8. Klik tombol Simpan	
	9. Menyimpan data perusahaan kedalam perusahaan pada <i>database</i>
	10. Mengambil data perusahaan di <i>database</i>
	<p>11. Menampilkan Halaman Perusahaan Berisi tabel Perusahaan :</p> <ul style="list-style-type: none"> a. Id b. Perusahaan c. Aksi <p>Serta terdapat tombol Tambah Perusahaan, List Merk, Update dan Delete</p>

SCENARIO ALTERNATIF

“Data Belum Diisi”

Aksi Aktor	Reaksi Sistem
7a. Tidak mengisi form tambah perusahaan	
8a. Klik tombol Simpan	
	9a. Menampilkan notifikasi “ Please fill out this field”

SCENARIO ALTERNATIF	
“Batal Mengubah Data”	
Aksi Aktor	Reaksi Sistem
7b. Mengisi form tambah perusahaan	
8b. Klik tombol Batal	
	9b. Mengambil data perusahaan di <i>database</i>
	10b. Menampilkan Halaman Perusahaan Berisi tabel Perusahaan : a. Id b. Perusahaan c. Aksi Serta terdapat tombol Tambah Perusahaan, List Merk, Update dan Delete
SKENARIO NORMAL	
“Menghapus Data Perusahaan”	
Aksi Aktor	Reaksi Sistem
4. Klik tombol Delete	
	5. Menampilkan notifikasi “Apakah Anda Yakin Ingin Menghapus” dan tombol Ok dan Cancel
6. Klik tombol Ok	
	7. Menghapus data perusahaan terpilih
	8. Mengambil data perusahaan di <i>database</i>
	9. Menampilkan Halaman Perusahaan Berisi tabel Perusahaan : a. Id

- b. Perusahaan
- c. Aksi

Serta terdapat tombol Tambah Perusahaan, List Merk, Update dan Delete

SCENARIO ALTERNATIF
“Batal Menghapus Data”

Aksi Aktor	Reaksi Sistem
6b. Klik tombol Cancel	7b. Mengambil data perusahaan di <i>database</i>
	8b. Menampilkan Halaman Perusahaan Berisi tabel Perusahaan : <ul style="list-style-type: none"> a. Id b. Perusahaan c. Aksi <p>Serta terdapat tombol Tambah Perusahaan, List Merk, Update dan Delete</p>

A3. Lampiran *Scenario* Mengelola Data *Merk*

Nama	Mengelola Data <i>Merk</i>
Aktor	Pemilik
Pre-Condition	Pemilik Memilih Tombol Data <i>Merk</i> Pada Halaman Perusahaan
Pra-Condition	Pemilik berhasil melihat data <i>merk</i> , tambah data <i>merk</i> , mengubah data <i>merk</i> dan hapus data <i>merk</i>

SKENARIO NORMAL	
“Melihat Data Merk”	
Aksi Aktor	Reaksi Sistem
1. Memilih Tombol List Merk	
	2. Mengambil data merk dari <i>database</i>
	3. Menampilkan Halaman Merk Berisi tabel Merk : a. Id b. Merk c. Aksi Serta terdapat tombol Tambah Merk, List Sparepart, Update dan Delete

SKENARIO NORMAL	
“Menambah Data Merk”	
Aksi Aktor	Reaksi Sistem
4. Klik Tombol Tambah Merk	
	5. Menampilkan form tambah merk yang berisi : c. Id (<i>Hidden</i>) d. Nama Merk Serta terdapat tombol Simpan dan Kembali pada halam form tambah merk.
6. Mengisi form tambah merk	
7. Klik tombol Simpan	
	8. Menyimpan data merk kedalam perusahaan pada <i>database</i>
	9. Mengambil data merk di <i>database</i>

10. Menampilkan Halaman Merk Berisi tabel Merk :
- a. Id
 - b. Merk
 - c. Aksi
- Serta terdapat tombol Tambah Merk, List Sparepart, Update dan Delete

SCENARIO ALTERNATIF

“Data Belum Diisi”

Aksi Aktor	Reaksi Sistem
6a. Tidak mengisi form tambah merk	
7a. Klik tombol Simpan	
	8a. Menampilkan notifikasi “ Please fill out this field”

SCENARIO ALTERNATIF

“Batal Menambah Data”

Aksi Aktor	Reaksi Sistem
6b. Mengisi form tambah merk	
7b. Klik tombol Batal	
	8b. Mengambil data merk di <i>database</i>
	9b. Menampilkan Halaman Merk Berisi tabel Merk : <ol style="list-style-type: none"> a. Id b. Perusahaan c. Aksi

Serta terdapat tombol Tambah Perusahaan, List Sparepart, Update dan Delete

SKENARIO NORMAL	
“Mengubah Data Merk”	
Aksi Aktor	Reaksi Sistem
12. Klik tombol Update	
	13. Mengambil data merk yang terpilih sesuai Id dari <i>database</i>
	14. Menampilkan form tambah merk yang berisi : <ul style="list-style-type: none"> a. Id (<i>Hidden</i>) b. Nama Merk Serta terdapat tombol Simpan dan Kembali pada halaman form tambah merk.
15. Mengubah nama merk	
16. Klik tombol Simpan	
	17. Menyimpan data merk ke dalam merk pada <i>database</i>
	18. Mengambil data merk di <i>database</i>
	19. Menampilkan Halaman Perusahaan <p>Berisi tabel Perusahaan :</p> <ul style="list-style-type: none"> a. Id b. Perusahaan c. Aksi Serta terdapat tombol Tambah merk, List Sparepart, Update dan Delete

SCENARIO ALTERNATIF	
"Data belum diisi"	
Aksi Aktor	Reaksi Sistem
7a. Tidak mengisi form tambah merk	
8a. Klik tombol Simpan	
	9a. Menampilkan notifikasi " Please fill out this field"
SCENARIO ALTERNATIF	
"Batal Mengubah Data"	
Aksi Aktor	Reaksi Sistem
7b. Mengisi form tambah merk	
8b. Klik tombol Batal	
	9b. Mengambil data perusahaan di <i>database</i>
	10b. Menampilkan Halaman Merk Berisi tabel Merk : a. Id b. Perusahaan c. Aksi Serta terdapat tombol Tambah Merk, List Sparepart, Update dan Delete
SKENARIO NORMAL	
"Menghapus Data Merk"	
Aksi Aktor	Reaksi Sistem
10. Klik tombol Delete	

	11. Menampilkan notifikasi “Apakah Anda Yakin Ingin Menghapus” dan tombol Ok dan Cancel
12. Klik tombol Ok	
	13. Menghapus data merk terpilih
	14. Mengambil data merk di <i>database</i>
	15. Menampilkan Halaman Merk Berisi tabel Merk : <ul style="list-style-type: none"> a. Id b. Perusahaan c. Aksi <p>Serta terdapat tombol Tambah Merk, List Sparepart, Update dan Delete</p>

SCENARIO ALTERNATIF
“Batal Menghapus Data”

Aksi Aktor	Reaksi Sistem
6b. Klik tombol Cancel	
	7b. Mengambil data merk di <i>database</i>
	8b. Menampilkan Halaman Merk Berisi tabel Merk : <ul style="list-style-type: none"> a. Id b. Perusahaan c. Aksi <p>Serta terdapat tombol Tambah Merk, List Sparepart, Update dan Delete</p>

A4. Lampiran *Scenario* Mengelola Data *Spare Part*

Nama	Mengelola Data <i>Spare Part</i>
Aktor	Pemilik
Pre-Condition	Pemilik Memilih Tombol Data <i>Spare Part</i> Pada Halaman <i>Merk</i>
Pra-Condition	Pemilik berhasil melihat data <i>spare part</i> , tambah data <i>spare part</i> dan hapus data <i>spare part</i>

SKENARIO NORMAL

“Melihat Data *Spare Part*”

Aksi Aktor	Reaksi Sistem
1. Memilih Tombol List Sparepart	
	2. Mengambil data sparepart dari <i>database</i>
	3. Menampilkan Halaman Sparepart Berisi tabel Sparepart : a. Id b. Grup Barang c. Sparepart d. Harga e. Garansi f. Ketahanan g. Aksi Serta terdapat tombol Tambah Sparepart dan Delete

SKENARIO NORMAL

“Menambah Data *Spare Part*”

Aksi Aktor	Reaksi Sistem
4. Klik Tombol Tambah Merk	

	<p>5. Menampilkan form tambah <i>spare part</i> yang berisi :</p> <ol style="list-style-type: none"> a. Id (<i>Hidden</i>) b. Nama Sparepart c. Grup Barang d. Harga e. Garansi f. Ketahanan g. Nilai Harga h. Nilai Garansi i. Nilai Ketahanan <p>Serta terdapat tombol Simpan dan Kembali pada halaman form tambah merk.</p>
6. Mengisi form tambah merk	
7. Klik tombol Simpan	
	8. Menyimpan data sparepart kedalam sparepart pada <i>database</i>
	9. Mengambil data merk di <i>database</i>
	<p>10. Menampilkan Halaman Sparepart Berisi tabel Sparepart :</p> <ol style="list-style-type: none"> a. Id b. Grup Barang c. Sparepart d. Harga e. Garansi f. Ketahanan g. Aksi

Serta terdapat tombol Tambah Sparepart dan Delete

SCENARIO ALTERNATIF

“Data Belum Diisi”

Aksi Aktor	Reaksi Sistem
6a. Tidak mengisi form tambah sparepart	
7a. Klik tombol Simpan	
	8a. Menampilkan notifikasi “ Please fill out this field”

SCENARIO ALTERNATIF

“Batal Menambah Data”

Aksi Aktor	Reaksi Sistem
6b. Mengisi form tambah sparepart	
7b. Klik tombol Batal	
	8b. Mengambil data sparepart di <i>database</i>
	9b. Menampilkan Halaman Sparepart Berisi tabel Sparepart : a. Id b. Grup Barang c. Sparepart d. Harga e. Garansi f. Ketahanan g. Aksi

Serta terdapat tombol Tambah Sparepart dan Delete

SKENARIO NORMAL	
“Menghapus Data Spare Part”	
Aksi Aktor	Reaksi Sistem
4. Klik tombol Delete	
	5. Menampilkan notifikasi “Apakah Anda Yakin Ingin Menghapus” dan tombol Ok dan Cancel
6. Klik tombol Ok	
	7. Menghapus data merk terpilih
	8. Mengambil data merk di <i>database</i>
	9. Menampilkan Halaman Sparepart Berisi tabel Sparepart : <ul style="list-style-type: none"> a. Id b. Grup Barang c. Sparepart d. Harga e. Garansi f. Ketahanan g. Aksi <p>Serta terdapat tombol Tambah Sparepart dan Delete</p>
SCENARIO ALTERNATIF	
“Batal Menghapus Data”	
Aksi Aktor	Reaksi Sistem
6b. Klik tombol Cancel	
	7b. Mengambil data merk di <i>database</i>
	8b. Menampilkan Halaman Sparepart Berisi tabel Sparepart :

- a. Id
- b. Grup Barang
- c. Sparepart
- d. Harga
- e. Garansi
- f. Ketahanan
- g. Aksi

Serta terdapat tombol Tambah Sparepart dan Delete

A5. Lampiran *Scenaro* Mengelola Data Grup Barang

Nama	Mengelola Data Grup Barang
Aktor	Pemilik
Pre-Condition	Pemilik Memilih Menu Data Grup Barang
Pra-Condition	Pemilik berhasil melihat data grup barang, tambah data grup barang, mengubah data grup barang dan hapus data grup barang

SKENARIO NORMAL

“Melihat Data Grup Barang”

Aksi Aktor	Reaksi Sistem
1. Memilih Menu Grup Barang	
	2. Mengambil data grup barang dari <i>database</i>
	3. Menampilkan Halaman Grup Barang Berisi tabel Grup Barang : <ul style="list-style-type: none"> a. Id b. Grup Barang c. Aksi

Serta terdapat tombol Tambah Grup Barang, Kriteria, Update dan Delete

SKENARIO NORMAL

“Menambah Data Grup Barang”

Aksi Aktor	Reaksi Sistem
4. Klik Tombol Tambah Grup Barang	
	5. Menampilkan form tambah grup barang yang berisi : <ul style="list-style-type: none"> a. Id (Hidden) b. Nama Grup Barang Serta terdapat tombol Simpan dan Kembali pada halaman form tambah perusahaan.
6. Mengisi form tambah grup barang	
7. Klik tombol Simpan	
	8. Menyimpan data grup barang ke dalam tabel grup barang pada <i>database</i>
	9. Mengambil data grup barang di <i>database</i>
	10. Menampilkan Halaman Grup Barang Berisi tabel Grup Barang : <ul style="list-style-type: none"> a. Id b. Grup Barang c. Aksi Serta terdapat tombol Tambah Grup Barang, Kriteria, Update dan Delete

SCENARIO ALTERNATIF**“Data Belum Diisi”**

Aksi Aktor	Reaksi Sistem
6a. Tidak mengisi form tambah grup barang	
7a. Klik tombol Simpan	
	8a. Menampilkan notifikasi “ Please fill out this field”

SCENARIO ALTERNATIF**“Batal Menambah Data”**

Aksi Aktor	Reaksi Sistem
6b. Mengisi form tambah grup barang	
7b. Klik tombol Batal	
	8b. Mengambil data grup barang di <i>database</i>
	9b. Menampilkan Halaman Grup Barang Berisi tabel Grup Barang : a. Id b. Grup Barang c. Aksi Serta terdapat tombol Tambah Grup Barang, Kriteria, Update dan Delete

SKENARIO NORMAL**“Mengubah Data Grup Barang”**

Aksi Aktor	Reaksi Sistem
4. Klik tombol Update	
	5. Mengambil data grup barang yang terpilih sesuai Id dari <i>database</i>

	<p>6. Menampilkan form tambah grup barang yang berisi :</p> <ul style="list-style-type: none"> a. Id (Hidden) b. Nama Grup Barang <p>Serta terdapat tombol Simpan dan Kembali pada halaman form tambah grup barang.</p>
7. Mengubah nama grup barang	
8. Klik tombol Simpan	
	9. Menyimpan data grup barang kedalam tabel grup barang pada <i>database</i>
	10. Mengambil data grup barang di <i>database</i>
	<p>11. Menampilkan Halaman Grup Barang Berisi tabel Grup Barang:</p> <ul style="list-style-type: none"> a. Id b. Grup Barang c. Aksi <p>Serta terdapat tombol Tambah Grup Barang, Kriteria, Update dan Delete</p>

SCENARIO ALTERNATIF

“Data Belum Diisi”

Aksi Aktor	Reaksi Sistem
7a. Tidak mengisi form tambah grup barang	
8a. Klik tombol Simpan	

9a. Menampilkan notifikasi “ Please fill out this field”

SCENARIO ALTERNATIF

“Batal Mengubah Data”

Aksi Aktor	Reaksi Sistem
7b. Mengisi form tambah grup barang	
8b. Klik tombol Batal	
	9b. Mengambil data grup barang di <i>database</i>
	10b. Menampilkan Halaman Grup Barang Berisi tabel Grup Barang: a. Id b. Perusahaan c. Aksi Serta terdapat tombol Tambah Grup Barang, Kriteria, Update dan Delete

SKENARIO NORMAL

“Menghapus Data Grup Barang”

Aksi Aktor	Reaksi Sistem
4. Klik tombol Delete	
	5. Menampilkan notifikasi “Apakah Anda Yakin Ingin Menghapus” dan tombol Ok dan Cancel
6. Klik tombol Ok	
	7. Menghapus data grup barang terpilih
	8. Mengambil data grup barang di <i>database</i>

9. Menampilkan Halaman Grup Barang
 Berisi tabel Grup Barang :
- a. Id
 - b. Grup Barang
 - c. Aksi
- Serta terdapat tombol Tambah Perusahaan, List Merk, Update dan Delete

SCENARIO ALTERNATIF

“Batal Menghapus Data”

Aksi Aktor	Reaksi Sistem
6b. Klik tombol Cancel	
	7b. Mengambil data grup barang di <i>database</i>
	8b. Menampilkan Halaman Grup Barang Berisi tabel Grup Barang: a. Id b. grup barang c. Aksi Serta terdapat tombol Tambah Perusahaan, Kriteria, Update dan Delete

A6. Lampiran *Scenaro* Mengelola Data Kriteria

Nama	Mengelola Data Kriteria
Aktor	Pemilik
Pre-Condition	Pemilik Memilih Tombol Kriteria Pada Halaman Grup Barang

Pra-Condition	Pemilik berhasil melihat data kriteria, tambah data kriteria, mengubah data kriteria dan hapus data kriteria
---------------	--

SKENARIO NORMAL
“Melihat Data Kriteria”

Aksi Aktor	Reaksi Sistem
1. Memilih Menu Kriteria	
	2. Mengambil data kriteria dari <i>database</i>
	3. Menampilkan Halaman Kriteria Berisi tabel Kriteria : a. Id b. Keterangan c. Kriteria d. Aksi Serta terdapat tombol Tambah Kriteria, Edit Sub, Update dan Delete

SKENARIO NORMAL
“Menambah Data Kriteria”

Aksi Aktor	Reaksi Sistem
4. Klik Tombol Tambah Kriteria	
	5. Menampilkan form tambah kriteria yang berisi : c. Id (Hidden) d. Nama Kriteria Serta terdapat tombol Simpan dan Kembali pada halam form tambah kriteria.
6. Mengisi form tambah kriteria	

7. Klik tombol Simpan	
	8. Menyimpan data kriteria ke dalam tabel kriteria pada <i>database</i>
	9. Mengambil data kriteria di <i>database</i>
	10. Menampilkan Halaman Kriteria Berisi tabel Kriteria : a. Id b. Keterangan c. Kriteria d. Aksi Serta terdapat tombol Tambah Kriteria, Edit Sub, Update dan Delete

SCENARIO ALTERNATIF

“Data Belum Diisi”

Aksi Aktor	Reaksi Sistem
6a. Tidak mengisi form tambah kriteria	
7a. Klik tombol Simpan	
	8a. Menampilkan notifikasi “ Please fill out this field”

SCENARIO ALTERNATIF

“Batal Menambah Data”

Aksi Aktor	Reaksi Sistem
6b. Mengisi form tambah kriteria	
7b. Klik tombol Batal	
	8b. Mengambil data kriteria di <i>database</i>
	9b. Menampilkan Halaman Kriteria Berisi tabel Kriteria :

- a. Id
- b. Keterangan
- c. Kriteria
- d. Aksi

Serta terdapat tombol Tambah Kriteria, Edit Sub, Update dan Delete

SKENARIO NORMAL	
“Mengubah Data Kriteria”	
Aksi Aktor	Reaksi Sistem
4. Klik tombol Update	
	5. Mengambil data kriteria yang terpilih sesuai Id dari <i>database</i>
	6. Menampilkan form ubah kriteria sesuai id terpilih yang berisi : <ul style="list-style-type: none"> e. Id (Hidden) f. Nama Kriteria Serta terdapat tombol Simpan dan Kembali pada halaman form tambah kriteria.
7. Mengubah nama kriteria	
8. Klik tombol Simpan	
	9. Menyimpan data kriteria kedalam tabel kriteria pada <i>database</i>
	10. Mengambil data kriteria di <i>database</i>
	11. Menampilkan Halaman Kriteria Berisi tabel Kriteria : <ul style="list-style-type: none"> a. Id b. Keterangan

- c. Kriteria
 - d. Aksi
- Serta terdapat tombol Tambah Kriteria, Kembali, Edit Sub, Update dan Delete

SCENARIO ALTERNATIF
“Data Belum Diisi”

Aksi Aktor	Reaksi Sistem
7a. Tidak mengisi form tambah kriteria	
8a. Klik tombol Simpan	
	9a. Menampilkan notifikasi “ Please fill out this field”

SCENARIO ALTERNATIF
“Batal Mengubah Data”

Aksi Aktor	Reaksi Sistem
7b. Mengisi form tambah kriteria	
8b. Klik tombol Batal	
	9b. Mengambil data kriteria di <i>database</i>
	10b. Menampilkan Halaman Kriteria Berisi tabel Kriteria : <ul style="list-style-type: none"> a. Id b. Keterangan c. Kriteria d. Aksi

Serta terdapat tombol Tambah Sub Kriteria, Kembali, Edit Sub, Update dan Delete

SKENARIO NORMAL	
“Menghapus Data Kriteria”	
Aksi Aktor	Reaksi Sistem
4. Klik tombol Delete	
	5. Menampilkan notifikasi “Apakah Anda Yakin Ingin Menghapus” dan tombol Ok dan Cancel
6. Klik tombol Ok	
	7. Menghapus data kriteria terpilih
	8. Mengambil data kriteria di <i>database</i>
	9. Menampilkan Halaman Kriteria Berisi tabel Kriteria : e. Id f. Keterangan g. Kriteria h. Aksi Serta terdapat tombol Tambah Kriteria, Edit Sub, Update dan Delete
SCENARIO ALTERNATIF	
“Batal Menghapus Data”	
Aksi Aktor	Reaksi Sistem
6b. Klik tombol Cancel	
	7b. Mengambil data kriteria di <i>database</i>
	8b. Menampilkan Halaman Kriteria Berisi tabel Kriteria :

- a. Id
- b. Keterangan
- c. Kriteria
- d. Aksi

Serta terdapat tombol Tambah Kriteria, Edit Sub, Update dan Delete

A7. Lampiran *Scenaro* Mengelola Data Sub Kriteria

Nama	Mengelola Data Sub Kriteria
Aktor	Pemilik
Pre-Condition	Pemilik Memilih Tombol Edit Sub Pada Halaman Kriteria
Pra-Condition	Pemilik berhasil melihat data sub kriteria, tambah data sub kriteria, mengubah data sub kriteria dan hapus data sub kriteria

SKENARIO NORMAL

“Melihat Data Sub Kriteria”

Aksi Aktor	Reaksi Sistem
1. Memilih Tombol Edit Sub	2. Mengambil data sub kriteria dari <i>database</i>
	3. Menampilkan Halaman Sub Kriteria Berisi tabel Sub Kriteria : <ul style="list-style-type: none"> a. Id b. Sub Keterangan c. Bobot d. Kriteria e. Aksi

Serta terdapat tombol Tambah Sub Kriteria, Kembali, Update dan Delete

SKENARIO NORMAL

“Menambah Data Sub Kriteria”

Aksi Aktor	Reaksi Sistem
4. Klik Tombol Tambah Sub Kriteria	
	<p>5. Menampilkan form tambah sub kriteria yang berisi :</p> <ul style="list-style-type: none"> a. Id (Hidden) b. Nama Sub Kriteria c. Bobot d. Keterangan <p>Serta terdapat tombol Simpan dan Kembali pada halaman form tambah kriteria.</p>
6. Mengisi form tambah sub kriteria	
7. Klik tombol Simpan	
	8. Menyimpan data sub kriteria ke dalam tabel sub kriteria pada <i>database</i>
	9. Mengambil data sub kriteria di <i>database</i>
	<p>10. Menampilkan Halaman Sub Kriteria Berisi tabel Sub Kriteria :</p> <ul style="list-style-type: none"> a. Id b. Sub Keterangan c. Bobot d. Kriteria e. Aksi

Serta terdapat tombol Tambah Sub Kriteria, Kembali, Update dan Delete

SCENARIO ALTERNATIF "Data Belum Diisi"	
Aksi Aktor	Reaksi Sistem
6a. Tidak mengisi form tambah sub kriteria	
7a. Klik tombol Simpan	
	8a. Menampilkan notifikasi " Please fill out this field"

SCENARIO ALTERNATIF "Batal Menambah Data"	
Aksi Aktor	Reaksi Sistem
6b. Mengisi form tambah sub kriteria	
7b. Klik tombol Batal	
	8b. Mengambil data sub kriteria di <i>database</i>
	9b. Menampilkan Halaman Sub Kriteria Berisi tabel Sub Kriteria : a. Id b. Sub Keterangan c. Bobot d. Kriteria e. Aksi

Serta terdapat tombol Tambah Sub Kriteria, Kembali, Update dan Delete

SKENARIO NORMAL	
“Mengubah Data Sub Kriteria”	
Aksi Aktor	Reaksi Sistem
4. Klik tombol Update	
	5. Mengambil data sub kriteria yang terpilih sesuai Id dari <i>database</i>
	6. Menampilkan form ubah sub kriteria sesuai id terpilih yang berisi : <ol style="list-style-type: none"> a. Id (Hidden) b. Nama Sub Kriteria c. Bobot d. Keterangan <p>Serta terdapat tombol Simpan dan <i>Close</i> pada form ubah Sub kriteria.</p>
7. Mengubah form ubah sub kriteria	
8. Klik tombol Simpan	
	9. Menyimpan data sub kriteria kedalam tabel sub kriteria pada <i>database</i>
	10. Mengambil data sub kriteria di <i>database</i>
	11. Menampilkan Halaman Sub Kriteria <p>Berisi tabel Sub Kriteria :</p> <ol style="list-style-type: none"> a. Id b. Sub Keterangan c. Bobot d. Kriteria e. Aksi

Serta terdapat tombol Tambah Sub Kriteria, Kembali, Update dan Delete

SCENARIO ALTERNATIF

“Data Belum Diisi”

Aksi Aktor	Reaksi Sistem
7a. Tidak mengisi form ubah sub kriteria	
8a. Klik tombol Simpan	
	9a. Menampilkan notifikasi “ Please fill out this field”

SCENARIO ALTERNATIF

“Batal Mengubah Data”

Aksi Aktor	Reaksi Sistem
7b. Mengisi form ubah sub kriteria	
8b. Klik tombol Batal	
	9b. Mengambil data sub kriteria di <i>database</i>
	10b. Menampilkan Halaman Sub Kriteria Berisi tabel sub kriteria : <ul style="list-style-type: none"> a. Id b. Sub Kriteria c. Keterangan d. Bobot e. Aksi <p>Serta terdapat tombol Tambah Sub Kriteria, Kembali, Update dan Delete</p>

SKENARIO NORMAL

“Menghapus Data Sub Kriteria”	
Aksi Aktor	Reaksi Sistem
4. Klik tombol Delete	
	5. Menampilkan notifikasi “Apakah Anda Yakin Ingin Menghapus” dan tombol Ok dan Cancel
6. Klik tombol Ok	
	7. Menghapus data sub kriteria terpilih
	8. Mengambil data sub kriteria di <i>database</i>
	9. Menampilkan Halaman Sub Kriteria Berisi tabel sub kriteria : <ul style="list-style-type: none"> a. Id b. Sub Kriteria c. Keterangan d. Bobot e. Aksi Serta terdapat tombol Tambah Sub Kriteria, Kembali, Update dan Delete

SCENARIO ALTERNATIF**“Batal Menghapus Data”**

Aksi Aktor	Reaksi Sistem
6b. Klik tombol Cancel	
	7b. Mengambil data sub kriteria di <i>database</i>
	8b. Menampilkan Halaman Sub Kriteria Berisi tabel sub kriteria : <ul style="list-style-type: none"> a. Id b. Sub Kriteria

c. Keterangan

d. Bobot

e. Aksi

Serta terdapat tombol Tambah Sub Kriteria, Kembali, Update dan Delete

A8. Lampiran *Scenaro* Mengelola Data Bobot Kriteria

Nama	Mengelola Data Bobot Kriteria
Aktor	Pemilik
Pre-Condition	Pemilik Memilih Menu Bobot Pada Halaman Awal
Pra-Condition	Pemilik berhasil melihat data bobot kriteria, dan mengubah data bobot kriteria.

SKENARIO NORMAL

“Melihat Data Bobot Kriteria”

Aksi Aktor	Reaksi Sistem
1. Memilih Menu Bobot	
	2. Mengambil data bobot kriteria dari <i>database</i>
	3. Menampilkan Halaman Bobot Kriteria Berisi tabel Bobot Kriteria : a. Id b. Nama Kriteria c. Keterangan d. Bobot Kriteria e. Aksi Serta terdapat Update.

SKENARIO NORMAL	
“Mengubah Data Sub Kriteria”	
Aksi Aktor	Reaksi Sistem
4. Klik tombol Update	
	5. Mengambil data bobot kriteria yang terpilih sesuai Id dari <i>database</i>
	6. Menampilkan form ubah bobot kriteria sesuai id terpilih yang berisi : <ul style="list-style-type: none"> a. Id (Hidden) b. Bobot Serta terdapat tombol Simpan dan <i>Close</i> pada halaman form ubah Sub kriteria.
7. Mengubah form ubah bobot kriteria	
8. Klik tombol Simpan	
	9. Menyimpan data bobot kriteria ke dalam tabel bobot kriteria pada <i>database</i>
	10. Mengambil data bobot kriteria di <i>database</i>
	11. Menampilkan Halaman Bobot Kriteria Berisi tabel Bobot Kriteria : <ul style="list-style-type: none"> a. Id b. Nama Kriteria c. Keterangan d. Bobot Kriteria e. Aksi Serta terdapat Update.

SCENARIO ALTERNATIF	
“Data Belum Diisi”	
Aksi Aktor	Reaksi Sistem
7a. Tidak mengisi form ubah bobot kriteria	
8a. Klik tombol Simpan	
	9a. Menampilkan notifikasi “ Please fill out this field”
SCENARIO ALTERNATIF	
“Batal Mengubah Data”	
Aksi Aktor	Reaksi Sistem
7b. Mengisi form ubah bobot kriteria	
8b. Klik tombol Batal	
	9b. Mengambil data bobot kriteria di <i>database</i>
	10b. Menampilkan Halaman Bobot Kriteria Berisi tabel Bobot kriteria : <ul style="list-style-type: none"> a. Id b. Nama Kriteria c. Keterangan d. Bobot e. Aksi Serta terdapat tombol Update.

A9. Lampiran *Scenario* Mengelola Data Rekomendasi Perhitungan (Karyawan)

Nama	Mengelola Rekomendasi Pemilihan
Aktor	Karyawan

Pre-Condition	Karyawan Memilih Tombol Perhitungan Pada Halaman Login
Pra-Condition	Pemilik berhasil melakukan perangkingan rekomendasi <i>spare part</i>

SKENARIO NORMAL

“Menghitung Rekomendasi *Spare Part*”

Aksi Aktor	Reaksi Sistem
1. Pilih menu Perhitungan	2. Menampilkan halaman perhitungan dengan tabel. <ul style="list-style-type: none"> a. Keterangan <ul style="list-style-type: none"> i. Id ii. Perusahaan iii. <i>Merek</i> iv. <i>Spare Part</i> b. Menghitung Nilai Preferensi <ul style="list-style-type: none"> i. S1 ii. S2 iii. S3 iv. S4 v. S5 vi. S6 c. Hasil Perkalian d. Total e. Rangking <p>Serta terdapat tombol Lakukan Perhitungan</p>
3. Klik tombol Lakukan Perhitungam	

	<p>4. Menampilkan modal show form lakukan perhiungan</p> <ul style="list-style-type: none"> f. <i>Drobdown</i> Grub Barang g. <i>Drobdown</i> Perusahaan 1 h. <i>Drobdown</i> Perusahaan 2 i. <i>Drobdown</i> Perusahaan 3 j. <i>Tab Conten Merk</i> <p>Serta terdapat tombol Hitung dan Close</p>
5. Klik tombol Lakukan Hitung	
	6. Melakukan proses perhitungan
	<p>7. Menampilkan halaman perhitungan dan tabel berisi perhitungan yang telah dilakukan.</p> <ul style="list-style-type: none"> a. Keterangan <ul style="list-style-type: none"> i. Id ii. Perusahaan iii. <i>Merek</i> iv. <i>Spare Part</i> b. Menghitung Nilai Preferensi <ul style="list-style-type: none"> i. S1 ii. S2 iii. S3 iv. S4 v. S5 vi. S6 c. Hasil Perkalian d. Total e. Rangking

Serta terdapat tombol Lakukan Perhitungan

SCENARIO ALTERNATIF

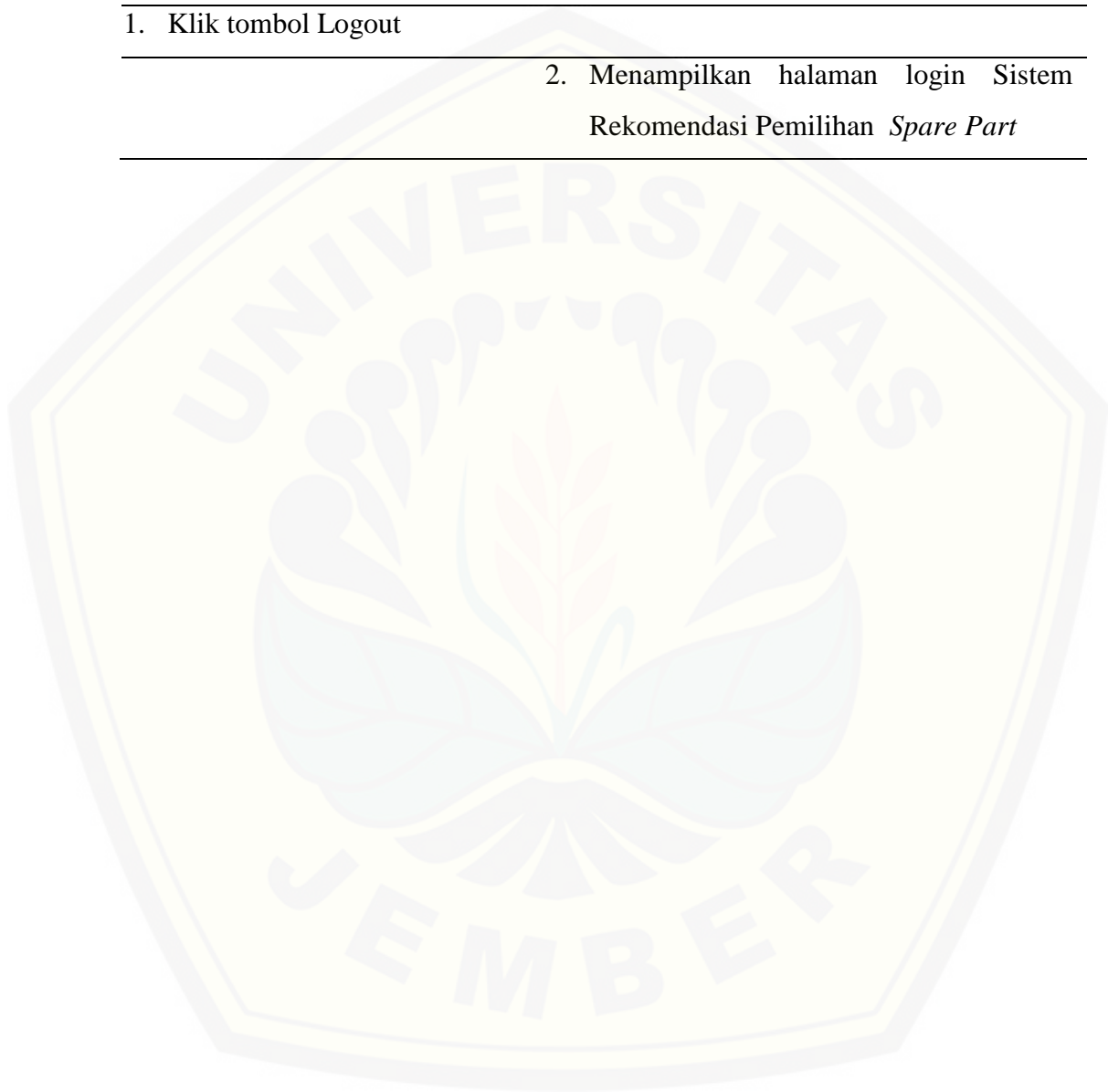
“Batal Menghitung Rekomendasi”

Aksi Aktor	Reaksi Sistem
5a. Klik tombol Close	<p>6a. Menampilkan halaman perhitungan dengan tabel.</p> <ul style="list-style-type: none"> a. Keterangan <ul style="list-style-type: none"> i. Id ii. Perusahaan iii. <i>Merek</i> iv. <i>Spare Part</i> b. Menghitung Nilai Preferensi <ul style="list-style-type: none"> i. S1 ii. S2 iii. S3 iv. S4 v. S5 vi. S6 c. Hasil Perkalian d. Total e. Rangking <p>Serta terdapat tombol Lakukan Perhitungan</p>

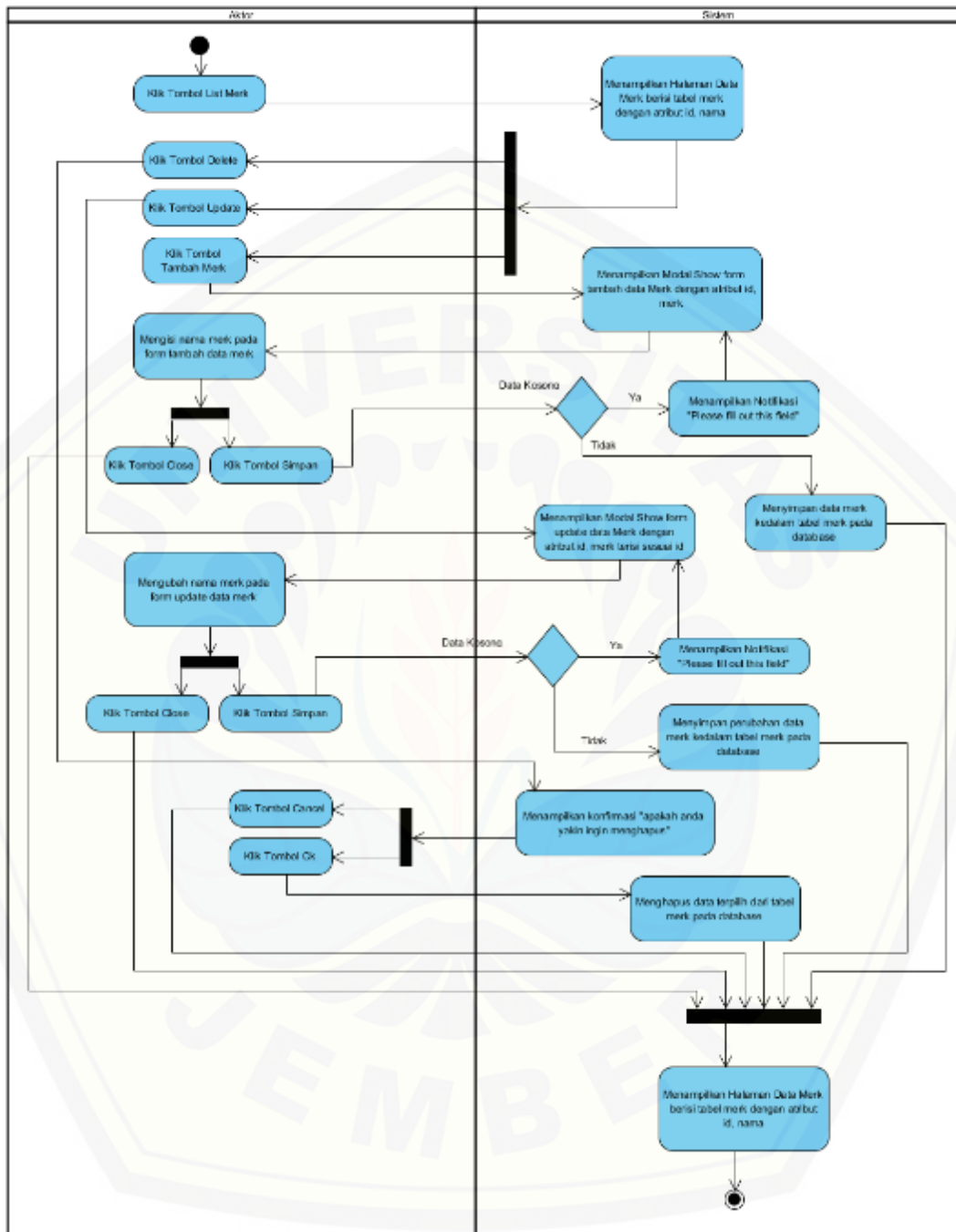
A10. Lampiran *Scenario Logout*

Nama	Logout
Aktor	Pemilik
Pre-Kondisi	Pemilik keluar dari system

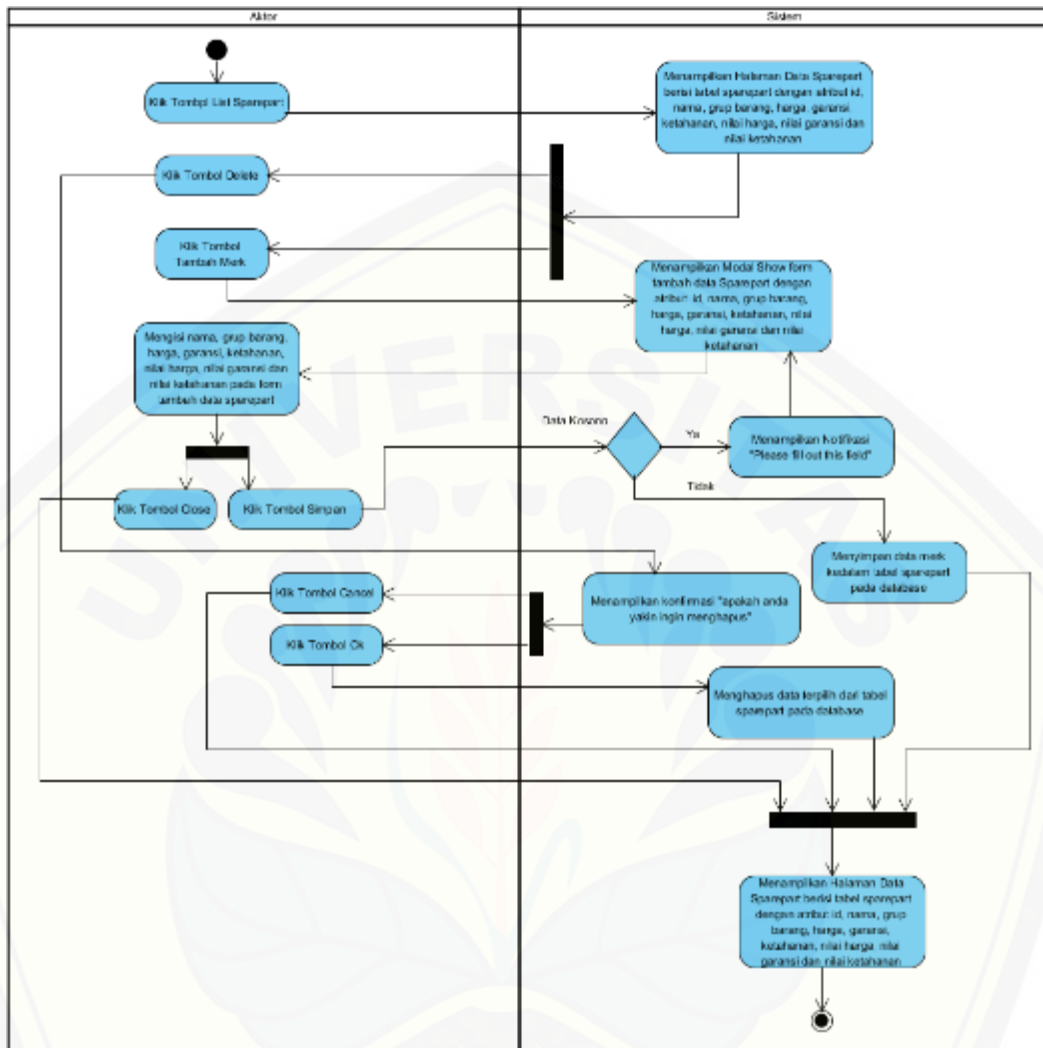
Pra-Kondisi	Pemilik sudah melakukan logout
SKENARIO NORMAL	
“Logout”	
Aktor	Sistem
1. Klik tombol Logout	2. Menampilkan halaman login Sistem Rekomendasi Pemilihan <i>Spare Part</i>



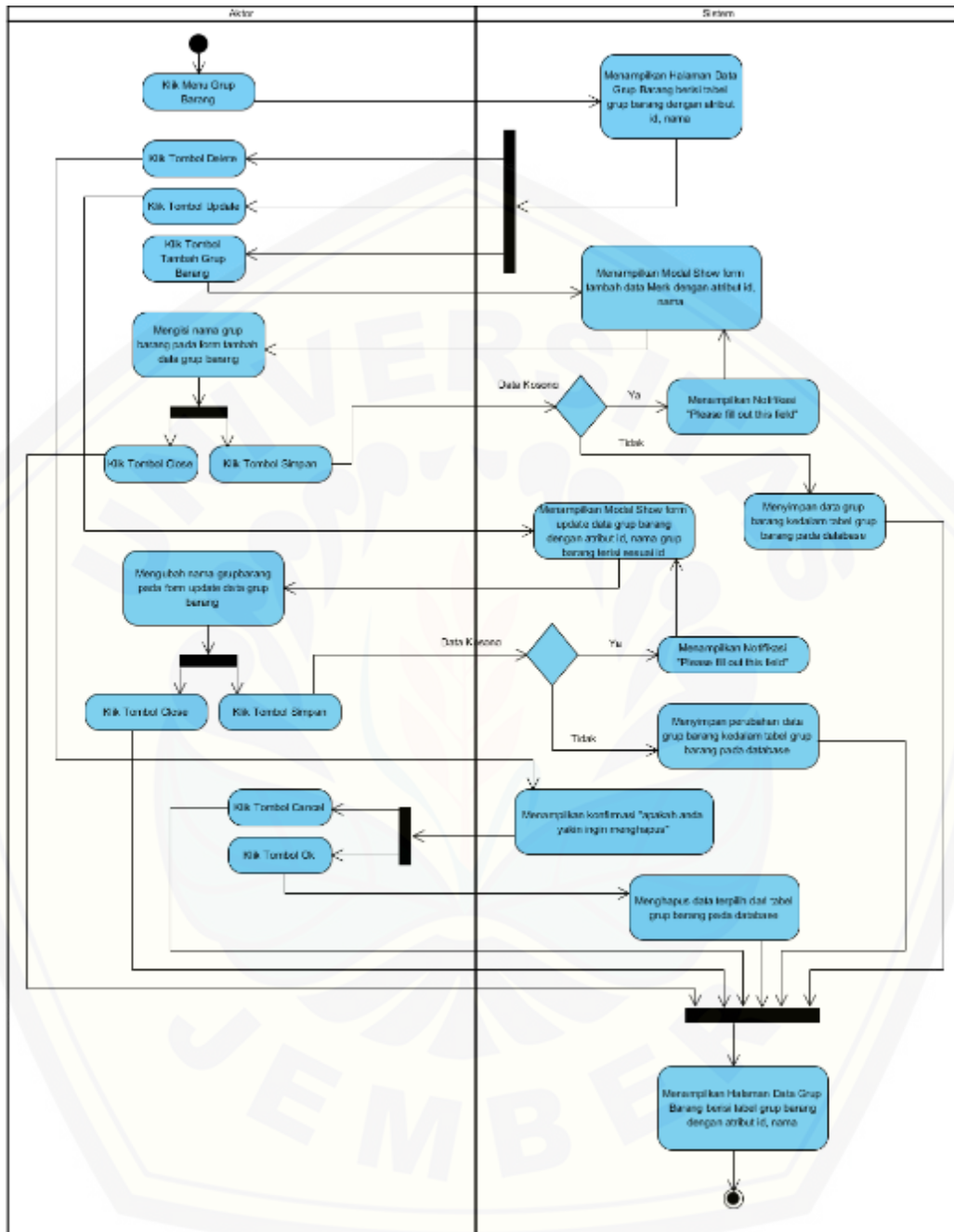
B2. Lampiran Activity Diagram Mengelola Data Merk



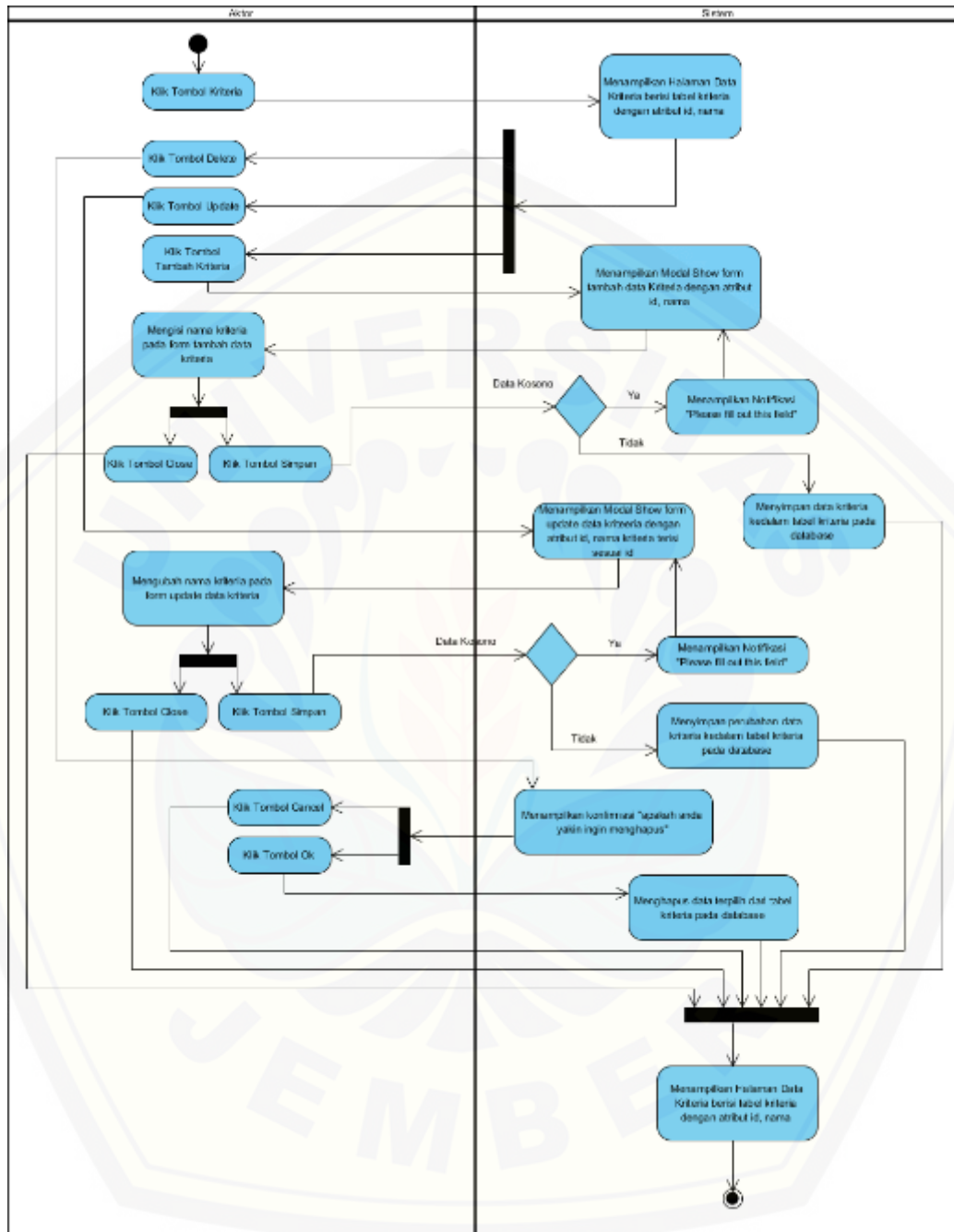
B3. Lampiran Activity Diagram Mengelola Data Spare Part



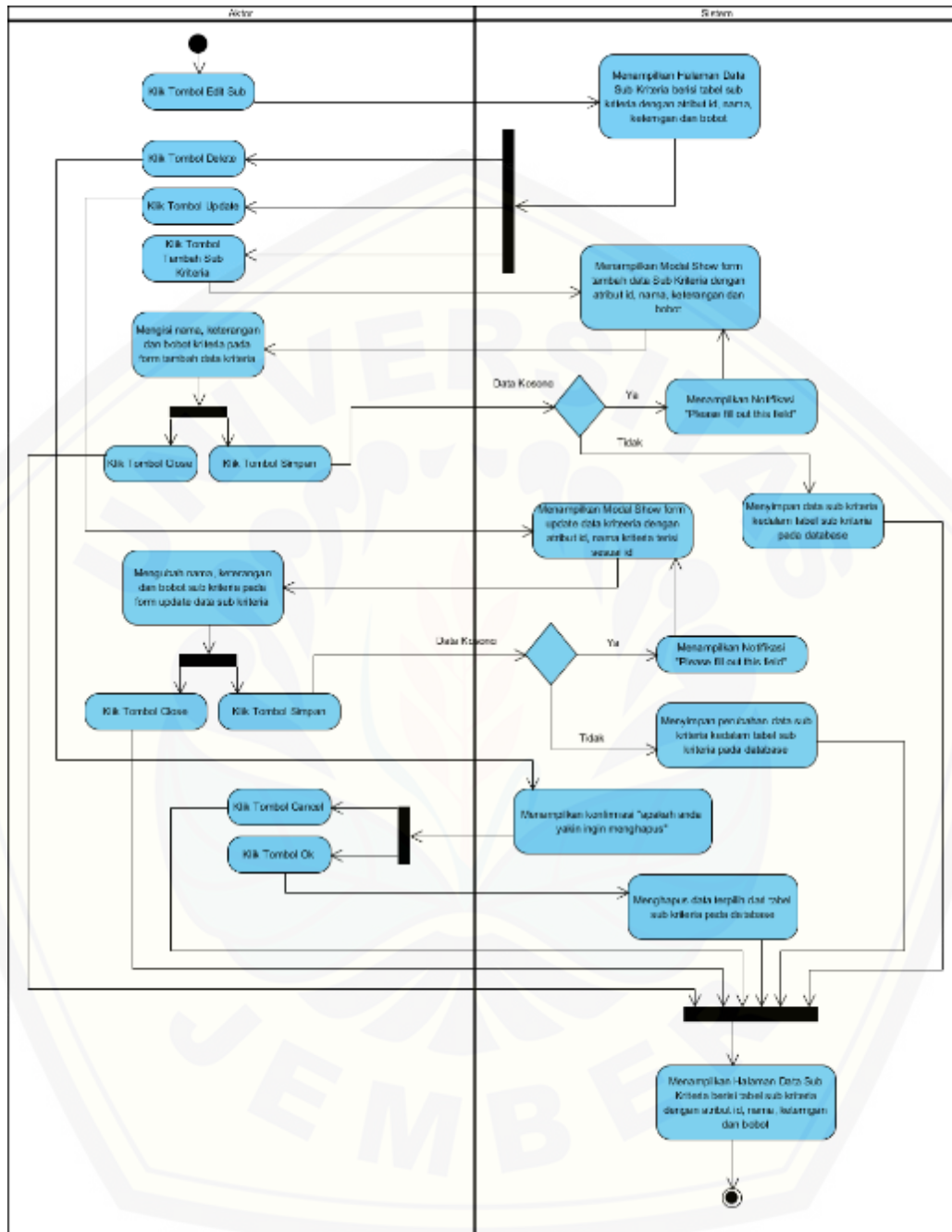
B4. Lampiran Activity Diagram Mengelola Data Grup Barang



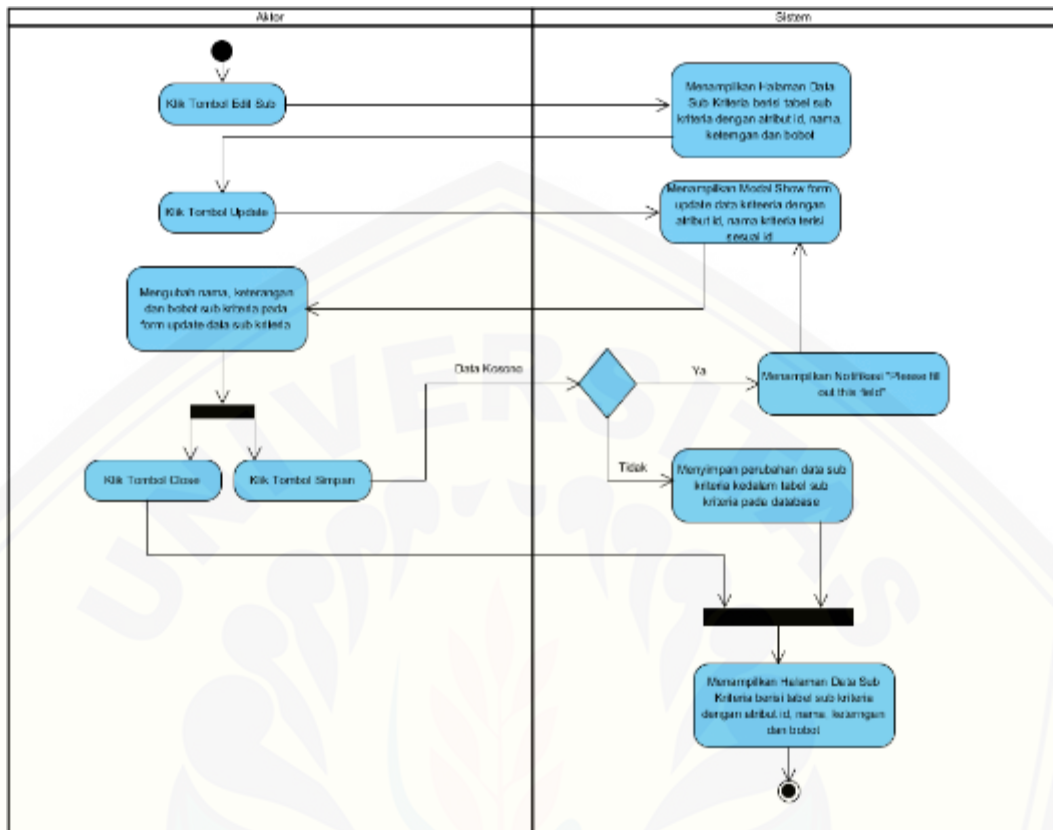
B5. Lampiran Activity Diagram Mengelola Data Kriteria



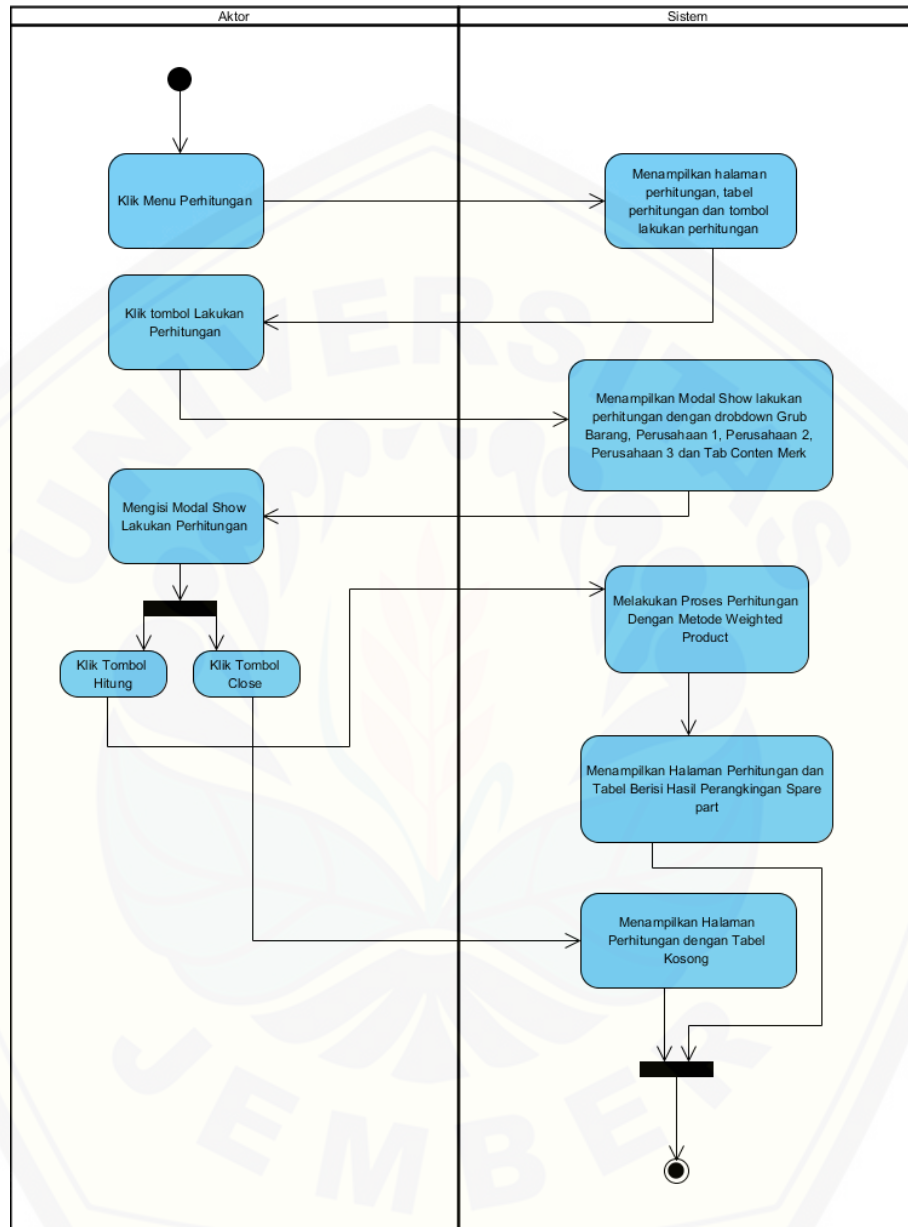
B6. Lampiran Activity Diagram Mengelola Data Sub Kriteria



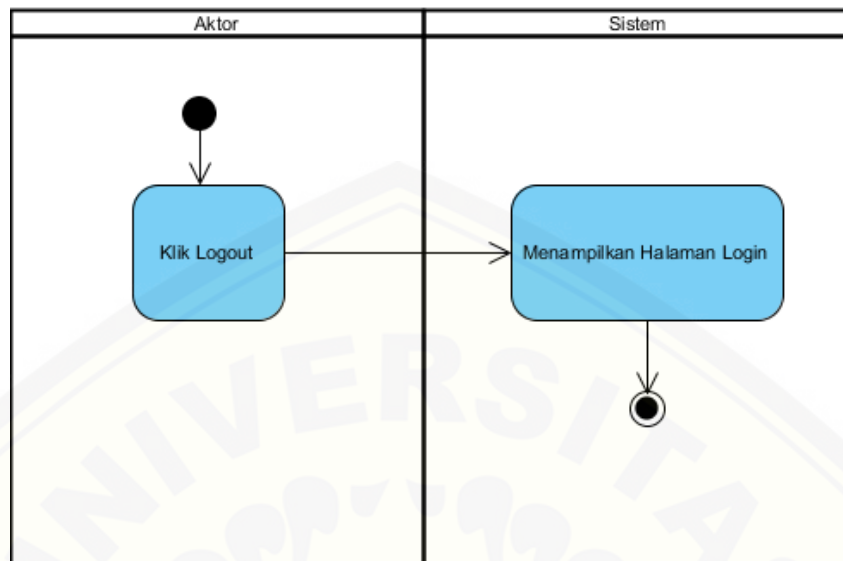
B7. Lampiran Activity Diagram Mengelola Data Bobot Kriteria



B8. Lampiran *Activity Diagram* Mengelola Data Rekomendasi Perhitungan (Karyawan)

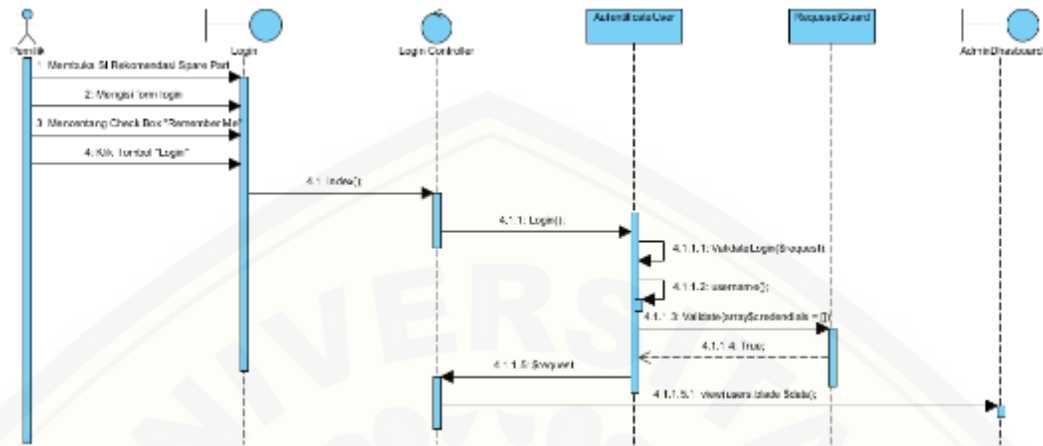


B9. Lampiran *Activity Diagram Logout*

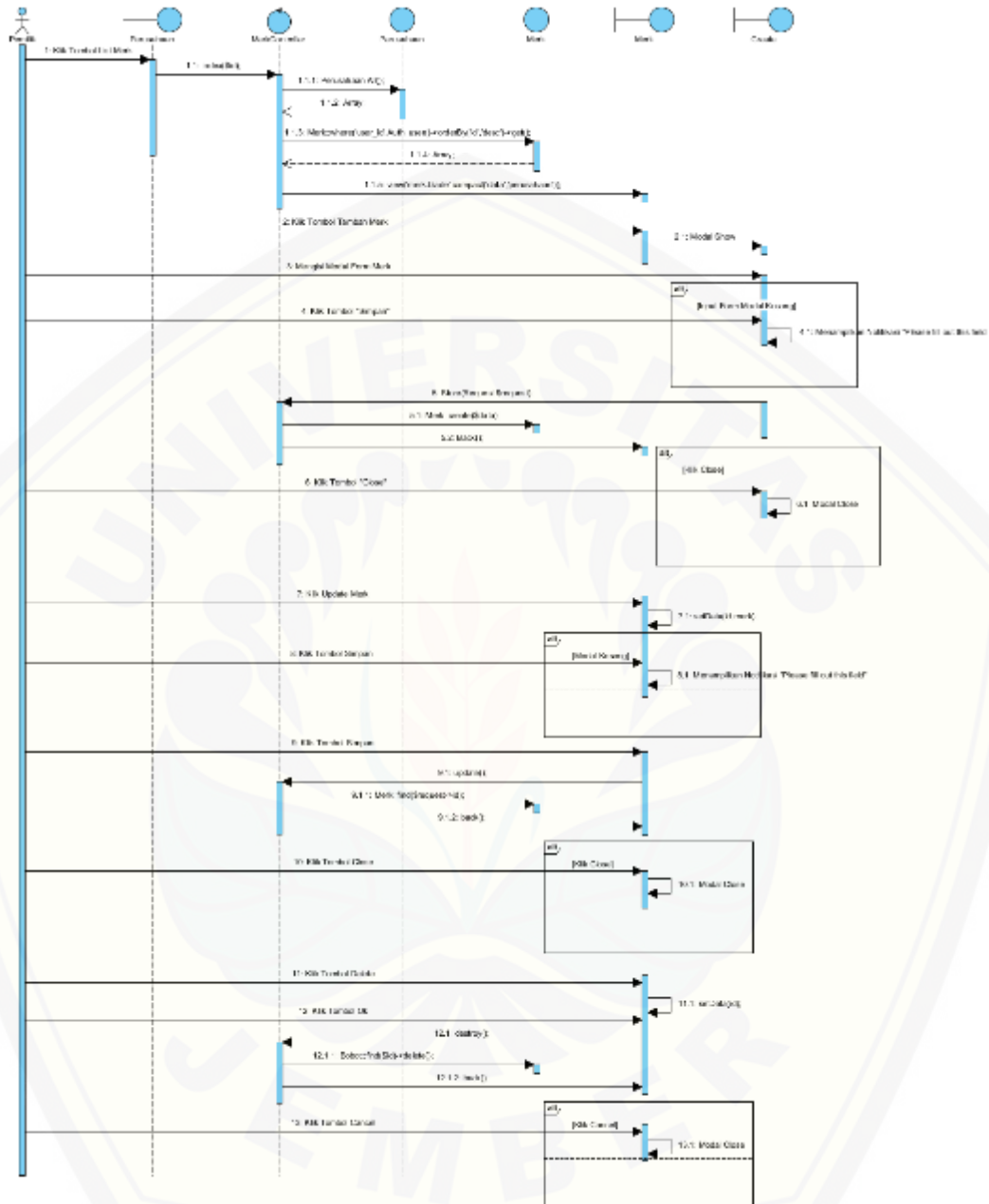


C. Lampiran Sequence Diagram

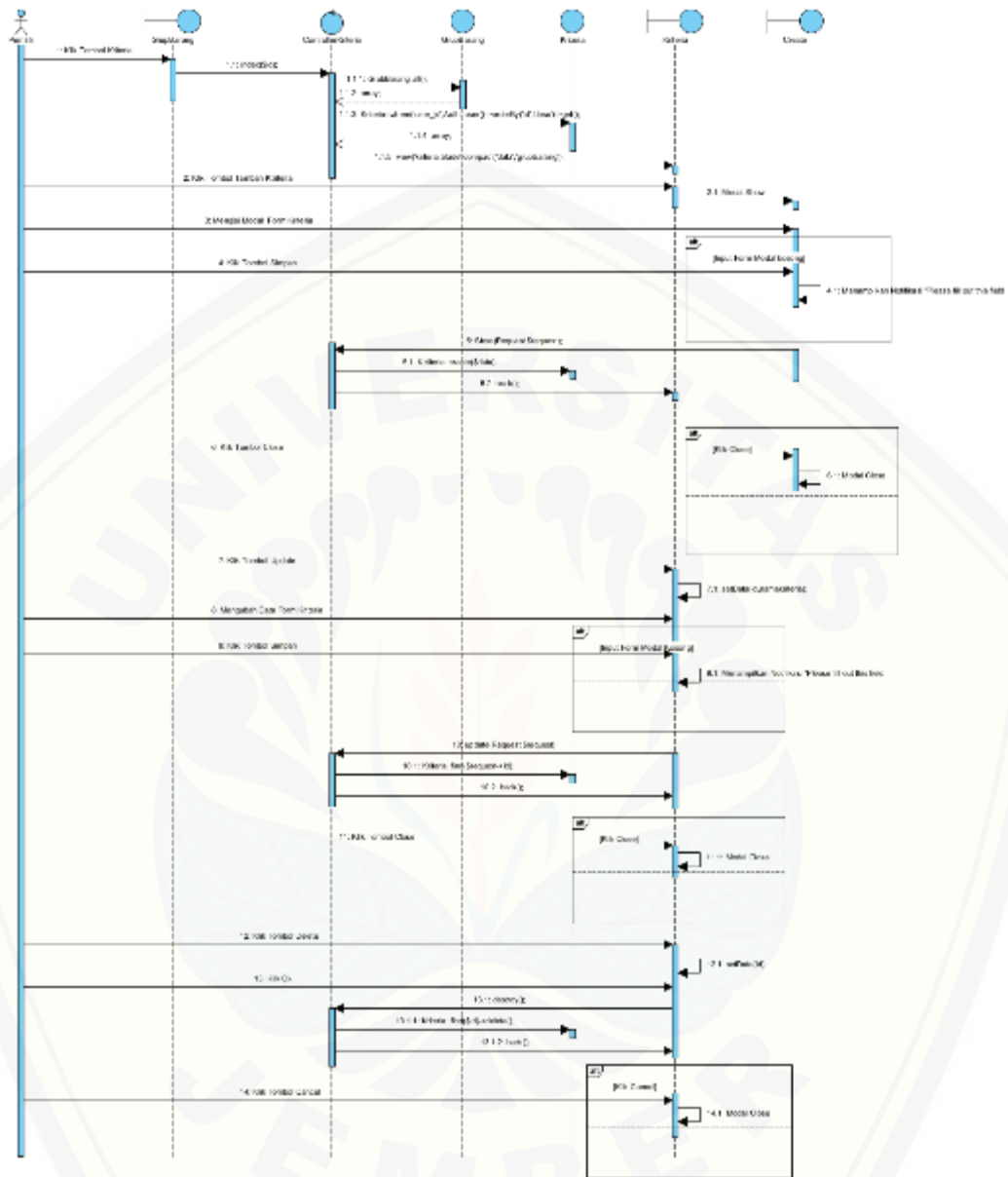
C1. Lampiran Sequence Diagram Login



C2. Lampiran Sequence Diagram Mengelola Data Merk



C5. Lampiran Sequence Diagram Mengelola Data Kriteria



D. Lampiran Kode Program**D1. Lampiran Kode Program Fitur Mengelola Data Perusahaan****Kode Program *Controller* PerusahaanController.php**

```
<?php

namespace App\Http\Controllers;

use App\Perusahaan;
use Illuminate\Http\Request;

class PerusahaanController extends Controller
{
    public function index()
    {
        $data = Perusahaan::all();
        return view('admin.Perusahaan', compact('data'));
    }

    public function store(Request $request)
    {
        $request->offsetUnset('_token');
        $data = $request->all();
        Perusahaan::create($data);
        return back();
    }

    public function update(Request $request)
    {
        $request->offsetUnset('_token');
        $data = $request->all();
        $dat = Perusahaan::find($request->id);
        $dat->update($data);
        return back();
    }

    public function destroy($id)
```

```
{
    $data = Perusahaan::find($id)->delete();
    return back();
}
}
```

Kode Program Model Perusahaan.php

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Perusahaan extends Model
{
    protected $table = "perusahaan";
    public $timestamps = false;
    protected $fillable = ['perusahaan'];

    public function merks(){
        return $this->hasMany('App\Merk');
    }
}
```

Kode Program View perusahaan.blade.php

```
@extends('layouts.app2')

@section('content')
<div class="container">
    <div class="row">
        <div class="col-md-11">
            <div class="panel panel-default">
```

```
<div class="panel-heading text-center">Perusahaan
    <button type="button"
class="btn-sm btn-primary pull-right" data-
toggle="modal" data-target="#create">Tambah
Perusahaan</button>
</div>

<div class="panel-body">
    @if (session('status'))
    <div class="alert alert-
success">
        {{ session('status') }}
    </div>
    @endif
    <table class="table table-
responsive">
        <thead>
            <tr>
                <th>ID</th>
                <th>Perusahaan</th>
                <th
style="width:300px;">Aksi</th>
            </tr>
        </thead>
        <tbody>
            @foreach($data as
$row)
                <tr>
```

```

<td>{{ $row-
>id}}</td>
<td>{{ $row-
>perusahaan}}</td>
<td>
<a
href="{{ route('Merk', ['id' => $row->id]) }}"><button
type="button" class="btn-sm btn-info">List
Merk</button></a>
<button
type="button" class="btn-sm btn-warning" data-
toggle="modal" data-target="#update"
onClick='setData({{ $row->id}}, "{{ $row-
>perusahaan}}");'>Update</button>
<a
onclick="return confirm('Apakah Anda Yakin ingin
Menghapus');" href="{{ route('deletePerusahaan', ['id'
=> $row->id]) }}"><button type="button" class="btn-sm
btn-danger">Delete</button></a>
</td>
</tr>
@endforeach
</tbody>
</table>
</div>
</div>
</div>
</div>
<div id="create" class="modal fade" role="dialog">
<div class="modal-dialog">

```

```
<div class="modal-content">
  <div class="modal-header">
    <button type="button"
class="close" data-dismiss="modal">&times;</button>
    <h4 class="modal-title">Tambah
Data</h4>
  </div>
  <form
action="{{route('createPerusahaan')}}" method="post"
accept-charset="utf-8">
    {{csrf_field()}}
    <div class="modal-body">
      <label>Nama
Perusahaan</label>
      <input type="text"
name="perusahaan" value="" placeholder="nama
Perusahaan" class="form-control" required>
    </div>
    <div class="modal-footer">
      <button class="btn-sm btn-
primary" type='submit'>Simpan</button>
      <button type="button"
class="btn btn-default pull-left" data-
dismiss="modal">Close</button>
    </div>
  </form>
</div>
</div>
</div>
<div id="update" class="modal fade" role="dialog">
```

```
<div class="modal-dialog">
  <div class="modal-content">
    <div class="modal-header">
      <button type="button"
class="close" data-dismiss="modal">&times;</button>
      <h4 class="modal-title">Update
Data</h4>
    </div>
    <form
action="{{route('updatePerusahaan')}}" method="post"
accept-charset="utf-8">
      {{csrf_field()}}
      <div class="modal-body">
        <input class="hidden"
name="id" type="hidden" id="id"></input>
        <label>Nama
Perusahaan</label>
        <input type="text"
id="perusahaan" name="perusahaan" value=""
placeholder="nama Perusahaan" class="form-control"
required>
      </div>
      <div class="modal-footer">
        <button class="btn-sm btn-
primary" type='submit'>Simpan</button>
        <button type="button"
class="btn btn-default pull-left" data-
dismiss="modal">Close</button>
      </div>
    </form>
  </div>
</div>
```

```
</div>
</div>
<script>
    function setData(id,perusahaan){
        // console.log(id+Perusahaan+nilai);
        $('#id').val(id);
        $('#perusahaan').val(perusahaan);
        // $('#nilai').val(nilai);
    }
</script>
@endsection
```

D2. Lampiran Kode Program Fitur Mengelola Data *Merk*

Kode Program *Controller* MerkController.php

```
<?php

namespace App\Http\Controllers;

use App\Merk;
use Illuminate\Http\Request;

class MerkController extends Controller
{

    public function index($id)
    {
        $data = Merk::where('perusahaan_id',$id)->get();
        return view('admin.Merk',compact('data','id'));
    }

    public function index2($id)
    {
        $data = Merk::where('perusahaan_id',$id)->get();
        echo json_encode($data);
    }
}
```

```
}

public function store(Request $request)
{
    $request->offsetUnset('_token');
    $data = $request->all();
    Merk::create($data);
    return back();
}

public function update(Request $request)
{
    $request->offsetUnset('_token');
    $data = $request->all();
    $dat = Merk::find($request->id);
    $dat->update($data);
    return back();
}

public function destroy($id)
{
    $data = Merk::find($id)->delete();
    return back();
}
}
```

Kode Program *Model* Merk.php

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Merk extends Model
```



```
{
    protected $table = "merk";
    public $timestamps = false;
    protected $fillable = ['merk','perusahaan_id'];
}
```

Kode Program View merk.blade.php

```
@extends('layouts.app2')

@section('content')
<div class="container">
    <div class="row">
        <div class="col-md-11">
            <div class="panel panel-default">
                <div class="panel-heading text-
center">Merk
                    <button type="button"
class="btn-sm btn-primary pull-right" data-
toggle="modal" data-target="#create">Tambah
Merk</button>
                    <a
href="{{url('Perusahaan')}}" class="btn-sm btn-danger
pull-left">Kembali</a>
                </div>
                <div class="panel-body">
                    @if (session('status'))
                    <div class="alert alert-
success">
                        {{ session('status') }}
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
```

```
@endif
<table class="table table-
responsive">
    <thead>
        <tr>
            <th>ID</th>
            <th>Merk</th>
            <th
style="width:300px;">Aksi</th>
        </tr>
    </thead>
    <tbody>
        @foreach($data as
$row)
            <tr>
                <td>{{$row-
>id}}</td>
                <td>{{$row-
>merk}}</td>
                <td>
                    <a
href="{{route('Sparepart',['id' => $row->id,'id2' =>
$id])}}"><button type="button" class="btn-sm btn-
info">List Sparepart</button></a>
                    <button
type="button" class="btn-sm btn-warning" data-
toggle="modal" data-target="#update"
onClick='setData({{$row->id}},"{{$row-
>merk}}");'>Update</button>
                    <a
onclick="return confirm('Apakah Anda Yakin ingin
```

```
Menghapus');" href="{{route('deleteMerk', ['id' =>
$row->id])}}"><button type="button" class="btn-sm
btn-danger">Delete</button></a>
</td>
</tr>
@endforeach
</tbody>
</table>
</div>
</div>
</div>
</div>
<div id="create" class="modal fade" role="dialog">
<div class="modal-dialog">
<div class="modal-content">
<div class="modal-header">
<button type="button"
class="close" data-dismiss="modal">&times;</button>
<h4 class="modal-title">Tambah
Data</h4>
</div>
<form action="{{route('createMerk')}}"
method="post" accept-charset="utf-8">
{{csrf_field()}}
<div class="modal-body">
<label>Nama Merk</label>
<input type="text"
name="merk" value="" placeholder="nama Merk"
class="form-control" required>
```



```
                <input type="text" id="merk"
name="merk" value="" placeholder="nama Merk"
class="form-control" required>
            </div>
            <div class="modal-footer">
                <button class="btn-sm btn-
primary" type='submit'>Simpan</button>
                <button type="button"
class="btn btn-default pull-left" data-
dismiss="modal">Close</button>
            </div>
        </form>
    </div>
</div>
</div>
<script>
    function setData(id,merk){
        // console.log(id+Merk+nilai);
        $('#id').val(id);
        $('#merk').val(merk);
        // $('#nilai').val(nilai);
    }
</script>
@endsection
```

D3. Lampiran Kode Program Fitur Mengelola Data *Spare Part*

Kode Program *Controller SparepartController.php*

```
<?php

namespace App\Http\Controllers;
```

```
use App\Sparepart;
use App\GrupBarang;
use Illuminate\Http\Request;

class SparepartController extends Controller
{
    public function index($id,$id2)
    {
        $data = Sparepart::where('merk_id',$id)->get();
        $grupBarang = GrupBarang::all();
        return
view('admin.Sparepart',compact('data','id','id2','grupBarang'));
    }

    public function store(Request $request)
    {
        $request->offsetUnset('_token');
        $data = $request->all();
        Sparepart::create($data);
        return back();
    }

    public function update(Request $request)
    {
        $request->offsetUnset('_token');
        $data = $request->all();
        $dat = Sparepart::find($request->id);
        $dat->update($data);
        return back();
    }

    public function destroy($id)
    {
        $data = Sparepart::find($id)->delete();
        return back();
    }
}
```

Kode Program *Model* Sparepart.php

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Sparepart extends Model
{
    protected $table = "sparepart";
    public $timestamps = false;
    protected $fillable =
['sparepart','merk_id','harga','garansi','ketahanan',
'grup_barang_id','Harga_id','Garansi_id','Ketahanan_i
d'];

    public function grupBarang(){
        return $this-
>belongsTo('App\GrupBarang','grup_barang_id','id');
    }
}
```

Kode Program *View* sparepart.blade.php

```
@extends('layouts.app2')

@section('content')
<div class="container">
    <div class="row">
        <div class="col-md-11">
            <div class="panel panel-default">
```



```

        <th>Ketahanan</th>
                                <th>Aksi</th>
                                </tr>
                                </thead>
                                <tbody>
                                @foreach($data as
$row)
                                <tr>
                                <td>{{ $row-
>id}}</td>
                                <td>{{ $row-
>grupBarang->nama}}</td>
                                <td>{{ $row-
>sparepart}}</td>
                                <td>{{ $row-
>harga}}</td>
                                <td>{{ $row-
>garansi}}</td>
                                <td>{{ $row-
>ketahanan}}</td>
                                <td>
                                <!--
                                <button type="button" class="btn-sm btn-warning"
                                data-toggle="modal" data-target="#update"
                                onClick='setData({{ $row->id}}, "{{ $row-
                                >sparepart}}", "{{ $row->harga}}", "{{ $row-
                                >garansi}}", "{{ $row->ketahanan}}");'>Update</button>
                                -->
                                <a
                                onclick="return confirm('Apakah Anda Yakin ingin

```

```
Menghapus');" href="{{route('deleteSparepart', ['id'
=> $row->id])}}"><button type="button" class="btn-sm
btn-danger">Delete</button></a>
</td>
</tr>
@endforeach
</tbody>
</table>
</div>
</div>
</div>
</div>
</div>
<div id="create" class="modal fade" role="dialog">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button"
class="close" data-dismiss="modal">&times;</button>
        <h4 class="modal-title">Tambah
Data</h4>
      </div>
      <form
action="{{route('createSparepart')}}" method="post"
accept-charset="utf-8">
        {{csrf_field()}}
        <div class="modal-body">
          <label>Nama
Sparepart</label>
```

```
        <input type="text"
name="sparepart" value="" placeholder="nama
sparepart" class="form-control" required>
        <label for="">Grup
Barang</label>
        <select
name="grup_barang_id" id="grupBarang" class="form-
control" onchange="getGrup();" required>
            <option value="">----
</option>
            @foreach($grupBarang as
$row)
            <option value="{{ $row-
>id }}">{{ $row->nama }}</option>
            @endforeach
        </select>
        <div id="pilihan">
</div>
        <label for="">Nilai
Harga</label>
        <input type="number"
name="harga" placeholder="Harga Dalam Rupiah"
class="form-control" required>
        <label for="">Nilai
Garansi</label>
        <input type="number"
name="garansi" placeholder="Garansi Dalam Bulan"
class="form-control" required>
        <label for="">Nilai
Ketahanan</label>
```

```

        <input type="number"
name="ketahanan" placeholder="Kapasitas Tahan
Sparepart" class="form-control" required>
        <input type="hidden"
name="merk_id" value="{{ $id }}">
    </div>
    <div class="modal-footer">
        <button class="btn-sm btn-
primary" type='submit'>Simpan</button>
        <button type="button"
class="btn btn-default pull-left" data-
dismiss="modal">Close</button>
    </div>
</form>
</div>
</div>
</div>
</div>
<div id="update" class="modal fade" role="dialog">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <button type="button"
class="close" data-dismiss="modal">&times;</button>
                <h4 class="modal-title">Update
Data</h4>
            </div>
            <form
action="{{ route('updateSparepart') }}" method="post"
accept-charset="utf-8">
                {{ csrf_field() }}

```

```
        <div class="modal-body">
            <input class="hidden"
name="id" type="hidden" id="id"></input>
            <label>Nama
Sparepart</label>
            <input type="text"
id="sparepart" name="sparepart" value=""
placeholder="nama Sparepart" class="form-control"
required>
            <label for="">Nilai
Harga</label>
            <input type="number"
id="harga" name="harga" placeholder="Harga Dalam
Rupiah" class="form-control" required>
            <label for="">Nilai
Garansi</label>
            <input type="number"
id="garansi" name="garansi" placeholder="Garansi
Dalam Bulan" class="form-control" required>
            <label for="">Nilai
Ketahanan</label>
            <input type="number"
id="ketahanan" name="ketahanan"
placeholder="Kapasitas Tahan Sparepart" class="form-
control" required>
        </div>
        <div class="modal-footer">
            <button class="btn-sm btn-
primary" type='submit'>Simpan</button>
```

```
                <button type="button"
class="btn btn-default pull-left" data-
dismiss="modal">Close</button>
            </div>
        </form>
    </div>
</div>
<script>
    function
setData(id,sparepart,harga,garansi,ketahanan) {
    // console.log(id+Sparepart+nilai);
    $('#id').val(id);
    $('#sparepart').val(sparepart);
    $('#harga').val(harga);
    $('#garansi').val(garansi);
    $('#ketahanan').val(ketahanan);

    // $('#nilai').val(nilai);
}
getGrup();

function getGrup() {
    var dt = $('#grupBarang').val();
    url = '{{url("Api/Kriteria")}}/'+dt;
    // console.log(url);
    $.getJSON(url,function(data,success) {
        var txt = "";
        $.each(data,function(i,item) {
            txt +=
"<label>"+item.kriteria+"</label>";
```

```
        txt += "<select
name='"+item.kriteria+"_id' class='form-control'>";
        txt += "<option value=''>---
</option>";

        $.each(item.sub,function(j,it){
            // console.log(it);
            txt += "<option
value='"+it.id+"'>"+it.sub_kriteria+"</option>";
        });
        txt += "</select>"
    });
    $('#pilihan').empty();
    $('#pilihan').append(txt);
});
}
</script>
@endsection
```

D4. Lampiran Kode Program Fitur Mengelola Data Grup Barang

Kode Program *Controller* GrupBarangController.php

```
<?php
namespace App\Http\Controllers;

use App\GrupBarang;
use Illuminate\Http\Request;

class GrupBarangController extends Controller
{
    public function index()
    {
        $data = GrupBarang::all();
```

```
        return view('admin.grupBarang', compact('data'));
    }

    public function store(Request $request)
    {
        $request->offsetUnset('_token');
        $data = $request->all();
        GrupBarang::create($data);
        return back();
    }

    public function update(Request $request)
    {
        $request->offsetUnset('_token');
        $data = $request->all();
        // dd($data);
        $dat = GrupBarang::find($request->id);
        $dat->update($data);
        return back();
    }

    public function destroy($id)
    {
        $dat = GrupBarang::find($id);
        $dat->delete();
        return back();
    }
}
```

Kode Program *Model* GrupBarang.php

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;
```



```
class GrupBarang extends Model
{
    protected $table = "grup_barang";
    protected $fillable = ['id','nama'];
    public $timestamps = false;
}
```

Kode Program View grupBarang.blade.php

```
@extends('layouts.app2')

@section('content')
<div class="container">
    <div class="row">
        <div class="col-md-11">
            <div class="panel panel-default">
                <div class="panel-heading text-
center">Grup Barang
                    <button type="button"
class="btn-sm btn-primary pull-right" data-
toggle="modal" data-target="#create">Tambah Grup
Barang</button>
                </div>
                <div class="panel-body">
                    @if (session('status'))
                        <div class="alert alert-
success">
                            {{ session('status') }}
                        </div>
                    @endif
                </div>
            </div>
        </div>
    </div>
</div>
```

```
@endif
<table class="table table-
responsive">
    <thead>
        <tr>
            <th>ID</th>
            <th>Grup
Barang</th>
            <th
style="width:300px;">Aksi</th>
        </tr>
    </thead>
    <tbody>
        @foreach($data as
$row)
            <tr>
                <td>{{$row-
>id}}</td>
                <td>{{$row-
>nama}}</td>
                <td>
                    <button
type="button" class="btn-sm btn-warning" data-
toggle="modal" data-target="#update"
onClick='setData({{$row->id}}, "{{$row-
>nama}}");'>Update</button>
                    <a
onclick="return confirm('Apakah Anda Yakin ingin
Menghapus');" href="{{route('deleteGrupBarang', ['id'
=> $row->id])}}"><button type="button" class="btn-sm
btn-danger">Delete</button></a>

```

```
<a
href="{{route('Kriteria',['id' => $row-
>id])}}"><button type="button" class="btn-sm btn-
info">Kriteria</button></a>
</td>
</tr>
@endforeach
</tbody>
</table>
</div>
</div>
</div>
</div>
</div>
<div id="create" class="modal fade" role="dialog">
<div class="modal-dialog">
<div class="modal-content">
<div class="modal-header">
<button type="button"
class="close" data-dismiss="modal">&times;</button>
<h4 class="modal-title">Tambah
Data</h4>
</div>
<form
action="{{route('createGrupBarang')}}" method="post"
accept-charset="utf-8">
{{csrf_field()}}
<div class="modal-body">
<label>Nama Grup
Barang</label>
```

```
        <input type="text"
name="nama" value="" placeholder="nama Grup Barang"
class="form-control" required>
    </div>
    <div class="modal-footer">
        <button class="btn-sm btn-
primary" type='submit'>Simpan</button>
        <button type="button"
class="btn btn-default pull-left" data-
dismiss="modal">Close</button>
    </div>
</form>
</div>
</div>
</div>
</div>
<div id="update" class="modal fade" role="dialog">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <button type="button"
class="close" data-dismiss="modal">&times;</button>
                <h4 class="modal-title">Update
Data</h4>
            </div>
            <form
action="{{route('updateGrupBarang')}}" method="post"
accept-charset="utf-8">
                {{csrf_field()}}
                <div class="modal-body">
```

```
        <input class="hidden"
name="id" type="hidden" id="id"></input>
        <label>Nama Grup
Barang</label>
        <input type="text" id="nama"
name="nama" value="" placeholder="nama Grup Barang"
class="form-control" required>
    </div>
    <div class="modal-footer">
        <button class="btn-sm btn-
primary" type='submit'>Simpan</button>
        <button type="button"
class="btn btn-default pull-left" data-
dismiss="modal">Close</button>
    </div>
</form>
</div>
</div>
</div>
<script>
    function setData(id,nama){
        // console.log(id+Grup Barang+nilai);
        $('#id').val(id);
        $('#nama').val(nama);
        // $('#nilai').val(nilai);
    }
</script>
@endsection
```

D5. Lampiran Kode Program Fitur Mengelola Data Kriteria

Kode Program *Controller* KriteriaController.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Kriteria;

use App\GrupBarang;

class KriteriaController extends Controller
{
    public function index($id)
    {
        $data = Kriteria::where('grup_barang_id',$id)->get();
        $grupBarang= GrupBarang::find($id);
        return
        view('admin.kriteria',compact('data','id','grupBarang'));
    }

    public function kriteria($id){
        $data[] = array();
        $i = 0;
        $d = Kriteria::where('grup_barang_id',$id)->get();
        foreach ($d as $key => $value) {
            // dd($value->sub());
            $data[$i]['id'] = $value->id;
            $data[$i]['kriteria'] = $value->kriteria;
            $data[$i++]['sub'] = $value->sub;
        }
        echo json_encode($data);
    }

    public function store(Request $request)
```

```
{
    $request->offsetUnset('_token');
    $data = $request->all();
    Kriteria::create($data);
    return back();
}

public function update(Request $request)
{
    $request->offsetUnset('_token');
    $data = $request->all();
    $dat = Kriteria::find($request->id);
    $dat->update($data);
    return back();
}

public function destroy($id)
{
    $data = Kriteria::find($id)->delete();
    return back();
}
}
```

Kode Program *Model* Kriteria.php

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Kriteria extends Model
{
    protected $table = 'kriteria';
}
```

```

        protected $fillable =
        ['kriteria','nilai','grup_barang_id'];
        public $timestamps = false;

        public function sub(){
            return $this-
>hasMany('App\SubKriteria','kriteria_id');
        }
    }
}

```

Kode Program View kriteria.blade.php

```

@extends('layouts.app2')

@section('content')
<div class="container">
    <div class="row">
        <div class="col-md-11">
            <div class="panel panel-default">
                <div class="panel-heading text-
center">Kriteria dari Grup Barang "{{$grupBarang-
>nama}}"
                    <button type="button"
class="btn-sm btn-primary pull-right" data-
toggle="modal" data-target="#create">Tambah
Kriteria</button>
                    <a
href="{{route('GrupBarang')}}"><button type="button"
class="btn-sm btn-danger pull-
left">Kembali</button></a>
                </div>
            </div>
        </div>
    </div>
</div>

```



```
<div class="panel-body">
    @if (session('status'))
    <div class="alert alert-
success">
        {{ session('status') }}
    </div>
    @endif
    <table class="table table-
responsive">
        <thead>
            <tr>
                <th>ID</th>
                <th>Kriteria</th>
                <!--
<th>Nilai</th> -->
                <th>Keterangan</th>
                <th
style="width:300px;">Aksi</th>
            </tr>
        </thead>
        <tbody>
            @foreach($data as
$row)
                <tr>
                    <td>{{ $row-
>id}}</td>
                    <td>{{ $row-
>kriteria}}</td>
```

```

<!--
<td>{{ $row->nilai}}</td> -->

<td>

    @foreach($row->sub as $r)

        {{ $r-
>sub_kriteria}} : {{ $r->nilai}}<br>

    @endforeach

</td>
<td>
<a
href="{{ route('SubKriteria', ['id' => $row->id, 'id2'
=> $id]) }}"><button type="button" class="btn-sm btn-
info">Edit Sub</button></a>

<button
type="button" class="btn-sm btn-warning" data-
toggle="modal" data-target="#update"
onClick='setData({{ $row->id}}, "{{ $row-
>kriteria}}", {{ $row->nilai}});'>Update</button>

<a
onclick="return confirm('Apakah Anda Yakin ingin
Menghapus');" href="{{ route('deleteKriteria', ['id'
=> $row->id]) }}"><button type="button" class="btn-sm
btn-danger">Delete</button></a>

</td>
</tr>

@endforeach

</tbody>
</table>
</div>

```

```
</div>
</div>
</div>
</div>
<div id="create" class="modal fade" role="dialog">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button"
class="close" data-dismiss="modal">&times;</button>
        <h4 class="modal-title">Tambah
Data</h4>
      </div>
      <form
action="{{route('createKriteria')}}" method="post"
accept-charset="utf-8">
        {{csrf_field()}}
        <div class="modal-body">
          <label>Nama Kriteria</label>
          <input type="text"
name="kriteria" value="" placeholder="nama kriteria"
class="form-control">
          <input type="hidden"
name="nilai" value="0" placeholder="Nilai Kriteria"
class="form-control">
          <input type="hidden"
name="grup_barang_id" value="{{ $id }}">
        </div>
        <div class="modal-footer">
          <button class="btn-sm btn-
primary" type='submit'>Simpan</button>

```

```
                <button type="button"
class="btn btn-default pull-left" data-
dismiss="modal">Close</button>
            </div>
        </form>
    </div>
</div>
</div>
</div>
<div id="update" class="modal fade" role="dialog">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <button type="button"
class="close" data-dismiss="modal">&times;</button>
                <h4 class="modal-title">Update
Data</h4>
            </div>
            <form
action="{route('updateKriteria')}" method="post"
accept-charset="utf-8">
                {{csrf_field()}}
                <div class="modal-body">
                    <input class="hidden"
name="id" type="hidden" id="id"></input>
                    <label>Nama Kriteria</label>
                    <input type="text"
id="kriteria" name="kriteria" value=""
placeholder="nama kriteria" class="form-control">
                    <!--
<label>Bobot/Nilai</label> -->
```



```
use Illuminate\Http\Request;

use App\SubKriteria;
use App\Kriteria;
class SubKriteriaController extends Controller
{

    public function index($id,$id2)
    {
        $data = SubKriteria::where('kriteria_id','=', $id)-
>get();
        $kriteria = Kriteria::find($id);
        return
view('admin.subkriteria',compact('data','id','id2','kriteria'));
    }

    public function store(Request $request)
    {
        $request->offsetUnset('_token');
        $data = $request->all();
        SubKriteria::create($data);
        return back();
    }

    public function update(Request $request)
    {
        $request->offsetUnset('_token');
        $data = SubKriteria::find($request->id);
        $data->update($request->all());
        return back();
    }

    public function destroy($id)
    {
        $data = SubKriteria::find($id)->delete();
        return back();
    }
}
```

```
}  
}
```

Kode Program *Model* SubKriteria.php

```
<?php  
  
namespace App;  
  
use Illuminate\Database\Eloquent\Model;  
  
class SubKriteria extends Model  
{  
    protected $table = 'sub_kriteria';  
    public $timestamps = false;  
    protected $fillable =  
['sub_kriteria', 'nilai', 'kriteria_id', 'keterangan'];  
  
    public function krit(){  
        return $this-  
>belongsTo('App\ViewKriteria', 'kriteria_id', 'id');  
    }  
}
```

Kode Program *View* subkriteria.blade.php

```
@extends('layouts.app2')  
  
@section('content')  
<div class="container">  
    <div class="row">  
        <div class="col-md-11">
```

```
<div class="panel panel-default">
  <div class="panel-heading text-center">Sub Kriteria "{{ $kriteria->kriteria }}"
    <button type="button"
class="btn-sm btn-primary pull-right" data-
toggle="modal" data-target="#create">Tambah Sub
Kriteria</button>
    <a
href="{{ route('Kriteria', ['id' => $id2]) }}"><button
type="button" class="btn-sm btn-danger pull-
left">Kembali</button></a>
  </div>

  <div class="panel-body">
    @if (session('status'))
      <div class="alert alert-
success">
        {{ session('status') }}
      </div>
    @endif
    <table class="table table-
responsive">
      <thead>
        <tr>
          <th>ID</th>

          <th>Kriteria</th>

          <th>Bobot</th>

          <th>Keterangan</th>
```



```

                <th>Aksi</th>
            </tr>
        </thead>
        <tbody>
            @foreach($data as
$row)
                <tr>
                    <td>{{ $row-
>id}}</td>
                    <td>{{ $row-
>sub_kriteria}}</td>
                    <td>{{ $row-
>nilai}}</td>
                    <td>{{ $row-
>keterangan}}</td>
                    <td><button
type="button" class="btn-sm btn-warning" data-
toggle="modal" data-target="#update"
onClick='setData({{ $row->id}}, "{{ $row-
>sub_kriteria}}", {{ $row->nilai}}, "{{ $row-
>keterangan}}");'>Update</button>
                    <a
onclick="return confirm('Apakah Anda Yakin ingin
Menghapus');"
href="{{ route('deleteSubKriteria', ['id' => $row-
>id]) }}"><button type="button" class="btn-sm btn-
danger">Delete</button></a>
                </td>
            </tr>
        @endforeach
    </tbody>

```

```
        </table>
    </div>
</div>
</div>
</div>
</div>
<div id="create" class="modal fade" role="dialog">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <button type="button"
class="close" data-dismiss="modal">&times;</button>
                <h4 class="modal-title">Tambah
Data</h4>
            </div>
            <form
action="{{route('createSubKriteria')}}" method="post"
accept-charset="utf-8">
                {{csrf_field()}}
                <div class="modal-body">
                    <input type="hidden"
name="kriteria_id" value="{{ $id }}" placeholder=""
required>
                    <label>Nama Kriteria</label>
                    <input type="text"
name="sub_kriteria" value="" placeholder="Nama
SubKriteria" class="form-control" required>
                    <label>Bobot/Nilai</label>
                    <input type="text"
name="nilai" value="" placeholder="Nilai SubKriteria"
class="form-control" required>
```

```
        <label>Keterangan</label>
        <textarea name="keterangan"
class="form-control"></textarea>
    </div>
    <div class="modal-footer">
        <button class="btn-sm btn-
primary" type='submit'>Simpan</button>
        <button type="button"
class="btn btn-default pull-left" data-
dismiss="modal">Close</button>
    </div>
</form>
</div>
</div>
</div>
</div>
<div id="update" class="modal fade" role="dialog">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <button type="button"
class="close" data-dismiss="modal">&times;</button>
                <h4 class="modal-title">Update
Data</h4>
            </div>
            <form
action="{{route('updateSubKriteria')}}" method="post"
accept-charset="utf-8">
                {{csrf_field()}}
                <div class="modal-body">
```



```
        $('#nilai').val(nilai);  
        $('#keterangan').val(keterangan);  
    }  
</script>  
@endsection
```

D7. Lampiran Kode Program Fitur Mengelola Data Bobot Kriteria

Kode Program *Controller* BobotController.php

```
<?php  
  
namespace App\Http\Controllers;  
  
use App\Bobot;  
use Illuminate\Http\Request;  
  
class BobotController extends Controller  
{  
    public function index()  
    {  
        $data = Bobot::all();  
        return view('admin.Bobot', compact('data'));  
    }  
  
    public function store(Request $request)  
    {  
        $request->offsetUnset('_token');  
        $data = $request->all();  
        Bobot::create($data);  
        return back();  
    }  
  
    public function update(Request $request)  
    {  
        $request->offsetUnset('_token');  
        $data = $request->all();
```

```
        $dat = Bobot::find($request->id);
        $dat->update($data);
        return back();
    }

    public function destroy($id)
    {
        $data = Bobot::find($id)->delete();
        return back();
    }
}
```

Kode Program *Model* Bobot.php

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Bobot extends Model
{
    protected $table = "bobot";
    protected $fillable =
    ['nama', 'keterangan', 'bobot'];
    public $timestamps = false;
}
```

Kode Program *View* bobot.blade.php

```
@extends('layouts.app2')

@section('content')
<div class="container">
```

```
<div class="row">
  <div class="col-md-11">
    <div class="panel panel-default">
      <div class="panel-heading text-center">Data Kriteria Pembobotan
        <!-- <button type="button"
class="btn-sm btn-primary pull-right" data-
toggle="modal" data-target="#create">Tambah Data
Bobot</button> -->
      </div>
      <div class="panel-body">
        @if (session('status'))
          <div class="alert alert-
success">
            {{ session('status') }}
          </div>
        @endif
        <table class="table table-
responsive">
          <thead>
            <tr>
              <th>ID</th>
              <th>Nama</th>
              <th>Keterangan</th>
              <th>Bobot</th>
              <th>Aksi</th>
            </tr>
          </thead>
```



```
<div class="modal-content">
  <div class="modal-header">
    <button type="button"
class="close" data-dismiss="modal">&times;</button>
    <h4 class="modal-title">Update
Data</h4>
  </div>
  <form
action="{{route('updateBobot')}}" method="post"
accept-charset="utf-8">
    {{csrf_field()}}
    <div class="modal-body">
      <input class="hidden"
name="id" type="hidden" id="id" required>
      <label>Bobot</label>
      <input type="text"
id="perusahaan" name="bobot" value=""
placeholder="Nilai Bobot" class="form-control"
required>
    </div>
    <div class="modal-footer">
      <button class="btn-sm btn-
primary" type='submit'>Simpan</button>
      <button type="button"
class="btn btn-default pull-left" data-
dismiss="modal">Close</button>
    </div>
  </form>
</div>
</div>
```

```
<script>
    function setData(id,perusahaan){
        // console.log(id+Perusahaan+nilai);
        $('#id').val(id);
        $('#perusahaan').val(perusahaan);
        // $('#nilai').val(nilai);
    }
</script>
@endsection
```

D8. Lampiran Kode Program Mengelola Rekomendasi Pemilihan

Kode Program *View penjualan.blade.php* (Bagian Pemilik)

```
@extends('layouts.app2')

@section('content')
<div class="container" style="overflow-x:auto">
    <div class="row">
        <div class="col-md-10">
            <div class="panel panel-default">
                <div class="panel-heading text-center">PERHITUNGAN
                    <button type="button" class="btn-sm btn-primary pull-right" data-toggle="modal" data-target="#creates">Lakukan Perhitungan</button>
                </div>

                <div class="panel-body">
                    <table class="table table-responsive">
                        <thead
                            style="color:white;background-color:#f0f0f0;">
```

```

                <th colspan="4"
class="text-center" style="border-right:solid 1px black;border-
left:solid 1px black;border-top:solid 1px black;border-
bottom:solid 1px black;">Keterangan</th>
                <th colspan="6"
class="text-center" style="border-right:solid 1px black;border-
top:solid 1px black;border-bottom:solid 1px black;">Menghitung
Nilai Preferensi</th>
                <th rowspan="2"
style="border-right:solid 1px black;border-top:solid 1px
black;">Hasil Perkalian</th>
                <th rowspan="2"
style="border-right:solid 1px black;border-top:solid 1px
black;">Total</th>
                <th rowspan="2"
style="border-right:solid 1px black;border-top:solid 1px
black;">Rangking</th>
            </thead>
            <thead
style="color:white;background-color:#f0f0f0;">
                <th style="border-
left:solid 1px black;border-bottom:solid 1px black;">ID</th>
                <th style="border-
bottom:solid 1px black;">Perusahaan</th>
                <th style="border-
bottom:solid 1px black;">Merk</th>
                <th style="border-
right:solid 1px black;border-bottom:solid 1px black;">Spare
Part</th>
                <th style="border-
bottom:solid 1px black;">S1</th>
                <th style="border-
bottom:solid 1px black;">S2</th>
                <th style="border-
bottom:solid 1px black;">S3</th>
                <th style="border-
bottom:solid 1px black;">S4</th>

```

```

<th style="border-
bottom:solid 1px black;">S5</th>
<th style="border-
right:solid 1px black;border-bottom:solid 1px black;">S6</th>
<th style="border-
right:solid 1px black;border-bottom:solid 1px black;"></th>
<th style="border-
right:solid 1px black;border-bottom:solid 1px black;"></th>
<th style="border-
right:solid 1px black;border-bottom:solid 1px black;"></th>
</thead>
<tbody>
<?php $i = 1; ?>
@foreach($data as
$row)
    @if($i <= 50)
    <tr>
        <td
style="border-left:solid 1px black;">{{$row['id']}}</td>
        <td>{{$row['per']}}</td>
        <td>{{$row['merk']}}</td>
        <td
style="border-right:solid 1px black;">{{$row['sp']}}</td>
        <td>{{$row['s1']}}</td>
        <td>{{$row['s2']}}</td>
        <td>{{$row['s3']}}</td>
        <td>{{$row['s4']}}</td>
        <td>{{$row['s5']}}</td>
        <td
style="border-right:solid 1px black;">{{$row['s6']}}</td>

```

```

                <td
style="border-right:solid 1px black;">{{$row['hPer']}}</td>
                <td
style="border-right:solid 1px black;">{{$row['tot']}}</td>
                <td
style="border-right:solid 1px black;">{{$i++}}</td>
            </tr>
        @endif
    @endforeach
</tbody>
</table>
</div>
</div>
</div>
</div>
</div>
<div id="creates" class="modal fade" role="dialog">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <button type="button" class="close"
data-dismiss="modal">&times;</button>
                <h4 class="modal-title">Masukkan
Kriteria</h4>
            </div>
            <form action="" method="get" accept-
charset="utf-8">
                <div class="modal-body">
                    <div class="row">
                        <div class="col-md-12">
                            <label for="">Grup
Barang</label>
                            <select
name="grup_barang" class="form-control" id="">
                                <option
value="">---</option>

```

```

        @foreach($grupBarang as $row)
            <option
value="{{ $row->id }}">{{ $row->nama }}</option>
            @endforeach
        </select>
    </div>
</div>
<br>
<br>
<div class="row">
    <div class="col-md-4">
        <label
for="">Perusahaan 1</label>
        <select name="s1[]"
id="vs1" class="form-control" onchange="setPerusahaan1();"
required>
            <option
value="">---</option>
            @foreach($perusahaan as $row)
                <option
value="{{ $row->id }}">{{ $row->perusahaan }}</option>
            @endforeach
        </select>
    </div>
    <div class="col-md-4">
        <label
for="">Perusahaan 2</label>
        <select name="s1[]"
id="vs2" class="form-control" onchange="setPerusahaan2();"
required>
            <option
value="">---</option>
            @foreach($perusahaan as $row)

```

```

                                <option
value="{{ $row->id }}">{{ $row->perusahaan }}</option>
                                @endforeach
                                </select>
                                </div>
                                <div class="col-md-4">
                                <label
for="">Perusahaan 3</label>
                                <select name="s1[]"
id="vs3" class="form-control" onchange="setPerusahaan3();"
required>
                                <option
value="">---</option>
                                @foreach($perusahaan as $row)
                                <option
value="{{ $row->id }}">{{ $row->perusahaan }}</option>
                                @endforeach
                                </select>
                                </div>
                                </div>
                                <br>
                                <br>
                                <div class="row">
                                <div class="col-md-12">
                                <label
for="">Merk</label>
                                <ul class="nav nav-
tabs">
                                <li><a data-
toggle="tab" href="#home1" id="h1"></a></li>
                                <li><a data-
toggle="tab" href="#home2" id="h2"></a></li>
                                <li><a data-
toggle="tab" href="#home3" id="h3"></a></li>
                                </ul>

```

```

<div class="tab-
content">
                                <div id="home1"
class="tab-pane fade in active">
                                </div>
                                <div id="home2"
class="tab-pane fade">
                                </div>
                                <div id="home3"
class="tab-pane fade">
                                </div>
                                </div>
                                </div>
                                </div>
                                <div class="modal-footer">
                                <button class="btn-sm btn-primary"
type='submit'>HITUNG</button>
                                <button type="button" class="btn
btn-default pull-left" data-dismiss="modal">Close</button>
                                </div>
                                </form>
                                </div>
                                </div>
</div>
<script>
function setPerusahaan1(){
    $('#h1').empty();
    $('#h1').append($('#vs1 option:selected').text());
    url = "{{url('Merk2') . '/'}}"+$('#vs1').val();
    $.getJSON(url, function(result){
        // console.log(result);
        text = "<div class='col-md-4'><select
class='form-control' name='merk[]' required>";

```



```
        $.each(result, function(i, field){
            text += "<option
value='"+field.id+"'>"+field.merk+"</option>"
        });
        text += "</select></div>";
        txt = "<div class='row'><div class='col-md-
12'>"+text+text+text+"</div></div>";
        $('#home1').empty();
        $('#home1').append(txt);
    });
}

function setPerusahaan2(){
    $('#h2').empty();
    $('#h2').append($('#vs2 option:selected').text());
    url = "{{url('Merk2').'/'}}"+$('#vs2').val();
    $.getJSON(url, function(result){
        // console.log(result);
        text = "<div class='col-md-4'><select
class='form-control' name='merk[]' required>";
        $.each(result, function(i, field){
            text += "<option
value='"+field.id+"'>"+field.merk+"</option>"
        });
        text += "</select></div>";
        txt = "<div class='row'><div class='col-md-
12'>"+text+text+text+"</div></div>";
        $('#home2').empty();
        $('#home2').append(txt);
    });
}

function setPerusahaan3(){
    $('#h3').empty();
    $('#h3').append($('#vs3 option:selected').text());
    url = "{{url('Merk2').'/'}}"+$('#vs3').val();
```

```
$.getJSON(url, function(result) {
    // console.log(result);
    text = "<div class='col-md-4'><select
class='form-control' name='merk[]' required>";
    $.each(result, function(i, field) {
        text += "<option
value='"+field.id+"'>"+field.merk+"</option>"
    });
    text += "</select></div>";
    txt = "<div class='row'><div class='col-md-
12'>"+text+text+text+"</div></div>";
    $('#home3').empty();
    $('#home3').append(txt);
});
}

function sf2() {
    perusahaan = $('#s2').val();
    // alert(perusahaan);

    $.getJSON("{{url('Merks')}}/"+perusahaan, function(result) {
        // console.log("yes"+result);
        txt = "<option>---Pilih Merk---</option>";
        $.each(result, function(i, field) {
            txt += "<option
value='"+field.id+"'>"+field.merk+"</option>";
        });
        $('#s3').empty();
        $('#s3').append(txt);
    });
}

function sf1() {
    sparepart = $('#s1').val();

    $.getJSON("{{url('Hargas')}}/"+sparepart, function(result) {
        // console.log("yes"+result);
```

```
txt4 = "<option>---Pilih Harga---</option>";
txt5 = "<option>---Pilih Garansi---</option>";
txt6 = "<option>---Pilih Ketahanan---
</option>";

$.each(result, function(i, field){
    tmp = "<option
value='"+field.nilai+"'>"+field.sub_kriteria+"</option>";
    if(field.kriteria == "Harga"){
        txt4 +=tmp;
    }
    if(field.kriteria == "Garansi"){
        txt5 +=tmp;
    }
    if(field.kriteria == "Ketahanan"){
        txt6 +=tmp;
    }
});
$('#s4').empty();
$('#s5').empty();
$('#s6').empty();
$('#s4').append(txt4);
$('#s5').append(txt5);
$('#s6').append(txt6);
});
}
</script>
@endsection
```

Kode Program View *penjualan2.blade.php* (Bagian Karyawan)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
```

```
<meta http-equiv="X-UA-Compatible"
content="IE=edge">
  <meta name="viewport" content="width=device-
width, initial-scale=1">
  <meta name="description" content="">
  <meta name="author" content="">
  <link rel="icon" type="image/png" sizes="16x16"
href="../plugins/images/favicon.png">
  <title>SPAREPART</title>
  <!-- Bootstrap Core CSS -->
  <link
href="{{url('bootstrap/dist/css/bootstrap.min.css')}}
" rel="stylesheet">
  <!-- Menu CSS -->
  <link
href="{{url('../plugins/bower_components/sidebar-
nav/dist/sidebar-nav.min.css')}}" rel="stylesheet">
  <!-- toast CSS -->
  <link
href="{{url('../plugins/bower_components/toast-
master/css/jquery.toast.css')}}" rel="stylesheet">
  <!-- morris CSS -->
  <link
href="{{url('../plugins/bower_components/morrisjs/mor-
ris.css')}}" rel="stylesheet">
  <!-- animation CSS -->
  <link href="{{url('css/animate.css')}}"
rel="stylesheet">
  <!-- Custom CSS -->
  <link href="{{url('css/style.css')}}"
rel="stylesheet">
```

```
<!-- color CSS -->
<link href="{{url('css/colors/blue-dark.css')}}"
id="theme" rel="stylesheet">
<script
src="{{url('../plugins/bower_components/jquery/dist/j
query.min.js')}}"></script>
<!-- HTML5 Shim and Respond.js IE8 support of
HTML5 elements and media queries -->
<!-- WARNING: Respond.js doesn't work if you view
the page via file:// -->
<!--[if lt IE 9]>
<script
src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html
5shiv.js"></script>
<script
src="https://oss.maxcdn.com/libs/respond.js/1.4.2/res
pond.min.js"></script>
<![endif]-->
</head>

<body style="overflow-x: scroll;background-color:
white;">
<div class="container" style="overflow-x:auto;margin-
left:100px;margin-right:100px;">
<!-- <img src="" style="position: absolute; margin:
0px; padding: 0px; border: none; width: 1349px;
height: 899.503px; max-width: none; z-index: -999999;
left: 0px; top: -271.252px;"> -->
<div class="row">
<div class="col-md-12">
<div class="panel panel-default">
```

```

        <div class="panel-heading text-
center">PERHITUNGAN
                <button type="button"
class="btn-sm btn-primary pull-right" data-
toggle="modal" data-target="#creates">Lakukan
Perhitungan</button>
                <a href="/" class="btn-sm btn-
danger pull-left">Kembali</a>
        </div>

        <div class="panel-body">
                <table class="table table-
responsive" style="min-width:100%;">
                        <thead
style="color:white;background-color:#f0f0f0;">
                                <th colspan="4"
class="text-center" style="border-right:solid 1px
black;border-left:solid 1px black;border-top:solid
1px black;border-bottom:solid 1px
black;">Keterangan</th>
                                <th colspan="6"
class="text-center" style="border-right:solid 1px
black;border-top:solid 1px black;border-bottom:solid
1px black;">Menghitung Nilai Preferensi</th>
                                <th rowspan="2"
style="border-right:solid 1px black;border-top:solid
1px black;">Hasil Perkalian</th>
                                <th rowspan="2"
style="border-right:solid 1px black;border-top:solid
1px black;">Total</th>

```

```

                <th rowspan="2"
style="border-right:solid 1px black;border-top:solid
1px black;">Rangking</th>
                </thead>
            <thead
style="color:white;background-color:#f0f0f0;">
                <th style="border-
left:solid 1px black;border-bottom:solid 1px
black;">ID</th>
                <th style="border-
bottom:solid 1px black;">Perusahaan</th>
                <th style="border-
bottom:solid 1px black;">Merk</th>
                <th style="border-
right:solid 1px black;border-bottom:solid 1px
black;">Spare Part</th>
                <th style="border-
bottom:solid 1px black;">S1</th>
                <th style="border-
bottom:solid 1px black;">S2</th>
                <th style="border-
bottom:solid 1px black;">S3</th>
                <th style="border-
bottom:solid 1px black;">S4</th>
                <th style="border-
bottom:solid 1px black;">S5</th>
                <th style="border-
right:solid 1px black;border-bottom:solid 1px
black;">S6</th>

```



```
<button type="button"
class="close" data-dismiss="modal">&times;</button>
    <h4 class="modal-title">Masukkan
Kriteria</h4>
</div>
<form action="" method="get" accept-
charset="utf-8">
    <div class="modal-body">
        <div class="row">
            <div class="col-md-12">
                <label for="">Grup
Barang</label>
                <select
name="grup_barang" class="form-control" id="">
                    <option
value="">---</option>
                    @foreach($grupBarang as $row)
                        <option
value="{{ $row->id }}">{{ $row->nama }}</option>
                    @endforeach
                </select>
            </div>
        </div>
        <br>
        <br>
        <div class="row">
            <div class="col-md-4">
                <label
for="">Perusahaan 1</label>
```

```
                                <select
name="s1[]" id="vs1" class="form-control"
onchange="setPerusahaan1();" required>
                                <option
value="">---</option>

                                @foreach($perusahaan as $row)
                                <option
value="{{ $row->id }}">{{ $row->perusahaan }}</option>
                                @endforeach
                                </select>
                                </div>
                                <div class="col-md-4">
                                <label
for="">Perusahaan 2</label>
                                <select
name="s1[]" id="vs2" class="form-control"
onchange="setPerusahaan2();" required>
                                <option
value="">---</option>

                                @foreach($perusahaan as $row)
                                <option
value="{{ $row->id }}">{{ $row->perusahaan }}</option>
                                @endforeach
                                </select>
                                </div>
                                <div class="col-md-4">
                                <label
for="">Perusahaan 3</label>
```



```

        <div
id="home1" class="tab-pane fade in active">

        </div>

        <div

id="home2" class="tab-pane fade">

        </div>

        <div

id="home3" class="tab-pane fade">

        </div>

        </div>

        </div>

        </div>

        <div class="modal-footer">
            <button class="btn-sm btn-
primary" type='submit'>HITUNG</button>
            <button type="button"
class="btn btn-default pull-left" data-
dismiss="modal">Close</button>
        </div>
    </form>
</div>
</div>
</div>
<script
src="{{url('bootstrap/dist/js/bootstrap.min.js')}}"><
/script>
    <!-- Menu Plugin JavaScript -->

```

```
<script
src="{{url('../plugins/bower_components/sidebar-
nav/dist/sidebar-nav.min.js')}}"></script>
<!--slimscroll JavaScript -->
<script
src="{{url('js/jquery.slimscroll.js')}}"></script>
<!--Wave Effects -->
<script src="{{url('js/waves.js')}}"></script>
<!-- Custom Theme JavaScript -->
<script
src="{{url('js/custom.min.js')}}"></script>
<script>
function setPerusahaan1(){
    $('#h1').empty();
    $('#h1').append($('#vs1
option:selected').text());
    url =
"{{url('Merk2') . '/' }}" + $('#vs1').val();
    $.getJSON(url, function(result){
        // console.log(result);
        text = "<div class='col-md-4'><select
class='form-control' name='merk[]' required>";
        $.each(result, function(i, field){
            text += "<option
value='"+field.id+"'>" + field.merk + "</option>"
        });
        text += "</select></div>";
        txt = "<div class='row'><div
class='col-md-12'>" + text + text + text + "</div></div>";
        $('#home1').empty();
```

```
        $('#home1').append(txt);
    });

}

function setPerusahaan2() {
    $('#h2').empty();
    $('#h2').append($('#vs2
option:selected').text());
    url =
    "{{url('Merk2').'/'}}"+$('#vs2').val();
    $.getJSON(url, function(result) {
        // console.log(result);
        text = "<div class='col-md-4'><select
class='form-control' name='merk[]' required>";
        $.each(result, function(i, field) {
            text += "<option
value='"+field.id+"'>"+field.merk+"</option>"
        });
        text += "</select></div>";
        txt = "<div class='row'><div
class='col-md-12'>"+text+text+text+"</div></div>";
        $('#home2').empty();
        $('#home2').append(txt);
    });
}

function setPerusahaan3() {
    $('#h3').empty();
    $('#h3').append($('#vs3
option:selected').text());
```

```
url =
"{{url('Merk2')}}"+$('#vs3').val();
$.getJSON(url, function(result) {
    // console.log(result);
    text = "<div class='col-md-4'><select
class='form-control' name='merk[]' required>";
    $.each(result, function(i, field) {
        text += "<option
value='"+field.id+"'>"+field.merk+"</option>"
    });
    text += "</select></div>";
    txt = "<div class='row'><div
class='col-md-12'>"+text+text+text+"</div></div>";
    $('#home3').empty();
    $('#home3').append(txt);
});
}

function sf2() {
    perusahaan = $('#s2').val();
    // alert(perusahaan);

    $.getJSON("{{url('Merks')}}/"+perusahaan, function(
result) {
        // console.log("yes"+result);
        txt = "<option>---Pilih Merk---
</option>";
        $.each(result, function(i, field) {
            txt += "<option
value='"+field.id+"'>"+field.merk+"</option>";
        });
    });
}
```



```
        $('#s3').empty();
        $('#s3').append(txt);
    });
}

function sf1(){
    sparepart = $('#s1').val();

    $.getJSON("{url('Hargas')}"+sparepart,function
n(result){
        // console.log("yes"+result);
        txt4 = "<option>---Pilih Harga---
</option>";
        txt5 = "<option>---Pilih Garansi---
</option>";
        txt6 = "<option>---Pilih Ketahanan---
</option>";

        $.each(result, function(i, field){
            tmp = "<option
value='"+field.nilai+"'>"+field.sub_kriteria+"</optio
n>";

            if(field.kriteria == "Harga"){
                txt4 +=tmp;
            }
            if(field.kriteria == "Garansi"){
                txt5 +=tmp;
            }
            if(field.kriteria ==
"Ketahanan"){
```

```

        txt6 +=tmp;
    }
});
$('#s4').empty();
$('#s5').empty();
$('#s6').empty();
$('#s4').append(txt4);
$('#s5').append(txt5);
$('#s6').append(txt6);
});
}
</script>
</body>
</html>

```

E. Lampiran Black Box

No	Menu	Fungsi	Aksi	Hasil	Ket
1.	Mengelola Data Grup Barang	Digunakan untuk lihat tabel grup barang, tambah data grup barang, mengubah data grup barang, dan menghapus data grup barang	Memilih tombol tambah grup barang	Menampilkan <i>modal show</i> form tambah grup barang	√
			Mengisi <i>modal show</i> form grup barang dan klik tombol Simpan	Menyimpan data ke <i>database</i> dan menampilkan halaman grup barang	√
			Form <i>modal show</i> grup barang tidak	Menampilkan notifikasi	√

			terisi dan klik tombol Simpan	"Please fill out this field"	
			Klik tombol Cancel	Menampilkan halaman grup barang	√
			Memilih tombol Update	Menampilkan <i>modal show form Update</i>	√
			Mengubah <i>modal show form grup</i> barang dan klik tombol Simpan	Menyimpan perubahan data ke <i>database</i> dan menampilkan halaman grup barang	√
			Form <i>modal show form Update</i> tidak terisi dan klik tombol Simpan	Menampilkan notifikasi "Please fill out this field"	√
			Klik tombol Cancel	Menampilkan halaman grup barang	√
			Memilih tombol Delete	Menampilkan peringatan "Apakah Anda Yakin Untuk Menghapus", tombol Ok	√

				dan Tombol Cancel	
			Klik Tombol Ok	Menghapus data perusahaan terpilih dan menampilkan halaman grup barang	√
			Klik Tombol Cancel	Menampilkan halaman grup barang	√
2.	Mengelola Data Kriteria	Digunakan untuk lihat tabel kriteria tambah data kriteria, mengubah data kriteria, dan menghapus data kriteria	Memilih tombol tambah kriteria	Menampilkan <i>modal show</i> form tambah kriteria	√
			Mengisi <i>modal show</i> form kriteria dan klik tombol Simpan	Menyimpan data ke <i>database</i> dan menampilkan halaman kriteria	√
			Form <i>modal show</i> kriteria tidak terisi dan klik tombol Simpan	Menampilkan notifikasi "Please fill out this field"	√
			Klik tombol Cancel	Menampilkan halaman kriteria	√

			Memilih tombol Update	Menampilkan <i>modal show form Update</i>	√
			Mengubah <i>modal show form kriteria</i> dan klik tombol Simpan	Menyimpan data kriteria ke <i>database</i> dan menampilkan halaman kriteria	√
			Form <i>modal show form Update</i> tidak terisi dan klik tombol Simpan	Menampilkan notifikasi "Please fill out this field"	√
			Klik tombol Cancel	Menampilkan halaman kriteria	√
			Memilih tombol Delete	Menampilkan peringatan "Apakah Anda Yakin Untuk Menghapus", tombol Ok dan Tombol Cancel	√
			Klik Tombol Ok	Menghapus data kriteria terpilih dan	√

				menampilkan halaman kriteria	
			Klik Tombol Cancel	Menampilkan halaman kriteria	√
3.	Mengelola Data Sub Kriteria	Digunakan untuk lihat tabel sub kriteria tambah data kriteria, mengubah data kriteria, dan menghapus data kriteria	Memilih tombol tambah sub kriteria	Menampilkan <i>modal show</i> form tambah sub kriteria	√
			Mengisi <i>modal show</i> form sub kriteria dan klik tombol Simpan	Menyimpan data ke <i>database</i> dan menampilkan halaman sub kriteria	√
			Form <i>modal show</i> sub kriteria tidak terisi dan klik tombol Simpan	Menampilkan notifikasi "Please fill out this field"	√
			Klik tombol Cancel	Menampilkan halaman sub kriteria	√
			Memilih tombol Update	Menampilkan <i>modal show</i> form <i>Upadate</i>	√
			Mengubah <i>modal show</i> form sub	Menyimpan data sub kriteria ke	√

			kriteria dan klik tombol Simpan	<i>database</i> dan menampilkan halaman sub kriteria	
			Form <i>modal show</i> form <i>Update</i> tidak terisi dan klik tombol Simpan	Menampilkan notifikasi "Please fill out this field"	√
			Klik tombol Cancel	Menampilkan halaman sub kriteria	√
			Memilih tombol Delete	Menampilkan peringatan "Apakah Anda Yakin Untuk Menghapus", tombol Ok dan Tombol Cancel	√
			Klik Tombol Ok	Menghapus data sub kriteria terpilih dan menampilkan halaman sub kriteria	√

			Klik Tombol Cancel	Menampilkan halaman sub kriteria	√
4.	Mengelola Data Bobot Kriteria	Digunakan untuk lihat tabel bobot kriteria tambah data kriteria, mengubah data kriteria, dan menghapus data kriteria	Memilih tombol Update	Menampilkan <i>modal show form Update</i>	√
			Mengubah <i>modal show form bobot kriteria</i> dan klik tombol Simpan	Menyimpan data bobot kriteria ke <i>database</i> dan menampilkan halaman bobot kriteria	√
			Form <i>modal show form Update</i> tidak terisi dan klik tombol Simpan	Menampilkan notifikasi "Please fill out this field"	√
			Klik tombol Cancel	Menampilkan halaman bobot kriteria	√
5.	Mengelola Rekomendasi Perhitungan	Digunakan untuk melakukan rekomendasi perhitungan <i>spare part</i> dengan menggunakan metode	Memilih tombol lakukan perhitungan	Menampilkan form rekomendasi perhitungan	√
			Mengisi form rekomendasi pemilihan kemudian klik tombol hitung	Melakukan proses perhitungan dan menampilkan	√

		<i>Weighted Product (WP)</i>		pada hasil perhitungan pada tabel di halaman rekomendasi perhitungan	
6.	<i>Logout</i>	Digunakan untuk keluar dari sistem	Klik admid kemudian pilih klik logout pada pojok kanan atas sistem	Melakukan proses keluar dari sistem dan menampilkan halaman <i>login</i>	√