

**RANCANG BANGUN SISTEM PENGKLASIFIKASI KECEPATAN
MAKSIMUM KERETA API PADA JALUR KLAKAH-PASIRIAN
MENGUNAKAN METODE *SINGLE LAYER PERCEPTRON***

SKRIPSI

Oleh

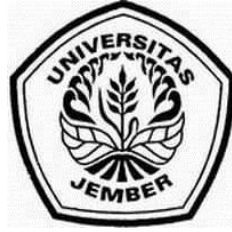
Tri Mada Abdillah

NIM 122410101043

PROGRAM STUDI SISTEM INFORMASI

UNIVERSITAS JEMBER

2017



**RANCANG BANGUN SISTEM PENGKLASIFIKASI KECEPATAN
MAKSIMUM KERETA API PADA JALUR KLAKAH-PASIRIAN
MENGUNAKAN METODE *SINGLE LAYER PERCEPTRON***

SKRIPSI

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

Oleh

**Tri Mada Abdillah
NIM 122410101043**

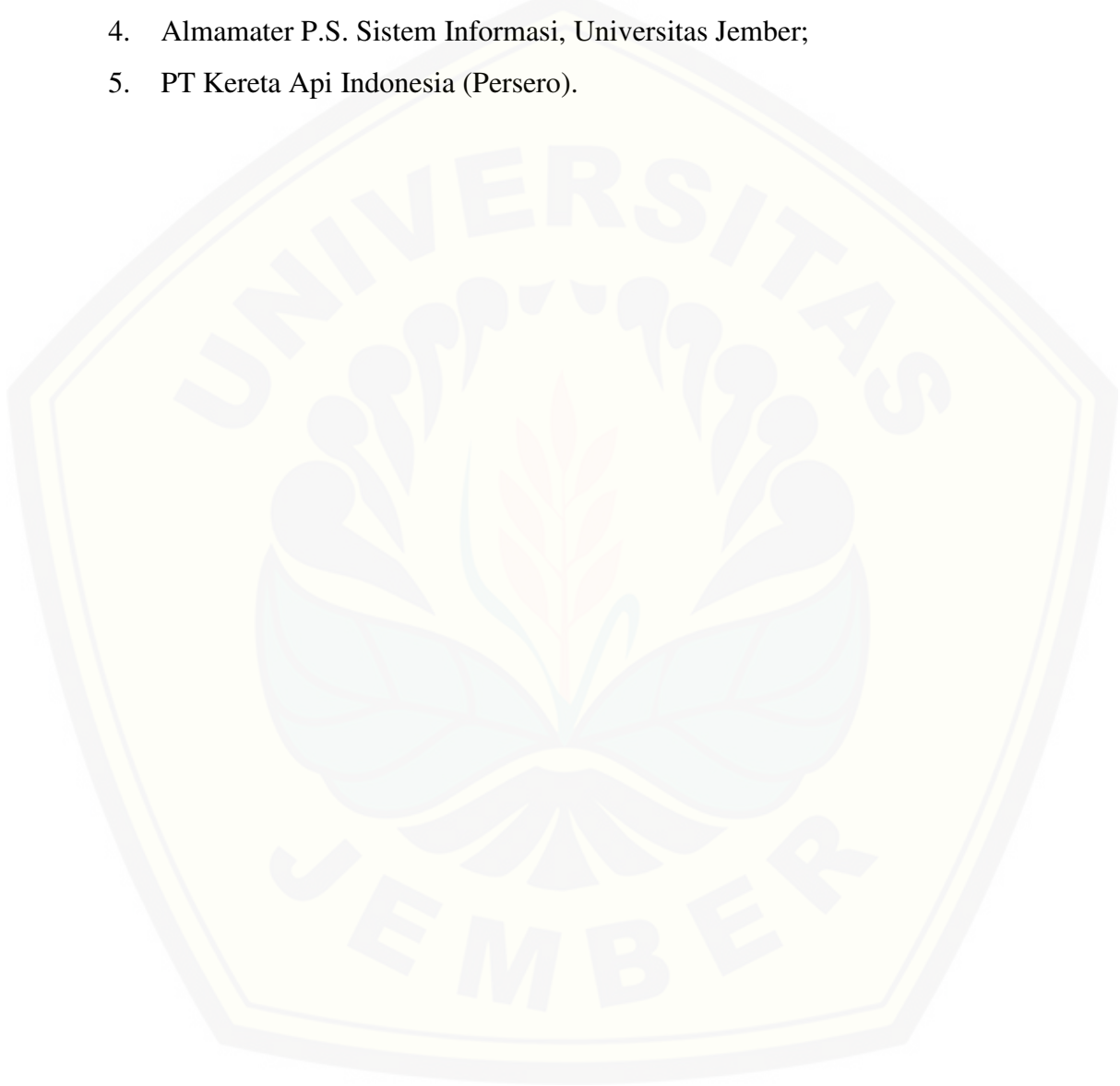
**PROGRAM STUDI SISTEM INFORMASI
UNIVERSITAS JEMBER**

2017

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Ibunda Mamik Megowati dan Ayahanda Djoko Soerjanto yang tercinta;
2. Mbak Ika Ratmawati dan Mbak Dwi Ohta Pebriyanti;
3. Ibu/Bapak Guru sejak taman kanak-kanak sampai dengan perguruan tinggi;
4. Almamater P.S. Sistem Informasi, Universitas Jember;
5. PT Kereta Api Indonesia (Persero).



MOTO

Semakin dangkal ilmu seseorang, maka tentunya ia semakin mudah berfatwa dan menghukumi. Semakin ahli dan tingginya ilmu seseorang, maka semakin ia berhati-hati dalam berfatwa dan tidak ceroboh dalam menghukumi.¹



¹ Habib Munzir Almusawa. 2007. *Kenalilah Aqidahmu*. Jakarta: Majelis Rasulullah.

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

nama : Tri Mada Abdillah

NIM : 122410101043

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Rancang Bangun Sistem Pengklasifikasi Kecepatan Maksimum Kereta Api pada Jalur Klakah-Pasirian Menggunakan Metode *Single Layer Perceptron*” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

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

Jember, 17 Mei 2017

Yang menyatakan,

Tri Mada Abdillah

NIM 122410101043

SKRIPSI

**RANCANG BANGUN SISTEM PENGKLASIFIKASI KECEPATAN
MAKSIMUM KERETA API PADA JALUR KLAKAH-PASIRIAN
MENGUNAKAN METODE *SINGLE LAYER PERCEPTRON***

Oleh

Tri Mada Abdillah

NIM 122410101043

Pembimbing

Dosen Pembimbing Utama : Anang Andrianto, S.T., M.T.

Dosen Pembimbing Anggota : M. Arief Hidayat, S.Kom, M.Kom.

PENGESAHAN PEMBIMBING

Skripsi berjudul “Rancang Bangun Sistem Pengklasifikasi Kecepatan Maksimum Kereta Api pada Jalur Klakah-Pasirian Menggunakan Metode *Single Layer Perceptron*” telah diuji dan disahkan pada:

hari, tanggal : Jum’at, 28 April 2017

tempat : Program Studi Sistem Informasi, Universitas Jember

Disetujui oleh:

Pembimbing I,

Pembimbing II,

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

NIP 19690615 199702 1 002

M. Arief Hidayat, S.Kom. M.Kom.

NIP 19810123 201012 1 003

PENGESAHAN

Skripsi berjudul “Rancang Bangun Sistem Pengklasifikasi Kecepatan Maksimum Kereta Api pada Jalur Klakah-Pasirian Menggunakan Metode *Single Layer Perceptron*” telah diuji dan disahkan pada:

hari, tanggal : Jum’at, 28 April 2017

tempat : Program Studi Sistem Informasi, Universitas Jember

Penguji I,

Penguji II,

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

Priza Pandunata, S.Kom., M.Comp.Sc.

NIP 19670420 199201 1 001

NIP 19830131 201504 1 001

Mengesahkan

Ketua

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

NIP 19670420 199201 1 001

RINGKASAN

Rancang Bangun Sistem Pengklasifikasi Kecepatan Maksimum Kereta Api pada Jalur Klakah-Pasirian Menggunakan Metode *Single Layer Perceptron*; Tri Mada Abdillah, 122410101043; 2017; 139 halaman; Program Studi Sistem Informasi Universitas Jember.

Kecepatan maksimum kereta api merupakan kecepatan tertinggi yang diizinkan untuk operasi suatu rangkaian kereta api pada lintas tertentu. Selain itu, kecepatan maksimum kereta api merupakan salah satu parameter dari perencanaan konstruksi jalur kereta api yang bertujuan untuk mendapatkan keluaran terbaik pada pembangunan dan pemeliharaan konstruksi serta menjamin keamanan dan kenyamanan dalam pengoperasian suatu rangkaian kereta api. Pada penelitian terdahulu, didapati bahwa kecepatan maksimum Jalur Kereta Api Klakah-Pasirian adalah 120 km/jam dan nilai tebal balas atas adalah 55 cm. Sedangkan pada Peraturan Dinas No. 10 Tahun 1986 Perusahaan Jawatan Kereta Api, suatu lintas tidak dapat ditentukan nilai kecepatan maksimum kereta api jika memiliki nilai tebal balas atas 55 cm. Sehingga, diperlukan cara baru untuk menentukan nilai kecepatan maksimum kereta api pada suatu jalur atau lintas. Tujuan penelitian untuk memahami tentang cara menentukan kecepatan maksimum kereta api pada Jalur Kereta Api Klakah-Pasirian menggunakan metode *Single Layer Perceptron* dan memetakan hasil kecepatan maksimum kereta api tersebut.

Penelitian ini merancang dan membangun sebuah sistem informasi untuk klasifikasi kecepatan maksimum kereta api pada Jalur Klakah-Pasirian menggunakan metode *Single Layer Perceptron* di mana merupakan salah satu arsitektur *perceptron* dari Jaringan Syaraf Tiruan yang memiliki kelebihan seperti mampu mengakuisisi atau memperoleh pengetahuan walau tidak ada kepastian pada data yang digunakan, melakukan generalisasi dan ekstraksi dari suatu pola data tertentu, dan dapat menciptakan suatu pola pengetahuan melalui pengaturan diri atau kemampuan belajar. Model pengembangan perangkat lunak untuk perancangan dan pembangunan dari sistem ini menggunakan model *waterfall*. Pada penelitian ini, perancangan sistem menggunakan UML dengan konsep

OOAD. Selain itu, pembangunan sistem menggunakan kerangka pemrograman atau *framework* CodeIgniter, *web server* Apache dengan aplikasi *local server* XAMPP, dan sistem pengelolaan basis data PostgreSQL. Kemudian, pengujian sistem menggunakan *white box testing* dan *black box testing*.

Sistem ini dapat melakukan klasifikasi *training-testing* pada *dataset* kecepatan maksimum kereta api secara utuh menggunakan metode *Single Layer Perceptron*. Parameter-parameter yang digunakan antara lain nilai *learning rate* 0,1 dan epoch 10000. Nilai akurasi dari hasil klasifikasi tersebut adalah 100% dengan epoch berhenti pada pengulangan ke-1806. Dari klasifikasi tersebut, didapatkan pula nilai bobot dan bias yang selanjutnya dapat digunakan sebagai klasifikasi data baru berdasarkan nilai akurasi tersebut untuk mendapatkan nilai kecepatan maksimum kereta api pada suatu ruas lintas kereta api. Selain itu, sistem dapat memetakan nilai-nilai kecepatan maksimum pada suatu lintas atau jalur kereta api yang dalam objek penelitian ini menggunakan Jalur Kereta Api Klakah-Pasirian.

PRAKATA

Alhamdulillah Robbil ‘Alamin, segala puji dan syukur ke hadirat Allah Subhanahu wa Ta’ala atas segala rahmat, karunia, hidayah, dan inayah-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Rancang Bangun Sistem Pengklasifikasi Kecepatan Maksimum Kereta Api pada Jalur Klakah-Pasirian Menggunakan Metode *Single Layer Perceptron*”. 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. Drs. Slamin, M.Comp.Sc., Ph.D selaku Ketua Program Studi Sistem Informasi, Universitas Jember yang telah bekerja keras untuk mewujudkan kampus P.S. Sistem Informasi yang lebih baik dari hari ke hari;
2. Bapak Anang Andrianto, S.T., M.T. selaku Dosen Pembimbing Utama dan Bapak Muhamad Arief Hidayat, S.Kom, M.Kom. selaku Dosen Pembimbing Anggota yang telah berbagi waktu dan pikiran selama penulis melakukan bimbingan skripsi;
3. Bapak Anang Andrianto, S.T., M.T. selaku Dosen Wali yang telah berbagi waktu dan pikiran selama ini kepada penulis;
4. Bapak dan Ibu Dosen P.S. Sistem Informasi, Universitas Jember yang telah memberikan kepada penulis ilmu-ilmu yang bermanfaat;
5. Bapak dan Ibu Karyawan Tata Usaha P.S. Sistem Informasi, Universitas Jember yang telah berbagi waktu, penjelasan, dan penyelesaian dalam pengurusan administrasi skripsi dari penulis;
6. Ibunda Mamik Megowati dan Ayahanda Djoko Soerjanto yang senantiasa memberikan dorongan, pikiran, dan tenaga yang begitu besar kepada penulis;
7. Mbak Ika Ratmawati dan Mbak Dwi Ochta Pebriyanti yang senantiasa berbagi pengalaman dan motivasi kepada penulis;
8. PT Kereta Api Indonesia (Persero), khususnya para pegawai dan pimpinan Daerah Operasi IX Jember, Resor Jalan Rel 9.2 Probolinggo, dan Resor Jalan

Rel 9.4 Klakah yang telah memberikan data-data utama dalam penelitian penulis ini;

9. Badan Kesatuan Bangsa dan Politik atau Bankesbangpol, Dinas Perhubungan atau Dishub, dan Badan Perencanaan dan Pembangunan Daerah BAPPEDA Kabupaten Lumajang yang telah memberikan data-data penunjang dalam penelitian penulis ini;
10. Seluruh teman-teman Program Studi Sistem Informasi, khususnya angkatan 2012 / Formation yang telah memberikan kepada penulis tentang arti dari persahabatan;
11. Mas M. Shidiq dan Mbak Warda yang telah menyediakan tempat kos semasa penulis kuliah, khususnya semasa penulis mengerjakan skripsi;
12. semua pihak yang tidak dapat disebutkan satu per satu yang telah berbagi waktu, pikiran, dan tenaga;

Penulis menerima segala kritik dan saran dari semua pihak atas penyusunan skripsi ini. Akhirnya penulis berharap, semoga skripsi ini dapat bermanfaat.

Jember, Mei 2017

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PERSEMBAHAN	ii
HALAMAN MOTO	iii
HALAMAN PERNYATAAN	iv
HALAMAN PENGESAHAN PEMBIMBING	vi
HALAMAN PENGESAHAN PENGUJI	vii
RINGKASAN	viii
PRAKATA	x
DAFTAR ISI	xii
DAFTAR TABEL	xiv
DAFTAR GAMBAR	xvi
DAFTAR NOTASI	xix
DAFTAR SINGKATAN	xx
DAFTAR LAMPIRAN	xxi
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	3
1.3 Tujuan dan Manfaat	3
1.4 Batasan Masalah	4
1.5 Sistematika Penulisan	4
BAB 2. TINJAUAN PUSTAKA	6
2.1 Penelitian Terdahulu	6
2.2 Algoritme Klasifikasi	7
2.3 Kecepatan Maksimum	7
2.4 Jalur Kereta Api Klakah-Pasirian	12
2.5 Metode <i>Single Layer Perceptron</i>	16
2.6 Model <i>Waterfall</i>	18
2.7 Perumusan Premis	20
2.8 Perumusan Hipotesis	21
BAB 3. METODE PENELITIAN	22
3.1 Jenis Penelitian	22

3.2	Tempat dan Waktu Penelitian	22
3.3	Rancangan Penelitian	23
3.4	Teknik Perolehan Data	25
3.5	Analisis Kebutuhan Sistem.....	25
3.6	Perancangan Sistem.....	29
3.7	Implementasi Sistem	29
3.8	Pengujian Sistem	30
3.9	Pemeliharaan Sistem	30
BAB 4. PERANCANGAN DAN PEMBANGUNAN SISTEM		31
4.1	Gambaran Umum Sistem	31
4.2	Analisis Kebutuhan Perangkat Lunak	33
4.3	Perancangan Sistem.....	34
4.4	Penerapan Perancangan Sistem	79
4.5	Pengujian Sistem	79
4.6	Pemeliharaan Sistem	111
BAB 5. HASIL DAN PEMBAHASAN		112
5.1.	Hasil Pembuatan Sistem.....	112
5.2.	Pembahasan.....	123
BAB 6. PENUTUP		136
6.1.	Kesimpulan.....	136
6.2.	Saran.....	137
DAFTAR PUSTAKA		138
LAMPIRAN.....		140

DAFTAR TABEL

	Halaman
Tabel 2.1 Kelas Jalan Rel.....	8
Tabel 2.2 Lebar Badan Jalan Rel	8
Tabel 2.3 Jari-jari Minimum Lengkung Horizontal.....	8
Tabel 2.4 Jari-jari Minimal Lengkung Vertikal	9
Tabel 2.5 Landai Penentu Maksimum	9
Tabel 2.6 Penampang Melintang Jalan Rel.....	9
Tabel 2.7 Data Lintas Klakah-Pasirian	15
Tabel 4.1 Penjelasan <i>Use Case Diagram</i>	37
Tabel 4.2 Skenario <i>Login</i> Manajer Jalan dan Jembatan.....	40
Tabel 4.3 Skenario <i>Login</i> Asisten Manajer Konstruksi Jalan Rel dan Jembatan..	41
Tabel 4.4 Skenario <i>Login</i> Kepala Sub Direktorat Jalan Rel dan Jembatan Pusat.	42
Tabel 4.5 Skenario Mengelola Data Peta “Tambah”	44
Tabel 4.6 Skenario Mengelola Data Peta “Ubah”	45
Tabel 4.7 Skenario Mengelola Data Peta “Tampil”	46
Tabel 4.8 Skenario Melakukan Validasi Perancangan Kecepatan Maksimum pada Data Peta	47
Tabel 4.9 Skenario Mengelola <i>Dataset</i> “tampil”	49
Tabel 4.10 Skenario Mengelola <i>Dataset</i> “tambah”	50
Tabel 4.11 Skenario Mengelola <i>Dataset</i> “ubah”	51
Tabel 4.12 Skenario Mengelola <i>Dataset</i> “hapus”	52
Tabel 4.13 Skenario Melakukan Pencarian <i>Dataset</i>	53
Tabel 4.14 Skenario Memproses Klasifikasi SLP pada <i>dataset</i>	54
Tabel 4.15 Skenario Meninjau Tabel Status Perancangan.....	55
Tabel 4.16 Skenario Mengisi Nilai Kriteria Lintas	56
Tabel 4.17 Skenario Melakukan Klasifikasi per-Ruas Lintas.....	57
Tabel 4.18 Skenario Menyelesaikan Klasifikasi	58
Tabel 4.19 Skenario Mencetak Laporan Desain Kecepatan Maksimum	59
Tabel 4.20 Skenario <i>Logout</i>	60
Tabel 4.21 <i>Test Case index() – c_perceptron.php</i>	97

Tabel 4.22 Test Case <i>index()</i> – <i>c_login.php</i>	98
Tabel 4.23 Test Case <i>index()</i> – <i>c_utama.php</i>	98
Tabel 4.24 Test Case <i>proses()</i>	99
Tabel 4.25 Test Case <i>user(\$nipp)</i>	99
Tabel 4.26 Test Case <i>jml_kolom()</i>	100
Tabel 4.27 Test Case <i>datatrain()</i>	100
Tabel 4.28 Test Case <i>awal(\$jmlTitik, \$learning_rate)</i>	100
Tabel 4.29 Test Case <i>traintest(\$data, \$numEpoch)</i>	102
Tabel 4.30 Test Case <i>result_conv(\$res)</i>	108
Tabel 4.31 Test Case <i>get_hitungakh(\$nipp)</i>	108
Tabel 4.32 Test Case <i>tambah_hitungakh(\$na, \$sepoh, \$lr, \$nipp, \$larik)</i>	109
Tabel 4.33 Test Case <i>hasil_bobot()</i>	109
Tabel 4.34 Test Case <i>hasil_bias()</i>	110
Tabel 4.35 Test Case <i>keterangan()</i>	110
Tabel 5.1 Hasil Penghitungan dari Klasifikasi SLP pada Data Baru.....	131
Tabel 5.2 Hasil Pengujian Validasi.....	133

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Bentuk Penampang Jalan Rel secara Melintang	11
Gambar 2.2 Jalur dengan Lengkung Vertikal	11
Gambar 2.3 Jalur dengan Lengkung Horizontal	11
Gambar 2.4 Jalur dengan Lengkung Horizontal Peralihan	12
Gambar 2.5 Peta Rencana Jaringan Kereta Api Kabupaten Lumajang	13
Gambar 2.6 Peta Jalur Kereta Api Klakah-Pasirian.....	14
Gambar 2.7 Arsitektur <i>Single Layer Perceptron</i>	17
Gambar 2.8 Model <i>Waterfall</i>	18
Gambar 3.1 Rancangan Penelitian	24
Gambar 3.2 Alur Proses Klasifikasi Metode SLP pada <i>Dataset</i>	27
Gambar 3.3 Alur Klasifikasi Metode SLP untuk Data Baru.....	28
Gambar 4.1 <i>Business Process</i> SIKEMAKA	35
Gambar 4.2 <i>Use Case Diagram</i> SIKEMAKA	37
Gambar 4.3 <i>Activity Diagram</i> Memproses Klasifikasi Dataset	61
Gambar 4.4 <i>Activity Diagram</i> Melakukan Klasifikasi per Ruas Lintas.....	62
Gambar 4.5 <i>Sequence Diagram</i> Memproses Klasifikasi <i>Dataset</i>	68
Gambar 4.6 <i>Sequence Diagram</i> Melakukan Klasifikasi per Ruas Lintas.....	70
Gambar 4.7 <i>Class Diagram</i> SIKEMAKA	76
Gambar 4.8 ERD SIKEMAKA.....	78
Gambar 4.9 <i>Program Code</i> <code>index() – c_perceptron.php</code>	80
Gambar 4.10 <i>Program Code</i> <code>index() – c_login.php</code>	80
Gambar 4.11 <i>Program Code</i> <code>index() – c_utama.php</code>	81
Gambar 4.12 <i>Program Code</i> <code>proses()</code>	81
Gambar 4.13 <i>Program Code</i> <code>user(\$nipp)</code>	82
Gambar 4.14 <i>Program Code</i> <code>jml_kolom()</code>	82
Gambar 4.15 <i>Program Code</i> <code>datatrain()</code>	82
Gambar 4.16 <i>Program Code</i> <code>awal(\$jmlTitik, \$learning_rate)</code>	82
Gambar 4.17 <i>Program Code</i> <code>traintest(\$data, \$numEpoch)</code>	83
Gambar 4.18 <i>Program Code</i> <code>result_conv(\$res)</code>	84

Gambar 4.19 Program Code <i>get_hitungakh(\$nipp)</i>	84
Gambar 4.20 Program Code <i>tambah_hitungakh(\$na, \$epoh, \$lr, \$nipp, \$larik)</i> 84	84
Gambar 4.21 Program Code <i>hasil_bobot()</i>	84
Gambar 4.22 Program Code <i>hasil_bias()</i>	84
Gambar 4.23 Program Code <i>keterangan()</i>	85
Gambar 4.24 Diagram Alir <i>index() – c_perceptron.php</i>	85
Gambar 4.25 Diagram Alir <i>index() – c_login.php</i>	85
Gambar 4.26 Diagram Alir <i>index() – c_utama.php</i>	86
Gambar 4.27 Diagram Alir <i>proses()</i>	86
Gambar 4.28 Diagram Alir <i>user(\$nipp)</i>	87
Gambar 4.29 Diagram Alir <i>jml_kolom()</i>	87
Gambar 4.30 Diagram Alir <i>datatrain()</i>	87
Gambar 4.31 Diagram Alir <i>awal(\$jmlTitik, \$learning_rate)</i>	87
Gambar 4.32 Diagram Alir <i>traintest(\$data, \$numEpoch)</i>	88
Gambar 4.33 Diagram Alir <i>result_conv(\$res)</i>	88
Gambar 4.34 Diagram Alir <i>get_hitungakh(\$nipp)</i>	88
Gambar 4.35 Diagram Alir <i>tambah_hitungakh(\$na, \$epoh, \$lr, \$nipp, \$larik)</i> ... 89	89
Gambar 4.36 Diagram Alir <i>hasil_bobot()</i>	89
Gambar 4.37 Diagram Alir <i>hasil_bias()</i>	89
Gambar 4.38 Diagram Alir <i>keterangan()</i>	89
Gambar 5.1 Halaman <i>Login</i> Sistem	113
Gambar 5.2 Halaman Utama Manajer Jalan dan Jembatan	114
Gambar 5.3 <i>Pop-up</i> Proses Tambah Data Lintas	114
Gambar 5.4 <i>Pop-up</i> Proses Menandai Ubah Data Lintas	115
Gambar 5.5 <i>Pop-up</i> Proses Validasi Data Lintas.....	115
Gambar 5.6 Halaman Utama Kepala Sub Direktorat Jalan Rel dan Jembatan Pusat	116
Gambar 5.7 Halaman Utama Asisten Manajer Konstruksi Jalan Rel dan Jembatan	117
Gambar 5.8 Halaman Tampil Peta.....	117
Gambar 5.9 Halaman Tampil <i>Dataset</i>	118
Gambar 5.10 <i>Pop-up</i> Proses Hapus Data dari <i>Dataset</i>	118

Gambar 5.11 Halaman Tambah <i>Dataset</i>	119
Gambar 5.12 Halaman Ubah <i>Dataset</i>	120
Gambar 5.13 Halaman Klasifikasi SLP pada <i>Dataset</i>	120
Gambar 5.14 Halaman <i>Form</i> Nilai Kriteria Jalur	121
Gambar 5.15 Halaman Klasifikasi Data Baru.....	122
Gambar 5.16 Halaman Laporan Peta	123
Gambar 5.17 Tampilan Awal Halaman Klasifikasi SLP pada <i>Dataset</i>	126
Gambar 5.18 Tampilan Halaman Klasifikasi <i>Dataset</i> dengan Kesalahan Berupa Kekosongan Salah Satu <i>Field</i> Parameter SLP	126
Gambar 5.19 Tampilan Halaman Klasifikasi <i>Dataset</i> pada Kasus Pertama.....	127
Gambar 5.20 Tampilan Halaman Klasifikasi <i>Dataset</i> pada Kasus Kedua.....	127
Gambar 5.21 Tampilan Halaman Klasifikasi <i>Dataset</i> pada Kasus Ketiga	129
Gambar 5.22 Tampilan Pengisian <i>Form</i> Nilai Kriteria Ruas Lintas.....	130
Gambar 5.23 Tampilan Halaman Klasifikasi Data Baru dengan Kesalahan Berupa Kekosongan Salah Satu <i>Field Form</i> Nilai Kriteria Ruas Lintas	130

DAFTAR NOTASI

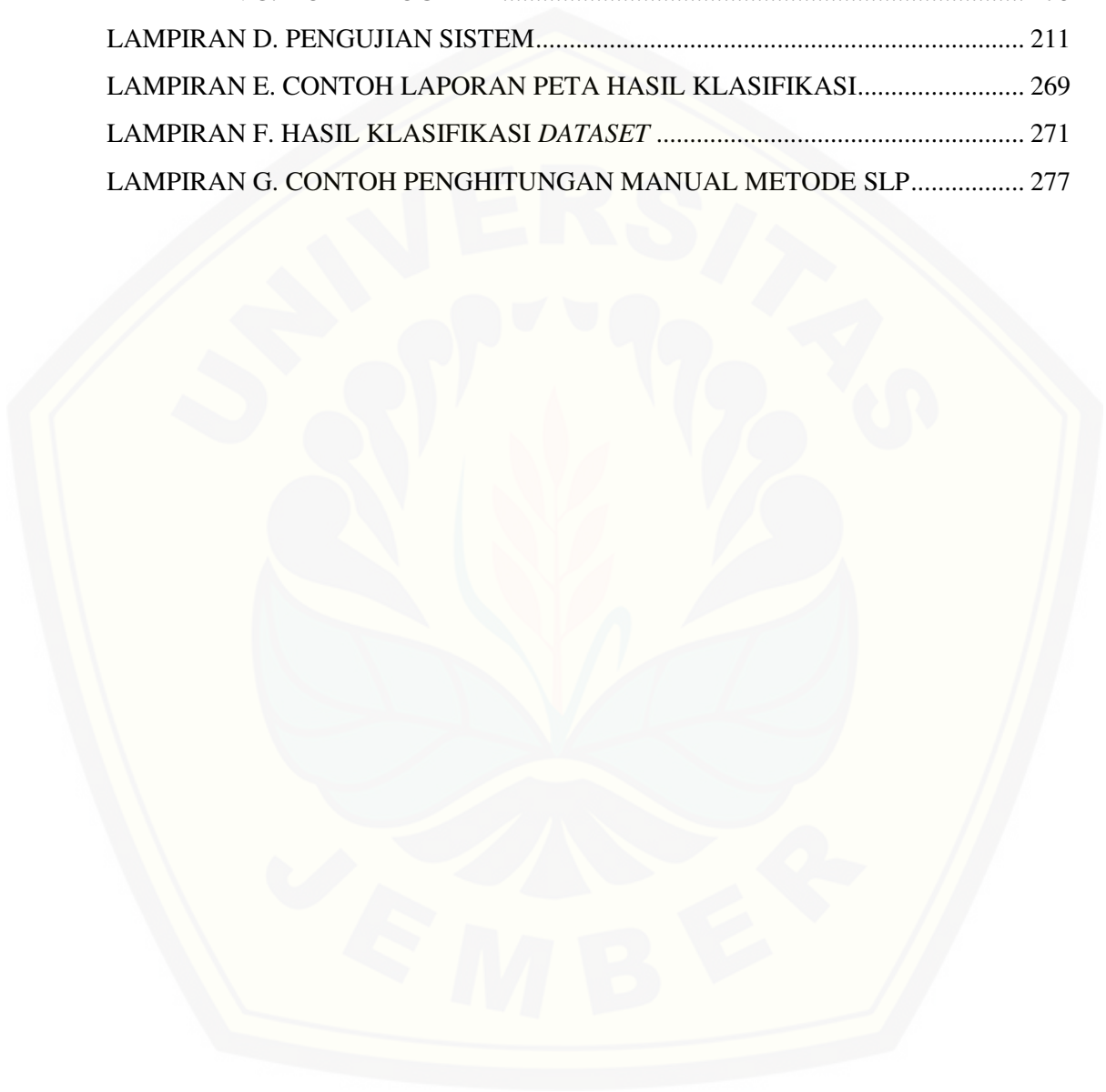
d_1	=	nilai tebal balas atas
b	=	nilai lebar permukaan balas atas
c	=	nilai lebar dasar balas atas
k_1	=	nilai lebar permukaan balas bawah
d_2	=	nilai tebal balas bawah
e	=	nilai tebal antara dasar balas dengan balas bawah
k_2	=	nilai lebar dasar balas bawah
α	=	nilai laju pembelajaran atau <i>learning rate</i>
Θ	=	nilai ambang batas atau <i>threshold</i>
k	=	nilai epoh
q	=	jumlah unit masukan dari jaringan syaraf tiruan
$s^{(q)}$	=	nilai atribut <i>dataset</i> untuk setiap unit masukan
x	=	nilai vektor masukan
y_{in}	=	nilai respon untuk unit <i>output</i>
w	=	nilai bobot
b	=	nilai bias
e	=	bilangan Euler
y	=	nilai keluaran
c	=	nilai kelas atau <i>class</i> pada <i>dataset</i>
t	=	nilai target atau nilai galat / <i>error</i>
$V(G)$	=	nilai <i>cyclomatic complexity</i>
<i>Egde</i>	=	nilai garis penghubung antar-titik pada <i>cyclomatic complexity</i>
<i>Node</i>	=	nilai titik pada <i>cyclomatic complexity</i>

DAFTAR SINGKATAN

km/jam	: kilometer per jam
cm	: sentimeter
mm	: milimeter
m	: meter
kg	: kilogram
UML	: <i>Unified Modeling Language</i>
OOAD	: <i>Object Oriented Analysis Design</i>
XAMPP	: X (Sistem Operasi Windows, Linux, Mac OS, dan Solaris), Apache, MySQL, PHP, dan Perl
PHP	: <i>Page Hypertext Pre-Processor</i>
OOP	: <i>Object-Oriented Programming</i>
MD5	: <i>Message-Direct algoritihm 5</i>
e-Office	: <i>electronic office</i>
NIPP	: Nomor Induk Pegawai Perusahaan
PT	: Perseroan Terbatas
KAI	: Kereta Api Indonesia
CRUD	: <i>Create, Read, Update, dan Delete</i>
DAOP	: daerah operasi
DIVRE	: divisi regional
TBA	: tebal balas atas
LBB	: lebar bahu balas
LPM	: landai penentu maksimum
JMPH	: jari-jari minimal horizontal peralihan
LBJ	: lebar badan jalan

DAFTAR LAMPIRAN

	Halaman
LAMPIRAN A. <i>DATASET</i> PENELITIAN	140
LAMPIRAN B. PERANCANGAN SISTEM.....	156
LAMPIRAN C. KODE PROGRAM	178
LAMPIRAN D. PENGUJIAN SISTEM.....	211
LAMPIRAN E. CONTOH LAPORAN PETA HASIL KLASIFIKASI.....	269
LAMPIRAN F. HASIL KLASIFIKASI <i>DATASET</i>	271
LAMPIRAN G. CONTOH PENGHITUNGAN MANUAL METODE SLP.....	277



BAB 1. PENDAHULUAN

Bab ini merupakan tahapan awal dari penulisan tugas akhir ini. Bab ini memuat uraian tentang latar belakang, perumusan masalah, tujuan dan manfaat, batasan masalah, dan sistematika penulisan.

1.1 Latar Belakang

Jalur Kereta Api Klakah-Pasirian merupakan jalur kereta api yang menghubungkan antara Stasiun Klakah dengan Stasiun Pasirian. Jalur tersebut merupakan jalur non-aktif yang akan direncanakan aktif sesuai dengan Peta Rencana Induk Perkeretaapian Nasional atau RIPNAS tahun 2030 sehingga perlu adanya perencanaan dalam proses aktivasi kembali jalur ini (Ditjen Perkeretaapian Kementerian Perhubungan, 2011). Selain itu, jalur tersebut merupakan solusi yang paling potensial untuk mengurangi berbagai dampak negatif dari pendistribusian pasir menggunakan moda transportasi truk, seperti kemacetan dan kerusakan di jalan raya Lumajang-Probolinggo (Arifianto & Prastyanto, 2012). Oleh karena itu, dibutuhkan perencanaan konstruksi jalur kereta api pada Jalur Kereta Api Klakah-Pasirian yang dalam hal ini dilakukan penelitian terhadap penentuan nilai kecepatan maksimum.

Berdasarkan Peraturan Dinas No. 10 Tahun 1986 Perusahaan Jawatan Kereta Api atau PD 10 mengenai peraturan konstruksi jalan rel, kecepatan maksimum kereta api yang selanjutnya disebut kecepatan maksimum merupakan kecepatan tertinggi yang diizinkan untuk operasi suatu rangkaian kereta api pada lintas tertentu. Selain itu, kecepatan maksimum merupakan salah satu parameter yang digunakan dalam perencanaan konstruksi jalur kereta api selain jumlah beban, beban gandar, dan pola operasi. Perencanaan konstruksi jalur kereta api tersebut dilakukan agar pembangunan dan pemeliharaan konstruksi dapat diselenggarakan dengan kualitas keluaran atau *output* terbaik dan tetap menjamin keamanan dan kenyamanan dalam pengoperasian suatu rangkaian kereta api. Penentuan kecepatan maksimum memerlukan beberapa data kriteria kecepatan maksimum seperti klasifikasi jalan rel, lebar badan jalan rel, jari-jari

minimum lengkung horizontal, landai penentu maksimum, jari-jari minimal lengkung vertikal, dan penampang melintang jalan rel (Dwiatmoko, 2013).

Pada penelitian terdahulu yang berjudul “Desain Geometrik, Struktur beserta Perkiraan Biaya Perencanaan Jalan Rel sebagai Alternatif Transportasi Angkutan Tambang Pasir di Kabupaten Lumajang” didapati bahwa kecepatan maksimum pada Jalur Kereta Api Klakah-Pasirian adalah 120 km/jam. Namun, didapati pula bahwa tebal balas atas yang didesain sebesar 55 cm. Jika ditinjau pada PD 10, khususnya pada Tabel Klasifikasi Jalan Rel, maka kecepatan maksimum suatu jalan rel yang memiliki nilai tebal balas atas 55 cm tidak dapat ditentukan nilainya. Selain itu, tidak terdapat rumus matematis kecepatan maksimum dengan variabel yang menggunakan atribut-atribut kriteria kecepatan maksimum pada PD 10.

Pada penelitian ini, metode klasifikasi yang akan digunakan adalah metode *Single Layer Perceptron*. Hal ini disesuaikan dengan jenis atribut pada *dataset* kecepatan maksimum, yaitu jenis nominal dan jenis numerik, untuk digunakan pada penelitian ini, antara lain tipe rel, jenis bantalan, jenis penambat, tebal balas atas, lebar bahu balas, kelandaian, jari-jari minimal vertikal, jari-jari minimal horizontal tanpa peralihan, jari-jari minimal horizontal peralihan, lebar badan jalan, dan ruang bebas. Metode ini merupakan salah satu arsitektur *perceptron* dari Jaringan Syaraf Tiruan atau JST yang paling sederhana karena hanya memiliki satu lapisan dengan bobot yang selanjutnya menerima masukan kemudian secara langsung akan mengolahnya menjadi keluaran tanpa harus melalui lapisan tersembunyi (Wuryandari & Afrianto, 2012). JST sendiri memiliki beberapa kelebihan seperti mampu mengakuisisi pengetahuan walau tidak ada kepastian, melakukan generalisasi dan ekstraksi dari suatu pola data tertentu, dan dapat menciptakan suatu pola pengetahuan melalui pengaturan diri atau kemampuan belajar (Atmawati, 2011). Sehingga, untuk lebih mengoptimalkan proses dan hasil penghitungan tersebut perlu dilakukan penelitian untuk men-sistem-kan penghitungan tersebut secara terkomputerisasi. Didasari atas berbagai data-data kriteria kecepatan maksimum yang selanjutnya diolah menjadi *dataset* kecepatan maksimum,

maka penelitian ini mengajukan sistem pengklasifikasi kecepatan maksimum kereta api.

1.2 Perumusan Masalah

Bertitik tolak dari latar belakang tersebut, maka dapat dirumuskan masalahnya sebagai berikut:

1. Bagaimana cara menentukan kecepatan maksimum pada Jalur Kereta Api Klakah-Pasirian menggunakan metode *Single Layer Perceptron*?
2. Bagaimana cara memetakan kecepatan maksimum pada Jalur Kereta Api Klakah-Pasirian?

1.3 Tujuan dan Manfaat

Dari perumusan masalah tersebut, maka dapat diambil tujuan dan manfaat penelitian ini sebagai berikut:

1.3.1 Tujuan

Tujuan dari penelitian ini adalah:

1. Memahami tentang cara menentukan kecepatan maksimum pada Jalur Kereta Api Klakah-Pasirian menggunakan metode *Single Layer Perceptron*.
2. Memahami tentang cara memetakan kecepatan maksimum pada Jalur Kereta Api Klakah-Pasirian.

1.3.2 Manfaat

Manfaat dari penelitian ini adalah:

1. Memberikan informasi tentang cara menentukan kecepatan maksimum pada Jalur Kereta Api Klakah-Pasirian menggunakan metode *Single Layer Perceptron*.
2. Memberikan informasi tentang cara memetakan kecepatan maksimum pada Jalur Kereta Api Klakah-Pasirian.

1.4 Batasan Masalah

Batasan masalah yang ditentukan dalam penelitian ini sebagai berikut:

1. Studi kasus yang digunakan adalah Jalur Kereta Api Klakah-Pasirian dengan kondisi jalur 100%.
2. Nilai kecepatan maksimum pada ruas lintas dengan nilai jari-jari minimum lengkung lingkaran tanpa lengkung peralihan 810-1050 meter adalah 70 km/jam, jari-jari minimum lengkung lingkaran dengan lengkung peralihan 270-350 meter adalah 70 km/jam, jari-jari minimum lengkung lingkaran tanpa lengkung peralihan 600-810 meter adalah 60 km/jam, jari-jari minimum lengkung lingkaran dengan lengkung peralihan 200-270 meter adalah 60 km/jam, jembatan dengan ruang bebas hingga 4,5 meter adalah 60 km/jam, dan viaduk dengan ruang bebas hingga 4,7 meter adalah 60 km/jam.
3. Tidak ditentukan posisi sinyal masuk dan sinyal keluar pada setiap stasiun serta semboyan-semboyan pembatas kecepatan maksimum.
4. Pembuatan sistem tersebut berbasis *website* menggunakan *framework* CodeIgniter.
5. Petaan yang digunakan adalah Mapbox dengan pustaka petaan adalah Leaflet.

1.5 Sistematika Penulisan

1. Pendahuluan

Bab ini memuat tentang latar belakang, perumusan masalah, tujuan dan manfaat, batasan masalah, dan sistematika penulisan skripsi ini.

2. Tinjauan Pustaka

Bab ini memuat tentang tinjauan terhadap hasil penelitian terdahulu, kajian materi, perumusan premis, dan perumusan hipotesis. Kajian materi yang dimaksud meliputi algoritme klasifikasi, kecepatan maksimum kereta api, Jalur Kereta Api Klakah-Pasirian, metode *Single Layer Perceptron* sebagai metode klasifikasi, dan model *Waterfall* sebagai model siklus hidup pengembangan perangkat lunak.

3. Metode Penelitian

Bab ini memuat tentang metode yang diterapkan dalam penelitian ini. Dalam hal ini, metode yang dimaksud adalah metode perancangan dan pembangunan sistem yang meliputi jenis penelitian, tempat dan waktu penelitian, perancangan penelitian, teknik perolehan data, analisis kebutuhan sistem, perancangan sistem, implementasi sistem, pengujian sistem, dan pemeliharaan sistem.

4. Perancangan dan Pembangunan Sistem

Bab ini memuat tentang gambaran umum, kebutuhan, perancangan, pembangunan, pengujian, dan pemeliharaan sistem. Kebutuhan sistem meliputi kebutuhan fungsional dan kebutuhan non-fungsional. Perancangan sistem menggunakan model OOAD yang meliputi *business process*, *use case diagram*, *use case scenario*, *activity diagram*, *sequence diagram*, *class diagram*, dan *entity relationship diagram* atau ERD. Pembangunan sistem meliputi *coding* yang didasarkan atas perancangan sistem. Pengujian sistem meliputi pengujian *white box* dan *black box* serta pengujian validasi. Pemeliharaan sistem meliputi penjelasan jika di kemudian hari ditemukan kesalahan-kesalahan sistem atau *bug*.

5. Hasil dan Pembahasan

Bab ini memuat tentang hasil pembuatan sistem dan pembahasan dari penelitian yang telah dilakukan untuk mendapati penerapan metode *Single Layer Perceptron* sebagai metode klasifikasi kecepatan maksimum kereta api pada sebuah sistem informasi.

6. Penutup

Bab ini memuat tentang kesimpulan dari penelitian yang telah dilakukan dan saran untuk perlakuan dan penelitian selanjutnya.

BAB 2. TINJAUAN PUSTAKA

Bab ini merupakan tinjauan terhadap hasil-hasil penelitian terdahulu yang berkaitan dengan masalah yang dibahas, kajian teori yang berkaitan dengan masalah seperti algoritme klasifikasi, kecepatan maksimum, jalur kereta api Klakah-Pasirian, dan metode *single layer perceptron*, perumusan premis yang didapat dari hasil-hasil penelitian terdahulu dan teori-teori tersebut, serta perumusan hipotesis sebagai akhir dari kajian teori.

2.1 Penelitian Terdahulu

Penelitian terdahulu yang digunakan pada penelitian ini berjudul “Desain Geometrik, Struktur beserta Perkiraan Biaya Perencanaan Jalan Rel sebagai Alternatif Transportasi Angkutan Tambang Pasir di Kabupaten Lumajang” oleh Dodik Teguh Arifianto dari Jurusan Teknik Sipil, Fakultas Teknik Sipil dan Perencanaan, Institut Teknologi Sepuluh Nopember (ITS). Beberapa tujuan yang didapat pada penelitian terdahulu ini antara lain merencanakan desain geometri jalan rel kereta api yang sesuai dengan persyaratan dan menghitung konstruksi jalan rel yang digunakan. Dalam abstrak jurnal penelitian terdahulu ini, disebutkan bahwa Jalur Kereta Api Klakah-Pasirian dirancang dengan nilai kecepatan rencana maksimum 120 km/jam. Selain itu, disebutkan pula bahwa tebal balas atas adalah 55 cm. Jika ditinjau pada PD 10, khususnya pada Tabel Klasifikasi Jalan Rel, maka tidak dapat ditentukan nilai kecepatan maksimum jika dalam perancangan menentukan nilai tebal balas atas sebesar 55 cm. Selain itu, pada PD 10 pula tidak ditemukan rumus matematis kecepatan maksimum dengan variabel yang menggunakan atribut-atribut kriteria kecepatan maksimum sehingga diperlukan suatu penghitungan yang dapat menentukan kecepatan maksimum sesuai dengan atribut-atribut kriteria kecepatan maksimum.

2.2 Algoritme Klasifikasi

Klasifikasi merupakan salah satu algoritme *data mining* yang merupakan proses menemukan model atau pola yang nantinya digunakan untuk memprediksi kelas pada data-data yang telah ditentukan pada suatu *dataset*. Dalam proses klasifikasi, terdapat dua tahap yaitu tahap *training* dan *testing* di mana *dataset* tersebut dipecah menjadi dua bagian, yaitu data *training* dan data *testing*. Pada tahap *training*, dilakukan analisis data *training* lalu direpresentasikan ke dalam aturan atau model klasifikasi. Sedangkan pada tahap *testing*, diterapkan model klasifikasi dari tahap *training* untuk memperkirakan akurasi dari model klasifikasi tersebut pada data *testing* sehingga didapat nilai akurasi atau nilai ketepatan antara *class* asli dengan *class* hasil prediksi tersebut (Raharja, 2014).

Pada proses klasifikasi, terdapat empat unsur yang digunakan, yaitu *class*, *predictor*, data *training*, dan data *testing*. *Class* merupakan variabel terikat yang bertipe kategorikal dan mempresentasikan label dari setiap data pada *dataset*. *Predictor* merupakan variabel bebas yang direpresentasikan oleh karakteristik dari *class*. Istilah lain dari *predictor* adalah atribut-atribut data. Data *training* merupakan bagian dari *dataset* yang digunakan untuk mempelajari model dalam menentukan kelas berdasarkan *predictor*. Data *testing* merupakan bagian dari *dataset* yang digunakan untuk menentukan kelas di mana dalam menentukan klasifikasi tersebut menggunakan model yang telah didapat sebelumnya (Raharja, 2014).

2.3 Kecepatan Maksimum

Kecepatan maksimum kereta api atau selanjutnya disebut kecepatan maksimum merupakan kecepatan tertinggi yang diizinkan dalam pengoperasian kereta api pada suatu lintasan tertentu. Kecepatan maksimum digunakan untuk mengejar keterlambatan yang terjadi akibat hal-hal yang mengganggu perjalanan kereta api (Kartikasari & Widyani, 2007). Kecepatan maksimum memiliki berbagai data berupa tabel dan hasil pengolahan berbagai data berupa tabel tersebut ke dalam *dataset* penelitian sebagai berikut:

Tabel 2.1 Kelas Jalan Rel

Kelas Jalan	Daya Angkut Lintas (ton/tahun)	Vmaks (km/jam)	P maks gandar (ton)	Tipe Rel	Jenis Bantalan	Jenis Penambat	Tebal Balas Atas (cm)	Lebar Bahu Balas (cm)
					Jarak (As)			
I	$>20.10^6$	120	18	R.60/R54	Beton 600	EG	30	60
II	$10.10^6 - 20.10^6$	110	18	R.54/R.50	Beton/Kayu 600	EG	30	50
III	$5.10^6 - 10.10^6$	100	18	R.54/R.50/ R.42	Beton/Kayu /Baja 600	EG	30	40
IV	$2.5.10^6 - 5.10^6$	90	18	R.54/R.50/ R.42	Beton/Kayu /Baja 600	EG/ET	25	40
V	$<2.5.10^6$	80	18	R.42	Kayu/Baja 600	ET	25	35

Sumber: (Dwiatmoko, 2013)

Tabel 2.2 Lebar Badan Jalan Rel

Kecepatan Maksimum Desain	Lebar Jalan Rel (meter)
120 km/jam dan 110 km/jam jalur	3,15 (3,00)
100 km/jam jalur	2,95 (2,85)
90 km/jam jalur	2,85 (2,75)
80 km/jam jalur	2,50 (2,40)

Sumber: (Dwiatmoko, 2013)

Tabel 2.3 Jari-jari Minimum Lengkung Horizontal

Kecepatan Rencana (km/jam)	Jari-jari minimum lengkung lingkaran tanpa lengkung peralihan (m)	Jari-jari minimum lengkung lingkaran yang diijinkan dengan lengkung peralihan (m)
120	2370	780
110	1990	660
100	1650	550
90	1330	440
80	1050	350
70	810	270
60	600	200

Sumber: (Dwiatmoko, 2013)

Tabel 2.4 Jari-jari Minimal Lengkung Vertikal

Kecepatan Rencana (km/jam)	Jari-jari Minimum Lengkung Vertikal (m)
Lebih besar dari 100	8000
Sampai 100	6000

Sumber: (Dwiatmoko, 2013)

Tabel 2.5 Landai Penentu Maksimum

Kelas Jalan Rel	Landai Penentu Maksimum
1	10 ‰
2	10 ‰
3	20 ‰
4	25 ‰
5	25 ‰

Sumber: (Dwiatmoko, 2013)

Tabel 2.6 Penampang Melintang Jalan Rel

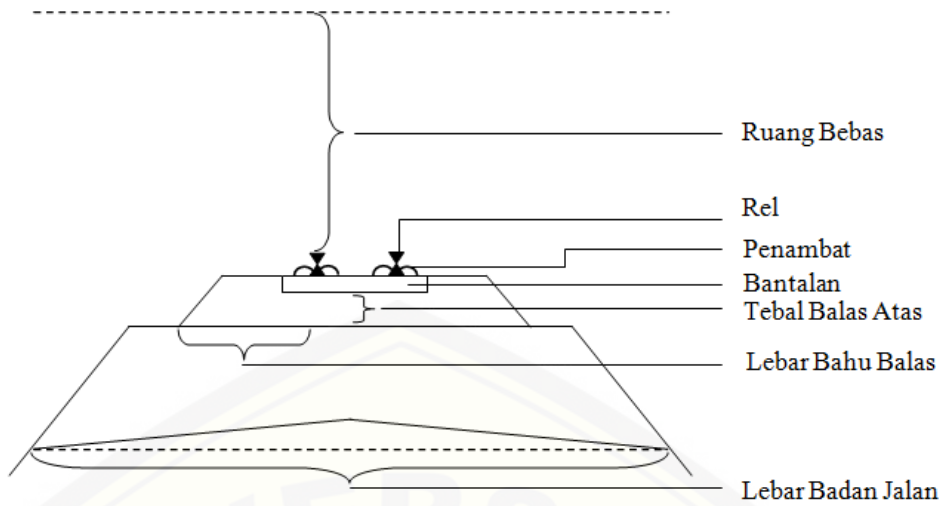
Kelas Jalan	Vmaks (km/jam)	d ₁ (cm)	b (cm)	c (cm)	k ₁ (cm)	d ₂ (cm)	e (cm)	k ₂ (cm)
I	120	30	150	235	265	15-50	25	375
II	110	30	150	235	265	15-50	25	375
III	100	30	140	225	240	15-50	22	325
IV	90	25	140	215	240	15-35	20	300
V	80	25	135	210	240	15-35	20	300

Sumber: (Dwiatmoko, 2013)

Dataset kecepatan maksimum yang digunakan pada penelitian ini merupakan hasil penggabungan dari beberapa atribut pada tabel-tabel di atas. Atribut-atribut yang digunakan pada *dataset* penelitian tersebut antara lain antara lain tipe rel, jenis bantalan, jenis penambat, tebal balas atas, lebar bahu balas, kelandaian, jari-jari minimal vertikal, jari-jari minimal horizontal tanpa peralihan, jari-jari minimal horizontal peralihan, lebar badan jalan, dan ruang bebas. Sedangkan atribut-atribut lain pada tabel-tabel di atas tersebut tidak digunakan karena *dataset* penelitian ini merupakan *dataset* prediksi. Tabel *dataset* kecepatan maksimum dapat ditinjau pada lampiran A.

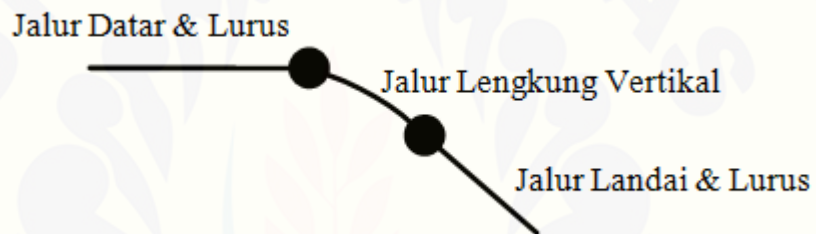
Atribut-atribut nominal memiliki perlakuan tersendiri, yaitu pembagian menjadi atribut-atribut baru untuk menghasilkan nilai-nilai data yang berjenis numerik sehingga dapat diolah pada klasifikasi dari penelitian ini. Atribut tipe rel terbagi menjadi empat atribut, yaitu R.42, R.50, R.54, R.60. Pembagian atribut tipe rel didasarkan atas berat rel dalam setiap meter, contoh R.42 yang berarti rel tersebut memiliki berat 42 kg dengan panjang satu meter. Atribut bantalan terbagi menjadi tiga atribut, yaitu beton, baja, dan kayu. Pembagian atribut bantalan didasarkan atas bahan dari bantalan rel kereta api. Atribut penambat terbagi menjadi dua atribut, yaitu tunggal dan ganda. Pembagian atribut penambat didasarkan atas dua macam penambat yang digunakan untuk mengikat rel pada bantalan rel kereta api.

Sedangkan untuk atribut-atribut lain yang merupakan atribut numerik antara lain tebal balas atas, lebar bahu balas, kelandaian, jari-jari minimal vertikal, jari-jari minimal horizontal tanpa peralihan, jari-jari minimal horizontal peralihan, lebar badan jalan, dan ruang bebas. Atribut tebal balas atas merupakan nilai tebal pada balas bagian atas. Atribut lebar bahu balas merupakan nilai jarak dari ujung balas ke rel. Atribut kelandaian merupakan nilai kelandaian jalur lurus dari jalur datar ke jalur miring. Atribut jari-jari minimal vertikal merupakan nilai jari-jari dari lengkung vertikal suatu jalur, baik terhubung dengan peralihan maupun jalur lurus. Atribut jari-jari minimal horizontal tanpa peralihan merupakan nilai jari-jari dari lengkung horizontal yang tidak memiliki peralihan sehingga kedua ujung jalur terhubung langsung dengan jalur lurus. Atribut jari-jari minimal horizontal peralihan merupakan nilai jari-jari dari dari lengkung horizontal yang memiliki peralihan sehingga kedua ujung jalur terhubung tidak langsung dengan jalur lurus atau melalui peralihan dahulu. Atribut lebar badan jalan merupakan nilai lebar dari badan jalan rel. Atribut ruang bebas merupakan nilai jarak dari permukaan atas rel hingga batas ruang yang ditentukan.



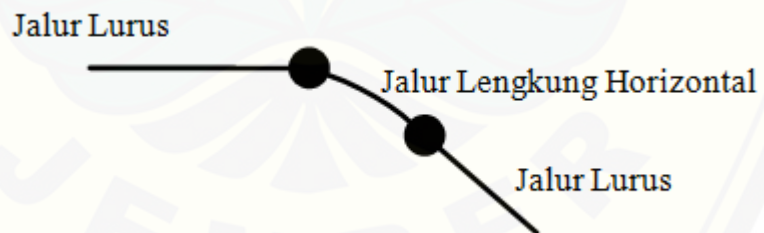
Gambar 2.1 Bentuk Penampang Jalan Rel secara Melintang

Sumber: (Hasil Analisis, 2017)



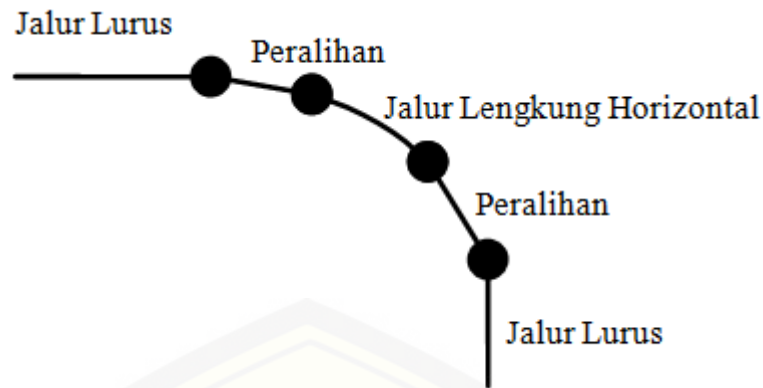
Gambar 2.2 Jalur dengan Lengkung Vertikal

Sumber: (Hasil Analisis, 2017)



Gambar 2.3 Jalur dengan Lengkung Horizontal

Sumber: (Hasil Analisis, 2017)

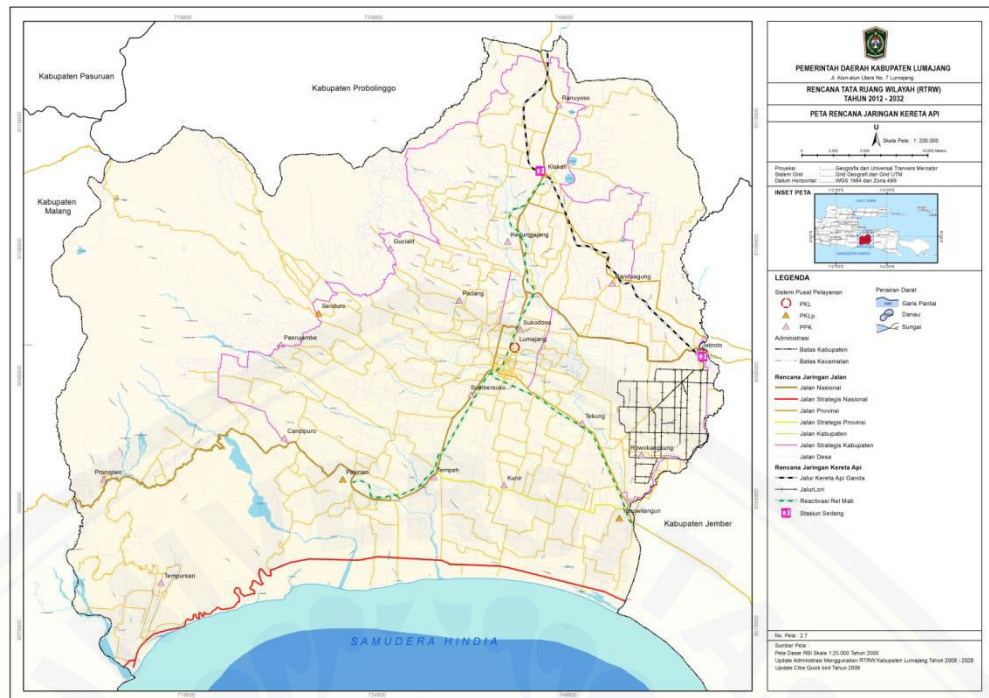


Gambar 2.4 Jalur dengan Lengkung Horizontal Peralihan

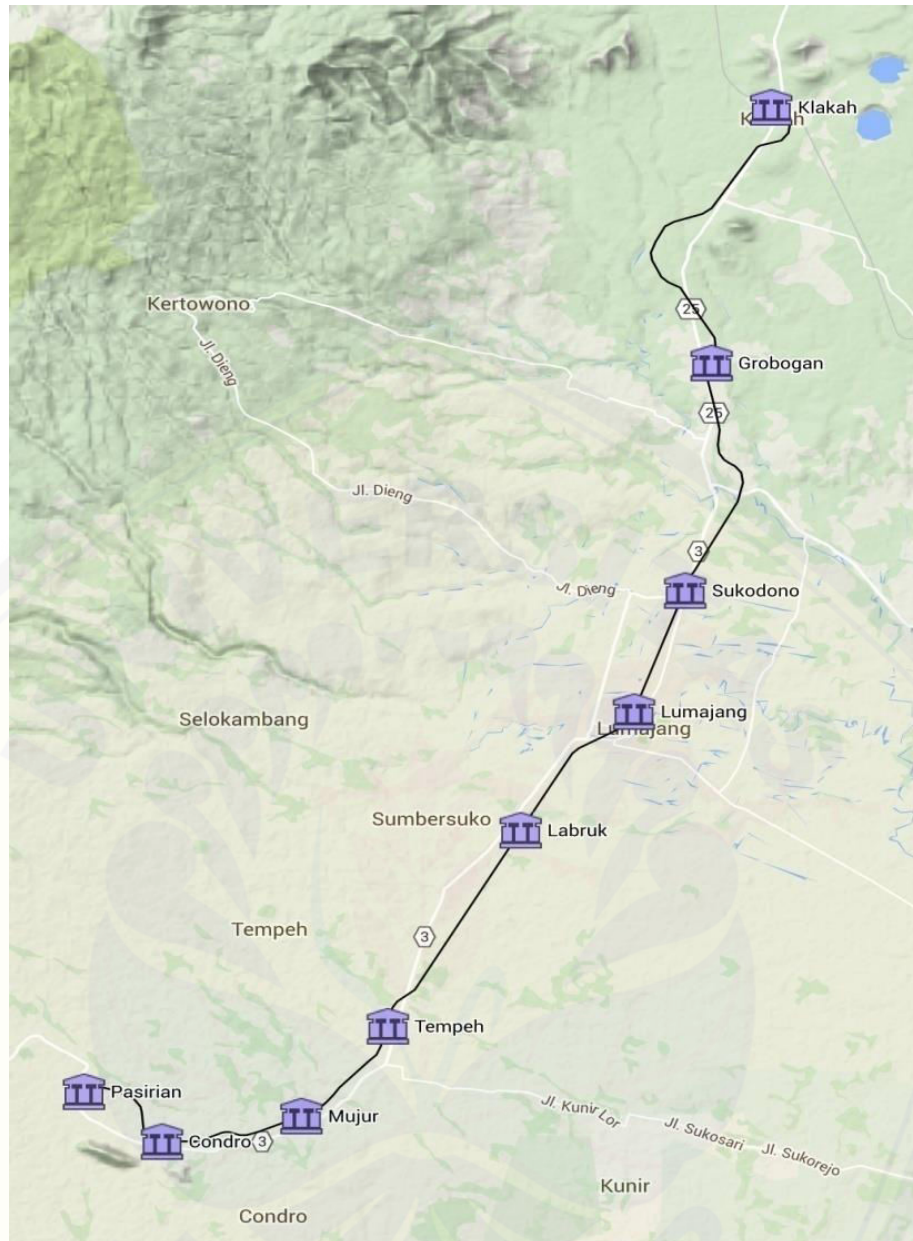
Sumber: (Hasil Analisis, 2017)

2.4 Jalur Kereta Api Klakah-Pasirian

Jalur Kereta Api Klakah-Pasirian merupakan jalur kereta api yang menghubungkan antara Stasiun Klakah dengan Stasiun Pasirian. Jalur kereta api ini berstatus jalur non-aktif sejak tahun 1987. Jalur Kereta Api Klakah-Pasirian memiliki panjang jalur 36,2 km. Secara keseluruhan, lokasi jalur kereta api ini terletak di wilayah kabupaten Lumajang, Jawa Timur sebagaimana pada gambar 2.5. Jalur ini melewati berbagai wilayah pemukiman atau kecamatan, antara lain Klakah, Kedungjajang, Sukodono, Lumajang, Sumpalsoko, Tempeh, dan Pasirian (Arifianto & Prastyanto, 2012). Semasa aktif pada jalur yang mulai dibuka pada tanggal 16 Mei 1896 ini, stasiun-stasiun yang beroperasi antara lain Klakah, Grobogan, Sukodono, Lumajang, Labruk, Tempeh, Mujur, Condro, dan Pasirian sebagaimana pada gambar 2.6.



Gambar 2.5 Peta Rencana Jaringan Kereta Api Kabupaten Lumajang
 Sumber: (Badan Perencanaan dan Pembangunan Daerah Kabupaten Lumajang, 2012)



Gambar 2.6 Peta Jalur Kereta Api Klakah-Pasirian

Sumber: (Hasil Analisis, 2015)

Dari hasil penelitian terdahulu tersebut (Arifianto & Prastyanto, 2012) didapatkan desain struktur dari Jalur Kereta Api Klakah-Pasirian sebagai berikut:

1. Desain menggunakan kelas jalan I
2. Tipe rel adalah R.54
3. Kapasitas angkut lintas sebesar lebih dari 20 juta ton/tahun

4. Beban gandar sebesar 18 ton
5. Lebar sepur sejumlah 1067 mm
6. Jarak bantalan beton sejumlah 50 cm (maksimal 60 cm)
7. Tebal balas di bawah bantalan sejumlah 25 cm (maksimal 30 cm)
8. Lebar bahu balas sejumlah 50 cm
9. Tipe penambat adalah elastik ganda
10. Sambungan adalah las ditempat
11. Tebal balas atas sejumlah 55 cm
12. Tebal balas bawah (sub balas) sejumlah 21 cm
13. Jarak dari sumbu jalan rel ke tepi atas lapisan bawah pada sepur lurus sejumlah 200 cm
14. Jarak dari sumbu jalan rel ke tepi atas lapisan bawah pada tikungan sejumlah 260 cm.

Dari hasil *tracking* yang telah dilakukan secara tidak langsung melalui aplikasi ArcGIS dengan Peta Rencana Jaringan Kereta Api Kabupaten Lumajang yang berbasis vektor, maka didapat hasil sebagai berikut:

Tabel 2.7 Data Lintas Klakah-Pasirian

No	Area Ruas Lintas	Bentuk Rel	Nilai Landai Penentu	Jari-jari Minimal (m)	Lebar Badan Jalan (m)
1	0+000 – 0+130	Lurus	20‰	-	0,02
2	0+130 – 1+692	Kumpulan Lengkung Horizontal Peralihan	20‰	400	0,026
3	1+692 – 3+412	Lurus	20‰	-	0,02
4	3+412 – 4+245	Kumpulan Lengkung Horizontal Peralihan	20‰	400	0,026
5	4+245 – 4+303	Jembatan	0	-	0,02
6	4+303 – 7+039	Kumpulan Lengkung Horizontal Peralihan	20‰	400	0,026
7	7+039 – 8+878	Lengkung Horizontal Peralihan	20‰	500	0,026
8	8+878 – 9+812	Lurus	20‰	-	0,02
9	9+812 – 11+092	Lengkung Horizontal Peralihan	20‰	665	0,026
10	11+092 – 12+154	Lengkung Horizontal Peralihan	20‰	500	0,026
11	12+154 – 12+180	Viaduk	0	-	0,02

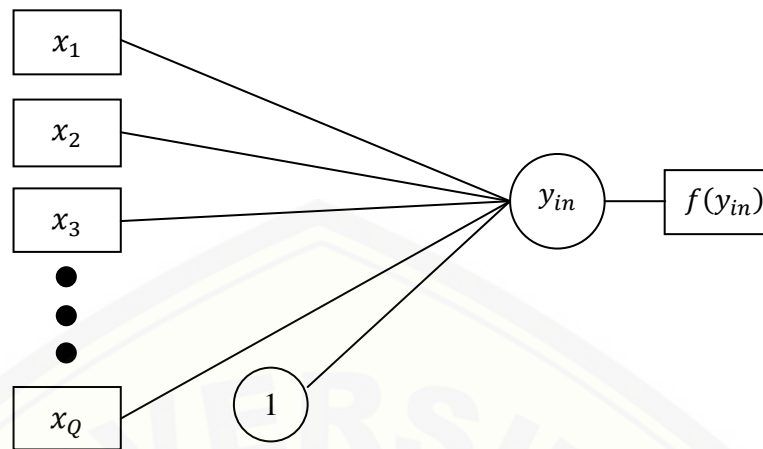
12	12+180 – 12+340	Lurus	10‰	-	0,02
13	12+340 – 12+374	Jembatan	0	-	0,02
14	12+374 – 14+160	Lurus	10‰	-	0,02
15	14+160 – 14+274	Lengkung Horizontal Peralihan	10‰	665	0,026
16	14+274 – 18+044	Lurus	0	-	0,02
17	18+044 – 18+447	Lengkung Horizontal Peralihan	0	500	0,02
18	18+447 – 18+856	Lurus	0	-	0,02
19	18+856 – 18+880	Jembatan	0	-	0,02
20	18+880 – 19+422	Lurus	0	-	0,02
21	19+422 – 19+614	Lengkung Horizontal Peralihan	0	665	0,02
22	19+614 – 26+384	Lurus	10‰	-	0,02
23	26+384 – 27+103	Kumpulan Lengkung Horizontal Peralihan	10‰	500	0,02
24	27+103 – 28+005	Lurus	10‰	-	0,02
25	28+005 – 28+472	Lengkung Horizontal Peralihan	10‰	450	0,02
26	28+472 – 29+651	Lurus	10‰	-	0,02
27	29+651 – 29+872	Jembatan	0	-	0,02
28	29+872 – 30+065	Lengkung Horizontal Peralihan	10‰	500	0,02
29	30+065 – 31+535	Lurus	10‰	-	0,02
30	31+535 – 32+345	Kumpulan Lengkung Horizontal Peralihan	10‰	500	0,02
31	32+345 – 32+432	Jembatan	0	-	0,02
32	32+432 – 32+706	Lengkung Horizontal Peralihan	10‰	500	0,02
33	32+706 – 32+943	Lurus	10‰	-	0,02
34	32+943 – 34+003	Lengkung Vertikal	10‰	385	0,026
35	34+003 – 34+933	Kumpulan Lengkung Horizontal Peralihan	10‰	385	0,02
36	34+933 – 36+200	Kumpulan Lengkung Horizontal Peralihan	10‰	500	0,02

Sumber: (Hasil Analisis, 2016)

2.5 Metode *Single Layer Perceptron*

Single Layer Perceptron atau SLP merupakan salah satu arsitektur dari Jaringan Syaraf Tiruan atau JST di mana hanya memiliki satu lapisan dengan bobot terhubung yang selanjutnya menerima masukan kemudian secara langsung akan mengolahnya menjadi keluaran tanpa harus melalui lapisan tersembunyi (Wuryandari & Afrianto, 2012). Algoritme ini terdiri

atas lapisan nilai *input*, pembobotan, lapisan *output*, dan nilai *output* seperti pada gambar berikut:



Gambar 2.7 Arsitektur *Single Layer Perceptron*

Sumber: (Hasil Analisis, 2016)

Untuk menentukan nilai kemelesetan atau nilai *error* pada metode SLP ini, maka digunakan *delta learning rule* dengan fungsi aktivasi *unipolar sigmoidal* (Fuller, 2010). Menurut (Leung, 2008), langkah-langkah dalam algoritme SLP yang menyesuaikan dengan *delta learning rule* sebagai berikut:

1. Mengatur nilai *learning rate* (α) lebih dari 0 dan *threshold* θ sama dengan 0. Inisialisasi nilai bobot (untuk unit masukan) dan bias (nilai unit masukan = 1) sama dengan 0 atau beberapa nilai secara acak.
2. Mengatur aktivasi untuk vektor masukan $x = s^{(q)}$ (1)
3. Menghitung respon untuk unit *output*:

$$y_{in} = \sum_q x \cdot w_q + b \text{(2)}$$

4. Menghitung nilai keluaran dan nilai kemelesetan:

$$y = f(y_{in}) = \frac{1}{1 + e^{-y_{in}}} \text{(3)}$$

$$t = c - y \text{(4)}$$

5. Jika klasifikasi sesuai atau $y \neq t$ atau nilai $t = \theta$ pada langkah 5, maka kondisi berhenti akan bernilai “benar”, namun kondisi berhenti akan bernilai “salah” jika

nilai $t = \theta$ sehingga dilakukan perbaikan bobot dan bias pola jika terjadi kesalahan:

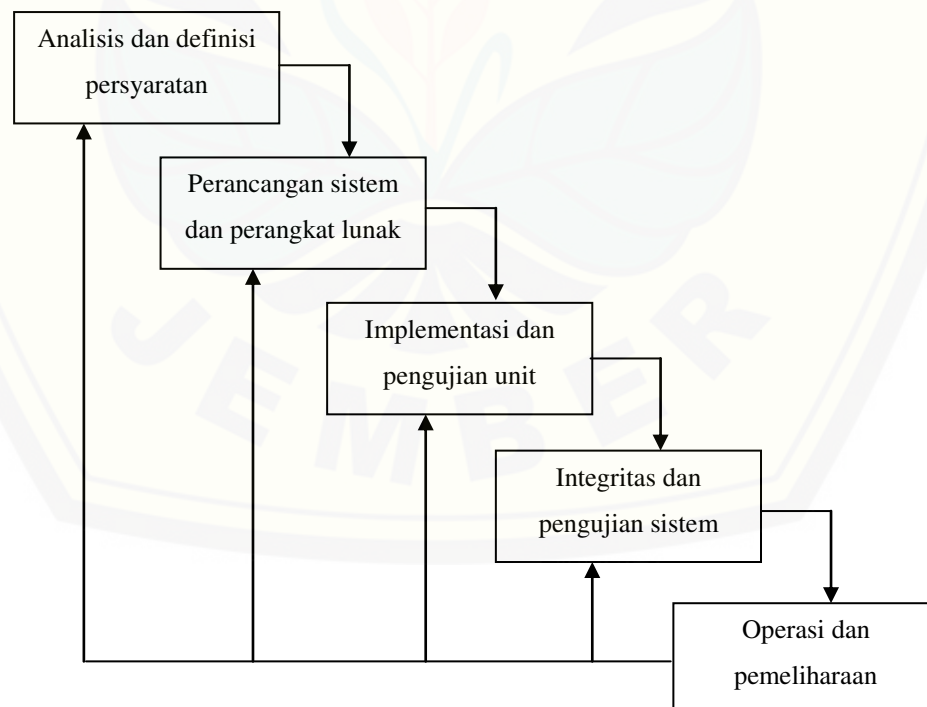
$$w^{baru} = w^{lama} + \alpha \cdot t^{(q)} \cdot x \dots\dots\dots(5)$$

$$b^{baru} = b^{lama} + \alpha \cdot t^{(q)} \dots\dots\dots(6)$$

6. Ulangi langkah-langkah 2 – 6 saat pengulangan dengan batasan nilai epoch k dilakukan untuk memproses *train-test* dengan mengatur *dataset* yang memiliki nilai unit masukan $q = 1, 2, 3 \dots, Q$ hingga proses tersebut menghasilkan nilai kelas yang tepat.

2.6 Model Waterfall

Model *Waterfall* merupakan model pengembangan perangkat lunak yang memiliki pendekatan secara sistematis dan sekuensial untuk pengembangan perangkat lunak yang berjalan melalui tahapan demi tahapan. Berikut merupakan gambar dari model *Waterfall*:



Gambar 2.8 Model *Waterfall*

Sumber: (Suryani, Sasongko, & Suharto, 2012)

Langkah-langkah pada model *Waterfall* sebagai berikut:

1. Analisis dan definisi persyaratan

Pada tahap ini, ditentukan berbagai persyaratan atau kebutuhan yang telah dianalisis sebelumnya, baik kebutuhan fungsional maupun kebutuhan non-fungsional. Selain itu, data yang dibutuhkan dalam penelitian ini adalah data geospasial berupa peta, citra satelit, dan tabel statistika.

2. Perancangan sistem dan perangkat lunak

Pada tahap ini, ditentukan *business process*, *use case diagram*, skenario, *activity diagram*, *sequence diagram*, *class diagram*, dan *entity relationship diagram*. Perangkat perancangan sistem dan perangkat lunak yang dibutuhkan adalah Visual Paradigm for UML 8.0.

3. Implementasi dan pengujian unit

Pada tahap ini, ditentukan implementasi dalam pembuatan sistem berbasis web menggunakan *framework* CodeIgniter dan sistem manajemen basis data menggunakan PostgreSQL dengan memanfaatkan aplikasi XAMPP sebagai *localhost server* untuk Apache. Pengujian unit dilakukan untuk menyesuaikan antara seluruh perancangan sistem dengan *coding* yang telah dikerjakan.

4. Integritas dan pengujian sistem

Pada tahap ini, sistem telah dipastikan saling terintegrasi antar-unit atau fitur pada sistem. Selain itu, ditentukan kemampuan dan efektivitas dari sistem yang telah dibuat dengan dua metode, yaitu *white box* yang merupakan metode yang dilakukan dengan membaca alur logika yang berjalan pada sistem dan *black box* yang merupakan metode yang dilakukan pada fungsionalitas sistem tanpa pengetahuan *source code* sistem tersebut. Selain itu, terdapat pula pengujian validitas untuk hasil prediksi dari sistem tersebut, baik perbandingan antara hasil

prediksi dengan PD 10, maupun dengan metode *K-Fold Cross Validation*.

5. Operasi dan pemeliharaan

Pada tahap ini, ditentukan apakah terdapat kesalahan-kesalahan atau *bug* pada sistem yang telah dibuat di mana kesalahan-kesalahan tersebut muncul dikarenakan kondisi-kondisi tertentu seperti perubahan kebutuhan-kebutuhan dan performa sistem. Jika muncul kesalahan-kesalahan tersebut, maka sistem akan diubah pada tahap tertentu yang terus berkelanjutan hingga tahap ini sehingga kesalahan tersebut dapat teratasi.

2.7 Perumusan Premis

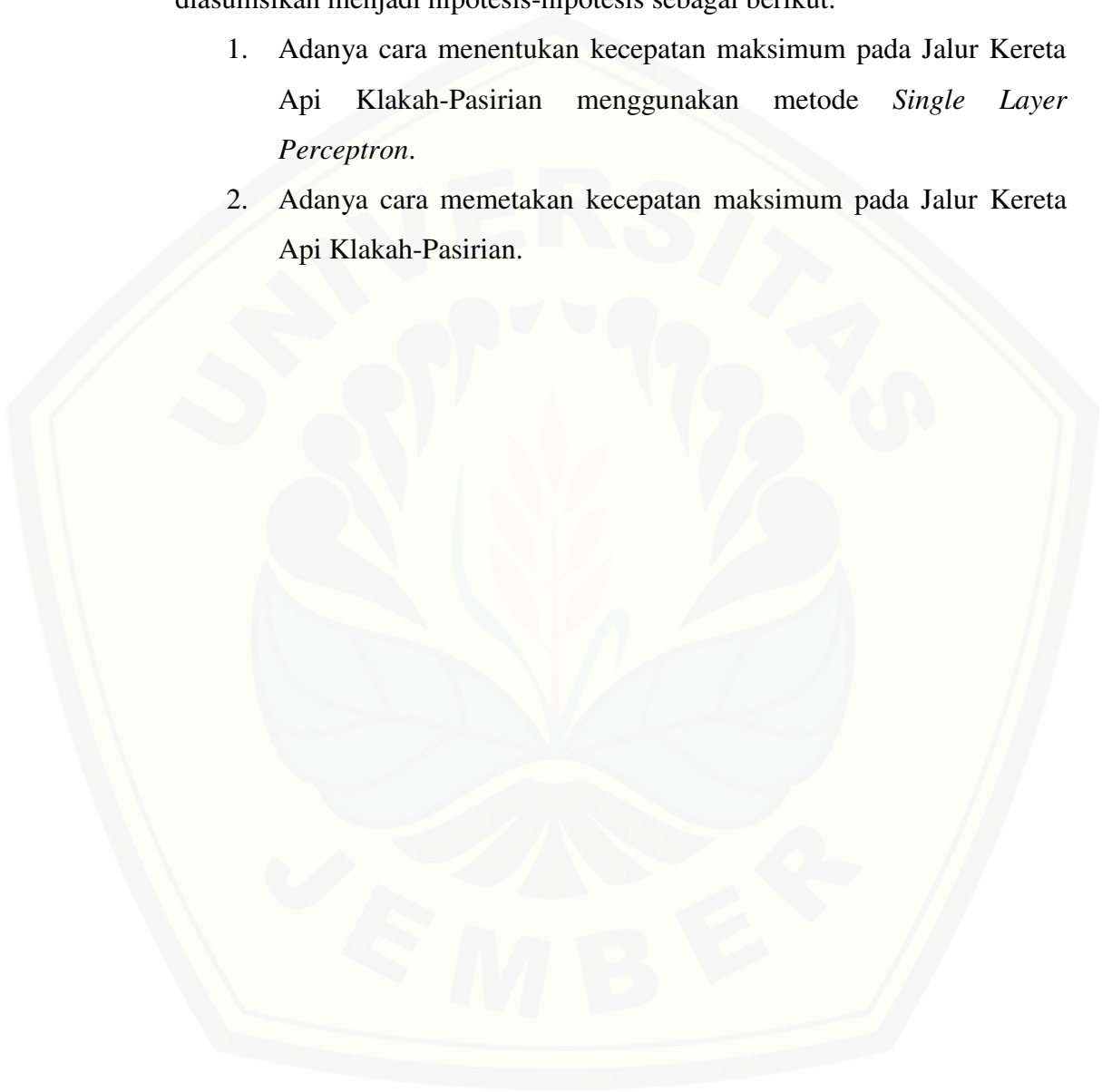
Melalui titik-titik pembahasan tersebut, maka dapat disusun beberapa premis sebagai berikut:

1. Algoritme klasifikasi merupakan algoritme yang digunakan untuk memprediksi kelas berdasarkan atribut-atribut yang ada.
2. Kecepatan maksimum merupakan kecepatan tertinggi pada suatu kereta api yang beroperasi pada suatu jalur kereta api tertentu yang memiliki berbagai kriteria seperti tipe rel, jenis bantalan, jenis penambat, tebal balas atas, lebar bahu balas, kelandaian, jari-jari minimal vertikal, jari-jari minimal horizontal tanpa peralihan, jari-jari minimal horizontal peralihan, dan lebar badan jalan.
3. Jalur Kereta Api Klakah-Pasirian merupakan jalur kereta api nonaktif yang menghubungkan antara Stasiun Klakah dengan Stasiun Pasirian dan memiliki petakan-petakan lintas tertentu.
4. Metode *Single Layer Perceptron* merupakan metode dari Jaringan Syaraf Tiruan yang hanya memiliki satu lapisan dengan bobot terhubung yang selanjutnya menerima masukan kemudian secara langsung akan mengolahnya menjadi keluaran.
5. Model *Waterfall* merupakan model pengembangan perangkat lunak yang berjalan bertahap secara sekuensial dan sistematis.

2.8 Perumusan Hipotesis

Dari premis-premis dan batasan masalah tersebut, maka dapat diasumsikan menjadi hipotesis-hipotesis sebagai berikut:

1. Adanya cara menentukan kecepatan maksimum pada Jalur Kereta Api Klakah-Pasirian menggunakan metode *Single Layer Perceptron*.
2. Adanya cara memetakan kecepatan maksimum pada Jalur Kereta Api Klakah-Pasirian.



BAB 3. METODE PENELITIAN

Bab ini merupakan penguraian teknik-teknik penelitian yang terdiri atas jenis penelitian, tempat dan waktu penelitian, rancangan penelitian, teknik perolehan data, analisis kebutuhan sistem, perancangan sistem, implementasi sistem, pengujian sistem, dan pemeliharaan sistem yang disesuaikan dengan model *Waterfall* yang telah dipaparkan pada tinjauan pustaka.

3.1 Jenis Penelitian

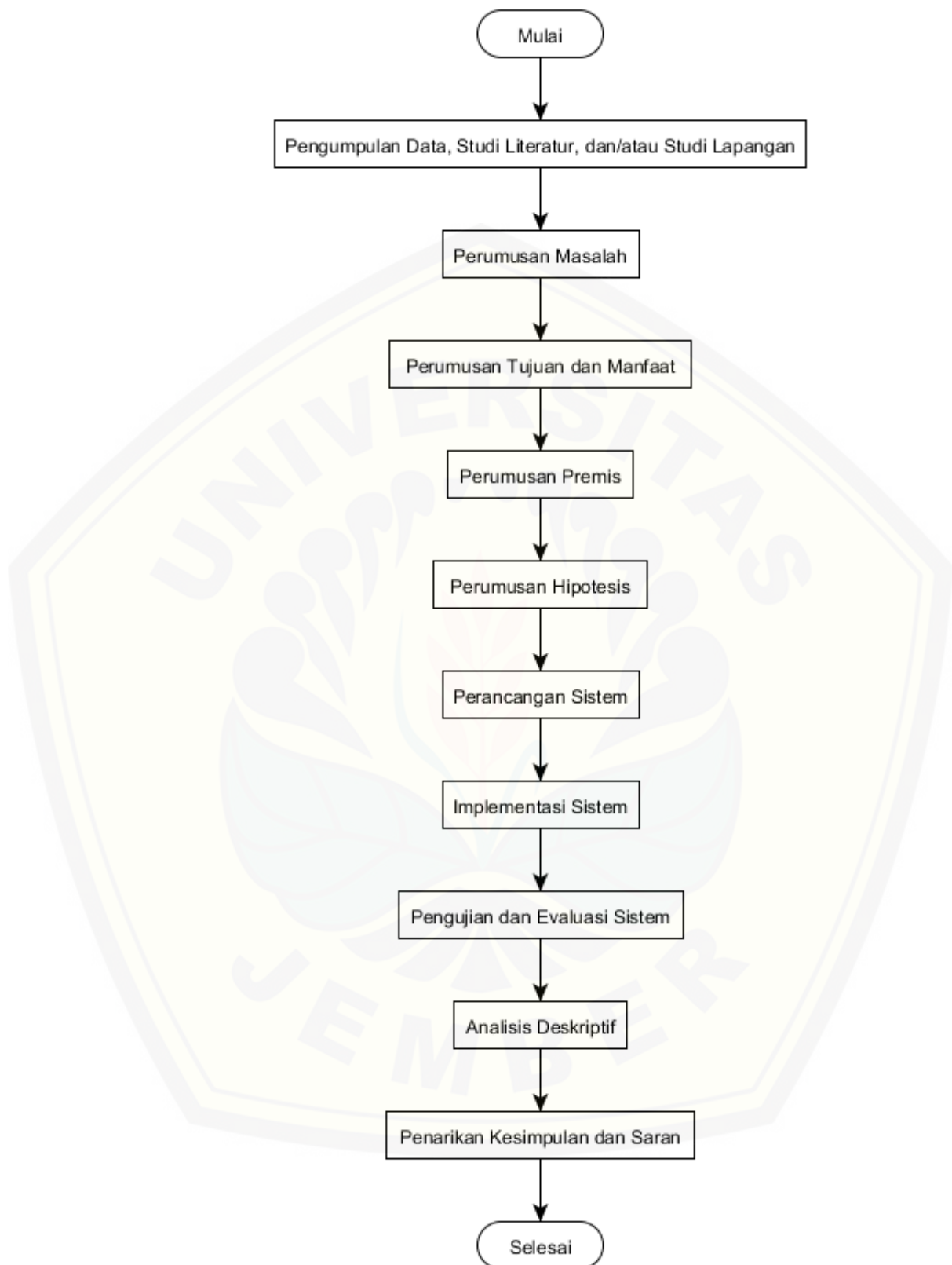
Jenis penelitian yang digunakan adalah penelitian kuantitatif berdasarkan jenis data, penelitian observasional berdasarkan ada/tidak adanya perlakuan, dan penelitian perpustakaan berdasarkan tempat penelitian.

3.2 Tempat dan Waktu Penelitian

Tahap awal penelitian dilakukan di PT. Kereta Api Indonesia Daerah Operasi IX Jember beralamatkan Jalan Dahlia 2 Kec. Patrang Kab. Jember, Resor Jalan Rel 9.2 beralamatkan Jalan K.H. Mas Mansyur Kota Probolinggo, dan Resor Jalan Rel 9.4 beralamatkan Jalan Stasiun Kec. Klakah Kab. Lumajang dengan waktu penelitian pada tanggal 4-18 November 2015. Tahap penelitian selanjutnya dilakukan di Badan Perencanaan dan Pembangunan Daerah Kabupaten Lumajang beralamatkan Jalan Alun-Alun Utara 7 Kabupaten Lumajang dengan waktu penelitian dilakukan pada tanggal 28 Desember 2015. Tahap penelitian selanjutnya dilakukan di Kampus Program Studi Sistem Informasi Universitas Jember beralamatkan Jalan Kalimantan 37 Kab. Jember dengan waktu penelitian dilakukan pada bulan Januari 2016 hingga Januari 2017.

3.3 Rancangan Penelitian

Gambar 3.1 merupakan alur rancangan dari penelitian ini. Awal proses pada alur ini adalah pengumpulan data, studi literatur, dan/atau studi lapangan. Selanjutnya, penelitian ini melakukan perumusan masalah yang dilanjutkan perumusan tujuan dan manfaat. Proses-proses tersebut merupakan tahapan analisis dan definisi persyaratan persyaratan pada model *waterfall*. Kemudian, penelitian ini melakukan perumusan premis yang didapat dari pustaka-pustaka yang diperlukan sehingga dapat dilakukan perumusan hipotesis. Setelah didapatkan hipotesis dari penelitian ini, proses dilanjutkan ke perancangan sistem yang merupakan tahapan perancangan sistem dan perangkat lunak pada model *waterfall*. Proses selanjutnya adalah implementasi sistem yang merupakan tahapan implementasi dan pengujian unit pada model *waterfall*. Kemudian, dilakukan proses pengujian dan evaluasi sistem yang merupakan tahapan integritas dan pengujian sistem yang dilanjutkan ke tahapan operasi dan pemeliharaan pada model *waterfall*. Setelah itu, dilakukan proses analisis deskriptif berupa hasil dan pembahasan sebagaimana yang dipaparkan pada bab 5 hingga rancangan penelitian ini berakhir pada proses penarikan kesimpulan dan saran sebagaimana yang dipaparkan pada bab 6. Gambar dari rancangan penelitian pada penelitian ini sebagai berikut:



Gambar 3.1 Rancangan Penelitian

3.4 Teknik Perolehan Data

Beberapa teknik perolehan data pada penelitian ini antara lain:

1. Teknik wawancara yang merupakan teknik di mana dilakukan tanya-jawab kepada para narasumber yang telah tercantum di subbab 3.2, yaitu tempat dan waktu penelitian.
2. Teknik observasi yang merupakan teknik di mana dilakukan peninjauan secara cermat untuk proses pengumpulan data dan studi literatur dari berbagai sumber seperti buku, jurnal dan *internet*.

3.5 Analisis Kebutuhan Sistem

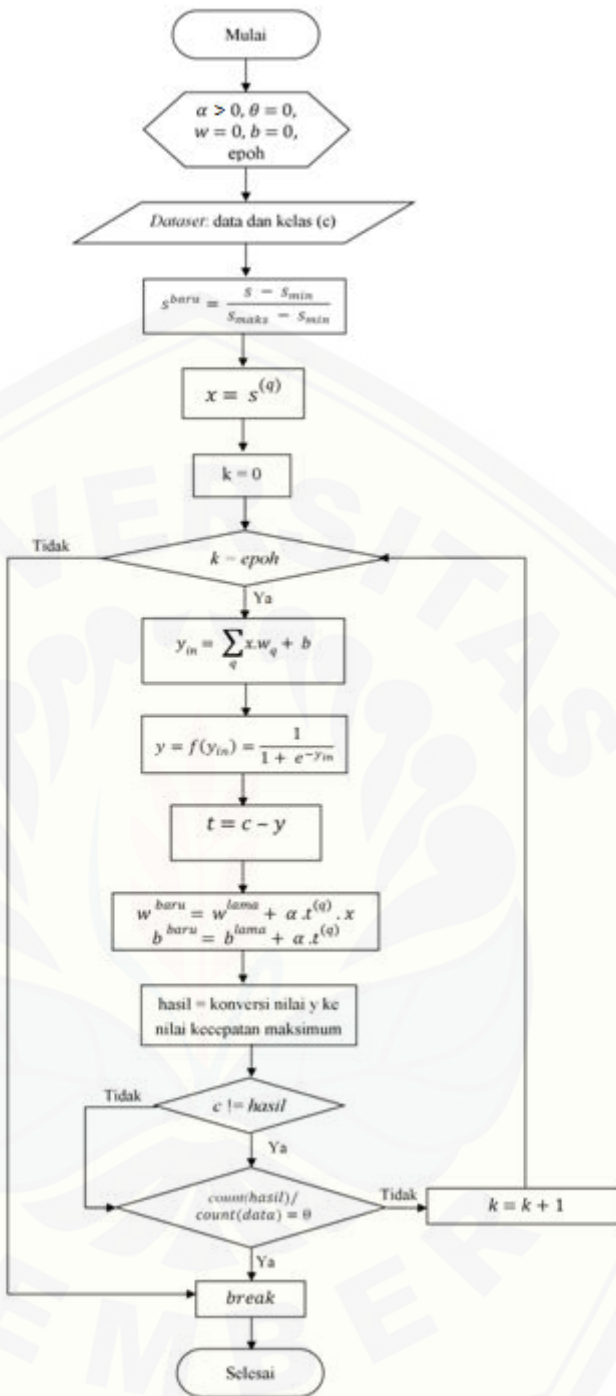
Analisis kebutuhan sistem merupakan tahapan awal dalam perancangan dan pengembangan sistem yang bertujuan untuk menentukan kebutuhan-kebutuhan pada sistem dari penelitian ini. Pada tahapan ini pula dilakukan teknik perolehan data sesuai dengan subbab 3.4 untuk mendapatkan gambaran tentang metode SLP yang akan digunakan pada sistem sebagai metode klasifikasi kecepatan maksimum kereta api dan data apa saja yang dibutuhkan untuk membangun sistem pada penelitian ini.

3.6.1 Metode SLP sebagai Metode Sistem Pengklasifikasi

Perancangan dan pembangunan sistem pengklasifikasi kecepatan maksimum menggunakan metode SLP ini didasarkan atas kelebihan-kelebihan metode ini seperti mampu mempelajari pola data sehingga dapat menciptakan pola sendiri yang menjadi rumus penghitungan untuk data yang baru. SLP sendiri merupakan salah satu arsitektur dari JST dan merupakan arsitektur JST yang paling sederhana. Selain itu, SLP memiliki satu lapisan bobot sehingga prosesnya lebih cepat daripada *Multi-Layer Perceptron* yang terdiri atas lebih dari satu lapisan bobot. Proses dari klasifikasi metode SLP untuk *dataset* dimulai pada proses inialisasi variabel seperti *learning rate* yang lebih dari 0, *threshold* dengan nilai 0, bobot-bobot dengan nilai 0, dan bias dengan nilai 0 serta epoch yang nilai tersebut ditentukan oleh pengguna atau aktor. Selanjutnya, nilai-nilai dari atribut-atribut kecepatan maksimum dilakukan normalisasi *min-max* akan dimasukkan sebagai nilai masukan.

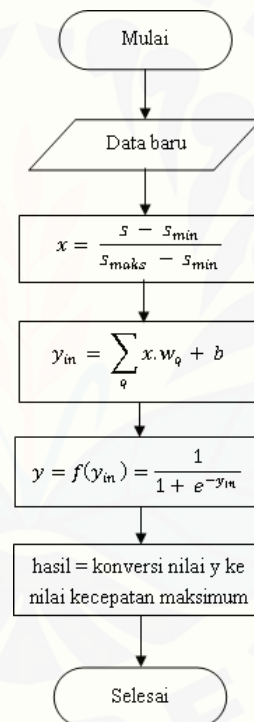
Proses selanjutnya adalah pengulangan proses *training-testing* dengan batasan nilai berupa epoch. Kemudian, proses selanjutnya adalah penghitungan nilai y_{in} sebagaimana pada persamaan (2) subbab 2.5, yaitu melalui hasil penjumlahan dari perkalian antara nilai data dengan bobot lalu ditambah dengan nilai bias sehingga didapat nilai keluaran. Nilai keluaran atau nilai y tersebut diolah menjadi nilai hasil melalui fungsi *unipolar sigmoidal* sebagaimana pada persamaan (3) subbab 2.5, lalu nilai y tersebut dikonversikan untuk didapat hasil nilai kecepatan maksimum. Konversi nilai hasil tersebut memiliki lima kondisi, yaitu nilai kecepatan maksimum adalah 120 km/jam jika nilai y lebih dari sama dengan 0,875, nilai kecepatan maksimum adalah 110 km/jam jika nilai y kurang dari 0,875 dan lebih dari sama dengan 0,625, nilai kecepatan maksimum adalah 100 km/jam jika nilai y kurang dari 0,625 dan lebih dari sama dengan 0,375, nilai kecepatan maksimum adalah 90 km/jam jika nilai y kurang dari 0,375 dan lebih dari sama dengan 0,125, serta nilai kecepatan maksimum adalah 80 km/jam jika nilai y kurang dari 0,125.

Proses selanjutnya dari klasifikasi SLP tersebut adalah perbandingan antara nilai kelas *dataset* dengan nilai hasil. Kemudian, dilakukan proses permisalan di mana jika nilai perbandingan antara jumlah data yang salah dengan nilai jumlah data sama dengan nilai *threshold*, maka proses pengulangan *training-testing* berhenti dan jika kondisi tersebut bernilai salah, maka proses pengulangan *training-testing* dilanjutkan pada epoch selanjutnya. Alur dari proses metode SLP pada *dataset* untuk mendapatkan nilai akurasi sebagai berikut:



Gambar 3.2 Alur Proses Klasifikasi Metode SLP pada Dataset

Alur proses metode SLP untuk diterapkan pada klasifikasi data baru pada penelitian ini dimulai pada masukan data berupa data baru yang merupakan data nilai kriteria suatu ruas lintas yang selanjutnya dilakukan normalisasi *min-max*. Proses selanjutnya adalah penghitungan nilai y_{in} sebagaimana pada persamaan (2) subbab 2.5. yang kemudian dilakukan penghitungan nilai y sebagaimana pada persamaan (3) subbab 2.5. Pada proses akhir dari alur ini adalah konversi nilai y untuk dijadikan nilai kecepatan maksimum sebagaimana pada proses klasifikasi SLP pada *dataset* untuk mencari nilai akurasi. Untuk alur klasifikasi metode SLP untuk data baru sebagai berikut:



Gambar 3.3 Alur Klasifikasi Metode SLP untuk Data Baru

3.6.2 Data yang Dibutuhkan Sistem

Data yang dibutuhkan untuk membangun sistem pada penelitian ini antara lain data peta sebagai data penunjang untuk klasifikasi dan laporan, data kriteria ruas lintas sebagai data baru untuk dilakukan klasifikasi supaya menghasilkan keluaran berupa nilai kecepatan maksimum, *dataset* kecepatan maksimum sebagai data utama dalam penghitungan metode SLP, data parameter SLP sebagai data masukan untuk penghitungan metode SLP dengan menggunakan *dataset*, dan data kepegawaian PT. Kereta Api Indonesia sebagai data untuk membedakan hak akses yang dapat menggunakan sistem pengklasifikasi ini.

3.6 Perancangan Sistem

Pada tahap ini, dilakukan perancangan dengan menggunakan UML yang mendukung konsep pemodelan pemrograman berbasis OOAD seperti yang akan diterapkan pada tahap implementasi sistem. Hasil perancangan sistem yang didapat antara lain *business process*, *use case diagram*, skenario, *sequence diagram*, *collaboration diagram*, *state diagram*, *class diagram*, dan *entity relationship diagram* atau ERD.

3.7 Implementasi Sistem

Pada tahap ini, dilakukan implementasi dari perancangan sistem sehingga menghasilkan kode-kode program untuk sistem ini. Sistem dibangun menggunakan bahasa PHP, *Hyper Text Markup Language* (HTML), JavaScript, dan *Structured Query Language* (SQL). Kerangka pemrograman atau *framework* sistem ini menggunakan CodeIgniter yang merupakan *framework* berkonsep OOP. Sistem ini menggunakan *local server* dengan menggunakan aplikasi XAMPP yang telah mendukung Apache sebagai *web server* untuk memfungsikan suatu web. Sistem pengelolaan basis data yang digunakan adalah PostgreSQL.

3.8 Pengujian Sistem

Pada tahap ini, dilakukan pengujian pada sistem secara keseluruhan. Metode yang digunakan untuk pengujian sistem adalah *black box* dan *white box*. *White box* merupakan metode untuk membaca alur logika yang berjalan pada sistem. Sedangkan *black box* merupakan metode yang dilakukan pada fungsionalitas sistem tanpa pengetahuan *source code* sistem tersebut. Selain itu, dilakukan pula pengujian validitas untuk membandingkan hasil prediksi dengan tabel kelas jalan rel sebagaimana yang tertera pada tabel 2.1, tabel jari-jari minimum lengkung horizontal sebagaimana yang tertera pada tabel 2.3, tabel jari-jari minimum lengkung vertikal sebagaimana yang tertera pada tabel 2.4, dan nomor 3 subbab 1.4. Selain itu, pengujian validitas juga menggunakan metode *K-Fold Cross Validation* untuk mendapatkan nilai akurasi pada setiap hasil pembagian *dataset* dengan K sebagai nilai pembagian *dataset* tersebut. Proses nilai akurasi dari metode ini sama dengan proses nilai akurasi pada seluruh data dari *dataset* secara keseluruhan, yaitu nilai perbandingan antara jumlah data yang salah dengan nilai jumlah data. Nilai K yang digunakan pada penelitian ini adalah 3.

3.9 Pemeliharaan Sistem

Pada tahap ini, dilakukan pencarian kesalahan-kesalahan atau *bug* pada sistem yang telah dibuat di mana kesalahan-kesalahan tersebut muncul dikarenakan kondisi-kondisi tertentu seperti perubahan kebutuhan-kebutuhan dan performa sistem. Jika muncul kesalahan-kesalahan tersebut, maka sistem akan diubah pada tahap tertentu yang terus berkelanjutan hingga tahap ini sehingga kesalahan tersebut dapat teratasi.

BAB 4. PERANCANGAN DAN PEMBANGUNAN SISTEM

Bab ini menguraikan tentang langkah-langkah dalam perancangan dan pembangunan Sistem Pengklasifikasi Kecepatan Maksimum Kereta Api (SIKEMAKA) yang terdiri atas gambaran umum, kebutuhan, perancangan, pembangunan, pengujian, dan pemeliharaan sistem.

4.1 Gambaran Umum Sistem

Gambaran umum dari SIKEMAKA merupakan gambaran secara keseluruhan dari SIKEMAKA sebagai sistem informasi. Gambaran umum ini dipaparkan pada bagian *statement of purpose* (SOP) dan hak akses pada sistem.

4.1.1. *Statement of Purpose* (SOP)

SIKEMAKA merupakan sistem yang dapat digunakan oleh pegawai-pegawai perkeretaapian berhak akses tertentu untuk memproses klasifikasi kecepatan maksimum kereta api pada suatu ruas lintas berdasarkan kriteria-kriteria seperti tipe rel, jenis bantalan, jenis penambat, tebal balas atas, lebar bahu balas, kelandaian, jari-jari minimal vertikal, jari-jari minimal horizontal tanpa peralihan, jari-jari minimal horizontal peralihan, lebar badan jalan, dan ruang bebas. Kriteria-kriteria tersebut digunakan sebagai atribut yang memiliki nilai kelas berdasarkan PD 10 sehingga terbentuk *dataset* yang selanjutnya diproses menggunakan metode SLP bersama dengan dua parameter metode ini, yaitu *learning rate* dan epoch untuk mendapatkan nilai akurasi, bobot, dan bias. Dari bobot dan bias tersebut, data baru dari setiap ruas lintas dapat dicari nilai kecepatan maksimumnya berdasarkan kriteria-kriteria kecepatan maksimum. Terdapat pula data peta lintas sebagai bahan untuk laporan kecepatan maksimum untuk suatu lintas atau jalur kereta api.

4.1.2. Hak akses pada Sistem

SIKEMAKA merupakan sistem yang memiliki batasan akses oleh pengguna-pengguna tertentu yang memiliki hak akses. Fitur yang digunakan untuk mengatur hak akses pada SIKEMAKA adalah fitur *login*, yaitu cara verifikasi yang dalam kasus ini dilakukan dengan mengisi NIPP dan kata sandi. Pengguna-pengguna dalam SIKEMAKA ini antara lain:

1. Kepala Sub Direktorat Jalan Rel dan Jembatan Pusat

Kepala Sub Direktorat Jalan Rel dan Jembatan Pusat memiliki akses untuk mengetahui tinjauan penghitungan metode SLP terakhir, pengelolaan *dataset*, dan pengelolaan klasifikasi *dataset*. Tinjauan penghitungan metode SLP terakhir merupakan fitur untuk menampilkan hasil penghitungan yang telah dilakukan terakhir sesuai dengan yang tersimpan pada basis data. Pengelolaan *dataset* merupakan fitur yang dapat meninjau, menambah, mengubah, menghapus, dan mencari data pada *dataset* kecepatan maksimum. Pengelolaan klasifikasi *dataset* merupakan fitur untuk menghasilkan bobot, bias, dan nilai akurasi dari penghitungan metode SLP menggunakan *dataset* kecepatan maksimum dan dua parameter yang digunakan, yaitu *learning rate* dan *epoch*.

2. Manajer Jalan dan Jembatan

Manajer Jalan dan Jembatan memiliki akses untuk melakukan penambahan data lintas pada tabel status perancangan data lintas yang diakses oleh Asisten Manajer Konstruksi Jalan Rel dan Jembatan. Selain penambahan data lintas, pengguna ini juga dapat melakukan penandaan bahwa data lintas tersebut perlu diubah oleh Asisten Manajer Program Jalan Rel dan Jembatan dan disetujui oleh Kepala Daerah Operasi (KADAOP) jika menurut pengguna terdapat kesalahan, namun untuk proses pengubahan peta dilakukan di sistem lain yang terintegrasi. Pengguna dapat pula melakukan validasi jika Asisten Manajer Konstruksi Jalan Rel dan Jembatan telah menyelesaikan klasifikasi per-lintas. Pengguna

dapat melihat hasil pengerjaan klasifikasi per-lintas dengan tampilan peta.

3. Asisten Manajer Konstruksi Jalan Rel dan Jembatan

Asisten Manajer Konstruksi Jalan Rel dan Jembatan memiliki akses untuk meninjau status perancangan, mengelola data lintas, dan mencetak laporan desain kecepatan maksimum. Tinjauan status perancangan merupakan fitur untuk mengetahui sejauh mana penyelesaian dalam pengerjaan klasifikasi kecepatan maksimum suatu lintas. Status tersebut antara lain mulai, proses perancangan, belum tervalidasi dan tervalidasi. Status mulai berarti data lintas tersebut belum memiliki nilai kriteria kecepatan maksimum. Status proses perancangan berarti masih dalam tahap pengerjaan. Status belum tervalidasi berarti selesai pengerjaan dan siap untuk di-validasi oleh Manajer Jalan dan Jembatan. Status tervalidasi berarti data lintas tersebut telah disetujui keabsahan data oleh Manajer Jalan dan Jembatan. Mengelola data lintas merupakan fitur untuk memasukkan nilai kriteria-kriteria kecepatan maksimum sehingga setiap ruas lintas dapat ditentukan nilai kecepatan maksimum yang dihitung oleh sistem. Sedangkan mencetak laporan desain kecepatan maksimum merupakan fitur untuk dapat melakukan cetak laporan melalui *printer* sesuai dengan pengaturan cetak yang telah tersedia.

4.2 Analisis Kebutuhan Perangkat Lunak

Analisis Kebutuhan Perangkat Lunak merupakan tahapan untuk menentukan kebutuhan apa saja yang digunakan dalam perancangan dan pembangunan SIKEMAKA. Tahapan ini meliputi dua macam kebutuhan, yaitu kebutuhan fungsional untuk menetapkan fitur-fitur yang penting dalam menjalankan SIKEMAKA dan kebutuhan non-fungsional sebagai tambahan untuk menunjang pembangunan SIKEMAKA.

4.2.1. Kebutuhan Fungsional

1. Sistem menggunakan fitur *login* sebagai menu verifikasi pada pengguna sesuai dengan hak akses yang dimiliki.

2. Sistem dapat mengelola (tambah, ubah, tampil, hapus) *dataset* kecepatan maksimum.
3. Sistem dapat melakukan penghitungan metode SLP sesuai menggunakan *dataset* kecepatan maksimum sebagai sumber data dan *learning rate* beserta epoch sebagai parameter SLP.
4. Sistem dapat menampilkan penghitungan akhir SLP.
5. Sistem dapat menampilkan tabel data lintas.
6. Sistem dapat menampilkan peta dalam sistem.
7. Sistem dapat menampilkan status perancangan pada data lintas.
8. Sistem dapat menambah nilai kriteria lintas untuk membantu proses masukan nilai kriteria ruas lintas.
9. Sistem dapat melakukan klasifikasi kecepatan maksimum suatu ruas lintas berdasarkan nilai kriteria ruas lintas beserta bobot dan bias yang telah ditetapkan pada penghitungan akhir SLP.
10. Sistem dapat menyimpan hasil klasifikasi kecepatan maksimum sebelum dilakukan validasi.
11. Sistem dapat melakukan validasi atas suatu lintas yang telah dianggap selesai dalam proses klasifikasi kecepatan maksimum ruas-ruas lintas pada lintas tersebut.
12. Sistem dapat mencetak laporan hasil klasifikasi kecepatan maksimum suatu lintas setelah dilakukan validasi sebagaimana yang tercantum pada nomor 11 subbab 4.2.1.

4.2.2. Kebutuhan Non-Fungsional

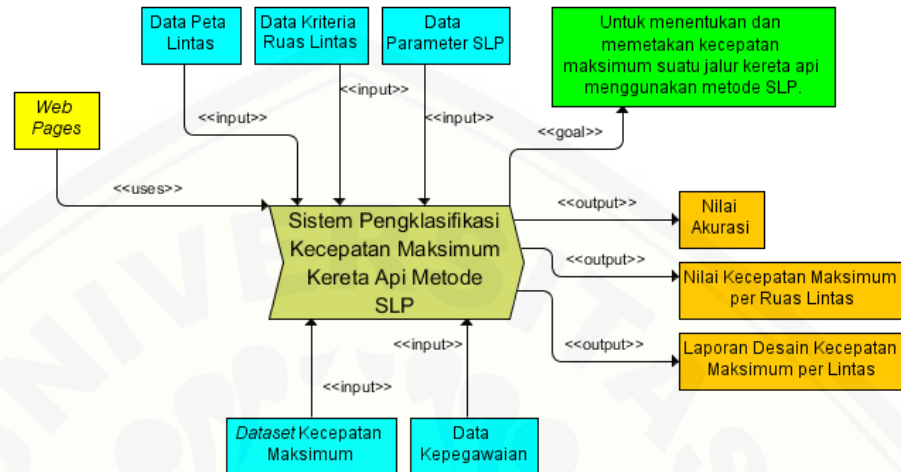
1. Sistem menggunakan Bootstrap untuk mengatur tampilan web.
2. Sistem membutuhkan koneksi *internet* untuk memperjelas posisi dan peta wilayah sekitar suatu lintas.
3. Sistem menggunakan Algoritme MD5 untuk keamanan *login*.

4.3 Perancangan Sistem

Rancangan sistem merupakan tahapan untuk pembuatan model sistem sebelum pembangunan sistem tersebut. Dokumen pada tahap ini antara lain *business process*, *use case diagram*, skenario, *activity diagram*, *sequence diagram*, *class diagram*, dan *entity relationship diagram*.

4.3.1. Business Process

Business process merupakan gambaran masukan, keluaran, penggunaan tampilan dasar, dan tujuan dari suatu sistem. Gambar untuk *business Process* pada SIKEMAKA sebagai berikut:



Gambar 4.1 *Business Process* SIKEMAKA

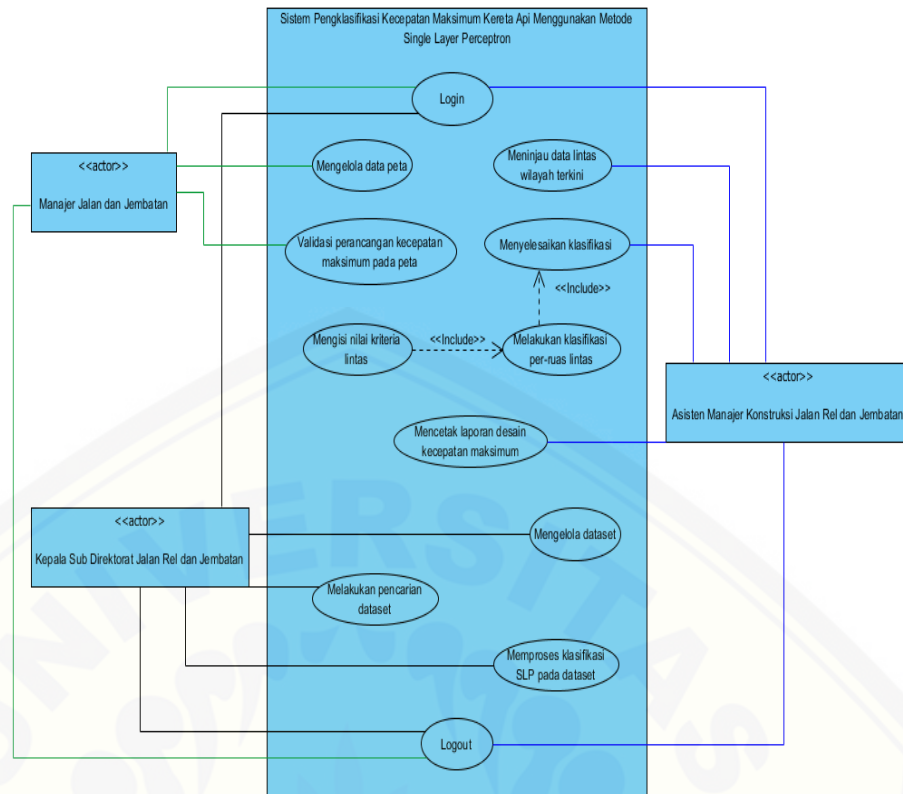
Berdasarkan gambar 4.1, sistem ini menggunakan tampilan dasar *web pages* atau dengan kata lain berbasis *web*. Sistem ini bertujuan untuk menentukan dan memetakan kecepatan maksimum suatu jalur kereta api menggunakan metode SLP. Sistem ini memiliki lima masukan, antara lain data peta lintas yang merupakan data peta yang disesuaikan dengan lintas atau jalur kereta api yang dimaksud, data kriteria ruas lintas yang merupakan data yang terdiri atas kriteria-kriteria suatu ruas lintas untuk penghitungan metode SLP sehingga menghasilkan nilai kecepatan maksimum, data parameter SLP yang merupakan data penghitungan metode SLP yang digunakan untuk klasifikasi kecepatan maksimum suatu ruas lintas, *dataset* kecepatan maksimum yang merupakan data untuk penghitungan metode SLP, dan data kepegawaian yang merupakan data untuk proses *login* sehingga setiap pengguna dapat masuk ke sistem sesuai dengan hak akses masing-masing. Sistem ini memiliki tiga keluaran, yaitu nilai akurasi yang didapat dari hasil penghitungan metode SLP

menggunakan parameter SLP dan *dataset* kecepatan maksimum, nilai kecepatan maksimum per-ruas lintas yang didapat dari hasil penghitungan klasifikasi metode SLP menggunakan data kriteria ruas lintas sebagai data untuk klasifikasi, dan laporan desain kecepatan maksimum per-lintas yang didapat dari data peta lintas.

4.3.2. Use Case Diagram

Use Case Diagram merupakan gambaran dari para pengguna yang memiliki hak akses atau aktor yang terhubung dengan fitur-fitur yang dapat diakses atau yang disebut pula dengan *use case*. Sistem dilambangkan dengan kotak biru yang didalamnya terdapat berbagai *use case* yang dilambangkan dengan elips biru. Sedangkan aktor dilambangkan dengan kotak biru bertanda “<<actor>>” dengan nama aktor di bawah tanda tersebut. Pada *use case diagram* SIKEMAKA, terdapat tiga aktor, yaitu Kepala Sub Direktorat Jalan Rel dan Jembatan, Manajer Jalan dan Jembatan, dan Asisten Manajer Konstruksi Jalan Rel dan Jembatan.

Kepala Sub Direktorat Jalan Rel dan Jembatan dapat mengelola *dataset*, melakukan pencarian data pada *dataset*, dan memproses klasifikasi menggunakan metode SLP pada *dataset*. Manajer Jalan dan Jembatan dapat mengelola data peta dan melakukan validasi perancangan kecepatan maksimum pada peta. Asisten Manajer Konstruksi Jalan Rel dan Jembatan dapat meninjau data lintas wilayah terkini, mengisi nilai kriteria lintas yang kemudian dilanjutkan dengan melakukan klasifikasi per-ruas lintas lalu diakhiri dengan menyelesaikan klasifikasi, dan mencetak laporan desain kecepatan maksimum. Gambar untuk *use case diagram* pada SIKEMAKA sebagai berikut:



Gambar 4.2 Use Case Diagram SIKEMAKA

Rincian dari penjelasan *use case diagram* SIKEMAKA secara yang terdiri atas nama *use case*, aktivitas, aktor, dan penjelasan tertuang pada tabel 4.1 sebagai berikut:

Tabel 4.1 Penjelasan Use Case Diagram

No.	Nama Use case	Aktivitas	Aktor	Penjelasan
1	<i>Login</i>	<i>Login</i> ke sistem	Semua	Merupakan <i>use case</i> yang menggambarkan proses <i>login</i> sistem
2		Data peta “tambah”		Merupakan <i>use case</i> yang menggambarkan penambahan data peta pada tabel status perancangan
3	Mengelola data peta	Data peta “ubah”	Manajer Jalan dan Jembatan	Merupakan <i>use case</i> yang menggambarkan proses peringatan data peta untuk diubah di sistem lain yang terintegrasi
4		Menampilkan data peta		Merupakan <i>use case</i> yang menggambarkan tampilan peta sesuai pilihan data peta

5	Validasi perancangan kec. maks. pada peta	Validasi perancangan kec. maks. pada peta		Merupakan <i>use case</i> yang menggambarkan validasi data peta setelah dilakukan penyelesaian klasifikasi
6	Mengelola <i>dataset</i>	<i>Dataset</i> “tampil”	Kepala Sub Direktorat Jalan Rel dan Jembatan Pusat	Merupakan <i>use case</i> yang menggambarkan tampilan tabel <i>dataset</i>
7		<i>Dataset</i> “tambah”		Merupakan <i>use case</i> yang menggambarkan proses penambahan data pada <i>dataset</i>
8		<i>Dataset</i> “ubah”		Merupakan <i>use case</i> yang menggambarkan proses perubahan data pada <i>dataset</i>
9		<i>Dataset</i> “hapus”		Merupakan <i>use case</i> yang menggambarkan proses penghapusan data pada tabel <i>dataset</i>
10	Melakukan pencarian <i>dataset</i>			Merupakan <i>use case</i> yang menggambarkan proses pencarian suatu kata kunci pada tabel <i>dataset</i>
11	Memproses Klasifikasi SLP pada <i>dataset</i>			Merupakan <i>use case</i> yang menggambarkan proses klasifikasi metode SLP menggunakan parameter-parameter SLP dan <i>dataset</i> sebagai data klasifikasi penentuan bobot dan bias
12	Meninjau data lintas wilayah terkini	Meninjau tabel status perancangan	Asisten Manajer Konstruksi Jalan Rel dan Jembatan	Merupakan <i>use case</i> yang menggambarkan tinjauan tabel status perancangan
13	Mengisi nilai kriteria lintas	Mengisi nilai kriteria lintas		Merupakan <i>use case</i> yang menggambarkan proses penentuan nilai kriteria suatu lintas
14	Melakukan klasifikasi per-ruas lintas	Melakukan klasifikasi per-ruas lintas		Merupakan <i>use case</i> yang menggambarkan proses klasifikasi berdasarkan data per-ruas lintas
15	Menyelesaikan klasifikasi	Menyelesaikan klasifikasi		Merupakan <i>use case</i> yang menggambarkan proses penyelesaian klasifikasi pada seluruh ruas lintas dalam suatu lintas
16	Mencetak laporan desain kec. Maks.	Mencetak laporan desain kec. Maks.		Merupakan <i>use case</i> yang menggambarkan proses pencetakan laporan desain kecepatan maksimum suatu lintas
17	<i>Logout</i>	<i>Logout</i> dari sistem	Semua	Merupakan <i>use case</i> yang menggambarkan proses <i>logout</i> sistem

4.3.3. Use Case Scenario

Use Case Scenario atau yang selanjutnya disebut skenario merupakan penjelasan alur fitur dari setiap *use case* pada *use case diagram* yang disertai dengan alur alternatif dari fitur tersebut. Skenario terdiri atas nama skenario, aktor, kondisi awal skenario, kondisi akhir skenario, skenario utama, dan skenario alternatif. Pembuatan skenario didasarkan atas *use case diagram* yang merupakan langkah sebelum tahapan ini. Dokumentasi skenario dari SIKEMAKA sebagai berikut:

1. Skenario dari *Use Case Login* Sistem

Skenario *login* sistem menjelaskan alur dari proses *login* aktor ke sistem. Skenario utama pada skenario ini dimulai saat aktor membuka halaman web SIKEMAKA, mengisi NIPP dan kata sandi lalu masuk ke sistem yang menampilkan halaman utama masing-masing aktor sesuai hak akses. Sedangkan skenario alternatif pada skenario ini antara lain aktor tidak mengisi NIPP atau kata sandi, ketidaksesuaian NIPP dengan kata sandi, dan tertolak masuk sistem karena aktor tidak memiliki hak akses sekalipun NIPP dengan kata sandi terdapat kesesuaian. Skenario *login* ke sistem dapat ditinjau pada tabel 4.2 untuk Manajer Jalan dan Jembatan, tabel 4.3 untuk Asisten Manajer Konstruksi Jalan Rel dan Jembatan, dan tabel 4.4 untuk Kepala Sub Direktorat Jalan Rel dan Jembatan Pusat.

Tabel 4.2 Skenario *Login* Manajer Jalan dan Jembatan

Nama	: <i>Login</i> Manajer Jalan dan Jembatan
Aktor	: Manajer Jalan dan Jembatan
Kondisi awal	: Membuka halaman <i>web</i> SIKEMAKA
Kondisi akhir	: Tampilan halaman utama Manajer Jalan dan Jembatan berupa menu tampil, tambah, ubah, dan validasi data peta
Skenario utama: <i>Login</i> Manajer Jalan dan Jembatan	
Aktor	Sistem
1. Membuka <i>web</i> SIKEMAKA.	
	2. Menampilkan halaman <i>login</i> .
3. Mengisi NIPP dan kata sandi.	
4. Meng-klik tombol “Masuk”.	
	5. Mengecek NIPP dan kata sandi.
	6. Tampilan halaman utama Manajer Jalan dan Jembatan berupa menu tampil, tambah, ubah, dan validasi data peta.
Skenario alternatif: <i>Login</i> Manajer Jalan dan Jembatan	
Tidak mengisi NIPP atau kata sandi.	
	5a. Menampilkan pesan “NIPP atau kata sandi belum terisi.”
Tidak sesuai NIPP dan kata sandi.	
	5b. Menampilkan pesan “Kombinasi NIPP dengan kata sandinya salah.”
	6b. Kembali ke halaman <i>login</i> .
Kombinasi NIPP dan kata sandi benar, tetapi tidak berhak akses.	
	5c. Mengalihkan ke halaman <i>web</i> PT. KAI

Tabel 4.3 Skenario *Login* Asisten Manajer Konstruksi Jalan Rel dan Jembatan

Nama	: <i>Login</i> Asisten Manajer Konstruksi Jalan Rel dan Jembatan	
Aktor	: Asisten Manajer Konstruksi Jalan Rel dan Jembatan	
Kondisi awal	: Membuka halaman <i>web</i> SIKEMAKA	
Kondisi akhir	: Tampilan halaman utama Asisten Manajer Konstruksi Jalan Rel dan Jembatan berupa data status perancangan lintas	
Skenario utama: <i>Login</i> Asisten Manajer Konstruksi Jalan Rel dan Jembatan		
	Aktor	Sistem
	1. Membuka <i>web</i> SIKEMAKA.	
		2. Menampilkan halaman <i>login</i> .
	3. Mengisi NIPP dan kata sandi.	
	4. Meng-klik tombol “Masuk”.	
		5. Mengecek NIPP dan kata sandi.
		6. Tampilan halaman utama Asisten Manajer Konstruksi Jalan Rel dan Jembatan berupa data status perancangan lintas
Skenario alternatif: <i>Login</i> Asisten Manajer Konstruksi Jalan Rel dan Jembatan		
Tidak mengisi NIPP atau kata sandi.		
		5a. Menampilkan pesan “NIPP atau kata sandi belum terisi.”
Tidak sesuai NIPP dan kata sandi.		
		5b. Menampilkan pesan “Kombinasi NIPP dengan kata sandinya salah.”
		6b. Kembali ke halaman <i>login</i> .
Kombinasi NIPP dan kata sandi benar, tetapi tidak berhak akses.		
		5c. Mengalihkan ke halaman <i>web</i> PT. KAI

Tabel 4.4 Skenario *Login* Kepala Sub Direktorat Jalan Rel dan Jembatan Pusat

Nama	: <i>Login</i> Kepala Sub Direktorat Jalan Rel dan Jembatan Pusat
Aktor	: Kepala Sub Direktorat Jalan Rel dan Jembatan Pusat
Kondisi awal	: Membuka halaman <i>web</i> SIKEMAKA
Kondisi akhir	: Tampilan halaman utama Kepala Sub Direktorat Jalan Rel dan Jembatan Pusat berupa data penghitungan terakhir metode SLP
Skenario utama: <i>Login</i> Kepala Sub Direktorat Jalan Rel dan Jembatan Pusat	
Aktor	Sistem
1. Membuka <i>web</i> SIKEMAKA.	
	2. Menampilkan halaman <i>login</i> .
3. Mengisi NIPP dan kata sandi.	
4. Meng-klik tombol “Masuk”.	
	5. Mengecek NIPP dan kata sandi.
	6. Tampilan halaman utama Kepala Sub Direktorat Jalan Rel dan Jembatan Pusat berupa data penghitungan terakhir metode SLP
Skenario alternatif: <i>Login</i> Kepala Sub Direktorat Jalan Rel dan Jembatan Pusat	
Tidak mengisi NIPP atau kata sandi.	
	5a. Menampilkan pesan “NIPP atau kata sandi belum terisi.”
Tidak sesuai NIPP dan kata sandi.	
	5b. Menampilkan pesan “Kombinasi NIPP dengan kata sandinya salah.”
	6b. Kembali ke halaman <i>login</i> .
Kombinasi NIPP dan kata sandi benar, tetapi tidak berhak akses.	
	5c. Mengalihkan ke halaman <i>web</i> PT. KAI

2. Skenario dari *Use Case* Mengelola Data Peta

Skenario mengelola data peta menjelaskan alur dari proses pengelolaan data peta yang terdiri atas menambah, mengubah, dan menampilkan data peta. Skenario utama pada skenario mengelola data peta “tambah” dimulai saat aktor telah mendapatkan tampilan tabel data peta pada halaman utama Manajer Jalan dan Jembatan. Selanjutnya, aktor meng-klik tombol “Tambah” sehingga sistem menampilkan pesan “Apakah data peta tersebut perlu dikelola di SIKEMAKA?”. Kemudian, aktor mengklik tombol “Ya” sehingga pada basis data terjadi perubahan status data peta tersebut menjadi berstatus tambah. Selanjutnya, menampilkan kembali halaman utama Manajer Jalan dan Jembatan. Sedangkan skenario alternatif pada skenario ini adalah jika aktor mengklik tombol “Tidak”, maka akan langsung menampilkan halaman utama Manajer Jalan dan Jembatan.

Skenario utama pada skenario mengelola data peta “ubah” dimulai saat aktor telah mendapatkan tampilan tabel data peta pada halaman utama Manajer Jalan dan Jembatan. Selanjutnya, aktor meng-klik tombol “Ubah” sehingga sistem menampilkan pesan “Apakah data peta tersebut perlu diubah? SIKEMAKA akan menyampaikan pesan ke e-Office untuk dilakukan peninjauan ulang pada data peta tersebut. Catatan: data peta tersebut tidak dapat digunakan di SIKEMAKA, kecuali telah disetujui oleh Kepala DAOP.”. Kemudian, aktor mengklik tombol “Ya” sehingga pada basis data terjadi perubahan status data peta tersebut menjadi berstatus ubah. Selanjutnya, menampilkan kembali halaman utama Manajer Jalan dan Jembatan. Sedangkan skenario alternatif pada skenario ini adalah jika aktor mengklik tombol “Tidak”, maka akan langsung menampilkan halaman utama Manajer Jalan dan Jembatan.

Skenario utama pada skenario mengelola data peta “tampil” dimulai saat aktor yang telah mendapatkan tampilan tabel data peta pada halaman utama Manajer Jalan dan Jembatan memilih lintas pada tabel data peta dengan cara meng-klik salah satu nama data peta pada kolom lintas wilayah sehingga sistem menampilkan tampilan peta dari data peta yang telah dipilih tersebut. Selanjutnya, aktor meng-klik tombol “Halaman Utama” sehingga sistem menampilkan kembali halaman utama Manajer Jalan dan Jembatan. Selanjutnya, menampilkan kembali halaman utama Manajer Jalan dan Jembatan. Selain itu, skenario ini tidak terdapat skenario alternatif.

Skenario mengelola data peta “tambah” dapat ditinjau pada tabel 4.5, tabel 4.6 untuk mengelola data peta “ubah”, dan tabel 4.7 untuk mengelola data peta “tampil”.

Tabel 4.5 Skenario Mengelola Data Peta “Tambah”

Nama	: Mengelola data peta “tambah”
Aktor	: Manajer Jalan dan Jembatan
Kondisi awal	: Tampilan tabel data peta pada halaman utama Manajer Jalan dan Jembatan
Kondisi akhir	: Data peta yang ditandai tambah telah muncul pada tabel status perancangan.
Skenario utama: Mengelola data peta “tambah”	
Aktor	Sistem
1. Meng-klik tombol “Tambah”.	
	2. Menampilkan peringatan “Apakah data peta tersebut perlu dikelola di SIKEMAKA?”
3. Meng-klik tombol “Ya”.	
	4. Data peta berstatus tambah di basis data.
	5. Menampilkan halaman utama Manajer Jalan dan Jembatan
Skenario alternatif: Mengelola data peta “tambah”	
Membatalkan proses melakukan penambahan data peta	
3a. Meng-klik tombol “Tidak”.	
	4a. Menampilkan halaman utama Manajer Jalan dan Jembatan

Tabel 4.6 Skenario Mengelola Data Peta “Ubah”

Nama	: Mengelola data peta “ubah”
Aktor	: Manajer Jalan dan Jembatan
Kondisi awal	: Menampilkan tabel data peta pada halaman utama Manajer Jalan dan Jembatan
Kondisi akhir	: Aktor telah mengubah data peta.
Skenario utama: Mengelola data peta “ubah”	
Aktor	Sistem
1. Meng-klik tombol “Ubah”.	
	2. Menampilkan peringatan “Apakah data peta tersebut perlu diubah? SIKEMAKA akan menyampaikan pesan ke e-Office untuk dilakukan peninjauan ulang pada data peta tersebut. Catatan: data peta tersebut tidak dapat digunakan di SIKEMAKA, kecuali telah disetujui oleh Kepala DAOP.”
3. Meng-klik tombol “Ya”.	
	4. Data peta berstatus ubah di basis data.
	5. Menampilkan halaman utama Manajer Jalan dan Jembatan
Skenario alternatif: Mengelola data peta “ubah”	
Membatalkan proses menandai data peta untuk diubah	
3a. Meng-klik tombol “Tidak”.	
	4a. Menampilkan halaman utama Manajer Jalan dan Jembatan

Tabel 4.7 Skenario Mengelola Data Peta “Tampil”

Nama	: Mengelola data peta “tampil”
Aktor	: Manajer Jalan dan Jembatan
Kondisi awal	: Menampilkan tabel data peta pada halaman utama Manajer Jalan dan Jembatan
Kondisi akhir	: Menampilkan halaman peta berdasarkan data peta yang dipilih.
Skenario utama: Mengelola data peta “tampil”	
Aktor	Sistem
1. Memilih lintas pada tabel data peta	
	2. Menampilkan halaman peta lintas yang telah dipilih
3. Meng-klik tombol “Halaman Utama”	
	4. Menampilkan halaman utama Manajer Jalan dan Jembatan
Skenario alternatif: Mengelola data peta “tampil”	
-	-

3. Skenario dari *Use Case* Melakukan Validasi Perancangan Kecepatan Maksimum pada Data Peta

Skenario melakukan validasi perancangan kecepatan maksimum pada peta menjelaskan alur dari proses validasi data peta yang telah dilakukan penyelesaian klasifikasi pada perancangan kecepatan maksimum oleh Asisten Manajer Konstruksi Jalan Rel dan Jembatan Pusat. Skenario utama pada skenario validasi perancangan kecepatan maksimum pada data peta dimulai saat aktor telah mendapatkan tampilan tabel data peta pada halaman utama Manajer Jalan dan Jembatan. Selanjutnya, aktor meng-klik tombol “Validasi” sehingga sistem menampilkan pesan “Apakah data peta tersebut divalidasi? Catatan: Data peta yang telah divalidasi tidak dapat diubah validasinya.” Kemudian, aktor mengklik tombol “Ya” sehingga pada basis data terjadi perubahan status data peta tersebut menjadi berstatus tervalidasi. Selanjutnya, menampilkan kembali halaman utama Manajer Jalan dan Jembatan. Sedangkan skenario alternatif pada skenario ini adalah jika aktor

mengklik tombol “Tidak”, maka akan langsung menampilkan halaman utama Manajer Jalan dan Jembatan. Skenario validasi perancangan kecepatan maksimum pada peta dapat ditinjau sebagai berikut:

Tabel 4.8 Skenario Melakukan Validasi Perancangan Kecepatan Maksimum pada Data Peta

Nama	: Melakukan validasi perancangan kecepatan maksimum pada data peta	
Aktor	: Manajer Jalan dan Jembatan	
Kondisi awal	: Menampilkan tabel data peta pada halaman utama Manajer Jalan dan Jembatan	
Kondisi akhir	: Data peta telah tervalidasi	
Skenario utama: Melakukan validasi perancangan kecepatan maksimum pada data peta		
	Aktor	Sistem
1.	Meng-klik tombol “Belum Tervalidasi”.	
		2. Menampilkan peringatan “Apakah data peta tersebut divalidasi? Catatan: Data peta yang telah divalidasi tidak dapat diubah validasinya.”
3.	Meng-klik tombol “Ya”.	
		4. Data peta berstatus tervalidasi di basis data.
		5. Menampilkan halaman utama Manajer Jalan dan Jembatan
Skenario alternatif: Melakukan validasi perancangan kecepatan maksimum pada data peta		
Membatalkan proses validasi data peta		
3a.	Meng-klik tombol “Tidak”.	
		4a. Menampilkan halaman utama Manajer Jalan dan Jembatan

4. Skenario dari *Use Case* Mengelola *Dataset*

Skenario mengelola *dataset* menjelaskan alur dari proses pengelolaan *dataset* yang terdiri atas menampilkan, menambah, mengubah, dan menghapus suatu data pada tabel *dataset*. Skenario utama pada skenario mengelola *dataset* “tampil” dimulai saat aktor telah mendapati tampilan halaman utama Kepala Sub Direktorat Jalan Rel dan Jembatan Pusat.

Selanjutnya, aktor meng-klik tombol “*Dataset*” sehingga sistem menampilkan halaman *dataset*. Untuk menampilkan semua data pada tabel *dataset* secara keseluruhan, aktor meng-klik tombol “Tampilkan Semua”.

Skenario utama pada skenario mengelola *dataset* “tambah” dimulai saat aktor telah mendapati tampilan halaman utama *dataset*. Selanjutnya, aktor mengklik tombol “Tambah *Dataset*” sehingga sistem menampilkan halaman tambah *dataset*. Aktor kemudian mengisi *form* tambah *dataset* yang dilanjutkan dengan meng-klik tombol “Simpan”. Sistem kemudian isian *form* tambah *dataset*, lalu menyimpan data tersebut ke basis data, dan kemudian menampilkan halaman *dataset*. Sedangkan skenario alternatif pada skenario mengelola *dataset* “tambah” antara lain jika aktor meng-klik tombol “Hapus Masukan”, maka isian *form* tambah *dataset* terhapus. Selain itu, jika aktor tidak mengisi *form* tambah *dataset*, maka sistem akan menampilkan pesan “Please fill out this field.” yang dilanjutkan dengan menampilkan kembali halaman tambah *dataset*.

Skenario utama pada skenario mengelola *dataset* “ubah” dimulai saat aktor telah mendapati tampilan halaman *dataset*. Selanjutnya, aktor mengklik tombol “ubah” pada suatu data sehingga sistem menampilkan halaman ubah *dataset*. Aktor kemudian mengisi *form* ubah *dataset* yang dilanjutkan dengan meng-klik tombol “Simpan”. Sistem kemudian isian *form* ubah *dataset*, lalu menyimpan perubahan data tersebut ke basis data, dan kemudian menampilkan halaman *dataset*. Sedangkan skenario alternatif pada skenario mengelola *dataset* “ubah” antara lain jika aktor meng-klik tombol “Hapus Masukan”, maka isian *form* ubah *dataset* terhapus. Selain itu, jika terdapat *form* ubah *dataset* yang kosong, maka sistem akan menampilkan

pesan “Please fill out this field.” yang dilanjutkan dengan menampilkan kembali halaman ubah *dataset*.

Skenario utama pada skenario mengelola *dataset* “hapus” dimulai saat aktor telah mendapati tampilan halaman *dataset*. Selanjutnya, aktor meng-klik tombol “Hapus” pada data tertentu sehingga sistem menampilkan peringatan “Apakah data *dataset* tersebut dihapus?”. Jika aktor meng-klik tombol “Ya”, maka data tersebut dihapuskan yang selanjutnya menampilkan kembali halaman *dataset*. Sedangkan skenario alternatif pada skenario mengelola *dataset* “hapus” antara lain jika aktor meng-klik tombol “Tidak”, maka sistem menampilkan kembali halaman *dataset*.

Skenario mengelola *dataset* “tampil” dapat ditinjau pada tabel 4.9, tabel 4.10 untuk mengelola *dataset* “tambah”, tabel 4.11 untuk mengelola *dataset* “ubah”, dan tabel 4.12 untuk mengelola *dataset* “hapus”.

Tabel 4.9 Skenario Mengelola *Dataset* “tampil”

Nama	: Mengelola <i>dataset</i> “tampil”	
Aktor	: Kepala Sub Direktorat Jalan Rel dan Jembatan Pusat	
Kondisi awal	: Menampilkan halaman utama Kepala Sub Direktorat Jalan Rel dan Jembatan Pusat	
Kondisi akhir	: Menampilkan tabel <i>dataset</i> pada halaman <i>dataset</i>	
Skenario utama: Mengelola <i>dataset</i> “tampil”		
	Aktor	Sistem
1.	Meng-klik tombol “ <i>Dataset</i> ”	
		2. Menampilkan halaman <i>dataset</i>
3.	Meng-klik tombol “Tampilkan Semua”	
		4. Menampilkan semua data pada tabel <i>dataset</i>
Skenario alternatif: Mengelola <i>dataset</i> “tampil”		
	-	-

Tabel 4.10 Skenario Mengelola *Dataset* “tambah”

Nama	: Mengelola <i>dataset</i> “tambah”
Aktor	: Kepala Sub Direktorat Jalan Rel dan Jembatan Pusat
Kondisi awal	: Menampilkan halaman <i>dataset</i>
Kondisi akhir	: <i>Dataset</i> telah ditambahkan
Skenario utama: Mengelola <i>dataset</i> “tambah”	
Aktor	Sistem
1. Meng-klik tombol “Tambah <i>Dataset</i> ”	.
	2. Menampilkan halaman tambah <i>dataset</i>
3. Mengisi <i>form</i> tambah <i>dataset</i>	
4. Meng-klik tombol “Simpan”	
	5. Mengecek isian <i>form</i> tambah <i>dataset</i>
	6. Menyimpan data <i>dataset</i> ke basis data
	7. Menampilkan halaman <i>dataset</i>
Skenario alternatif: Mengelola <i>dataset</i> “tambah”	
Menghapus seluruh isian pada <i>form</i> tambah <i>dataset</i>	
4a. Meng-klik tombol “Hapus Masukan”	
	5a. Menghapus isian <i>form</i> tambah <i>dataset</i>
Tidak mengisi <i>form</i> tambah <i>dataset</i>	
	6b. Menampilkan pesan “Please fill out this field.”
	7b. Kembali ke halaman tambah <i>dataset</i> .

Tabel 4.11 Skenario Mengelola *Dataset* “ubah”

Nama	: Mengelola <i>dataset</i> “ubah”
Aktor	: Kepala Sub Direktorat Jalan Rel dan Jembatan Pusat
Kondisi awal	: Menampilkan tabel <i>dataset</i> pada halaman <i>dataset</i>
Kondisi akhir	: Aktor telah mengubah <i>dataset</i>
Skenario utama: Mengelola <i>dataset</i> “ubah”	
Aktor	Sistem
1. Meng-klik tombol “Ubah”	
	2. Menampilkan halaman <i>form</i> ubah <i>dataset</i>
3. Mengubah isian <i>form</i> ubah <i>dataset</i>	
4. Meng-klik tombol “Ubah”	
	5. Mengecek isian <i>form</i> ubah <i>dataset</i>
	6. Menyimpan perubahan data <i>dataset</i> ke basis data
	7. Menampilkan halaman <i>dataset</i>
Skenario alternatif: Mengelola <i>dataset</i> “ubah”	
Mengembalikan nilai awal isian pada <i>form</i> ubah <i>dataset</i>	
4a. Meng-klik tombol “Hapus Masukan”	
	5a. Mengembalikan isian <i>form</i> ubah <i>dataset</i>
Terdapat bagian <i>form</i> ubah <i>dataset</i> yang kosong	
	6b. Menampilkan pesan “Please fill out this field.”
	8b. Kembali ke halaman ubah <i>dataset</i> .

Tabel 4.12 Skenario Mengelola *Dataset* “hapus”

Nama	: Mengelola data peta “hapus”
Aktor	: Kepala Sub Direktorat Jalan Rel dan Jembatan Pusat
Kondisi awal	: Menampilkan halaman <i>dataset</i>
Kondisi akhir	: Aktor telah menghapus data <i>dataset</i>
Skenario utama: Mengelola <i>dataset</i> “hapus”	
Aktor	Sistem
1. Meng-klik tombol “Hapus”.	
	2. Menampilkan peringatan “Apakah data <i>dataset</i> tersebut dihapus?”
3. Meng-klik tombol “Ya”.	
	4. Menghapus data <i>dataset</i> pada basis data.
	5. Menampilkan halaman <i>dataset</i> .
Skenario alternatif: Mengelola <i>dataset</i> “hapus”	
Membatalkan proses menghapus data pada <i>dataset</i>	
3a. Meng-klik tombol “Tidak”.	
	4a. Menampilkan halaman <i>dataset</i> .

5. Skenario dari *Use Case* Melakukan Pencarian *Dataset*

Skenario melakukan pencarian *dataset* menjelaskan alur dari proses. Skenario utama pada skenario ini dimulai saat aktor mendapati tampilan halaman *dataset*. Selanjutnya, aktor mengisi *form* pencarian yang dilanjutkan dengan meng-klik tombol “Cari”. Sistem kemudian mengecek isian *form* pencarian yang dilanjutkan dengan menampilkan hasil pencarian pada tabel *dataset*. Sedangkan skenario alternatif pada skenario ini antara lain aktor jika aktor tidak mengisi *form* pencarian tersebut, maka sistem menampilkan pesan “Data pencarian belum terisi.”. Skenario melakukan pencarian *dataset* dapat ditinjau sebagai berikut:

Tabel 4.13 Skenario Melakukan Pencarian *Dataset*

Nama	: Melakukan pencarian <i>dataset</i>
Aktor	: Kepala Sub Direktorat Jalan Rel dan Jembatan Pusat
Kondisi awal	: Menampilkan halaman <i>dataset</i>
Kondisi akhir	: Menampilkan hasil pencarian pada tabel <i>dataset</i>
Skenario utama: Melakukan pencarian <i>dataset</i>	
Aktor	Sistem
1. Mengisi <i>form</i> pencarian	
2. Meng-klik tombol “Cari”	
	3. Mengecek isian <i>form</i> pencarian
	4. Menampilkan hasil pencarian pada tabel <i>dataset</i>
Skenario alternatif: Melakukan pencarian <i>dataset</i>	
Tidak mengisi <i>form</i> pencarian	
	4a. Menampilkan pesan “Data pencarian belum terisi.”

6. Skenario dari *Use case* Memproses Klasifikasi SLP pada *Dataset*

Skenario memproses klasifikasi SLP pada *dataset* menjelaskan alur dari proses. Skenario utama pada skenario ini dimulai saat aktor mendapati tampilan halaman *perceptron*. Selanjutnya, aktor mengisi *form* nilai parameter SLP yang dilanjutkan dengan meng-klik tombol “Hitung”. Sistem kemudian memproses penghitungan klasifikasi *dataset* tersebut, lalu menyimpan hasil klasifikasi pada basis data, menetapkan bobot *perceptron* sesuai dengan penghitungan klasifikasi terakhir, dan diakhiri dengan menampilkan hasil klasifikasi *dataset* tersebut. Sedangkan skenario alternatif pada skenario ini antara lain aktor tidak mengisi *form* nilai parameter SLP sehingga sistem menampilkan pesan “Please fill out this field.” dan kembali ke halaman *perceptron* yang terdiri atas *form* parameter SLP dan *field* hasil klasifikasi. Selain itu, jika nilai akurasi yang didapat lebih kecil daripada klasifikasi

sebelumnya, maka sistem menampilkan pesan “Nilai akurasi yang didapat lebih kecil daripada sebelumnya. Silakan atur kembali parameter SLP anda.” dan kembali ke halaman *perceptron* yang terdiri atas *form* parameter SLP dan *field* hasil klasifikasi. Skenario memproses klasifikasi SLP pada *dataset* dapat ditinjau sebagai berikut:

Tabel 4.14 Skenario Memproses Klasifikasi SLP pada *dataset*

Nama	: Memproses klasifikasi SLP pada <i>dataset</i>
Aktor	: Kepala Sub Direktorat Jalan Rel dan Jembatan Pusat
Kondisi awal	: Menampilkan halaman <i>perceptron</i>
Kondisi akhir	: Menampilkan hasil klasifikasi <i>dataset</i>
Skenario utama: Memproses klasifikasi SLP pada <i>dataset</i>	
Aktor	Sistem
1. Mengisi <i>form</i> nilai parameter SLP.	
2. Meng-klik tombol “Hitung”	
	3. Memproses klasifikasi
	4. Menyimpan hasil klasifikasi pada basis data
	5. Menetapkan bobot <i>perceptron</i> sesuai dengan penghitungan klasifikasi terakhir
	6. Menampilkan hasil klasifikasi <i>dataset</i>
Skenario alternatif: Memproses klasifikasi SLP pada <i>dataset</i>	
Tidak mengisi <i>form</i> nilai parameter SLP.	
	4a. Menampilkan pesan “Please fill out this field.”
	5a. Kembali ke halaman <i>perceptron</i> yang terdiri atas <i>form</i> parameter SLP dan <i>field</i> hasil klasifikasi
Nilai akurasi lebih kecil daripada sebelumnya	
	4b. Menampilkan pesan “Nilai akurasi yang didapat lebih kecil daripada sebelumnya. Silakan atur kembali parameter SLP anda.”
	5b. Kembali ke halaman <i>perceptron</i> yang terdiri atas <i>form</i> parameter SLP dan <i>field</i> hasil klasifikasi

7. Skenario dari *Use Case* Meninjau Tabel Status Perancangan

Skenario meninjau tabel status perancangan menjelaskan alur dari proses. Skenario utama pada skenario ini dimulai saat aktor mendapati tampilan halaman utama Asisten Manajer Konstruksi Jalan Rel dan Jembatan. Jika aktor meng-klik tombol “Mulai”, maka sistem menampilkan halaman *form* nilai kriteria lintas yang baru. Jika aktor meng-klik tombol “Proses Perancangan”, maka sistem menampilkan halaman *form* nilai kriteria ruas lintas yang dalam proses pengerjaan. Jika aktor meng-klik tombol “Belum Tervalidasi”, maka sistem menampilkan halaman *form* nilai kriteria ruas lintas yang selesai pada pengerjaan. Skenario meninjau tabel status perancangan dapat ditinjau sebagai berikut:

Tabel 4.15 Skenario Meninjau Tabel Status Perancangan

Nama	: Meninjau tabel status perancangan
Aktor	: Asisten Manajer Konstruksi Jalan Rel dan Jembatan
Kondisi awal	: Tampilan halaman utama Asisten Manajer Konstruksi Jalan Rel dan Jembatan
Kondisi akhir	: Aktor meninjau halaman sesuai dengan jenis status perancangan yang dipilih
Skenario utama: Meninjau tabel status perancangan	
Aktor	Sistem
	1. Menampilkan halaman utama Asisten Manajer Konstruksi Jalan Rel dan Jembatan
2. Meng-klik tombol “Mulai”	
	3. Menampilkan halaman <i>form</i> nilai kriteria lintas yang baru
4. Meng-klik tombol “Proses Perancangan”	
	5. Menampilkan halaman <i>form</i> nilai kriteria ruas lintas yang dalam proses pengerjaan
6. Meng-klik tombol “Belum Tervalidasi”	
	7. Menampilkan halaman <i>form</i>

	nilai kriteria ruas lintas yang selesai pada pengerjaan
Skenario alternatif: Meninjau tabel status perancangan	
-	-

8. Skenario dari *Use Case* Mengisi Nilai Kriteria Lintas

Skenario mengisi nilai kriteria lintas menjelaskan alur dari proses. Skenario utama pada skenario ini dimulai saat aktor mendapati tampilan halaman *form* nilai kriteria lintas. Selanjutnya, aktor menentukan isian *form* nilai kriteria lintas yang dilanjutkan dengan meng-klik tombol “Lanjut” sehingga sistem menyimpan data nilai kriteria lintas pada basis data dan kemudian menampilkan halaman *form* nilai kriteria ruas lintas. Sedangkan skenario alternatif pada skenario ini antara lain aktor tidak mengisi *form* nilai kriteria ruas lintas yang dilanjutkan dengan meng-klik tombol “Lanjut” sehingga sistem menampilkan pesan “Please fill out this field.” yang kemudian sistem kembali ke halaman *form* nilai kriteria lintas. Skenario mengisi nilai kriteria lintas dapat ditinjau sebagai berikut:

Tabel 4.16 Skenario Mengisi Nilai Kriteria Lintas

Nama	: Mengisi nilai kriteria lintas	
Aktor	: Asisten Manajer Konstruksi Jalan Rel dan Jembatan	
Kondisi awal	: Menampilkan halaman <i>form</i> nilai kriteria lintas	
Kondisi akhir	: Menampilkan halaman rancang	
Skenario utama: Mengisi nilai kriteria lintas		
	Aktor	Sistem
1.	Menentukan isian <i>form</i> nilai kriteria lintas	
2.	Meng-klik tombol “Lanjut”	
		3. Menyimpan data nilai kriteria lintas pada basis data
		4. Menampilkan halaman <i>form</i> nilai kriteria ruas lintas
Skenario alternatif: Mengisi nilai kriteria lintas		
Terdapat kekosongan pada <i>form</i> nilai kriteria lintas.		
		3a. Menampilkan pesan “Please fill out this field.”

	4a. Kembali ke halaman <i>form</i> nilai kriteria lintas
--	--

9. Skenario dari *Use Case* Melakukan Klasifikasi per-Ruas Lintas

Skenario melakukan klasifikasi per-ruas lintas menjelaskan alur dari proses. Skenario utama pada skenario ini dimulai saat aktor menampilkan halaman rancang. Selanjutnya, aktor memilih ruas lintas, menentukan isian *form* nilai kriteria ruas lintas sesuai pilihan, dan meng-klik tombol “Simpan” sehingga data nilai kriteria lintas dan ruas lintas tersimpan pada basis data. Sedangkan skenario alternatif pada skenario ini antara lain aktor tidak mengisi sebagian *form* nilai kriteria ruas lintas, lalu meng-klik tombol “Simpan” sehingga sistem menampilkan pesan “Terdapat data yang belum terisi.”, lalu menampilkan halaman *form* nilai kriteria ruas lintas. Jika aktor meng-klik tombol “Reset”, maka sistem mengembalikan nilai awal sesuai dengan nilai kriteria. Skenario melakukan klasifikasi per-ruas lintas dapat ditinjau sebagai berikut:

Tabel 4.17 Skenario Melakukan Klasifikasi per-Ruas Lintas

Nama	: Melakukan klasifikasi per-ruas lintas	
Aktor	: Asisten Manajer Konstruksi Jalan Rel dan Jembatan	
Kondisi awal	: Menampilkan halaman rancang	
Kondisi akhir	: Menampilkan nilai kecepatan maksimum ruas lintas	
Skenario utama: Melakukan klasifikasi per-ruas lintas		
	Aktor	Sistem
	1. Memilih ruas lintas	
	2. Menentukan isian <i>form</i> nilai kriteria ruas lintas sesuai pilihan	
	3. Meng-klik tombol “Simpan”	
		4. Data nilai kriteria lintas dan ruas lintas tersimpan pada basis data.
		5. Menampilkan halaman <i>form</i> nilai kriteria ruas lintas
Skenario alternatif: Melakukan klasifikasi per-ruas lintas		
Terdapat kekosongan pada <i>form</i> nilai kriteria ruas lintas		

	4a. Menampilkan pesan “Terdapat data yang belum terisi.”
	5a. Menampilkan halaman <i>form</i> nilai kriteria ruas lintas
Nilai awal <i>form</i> kriteria ruas lintas dikembalikan	
3b. Meng-klik tombol “Reset”	
	4b. Mengembalikan nilai awal sesuai dengan nilai kriteria.

10. Skenario dari *Use Case* Menyelesaikan Klasifikasi

Skenario menyelesaikan klasifikasi menjelaskan alur dari proses. Skenario utama pada skenario ini dimulai saat aktor mendapati data nilai kriteria lintas dan ruas lintas telah tersimpan. Selanjutnya, aktor meng-klik tombol “Selesai” sehingga sistem menampilkan halaman utama Asisten Manajer Konstruksi Jalan Rel dan Jembatan. Skenario menyelesaikan klasifikasi dapat ditinjau sebagai berikut:

Tabel 4.18 Skenario Menyelesaikan Klasifikasi

Nama	: Menyelesaikan klasifikasi
Aktor	: Asisten Manajer Konstruksi Jalan Rel dan Jembatan
Kondisi awal	: Data nilai kriteria lintas dan ruas lintas telah tersimpan
Kondisi akhir	: Tombol “Belum Tervalidasi” muncul pada halaman utama Manajer Jalan dan Jembatan
Skenario utama: Menyelesaikan klasifikasi	
Aktor	Sistem
	1. Data nilai kriteria keseluruhan telah terisi
2. Meng-klik tombol “Selesai”	
	3. Menampilkan halaman utama Asisten Manajer Konstruksi Jalan Rel dan Jembatan
Skenario alternatif: Menyelesaikan klasifikasi	
-	-

11. Skenario dari *Use Case* Mencetak Laporan Desain Kecepatan Maksimum

Skenario mencetak laporan desain kecepatan maksimum menjelaskan alur dari proses. Skenario utama pada skenario ini dimulai saat aktor mendapati halaman utama Asisten Manajer Konstruksi Jalan Rel dan Jembatan. Selanjutnya, aktor mengklik tombol “Tervalidasi” sehingga sistem Menampilkan halaman laporan peta jalur. Skenario mencetak laporan desain kecepatan maksimum dapat ditinjau sebagai berikut:

Tabel 4.19 Skenario Mencetak Laporan Desain Kecepatan Maksimum

Nama	: Mencetak laporan desain kecepatan maksimum	
Aktor	: Asisten Manajer Konstruksi Jalan Rel dan Jembatan	
Kondisi awal	: Menampilkan halaman utama Asisten Manajer Konstruksi Jalan Rel dan Jembatan	
Kondisi akhir	: Aktor menyimpan/mencetak peta jalur	
Skenario utama: Mencetak laporan desain kecepatan maksimum		
	Aktor	Sistem
		1. Menampilkan tabel data lintas wilayah terkini
2. Meng-klik tombol “Tervalidasi”		
		3. Menampilkan halaman laporan peta jalur
Skenario alternatif: Mencetak laporan desain kecepatan maksimum		
	-	-

12. Skenario dari *Use Case Logout* Sistem

Skenario *logout* sistem menjelaskan alur dari proses. Skenario utama pada skenario ini dimulai saat seluruh aktor telah masuk ke SIKEMAKA. Selanjutnya, aktor mengklik tombol “Logout” sehingga sistem melakukan proses *logout*

dilanjutkan dengan menampilkan halaman *login*. Skenario *logout* dapat ditinjau sebagai berikut:

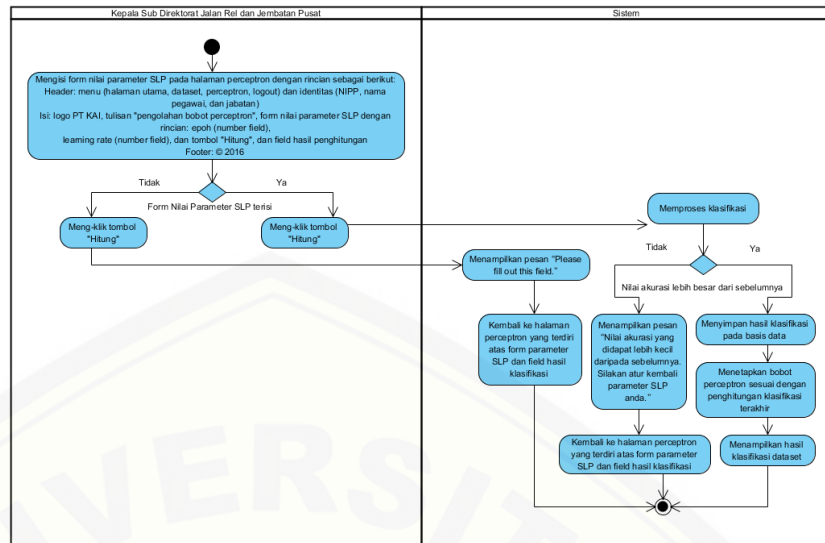
Tabel 4.20 Skenario *Logout*

Nama	: <i>Logout</i>	
Aktor	: Manajer Jalan dan Jembatan, Asisten Manajer Konstruksi Jalan Rel dan Jembatan, dan Kepala Sub Direktorat Jalan Rel dan Jembatan Pusat	
Kondisi awal	: Tampilan sistem terpilih sesuai dengan hak akses	
Kondisi akhir	: Menampilkan halaman <i>login</i>	
Skenario utama: <i>Logout</i>		
	Aktor	Sistem
1.	Meng-klik tombol " <i>Logout</i> "	
		2. Melakukan proses <i>logout</i>
		3. Menampilkan halaman <i>login</i>
Skenario alternatif: <i>Logout</i>		
	-	-

4.3.4. Activity Diagram

Activity Diagram merupakan gambaran aktivitas suatu sistem yang akan dibangun berupa alur yang berdasarkan skenario yang telah dibuat. Dokumentasi *activity diagram* dari SIKEMAKA dapat ditinjau pada Lampiran B.1. Berikut penjelasan seluruh fitur dan contoh *activity diagram* untuk fitur klasifikasi SLP pada *dataset* dan klasifikasi per ruas lintas.

1. Activity Diagram Memproses Klasifikasi Dataset

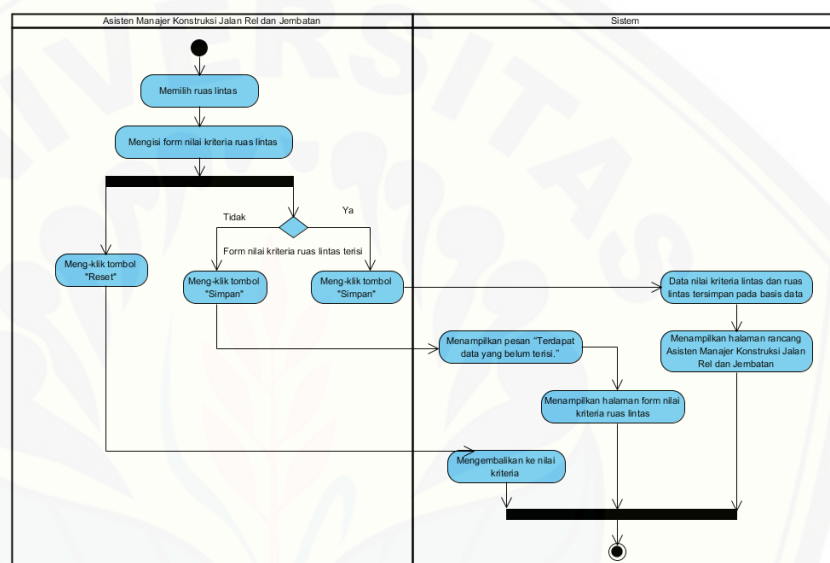


Gambar 4.3 Activity Diagram Memproses Klasifikasi Dataset

Gambar 4.3 merupakan *activity diagram* yang menggambarkan proses klasifikasi metode SLP pada *dataset* kecepatan maksimum. Proses diawali dengan aktor yang melakukan pengisian *form* nilai parameter untuk SLP yang terdiri atas nilai *learning rate* dan nilai epoch. Selanjutnya, aktor meng-klik tombol “Hitung” di mana jika *form* nilai parameter SLP terisi semua, maka sistem kemudian memproses klasifikasi hingga aktor mendapati hasil klasifikasi tersebut dan jika terdapat atau seluruh masukan dari *form* nilai parameter SLP yang tidak terisi, maka sistem akan menampilkan pesan “Please fill out this field.” dan kembali ke halaman *perceptron* yang terdiri atas *form* parameter SLP dan *field* hasil klasifikasi. *Field* hasil klasifikasi dapat terisi jika terdapat dua kondisi setelah sistem memproses klasifikasi, yaitu jika nilai akurasi yang didapat dari klasifikasi tersebut lebih kecil dari klasifikasi sebelumnya, maka sistem kemudian menampilkan pesan “Nilai akurasi yang didapat lebih kecil daripada sebelumnya. Silakan atur kembali parameter SLP anda.” yang dilanjutkan ke proses kembali ke halaman *perceptron* tersebut dan jika nilai akurasi

yang didapat dari klasifikasi tersebut lebih besar daripada klasifikasi sebelumnya, maka sistem akan menyimpan hasil klasifikasi tersebut ke basis data, lalu menetapkan bobot *perceptron*, yaitu nilai bobot dan nilai bias sesuai dengan proses klasifikasi terakhir tersebut, dan diakhiri dengan menampilkan hasil klasifikasi *dataset* tersebut pada *field* hasil klasifikasi halaman *perceptron*.

2. Activity Diagram Melakukan Klasifikasi per Ruas Lintas



Gambar 4.4 Activity Diagram Melakukan Klasifikasi per Ruas Lintas

Gambar 4.4 merupakan *activity diagram* yang menggambarkan proses klasifikasi metode SLP pada per-ruas lintas. Proses diawali dengan aktor yang melakukan pengisian *form* nilai kriteria ruas lintas. Selanjutnya, terdapat beberapa pilihan seperti meng-klik tombol “Reset” sehingga sistem mengembalikan ke nilai kriteria dan untuk *form* nilai kriteria ruas lintas yang terisi, jika meng-klik tombol “Simpan”, maka data nilai kriteria lintas dan ruas lintas tersimpan pada basis data kemudian menampilkan halaman rancang Asisten Manajer Konstruksi Jalan Rel dan Jembatan serta untuk *form* nilai

kriteria ruas lintas yang sebagian atau seluruhnya tidak terisi, jika meng-klik tombol “Simpan”, maka sistem menampilkan pesan “Terdapat data yang belum terisi.” Kemudian menampilkan halaman *form* nilai kriteria ruas lintas yang juga merupakan halaman rancang tersebut.

3. *Activity Diagram Login* Sistem untuk Asisten Manajer Konstruksi Jalan Rel dan Jembatan

Activity diagram ini merupakan gambaran alur proses *login* pada SIKEMAKA untuk pengguna Asisten Manajer Konstruksi Jalan Rel dan Jembatan. Gambar dari *activity diagram* ini dapat ditinjau pada Lampiran B.1.

4. *Activity Diagram Login* Sistem untuk Manajer Jalan dan Jembatan

Activity diagram ini merupakan gambaran alur proses *login* pada SIKEMAKA untuk pengguna Manajer Jalan dan Jembatan. Gambar dari *activity diagram* ini dapat ditinjau pada Lampiran B.1.

5. *Activity Diagram Login* Sistem untuk Kepala Sub Direktorat Jalan Rel dan Jembatan Pusat

Activity diagram ini merupakan gambaran alur proses *login* pada SIKEMAKA untuk pengguna Kepala Sub Direktorat Jalan Rel dan Jembatan Pusat. Gambar dari *activity diagram* ini dapat ditinjau pada Lampiran B.1.

6. *Activity Diagram* Menampilkan Tabel *Dataset*

Activity diagram ini merupakan gambaran alur untuk menampilkan tabel *dataset* baik secara terbagi maupun keseluruhan yang dilakukan oleh pengguna Kepala Sub

Direktorat Jalan Rel dan Jembatan Pusat. Gambar dari *activity diagram* ini dapat ditinjau pada Lampiran B.1.

7. *Activity Diagram Menambah Data pada Dataset*

Activity diagram ini merupakan gambaran alur untuk mengisi dan menyimpan data baru pada *dataset* yang dilakukan oleh pengguna Kepala Sub Direktorat Jalan Rel dan Jembatan Pusat. Gambar dari *activity diagram* ini dapat ditinjau pada Lampiran B.1.

8. *Activity Diagram Mengubah Data pada Dataset*

Activity diagram ini merupakan gambaran alur untuk mengisi dan mengubah data pada *dataset* yang ditentukan oleh pengguna Kepala Sub Direktorat Jalan Rel dan Jembatan Pusat. Gambar dari *activity diagram* ini dapat ditinjau pada Lampiran B.1.

9. *Activity Diagram Menghapus Data pada Dataset*

Activity diagram ini merupakan gambaran alur untuk menghapus data pada *dataset* yang ditentukan oleh pengguna Kepala Sub Direktorat Jalan Rel dan Jembatan Pusat. Gambar dari *activity diagram* ini dapat ditinjau pada Lampiran B.1.

10. *Activity Diagram Pencarian Data pada Dataset*

Activity diagram ini merupakan gambaran alur mencari data-data pada *dataset* berdasarkan kata kunci dan pilihan yang dilakukan oleh pengguna Kepala Sub Direktorat Jalan Rel dan Jembatan Pusat. Gambar dari *activity diagram* ini dapat ditinjau pada Lampiran B.1.

11. *Activity Diagram* Menambah Data Peta ke Tabel Status Perancangan

Activity diagram ini merupakan gambaran alur untuk menambah data peta yang dilakukan oleh pengguna Manajer Jalan dan Jembatan sehingga data peta tersebut muncul dengan status “Mulai” pada tabel status perancangan yang diakses oleh pengguna Asisten Manajer Konstruksi Jalan Rel dan Jembatan. Gambar dari *activity diagram* ini dapat ditinjau pada Lampiran B.1.

12. *Activity Diagram* Menandai Data Peta untuk Diubah di Sistem Lain

Activity diagram ini merupakan gambaran alur untuk menandai pada basis data yang dilakukan oleh pengguna Manajer Jalan dan Jembatan bahwa data peta tersebut perlu diubah pada sistem lain yang dapat mengubah data peta tersebut. Gambar dari *activity diagram* ini dapat ditinjau pada Lampiran B.1.

13. *Activity Diagram* Menampilkan Peta dari Data Peta yang Dipilih

Activity diagram ini merupakan gambaran alur untuk menampilkan peta dari data peta yang dipilih oleh pengguna Manajer Jalan dan Jembatan sehingga dapat dijadikan sebagai peninjauan untuk mengecek keabsahan peta dan proses pengerjaan klasifikasi kecepatan maksimum jalur kereta api pada data peta tersebut. Gambar dari *activity diagram* ini dapat ditinjau pada Lampiran B.1.

14. *Activity Diagram* Meninjau Tabel Status Perancangan

Activity diagram ini merupakan gambaran alur untuk meninjau tabel status perancangan yang dilakukan oleh pengguna Asisten Manajer Konstruksi Jalan Rel dan Jembatan.

Gambar dari *activity diagram* ini dapat ditinjau pada Lampiran B.1.

15. *Activity Diagram* Mengisi Nilai Kriteria Lintas

Activity diagram ini merupakan gambaran alur untuk mengisi dan menyimpan nilai kriteria suatu lintas yang dilakukan oleh pengguna Asisten Manajer Konstruksi Jalan Rel dan Jembatan. Gambar dari *activity diagram* ini dapat ditinjau pada Lampiran B.1.

16. *Activity Diagram* Menyelesaikan Klasifikasi

Activity diagram ini merupakan gambaran alur untuk memilih ruas lintas yang kemudian mengisi *form* nilai kriteria ruas lintas dan disimpan sehingga menghasilkan nilai klasifikasi untuk kecepatan maksimum dari ruas lintas tersebut hingga proses klasifikasi seluruh ruas lintas tersebut selesai yang dilakukan oleh pengguna Asisten Manajer Konstruksi Jalan Rel dan Jembatan. Gambar dari *activity diagram* ini dapat ditinjau pada Lampiran B.1.

17. *Activity Diagram* Melakukan Validasi Perancangan Kecepatan Maksimum pada Data Peta

Activity diagram ini merupakan gambaran alur untuk melakukan validasi perancangan kecepatan maksimum suatu data peta yang dilakukan oleh pengguna Manajer Jalan dan Jembatan. Gambar dari *activity diagram* ini dapat ditinjau pada Lampiran B.1.

18. *Activity Diagram* Mencetak Laporan Desain Kecepatan Maksimum

Activity diagram ini merupakan gambaran alur untuk mencetak laporan desain kecepatan maksimum suatu lintas atau jalur kereta api yang dilakukan oleh pengguna Asisten Manajer

Konstruksi Jalan Rel dan Jembatan. Gambar dari *activity diagram* ini dapat ditinjau pada Lampiran B.1.

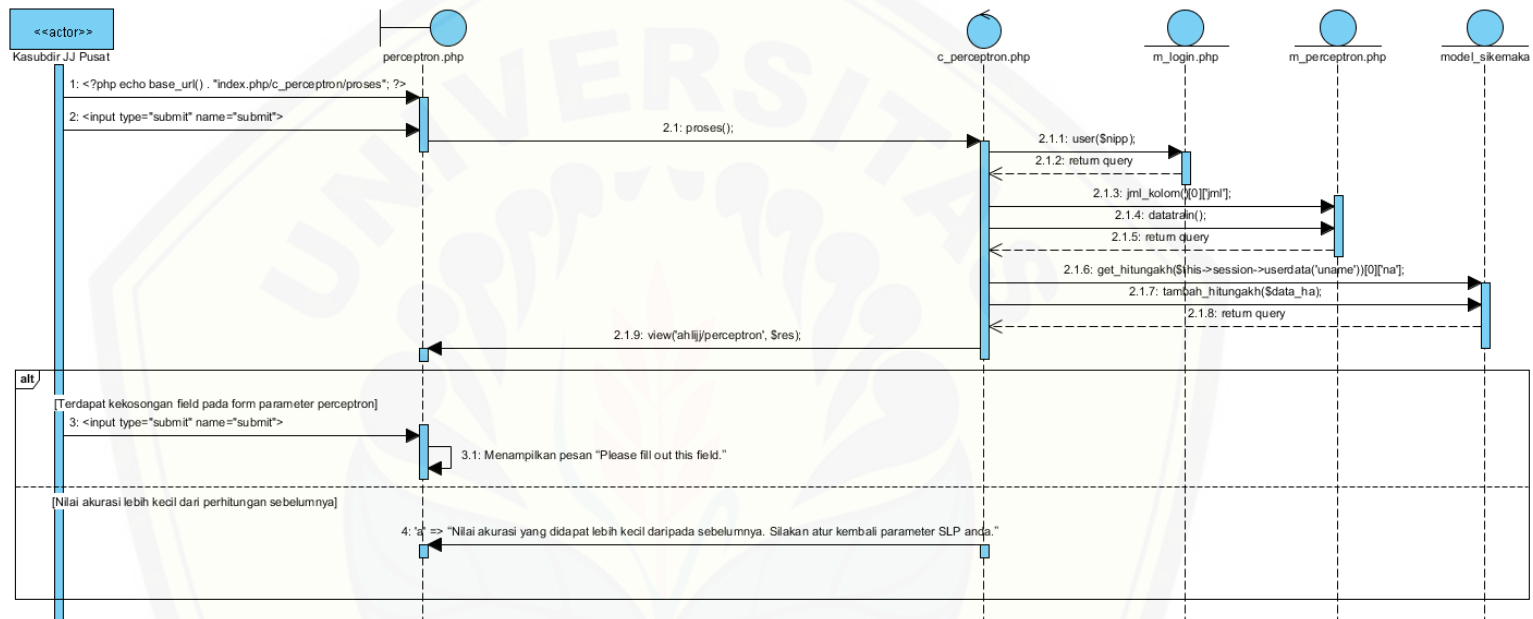
19. *Activity Diagram Logout Sistem*

Activity diagram ini merupakan gambaran alur proses *logout* dari SIKEMAKA yang dilakukan oleh seluruh pengguna SIKEMAKA. Gambar dari *activity diagram* ini dapat ditinjau pada Lampiran B.1.

4.3.5. *Sequence Diagram*

Sequence Diagram merupakan gambaran alur sistem secara terangkai di mana alur tersebut menggunakan bahasa pemrograman yang menyesuaikan dengan *activity diagram*. Alur tersebut menghubungkan antar-berkas di mana untuk berkas *controller* dan model masing-masing memiliki *class*. Dokumentasi *sequence diagram* dari SIKEMAKA dapat ditinjau pada Lampiran B.2. Berikut penjelasan seluruh fitur dan contoh *sequence diagram* untuk fitur klasifikasi SLP pada *dataset* dan klasifikasi per-ruas lintas.

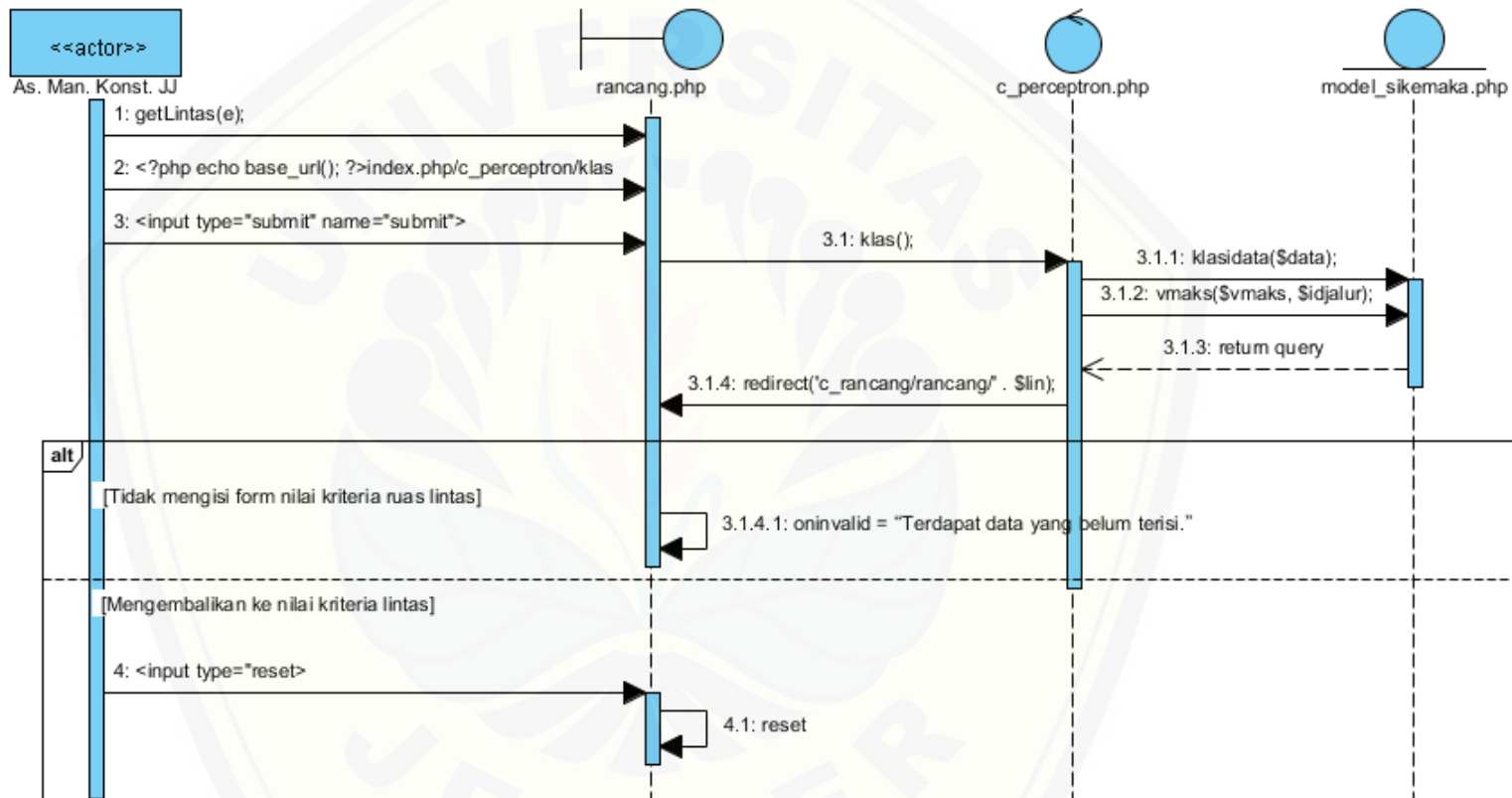
1. Sequence Diagram Memproses Klasifikasi Dataset



Gambar 4.5 Sequence Diagram Memproses Klasifikasi Dataset

Gambar 4.5 merupakan *sequence diagram* yang menggambarkan alur program dari proses klasifikasi *dataset*. Parameter SLP yang merupakan masukan dari aktor dikirim dari *view perceptron.php* ke *controller c_perceptron*. Selanjutnya, *c_perceptron* menerima hasil kueri berupa data pengguna atau pegawai dari *model m_login*, data *training* dari *model m_perceptron*, dan hasil penghitungan terakhir dari *model model_sikemaka*. Selain itu, pada *c_perceptron* yang merupakan *class* untuk pengaturan klasifikasi SLP mengirim hasil penghitungan klasifikasi sehingga tersimpan pada basis data melalui *model_sikemaka*. Hasil penghitungan terakhir yang terdapat pada *c_perceptron* dikirim ke *perceptron.php*. Selain itu, jika terdapat kekosongan *field* pada *form* parameter SLP atau *perceptron*, maka jika aktor melakukan klik *submit* yang pada *activity diagram* disebut dengan tombol “Hitung”, maka sistem menampilkan pesan “Please fill out this field.”. Jika pada penghitungan tersebut didapat nilai akurasi yang lebih kecil daripada penghitungan sebelumnya, maka sistem menampilkan pesan “Nilai akurasi yang didapat lebih kecil daripada sebelumnya. Silakan atur kembali parameter SLP anda.”

2. Sequence Diagram Melakukan Klasifikasi per Ruas Lintas



Gambar 4.6 Sequence Diagram Melakukan Klasifikasi per Ruas Lintas

Gambar 4.6 merupakan *sequence diagram* yang menggambarkan alur program dari proses klasifikasi per ruas lintas sehingga setiap ruas lintas memiliki nilai kecepatan maksimum kereta api. Aktor memilih ruas lintas yang ditentukan, lalu mengisi nilai kriteria ruas lintas dilanjutkan dengan meng-klik tombol “Simpan”, dan mengirim data tersebut dari *view rancang.php* ke *controller c_perceptron*. Pada *c_perceptron* data nilai kriteria tersebut diolah sehingga menghasilkan nilai kecepatan maksimum. Selanjutnya, data nilai kriteria tersebut beserta nilai kecepatan maksimum dikirim ke *model model_sikemaka* yang kemudian diolah untuk disimpan ke basis data. *Controller c_perceptron* mengatur untuk kembali ke halaman rancang atau *view rancang.php* yang menampilkan nilai kecepatan maksimum beserta nilai kriteria lintas dan ruas lintas yang telah diklasifikasikan. Selain itu, jika aktor tidak mengisi *form* nilai kriteria ruas lintas dan meng-klik tombol “Simpan”, maka sistem menampilkan pesan “Terdapat data yang belum terisi.”. Jika aktor ingin mengembalikan nilai awal nilai kriteria tersebut, maka aktor meng-klik tombol “Reset” yang kemudian sistem mengembalikan nilai-nilai pada *form* tersebut seperti semula.

3. *Sequence Diagram Login* Sistem untuk Asisten Manajer Konstruksi Jalan Rel dan Jembatan

Sequence diagram ini merupakan gambaran alur proses *login* pada SIKEMAKA untuk pengguna Asisten Manajer Konstruksi Jalan Rel dan Jembatan. Gambar dari *sequence diagram* ini dapat ditinjau pada Lampiran B.2.

4. *Sequence Diagram Login* Sistem untuk Manajer Jalan dan Jembatan

Sequence diagram ini merupakan gambaran alur proses *login* pada SIKEMAKA untuk pengguna Manajer Jalan dan Jembatan. Gambar dari *sequence diagram* ini dapat ditinjau pada Lampiran B.2.

5. *Sequence Diagram Login* Sistem untuk Kepala Sub Direktorat Jalan Rel dan Jembatan Pusat

Sequence diagram ini merupakan gambaran alur proses *login* pada SIKEMAKA untuk pengguna Kepala Sub Direktorat Jalan Rel dan Jembatan Pusat. Gambar dari *sequence diagram* ini dapat ditinjau pada Lampiran B.2.

6. *Sequence Diagram Menampilkan Tabel Dataset*

Sequence diagram ini merupakan gambaran alur untuk menampilkan tabel *dataset* baik secara terbagi maupun keseluruhan yang dilakukan oleh pengguna Kepala Sub Direktorat Jalan Rel dan Jembatan Pusat. Gambar dari *sequence diagram* ini dapat ditinjau pada Lampiran B.2.

7. *Sequence Diagram Menambah Data* pada *Dataset*

Sequence diagram ini merupakan gambaran alur untuk mengisi dan menyimpan data baru pada *dataset* yang dilakukan oleh pengguna Kepala Sub Direktorat Jalan Rel dan Jembatan Pusat. Gambar dari *sequence diagram* ini dapat ditinjau pada Lampiran B.2.

8. *Sequence Diagram* Mengubah Data pada *Dataset*

Sequence diagram ini merupakan gambaran alur untuk mengisi dan mengubah data pada *dataset* yang ditentukan oleh pengguna Kepala Sub Direktorat Jalan Rel dan Jembatan Pusat. Gambar dari *sequence diagram* ini dapat ditinjau pada Lampiran B.2.

9. *Sequence Diagram* Menghapus Data pada *Dataset*

Sequence diagram ini merupakan gambaran alur untuk menghapus data pada *dataset* yang ditentukan oleh pengguna Kepala Sub Direktorat Jalan Rel dan Jembatan Pusat. Gambar dari *sequence diagram* ini dapat ditinjau pada Lampiran B.2.

10. *Sequence Diagram* Pencarian Data pada *Dataset*

Sequence diagram ini merupakan gambaran alur mencari data-data pada *dataset* berdasarkan kata kunci dan pilihan yang dilakukan oleh pengguna Kepala Sub Direktorat Jalan Rel dan Jembatan Pusat. Gambar dari *sequence diagram* ini dapat ditinjau pada Lampiran B.2.

11. *Sequence Diagram* Menambah Data Peta ke Tabel Status Perancangan

Sequence diagram ini merupakan gambaran alur untuk menambah data peta yang dilakukan oleh pengguna Manajer Jalan dan Jembatan sehingga data peta tersebut muncul dengan status “Mulai” pada tabel status perancangan yang diakses oleh pengguna Asisten Manajer Konstruksi Jalan Rel dan Jembatan. Gambar dari *sequence diagram* ini dapat ditinjau pada Lampiran B.2.

12. *Sequence Diagram* Menandai Data Peta untuk Diubah di Sistem Lain

Sequence diagram ini merupakan gambaran alur untuk menandai pada basis data yang dilakukan oleh pengguna Manajer Jalan dan Jembatan bahwa data peta tersebut perlu diubah pada sistem lain yang dapat mengubah data peta tersebut. Gambar dari *sequence diagram* ini dapat ditinjau pada Lampiran B.2.

13. *Sequence Diagram* Menampilkan Peta dari Data Peta yang Dipilih

Sequence diagram ini merupakan gambaran alur untuk menampilkan peta dari data peta yang dipilih oleh pengguna Manajer Jalan dan Jembatan sehingga dapat dijadikan sebagai peninjauan untuk mengecek keabsahan peta dan proses pengerjaan klasifikasi kecepatan maksimum jalur kereta api pada data peta tersebut. Gambar dari *sequence diagram* ini dapat ditinjau pada Lampiran B.2.

14. *Sequence Diagram* Meninjau Tabel Status Perancangan

Sequence diagram ini merupakan gambaran alur untuk meninjau tabel status perancangan yang dilakukan oleh pengguna Asisten Manajer Konstruksi Jalan Rel dan Jembatan. Gambar dari *sequence diagram* ini dapat ditinjau pada Lampiran B.2.

15. *Sequence Diagram* Mengisi Nilai Kriteria Lintas

Sequence diagram ini merupakan gambaran alur untuk mengisi dan menyimpan nilai kriteria suatu lintas yang dilakukan oleh pengguna Asisten Manajer Konstruksi Jalan Rel dan Jembatan. Gambar dari *sequence diagram* ini dapat ditinjau pada Lampiran B.2.

16. *Sequence Diagram* Menyelesaikan Klasifikasi

Sequence diagram ini merupakan gambaran alur untuk memilih ruas lintas yang kemudian mengisi *form* nilai kriteria ruas lintas dan disimpan sehingga menghasilkan nilai klasifikasi untuk kecepatan maksimum dari ruas lintas tersebut hingga proses klasifikasi seluruh ruas lintas tersebut selesai yang dilakukan oleh pengguna Asisten Manajer Konstruksi Jalan Rel dan Jembatan. Gambar dari *sequence diagram* ini dapat ditinjau pada Lampiran B.2.

17. *Sequence Diagram* Melakukan Validasi Perancangan Kecepatan Maksimum pada Data Peta

Sequence diagram ini merupakan gambaran alur untuk melakukan validasi perancangan kecepatan maksimum suatu data peta yang dilakukan oleh pengguna Manajer Jalan dan Jembatan. Gambar dari *sequence diagram* ini dapat ditinjau pada Lampiran B.2.

18. *Sequence Diagram* Mencetak Laporan Desain Kecepatan Maksimum

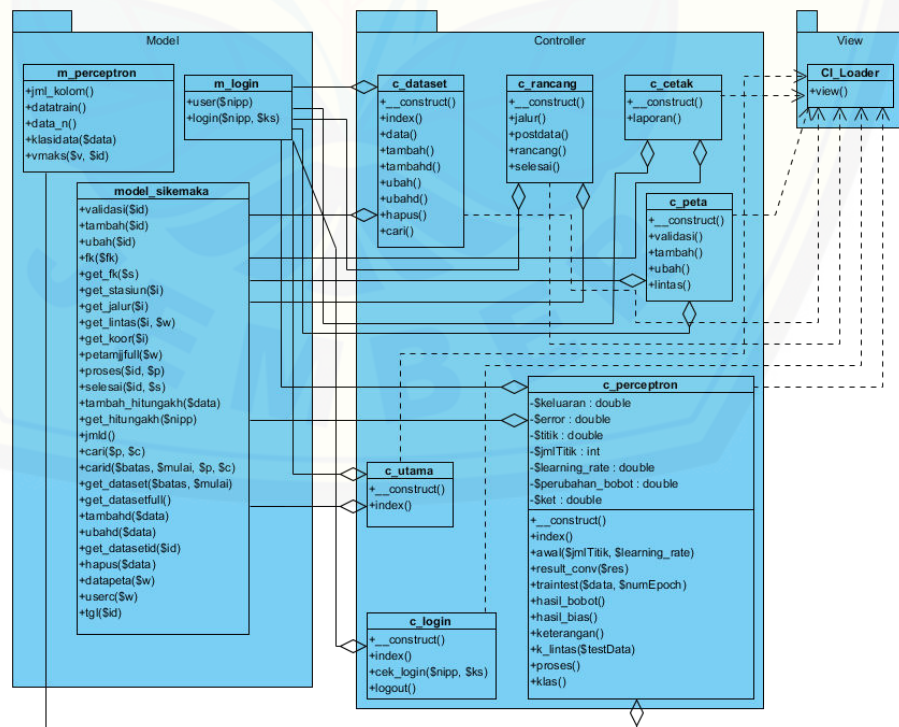
Sequence diagram ini merupakan gambaran alur untuk mencetak laporan desain kecepatan maksimum suatu lintas atau jalur kereta api yang dilakukan oleh pengguna Asisten Manajer Konstruksi Jalan Rel dan Jembatan. Gambar dari *sequence diagram* ini dapat ditinjau pada Lampiran B.2.

19. *Sequence Diagram* Logout Sistem

Sequence diagram ini merupakan gambaran alur proses *logout* dari SIKEMAKA yang dilakukan oleh seluruh pengguna SIKEMAKA. Gambar dari *sequence diagram* ini dapat ditinjau pada Lampiran B.2.

4.3.6. Class Diagram

Class Diagram merupakan gambaran dari berbagai *class* yang saling terhubung sesuai dengan model penerapan sistem atau *coding* yang dalam kasus ini menggunakan *Model-View-Controller* atau MVC berdasarkan *framework* CodeIgniter sehingga pada *class diagram* ini terdapat tiga *package*, yaitu *model*, *controller*, dan *view*. Suatu *class* terdiri atas berbagai atribut beserta tipe data dan *method* beserta parameter atau ada pula yang tanpa parameter. Setiap *class* pada *package* dari *model* terhubung dengan setiap *class* pada *package* dari *controller* menggunakan agregasi karena seluruh *class* pada *model* dapat diobjekkan ke *controller* di mana *model* memiliki fungsi mengelola transaksi *query* dari dan ke basis data dan *controller* memiliki fungsi mengelola alur-alur sistem sekaligus menjadi penghubung antara *view* dengan *model*. Selain itu, setiap *class* pada *package* dari *model* terhubung dengan setiap *class* pada *package* dari *controller* menggunakan dependensi karena seluruh *class* pada *controller* membutuhkan masukan atau *input* yang didapat dari *view*. Gambaran *class diagram* dari SIKEMAKA sebagai berikut:

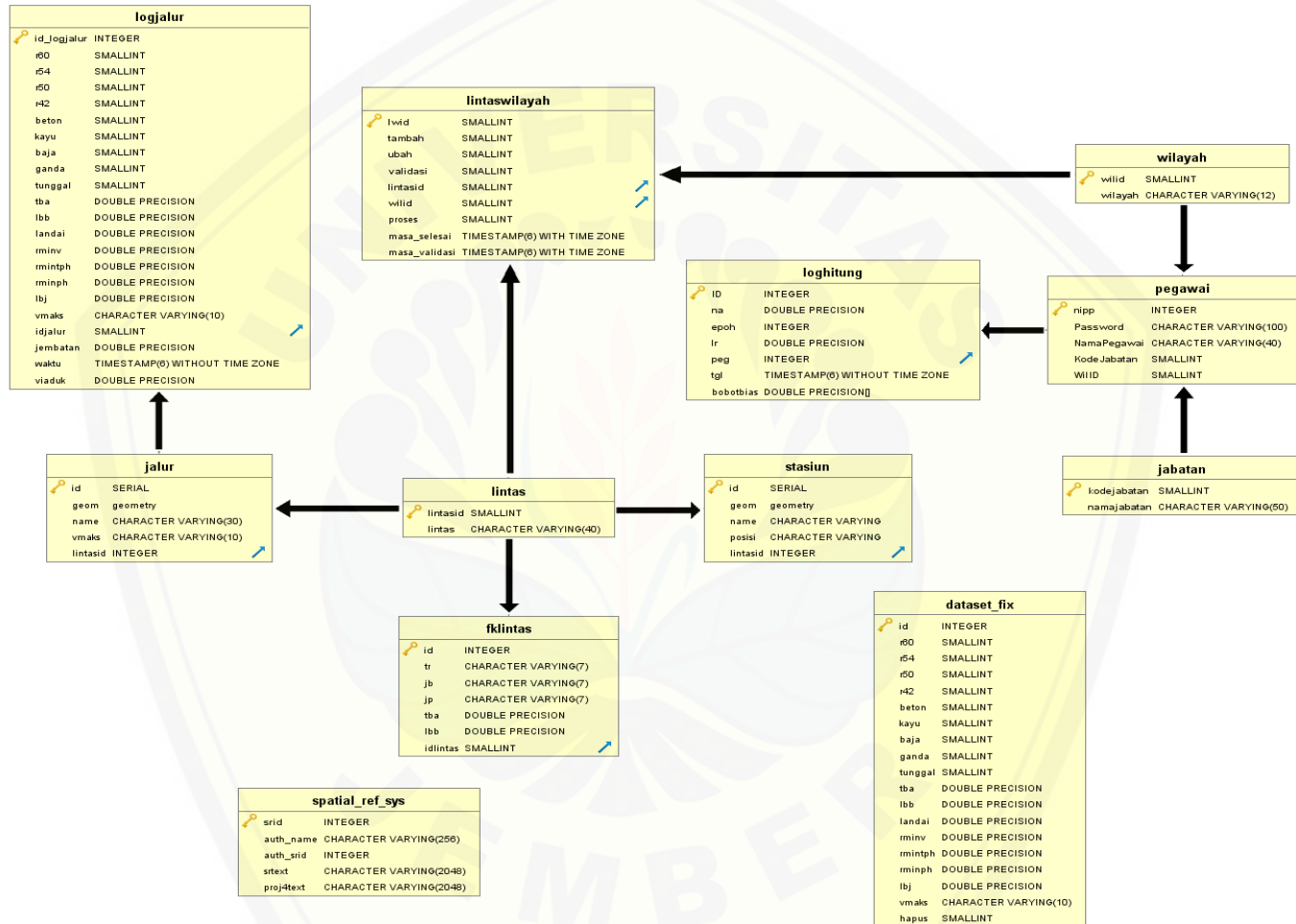


Gambar 4.7 Class Diagram SIKEMAKA

4.3.7. Entity Relationship Diagram (ERD)

Entity Relationship Diagram atau ERD merupakan gambaran dari entitas-entitas yang saling terhubung yang diterapkan pada basis data. Entitas sendiri merupakan tabel yang terdiri atas atribut-atribut yang masing-masing memiliki tipe data. Antar-entitas terhubung dengan relasi *one to many* yang berarti setiap anggota suatu entitas dapat terhubung dengan lebih dari satu anggota dari entitas lain yang terhubung. Gambaran ERD dari SIKEMAKA sebagai berikut:





Gambar 4.8 ERD SIKEMAKA

4.4 Penerapan Perancangan Sistem

Tahapan Penerapan Perancangan Sistem merupakan proses penerjemahan dari tahap perancangan sistem ke dalam kode-kode program. Sistem dibangun menggunakan bahasa PHP, *Hyper Text Markup Language* (HTML), JavaScript, dan *Structured Query Language* (SQL). Kerangka pemrograman atau *framework* sistem ini menggunakan CodeIgniter yang merupakan *framework* berkonsep OOP. Sistem ini menggunakan *local server* dengan menggunakan aplikasi XAMPP yang telah mendukung Apache sebagai *web server* untuk memfungsikan suatu web. Sistem pengelolaan basis data yang digunakan adalah PostgreSQL.

Penerapan Metode SLP dilakukan pada berkas atau *file controller* `c_perceptron.php` yang terdapat pada *folder controllers* sistem ini. Kode-kode program sistem untuk *folder controllers* dan *models* dapat ditinjau pada Lampiran C.

4.5 Pengujian Sistem

Tahapan Pengujian Sistem merupakan tahapan untuk pengujian pada sistem secara keseluruhan. Metode yang digunakan untuk pengujian sistem adalah *white box* dan *black box*. *White box* merupakan metode untuk membaca alur logika yang berjalan pada sistem. Sedangkan *black box* merupakan metode yang dilakukan pada fungsionalitas sistem tanpa pengetahuan *source code* sistem tersebut. Selain itu, terdapat pe

4.5.1 Pengujian White Box

Pengujian *White Box* merupakan pengujian sistem yang menggunakan metode untuk membaca alur logika yang berjalan pada sistem yang digunakan pada penelitian ini. Dalam pengujian ini, terdapat dokumen-dokumen yang digunakan, antara lain daftar *method*, diagram alir *method*, penghitungan *Cyclomatic Complexity* (CC), basis set, dan tes kasus. Dokumen-dokumen tersebut dapat ditinjau pada Lampiran D.1.

Khusus untuk contoh, berikut merupakan pengujian *white box* untuk alur klasifikasi *dataset* sebagai berikut:

1. Daftar Method

Method-method yang digunakan pada alur tersebut sebagai berikut:

<i>c_perceptron.php</i>	: <i>index()</i> , <i>proses()</i> , <i>awal(\$jmlTitik, \$learning_rate)</i> , <i>traintest(\$data, \$numEpoch)</i> , <i>hasil_bobot()</i> , <i>hasil_bias()</i> , <i>keterangan()</i> , dan <i>result_conv(\$res)</i>
<i>c_login.php</i>	: <i>index()</i>
<i>c_utama.php</i>	: <i>index()</i>
<i>m_login.php</i>	: <i>user(\$nipp)</i>
<i>model_sikemaka.php</i>	: <i>get_hitungakh(\$nipp)</i> dan <i>tambah_hitungakh(\$na, \$epoh, \$lr, \$nipp, \$larik)</i>
<i>m_perceptron.php</i>	: <i>jml_kolom()</i> dan <i>datatrain()</i>

```

23     public function index() {
24         $ambil_akun = $this->m_login->user($this->session->userdata('uname'));
25         $dataTr = $this->m_perceptron->datatrain();
26         $data = array(
27             'user' => $ambil_akun
28         );
29         $res = array(
30             'hasil' => 'Tentukan Epoch dan Learning Rate, lalu klik tombol Hitung.',
31             'epoh' =>
32                 $this->model_sikemaka->get_hitungakh($this->session->userdata('uname'))[0]
33                 ['epoh'],
34             'lr' =>
35                 $this->model_sikemaka->get_hitungakh($this->session->userdata('uname'))[0]
36                 ['lr'],
37             'datatr' => $dataTr
38         );
39         $stat = $this->session->userdata('lvl');
40         if ($stat == 3) {
41             $this->load->view('ahlijj/header', $data);
42             $this->load->view('ahlijj/perceptron', $res);
43             $this->load->view('Footer');
44         } else {
45             redirect('c_login', 'refresh');
46         }
47     }

```

Gambar 4.9 Program Code *index()* – *c_perceptron.php*

```

10     public function index() {
11         $session = $this->session->userdata('isLogin');
12         if ($session == FALSE) {
13             $this->load->view('login');
14         } else {
15             redirect('c_utama');
16         }
17     }

```

Gambar 4.10 Program Code *index()* – *c_login.php*

```

11     public function index() {
12         $ambil_akun = $this->m_login->user($this->session->userdata('uname'));
13         $ambil_data =
14         $this->model_sikemaka->get_hitungakh($this->session->userdata('uname'));
15         $ambil_data2 = $this->model_sikemaka->jmlid();
16         $ambil_data3 =
17         $this->model_sikemaka->datapeta($this->session->userdata('wil'))->result();
18         $data = array(
19             'user' => $ambil_akun
20         );
21         $data2 = array(
22             'ta' => $ambil_data,
23             'd' => $ambil_data2
24         );
25         $data3['dp'] = $ambil_data3;
26         $data4['dp'] =
27         $this->model_sikemaka->petamjffull($this->session->userdata('wil'));
28         $stat = $this->session->userdata('lvl');
29         if ($stat == 1) {
30             $this->load->view('manjj/header', $data);
31             $this->load->view('manjj/utama', $data4);
32             $this->load->view('footer');
33         } else if ($stat == 2) {
34             $this->load->view('pegawai/header', $data);
35             $this->load->view('pegawai/utama', $data3);
36             $this->load->view('footer');
37         } else if ($stat == 3) {
38             $this->load->view('ahlijj/header', $data);
39             $this->load->view('ahlijj/utama', $data2);
40             $this->load->view('footer');
41         } else if ($stat > 3) {
42             redirect('http://kereta-api.co.id');
43         } else {
44             redirect('c_login', 'refresh');
45         }
46     }
47 }

```

Gambar 4.11 Program Code index() – c_utama.php

```

243     public function proses() {
244         $snipp = $this->session->userdata('uname');
245         $ambil_akun = $this->m_login->user($snipp);
246         $data = array(
247             'user' => $ambil_akun
248         );
249
250         $jml_data = $this->m_perceptron->jml_kolom()[0]['jml'];
251         $dataTr = $this->m_perceptron->dataTrain();
252         $learning_rate = $this->input->post('lr'); // learning rate
253         $max_epoh = $this->input->post('e'); // maksimal iterasi
254         $jmlTitik = array($jml_data, 1);
255
256         $this->awal($jmlTitik, $learning_rate);
257         $hasil = $this->traintest($dataTr, $max_epoh); // melakukan proses
258         // klasifikasi dengan data uji
259
260         $na_db =
261         $this->model_sikemaka->get_hitungakh($this->session->userdata('uname'))[0]['na
262         '];
263         if ($this->ket[1] < $na_db) {
264             $na_kurang = "Nilai akurasi yang didapat lebih kecil daripada
265             sebelumnya. Silakan atur kembali parameter SLP anda.";
266         } else {
267             $na_kurang = "";
268         }
269
270         $this->model_sikemaka->tambah_hitungakh(round($this->ket[1], 2), $max_epoh,
271         $learning_rate, $snipp, json_encode($this->titik[1][0]));
272
273         $res = array(
274             'epoh' => $max_epoh,
275             'lr' => $learning_rate,
276             'hasil' => '',
277             'jmlData' => count($dataTr),
278             'titik' => ($jmlTitik[0] - 1),
279             'keluaran' => $jmlTitik[1],
280             'bobot' => $this->hasil_bobot(),
281             'bias' => $this->hasil_bias(),
282             'hasilp' => $hasil,
283             'ket' => $this->keterangan(),
284             'a' => $na_kurang
285         );
286
287         $stat = $this->session->userdata('lvl');
288         if ($stat == 3) {
289             $this->load->view('ahlijj/header', $data);
290             $this->load->view('ahlijj/perceptron', $res);
291             $this->load->view('footer');
292         } else {
293             redirect('c_login', 'refresh');
294         }
295     }
296 }

```

Gambar 4.12 Program Code proses()


```

5     public function user($nipp) {
6         $this->db->select('*');
7         $this->db->from('pegawai');
8         $this->db->join('jabatan', 'jabatan.kodejabatan = pegawai.KodeJabatan');
9         $this->db->join('wilayah', 'pegawai.WilID = wilayah.wilid');
10        $this->db->where('pegawai.nipp', $nipp);
11        $q = $this->db->get()->result_array();
12        if ($q)
13            return $q[0];
14    }

```

Gambar 4.13 Program Code user(\$nipp)

```

5     public function jml_kolom() {
6         $query = $this->db->query("SELECT (COUNT(*) - 2) AS jml FROM
            information_schema.columns WHERE table_name = 'dataset_fix'");
7         return $query->result_array();
8     }

```

Gambar 4.14 Program Code jml_kolom()

```

10    public function datatrain() {
11        $this->db->select('id, r60 AS "0", r54 AS "1", r50 AS "2", r42 AS "3", beton
            AS "4", kayu AS "5",
12        baja AS "6", ganda AS "7", tunggal AS "8",
13        COALESCE((tba - (SELECT MIN(tba) FROM dataset_fix WHERE hapus = 0)) /
            NULLIF((SELECT MAX(tba) FROM dataset_fix WHERE hapus = 0) - (SELECT
            MIN(tba) FROM dataset_fix WHERE hapus = 0), 0), 0) AS "9",
14        COALESCE((lbb - (SELECT MIN(lbb) FROM dataset_fix WHERE hapus = 0)) /
            NULLIF((SELECT MAX(lbb) FROM dataset_fix WHERE hapus = 0) - (SELECT
            MIN(lbb) FROM dataset_fix WHERE hapus = 0), 0), 0) AS "10",
15        COALESCE((landai - (SELECT MIN(landai) FROM dataset_fix WHERE hapus =
            0)) / NULLIF((SELECT MAX(landai) FROM dataset_fix WHERE hapus = 0) -
            (SELECT MIN(landai) FROM dataset_fix WHERE hapus = 0), 0), 0) AS "11",
16        COALESCE((rminv - (SELECT MIN(rminv) FROM dataset_fix WHERE hapus = 0))
            / NULLIF((SELECT MAX(rminv) FROM dataset_fix WHERE hapus = 0) - (SELECT
            MIN(rminv) FROM dataset_fix WHERE hapus = 0), 0), 0) AS "12",
17        COALESCE((rmintph - (SELECT MIN(rmintph) FROM dataset_fix WHERE hapus =
            0)) / NULLIF((SELECT MAX(rmintph) FROM dataset_fix WHERE hapus = 0) -
            (SELECT MIN(rmintph) FROM dataset_fix WHERE hapus = 0), 0), 0) AS "13",
18        COALESCE((rminph - (SELECT MIN(rminph) FROM dataset_fix WHERE hapus =
            0)) / NULLIF((SELECT MAX(rminph) FROM dataset_fix WHERE hapus = 0) -
            (SELECT MIN(rminph) FROM dataset_fix WHERE hapus = 0), 0), 0) AS "14",
19        COALESCE((lbj - (SELECT MIN(lbj) FROM dataset_fix WHERE hapus = 0)) /
            NULLIF((SELECT MAX(lbj) FROM dataset_fix WHERE hapus = 0) - (SELECT
            MIN(lbj) FROM dataset_fix WHERE hapus = 0), 0), 0) AS "15", vmaks AS
            "16");
20        $this->db->from('dataset_fix');
21        $this->db->where('hapus != 1');
22        $this->db->order_by('id');
23        $q = $this->db->get()->result_array();
24        return $q;
25    }

```

Gambar 4.15 Program Code datatrain()

```

46    public function awal($jmlTitik, $learning_rate) {
47        $this->learning_rate = $learning_rate;
48        $this->jmlTitik = $jmlTitik;
49
50        // Menempatkan pada dan menetapkan bobot dengan nilai awal 0
51        for ($l = 1; $l < 2; $l++) {
52            for ($i = 0; $i < $this->jmlTitik[$l]; $i++) {
53                for ($j = 0; $j < $this->jmlTitik[$l - 1] - 1; $j++) {
54                    $this->titik[$l][$i][$j] = 0;
55                }
56                // bias pada neuron dengan nilai awal 0
57                $this->titik[$l][$i][$this->jmlTitik[$l - 1] - 1] = 0;
58            }
59        }
60
61        // Memulakan nilai perubahan bobot sebelumnya ke 0 untuk pengulangan
        pertama
62        for ($l = 1; $l < 2; $l++) {
63            for ($i = 0; $i < $this->jmlTitik[$l]; $i++) {
64                for ($j = 0; $j < $this->jmlTitik[$l - 1] - 1; $j++) {
65                    $this->perubahan_bobot[$l][$i][$j] = (double) 0.0;
66                }
67            }
68        }
69    }

```

Gambar 4.16 Program Code awal(\$jmlTitik, \$learning_rate)

```

87 public function traintest($data, $numEpoch) {
88     $numPattern = count($data); //Baris atau jumlah data
89     $numInput = count($data[0]) - 1; //Kolom atau jumlah atribut
90
91     /* Proses Training */
92     for ($sep = 0; $sep < $numEpoch; $sep++) {
93         $inputSource = $data[$sep % $numPattern];
94         $target = $data[$sep % $numPattern][$numInput - 1];
95
96         // Menetapkan data ke lapisan input
97         for ($k = 0; $k < $this->jmlTitik[1]; $k++) {
98             for ($l = 0; $l < $this->jmlTitik[0]; $l++) {
99                 $this->keluaran[$k][$l] = $inputSource[$l];
100             }
101         }
102
103         // Menetapkan nilai keluaran
104         for ($l = 1; $l < 2; $l++) {
105             for ($i = 0; $i < $this->jmlTitik[$l]; $i++) {
106                 $sum = 0.0;
107                 for ($j = 0; $j < $this->jmlTitik[$l - 1] - 1; $j++) {
108                     $sum += $this->keluaran[$i][$j] * $this->titik[$l][$i][$j]; //
Menerapkan Penjumlahan dari perkalian antara nilai masukan
dengan bobt
109                 }
110                 $sum += $this->titik[$l][$i][$this->jmlTitik[$l - 1] - 1];
111                 //Menerapkan bias
112                 // Menerapkan fungsi sigmoid
113                 $this->keluaran[$l][$i] = 1.0 / (1.0 + exp(-$sum));
114             }
115         }
116         for ($l = 0; $l < $this->jmlTitik[1]; $l++) {
117             $this->error[1][$l] = ($target - $this->keluaran[1][$l]);
118         }
119
120         for ($l = 1; $l < 2; $l++) {
121             for ($i = 0; $i < $this->jmlTitik[$l]; $i++) {
122                 for ($j = 0; $j < $this->jmlTitik[$l - 1] - 1; $j++) {
123                     /* Perubahan Bobot-bobot */
124                     $this->perubahan_bobot[$l][$i][$j] = $this->learning_rate *
125                     $this->error[1][$i] * $this->keluaran[$i][$j];
126                     $this->titik[$l][$i][$j] += $this->perubahan_bobot[$l][$i][$j];
127                 }
128                 /* Perubahan Bias */
129                 $this->perubahan_bobot[$l][$i][$this->jmlTitik[$l - 1] - 1] =
130                 $this->learning_rate * $this->error[1][$i];
131                 $this->titik[$l][$i][$this->jmlTitik[$l - 1] - 1] +=
132                 $this->perubahan_bobot[$l][$i][$this->jmlTitik[$l - 1] - 1];
133             }
134         }
135
136         /* Proses Testing */
137         $na = array();
138         $hp_pred = array();
139         for ($m = 0; $m < $numPattern; $m++) {
140             $sum = 0.0;
141
142             for ($j = 0; $j < $this->jmlTitik[1]; $j++) {
143                 for ($k = 0; $k < $this->jmlTitik[0]; $k++) {
144                     $this->keluaran[$j][$k] = $data[$m][$k];
145                 }
146             }
147
148             for ($l = 1; $l < 2; $l++) {
149                 for ($i = 0; $i < $this->jmlTitik[$l]; $i++) {
150                     $sum = 0.0;
151                     for ($j = 0; $j < $this->jmlTitik[$l - 1] - 1; $j++) {
152                         $sum += $this->keluaran[$i][$j] * $this->titik[$l][$i][$j];
153                     }
154                     $sum += $this->titik[$l][$i][$this->jmlTitik[$l - 1] - 1];
155                     $this->keluaran[$l][$i] = 1.0 / (1.0 + exp(-$sum));
156                 }
157             }
158
159             $result = (double) $this->keluaran[1][0];
160             $vmaks = $this->result_conv($data[$m][$numInput - 1]);
161             $prediksi = $this->result_conv($result);
162             if ($vmaks != $prediksi) {
163                 $na[$m] = $prediksi / $numPattern;
164                 $hp_pred[$m] = "<a style='background: yellow;'>Data ke-" . ($m +
165                 1) . ". " . $vmaks . " || Hasil: " . $result . " - Hasil
166                 Prediksi: " . $prediksi . "</a>";
167             } else {
168                 $hp_pred[$m] = "Data ke-" . ($m + 1) . ". " . $vmaks . " ||
169                 Hasil: " . $result . " - Hasil Prediksi: " . $prediksi;
170             }
171         }
172
173         $keliru = count($na);
174         $threshold = 0;
175         $akurasi = (($numPattern - $keliru) / $numPattern) * 100;
176         $sepohAkhir = 0;
177         if ($keliru == $threshold) {
178             $numEpoch = $sep + 1;
179             $sepohAkhir = $numEpoch;
180             break;
181         } else {
182             $sepohAkhir = $numEpoch;
183         }
184     }
185
186     $this->ket = array($keliru, $akurasi, $sepohAkhir);
187     return $hp_pred;
188 }

```

Gambar 4.17 Program Code traintest(\$data, \$numEpoch)


```

71 public function result_conv($res) {
72 //Konversi hasil
73 if ($res >= 0.875) {
74 $sts = "120 km/jam";
75 } else if ($res >= 0.625 && $res < 0.875) {
76 $sts = "110 km/jam";
77 } else if ($res >= 0.375 && $res < 0.625) {
78 $sts = "100 km/jam";
79 } else if ($res >= 0.125 && $res < 0.375) {
80 $sts = "90 km/jam";
81 } else {
82 $sts = "80 km/jam";
83 }
84 return $sts;
85 }

```

Gambar 4.18 Program Code result_conv(\$res)

```

78 public function get_hitungakh($nipp) {
79 $query = $this->db->query("SELECT epoch, lr, na, to_char(tgl, 'TMDay, DD
TMMonth YYYY HH24:MI:SS') AS tgl, array_to_json(bobotbias) AS bb FROM
loghitung
80 JOIN pegawai ON loghitung.peg = pegawai.nipp WHERE loghitung.peg = $nipp ORDER BY na
DESC, bobotbias ASC LIMIT 1");
81 return $query->result_array();
82 }

```

Gambar 4.19 Program Code get_hitungakh(\$nipp)

```

74 public function tambah_hitungakh($na, $epoch, $lr, $nipp, $larik) {
75 $this->db->query("INSERT INTO loghitung (na, epoch, lr, peg, tgl, bobotbias)
VALUES ($na, '$epoch', $lr, $nipp, now(), ARRAY$larik)");
76 }

```

Gambar 4.20 Program Code tambah_hitungakh(\$na, \$epoch, \$lr, \$nipp, \$larik)

```

181 public function hasil_bobot() {
182 $h_bo = array();
183 for ($l = 1; $l < 2; $l++) {
184 for ($i = 0; $i < $this->jmlTitik[$l]; $i++) {
185 for ($j = 0; $j < $this->jmlTitik[$l - 1] - 1; $j++) {
186 $h_bo[$i][$j] = "Botot ke-" . $i . ", " . $j . ": " .
$this->titik[$l][$i][$j];
187 }
188 }
189 }
190 return $h_bo;
191 }

```

Gambar 4.21 Program Code hasil_bobot()

```

193 public function hasil_bias() {
194 $h_bi = array();
195 for ($l = 1; $l < 2; $l++) {
196 for ($j = 0; $j < $this->jmlTitik[$l]; $j++) {
197 $h_bi[$j] = "Bias ke-" . $j . ": " .
$this->titik[$l][$j][$this->jmlTitik[$l - 1] - 1];
198 }
199 }
200 return $h_bi;
201 }

```

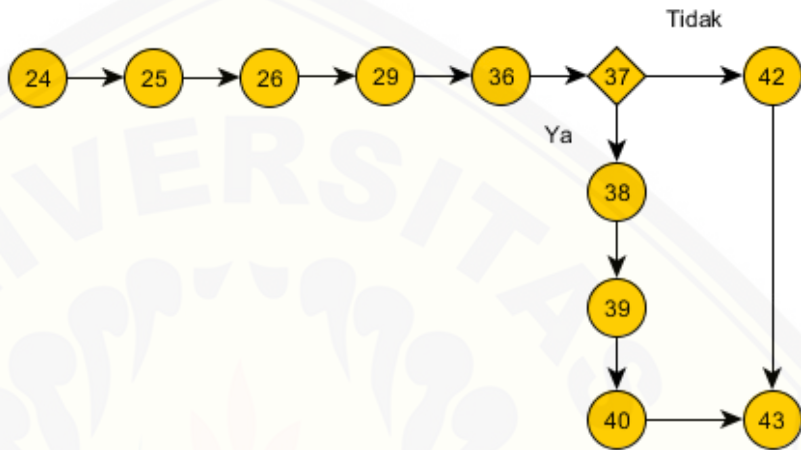
Gambar 4.22 Program Code hasil_bias()

```

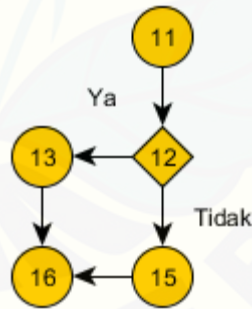
203 public function keterangan() {
204     return "Jumlah data prediksi yang salah: " . $this->ket[0] . "%#13;##10Nilai
        akurasi: " . $this->ket[1] . "% %#13;##10EpoH berakhir pada: " .
        $this->ket[2] . " pengulangan";
205 }
    
```

Gambar 4.23 Program Code keterangan()

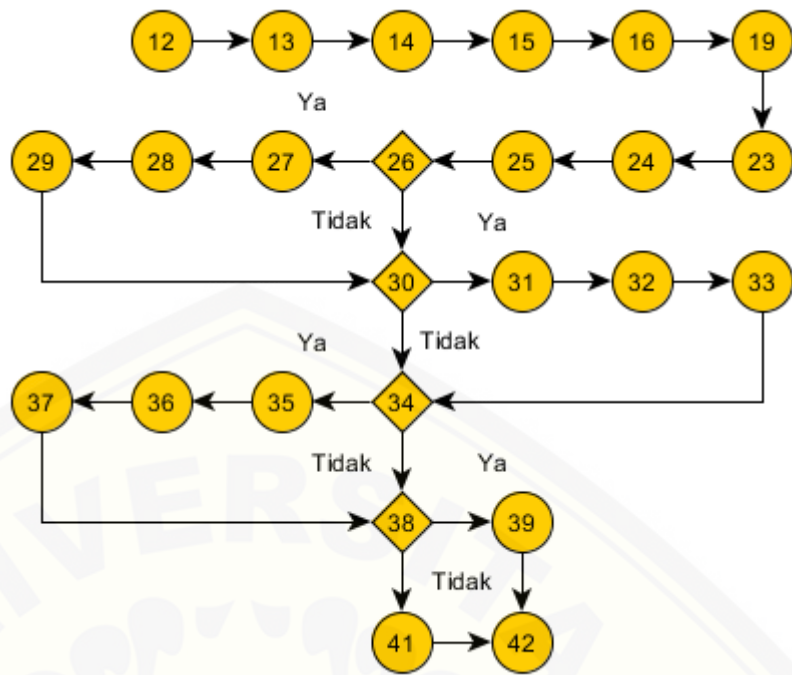
2. Diagram Alir Method



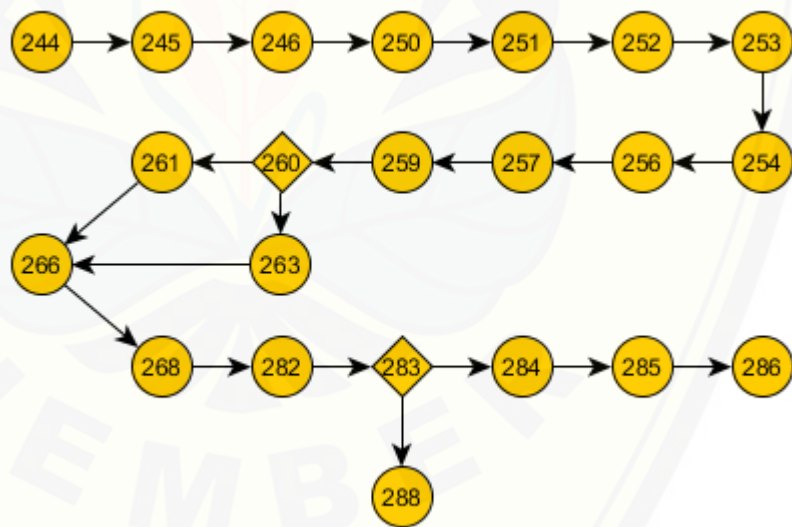
Gambar 4.24 Diagram Alir `index() - c_perceptron.php`



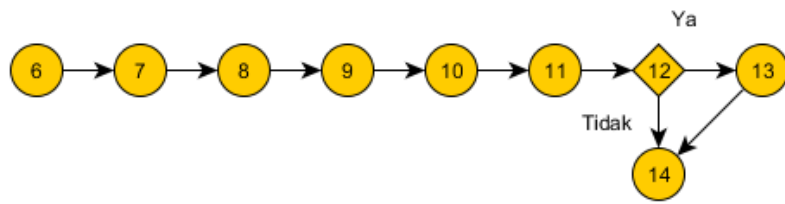
Gambar 4.25 Diagram Alir `index() - c_login.php`



Gambar 4.26 Diagram Alir *index() - c_utama.php*



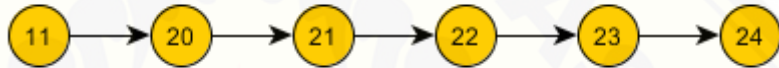
Gambar 4.27 Diagram Alir *proses()*



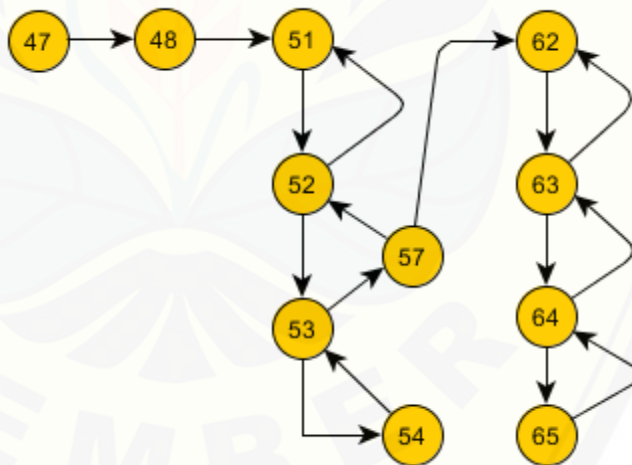
Gambar 4.28 Diagram Alir *user(\$snipp)*



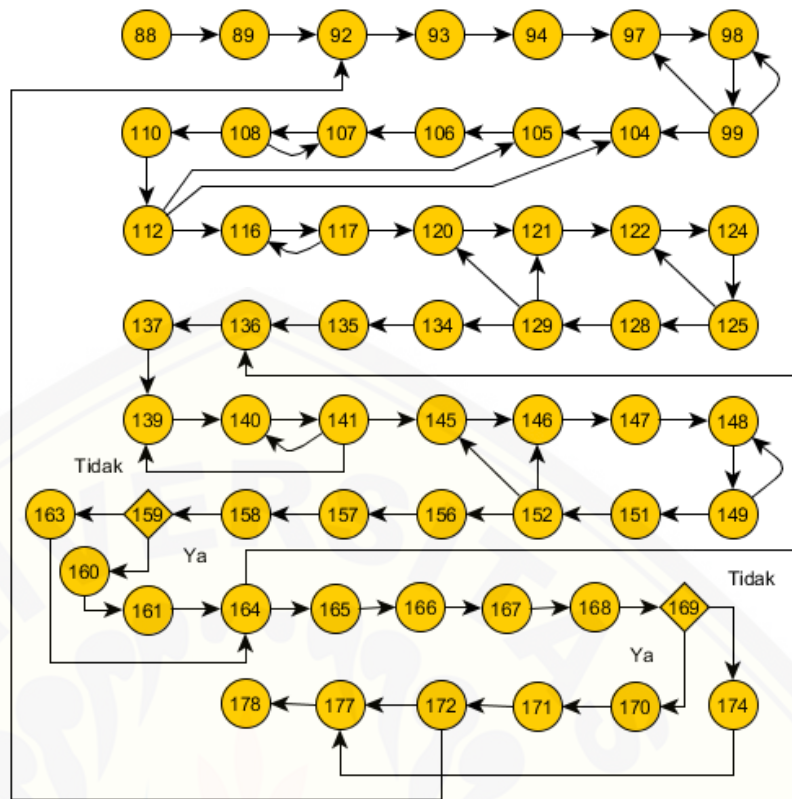
Gambar 4.29 Diagram Alir *jml_kolom()*



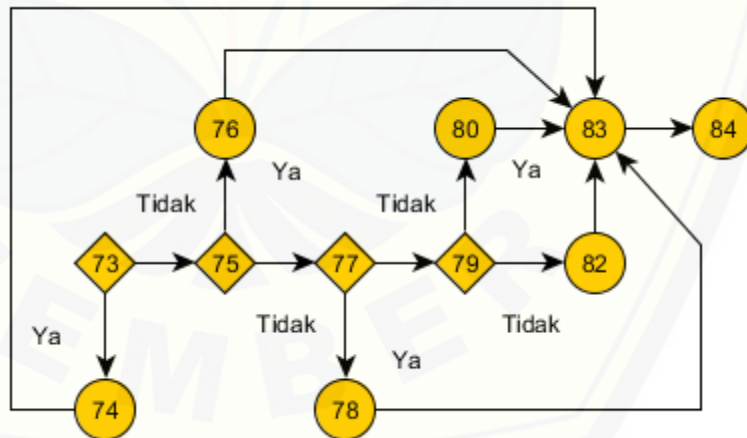
Gambar 4.30 Diagram Alir *datatrain()*



Gambar 4.31 Diagram Alir *awal(\$jmlTitik, \$learning_rate)*



Gambar 4.32 Diagram Alir *traintest(\$data, \$numEpoch)*



Gambar 4.33 Diagram Alir *result_conv(\$res)*



Gambar 4.34 Diagram Alir *get_hitungakh(\$nipp)*



Gambar 4.35 Diagram Alir *tambah_hitungakh(\$na, \$sepoh, \$lr, \$nipp, \$larik)*



Gambar 4.36 Diagram Alir *hasil_bobot()*



Gambar 4.37 Diagram Alir *hasil_bias()*



Gambar 4.38 Diagram Alir *keterangan()*

3. Penghitungan *Cyclomatic Complexity (CC)*

Rumus: $V(G) = Edge - Node + 2$ (7)

index() - c_perceptron.php : $V(G) = 11 - 11 + 2 = 2$

index() - c_login.php : $V(G) = 5 - 5 + 2 = 2$

index() - c_utama.php : $V(G) = 28 - 25 + 2 = 5$

proses() : $V(G) = 22 - 22 + 2 = 2$

user(\$nipp) : $V(G) = 9 - 9 + 2 = 2$

jml_kolom() : $V(G) = 1 - 2 + 2 = 1$

datatrain() : $V(G) = 5 - 6 + 2 = 1$

awal(\$jmlTitik, \$learning_rate) : $V(G) = 16 - 11 + 2 = 7$

traintest(\$data, \$numEpoch) : V(G) = 74 - 57 + 2 = 19

result_conv(\$res) : V(G) = 14 - 11 + 2 = 5

get_hitungakh(\$nipp) : V(G) = 1 - 2 + 2 = 1

tambah_hitungakh(\$na, \$epoh, \$lr, \$nipp, \$larik) : V(G) = 1

hasil_bobot() : V(G) = 4

hasil_bias() : V(G) = 3

keterangan : V(G) = 1

4. Basis Set

index() - c_perceptron.php : 24-25-26-29-36-37-38-39-40-43

(Alur 1)

24-25-26-29-36-37-42-43

(Alur 2)

index() - c_login.php : 11-12-13-16 (Alur 1)

11-12-15-16 (Alur 2)

index() - c_utama.php : 12-13-14-15-16-19-23-24-25-26-27-28-

29-30-34-38-41-42 (Alur 1)

12-13-14-15-16-19-23-24-25-26-30-

31-32-33-34-38-41-42 (Alur 2)

12-13-14-15-16-19-23-24-25-26-30-

34-35-36-37-38-41-42 (Alur 3)

12-13-14-15-16-19-23-24-25-26-30-

34-38-39-42 (Alur 4)

12-13-14-15-16-19-23-24-25-26-30-

34-38-41-42 (Alur 5)

proses() : 244-245-246-250-251-252-253-254-256-

257-259-260-261-266-268-282-283-284-

285-286 (Alur 1)

244-245-246-250-251-252-253-254-256-

257-259-260-261-266-268-282-283-288

(Alur 2)

user(\$nipp) : 6-7-8-9-10-11-12-14 (Alur 1)

6-7-8-9-10-11-13-14 (Alur 2)

jml_kolom() : 6-7 (Alur 1)

datatrain() : 11-20-21-22-23-24 (Alur 1)

awal(\$jmlTitik, \$learning_rate) : 47-48-51-52-51-52-53-54-57-62-63-64-65 (Alur 1)

47-48-51-52-53-57-52-53-54-57-62-63-64-65 (Alur 2)

47-48-51-52-53-54-53-54-57-62-63-64-65 (Alur 3)

47-48-51-52-53-54-57-62-63-62-63-64-65 (Alur 4)

47-48-51-52-53-54-57-62-63-64-63-64-65 (Alur 5)

47-48-51-52-53-54-57-62-63-64-65-64-65 (Alur 6)

47-48-51-52-53-54-57-62-63-64-65 (Alur 7)

traintest(\$data, \$numEpoch) : 88-89-92-93-94-97-98-99-104-105-106-107-108-110-112-116-117-120-121-122-124-125-128-129-134-135-136-137-139-140-141-145-146-147-148-149-151-152-156-157-158-159-160-161-164-165-167-168-169-170-171-172-92-93-94-97-98-99-104-105-106-107-108-110-112-116-117-120-121-122-124-125-128-129-134-135-136-137-139-140-141-145-146-147-148-149-151-152-156-157-158-159-160-161-164-165-167-168-169-170-171-172-177-178 (Alur 1)

88-89-92-93-94-97-98-99-97-98-
99-104-105-106-107-108-110-
112-116-117-120-121-122-124-
125-128-129-134-135-136-137-
139-140-141-145-146-147-148-
149-151-152-156-157-158-159-
163-164-165-167-168-169-170-
171-172-177-178 (Alur 2)

88-89-92-93-94-97-98-99-98-99-
104-105-106-107-108-110-112-
116-117-120-121-122-124-125-
128-129-134-135-136-137-139-
140-141-145-146-147-148-149-
151-152-156-157-158-159-160-
161-164-165-167-168-169-174-
177-178 (Alur 3)

88-89-92-93-94-97-98-99-104-
105-106-107-108-110-112-104-
105-106-107-108-110-112-116-
117-120-121-122-124-125-128-
129-134-135-136-137-139-140-
141-145-146-147-148-149-151-
152-156-157-158-159-163-164-
165-167-168-169-174-177-178
(Alur 4)

88-89-92-93-94-97-98-99-104-
105-106-107-108-110-112-105-
106-107-108-110-112-116-117-
120-121-122-124-125-128-129-
134-135-136-137-139-140-141-
145-146-147-148-149-151-152-
156-157-158-159-160-161-164-

165-167-168-169-170-171-172-
177-178 (Alur 5)

88-89-92-93-94-97-98-99-104-
105-106-107-108-107-108-110-
112-116-117-120-121-122-124-
125-128-129-134-135-136-137-
139-140-141-145-146-147-148-
149-151-152-156-157-158-159-
163-164-165-167-168-169-170-
171-172-177-178 (Alur 6)

88-89-92-93-94-97-98-99-104-
105-106-107-108-110-112-116-
117-116-117-120-121-122-124-
125-128-129-134-135-136-137-
139-140-141-145-146-147-148-
149-151-152-156-157-158-159-
160-161-164-165-167-168-169-
174-177-178 (Alur 7)

88-89-92-93-94-97-98-99-104-
105-106-107-108-110-112-116-
117-120-121-122-124-125-128-
129-120-121-122-124-125-128-
129-134-135-136-137-139-140-
141-145-146-147-148-149-151-
152-156-157-158-159-163-164-
165-167-168-169-174-177-178
(Alur 8)

88-89-92-93-94-97-98-99-104-
105-106-107-108-110-112-116-
117-120-121-122-124-125-128-
129-121-122-124-125-128-129-
134-135-136-137-139-140-141-

145-146-147-148-149-151-152-
156-157-158-159-160-161-164-
165-167-168-169-170-171-172-
177-178 (Alur 9)

88-89-92-93-94-97-98-99-104-
105-106-107-108-110-112-116-
117-120-121-122-124-125-122-
124-125-128-129-134-135-136-
137-139-140-141-145-146-147-
148-149-151-152-156-157-158-
159-163-164-165-167-168-169-
170-171-172-177-178 (Alur 10)

88-89-92-93-94-97-98-99-104-
105-106-107-108-110-112-116-
117-120-121-122-124-125-128-
129-134-135-136-137-139-140-
141-145-146-147-148-149-151-
152-156-157-158-159-160-161-
164-136-137-139-140-141-145-
146-147-148-149-151-152-156-
157-158-159-160-161-164-165-
167-168-169-174-177-178 (Alur
11)

88-89-92-93-94-97-98-99-104-
105-106-107-108-110-112-116-
117-120-121-122-124-125-128-
129-134-135-136-137-139-140-
141-139-140-141-145-146-147-
148-149-151-152-156-157-158-
159-163-164-165-167-168-169-
174-177-178 (Alur 12)

88-89-92-93-94-97-98-99-104-
105-106-107-108-110-112-116-
117-120-121-122-124-125-128-
129-134-135-136-137-139-140-
141-140-141-145-146-147-148-
149-151-152-156-157-158-159-
160-161-164-165-167-168-169-
170-171-172-177-178 (Alur 13)

88-89-92-93-94-97-98-99-104-
105-106-107-108-110-112-116-
117-120-121-122-124-125-128-
129-134-135-136-137-139-140-
141-145-146-147-148-149-151-
152-145-146-147-148-149-151-
152-156-157-158-159-163-164-
165-167-168-169-170-171-172-
177-178 (Alur 14)

88-89-92-93-94-97-98-99-104-
105-106-107-108-110-112-116-
117-120-121-122-124-125-128-
129-134-135-136-137-139-140-
141-145-146-147-148-149-151-
152-146-147-148-149-151-152-
156-157-158-159-160-161-164-
165-167-168-169-174-177-178
(Alur 15)

88-89-92-93-94-97-98-99-104-
105-106-107-108-110-112-116-
117-120-121-122-124-125-128-
129-134-135-136-137-139-140-
141-145-146-147-148-149-148-
149-151-152-156-157-158-159-

163-164-165-167-168-169-174-
177-178 (Alur 16)

88-89-92-93-94-97-98-99-104-
105-106-107-108-110-112-116-
117-120-121-122-124-125-128-
129-134-135-136-137-139-140-
141-145-146-147-148-149-151-
152-156-157-158-159-163-164-
165-167-168-169-170-171-172-
177-178 (Alur 17)

88-89-92-93-94-97-98-99-104-
105-106-107-108-110-112-116-
117-120-121-122-124-125-128-
129-134-135-136-137-139-140-
141-145-146-147-148-149-151-
152-156-157-158-159-163-164-
165-167-168-169-174-177-178
(Alur 18)

88-89-92-93-94-97-98-99-104-
105-106-107-108-110-112-116-
117-120-121-122-124-125-128-
129-134-135-136-137-139-140-
141-145-146-147-148-149-151-
152-156-157-158-159-160-161-
164-165-167-168-169-174-177-
178 (Alur 19)

result_conv(\$res) : 73-74-83-84 (Alur 1)

73-75-76-83-84 (Alur 2)

73-75-77-78-83-84 (Alur 3)

73-75-77-79-80-83-84 (Alur 4)

73-75-77-79-82-83-84 (Alur 5)

get_hitungakh(\$snipp) : 79-81 (Alur 1)

tambah_hitungakh(\$na, \$epoh, \$lr, \$nipp, \$larik) : 75 (Alur 1)

hasil_bobot() : 182-183-184-183-184-185-186-190

(Alur 1)

182-183-184-185-184-185-186-190

(Alur 2)

182-183-184-185-186-185-186-190

(Alur 3)

182-183-184-185-186-190 (Alur 4)

hasil_bias() : 194-195-196-195-196-197-200 (Alur 1)

194-195-196-197-196-197-200 (Alur 2)

194-195-196-197-200 (Alur 3)

keterangan() : 204 (Alur 1)

5. Kasus Uji atau *Test Case*

Tabel 4.21 *Test Case index() – c_perceptron.php*

Alur 1	
<i>Test Case</i>	<i>index() – c_perceptron.php</i>
Target yang Diharapkan	Menampilkan halaman <i>perceptron</i> yang berhak akses Kepala Sub Direktorat Jalan Rel dan Jembatan Pusat
Hasil Pengujian	Benar
Alur / Path	24-25-26-29-36-37-38-39-40-43
Alur 2	
<i>Test Case</i>	<i>index() – c_perceptron.php</i>
Target yang Diharapkan	Kembali ke <i>method c_login</i>
Hasil Pengujian	Benar
Alur / Path	24-25-26-29-36-37-42-43

Tabel 4.22 Test Case *index() – c_login.php*

Alur 1	
Test Case	<i>index() – c_login.php</i>
Target yang Diharapkan	Menampilkan halaman <i>login</i> jika pengguna tidak memiliki hak akses untuk masuk ke SIKEMAKA
Hasil Pengujian	Benar
Alur / Path	11-12-13-16
Alur 2	
Test Case	<i>index() – c_login.php</i>
Target yang Diharapkan	Kembali ke <i>method c_utama</i>
Hasil Pengujian	Benar
Alur / Path	11-12-15-16

Tabel 4.23 Test Case *index() – c_utama.php*

Alur 1	
Test Case	<i>index() – c_utama.php</i>
Target yang Diharapkan	Mengarahkan pengguna sesuai dengan hak akses Manajer Jalan dan Jembatan
Hasil Pengujian	Benar
Alur / Path	12-13-14-15-16-19-23-24-25-26-27-28-29-30-34-38-41-42
Alur 2	
Test Case	<i>index() – c_utama.php</i>
Target yang Diharapkan	Mengarahkan pengguna sesuai dengan hak akses Asisten Manajer Konstruksi Jalan dan Jembatan
Hasil Pengujian	Benar
Alur / Path	12-13-14-15-16-19-23-24-25-26-30-31-32-33-34-38-41-42
Alur 3	
Test Case	<i>index() – c_utama.php</i>
Target yang Diharapkan	Mengarahkan pengguna sesuai dengan hak akses Kepala Sub Direktorat Jalan dan Jembatan Pusat
Hasil Pengujian	Benar
Alur / Path	12-13-14-15-16-19-23-24-25-26-30-34-35-36-37-38-41-42
Alur 4	
Test Case	<i>index() – c_utama.php</i>
Target yang Diharapkan	Mengarahkan pengguna ke situs http://kereta-api.co.id
Hasil Pengujian	Benar

Alur / Path	12-13-14-15-16-19-23-24-25-26-30-34-38-39-42
Alur 5	
Test Case	<i>index() – c_utama.php</i>
Target yang Diharapkan	Kembali ke <i>method c_login</i>
Hasil Pengujian	Benar
Alur / Path	12-13-14-15-16-19-23-24-25-26-30-34-38-41-42

Tabel 4.24 Test Case proses()

Alur 1	
Test Case	<i>proses()</i>
Target yang Diharapkan	Mengirim variabel data hasil proses klasifikasi <i>dataset</i> ke <i>view</i> klasifikasi SLP pada <i>dataset</i>
Hasil Pengujian	Benar
Alur / Path	244-245-246-250-251-252-253-254-256-257-259-260-261-266-268-282-283-284-285-286
Alur 2	
Test Case	<i>proses()</i>
Target yang Diharapkan	Kembali ke <i>method c_login</i>
Hasil Pengujian	Benar
Alur / Path	244-245-246-250-251-252-253-254-256-257-259-260-261-266-268-282-283-288

Tabel 4.25 Test Case user(\$nipp)

Alur 1	
Test Case	<i>user(\$nipp)</i>
Target yang Diharapkan	Mengambil data pengguna dari basis data berdasarkan variabel <i>\$nipp</i> yang didapat dari <i>session</i> hak akses SIKEMAKA
Hasil Pengujian	Benar
Alur / Path	6-7-8-9-10-11-12-14
Alur 2	
Test Case	<i>user(\$nipp)</i>
Target yang Diharapkan	Proses permissalan tersebut selesai
Hasil Pengujian	Benar
Alur / Path	6-7-8-9-10-11-13-14

Tabel 4.26 Test Case *jml_kolom()*

Alur 1	
<i>Test Case</i>	<i>jml_kolom()</i>
Target yang Diharapkan	Mendapatkan nilai jumlah kolom dari tabel <i>dataset_fix</i> yang tersimpan pada basis data
Hasil Pengujian	Benar
Alur / Path	6-7

Tabel 4.27 Test Case *datatrain()*

Alur 1	
<i>Test Case</i>	<i>datatrain()</i>
Target yang Diharapkan	Mendapatkan nilai atribut-atribut beserta nilai kelas dari basis data di mana atribut-atribut tersebut telah mengalami proses normalisasi <i>min-max</i> pada <i>query</i> tersebut
Hasil Pengujian	Benar
Alur / Path	11-20-21-22-23-24

Tabel 4.28 Test Case *awal(\$jmlTitik, \$learning_rate)*

Alur 1	
<i>Test Case</i>	<i>awal(\$jmlTitik, \$learning_rate)</i>
Target yang Diharapkan	Melakukan pengulangan yang menggunakan variabel \$l sebanyak satu lapisan untuk mendapatkan nilai awal bobot dan bias sama dengan 0
Hasil Pengujian	Benar
Alur / Path	47-48-51-52-51-52-53-54-57-62-63-64-65
Alur 2	
<i>Test Case</i>	<i>awal(\$jmlTitik, \$learning_rate)</i>
Target yang Diharapkan	Melakukan pengulangan yang menggunakan variabel \$i sebanyak satu bias untuk mendapatkan nilai awal bobot dan bias sama dengan 0
Hasil Pengujian	Benar
Alur / Path	47-48-51-52-53-57-52-53-54-57-62-63-64-65
Alur 3	
<i>Test Case</i>	<i>awal(\$jmlTitik, \$learning_rate)</i>
Target yang Diharapkan	Melakukan pengulangan yang menggunakan variabel \$j sebanyak jumlah titik bobot yang dikurangi satu untuk mendapatkan nilai awal bobot dan bias

	sama dengan 0
Hasil Pengujian	Benar
Alur / Path	47-48-51-52-53-54-53-54-57-62-63-64-65
Alur 4	
<i>Test Case</i>	<i>awal(\$jmlTitik, \$learning_rate)</i>
Target yang Diharapkan	Melakukan pengulangan yang menggunakan variabel \$l sebanyak satu lapisan untuk mendapatkan nilai awal perubahan bobot sama dengan 0
Hasil Pengujian	Benar
Alur / Path	47-48-51-52-53-54-57-62-63-62-63-64-65
Alur 5	
<i>Test Case</i>	<i>awal(\$jmlTitik, \$learning_rate)</i>
Target yang Diharapkan	Melakukan pengulangan yang menggunakan variabel \$i sebanyak satu bias untuk mendapatkan nilai awal perubahan bobot sama dengan 0
Hasil Pengujian	Benar
Alur / Path	47-48-51-52-53-54-57-62-63-64-63-64-65
Alur 6	
<i>Test Case</i>	<i>awal(\$jmlTitik, \$learning_rate)</i>
Target yang Diharapkan	Melakukan pengulangan yang menggunakan variabel \$j sebanyak jumlah titik bobot yang dikurangi satu untuk mendapatkan nilai awal perubahan bobot sama dengan 0
Hasil Pengujian	Benar
Alur / Path	47-48-51-52-53-54-57-62-63-64-65-64-65
Alur 7	
<i>Test Case</i>	<i>awal(\$jmlTitik, \$learning_rate)</i>
Target yang Diharapkan	Menghasilkan nilai 0 sebagai nilai awal bobot, bias, dan perubahan bobot
Hasil Pengujian	Benar
Alur / Path	47-48-51-52-53-54-57-62-63-64-65

Tabel 4.29 Test Case *traintest(\$data, \$numEpoch)*

Alur 1	
<i>Test Case</i>	<i>traintest(\$data, \$numEpoch)</i>
Target yang Diharapkan	Melakukan pengulangan klasifikasi dengan batas pengulangan adalah nilai epoch
Hasil Pengujian	Benar
Alur / Path	88-89-92-93-94-97-98-99-104-105-106-107-108-110-112-116-117-120-121-122-124-125-128-129-134-135-136-137-139-140-141-145-146-147-148-149-151-152-156-157-158-159-160-161-164-165-167-168-169-170-171-172-92-93-94-97-98-99-104-105-106-107-108-110-112-116-117-120-121-122-124-125-128-129-134-135-136-137-139-140-141-145-146-147-148-149-151-152-156-157-158-159-160-161-164-165-167-168-169-170-171-172-177-178
Alur 2	
<i>Test Case</i>	<i>traintest(\$data, \$numEpoch)</i>
Target yang Diharapkan	Melakukan pengulangan pada variabel <i>keluaran</i> sebanyak jumlah atribut <i>dataset</i> untuk menetapkan data dari <i>dataset</i> ke lapisan masukan pada proses <i>training</i>
Hasil Pengujian	Benar
Alur / Path	88-89-92-93-94-97-98-99-97-98-99-104-105-106-107-108-110-112-116-117-120-121-122-124-125-128-129-134-135-136-137-139-140-141-145-146-147-148-149-151-152-156-157-158-159-163-164-165-167-168-169-170-171-172-177-178
Alur 3	
<i>Test Case</i>	<i>traintest(\$data, \$numEpoch)</i>
Target yang Diharapkan	Melakukan pengulangan pada variabel <i>keluaran</i> sebanyak satu kali untuk menetapkan data dari <i>dataset</i> ke lapisan masukan pada proses <i>training</i>
Hasil Pengujian	Benar
Alur / Path	88-89-92-93-94-97-98-99-98-99-104-105-106-107-108-110-112-116-117-120-121-122-124-125-128-129-134-135-136-137-139-140-141-145-146-147-148-149-151-152-156-157-158-159-160-161-164-165-167-168-169-174-177-178

Alur 4	
<i>Test Case</i>	<i>traintest(\$data, \$numEpoch)</i>
Target yang Diharapkan	Melakukan pengulangan yang menggunakan variabel \$l sebanyak satu lapisan untuk mendapatkan nilai y_{in} pada proses <i>training</i>
Hasil Pengujian	Benar
Alur / Path	88-89-92-93-94-97-98-99-104-105-106-107-108-110-112-104-105-106-107-108-110-112-116-117-120-121-122-124-125-128-129-134-135-136-137-139-140-141-145-146-147-148-149-151-152-156-157-158-159-163-164-165-167-168-169-174-177-178
Alur 5	
<i>Test Case</i>	<i>traintest(\$data, \$numEpoch)</i>
Target yang Diharapkan	Melakukan pengulangan yang menggunakan variabel \$i sebanyak satu bias untuk mendapatkan nilai y_{in} pada proses <i>training</i>
Hasil Pengujian	Benar
Alur / Path	88-89-92-93-94-97-98-99-104-105-106-107-108-110-112-105-106-107-108-110-112-116-117-120-121-122-124-125-128-129-134-135-136-137-139-140-141-145-146-147-148-149-151-152-156-157-158-159-160-161-164-165-167-168-169-170-171-172-177-178
Alur 6	
<i>Test Case</i>	<i>traintest(\$data, \$numEpoch)</i>
Target yang Diharapkan	Melakukan pengulangan yang menggunakan variabel \$j sebanyak jumlah titik bobot yang dikurangi satu untuk mendapatkan nilai y_{in} pada proses <i>training</i>
Hasil Pengujian	Benar
Alur / Path	88-89-92-93-94-97-98-99-104-105-106-107-108-107-108-110-112-116-117-120-121-122-124-125-128-129-134-135-136-137-139-140-141-145-146-147-148-149-151-152-156-157-158-159-163-164-165-167-168-169-170-171-172-177-178
Alur 7	
<i>Test Case</i>	<i>traintest(\$data, \$numEpoch)</i>
Target yang Diharapkan	Melakukan pengulangan nilai <i>error</i> sebanyak jumlah atribut
Hasil Pengujian	Benar
Alur / Path	88-89-92-93-94-97-98-99-104-105-106-

	107-108-110-112-116-117-116-117-120-121-122-124-125-128-129-134-135-136-137-139-140-141-145-146-147-148-149-151-152-156-157-158-159-160-161-164-165-167-168-169-174-177-178
Alur 8	
<i>Test Case</i>	<i>traintest(\$data, \$numEpoch)</i>
Target yang Diharapkan	Melakukan perulangan yang menggunakan variabel \$l sebanyak satu lapisan untuk mendapatkan nilai perubahan bobot dan bias
Hasil Pengujian	Benar
Alur / Path	88-89-92-93-94-97-98-99-104-105-106-107-108-110-112-116-117-120-121-122-124-125-128-129-120-121-122-124-125-128-129-134-135-136-137-139-140-141-145-146-147-148-149-151-152-156-157-158-159-163-164-165-167-168-169-174-177-178
Alur 9	
<i>Test Case</i>	<i>traintest(\$data, \$numEpoch)</i>
Target yang Diharapkan	Melakukan perulangan yang menggunakan variabel \$i sebanyak satu bias untuk mendapatkan nilai perubahan bobot dan bias
Hasil Pengujian	Benar
Alur / Path	88-89-92-93-94-97-98-99-104-105-106-107-108-110-112-116-117-120-121-122-124-125-128-129-121-122-124-125-128-129-134-135-136-137-139-140-141-145-146-147-148-149-151-152-156-157-158-159-160-161-164-165-167-168-169-170-171-172-177-178
Alur 10	
<i>Test Case</i>	<i>traintest(\$data, \$numEpoch)</i>
Target yang Diharapkan	Melakukan perulangan yang menggunakan variabel \$j sebanyak jumlah titik bobot yang dikurangi satu untuk mendapatkan nilai perubahan bobot dan bias
Hasil Pengujian	Benar
Alur / Path	88-89-92-93-94-97-98-99-104-105-106-107-108-110-112-116-117-120-121-122-124-125-122-124-125-128-129-134-135-136-137-139-140-141-145-146-147-148-149-151-152-156-157-158-159-163-164-165-167-168-169-170-171-172-177-178
Alur 11	

<i>Test Case</i>	<i>traintest(\$data, \$numEpoch)</i>
Target yang Diharapkan	Melakukan pengulangan sebanyak jumlah data untuk proses <i>testing</i>
Hasil Pengujian	Benar
Alur / Path	88-89-92-93-94-97-98-99-104-105-106-107-108-110-112-116-117-120-121-122-124-125-128-129-134-135-136-137-139-140-141-145-146-147-148-149-151-152-156-157-158-159-160-161-164-136-137-139-140-141-145-146-147-148-149-151-152-156-157-158-159-160-161-164-165-167-168-169-174-177-178
Alur 12	
<i>Test Case</i>	<i>traintest(\$data, \$numEpoch)</i>
Target yang Diharapkan	Melakukan pengulangan pada variabel <i>keluaran</i> sebanyak jumlah atribut <i>dataset</i> untuk menetapkan data dari <i>dataset</i> ke lapisan masukan pada proses <i>testing</i>
Hasil Pengujian	Benar
Alur / Path	88-89-92-93-94-97-98-99-104-105-106-107-108-110-112-116-117-120-121-122-124-125-128-129-134-135-136-137-139-140-141-139-140-141-145-146-147-148-149-151-152-156-157-158-159-163-164-165-167-168-169-174-177-178
Alur 13	
<i>Test Case</i>	<i>traintest(\$data, \$numEpoch)</i>
Target yang Diharapkan	Melakukan pengulangan pada variabel <i>keluaran</i> sebanyak satu kali untuk menetapkan data dari <i>dataset</i> ke lapisan masukan pada proses <i>testing</i>
Hasil Pengujian	Benar
Alur / Path	88-89-92-93-94-97-98-99-104-105-106-107-108-110-112-116-117-120-121-122-124-125-128-129-134-135-136-137-139-140-141-140-141-145-146-147-148-149-151-152-156-157-158-159-160-161-164-165-167-168-169-170-171-172-177-178
Alur 14	
<i>Test Case</i>	<i>traintest(\$data, \$numEpoch)</i>
Target yang Diharapkan	Melakukan pengulangan yang menggunakan variabel <i>\$l</i> sebanyak satu lapisan untuk mendapatkan nilai <i>y_in</i> pada proses <i>testing</i>
Hasil Pengujian	Benar
Alur / Path	88-89-92-93-94-97-98-99-104-105-106-107-108-110-112-116-117-120-121-122-

	124-125-128-129-134-135-136-137-139-140-141-145-146-147-148-149-151-152-145-146-147-148-149-151-152-156-157-158-159-163-164-165-167-168-169-170-171-172-177-178
Alur 15	
<i>Test Case</i>	<i>traintest(\$data, \$numEpoch)</i>
Target yang Diharapkan	Melakukan pengulangan yang menggunakan variabel \$i sebanyak satu bias untuk mendapatkan nilai y_{in} pada proses <i>testing</i>
Hasil Pengujian	Benar
Alur / Path	88-89-92-93-94-97-98-99-104-105-106-107-108-110-112-116-117-120-121-122-124-125-128-129-134-135-136-137-139-140-141-145-146-147-148-149-151-152-146-147-148-149-151-152-156-157-158-159-160-161-164-165-167-168-169-174-177-178
Alur 16	
<i>Test Case</i>	<i>traintest(\$data, \$numEpoch)</i>
Target yang Diharapkan	Melakukan pengulangan yang menggunakan variabel \$j sebanyak jumlah titik bobot yang dikurangi satu untuk mendapatkan nilai y_{in} pada proses <i>training</i>
Hasil Pengujian	Benar
Alur / Path	88-89-92-93-94-97-98-99-104-105-106-107-108-110-112-116-117-120-121-122-124-125-128-129-134-135-136-137-139-140-141-145-146-147-148-149-148-149-151-152-156-157-158-159-163-164-165-167-168-169-174-177-178
Alur 17	
<i>Test Case</i>	<i>traintest(\$data, \$numEpoch)</i>
Target yang Diharapkan	Melakukan permisalan pada proses <i>testing</i> dengan hasil salah, sehingga jika nilai kecepatan maksimum tidak sama dengan nilai prediksi, maka proses permisalan tersebut selesai
Hasil Pengujian	Benar
Alur / Path	88-89-92-93-94-97-98-99-104-105-106-107-108-110-112-116-117-120-121-122-124-125-128-129-134-135-136-137-139-140-141-145-146-147-148-149-151-152-156-157-158-159-163-164-165-167-168-169-170-171-172-177-178
Alur 18	

<i>Test Case</i>	<i>traintest(\$data, \$numEpoch)</i>
Target yang Diharapkan	Melakukan permisalan pada proses <i>testing</i> dengan hasil yang salah, sehingga jika nilai data yang keliru sama dengan nilai <i>threshold</i> , maka didapatkan nilai epoch akhir sama dengan nilai epoch masukan
Hasil Pengujian	Benar
Alur / Path	88-89-92-93-94-97-98-99-104-105-106-107-108-110-112-116-117-120-121-122-124-125-128-129-134-135-136-137-139-140-141-145-146-147-148-149-151-152-156-157-158-159-163-164-165-167-168-169-174-177-178
Alur 19	
<i>Test Case</i>	<i>traintest(\$data, \$numEpoch)</i>
Target yang Diharapkan	Menghasilkan prediksi klasifikasi SLP pada <i>dataset</i>
Hasil Pengujian	Benar
Alur / Path	88-89-92-93-94-97-98-99-104-105-106-107-108-110-112-116-117-120-121-122-124-125-128-129-134-135-136-137-139-140-141-145-146-147-148-149-151-152-156-157-158-159-160-161-164-165-167-168-169-174-177-178

Tabel 4.30 Test Case *result_conv(\$res)*

Alur 1	
<i>Test Case</i>	<i>result_conv(\$res)</i>
Target yang Diharapkan	Menghasilkan nilai \$res = “120 km/jam”
Hasil Pengujian	Benar
Alur / Path	73-74-83-84
Alur 2	
<i>Test Case</i>	<i>result_conv(\$res)</i>
Target yang Diharapkan	Menghasilkan nilai \$res = “110 km/jam”
Hasil Pengujian	Benar
Alur / Path	73-75-76-83-84
Alur 3	
<i>Test Case</i>	<i>result_conv(\$res)</i>
Target yang Diharapkan	Menghasilkan nilai \$res = “100 km/jam”
Hasil Pengujian	Benar
Alur / Path	73-75-77-78-83-84
Alur 4	
<i>Test Case</i>	<i>result_conv(\$res)</i>
Target yang Diharapkan	Menghasilkan nilai \$res = “90 km/jam”
Hasil Pengujian	Benar
Alur / Path	73-75-77-79-80-83-84
Alur 5	
<i>Test Case</i>	<i>result_conv(\$res)</i>
Target yang Diharapkan	Menghasilkan nilai \$res = “80 km/jam”
Hasil Pengujian	Benar
Alur / Path	73-75-77-79-82-83-84

Tabel 4.31 Test Case *get_hitungakh(\$nipp)*

Alur 1	
<i>Test Case</i>	<i>get_hitungakh(\$nipp)</i>
Target yang Diharapkan	Mendapatkan nilai-nilai penghitungan terakhir dari basis data berdasarkan variabel <i>\$nipp</i> yang didapat dari <i>session</i> hak akses SIKEMAKA yang dapat mengelola klasifikasi SLP pada <i>dataset</i>
Hasil Pengujian	Benar

Alur / Path	79-81
-------------	-------

Tabel 4.32 Test Case tambah_hitungakh(\$na, \$epoh, \$lr, \$nipp, \$larik)

Alur 1	
Test Case	tambah_hitungakh(\$na, \$epoh, \$lr, \$nipp, \$larik)
Target yang Diharapkan	Menambahkan nilai-nilai penghitungan terakhir yang didapat dari hasil klasifikasi SLP pada <i>dataset</i> ke basis data
Hasil Pengujian	Benar
Alur / Path	75

Tabel 4.33 Test Case hasil_bobot()

Alur 1	
Test Case	hasil_bobot()
Target yang Diharapkan	Melakukan pengulangan yang menggunakan variabel \$l sebanyak satu lapisan untuk mendapatkan nilai bobot hasil klasifikasi
Hasil Pengujian	Benar
Alur / Path	182-183-184-183-184-185-186-190
Alur 2	
Test Case	hasil_bobot()
Target yang Diharapkan	Melakukan pengulangan yang menggunakan variabel \$i sebanyak satu bias untuk mendapatkan nilai bobot hasil klasifikasi
Hasil Pengujian	Benar
Alur / Path	182-183-184-185-184-185-186-190
Alur 3	
Test Case	hasil_bobot()
Target yang Diharapkan	Melakukan pengulangan yang menggunakan variabel \$j sebanyak jumlah titik bobot yang dikurangi satu untuk mendapatkan nilai bobot hasil klasifikasi
Hasil Pengujian	Benar
Alur / Path	182-183-184-185-186-185-186-190
Alur 4	
Test Case	hasil_bobot()
Target yang Diharapkan	Melakukan pengolahan format nilai bobot dari hasil klasifikasi
Hasil Pengujian	Benar

Alur / Path	182-183-184-185-186-190
-------------	-------------------------

Tabel 4.34 Test Case hasil_bias()

Alur 1	
Test Case	hasil_bias()
Target yang Diharapkan	Melakukan pengulangan yang menggunakan variabel \$l sebanyak satu lapisan untuk mendapatkan nilai bias hasil klasifikasi
Hasil Pengujian	Benar
Alur / Path	194-195-196-195-196-197-200
Alur 2	
Test Case	hasil_bias()
Target yang Diharapkan	Melakukan pengulangan yang menggunakan variabel \$j sebanyak satu bias yang untuk mendapatkan nilai bias hasil klasifikasi
Hasil Pengujian	Benar
Alur / Path	194-195-196-197-196-197-200
Alur 3	
Test Case	hasil_bias()
Target yang Diharapkan	Melakukan pengolahan format nilai bias dari hasil klasifikasi
Hasil Pengujian	Benar
Alur / Path	194-195-196-197-200

Tabel 4.35 Test Case keterangan()

Alur 1	
Test Case	keterangan()
Target yang Diharapkan	Melakukan pengolahan format hasil keterangan klasifikasi berupa jumlah data prediksi yang salah, nilai akurasi, dan epoch terakhir dari pengulangan klasifikasi yang telah dilakukan tersebut
Hasil Pengujian	Benar
Alur / Path	204

4.5.2 Pengujian Black Box

Pengujian *Black Box* merupakan pengujian sistem yang dilakukan pada fungsionalitas sistem tanpa pengetahuan *source code* sistem yang digunakan pada penelitian ini. Dokumentasi tersebut dapat ditinjau pada Lampiran D.2.

4.5.3 Pengujian Validasi

Pengujian validasi merupakan pengujian dengan cara perbandingan antara hasil prediksi dengan data yang telah valid. Pada penelitian ini, hasil prediksi merupakan nilai kecepatan maksimum yang didapat dari proses klasifikasi tersebut. Sedangkan data yang telah valid merupakan nilai kecepatan maksimum yang didapat dari tabel kelas jalan rel untuk jalur lurus dengan memperhatikan atribut-atribut studi kasus tersebut, tabel jari-jari minimum lengkung horizontal, tabel jari-jari minimum lengkung vertikal, dan nomor 3 subbab 1.4. Selain itu, pengujian validasi ini juga menggunakan metode *K-Fold Cross Validation* untuk menguji *dataset* dengan cara pembagian sebanyak tiga bagian, lalu dilakukan proses penghitungan nilai akurasi sebanyak tiga kali dengan setiap bagian adalah *test-set* dan dua bagian lain sebagai *train-set* sehingga metode ini menghasilkan pula tiga nilai akurasi untuk setiap bagian *dataset* tersebut.

4.6 Pemeliharaan Sistem

Tahapan Pemeliharaan Sistem merupakan tahapan untuk pencarian kesalahan-kesalahan atau *bug* pada sistem yang telah dibuat di mana kesalahan-kesalahan tersebut muncul dikarenakan kondisi-kondisi tertentu seperti perubahan kebutuhan-kebutuhan dan performa sistem. Jika muncul kesalahan-kesalahan tersebut, maka sistem akan diubah pada tahap tertentu yang terus berkelanjutan hingga tahap ini sehingga kesalahan tersebut dapat teratasi sesuai dengan alur Model *Waterfall*.

BAB 6. PENUTUP

Bab ini menjelaskan kesimpulan dan saran yang diperoleh berdasarkan hasil dan pembahasan yang telah dipaparkan pada bab 5. Adapun kesimpulan merupakan pembuktian dari perumusan hipotesis dan saran merupakan anjuran yang dapat menjadi keterangan untuk pengembangan penelitian atau penerapan ini di masa mendatang.

6.1. Kesimpulan

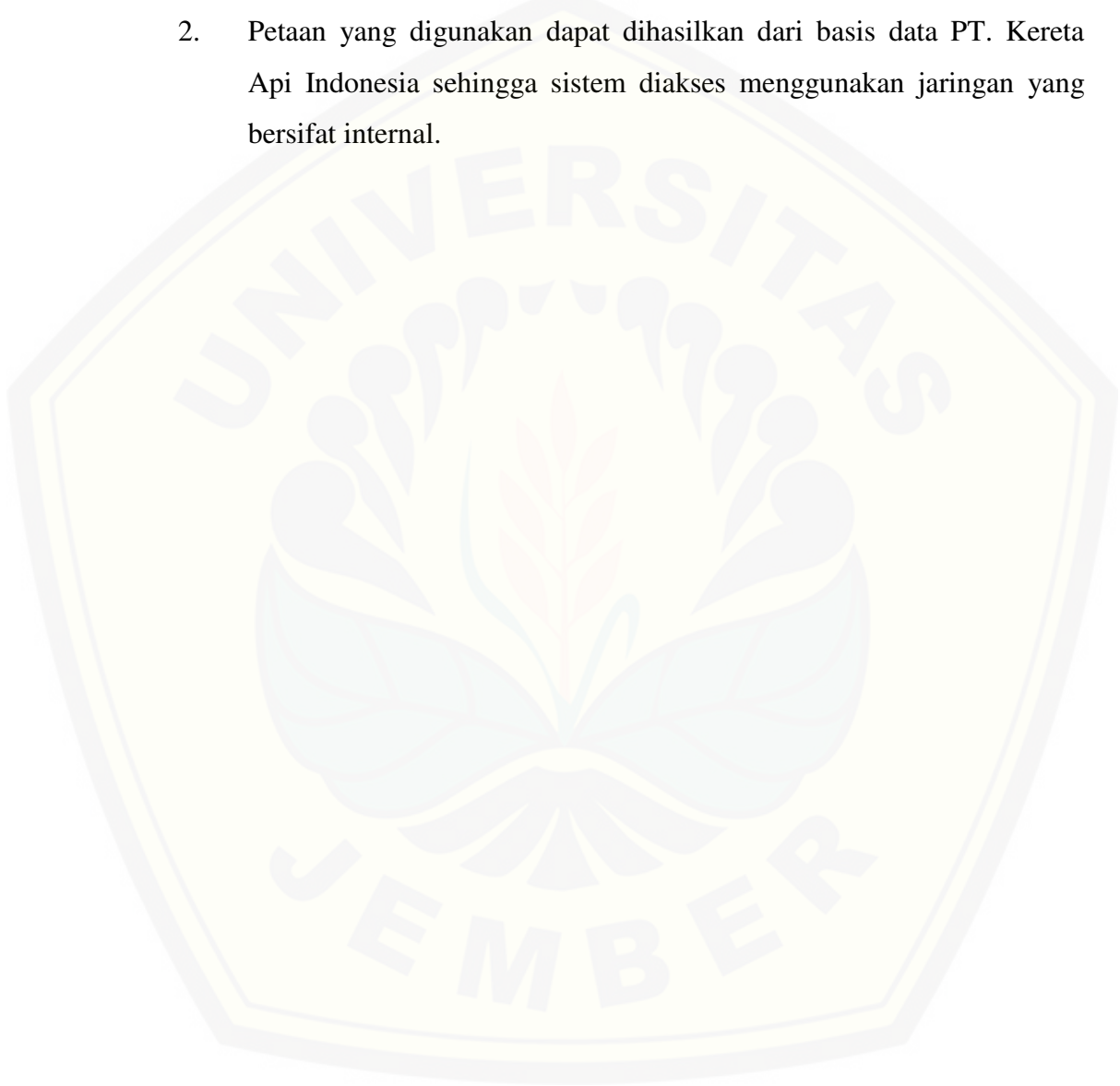
Kesimpulan dari penelitian yang telah dilakukan ini sebagai berikut:

1. Sistem Pengklasifikasi Kecepatan Maksimum Kereta Api (SIKEMAKA) merupakan suatu sistem informasi yang dapat melakukan klasifikasi *dataset* kecepatan maksimum dan klasifikasi data baru untuk menentukan nilai kecepatan maksimum menggunakan metode *Single Layer Perceptron* dengan objek penelitian Jalur Kereta Api Klakah-Pasirian.
2. Nilai bobot dan bias yang didapatkan dari klasifikasi SLP pada *dataset* dapat digunakan sebagai nilai bobot dan bias untuk klasifikasi SLP pada data baru karena nilai akurasi dari hasil klasifikasi adalah 100% dengan nilai epoch berakhir pada pengulangan ke-1806 dan *learning rate* adalah 0,1.
3. Nilai kecepatan maksimum dari hasil klasifikasi SLP pada Jalur Kereta Api Klakah-Pasirian bervariasi mulai 80 km/jam hingga 110 km/jam.
4. SIKEMAKA dapat memetakan kecepatan maksimum yang dibentuk dalam hal perancangan dan laporan hasil dengan objek penelitian Jalur Kereta Api Klakah-Pasirian.

6.2. Saran

Saran yang diharapkan dari penelitian yang telah dilakukan ini sebagai berikut:

1. Sistem perlu penetapan berbagai data yang dibutuhkan untuk menentukan kecepatan maksimum kereta api berdasarkan *Standart Operational Procedure* (SOP) dari PT. Kereta Api Indonesia.
2. Petaan yang digunakan dapat dihasilkan dari basis data PT. Kereta Api Indonesia sehingga sistem diakses menggunakan jaringan yang bersifat internal.



DAFTAR PUSTAKA

Buku dan Buku Elektronik

- Ditjen Perkeretaapian Kementerian Perhubungan. 2011. *Rencana Induk Perkeretaapian Nasional*. Jakarta: Ditjen Perkeretaapian Kemenhub RI.
- Dwiatmoko, H. 2013. *Keselamatan Jalur & Bangunan Kereta Api*. Jakarta: Kencana Prenada Media Group.
- Fuller, Robert. 2010. *The Delta Learning Rule - Tutorial*. Turku: Abo Akademi University.
- Kartikasari, D., & Widyani, E. 2007. *Perencanaan Jalur Ganda Kereta Api dari Stasiun Pekalongan ke Stasiun Tegal* (Tidak Diterbitkan. Skripsi ed.). Semarang: Fakultas Teknik Universitas Diponegoro.
- Leung, K. M. 2008. *Perceptron For Pattern Classification*. New York: New York University Tandon School of Engineering.
- Raharja, Y. P. 2014. *Rancang Bangun Sistem Rekomendasi Beasiswa Menggunakan Algoritma Klasifikasi C4.5 pada Universitas Dian Nuswantoro* (Tidak Diterbitkan. Skripsi ed.). Semarang: Universitas Dian Nuswantoro.

Jurnal

- Arifianto, D. T., & Prastyanto, C. A. 2012. *Desain Geometrik, Struktur beserta Perkiraan Biaya Perencanaan Jalan Rel sebagai Alternatif Transportasi Angkutan Tambang Pasir di Kabupaten Lumajang*. *Jurnal Teknik Pomits Vol. 1, No. 1, (2012) 1-6*. Surabaya: Institut Teknologi Sepuluh Nopember.
- Suryani, S., Sasongko, P., & Suharto, E. 2012. *Sistem Informasi Geografis Pemetaan Sekolah Tingkat Pendidikan Dasar dan Menengah di Kota Serang*. *Jurnal Masyarakat Informatika, Volume 2, Nomor 3, ISSN 2086-4930*. Semarang: Universitas Diponegoro.
- Wuryandari, M. D., & Afrianto, I. 2012. *Perbandingan Metode Jaringan Syaraf Tiruan Backpropagation dan Learning Vector Quantization pada*

Pengenalan Wajah. Jurnal Komputer dan Informatika (KOMPUTA), Edisi I, Volume I. Bandung: Universitas Komputer Indonesia.

Internet

Atmawati, Ersti Ulfa. 2011. *Jaringan Syaraf Tiruan*. http://ersti-fst08.web.unair.ac.id/artikel_detail-35314-Kuliah%20Komputer%20Cerdas-Jaringan%20Syaraf%20Tiruan.html. [26 November 2015].

Peta

Badan Perencanaan dan Pembangunan Daerah Kabupaten Lumajang. 2012. *Peta Rencana Jaringan Kereta Api – Rencana Tata Ruang dan Wilayah Tahun 2012-2032*. Lumajang: Pemerintah Daerah Kabupaten Lumajang.

LAMPIRAN

LAMPIRAN A. DATASET PENELITIAN

No	Tipe Rel	Jenis Bantalan	Jenis Penambat	Tebal Balas Atas (cm)	Lebar Bahu Balas (cm)	Landai Penentu Maksimum	Jari-jari minimal vertikal (m)	Jari-jari minimal horizontal tanpa peralihan (m)	Jari-jari minimal horizontal peralihan (m)	Lebar Badan Jalan (m)	Kecepatan Maksimum (km/jam)
1	R.60	Beton	Ganda	30	60	0.01	0	0	0	3.15	120
2	R.60	Beton	Ganda	30	60	0.01	0	0	0	3	120
3	R.60	Beton	Ganda	30	60	0.01	8000	0	0	3.15	120
4	R.60	Beton	Ganda	30	60	0.01	8000	0	0	3	120
5	R.60	Beton	Ganda	30	60	0.01	0	2370	0	3.15	120
6	R.60	Beton	Ganda	30	60	0.01	0	2370	0	3	120
7	R.60	Beton	Ganda	30	60	0.01	0	0	780	3.15	120
8	R.60	Beton	Ganda	30	60	0.01	0	0	780	3	120
9	R.54	Beton	Ganda	30	60	0.01	0	0	0	3.15	120
10	R.54	Beton	Ganda	30	60	0.01	0	0	0	3	120
11	R.54	Beton	Ganda	30	60	0.01	8000	0	0	3.15	120

12	R.54	Beton	Ganda	30	60	0.01	8000	0	0	3	120
13	R.54	Beton	Ganda	30	60	0.01	0	2370	0	3.15	120
14	R.54	Beton	Ganda	30	60	0.01	0	2370	0	3	120
15	R.54	Beton	Ganda	30	60	0.01	0	0	780	3.15	120
16	R.54	Beton	Ganda	30	60	0.01	0	0	780	3	120
17	R.54	Beton	Ganda	30	50	0.01	0	0	0	3.15	110
18	R.54	Beton	Ganda	30	50	0.01	0	0	0	3	110
19	R.54	Beton	Ganda	30	50	0.01	8000	0	0	3.15	110
20	R.54	Beton	Ganda	30	50	0.01	8000	0	0	3	110
21	R.54	Beton	Ganda	30	50	0.01	0	1990	0	3.15	110
22	R.54	Beton	Ganda	30	50	0.01	0	1990	0	3	110
23	R.54	Beton	Ganda	30	50	0.01	0	0	660	3.15	110
24	R.54	Beton	Ganda	30	50	0.01	0	0	660	3	110
25	R.50	Beton	Ganda	30	50	0.01	0	0	0	3.15	110
26	R.50	Beton	Ganda	30	50	0.01	0	0	0	3	110
27	R.50	Beton	Ganda	30	50	0.01	8000	0	0	3.15	110
28	R.50	Beton	Ganda	30	50	0.01	8000	0	0	3	110
29	R.50	Beton	Ganda	30	50	0.01	0	1990	0	3.15	110

30	R.50	Beton	Ganda	30	50	0.01	0	1990	0	3	110
31	R.50	Beton	Ganda	30	50	0.01	0	0	660	3.15	110
32	R.50	Beton	Ganda	30	50	0.01	0	0	660	3	110
33	R.54	Kayu	Ganda	30	50	0.01	0	0	0	3.15	110
34	R.54	Kayu	Ganda	30	50	0.01	0	0	0	3	110
35	R.54	Kayu	Ganda	30	50	0.01	8000	0	0	3.15	110
36	R.54	Kayu	Ganda	30	50	0.01	8000	0	0	3	110
37	R.54	Kayu	Ganda	30	50	0.01	0	1990	0	3.15	110
38	R.54	Kayu	Ganda	30	50	0.01	0	1990	0	3	110
39	R.54	Kayu	Ganda	30	50	0.01	0	0	660	3.15	110
40	R.54	Kayu	Ganda	30	50	0.01	0	0	660	3	110
41	R.50	Kayu	Ganda	30	50	0.01	0	0	0	3.15	110
42	R.50	Kayu	Ganda	30	50	0.01	0	0	0	3	110
43	R.50	Kayu	Ganda	30	50	0.01	8000	0	0	3.15	110
44	R.50	Kayu	Ganda	30	50	0.01	8000	0	0	3	110
45	R.50	Kayu	Ganda	30	50	0.01	0	1990	0	3.15	110
46	R.50	Kayu	Ganda	30	50	0.01	0	1990	0	3	110
47	R.50	Kayu	Ganda	30	50	0.01	0	0	660	3.15	110

48	R.50	Kayu	Ganda	30	50	0.01	0	0	660	3	110
49	R.54	Beton	Ganda	30	40	0.02	0	0	0	2.95	100
50	R.54	Beton	Ganda	30	40	0.02	0	0	0	2.85	100
51	R.54	Beton	Ganda	30	40	0.02	6000	0	0	2.95	100
52	R.54	Beton	Ganda	30	40	0.02	6000	0	0	2.85	100
53	R.54	Beton	Ganda	30	40	0.02	0	1650	0	2.95	100
54	R.54	Beton	Ganda	30	40	0.02	0	1650	0	2.85	100
55	R.54	Beton	Ganda	30	40	0.02	0	0	550	2.95	100
56	R.54	Beton	Ganda	30	40	0.02	0	0	550	2.85	100
57	R.50	Beton	Ganda	30	40	0.02	0	0	0	2.95	100
58	R.50	Beton	Ganda	30	40	0.02	0	0	0	2.85	100
59	R.50	Beton	Ganda	30	40	0.02	6000	0	0	2.95	100
60	R.50	Beton	Ganda	30	40	0.02	6000	0	0	2.85	100
61	R.50	Beton	Ganda	30	40	0.02	0	1650	0	2.95	100
62	R.50	Beton	Ganda	30	40	0.02	0	1650	0	2.85	100
63	R.50	Beton	Ganda	30	40	0.02	0	0	550	2.95	100
64	R.50	Beton	Ganda	30	40	0.02	0	0	550	2.85	100
65	R.42	Beton	Ganda	30	40	0.02	0	0	0	2.95	100

66	R.42	Beton	Ganda	30	40	0.02	0	0	0	2.85	100
67	R.42	Beton	Ganda	30	40	0.02	6000	0	0	2.95	100
68	R.42	Beton	Ganda	30	40	0.02	6000	0	0	2.85	100
69	R.42	Beton	Ganda	30	40	0.02	0	1650	0	2.95	100
70	R.42	Beton	Ganda	30	40	0.02	0	1650	0	2.85	100
71	R.42	Beton	Ganda	30	40	0.02	0	0	550	2.95	100
72	R.42	Beton	Ganda	30	40	0.02	0	0	550	2.85	100
73	R.54	Kayu	Ganda	30	40	0.02	0	0	0	2.95	100
74	R.54	Kayu	Ganda	30	40	0.02	0	0	0	2.85	100
75	R.54	Kayu	Ganda	30	40	0.02	6000	0	0	2.95	100
76	R.54	Kayu	Ganda	30	40	0.02	6000	0	0	2.85	100
77	R.54	Kayu	Ganda	30	40	0.02	0	1650	0	2.95	100
78	R.54	Kayu	Ganda	30	40	0.02	0	1650	0	2.85	100
79	R.54	Kayu	Ganda	30	40	0.02	0	0	550	2.95	100
80	R.54	Kayu	Ganda	30	40	0.02	0	0	550	2.85	100
81	R.50	Kayu	Ganda	30	40	0.02	0	0	0	2.95	100
82	R.50	Kayu	Ganda	30	40	0.02	0	0	0	2.85	100
83	R.50	Kayu	Ganda	30	40	0.02	6000	0	0	2.95	100

84	R.50	Kayu	Ganda	30	40	0.02	6000	0	0	2.85	100
85	R.50	Kayu	Ganda	30	40	0.02	0	1650	0	2.95	100
86	R.50	Kayu	Ganda	30	40	0.02	0	1650	0	2.85	100
87	R.50	Kayu	Ganda	30	40	0.02	0	0	550	2.95	100
88	R.50	Kayu	Ganda	30	40	0.02	0	0	550	2.85	100
89	R.42	Kayu	Ganda	30	40	0.02	0	0	0	2.95	100
90	R.42	Kayu	Ganda	30	40	0.02	0	0	0	2.85	100
91	R.42	Kayu	Ganda	30	40	0.02	6000	0	0	2.95	100
92	R.42	Kayu	Ganda	30	40	0.02	6000	0	0	2.85	100
93	R.42	Kayu	Ganda	30	40	0.02	0	1650	0	2.95	100
94	R.42	Kayu	Ganda	30	40	0.02	0	1650	0	2.85	100
95	R.42	Kayu	Ganda	30	40	0.02	0	0	550	2.95	100
96	R.42	Kayu	Ganda	30	40	0.02	0	0	550	2.85	100
97	R.54	Baja	Ganda	30	40	0.02	0	0	0	2.95	100
98	R.54	Baja	Ganda	30	40	0.02	0	0	0	2.85	100
99	R.54	Baja	Ganda	30	40	0.02	6000	0	0	2.95	100
100	R.54	Baja	Ganda	30	40	0.02	6000	0	0	2.85	100
101	R.54	Baja	Ganda	30	40	0.02	0	1650	0	2.95	100

102	R.54	Baja	Ganda	30	40	0.02	0	1650	0	2.85	100
103	R.54	Baja	Ganda	30	40	0.02	0	0	550	2.95	100
104	R.54	Baja	Ganda	30	40	0.02	0	0	550	2.85	100
105	R.50	Baja	Ganda	30	40	0.02	0	0	0	2.95	100
106	R.50	Baja	Ganda	30	40	0.02	0	0	0	2.85	100
107	R.50	Baja	Ganda	30	40	0.02	6000	0	0	2.95	100
108	R.50	Baja	Ganda	30	40	0.02	6000	0	0	2.85	100
109	R.50	Baja	Ganda	30	40	0.02	0	1650	0	2.95	100
110	R.50	Baja	Ganda	30	40	0.02	0	1650	0	2.85	100
111	R.50	Baja	Ganda	30	40	0.02	0	0	550	2.95	100
112	R.50	Baja	Ganda	30	40	0.02	0	0	550	2.85	100
113	R.42	Baja	Ganda	30	40	0.02	0	0	0	2.95	100
114	R.42	Baja	Ganda	30	40	0.02	0	0	0	2.85	100
115	R.42	Baja	Ganda	30	40	0.02	6000	0	0	2.95	100
116	R.42	Baja	Ganda	30	40	0.02	6000	0	0	2.85	100
117	R.42	Baja	Ganda	30	40	0.02	0	1650	0	2.95	100
118	R.42	Baja	Ganda	30	40	0.02	0	1650	0	2.85	100
119	R.42	Baja	Ganda	30	40	0.02	0	0	550	2.95	100

120	R.42	Baja	Ganda	30	40	0.02	0	0	550	2.85	100
121	R.54	Beton	Ganda	25	40	0.025	0	0	0	2.85	90
122	R.54	Beton	Ganda	25	40	0.025	0	0	0	2.75	90
123	R.54	Beton	Ganda	25	40	0.025	6000	0	0	2.85	90
124	R.54	Beton	Ganda	25	40	0.025	6000	0	0	2.75	90
125	R.54	Beton	Ganda	25	40	0.025	0	1330	0	2.85	90
126	R.54	Beton	Ganda	25	40	0.025	0	1330	0	2.75	90
127	R.54	Beton	Ganda	25	40	0.025	0	0	440	2.85	90
128	R.54	Beton	Ganda	25	40	0.025	0	0	440	2.75	90
129	R.50	Beton	Ganda	25	40	0.025	0	0	0	2.85	90
130	R.50	Beton	Ganda	25	40	0.025	0	0	0	2.75	90
131	R.50	Beton	Ganda	25	40	0.025	6000	0	0	2.85	90
132	R.50	Beton	Ganda	25	40	0.025	6000	0	0	2.75	90
133	R.50	Beton	Ganda	25	40	0.025	0	1330	0	2.85	90
134	R.50	Beton	Ganda	25	40	0.025	0	1330	0	2.75	90
135	R.50	Beton	Ganda	25	40	0.025	0	0	440	2.85	90
136	R.50	Beton	Ganda	25	40	0.025	0	0	440	2.75	90
137	R.42	Beton	Ganda	25	40	0.025	0	0	0	2.85	90

138	R.42	Beton	Ganda	25	40	0.025	0	0	0	2.75	90
139	R.42	Beton	Ganda	25	40	0.025	6000	0	0	2.85	90
140	R.42	Beton	Ganda	25	40	0.025	6000	0	0	2.75	90
141	R.42	Beton	Ganda	25	40	0.025	0	1330	0	2.85	90
142	R.42	Beton	Ganda	25	40	0.025	0	1330	0	2.75	90
143	R.42	Beton	Ganda	25	40	0.025	0	0	440	2.85	90
144	R.42	Beton	Ganda	25	40	0.025	0	0	440	2.75	90
145	R.54	Kayu	Ganda	25	40	0.025	0	0	0	2.85	90
146	R.54	Kayu	Ganda	25	40	0.025	0	0	0	2.75	90
147	R.54	Kayu	Ganda	25	40	0.025	6000	0	0	2.85	90
148	R.54	Kayu	Ganda	25	40	0.025	6000	0	0	2.75	90
149	R.54	Kayu	Ganda	25	40	0.025	0	1330	0	2.85	90
150	R.54	Kayu	Ganda	25	40	0.025	0	1330	0	2.75	90
151	R.54	Kayu	Ganda	25	40	0.025	0	0	440	2.85	90
152	R.54	Kayu	Ganda	25	40	0.025	0	0	440	2.75	90
153	R.50	Kayu	Ganda	25	40	0.025	0	0	0	2.85	90
154	R.50	Kayu	Ganda	25	40	0.025	0	0	0	2.75	90
155	R.50	Kayu	Ganda	25	40	0.025	6000	0	0	2.85	90

156	R.50	Kayu	Ganda	25	40	0.025	6000	0	0	2.75	90
157	R.50	Kayu	Ganda	25	40	0.025	0	1330	0	2.85	90
158	R.50	Kayu	Ganda	25	40	0.025	0	1330	0	2.75	90
159	R.50	Kayu	Ganda	25	40	0.025	0	0	440	2.85	90
160	R.50	Kayu	Ganda	25	40	0.025	0	0	440	2.75	90
161	R.42	Kayu	Ganda	25	40	0.025	0	0	0	2.85	90
162	R.42	Kayu	Ganda	25	40	0.025	0	0	0	2.75	90
163	R.42	Kayu	Ganda	25	40	0.025	6000	0	0	2.85	90
164	R.42	Kayu	Ganda	25	40	0.025	6000	0	0	2.75	90
165	R.42	Kayu	Ganda	25	40	0.025	0	1330	0	2.85	90
166	R.42	Kayu	Ganda	25	40	0.025	0	1330	0	2.75	90
167	R.42	Kayu	Ganda	25	40	0.025	0	0	440	2.85	90
168	R.42	Kayu	Ganda	25	40	0.025	0	0	440	2.75	90
169	R.54	Baja	Ganda	25	40	0.025	0	0	0	2.85	90
170	R.54	Baja	Ganda	25	40	0.025	0	0	0	2.75	90
171	R.54	Baja	Ganda	25	40	0.025	6000	0	0	2.85	90
172	R.54	Baja	Ganda	25	40	0.025	6000	0	0	2.75	90
173	R.54	Baja	Ganda	25	40	0.025	0	1330	0	2.85	90

174	R.54	Baja	Ganda	25	40	0.025	0	1330	0	2.75	90
175	R.54	Baja	Ganda	25	40	0.025	0	0	440	2.85	90
176	R.54	Baja	Ganda	25	40	0.025	0	0	440	2.75	90
177	R.50	Baja	Ganda	25	40	0.025	0	0	0	2.85	90
178	R.50	Baja	Ganda	25	40	0.025	0	0	0	2.75	90
179	R.50	Baja	Ganda	25	40	0.025	6000	0	0	2.85	90
180	R.50	Baja	Ganda	25	40	0.025	6000	0	0	2.75	90
181	R.50	Baja	Ganda	25	40	0.025	0	1330	0	2.85	90
182	R.50	Baja	Ganda	25	40	0.025	0	1330	0	2.75	90
183	R.50	Baja	Ganda	25	40	0.025	0	0	440	2.85	90
184	R.50	Baja	Ganda	25	40	0.025	0	0	440	2.75	90
185	R.42	Baja	Ganda	25	40	0.025	0	0	0	2.85	90
186	R.42	Baja	Ganda	25	40	0.025	0	0	0	2.75	90
187	R.42	Baja	Ganda	25	40	0.025	6000	0	0	2.85	90
188	R.42	Baja	Ganda	25	40	0.025	6000	0	0	2.75	90
189	R.42	Baja	Ganda	25	40	0.025	0	1330	0	2.85	90
190	R.42	Baja	Ganda	25	40	0.025	0	1330	0	2.75	90
191	R.42	Baja	Ganda	25	40	0.025	0	0	440	2.85	90

192	R.42	Baja	Ganda	25	40	0.025	0	0	440	2.75	90
193	R.54	Beton	Tunggal	25	40	0.025	0	0	0	2.85	90
194	R.54	Beton	Tunggal	25	40	0.025	0	0	0	2.75	90
195	R.54	Beton	Tunggal	25	40	0.025	6000	0	0	2.85	90
196	R.54	Beton	Tunggal	25	40	0.025	6000	0	0	2.75	90
197	R.54	Beton	Tunggal	25	40	0.025	0	1330	0	2.85	90
198	R.54	Beton	Tunggal	25	40	0.025	0	1330	0	2.75	90
199	R.54	Beton	Tunggal	25	40	0.025	0	0	440	2.85	90
200	R.54	Beton	Tunggal	25	40	0.025	0	0	440	2.75	90
201	R.50	Beton	Tunggal	25	40	0.025	0	0	0	2.85	90
202	R.50	Beton	Tunggal	25	40	0.025	0	0	0	2.75	90
203	R.50	Beton	Tunggal	25	40	0.025	6000	0	0	2.85	90
204	R.50	Beton	Tunggal	25	40	0.025	6000	0	0	2.75	90
205	R.50	Beton	Tunggal	25	40	0.025	0	1330	0	2.85	90
206	R.50	Beton	Tunggal	25	40	0.025	0	1330	0	2.75	90
207	R.50	Beton	Tunggal	25	40	0.025	0	0	440	2.85	90
208	R.50	Beton	Tunggal	25	40	0.025	0	0	440	2.75	90
209	R.42	Beton	Tunggal	25	40	0.025	0	0	0	2.85	90

210	R.42	Beton	Tunggal	25	40	0.025	0	0	0	2.75	90
211	R.42	Beton	Tunggal	25	40	0.025	6000	0	0	2.85	90
212	R.42	Beton	Tunggal	25	40	0.025	6000	0	0	2.75	90
213	R.42	Beton	Tunggal	25	40	0.025	0	1330	0	2.85	90
214	R.42	Beton	Tunggal	25	40	0.025	0	1330	0	2.75	90
215	R.42	Beton	Tunggal	25	40	0.025	0	0	440	2.85	90
216	R.42	Beton	Tunggal	25	40	0.025	0	0	440	2.75	90
217	R.54	Kayu	Tunggal	25	40	0.025	0	0	0	2.85	90
218	R.54	Kayu	Tunggal	25	40	0.025	0	0	0	2.75	90
219	R.54	Kayu	Tunggal	25	40	0.025	6000	0	0	2.85	90
220	R.54	Kayu	Tunggal	25	40	0.025	6000	0	0	2.75	90
221	R.54	Kayu	Tunggal	25	40	0.025	0	1330	0	2.85	90
222	R.54	Kayu	Tunggal	25	40	0.025	0	1330	0	2.75	90
223	R.54	Kayu	Tunggal	25	40	0.025	0	0	440	2.85	90
224	R.54	Kayu	Tunggal	25	40	0.025	0	0	440	2.75	90
225	R.50	Kayu	Tunggal	25	40	0.025	0	0	0	2.85	90
226	R.50	Kayu	Tunggal	25	40	0.025	0	0	0	2.75	90
227	R.50	Kayu	Tunggal	25	40	0.025	6000	0	0	2.85	90

228	R.50	Kayu	Tunggal	25	40	0.025	6000	0	0	2.75	90
229	R.50	Kayu	Tunggal	25	40	0.025	0	1330	0	2.85	90
230	R.50	Kayu	Tunggal	25	40	0.025	0	1330	0	2.75	90
231	R.50	Kayu	Tunggal	25	40	0.025	0	0	440	2.85	90
232	R.50	Kayu	Tunggal	25	40	0.025	0	0	440	2.75	90
233	R.42	Kayu	Tunggal	25	40	0.025	0	0	0	2.85	90
234	R.42	Kayu	Tunggal	25	40	0.025	0	0	0	2.75	90
235	R.42	Kayu	Tunggal	25	40	0.025	6000	0	0	2.85	90
236	R.42	Kayu	Tunggal	25	40	0.025	6000	0	0	2.75	90
237	R.42	Kayu	Tunggal	25	40	0.025	0	1330	0	2.85	90
238	R.42	Kayu	Tunggal	25	40	0.025	0	1330	0	2.75	90
239	R.42	Kayu	Tunggal	25	40	0.025	0	0	440	2.85	90
240	R.42	Kayu	Tunggal	25	40	0.025	0	0	440	2.75	90
241	R.54	Baja	Tunggal	25	40	0.025	0	0	0	2.85	90
242	R.54	Baja	Tunggal	25	40	0.025	0	0	0	2.75	90
243	R.54	Baja	Tunggal	25	40	0.025	6000	0	0	2.85	90
244	R.54	Baja	Tunggal	25	40	0.025	6000	0	0	2.75	90
245	R.54	Baja	Tunggal	25	40	0.025	0	1330	0	2.85	90

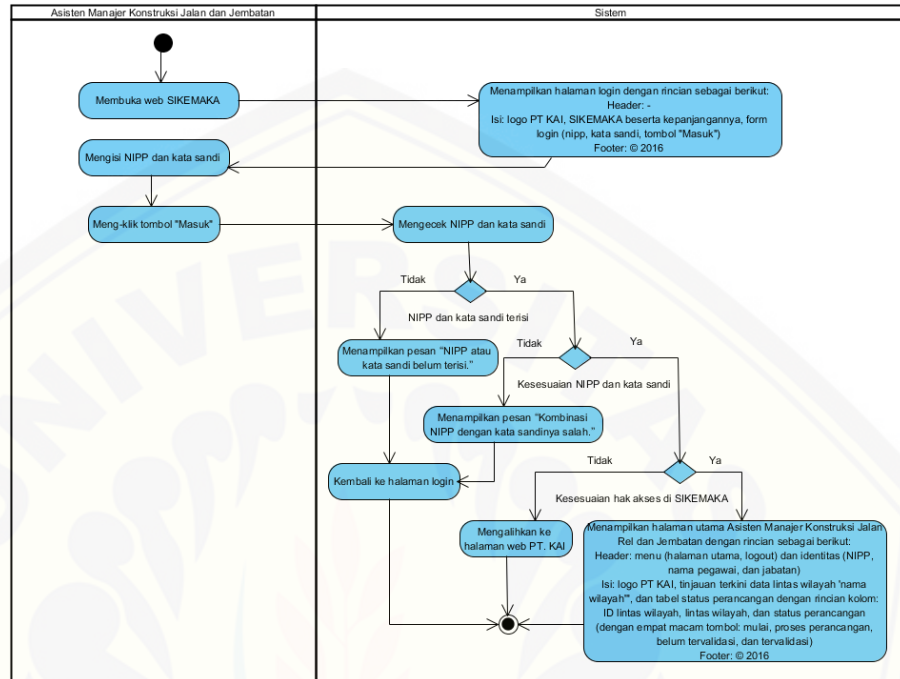
246	R.54	Baja	Tunggal	25	40	0.025	0	1330	0	2.75	90
247	R.54	Baja	Tunggal	25	40	0.025	0	0	440	2.85	90
248	R.54	Baja	Tunggal	25	40	0.025	0	0	440	2.75	90
249	R.50	Baja	Tunggal	25	40	0.025	0	0	0	2.85	90
250	R.50	Baja	Tunggal	25	40	0.025	0	0	0	2.75	90
251	R.50	Baja	Tunggal	25	40	0.025	6000	0	0	2.85	90
252	R.50	Baja	Tunggal	25	40	0.025	6000	0	0	2.75	90
253	R.50	Baja	Tunggal	25	40	0.025	0	1330	0	2.85	90
254	R.50	Baja	Tunggal	25	40	0.025	0	1330	0	2.75	90
255	R.50	Baja	Tunggal	25	40	0.025	0	0	440	2.85	90
256	R.50	Baja	Tunggal	25	40	0.025	0	0	440	2.75	90
257	R.42	Baja	Tunggal	25	40	0.025	0	0	0	2.85	90
258	R.42	Baja	Tunggal	25	40	0.025	0	0	0	2.75	90
259	R.42	Baja	Tunggal	25	40	0.025	6000	0	0	2.85	90
260	R.42	Baja	Tunggal	25	40	0.025	6000	0	0	2.75	90
261	R.42	Baja	Tunggal	25	40	0.025	0	1330	0	2.85	90
262	R.42	Baja	Tunggal	25	40	0.025	0	1330	0	2.75	90
263	R.42	Baja	Tunggal	25	40	0.025	0	0	440	2.85	90

264	R.42	Baja	Tunggal	25	40	0.025	0	0	440	2.75	90
265	R.42	Kayu	Tunggal	25	35	0.025	0	0	0	2.5	80
266	R.42	Kayu	Tunggal	25	35	0.025	0	0	0	2.4	80
267	R.42	Kayu	Tunggal	25	35	0.025	6000	0	0	2.5	80
268	R.42	Kayu	Tunggal	25	35	0.025	6000	0	0	2.4	80
269	R.42	Kayu	Tunggal	25	35	0.025	0	1050	0	2.5	80
270	R.42	Kayu	Tunggal	25	35	0.025	0	1050	0	2.4	80
271	R.42	Kayu	Tunggal	25	35	0.025	0	0	350	2.5	80
272	R.42	Kayu	Tunggal	25	35	0.025	0	0	350	2.4	80
273	R.42	Baja	Tunggal	25	35	0.025	0	0	0	2.5	80
274	R.42	Baja	Tunggal	25	35	0.025	0	0	0	2.4	80
275	R.42	Baja	Tunggal	25	35	0.025	6000	0	0	2.5	80
276	R.42	Baja	Tunggal	25	35	0.025	6000	0	0	2.4	80
277	R.42	Baja	Tunggal	25	35	0.025	0	1050	0	2.5	80
278	R.42	Baja	Tunggal	25	35	0.025	0	1050	0	2.4	80
279	R.42	Baja	Tunggal	25	35	0.025	0	0	350	2.5	80
280	R.42	Baja	Tunggal	25	35	0.025	0	0	350	2.4	80

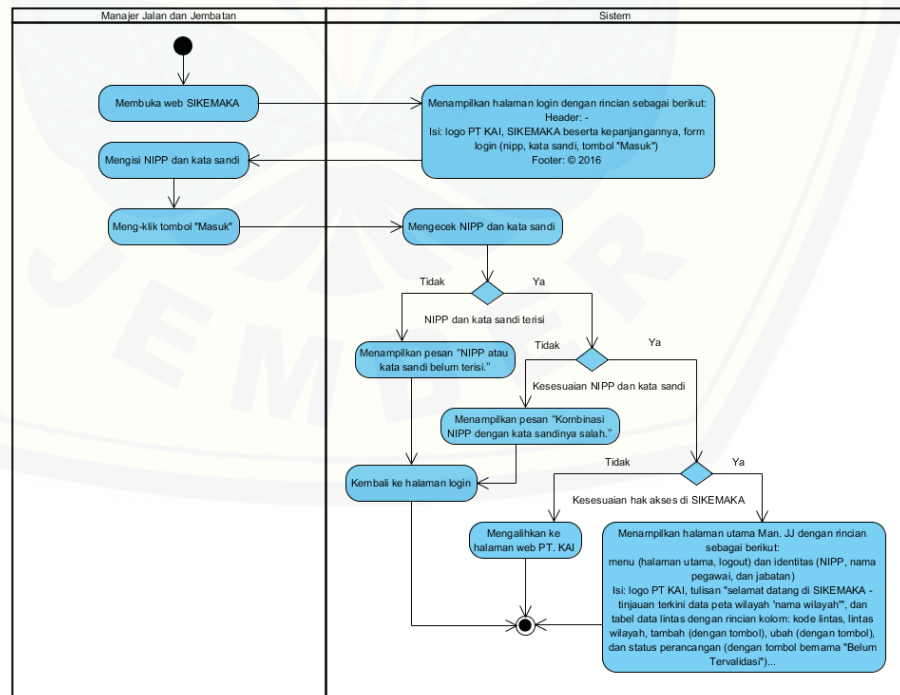
LAMPIRAN B. PERANCANGAN SISTEM

B.1. Perancangan Activity Diagram

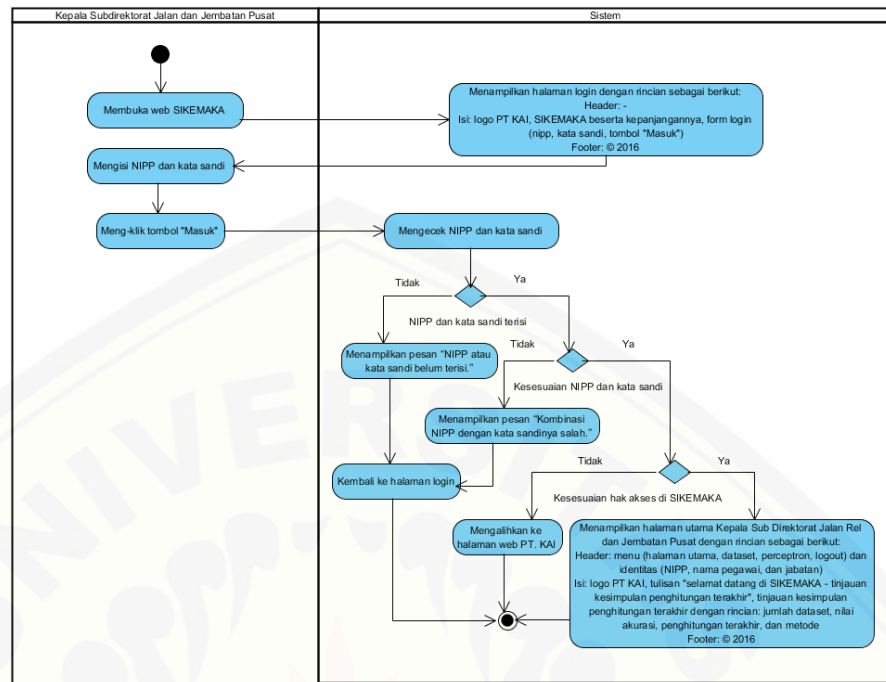
1. Activity Diagram Login Sistem untuk Asisten Manajer Konstruksi Jalan Rel dan Jembatan



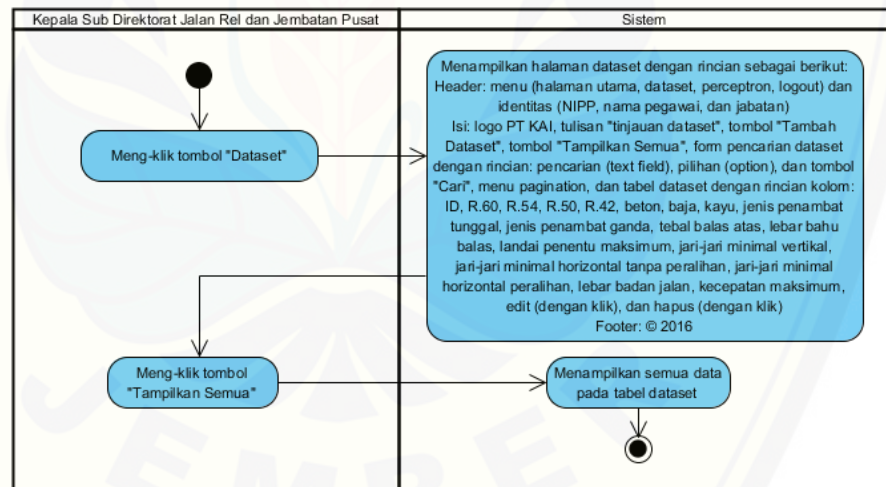
2. Activity Diagram Login Sistem untuk Manajer Jalan dan Jembatan



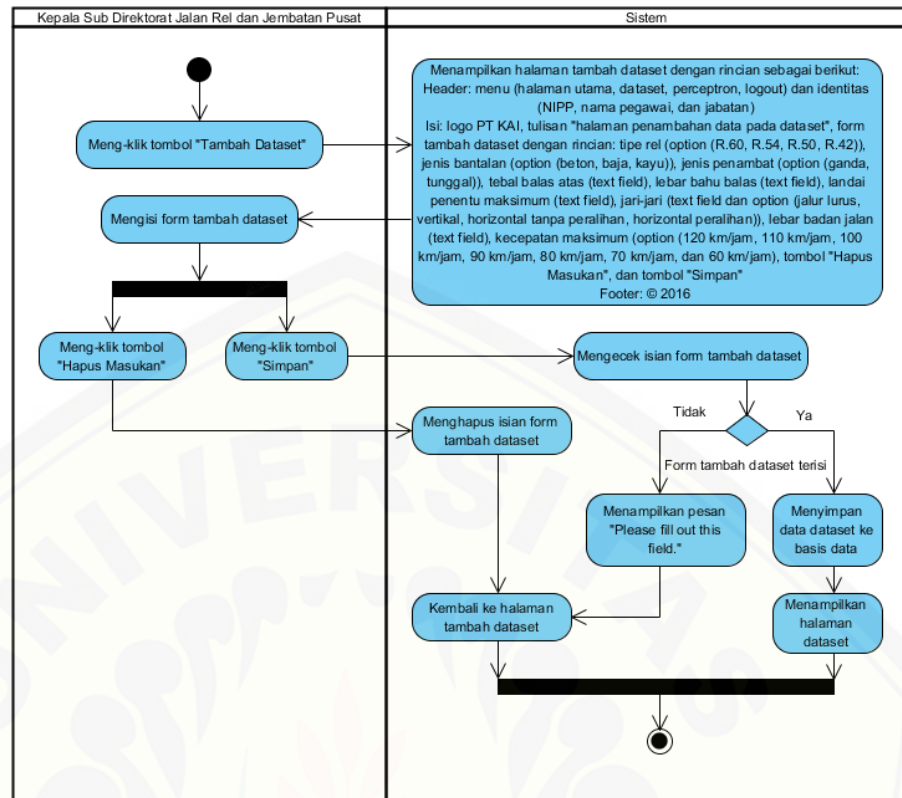
3. Activity Diagram Login Sistem untuk Kepala Sub Direktorat Jalan Rel dan Jembatan Pusat



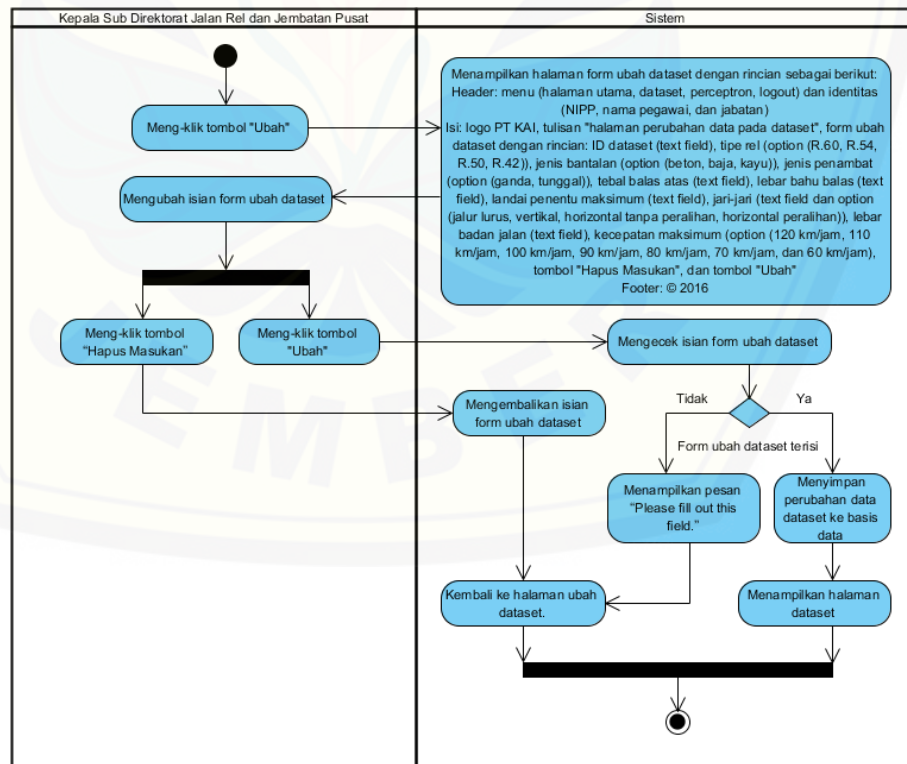
4. Activity Diagram Menampilkan Tabel Dataset



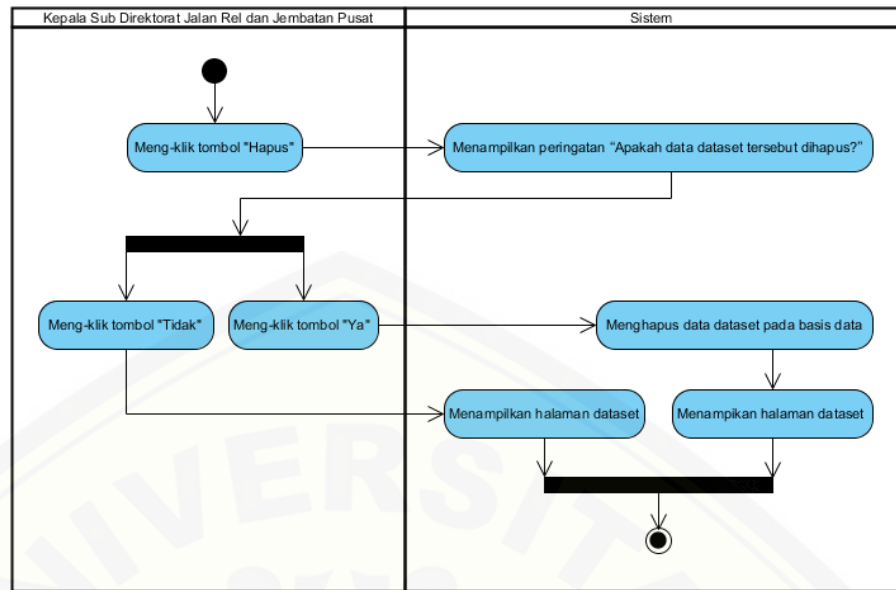
5. Activity Diagram Menambah Data pada Dataset



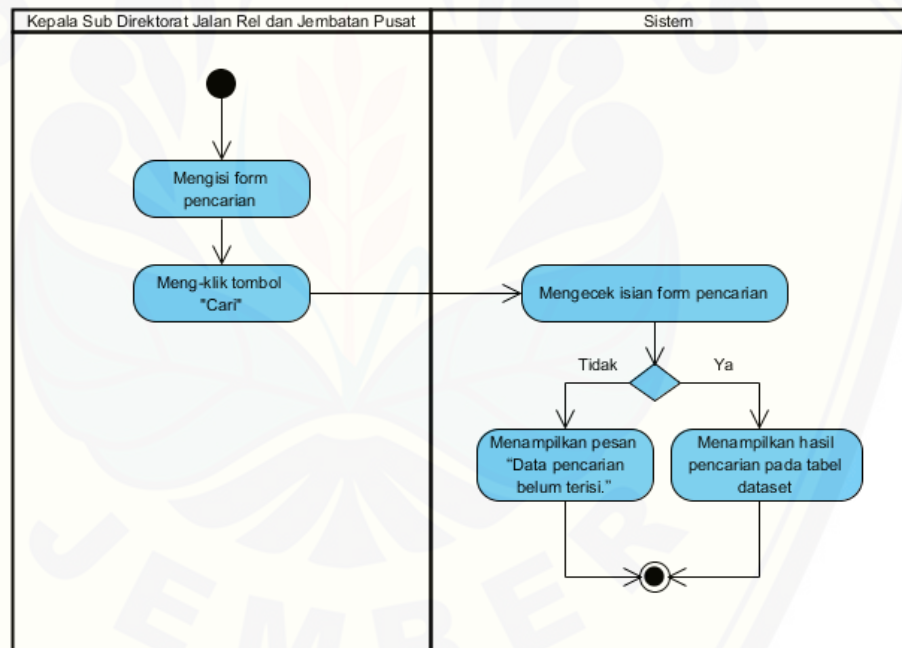
6. Activity Diagram Mengubah Data pada Dataset



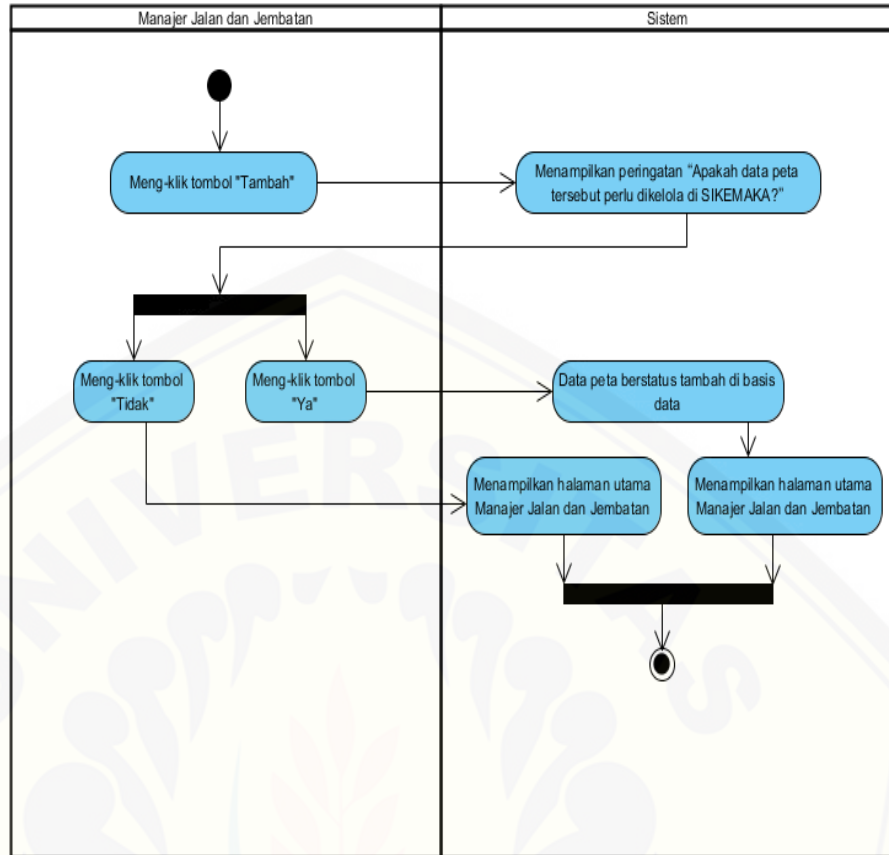
7. Activity Diagram Menghapus Data pada Dataset



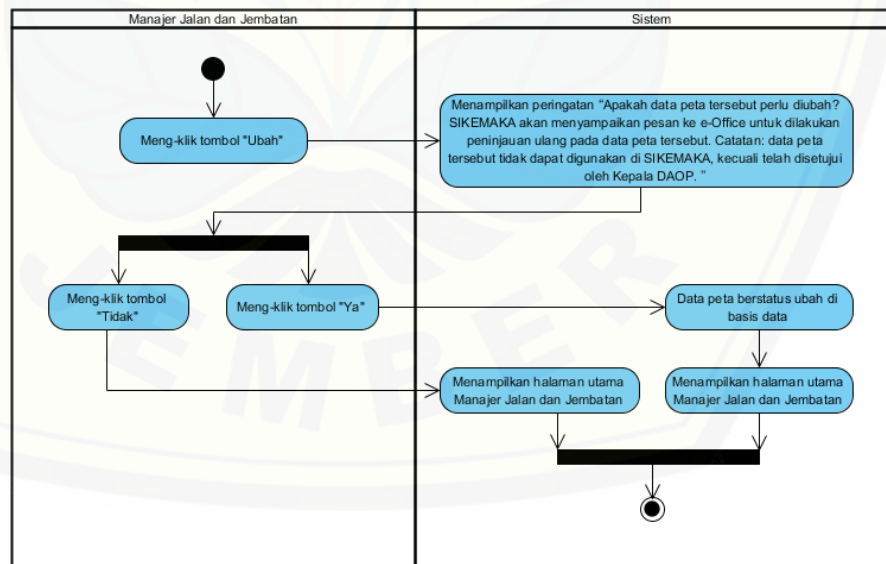
8. Activity Diagram Pencarian Data pada Dataset



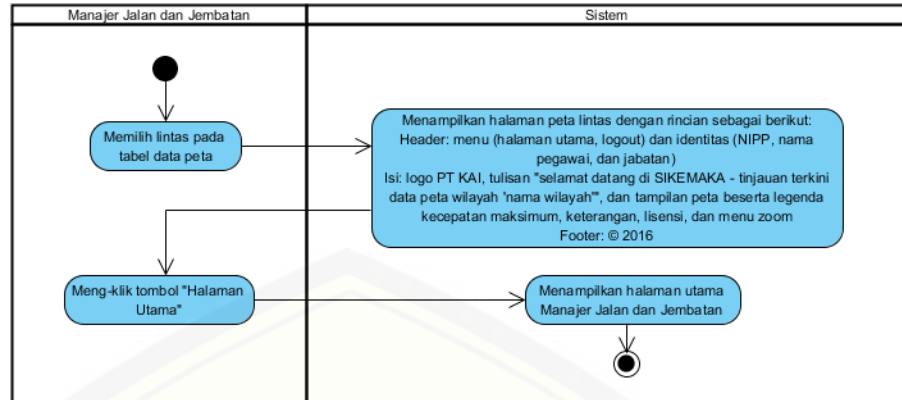
9. Activity Diagram Menambah Data Peta ke Tabel Status Perancangan



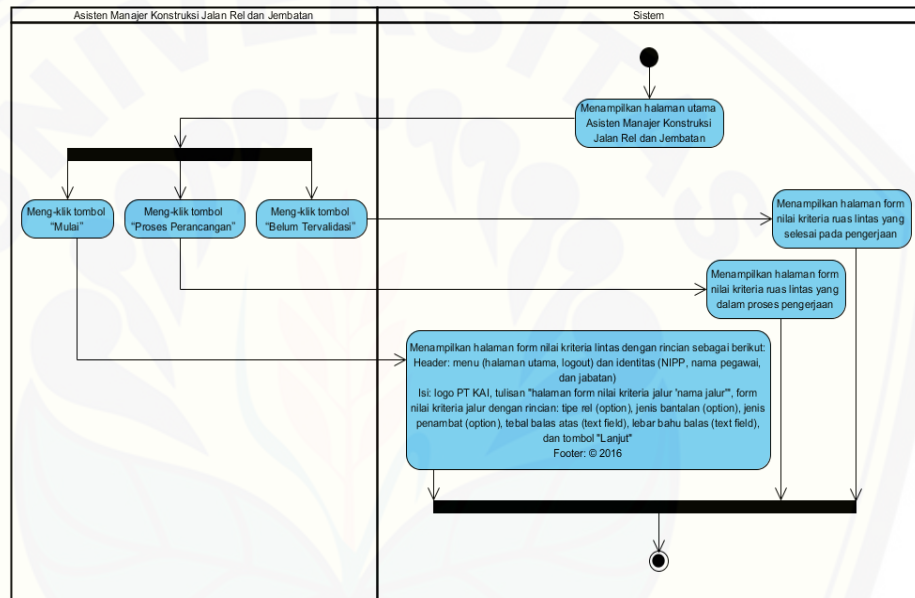
10. Activity Diagram Menandai Data Peta untuk Diubah di Sistem Lain



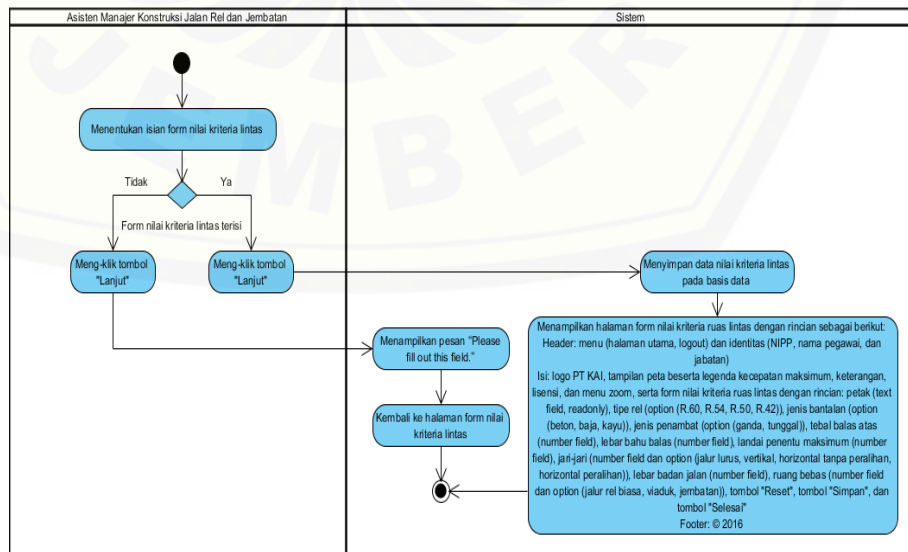
11. Activity Diagram Menampilkan Peta dari Data Peta yang Dipilih



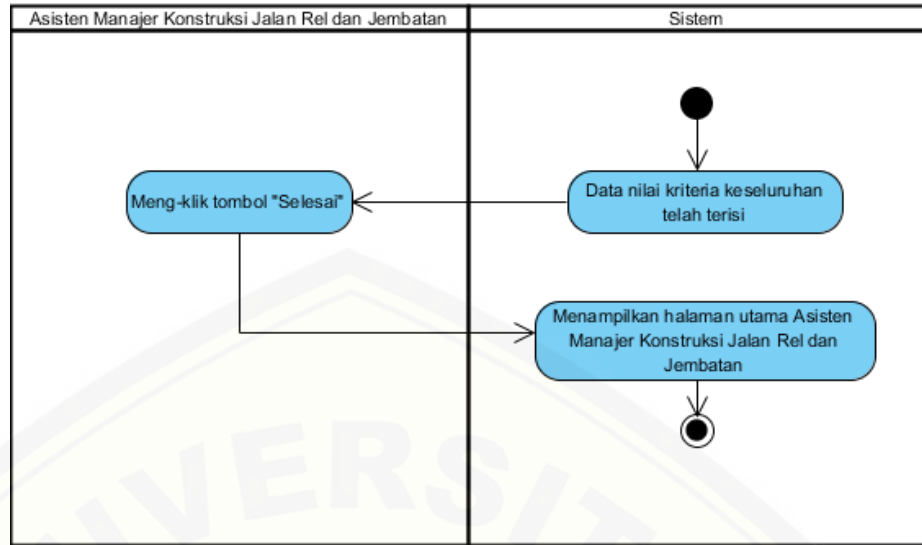
12. Activity Diagram Meninjau Tabel Status Perancangan



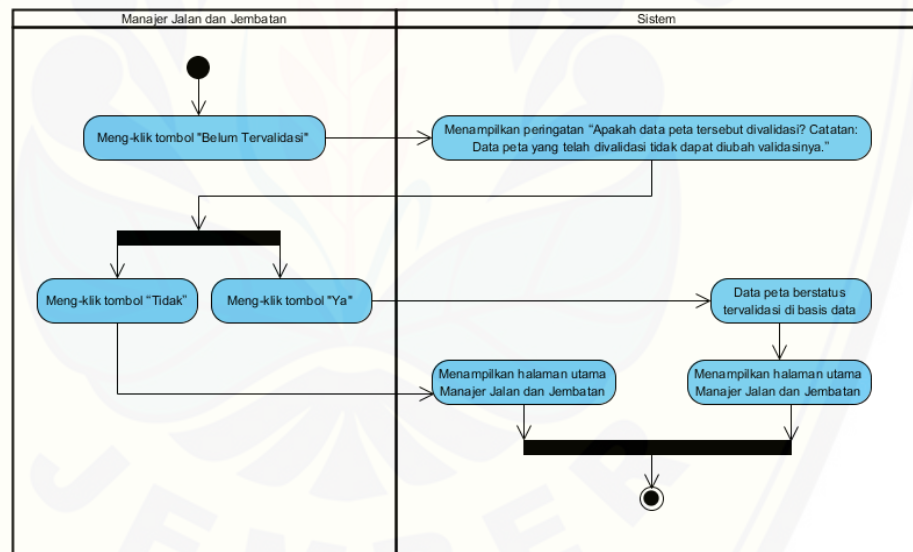
13. Activity Diagram Mengisi Nilai Kriteria Lintas



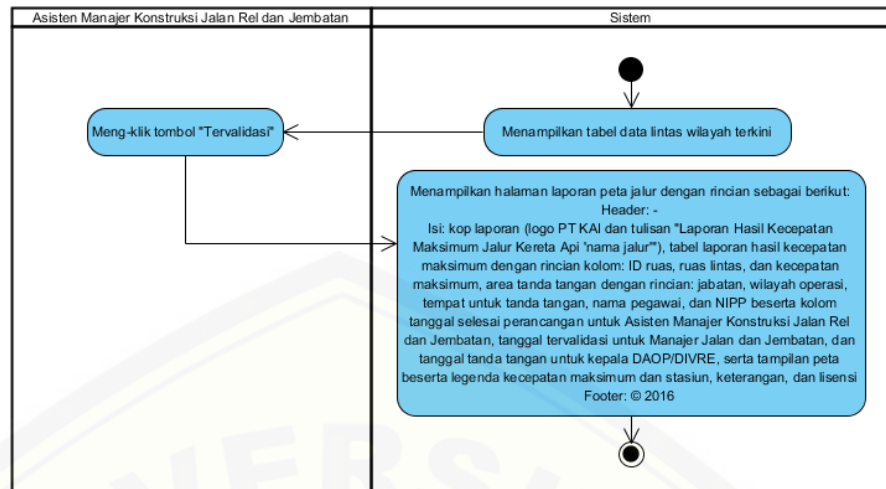
14. Activity Diagram Menyelesaikan Klasifikasi



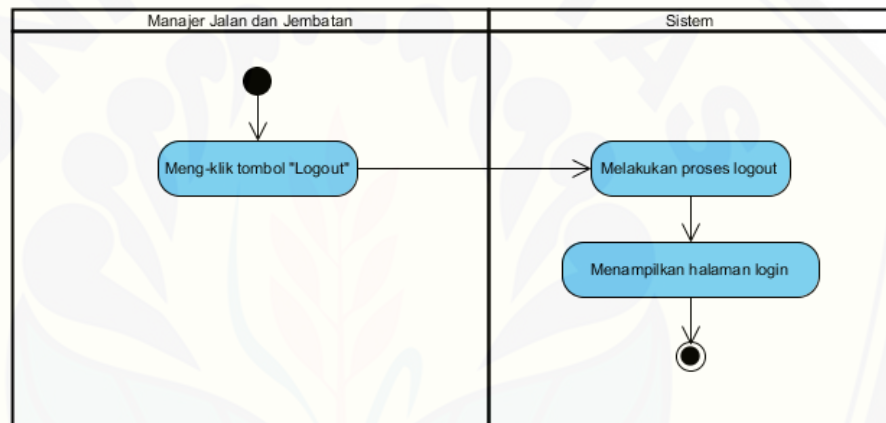
15. Activity Diagram Melakukan Validasi Perancangan Kecepatan Maksimum pada Data Peta



16. Activity Diagram Mencetak Laporan Desain Kecepatan Maksimum

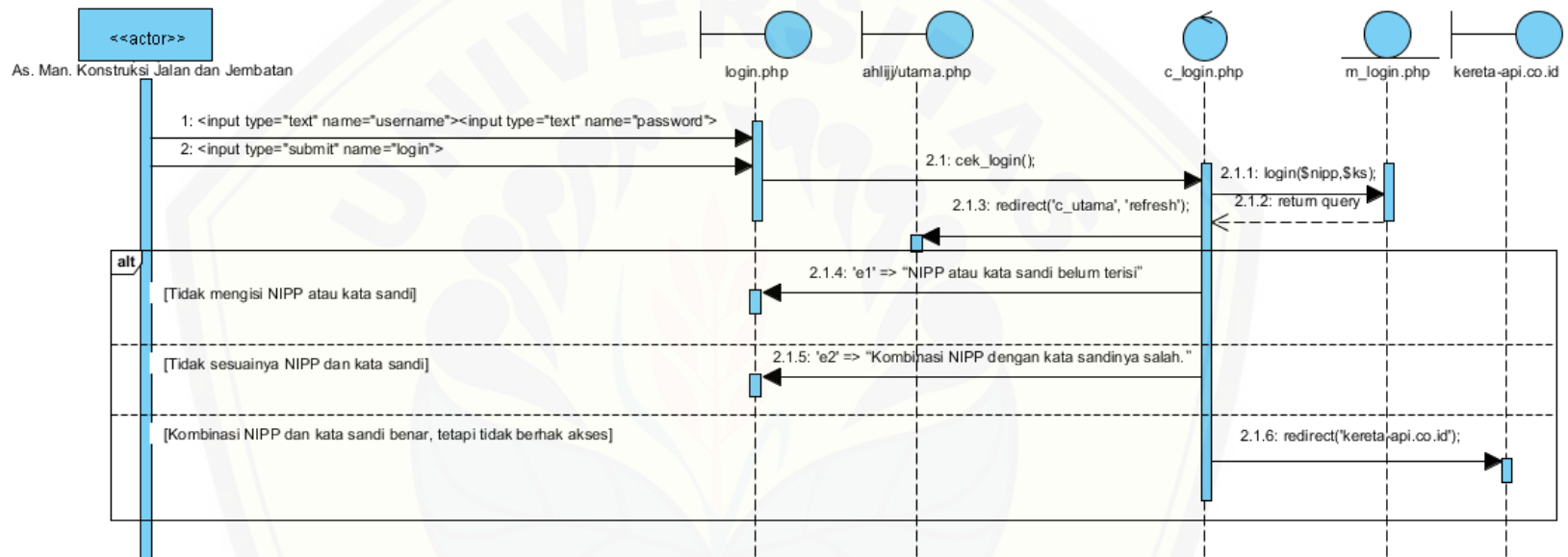


17. Activity Diagram Logout Sistem

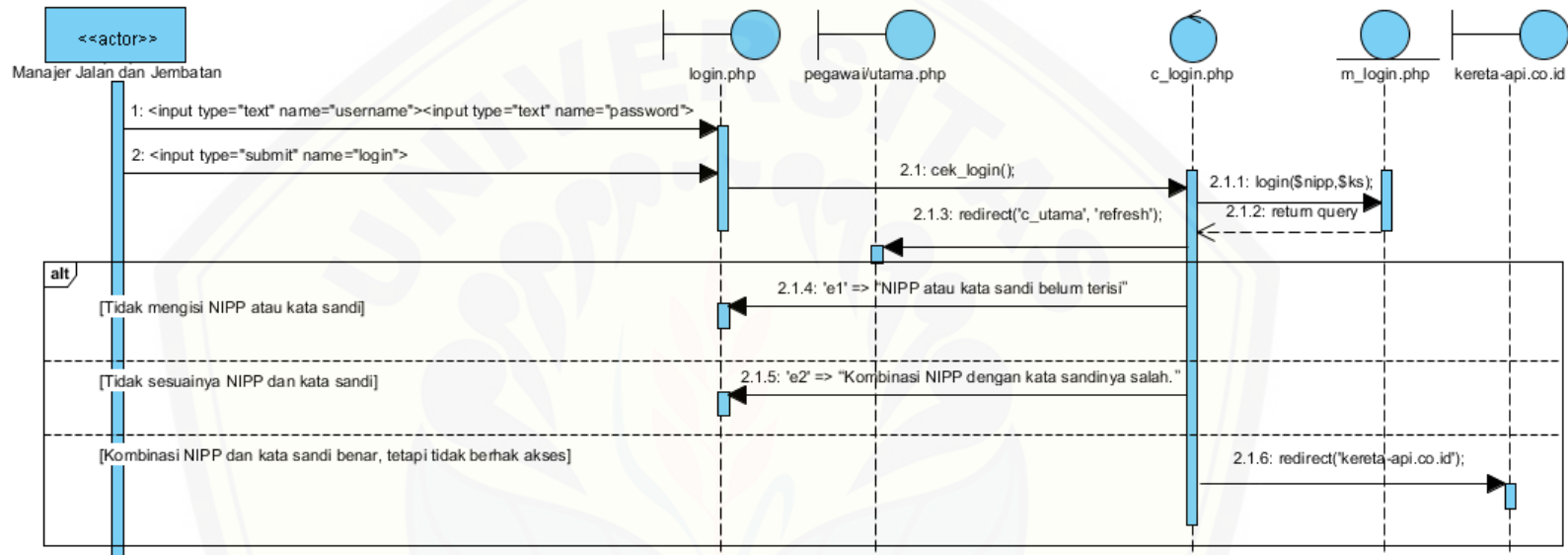


B.2. Perancangan *Sequence Diagram*

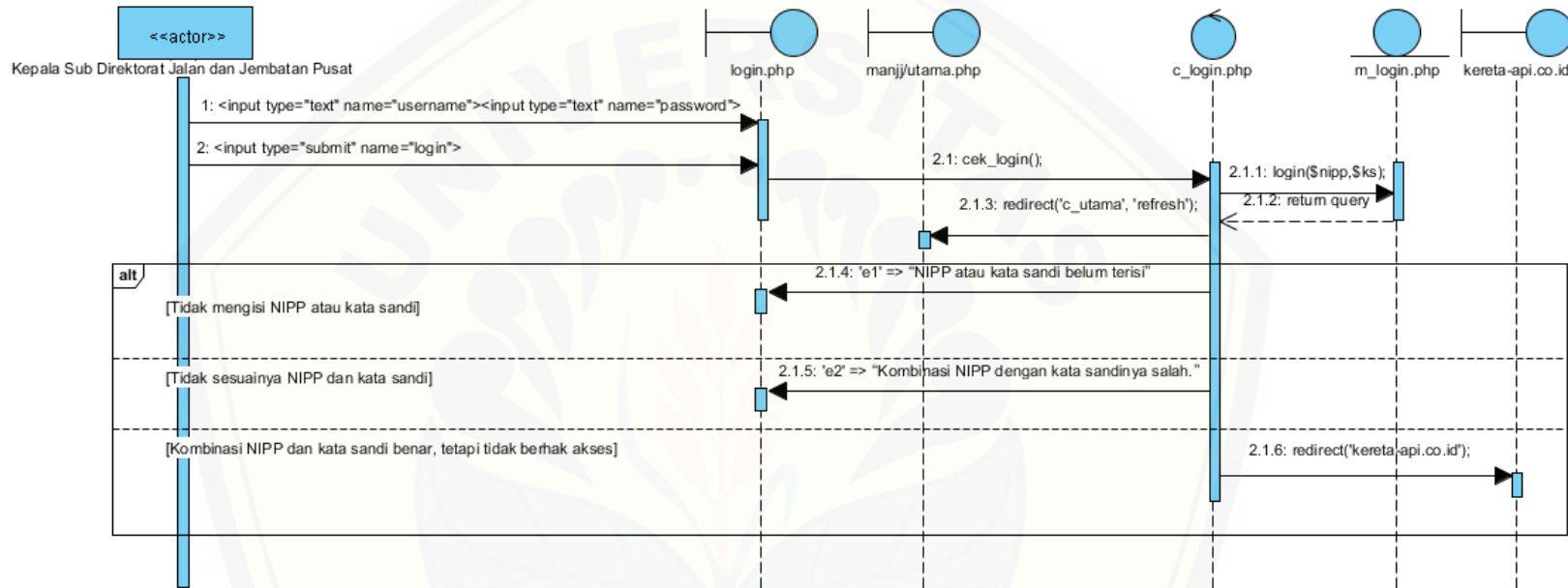
1. *Sequence Diagram Login* Sistem untuk Asisten Manajer Konstruksi Jalan Rel dan Jembatan



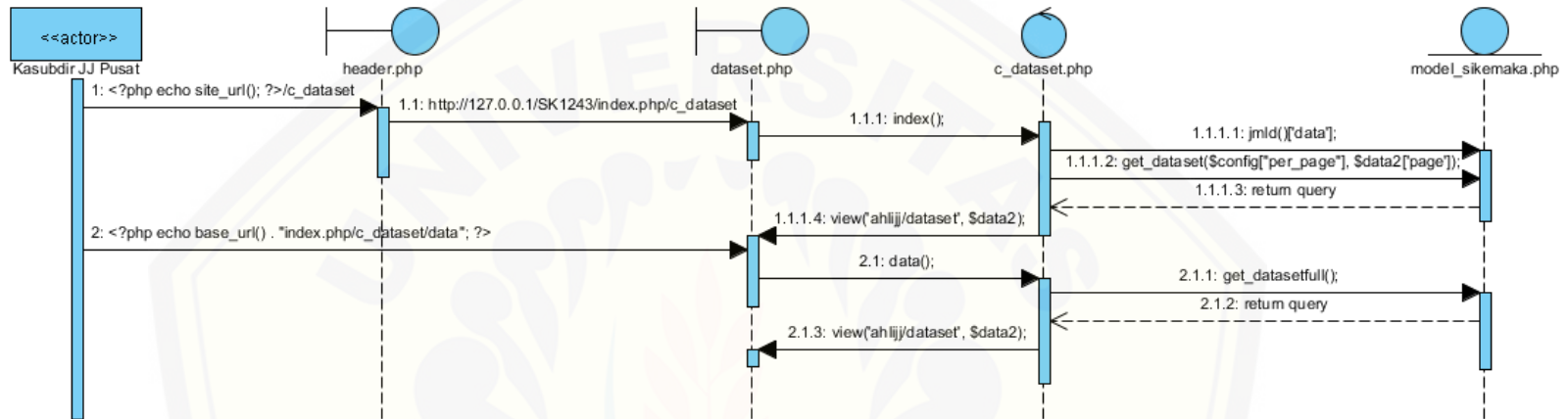
2. Sequence Diagram Login Sistem untuk Manajer Jalan dan Jembatan



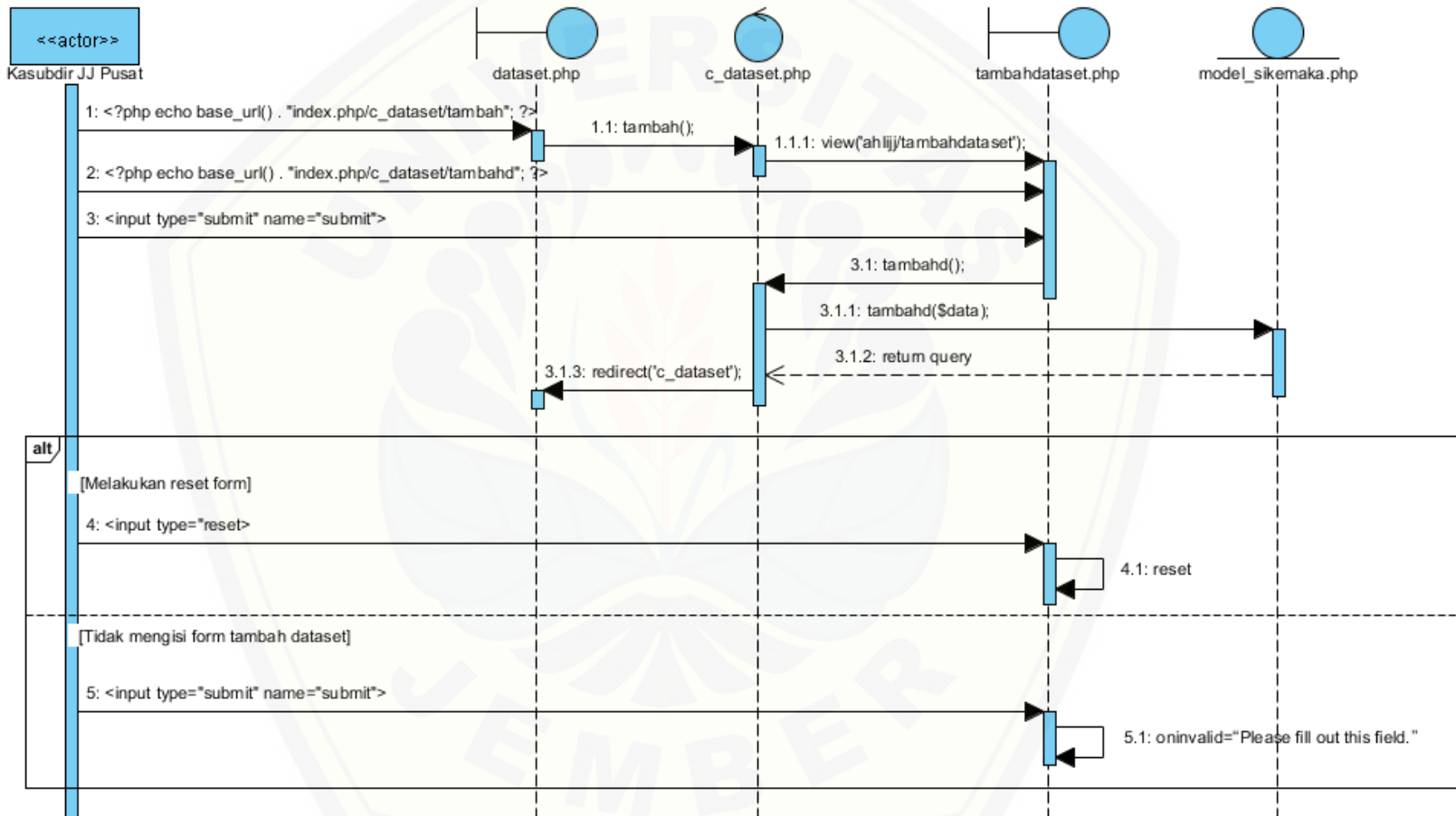
3. Sequence Diagram Login Sistem untuk Kepala Sub Direktorat Jalan Rel dan Jembatan Pusat



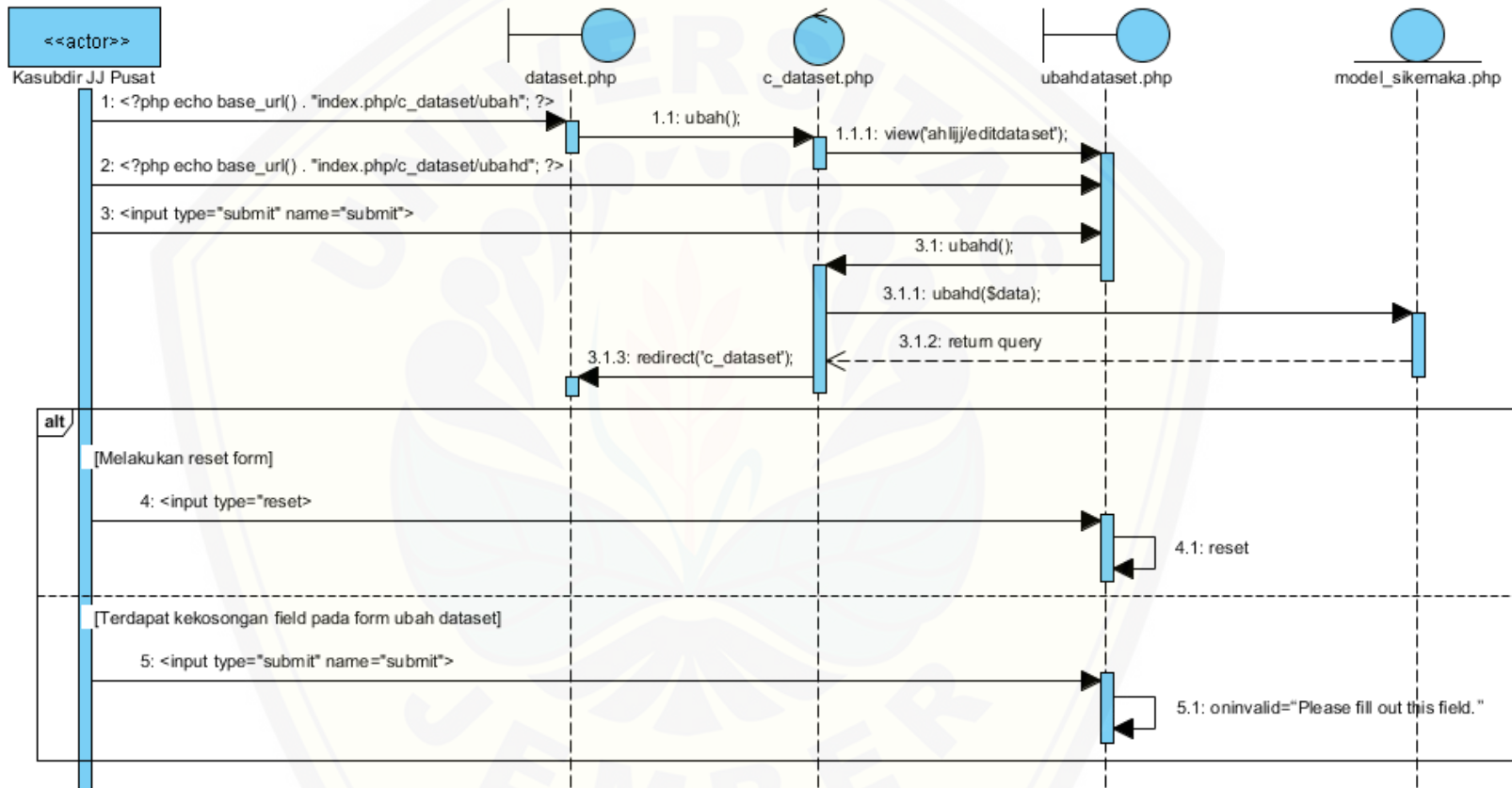
4. Sequence Diagram Menampilkan Tabel Dataset



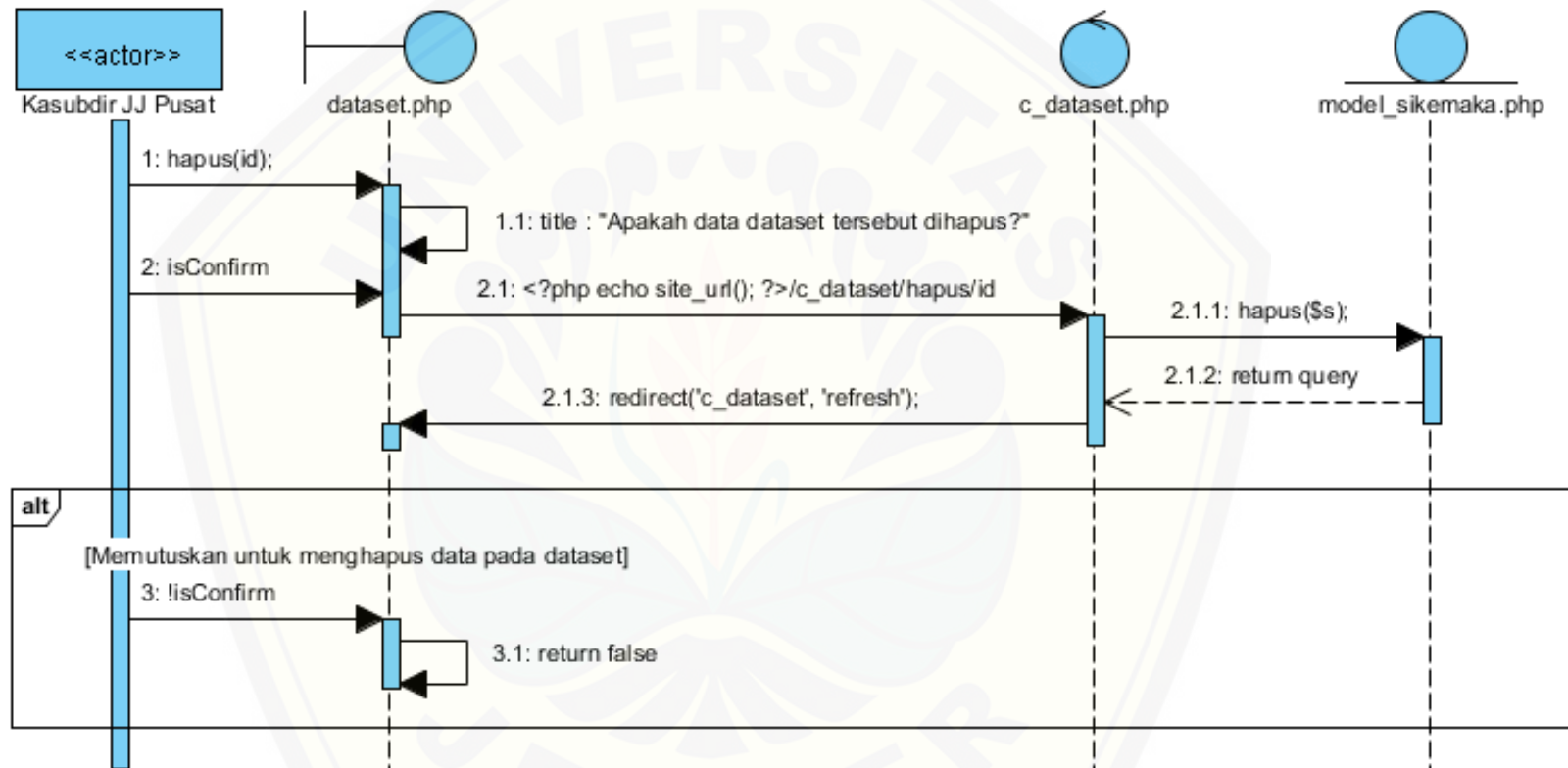
5. Sequence Diagram Menambah Data pada Dataset



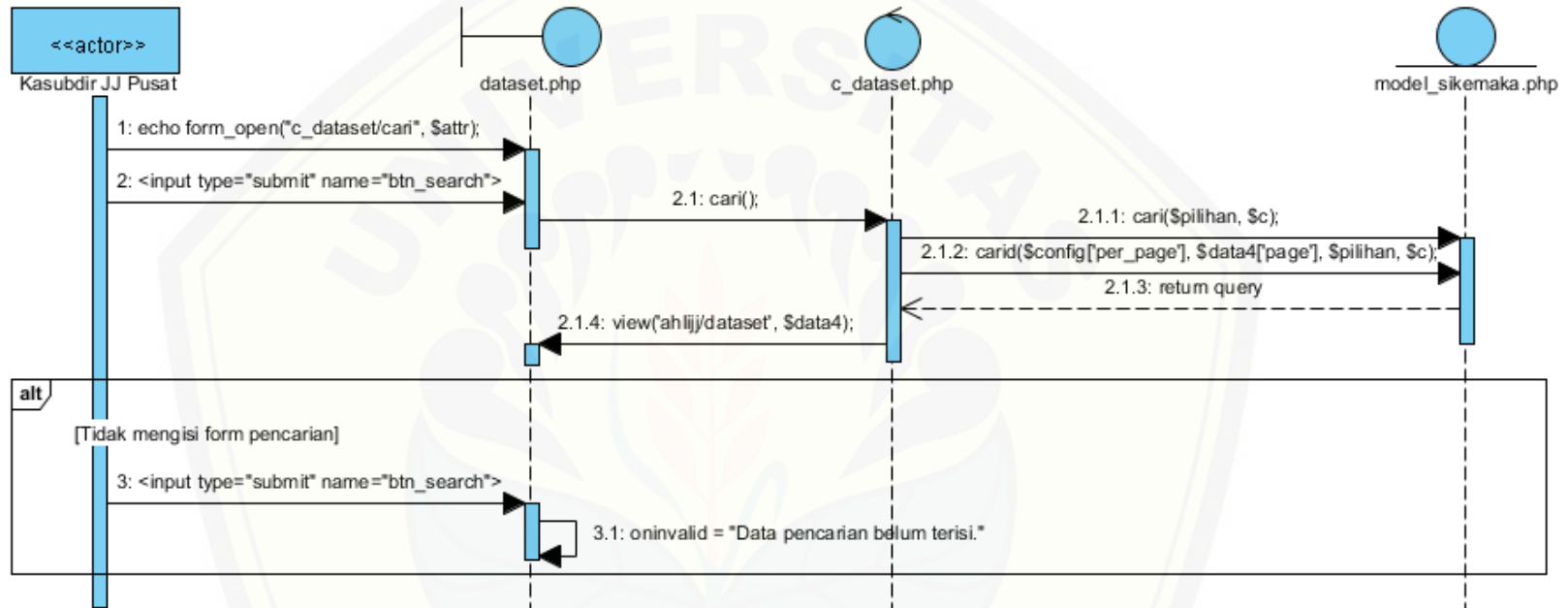
6. Sequence Diagram Mengubah Data pada Dataset



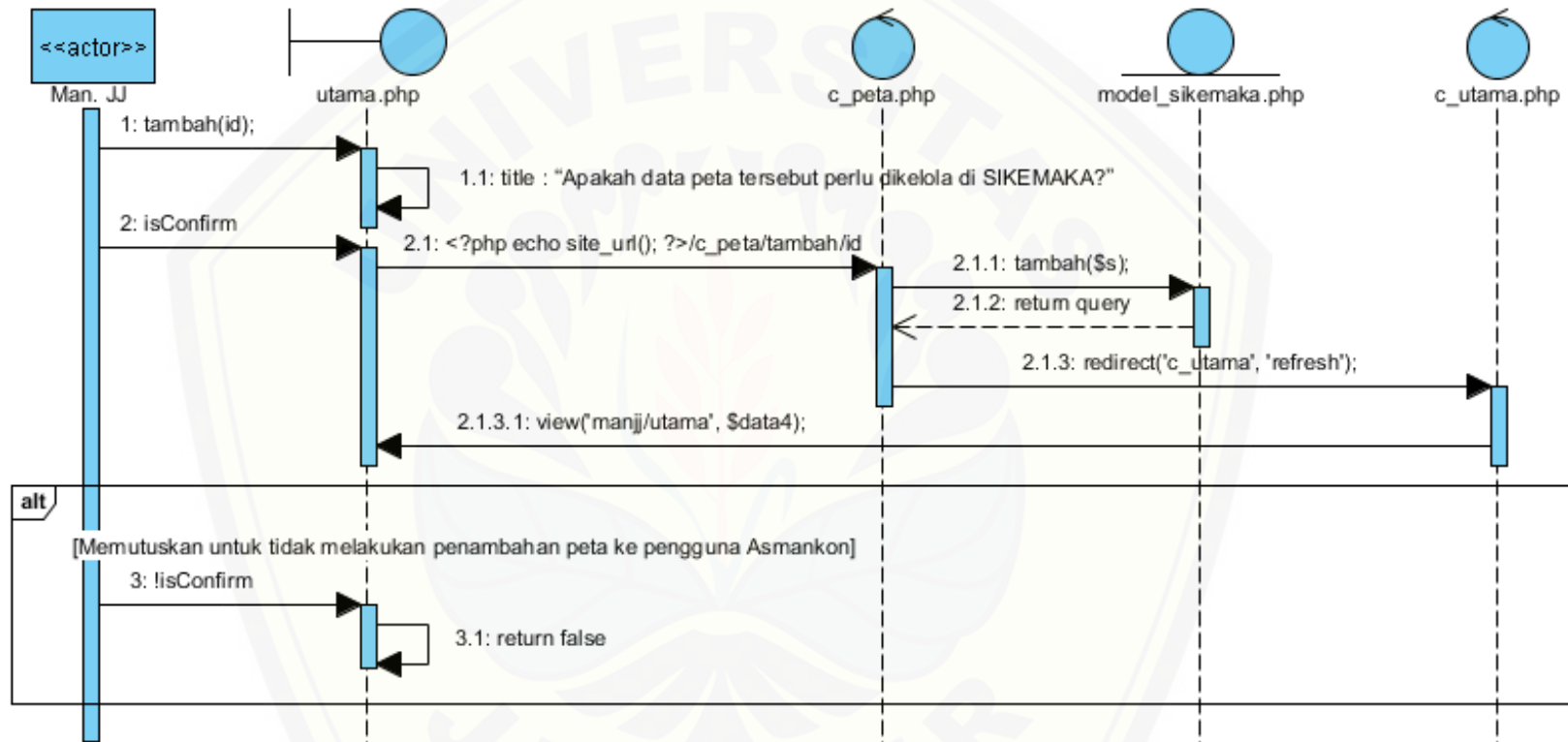
7. Sequence Diagram Menghapus Data pada Dataset



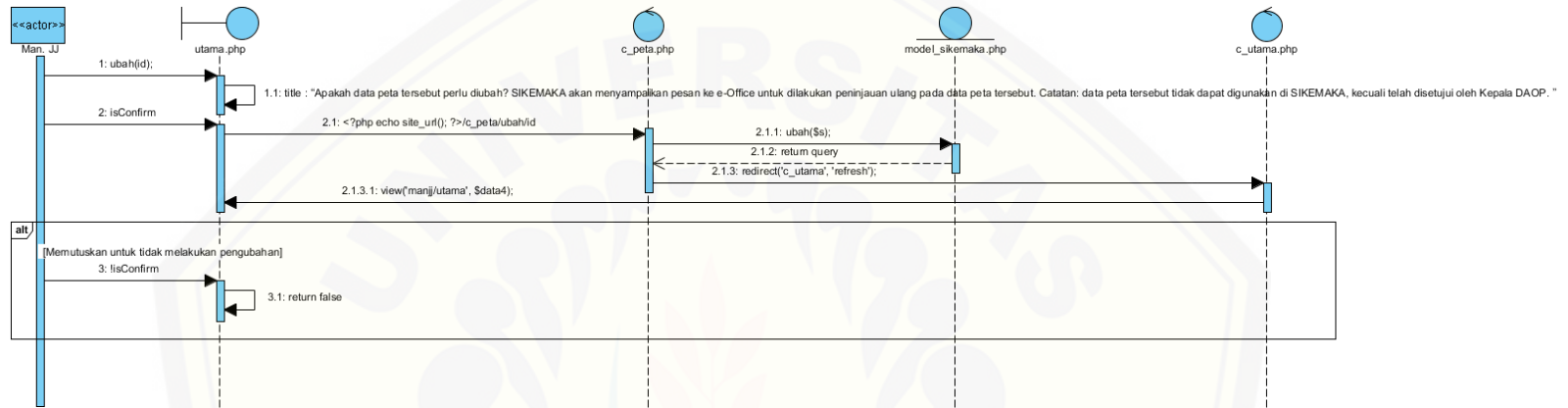
8. Sequence Diagram Pencarian Data pada Dataset



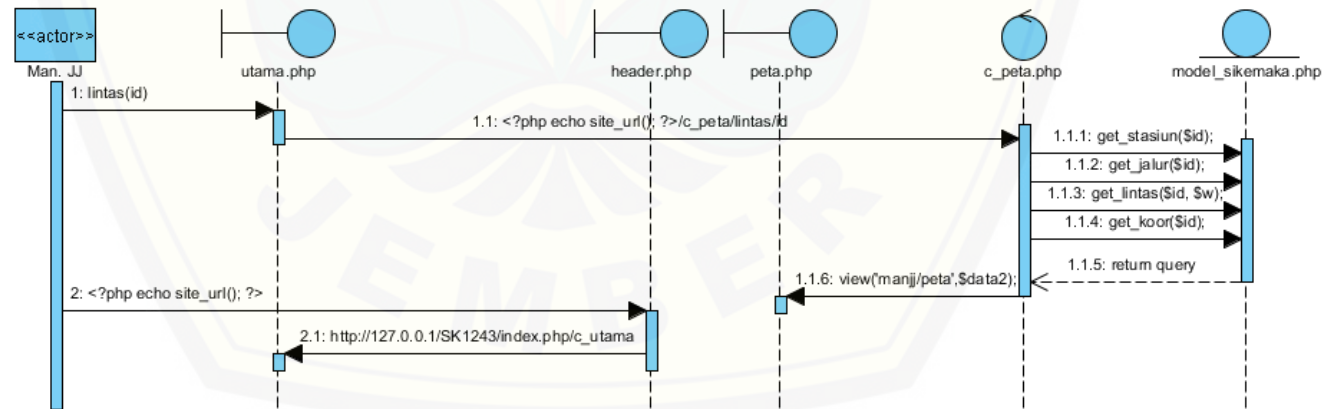
9. Sequence Diagram Menambah Data Peta ke Tabel Status Perancangan



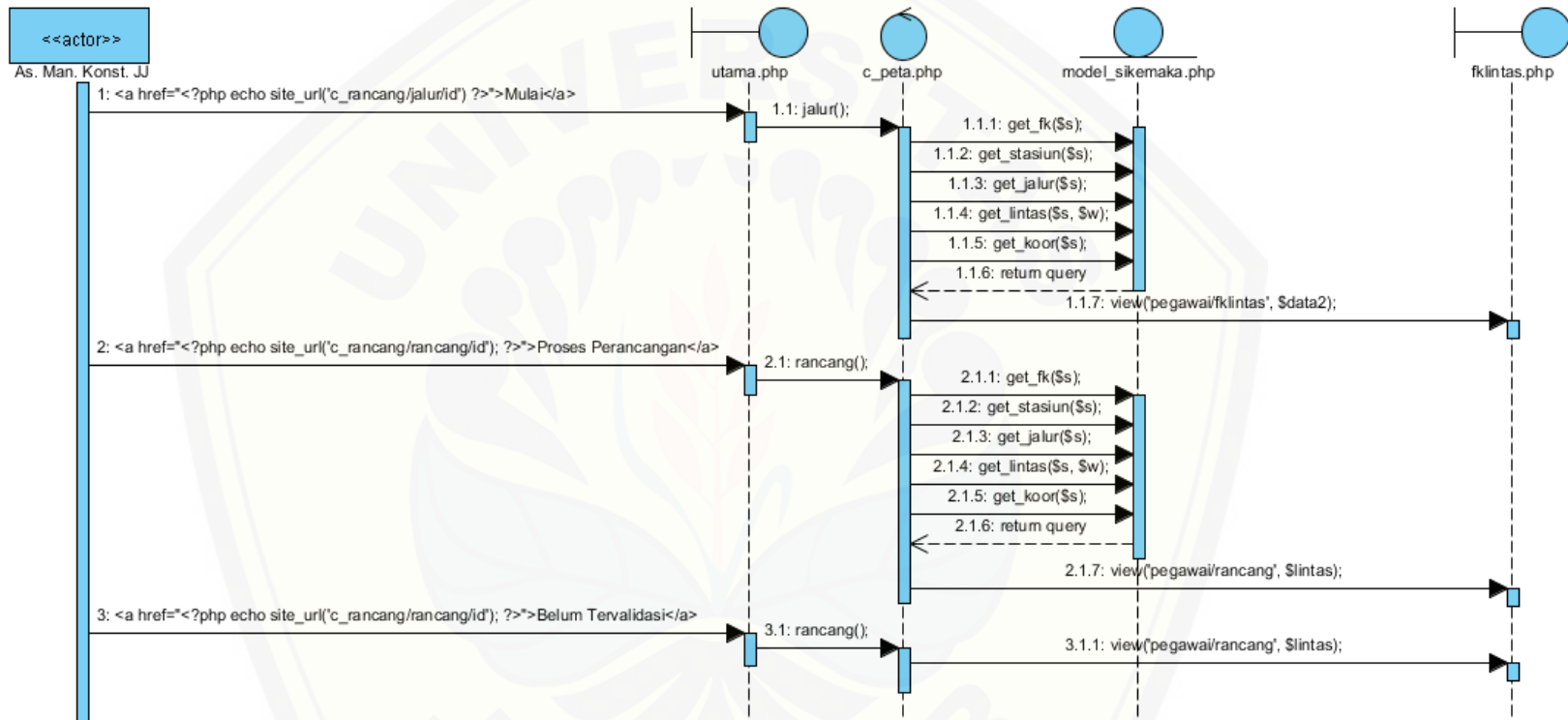
10. Sequence Diagram Menandai Data Peta untuk Diubah di Sistem Lain



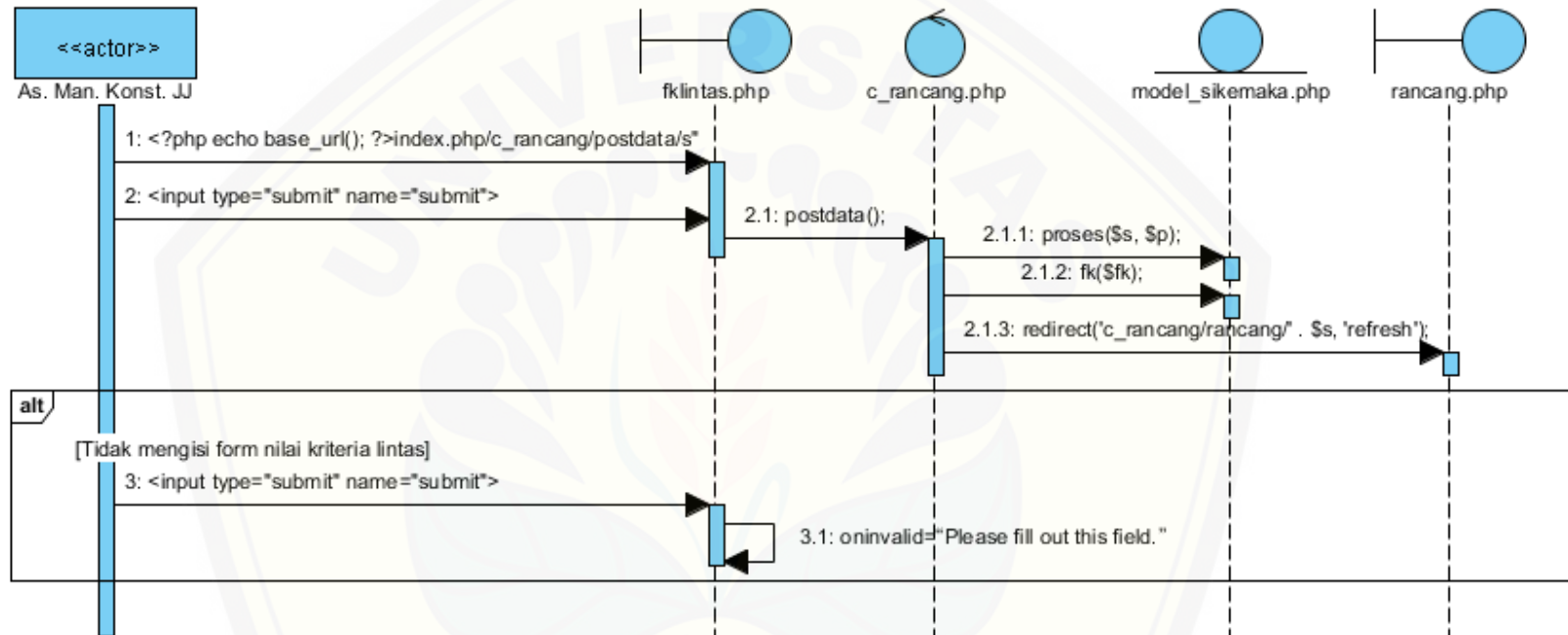
11. Sequence Diagram Menampilkan Peta dari Data Peta yang Dipilih



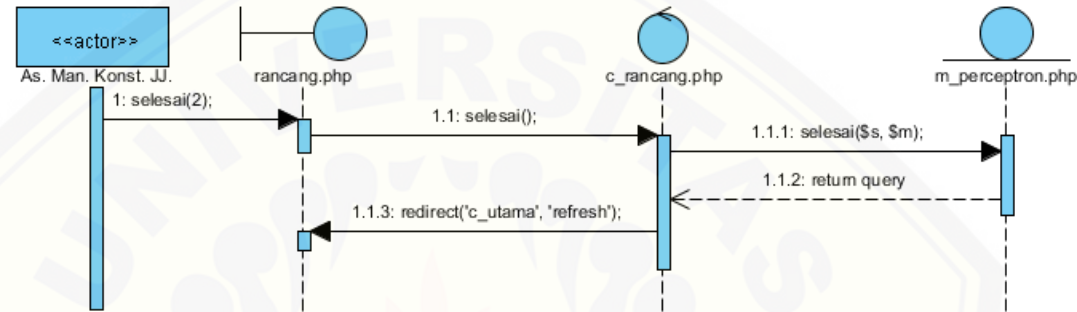
12. Sequence Diagram Meninjau Tabel Status Perancangan



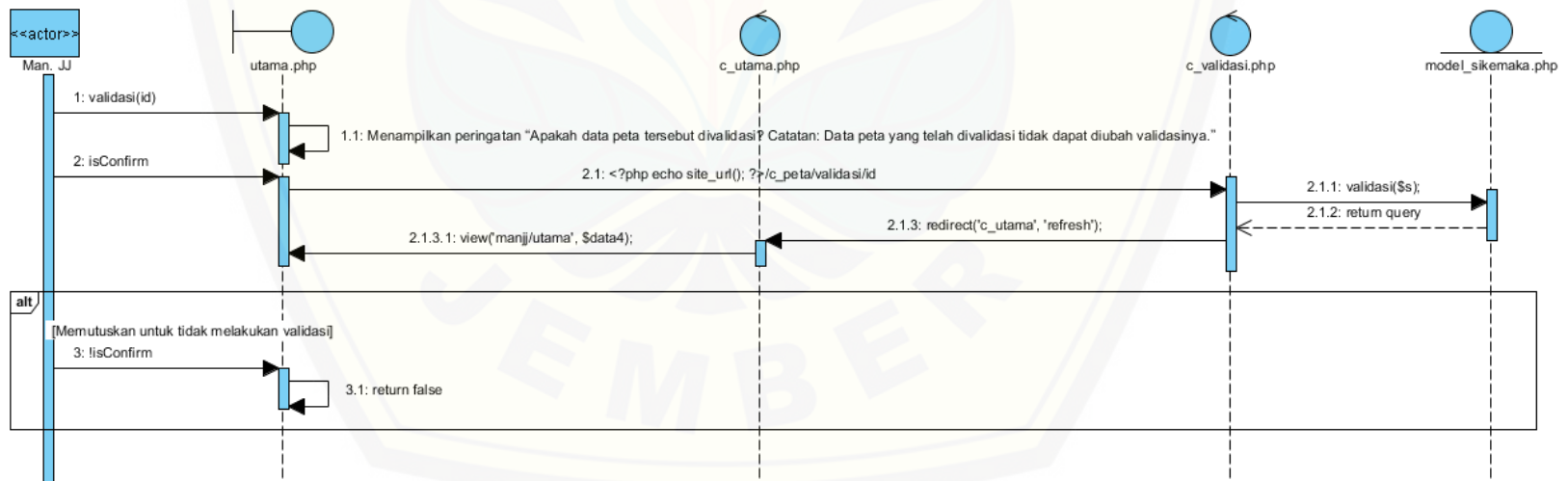
13. Sequence Diagram Mengisi Nilai Kriteria Lintas



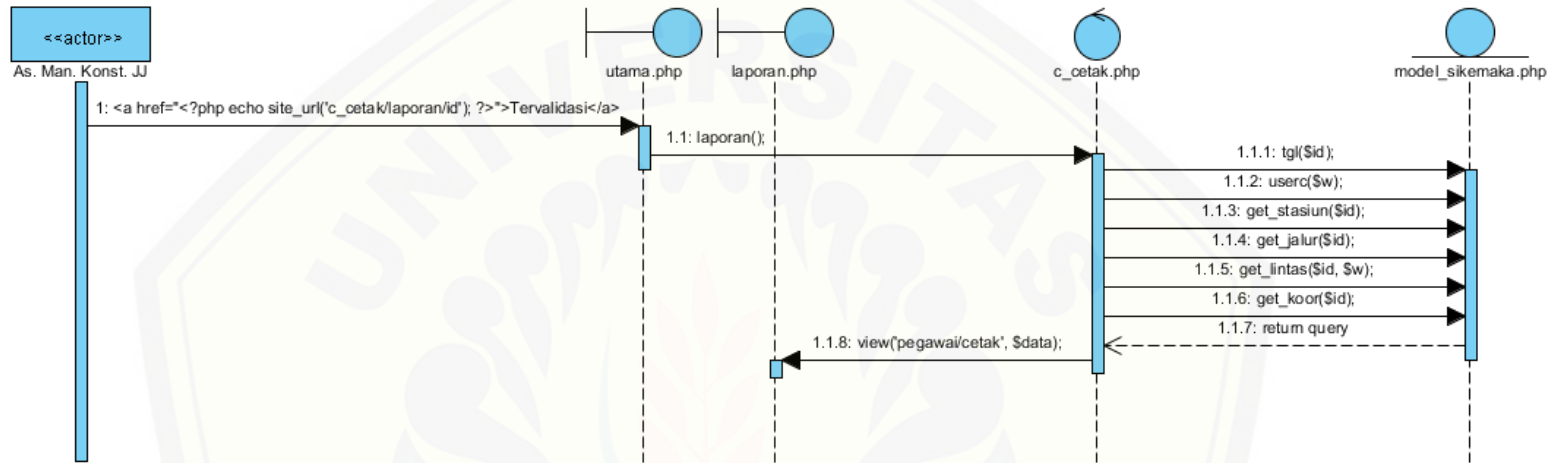
14. Sequence Diagram Menyelesaikan Klasifikasi



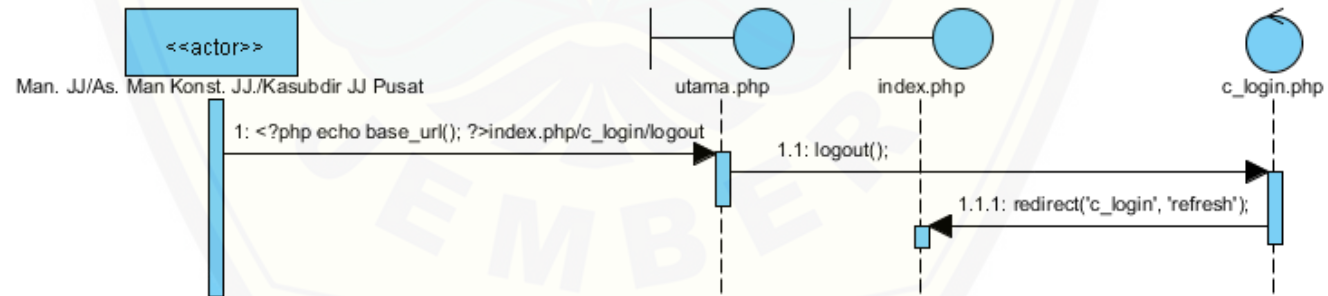
15. Sequence Diagram Melakukan Validasi Perancangan Kecepatan Maksimum pada Data Peta



16. Sequence Diagram Mencetak Laporan Desain Kecepatan Maksimum



17. Sequence Diagram Logout Sistem



LAMPIRAN C. KODE PROGRAM

1. Kode Program m_login.php

```

1  <?php
2
3  class M_login extends CI_Model {
4
5      public function user($nipp) {
6          $this->db->select('*');
7          $this->db->from('pegawai');
8          $this->db->join('jabatan',
9              'jabatan.kodejabatan = pegawai.KodeJabatan');
10         $this->db->join('wilayah', 'pegawai.WilID
11         = wilayah.wilid');
12         $this->db->where('pegawai.nipp', $nipp);
13         $q = $this->db->get()->result_array();
14         if ($q)
15             return $q[0];
16     }
17
18     public function login($nipp, $ks) {
19         if ($nipp != '' && $ks != '') {
20             $query = $this->db-
21 >get_where("pegawai", array('nipp' => $nipp,
22 'Password' => md5($ks)));
23             return $query->result_array();
24         }
25     }
26 }
27
28 }
29
30 }
31
32 }
33
34 }
35
36 }
37
38 }
39
40 }
41
42 }
43
44 }
45
46 }
47
48 }
49
50 }
51
52 }
53
54 }
55
56 }
57
58 }
59
60 }
61
62 }
63
64 }
65
66 }
67
68 }
69
70 }
71
72 }
73
74 }
75
76 }
77
78 }
79
80 }
81
82 }
83
84 }
85
86 }
87
88 }
89
90 }
91
92 }
93
94 }
95
96 }
97
98 }
99
100 }
101
102 }
103
104 }
105
106 }
107
108 }
109
110 }
111
112 }
113
114 }
115
116 }
117
118 }
119
120 }
121
122 }
123
124 }
125
126 }
127
128 }
129
130 }
131
132 }
133
134 }
135
136 }
137
138 }
139
140 }
141
142 }
143
144 }
145
146 }
147
148 }
149
150 }
151
152 }
153
154 }
155
156 }
157
158 }
159
160 }
161
162 }
163
164 }
165
166 }
167
168 }
169
170 }
171
172 }
173
174 }
175
176 }
177
178 }
179
180 }
181
182 }
183
184 }
185
186 }
187
188 }
189
190 }
191
192 }
193
194 }
195
196 }
197
198 }
199
200 }
201
202 }
203
204 }
205
206 }
207
208 }
209
210 }
211
212 }
213
214 }
215
216 }
217
218 }
219
220 }
221
222 }
223
224 }
225
226 }
227
228 }
229
230 }
231
232 }
233
234 }
235
236 }
237
238 }
239
240 }
241
242 }
243
244 }
245
246 }
247
248 }
249
250 }
251
252 }
253
254 }
255
256 }
257
258 }
259
260 }
261
262 }
263
264 }
265
266 }
267
268 }
269
270 }
271
272 }
273
274 }
275
276 }
277
278 }
279
280 }
281
282 }
283
284 }
285
286 }
287
288 }
289
290 }
291
292 }
293
294 }
295
296 }
297
298 }
299
300 }
301
302 }
303
304 }
305
306 }
307
308 }
309
310 }
311
312 }
313
314 }
315
316 }
317
318 }
319
320 }
321
322 }
323
324 }
325
326 }
327
328 }
329
330 }
331
332 }
333
334 }
335
336 }
337
338 }
339
340 }
341
342 }
343
344 }
345
346 }
347
348 }
349
350 }
351
352 }
353
354 }
355
356 }
357
358 }
359
360 }
361
362 }
363
364 }
365
366 }
367
368 }
369
370 }
371
372 }
373
374 }
375
376 }
377
378 }
379
380 }
381
382 }
383
384 }
385
386 }
387
388 }
389
390 }
391
392 }
393
394 }
395
396 }
397
398 }
399
400 }
401
402 }
403
404 }
405
406 }
407
408 }
409
410 }
411
412 }
413
414 }
415
416 }
417
418 }
419
420 }
421
422 }
423
424 }
425
426 }
427
428 }
429
430 }
431
432 }
433
434 }
435
436 }
437
438 }
439
440 }
441
442 }
443
444 }
445
446 }
447
448 }
449
450 }
451
452 }
453
454 }
455
456 }
457
458 }
459
460 }
461
462 }
463
464 }
465
466 }
467
468 }
469
470 }
471
472 }
473
474 }
475
476 }
477
478 }
479
480 }
481
482 }
483
484 }
485
486 }
487
488 }
489
490 }
491
492 }
493
494 }
495
496 }
497
498 }
499
500 }
501
502 }
503
504 }
505
506 }
507
508 }
509
510 }
511
512 }
513
514 }
515
516 }
517
518 }
519
520 }
521
522 }
523
524 }
525
526 }
527
528 }
529
530 }
531
532 }
533
534 }
535
536 }
537
538 }
539
540 }
541
542 }
543
544 }
545
546 }
547
548 }
549
550 }
551
552 }
553
554 }
555
556 }
557
558 }
559
560 }
561
562 }
563
564 }
565
566 }
567
568 }
569
570 }
571
572 }
573
574 }
575
576 }
577
578 }
579
580 }
581
582 }
583
584 }
585
586 }
587
588 }
589
590 }
591
592 }
593
594 }
595
596 }
597
598 }
599
600 }
601
602 }
603
604 }
605
606 }
607
608 }
609
610 }
611
612 }
613
614 }
615
616 }
617
618 }
619
620 }
621
622 }
623
624 }
625
626 }
627
628 }
629
630 }
631
632 }
633
634 }
635
636 }
637
638 }
639
640 }
641
642 }
643
644 }
645
646 }
647
648 }
649
650 }
651
652 }
653
654 }
655
656 }
657
658 }
659
660 }
661
662 }
663
664 }
665
666 }
667
668 }
669
670 }
671
672 }
673
674 }
675
676 }
677
678 }
679
680 }
681
682 }
683
684 }
685
686 }
687
688 }
689
690 }
691
692 }
693
694 }
695
696 }
697
698 }
699
700 }
701
702 }
703
704 }
705
706 }
707
708 }
709
710 }
711
712 }
713
714 }
715
716 }
717
718 }
719
720 }
721
722 }
723
724 }
725
726 }
727
728 }
729
730 }
731
732 }
733
734 }
735
736 }
737
738 }
739
740 }
741
742 }
743
744 }
745
746 }
747
748 }
749
750 }
751
752 }
753
754 }
755
756 }
757
758 }
759
760 }
761
762 }
763
764 }
765
766 }
767
768 }
769
770 }
771
772 }
773
774 }
775
776 }
777
778 }
779
780 }
781
782 }
783
784 }
785
786 }
787
788 }
789
790 }
791
792 }
793
794 }
795
796 }
797
798 }
799
800 }
801
802 }
803
804 }
805
806 }
807
808 }
809
810 }
811
812 }
813
814 }
815
816 }
817
818 }
819
820 }
821
822 }
823
824 }
825
826 }
827
828 }
829
830 }
831
832 }
833
834 }
835
836 }
837
838 }
839
840 }
841
842 }
843
844 }
845
846 }
847
848 }
849
850 }
851
852 }
853
854 }
855
856 }
857
858 }
859
860 }
861
862 }
863
864 }
865
866 }
867
868 }
869
870 }
871
872 }
873
874 }
875
876 }
877
878 }
879
880 }
881
882 }
883
884 }
885
886 }
887
888 }
889
890 }
891
892 }
893
894 }
895
896 }
897
898 }
899
900 }
901
902 }
903
904 }
905
906 }
907
908 }
909
910 }
911
912 }
913
914 }
915
916 }
917
918 }
919
920 }
921
922 }
923
924 }
925
926 }
927
928 }
929
930 }
931
932 }
933
934 }
935
936 }
937
938 }
939
940 }
941
942 }
943
944 }
945
946 }
947
948 }
949
950 }
951
952 }
953
954 }
955
956 }
957
958 }
959
960 }
961
962 }
963
964 }
965
966 }
967
968 }
969
970 }
971
972 }
973
974 }
975
976 }
977
978 }
979
980 }
981
982 }
983
984 }
985
986 }
987
988 }
989
990 }
991
992 }
993
994 }
995
996 }
997
998 }
999
1000 }
1001
1002 }
1003
1004 }
1005
1006 }
1007
1008 }
1009
1010 }
1011
1012 }
1013
1014 }
1015
1016 }
1017
1018 }
1019
1020 }
1021
1022 }
1023
1024 }
1025
1026 }
1027
1028 }
1029
1030 }
1031
1032 }
1033
1034 }
1035
1036 }
1037
1038 }
1039
1040 }
1041
1042 }
1043
1044 }
1045
1046 }
1047
1048 }
1049
1050 }
1051
1052 }
1053
1054 }
1055
1056 }
1057
1058 }
1059
1060 }
1061
1062 }
1063
1064 }
1065
1066 }
1067
1068 }
1069
1070 }
1071
1072 }
1073
1074 }
1075
1076 }
1077
1078 }
1079
1080 }
1081
1082 }
1083
1084 }
1085
1086 }
1087
1088 }
1089
1090 }
1091
1092 }
1093
1094 }
1095
1096 }
1097
1098 }
1099
1100 }
1101
1102 }
1103
1104 }
1105
1106 }
1107
1108 }
1109
1110 }
1111
1112 }
1113
1114 }
1115
1116 }
1117
1118 }
1119
1120 }
1121
1122 }
1123
1124 }
1125
1126 }
1127
1128 }
1129
1130 }
1131
1132 }
1133
1134 }
1135
1136 }
1137
1138 }
1139
1140 }
1141
1142 }
1143
1144 }
1145
1146 }
1147
1148 }
1149
1150 }
1151
1152 }
1153
1154 }
1155
1156 }
1157
1158 }
1159
1160 }
1161
1162 }
1163
1164 }
1165
1166 }
1167
1168 }
1169
1170 }
1171
1172 }
1173
1174 }
1175
1176 }
1177
1178 }
1179
1180 }
1181
1182 }
1183
1184 }
1185
1186 }
1187
1188 }
1189
1190 }
1191
1192 }
1193
1194 }
1195
1196 }
1197
1198 }
1199
1200 }
1201
1202 }
1203
1204 }
1205
1206 }
1207
1208 }
1209
1210 }
1211
1212 }
1213
1214 }
1215
1216 }
1217
1218 }
1219
1220 }
1221
1222 }
1223
1224 }
1225
1226 }
1227
1228 }
1229
1230 }
1231
1232 }
1233
1234 }
1235
1236 }
1237
1238 }
1239
1240 }
1241
1242 }
1243
1244 }
1245
1246 }
1247
1248 }
1249
1250 }
1251
1252 }
1253
1254 }
1255
1256 }
1257
1258 }
1259
1260 }
1261
1262 }
1263
1264 }
1265
1266 }
1267
1268 }
1269
1270 }
1271
1272 }
1273
1274 }
1275
1276 }
1277
1278 }
1279
1280 }
1281
1282 }
1283
1284 }
1285
1286 }
1287
1288 }
1289
1290 }
1291
1292 }
1293
1294 }
1295
1296 }
1297
1298 }
1299
1300 }
1301
1302 }
1303
1304 }
1305
1306 }
1307
1308 }
1309
1310 }
1311
1312 }
1313
1314 }
1315
1316 }
1317
1318 }
1319
1320 }
1321
1322 }
1323
1324 }
1325
1326 }
1327
1328 }
1329
1330 }
1331
1332 }
1333
1334 }
1335
1336 }
1337
1338 }
1339
1340 }
1341
1342 }
1343
1344 }
1345
1346 }
1347
1348 }
1349
1350 }
1351
1352 }
1353
1354 }
1355
1356 }
1357
1358 }
1359
1360 }
1361
1362 }
1363
1364 }
1365
1366 }
1367
1368 }
1369
1370 }
1371
1372 }
1373
1374 }
1375
1376 }
1377
1378 }
1379
1380 }
1381
1382 }
1383
1384 }
1385
1386 }
1387
1388 }
1389
1390 }
1391
1392 }
1393
1394 }
1395
1396 }
1397
1398 }
1399
1400 }
1401
1402 }
1403
1404 }
1405
1406 }
1407
1408 }
1409
1410 }
1411
1412 }
1413
1414 }
1415
1416 }
1417
1418 }
1419
1420 }
1421
1422 }
1423
1424 }
1425
1426 }
1427
1428 }
1429
1430 }
1431
1432 }
1433
1434 }
1435
1436 }
1437
1438 }
1439
1440 }
1441
1442 }
1443
1444 }
1445
1446 }
1447
1448 }
1449
1450 }
1451
1452 }
1453
1454 }
1455
1456 }
1457
1458 }
1459
1460 }
1461
1462 }
1463
1464 }
1465
1466 }
1467
1468 }
1469
1470 }
1471
1472 }
1473
1474 }
1475
1476 }
1477
1478 }
1479
1480 }
1481
1482 }
1483
1484 }
1485
1486 }
1487
1488 }
1489
1490 }
1491
1492 }
1493
1494 }
1495
1496 }
1497
1498 }
1499
1500 }
1501
1502 }
1503
1504 }
1505
1506 }
1507
1508 }
1509
1510 }
1511
1512 }
1513
1514 }
1515
1516 }
1517
1518 }
1519
1520 }
1521
1522 }
1523
1524 }
1525
1526 }
1527
1528 }
1529
1530 }
1531
1532 }
1533
1534 }
1535
1536 }
1537
1538 }
1539
1540 }
1541
1542 }
1543
1544 }
1545
1546 }
1547
1548 }
1549
1550 }
1551
1552 }
1553
1554 }
1555
1556 }
1557
1558 }
1559
1560 }
1561
1562 }
1563
1564 }
1565
1566 }
1567
1568 }
1569
1570 }
1571
1572 }
1573
1574 }
1575
1576 }
1577
1578 }
1579
1580 }
1581
1582 }
1583
1584 }
1585
1586 }
1587
1588 }
1589
1590 }
1591
1592 }
1593
1594 }
1595
1596 }
1597
1598 }
1599
1600 }
1601
1602 }
1603
1604 }
1605
1606 }
1607
1608 }
1609
1610 }
1611
1612 }
1613
1614 }
1615
1616 }
1617
1618 }
1619
1620 }
1621
1622 }
1623
1624 }
1625
1626 }
1627
1628 }
1629
1630 }
1631
1632 }
1633
1634 }
1635
1636 }
1637
1638 }
1639
1640 }
1641
1642 }
1643
1644 }
1645
1646 }
1647
1648 }
1649
1650 }
1651
1652 }
1653
1654 }
1655
1656 }
1657
1658 }
1659
1660 }
1661
1662 }
1663
1664 }
1665
1666 }
1667
1668 }
1669
1670 }
1671
1672 }
1673
1674 }
1675
1676 }
1677
1678 }
1679
1680 }
1681
1682 }
1683
1684 }
1685
1686 }
1687
1688 }
1689
1690 }
1691
1692 }
1693
1694 }
1695
1696 }
1697
1698 }
1699
1700 }
1701
1702 }
1703
1704 }
1705
1706 }
1707
1708 }
1709
1710 }
1711
1712 }
1713
1714 }
1715
1716 }
1717
1718 }
1719
1720 }
1721
1722 }
1723
1724 }
1725
1726 }
1727
1728 }
1729
1730 }
1731
1732 }
1733
1734 }
1735
1736 }
1737
1738 }
1739
1740 }
1741
1742 }
1743
1744 }
1745
1746 }
1747
1748 }
1749
1750 }
1751
1752 }
1753
1754 }
1755
1756 }
1757
1758 }
1759
1760 }
1761
1762 }
1763
1764 }
1765
1766 }
1767
1768 }
1769
1770 }
1771
1772 }
1773
1774 }
1775
1776 }
1777
1778 }
1779
1780 }
1781
1782 }
1783
1784 }
1785
1786 }
1787
1788 }
1789
1790 }
1791
1792 }
1793
1794 }
1795
1796 }
1797
1798 }
1799
1800 }
1801
1802 }
1803
1804 }
1805
1806 }
1807
1808 }
1809
1810 }
1811
1812 }
1813
1814 }
1815
1816 }
1817
1818 }
1819
1820 }
1821
1822 }
1823
1824 }
1825
1826 }
1827
1828 }
1829
1830 }
1831
1832 }
1833
1834 }
1835
1836 }
1837
1838 }
1839
1840 }
1841
1842 }
1843
1844 }
1845
1846 }
1847
1848 }
1849
1850 }
1851
1852 }
1853
1854 }
1855
1856 }
1857
1858 }
1859
1860 }
1861
1862 }
1863
1864 }
1865
1866 }
1867
1868 }
1869
1870 }
1871
1872 }
1873
1874 }
1875
1876 }
1877
1878 }
1879
1880 }
1881
1882 }
1883
1884 }
1885
1886 }
1887
1888 }
1889
1890 }
1891
1892 }
1893
1894 }
1895
1896 }
1897
1898 }
1899
1900 }
1901
1902 }
1903
1904 }
1905
1906 }
1907
1908 }
1909
1910 }
1911
1912 }
1913
1914 }
1915
1916 }
1917
1918 }
1919
1920 }
1921
1922 }
1923
1924 }
1925
1926 }
1927
1928 }
1929
1930 }
1931
1932 }
1933
1934 }
1935
1936 }
1937
1938 }
1939
1940 }
1941
1942 }
1943
1944 }
1945
1946 }
1947
1948 }
1949
1950 }
1951
1952 }
1953
1954 }
1955
1956 }
1957
1958 }
1959
1960 }
1961
1962 }
1963
1964 }
1965
1966 }
1967
1968 }
1969
1970 }
1971
1972 }
1973
1974 }
1975
1976 }
1977
1978 }
1979
1980 }
1981
1982 }
1983
1984 }
1985
1986 }
1987
1988 }
1989
1990 }
1991
1992 }
1993
1994 }
1995
1996 }
1997
1998 }
1999
2000 }
2001
2002 }
2003
2004 }
2005
2006 }
2007
2008 }
2009
2010 }
2011
2012 }
2013
2014 }
2015
2016 }
2017
2018 }
2019
2020 }
2021
2022 }
2023
2024 }
2025
2026 }
2027
2028 }
2029
2030 }
2031
2032 }
2033
2034 }
2035
2036 }
2037
2038 }
2039
2040 }
2041
2042 }
2043
2044 }
2045
2046 }
2047
2048 }
2049
2050 }
2051
2052 }
2053
2054 }
2055
2056 }
2057
2058 }
2059
2060 }
2061
2062 }
2063
2064 }
2065
2066 }
2067
2068 }
2069
2070 }
2071
2072 }
2073
2074 }
2075
2076 }
2077
2078 }
2079
2080 }
2081
2082 }
2083
2084 }
2085
2086 }
2087
2088 }
2089
2090 }
2091
2092 }
2093
2094 }
2095
2096 }
2097
2098 }
2099
2100 }
2101
2102 }
2103
2104 }
2105
2106 }
2107
2108 }
2109
2110 }
2111
2112 }
2113
2114 }
2115
2116 }
2117
2118 }
2119
2120 }
2121
2122 }
2123
2124 }
2125
2126 }
2127
2128 }
2129
2130 }
2131
2132 }
2133
2134 }
2135
2136 }
2137
2138 }
2139
2140 }
2141
2142 }
2143
2144 }
2145
2146 }
2147
2148 }
2149
2150 }
2151
2152 }
2153
2154 }
2155
2156 }
2157
2158 }
2159
2160 }
2161
2162 }
2163
2164 }
2165
2166 }
2167
2168 }
2169
2170 }
2171
2172 }
2173
2174 }
2175
2176 }
2177
2178 }
2179
2180 }
2181
2182 }
2183
2184 }
2185
2186 }
2187
2188 }
2189
2190 }
2191
2192 }
2193
2194 }
2195
2196 }
2197
2198 }
2199
2200 }
2201
2202 }
2203
2204 }
2205
2206 }
2207
2208 }
2209
2210 }
2211
2212 }
2213
2214 }
2215
2216 }
2217
2218 }
2219
2220 }
2221
2222 }
2223
2224 }
2225
2226 }
2227
2228 }
2229
2230 }
2231
2232 }
2233
2234 }
2235
2236 }
2237
2238 }
2239
2240 }
2241
2242 }
2243
2244 }
2245
2246 }
2247
2248 }
2249
2250 }
2251
2252 }
2253
2254 }
2255
2256 }
2257
2258 }
2259
2260 }
2261
2262 }
2263
2264 }
2265
2266 }
2267
2268 }
2269
2270 }
2271
2272 }
2273
2274 }
2275
2276 }
2277
2278 }
2279
2280 }
2281
2282 }
2283
2284 }
2285
2286 }
2287
2288 }
2289
2290 }
2291
2292 }
2293
2294 }
2295
2296 }
2297
2298 }
2299
2300 }
2301
2302 }
2303
2304 }
2305
2306 }
2307
2308 }
2309
2310 }
2311
2312 }
2313
2314 }
2315
2316 }
2317
2318 }
2319
2320 }
2321
2322 }
2323
2324 }
2325
2326 }
2327
2328 }
2329
2330 }
2331
2332 }
2333
2334 }
2335
2336 }
2337
2338 }
2339
2340 }
2341
2342 }
2343
2344 }
2345
2346 }
2347
2348 }
2349
2350 }
2351
2352 }
2353
2354 }
2355
2356 }
2357
2358 }
2359
2360 }
2361
2362 }
2363
2364 }
2365
2366 }
2367
2368 }
2369
2370 }
2371
2372 }
2373
2374 }
2375
2376 }
2377
2378 }
2379
2380 }
2381
2382 }
238
```

```

13      "8",
          COALESCE((tba - (SELECT MIN(tba) FROM
dataset_fix WHERE hapus = 0)) / NULLIF((55) -
(SELECT MIN(tba) FROM dataset_fix WHERE hapus =
0), 0), 0) AS "9",
14      COALESCE((lbb - (SELECT MIN(lbb) FROM
dataset_fix WHERE hapus = 0)) / NULLIF((SELECT
MAX(lbb) FROM dataset_fix WHERE hapus = 0) -
(SELECT MIN(lbb) FROM dataset_fix WHERE hapus =
0), 0), 0) AS "10",
15      COALESCE((landai - (SELECT
MIN(landai) FROM dataset_fix WHERE hapus = 0)) /
NULLIF((SELECT MAX(landai) FROM dataset_fix WHERE
hapus = 0) - (SELECT MIN(landai) FROM dataset_fix
WHERE hapus = 0), 0), 0) AS "11",
16      COALESCE((rminv - (SELECT MIN(rminv)
FROM dataset_fix WHERE hapus = 0)) /
NULLIF((SELECT MAX(rminv) FROM dataset_fix WHERE
hapus = 0) - (SELECT MIN(rminv) FROM dataset_fix
WHERE hapus = 0), 0), 0) AS "12",
17      COALESCE((rmintph - (SELECT
MIN(rmintph) FROM dataset_fix WHERE hapus = 0)) /
NULLIF((SELECT MAX(rmintph) FROM dataset_fix
WHERE hapus = 0) - (SELECT MIN(rmintph) FROM
dataset_fix WHERE hapus = 0), 0), 0) AS "13",
18      COALESCE((rminph - (SELECT
MIN(rminph) FROM dataset_fix WHERE hapus = 0)) /
NULLIF((SELECT MAX(rminph) FROM dataset_fix WHERE
hapus = 0) - (SELECT MIN(rminph) FROM dataset_fix
WHERE hapus = 0), 0), 0) AS "14",
19      COALESCE((lbj - (SELECT MIN(lbj) FROM
dataset_fix WHERE hapus = 0)) / NULLIF((SELECT
MAX(lbj) FROM dataset_fix WHERE hapus = 0) -
(SELECT MIN(lbj) FROM dataset_fix WHERE hapus =
0), 0), 0) AS "15", vmaks AS "16");
20      $this->db->from('dataset_fix');
21      $this->db->where('hapus != 1');
22      $this->db->order_by('id');
23      $q = $this->db->get()->result_array();
24      return $q;
25      }
26
27      public function data_n() {
28          $query = $this->db->query("SELECT
MIN(tba) AS tba, MAX(tba) AS tbax, MIN(lbb) AS
lbb, MAX(lbb) AS lbbx,
29          MIN(landai) AS landai, MAX(landai) AS
landaix, MIN(rminv) AS rminv, MAX(rminv) AS
rminvx, MIN(rmintph) AS rmintph,
30          MAX(rmintph) AS rmintphx, MIN(rminph)
AS rminph, MAX(rminph) AS rminphx, MIN(lbj) AS
lbj, MAX(lbj) AS lbjx
31          FROM dataset_fix WHERE hapus = 0");

```



```

32         return $query->result_array();
33     }
34
35     public function klasidata($data) {
36         $this->db->insert('logjalur', $data);
37     }
38
39     public function vmaks($v, $id) {
40         $this->db->query("UPDATE jalur SET vmaks
= '$v' WHERE id = $id");
41     }
42
43     public function get_bobotbias() {
44         $query = $this->db->query("SELECT
array_to_json(bobotbias) AS bb FROM loghitung
ORDER BY na DESC, bobotbias ASC LIMIT 1");
45         return $query->result_array();
46     }
47
48 }
49
50 ?>

```

3. Kode Program model_sikemaka.php

```

1 <?php
2
3 class Model_sikemaka extends CI_Model {
4
5     public function validasi($id) {
6         $this->db->query("UPDATE lintaswilayah
SET validasi = 1, ubah = 2, masa_validasi = now()
7         WHERE LintasID = $id");
8     }
9
10    public function tambah($id) {
11        $this->db->query("UPDATE lintaswilayah
SET tambah = 1 WHERE LintasID = $id");
12    }
13
14    public function ubah($id) {
15        $this->db->query("UPDATE lintaswilayah
SET ubah = 1, tambah = 2 WHERE LintasID = $id");
16    }
17
18    public function fk($fk) {
19        $this->db->insert('fklintas', $fk);
20    }
21
22    public function get_fk($s) {
23        $this->db->select('tr, jp, jb, lbb,
tba');
24        $this->db->from('fklintas');

```

```
25     $this->db->where('idlintas', $s);
26     $q = $this->db->get();
27     return $q->result();
28 }
29
30 public function get_stasiun($i) {
31     $query = $this->db->query("SELECT
32 st_asgeojson(geom) AS st, name, posisi
33     FROM stasiun WHERE lintasid = $i");
34     return $query->result();
35 }
36
37 public function get_jalur($i) {
38     $query = $this->db->query("SELECT dups.*,
39 id, st_asgeojson(geom) AS lintas, name, vmaks
40     FROM (SELECT idjalur, r60, r54, r50,
41 r42, beton, kayu, baja, ganda, tunggal, tba, lbb,
42 landai, rminv, rmintph,
43     rminph, lbj, jembatan, viaduk,
44 ROW_NUMBER() OVER(PARTITION BY idjalur ORDER BY
45 waktu DESC) AS baris FROM logjalur) dups
46     FULL OUTER JOIN jalur ON dups.idjalur
47 = jalur.id
48     WHERE lintasid = $i AND baris = 1 OR
49 baris IS NULL ORDER BY id");
50     return $query->result();
51 }
52
53 public function get_lintas($i, $w) {
54     $query = $this->db->query("SELECT lintas
55 AS l FROM lintas
56     JOIN lintaswilayah ON lintas.lintasid
57 = lintaswilayah.lintasid
58     WHERE lintas.lintasid = $i AND wilid
59 = $w");
60     return $query->result();
61 }
62
63 public function get_koor($i) {
64     $query = $this->db->query("SELECT
65 st_x(geom) AS koorx, st_y(geom) AS koory, name,
66 posisi
67     FROM stasiun WHERE lintasid = $i");
68     return $query->result();
69 }
70
71 public function petamjjfull($w) {
72     $query = $this->db->query("SELECT
73 CONCAT('L', '$w', LPAD(CAST(COALESCE(CAST(LWID AS
74 INT), 0) AS VARCHAR(2)), 2, '0'))
75     IDLINTAS, lintas, tambah, ubah,
76 validasi
77     FROM lintas JOIN lintaswilayah ON
```

```
lintas.lintasid = lintaswilayah.lintasid WHERE
62 wilid = $w ORDER BY IDLINTAS");
63     return $query->result();
64 }
65 public function proses($id, $p) {
66     $this->db->query("UPDATE lintaswilayah
SET proses = $p WHERE LintasID = $id");
67 }
68
69 public function selesai($id, $s) {
70     $this->db->query("UPDATE lintaswilayah
SET proses = 2, validasi = $s, masa_selesai =
now()
71     WHERE LintasID = $id");
72 }
73
74 public function tambah_hitungakh($na, $epoh,
75 $lr, $nipp, $larik) {
76     $this->db->query("INSERT INTO loghitung
(na, epoh, lr, peg, tgl, bobotbias) VALUES ($na,
'$epoh', $lr, $nipp, now(), ARRAY$larik)");
77 }
78 public function get_hitungakh($nipp) {
79     $query = $this->db->query("SELECT epoh,
lr, na, to_char(tgl, 'TMDay, DD TMMonth YYYY
HH24:MI:SS') AS tgl, array_to_json(bobotbias) AS
bb FROM loghitung
80 JOIN pegawai ON loghitung.peg = pegawai.nipp
WHERE loghitung.peg = $nipp ORDER BY na DESC,
bobotbias ASC LIMIT 1");
81     return $query->result_array();
82 }
83
84 public function jmld() {
85     $this->db->select('COUNT(*) AS data');
86     $this->db->from('dataset_fix');
87     $this->db->where('hapus != 1');
88     $q = $this->db->get()->result_array();
89     if ($q)
90         return $q[0];
91 }
92
93 public function cari($p, $c) {
94     $query = $this->db->query("SELECT * FROM
dataset_fix
95     WHERE hapus != 1 AND CASE WHEN
LOWER(vmaks) LIKE '%$c%' THEN $p::text LIKE
'%$c%'
96     ELSE $p::text LIKE '%$c%' END ORDER
97 BY ID");
    return $query->num_rows();
```

```
98     }
99
100     public function carid($batas, $mulai, $p, $c)
101     {
102         $query = $this->db->query("SELECT * FROM
dataset_fix
103         WHERE hapus != 1 AND CASE WHEN
LOWER(vmaks) LIKE '%$c%' THEN $p::text LIKE
'%$c%'
104         ELSE $p::text LIKE '%$c%' END ORDER
BY ID OFFSET $mulai LIMIT $batas");
        return $query->result();
105     }
106
107     public function get_dataset($batas, $mulai) {
108         $query = $this->db->query("SELECT * FROM
109 dataset_fix WHERE hapus != 1 ORDER BY ID OFFSET
$mulai LIMIT $batas");
        return $query->result();
110     }
111
112     public function get_datasetfull() {
113         $query = $this->db->query("SELECT * FROM
114 dataset_fix WHERE hapus != 1 ORDER BY ID");
        return $query->result();
115     }
116
117     public function tambahd($data) {
118         $this->db->insert('dataset_fix', $data);
119     }
120
121     public function ubahd($data) {
122         $this->db->where('id', $data['id']);
123         $this->db->update('dataset_fix', $data);
124     }
125
126     public function get_datasetid($id) {
127         $query = $this->db->query("SELECT * FROM
128 dataset_fix WHERE hapus != 1 AND id = $id ORDER
BY ID");
        return $query->result();
129     }
130
131     public function hapus($data) {
132         $this->db->query("UPDATE dataset_fix SET
133 hapus = 1 WHERE ID = $data");
    }
134
135     public function datapeta($w) {
136         return $this->db->query("SELECT
CONCAT('L', '$w', LPAD(CAST(COALESCE(CAST(LWID AS
INT), 0) AS VARCHAR(2)), 2, '0')) IDLINTAS,
lintas, tambah, validasi, proses
```

```
137         FROM lintas JOIN lintaswilayah ON
138         lintas.lintasid = lintaswilayah.lintasid
139         WHERE tambah = 1 AND wilid = $w ORDER
140         BY idlintas");
141     }
142     public function userc($w) {
143         $this->db->select('MAX(CASE WHEN
144         pegawai."KodeJabatan" = 1 THEN pegawai."nipp"
145         END) AS n1,
146         MAX(CASE WHEN pegawai."KodeJabatan" = 2
147         THEN pegawai."nipp" END) AS n2,
148         MAX(CASE WHEN pegawai."KodeJabatan" = 4
149         THEN pegawai."nipp" END) AS n3,
150         MAX(CASE WHEN pegawai."KodeJabatan" = 1
151         THEN pegawai."NamaPegawai" END) AS np1,
152         MAX(CASE WHEN pegawai."KodeJabatan" = 2
153         THEN pegawai."NamaPegawai" END) AS np2,
154         MAX(CASE WHEN pegawai."KodeJabatan" = 4
155         THEN pegawai."NamaPegawai" END) AS np3,
156         MAX(CASE WHEN pegawai."KodeJabatan" = 1
157         THEN jabatan.namajabatan END) AS j1,
158         MAX(CASE WHEN pegawai."KodeJabatan" = 2
159         THEN jabatan.namajabatan END) AS j2,
160         MAX(CASE WHEN pegawai."KodeJabatan" = 4
161         THEN jabatan.namajabatan END) AS j3,
162         MAX(wilayah.wilayah) AS w');
163         $this->db->from('pegawai');
164         $this->db->join('jabatan',
165         'jabatan.kodejabatan = pegawai.KodeJabatan');
166         $this->db->join('wilayah', 'pegawai.WilID
167         = wilayah.wilid');
168         $this->db->where('pegawai.WilID', $w);
169         $q = $this->db->get();
170         return $q->result();
171     }
172     public function tgl($id) {
173         $query = $this->db->query("SELECT
174         to_char(masa_selesai, 'TMDay, DD TMMonth YYYY
175         HH24:MI:SS') AS masa_selesai,
176         to_char(masa_validasi, 'TMDay, DD
177         TMMonth YYYY HH24:MI:SS') AS masa_validasi
178         FROM lintaswilayah WHERE lintasid =
179         $id");
180         return $query->result();
181     }
182 }
183 ?>
```


4. Kode Program c_perceptron.php

```
1 <?php
2
3 ini_set('max_execution_time', 100000);
4
5 class C_perceptron extends CI_Controller {
6
7     protected $keluaran = null; /* Nilai keluaran
8     pada setiap neuron */
9     protected $error = null; /* Nilai error /
10    kemelesetan */
11    protected $titik = null; /* Larik bobot atau
12    bias */
13    protected $jmlTitik = null; /* Larik jumlah
14    neuron pada setiap lapisan */
15    protected $learning_rate = null; /* Laju
16    Pembelajaran atau Learning Rate */
17    protected $perubahan_bobot = null; /*
18    Penyimpanan perubahan bobot yang dibuat dari epoh
19    sebelumnya */
20    protected $ket = null; /* Keterangan */
21
22    public function __construct() {
23        parent::__construct();
24        $this->load->helper('array');
25        $this->load->model('m_login');
26        $this->load->model('m_perceptron');
27        $this->load->model('model_sikemaka');
28    }
29
30    public function index() {
31        $ambil_akun = $this->m_login->user($this->
32        >session->userdata('uname'));
33        $dataTr = $this->m_perceptron->
34        >datatrain();
35        $data = array(
36            'user' => $ambil_akun
37        );
38        $res = array(
39            'hasil' => 'Tentukan Epoch dan
40            Learning Rate, lalu klik tombol Hitung.',
41            'epoh' => $this->model_sikemaka->
42            >get_hitungakh($this->session->
43            >userdata('uname'))[0]['epoh'],
44            'lr' => $this->model_sikemaka->
45            >get_hitungakh($this->session->
46            >userdata('uname'))[0]['lr'],
47            'datatr' => $dataTr
48        );
49
50        $stat = $this->session->userdata('lvl');
51        if ($stat == 3) {
52            $this->load->view('ahlijj/header',
```



```
38 $data);
    $this->load-
39 >view('ahlijj/perceptron', $res);
40     $this->load->view('footer');
41     } else {
42         redirect('c_login', 'refresh');
43     }
44 }
45
    public function awal($jmlTitik,
46 $learning_rate) {
47     $this->learning_rate = $learning_rate;
48     $this->jmlTitik = $jmlTitik;
49
    // Menempatkan pada dan menetapkan bobot
50 dengan nilai awal 0
51     for ($l = 1; $l < 2; $l++) {
52         for ($i = 0; $i < $this-
53 >jmlTitik[$l]; $i++) {
54             for ($j = 0; $j < $this-
55 >jmlTitik[$l - 1] - 1; $j++) {
56                 $this->titik[$l][$i][$j] = 0;
57             }
58             // bias pada neuron dengan nilai
59 awal 0
60             $this->titik[$l][$i][$this-
61 >jmlTitik[$l - 1] - 1] = 0;
62         }
63     }
64
    // Memulakan nilai perubahan bobot
65 sebelumnya ke 0 untuk pengulangan pertama
66     for ($l = 1; $l < 2; $l++) {
67         for ($i = 0; $i < $this-
68 >jmlTitik[$l]; $i++) {
69             for ($j = 0; $j < $this-
70 >jmlTitik[$l - 1] - 1; $j++) {
71                 $this-
72 >perubahan_bobot[$l][$i][$j] = (double) 0.0;
73             }
74         }
75     }
76
    public function result_conv($res) {
77     //Konversi hasil
78     if ($res >= 0.875) {
79         $sts = "120 km/jam";
80     } else if ($res >= 0.625 && $res < 0.875)
81     {
82         $sts = "110 km/jam";
83     } else if ($res >= 0.375 && $res < 0.625)
84     {
```

```

78         $sts = "100 km/jam";
           } else if ($res >= 0.125 && $res < 0.375)
79     {
80         $sts = "90 km/jam";
81     } else {
82         $sts = "80 km/jam";
83     }
84     return $sts;
85 }
86
87     public function traintest($data, $numEpoch) {
           $NumPattern = count($data); //Baris atau
88 jumlah data
           $NumInput = count($data[0]) - 1; //Kolom
89 atau jumlah atribut
90
91     /* Proses Training */
92     for ($sep = 0; $sep < $numEpoch; $sep++) {
93         $inputSource = $data[$sep %
$NumPattern];
94         $target = $data[$sep %
$NumPattern][$NumInput - 1];
95
96         // Menetapkan data ke lapisan input
97         for ($k = 0; $k < $this->jmlTitik[1];
$K++) {
98             for ($l = 0; $l < $this-
>jmlTitik[0]; $l++) {
99                 $this->keluaran[$k][$l] =
$inputSource[$l];
100            }
101        }
102
103        // Menetapkan nilai keluaran
104        for ($l = 1; $l < 2; $l++) {
105            for ($i = 0; $i < $this-
>jmlTitik[$l]; $i++) {
106                $sum = 0.0;
107                for ($j = 0; $j < $this-
>jmlTitik[$l - 1] - 1; $j++) {
108                    $sum+=$this-
>keluaran[$i][$j] * $this->titik[$l][$i][$j]; //
Menerapkan Penjumlahan dari perkalian antara
nilai masukan dengan bobt
109                }
110                $sum+=$this-
>titik[$l][$i][$this->jmlTitik[$l - 1] - 1];
//Menerapkan bias
111                // Menerapkan fungsi sigmoid
112                $this->keluaran[$l][$i] = 1.0
/ (1.0 + exp(-$sum));
113            }

```

```

114         }
115
116         for ($l = 0; $l < $this->jmlTitik[1];
117 $l++) {
118             $this->error[1][$l] = ($target -
119 $this->keluaran[1][$l]);
120         }
121         for ($l = 1; $l < 2; $l++) {
122             for ($i = 0; $i < $this-
123 >jmlTitik[$l]; $i++) {
124                 for ($j = 0; $j < $this-
125 >jmlTitik[$l - 1] - 1; $j++) {
126                     /* Perubahan Bobot-bobot
127 */
128                     $this-
129 >perubahan_bobot[$l][$i][$j] = $this-
130 >learning_rate * $this->error[$l][$i] * $this-
131 >keluaran[$i][$j];
132
133                     $this-
134 >titik[$l][$i][$j]+=$this-
135 >perubahan_bobot[$l][$i][$j];
136                 }
137             }
138             /* Perubahan Bias */
139             $this-
140 >perubahan_bobot[$l][$i][$this->jmlTitik[$l - 1]
141 - 1] = $this->learning_rate * $this-
142 >error[$l][$i];
143
144             $this->titik[$l][$i][$this-
145 >jmlTitik[$l - 1] - 1]+=$this-
146 >perubahan_bobot[$l][$i][$this->jmlTitik[$l - 1]
147 - 1];
148         }
149     }
150
151     /* Proses Testing */
152     $na = array();
153     $h_pred = array();
154     for ($m = 0; $m < $NumPattern; $m++)
155     {
156         $sum = 0.0;
157
158         for ($j = 0; $j < $this-
159 >jmlTitik[1]; $j++) {
160             for ($k = 0; $k < $this-
161 >jmlTitik[0]; $k++) {
162                 $this->keluaran[$j][$k] =
163 $data[$m][$k];
164             }
165         }
166
167         for ($l = 1; $l < 2; $l++) {
168             for ($i = 0; $i < $this-

```

```

>jmlTitik[$l]; $i++) {
147     $sum = 0.0;
148     for ($j = 0; $j < $this-
>jmlTitik[$l - 1] - 1; $j++) {
149         $sum+=$this-
>keluaran[$i][$j] * $this->titik[$l][$i][$j];
150     }
151     $sum+=$this-
>titik[$l][$i][$this->jmlTitik[$l - 1] - 1];
152     $this->keluaran[$l][$i] =
1.0 / (1.0 + exp(-$sum));
153     }
154     }
155
156     $result = (double) $this-
>keluaran[1][0];
157     $vmaks = $this-
>result_conv($data[$m][$NumInput - 1]);
158     $prediksi = $this-
>result_conv($result);
159     if ($vmaks != $prediksi) {
160         $na[$m] = $prediksi /
$NumPattern;
161         $h_pred[$m] = "<a
style='background: yellow;'>Data ke-" . ($m + 1)
. ". " . $vmaks . " || Hasil: " . $result . " -
Hasil Prediksi: " . $prediksi . "</a>";
162     } else
163         $h_pred[$m] = "Data ke-" .
($m + 1) . ". " . $vmaks . " || Hasil: " .
$result . " - Hasil Prediksi: " . $prediksi;
164     }
165     $keliru = count($na);
166     $threshold = 0;
167     $akurasi = (($NumPattern - $keliru) /
$NumPattern) * 100;
168     $epohAkhir = 0;
169     if ($keliru == $threshold) {
170         $numEpoch = $ep + 1;
171         $epohAkhir = $numEpoch;
172         break;
173     } else {
174         $epohAkhir = $numEpoch;
175     }
176     }
177     $this->ket = array($keliru, $akurasi,
$epohAkhir);
178     return $h_pred;
179     }
180
181     public function hasil_bobot() {
182         $h_bo = array();
183         for ($l = 1; $l < 2; $l++) {

```

```

184         for ($i = 0; $i < $this-
>jmlTitik[$l]; $i++) {
185             for ($j = 0; $j < $this-
>jmlTitik[$l - 1] - 1; $j++) {
186                 $h_bo[$i][$j] = "Bobot ke-" .
$i . "," . $j . ": " . $this->titik[$l][$i][$j];
187             }
188         }
189     }
190     return $h_bo;
191 }
192
193     public function hasil_bias() {
194         $h_bi = array();
195         for ($l = 1; $l < 2; $l++) {
196             for ($j = 0; $j < $this-
>jmlTitik[$l]; $j++) {
197                 $h_bi[$j] = "Bias ke-" . $j . ":
" . $this->titik[$l][$j][$this->jmlTitik[$l - 1]
- 1];
198             }
199         }
200         return $h_bi;
201     }
202
203     public function keterangan() {
204         return "Jumlah data prediksi yang salah:
" . $this->ket[0] . "&#13;&#10Nilai akurasi: " .
$this->ket[1] . "% &#13;&#10Epoh berakhir pada: "
. $this->ket[2] . " pengulangan";
205     }
206
207     public function k_lintas($testData) {
208         $titik = json_decode($this->m_perceptron-
>get_bobotbias()[0]['bb']);
209         $NumPattern = count($testData);
210         $h_pred = array();
211
212         // for ($l = 1; $l < 2; $l++) {
213             // for ($i = 0; $i < 1; $i++) {
214                 // for ($j = 0; $j <
count($titik) - 1; $j++) {
215                     // $titik[$j] = 0;
216                 // }
217                 // $titik[count($titik) - 1] = 0;
218             // }
219         // }
220     // }
221
222     for ($m = 0; $m < $NumPattern; $m++) {
223         $sum = 0.0;
224         for ($k = 0; $k < count($titik);
$k++) {
225             $this->keluaran[0][$k] =

```

```

226     $testData[$m][$k];
227     }
228     for ($l = 1; $l < 2; $l++) {
229         for ($i = 0; $i < 1; $i++) {
230             $sum = 0.0;
231             for ($j = 0; $j <
(count($titik) - 1); $j++) {
232                 $sum+=$this->keluaran[$l
- 1][$j] * $titik[$j];
233             }
234             $sum+=$titik[count($titik) -
1];
235             $this->keluaran[$l][$i] = 1.0
/ (1.0 + exp(-$sum));
236         }
237     }
238     $result = (double) $this-
>keluaran[1][0];
239     $h_pred[$m] = $this-
>result_conv($result);
240     }
241     return $h_pred;
242 }
243 public function proses() {
244     $nipp = $this->session-
>userdata('uname');
245     $ambil_akun = $this->m_login-
>user($nipp);
246     $data = array(
247         'user' => $ambil_akun
248     );
249
250     $jml_data = $this->m_perceptron-
>jml_kolom()[0]['jml'];
251     $dataTr = $this->m_perceptron-
>datatrain();
252     $learning_rate = $this->input-
>post('lr'); // learning rate
253     $max_epoh = $this->input->post('e');
// maksimal iterasi
254     $jmlTitik = array($jml_data, 1);
255
256     $this->awal($jmlTitik, $learning_rate);
257     $hasil = $this->traintest($dataTr,
$max_epoh); // melakukan proses klasifikasi
dengan data uji
258
259     $na_db = $this->model_sikemaka-
>get_hitungakh($this->session-
>userdata('uname'))[0]['na'];
260     if ($this->ket[1] < $na_db) {

```



```
261     $na_kurang = "Nilai akurasi yang
didapat lebih kecil daripada sebelumnya. Silakan
atur kembali parameter SLP anda.";
262     } else {
263         $na_kurang = "";
264     }
265
266     $this->model_sikemaka-
>tambah_hitungakh(round($this->ket[1], 2),
$max_epoh, $learning_rate, $nipp,
json_encode($this->titik[1][0]));
267
268     $res = array(
269         'epoh' => $max_epoh,
270         'lr' => $learning_rate,
271         'hasil' => '',
272         'jmlData' => count($dataTr),
273         'titik' => ($jmlTitik[0] - 1),
274         'keluaran' => $jmlTitik[1],
275         'bobot' => $this->hasil_bobot(),
276         'bias' => $this->hasil_bias(),
277         'hasilp' => $hasil,
278         'ket' => $this->keterangan(),
279         'a' => $na_kurang
280     );
281
282     $stat = $this->session->userdata('lvl');
283     if ($stat == 3) {
284         $this->load->view('ahlijj/header',
$data);
285         $this->load-
>view('ahlijj/perceptron', $res);
286         $this->load->view('footer');
287     } else {
288         redirect('c_login', 'refresh');
289     }
290 }
291
292 public function klas() {
293     $lin = $this->input->post('jalur');
294     $tr = $this->input->post('trr');
295     $jb = $this->input->post('jbr');
296     $jp = $this->input->post('jpr');
297     $r60 = 0;
298     $r54 = 0;
299     $r50 = 0;
300     $r42 = 0;
301     $beton = 0;
302     $kayu = 0;
303     $baja = 0;
304     $tunggal = 0;
305     $ganda = 0;
306     $tba = $this->input->post('tbar');
```

```
307 $lbb = $this->input->post('lbbr');
308 $landai = $this->input->post('landai');
309 $jari2 = $this->input->post('jari2');
310 $jj = $this->input->post('jj');
311 $rminv = 0;
312 $rmintph = 0;
313 $rminph = 0;
314 $lbj = $this->input->post('lbj');
315 $rb = $this->input->post('rb');
316 $j = 0;
317 $v = 0;
318 $jv = $this->input->post('jv');
319 $idjalur = $this->input->post('idjalur');
320
321 if ($tr == 'R.60') {
322     $r60 = 1;
323     $r54 = 0;
324     $r50 = 0;
325     $r42 = 0;
326 } else if ($tr == 'R.54') {
327     $r60 = 0;
328     $r54 = 1;
329     $r50 = 0;
330     $r42 = 0;
331 } else if ($tr == 'R.50') {
332     $r60 = 0;
333     $r54 = 0;
334     $r50 = 1;
335     $r42 = 0;
336 } else if ($tr == 'R.42') {
337     $r60 = 0;
338     $r54 = 0;
339     $r50 = 0;
340     $r42 = 1;
341 }
342
343 if ($jb == 'Beton') {
344     $beton = 1;
345     $kayu = 0;
346     $baja = 0;
347 } else if ($jb == 'Kayu') {
348     $beton = 0;
349     $kayu = 1;
350     $baja = 0;
351 } else if ($jb == 'Baja') {
352     $beton = 0;
353     $kayu = 0;
354     $baja = 1;
355 }
356
357 if ($jp == 'Tunggal') {
358     $tunggal = 1;
359     $ganda = 0;
```

```
360     } else if ($jp == 'Ganda') {
361         $tunggal = 0;
362         $ganda = 1;
363     }
364
365     if ($jj == 'v') {
366         $rminv = $jari2;
367         $rmintph = 0;
368         $rminph = 0;
369     } else if ($jj == 'tph') {
370         $rminv = 0;
371         $rmintph = $jari2;
372         $rminph = 0;
373     } else if ($jj == 'ph') {
374         $rminv = 0;
375         $rmintph = 0;
376         $rminph = $jari2;
377     } else if ($jj == 'l') {
378         $rminv = 0;
379         $rmintph = 0;
380         $rminph = 0;
381     }
382
383     if ($jv == 'j') {
384         $j = $rb;
385         $v = 0;
386     } else if ($jv == 'v') {
387         $j = 0;
388         $v = $rb;
389     } else if ($jv == 'b') {
390         $j = 0;
391         $v = 0;
392     }
393
394     if ($rb != 0 && $rb <= 4.5 && $jv = 'j')
395     {
396         $vmaks = '60 km/jam';
397     } else if ($rb != 0 && $rb <= 4.7 && $jv
398 = 'v') {
399         $vmaks = '60 km/jam';
400     } else if ($rmintph > 810 && $rmintph <=
401 1050) {
402         $vmaks = '70 km/jam';
403     } else if ($rminph > 270 && $rminph <=
404 350) {
405         $vmaks = '70 km/jam';
406     } else if ($rmintph >= 600 && $rmintph <=
407 810) {
408         $vmaks = '60 km/jam';
409     } else if ($rminph >= 200 && $rminph <=
410 270) {
411         $vmaks = '60 km/jam';
412     } else {
```

```
407         $n_tba = ($tba - $this->m_perceptron-
>data_n()[0]['tba']) / ($this->m_perceptron-
>data_n()[0]['tbax'] - $this->m_perceptron-
>data_n()[0]['tba']);
408         $n_lbb = ($lbb - $this->m_perceptron-
>data_n()[0]['lbb']) / ($this->m_perceptron-
>data_n()[0]['lbbx'] - $this->m_perceptron-
>data_n()[0]['lbb']);
409         $n_landai = ($landai - $this-
>m_perceptron->data_n()[0]['landai']) / ($this-
>m_perceptron->data_n()[0]['landaix'] - $this-
>m_perceptron->data_n()[0]['landai']);
410         $n_rminv = ($rminv - $this-
>m_perceptron->data_n()[0]['rminv']) / ($this-
>m_perceptron->data_n()[0]['rminvx'] - $this-
>m_perceptron->data_n()[0]['rminv']);
411         $n_rmintph = ($rmintph - $this-
>m_perceptron->data_n()[0]['rmintph']) / ($this-
>m_perceptron->data_n()[0]['rmintphx'] - $this-
>m_perceptron->data_n()[0]['rmintph']);
412         $n_rminph = ($rminph - $this-
>m_perceptron->data_n()[0]['rminph']) / ($this-
>m_perceptron->data_n()[0]['rminphx'] - $this-
>m_perceptron->data_n()[0]['rminph']);
413         $n_lbj = ($lbj - $this->m_perceptron-
>data_n()[0]['lbj']) / ($this->m_perceptron-
>data_n()[0]['lbjx'] - $this->m_perceptron-
>data_n()[0]['lbj']);
414         $data_klas = array(array($r60, $r54,
$r50, $r42, $beton, $kayu, $baja, $ganda,
$tunggal,
415         $n_tba, $n_lbb, $n_landai,
$n_rminv, $n_rmintph, $n_rminph, $n_lbj, 0));
416         $vmaks = $this-
>k_lintas($data_klas)[0];
417     }
418
419
420     $data = array(
421         'r60' => $r60,
422         'r54' => $r54,
423         'r50' => $r50,
424         'r42' => $r42,
425         'beton' => $beton,
426         'kayu' => $kayu,
427         'baja' => $baja,
428         'ganda' => $ganda,
429         'tunggal' => $tunggal,
430         'tba' => $tba,
431         'lbb' => $lbb,
432         'landai' => $landai,
433         'rminv' => $rminv,
434         'rmintph' => $rmintph,
```

```

435         'rminph' => $rminph,
436         'lbj' => $lbj,
437         'vmaks' => $vmaks,
438         'idjalur' => $idjalur,
439         'jembatan' => $j,
440         'waktu' => 'now()',
441         'viaduk' => $v
442     );
443
444     $this->m_perceptron->klasidata($data);
445     $this->m_perceptron->vmaks($vmaks,
446     $idjalur);
447     redirect('c_rancang/rancang/' . $lin);
448 }
449 }
450
451 ?>

```

5. Kode Program c_cetak.php

```

1  <?php
2
3  class C_cetak extends CI_Controller {
4
5      public function __construct() {
6          parent::__construct();
7          $this->load->database();
8          $this->load->model('m_login');
9          $this->load->model('model_sikemaka');
10     }
11
12     public function laporan() {
13         $id = $this->uri->segment(3);
14         $w = $this->session->userdata('wil');
15
16         $data['v'] = $this->model_sikemaka-
17         >tgl($id);
18         $data['user'] = $this->model_sikemaka-
19         >userc($w);
20         $data['st'] = $this->model_sikemaka-
21         >get_stasiun($id);
22         $data['j'] = $this->model_sikemaka-
23         >get_jalur($id);
24         $data['lin'] = $this->model_sikemaka-
25         >get_lintas($id, $w);
26         $data['koor'] = $this->model_sikemaka-
27         >get_koor($id);
28
29         $stat = $this->session->userdata('lvl');
30         if ($stat == 2) {
31             $this->load->view('pegawai/cetak',
32             $data);
33         }
34     }
35 }

```

```
26         } else {
27             redirect('c_login', 'refresh');
28         }
29     }
30
31 }
32
33 ?>
```

6. Kode Program c_dataset.php

```
1 <?php
2
3 class C_dataset extends CI_Controller {
4
5     public function __construct() {
6         parent::__construct();
7         $this->load->database();
8         $this->load->library('pagination');
9         $this->load->model('m_login');
10        $this->load->model('model_sikemaka');
11    }
12
13    public function index() {
14        $config['base_url'] =
15        site_url('c_dataset/index');
16        $config['total_rows'] = $this-
17        >model_sikemaka->jmld()['data'];
18        $config['per_page'] = "30";
19        $config["uri_segment"] = 3;
20        $choice = $config["total_rows"] /
21        $config["per_page"];
22        $config["num_links"] = floor($choice);
23
24        $config['full_tag_open'] = '<ul
25        class="pagination">';
26        $config['full_tag_close'] = '</ul>';
27        $config['first_link'] = false;
28        $config['last_link'] = false;
29        $config['first_tag_open'] = '<li>';
30        $config['first_tag_close'] = '</li>';
31        $config['prev_link'] = '<';
32        $config['prev_tag_open'] = '<li
33        class="prev">';
34        $config['prev_tag_close'] = '</li>';
35        $config['next_link'] = '>';
36        $config['next_tag_open'] = '<li>';
37        $config['next_tag_close'] = '</li>';
38        $config['last_tag_open'] = '<li>';
39        $config['last_tag_close'] = '</li>';
40        $config['cur_tag_open'] = '<li
41        class="active"><a href="#">';
42        $config['cur_tag_close'] = '</a></li>';
```



```
37     $config['num_tag_open'] = '<li>';
38     $config['num_tag_close'] = '</li>';
39     $this->pagination->initialize($config);
40
41     $data2['page'] = ($this->uri->segment(3))
? $this->uri->segment(3) : 0;
42
43     $data2['d'] = $this->model_sikemaka-
>get_dataset($config["per_page"],
$data2['page']);
44
45     $data2['pagination'] = $this->pagination-
>create_links();
46
47 //=====
=====
48
49     $ambil_akun = $this->m_login->user($this-
>session->userdata('uname'));
50     $data = array(
51         'user' => $ambil_akun
52     );
53     $stat = $this->session->userdata('lvl');
54     if ($stat == 3) {
55         $this->load->view('ahlijj/header',
$data);
56         $this->load->view('ahlijj/dataset',
$data2);
57         $this->load->view('footer');
58     } else {
59         redirect('c_login', 'refresh');
60     }
61 }
62
63 public function data() {
64     $ambil_akun = $this->m_login->user($this-
>session->userdata('uname'));
65     $data = array(
66         'user' => $ambil_akun
67     );
68
69     $data2['d'] = $this->model_sikemaka-
>get_datasetfull();
70     $data2['pagination'] = '';
71
72     if ($this->session->userdata('lvl') == 3)
{
73         $this->load->view('ahlijj/header',
$data);
74         $this->load->view('ahlijj/dataset',
$data2);
75         $this->load->view('footer');
76     } else {
```

```
77         redirect('c_login', 'refresh');
78     }
79 }
80
81     public function tambah() {
82         $ambil_akun = $this->m_login->user($this->
>session->userdata('uname'));
83         $data = array(
84             'user' => $ambil_akun
85         );
86
87         if ($this->session->userdata('lvl') == 3)
88     {
89         $this->load->view('ahlijj/header',
90 $data);
91         $this->load-
92 >view('ahlijj/tambahdataset');
93         $this->load->view('footer');
94     } else {
95         redirect('c_login', 'refresh');
96     }
97 }
98
99     public function tambahd() {
100         $tr = $this->input->post('tr');
101         $jb = $this->input->post('jb');
102         $jp = $this->input->post('jp');
103         $r60 = 0;
104         $r54 = 0;
105         $r50 = 0;
106         $r42 = 0;
107         $beton = 0;
108         $kayu = 0;
109         $baja = 0;
110         $tunggal = 0;
111         $ganda = 0;
112
113         if ($tr == 'r60') {
114             $r60 = 1;
115             $r54 = 0;
116             $r50 = 0;
117             $r42 = 0;
118         } else if ($tr == 'r54') {
119             $r60 = 0;
120             $r54 = 1;
121             $r50 = 0;
122             $r42 = 0;
123         } else if ($tr == 'r50') {
124             $r60 = 0;
125             $r54 = 0;
126             $r50 = 1;
127             $r42 = 0;
128         } else if ($tr == 'r42') {
```

```
126         $r60 = 0;
127         $r54 = 0;
128         $r50 = 0;
129         $r42 = 1;
130     }
131
132     if ($jb == 'beton') {
133         $beton = 1;
134         $kayu = 0;
135         $baja = 0;
136     } else if ($jb == 'kayu') {
137         $beton = 0;
138         $kayu = 1;
139         $baja = 0;
140     } else if ($jb == 'baja') {
141         $beton = 0;
142         $kayu = 0;
143         $baja = 1;
144     }
145
146     if ($jp == 'tunggal') {
147         $tunggal = 1;
148         $ganda = 0;
149     } else if ($jp == 'ganda') {
150         $tunggal = 0;
151         $ganda = 1;
152     }
153
154     $data = array(
155         'r60' => $r60,
156         'r54' => $r54,
157         'r50' => $r50,
158         'r42' => $r42,
159         'beton' => $beton,
160         'kayu' => $kayu,
161         'baja' => $baja,
162         'tunggal' => $tunggal,
163         'ganda' => $ganda,
164         'tba' => $this->input->post('tba'),
165         'lbb' => $this->input->post('lbb'),
166         'landai' => $this->input-
>post('landai'),
167         'rminv' => $this->input-
>post('rminv'),
168         'rmintph' => $this->input-
>post('rmintph'),
169         'rminph' => $this->input-
>post('rminph'),
170         'lbj' => $this->input->post('lbj'),
171         'vmaks' => $this->input-
>post('vmaks'),
172         'hapus' => 0
173     );
```

```
174
175     $this->model_sikemaka->tambahd($data);
176     redirect('c_dataset');
177 }
178
179     public function ubah() {
180         $ambil_akun = $this->m_login->user($this-
>session->userdata('uname'));
181         $data = array(
182             'user' => $ambil_akun
183         );
184
185         $id = $this->uri->segment(3);
186         $data2['ubah'] = $this->model_sikemaka-
>get_datasetid($id);
187
188         if ($this->session->userdata('lvl') == 3)
{
189             $this->load->view('ahlijj/header',
$data);
190             $this->load-
>view('ahlijj/editdataset', $data2);
191             $this->load->view('footer');
192         } else {
193             redirect('c_login', 'refresh');
194         }
195     }
196
197     public function ubahd() {
198         $tr = $this->input->post('tr');
199         $jb = $this->input->post('jb');
200         $jp = $this->input->post('jp');
201         $r60 = 0;
202         $r54 = 0;
203         $r50 = 0;
204         $r42 = 0;
205         $beton = 0;
206         $kayu = 0;
207         $baja = 0;
208         $tunggal = 0;
209         $ganda = 0;
210
211         if ($tr == 'r60') {
212             $r60 = 1;
213             $r54 = 0;
214             $r50 = 0;
215             $r42 = 0;
216         } else if ($tr == 'r54') {
217             $r60 = 0;
218             $r54 = 1;
219             $r50 = 0;
220             $r42 = 0;
221         } else if ($tr == 'r50') {
```

```
222         $r60 = 0;
223         $r54 = 0;
224         $r50 = 1;
225         $r42 = 0;
226     } else if ($str == 'r42') {
227         $r60 = 0;
228         $r54 = 0;
229         $r50 = 0;
230         $r42 = 1;
231     }
232
233     if ($jtb == 'beton') {
234         $beton = 1;
235         $kayu = 0;
236         $baja = 0;
237     } else if ($jtb == 'kayu') {
238         $beton = 0;
239         $kayu = 1;
240         $baja = 0;
241     } else if ($jtb == 'baja') {
242         $beton = 0;
243         $kayu = 0;
244         $baja = 1;
245     }
246
247     if ($jtp == 'tunggal') {
248         $tunggal = 1;
249         $ganda = 0;
250     } else if ($jtp == 'ganda') {
251         $tunggal = 0;
252         $ganda = 1;
253     }
254
255     $data = array(
256         'id' => $this->input->post('id'),
257         'r60' => $r60,
258         'r54' => $r54,
259         'r50' => $r50,
260         'r42' => $r42,
261         'beton' => $beton,
262         'kayu' => $kayu,
263         'baja' => $baja,
264         'tunggal' => $tunggal,
265         'ganda' => $ganda,
266         'tba' => $this->input->post('tba'),
267         'lbb' => $this->input->post('lbb'),
268         'landai' => $this->input-
>post('landai'),
269         'rminv' => $this->input-
>post('rminv'),
270         'rmintph' => $this->input-
>post('rmintph'),
271         'rminph' => $this->input-
```

```
272 >post('rminph'),
      'lbj' => $this->input->post('lbj'),
      'vmaks' => $this->input-
273 >post('vmaks'),
274      'hapus' => 0
275     );
276
277     $this->model_sikemaka->ubahd($data);
278     redirect('c_dataset');
279 }
280
281 public function hapus() {
282     $s = $this->uri->segment(3);
283     $this->model_sikemaka->hapus($s);
284     redirect('c_dataset', 'refresh');
285 }
286
287 public function cari() {
288     $ambil_akun = $this->m_login->user($this-
>session->userdata('uname'));
289     $data = array(
290         'user' => $ambil_akun
291     );
292
293     $pilihan = $this->input->post('pilihan');
294     $search = $this->input->post("cr");
295
296     $search = ($this->uri->segment(3)) ?
$this->uri->segment(3) : $search;
297     $pilihan = ($this->uri->segment(4)) ?
$this->uri->segment(4) : $pilihan;
298     $c = strtolower($search);
299     if ($pilihan == 'vmaks') {
300         if ($c == 120 || $c == '120 ' || $c
== '120 km' || $c == '120 km/' || $c == '120
km/jam') {
301             $c = 1;
302         } else if ($c == 110 || $c == '110 '
|| $c == '110 km' || $c == '110 km/' || $c ==
'110 km/jam') {
303             $c = 0.75;
304         } else if ($c == 100 || $c == '100 '
|| $c == '100 km' || $c == '100 km/' || $c ==
'100 km/jam') {
305             $c = 0.5;
306         } else if ($c == 90 || $c == '90 ' ||
$c == '90 km' || $c == '90 km/' || $c == '90
km/jam') {
307             $c = 0.25;
308         } if ($c == 80 || $c == '80 ' || $c
== '80 km' || $c == '80 km/' || $c == '80
km/jam') {
309             $c = 0;
```



```
310     }
311   }
312
313   $config = array();
314   $config['base_url'] =
site_url("c_dataset/cari/$search/$pilihan");
315   $config['total_rows'] = $this-
>model_sikemaka->cari($pilihan, $c);
316   $config['per_page'] = "10";
317   $config["uri_segment"] = 5;
318   $choice = $config["total_rows"] /
$config["per_page"];
319   $config["num_links"] = floor($choice);
320
321   $config['full_tag_open'] = '<ul
class="pagination">';
322   $config['full_tag_close'] = '</ul>';
323   $config['first_link'] = false;
324   $config['last_link'] = false;
325   $config['first_tag_open'] = '<li>';
326   $config['first_tag_close'] = '</li>';
327   $config['prev_link'] = '<';
328   $config['prev_tag_open'] = '<li
class="prev">';
329   $config['prev_tag_close'] = '</li>';
330   $config['next_link'] = '>';
331   $config['next_tag_open'] = '<li>';
332   $config['next_tag_close'] = '</li>';
333   $config['last_tag_open'] = '<li>';
334   $config['last_tag_close'] = '</li>';
335   $config['cur_tag_open'] = '<li
class="active"><a href="#">';
336   $config['cur_tag_close'] = '</a></li>';
337   $config['num_tag_open'] = '<li>';
338   $config['num_tag_close'] = '</li>';
339   $this->pagination->initialize($config);
340
341   $data4['page'] = ($this->uri->segment(5))
? $this->uri->segment(5) : 0;
342
343   $data4['d'] = $this->model_sikemaka-
>carid($config['per_page'], $data4['page'],
$pilihan, $c);
344
345   $data4['pagination'] = $this->pagination-
>create_links();
346
347   if ($this->session->userdata('lvl') == 3)
348   {
349     $this->load->view('ahlijj/header',
$data);
350     $this->load->view('ahlijj/dataset',
$data4);
```

```
350         $this->load->view('footer');
351     } else {
352         redirect('c_login', 'refresh');
353     }
354 }
355 }
356 }
357 }
358 ?>
```

7. Kode Program c_login.php

```
1  <?php
2
3  class C_login extends CI_Controller {
4
5      public function __construct() {
6          parent::__construct();
7          $this->load->model('m_login');
8      }
9
10     public function index() {
11         $session = $this->session-
>userdata('isLoggedIn');
12         if ($session == FALSE) {
13             $this->load->view('login');
14         } else {
15             redirect('c_utama');
16         }
17     }
18
19     public function cek_login() {
20         $nipp = $this->input->post('username');
21         $ks = $this->input->post('password');
22         $data['e1'] = 'NIPP atau kata sandi belum
terisi.';
23         $data2 = array(
24             'e2' => 'Kombinasi NIPP dengan kata
sandinya salah.'
25         );
26
27         $nalar = $this->m_login->login($nipp,
$ks);
28         if (count($nalar) == 1) {
29             foreach ($nalar as $nalar) {
30                 $kd = $nalar['KodeJabatan'];
31                 $w = $nalar['WilID'];
32             }
33             $this->session->set_userdata(array(
34                 'isLoggedIn' => TRUE,
35                 'uname' => $nipp,
36                 'lvl' => $kd,
37                 'wil' => $w
```

```
38         });
39
40     redirect('c_utama', 'refresh');
41     } else if ($nipp == '' || $ks == '') {
42         $this->load->view('login', $data);
43     } else if ($nipp != '' || $ks != '' ||
44 count($nalar) != 1) {
45         $this->load->view('login', $data2);
46     }
47
48     public function logout() {
49         $this->session->sess_destroy();
50         redirect('c_login', 'refresh');
51     }
52
53 }
54
55 ?>
```

8. Kode Program c_peta.php

```
1 <?php
2
3 class C_peta extends CI_Controller {
4
5     public function __construct() {
6         parent::__construct();
7         $this->load->model('m_login');
8         $this->load->model('model_sikemaka');
9     }
10
11     public function validasi() {
12         $s = $this->uri->segment(3);
13         $this->model_sikemaka->validasi($s);
14         redirect('c_utama', 'refresh');
15     }
16
17     public function tambah() {
18         $s = $this->uri->segment(3);
19         $this->model_sikemaka->tambah($s);
20         redirect('c_utama', 'refresh');
21     }
22
23     public function ubah() {
24         $s = $this->uri->segment(3);
25         $this->model_sikemaka->ubah($s);
26         redirect('c_utama', 'refresh');
27     }
28
29     public function lintas() {
30         $ambil_akun = $this->m_login->user($this->
31 session->userdata('uname'));
```

```

31
32     $sid = $this->uri->segment(3);
33     $w = $this->session->userdata('wil');
34
35     $data = array('user' => $ambil_akun);
36     $data2['st'] = $this->model_sikemaka-
>get_stasiun($id);
37     $data2['j'] = $this->model_sikemaka-
>get_jalur($id);
38     $data2['lin'] = $this->model_sikemaka-
>get_lintas($id, $w);
39     $data2['koor'] = $this->model_sikemaka-
>get_koor($id);
40
41     $stat = $this->session->userdata('lvl');
42     if ($stat == 1) {
43         $this->load->view('manjj/header',
$data);
44         $this->load->view('manjj/peta',
$data2);
45         $this->load->view('footer');
46     } else {
47         redirect('c_login', 'refresh');
48     }
49 }
50
51 }
52
53 ?>

```

9. Kode Program c_rancang.php

```

1 <?php
2
3 class C_rancang extends CI_Controller {
4
5     public function __construct() {
6         parent::__construct();
7         $this->load->model('m_login');
8         $this->load->model('model_sikemaka');
9     }
10
11     public function jalur() {
12         $ambil_akun = $this->m_login->user($this-
>session->userdata('uname'));
13         $s = $this->uri->segment(3);
14         $w = $this->session->userdata('wil');
15         $data = array('user' => $ambil_akun);
16         $data2['lin'] = $this->model_sikemaka-
>get_lintas($s, $w);
17         $data2['s'] = $s;
18         $stat = $this->session->userdata('lvl');
19

```

```
20         if ($stat == 2) {
21             $this->load->view('pegawai/header',
22 $data);
23             $this->load->view('pegawai/fklintas',
24 $data2);
25             $this->load->view('footer');
26         } else {
27             redirect('c_login', 'refresh');
28         }
29     }
30     public function postdata() {
31         $s = $this->uri->segment(3);
32         $p = $this->input->post('proses');
33         $this->model_sikemaka->proses($s, $p);
34         $fk = array(
35             'tr' => $this->input->post('tr'),
36             'jb' => $this->input->post('jb'),
37             'jp' => $this->input->post('jp'),
38             'tba' => $this->input->post('tba'),
39             'lbb' => $this->input->post('lbb'),
40             'idlintas' => $s
41         );
42         $this->model_sikemaka->fk($fk);
43         $stat = $this->session->userdata('lvl');
44         if ($stat == 2) {
45             redirect('c_rancang/rancang/' . $s,
46 'refresh');
47         } else {
48             redirect('c_login', 'refresh');
49         }
50     }
51     public function rancang() {
52         $ambil_akun = $this->m_login->user($this->
53 session->userdata('uname'));
54         $s = $this->uri->segment(3);
55         $w = $this->session->userdata('wil');
56         $data = array('user' => $ambil_akun);
57         $lintas = array(
58             'fk' => $this->model_sikemaka-
59 >get_fk($s),
60             'st' => $this->model_sikemaka-
61 >get_stasiun($s),
62             'j' => $this->model_sikemaka-
63 >get_jalur($s),
64             'lin' => $this->model_sikemaka-
65 >get_lintas($s, $w),
66             'koor' => $this->model_sikemaka-
67 >get_koor($s),
68             's' => $s
69         );
```

```

64     );
65     $stat = $this->session->userdata('lvl');
66
67     if ($stat == 2) {
68         $this->load->view('pegawai/header',
69 $data);
69         $this->load->view('pegawai/rancang',
70 $lintas);
70         $this->load->view('footer');
71     } else {
72         redirect('c_login', 'refresh');
73     }
74 }
75
76 public function selesai() {
77     $s = $this->uri->segment(3);
78     $m = $this->uri->segment(4);
79     $this->model_sikemaka->selesai($s, $m);
80     $stat = $this->session->userdata('lvl');
81
82     if ($stat == 2) {
83         redirect('c_utama', 'refresh');
84     } else {
85         redirect('c_login', 'refresh');
86     }
87 }
88
89 }
90
91 ?>

```

10. Kode Program c_utama.php

```

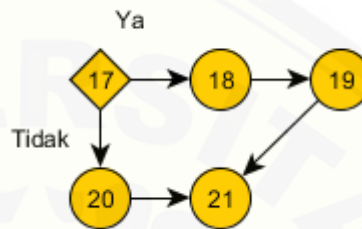
1  <?php
2
3  class C_utama extends CI_Controller {
4
5      public function __construct() {
6          parent::__construct();
7          $this->load->model('m_login');
8          $this->load->model('model_sikemaka');
9      }
10
11     public function index() {
12         $ambil_akun = $this->m_login->user($this->
13 >session->userdata('uname'));
13         $ambil_data = $this->model_sikemaka-
14 >get_hitungakh($this->session-
15 >userdata('uname'));
14         $ambil_data2 = $this->model_sikemaka-
15 >jmld();
15         $ambil_data3 = $this->model_sikemaka-
16 >datapeta($this->session->userdata('wil'))-

```


LAMPIRAN D. PENGUJIAN SISTEM

D.1. Pengujian White Box

1. Berkas *m_login.php*
 1. Daftar *Method*
Method-method lain yang digunakan pada alur tersebut sebagai berikut: *login(\$nipp, \$ks)*
 2. Diagram Alir *Method*



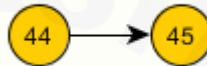
Gambar D.0.1 Diagram Alir *login(\$nipp, \$ks)*

3. Penghitungan *Cyclomatic Complexity* (CC)
login(\$nipp, \$ks) : $V(G) = 5 - 5 + 2 = 2$
4. Basis Set
login(\$nipp, \$ks) : 17-18-19-21 (Alur 1)
17-20-21 (Alur 2)
5. Kasus Uji atau *Test Case*

Alur 1	
<i>Test Case</i>	<i>login(\$nipp, \$ks)</i>
Target yang Diharapkan	Mendapatkan NIPP dan kata sandi dari tabel pegawai pada basis data sesuai dengan kedua parameter tersebut apabila tidak bernilai kosong
Hasil Pengujian	Benar
Alur / Path	17-18-19-21
Alur 2	
<i>Test Case</i>	<i>login(\$nipp, \$ks)</i>
Target yang Diharapkan	Proses permisalan tersebut selesai
Hasil Pengujian	Benar
Alur / Path	17-20-21

2. Berkas *m_perceptron.php*1. Daftar *Method*

Method-method lain yang digunakan pada alur tersebut sebagai berikut: *data_n()*, *get_bobotbias()*, *klasidata(\$data)*, dan *vmaks(\$v, \$id)*

2. Diagram Alir *Method*Gambar D.0.2 Diagram Alir *data_n()*Gambar D.0.3 Diagram Alir *get_bobotbias()*Gambar D.0.4 Diagram Alir *klasidata(\$data)*Gambar D.0.5 Diagram Alir *vmaks(\$v, \$id)*3. Penghitungan *Cyclomatic Complexity (CC)*

data_n() : $V(G) = 1 - 2 + 2 = 1$

get_bobotbias() : $V(G) = 1 - 2 + 2 = 1$

klasidata(\$data) : $V(G) = 0 - 1 + 2 = 1$

vmaks(\$v, \$id) : $V(G) = 0 - 1 + 2 = 1$

4. Basis Set

data_n() : 28-32 (Alur 1)

get_bobotbias() : 44-45 (Alur 1)

klasidata(\$data) : 36 (Alur 1)

vmaks(\$v, \$id) : 40 (Alur 1)

5. Kasus Uji atau *Test Case*

Alur 1	
<i>Test Case</i>	<i>data_n()</i>
Target yang Diharapkan	Mendapatkan nilai minimal dan nilai maksimal pada setiap atribut <i>dataset</i> dari basis data
Hasil Pengujian	Benar
Alur / Path	28-31
Alur 1	
<i>Test Case</i>	<i>get_bobotbias()</i>
Target yang Diharapkan	Mendapatkan larik bobot dan bias yang merupakan penghitungan terakhir yang bernilai akurasi tertinggi dalam bentuk JSON dari basis data
Hasil Pengujian	Benar
Alur / Path	44-45
Alur 1	
<i>Test Case</i>	<i>klasidata(\$data)</i>
Target yang Diharapkan	Menambahkan data <i>log</i> klasifikasi data baru ke tabel <i>logjalur</i> pada basis data
Hasil Pengujian	Benar
Alur / Path	36
Alur 1	
<i>Test Case</i>	<i>vmaks(\$v, \$id)</i>
Target yang Diharapkan	Mengubah nilai kecepatan maksimum suatu ruas lintas berdasarkan parameter kecepatan maksimum dan <i>id</i> jalur yang ditentukan
Hasil Pengujian	Benar
Alur / Path	40

3. Berkas *model_sikemaka.php*1. Daftar *Method*

Method-method lain yang digunakan pada alur tersebut sebagai berikut: *cari(\$p, \$c)*, *carid(\$batas, \$mulai, \$p, \$c)*, *datapeta(\$w)*, *fk(\$fk)*, *get_dataset(\$batas, \$mulai)*, *get_datasetfull()*, *get_datasetid(\$id)*, *get_fk(\$s)*, *get_jalur(\$i)*, *get_koor(\$i)*, *get_lintas(\$i, \$w)*, *get_stasiun(\$i)*, *hapus(\$data)*, *jmlid()*, *petamjfull(\$w)*, *proses(\$id, \$p)*, *selesai(\$id, \$s)*, *tambah(\$id)*, *tambahd(\$data)*, *tgl(\$id)*, *ubah(\$id)*, *ubahd(\$data)*, *userc(\$w)*, dan *validasi(\$id)*

2. Diagram Alir *Method*



Gambar D.0.6 Diagram Alir *cari(\$p, \$c)*



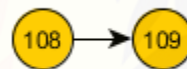
Gambar D.0.7 Diagram Alir *carid(\$batas, \$mulai, \$p, \$c)*



Gambar D.0.8 Diagram Alir *datapeta(\$w)*



Gambar D.0.9 Diagram Alir *fk(\$fk)*



Gambar D.0.10 Diagram Alir *get_dataset(\$batas, \$mulai)*



Gambar D.0.11 Diagram Alir *get_datasetfull()*



Gambar D.0.12 Diagram Alir *get_datasetid(\$id)*



Gambar D.0.13 Diagram Alir *get_fk(\$s)*



Gambar D.0.14 Diagram Alir *get_jalur(\$i)*



Gambar D.0.15 Diagram Alir *get_koor(\$i)*



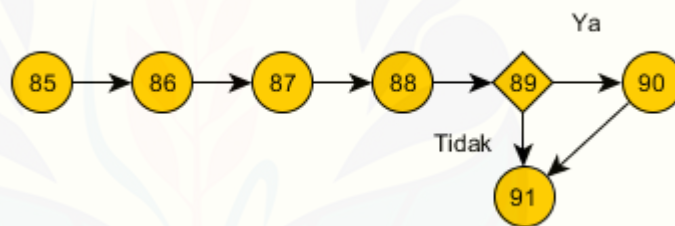
Gambar D.0.16 Diagram Alir *get_lintas(\$i, \$w)*



Gambar D.0.17 Diagram Alir *get_stasiun(\$i)*



Gambar D.0.18 Diagram Alir *hapus(\$data)*



Gambar D.0.19 Diagram Alir *jml()*



Gambar D.0.20 Diagram Alir *petamjfull(\$w)*



Gambar D.0.21 Diagram Alir *proses(\$id, \$p)*



Gambar D.0.22 Diagram Alir *selesai(\$id, \$s)*

Gambar D.0.23 Diagram Alir *tambah(\$id)*Gambar D.0.24 Diagram Alir *tambahd(\$data)*Gambar D.0.25 Diagram Alir *tgl(\$id)*Gambar D.0.26 Diagram Alir *ubah(\$id)*Gambar D.0.27 Diagram Alir *ubahd(\$data)*Gambar D.0.28 Diagram Alir *userc(\$w)*Gambar D.0.29 Diagram Alir *validasi(\$id)*

3. Penghitungan *Cyclomatic Complexity* (CC)

cari(\$p, \$c) : $V(G) = 1 - 2 + 2 = 1$

carid(\$batas, \$mulai, \$p, \$c) : $V(G) = 1 - 2 + 2 = 1$

datapeta(\$w) : $V(G) = 0 - 1 + 2 = 1$

fk(\$fk) : $V(G) = 0 - 1 + 2 = 1$

get_dataset(\$batas, \$mulai) : $V(G) = 1 - 2 + 2 = 1$

get_datasetfull() : $V(G) = 1 - 2 + 2 = 1$

<i>get_datasetid(\$id)</i>	: $V(G) = 1 - 2 + 2 = 1$
<i>get_fk(\$s)</i>	: $V(G) = 4 - 5 + 2 = 1$
<i>get_jalur(\$i)</i>	: $V(G) = 1 - 2 + 2 = 1$
<i>get_koor(\$i)</i>	: $V(G) = 1 - 2 + 2 = 1$
<i>get_lintas(\$i, \$w)</i>	: $V(G) = 1 - 2 + 2 = 1$
<i>get_stasiun(\$i)</i>	: $V(G) = 1 - 2 + 2 = 1$
<i>hapus(\$data)</i>	: $V(G) = 0 - 1 + 2 = 1$
<i>jmlid()</i>	: $V(G) = 7 - 7 + 2 = 2$
<i>petamjfull(\$w)</i>	: $V(G) = 1 - 2 + 2 = 1$
<i>proses(\$id, \$p)</i>	: $V(G) = 0 - 1 + 2 = 1$
<i>selesai(\$id, \$s)</i>	: $V(G) = 0 - 1 + 2 = 1$
<i>tambah (\$id)</i>	: $V(G) = 0 - 1 + 2 = 1$
<i>tambahd(\$data)</i>	: $V(G) = 0 - 1 + 2 = 1$
<i>tgl(\$id)</i>	: $V(G) = 1 - 2 + 2 = 1$
<i>ubah(\$id)</i>	: $V(G) = 0 - 1 + 2 = 1$
<i>ubahd(\$data)</i>	: $V(G) = 1 - 2 + 2 = 1$
<i>userc(\$w)</i>	: $V(G) = 6 - 7 + 2 = 1$
<i>validasi(\$id)</i>	: $V(G) = 0 - 1 + 2 = 1$

4. Basis Set

<i>cari(\$p, \$c)</i>	: 94-97 (Alur 1)
<i>carid(\$batas, \$mulai, \$p, \$c)</i>	: 101-104 (Alur 1)
<i>datapeta(\$w)</i>	: 136 (Alur 1)
<i>fk(\$fk)</i>	: 19 (Alur 1)
<i>get_dataset(\$batas, \$mulai)</i>	: 108-109 (Alur 1)
<i>get_datasetfull()</i>	: 113-114 (Alur 1)
<i>get_datasetid(\$id)</i>	: 127-128 (Alur 1)
<i>get_fk(\$s)</i>	: 23-24-25-26-27 (Alur 1)
<i>get_jalur(\$i)</i>	: 37-42 (Alur 1)
<i>get_koor(\$i)</i>	: 53-55 (Alur 1)
<i>get_lintas(\$i, \$w)</i>	: 46-49 (Alur 1)
<i>get_stasiun(\$i)</i>	: 31-33 (Alur 1)
<i>hapus(\$data)</i>	: 132 (Alur 1)

<i>jml</i> (<i>d</i>)	: 85-86-87-88-89-90 (Alur 1) 85-86-87-88-89-91 (Alur 2)
<i>petamjfull</i> (\$ <i>w</i>)	: 59-62 (Alur 1)
<i>proses</i> (\$ <i>id</i> , \$ <i>p</i>)	: 66 (Alur 1)
<i>selesai</i> (\$ <i>id</i> , \$ <i>s</i>)	: 70 (Alur 1)
<i>tambah</i> (\$ <i>id</i>)	: 11 (Alur 1)
<i>tambahd</i> (\$ <i>data</i>)	: 118 (Alur 1)
<i>tgl</i> (\$ <i>id</i>)	: 161-164 (Alur 1)
<i>ubah</i> (\$ <i>id</i>)	: 15 (Alur 1)
<i>ubahd</i> (\$ <i>data</i>)	: 122-123 (Alur 1)
<i>userc</i> (\$ <i>w</i>)	: 142-152-153-154-155-156- 157(Alur 1)
<i>validasi</i> (\$ <i>id</i>)	: 6 (Alur 1)

5. Kasus Uji atau *Test Case*

Alur 1	
<i>Test Case</i>	<i>cari</i> (\$ <i>p</i> , \$ <i>c</i>)
Target yang Diharapkan	Mendapatkan nilai jumlah baris berdasarkan pencarian yang menggunakan nilai parameter pilihan dan cari pada <i>dataset</i> dari basis data
Hasil Pengujian	Benar
Alur / Path	94-97
Alur 1	
<i>Test Case</i>	<i>carid</i> (\$ <i>batas</i> , \$ <i>mulai</i> , \$ <i>p</i> , \$ <i>c</i>)
Target yang Diharapkan	Mendapatkan hasil pencarian yang menggunakan nilai parameter pilihan dan cari pada <i>dataset</i> dari basis data dengan nilai batasan data untuk setiap halaman didapat dari parameter batas dan nilai mulai untuk setiap halaman hasil pencarian didapat dari parameter mulai
Hasil Pengujian	Benar
Alur / Path	101-104
Alur 1	
<i>Test Case</i>	<i>datapeta</i> (\$ <i>w</i>)
Target yang Diharapkan	Mendapatkan nilai-nilai data dari basis data yang akan digunakan untuk menampilkan data pada tabel tinjauan terkini data lintas suatu wilayah
Hasil Pengujian	Benar
Alur / Path	136

Alur 1	
<i>Test Case</i>	<i>fk(\$fk)</i>
Target yang Diharapkan	Menambah data kriteria lintas ke basis data
Hasil Pengujian	Benar
Alur / Path	19
Alur 1	
<i>Test Case</i>	<i>get_dataset(\$batas, \$mulai)</i>
Target yang Diharapkan	Mendapatkan atribut-atribut dan kelas dari tabel <i>dataset</i> pada basis data dengan nilai batasan data untuk setiap halaman didapat dari parameter batas dan nilai mulai untuk setiap halaman hasil pencarian didapat dari parameter mulai
Hasil Pengujian	Benar
Alur / Path	108-109
Alur 1	
<i>Test Case</i>	<i>get_datasetfull()</i>
Target yang Diharapkan	Mendapatkan atribut-atribut dan kelas dari tabel <i>dataset</i> pada basis data
Hasil Pengujian	Benar
Alur / Path	113-114
Alur 1	
<i>Test Case</i>	<i>get_datasetid(\$id)</i>
Target yang Diharapkan	Mendapatkan nilai atribut-atribut dan kelas dari tabel <i>dataset</i> pada basis data berdasarkan parameter <i>id</i>
Hasil Pengujian	Benar
Alur / Path	127-128
Alur 1	
<i>Test Case</i>	<i>get_fk(\$s)</i>
Target yang Diharapkan	Menampilkan nilai data kriteria lintas ke basis data berdasarkan parameter untuk nilai <i>idlintas</i>
Hasil Pengujian	Benar
Alur / Path	23-24-25-26-27
Alur 1	
<i>Test Case</i>	<i>get_jalur(\$i)</i>
Target yang Diharapkan	Mendapatkan nilai JSON untuk jalur beserta <i>properties</i> dari basis data berdasarkan <i>lintasid</i> yang nilainya sesuai dengan parameter
Hasil Pengujian	Benar
Alur / Path	37-42
Alur 1	
<i>Test Case</i>	<i>get_koor(\$i)</i>
Target yang Diharapkan	Mendapatkan nilai koordinat stasiun dari

Diharapkan	basis data beserta nama dan posisi jarak dari stasiun terminus awal
Hasil Pengujian	Benar
Alur / Path	53-55
Alur 1	
Test Case	<i>get_lintas(\$i, \$w)</i>
Target yang Diharapkan	Mendapatkan nama lintas dari basis data berdasarkan <i>lintasid</i> dan <i>wilid</i> yang nilainya sesuai dengan parameter
Hasil Pengujian	Benar
Alur / Path	46-49
Alur 1	
Test Case	<i>get_stasiun(\$i)</i>
Target yang Diharapkan	Mendapatkan nilai JSON untuk stasiun beserta <i>properties</i> dari basis data berdasarkan <i>lintasid</i> yang nilainya sesuai dengan parameter
Hasil Pengujian	Benar
Alur / Path	31-33
Alur 1	
Test Case	<i>hapus(\$data)</i>
Target yang Diharapkan	Mengubah nilai hapus menjadi “1” untuk data pada <i>dataset</i> di basis data berdasarkan <i>id</i> dari larik parameter
Hasil Pengujian	Benar
Alur / Path	132
Alur 1	
Test Case	<i>jml()</i>
Target yang Diharapkan	Mendapatkan nilai jumlah data pada <i>dataset</i> dari basis data
Hasil Pengujian	Benar
Alur / Path	85-86-87-88-89-90
Alur 2	
Test Case	<i>jml()</i>
Target yang Diharapkan	Proses permisalan tersebut selesai
Hasil Pengujian	Benar
Alur / Path	85-86-87-88-89-91
Alur 1	
Test Case	<i>petamjfull(\$w)</i>
Target yang Diharapkan	Mendapatkan nilai-nilai data dari basis data yang akan digunakan untuk menampilkan data pada tabel tinjauan terkini data peta suatu wilayah
Hasil Pengujian	Benar
Alur / Path	59-62
Alur 1	

<i>Test Case</i>	<i>proses(\$id, \$p)</i>
Target yang Diharapkan	Menjadikan status “Proses Perancangan” pada basis data berdasarkan suatu jalur yang ditentukan dari parameter tersebut
Hasil Pengujian	Benar
Alur / Path	66
Alur 1	
<i>Test Case</i>	<i>selesai(\$id, \$s)</i>
Target yang Diharapkan	Menjadikan status “Belum Tervalidasi” pada basis data berdasarkan suatu jalur yang ditentukan dari parameter tersebut beserta tanggal selesai tersebut
Hasil Pengujian	Benar
Alur / Path	70
Alur 1	
<i>Test Case</i>	<i>tambah (\$id)</i>
Target yang Diharapkan	Menjadikan status “tambah” pada basis data berdasarkan suatu jalur yang ditentukan dari parameter tersebut
Hasil Pengujian	Benar
Alur / Path	11
Alur 1	
<i>Test Case</i>	<i>tambahd(\$data)</i>
Target yang Diharapkan	Menambahkan data pada <i>dataset</i> berdasarkan parameter ke basis data
Hasil Pengujian	Benar
Alur / Path	118
Alur 1	
<i>Test Case</i>	<i>tgl(\$id)</i>
Target yang Diharapkan	Mendapatkan tanggal penyelesaian seluruh klasifikasi kecepatan maksimum suatu jalur kereta api dan tanggal validasi tersebut dari basis data berdasarkan jalur yang ditentukan dari parameter
Hasil Pengujian	Benar
Alur / Path	161-164
Alur 1	
<i>Test Case</i>	<i>ubah(\$id)</i>
Target yang Diharapkan	Menjadikan status “ubah” pada basis data berdasarkan suatu jalur yang ditentukan dari parameter tersebut
Hasil Pengujian	Benar
Alur / Path	15
Alur 1	
<i>Test Case</i>	<i>ubahd(\$data)</i>
Target yang Diharapkan	Mengubah data pada <i>dataset</i> berdasarkan <i>id</i> yang menjadi isi dari larik parameter

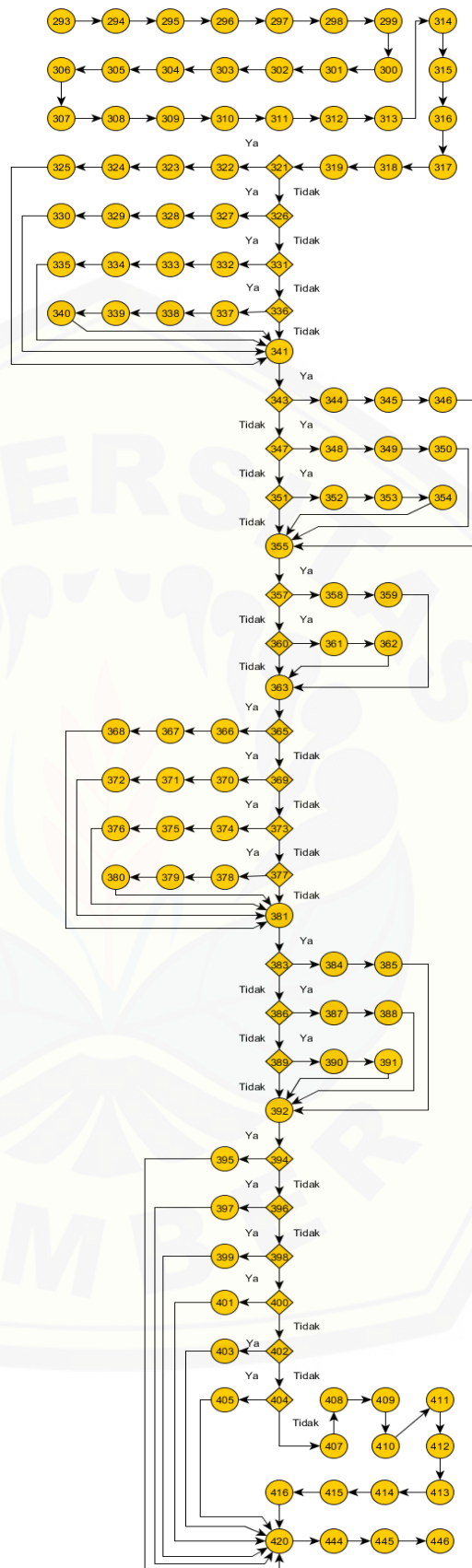
	tersebut pada basis data
Hasil Pengujian	Benar
Alur / Path	122-123
Alur 1	
Test Case	<i>userc(\$w)</i>
Target yang Diharapkan	Mendapatkan nama pegawai, NIPP, dan jabatan dari basis data sesuai dengan wilayah yang nilainya didapat dari parameter
Hasil Pengujian	Benar
Alur / Path	142-152-153-154-155-156-157
Alur 1	
Test Case	<i>validasi(\$id)</i>
Target yang Diharapkan	Menjadikan status “validasi” pada basis data berdasarkan suatu jalur yang ditentukan dari parameter tersebut beserta tanggal validasi tersebut
Hasil Pengujian	Benar
Alur / Path	6

4. Berkas *c_perceptron.php*

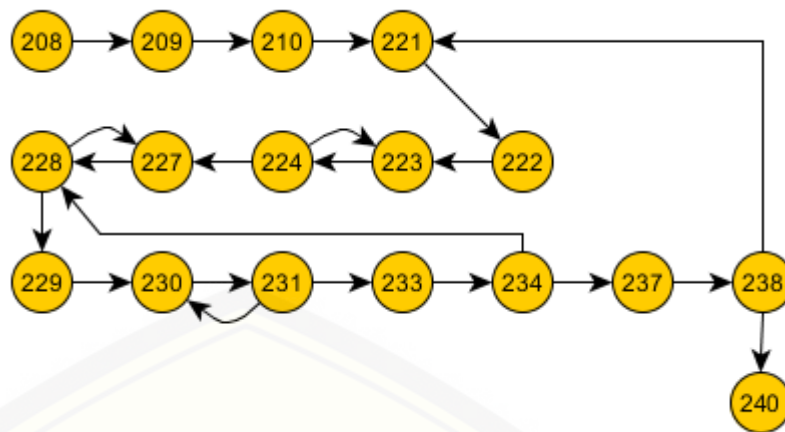
1. Daftar *Method*

Method-method lain yang digunakan pada alur tersebut sebagai berikut: *klas()* dan *k_lintas(\$testData)*

2. Diagram Alir *Method*



Gambar D.0.30 Diagram Alir klas()



Gambar D.0.31 Diagram Alir $k_lintas(\$testData)$

3. Penghitungan *Cyclomatic Complexity* (CC)

$$klas() : V(G) = 142 - 121 + 2 = 23$$

$$k_lintas(\$testData) : V(G) = 21 - 17 + 2 = 6$$

4. Basis Set

$klas()$: 293-294-295-296-297-298-299-300-301-302-303-304-305-306-307-308-309-310-311-312-313-314-315-316-317-318-319-321-322-323-324-325-341-343-347-351-355-357-360-363-365-369-373-377-381-383-386-389-392-394-396-398-400-402-404-407-408-409-410-411-412-413-414-415-416-420-444-445-456 (Alur 1)

293-294-295-296-297-298-299-300-301-302-303-304-305-306-307-308-309-310-311-312-313-314-315-316-317-318-319-321-326-327-328-329-330-341-343-347-351-355-357-360-363-365-369-373-377-381-383-386-389-392-394-396-398-400-402-404-407-408-409-410-411-412-413-414-415-416-420-444-445-456 (Alur 2)

293-294-295-296-297-298-299-300-301-302-303-304-305-306-307-308-309-310-311-312-313-314-315-316-317-318-319-321-326-331-332-333-334-

335-341-343-347-351-355-357-360-363-365-369-
373-377-381-383-386-389-392-394-396-398-400-
402-404-407-408-409-410-411-412-413-414-415-
416-420-444-445-456 (Alur 3)

293-294-295-296-297-298-299-300-301-302-303-
304-305-306-307-308-309-310-311-312-313-314-
315-316-317-318-319-321-326-331-336-337-338-
339-340-341-343-347-351-355-357-360-363-365-
369-373-377-381-383-386-389-392-394-396-398-
400-402-404-407-408-409-410-411-412-413-414-
415-416-420-444-445-456 (Alur 4)

293-294-295-296-297-298-299-300-301-302-303-
304-305-306-307-308-309-310-311-312-313-314-
315-316-317-318-319-321-326-331-336-341-343-
344-345-346-355-357-360-363-365-369-373-377-
381-383-386-389-392-394-396-398-400-402-404-
407-408-409-410-411-412-413-414-415-416-420-
444-445-456 (Alur 5)

293-294-295-296-297-298-299-300-301-302-303-
304-305-306-307-308-309-310-311-312-313-314-
315-316-317-318-319-321-326-331-336-341-343-
347-348-349-350-355-357-360-363-365-369-373-
377-381-383-386-389-392-394-396-398-400-402-
404-407-408-409-410-411-412-413-414-415-416-
420-444-445-456 (Alur 6)

293-294-295-296-297-298-299-300-301-302-303-
304-305-306-307-308-309-310-311-312-313-314-
315-316-317-318-319-321-326-331-336-341-343-
347-351-352-353-354-355-357-360-363-365-369-
373-377-381-383-386-389-392-394-396-398-400-
402-404-407-408-409-410-411-412-413-414-415-
416-420-444-445-456 (Alur 7)

293-294-295-296-297-298-299-300-301-302-303-
304-305-306-307-308-309-310-311-312-313-314-
315-316-317-318-319-321-326-331-336-341-343-
347-351-355-357-358-359-363-365-369-373-377-
381-383-386-389-392-394-396-398-400-402-404-
407-408-409-410-411-412-413-414-415-416-420-
444-445-456 (Alur 8)

293-294-295-296-297-298-299-300-301-302-303-
304-305-306-307-308-309-310-311-312-313-314-
315-316-317-318-319-321-326-331-336-341-343-
347-351-355-357-360-361-362-363-365-369-373-
377-381-383-386-389-392-394-396-398-400-402-
404-407-408-409-410-411-412-413-414-415-416-
420-444-445-456 (Alur 9)

293-294-295-296-297-298-299-300-301-302-303-
304-305-306-307-308-309-310-311-312-313-314-
315-316-317-318-319-321-326-331-336-341-343-
347-351-355-357-360-363-365-366-367-368-381-
383-386-389-392-394-396-398-400-402-404-407-
408-409-410-411-412-413-414-415-416-420-444-
445-456 (Alur 10)

293-294-295-296-297-298-299-300-301-302-303-
304-305-306-307-308-309-310-311-312-313-314-
315-316-317-318-319-321-326-331-336-341-343-
347-351-355-357-360-363-365-369-370-371-372-
381-383-386-389-392-394-396-398-400-402-404-
407-408-409-410-411-412-413-414-415-416-420-
444-445-456 (Alur 11)

293-294-295-296-297-298-299-300-301-302-303-
304-305-306-307-308-309-310-311-312-313-314-
315-316-317-318-319-321-326-331-336-341-343-
347-351-355-357-360-363-365-369-373-374-375-

376-381-383-386-389-392-394-396-398-400-402-
404-407-408-409-410-411-412-413-414-415-416-
420-444-445-456 (Alur 12)

293-294-295-296-297-298-299-300-301-302-303-
304-305-306-307-308-309-310-311-312-313-314-
315-316-317-318-319-321-326-331-336-341-343-
347-351-355-357-360-363-365-369-373-377-378-
379-380-381-383-386-389-392-394-396-398-400-
402-404-407-408-409-410-411-412-413-414-415-
416-420-444-445-456 (Alur 13)

293-294-295-296-297-298-299-300-301-302-303-
304-305-306-307-308-309-310-311-312-313-314-
315-316-317-318-319-321-326-331-336-341-343-
347-351-355-357-360-363-365-369-373-377-381-
383-384-385-392-394-396-398-400-402-404-407-
408-409-410-411-412-413-414-415-416-420-444-
445-456 (Alur 14)

293-294-295-296-297-298-299-300-301-302-303-
304-305-306-307-308-309-310-311-312-313-314-
315-316-317-318-319-321-326-331-336-341-343-
347-351-355-357-360-363-365-369-373-377-381-
383-386-387-388-392-394-396-398-400-402-404-
407-408-409-410-411-412-413-414-415-416-420-
444-445-456 (Alur 15)

293-294-295-296-297-298-299-300-301-302-303-
304-305-306-307-308-309-310-311-312-313-314-
315-316-317-318-319-321-326-331-336-341-343-
347-351-355-357-360-363-365-369-373-377-381-
383-386-389-390-391-392-394-396-398-400-402-
404-407-408-409-410-411-412-413-414-415-416-
420-444-445-456 (Alur 16)

293-294-295-296-297-298-299-300-301-302-303-
304-305-306-307-308-309-310-311-312-313-314-
315-316-317-318-319-321-326-331-336-341-343-
347-351-355-357-360-363-365-369-373-377-381-
383-386-389-392-394-395-420-444-445-456 (Alur
17)

293-294-295-296-297-298-299-300-301-302-303-
304-305-306-307-308-309-310-311-312-313-314-
315-316-317-318-319-321-326-331-336-341-343-
347-351-355-357-360-363-365-369-373-377-381-
383-386-389-392-394-396-397-420-444-445-456
(Alur 18)

293-294-295-296-297-298-299-300-301-302-303-
304-305-306-307-308-309-310-311-312-313-314-
315-316-317-318-319-321-326-331-336-341-343-
347-351-355-357-360-363-365-369-373-377-381-
383-386-389-392-394-396-398-389-420-444-445-456
(Alur 19)

293-294-295-296-297-298-299-300-301-302-303-
304-305-306-307-308-309-310-311-312-313-314-
315-316-317-318-319-321-326-331-336-341-343-
347-351-355-357-360-363-365-369-373-377-381-
383-386-389-392-394-396-398-400-401-420-444-
445-456 (Alur 20)

293-294-295-296-297-298-299-300-301-302-303-
304-305-306-307-308-309-310-311-312-313-314-
315-316-317-318-319-321-326-331-336-341-343-
347-351-355-357-360-363-365-369-373-377-381-
383-386-389-392-394-396-398-400-402-403-420-
444-445-456 (Alur 21)

293-294-295-296-297-298-299-300-301-302-303-
304-305-306-307-308-309-310-311-312-313-314-

315-316-317-318-319-321-326-331-336-341-343-
347-351-355-357-360-363-365-369-373-377-381-
383-386-389-392-394-396-398-400-402-404-405-
420-444-445-456 (Alur 22)

293-294-295-296-297-298-299-300-301-302-303-
304-305-306-307-308-309-310-311-312-313-314-
315-316-317-318-319-321-326-331-336-341-343-
347-351-355-357-360-363-365-369-373-377-381-
383-386-389-392-394-396-398-400-402-404-407-
408-409-410-411-412-413-414-415-416-420-444-
445-456 (Alur 23)

k_lintas(\$testData) : 208-209-210-221-222-223-224-223-224-
227-228-229-230-231-233-234-237-238-
240 (Alur 1)

208-209-210-221-222-223-224-227-228-
227-228-229-230-231-233-234-237-238-
240 (Alur 2)

208-209-210-221-222-223-224-227-228-
229-230-231-230-231-233-234-237-238-
240 (Alur 3)

208-209-210-221-222-223-224-227-228-
229-230-231-233-234-228-229-230-231-
233-234-237-238-240 (Alur 4)

208-209-210-221-222-223-224-227-228-
229-230-231-233-234-237-238-222-223-
224-227-228-229-230-231-233-234-237-
238-240 (Alur 5)

208-209-210-221-222-223-224-227-228-
229-230-231-233-234-237-238-240 (Alur
6)

5. Kasus Uji atau *Test Case*

Alur 1	
<i>Test Case</i>	<i>klas()</i>
Target yang Diharapkan	Menentukan nilai \$r60 sama dengan 1 jika nilai \$tr sama dengan 'R.60'
Hasil Pengujian	Benar
Alur / Path	293-294-295-296-297-298-299-300-301-302-303-304-305-306-307-308-309-310-311-312-313-314-315-316-317-318-319-321-322-323-324-325-341-343-347-351-355-357-360-363-365-369-373-377-381-383-386-389-392-394-396-398-400-402-404-407-408-409-410-411-412-413-414-415-416-420-444-445-456
Alur 2	
<i>Test Case</i>	<i>klas()</i>
Target yang Diharapkan	Menentukan nilai \$r54 sama dengan 1 jika nilai \$tr sama dengan 'R.54'
Hasil Pengujian	Benar
Alur / Path	293-294-295-296-297-298-299-300-301-302-303-304-305-306-307-308-309-310-311-312-313-314-315-316-317-318-319-321-326-327-328-329-330-341-343-347-351-355-357-360-363-365-369-373-377-381-383-386-389-392-394-396-398-400-402-404-407-408-409-410-411-412-413-414-415-416-420-444-445-456
Alur 3	
<i>Test Case</i>	<i>klas()</i>
Target yang Diharapkan	Menentukan nilai \$r50 sama dengan 1 jika nilai \$tr sama dengan 'R.50'
Hasil Pengujian	Benar
Alur / Path	293-294-295-296-297-298-299-300-301-302-303-304-305-306-307-308-309-310-311-312-313-314-315-316-317-318-319-321-326-331-332-333-334-335-341-343-347-351-355-357-360-363-365-369-373-377-381-383-386-389-392-394-396-398-400-402-404-407-408-409-410-411-412-413-414-415-416-420-444-445-456
Alur 4	
<i>Test Case</i>	<i>klas()</i>
Target yang Diharapkan	Menentukan nilai \$r42 sama dengan 1 jika nilai \$tr sama dengan 'R.42'
Hasil Pengujian	Benar
Alur / Path	293-294-295-296-297-298-299-300-301-302-303-304-305-306-307-308-309-310-

	311-312-313-314-315-316-317-318-319-321-326-331-336-337-338-339-340-341-343-347-351-355-357-360-363-365-369-373-377-381-383-386-389-392-394-396-398-400-402-404-407-408-409-410-411-412-413-414-415-416-420-444-445-456
Alur 5	
<i>Test Case</i>	<i>klas()</i>
Target yang Diharapkan	Menentukan nilai \$beton sama dengan 1 jika nilai \$jb sama dengan 'Beton'
Hasil Pengujian	Benar
Alur / Path	293-294-295-296-297-298-299-300-301-302-303-304-305-306-307-308-309-310-311-312-313-314-315-316-317-318-319-321-326-331-336-341-343-344-345-346-355-357-360-363-365-369-373-377-381-383-386-389-392-394-396-398-400-402-404-407-408-409-410-411-412-413-414-415-416-420-444-445-456
Alur 6	
<i>Test Case</i>	<i>klas()</i>
Target yang Diharapkan	Menentukan nilai \$kayu sama dengan 1 jika nilai \$jb sama dengan 'Kayu'
Hasil Pengujian	Benar
Alur / Path	293-294-295-296-297-298-299-300-301-302-303-304-305-306-307-308-309-310-311-312-313-314-315-316-317-318-319-321-326-331-336-341-343-347-348-349-350-355-357-360-363-365-369-373-377-381-383-386-389-392-394-396-398-400-402-404-407-408-409-410-411-412-413-414-415-416-420-444-445-456
Alur 7	
<i>Test Case</i>	<i>klas()</i>
Target yang Diharapkan	Menentukan nilai \$baja sama dengan 1 jika nilai \$jb sama dengan 'Baja'
Hasil Pengujian	Benar
Alur / Path	293-294-295-296-297-298-299-300-301-302-303-304-305-306-307-308-309-310-311-312-313-314-315-316-317-318-319-321-326-331-336-341-343-347-351-352-353-354-355-357-360-363-365-369-373-377-381-383-386-389-392-394-396-398-400-402-404-407-408-409-410-411-412-413-414-415-416-420-444-445-456
Alur 8	
<i>Test Case</i>	<i>klas()</i>

Target yang Diharapkan	Menentukan nilai \$tunggal sama dengan 1 jika nilai \$jp sama dengan 'Tunggal'
Hasil Pengujian	Benar
Alur / Path	293-294-295-296-297-298-299-300-301-302-303-304-305-306-307-308-309-310-311-312-313-314-315-316-317-318-319-321-326-331-336-341-343-347-351-355-357-358-359-363-365-369-373-377-381-383-386-389-392-394-396-398-400-402-404-407-408-409-410-411-412-413-414-415-416-420-444-445-456
Alur 9	
Test Case	klas()
Target yang Diharapkan	Menentukan nilai \$ganda sama dengan 1 jika nilai \$jp sama dengan 'Ganda'
Hasil Pengujian	Benar
Alur / Path	293-294-295-296-297-298-299-300-301-302-303-304-305-306-307-308-309-310-311-312-313-314-315-316-317-318-319-321-326-331-336-341-343-347-351-355-357-360-361-362-363-365-369-373-377-381-383-386-389-392-394-396-398-400-402-404-407-408-409-410-411-412-413-414-415-416-420-444-445-456
Alur 10	
Test Case	klas()
Target yang Diharapkan	Menentukan nilai \$rminv sama dengan 1 jika nilai \$jj sama dengan 'v'
Hasil Pengujian	Benar
Alur / Path	293-294-295-296-297-298-299-300-301-302-303-304-305-306-307-308-309-310-311-312-313-314-315-316-317-318-319-321-326-331-336-341-343-347-351-355-357-360-363-365-366-367-368-381-383-386-389-392-394-396-398-400-402-404-407-408-409-410-411-412-413-414-415-416-420-444-445-456
Alur 11	
Test Case	klas()
Target yang Diharapkan	Menentukan nilai \$rmintph sama dengan 1 jika nilai \$jj sama dengan 'tph'
Hasil Pengujian	Benar
Alur / Path	293-294-295-296-297-298-299-300-301-302-303-304-305-306-307-308-309-310-311-312-313-314-315-316-317-318-319-321-326-331-336-341-343-347-351-355-357-360-363-365-369-370-371-372-381-

	383-386-389-392-394-396-398-400-402-404-407-408-409-410-411-412-413-414-415-416-420-444-445-456
Alur 12	
<i>Test Case</i>	<i>klas()</i>
Target yang Diharapkan	Menentukan nilai \$rminph sama dengan 1 jika nilai \$jj sama dengan 'ph'
Hasil Pengujian	Benar
Alur / Path	293-294-295-296-297-298-299-300-301-302-303-304-305-306-307-308-309-310-311-312-313-314-315-316-317-318-319-321-326-331-336-341-343-347-351-355-357-360-363-365-369-373-374-375-376-381-383-386-389-392-394-396-398-400-402-404-407-408-409-410-411-412-413-414-415-416-420-444-445-456
Alur 13	
<i>Test Case</i>	<i>klas()</i>
Target yang Diharapkan	Menentukan nilai \$rminv, \$rminph, dan \$rminph sama dengan 0 jika nilai \$jj sama dengan 'l'
Hasil Pengujian	Benar
Alur / Path	293-294-295-296-297-298-299-300-301-302-303-304-305-306-307-308-309-310-311-312-313-314-315-316-317-318-319-321-326-331-336-341-343-347-351-355-357-360-363-365-369-373-377-378-379-380-381-383-386-389-392-394-396-398-400-402-404-407-408-409-410-411-412-413-414-415-416-420-444-445-456
Alur 14	
<i>Test Case</i>	<i>klas()</i>
Target yang Diharapkan	Menentukan nilai \$j sama dengan 1 jika nilai \$jv sama dengan 'j'
Hasil Pengujian	Benar
Alur / Path	293-294-295-296-297-298-299-300-301-302-303-304-305-306-307-308-309-310-311-312-313-314-315-316-317-318-319-321-326-331-336-341-343-347-351-355-357-360-363-365-369-373-377-381-383-384-385-392-394-396-398-400-402-404-407-408-409-410-411-412-413-414-415-416-420-444-445-456
Alur 15	
<i>Test Case</i>	<i>klas()</i>
Target yang Diharapkan	Menentukan nilai \$v sama dengan 1 jika nilai \$jv sama dengan 'v'

Hasil Pengujian	Benar
Alur / Path	293-294-295-296-297-298-299-300-301-302-303-304-305-306-307-308-309-310-311-312-313-314-315-316-317-318-319-321-326-331-336-341-343-347-351-355-357-360-363-365-369-373-377-381-383-386-387-388-392-394-396-398-400-402-404-407-408-409-410-411-412-413-414-415-416-420-444-445-456
Alur 16	
Test Case	klas()
Target yang Diharapkan	Menentukan nilai \$j dan \$v sama dengan 0 jika nilai \$jv sama dengan 'b'
Hasil Pengujian	Benar
Alur / Path	293-294-295-296-297-298-299-300-301-302-303-304-305-306-307-308-309-310-311-312-313-314-315-316-317-318-319-321-326-331-336-341-343-347-351-355-357-360-363-365-369-373-377-381-383-386-389-390-391-392-394-396-398-400-402-404-407-408-409-410-411-412-413-414-415-416-420-444-445-456
Alur 17	
Test Case	klas()
Target yang Diharapkan	Menentukan nilai \$vmaks sama dengan '60 km/jam' jika nilai \$rb tidak sama dengan 0 dan kurang dari 4,5 serta \$jv sama dengan 'j'
Hasil Pengujian	Benar
Alur / Path	293-294-295-296-297-298-299-300-301-302-303-304-305-306-307-308-309-310-311-312-313-314-315-316-317-318-319-321-326-331-336-341-343-347-351-355-357-360-363-365-369-373-377-381-383-386-389-392-394-395-420-444-445-456
Alur 18	
Test Case	klas()
Target yang Diharapkan	Menentukan nilai \$vmaks sama dengan '60 km/jam' jika nilai \$rb tidak sama dengan 0 dan kurang dari 4,7 serta \$jv sama dengan 'v'
Hasil Pengujian	Benar
Alur / Path	293-294-295-296-297-298-299-300-301-302-303-304-305-306-307-308-309-310-311-312-313-314-315-316-317-318-319-321-326-331-336-341-343-347-351-355-357-360-363-365-369-373-377-381-383-386-389-392-394-396-397-420-444-445-

	456
Alur 19	
<i>Test Case</i>	<i>klas()</i>
Target yang Diharapkan	Menentukan nilai \$vmaks sama dengan '70 km/jam' jika nilai \$rmintph lebih dari 810 hingga lebih dari sama dengan 1050
Hasil Pengujian	Benar
Alur / Path	293-294-295-296-297-298-299-300-301-302-303-304-305-306-307-308-309-310-311-312-313-314-315-316-317-318-319-321-326-331-336-341-343-347-351-355-357-360-363-365-369-373-377-381-383-386-389-392-394-396-398-389-420-444-445-456
Alur 20	
<i>Test Case</i>	<i>klas()</i>
Target yang Diharapkan	Menentukan nilai \$vmaks sama dengan '70 km/jam' jika nilai \$rminph lebih dari 270 hingga lebih dari sama dengan 350
Hasil Pengujian	Benar
Alur / Path	293-294-295-296-297-298-299-300-301-302-303-304-305-306-307-308-309-310-311-312-313-314-315-316-317-318-319-321-326-331-336-341-343-347-351-355-357-360-363-365-369-373-377-381-383-386-389-392-394-396-398-400-401-420-444-445-456
Alur 21	
<i>Test Case</i>	<i>klas()</i>
Target yang Diharapkan	Menentukan nilai \$vmaks sama dengan '60 km/jam' jika nilai \$rmintph lebih dari sama dengan 600 hingga lebih dari sama dengan 810
Hasil Pengujian	Benar
Alur / Path	293-294-295-296-297-298-299-300-301-302-303-304-305-306-307-308-309-310-311-312-313-314-315-316-317-318-319-321-326-331-336-341-343-347-351-355-357-360-363-365-369-373-377-381-383-386-389-392-394-396-398-400-402-403-420-444-445-456
Alur 22	
<i>Test Case</i>	<i>klas()</i>
Target yang Diharapkan	Menentukan nilai \$vmaks sama dengan '60 km/jam' jika nilai \$rminph lebih dari sama dengan 200 hingga lebih dari sama dengan 270

Hasil Pengujian	Benar
Alur / Path	293-294-295-296-297-298-299-300-301-302-303-304-305-306-307-308-309-310-311-312-313-314-315-316-317-318-319-321-326-331-336-341-343-347-351-355-357-360-363-365-369-373-377-381-383-386-389-392-394-396-398-400-402-404-405-420-444-445-456
Alur 23	
Test Case	<i>klas()</i>
Target yang Diharapkan	Menentukan nilai \$vmaks berdasarkan klasifikasi yang dilakukan oleh <i>method k_lintas(\$testData)</i>
Hasil Pengujian	Benar
Alur / Path	293-294-295-296-297-298-299-300-301-302-303-304-305-306-307-308-309-310-311-312-313-314-315-316-317-318-319-321-326-331-336-341-343-347-351-355-357-360-363-365-369-373-377-381-383-386-389-392-394-396-398-400-402-404-407-408-409-410-411-412-413-414-415-416-420-444-445-456
Alur 1	
Test Case	<i>k_lintas(\$testData)</i>
Target yang Diharapkan	Melakukan pengulangan yang menggunakan variabel \$m sebanyak jumlah titik bobot untuk mendapatkan nilai variabel \$keluaran yaitu nilai data baru
Hasil Pengujian	Benar
Alur / Path	208-209-210-221-222-223-224-223-224-227-228-229-230-231-233-234-237-238-240
Alur 2	
Test Case	<i>k_lintas(\$testData)</i>
Target yang Diharapkan	Melakukan pengulangan yang menggunakan variabel \$l sebanyak satu lapisan untuk mendapatkan nilai variabel \$keluaran yang berubah menjadi nilai hasil penghitungan klasifikasi data baru
Hasil Pengujian	Benar
Alur / Path	208-209-210-221-222-223-224-227-228-227-228-229-230-231-233-234-237-238-240
Alur 3	
Test Case	<i>k_lintas(\$testData)</i>
Target yang Diharapkan	Melakukan pengulangan yang menggunakan variabel \$j sebanyak jumlah

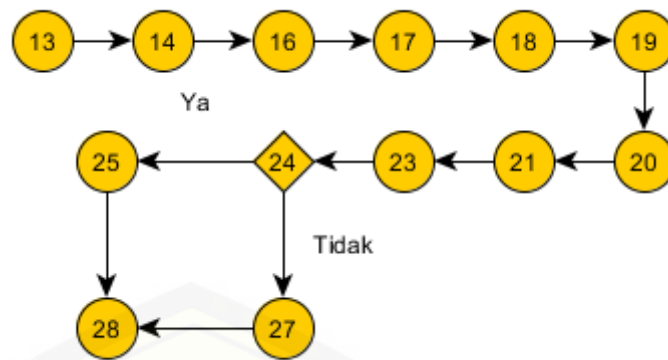
	titik bobot yang dikurangi satu untuk mendapatkan nilai variabel \$keluaran yang berubah menjadi nilai hasil penghitungan klasifikasi data baru
Hasil Pengujian	Benar
Alur / Path	208-209-210-221-222-223-224-227-228-229-230-231-230-231-233-234-237-238-240
Alur 4	
Test Case	$k_lintas(\$testData)$
Target yang Diharapkan	Melakukan pengulangan yang menggunakan variabel \$i sebanyak satu bias untuk mendapatkan nilai variabel \$keluaran yang berubah menjadi nilai hasil penghitungan klasifikasi data baru
Hasil Pengujian	Benar
Alur / Path	208-209-210-221-222-223-224-227-228-229-230-231-233-234-228-229-230-231-233-234-237-238-240
Alur 5	
Test Case	$k_lintas(\$testData)$
Target yang Diharapkan	Melakukan pengulangan yang menggunakan variabel \$m sebanyak jumlah data baru untuk mendapatkan nilai variabel \$h_pred yang merupakan nilai hasil klasifikasi
Hasil Pengujian	Benar
Alur / Path	208-209-210-221-222-223-224-227-228-229-230-231-233-234-237-238-222-223-224-227-228-229-230-231-233-234-237-238-240
Alur 6	
Test Case	$k_lintas(\$testData)$
Target yang Diharapkan	Menghasilkan nilai kecepatan maksimum berdasarkan parameter data untuk klasifikasi SLP
Hasil Pengujian	Benar
Alur / Path	208-209-210-221-222-223-224-227-228-229-230-231-233-234-237-238-240

5. Berkas *c_cetak.php*

1. Daftar *Method*

Method yang digunakan pada alur tersebut adalah *laporan()*

2. Diagram Alir *Method*



Gambar D.0.32 Diagram Alir *laporan()*

3. Penghitungan *Cyclomatic Complexity* (CC)

laporan() : $V(G) = 13 - 13 + 2 = 2$

4. Basis Set

laporan() : 13-14-16-17-18-19-20-21-23-24-25-28

(Alur 1)

13-14-16-17-18-19-20-21-23-24-27-28

(Alur 2)

5. Kasus Uji atau *Test Case*

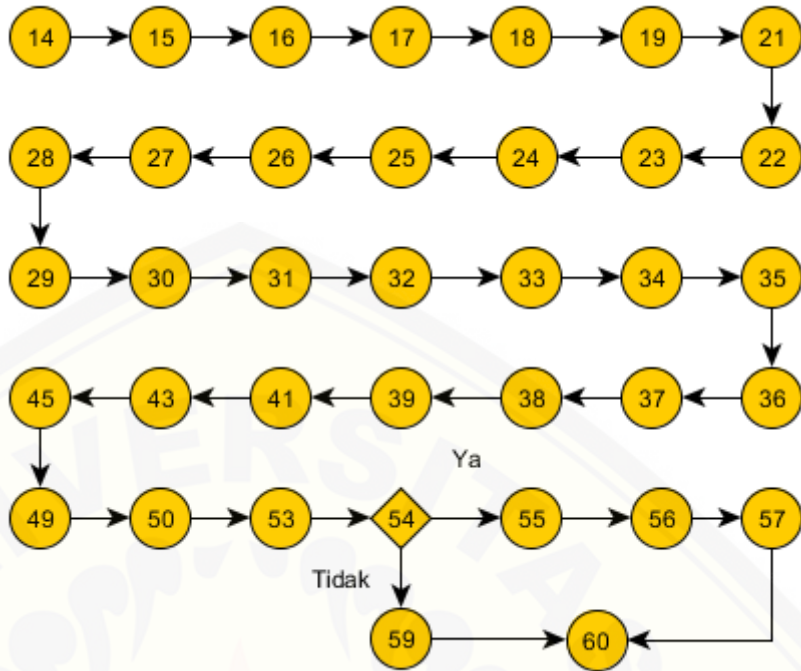
Alur 1	
<i>Test Case</i>	<i>laporan()</i>
Target yang Diharapkan	Mengolah data-data yang digunakan pada laporan peta
Hasil Pengujian	Benar
Alur / Path	13-14-16-17-18-19-20-21-23-24-25-28
Alur 2	
<i>Test Case</i>	<i>laporan()</i>
Target yang Diharapkan	Kembali ke <i>method c_login</i>
Hasil Pengujian	Benar
Alur / Path	13-14-16-17-18-19-20-21-23-24-27-28

6. Berkas *c_dataset.php*

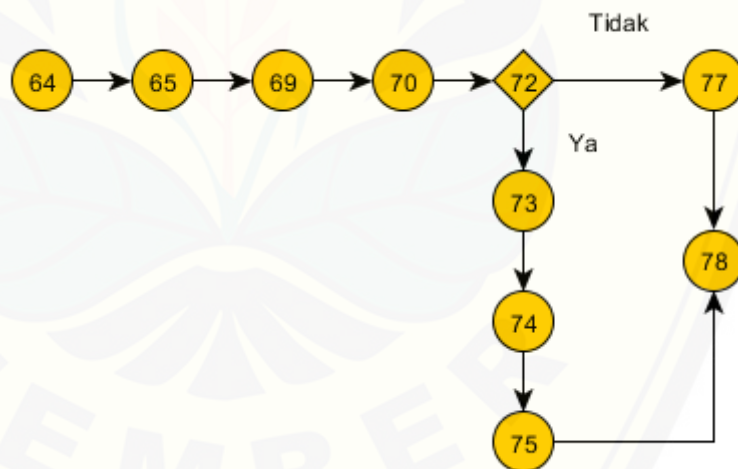
1. Daftar *Method*

Method-method yang digunakan pada alur tersebut sebagai berikut: *index()*, *data()*, *tambah()*, *tambahd()*, *ubah()*, *ubahd()*, *hapus()*, dan *cari()*

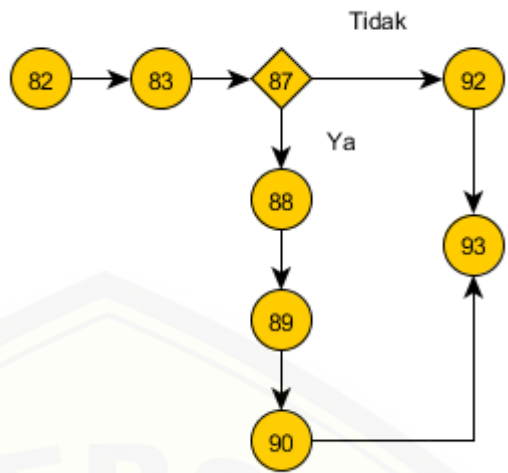
2. Diagram Alir *Method*



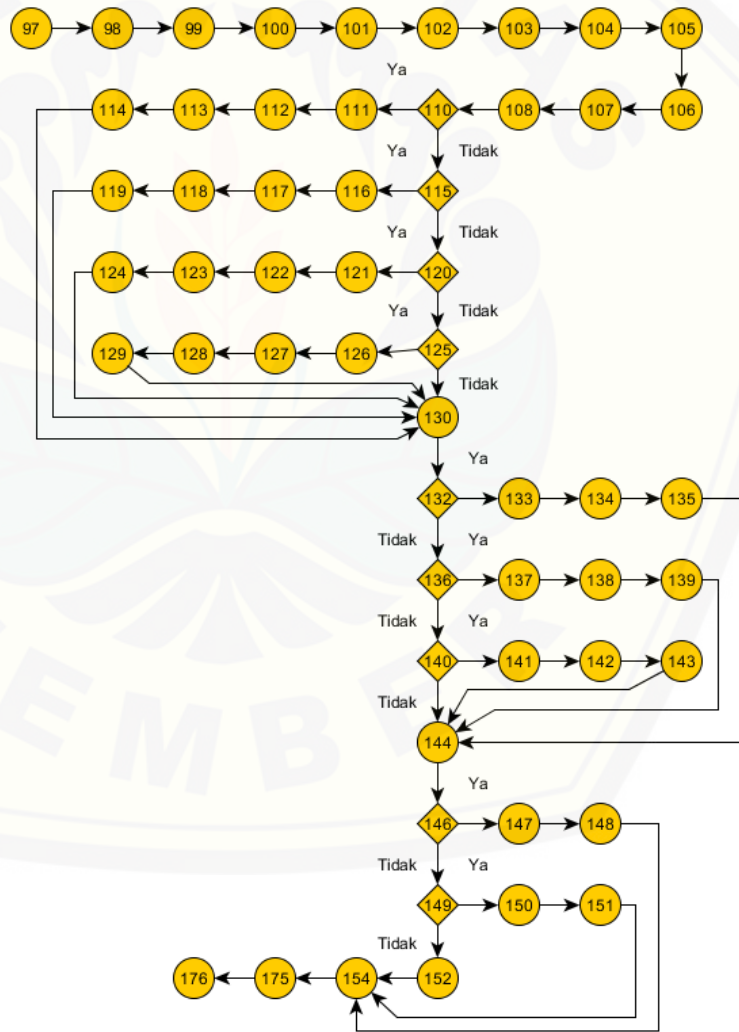
Gambar D.0.33 Diagram Alir *index()*



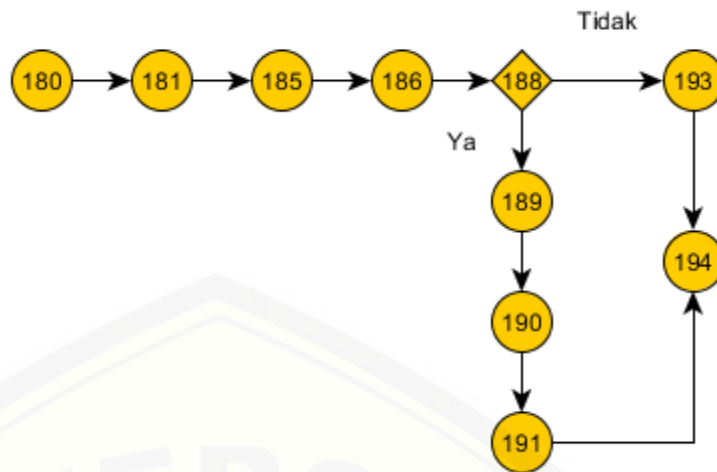
Gambar D.0.34 Diagram Alir *data()*



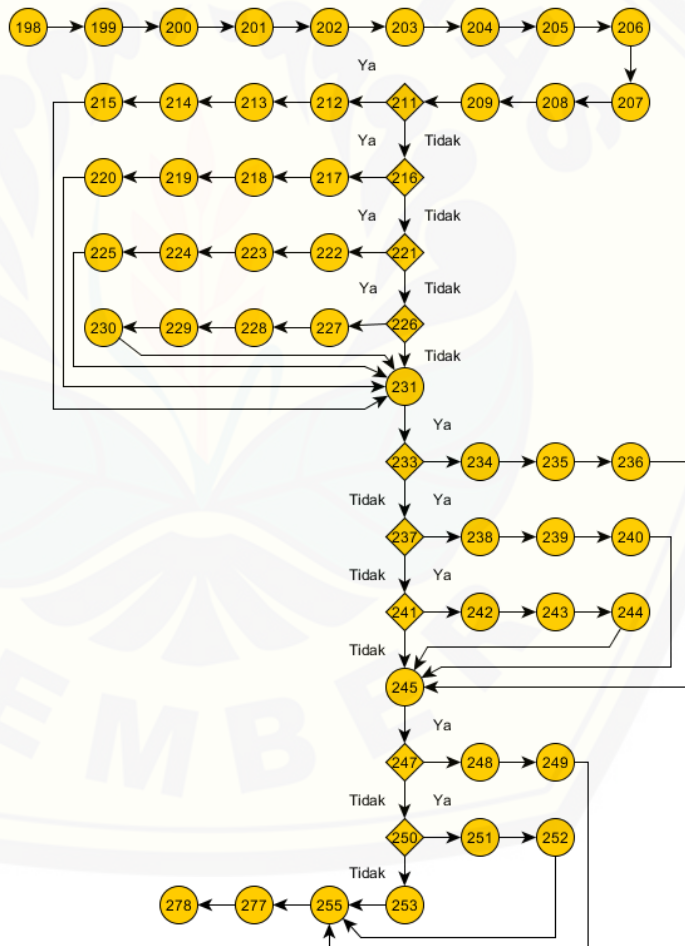
Gambar D.0.35 Diagram Alir tambah()



Gambar D.0.36 Diagram Alir tambahd()



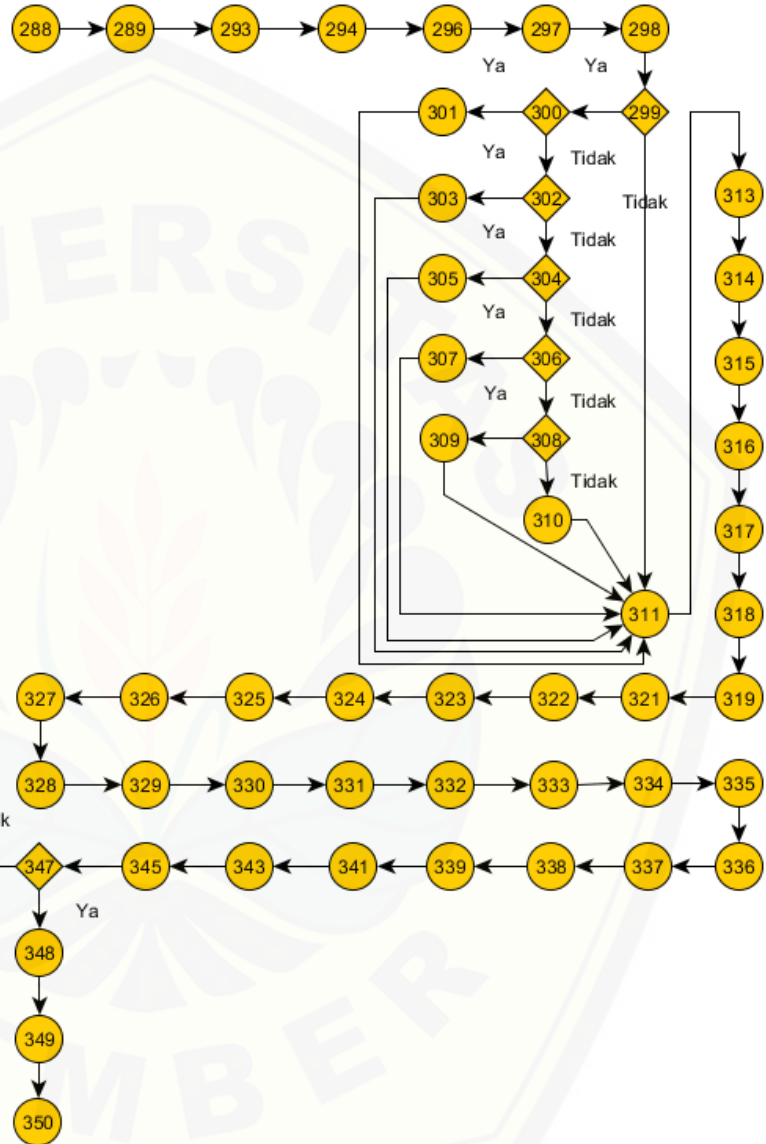
Gambar D.0.37 Diagram Alir ubah()



Gambar D.0.38 Diagram Alir ubahd()



Gambar D.0.39 Diagram Alir *hapus()*



Gambar D.0.40 Diagram Alir *cari()*

3. Penghitungan *Cyclomatic Complexity (CC)*

index() : $V(G) = 37 - 37 + 2 = 2$

data() : $V(G) = 10 - 10 + 2 = 2$

tambah() : $V(G) = 8 - 8 + 2 = 2$

tambahd() : $V(G) = 64 - 56 + 2 = 10$

ubah() : $V(G) = 10 - 10 + 2 = 2$

ubahd() : $V(G) = 64 - 56 + 2 = 10$

hapus() : $V(G) = 2 - 3 + 2 = 1$

cari() : $V(G) = 59 - 54 + 2 = 7$

4. Basis Set

index() : 14-15-16-17-18-19-21-22-23-24-25-26-27-28-29-
30-31-32-33-34-35-36-37-38-39-41-43-45-49-50-
53-54-55-56-57-60 (Alur 1)

14-15-16-17-18-19-21-22-23-24-25-26-27-28-29-
30-31-32-33-34-35-36-37-38-39-41-43-45-49-50-
53-54-59-60 (Alur 2)

data() : 64-65-69-70-72-73-74-75-78 (Alur 1)

64-65-69-70-72-77-78 (Alur 2)

tambah() : 82-83-87-88-89-90-93 (Alur 1)

82-83-87-92-93 (Alur 2)

tambahd() : 97-98-99-100-101-102-103-104-105-106-107-
108-110-115-120-125-130-132-136-140-144-
146-149-152-154-175-176 (Alur 1)

97-98-99-100-101-102-103-104-105-106-107-
108-110-111-112-113-114-130-132-136-140-
144-146-149-152-154-175-176 (Alur 2)

97-98-99-100-101-102-103-104-105-106-107-
108-110-115-116-117-118-119-130-132-136-
140-144-146-149-152-154-175-176 (Alur 3)

97-98-99-100-101-102-103-104-105-106-107-
108-110-115-120-121-122-123-124-130-132-
136-140-144-146-149-152-154-175-176 (Alur 4)

97-98-99-100-101-102-103-104-105-106-107-
108-110-115-120-125-126-127-128-129-130-
132-136-140-144-146-149-152-154-175-176

(Alur 5)

97-98-99-100-101-102-103-104-105-106-107-
108-110-115-120-125-130-132-133-134-135-
144-146-149-152-154-175-176 (Alur 6)

97-98-99-100-101-102-103-104-105-106-107-
108-110-115-120-125-130-132-136-137-138-
139-144-146-149-152-154-175-176 (Alur 7)

97-98-99-100-101-102-103-104-105-106-107-
108-110-115-120-125-130-132-136-140-141-
142-143-144-146-149-152-154-175-176 (Alur 8)

97-98-99-100-101-102-103-104-105-106-107-
108-110-115-120-125-130-132-136-140-144-
146-147-148-154-175-176 (Alur 9)

97-98-99-100-101-102-103-104-105-106-107-
108-110-115-120-125-130-132-136-140-144-
146-149-150-151-154-175-176 (Alur 10)

ubah() : 180-181-185-186-188-189-190-191-194 (Alur 1)
180-181-185-186-188-193-194 (Alur 2)

ubahd() : 198-199-200-201-202-203-204-205-206-207-
208-209-211-216-221-226-231-233-237-241-
245-247-250-253-255-277-278 (Alur 1)

198-199-200-201-202-203-204-205-206-207-
208-209-211-212-213-214-215-231-233-237-
241-245-247-250-253-255-277-278 (Alur 2)

198-199-200-201-202-203-204-205-206-207-
208-209-211-216-217-218-219-220-231-233-
237-241-245-247-250-253-255-277-278 (Alur 3)

198-199-200-201-202-203-204-205-206-207-
208-209-211-216-221-222-223-224-225-231-
233-237-241-245-247-250-253-255-277-278
(Alur 4)

198-199-200-201-202-203-204-205-206-207-
208-209-211-216-221-226-227-228-229-230-

231-233-237-241-245-247-250-253-255-277-278
(Alur 5)

198-199-200-201-202-203-204-205-206-207-
208-209-211-216-221-226-231-233-234-235-
236-245-247-250-253-255-277-278 (Alur 6)

198-199-200-201-202-203-204-205-206-207-
208-209-211-216-221-226-231-233-237-238-
239-240-245-247-250-253-255-277-278 (Alur 7)

198-199-200-201-202-203-204-205-206-207-
208-209-211-216-221-226-231-233-237-241-
242-243-244-245-247-250-253-255-277-278
(Alur 8)

198-199-200-201-202-203-204-205-206-207-
208-209-211-216-221-226-231-233-237-241-
245-247-248-249-255-277-278 (Alur 9)

198-199-200-201-202-203-204-205-206-207-
208-209-211-216-221-226-231-233-237-241-
245-247-250-251-252-255-277-278 (Alur 10)

hapus() : 282-283-284 (Alur 1)

cari() : 288-289-290-294-296-297-298-299-311-313-
314-315-316-317-318-319-321-322-323-324-
325-326-327-328-329-330-331-332-333-334-
335-336-337-338-339-341-343-345-347-348-
349-350-353 (Alur 1)

288-289-290-294-296-297-298-299-300-302-
304-306-308-310-311-313-314-315-316-317-
318-319-321-322-323-324-325-326-327-328-
329-330-331-332-333-334-335-336-337-338-
339-341-343-345-347-348-349-350-353 (Alur 2)

288-289-290-294-296-297-298-299-300-301-
311-313-314-315-316-317-318-319-321-322-
323-324-325-326-327-328-329-330-331-332-

333-334-335-336-337-338-339-341-343-345-
347-348-349-350-353 (Alur 3)

288-289-290-294-296-297-298-299-300-302-
303-311-313-314-315-316-317-318-319-321-
322-323-324-325-326-327-328-329-330-331-
332-333-334-335-336-337-338-339-341-343-
345-347-348-349-350-353 (Alur 4)

288-289-290-294-296-297-298-299-300-302-
304-305-311-313-314-315-316-317-318-319-
321-322-323-324-325-326-327-328-329-330-
331-332-333-334-335-336-337-338-339-341-
343-345-347-348-349-350-353 (Alur 5)

288-289-290-294-296-297-298-299-300-302-
304-306-307-311-313-314-315-316-317-318-
319-321-322-323-324-325-326-327-328-329-
330-331-332-333-334-335-336-337-338-339-
341-343-345-347-348-349-350-353 (Alur 6)

288-289-290-294-296-297-298-299-300-302-
304-306-308-309-311-313-314-315-316-317-
318-319-321-322-323-324-325-326-327-328-
329-330-331-332-333-334-335-336-337-338-
339-341-343-345-347-348-349-350-353 (Alur 7)

288-289-290-294-296-297-298-299-311-313-
314-315-316-317-318-319-321-322-323-324-
325-326-327-328-329-330-331-332-333-334-
335-336-337-338-339-341-343-345-347-352-353
(Alur 8)

5. Kasus Uji atau *Test Case*

Alur 1	
<i>Test Case</i>	<i>index()</i>
Target yang Diharapkan	Menampilkan halaman <i>dataset</i> beserta tabel <i>dataset</i> yang menampilkan pembagian 30 data per halaman
Hasil Pengujian	Benar
Alur / Path	14-15-16-17-18-19-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-41-43-45-49-50-53-54-55-56-57-60
Alur 2	
<i>Test Case</i>	<i>index()</i>
Target yang Diharapkan	Melakukan <i>redirect</i> ke <i>class c_login</i> jika hak akses tidak sesuai
Hasil Pengujian	Benar
Alur / Path	14-15-16-17-18-19-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-41-43-45-49-50-53-54-59-60
Alur 1	
<i>Test Case</i>	<i>data()</i>
Target yang Diharapkan	Menampilkan halaman <i>dataset</i> beserta tabel <i>dataset</i> yang menampilkan data atribut dan kelas secara keseluruhan
Hasil Pengujian	Benar
Alur / Path	64-65-69-70-72-73-74-75-78
Alur 2	
<i>Test Case</i>	<i>data()</i>
Target yang Diharapkan	Melakukan <i>redirect</i> ke <i>class c_login</i> jika hak akses tidak sesuai
Hasil Pengujian	Benar
Alur / Path	64-65-69-70-72-77-78
Alur 1	
<i>Test Case</i>	<i>tambah()</i>
Target yang Diharapkan	Menampilkan halaman penambahan data pada <i>dataset</i>
Hasil Pengujian	Benar
Alur / Path	82-83-87-88-89-90-93
Alur 2	
<i>Test Case</i>	<i>tambah()</i>
Target yang Diharapkan	Melakukan <i>redirect</i> ke <i>class c_login</i> jika hak akses tidak sesuai
Hasil Pengujian	Benar
Alur / Path	82-83-87-92-93
Alur 1	
<i>Test Case</i>	<i>tambahd()</i>
Target yang	Mengolah masukan dari <i>form</i> tambah

Diharapkan	<i>dataset</i> sehingga dikumpulkan dalam sebuah larik sebagai parameter ke <i>model</i>
Hasil Pengujian	Benar
Alur / Path	97-98-99-100-101-102-103-104-105-106-107-108-110-115-120-125-130-132-136-140-144-146-149-152-154-175-176
Alur 2	
<i>Test Case</i>	<i>tambahd()</i>
Target yang Diharapkan	Menentukan nilai \$r60 sama dengan 1 jika nilai \$tr sama dengan 'R.60'
Hasil Pengujian	Benar
Alur / Path	97-98-99-100-101-102-103-104-105-106-107-108-110-111-112-113-114-130-132-136-140-144-146-149-152-154-175-176
Alur 3	
<i>Test Case</i>	<i>tambahd()</i>
Target yang Diharapkan	Menentukan nilai \$r54 sama dengan 1 jika nilai \$tr sama dengan 'R.54'
Hasil Pengujian	Benar
Alur / Path	97-98-99-100-101-102-103-104-105-106-107-108-110-115-116-117-118-119-130-132-136-140-144-146-149-152-154-175-176
Alur 4	
<i>Test Case</i>	<i>tambahd()</i>
Target yang Diharapkan	Menentukan nilai \$r50 sama dengan 1 jika nilai \$tr sama dengan 'R.50'
Hasil Pengujian	Benar
Alur / Path	97-98-99-100-101-102-103-104-105-106-107-108-110-115-120-121-122-123-124-130-132-136-140-144-146-149-152-154-175-176
Alur 5	
<i>Test Case</i>	<i>tambahd()</i>
Target yang Diharapkan	Menentukan nilai \$r42 sama dengan 1 jika nilai \$tr sama dengan 'R.42'
Hasil Pengujian	Benar
Alur / Path	97-98-99-100-101-102-103-104-105-106-107-108-110-115-120-125-126-127-128-129-130-132-136-140-144-146-149-152-154-175-176
Alur 6	
<i>Test Case</i>	<i>tambahd()</i>
Target yang Diharapkan	Menentukan nilai \$beton sama dengan 1 jika nilai \$jb sama dengan 'Beton'
Hasil Pengujian	Benar
Alur / Path	97-98-99-100-101-102-103-104-105-106-

	107-108-110-115-120-125-130-132-133-134-135-144-146-149-152-154-175-176
Alur 7	
<i>Test Case</i>	<i>tambahd()</i>
Target yang Diharapkan	Menentukan nilai \$kayu sama dengan 1 jika nilai \$jb sama dengan 'Kayu'
Hasil Pengujian	Benar
Alur / Path	97-98-99-100-101-102-103-104-105-106-107-108-110-115-120-125-130-132-136-137-138-139-144-146-149-152-154-175-176
Alur 8	
<i>Test Case</i>	<i>tambahd()</i>
Target yang Diharapkan	Menentukan nilai \$baja sama dengan 1 jika nilai \$jb sama dengan 'Baja'
Hasil Pengujian	Benar
Alur / Path	97-98-99-100-101-102-103-104-105-106-107-108-110-115-120-125-130-132-136-140-141-142-143-144-146-149-152-154-175-176
Alur 9	
<i>Test Case</i>	<i>tambahd()</i>
Target yang Diharapkan	Menentukan nilai \$tunggal sama dengan 1 jika nilai \$jp sama dengan 'Tunggal'
Hasil Pengujian	Benar
Alur / Path	97-98-99-100-101-102-103-104-105-106-107-108-110-115-120-125-130-132-136-140-144-146-147-148-154-175-176
Alur 10	
<i>Test Case</i>	<i>tambahd()</i>
Target yang Diharapkan	Menentukan nilai \$ganda sama dengan 1 jika nilai \$jp sama dengan 'Ganda'
Hasil Pengujian	Benar
Alur / Path	97-98-99-100-101-102-103-104-105-106-107-108-110-115-120-125-130-132-136-140-144-146-149-150-151-154-175-176
Alur 1	
<i>Test Case</i>	<i>ubah()</i>
Target yang Diharapkan	Menampilkan halaman perubahan data pada <i>dataset</i> beserta data atribut dan kelas yang ditentukan dari nilai <i>id</i>
Hasil Pengujian	Benar
Alur / Path	180-181-185-186-188-189-190-191-194
Alur 2	
<i>Test Case</i>	<i>ubah()</i>
Target yang Diharapkan	Melakukan <i>redirect</i> ke <i>class c_login</i> jika hak akses tidak sesuai

Hasil Pengujian	Benar
Alur / Path	180-181-185-186-188-193-194
Alur 1	
Test Case	<i>ubahd()</i>
Target yang Diharapkan	Mengolah masukan dari <i>form</i> ubah <i>dataset</i> sehingga dikumpulkan dalam sebuah larik sebagai parameter ke <i>model</i>
Hasil Pengujian	Benar
Alur / Path	198-199-200-201-202-203-204-205-206-207-208-209-211-216-221-226-231-233-237-241-245-247-250-253-255-277-278
Alur 2	
Test Case	<i>ubahd()</i>
Target yang Diharapkan	Menentukan nilai \$r60 sama dengan 1 jika nilai \$tr sama dengan 'R.60'
Hasil Pengujian	Benar
Alur / Path	198-199-200-201-202-203-204-205-206-207-208-209-211-212-213-214-215-231-233-237-241-245-247-250-253-255-277-278
Alur 3	
Test Case	<i>ubahd()</i>
Target yang Diharapkan	Menentukan nilai \$r54 sama dengan 1 jika nilai \$tr sama dengan 'R.54'
Hasil Pengujian	Benar
Alur / Path	198-199-200-201-202-203-204-205-206-207-208-209-211-216-217-218-219-220-231-233-237-241-245-247-250-253-255-277-278
Alur 4	
Test Case	<i>ubahd()</i>
Target yang Diharapkan	Menentukan nilai \$r50 sama dengan 1 jika nilai \$tr sama dengan 'R.50'
Hasil Pengujian	Benar
Alur / Path	198-199-200-201-202-203-204-205-206-207-208-209-211-216-221-222-223-224-225-231-233-237-241-245-247-250-253-255-277-278
Alur 5	
Test Case	<i>ubahd()</i>
Target yang Diharapkan	Menentukan nilai \$r42 sama dengan 1 jika nilai \$tr sama dengan 'R.42'
Hasil Pengujian	Benar
Alur / Path	198-199-200-201-202-203-204-205-206-207-208-209-211-216-221-226-227-228-229-230-231-233-237-241-245-247-250-253-255-277-278

Alur 6	
<i>Test Case</i>	<i>ubahd()</i>
Target yang Diharapkan	Menentukan nilai \$beton sama dengan 1 jika nilai \$jb sama dengan 'Beton'
Hasil Pengujian	Benar
Alur / Path	198-199-200-201-202-203-204-205-206-207-208-209-211-216-221-226-231-233-234-235-236-245-247-250-253-255-277-278
Alur 7	
<i>Test Case</i>	<i>ubahd()</i>
Target yang Diharapkan	Menentukan nilai \$kayu sama dengan 1 jika nilai \$jb sama dengan 'Kayu'
Hasil Pengujian	Benar
Alur / Path	198-199-200-201-202-203-204-205-206-207-208-209-211-216-221-226-231-233-237-238-239-240-245-247-250-253-255-277-278
Alur 8	
<i>Test Case</i>	<i>ubahd()</i>
Target yang Diharapkan	Menentukan nilai \$baja sama dengan 1 jika nilai \$jb sama dengan 'Baja'
Hasil Pengujian	Benar
Alur / Path	198-199-200-201-202-203-204-205-206-207-208-209-211-216-221-226-231-233-237-241-242-243-244-245-247-250-253-255-277-278
Alur 9	
<i>Test Case</i>	<i>ubahd()</i>
Target yang Diharapkan	Menentukan nilai \$tunggal sama dengan 1 jika nilai \$jp sama dengan 'Tunggal'
Hasil Pengujian	Benar
Alur / Path	198-199-200-201-202-203-204-205-206-207-208-209-211-216-221-226-231-233-237-241-245-247-248-249-255-277-278
Alur 10	
<i>Test Case</i>	<i>ubahd()</i>
Target yang Diharapkan	Menentukan nilai \$ganda sama dengan 1 jika nilai \$jp sama dengan 'Ganda'
Hasil Pengujian	Benar
Alur / Path	198-199-200-201-202-203-204-205-206-207-208-209-211-216-221-226-231-233-237-241-245-247-250-251-252-255-277-278
Alur 1	
<i>Test Case</i>	<i>hapus()</i>
Target yang	Mengirim variabel <i>id</i> dari <i>dataset</i> yang telah

Diharapkan	ditentukan ke <i>model</i> sebagai parameter untuk dilakukan proses hapus data sesuai dengan <i>id</i> tersebut
Hasil Pengujian	Benar
Alur / Path	282-283-284
Alur 1	
Test Case	<i>cari()</i>
Target yang Diharapkan	Mengolah pencarian beserta hasil pencarian yang dikirim ke halaman <i>dataset</i>
Hasil Pengujian	Benar
Alur / Path	288-289-290-294-296-297-298-299-311-313-314-315-316-317-318-319-321-322-323-324-325-326-327-328-329-330-331-332-333-334-335-336-337-338-339-341-343-345-347-348-349-350-353
Alur 2	
Test Case	<i>cari()</i>
Target yang Diharapkan	Mengelola pencarian khusus untuk pilihan pencarian bernilai 'vmaks'
Hasil Pengujian	Benar
Alur / Path	288-289-290-294-296-297-298-299-300-302-304-306-308-310-311-313-314-315-316-317-318-319-321-322-323-324-325-326-327-328-329-330-331-332-333-334-335-336-337-338-339-341-343-345-347-348-349-350-353
Alur 3	
Test Case	<i>cari()</i>
Target yang Diharapkan	Mengelola pencarian khusus untuk isi pencarian bernilai 1 untuk '120 km/jam'
Hasil Pengujian	Benar
Alur / Path	288-289-290-294-296-297-298-299-300-301-311-313-314-315-316-317-318-319-321-322-323-324-325-326-327-328-329-330-331-332-333-334-335-336-337-338-339-341-343-345-347-348-349-350-353
Alur 4	
Test Case	<i>cari()</i>
Target yang Diharapkan	Mengelola pencarian khusus untuk isi pencarian bernilai 0,75 untuk '110 km/jam'
Hasil Pengujian	Benar
Alur / Path	288-289-290-294-296-297-298-299-300-302-303-311-313-314-315-316-317-318-319-321-322-323-324-325-326-327-328-329-330-331-332-333-334-335-336-337-338-339-341-343-345-347-348-349-350-353

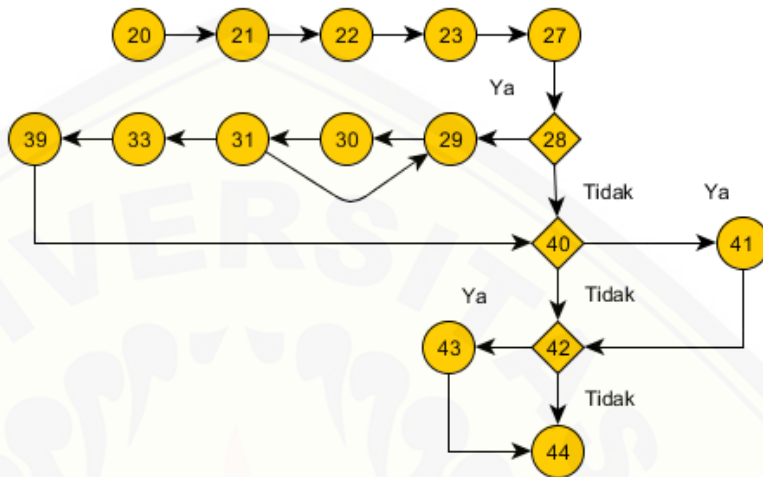
Alur 5	
<i>Test Case</i>	<i>cari()</i>
Target yang Diharapkan	Mengelola pencarian khusus untuk isi pencarian bernilai 0,5 untuk '100 km/jam'
Hasil Pengujian	Benar
Alur / Path	288-289-290-294-296-297-298-299-300-302-304-305-311-313-314-315-316-317-318-319-321-322-323-324-325-326-327-328-329-330-331-332-333-334-335-336-337-338-339-341-343-345-347-348-349-350-353
Alur 6	
<i>Test Case</i>	<i>cari()</i>
Target yang Diharapkan	Mengelola pencarian khusus untuk isi pencarian bernilai 0,25 untuk '90 km/jam'
Hasil Pengujian	Benar
Alur / Path	288-289-290-294-296-297-298-299-300-302-304-306-307-311-313-314-315-316-317-318-319-321-322-323-324-325-326-327-328-329-330-331-332-333-334-335-336-337-338-339-341-343-345-347-348-349-350-353
Alur 7	
<i>Test Case</i>	<i>cari()</i>
Target yang Diharapkan	Mengelola pencarian khusus untuk isi pencarian bernilai 0 untuk '80 km/jam'
Hasil Pengujian	Benar
Alur / Path	288-289-290-294-296-297-298-299-300-302-304-306-308-309-311-313-314-315-316-317-318-319-321-322-323-324-325-326-327-328-329-330-331-332-333-334-335-336-337-338-339-341-343-345-347-348-349-350-353
Alur 8	
<i>Test Case</i>	<i>cari()</i>
Target yang Diharapkan	Melakukan <i>redirect</i> ke <i>class c_login</i> jika hak akses tidak sesuai
Hasil Pengujian	Benar
Alur / Path	288-289-290-294-296-297-298-299-311-313-314-315-316-317-318-319-321-322-323-324-325-326-327-328-329-330-331-332-333-334-335-336-337-338-339-341-343-345-347-352-353

7. Berkas *c_login.php*

1. Daftar Method

Method-method lain yang digunakan pada alur tersebut sebagai berikut: *cek_login(\$nipp, \$ks)* dan *logout()*

2. Diagram Alir Method

Gambar D.0.41 Diagram Alir *cek_login(\$nipp, \$ks)*Gambar D.0.42 Diagram Alir *logout()*3. Penghitungan *Cyclomatic Complexity (CC)*

cek_login(\$nipp, \$ks) : $V(G) = 19 - 16 + 2 = 5$

logout() : $V(G) = 1 - 2 + 2 = 1$

4. Basis Set

cek_login(\$nipp, \$ks) : 20-21-22-23-27-28-29-30-31-29-30-31-33-39-40-42-44 (Alur 1)

20-21-22-23-27-28-29-30-31-33-39-40-42-44 (Alur 2)

20-21-22-23-27-28-40-41-42-44 (Alur 3)

20-21-22-23-27-28-40-42-43-44 (Alur 4)

20-21-22-23-27-28-40-42-44 (Alur
5)

logout() : 48-49 (Alur 1)

5. Kasus Uji atau *Test Case*

Alur 1	
<i>Test Case</i>	<i>cek_login(\$nipp, \$ks)</i>
Target yang Diharapkan	Melakukan pengulangan yang menggunakan variabel \$nalar sebanyak satu kali untuk mendapatkan nilai KodeJabatan dan Willd
Hasil Pengujian	Benar
Alur / Path	20-21-22-23-27-28-29-30-31-29-30-31-33-39-40-42-44
Alur 2	
<i>Test Case</i>	<i>cek_login(\$nipp, \$ks)</i>
Target yang Diharapkan	Melakukan proses validasi <i>login</i> terhadap kombinasi NIPP dan kata sandi yang apabila hal tersebut berhasil maka pengguna masuk atau <i>redirect</i> ke <i>c_utama</i>
Hasil Pengujian	Benar
Alur / Path	20-21-22-23-27-28-29-30-31-33-39-40-42-44
Alur 3	
<i>Test Case</i>	<i>cek_login(\$nipp, \$ks)</i>
Target yang Diharapkan	Mengirim nilai variabel \$data jika \$nipp atau \$ks bernilai kosong
Hasil Pengujian	Benar
Alur / Path	20-21-22-23-27-28-40-41-42-44
Alur 4	
<i>Test Case</i>	<i>cek_login(\$nipp, \$ks)</i>
Target yang Diharapkan	Mengirim nilai variabel \$data2 jika \$nipp atau \$ks, keduanya tidak bernilai kosong serta nilai <i>count(\$nalar)</i> tidak bernilai 1
Hasil Pengujian	Benar
Alur / Path	20-21-22-23-27-28-40-42-43-44
Alur 5	
<i>Test Case</i>	<i>cek_login(\$nipp, \$ks)</i>
Target yang Diharapkan	Proses permisalan tersebut selesai
Hasil Pengujian	Benar
Alur / Path	20-21-22-23-27-28-40-42-44
Alur 1	
<i>Test Case</i>	<i>logout()</i>
Target yang Diharapkan	Melakukan proses <i>logout</i> yang selanjutnya menampilkan halaman <i>login</i>

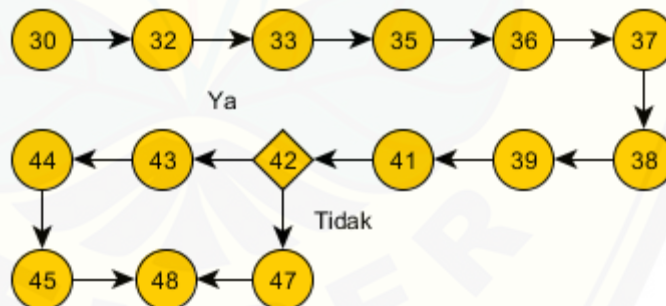
Hasil Pengujian	Benar
Alur / Path	48-49

8. Berkas *c_peta.php*

1. Daftar Method

Method-method yang digunakan pada alur tersebut sebagai berikut: *validasi()*, *tambah()*, *ubah()*, dan *lintas()*

2. Diagram Alir Method

Gambar D.0.43 Diagram Alir *validasi()*Gambar D.0.44 Diagram Alir *tambah()*Gambar D.0.45 Diagram Alir *ubah()*Gambar D.0.46 Diagram Alir *lintas()*3. Penghitungan *Cyclomatic Complexity (CC)*

validasi() : $V(G) = 2 - 3 + 2 = 1$

tambah() : $V(G) = 2 - 3 + 2 = 1$

ubah() : $V(G) = 2 - 3 + 2 = 1$

lintas() : $V(G) = 15 - 15 + 2 = 2$

4. Basis Set

<i>validasi()</i>	: 12-13-14 (Alur 1)
<i>tambah()</i>	: 18-19-20 (Alur 1)
<i>ubah()</i>	: 24-25-26 (Alur 1)
<i>lintas()</i>	: 30-32-33-35-36-37-38-39-41-42-43-44-45-48 (Alur 1)
	30-32-33-35-36-37-38-39-41-42-47-48 (Alur 2)

5. Kasus Uji atau *Test Case*

Alur 1	
<i>Test Case</i>	<i>validasi()</i>
Target yang Diharapkan	Mengirim variabel tanda tervalidasi-nya klasifikasi kecepatan maksimum suatu jalur sebagai parameter ke <i>model</i>
Hasil Pengujian	Benar
Alur / Path	12-13-14
Alur 1	
<i>Test Case</i>	<i>tambah()</i>
Target yang Diharapkan	Mengirim variabel tanda dimunculkan data peta jalur yang ditentukan di tabel tinjauan terkini data lintas ke <i>model</i>
Hasil Pengujian	Benar
Alur / Path	18-19-20
Alur 1	
<i>Test Case</i>	<i>ubah()</i>
Target yang Diharapkan	Mengirim variabel status “ubah” pada data peta jalur yang ditentukan sebagai parameter ke <i>model</i>
Hasil Pengujian	Benar
Alur / Path	24-25-26
Alur 1	
<i>Test Case</i>	<i>lintas()</i>
Target yang Diharapkan	Menampilkan halaman tampil peta beserta dengan data-data jalur kereta api beserta stasiun-stasiun, koordinat masing-masing stasiun, dan nama lintas atau jalur
Hasil Pengujian	Benar
Alur / Path	30-32-33-35-36-37-38-39-41-42-43-44-45-48
Alur 2	
<i>Test Case</i>	<i>lintas()</i>
Target yang Diharapkan	Kembali ke <i>method c_login</i>

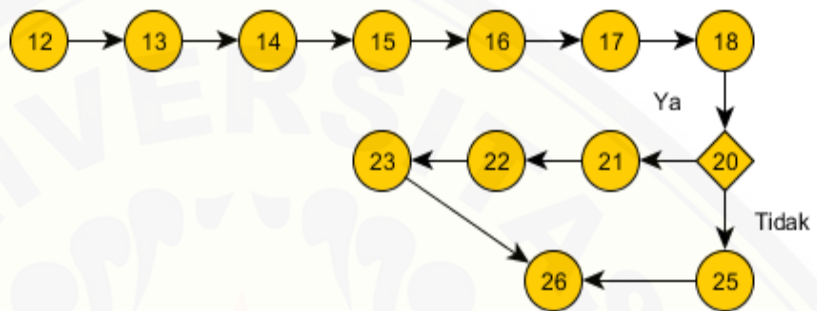
Hasil Pengujian	Benar
Alur / Path	30-32-33-35-36-37-38-39-41-42-47-48

9. Berkas *c_rancang.php*

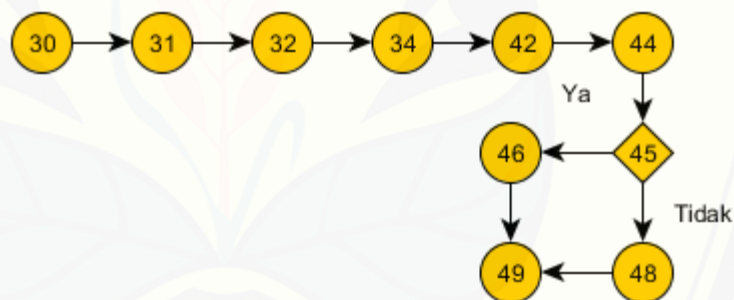
1. Daftar Method

Method-method yang digunakan pada alur tersebut sebagai berikut: *jalur()*, *postdata()*, *rancang()*, dan *selesai()*

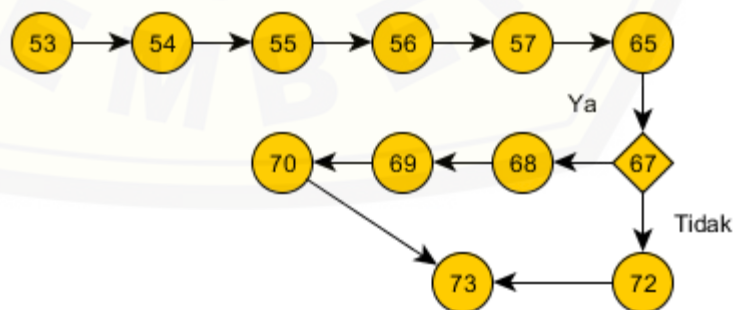
2. Diagram Alir Method



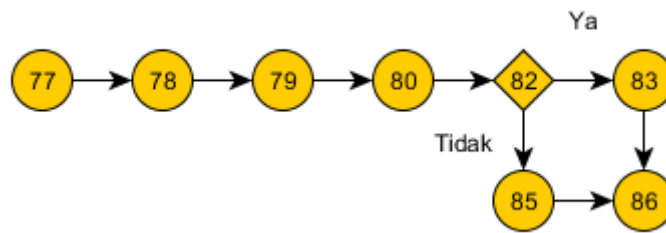
Gambar D.0.47 Diagram Alir *jalur()*



Gambar D.0.48 Diagram Alir *postdata()*



Gambar D.0.49 Diagram Alir *rancang()*



Gambar D.0.50 Diagram Alir *selesai()*

3. Penghitungan *Cyclomatic Complexity* (CC)

jalur() : $V(G) = 13 - 13 + 2 = 2$

postdata() : $V(G) = 10 - 10 + 2 = 2$

rancang() : $V(G) = 12 - 12 + 2 = 2$

selesai() : $V(G) = 8 - 8 + 2 = 2$

4. Basis Set

jalur() : 12-13-14-15-16-17-18-20-21-22-23-26
(Alur 1)

12-13-14-15-16-17-18-20-25-26 (Alur 2)

postdata() : 30-31-32-34-42-44-45-46-49 (Alur 1)

30-31-32-34-42-44-45-48-49 (Alur 2)

rancang() : 53-54-55-56-57-65-67-68-69-70-73 (Alur 1)

53-54-55-56-57-65-67-72-73 (Alur 2)

selesai() : 77-78-79-80-82-83-86 (Alur 1)

77-78-79-80-82-85-86 (Alur 2)

5. Kasus Uji atau *Test Case*

Alur 1	
<i>Test Case</i>	<i>jalur()</i>
Target yang Diharapkan	Menampilkan halaman <i>form</i> nilai kriteria jalur beserta nama jalur kereta api yang telah ditentukan dari <i>view</i>
Hasil Pengujian	Benar
Alur / Path	12-13-14-15-16-17-18-20-21-22-23-26
Alur 2	
<i>Test Case</i>	<i>jalur()</i>
Target yang Diharapkan	Kembali ke <i>method c_login</i>

Hasil Pengujian	Benar
Alur / Path	12-13-14-15-16-17-18-20-25-26
Alur 1	
Test Case	<i>postdata()</i>
Target yang Diharapkan	Mengirim data dari <i>form</i> nilai kriteria jalur dari <i>view</i> sebagai parameter di mana proses selanjutnya diarahkan tampilan <i>web</i> ke halaman rancang
Hasil Pengujian	Benar
Alur / Path	30-31-32-34-42-44-45-46-49
Alur 2	
Test Case	<i>postdata()</i>
Target yang Diharapkan	Kembali ke <i>method c_login</i>
Hasil Pengujian	Benar
Alur / Path	30-31-32-34-42-44-45-48-49
Alur 1	
Test Case	<i>rancang()</i>
Target yang Diharapkan	Menampilkan halaman rancang beserta dengan data-data yang dibutuhkan seperti data nilai kriteria lintas, jalur kereta api beserta stasiun-stasiun dan koordinat masing-masing stasiun, serta nama lintas atau jalur
Hasil Pengujian	Benar
Alur / Path	53-54-55-56-57-65-67-68-69-70-73
Alur 2	
Test Case	<i>rancang()</i>
Target yang Diharapkan	Kembali ke <i>method c_login</i>
Hasil Pengujian	Benar
Alur / Path	53-54-55-56-57-65-67-72-73
Alur 1	
Test Case	<i>selesai()</i>
Target yang Diharapkan	Mengirim variabel tanda selesainya klasifikasi kecepatan maksimum suatu jalur dan nilai wilayah dari tersebut sebagai parameter ke <i>model</i>
Hasil Pengujian	Benar
Alur / Path	77-78-79-80-82-83-86
Alur 2	
Test Case	<i>selesai()</i>
Target yang Diharapkan	Kembali ke <i>method c_login</i>
Hasil Pengujian	Benar
Alur / Path	77-78-79-80-82-85-86

D.2. Pengujian *Black Box*

1. Fitur *Login* Sistem untuk Asisten Manajer Konstruksi Jalan dan Jembatan yang berfungsi untuk melakukan *login* untuk pengguna tersebut sehingga dapat menampilkan halaman utama pengguna tersebut

Aksi	Hasil	Keterangan
NIPP dan kata sandi telah terisi, klik "Masuk", memiliki hak akses	Melakukan proses <i>login</i> yang selanjutnya menampilkan halaman utama Asisten Manajer Konstruksi Jalan Rel dan Jembatan berupa data status perancangan lintas	✓
NIPP dan kata sandi belum terisi, klik "Masuk"	Menampilkan pesan "NIPP atau kata sandi belum terisi."	✓
NIPP dan kata sandi telah terisi, klik "Masuk", ketidaksesuaian NIPP dan kata sandi	Menampilkan pesan "Kombinasi NIPP dengan kata sandinya salah."	✓
NIPP dan kata sandi telah terisi, klik "Masuk", tidak memiliki hak akses	Mengalihkan ke halaman <i>web</i> PT. KAI	✓

2. Fitur *Login* Sistem untuk Manajer Jalan dan Jembatan yang berfungsi untuk melakukan *login* untuk pengguna tersebut sehingga dapat menampilkan halaman utama pengguna tersebut

Aksi	Hasil	Keterangan
NIPP dan kata sandi telah terisi, klik "Masuk", memiliki hak akses	Melakukan proses <i>login</i> yang selanjutnya menampilkan halaman utama Manajer Jalan dan Jembatan berupa menu tampil, tambah, ubah, dan validasi data peta	✓
NIPP dan kata sandi belum terisi, klik "Masuk"	Menampilkan pesan "NIPP atau kata sandi belum terisi."	✓
NIPP dan kata sandi telah terisi, klik "Masuk",	Menampilkan pesan "Kombinasi NIPP dengan kata sandinya salah."	✓

ketidaksesuaian NIPP dan kata sandi		
NIPP dan kata sandi telah terisi, klik “Masuk”, tidak memiliki hak akses	Mengalihkan ke halaman <i>web</i> PT. KAI	✓

3. Fitur *Login* Sistem untuk Kepala Subdirektorat Jalan dan Jembatan Pusat yang berfungsi untuk melakukan *login* untuk pengguna tersebut sehingga dapat menampilkan halaman utama pengguna tersebut

Aksi	Hasil	Keterangan
NIPP dan kata sandi telah terisi, klik “Masuk”, memiliki hak akses	Melakukan proses <i>login</i> yang selanjutnya menampilkan halaman utama Kepala Sub Direktorat Jalan Rel dan Jembatan Pusat berupa data penghitungan terakhir metode SLP	✓
NIPP dan kata sandi belum terisi, klik “Masuk”	Menampilkan pesan “NIPP atau kata sandi belum terisi.”	✓
NIPP dan kata sandi telah terisi, klik “Masuk”, ketidaksesuaian NIPP dan kata sandi	Menampilkan pesan “Kombinasi NIPP dengan kata sandinya salah.”	✓
NIPP dan kata sandi telah terisi, klik “Masuk”, tidak memiliki hak akses	Mengalihkan ke halaman <i>web</i> PT. KAI	✓

4. Fitur Menampilkan Tabel *Dataset* yang berfungsi untuk menampilkan data-data pada tabel *dataset*

Aksi	Hasil	Keterangan
Klik “ <i>Dataset</i> ”	Menampilkan halaman <i>dataset</i>	✓
Klik “Tampilkan Semua”	Menampilkan semua data pada tabel <i>dataset</i>	✓

5. Fitur Menambah Data pada *Dataset* yang berfungsi untuk menambah data dari ke basis data

Aksi	Hasil	Keterangan
Klik “Tambah <i>Dataset</i> ”	Menampilkan halaman tambah <i>dataset</i>	✓
<i>Form</i> tambah <i>dataset</i> telah terisi, klik “Simpan”	Menyimpan data <i>dataset</i> tersebut ke basis data, lalu menampilkan halaman <i>dataset</i>	✓
<i>Form</i> tambah <i>dataset</i> telah terisi, klik “Hapus Masukan”	Menghapus isian <i>form</i> tambah <i>dataset</i>	✓
<i>Form</i> tambah <i>dataset</i> tidak terisi sepenuhnya, klik “Simpan”	Menampilkan pesan “Please fill out this field.”	✓

6. Fitur Mengubah Data pada *Dataset* yang berfungsi untuk mengubah data pada *dataset* yang menjadikan perubahan data tersebut pada basis data sesuai dengan pilihan pengguna

Aksi	Hasil	Keterangan
Klik “Ubah”	Menampilkan halaman <i>form</i> ubah <i>dataset</i>	✓
<i>Form</i> ubah <i>dataset</i> telah terisi, klik “Ubah”	Menyimpan perubahan data <i>dataset</i> tersebut ke basis data, lalu menampilkan halaman <i>dataset</i>	✓
<i>Form</i> ubah <i>dataset</i> telah terisi, klik “Hapus Masukan”	Mengembalikan isian <i>form</i> ubah <i>dataset</i>	✓
<i>Form</i> ubah <i>dataset</i> tidak terisi sepenuhnya, klik “Ubah”	Menampilkan pesan “Please fill out this field.”	✓

7. Fitur Menghapus Data pada *Dataset* yang berfungsi untuk menghapus data dari *dataset* pada basis data sesuai dengan pilihan pengguna

Aksi	Hasil	Keterangan
Klik "Hapus"	Menampilkan peringatan "Apakah data <i>dataset</i> tersebut dihapus?"	✓
Klik "Ya"	Menghapus data <i>dataset</i> pada basis data, lalu kembali ke halaman <i>dataset</i>	✓
Klik "Tidak"	Kembali ke halaman <i>dataset</i>	✓

8. Fitur Pencarian Data pada *Dataset* yang berfungsi untuk melakukan pencarian sesuai dengan kata kunci dan pilihan atribut dari pengguna

Aksi	Hasil	Keterangan
<i>Form</i> pencarian terisi, klik "Cari"	Menampilkan hasil pencarian pada tabel <i>dataset</i>	✓
<i>Form</i> pencarian tidak terisi, klik "Cari"	Menampilkan pesan "Data pencarian belum terisi."	✓

9. Fitur Memproses Klasifikasi *Dataset* yang berfungsi untuk mengolah data dari *dataset*, nilai epoch, dan nilai *learning rate* untuk dilakukan klasifikasi SLP

Aksi	Hasil	Keterangan
<i>Form</i> nilai parameter SLP terisi, klik "Hitung"	Menetapkan bobot <i>perceptron</i> sesuai dengan penghitungan klasifikasi terakhir, lalu menampilkan hasil klasifikasi <i>dataset</i>	✓
<i>Form</i> nilai parameter SLP tidak terisi, klik "Hitung"	Menampilkan pesan "Please fill out this field."	✓
Nilai akurasi penghitungan terakhir lebih kecil daripada penghitungan sebelumnya	Menampilkan pesan "Nilai akurasi yang didapat lebih kecil daripada sebelumnya. Silakan atur kembali parameter SLP anda."	✓

10. Fitur Menambah Data Peta ke Tabel Status Perancangan yang berfungsi untuk menambah data peta oleh Manajer Jalan dan Jembatan untuk proses selanjutnya pada tabel status perancangan oleh Asisten Manajer Konstruksi Jalan dan Jembatan

Aksi	Hasil	Keterangan
Klik “Tambah”	Menampilkan peringatan “Apakah data peta tersebut perlu dikelola di SIKEMAKA?”	✓
Klik “Ya”	Data peta berstatus tambah di basis data, lalu menampilkan halaman utama Manajer Jalan dan Jembatan	✓
Klik “Tidak	Menampilkan halaman utama Manajer Jalan dan Jembatan	✓

11. Fitur Menandai Data Peta untuk Diubah di Sistem Lain yang berfungsi untuk menandai data peta oleh Manajer Jalan dan Jembatan bahwa data peta tersebut tidak dapat dikelola pada SIKEMAKA

Aksi	Hasil	Keterangan
Klik “Ubah”	Menampilkan peringatan “Apakah data peta tersebut perlu diubah? SIKEMAKA akan menyampaikan pesan ke e-Office untuk dilakukan peninjauan ulang pada data peta tersebut. Catatan: data peta tersebut tidak dapat digunakan di SIKEMAKA, kecuali telah disetujui oleh Kepala DAOP.”	✓
Klik “Ya”	Data peta berstatus ubah di basis data, lalu menampilkan halaman utama Manajer Jalan dan Jembatan	✓
Klik “Tidak	Menampilkan halaman utama Manajer Jalan dan Jembatan	✓

12. Fitur Menampilkan Peta dari Data Peta yang Dipilih yang berfungsi untuk menampilkan peta dari data peta yang dipilih oleh Manajer Jalan dan Jembatan

Aksi	Hasil	Keterangan
Klik lintas pada tabel data peta	Menampilkan halaman peta lintas yang telah dipilih	✓
Klik "Halaman Utama"	Menampilkan halaman utama Manajer Jalan dan Jembatan	✓

13. Fitur Meninjau Tabel Status Perancangan yang berfungsi untuk menampilkan status data peta untuk proses memulai klasifikasi SLP, pengerjaan klasifikasi SLP, dan penyelesaian klasifikasi SLP

Aksi	Hasil	Keterangan
Klik "Mulai"	Menampilkan halaman <i>form</i> nilai kriteria lintas yang baru	✓
Klik "Proses Perancangan"	Menampilkan halaman <i>form</i> nilai kriteria ruas lintas yang dalam proses pengerjaan	✓
Klik "Belum Tervalidasi"	Menampilkan halaman <i>form</i> nilai kriteria ruas lintas yang selesai pada pengerjaan	✓

14. Fitur Mengisi Nilai Kriteria Lintas yang berfungsi untuk memberikan nilai kriteria lintas pada seluruh ruas lintas untuk lintas yang telah ditentukan

Aksi	Hasil	Keterangan
<i>Form</i> nilai kriteria lintas terisi, klik "Lanjut"	Menyimpan data nilai kriteria lintas pada basis data, lalu menampilkan halaman <i>form</i> nilai kriteria ruas lintas	✓
<i>Form</i> nilai kriteria lintas tidak terisi, klik "Lanjut"	Menampilkan pesan "Please fill out this field."	✓

15. Fitur Melakukan Klasifikasi per Ruas Lintas yang berfungsi untuk melakukan proses klasifikasi SLP untuk setiap ruas lintas pada lintas yang telah ditentukan

Aksi	Hasil	Keterangan
Ruas lintas dipilih, lalu <i>form</i> nilai kriteria ruas lintas yang telah dipilih terakhir menjadi terisi, klik “Simpan”	Menyimpan data nilai kriteria lintas dan ruas lintas pada basis data, lalu menampilkan rancang Asisten Manajer Konstruksi Jalan Rel dan Jembatan	✓
<i>Form</i> nilai kriteria ruas lintas yang telah dipilih terakhir tidak terisi, klik “Simpan”	Menampilkan pesan “Terdapat data yang belum terisi.”	✓
<i>Form</i> nilai kriteria ruas lintas yang telah dipilih terakhir terisi, klik “Reset”	Mengembalikan nilai awal sesuai dengan nilai kriteria tersebut.	✓

16. Fitur Menyelesaikan Klasifikasi yang berfungsi untuk melakukan penyelesaian klasifikasi setelah seluruh ruas lintas memiliki nilai kecepatan maksimum

Aksi	Hasil	Keterangan
Klik “Selesai”	Menampilkan halaman utama Asisten Manajer Konstruksi Jalan Rel dan Jembatan	✓

17. Fitur Melakukan Validasi Perancangan Kecepatan Maksimum pada Data Peta yang berfungsi untuk melakukan proses validasi untuk data peta yang telah selesai dalam klasifikasi SLP

Aksi	Hasil	Keterangan
Klik “Belum Tervalidasi”	Menampilkan peringatan “Apakah data peta tersebut divalidasi? Catatan: Data peta yang telah divalidasi tidak dapat diubah validasinya.”	✓
Klik “Ya”	Data peta berstatus tervalidasi di basis data, lalu menampilkan halaman	✓

	utama Manajer Jalan dan Jembatan	
Klik “Tidak”	Menampilkan halaman utama Manajer Jalan dan Jembatan	✓

18. Fitur Mencetak Laporan Desain Kecepatan Maksimum yang berfungsi untuk menampilkan menu cetak untuk laporan tersebut

Aksi	Hasil	Keterangan
Klik “Tervalidasi”	Menampilkan halaman laporan peta jalur	✓

19. Fitur *Logout* Sistem yang berfungsi untuk keluar dari sistem sehingga menampilkan kembali halaman *login*

Aksi	Hasil	Keterangan
Klik “Logout”	Melakukan proses <i>logout</i> , yang selanjutnya menampilkan halaman <i>login</i>	✓

LAMPIRAN E. CONTOH LAPORAN PETA HASIL KLASIFIKASI

LAMPIRAN HASIL KLASIFIKASI KECEPATAN MAKSIMUM JALUR KLAKAH-PASIRIAN

http://127.0.0.1/SK1243/index.php/c_cetak/laporan/03





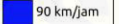
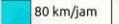




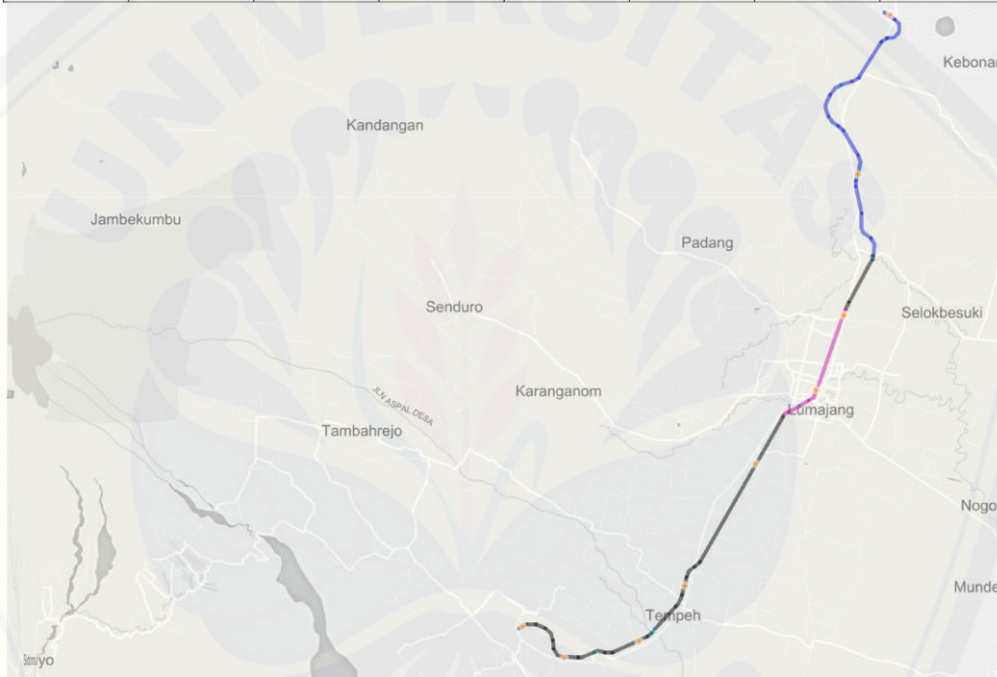
LAPORAN
HASIL KECEPATAN MAKSIMUM JALUR KERETA API KLAKAH-PASIRIAN

ID Ruas	Ruas Lintas	Kecepatan Maksimum
1	Area Terminus/Klakah	90 km/jam
2	Ruas Lintas 1	90 km/jam
3	Ruas Lintas 2	90 km/jam
4	Ruas Lintas 3	90 km/jam
5	Ruas Lintas 4	90 km/jam
6	Ruas Lintas 5	90 km/jam
7	Ruas Lintas 6	80 km/jam
8	Ruas Lintas 7	90 km/jam
9	Ruas Lintas 8	90 km/jam
10	Ruas Lintas 9	90 km/jam
11	Ruas Lintas 10	90 km/jam
12	Ruas Lintas 11	90 km/jam
13	Ruas Lintas 12	90 km/jam
14	Ruas Lintas 13	90 km/jam
15	Ruas Lintas 14	90 km/jam
16	Ruas Lintas 15	90 km/jam
17	Ruas Lintas 16	90 km/jam
18	Ruas Lintas 17	90 km/jam
19	Ruas Lintas 18	80 km/jam
20	Ruas Lintas 19	100 km/jam
21	Ruas Lintas 20	80 km/jam
22	Ruas Lintas 21	100 km/jam
23	Ruas Lintas 22	100 km/jam
24	Ruas Lintas 23	110 km/jam
25	Ruas Lintas 24	110 km/jam
26	Ruas Lintas 25	110 km/jam
27	Ruas Lintas 26	80 km/jam
28	Ruas Lintas 27	110 km/jam
29	Ruas Lintas 28	110 km/jam
30	Ruas Lintas 29	100 km/jam
31	Ruas Lintas 30	100 km/jam
32	Ruas Lintas 31	100 km/jam
33	Ruas Lintas 32	100 km/jam
34	Ruas Lintas 33	100 km/jam
35	Ruas Lintas 34	100 km/jam
36	Ruas Lintas 35	80 km/jam
37	Ruas Lintas 36	100 km/jam
38	Ruas Lintas 37	100 km/jam
39	Ruas Lintas 38	100 km/jam
40	Ruas Lintas 39	100 km/jam
41	Ruas Lintas 40	80 km/jam
42	Ruas Lintas 41	100 km/jam

ID Ruas	Ruas Lintas	Kecepatan Maksimum
43	Ruas Lintas 42	100 km/jam
44	Ruas Lintas 43	100 km/jam
45	Ruas Lintas 44	100 km/jam
46	Ruas Lintas 45	100 km/jam
47	Ruas Lintas 46	100 km/jam
48	Ruas Lintas 47	100 km/jam
49	Area Terminus / Pasirian	100 km/jam

Asisten Manajer Konstruksi Jalan Rel dan Jembatan DAOP IX	Manajer Jalan dan Jembatan DAOP IX	Kepala DAOP IX
Satya Laksana NIPP: 20101	Eka Wira Bhakti NIPP: 10901	Haryanto NIPP: 32767
Selesai Perancangan: Selasa, 02 Mei 2017 09:05:26	Tervalidasi: Selasa, 02 Mei 2017 09:07:01	Tanggal:

 Stasiun	 120 km/jam	 110 km/jam	 100 km/jam	 90 km/jam	 80 km/jam	 70 km/jam	 60 km/jam
---	--	--	--	---	--	---	---



Leaflet (<http://leafletjs.com>) | Sumber petaan © Mapbox (<http://mapbox.com>), CC-BY-SA (<http://creativecommons.org/licenses/by-sa/2.0/>)

LAMPIRAN F. HASIL KLASIFIKASI DATASET

Hasil bobot dan bias:

Bobot ke-0: 0.8644690483322
Bobot ke-1: -0.30772515156507
Bobot ke-2: -0.25918505295832
Bobot ke-3: -0.41077229905433
Bobot ke-4: 0.076650723348786
Bobot ke-5: -0.051047275681147
Bobot ke-6: -0.13881690291316
Bobot ke-7: 0.30694046319169
Bobot ke-8: -0.42015391843721
Bobot ke-9: 0.264515637985
Bobot ke-10: 1.2862110059297
Bobot ke-11: -1.3415304744576
Bobot ke-12: 0.026960583675785
Bobot ke-13: 0.06078484040598
Bobot ke-14: 0.070941338951397
Bobot ke-15: 0.84702029490635
Bias: -0.11321345524552

Rincian hasil prediksi:

Data ke-1. 120 km/jam || Hasil: 0.96484375884762 - Hasil Prediksi: 120 km/jam
Data ke-2. 120 km/jam || Hasil: 0.95862249435954 - Hasil Prediksi: 120 km/jam
Data ke-3. 120 km/jam || Hasil: 0.96574689615492 - Hasil Prediksi: 120 km/jam
Data ke-4. 120 km/jam || Hasil: 0.95967877232906 - Hasil Prediksi: 120 km/jam
Data ke-5. 120 km/jam || Hasil: 0.9668483401201 - Hasil Prediksi: 120 km/jam
Data ke-6. 120 km/jam || Hasil: 0.96096745665934 - Hasil Prediksi: 120 km/jam
Data ke-7. 120 km/jam || Hasil: 0.96717234352119 - Hasil Prediksi: 120 km/jam
Data ke-8. 120 km/jam || Hasil: 0.96134663828504 - Hasil Prediksi: 120 km/jam
Data ke-9. 120 km/jam || Hasil: 0.89472776787915 - Hasil Prediksi: 120 km/jam
Data ke-10. 120 km/jam || Hasil: 0.87767186852499 - Hasil Prediksi: 120 km/jam
Data ke-11. 120 km/jam || Hasil: 0.89724029424311 - Hasil Prediksi: 120 km/jam
Data ke-12. 120 km/jam || Hasil: 0.88053711532044 - Hasil Prediksi: 120 km/jam
Data ke-13. 120 km/jam || Hasil: 0.90031725918699 - Hasil Prediksi: 120 km/jam
Data ke-14. 120 km/jam || Hasil: 0.88404959440854 - Hasil Prediksi: 120 km/jam
Data ke-15. 120 km/jam || Hasil: 0.90122506645368 - Hasil Prediksi: 120 km/jam
Data ke-16. 120 km/jam || Hasil: 0.88508664150488 - Hasil Prediksi: 120 km/jam
Data ke-17. 110 km/jam || Hasil: 0.83555024730956 - Hasil Prediksi: 110 km/jam
Data ke-18. 110 km/jam || Hasil: 0.81093259704624 - Hasil Prediksi: 110 km/jam
Data ke-19. 110 km/jam || Hasil: 0.83922136054837 - Hasil Prediksi: 110 km/jam
Data ke-20. 110 km/jam || Hasil: 0.8150316085531 - Hasil Prediksi: 110 km/jam
Data ke-21. 110 km/jam || Hasil: 0.8424437246021 - Hasil Prediksi: 110 km/jam
Data ke-22. 110 km/jam || Hasil: 0.81863401458412 - Hasil Prediksi: 110 km/jam
Data ke-23. 110 km/jam || Hasil: 0.84363312494638 - Hasil Prediksi: 110 km/jam
Data ke-24. 110 km/jam || Hasil: 0.81996474397748 - Hasil Prediksi: 110 km/jam
Data ke-25. 110 km/jam || Hasil: 0.84211178930559 - Hasil Prediksi: 110 km/jam
Data ke-26. 110 km/jam || Hasil: 0.81826273882233 - Hasil Prediksi: 110 km/jam
Data ke-27. 110 km/jam || Hasil: 0.84566348138348 - Hasil Prediksi: 110 km/jam
Data ke-28. 110 km/jam || Hasil: 0.82223766780877 - Hasil Prediksi: 110 km/jam

Data ke-29. 110 km/jam	Hasil: 0.8487799961624 - Hasil Prediksi: 110 km/jam
Data ke-30. 110 km/jam	Hasil: 0.82572973845242 - Hasil Prediksi: 110 km/jam
Data ke-31. 110 km/jam	Hasil: 0.84993008444072 - Hasil Prediksi: 110 km/jam
Data ke-32. 110 km/jam	Hasil: 0.82701940582801 - Hasil Prediksi: 110 km/jam
Data ke-33. 110 km/jam	Hasil: 0.81724424565713 - Hasil Prediksi: 110 km/jam
Data ke-34. 110 km/jam	Hasil: 0.79057309781677 - Hasil Prediksi: 110 km/jam
Data ke-35. 110 km/jam	Hasil: 0.82123658432516 - Hasil Prediksi: 110 km/jam
Data ke-36. 110 km/jam	Hasil: 0.79500192440322 - Hasil Prediksi: 110 km/jam
Data ke-37. 110 km/jam	Hasil: 0.82474412917113 - Hasil Prediksi: 110 km/jam
Data ke-38. 110 km/jam	Hasil: 0.79889817217698 - Hasil Prediksi: 110 km/jam
Data ke-39. 110 km/jam	Hasil: 0.82603955379799 - Hasil Prediksi: 110 km/jam
Data ke-40. 110 km/jam	Hasil: 0.80033839071944 - Hasil Prediksi: 110 km/jam
Data ke-41. 110 km/jam	Hasil: 0.82438267818346 - Hasil Prediksi: 110 km/jam
Data ke-42. 110 km/jam	Hasil: 0.79849643966091 - Hasil Prediksi: 110 km/jam
Data ke-43. 110 km/jam	Hasil: 0.82825184799177 - Hasil Prediksi: 110 km/jam
Data ke-44. 110 km/jam	Hasil: 0.80279950406704 - Hasil Prediksi: 110 km/jam
Data ke-45. 110 km/jam	Hasil: 0.83164996242976 - Hasil Prediksi: 110 km/jam
Data ke-46. 110 km/jam	Hasil: 0.80658360531997 - Hasil Prediksi: 110 km/jam
Data ke-47. 110 km/jam	Hasil: 0.83290468367706 - Hasil Prediksi: 110 km/jam
Data ke-48. 110 km/jam	Hasil: 0.80798201269834 - Hasil Prediksi: 110 km/jam
Data ke-49. 100 km/jam	Hasil: 0.49769383815517 - Hasil Prediksi: 100 km/jam
Data ke-50. 100 km/jam	Hasil: 0.4694977352819 - Hasil Prediksi: 100 km/jam
Data ke-51. 100 km/jam	Hasil: 0.50274890354426 - Hasil Prediksi: 100 km/jam
Data ke-52. 100 km/jam	Hasil: 0.4745369682123 - Hasil Prediksi: 100 km/jam
Data ke-53. 100 km/jam	Hasil: 0.50827270673519 - Hasil Prediksi: 100 km/jam
Data ke-54. 100 km/jam	Hasil: 0.48005004882822 - Hasil Prediksi: 100 km/jam
Data ke-55. 100 km/jam	Hasil: 0.51019809204939 - Hasil Prediksi: 100 km/jam
Data ke-56. 100 km/jam	Hasil: 0.4819733134567 - Hasil Prediksi: 100 km/jam
Data ke-57. 100 km/jam	Hasil: 0.50982758061194 - Hasil Prediksi: 100 km/jam
Data ke-58. 100 km/jam	Hasil: 0.48160314511961 - Hasil Prediksi: 100 km/jam
Data ke-59. 100 km/jam	Hasil: 0.514879561084 - Hasil Prediksi: 100 km/jam
Data ke-60. 100 km/jam	Hasil: 0.48665311755682 - Hasil Prediksi: 100 km/jam
Data ke-61. 100 km/jam	Hasil: 0.52039716026045 - Hasil Prediksi: 100 km/jam
Data ke-62. 100 km/jam	Hasil: 0.49217511547224 - Hasil Prediksi: 100 km/jam
Data ke-63. 100 km/jam	Hasil: 0.52231968816702 - Hasil Prediksi: 100 km/jam
Data ke-64. 100 km/jam	Hasil: 0.49410079512492 - Hasil Prediksi: 100 km/jam
Data ke-65. 100 km/jam	Hasil: 0.47196148080294 - Hasil Prediksi: 100 km/jam
Data ke-66. 100 km/jam	Hasil: 0.44393478744889 - Hasil Prediksi: 100 km/jam
Data ke-67. 100 km/jam	Hasil: 0.47700338049536 - Hasil Prediksi: 100 km/jam
Data ke-68. 100 km/jam	Hasil: 0.44893183252994 - Hasil Prediksi: 100 km/jam
Data ke-69. 100 km/jam	Hasil: 0.48251880292641 - Hasil Prediksi: 100 km/jam
Data ke-70. 100 km/jam	Hasil: 0.45440468552929 - Hasil Prediksi: 100 km/jam
Data ke-71. 100 km/jam	Hasil: 0.48444274299266 - Hasil Prediksi: 100 km/jam
Data ke-72. 100 km/jam	Hasil: 0.45631537559213 - Hasil Prediksi: 100 km/jam
Data ke-73. 100 km/jam	Hasil: 0.4658227012134 - Hasil Prediksi: 100 km/jam
Data ke-74. 100 km/jam	Hasil: 0.43785826557695 - Hasil Prediksi: 100 km/jam
Data ke-75. 100 km/jam	Hasil: 0.47085749930955 - Hasil Prediksi: 100 km/jam
Data ke-76. 100 km/jam	Hasil: 0.44284138353966 - Hasil Prediksi: 100 km/jam
Data ke-77. 100 km/jam	Hasil: 0.47636658584034 - Hasil Prediksi: 100 km/jam
Data ke-78. 100 km/jam	Hasil: 0.44830039022008 - Hasil Prediksi: 100 km/jam
Data ke-79. 100 km/jam	Hasil: 0.47828866802657 - Hasil Prediksi: 100 km/jam
Data ke-80. 100 km/jam	Hasil: 0.45020659273213 - Hasil Prediksi: 100 km/jam
Data ke-81. 100 km/jam	Hasil: 0.47791871879358 - Hasil Prediksi: 100 km/jam

Data ke-82. 100 km/jam	Hasil: 0.44983963582485 - Hasil Prediksi: 100 km/jam
Data ke-83. 100 km/jam	Hasil: 0.48296605071834 - Hasil Prediksi: 100 km/jam
Data ke-84. 100 km/jam	Hasil: 0.45484877931155 - Hasil Prediksi: 100 km/jam
Data ke-85. 100 km/jam	Hasil: 0.48848602252405 - Hasil Prediksi: 100 km/jam
Data ke-86. 100 km/jam	Hasil: 0.4603350909683 - Hasil Prediksi: 100 km/jam
Data ke-87. 100 km/jam	Hasil: 0.49041120722767 - Hasil Prediksi: 100 km/jam
Data ke-88. 100 km/jam	Hasil: 0.46224800751016 - Hasil Prediksi: 100 km/jam
Data ke-89. 100 km/jam	Hasil: 0.44029377844469 - Hasil Prediksi: 100 km/jam
Data ke-90. 100 km/jam	Hasil: 0.41267792605226 - Hasil Prediksi: 100 km/jam
Data ke-91. 100 km/jam	Hasil: 0.44528265842609 - Hasil Prediksi: 100 km/jam
Data ke-92. 100 km/jam	Hasil: 0.41758735298481 - Hasil Prediksi: 100 km/jam
Data ke-93. 100 km/jam	Hasil: 0.45074741276049 - Hasil Prediksi: 100 km/jam
Data ke-94. 100 km/jam	Hasil: 0.42297138259301 - Hasil Prediksi: 100 km/jam
Data ke-95. 100 km/jam	Hasil: 0.45265548321692 - Hasil Prediksi: 100 km/jam
Data ke-96. 100 km/jam	Hasil: 0.42485282258215 - Hasil Prediksi: 100 km/jam
Data ke-97. 100 km/jam	Hasil: 0.44406206069633 - Hasil Prediksi: 100 km/jam
Data ke-98. 100 km/jam	Hasil: 0.41638568709234 - Hasil Prediksi: 100 km/jam
Data ke-99. 100 km/jam	Hasil: 0.44905938146225 - Hasil Prediksi: 100 km/jam
Data ke-100. 100 km/jam	Hasil: 0.42130758214493 - Hasil Prediksi: 100 km/jam
Data ke-101. 100 km/jam	Hasil: 0.45453250685253 - Hasil Prediksi: 100 km/jam
Data ke-102. 100 km/jam	Hasil: 0.42670443536942 - Hasil Prediksi: 100 km/jam
Data ke-103. 100 km/jam	Hasil: 0.456443284739 - Hasil Prediksi: 100 km/jam
Data ke-104. 100 km/jam	Hasil: 0.42859014697878 - Hasil Prediksi: 100 km/jam
Data ke-105. 100 km/jam	Hasil: 0.45607546130741 - Hasil Prediksi: 100 km/jam
Data ke-106. 100 km/jam	Hasil: 0.42822708591274 - Hasil Prediksi: 100 km/jam
Data ke-107. 100 km/jam	Hasil: 0.46109584605765 - Hasil Prediksi: 100 km/jam
Data ke-108. 100 km/jam	Hasil: 0.4331850592542 - Hasil Prediksi: 100 km/jam
Data ke-109. 100 km/jam	Hasil: 0.46659143128891 - Hasil Prediksi: 100 km/jam
Data ke-110. 100 km/jam	Hasil: 0.43861874015126 - Hasil Prediksi: 100 km/jam
Data ke-111. 100 km/jam	Hasil: 0.46850936137248 - Hasil Prediksi: 100 km/jam
Data ke-112. 100 km/jam	Hasil: 0.44051664635524 - Hasil Prediksi: 100 km/jam
Data ke-113. 100 km/jam	Hasil: 0.41879074234117 - Hasil Prediksi: 100 km/jam
Data ke-114. 100 km/jam	Hasil: 0.39158001673903 - Hasil Prediksi: 100 km/jam
Data ke-115. 100 km/jam	Hasil: 0.42372042613972 - Hasil Prediksi: 100 km/jam
Data ke-116. 100 km/jam	Hasil: 0.39640785667795 - Hasil Prediksi: 100 km/jam
Data ke-117. 100 km/jam	Hasil: 0.42912526774947 - Hasil Prediksi: 100 km/jam
Data ke-118. 100 km/jam	Hasil: 0.40170716037037 - Hasil Prediksi: 100 km/jam
Data ke-119. 100 km/jam	Hasil: 0.43101363447791 - Hasil Prediksi: 100 km/jam
Data ke-120. 100 km/jam	Hasil: 0.40356016543769 - Hasil Prediksi: 100 km/jam
Data ke-121. 90 km/jam	Hasil: 0.35127864359636 - Hasil Prediksi: 90 km/jam
Data ke-122. 90 km/jam	Hasil: 0.32599410562949 - Hasil Prediksi: 90 km/jam
Data ke-123. 90 km/jam	Hasil: 0.35590025724676 - Hasil Prediksi: 90 km/jam
Data ke-124. 90 km/jam	Hasil: 0.33045251451261 - Hasil Prediksi: 90 km/jam
Data ke-125. 90 km/jam	Hasil: 0.35909087347697 - Hasil Prediksi: 90 km/jam
Data ke-126. 90 km/jam	Hasil: 0.33353312960587 - Hasil Prediksi: 90 km/jam
Data ke-127. 90 km/jam	Hasil: 0.36045143566986 - Hasil Prediksi: 90 km/jam
Data ke-128. 90 km/jam	Hasil: 0.33484744847476 - Hasil Prediksi: 90 km/jam
Data ke-129. 90 km/jam	Hasil: 0.36241828569901 - Hasil Prediksi: 90 km/jam
Data ke-130. 90 km/jam	Hasil: 0.33674815076192 - Hasil Prediksi: 90 km/jam
Data ke-131. 90 km/jam	Hasil: 0.36710352248754 - Hasil Prediksi: 90 km/jam
Data ke-132. 90 km/jam	Hasil: 0.3412791645564 - Hasil Prediksi: 90 km/jam
Data ke-133. 90 km/jam	Hasil: 0.37033682736353 - Hasil Prediksi: 90 km/jam
Data ke-134. 90 km/jam	Hasil: 0.34440879247627 - Hasil Prediksi: 90 km/jam

Data ke-135. 90 km/jam	Hasil: 0.37171528650504 - Hasil Prediksi: 90 km/jam
Data ke-136. 90 km/jam	Hasil: 0.34574373517059 - Hasil Prediksi: 90 km/jam
Data ke-137. 90 km/jam	Hasil: 0.3281706396941 - Hasil Prediksi: 90 km/jam
Data ke-138. 90 km/jam	Hasil: 0.30377053423317 - Hasil Prediksi: 90 km/jam
Data ke-139. 90 km/jam	Hasil: 0.33264412460063 - Hasil Prediksi: 90 km/jam
Data ke-140. 90 km/jam	Hasil: 0.30806392465727 - Hasil Prediksi: 90 km/jam
Data ke-141. 90 km/jam	Hasil: 0.33573492591171 - Hasil Prediksi: 90 km/jam
Data ke-142. 90 km/jam	Hasil: 0.31103278121358 - Hasil Prediksi: 90 km/jam
Data ke-143. 90 km/jam	Hasil: 0.3370535332421 - Hasil Prediksi: 90 km/jam
Data ke-144. 90 km/jam	Hasil: 0.3122998278243 - Hasil Prediksi: 90 km/jam
Data ke-145. 90 km/jam	Hasil: 0.32275894787046 - Hasil Prediksi: 90 km/jam
Data ke-146. 90 km/jam	Hasil: 0.29858238606748 - Hasil Prediksi: 90 km/jam
Data ke-147. 90 km/jam	Hasil: 0.32719459031169 - Hasil Prediksi: 90 km/jam
Data ke-148. 90 km/jam	Hasil: 0.3028343437492 - Hasil Prediksi: 90 km/jam
Data ke-149. 90 km/jam	Hasil: 0.33025981478732 - Hasil Prediksi: 90 km/jam
Data ke-150. 90 km/jam	Hasil: 0.3057750736758 - Hasil Prediksi: 90 km/jam
Data ke-151. 90 km/jam	Hasil: 0.33156765201318 - Hasil Prediksi: 90 km/jam
Data ke-152. 90 km/jam	Hasil: 0.30703040037284 - Hasil Prediksi: 90 km/jam
Data ke-153. 90 km/jam	Hasil: 0.33345907041706 - Hasil Prediksi: 90 km/jam
Data ke-154. 90 km/jam	Hasil: 0.30884652024961 - Hasil Prediksi: 90 km/jam
Data ke-155. 90 km/jam	Hasil: 0.33796837979223 - Hasil Prediksi: 90 km/jam
Data ke-156. 90 km/jam	Hasil: 0.31317938169084 - Hasil Prediksi: 90 km/jam
Data ke-157. 90 km/jam	Hasil: 0.34108336760663 - Hasil Prediksi: 90 km/jam
Data ke-158. 90 km/jam	Hasil: 0.31617501011153 - Hasil Prediksi: 90 km/jam
Data ke-159. 90 km/jam	Hasil: 0.34241215293279 - Hasil Prediksi: 90 km/jam
Data ke-160. 90 km/jam	Hasil: 0.31745350878705 - Hasil Prediksi: 90 km/jam
Data ke-161. 90 km/jam	Hasil: 0.30065754240614 - Hasil Prediksi: 90 km/jam
Data ke-162. 90 km/jam	Hasil: 0.27745825600001 - Hasil Prediksi: 90 km/jam
Data ke-163. 90 km/jam	Hasil: 0.30492620455724 - Hasil Prediksi: 90 km/jam
Data ke-164. 90 km/jam	Hasil: 0.28153013602218 - Hasil Prediksi: 90 km/jam
Data ke-165. 90 km/jam	Hasil: 0.30787827720192 - Hasil Prediksi: 90 km/jam
Data ke-166. 90 km/jam	Hasil: 0.28434836560776 - Hasil Prediksi: 90 km/jam
Data ke-167. 90 km/jam	Hasil: 0.30913839344237 - Hasil Prediksi: 90 km/jam
Data ke-168. 90 km/jam	Hasil: 0.2855519094608 - Hasil Prediksi: 90 km/jam
Data ke-169. 90 km/jam	Hasil: 0.30387958322395 - Hasil Prediksi: 90 km/jam
Data ke-170. 90 km/jam	Hasil: 0.28053140809758 - Hasil Prediksi: 90 km/jam
Data ke-171. 90 km/jam	Hasil: 0.30817383269714 - Hasil Prediksi: 90 km/jam
Data ke-172. 90 km/jam	Hasil: 0.28463062219642 - Hasil Prediksi: 90 km/jam
Data ke-173. 90 km/jam	Hasil: 0.31114327216222 - Hasil Prediksi: 90 km/jam
Data ke-174. 90 km/jam	Hasil: 0.28746747061274 - Hasil Prediksi: 90 km/jam
Data ke-175. 90 km/jam	Hasil: 0.3124107197666 - Hasil Prediksi: 90 km/jam
Data ke-176. 90 km/jam	Hasil: 0.2886788910816 - Hasil Prediksi: 90 km/jam
Data ke-177. 90 km/jam	Hasil: 0.31424423291993 - Hasil Prediksi: 90 km/jam
Data ke-178. 90 km/jam	Hasil: 0.29043195483906 - Hasil Prediksi: 90 km/jam
Data ke-179. 90 km/jam	Hasil: 0.3186179107987 - Hasil Prediksi: 90 km/jam
Data ke-180. 90 km/jam	Hasil: 0.29461659769935 - Hasil Prediksi: 90 km/jam
Data ke-181. 90 km/jam	Hasil: 0.32164119839346 - Hasil Prediksi: 90 km/jam
Data ke-182. 90 km/jam	Hasil: 0.29751158185074 - Hasil Prediksi: 90 km/jam
Data ke-183. 90 km/jam	Hasil: 0.32293136225246 - Hasil Prediksi: 90 km/jam
Data ke-184. 90 km/jam	Hasil: 0.29874758264934 - Hasil Prediksi: 90 km/jam
Data ke-185. 90 km/jam	Hasil: 0.28253164292635 - Hasil Prediksi: 90 km/jam
Data ke-186. 90 km/jam	Hasil: 0.26021058427003 - Hasil Prediksi: 90 km/jam
Data ke-187. 90 km/jam	Hasil: 0.28664844013663 - Hasil Prediksi: 90 km/jam

Data ke-188. 90 km/jam	Hasil: 0.26412187067047 - Hasil Prediksi: 90 km/jam
Data ke-189. 90 km/jam	Hasil: 0.28949726106679 - Hasil Prediksi: 90 km/jam
Data ke-190. 90 km/jam	Hasil: 0.26683055586271 - Hasil Prediksi: 90 km/jam
Data ke-191. 90 km/jam	Hasil: 0.2907137453599 - Hasil Prediksi: 90 km/jam
Data ke-192. 90 km/jam	Hasil: 0.26798771857672 - Hasil Prediksi: 90 km/jam
Data ke-193. 90 km/jam	Hasil: 0.20742490328692 - Hasil Prediksi: 90 km/jam
Data ke-194. 90 km/jam	Hasil: 0.1894705707458 - Hasil Prediksi: 90 km/jam
Data ke-195. 90 km/jam	Hasil: 0.21076880806428 - Hasil Prediksi: 90 km/jam
Data ke-196. 90 km/jam	Hasil: 0.19259536742686 - Hasil Prediksi: 90 km/jam
Data ke-197. 90 km/jam	Hasil: 0.21308877551534 - Hasil Prediksi: 90 km/jam
Data ke-198. 90 km/jam	Hasil: 0.19476466172513 - Hasil Prediksi: 90 km/jam
Data ke-199. 90 km/jam	Hasil: 0.214080929096 - Hasil Prediksi: 90 km/jam
Data ke-200. 90 km/jam	Hasil: 0.19569271448468 - Hasil Prediksi: 90 km/jam
Data ke-201. 90 km/jam	Hasil: 0.21551823585295 - Hasil Prediksi: 90 km/jam
Data ke-202. 90 km/jam	Hasil: 0.19703751710088 - Hasil Prediksi: 90 km/jam
Data ke-203. 90 km/jam	Hasil: 0.21895656914097 - Hasil Prediksi: 90 km/jam
Data ke-204. 90 km/jam	Hasil: 0.20025627662156 - Hasil Prediksi: 90 km/jam
Data ke-205. 90 km/jam	Hasil: 0.22134138518325 - Hasil Prediksi: 90 km/jam
Data ke-206. 90 km/jam	Hasil: 0.20249021900243 - Hasil Prediksi: 90 km/jam
Data ke-207. 90 km/jam	Hasil: 0.22236110556661 - Hasil Prediksi: 90 km/jam
Data ke-208. 90 km/jam	Hasil: 0.20344578325387 - Hasil Prediksi: 90 km/jam
Data ke-209. 90 km/jam	Hasil: 0.19099388781232 - Hasil Prediksi: 90 km/jam
Data ke-210. 90 km/jam	Hasil: 0.17414929198351 - Hasil Prediksi: 90 km/jam
Data ke-211. 90 km/jam	Hasil: 0.19413778998689 - Hasil Prediksi: 90 km/jam
Data ke-212. 90 km/jam	Hasil: 0.17707660985478 - Hasil Prediksi: 90 km/jam
Data ke-213. 90 km/jam	Hasil: 0.19632023308053 - Hasil Prediksi: 90 km/jam
Data ke-214. 90 km/jam	Hasil: 0.17910988446343 - Hasil Prediksi: 90 km/jam
Data ke-215. 90 km/jam	Hasil: 0.19725388238443 - Hasil Prediksi: 90 km/jam
Data ke-216. 90 km/jam	Hasil: 0.17998001516685 - Hasil Prediksi: 90 km/jam
Data ke-217. 90 km/jam	Hasil: 0.1872139402658 - Hasil Prediksi: 90 km/jam
Data ke-218. 90 km/jam	Hasil: 0.17063240157446 - Hasil Prediksi: 90 km/jam
Data ke-219. 90 km/jam	Hasil: 0.19031025846402 - Hasil Prediksi: 90 km/jam
Data ke-220. 90 km/jam	Hasil: 0.17351302363408 - Hasil Prediksi: 90 km/jam
Data ke-221. 90 km/jam	Hasil: 0.19245994967987 - Hasil Prediksi: 90 km/jam
Data ke-222. 90 km/jam	Hasil: 0.1755141067122 - Hasil Prediksi: 90 km/jam
Data ke-223. 90 km/jam	Hasil: 0.1933796578428 - Hasil Prediksi: 90 km/jam
Data ke-224. 90 km/jam	Hasil: 0.17637052200296 - Hasil Prediksi: 90 km/jam
Data ke-225. 90 km/jam	Hasil: 0.1947124131664 - Hasil Prediksi: 90 km/jam
Data ke-226. 90 km/jam	Hasil: 0.17761186406239 - Hasil Prediksi: 90 km/jam
Data ke-227. 90 km/jam	Hasil: 0.19790255166655 - Hasil Prediksi: 90 km/jam
Data ke-228. 90 km/jam	Hasil: 0.18058465814373 - Hasil Prediksi: 90 km/jam
Data ke-229. 90 km/jam	Hasil: 0.20011680739737 - Hasil Prediksi: 90 km/jam
Data ke-230. 90 km/jam	Hasil: 0.18264927332597 - Hasil Prediksi: 90 km/jam
Data ke-231. 90 km/jam	Hasil: 0.20106399512434 - Hasil Prediksi: 90 km/jam
Data ke-232. 90 km/jam	Hasil: 0.18353275433474 - Hasil Prediksi: 90 km/jam
Data ke-233. 90 km/jam	Hasil: 0.17203640925714 - Hasil Prediksi: 90 km/jam
Data ke-234. 90 km/jam	Hasil: 0.1565402412232 - Hasil Prediksi: 90 km/jam
Data ke-235. 90 km/jam	Hasil: 0.17493573433234 - Hasil Prediksi: 90 km/jam
Data ke-236. 90 km/jam	Hasil: 0.15922863372761 - Hasil Prediksi: 90 km/jam
Data ke-237. 90 km/jam	Hasil: 0.17694971230207 - Hasil Prediksi: 90 km/jam
Data ke-238. 90 km/jam	Hasil: 0.16109708889768 - Hasil Prediksi: 90 km/jam
Data ke-239. 90 km/jam	Hasil: 0.17781162186132 - Hasil Prediksi: 90 km/jam
Data ke-240. 90 km/jam	Hasil: 0.16189697031249 - Hasil Prediksi: 90 km/jam

Data ke-241. 90 km/jam	Hasil: 0.17422345303717 - Hasil Prediksi: 90 km/jam
Data ke-242. 90 km/jam	Hasil: 0.1585680153544 - Hasil Prediksi: 90 km/jam
Data ke-243. 90 km/jam	Hasil: 0.17715175009562 - Hasil Prediksi: 90 km/jam
Data ke-244. 90 km/jam	Hasil: 0.16128457338088 - Hasil Prediksi: 90 km/jam
Data ke-245. 90 km/jam	Hasil: 0.1791856996315 - Hasil Prediksi: 90 km/jam
Data ke-246. 90 km/jam	Hasil: 0.16317247170211 - Hasil Prediksi: 90 km/jam
Data ke-247. 90 km/jam	Hasil: 0.18005611786385 - Hasil Prediksi: 90 km/jam
Data ke-248. 90 km/jam	Hasil: 0.16398064356842 - Hasil Prediksi: 90 km/jam
Data ke-249. 90 km/jam	Hasil: 0.1813176895185 - Hasil Prediksi: 90 km/jam
Data ke-250. 90 km/jam	Hasil: 0.16515226819581 - Hasil Prediksi: 90 km/jam
Data ke-251. 90 km/jam	Hasil: 0.18433860689659 - Hasil Prediksi: 90 km/jam
Data ke-252. 90 km/jam	Hasil: 0.16795911026284 - Hasil Prediksi: 90 km/jam
Data ke-253. 90 km/jam	Hasil: 0.18643637576274 - Hasil Prediksi: 90 km/jam
Data ke-254. 90 km/jam	Hasil: 0.16990930922483 - Hasil Prediksi: 90 km/jam
Data ke-255. 90 km/jam	Hasil: 0.18733397661768 - Hasil Prediksi: 90 km/jam
Data ke-256. 90 km/jam	Hasil: 0.1707440397181 - Hasil Prediksi: 90 km/jam
Data ke-257. 90 km/jam	Hasil: 0.15989189196179 - Hasil Prediksi: 90 km/jam
Data ke-258. 90 km/jam	Hasil: 0.14529767128066 - Hasil Prediksi: 90 km/jam
Data ke-259. 90 km/jam	Hasil: 0.16262674686964 - Hasil Prediksi: 90 km/jam
Data ke-260. 90 km/jam	Hasil: 0.14782682529804 - Hasil Prediksi: 90 km/jam
Data ke-261. 90 km/jam	Hasil: 0.16452727395675 - Hasil Prediksi: 90 km/jam
Data ke-262. 90 km/jam	Hasil: 0.1495852906912 - Hasil Prediksi: 90 km/jam
Data ke-263. 90 km/jam	Hasil: 0.16534083018446 - Hasil Prediksi: 90 km/jam
Data ke-264. 90 km/jam	Hasil: 0.15033825684837 - Hasil Prediksi: 90 km/jam
Data ke-265. 80 km/jam	Hasil: 0.097635147596581 - Hasil Prediksi: 80 km/jam
Data ke-266. 80 km/jam	Hasil: 0.088127349344797 - Hasil Prediksi: 80 km/jam
Data ke-267. 80 km/jam	Hasil: 0.09943117046702 - Hasil Prediksi: 80 km/jam
Data ke-268. 80 km/jam	Hasil: 0.089765872513712 - Hasil Prediksi: 80 km/jam
Data ke-269. 80 km/jam	Hasil: 0.10003359178696 - Hasil Prediksi: 80 km/jam
Data ke-270. 80 km/jam	Hasil: 0.090315607632742 - Hasil Prediksi: 80 km/jam
Data ke-271. 80 km/jam	Hasil: 0.10047582918561 - Hasil Prediksi: 80 km/jam
Data ke-272. 80 km/jam	Hasil: 0.090719213609467 - Hasil Prediksi: 80 km/jam
Data ke-273. 80 km/jam	Hasil: 0.090170818803325 - Hasil Prediksi: 80 km/jam
Data ke-274. 80 km/jam	Hasil: 0.081324394248862 - Hasil Prediksi: 80 km/jam
Data ke-275. 80 km/jam	Hasil: 0.091843508836808 - Hasil Prediksi: 80 km/jam
Data ke-276. 80 km/jam	Hasil: 0.082847924054289 - Hasil Prediksi: 80 km/jam
Data ke-277. 80 km/jam	Hasil: 0.092404675934014 - Hasil Prediksi: 80 km/jam
Data ke-278. 80 km/jam	Hasil: 0.08335917286357 - Hasil Prediksi: 80 km/jam
Data ke-279. 80 km/jam	Hasil: 0.092816665083472 - Hasil Prediksi: 80 km/jam
Data ke-280. 80 km/jam	Hasil: 0.08373455315941 - Hasil Prediksi: 80 km/jam

Keterangan:

Jumlah data prediksi yang salah: 0

Nilai akurasi: 100%

Epo berakhir pada: 1806 pengulangan

**LAMPIRAN G. CONTOH PENGHITUNGAN MANUAL METODE
SINGLE LAYER PERCEPTRON**

1. Proses Klasifikasi *Training-Testing*

Proses klasifikasi *training-testing* merupakan gabungan dari dua proses klasifikasi, yaitu *training* dan *testing* yang bertujuan untuk menemukan model klasifikasi yang digunakan untuk klasifikasi data baru. Model klasifikasi yang digunakan adalah penghitungan menggunakan bobot dan bias. Untuk memulai tahap *training*, diperlukan contoh data *training* sebagaimana pada tabel E.1

Untuk atribut-atribut yang bertipe nominal seperti tipe rel, jenis bantalan, dan jenis penambat, diperlukan konversi menjadi atribut-atribut baru sesuai dengan nilai-nilai dari atribut-atribut sebelumnya yang dilanjutkan dengan proses normalisasi *min-max*. Contohnya untuk atribut tipe rel, dikonversi menjadi tiga atribut, yaitu R.60, R.54, dan R.42. Selain itu, jika terdapat atribut yang seluruh nilai data-nya bernilai 0 atau kosong, maka atribut tersebut tidak digunakan untuk proses klasifikasi ini. Untuk kelas, nilai-nilai kelas tersebut dikonversi sesuai dengan jumlah kelas. Dalam data *training* tersebut, terdapat lima kelas antara lain: 120 km/jam, 110 km/jam, 100 km/jam, 90 km/jam, dan 80 km/jam. Sebagaimana nilai-nilai dari atribut-atribut, nilai-nilai kelas juga diproses normalisasi *min-max*, sehingga menghasilkan nilai 1 untuk 120 km/jam, 0,75 untuk 110 km/jam, 0,5 untuk 100 km/jam, 0,25 untuk 90 km/jam, dan 0 untuk 80 km/jam. Untuk data *training* yang telah dikonversi dipaparkan pada tabel E.2

Tabel E.1 Contoh Data *Training*

No	Tipe Rel	Jenis Bantalan	Jenis Penambat	Tebal Balas Atas (cm)	Lebar Bahu Balas (cm)	Landai Penentu Maksimum	Jari-jari minimal vertikal (m)	Jari-jari minimal horizontal tanpa peralihan (m)	Jari-jari minimal horizontal peralihan (m)	Lebar Badan Jalan (m)	Kecepatan Maksimum (km/jam)
1	R.60	Beton	Ganda	30	60	0.01	0	0	0	3.15	120
17	R.54	Beton	Ganda	30	50	0.01	0	0	0	3.15	110
95	R.42	Kayu	Ganda	30	40	0.02	0	0	550	2.95	100
175	R.54	Baja	Ganda	25	40	0.025	0	0	440	2.85	90
274	R.42	Baja	Tunggal	25	35	0.025	0	0	0	2.4	80

Tabel E.2 Contoh Data *Training* Hasil Konversi

No	R.60	R.54	R.42	Beton	Kayu	Baja	Ganda	Tunggal	TBA	LBB	LPM	JMPH	LBJ	Kecepatan Maksimum
1	1	0	0	1	0	0	1	0	1	1	0	0	1	1
2	0	1	0	1	0	0	1	0	1	0,6	0	0	1	0,75
3	0	0	1	0	1	0	1	0	1	0,2	0,667	1	0,733	0,5
4	0	1	0	0	0	1	1	0	0	0,2	1	0,8	0,6	0,25
5	0	0	1	0	0	1	0	1	0	0	1	0	0	0

Langkah selanjutnya adalah penghitungan pada setiap epoch di mana nilai epoch ditentukan berdasarkan pengulangan yang nilai batasannya adalah nilai epoch itu sendiri. Untuk dapat mendapatkan hasil klasifikasi yang optimal maka nilai epoch yang ditentukan adalah lebih besar daripada banyaknya data *training*. Pada kasus ini, ditentukan nilai epoch adalah 10. Apabila dengan nilai epoch 10 tidak menghasilkan nilai akurasi 100%, maka dilakukan kembali proses klasifikasi ini dengan nilai epoch yang lebih besar dari 10, sebagai contoh nilai epoch yang ditentukan adalah 100. Apabila dengan nilai epoch 10 tidak menghasilkan nilai akurasi 100%, maka dilakukan kembali proses klasifikasi ini dengan nilai epoch yang lebih besar dari 100, sebagai contoh nilai epoch yang ditentukan adalah 500. Hal itu dilakukan seterusnya hingga didapatkan nilai akurasi 100%. Selain epoch, parameter lain yang digunakan pada proses klasifikasi ini adalah *learning rate* dengan nilai 0,9.

Dari hasil penghitungan yang telah dilakukan, proses klasifikasi berhenti pada epoch ke-10. Berikut tabel E.3 yang merupakan contoh penghitungan manual proses *training* yang menggunakan data *training* pada tabel E.2.

Tabel E.3 Contoh Penghitungan Manual Proses *Training*

Epoth 1														
Keterangan	R.60	R.54	R.42	Beton	Kayu	Baja	Ganda	Tunggal	TBA	LBB	LPM	JMPH	LBJ	b
w awal	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Data 1	1	0	0	1	0	0	1	0	1	1	0	0	1	-
w*x	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$y_{in} = \sum_{q=1}^n [x \cdot w_q + b] = 0$ Fungsi Sigmoid ($y = 1 / (1 + e^{-y_{in}}) = 0,5$) Nilai Galat / Error ($t = c - y$) = 0,5 Hasil bobot = $w^{lama} + \alpha \cdot t^{(q)} \cdot x$ Nilai bias = $b^{lama} + \alpha \cdot t^{(q)}$														
Nilai	0,45	0	0	0,45	0	0	0,45	0	0,45	0,45	0	0	0,45	0,45
Epoth 2														
Ket	R.60	R.54	R.42	Beton	Kayu	Baja	Ganda	Tunggal	TBA	LBB	LPM	JMPH	LBJ	b
w awal	0,45	0	0	0,45	0	0	0,45	0	0,45	0,45	0	0	0,45	0,45
Data 2	0	1	0	1	0	0	1	0	1	0,6	0	0	1	-
w*x	0	0	0	0,45	0	0	0,45	0	0,45	0,27	0	0	0,45	0,45
$y_{in} = \sum_{q=1}^n [x \cdot w_q + b] = 2,52$ Fungsi Sigmoid ($y = 1 / (1 + e^{-y_{in}}) = 0,925532055$) Nilai Galat / Error ($t = c - y$) = -0,175532055 Nilai bobot = $w^{lama} + \alpha \cdot t^{(q)} \cdot x$ Nilai bias = $b^{lama} + \alpha \cdot t^{(q)}$														
Nilai	0,45	-0,16	0	0,29	0	0	0,29	0	0,29	0,36	0	0	0,29	0,29
Epoth 3														
Ket	R.60	R.54	R.42	Beton	Kayu	Baja	Ganda	Tunggal	TBA	LBB	LPM	JMPH	LBJ	B
w awal	0,45	-0,16	0	0,29	0	0	0,29	0	0,29	0,36	0	0	0,29	0,29
Data 3	0	0	1	0	1	0	1	0	1	0,2	0,67	1	0,73	-

w*x	0	0	0	0	0	0	0,29	0	0,29	0,07	0	0	0,21	0,29
$y_{in} = \sum_{q=1}^n [x \cdot w_q + b] = 1,161254834$ Fungsi Sigmoid ($y = 1 / (1 + e^{-y_{in}}) = 0,76156065$) Nilai Galat / Error ($t = c - y = -0,26156065$) Hasil bobot = $w^{lama} + \alpha \cdot t^{(q)} \cdot x$ Hasil bias = $b^{lama} + \alpha \cdot t^{(q)}$														
Hasil	0,45	-0,16	-0,24	0,29	-0,24	0	0,06	0	0,06	0,31	-0,16	-0,24	0,12	0,06
Epo 4														
Ket	R.60	R.54	R.42	Beton	Kayu	Baja	Ganda	Tunggal	TBA	LBB	LPM	JMPH	LBJ	b
w awal	0,45	-0,16	-0,24	0,29	-0,24	0	0,06	0	0,06	0,31	-0,16	-0,24	0,12	0,06
Data 4	0	1	0	0	0	1	1	0	0	0,2	1	0,8	0,6	-
w*x	0	-0,16	0	0	0	0	0,06	0	0	0,06	-0,16	-0,19	-0,07	0,06
$y_{in} = \sum_{q=1}^n [x \cdot w_q + b] = -0,256744748$ Fungsi Sigmoid ($y = 1 / (1 + e^{-y_{in}}) = 0,43616409$) Nilai Galat / Error ($t = c - y = -0,186164089$) Nilai bobot = $w^{lama} + \alpha \cdot t^{(q)} \cdot x$ Nilai bias = $b^{lama} + \alpha \cdot t^{(q)}$														
Hasil	0,45	-0,33	-0,24	0,29	-0,24	-0,2	-0,1	0	0,06	0,27	-0,32	-0,37	0,02	-0,11
Epo 5														
Ket	R.60	R.54	R.42	Beton	Kayu	Baja	Ganda	Tunggal	TBA	LBB	LPM	JMPH	LBJ	b
w awal	0,45	-0,33	-0,24	0,29	-0,24	-0,2	-0,1	0	0,06	0,27	-0,32	-0,37	0,02	-0,11
Data 5	0	0	1	0	0	1	0	1	0	0	1	0	0	-
w*x	0	0	-0,24	0	0	-0,2	0	0	0	0	-0,32	0	0	-0,11
$y_{in} = \sum_{q=1}^n [x \cdot w_q + b] = -0,838367449$ Fungsi Sigmoid ($y = 1 / (1 + e^{-y_{in}}) = 0,301878729$) Nilai Galat / Error ($t = c - y = -0,301878729$) Nilai bobot = $w^{lama} + \alpha \cdot t^{(q)} \cdot x$ Nilai bias = $b^{lama} + \alpha \cdot t^{(q)}$														

Nilai	0,45	-0,33	-0,51	0,29	-0,24	-0,4	-0,1	-0,27	0,06	0,27	-0,6	-0,37	0,02	-0,38
Epoth 6														
Ket	R.60	R.54	R.42	Beton	Kayu	Baja	Ganda	Tunggal	TBA	LBB	LPM	JMPH	LBJ	b
w awal	0,45	-0,33	-0,51	0,29	-0,24	-0,4	-0,1	-0,27	0,06	0,27	-0,6	-0,37	0,02	-0,38
Data 1	1	0	0	1	0	0	1	0	1	1	0	0	1	-
w*x	0,45	0	0	0,29	0	0	-0,1	0	0,06	0,27	0	0	0,02	-0,38
$y_{in} = \sum q \cdot [x \cdot w_q + b] = 0,598569382$ Fungsi Sigmoid ($y = 1/1 + e^{-y_{in}}$) = 0,645328935 Nilai Galat / Error ($t = c - y$) = 0,354671065 Nilai bobot = $w^{lama} + \alpha \cdot t^{(q)} \cdot x$ Nilai bias = $b^{lama} + \alpha \cdot t^{(q)}$														
Nilai	0,77	-0,33	-0,51	0,61	-0,24	-0,4	0,21	-0,27	0,38	0,59	-0,6	-0,37	0,34	-0,06
Epoth 7														
Ket	R.60	R.54	R.42	Beton	Kayu	Baja	Ganda	Tunggal	TBA	LBB	LPM	JMPH	LBJ	B
w awal	0,77	-0,33	-0,51	0,61	-0,24	-0,4	0,21	-0,27	0,38	0,59	-0,6	-0,37	0,34	-0,06
Data 2	0	1	0	1	0	0	1	0	1	0,6	0	0	1	-
w*x	0	-0,33	0	0,61	0	0	0,21	0	0,38	0,36	0	0	0,34	-0,06
$y_{in} = \sum q \cdot [x \cdot w_q + b] = 1,500736124$ Fungsi Sigmoid ($y = 1/1 + e^{-y_{in}}$) = 0,817684241 Nilai Galat / Error ($t = c - y$) = -0,067684241 Hasil bobot = $w^{lama} + \alpha \cdot t^{(q)} \cdot x$ Hasil bias = $b^{lama} + \alpha \cdot t^{(q)}$														
Hasil	0,77	-0,39	-0,51	0,55	-0,24	-0,4	0,15	-0,27	0,31	0,56	-0,6	-0,37	0,28	-0,12
Epoth 8														
Ket	R.60	R.54	R.42	Beton	Kayu	Baja	Ganda	Tunggal	TBA	LBB	LPM	JMPH	LBJ	B
w awal	0,77	-0,39	-0,51	0,55	-0,24	-0,4	0,15	-0,27	0,31	0,56	-0,6	-0,37	0,28	-0,12
Data 3	0	0	1	0	1	0	1	0	1	0,2	0,67	1	0,73	-
w*x	0	0	-0,51	0	-0,24	0	0,15	0	0,31	0,11	-0,4	-0,37	0,2	-0,12

$y_{in} = \sum_{q=1}^n [x \cdot w_q + b] = -0,856765646$ Fungsi Sigmoid ($y = 1/1 + e^{-y_{in}}$) = 0,298015537 Nilai Galat / Error ($t = c-y$) = 0,201984463 Nilai bobot = $w^{lama} + \alpha \cdot t^{(q)} \cdot x$ Nilai bias = $b^{lama} + \alpha \cdot t^{(q)}$														
Nilai	0,77	-0,39	-0,33	0,55	-0,05	-0,4	0,33	-0,27	0,5	0,59	-0,47	-0,19	0,41	-0,06
Epo 9														
Ket	R.60	R.54	R.42	Beton	Kayu	Baja	Ganda	Tunggal	TBA	LBB	LPM	JMPH	LBJ	B
w awal	0,77	-0,39	-0,33	0,55	-0,05	-0,4	0,33	-0,27	0,5	0,59	-0,47	-0,19	0,41	-0,06
Data 4	0	1	0	0	0	1	1	0	0	0,2	1	0,8	0,6	-
w*x	0	-0,39	0	0	0	-0,4	0,33	0	0	0,12	-0,47	-0,15	0,25	-0,06
$y_{in} = \sum_{q=1}^n [x \cdot w_q + b] = -0,699192249$ Fungsi Sigmoid ($y = 1/1 + e^{-y_{in}}$) = 0,331991341 Nilai Galat / Error ($t = c-y$) = -0,081991341 Nilai bobot = $w^{lama} + \alpha \cdot t^{(q)} \cdot x$ Nilai bias = $b^{lama} + \alpha \cdot t^{(q)}$														
Nilai	0,77	-0,46	-0,33	0,55	-0,05	-0,5	0,26	-0,27	0,5	0,58	-0,55	-0,25	0,37	-0,02
Epo 10														
Ket	R.60	R.54	R.42	Beton	Kayu	Baja	Ganda	Tunggal	TBA	LBB	LPM	JMPH	LBJ	B
w awal	0,77	-0,46	-0,33	0,55	-0,05	-0,5	0,26	-0,27	0,5	0,58	-0,55	-0,25	0,37	-0,02
Data 5	0	0	1	0	0	1	0	1	0	0	1	0	0	-
w*x	0	0	-0,33	0	0	-0,5	0	-0,27	0	0	-0,55	0	0	0
$y_{in} = \sum_{q=1}^n [x \cdot w_q + b] = -1,6751475$ Fungsi Sigmoid ($y = 1/1 + e^{-y_{in}}$) = 0,157739089 Nilai Galat / Error ($t = c-y$) = -0,157739089 Nilai bobot = $w^{lama} + \alpha \cdot t^{(q)} \cdot x$ Nilai bias = $b^{lama} + \alpha \cdot t^{(q)}$														
Nilai	0,77	-0,46	-0,33	0,55	-0,05	-0,5	0,26	-0,27	0,5	0,58	-0,55	-0,25	0,37	-0,16

Setelah dilakukan proses *training*, maka dilanjutkan ke proses *testing*. Proses *testing* dilakukan pada setiap epoch sehingga mendapatkan nilai akurasi yang menjadi acuan untuk menetapkan nilai bobot dan bias untuk proses klasifikasi data baru. Nilai akurasi merupakan hasil persentase antara banyaknya data yang benar dengan jumlah data *training*. Untuk mengetahui nilai akurasi, terdapat contoh nilai akurasi pada epoch ke-10 yang dipaparkan pada tabel E.4.

Nilai y_{in} merupakan nilai yang didapat dari hasil penjumlahan antar setiap bobot akhir yang dikalikan dengan setiap nilai data *training* yang dilanjutkan penjumlahan dengan bias akhir. Nilai y merupakan nilai yang didapat dari fungsi aktivasi *unipolar sigmoidal* pada nilai y_{in} . Hasil merupakan nilai konversi dari nilai y dengan ketentuan:

1. Kecepatan 120 km/jam untuk $1 > y > 0,875$
2. Kecepatan 110 km/jam untuk $0,875 > y > 0,625$
3. Kecepatan 100 km/jam untuk $0,625 > y > 0,375$
4. Kecepatan 90 km/jam untuk $0,375 > y > 0,125$
5. Kecepatan 80 km/jam untuk $0,125 > y > 0$

Dari hasil proses klasifikasi *training-testing* tersebut, didapatkan bahwa hasil prediksi data pertama adalah 120 km/jam, data kedua adalah 110 km/jam, data ketiga adalah 100 km/jam, data keempat adalah 90 km/jam, dan data kelima adalah 80 km/jam yang jika dibandingkan dengan nilai kecepatan maksimum pada data *training* adalah sama secara keseluruhan sebagaimana yang dipaparkan pada tabel E.4. Dengan demikian, nilai akurasi yang didapat adalah 100%.

Tabel E.4 Contoh Hasil Klasifikasi *Training-Testing* pada Epoch ke-10

No	R.60	R.54	R.42	Beton	Kayu	Baja	Ganda	Tunggal	TBA	LBB	LPM	JMPH	LBJ	y_in	y	Hasil
1	1	0	0	1	0	0	1	0	1	1	0	0	1	2,86	0,95	120 km/jam
2	0	1	0	1	0	0	1	0	1	0,6	0	0	1	1,4	0,8	110 km/jam
3	0	0	1	0	1	0	1	0	1	0,2	0,67	1	0,73	-0,25	0,44	100 km/jam
4	0	1	0	0	0	1	1	0	0	0,2	1	0,8	0,6	-1,57	0,17	90 km/jam
5	0	0	1	0	0	1	0	1	0	0	1	0	0	-2,39	0,08	80 jm/jam

2. Proses Klasifikasi Data Baru

Proses klasifikasi data baru merupakan proses klasifikasi untuk memprediksi nilai kelas dari data baru berdasarkan model yang telah ditentukan pada proses klasifikasi *training-testing*. Data baru tersebut merupakan data dengan atribut-atribut yang sama dengan *dataset* atau data *training* pada kasus ini, tetapi memiliki perbedaan pada nilai data jika dibandingkan dengan data pada data *training* sebagaimana yang dipaparkan pada tabel E.5. Selain itu, data baru tidak memiliki nilai kelas karena nilai kelas merupakan hasil dari klasifikasi ini. Nilai bobot dan bias yang digunakan pada klasifikasi ini didapatkan dari proses *training-testing* sebelumnya.

Data baru yang ditentukan tersebut selanjutnya dilakukan proses normalisasi *min-max* yang mengacu nilai minimal dan maksimal setiap atribut pada data *training* sebagaimana yang dipaparkan pada tabel E.6. Setelah data baru tersebut telah di-normalisasi, maka proses berikutnya adalah penghitungan dari klasifikasi ini yang memiliki proses yang sama dengan proses *testing* sebagaimana yang dipaparkan pada proses klasifikasi *training-testing*. Proses penghitungan klasifikasi data baru ini bermula pada proses mendapatkan nilai penjumlahan bobot dilakukan sebagaimana pada tabel E.7, lalu ditentukan nilai bias untuk dilakukan penjumlahan dengan nilai penjumlahan bobot sehingga didapatkan nilai y_{in} . Proses selanjutnya adalah penghitungan nilai y yang menggunakan fungsi sigmoid. Setelah nilai y didapatkan, maka nilai y dikonversi menjadi nilai hasil sebagaimana yang dilakukan pada proses *testing* sebelumnya.

Tabel E.5 Contoh Data Baru untuk Dilakukan Klasifikasi

R.60	R.54	R.42	Beton	Kayu	Baja	Ganda	Tunggal	TBA	LBB	LPM	JMPH	LBJ
0	1	0	0	0	1	1	0	25	40	0,025	0	2,75

Tabel E.6 Contoh Data Baru Hasil Normalisasi *Min-Max*

R.60	R.54	R.42	Beton	Kayu	Baja	Ganda	Tunggal	TBA	LBB	LPM	JMPH	LBJ
0	1	0	0	0	1	1	0	0	0,2	1	0	0,47

Tabel E.7 Contoh Penghitungan Klasifikasi Data Baru Hingga Nilai Bobot

	R.60	R.54	R.42	Beton	Kayu	Baja	Ganda	Tunggal	TBA	LBB	LPM	JMPH	LBJ
Data	0	1	0	0	0	1	1	0	0	0,2	1	0	0,47
Bobot	0,77	-0,46	-0,33	0,55	-0,05	-0,5	0,26	-0,27	0,5	0,58	-0,55	-0,25	0,37
w * x	0	-0,46	0	0	0	-0,7	0,26	0	0	0,12	-0,69	0	0,17

Nilai bias = 0,16

$$y_{in} = \sum_q x \cdot w_q + b = -1,422265014$$

$$y = f(y_{in}) = \frac{1}{1 + e^{-y_{in}}} = 0,194306746$$

Dari nilai $y = 0,194306746$ tersebut, selanjutnya dilakukan konversi hasil kecepatan maksimum sehingga didapat nilai kecepatan maksimum data baru yang didapat adalah 90 km/jam.

